Anaconda Documentation

Release 2.0

Anaconda Inc.

Sep 06, 2018

Contents

Anaconda Distribution 5 is a free, easy-to-install package manager, environment manager and Python distribution with a collection of 1,000+ open source packages with free community support. Anaconda is platform-agnostic, so you can use it whether you are on Windows, macOS or Linux.

Go to *Anaconda Distribution 5 documentation*.

CHAPTER 1

Anaconda Enterprise 5

Anaconda Enterprise is an enterprise-ready, secure and scalable data science platform that empowers teams to govern data science assets, collaborate and deploy data science projects.

Enterprise 5 includes these capabilities:

- Easily deploy your projects into interactive data applications, live notebooks and machine learning models with APIs.
- Share those applications with colleagues and collaborators.
- Manage your data science assets: notebooks, packages, environments and projects in an integrated data science experience.

Go to Anaconda Enterprise 5 documentation.

1.1 Anaconda Enterprise 4

Anaconda Enterprise 4 includes *Anaconda Distribution*, Anaconda Enterprise 4 Repository and Anaconda Enterprise 4 Notebooks. This is our previous-generation product, and documentation is provided for our current customers.

Anaconda Enterprise 4 Repository is an enterprise server on your network or your private cloud where open source and proprietary packages may be stored, retrieved and shared.

Anaconda Enterprise 4 Notebooks is a browser-based Python data analysis environment and visualization tool in a secure, governed environment.

Please refer to Anaconda Enterprise 5 documentation for our current-generation product.

1.1.1 Anaconda Enterprise 4 Repository

Open Data Science Hub

Anaconda Repository is an enterprise server on your network or your private cloud where open source and proprietary packages may be stored, retrieved and shared. It is used to govern access to data science assets including packages, environments and notebooks.

Anaconda Repository has a self-contained internal mirror of packages and can install securely behind an enterprise firewall or in an air gapped environment. It supports many repositories including PyPI, conda and the Anaconda Repository.

Many enterprises have customized local instances of Repository. Anaconda also makes an instance of Repository available for public use at Anaconda Cloud.

User guide

Getting started

Finding, downloading and installing packages

You do not need to have an Anaconda Repository account or be logged in to search for, download, or install packages, notebooks, environments or installers.

You do need an account to access *private packages* without an *access token* and to upload and share your own packages, notebooks, environments and installers with others.

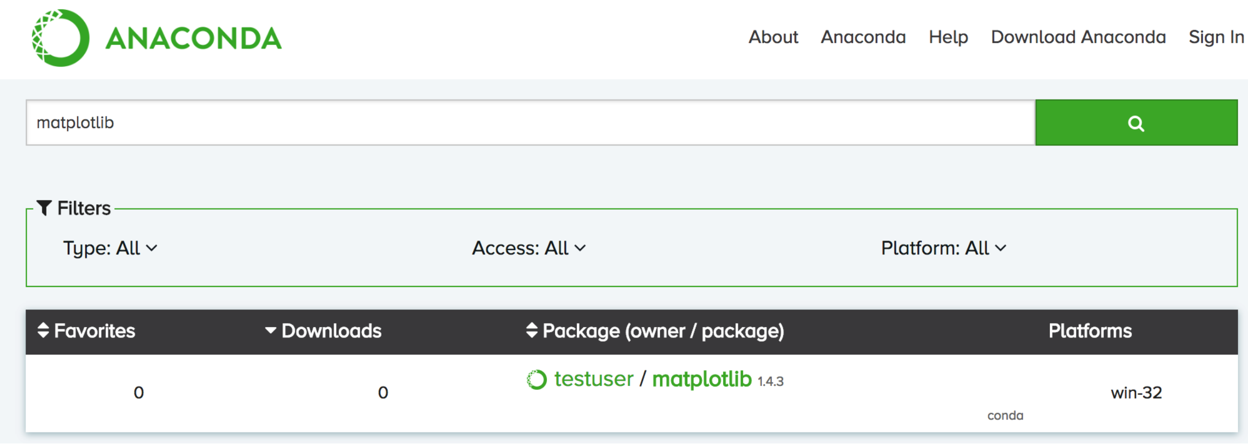
Searching for packages

1. In the top Search box, type part or all of the name of a file you are searching for, and then press Enter.
2. Packages that match your search string are displayed. To see more information, click the package name.

Refining your search results

You can filter search results using 3 filter controls:

- Type: All, conda only or PyPI only.
- Access: All, Public and/or Private—available only if you are logged in.
- Platform: All, Source, Linux-32, Linux-64, Noarch, OSX-64, Win-32 and Win-64.



Favorites	Downloads	Package (owner / package)	Platforms
0	0	testuser / matplotlib 1.4.3 conda	win-32

NOTE: Source packages are source code only, not yet built for any specific platform. Noarch packages are built to work on all platforms.

Downloading and installing packages from Anaconda Repository

You can download and install packages using *Anaconda Navigator*, the graphical user interface for Anaconda®. Advanced users may prefer a Terminal window or an Anaconda Prompt.

Using Navigator

Navigator is automatically installed when you install Anaconda.

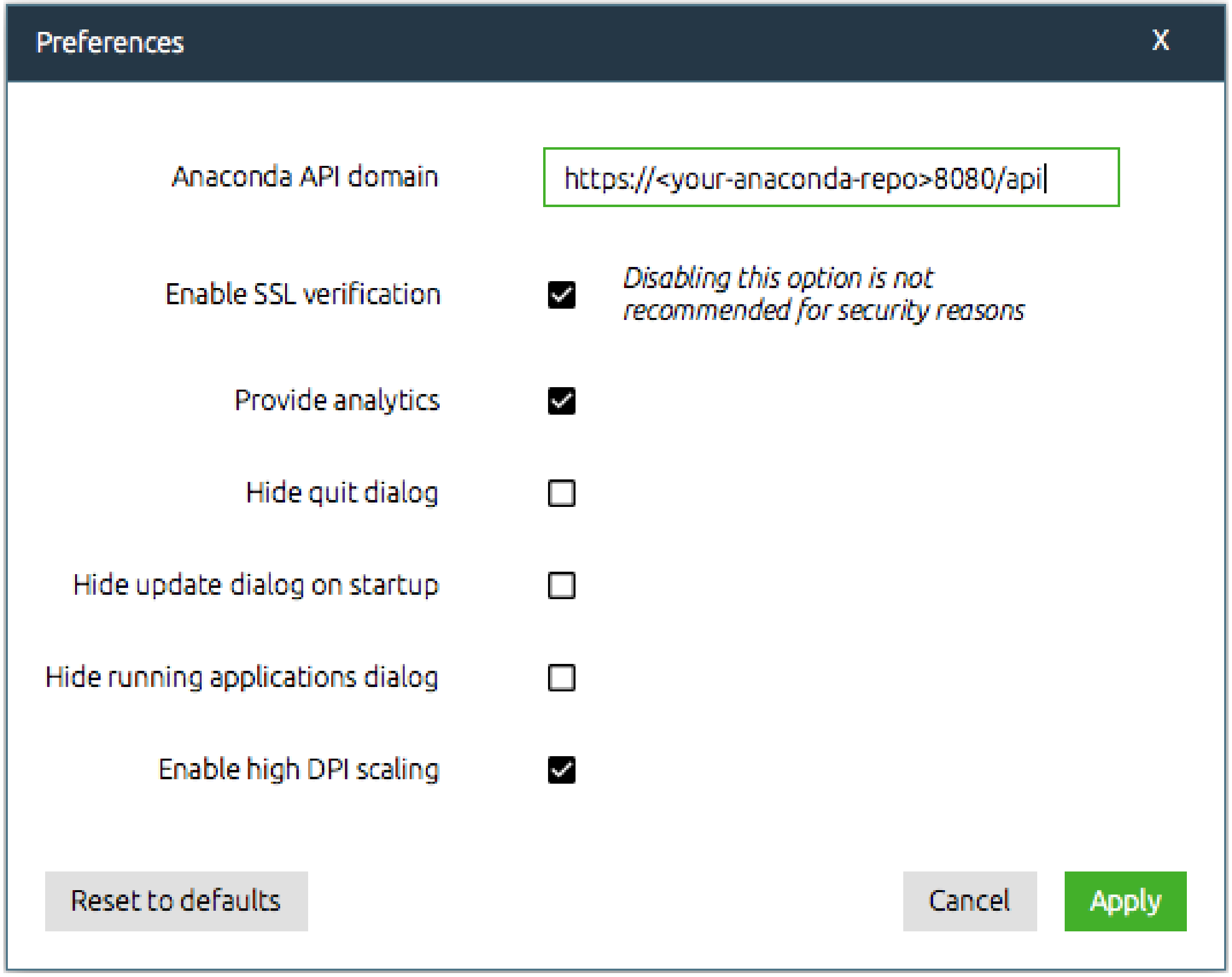
To download and install a package into its own environment:

1. Start Navigator by clicking its program icon on your desktop or in your programs menu.
2. Set up Navigator to search your local Repository:
   (a) From the top menu bar, select Preferences.
   (b) In the Anaconda API domain box, type the address of your local Repository:

   NOTE: If your organization does not use HTTPS, use `http` in the domain box and clear the Enable SSL verification checkbox.

   NOTE: If your organization uses subdomains, enter the address as `https://api.<your-anaconda-repo>:8080`. Replace `<your-anaconda-repo>` with the name of your local Repository. If that does not work, contact your system administrator.

   (c) Click the Apply button.
3. Sign Navigator into your local Anaconda Cloud—Repository— so you can search for packages marked as private:
   (a) Click the top right Sign in to Anaconda Cloud button.
   (b) Type your Repository username and password:



Preferences

Anaconda API domain https://<your-anaconda-repo>8080/api

Enable SSL verification ☑ *Disabling this option is not recommended for security reasons*

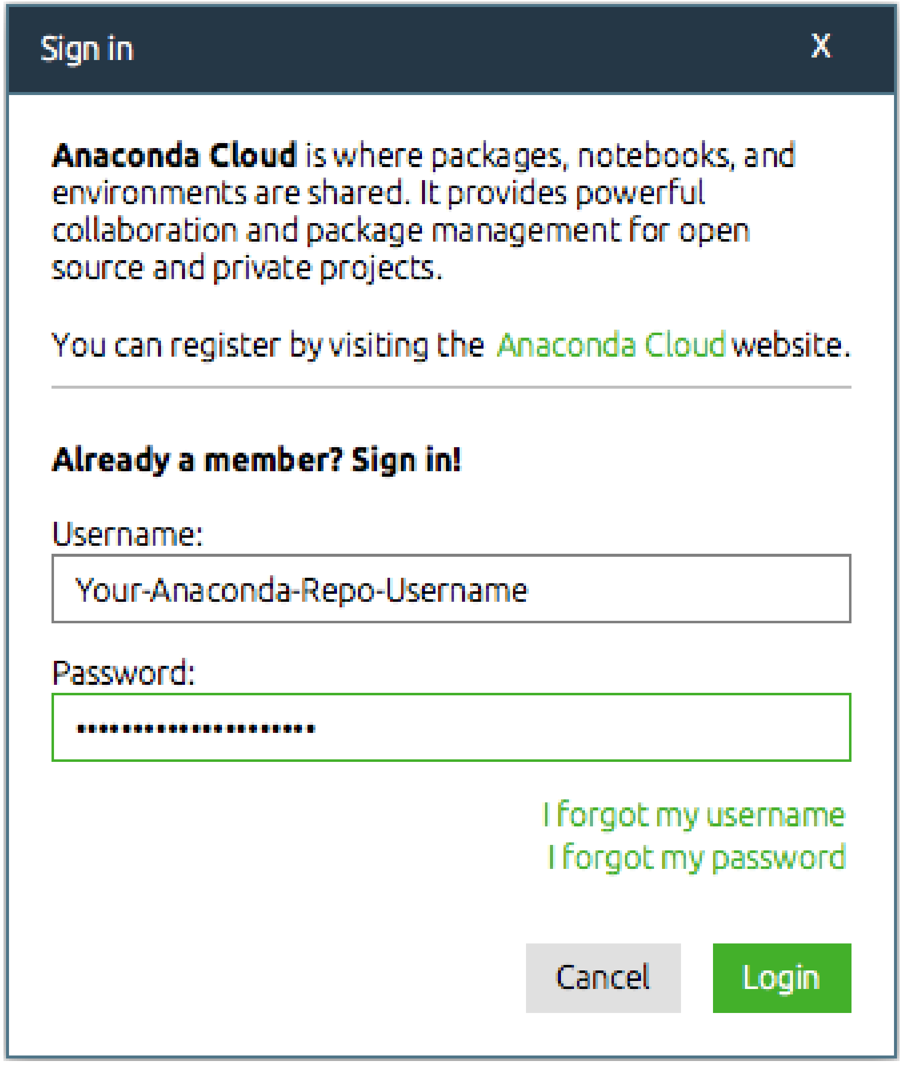
Provide analytics ☑

Hide quit dialog □

Hide update dialog on startup □

Hide running applications dialog □

Enable high DPI scaling ☑



Anaconda Cloud is where packages, notebooks, and environments are shared. It provides powerful collaboration and package management for open source and private projects.

You can register by visiting the Anaconda Cloud website.

Already a member? Sign in!

Username: Your-Anaconda-Repo-Username

Password: ••••••••••••••••••••

   (c) Click the Login button.

4. On the **Environments** tab, in the far-right Search packages box, type the name of the desired package.
5. In the list to the left of Channels, select either Not installed or All, then click the Search button.
6. Select the checkbox of the package you want to install, then click the Apply button.

For more information, see the full Navigator documentation *Anaconda Navigator*.

Using conda in a Terminal window or an Anaconda Prompt

Conda is automatically installed when you install Anaconda.

To download and install a package into its own environment:

1. Locate a package on Anaconda Repository that you want to download, then click the package name.

   A detail page displays specific installation instructions for the current operating system.

2. Enter the command into your Terminal window or Anaconda Prompt.

   EXAMPLE: To download and install a package with conda:

   ```
   conda install -c USERNAME PACKAGE
   ```

   TIP: Conda expands `USERNAME` to a URL such as `https://<your-anaconda-repo>/USERNAME`, based on the settings in the `.condarc` file.

Building and uploading new packages

Building and uploading new packages is optional, and best suited for advanced users who are comfortable using a Terminal application. It requires the anaconda-client, which is easy to get if you have installed Anaconda.

Use Terminal window or Anaconda Prompt to run the following command line commands.

1. To build and upload packages, first install the Anaconda Client CLI:

   ```
   conda install anaconda-client
   ```

2. Log into your Repository account:

   ```
   anaconda login
   ```

   At the prompt, enter your Repository username and password.

3. Choose the package you would like to build. For this example, you can download our public test package:

   ```
   git clone https://github.com/anaconda-platform/anaconda-client
   cd anaconda-client/example-packages/conda/
   ```

4. To build your test package, first install conda-build and turn off automatic Client uploading, then run the `conda build` command:

```
conda install conda-build
conda config --set anaconda_upload no
conda build .
```

5. Find the path where the newly-built package was placed, so that you can use it in the next step:

```
conda build . --output
```

6. Upload your test package to your Repository account:

```
anaconda login
anaconda upload /your/path/conda-package.tar.bz2
```

NOTE: Replace `/your/path/` with the path you found in the previous step.

For more information, see *Working with conda packages*.

Sharing notebooks

To upload a notebook to Anaconda Repository with anaconda-client, open Anaconda Prompt or Terminal and then enter:

```
anaconda upload my-notebook.ipynb
```

NOTE: Replace `my-notebook` with the name of your notebook.

Viewing notebooks

You can view an HTML version of your notebook in Anaconda Repository. Log into your account, then from the drop-down menu of the view button, select Notebooks. Click the name of the notebook you want to view.

You can also view an HTML version of your notebook directly from:

```
http://<your-anaconda-repo>/USERNAME/my-notebook
```

NOTE: Replace `<your-anaconda-repo>` with your Repository name, `USERNAME` with your username and `my-notebook` with the name of your notebook.

Anyone who has anaconda-client and access to Repository can download your notebook. To download the notebook, open Anaconda Prompt or Terminal and enter:

```
anaconda download USERNAME/my-notebook
```

Sharing environments

A saved conda environment can be uploaded to Anaconda Repository with the web interface or the `anaconda upload` command.

To save the environment, run this command in an Anaconda Prompt or Terminal window:

```
conda env export -n my-environment -f my-environment.yml
```

To upload it with the web interface go to:

```
https://<your-anaconda-repo>/<USERNAME>/environments
```

Then use the Upload button in the top right corner.

To upload it with the `anaconda upload` command:

```
anaconda upload my-environment.yml
```

NOTE: Replace `my-environment` with the name of your environment.

You can view a list of your uploaded environments at:

```
http://envs.<your-anaconda-repo>/USERNAME
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository and `USERNAME` with your username.

Anyone who has access can download and install your environment. Open Anaconda Prompt or Terminal and then enter:

```
conda env create user/my-environment
source activate my-environment
```

NOTE: Replace `user` with your username and `my-environment` with the name of your environment.

How to...

- *Use packages*
- *Use the Anaconda Client CLI*
- *Build packages*

Use packages

Find a package

From Anaconda Repository, you can search for packages by package name. From the top navigation bar of any page, in the search box enter the package name. You can filter your searches using type of the packages, access or labels and you can sort results by number of favorites or number of downloads by clicking the search results column heading.

Download and install a conda package from Repository

To install a conda package, in a Terminal window or an Anaconda Prompt run:

```
conda install -c USERNAME PACKAGE
```

NOTE: Conda expands `USERNAME` to a URL such as `https://<your-anaconda-repo>/username` based on the settings in the `.condarc` file.

NOTE: Replace `USERNAME` with your username and `PACKAGE` with the name of the desired package.

Download and install a PyPI package from Repository

To install a PyPI package, in a Terminal window or an Anaconda Prompt run:

```
pip install --index-url pypi.anaconda.org/USERNAME/PACKAGE
```

NOTE: Replace `USERNAME` with your username and `PACKAGE` with the name of the desired package.

Use the Anaconda Client CLI

Install Client

See *Installing the Anaconda Client CLI*.

Find my Client login credentials

Your credentials for Client are those that you used to create an account on Repository.

To get help:

1. In a browser, navigate to your Repository.
2. Select the **Sign In** tab.
3. Click either the I forgot my password link or the I forgot my username link.

Log into Client

After you have downloaded and configured Client, in a Terminal window or an Anaconda Prompt, run:

```
anaconda login
```

Display a list of Client commands

In a Terminal window or Anaconda Prompt, run:

```
anaconda --help
```

Find out more about a Client command

In a Terminal window or Anaconda Prompt, run:

```
anaconda COMMANDNAME -h
```

NOTE: Replace `COMMANDNAME` with the name of the command about which you want more information.

List all available Client configuration files

In a Terminal window or Anaconda Prompt, run:

```
anaconda config --files
```

List all of your Client configuration variables

In a Terminal window or Anaconda Prompt, run:

```
anaconda config --show
```

Find out more about Client

If you have a question that you cannot answer using the help command or documentation, contact your system administrator who has access to Anaconda Enterprise Support.

Build packages

Build and upload a package

For a quick example, see *Building and uploading new packages*.

Test a built package

In a Terminal window or Anaconda Prompt, specify the `--use-local` option:

```
conda create --use-local -n test PACKAGE
```

NOTE: Replace `PACKAGE` with the name of your package.

Upload a package to Repository

In a Terminal window or Anaconda Prompt, run:

```
anaconda upload PACKAGE
```

NOTE: Replace `PACKAGE` with the name of your package.

Find help for uploading packages

You can obtain a complete list of upload options, including:

- Package channel.
- Label.
- Availability to other users.

- Metadata.

In a Terminal window or Anaconda Prompt, run:

```
anaconda upload -h
```

Tutorials

Using labels in the development cycle

Anaconda Repository *labels* can be used to facilitate a development cycle and organize the code that is in development, in testing and in production, without affecting non-development users.

In this tutorial, we show how to use a "test" label, so that you can upload files without affecting your production-quality packages. Without a `--label` argument the default label is "main."

1. You need to begin with a conda package. If you do not have one, use our example conda package. Before you build the package, edit the version in the `meta.yaml` file in `anaconda-client/example-packages/conda/` to be `2.0`:

   ```
   git clone https://github.com/anaconda-platform/anaconda-client
   cd anaconda-client/example-packages/conda/
   nano meta.yaml # Bump version to 2.0
   conda config --set anaconda_upload no
   conda build .
   ```

2. Upload your test package to Repository using the Client *upload* command. Adding the `--label` option tells Repository to make the upload visible only to users who specify that label:

   ```
   anaconda upload /path/to/conda-package-2.0.tar.bz2 --label test
   ```

   NOTE: Replace `/path/to/` with the path to where you stored the package.

3. You now can see that even when you search conda "main," you do not see the 2.0 version of the test package. This is because you need to tell conda to look for your new "test" label.

4. The `--override` argument tells conda not to use any channels in your `~/.condarc` file.

   The following command produces no 2.0 results:

   ```
   conda search --override -c USERNAME conda-package
   ```

   NOTE: Replace USERNAME with your username.

   Your 2.0 package is here:

   ```
   conda search --override -c USERNAME/label/test conda-package
   ```

   NOTE: Replace `USERNAME` with your username.

5. You can give the label `USERNAME/label/test` to your testers.

   NOTE: Replace `USERNAME` with your username.

6. Once they finish testing, you may then want to copy the test packages back to your "main" label:

   ```
   anaconda label --copy test main
   ```

   Your version 2.0 is now in main:

   ```
   conda search --override -c USERNAME conda-package
   ```

   NOTE: Replace `USERNAME` with your username.

You can also manage your package labels from your dashboard: `https://<your-anaconda-repo>/USERNAME/conda-package`.

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, and `USERNAME` with your username.

If you use `anaconda-client` 1.7 or higher, you can use `anaconda move` to move packages from one label to another:

```
anaconda move --from-label OLD --to-label NEW SPEC
```

Replace OLD with the old label, NEW with the new label, and SPEC with the package to move. SPEC can be either "user/package/version/file", or "user/package/version" in which case it moves all files in that version.

Working with other file types

In addition to uploading or downloading *packages*, you can also upload or download other file types to/from Anaconda Repository.

Uploading other file types

You can upload any type of file with *Anaconda Client command line interface* (CLI) by using the steps below.

PyPI package files, conda package files and notebook files are automatically detected. There is no auto-detect for other types of files, so you must explicitly specify the `package`, `package-type` and `version` fields.

In the following example, we upload a spreadsheet named `baby-names` in comma separated value (CSV) format.

1. Create a new package, which creates a *namespace* that can hold multiple files:

   ```
   anaconda login
   anaconda package --create jsmith/baby-names
   ```

2. Upload the file to the new namespace:

   ```
   anaconda upload --user jsmith --package baby-names --package-type file --version␣
   ↪1 baby-names1.csv
   ```

NOTE: In this example:

- The user or organization name is "jsmith."
- The package name is "baby-names."
- The package type is "file."

- The version is “1.”
- The full filename is `baby-names1.csv`.

Downloading other file types

Files, such as the one created above, are available at:

```
https://<your-anaconda-repo>/USERNAME/PACKAGE
```

Anyone can download these files using Client:

```
anaconda download USERNAME/PACKAGE
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `USERNAME` with the desired username and `PACKAGE` with the desired package name.

If the repository has multiple files with the same name and different extensions, `anaconda download` will download all of them by default. If you use `anaconda-client` 1.7 or higher, you can use `anaconda download` with the option `--package-type` or `-t` to specify only one of these files. This option can work with the values `pypi`, `conda`, `ipynb`, and `env`.

Tasks

This guide covers all the everyday tasks for a user of Anaconda Repository.

Creating an account

The information below applies to personal Anaconda Repository accounts. For information on organization accounts, see *Working with organizations*.

You do not need an Anaconda Repository account to find, download and use packages.

However, you do need a Repository account to:

- Author packages.
- Upload packages, notebooks and environments.
- Access private packages that are shared with you.
- Create organizations.

If your administrator sent you an email with a login address, username and password, use them.

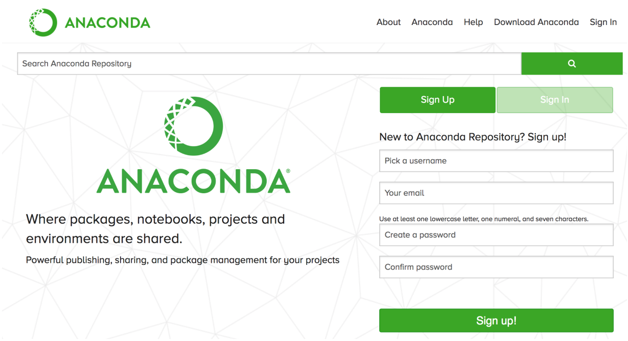
Otherwise, to sign up for an Anaconda Repository account:

1. In a browser, go to the address your administrator gave you.

1. Make sure the **Sign Up** tab is active.

   NOTE: There is also a **Sign In** tab for existing users.

2. Select a username.



Where packages, notebooks, projects and environments are shared.

Powerful publishing, sharing, and package management for your projects

New to Anaconda Repository? Sign up!

Pick a username

Your email

Use at least one lowercase letter, one numeral, and seven characters.

Create a password

Confirm password

3. Enter your email address.
4. Create a password that is at least 7 characters long.
5. Enter the password again to confirm it.
6. Read and accept the Terms and Conditions.
7. Click the Sign up button.

   The system creates your user account, logs you in and displays your *personal dashboard*.

Using your Repository dashboard

When you log in to Repository, your personal dashboard is displayed.

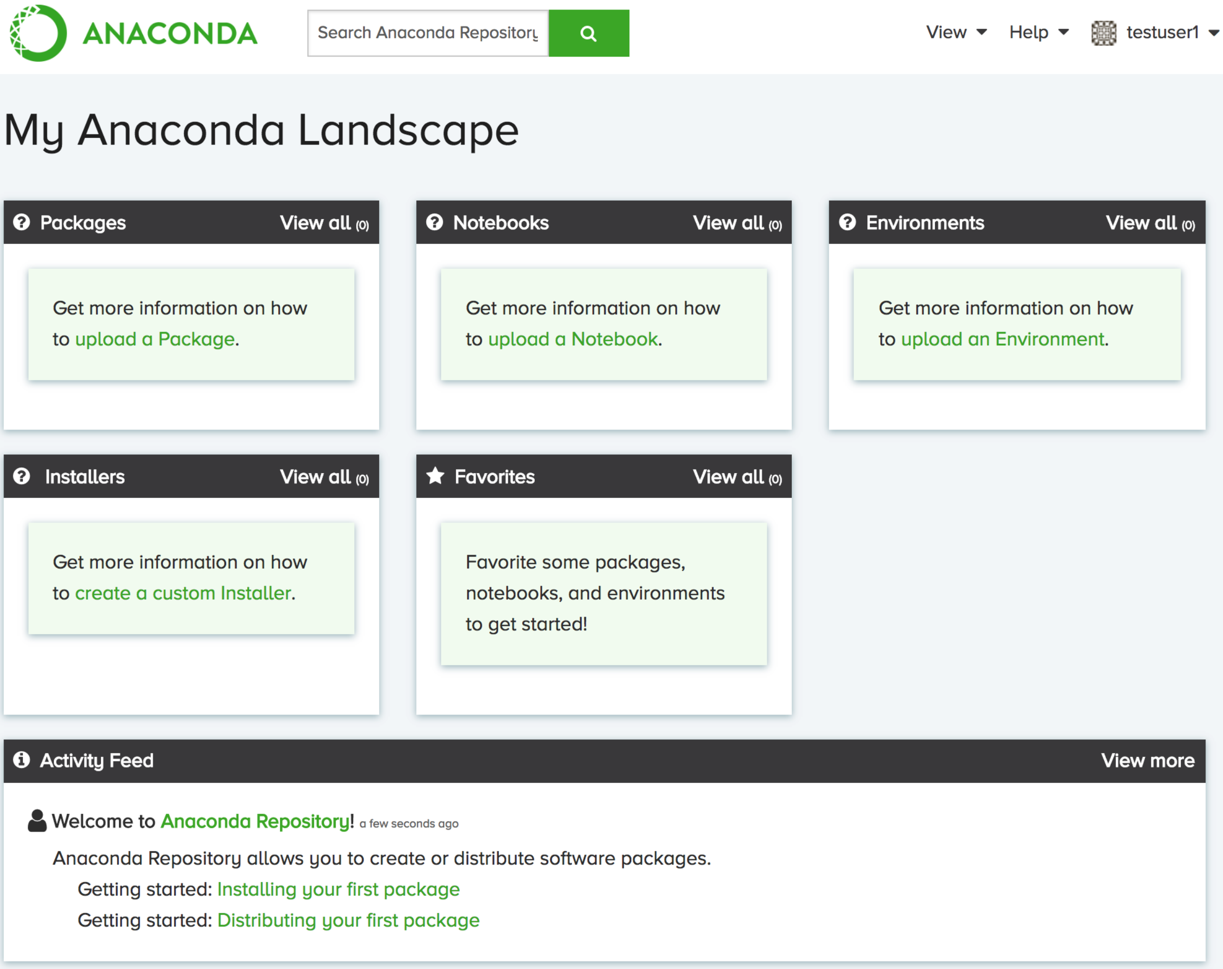
In the top navigation bar, the currently active user or organization is shown at the far right.

TIP: If the email address on your account is associated with a Gravatar account, Repository displays your profile photo. To associate your email address with Gravatar or to change your Gravatar profile photo, see gravatar.com.

Packages, notebooks, environments, projects and installers that you have created with this account appear on your Landscape.

Click the view button to see the following options:

- Landscape: Your home page.
- Favorites: Other users' packages that you have starred.
- Packages: Only packages you have created.
- Notebooks: Only notebooks you have created.
- Environments: Only environments you have created.
- Installers: If you have created and uploaded installers using *Cloudera*, they are displayed here.
- Projects: If you have created and uploaded *projects*, they are displayed here.



My Anaconda Landscape

Get more information on how to upload a Package.

Get more information on how to upload a Notebook.

Get more information on how to upload an Environment.

Get more information on how to create a custom Installer.

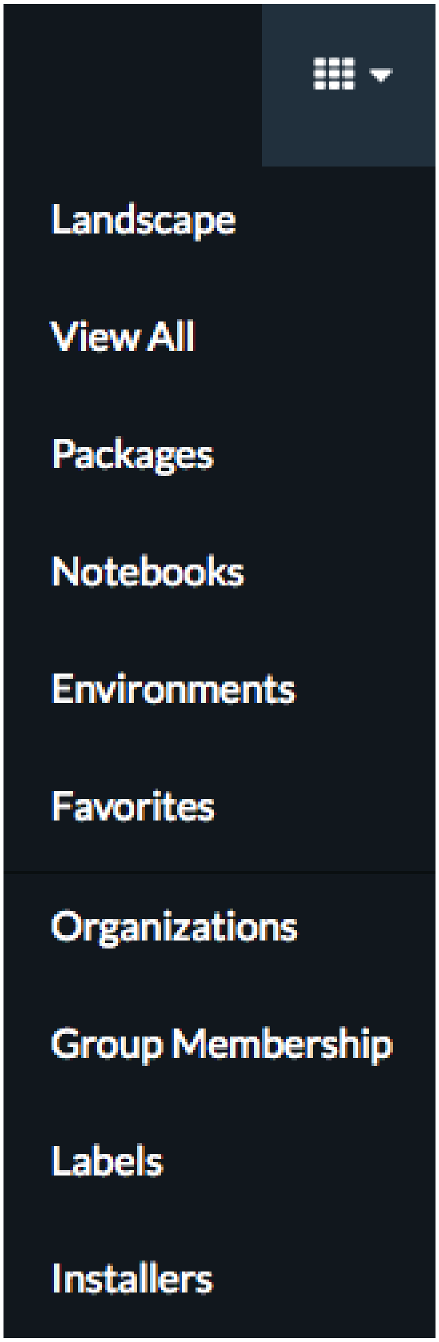
Favorite some packages, notebooks, and environments to get started!

Welcome to Anaconda Repository! a few seconds ago

Anaconda Repository allows you to create or distribute software packages.

Getting started: Installing your first package

Getting started: Distributing your first package



Landscape
View All
Packages
Notebooks
Environments
Favorites
Organizations
Group Membership
Labels
Installers

Installing the Anaconda Client CLI

You can use the Anaconda Client command line interface (CLI) in an Anaconda Prompt or Terminal window to:

- Connect to and manage your Anaconda Repository account.
- Upload *packages* you have created.
- Generate access *tokens* to allow access to private packages.

NOTE: Anaconda Client is not necessary to search for and download packages.

Anaconda Client requires conda to be installed. If you have Anaconda, conda is already installed.

To install with conda, open Anaconda Prompt or Terminal window and enter:

```
conda install anaconda-client
```

After installing, view the complete list of Client tasks with this command from Anaconda Prompt or Terminal window:

```
anaconda -h
```

Working with organizations

Creating an organization

1. Log in to your Anaconda Repository.
2. From the far-right drop-down menu next to your username, select New Organization.
3. Scroll to the “Create Organization” box and enter a name for your organization.

   NOTE: Organization names can include dashes, but not spaces or special characters.
4. Supply an email address for the organization, then click the Create Organization button.

   The system displays the dashboard for the new organization.

As the creator and owner of an organization, you have automatic administrative access to this organization and any packages associated with the organization.

From the far-right drop-down menu, Profile option shows a list of all organizations to which you belong.

Uploading packages to an organization

Only the co-owners of an organization may upload packages to the organization.

To upload a package to an organization, in a Terminal or Anaconda Prompt use the `-u/--user` option:

```
anaconda upload --user ORGANIZATION package.tar.bz2
```

NOTE: Replace `ORGANIZATION` with the name of the organization, and `package.tar.bz2` with the name of the package.

Transferring packages to an organization

See *Transferring a package to a new owner*.

Customizing users and groups

Only the co-owners of an organization may customize users and groups of the organization.

To add, remove, or edit group and user access for an organization you administer:

From the top right drop-down menu, select Groups, then click the name of the group you want to edit.

In the Members box, type the username of the user you want to add as a member, thenc lick the Add button.

To remove a member, in the list on the Members page, click the delete icon (trash can).

Users receive a dashboard notification when you add them to an organization.

Creating groups for differing access levels

Within an organization, you can create a group to customize access for a group of users:

1. From the top right drop-down menu, select Groups, then click the + New Group button.
2. Give the group a name and click the Create group button.
3. In the Members box, add the desired members by username.
4. Add installers, packages, notebooks, projects or environments that this group can access.
5. Click the Save Group button.

Deleting an organization

To delete an organization you administer and erase all data associated with it:

1. At the top right of the Repository interface, in the Profile list, select Switch To.
2. Select the organization you want to delete.
3. In the Profile list, select Settings.
4. Select the Account option. You may be asked to verify your password.
5. In the Delete Account section, click the Delete button.

   A confirmation page requests that you provide the full name of the organization.

Working with packages

All files uploaded to Anaconda Repository are stored in *packages*. Each Repository package is visible at its own unique URL based on the name of the user who owns the package and the name of the package. You can create a Repository package and then upload files into it.

Each user and organization has their own location called a *namespace* where they may host packages.

A *label* is part of the URLs for Repository where conda looks for packages. Each file within a package may be tagged with one or more labels, or not tagged at all to accept the default label of `main`. Labels are searched only if you specify a label.

Using package managers

Repository supports two package managers, conda and PyPI. To work with conda or PyPI packages, you must use their corresponding subdomains.

EXAMPLE: To install conda packages from the user "travis," use the Repository URL:

```
https://conda.<your-anaconda-repo>/travis
```

EXAMPLE: To install PyPI packages from the user "travis," use the Repository URL:

```
https://pypi.<your-anaconda-repo>/travis
```

Working with conda packages

Building a conda package

To build a package using `conda build`:

1. Install Anaconda Client and conda build:

   ```
   conda install anaconda-client conda-build
   ```

2. Choose the repository for which you would like to build the package. In this example, we use a simple, public conda test package:

   ```
   git clone https://github.com/anaconda-platform/anaconda-client
   cd anaconda-client/example-packages/conda/
   ```

   In this directory, there are two required files, build.sh, and meta.yaml.

   NOTE: Linux and macOS systems are Unix systems. Packages built for Unix systems require a `build.sh` file, packages built for Windows require a `bld.bat` file, and packages built for both Unix and Windows systems require both a `build.sh` file and a `bld.bat` file. All packages require a `meta.yaml` file.

3. To build the package, turn off automatic Client uploading and then run the `conda build` command:

```
conda config --set anaconda_upload no
conda build .
```

All packages built in this way are placed in a subdirectory of *Anaconda's* `conda-bld` directory.

4. You can check where the resulting file was placed with the `--output` option:

```
conda build . --output
```

For more information on conda's overall build framework, you may also want to read the articles Building conda packages and Tutorials on conda build.

Uploading a conda package

Upload the test package to Repository with the *anaconda upload* command:

```
anaconda login
anaconda upload /path/to/conda-package.tar.bz2
```

NOTE: Replace `/path/to/` with the path to where you stored the package.

Installing conda packages

You can install conda packages from Repository by adding channels to your conda configuration.

1. Because conda knows how to interact with Repository, specifying the channel "sean" translates to `https://<your-anaconda-repo>/sean`:

```
conda config --add channels sean
```

2. You can now install public conda packages from Sean's Repository account. Try installing the testci package at `https://<your-anaconda-repo>/sean/testci`:

```
conda install testci
```

You can also install a package from a channel with a token and a label:

```
conda install -c https://conda.anaconda.org/t/<token>/<channel>/label/<labelname>
↪<package>
```

NOTE: Replace `<token>` with the provided token,``<channel>`` with the user channel, `<labelname>` with the label name and `<package>` with the package name you want to install.

Working with PyPI packages

Uploading PyPI packages

You can test PyPI package uploading with a small, public example package saved in the anaconda-client repository:

1. Begin by cloning the repository from the command line:

```
git clone git@github.com:anaconda-platform/anaconda-client.git
cd anaconda-client/example-packages/pypi/
```

2. You can now create your PyPI package with the `setup.py` script:

   ```
   python setup.py sdist
   ```

3. Your package now is built as a source "tarball" and is ready to be uploaded with:

   ```
   anaconda upload dist/*.tar.gz
   ```

Your package is now available at:

```
http://<your-anaconda-repo>/USERNAME/PACKAGE
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `USERNAME` with your username and `PACKAGE` with the package name.

Installing PyPI packages

The best way to install a PyPI package is using pip. For the following command, you can use the package you authored in the above steps:

```
pip install --extra-index-url https://pypi.<your-anaconda-repo>/USERNAME/PACKAGE
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `USERNAME` with your username and `PACKAGE` with the test-package name.

Using cross-platform "noarch" packages

As of Anaconda Repository version 2.6.0, your Repository supports conda "noarch" packages that contain no operating system-specific files.

The conda build system allows you to specify "no architecture" when building a package, so it is compatible with all platforms and architectures. Noarch packages from your Repository instance can be downloaded and installed on any platform.

NOTE: Noarch packages are not compatible with Anaconda constructor. If you intend to use the packages with Anaconda constructor, build the packages for specific operating systems.

Building noarch packages

To specify a noarch build, use the `noarch` key in the `build` section of your conda recipe's `meta.yaml` file:

```
build:
      noarch: generic
```

See the conda documentation for full information on noarch packages.

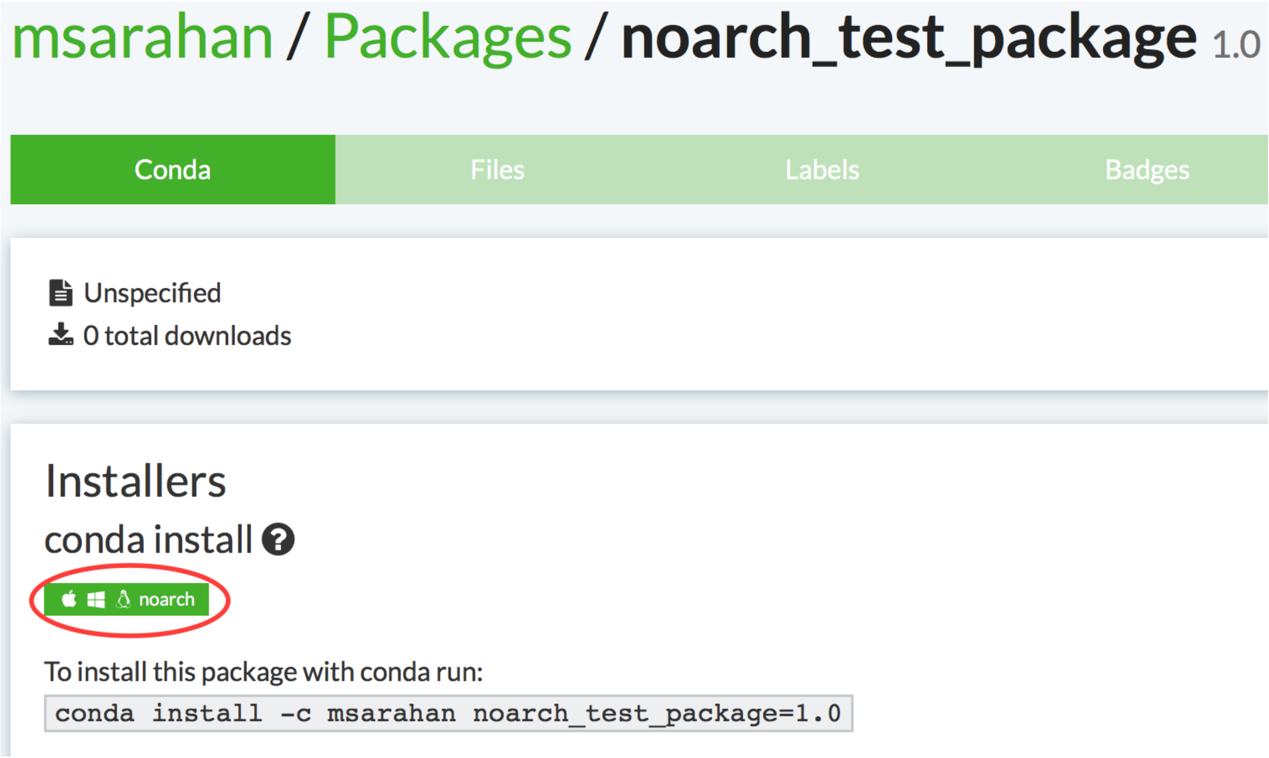
Additional examples can be found in the conda-recipes repository on github.

Uploading noarch packages

You can upload noarch packages to Repository in the same manner as any other package:

```
anaconda upload babel
```

Noarch packages are identified on Repository by a cross-platform icon:



msarahan / Packages / noarch_test_package 1.0

Unspecified

0 total downloads

Installers

conda install

To install this package with conda run:

```
conda install -c msarahan noarch_test_package=1.0
```

Uploading a package

To upload a package to Repository, using the Client CLI, run the *upload* command:

```
anaconda login
anaconda upload PACKAGE
```

NOTE: Replace `PACKAGE` with the name of the desired package.

Repository automatically detects packages and notebooks, package or notebook types, and their versions.

Your package is now available at:

```
https://<your-anaconda-repo>/USERNAME/PACKAGE
```

NOTE: `<your-anaconda-repo>` is the name of your local Repository, `USERNAME` is your username and `PACKAGE` is the package name.

Anyone can download your package by using Client:

```
anaconda download USERNAME/PACKAGE
```

NOTE: USERNAME is their username, and PACKAGE is your package name.

If you want to restrict access to your package, see *Controlling access to packages*.

Controlling access to packages

By default, all packages, notebooks and environments uploaded to Repository are public, meaning they are accessible to anyone who has access to Repository.

When you *make a package private*, only you and the users you authorize can access it.

You can authorize users to access your private package in two ways:

- *Use a group* inside an organization account—only group members who are logged in can access the package. This is the best way to control access to your private packages because it allows you to set separate permissions for each package, notebook or environment.
- *Use a token control system*—only users who have the appropriate *token* can access the private package or channel.

After you grant other users access, they can *download and install* your package using the Web UI or Client.

Making a package private

1. In the Web UI, in the **Tools** menu, select Packages.
2. OPTIONAL: If the packages you are looking for are not visible, under Filters, in the Type list, select All.
3. Select the checkbox next to each package you want to make private.
4. Click the **Settings** tab, and then click the **Admin** tab in the sidebar.

   NOTE: You can also reach this page at the following URL:

   ```
   https://<your-anaconda-repo>/USERNAME/PACKAGE/settings/admin
   ```

   Replace <your-anaconda-repo> with the name of your local Repository, USERNAME with your username and PACKAGE with the name of the package.
5. Click Set access, then select Private.

NOTE: You can use the same procedure and URL to make Jupyter Notebooks and conda environments private.

Using groups to allow access to private packages

1. *Create an organization*.
2. *Upload* or *transfer* the package to the organization.

3. Within the organization, *create a group* with the appropriate users, permissions, and packages.

Creating a token to allow access to a private package or channel

You can control access to private packages and channels with the *token* system. All Repository URLs can be prefixed with `/t/<token>` to allow access.

The degree of access a token grants is completely configurable when you generate it. You can generate multiple tokens to control which groups of users have access to certain features if they have the appropriate token.

Tokens provide access to all packages in a specified channel. Separate permissions per package, notebook or environment may be better handled with *organizations and groups*.

You can generate tokens using the Web UI or Anaconda Client.

NOTE: By default, tokens expire after one year.

Generating a token in the Web UI

1. Navigate to:

   ```
   https://<your-anaconda-repo>/<channel>/settings/access
   ```

   NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, and `<channel>` with the name of the desired channel.

2. In the Token Name box, type a name for the token:

3. Select the appropriate checkboxes for the type of access you want to allow for users of this token.

   EXAMPLE: To allow users to download private packages or packages from private channels, select Allow private downloads from Conda repositories.

4. Click the Create button.

Generating a token with Client

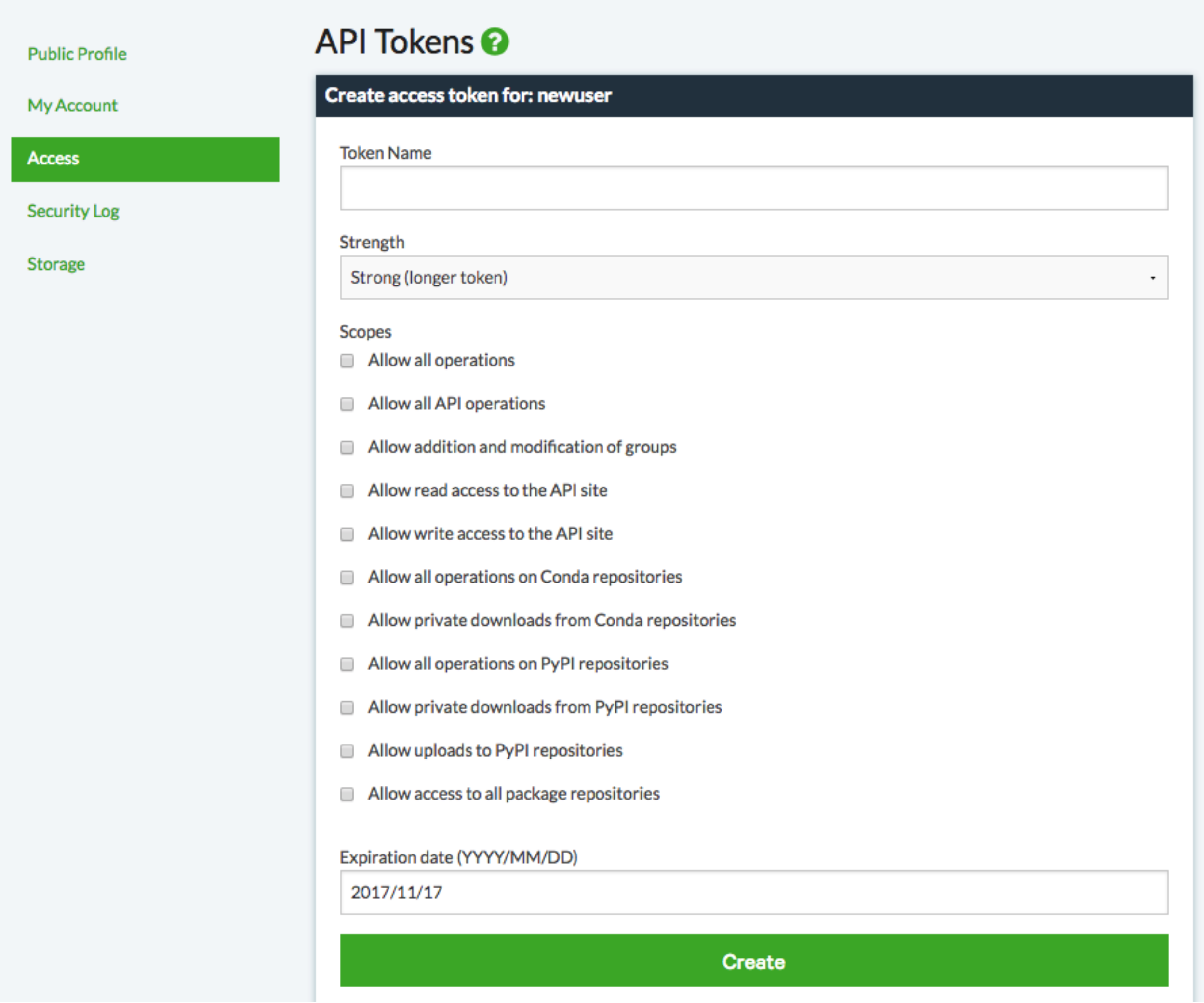
1. In a Client Terminal window or Anaconda Prompt, run:

   ```
   anaconda auth --create --name YOUR-TOKEN-NAME --scopes repos conda:download'
   ```

   NOTE: Replace `YOUR-TOKEN-NAME` with a name for the new token.

   Provide scopes as a space-separated, quoted list. The token produced by the above command provides access to download any of your private conda repositories. The available scopes are:

   - `all`: Allow all operations.
   - `api`: Allow all API operations.
   - `api:modify-group`: Allow addition and modification of groups.
   - `api:read`: Allow read access to the API site.



API Tokens

Create access token for: newuser

Token Name

Strength

Strong (longer token)

Scopes

□ Allow all operations

□ Allow all API operations

□ Allow addition and modification of groups

□ Allow read access to the API site

□ Allow write access to the API site

□ Allow all operations on Conda repositories

□ Allow private downloads from Conda repositories

□ Allow all operations on PyPI repositories

□ Allow private downloads from PyPI repositories

□ Allow uploads to PyPI repositories

□ Allow access to all package repositories

Expiration date (YYYY/MM/DD)

2017/11/17

- `api:write`: Allow write access to the API site.
- `conda`: Allow all operations on conda repositories.
- `conda:download`: Allow private downloads from conda repositories.
- `pypi`: Allow all operations on PyPI repositories.
- `pypi:download`: Allow private downloads from PyPI repositories.
- `pypi:upload`: Allow uploads to PyPI repositories.
- `repos`: Allow access to all package repositories.

2. You can enable the token with the `conda config` command:

```
conda config --add channels https://conda.anaconda.org/t/<token>/<channel>
```

Or to add a channel with a token and label:

```
conda config --add channels https://conda.anaconda.org/t/<token>/<channel>/label/
↪<labelname>
```

NOTE: Replace `<token>` with your token string,‘‘<channel>‘‘ with the desired channel name, and `<labelname>` with the label name.

NOTE: If you lose the token's random alphanumeric string, you must *revoke the token* and create a new one.

Using a token

The token can be used to:

- Add a channel from which to install private packages:

```
conda config --add channels https://conda.<your-anaconda-repo>/t/<token>/<channel>
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `<token>` with the provided token and `<channel>` with a user channel.

- Install a private package without first adding a channel:

```
conda install -c https://conda.<your-anaconda-repo>/t/<token>/<channel> <package>
```

To install a package from a channel using a token and a label name:

```
conda install -c https://conda.<your-anaconda-repo>/t/<token>/<channel>/label/
↪<labelname> <package>
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `<token>` with the provided token, `<channel>` with a user channel, `<labelname>` with the label name and `<package>` with the name of the package to install.

- Install a private PyPI package:

```
pip install --index-url https://pypi.<your-anaconda-repo>/t/<token>/<channel>/
↪PACKAGE
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `<token>` with the provided token, `<channel>` with a user channel and `PACKAGE` with the name of the desired package.

NOTE: Private PyPI packages can also be installed using:

```
https://pypi.<your-anaconda-repo>/t/<token>/<channel>
```

Revoking a token

You can revoke tokens using the Web UI or Client.

To revoke a token using the Web UI, from the far-right drop-down menu, select My Settings, then from the left navigation select Access.

At the bottom of the page, you will see a list of all tokens you have generated. Click the name of the token you want to revoke, then in the dialog box that appears, click the Revoke Token button.

Or to revoke a token using Client, run:

```
anaconda auth -r YOUR-TOKEN-NAME
```

NOTE: Replace `YOUR-TOKEN-NAME` with the name of the token you want to revoke.

Downloading and installing a package

To download a package using the Web UI, in a web browser, navigate to the organization's or user's channel.

To download a package using Client:

- Run:

  ```
  conda install anaconda-client
  anaconda login
  conda install -c OrgName PACKAGE
  ```

  NOTE: Replace `OrgName` with the organization or username and `PACKAGE` with the package name.

- Or run:

  ```
  conda install anaconda-client
  anaconda login
  conda install -c https://conda.<your-anaconda-repo>/OrgName PACKAGE
  ```

  NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `OrgName` with the organization name or username and `PACKAGE` with the package name.

Transferring a package to a new owner

When you create or add a package, by default it is attached to your individual profile. You can transfer ownership to another owner account you control, such as an organization profile you manage.

To transfer a package to a new owner:

1. On your dashboard—or the dashboard of an organization you administer—select the package for which you want to transfer ownership.

   The system displays options for that package.

2. To display the package settings, select the Settings option.

3. Select the Admin option.

4. Under Transfer this package to a new owner, click the Transfer button.
5. Select the organization name for the new owner.
6. Click the Transfer Ownership button.

Adding and removing collaborators

You can add other users that are not part of an organization to collaborate on your packages. You need the usernames of the other users. You can also remove collaborators at any time.

All collaborators have full read/write permissions to the package, even if the package is private.

1. On your dashboard, click the package name.
2. Select the Settings option.
3. In the package settings, select the Collaborators option.
4. To add a collaborator, in the current collaborators, type the username of the person you want to add, then click the Add button.
5. To remove a collaborator, click the red X button next to the collaborator name.

Removing a previous version of a package

To remove a previous version of one of your packages from Repository:

1. On your dashboard, click the package name.
2. Select the **Files** tab.
3. Select the checkbox to the left of the version you want to remove.
4. In the **Actions** menu, select Remove.

You can also use the Client CLI to remove a previous version of a package:

```
anaconda remove jsmith/testpack/0.2
```

NOTE: Replace `jsmith` with your username, `testpack` with the package name and `0.2` with the desired version.

You can now see the change on your profile page:

```
https://<your-anaconda-repo>/USERNAME/PACKAGE
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `USERNAME` with your username and `PACKAGE` with the package name.

Copying a package

To copy a package from the channel `conda-forge` to a personal channel such as `jsmith`:

```
anaconda copy conda-forge/glueviz/0.10.4 --to-owner jsmith
```

`conda-forge/glueviz/0.10.4` is a “spec” and can match either of two formats: `user/package/version` or `user/package/version/filename`.

Deprecated options

Previously labels were called "channels", and the `anaconda copy` command has deprecated options `from-channel` and `to-channel` that expect to operate on labels.

These deprecated options should not be used.

If you attempt to use them in a command such as `anaconda copy --from-channel conda-forge --to-channel jsmith glueviz`, you will get an error that `Label conda-forge does not exist.`

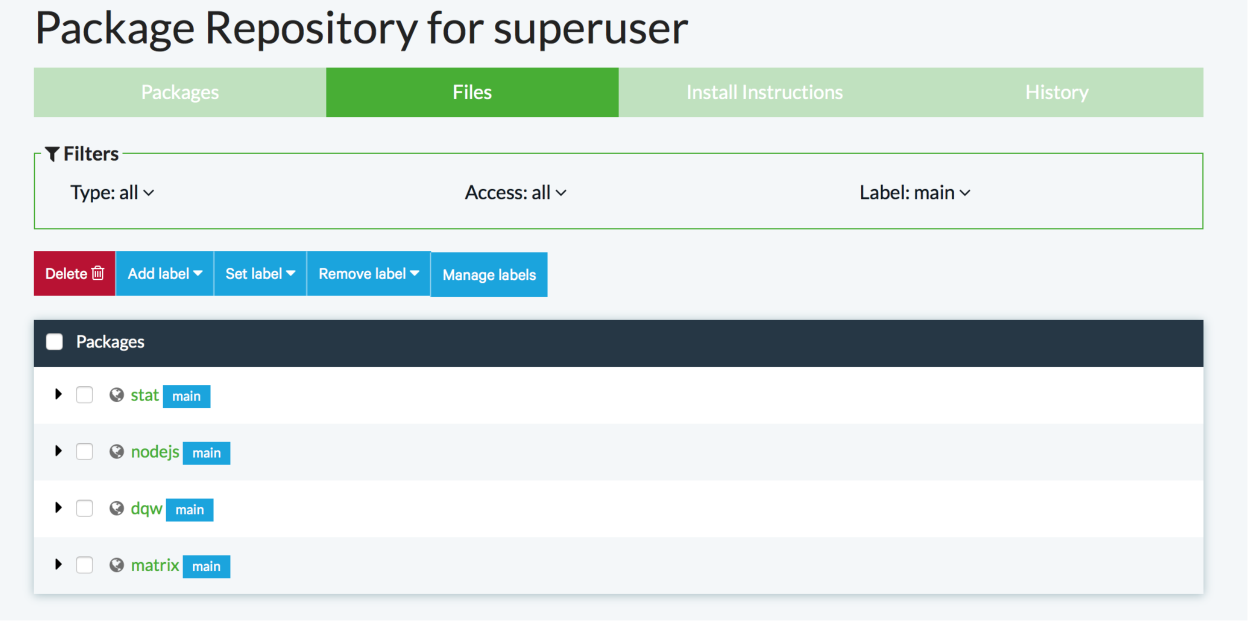
Deleting files from a package

You can delete individual files from a package in Repository, without deleting the entire package.

CAUTION: There is no undo for deleting a file.

To delete individual files from a package in Repository:

1. Access Repository using the Web UI.
2. In the **Tools** menu, select Packages.
3. Click the **Files** tab.
4. OPTIONAL: If the files you want to delete are not visible, under Filters, in the Type list, select All.



Package Repository for superuser

Filters

Type: all

Access: all

Label: main

5. To select individual files, expand the package in which the files are located.
6. Select the checkboxes next to the files you want to delete.
7. Click the Delete button.
8. Enter your account name in the confirmation window.

9. Click Delete to permanently delete the selected files.

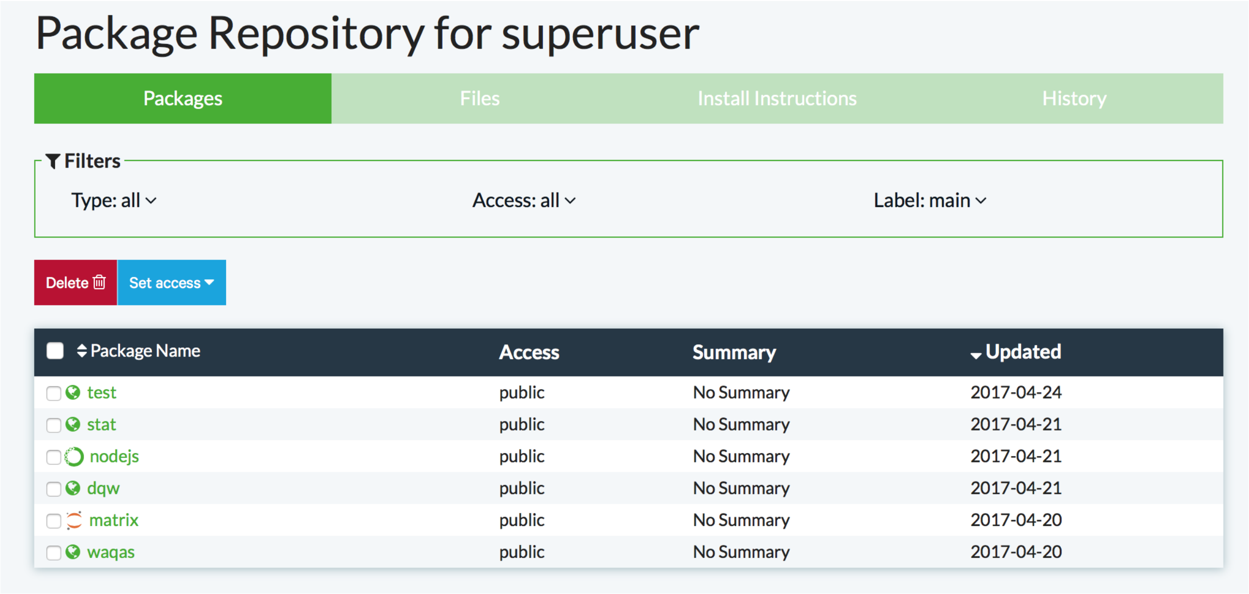
Deleting a package

You can delete an entire package from Repository, including all of its versions.

CAUTION: There is no undo for deleting a package.

To delete a package from Repository:

1. Access Repository using the Web UI.
2. In the **Tools** menu, select Packages.
3. OPTIONAL: If the packages that you want to delete are not visible, under Filters, in the Type list, select All.



Package Repository for superuser

Type: all
Access: all
Label: main

□ Package Name	Access	Summary	Updated
□ test	public	No Summary	2017-04-24
□ stat	public	No Summary	2017-04-21
□ nodejs	public	No Summary	2017-04-21
□ dqw	public	No Summary	2017-04-21
□ matrix	public	No Summary	2017-04-20
□ waqas	public	No Summary	2017-04-20

4. Select the checkbox next to the packages you want to delete.
5. Click the Delete button.
6. Enter the account name in the confirmation window.
7. Click Delete to permanently delete the selected package(s).

You can also use the Client CLI to delete a package:

```
anaconda remove jsmith/testpak
```

NOTE: Replace `jsmith` with your user name, and `testpak` with the package name.

You can now see the change on your profile page:

```
https://<your-anaconda-repo>/USERNAME
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository and `USERNAME` with your username.

Also see the tutorial *Using labels in the development cycle*.

Working with Jupyter notebooks

As of Anaconda Repository version 2.3.0, you can upload and download Jupyter notebooks like other files.

You can also view a static copy of notebooks in Repository. The ability to run notebooks inside Repository will become available in a future release.

To run notebooks, use Anaconda Navigator or AE Notebooks server.

Uploading a notebook

The default maximum allowed size for notebooks is 25 MB. This limit can be changed by setting the *MAX_IPYNB_SIZE* variable in the `config.yaml` file.

To upload a notebook to your user account, at the Anaconda Prompt or in a Terminal window, run:

```
anaconda upload -p my-notebook my-notebook.ipynb
```

NOTE: Replace `my-notebook` with the name of the notebook you want to upload.

To upload a new version of your notebook, while retaining the original version, upload it with the version switch from a Terminal window or an Anaconda Prompt:

```
anaconda upload -p my-notebook -v 1.1 my-notebook.ipynb
```

NOTE: Replace `my-notebook` with the name of the notebook you want to upload.

Finding a notebook

You can view an HTML version of your notebook at:

```
http://notebooks.<your-anaconda-repo>/USERNAME/my-notebook
```

NOTE: Replace `<your-anaconda-repo>` with the name of your local Repository, `USERNAME` with your username and `my-notebook` with the name of your notebook.

To see another user’s notebook, browse to the associated user account on your Repository installation.

Downloading a notebook

Anyone with access to Repository can download your notebook using the Anaconda Prompt or Terminal window:

```
anaconda download USERNAME/my-notebook
```

NOTE: Replace `USERNAME` with your username, and `my-notebook` with the name of your notebook.

Working with environments

A saved conda environment can be uploaded to Anaconda Repository with the web interface or the `anaconda upload` command.

# To save the environment, run this command in an Anaconda Prompt or Terminal window:

```
conda env export -n my-environment -f my-environment.yml
```

To upload it with the web interface go to:

```
https://<your-anaconda-repo>/<USERNAME>/environments
```

Then use the Upload button in the top right corner.

To upload it with the `anaconda upload` command:

```
anaconda upload -f my-environment.yml
```

NOTE: Replace `my-environment` with the name of your environment.

1. You can view a list of your uploaded environments in the web interface at:

```
http://envs.anaconda.org/USERNAME
```

NOTE: Replace `USERNAME` with your username.

2. Anyone who has access can download and install your environment. Open a Terminal window or an Anaconda Prompt and then enter:

```
conda env create user/my-environment
source activate my-environment
```

NOTE: Replace `my-environment` with the actual name of your environment.

Working with projects

You can add Anaconda Projects to Anaconda Repository. Projects can be any directory of code and assets. For example, projects often contain notebooks or Bokeh apps.

Adding a project

Use the Anaconda Client to add a project to Repository.

Accessing and managing a project

1. Access Repository using the Web UI.
2. From the drop-down menu of the view button, select Projects.

   The Projects page shows your existing projects.
3. Click on a project to display the Details page for that project, including the files, revisions, history and settings for the project.

4. Under Settings for an individual project, you can change options, set groups and collaborations, and manage administration of the project, including making it public, private or authenticated, transferring membership, or deleting it.

Working with Cloudera Manager parcels

Anaconda Repository provides a way to integrate with Cloudera Manager to distribute your Anaconda data science artifacts to your Hadoop cluster.

Creating parcels, management packs and installers

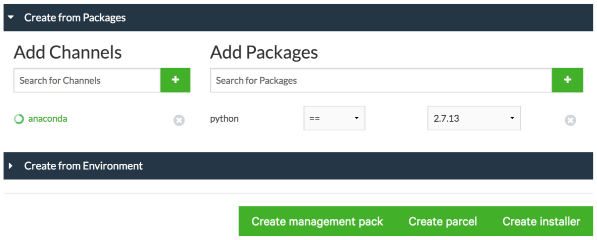
You can create custom Cloudera Manager parcels with the packages you want, including your own packages.

NOTE: Creating custom parcels requires a local mirror of the Anaconda packages.

When creating a parcel, Repository generates a 64-bit Linux installer including the specified packages and a file named `construct.yaml`, which can be used with conda constructor.

To create a custom parcel, management pack or installer:

1. From the view button, select Installers.
2. Click the Create new installer button.

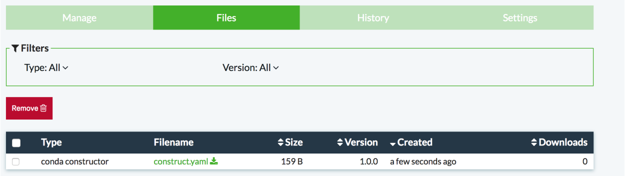


anaconda

python == 2.7.13

NOTE: Use only letters, numbers, dashes and underscores in the installer name.

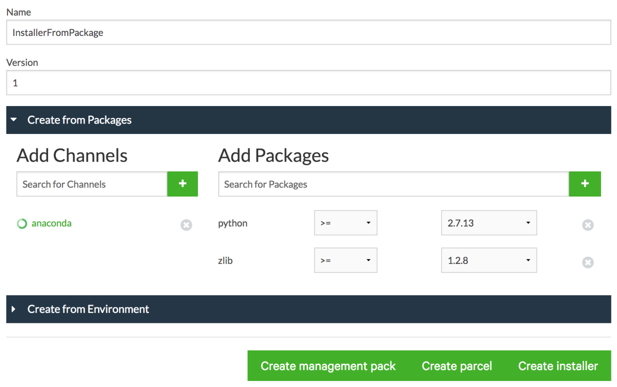
When creating a parcel, Anaconda Repository generates a 64-bit Linux installer with the specified packages, and a file named `construct.yaml` which can be used with conda constructor.



Type	Filename	Size	Version	Created	Downloads
conda constructor	construct.yaml	159 B	1.0.0	a few seconds ago	0

To create just the installer script, click **Create installer**; to create a parcel, click **Create parcel**.

Creating a parcel by selecting packages



Name: InstallerFromPackage

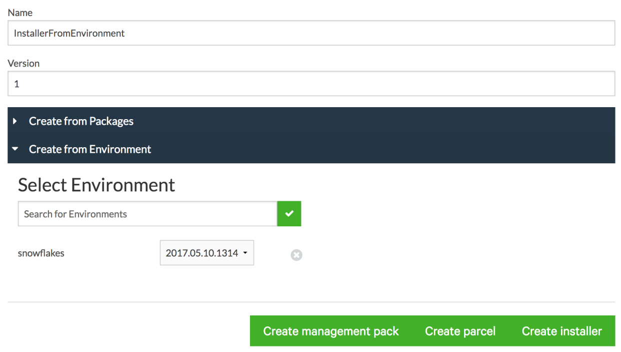
Version: 1

Package		Version
python	>=	2.7.13
zlib	>=	1.2.8

1. Click the **Create from Packages** tab.
2. Add channels from which to fetch packages into the Search for Channels box. Add each channel by clicking the green + (plus) button next to the Search for Channels box.

   NOTE: The `anaconda` user is added by default.
3. Add package names into the Search for Packages box. Add each package by clicking the green + (plus) button next to the Search for Packages box.
4. Set version requirements for each package using the list next to the package name.

Creating a parcel by selecting an environment



Name: InstallerFromEnvironment

Version: 1

snowflakes 2017.05.10.1314

1. Click the **Create from Environment** tab.
2. Type the environment name and click the green checkbox button.
3. Select the environment version from the list next to the environment name.

4. Click the Create management pack button, Create parcel button or Create installer button.

NOTE: By default, conda is not included in a custom parcel. To add additional packages to your environment, you can add them using the Repository Web UI.

A parcel is generated with the prefix of `/opt/cloudera/parcels/PARCEL_NAME`. This is the default location where activated parcels are loaded. If you are deploying parcels in a different directory, you can change this prefix with the PARCELS_ROOT *configuration setting*.

Viewing a list of packages in a custom parcel

To see a list of packages included in your custom parcel, see:

```
/opt/cloudera/parcels/PARCEL_NAME/meta/parcel.json
```

NOTE: Replace PARCEL_NAME with the name of the desired parcel.

Distributing custom parcels

After you have created a custom parcel, you can distribute it to your cluster by adding `http://<repository ip>:<port>/USERNAME/installers/parcels/` as a Remote Parcel Repository URL.

NOTE: Replace `<repository ip>` with the Repository IP address, `<port>` with the port address and `USERNAME` with your user name.

Cloudera Manager detects the parcels hosted on Repository and provides the option to download and distribute the parcels.

By default, Repository generates a parcel file for every compatible distribution.

You can customize which parcel distributions are created by configuring the PARCEL_DISTRO_SUFFIXES *configuration setting*.

NOTE: If you have configured conda via `~/.condarc` on your server for use of a proxy—for example, to mirror behind a proxy—you must disable proxying for Repository. For more information, see the conda documentation.

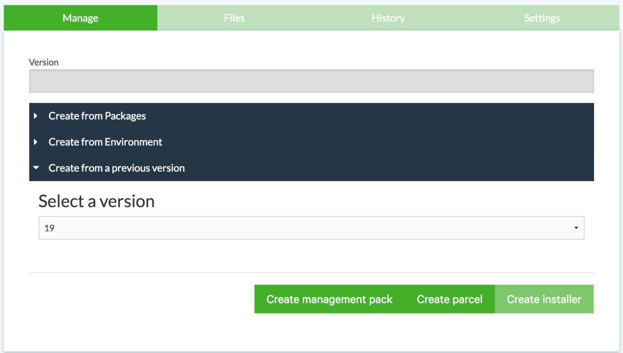
EXAMPLE:

```
proxy_servers:
    https: http://proxy.corp.example.com
    http: http://proxy.corp.example.com
    'http://<repository ip>': false
```

Creating from a previous version

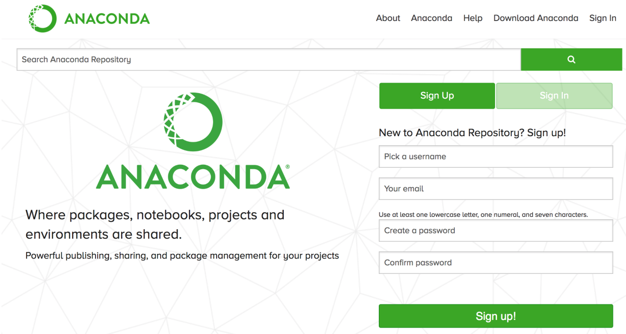
Once you have an installer created, you can return to this page and create a management pack or a parcel from a specific version. Use the **Create from a previous version** option to choose which version you want to use. The **Create Installer** button will be disabled since you have already created an installer. The other buttons will be disabled if you have already created management packs or parcels for those versions.

The versions listed on the drop-down list are the versions that successfully created an installer. An installer is needed to create a management pack or a parcel, so versions which failed won't be listed.



Resetting your password

Open the Anaconda Respository login page:



Where packages, notebooks, projects and environments are shared.

Powerful publishing, sharing, and package management for your projects

New to Anaconda Repository? Sign up!

Use at least one lowercase letter, one numeral, and seven characters.

The **Sign In** tab provides two links to help regain access to your account:

- `I forgot my username.` Click this link to have the username emailed to the email address of record.
- `I forgot my password.` Click this link to have a reset password link sent to the email address of record.

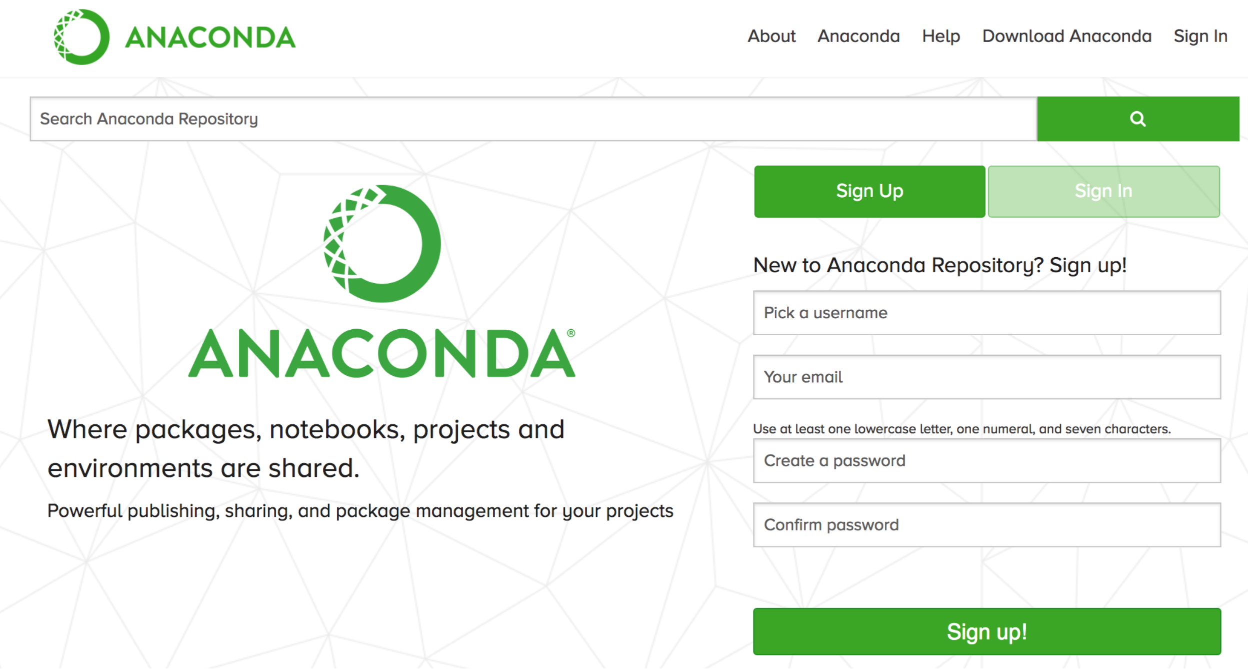
NOTE: The reset password link expires within 24 hours. If you no longer have access to the email account, you can create a new account or email your administrator for assistance.

Also see *Tutorials*.

Anaconda Repository is package management server software that makes it easy to find, access, store and share public and private notebooks, projects, installers, environments, and conda and PyPI packages. Repository also makes it easy to stay current with updates made to the packages and environments you are using.

Anaconda also makes an instance of Anaconda Repository for private enterprises at Anaconda Cloud.

To begin using Repository, read *Getting started*, then the remaining sections of the user guide.



Where packages, notebooks, projects and environments are shared.

Powerful publishing, sharing, and package management for your projects

New to Anaconda Repository? Sign up!

Pick a username

Your email

Use at least one lowercase letter, one numeral, and seven characters.

Create a password

Confirm password

Administration guide

This Anaconda Repository Administration guide is intended for installers and administrators of Anaconda Repository version 2.33.

Installation

This guide provides instructions for installing and configuring Anaconda Repository.

System requirements

Your server must meet the requirements for hardware, software, security and network. Please review and verify that you have met all system requirements before beginning your installation.

- *Hardware requirements*
- *Software requirements*
- *Security requirements*
- *Network requirements*
- *Hardware verification*
- *Software verification*
- *Security verification*

See also the system requirements for all of Anaconda Enterprise.

Hardware requirements

- Physical server or virtual machine.
- CPU: 2 x 64-bit, 2.8 GHz, 8.00 GT/s CPUs or better. *Verify machine architecture*.
- Memory: minimum RAM size of 32 GB, or 16 GB RAM with 1600 MHz DDR3 installed, for a typical installation with 50 regular users. *Verify memory requirements*.
- Storage: Recommended minimum of 100 GB, or 300 GB if you are planning to mirror both Anaconda Repository, which is approximately 90 GB, and the PyPI repository, which is approximately 100 GB, or at least 1 TB for an air gapped environment. Additional space is recommended if Repository is used to store packages built by your organization. *Verify storage requirements*.
- Internet access to download the files from Anaconda Cloud, or a USB drive containing all of the files you need with alternate instructions for air gapped installations.

Software requirements

- Linux environment: Installations have been tested on Red Hat Enterprise Linux/CentOS 6.7, 7.3, 7.4, and 7.5, and Ubuntu 12.04+. *Verify Linux version*.
- Client environment may be Windows, macOS or Linux.
- Ubuntu users may need to install cURL. *Verify cURL access*.
- MongoDB version 2.6+ installed as root and running. Versions through 3.6 are supported. *Verify MongoDB installation*.
- bzip2. *Verify bzip2 installation*.

Security requirements

- Root access or sudo capabilities. *Verify root access and sudo privileges*.
- OPTIONAL: Ability to make IPTables modifications.
- SELinux policy edit privileges.

  NOTE: SELinux does not have to be disabled for Repository operation.

Network requirements

TCP ports are used as follows:

- Inbound TCP 8080, 8443: Anaconda Repository.
- Inbound TCP 22: SSH.
- Outbound TCP 443: Anaconda Cloud.
- Outbound TCP 25: SMTP.
- Outbound TCP 389/636: LDAP(s).

You need your Anaconda.org—Repository in the cloud—account username and password and the installation token provided to you by Anaconda at the time of purchase. If you did not receive your token, please contact your sales representative or our Professional Support Team.

Hardware verification

Machine architecture

Repository is built to operate only on 64-bit computers.

To verify that you have a 64-bit or x86_64 computer, in a terminal window, run:

```
arch
```

This command displays what your system is: 32-bit "i686" or 64-bit "x86_64."

Memory requirements

You need a minimum RAM size of 32 GB, or 16 GB RAM with 1600 MHz DDR3.

In a terminal window, run:

```
free -m
```

This command returns the free memory size in MB.

Storage requirements

To check your available disk space—hard drive or virtual environment size—use the built-in Linux df utility with the `-h` parameter for human readable format:

```
df -h
```

Software verification

Other versions of the Linux environment

Please contact us by filing a GitHub issue if you have problems with a version other than Redhat, CentOS or Ubuntu. Prompts may vary slightly depending on your version.

cURL access for Ubuntu users

RedHat and CentOS Linux distributions have cURL pre-installed, but Ubuntu does not.

To verify cURL access, in a terminal window, run:

```
curl --version
```

If cURL is not found, Ubuntu users can use the Advanced Packaging Tool (APT) to get and install cURL:

```
sudo apt-get install curl
```

TIP: If you already have Miniconda or Anaconda installed, in all versions of Linux you can use the conda command:

```
conda install curl
```

MongoDB version 2.4+ installed

MongoDB version 2.4 or higher must installed as root and running. Versions through 3.4 are supported. To check for the existence of MongoDB and its version number, in a terminal window, run:

```
mongod --version
```

If you get a “not found” message or if the MongoDB version is 2.3 or earlier, then install MongoDB 2.4 or higher using the official installation instructions. Remember to install as root with the sudo command.

MongoDB must always be running before Repository can be started.

To start MongoDB:

```
sudo service mongod start
```

To verify that MongoDB is running:

```
mongo --eval 'db.serverStatus().ok'
```

bzip2 is installed

To check for the existence of bzip2 and its version number, in a terminal window, run:

```
bzip2 --version
```

Security verification

Root access and sudo privileges

The Repository installation process cannot be completed without root access.

To verify that you have sudo privileges, in a terminal window, run:

```
sudo -v
```

Enter your root password when prompted and press Enter.

If you receive a message like the following, contact your system administrator for root access:

```
Sorry, user [username] may not run sudo on [hostname].
```

Installing on an online system

These instructions are for normal Linux installations on machines that have access to the internet.

NOTE: If the destination server is an air gapped system or otherwise does not have access to the internet, see *Installing on an air gapped system*.

Before you start

Your server must meet the requirements for hardware, software, security and network. Please review and verify that you have met all *system requirements* before beginning your installation.

Your support representative provides you with a download URL for the Anaconda Repository installer. Make sure you have the download URL.

1. Install MongoDB 2.6

In a terminal window, create the `yum` repo file as the root user:

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.
↪rackcdn.com"
curl -O $RPM_CDN/mongodb-org-tools-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-shell-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-server-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-mongos-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-2.6.8-1.x86_64.rpm
```

NOTE: Ubuntu users use `apt-get` instead of `yum`.

MongoDB for Redhat and CentOS 7

1. Install MongoDB:

   ```
   sudo yum install -y mongodb-org*
   ```

2. Start MongoDB:

   ```
   sudo systemctl start mongod
   ```

3. Verify that MongoDB is running:

```
$ sudo systemctl status mongod
mongodb (pid 17258) is running...
```

MongoDB for Redhat and CentOS 6.7+

1. Install MongoDB:

   ```
   sudo yum install -y mongodb-org*
   ```

2. Start MongoDB:

   ```
   sudo /etc/init.d/mongod start
   ```

3. Verify that MongoDB is running:

   ```
   $ sudo /etc/init.d/mongod status
   mongodb (pid 17258) is running...
   ```

MongoDB for Ubuntu 12.04+

1. Install MongoDB:

   ```
   sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10

   echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' |␣
   ↪sudo tee /etc/apt/sources.list.d/mongodb.list

   sudo apt-get update

   sudo apt-get install -y mongodb-org=2.6.9 mongodb-org-server=2.6.9 mongodb-org-
   ↪shell=2.6.9 mongodb-org-mongos=2.6.9 mongodb-org-tools=2.6.9
   ```

   NOTE: If you do not specify a version, such as 2.6.9, `apt-get` installs the latest stable version, which is 3.x.

2. Start MongoDB:

   ```
   sudo /etc/init.d/mongod start
   ```

   Verify that MongoDB is running:

   ```
   $ sudo /etc/init.d/mongod status
   mongodb (pid 17258) is running...
   ```

   You receive verification that MongoDB is running:

   ```
   start: Job is already running: mongodb
   ```

Additional MongoDB resources

For additional MongoDB installation information see https://docs.mongodb.org/manual/.

2. Create the Repository administrator account

1. In a Terminal window, create a new user account for Anaconda Repository named “anaconda-server,” and switch to this new account:

   ```
   sudo useradd -m anaconda-server
   ```

   NOTE: The anaconda-server user is the default for installing Repository. Any username can be used, but using the root user is discouraged.

2. Create a Repository package storage directory:

   ```
   sudo mkdir -m 0770 -p /opt/anaconda-server/package-storage
   ```

3. Assign ownership of this directory to the anaconda-server user:

   ```
   sudo chown -R anaconda-server:anaconda-server /opt/anaconda-server
   ```

4. Switch to the Repository administrator account:

   ```
   sudo su - anaconda-server
   ```

3. Install Repository

Download the installer

Download the Repository installer from the download URL provided by your support representative:

```
curl "$INSTALLER_URL" > anaconda_repository.sh
```

Install Repository, following the prompts in the installation routine:

```
bash anaconda_repository.sh
```

1. Review and accept the license terms:

   ```
   Welcome to Anaconda Repository 2.33 (by Anaconda, Inc.)
   In order to continue the installation process, please review the license␣
   ↪agreement.
   Please, press ENTER to continue.
   ```

2. Once you have reviewed the license terms, approve them by typing `yes`:

   ```
   Do you approve the license terms? [yes|no] yes
   ```

3. Accept the default location or specify an alternative:

   ```
   anaconda_repository will now be installed into this location:
   /home/anaconda-server/repo  -Press ENTER to confirm the location
   -Press CTRL-C to abort the installation
   -Or specify a different location below
    [/home/anaconda-server/repo] >>> /home/anaconda-server/repo" [Press ENTER]
     PREFIX=/home/anaconda-server/repo
     installing: python-2.7.11-0
     ...
   ```

(continues on next page)

(continued from previous page)

```
  Python 2.7.11 :: Anaconda, Inc.
  creating default environment... installation finished.
```

4. At the end of the installation routine, update the anaconda-server user's path—prepending `/home/anaconda-server/repo`—by answering "yes" at the prompt to add the install location to your path:

```
Do you wish the installer to prepend the anaconda_repository install location to␣
↪PATH in your /home/anaconda-server/.bashrc ? [yes|no]
```

5. Type `yes` and press ENTER.

6. For the new path changes to take effect, source your `.bashrc`:

```
source ~/.bashrc
```

4. Configure Repository

1. Initialize the web server and indicate the filepath for the package storage location:

```
anaconda-server-config --init
anaconda-server-config --set fs_storage_root /opt/anaconda-server/package-storage
```

   NOTE: The location for file storage can be any location owned by the anaconda-server user that you created in section 2 above.

   NOTE: As of Repository 2.33.8, the `fs_storage_root` configuration setting is mandatory for local filesystem storage and the Repository server will not run without it.

2. Configure the connection to your MongoDB database:

```
anaconda-server-config --set MONGO_URL mongodb://localhost
```

   NOTE: You may also *configure an external MongoDB database*.

3. If you are not using LDAP or Kerberos authentication, create an initial superuser account for Repository. Set the environment variable `USER_PASSWORD` with the desired password for the initial user. Then run:

```
anaconda-server-create-user --username "superuser" --email "your@email.com" --
↪superuser
```

   NOTE: Replace `superuser` with a username of your choice and `your@email.com` with an email address where you wish to receive system email notifications.

   NOTE: To ensure the bash shell does not process any of the characters in this password, limit the password to letters and numbers, with no punctuation. After setup, you can change the password in the web UI.

4. Initialize the Repository database:

```
anaconda-server-db-setup --execute
```

   NOTE: The above command is also run when upgrading Repository. Upgrade and then run:

```
anaconda-server-db-setup --execute
```

5. Restart the server.

NOTE: More configuration options can be controlled with one or more .yaml configuration files. Repository reads configuration files in this order:

1. From /etc/anaconda-server/*.yaml.
2. From $PREFIX/etc/anaconda-server/*.yaml.
3. From the path specified in the environment variable ANACONDA_SERVER_CONFIG, if it is set and the command line argument --config-file was not used.
4. From the path specified in the command line argument --config-file, if it was used.

All configuration is merged, and options from files read earlier are overwritten by files read later. If there are multiple files in the same directory, they are read in alphabetical order.

5. Set up automatic restart on reboot, fail or error

1. Run the anaconda-server-install-supervisord-config.sh script to configure supervisord management of the Anaconda server and worker processes:

   ```
   anaconda-server-install-supervisord-config.sh
   ```

   This will generate the /home/anaconda-server/repo/etc/supervisord.conf file and add a crontab rule to restart supervisor after each reboot.

   It will also create the folder /home/anaconda-server/repo/etc/supervisord/conf.d/ where you can add .conf files with custom configuration.

   NOTE: If you don’t want to include the crontab rule, use the --no-crontab option when running the script.

   If an error message says that the user is disallowed from using cron and could not add the crontab rule, you can add it manually with sudo. Edit the crontab file:

   ```
   sudo crontab -e -u anaconda-server
   ```

   When the file is open for editing, add this entry:

   ```
   @reboot /home/anaconda-server/repo/bin/supervisord
   ```

2. Verify that the server is running:

   ```
   supervisorctl status
   ```

   If installed correctly, you see:

   ```
   anaconda-server RUNNING   pid 10831, uptime 0:00:05
   ```

3. View the log file at:

   ```
   $PREFIX/var/log/anaconda-server/application.log
   ```

6. Start and log in to Repository

1. Open your browser and log into Repository by visiting http://your.anaconda.repository:8080/ using the superuser account you created in section 4 above.

   NOTE: Replace your.anaconda.repository with the IP address or domain name of your repository.

2. If you are using LDAP or Kerberos authentication, modify your user account to be a superuser.

   EXAMPLE: If your user account is "jsmith":

   ```
   anaconda-server-admin set-superuser "jsmith"
   ```

NOTE: See *Troubleshooting* if you have issues starting the repo server.

7. Client configuration

Follow the *Configuring Anaconda Client* instructions so you can use one or more clients to communicate with the server.

8. Install the Repository license

1. In your browser, go to `http://your.anaconda.repository:8080`. Follow the onscreen instructions to upload the license file that you received in an email from your sales representative.

   NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

   Contact your sales representative or support representative if you cannot find or have any questions about your license.

2. After uploading the license file, you will see the login page. Log in using the superuser user and password that you created in section 4 above.

TIP: You can view the current license information and upload a new license file by visiting the URL `http://your.anaconda.repository:8080/admin/license`.

NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

Alternatively, you can install the license by copying the license file directly into the `/home/anaconda-server/.continuum` directory.

9. OPTIONAL: Mirror installers for Anaconda and Miniconda

Miniconda and Anaconda installers can be served by Repository via the static directory located at `/home/anaconda-server/repo/opt/anaconda-server/installers`. To serve up the latest installers for each platform, download them to this directory.

Define the URL for miniconda installers:

```
URL="https://repo.continuum.io/miniconda/"
```

The `Miniconda*latest*.sh` always point to the latest Miniconda installers. Either these can be mirrored or the ones with the latest version number can be mirrored. Define the list of installers to mirror:

```
versions="Miniconda2-4.5.4-Linux-ppc64le.sh
          Miniconda2-4.5.4-Linux-x86.sh
          Miniconda2-4.5.4-Linux-x86_64.sh
          Miniconda2-4.5.4-MacOSX-x86_64.pkg
          Miniconda2-4.5.4-MacOSX-x86_64.sh
          Miniconda2-4.5.4-Windows-x86.exe
          Miniconda2-4.5.4-Windows-x86_64.exe
          Miniconda3-4.5.4-Linux-ppc64le.sh
          Miniconda3-4.5.4-Linux-x86.sh
```

(continues on next page)

(continued from previous page)

```
          Miniconda3-4.5.4-Linux-x86_64.sh
          Miniconda3-4.5.4-MacOSX-x86_64.pkg
          Miniconda3-4.5.4-MacOSX-x86_64.sh
          Miniconda3-4.5.4-Windows-x86.exe
          Miniconda3-4.5.4-Windows-x86_64.exe"

# miniconda installers
pushd /home/anaconda-server/repo/opt/anaconda-server/installers

for installer in $versions
do
    curl -O $URL$installer
done
```

Define the URL for Anaconda installers:

```
URL="https://repo.continuum.io/archive/"
```

Define the anaconda version to mirror.

EXAMPLE: To mirror version 5.2.0:

```
versions="Anaconda3-5.2.0-Linux-ppc64le.sh
          Anaconda3-5.2.0-Linux-x86.sh
          Anaconda3-5.2.0-Linux-x86_64.sh
          Anaconda3-5.2.0-MacOSX-x86_64.pkg
          Anaconda3-5.2.0-MacOSX-x86_64.sh
          Anaconda3-5.2.0-Windows-x86.exe
          Anaconda3-5.2.0-Windows-x86_64.exe
          Anaconda2-5.2.0-Linux-ppc64le.sh
          Anaconda2-5.2.0-Linux-x86.sh
          Anaconda2-5.2.0-Linux-x86_64.sh
          Anaconda2-5.2.0-MacOSX-x86_64.pkg
          Anaconda2-5.2.0-MacOSX-x86_64.sh
          Anaconda2-5.2.0-Windows-x86.exe
          Anaconda2-5.2.0-Windows-x86_64.exe"

# miniconda installers
pushd /home/anaconda-server/repo/opt/anaconda-server/installers

for installer in $versions
do
    curl -O $URL$installer
done

popd
```

Users can download the installers using curl from the following URL:

```
# Fill in server name, port, and specific installer for your platform
curl -s -O http://your.anaconda.repository:8080/downloads/Miniconda-latest-Linux-x86_
↪64.sh
```

NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

10. Mirror Anaconda Cloud

The final step is to mirror the packages from a subset of channels on Anaconda Cloud to the local Repository. The channels to mirror are as follows:

Channel	Description
anaconda	Default anaconda channel containing all packages built and supported by Anaconda, Inc. Also contains custom packages.
r	If you would like conda packages for *r*, mirror this channel. It is typically done under an *r* account.
wakari anaconda-nb-extensions	If the local Repository will be used by Anaconda Enterprise Notebooks the recommended method is to mirror these channels under the *wakari* account.
anaconda-adam	Anaconda-adam is used to manage the environments on a cluster. If you plan to use anaconda-scale for cluster management, mirror the anaconda-adam packages.
msys2	msys2 is required by quite a few windows packages. See http://www.msys2.org/

The packages will be mirrored to the package store defined by the fs_storage_root key as described in section 4 above.

Mirror Anaconda

Mirror the Anaconda channel from Anaconda Cloud:

```
anaconda-server-sync-conda
```

NOTE: Due to the size of the Cloud repository and depending on the available internet bandwidth, the mirroring process can take hours.

Mirroring an Anaconda repository contains documentation and advanced `yaml` configuration for mirroring other channels.

Installing on an air gapped system

These instructions are for installation on air gapped systems or other machines that do not have access to the internet. The air gap archives contain installers, dependencies and packages to mirror.

Before you start

Your server must meet the requirements for hardware, software, security and network. Please review and verify that you have met all *system requirements* before beginning your installation.

Download the installers archive and the appropriate mirrors archive for your needs. The *Air gap archives* page lists the archives and their contents.

NOTE: These installation instructions assume the air gap media is available on the target server at $INSTALLER_PATH.

EXAMPLE:

```
tar xf <installer-archive> -C /installer/
export INSTALLER_PATH=/installer/anaconda-enterprise-`date +%Y-%m-%d`
```

Also download and expand the archive of conda packages you plan to mirror. These instructions assume packages are expanded to $INSTALLER_PATH:

```
tar xf <archive-of-pkgs-to-mirror> -C /installer/
export MIRRORS_ARCHIVE=/installer/repo-mirrors-`date +%Y-%m-%d`
```

Air gap archives

This section provides information about where to get the air gap archives and their contents.

The air gap archives are generated monthly, generally on the 1st of each month. Monthly archives are hosted at http://airgap.demo.continuum.io/ organized in folders by date.

Installers Archive

All the installers and the latest Miniconda and Anaconda installers for all platforms are in the archive titled:

```
anaconda-enterprise-`date +%Y-%m-%d`.tar
```

The archive size is about 14 GB. It contains everything to install Anaconda Repository, Anaconda Enterprise Notebooks, Anaconda Adam and Anaconda Scale.

The archive contains:

Contents	Description
aen-*.sh	anaconda-enterprise-notebooks server, gateway, compute installers
anaconda_repository*.sh	anaconda-repository installer
adam-installer*.sh	adam installer
conda/	latest version of Miniconda and Anaconda for all platforms
rpms6x/	dependencies for installing on RHEL-6x/CentOS-6x
rpms7x/	dependencies for installing on RHEL-7x/CentOS-7x

Mirror archives

In addition, the `anaconda-server-sync-conda` subdirectory contains mirror archives. These are platform-specific conda packages that must be mirrored after AE-Repo is installed. If you only need packages for a subset of platforms, download the platform-based installers as they will be much smaller in size.

Each component has an md5 file and a list file which are both small and included for convenience.

Tarball	Contents	Size
repo-mirrors-*date +%Y-%m-%d*.tar	All AE-channels for all platforms	160 GB
x64-repo-mirrors-+*%Y-%m-%d*.tar	x64 conda packages for all AE-channels	100 GB
linux-64-pkgs.tar	conda packages for linux-64 for all AE-channels	45 GB
win-64-pkgs.tar	conda packages for win-64	30 GB
osx-64-pkgs.tar	conda packages for osx-64	30 GB

NOTE: The archives contain packages for channels: Anaconda, R, Adam, Wakari. The anaconda-nb-extensions packages are in the anaconda-nb-extensions channel.

1. Install MongoDB 2.6

Change the directory to the appropriate `rpms*` directory to find dependencies:

```
cd $INSTALLER_PATH/rpms*x/
```

MongoDB for Redhat and CentOS 7

1. Install MongoDB:

   ```
   sudo yum install -y mongodb-org*
   ```

2. Start MongoDB:

   ```
   sudo systemctl start mongod
   ```

3. Verify that MongoDB is running:

   ```
   $ sudo systemctl status mongod
   mongodb (pid 17258) is running...
   ```

MongoDB for Redhat and CentOS 6.7+

1. Install MongoDB:

```
sudo yum install -y mongodb-org*
```

2. Start MongoDB:

```
sudo /etc/init.d/mongod start
```

3. Verify that MongoDB is running:

```
$ sudo /etc/init.d/mongod status
mongodb (pid 17258) is running...
```

MongoDB for Ubuntu 12.04+

1. Install MongoDB:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10

echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' |␣
↪sudo tee /etc/apt/sources.list.d/mongodb.list

sudo apt-get update

sudo apt-get install -y mongodb-org=2.6.9 mongodb-org-server=2.6.9 mongodb-org-
↪shell=2.6.9 mongodb-org-mongos=2.6.9 mongodb-org-tools=2.6.9
```

   NOTE: If you do not specify a version, such as 2.6.9, `apt-get` installs the latest stable version, which is 3.x.

2. Start MongoDB:

```
sudo /etc/init.d/mongod start
```

   Verify that MongoDB is running:

```
$ sudo /etc/init.d/mongod status
mongodb (pid 17258) is running...
```

   You receive verification that MongoDB is running:

```
start: Job is already running: mongodb
```

Additional MongoDB resources

For additional MongoDB installation information see https://docs.mongodb.org/manual/.

2. Create the Repository administrator account

1. In a Terminal window, create a new user account for Anaconda Repository named “anaconda-server,” and switch to this new account:

```
sudo useradd -m anaconda-server
```

NOTE: The anaconda-server user is the default for installing Repository. Any username can be used, but using the root user is discouraged.

2. Create a Repository package storage directory:

```
sudo mkdir -m 0770 -p /opt/anaconda-server/package-storage
```

3. Assign ownership of this directory to the anaconda-server user:

```
sudo chown -R anaconda-server:anaconda-server /opt/anaconda-server
```

4. Switch to the Repository administrator account:

```
sudo su - anaconda-server
```

3. Install Repository

Install Repository, following the prompts in the installation routine:

```
bash $INSTALLER_PATH/anaconda_repository-*-linux-64.sh
```

NOTE: Path should have only one installer that is for the latest stable version of Repository.

1. Review and accept the license terms:

```
Welcome to Anaconda Repository 2.33 (by Anaconda, Inc.)
In order to continue the installation process, please review the license␣
↪agreement.
Please, press ENTER to continue.
```

2. Once you have reviewed the license terms, approve them by typing `yes`:

```
Do you approve the license terms? [yes|no] yes
```

3. Accept the default location or specify an alternative:

```
anaconda_repository will now be installed into this location:
/home/anaconda-server/repo  -Press ENTER to confirm the location
-Press CTRL-C to abort the installation
-Or specify a different location below
 [/home/anaconda-server/repo] >>> /home/anaconda-server/repo" [Press ENTER]
  PREFIX=/home/anaconda-server/repo
  installing: python-2.7.11-0
  ...
  Python 2.7.11 :: Anaconda, Inc.
  creating default environment... installation finished.
```

4. At the end of the installation routine, update the anaconda-server user's path—prepending `/home/anaconda-server/repo`—by answering "yes" at the prompt to add the install location to your path:

```
Do you wish the installer to prepend the anaconda_repository install location to␣
↪PATH in your /home/anaconda-server/.bashrc ? [yes|no]
```

5. Type `yes` and press ENTER.

6. For the new path changes to take effect, source your `.bashrc`:

```
source ~/.bashrc
```

4. Configure Repository

1. Initialize the web server and indicate the filepath for the package storage location:

```
anaconda-server-config --init
anaconda-server-config --set fs_storage_root /opt/anaconda-server/package-storage
```

   NOTE: The location for file storage can be any location owned by the anaconda-server user that you created in section 2 above.

   NOTE: As of Repository 2.33.8, the `fs_storage_root` configuration setting is mandatory for local filesystem storage and the Repository server will not run without it.

2. Configure the connection to your MongoDB database:

```
anaconda-server-config --set MONGO_URL mongodb://localhost
```

   NOTE: You may also *configure an external MongoDB database*.

3. If you are not using LDAP or Kerberos authentication, create an initial superuser account for Repository. Set the environment variable `USER_PASSWORD` with the desired password for the initial user. Then run:

```
anaconda-server-create-user --username "superuser" --email "your@email.com" --
↪superuser
```

   NOTE: Replace `superuser` with a username of your choice and `your@email.com` with an email address where you wish to receive system email notifications.

   NOTE: To ensure the bash shell does not process any of the characters in this password, limit the password to letters and numbers, with no punctuation. After setup, you can change the password in the web UI.

4. Initialize the Repository database:

```
anaconda-server-db-setup --execute
```

   NOTE: The above command is also run when upgrading Repository. Upgrade and then run:

```
anaconda-server-db-setup --execute
```

5. Restart the server.

NOTE: More configuration options can be controlled with one or more `.yaml` configuration files. Repository reads configuration files in this order:

1. From `/etc/anaconda-server/*.yaml`.
2. From `$PREFIX/etc/anaconda-server/*.yaml`.
3. From the path specified in the environment variable ANACONDA_SERVER_CONFIG, if it is set and the command line argument `--config-file` was not used.
4. From the path specified in the command line argument `--config-file`, if it was used.

All configuration is merged, and options from files read earlier are overwritten by files read later. If there are multiple files in the same directory, they are read in alphabetical order.

5. Set up automatic restart on reboot, fail or error

1. Run the `anaconda-server-install-supervisord-config.sh` script to configure supervisord management of the Anaconda server and worker processes:

   ```
   anaconda-server-install-supervisord-config.sh
   ```

   This will generate the `/home/anaconda-server/repo/etc/supervisord.conf` file and add a crontab rule to restart supervisor after each reboot.

   It will also create the folder `/home/anaconda-server/repo/etc/supervisord/conf.d/` where you can add `.conf` files with custom configuration.

   NOTE: If you don't want to include the crontab rule, use the `--no-crontab` option when running the script.

   If an error message says that the user is disallowed from using cron and could not add the crontab rule, you can add it manually with sudo. Edit the crontab file:

   ```
   sudo crontab -e -u anaconda-server
   ```

   When the file is open for editing, add this entry:

   ```
   @reboot /home/anaconda-server/repo/bin/supervisord
   ```

2. Verify that the server is running:

   ```
   supervisorctl status
   ```

   If installed correctly, you see:

   ```
   anaconda-server RUNNING   pid 10831, uptime 0:00:05
   ```

3. View the log file at:

   ```
   $PREFIX/var/log/anaconda-server/application.log
   ```

6. Start and log in to Repository

1. Open your browser and log into Repository by visiting `http://your.anaconda.repository:8080/` using the superuser account you created in section 4 above.

   NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

2. If you are using LDAP or Kerberos authentication, modify your user account to be a superuser.

   EXAMPLE: If your user account is "jsmith":

   ```
   anaconda-server-admin set-superuser "jsmith"
   ```

NOTE: See *Troubleshooting* if you have issues starting the repo server.

7. Client configuration

Follow the *Configuring Anaconda Client* instructions so you can use one or more clients to communicate with the server.

8. Install the Repository license

1. In your browser, go to `http://your.anaconda.repository:8080`. Follow the onscreen instructions to upload the license file that you received in an email from your sales representative.

   NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

   Contact your sales representative or support representative if you cannot find or have any questions about your license.

2. After uploading the license file, you will see the login page. Log in using the superuser user and password that you created in section 4 above.

TIP: You can view the current license information and upload a new license file by visiting the URL `http://your.anaconda.repository:8080/admin/license`.

NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

Alternatively, you can install the license by copying the license file directly into the `/home/anaconda-server/.continuum` directory.

9. OPTIONAL: Mirror installers for Anaconda and Miniconda

Miniconda and Anaconda installers can be served by Repository via the static directory located at `/home/anaconda-server/repo/opt/anaconda-server/installers`. To serve up the latest installers for each platform, copy them from your *air gap archive* to this directory.:

```
cp Miniconda-latest-Linux-x86_64.sh /home/anaconda-server/repo/opt/anaconda-server/
↪installers
```

Replace `Miniconda-latest-Linux-x86_64.sh` with your Anaconda or Miniconda installer name.

NOTE: Air gap archive only contains the latest version of both Miniconda and Anaconda installers.

Users can download the installers using curl from the following URL:

```
# Fill in server name, port, and specific installer for your platform
curl -s -O http://your.anaconda.repository:8080/downloads/Miniconda-latest-Linux-x86_
↪64.sh
```

NOTE: Replace `your.anaconda.repository` with the IP address or domain name of your repository.

10. Mirror Anaconda Cloud

The final step is to mirror the packages from a subset of channels on Anaconda Cloud to the local Repository. The channels to mirror are as follows:

Channel	Description
anaconda	Default anaconda channel containing all packages built and supported by Anaconda, Inc. Also contains custom packages.
r	If you would like conda packages for *r*, mirror this channel. It is typically done under an *r* account.
wakari anaconda-nb-extensions	If the local Repository will be used by Anaconda Enterprise Notebooks the recommended method is to mirror these channels under the *wakari* account.
anaconda-adam	Anaconda-adam is used to manage the environments on a cluster. If you plan to use anaconda-scale for cluster management, mirror the anaconda-adam packages.
msys2	msys2 is required by quite a few windows packages. See http://www.msys2.org/

The packages will be mirrored to the package store defined by the fs_storage_root key as described in section 4 above.

Mirror Anaconda

Since we are mirroring from a local file system, some additional configuration is necessary. The steps are the same for each channel:

1. Create a mirror configuration `yaml` typically stored in `$PREFIX/etc/anaconda-server/mirror/`.
2. *Customize your mirror*. An example is if you only need to mirror packages for a subset of platforms. By default, it mirrors all packages found in the channels linux-64, osx-64, win-64, win-32 and linux-32.
3. Invoke the mirror command by pointing it to the config file:

```
echo "channels:" > ~/repo/etc/anaconda-server/mirror/conda.yaml
echo " - file://$MIRRORS_ARCHIVE/anaconda-suite/pkgs" >> \
       ~/repo/etc/anaconda-server/mirror/conda.yaml
```

4. Mirror the default Anaconda packages:

```
anaconda-server-sync-conda --mirror-config ~/repo/etc/anaconda-server/mirror/
↪conda.yaml
```

Mirroring an Anaconda repository contains documentation and advanced `yaml config` for mirroring other channels.

Configuration

Enabling HTTPS

Before you begin, purchase an SSL certificate and download the SSL `*.cert` file and SSL `*.key` file.

NOTE: If security is not an issue, for testing, you may set up a self-signed SSL certificate. For more information, see http://www.selfsignedcertificate.com/.

1. Save the SSL `*.cert` file and an SSL `*.key` file in your home directory.
2. Configure the server to use those keys and the correct ports:

```
anaconda-server-config --set ssl_options.keyfile ~/localhost.key
anaconda-server-config --set ssl_options.certfile ~/localhost.cert
anaconda-server-config --set port 8443
```

3. Restart your server for the changes to take effect:

```
supervisorctl restart all
```

4. To test, navigate to the site using `https` in the address bar.

   NOTE: If you use a self-signed SSL certificate, your web browser issues a warning that the website certificate cannot be verified.

Next, configure your client side tools `conda` and `anaconda-client` to pull packages from the local repo by setting the `ssl_verify` flags.

Configure conda

If your conda client is configured to point to this local repo, update the configuration file `.condarc` to contain the `ssl_verify` flag. If you're using a self-signed certificate, configure the `ssl_verify` flag in `.condarc` to point to the root CA used to sign the Anaconda Enterprise Repository server certificate.

Configure anaconda-client

If you're using `anaconda-client` to connect to Anaconda Enterprise Repository with the command line, set the `ssl_verify` flag.

Use `anaconda config --files` to find the `anaconda-client` configuration files.

SEE the *command reference* for updating the client sites configuration for `anaconda-client`.

Enabling email and SMTP

To send emails such as password reset emails, Repository must have the *email settings* configured.

Configuring a standard or alternate port

The easiest way to enable clients to access a Repository server on standard ports is to configure the server to redirect traffic received on standard HTTP port 80 to the standard Repository HTTP port 8080:

```
sudo iptables -t nat -F
sudo iptables -t nat -A OUTPUT -d localhost -p tcp --dport 80 -j REDIRECT --to-ports␣
↪8080
sudo iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

HTTPS

To use HTTPS, redirect traffic from standard HTTPS port 443 to standard Repository HTTPS port 8443:

```
sudo iptables -t nat -A OUTPUT -d localhost -p tcp --dport 443 -j REDIRECT --to-ports␣
↪8443
sudo iptables -t nat -I PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 8443
```

NOTE: See also *Enabling HTTPS*.

Alternate port

To run Repository on a port other than the standard port 8080:

1. Modify the usual instructions by adjusting the port numbers in your *iptables configuration*.
2. Specify the correct port in your `supervisord.conf` file.

Adjusting IPTables to accept requests on port 80

Enable clients to access a Repository on standard ports by configuring the server to redirect traffic received on standard HTTP port 80 to the standard Repository HTTP port 8080.

NOTE: These commands assume the default state of IPTables, which is `on` and allowing inbound SSH access on port 22. This is the factory default state for CentOS 6.7. If this default has been changed, you can reset it:

```
sudo iptables -L
```

CAUTION: Mistakes with IPTables rules can render a remote machine inaccessible.

1. Allow inbound access to tcp port 80:

   ```
   sudo iptables -I INPUT -i eth0 -p tcp --dport 80 -m comment --comment "# Anaconda␣
   ↪Repo #" -j ACCEPT
   ```

2. Allow inbound access to tcp port 8080:

   ```
   sudo iptables -I INPUT -i eth0 -p tcp --dport 8080 -m comment --comment "#␣
   ↪Anaconda Repo #" -j ACCEPT
   ```

3. Redirect inbound requests to port 80 to port 8080:

   ```
   sudo iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -m comment --comment
   ↪"# Anaconda Repo #" -j REDIRECT --to-port 8080
   ```

4. Display the current IPTables rules:

   ```
   iptables -L -n
   Chain INPUT (policy ACCEPT)
   target     prot opt source               destination
   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:8080 /* #␣
   ↪Anaconda Repo # */
   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 /* #␣
   ↪Anaconda Repo # */
   ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state RELATED,
   ↪ESTABLISHED
   ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
   ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp dpt:22
   ```

(continues on next page)

(continued from previous page)

```
REJECT     all  --  0.0.0.0/0            0.0.0.0/0            reject-with icmp-
↪host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  0.0.0.0/0            0.0.0.0/0            reject-with icmp-
↪host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

NOTE: The PREROUTING (nat) IPTables chain is not displayed by default. To display the chain:

```
iptables -L -n -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 /* #␣
↪Anaconda Repo # */ redir ports 8080

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

5. Save the running IPTables configuration to `/etc/sysconfig/iptables`:

```
sudo service iptables save
```

Connecting to an existing MongoDB database

If you already have a MongoDB server running, you can connect to it by setting the MONGO_URL configuration variable:

```
anaconda-server-config --set MONGO_URL 'mongodb://<hostname>'
```

For more information, see the MongoDB Connection String URI Format manual.

See also *Configuring MongoDB authentication*.

Configuring MongoDB authentication

By default, MongoDB does not require a username or password to access or modify the database. We recommend enabling and configuring mandatory authentication.

1. Open a MongoDB shell:

```
mongo
```

2. Repository requires read/write access to the database `binstar`. Enter the following commands into the MongoDB shell to create an administrative user and a service user:

```
use admin
```

3. Create an administrative user to manage database users:

```
db.createUser({user:'siteUserAdmin', pwd: '<secure password #1>', roles:[
↪'userAdminAnyDatabase']})
```

4. Authorize as that user to verify the password:

```
db.auth('siteUserAdmin', '<secure password #1>')
```

5. Create a service user for Repository:

```
db.createUser({user:'anaconda', pwd: '<secure password #2>', roles:[{db:'binstar',
↪ role:'readWrite'}]})
```

6. Enable mandatory authentication in MongoDB:

   - If you are using the legacy MongoDB configuration format, add the auth key to `/etc/mongod.conf`:

   ```
   auth=true
   ```

   - If you are using the current MongoDB configuration format, add the security.authorization key to `/etc/mongod.conf`:

   ```
   security:
       authorization: enabled
   ```

7. Restart MongoDB to reload the configuration:

```
sudo service mongod restart
```

8. Edit the Repository configuration file and set the `MONGO_URL` parameter to `mongodb://<username>:<password>@<hostname>`.

   After editing the configuration file, restart Repository for the changes to take effect.

9. Edit the Repository configuration file and set the `MONGO_URL` parameter to `mongodb://<username>:<password>@<hostname>`.

   After editing the configuration file, restart Repository for the changes to take effect.

NOTE: For more information about MongoDB authentication and authorization, see https://docs.mongodb.com/v2.6/core/authentication/ and https://docs.mongodb.com/v2.6/core/authorization/.

Whitelisting or blacklisting packages

Sometimes you do not want to replicate all the packages from Repository into your mirror. The `anaconda-server-sync-conda` tool includes whitelist/blacklist functionality to manipulate your list of mirrored packages in a variety of ways.

A mirror config file can be specified when you run `anaconda-server-sync-conda` with the flag `--mirror-config=FILEPATH` and replace FILEPATH with the path to your config file.

NOTE: Configuration files are `yaml` files.

To customize your distribution, you have the following options:

- remote_url: Repository mirrors packages from this source URL.
- mirror_dir: Repository stores packages in this directory on the machine where the script is executed.

- platforms: Repository mirrors packages for these platforms.
- license_blacklist: Repository omits packages with these licenses.
- blacklist: Repository omits these packages.
- whitelist: Repository always mirrors these packages.

TIP: You do not need to set up every option manually. If you only want to adjust one or two options, that is allowed. Untouched options remain defined by the default setting.

EXAMPLE: The following example only selects packages that are available for linux-32 and linux-64 platforms. Win-32 or win-64 packages are not mirrored at all:

```
mirror_dir: /opt/anaconda-server/package-storage
platforms:
  - linux-32
  - linux-64
license_blacklist: GPL
whitelist:
  - distribute
  - conda
blacklist:
  - flask
  - readline
```

The step-by-step algorithm that is used by cas-mirror to create the ultimate list of packages to mirror follows this order:

1. Get a full list of packages from `default_url`.
2. If the platforms option is present, only those packages available to the platforms listed here are left on the list.
3. If license_blacklist is present, then all the packages subject to any of the licenses mentioned here are removed from the list. See the *list of license families that can be blacklisted*.
4. If blacklist is present, then all member packages explicitly mentioned here are removed from the list.
5. If whitelist is present, then those assigned member packages are added to the list. The whitelist option overrides license_blacklist and blacklist, so that a package listed here is mirrored even when under a GPL license or if it appears in the blacklist option.

After performing all of the above actions sequentially, the script produces the ultimate list of packages that are mirrored.

Securing user-created content

To prevent cross-site scripting attacks (XSS), user content—such as Jupyter Notebooks—can be served from a separate domain.

To enable this:

1. Configure the project to use a separate content domain:

   ```
   anaconda-server-config --set SERVER_NAME your.anaconda.repository
   anaconda-server-config --set USER_CONTENT_DOMAIN your.usercontent.server
   ```

NOTE: Replace `your.anaconda.repository` and `usercontent.your.anaconda.repository` with the respective server IP address or domain name.

1. If your user content domain is a subdomain of your Repository domain, you must also configure the session cookie to only send to the root domain:

```
anaconda-server-config --set SERVER_NAME your.anaconda.repository
anaconda-server-config --set USER_CONTENT_DOMAIN usercontent.your.anaconda.
↪repository
anaconda-server-config --set SESSION_COOKIE_DOMAIN your.anaconda.repository
```

NOTE: Replace `your.anaconda.repository` and `usercontent.your.anaconda.repository` with the respective server IP address or domain name.

Configuring Repository to use LDAP

To enable Lightweight Directory Access Protocol (LDAP) support:

1. Open the Repository configuration file `$PREFIX/etc/anaconda-server/config.yaml` and add the following configuration:

```
account_names_filter: false
USER_REGEX: ^[a-z0-9_][a-z0-9_-.]+$
LDAP:
  # Replace with company LDAP server
  URI: 'ldap://<ldap.company.com>'

  # Replace <uid=%(username)s,ou=People,dc=company,dc=com> with your company␣
↪specific LDAP Bind/Base DN
  # Bind directly to this Base DN.
  BIND_DN: '<uid=%(username)s,ou=People,dc=company,dc=com>'

  # Map LDAP keys into application specific keys
  KEY_MAP:
      name: 'cn'
      company: 'o'
      location:'l'
      email: 'mail'
```

2. When switching authentication to LDAP, the admin account is lost, so you need to add your admin account again:

```
anaconda-server-admin set-superuser "jsmith"
```

3. Run the `flask-ldap-login-check` command to verify LDAP connectivity:

```
flask-ldap-login-check binstar.wsgi:app --username 'jsmith' --password 'abc123DEF'
```

   NOTE: Replace `jsmith` and `abc123DEF` with your LDAP username and password.

4. To apply the changes, restart the Repository server:

```
supervisorctl restart all
```

5. Open a new browser window and navigate to your local Repository installation:

```
http://your.anaconda.repository
```

   NOTE: Replace `your.anaconda.repository` with your Repository server IP address or domain name.

6. Log in using your LDAP credentials.

7. Optional. You may set an LDAP network timeout in seconds with the options OPT_NETWORK_TIMEOUT and OPT_TIMEOUT. The default value is 0, meaning no timeout.

   For example, to set the timeout to 60 seconds, add this block to the LDAP settings in your configuration file:

   ```
   OPTIONS:
      OPT_NETWORK_TIMEOUT: 60
      OPT_TIMEOUT: 60
   ```

Configuring Repository to use Active Directory

Microsoft Active Directory is a server program that provides directory services and uses the open industry standard Lightweight Directory Access Protocol (LDAP).

To enable Active Directory support:

1. Open the Repository configuration file `$PREFIX/etc/anaconda-server/config.yaml` and add the following configuration:

   ```
   account_names_filter: false
   USER_REGEX: ^[a-z0-9_][a-z0-9_-.]+$
   LDAP:
       # Replace with company LDAP server
       'URI': 'ldap://<ldap.server.url>'

       # This BIND_DN/BIND_PASSWORD default to '', this is shown here for
       # demonstrative purposes. To enable Authorized Bind, insert the AD
       # BIND_DN and BIND_AUTH password for and authorized AD user.
       #
       #e.g. 'BIND_DN': '<cn=Authorized User,cn=users,dc=company,dc=local>'
       #e.g. 'BIND_AUTH': '<AuthUsrPassword>'

       # The values '' perform an anonymous bind so we may use search/bind method
       BIND_DN: ''
       BIND_AUTH: ''

       # Adding the USER_SEARCH field tells the flask-ldap-login that we
       # are using the search/bind method
       USER_SEARCH:
           base: <cn=users,dc=company,dc=local>
           filter: sAMAccountName=%(username)s

       # Map ldap keys into application specific keys
       KEY_MAP:
           name: 'cn'
           company: 'o'
           location: 'l'
           email: 'userPrincipalName'
   ```

2. To apply the changes, restart the Repository server:

   ```
   supervisorctl restart all
   ```

3. Run the `flask-ldap-login-check` command to verify Active Directory connectivity:

   ```
   flask-ldap-login-check binstar.wsgi:app --username 'jsmith' --password 'abc123DEF'
   ```

NOTE: Replace jsmith and abc123DEF with your Active Directory username and password.

You see a response similar to the following:

```
[anaconda.server] Started Site
Got userdata for jsmith
{'company': None, 'email': None, 'location': None, 'name': 'Jane Smith'}
```

4. Open your browser and navigate to your local Repository installation:

   ```
   http://your.anaconda.repository
   ```

   NOTE: Replace your.anaconda.repository with your Repository IP address or domain name.

5. Log in with Active Directory.

Configuring Repository to use LDAP groups

Repository can be configured to allow synchronizing the membership of organization groups with groups in an LDAP directory. Owners of an organization can select a specific LDAP group as the source of group members.

Once this is enabled, users who sign in to Repository who are members of the LDAP group automatically are granted the permissions of the organization group.

To enable LDAP groups, configure the following:

- Authenticated bind to LDAP. Repository needs to perform searches against the directory to determine the available groups and the membership of those groups.
- A query for Repository to identify the groups in your LDAP directory. For more information, see *GROUP_SEARCH*.

If LDAP synchronization is disabled or the LDAP server is unreachable, the member list at the time is used for the group.

To administer and debug LDAP synchronization, a superuser can visit:

```
http://your.anaconda.repository/admin/ldap
```

NOTE: Replace your.anaconda.repository with your Repository IP address or domain name.

Enabling TLS on LDAP/Active Directory

To enable a secure Transport Layer Security (TLS) connection on LDAP/Active Directory, add the following to the LDAP configuration section of the file $PREFIX/etc/anaconda-server/config.yaml:

```
LDAP:
  ...  # Rest of the LDAP config
  START_TLS: true,
  OPTIONS:
    OPT_PROTOCOL_VERSION: 3
    OPT_X_TLS_DEMAND: true
    OPT_X_TLS_REQUIRE_CERT: 'OPT_X_TLS_NEVER'
    OPT_X_TLS_CACERTFILE: '/path/to/certfile'
```

NOTE: Replace /path/to/certfile with the actual path to certfile.

NOTE: START_TLS is not compatible with LDAPS. When using START_TLS the URI value in the LDAP configuration section must start with ldap://. When using START_TLS the connection starts as a regular connection, and after it has been established it is upgraded to use TLS.

If you're using self-signed certificates you'll need to add OPT_X_TLS_NEWCTX as the **last entry** of the OPTIONS field of the LDAP options:

```
LDAP:
  ...  # Rest of the LDAP config
  START_TLS: true,
  OPTIONS:
    OPT_PROTOCOL_VERSION: 3
    OPT_X_TLS_DEMAND: true
    OPT_X_TLS_REQUIRE_CERT: 'OPT_X_TLS_NEVER'
    OPT_X_TLS_CACERTFILE: '/path/to/certfile'
    OPT_X_TLS_NEWCTX: 0
```

Using LDAP and TLS configuration options

URI

Start by setting URI to point to your server. The value of this setting can be anything that your LDAP library supports. For instance, openldap may allow you to give a comma- or space-separated list of URI values to try in sequence.

BIND_DN

The distinguished name to use when binding to the LDAP server with BIND_AUTH. Use the empty string—the default—for an anonymous bind.

BIND_AUTH

The password to use with BIND_DN.

USER_SEARCH

A dictionary that locates a user in the directory. The dict object must contain the required entries base and filter and may contain the optional entry scope.

- base: The base DN to search.
- filter: Should contain the placeholder %(username)s for the username.
- scope: One of LDAP_SCOPE_BASE, LDAP_SCOPE_ONELEVEL or LDAP_SCOPE_SUBTREE.

EXAMPLE:

```
{'base': 'dc=example,dc=com', 'filter': 'uid=%(username)s'}
```

SUPERUSER_SEARCH

A dict that will determine whether a valid user is a superuser. The dict object must contain the required entries `base` and `filter` and may contain the optional entry `scope`. If the search is successful, then something is returned by the LDAP server, and the user is given superuser permissions.

- base: The base DN to search.
- filter: Should contain the placeholder `%(username)s` for the username.
- scope: One of `LDAP_SCOPE_BASE`, `LDAP_SCOPE_ONELEVEL`, or `LDAP_SCOPE_SUBTREE`.

For example:

```
{'base': 'cn=admin,ou=Groups,dc=example,dc=com', 'filter': 'memberUid=%(username)s'}
```

Notice that this check is done during the login procedure, so even though privileges might have been removed from (or added to) the LDAP server, the user will have to authenticate again to see the changes.

ENABLE_GROUPS

This attribute enables LDAP group synchronization, allowing users to synchronize group membership with an LDAP directory. Defaults to `false`.

EXAMPLE:

```
ENABLE_GROUPS: true
```

GROUP_SEARCH

A dictionary that locates a group in the directory. An LDAP search is performed using the `base` distinguished name and `filter`.

EXAMPLE:

```
GROUP_SEARCH:
    base: dc=example,dc=com
    filter: (objectClass=group)
```

NOTE: Anaconda Repository assumes that the groups' *objectClass* is *groupOfNames* (or a compatible schema). The following LDIF snippet shows an example group instance:

```
dn: cn=Analysts,ou=Anaconda Groups,dc=example,dc=com
cn: Analysts
member: cn=John Doe,ou=Users,dc=example,dc=com
member: cn=Jane Doe,ou=Users,dc=example,dc=com
member: cn=John Q. Public,ou=Users,dc=example,dc=com
member: cn=Guy Incognito,ou=Users,dc=example,dc=com
objectclass: groupOfNames
objectclass: top
```

GROUP_MEMBERS_ATTR

The LDAP attribute on a group object that indicates the users that are members of the group. Defaults to `member`.

EXAMPLE:

```
GROUP_MEMBERS_ATTR: 'member'
```

NOTE: Anaconda Repository assumes that the groups' *objectClass* is *groupOfNames* (or a compatible schema).

REFRESH_INTERVAL

The number of seconds that group membership information from LDAP is used before being fetched from the directory server again. Defaults to `3600`, which is 1 hour.

EXAMPLE:

```
REFRESH_INTERVAL: 600
```

KEY_MAP

This is a dict mapping application context to LDAP. An application may expect user data to be consistent, and not all LDAP setups use the same configuration:

```
'application_key': 'ldap_key'
```

EXAMPLE:

```
KEY_MAP={'name': 'cn', 'company': 'o', 'email': 'mail'}
```

START_TLS

If `true`, each connection to the LDAP server calls `start_tls_s()` to enable TLS encryption over the standard LDAP port. There are a number of configuration options that can be given to OPTIONS that affect the TLS connection. For example, OPT_X_TLS_REQUIRE_CERT can be set to `OPT_X_TLS_NEVER` to disable certificate verification, perhaps to allow self-signed certificates.

OPTIONS

This stores LDAP specific options.

EXAMPLE:

```
LDAP:
    OPTIONS:
        OPT_PROTOCOL_VERSION: 3
        OPT_X_TLS_REQUIRE_CERT: 'OPT_X_TLS_NEVER'
```

TLS—secure LDAP

To enable a secure TLS connection you must set START_TLS to `true`. There are a number of configuration options for OPTIONS that affect the TLS connection.

EXAMPLE: OPT_X_TLS_REQUIRE_CERT set to `OPT_X_TLS_NEVER` disables certificate verification, perhaps to allow self-signed certificates:

```
LDAP:
    START_TLS: true
    OPTIONS:
        OPT_PROTOCOL_VERSION: 3
        OPT_X_TLS_DEMAND: true
        OPT_X_TLS_REQUIRE_CERT: 'OPT_X_TLS_NEVER'
        OPT_X_TLS_CACERTFILE: '/path/to/certfile'
```

Configuring Repository to use Kerberos

Kerberos is an authentication protocol designed to allow nodes communicating over an insecure network to verify identity. Repository can use Kerberos to authenticate users.

The Kerberos protocol uses timestamps to prevent replay attacks on expired credentials, so the Network Time Protocol (NTP) service must be set up and working correctly.

Several aspects of Kerberos rely on name service. Your domain name system (DNS) entries and your hosts must have the correct information. The `hostname` command and the configuration file `/etc/hostname` must reflect the fully-qualified domain name (FQDN) of the machine. The configuration file `/etc/hosts` must include an entry with the FQDN, to allow reverse-DNS lookups to be performed.

To allow clients to authenticate against Anaconda Repository, create a principal for the service with a private key that identifies the service. Create a service principal `HTTP/your.anaconda.repository`, and create the keytab containing this principal to `$PREFIX/etc/anaconda-server/http.keytab`:

```
SERVER_NAME=your.anaconda.repository
```

NOTE: Replace `your.anaconda.repository` with your server IP address or domain name.

If you are using MIT Kerberos:

```
kadmin -q "addprinc HTTP/${SERVER_NAME}"
kadmin -q "ktadd -k $PREFIX/etc/anaconda-server/http.keytab HTTP/${SERVER_NAME}"
chown anaconda-server:anaconda-server $PREFIX/etc/anaconda-server/http.keytab
chmod 600 $PREFIX/etc/anaconda-server/http.keytab
```

If you are using Active Directory:

1. Open Active Directory Users and Computers.
2. Select the Users container.
3. In the **Action** menu, select New, then select User.
4. In the New Object - User dialog, type the user information. In this example, we use `your-anaconda-repository` as the login.
5. In the next dialog, select the options Password never expires and User cannot change password.
6. Right-click on the newly created user, and select Properties.
7. In the Properties dialog, select the **Account** tab, and ensure the Do not require Kerberos preauthentication option is selected.
8. Open an Administrative prompt and run:

   ```
   ktpass -princ HTTP/your.anaconda.repository@YOUR.DOMAIN -out http.keytab -pass "*
   ↪" -mapUser your-anaconda-user@your-anaconda-server -ptype KRB5_NT_PRINCIPAL
   ```

9. Copy the newly created file `http.keytab` to `$PREFIX/etc/anaconda-server/http.keytab` on your Repository server.

To enable Kerberos authentication on Repository, add the configuration options to `$PREFIX/etc/anaconda-server/config.yaml`:

```
AUTH_TYPE: KERBEROS
KRB5_KTNAME: /home/anaconda-server/repo/etc/anaconda-server/http.keytab
```

For a minimal configuration example see *Kerberos-Anaconda Repository setup example*.

Kerberos configuration options

AUTH_TYPE	string	Configures the authentication scheme used for Repository. Set to `KERBEROS` to enable Kerberos authentication. Default: `NATIVE`.
KRB5_KTNAME	string	The file path of the keytab containing the service principal for Repository. Default: `/etc/krb5.keytab`.
KRB5_SERVICE_NAME	string	The service type used to identify the service principal for Repository. `HTTP` in `HTTP/your.anaconda.repository@YOUR.REALM`. Default: `HTTP`.
KRB5_HOSTNAME	string	The hostname used to identify the service principal for Repository. `your.anaconda.repository` in `HTTP/your.anaconda.repository@YOUR.REALM`. Default: the hostname of the machine on which Repository is running.

Kerberos-Anaconda Repository setup example

Kerberos authentication adds a layer of security to Anaconda Repository. The following example show how to set up a minimal working installation with three machines: One running anaconda server, one running the MIT Kerberos Key Distribution Center (KDC), and a client from where we are going to connect to both services.

For this example we assume that both the KDC and Anaconda Repository are already configured and the 3 systems have the Network Time Protocol (NTP) service working.

Initial Setup

All 3 machines are running CentOS 7 but the configurations mentioned here apply for many other Linux distributions. We are going to use the following domain names:

- Anaconda Repository: `anaconda.kerberos.local`
- Kerberos KDC: `kdc.kerberos.local`
- Client: `client.kerberos.local`

Make sure that the information is correct in the configuration files `/etc/hostname` and `/etc/hosts` to allow reverse-DNS lookups.

The name of the Kerberos realm is `KERBEROS.LOCAL`. The 3 machines have the same configuration file `/etc/krb5.conf`:

```
[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log
    default = SYSLOG:NOTICE:DAEMON

[libdefaults]
    dns_lookup_realm = true
    dns_lookup_kdc = true
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true
    rdns = false
    default_realm = KERBEROS.LOCAL
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
    KERBEROS.LOCAL = {
        kdc = kdc.kerberos.local
        admin_server = kdc.kerberos.local
 }

[domain_realm]
    .kerberos.local = KERBEROS.LOCAL
    kerberos.local = KERBEROS.LOCAL
```

On `kdc.kerberos.local` the files `/var/kerberos/krb5kdc/kdc.conf` and `/var/kerberos/krb5kdc/kadm5.acl` should be configured accordingly.

Configure Anaconda Repostiory

At this point Anaconda Repository is up and running, it's installed on `/home/anaconda-server/repo`, the administrator account in this example is `superuser`. To allow authentication we first create a service principal and the keytab containing this principal. This is accomplished running the following commands as root from a terminal on `anaconda.kerberos.local`.

```
kadmin -q "addprinc HTTP/anaconda.kerberos.local"
kadmin -q "ktadd -k /home/anaconda-server/repo/etc/anaconda-server/http.keytab HTTP/
↪anaconda.kerberos.local"
chown anaconda-server:anaconda-server \
   /home/anaconda-server/repo/etc/anaconda-server/http.keytab
chmod 600 /home/anaconda-server/repo/etc/anaconda-server/http.keytab
```

Now edit the configuration file `/home/anaconda-server/repo/etc/anaconda-server/config.yaml` and add the following lines:

```
AUTH_TYPE: KERBEROS
KRB5_KTNAME: /home/anaconda-server/repo/etc/anaconda-server/http.keytab
```

Finally, add the principal for the admin account on the kerberos realm:

```
kadmin -q "addprinc superuser@KERBEROS.LOCAL"
```

Reboot the server for the changes to take effect.

Client Configuration

To log in to Anaconda Repository with Kerberos Authentication, a browser that supports said authentication protocol is necessary. In this example we are using Firefox. Some extra tweaking is required.

- Open Firefox and type **about:config** in the navigation bar, click the confirmation button if necessary to proceed to the configuration page.
- Type **negotiate** in the Search field to filter out the options, double click **network.negotiate-auth.trusted-uris** and enter `.kerberos.local` in the text box.
- Do the same for **network.negotiate-auth.delegation-uris**.

Finally a ticket for the `superuser` should be stored on the local machine. The following command will request it:

```
kinit superuser@KERBEROS.LOCAL
```

Now it is possible to open anaconda server on firefox, in this case the URL is `anaconda.kerberos.local:8080`, after clicking **Sign In**, the user should be able to log in immediately without having to enter any credentials.

Configure Anaconda Repository to use PAM

Open the Anaconda Repository configuration file `$PREFIX/etc/anaconda-server/config.yaml` and add the following configuration to enable Pluggable Authentication Module (PAM) support:

```
AUTH_TYPE: PAM
```

When switching authentication to PAM the admin account is lost, so you need to add your admin account again:

```
anaconda-server-admin set-superuser "jsmith"
```

To apply the changes, restart the Anaconda Repository server:

```
supervisorctl restart all
```

Open a new browser window and navigate to your local Anaconda Repository installation:

```
http://your.anaconda.server
```

NOTE: Replace "your.anaconda.server" with your actual Anaconda Repository server IP address or domain name.

You can now log in using your PAM credentials.

NOTE: To use the "shadow" PAM backend, add the user under which Anaconda Repository is running (usually "anaconda-server") to the "shadow" group:

```
sudo usermod -a -G shadow anaconda-server
```

Read only mode

The site can be put into **read only** mode to disable any action that modifies the database.

This may be useful when mirroring from the site.

NOTE: Logging in modifies the database, so in **read only** mode users and admins may not log in.

To enable **read only** mode use the setting:

```
READ_ONLY: true
```

Configuring Anaconda Client

Anaconda Client gives you the ability to upload packages to your on-site Anaconda Repository and provides highly granular access control capabilities. The instructions below describe how to configure Client to use your local Repository instead of Anaconda Cloud.

Client configuration

On each machine that accesses your on-site Repository, run this command as the machine's local user:

```
anaconda config --set url http://your.server.name:<port>/api
```

Or, to set the default repo on a system-wide basis, run this command:

```
anaconda config --set url http://your.server.name:<port>/api --site
```

NOTE: Replace `your.server.name` with the name of your local Repository and `<port>` with the name of the port used by Repository.

The system level `config` file is used only if no user-level `config` file is present.

To show the system and user `config` file locations and configuration settings:

```
anaconda config --show
```

Conda configuration

When the above `anaconda config` steps are completed, you can access all packages and channels from the local on-site Repository instead of the public Anaconda.org.

Users can then add individual accounts to their `.condarc` file by running the following command:

```
conda config --add channels USERNAME
```

If you still want to access certain channels from the public Anaconda.org, run:

```
conda config --add channels http://conda.anaconda.org/USERNAME
```

NOTE: Replace `USERNAME` with your username.

Conda channel priority

To set a preferred priority for the channels conda searches for package installs, edit your `~/.condarc file` and change the order. Channels at the top are searched first.

For example:

```
channels:
  - channel
  - https://conda.anaconda.org/t/<token>/<channel2>
  - http://conda.anaconda.org/<channel1>
  - defaults
```

The order of search is:

1. Private on-site Repository channel.
2. Private Anaconda.org channel2.
3. Public Anaconda.org channel1.
4. Default channel on the on-site Repository.

Pip configuration

To install PyPI packages from your Repository, add your channel to your `~/.pip/pip.conf` configuration file.

Edit the file and add an extra-index-url entry to the global config section:

```
[global]
extra-index-url = http://your.server.name:<port>/pypi/USERNAME/simple
```

NOTE: Replace `your.server.name` with the name of your local Repository, `<port>` with the name of the port used by Repository and `USERNAME` with your username.

Kerberos configuration

If you have enabled Kerberos authentication as described in *Configuring Repository to use Kerberos*, your browser and Client should be able to authenticate to Repository using Kerberos.

In macOS/Unix, configure the file `/etc/krb5.conf`:

```
[libdefaults]
default_realm = YOUR.DOMAIN

[realms]
YOUR.DOMAIN = {
  kdc = your.kdc.server
}

[domain_realm]
your.anaconda.repository = YOUR.DOMAIN
```

NOTE: Replace `YOUR.DOMAIN` with your domain, `your.kdc.server` with your Kerberos key distribution center (KDC) and `your.anaconda.repository` with your local Repository server.

If your configuration is correct, you should be able to authenticate using the command line tool `kinit`:

```
kinit jsmith
anaconda login
```

NOTE: Replace `jsmith` with your username.

Browser Setup

Many browsers do not present your Kerberos credentials by default, to prevent leaking credentials to untrusted parties. In order to use Kerberos authentication, you must whitelist Repository as a trusted party to receive credentials.

You must restart your browser after configuring the whitelist in order for changes to be reflected.

Safari

Safari requires no configuration—it automatically presents your credentials without whitelisting.

Chrome

The AuthServerWhitelist policy must be set to `your.anaconda.repository` to allow Chrome to present credentials to Repository with the hostname `your.anaconda.repository`. Depending on your DNS configuration, DisableAuthNegotiateCnameLookup may also be required to prevent Chrome from canonicalizing the hostname before generating a service name.

NOTE: Replace `your.anaconda.repository` with your local Repository server.

To configure on macOS:

```
defaults write com.google.Chrome AuthServerWhitelist "your.anaconda.repository"
```

On Linux:

```
mkdir -p /etc/opt/chrome/policies/managed
mkdir -p /etc/opt/chrome/policies/recommended
chmod -w /etc/opt/chrome/policies/managed
echo '{"AuthServerWhitelist": "your.anaconda.repository"}' > /etc/opt/chrome/policies/
↪managed/anaconda_repo_policy.json
```

On Windows, use Group Policy objects to set the Authentication server whitelist setting to `your.anaconda.repository`.

For more information, see Chrome's SPNEGO authentication and administration documentation.

Firefox

1. Navigate to the configuration page `about:config`.
2. Search for `negotiate`.
3. Set the configuration item `network.negotiate-auth.trusted-uris` to `your.anaconda.repository`

NOTE: Replace `your.anaconda.repository` with your local Repository server.

Internet Explorer

1. In the **Tools** menu, select Internet Options.
2. On the **Advanced** tab, in the Security section, select Enable Integrated Windows Authentication.

Configuring local mirrors

You can add a local copy—mirror—of Anaconda or PyPI repositories to your Anaconda Repository installation.

This section explains how to use Repository's convenient syncing tools to create and configure local mirrors:

Mirroring an Anaconda repository

NOTE: This documentation is for versions of Anaconda Repository 2.26 and newer. For versions from 2.23 through 2.25, see *2.23 documentation*. For versions before 2.23, see *2.20 documentation*.

Before you start

You need to have already installed and configured your Repository instance. Due to the size of Repository, it is important that you have configured a file storage location with sufficient disk space. If necessary, see the *requirements for the file storage location*.

A full Anaconda mirror requires approximately 90 GB.

You will also need to install `cas-mirror` as it is the recommended mirroring tool.

NOTE: The `anaconda-mirror` tool has been deprecated and will not be updated any further.

Mirroring all packages

You can mirror some or all of the contents of the Anaconda repository using the `cas-sync-api-v4` command:

```
$ cas-sync-api-v4 --help
usage: cas-sync-api-v4 [-h] [-f FILENAME] [--config] [--version]
                       [-l LOG_LEVEL] [-v]

Updates an Anaconda repository instance

```

(continues on next page)

(continued from previous page)

```
optional arguments:
  -h, --help            show this help message and exit
  -f FILENAME, --file FILENAME
                        Configuration file location (Defaults to:
                        /home/abarto/.cas-mirror or /etc/cas-mirror)
  --config, --show-config
                        Show running configuration and exit
  --version             Print version and exit
  -l LOG_LEVEL, --log-level LOG_LEVEL
                        Set the log level (CRITICAL, ERROR, WARNING, INFO,
                        DEBUG)Default: INFO.
  -v, --verbose         Shorthand for --log-level DEBUG
```

This tools mirrors all of the packages from the Anaconda repository default channels into the `anaconda` user account. It leverages the functionality exposed by the `anaconda-client` package to import the remote packages into an existing site.

1. Associate the URL of your Anaconda repository with a site using the `anaconda` command:

   ```
   anaconda config --set sites.myrepo.url "http://your-anaconda-repo/"
   ```

   NOTE: Replace `your-anaconda-repo` with the URL to your installation of Repository.

2. Create a configuration file `sync.yaml` that tells `cas-sync-api-v4` which site to use:

   ```
   dest_site: myrepo
   ```

3. Check that the configuration is valid with the `--config` parameter:

   ```
   $ cas-sync-api-v4 -f sync.yaml --config
   path: /home/ec2-user/sync.yaml
   remote_url: https://repo.continuum.io/
   mirror_dir: /opt/cas-mirror
   platforms: ['osx-32', 'osx-64', 'win-32', 'win-64', 'linux-32', 'linux-64',
   ↪'linux-armv6l', 'linux-armv7l', 'linux-ppc64le']
   fetch_installers: True
   repodata_source: False
   dir_names: ['archive']
   server_log_dir: None
   server_port: None
   dest_site: myrepo
   dest_channel: anaconda
   verify_checksum: False
   delta: False
   delta_dir: None
   log_dir: None
   log_level: 20
   python_versions: []
   pkg_list: []
   license_blacklist: []
   blacklist: []
   whitelist: []
   channels:
       - https://repo.continuum.io/pkgs/main/
       - https://repo.continuum.io/pkgs/free/
       - https://repo.continuum.io/pkgs/pro/
   ```

With this configuration, cas-sync-api-v4 mirrors the contents of all the default channels into the anaconda account of the myrepo site.

4. Run cas-sync-api-v4:

```
cas-sync-api-v4 -f sync.yaml
```

5. Verify the mirroring by opening a browser and loading this URL:

```
http://your-anaconda-repo/anaconda/
```

NOTE: Replace your-anaconda-repo with the URL to your installation of Repository.

Mirroring some packages

Alternately, you may not want to mirror all packages. To mirror a subset of the total repository, specify which platforms you want to include, or use the whitelist, blacklist or license_blacklist functionality to control which packages are mirrored, by copying the default configuration file $PREFIX/etc/anaconda-server/mirror/anaconda.yaml to $PREFIX/etc/anaconda-server/mirror/anaconda-custom.yaml.

For Repository 2.27 or newer, there are sample yaml config files located at: $PREFIX/etc/anaconda-server/mirror. PREFIX is the install location of Repository, which by default is ~anaconda-server/repo/etc/anaconda-server/mirror.

This command mirrors the repository according to the settings in the configuration file anaconda-custom.yaml:

```
cas-sync-api-v4 -f anaconda-custom.yaml
```

For more information, see *Customizing mirrors*.

Offline mirroring

Offline mirroring is done by using both cas-sync and cas-sync-api-v4. First download all the packages onto a host with Internet access.

EXAMPLE:

To download the packages, create a configuration file named export.yaml:

```
mirror_dir: /opt/mirror/export/
platforms:
  - linux-64
  - win-64
python_versions:
  - 2.7
  - 3.6
fetch_installers: false
pkg_list:
  - ca-certificates
  - certifi
  - libedit
  - libffi
  - libgcc-ng
  - libstdcxx-ng
  - ncurses
  - openssl
```

(continues on next page)

(continued from previous page)

```
  - pip
  - python
  - readline
  - setuptools
  - sqlite
  - tk
  - wheel
  - xz
  - zlib
```

This example downloads only a subset of the default channels.

Run `cas-sync`:

```
cas-sync -f export.yaml
```

After it finishes, a conda repository for each platform is created in the directory `/opt/mirror/export/pkgs`. We can now take the contents of the directory `/opt/mirror/export/` to the air-gapped environment.

To import the packages, create a config file named `import.yaml`:

```
dest_site: mysite
dest_channel: anaconda
channels:
  - file:///opt/mirror/export/pkgs/
platforms:
  - linux-64
  - win-64
python_versions:
  - 2.7
  - 3.6
```

Make sure these requirements are true:

- The files mirrored onto the connected box have been put in the `/opt/mirror/export` directory.
- An anaconda site named `mysite` is properly configured and the user has logged into it using the `anaconda` command.
- An “anaconda” account exists in the “mysite” site and the logged in user has access to it.

After these requirements are true, run `cas-sync-api-v4`:

```
cas-sync-api-v4 -f import.yaml
```

Mirroring with the deprecated anaconda-mirror command

The `anaconda-mirror` command is deprecated but still in use at some installations.

It is used with the command `anaconda-mirror sync`, or with a configuration file such as `anaconda-custom.yaml` with the command `anaconda-mirror --config-file anaconda-custom sync`.

Resetting packages

Use the `--reset` option to reset the previously mirrored packages:

```
anaconda-mirror --config-file anaconda-custom sync --reset
```

This resets the "last sync" time for the repository, so `anaconda-mirror` requests all packages, not just those changed or added since the last sync. As the requests are processed, `anaconda-mirror` still automatically downloads only those files that differ from the files currently in the repository.

Exporting a mirror

To generate a mirror archive:

```
anaconda-mirror export mirror.tar
```

This command dumps the packages, according to the *configured settings*, into the file `mirror.tar`.

This mirror can be used in an air gapped environment.

Importing a mirror

To mirror the Anaconda repository in an air gapped environment, point `anaconda-mirror` to the exported mirror archive.

Mount the USB drive and then run:

```
anaconda-mirror import $USB/mirror.tar
```

This command mirrors the contents of the local Anaconda repository to your Repository installation under the username "anaconda."

Filtering

If you want to update the filters on your mirror—for example, to exclude additional licenses—running `anaconda-mirror sync` again retrieves new packages that match this filter, but it does not remove existing packages that no longer match the filter.

To see which packages no longer match your filter:

```
anaconda-mirror clean --dry-run
```

To remove these packages from your mirror:

```
anaconda-mirror clean
```

Mirroring additional channels

If mirroring from an air gap archive, the channel in the following configuration points to a local directory to which the archive is expanded.

In addition, if a platform-specific archive is downloaded, then the config file needs the platforms section. The examples in the following sections assume `x64-repo-mirrors-\*.tar <airgap-archive-mirrors>` is expanded to `$MIRRORS_ARCHIVE`.

Similarly, for an online system, the channel points to Anaconda Cloud. The platforms are optional and limit the mirrored conda packages to the specified platforms.

Mirroring R channel

1. Create the `yaml` config file.

   EXAMPLE: The following is a config to mirror from an air gap archive containing only x64 packages:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/r.yaml

   channels:
     - file://$MIRRORS_ARCHIVE/r/pkgs

   # The platforms should correspond to the platforms contained in
   # the archive. Omit if the archive contains conda packages for all platforms.
   platforms:
     - linux-64
     - osx-64
     - win-64
   ```

   EXAMPLE: The following is for an online system:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/r.yaml

   channels:
     - https://conda.anaconda.org/r
   ```

2. Mirror the packages to r-channel:

   ```
   anaconda-server-sync-conda --mirror-config \
       $PREFIX/etc/anaconda-server/mirror/r.yaml --account=r-channel
   ```

Mirroring Wakari channel for AEN

1. Create the `yaml` config file.

   EXAMPLE: The following is a config to mirror from an air gap archive containing only x64 packages:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/wakari.yaml

   channels:
     - file://$MIRRORS_ARCHIVE/wakari/pkgs

   # The platforms should correspond with the platforms contained in
   # the archive. Omit if the archive contains conda packages for all platforms.
   platforms:
     - linux-64
     - osx-64
     - win-64
   ```

   EXAMPLE: The following is for an online system:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/wakari.yaml

   channels:
     - https://conda.anaconda.org/t/<TOKEN>/anaconda-nb-extensions
     - https://conda.anaconda.org/wakari
   ```

   NOTE: Replace <TOKEN> with the token for the anaconda-nb-extensions channel that you should have received along with your Repository license.

2. Mirror the packages to the Wakari channel:

   ```
   anaconda-server-sync-conda --mirror-config \
       $PREFIX/etc/anaconda-server/mirror/wakari.yaml --account=wakari
   ```

Mirroring anaconda-adam channel for cluster management

1. Create the yaml config file.

   EXAMPLE: The following is a config to mirror from an air gap archive containing only x64 packages:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/anaconda-adam.yaml

   channels:
     - file://$MIRRORS_ARCHIVE/anaconda-adam/pkgs

   # The platforms should correspond with the platforms contained in
   # the archive. Omit if the archive contains conda packages for all
   # platforms.
   platforms:
         * linux-64
         * osx-64
         * win-64
   ```

   EXAMPLE: The following is for an online system:

   ```
   cat $PREFIX/etc/anaconda-server/mirror/anaconda-adam.yaml

   channels:
     - https://conda.anaconda.org/anaconda-adam
   ```

2. Mirror the packages to anaconda-adam channel:

   ```
   anaconda-server-sync-conda --mirror-config \
       $PREFIX/etc/anaconda-server/mirror/anaconda-adam.yaml --account=anaconda-adam
   ```

Configuring conda

Having created the mirror, you still need to configure conda to search for packages here rather than on the default Anaconda repository. You can do that by editing your ~/.condarc file to add the appropriate channel:

```
channels:
    - http://<anaconda.repo.ipaddress>:<port>/conda/anaconda/
```

NOTE: Replace <anaconda.repo.ipaddress> with the URL to your installation of Repository.

NOTE: This configuration change can be made at the user level or via an administrative conda file, to force all internal users to use your local Anaconda mirror rather than querying the Anaconda repository.

NOTE: Users can download Anaconda installers that are pre-configured to search your Repository from http://<anaconda.repository.addr>/downloads. To learn how to generate these installers, see *Customizing installers*.

Mirroring a PyPI repository

Before you start

You need to have already installed and configured your Repository instance. Due to the size of Repository, it is important that you have configured a file storage location with sufficient disk space. If necessary, see the *requirements for the file storage location*.

The full PyPI mirror requires approximately 120 GB.

Running the PyPI mirror command

To create a PyPI mirror:

```
anaconda-server-sync-pypi
```

This command loads all of the packages on pypi.python.org into the ~pypi binstar user account.

Verify that the command ran successfully by opening your browser to `http://your-anaconda-repo/pypi/~pypi`.

NOTE: Replace `your-anaconda-repo` with the URL to your installation of Repository.

Customizing the mirror

It is possible to customize the mirror behavior by creating a configuration file such as `$PREFIX/etc/anaconda-server/mirror/pypi.yaml` and using the `--mirror-config` option:

```
anaconda-server-sync-pypi --mirror-config /etc/binstar/mirrors/pypi.yaml
```

The following configuration options are available:

Name	Description
user	The local user under which the PyPI packages are imported. Default: pypi.
pkg_list	A list of packages to mirror. Only packages listed are mirrored. If this is set, blacklist and whitelist settings are ignored. **Default:** [].
whitelist	A list of packages to mirror. Only packages listed are mirrored. If the list is empty, all packages are checked. Default: [].
blacklist	A list of packages to skip. The packages listed are ignored. Default: [].
latest_only	Only download the latest versions of the packages. Default: false.
remote_url	The URL of the PyPI mirror. /pypi is appended to build the XML RPC API URL, /simple for the simple index and /pypi/{package}/{version}/json for the JSON API. Default: https://pypi.python.org/.
xml_rpc_api_url	A custom value for XML RPC URL. If this value is present, it takes precedence over the URL built using remote_url. Default: null.
simple_index_url	A custom value for the simple index URL. If this value is present, it takes precedence over the URL built using remote_url. Default: null.
use_xml_rpc	Whether to use the XML RPC API as specified by PEP381. If this is set to true, the XML RPC API is used to determine which packages to check. Otherwise the scripts falls back to the simple index. If the XML RPC fails, the simple index is used. Default: true.
use_serial	Whether to use the serial number provided by the XML RPC API. Only packages updated since the last serial saved are checked. If this is set to false, all PyPI packages are checked for updates. Default: true.
create_org	Create the mirror user as an organization instead of a regular user account. All superusers are added to the "Owners" group of the organization. Default: false.
private	Save the mirrored packages as private. **Default:** false.

EXAMPLE:

```
whitelist:
  - requests
  - six
  - numpy
  - simplejson
latest_only: true
remote_url: http://pypimirror.local/
use_xml_rpc: true
```

Configuring pip

To configure pip to use this new mirror, edit /etc/pip.conf as follows:

```
[global]
index-url=https://pypi.anaconda.org/pypi/simple
```

Customizing mirrors

This section explains how you can customize your PyPI or Anaconda repository mirror.

NOTE: This documentation applies to versions of Repository 2.26 and newer. For versions between 2.23 and 2.26, see *2.23 documentation*. For older versions, see *2.20 documentation*.

Before you start

Your Anaconda Repository should already be installed and configured, as well as the mirroring tool `cas-mirror`. See *mirroring an Anaconda repository*.

Customizing your mirror

The following commands can be used with the `cas-mirror` tool:

- `cas-sync`: Creates or updates an existing local Anaconda package repository. The packages are saved into a local directory. This directory is defined by the `mirror_dir` configuration setting.
- `cas-sync-api-v4`: Creates or updates an existing remote Anaconda package repository. The packages are uploaded into an account of an existing Anaconda repository instance. The account and instance are controlled by the `dest_channel` and `dest_site` configuration settings.
- `cas-merge`: Combines delta repositories into an existing Anaconda package repository. A delta repository contains the changes between the local and remote Anaconda package repositories. Delta repositories are generated by `cas-sync` using the `delta` configuration setting.

All the commands can read a YAML configuration file specified by the `--file` (or `-f`) command line option. The YAML file can contain any of the following valid keys:

remote_url

Conda packages, Anaconda installers and Miniconda installers are fetched from this remote URL.

DEFAULT: `https://repo.continuum.io/`

channels

Conda packages are fetched from these remote channels.

DEFAULT: A list of these channels:

- `<remote_url>/pkgs/main/`
- `<remote_url>/pkgs/free/`
- `<remote_url>/pkgs/pro/`

mirror_dir

The mirror is saved in this local directory.

NOTE: A full mirror of the Anaconda repository uses about 100 GB of disk space. Make sure the mirror directory has enough space.

DEFAULT: `/opt/cas-mirror`

platforms

Conda packages and installers for this list of platforms are mirrored.

DEFAULT: A list of all platforms. This is:

```
['osx-32', 'osx-64', 'win-32', 'win-64', 'linux-32', 'linux-64',
'linux-armv6l', 'linux-armv7l', 'linux-ppc64le']
```

fetch_installers

Whether to fetch all Anaconda and Miniconda installers from `remote_url`.

If `fetch_installers` is set to `false` then installers are not fetched.

DEFAULT: `true`

python_versions

Python versions to mirror.

DEFAULT: All versions.

EXAMPLE: `['2.7', '3.6']`

pkg_list

An explicit list of package names to be mirrored.

When this list is provided, the `license_blacklist`, `blacklist` and `whitelist` keys are not allowed to be set.

license_blacklist

A list of licenses to be excluded from the mirror.

The license families that can be blacklisted are:

- AGPL
- APACHE
- BSD
- GPL2
- GPL3

- LGPL
- MIT
- PROPRIETARY
- PUBLICDOMAIN
- PSF
- OTHER
- NONE

blacklist

A list of package names to be excluded from the mirror.

whitelist

A list of package names to be included in the mirror.

The whitelist overrides the blacklists. If a package is both blacklisted and whitelisted, then it is included and mirrored.

EXAMPLE: The package `numpy` has a license in the license family `BSD`.

If `license_blacklist` contains `BSD` and `whitelist` is empty, then `numpy` and all other BSD licensed packages are excluded and not mirrored.

If `license_blacklist` contains `BSD` and `whitelist` contains `numpy`, then the `numpy` package is included and mirrored and other BSD licensed packages are excluded and not mirrored.

dest_channel

Optional channel to use when synchronizing with a local Repository instance.

DEFAULT: “anaconda”

dest_site

Optional site to use when synchronizing with a local Repository instance.

DEFAULT: None

delta

If `delta` is true, then a delta is generated from `mirror_dir`.

If `delta` is false, then the changes are applied directly.

delta_dir

The delta is generated (or merged) onto this directory.

If this is not specified, the generated delta directory is named `delta-<timestamp>-pkgs`. `<timestamp>` is replaced with a timestamp.

max_retries

The number of retries to allow before failing.

When it is set to 0, `cas-mirror` fails at the first error. Default is 0.

This is supported only for the `cas-sync` and `cas-sync-api-v4` commands.

DEFAULT: 0

safe

If `safe` is `true`, synchronizing repositories or merging delta directories never delete anything.

DEFAULT: `false`

repodata_source

Uses a `repodata.json` (or `repodata.json.bz2`) as the source of existing packages.

This file must be in the mirror directory for a specific platform.

EXAMPLE: `/mirror/linux-64/repodata.json`

Mirroring a platform-specific list

By default, `cas-sync` and `cas-sync-api-v4` mirror all platforms. If you do not need all platforms, you can save time and disk space by editing the `yaml` file to specify which platform(s) you want to mirror.

EXAMPLE:

```
platforms:
  - linux-64
  - win-32
```

Mirroring a package-specific list

You may want to mirror only a small subset of Repository. Rather than blacklisting a long list of packages you do not want to be mirrored, you can instead simply enumerate the list of packages you do want to mirror.

EXAMPLE: This example mirrors only the three packages Accelerate, PyQt and Zope. All other packages are ignored:

```
package_list:
- accelerate
- pyqt
- zope
```

Mirroring Python version-specific packages

You may want to mirror only a subset of versions.

EXAMPLE: This example mirrors only Anaconda packages built for Python 3.3:

```
python_versions:
  - 3.3
```

Mirroring with a license blacklist

As of Repository 2.26.0, the Anaconda mirroring script supports license blacklisting for the following license families:

- AGPL
- APACHE
- BSD
- GPL2
- GPL3
- LGPL
- MIT
- PROPRIETARY
- PUBLICDOMAIN
- PSF
- OTHER
- NONE

EXAMPLE: This example mirrors all the packages in the repository except those that are GPL2-, GPL3- or BSD-licensed:

```
license_blacklist:
  - GPL2
  - GPL3
  - BSD
```

NOTE: Older versions of Anaconda mirror support only license blacklisting for GPL. If you are using an older version of Repository, see the documentation for *customizing your PyPI or Anaconda Repository mirror*.

Mirroring with a blacklist

The `blacklist` allows access to all packages except those explicitly listed.

EXAMPLE: This example mirrors the entire Repository except the bzip2, tk and openssl packages:

```
blacklist:
  - bzip2
  - tk
  - openssl
```

Mirroring with a whitelist

The whitelist functions in combination with either the license_blacklist or blacklist arguments, and re-adds packages that were excluded by a previous argument.

EXAMPLE: This example mirrors the entire Repository except any GPL2- or GPL3-licensed packages, but including readline, despite the fact that it is GPL3-licensed:

```
license_blacklist:
  - GPL2
  - GPL3
whitelist:
  - readline
```

Combining multiple mirror configurations

You may find that combining two or more of the arguments above is the simplest way to get the exact combination of packages that you want.

The platforms argument is evaluated before any other argument.

EXAMPLE: This example mirrors only linux-64 distributions of the dnspython, shapely and gdal packages:

```
platforms:
  - linux-64
package_list:
  - dnspython
  - shapely
  - gdal
```

If the license_blacklist and blacklist arguments are combined, the license_blacklist is evaluated first, and the blacklist is a supplemental modifier.

EXAMPLE: In this example, the mirror configuration does not mirror GPL2-licensed packages. It does not mirror the GPL3-licensed package PyQt because it has been blacklisted. It does mirror all other packages in Repository:

```
license_blacklist:
  - GPL2
blacklist:
  - pyqt
```

If the blacklist and whitelist arguments are both employed, the blacklist is evaluated first, with the whitelist functioning as a modifier.

EXAMPLE: This example mirrors all packages in the repository except astropy and pygments. Despite being listed on the blacklist, Accelerate is mirrored because it is listed on the whitelist:

```
blacklist:
 - accelerate
 - astropy
 - pygments
whitelist:
 - accelerate
```

Verifying the checksum of a file

To help ensure that a file was correctly uploaded or synced you can use the checksum tool. This routine fetches a file from a database and verifies that the stored hash checksum and the calculated hash checksum of the file on disk are the same.

On a package's page, view the file list and click the Info button next to a file to see the file's keyname.

To check the file's hash checksum run:

```
anaconda-server-checksum keyname
```

NOTE: Replace "keyname" with the file's keyname.

The output will be either `Hashes are the same` or `Hashes differ`.

To overwrite the old hash checksum in the database with the new hash checksum you calculated, use the option `--fix`.

Customizing installers

Anaconda Repository can distribute copies of the Anaconda distribution and the Miniconda installer that are pre-configured to use your installation of Repository.

This applies to Anaconda Distribution version 4.1 and higher, and Miniconda version 4.1.11 and higher.

By default the installers will be stored in `$PREFIX/opt/anaconda-server/installers`. If you prefer to store the installers in a different location, configure a new path:

```
anaconda-server-config --set INSTALLER_DIR /preferred/directory
```

NOTE: Replace "/preferred/directory" with the path to the directory where you prefer to store the installers.

If necessary, edit the script below and replace "5.2.0" with the current version number.

To download the installers:

```
mkdir -p /tmp/extras
pushd /tmp/extras

URL="https://repo.continuum.io"

version="5.2.0"
miniconda="Miniconda3-latest-Linux-x86_64.sh \
    Miniconda3-latest-MacOSX-x86_64.sh \
    Miniconda3-latest-Windows-x86.exe \
    Miniconda3-latest-Windows-x86_64.exe \
    Miniconda-latest-Linux-x86_64.sh \
    Miniconda-latest-MacOSX-x86_64.sh \
    Miniconda-latest-Windows-x86.exe \
    Miniconda-latest-Windows-x86_64.exe"
anaconda="Anaconda2-$version-Linux-x86_64.sh \
    Anaconda3-$version-Linux-x86_64.sh \
    Anaconda2-$version-MacOSX-x86_64.sh \
    Anaconda3-$version-MacOSX-x86_64.sh \
    Anaconda2-$version-MacOSX-x86_64.pkg \
    Anaconda3-$version-MacOSX-x86_64.pkg \
    Anaconda2-$version-Windows-x86_64.exe \
    Anaconda3-$version-Windows-x86_64.exe"
```

(continues on next page)

(continued from previous page)

```

for installer in $miniconda; do
    curl -O $URL/miniconda/$installer
done
for installer in $anaconda; do
    curl -O $URL/archive/$installer
done

# Move the files into the installers directory
popd
cp -a /tmp/extras $PREFIX/opt/anaconda-server/installers
```

The installers will be available for download from `http://your.anaconda.server:port/downloads`.

NOTE: Replace "your.anaconda.server:port" with the name or IP address and port of your Anaconda server.

The downloadable file will be a zip file containing the Anaconda distribution and the configuration files specific to your Repository. These zip files are cached in the server's configured storage for quick retrieval.

Check that the `SERVER_NAME` *setting* has been set so you generate the correct URLs in the next step.

To pre-generate these installer bundles based on the downloaded installers, or to re-generate after downloading new installers, execute the command:

```
anaconda-server-admin update-installers
```

This command requires that the *SERVER_NAME setting* be set in order to generate the correct URLs.

By default, the included `conda` installation will point to the default `anaconda` and `r-channel` accounts on your Anaconda Repository server, if those accounts exist.

You can change these default channels by setting the `DEFAULT_CHANNELS` *setting*, and then running the `anaconda-server-admin update-installers` command.

Configuration reference

Files

Anaconda Repository loads configuration files with the extension `.yaml` from the following locations:

- `/etc/binstar/`
- `/etc/anaconda-server/`
- `$PREFIX/etc/anaconda-server`

NOTE: `$PREFIX` is the location where repository is installed.

Files are loaded from these directories in order, with later files overriding earlier files. Files are loaded from each directory in alphabetical order.

If an environment variable `ANACONDA_SERVER_CONFIG` is set with the path of a configuration file, this file is loaded after the three already listed. Its settings override any conflicting settings in the earlier files.

Each configuration setting variable can have its value set with the `anaconda-server-config --set` command, or by editing a configuration file.

EXAMPLE: To set a value named VALUE_ONE to 50, add this to a configuration file:

```
VALUE_ONE: 50
```

Or, you can set a value named VALUE_ONE to 50 with this command:

```
anaconda-server-config --set VALUE_ONE 50
```

Logging

The location of the server's log file is defined in the supervisord configuration file `$PREFIX/etc/supervisord.conf` by the `stdout_logfile` config entry located in the [program:anaconda-server] section.

Advanced configuration of logging requires setting a LOGGING key on the server's config.yaml. It uses Python's logging module config structure.

Usernames

USER_REGEX

A regular expression that defines the allowable user names.

For example, this setting specifies that user names contain only lowercase letters, periods, plus and minus characters (`.`, + and –):

```
USER_REGEX: '^[a-z.+-]+$'
```

NOTE: The default value for `USER_REGEX` is `^[a-z0-9_][a-z0-9_-]+$` which translates to: at least one alphanumeric character or underscore, followed by zero or more alphanumeric, dash or underscore characters.

NOTE: Escape any extra instances of the single quote character `'` as `\'`. Do not use the slash and ampersand characters `/` and `&`, which have special meanings in URLs.

NOTE: If USER_REGEX is changed and the server is restarted, existing usernames that do not match the new USER_REGEX do not cause errors.

Database

Repository uses MongoDB as the database back end.

MONGO_URL

A MongoDB connection URI is used to connect to the MongoDB database server. It can be used to configure the hostname and port, as well as database authentication.

For example:

```
MONGO_URL: mongodb://anaconda-server:Pa55w0rd@mongodb.serv/
```

MONGO_DBNAME

The MongoDB database where Repository stores its data.

MQ_DBNAME

The MongoDB database where Repository stores data used for asynchronous processing.

MONGO_REPLICA_SET

The name of a MongoDB replica set Repository connects to after establishing a connection to the database server.

File storage

Repository can serve package contents from a local file-system, or from Amazon Web Services Simple Storage Service: AWS S3.

Storage_type

The storage mechanism to use. Valid choices are `fs`, for file-system storage, or `s3`, for AWS S3 storage.

keyname_full_path

When this option is set, Repository stores the files by full paths and not just by hashes. This way a tensorflow file uploaded by the user *Bob* will be stored on `<fs_storage_root>/Bob/tensorflow/osx-64/tensorflow-1.1.0-np112py36_0.tar.bz2-594ac56e7e042600648defdb`.

NOTE: The storage path does not always contain the current file owner and their user name. This is because the file location on the storage does not change when you rename a user or transfer a file to a different user.

Fs_storage_root

If configured to use file-system storage, the absolute path to a directory where Repository stores all uploaded packages.

PACKAGE_BUCKET_ID

If configured to use AWS S3 storage, the name of an AWS S3 bucket where Repository stores uploaded packages.

You can identify the name of your bucket by using `<bucket>` in your `http://<bucket>.s3.amazonaws.com` URL.

S3_REGION_NAME

The S3 region that the bucket is located in. The available regions can be found in the Amazon AWS documentation.

S3_SERVER_SIDE_ENCRYPTION

This variable can be set to `AES256` to enable server-side encryption for packages stored in the S3 bucket.

Notebooks

MAX_IPYNB_SIZE

Specifies the maximum allowed size when uploading notebooks to the server. The default is 25 MB. This variable can be set in `config.yaml`.

Web server

SERVER_NAME

The name and port number of the server. This option is required for subdomain support.

For example:

```
SERVER_NAME: anaconda.srv:8080
```

port

The port number of the server. Defaults to `8080`.

subdomains

If set to `true`, Repository serves `conda` package from a separate subdomain. Defaults to `false`.

For example:

```
SERVER_NAME: anaconda.srv:8080
subdomains: true
```

Allows access to conda packages at `http://conda.anaconda.srv:8080/`.

SESSION_COOKIE_DOMAIN

The domain that Repository sets on the session cookie. If this is not set, the cookie is valid for all subdomains of `SERVER_NAME`.

See *Securing user-created content*.

USER_CONTENT_DOMAIN

As a cross-site scripting (XSS) protection, notebook content can be served from a separate domain name. If this option is configured, Repository only serves rendered notebooks from this domain.

See *Securing user-created content*.

ssl_options

Repository can serve content over HTTPS, using user-provided SSL certificates.

For example:

```
ssl_options:
    certfile: /etc/anaconda-server/server.crt
    keyfile: /etc/anaconda-server/server.key
PREFERRED_URL_SCHEME: https
```

certfile

The absolute path to a PEM-formatted X.509 certificate file.

keyfile

The absolute path to a PEM-formatted private key for the associated certificate.

ssl_version

An integer that specifies the SSL protocol version as defined by Python's `ssl` module:

```
PROTOCOL_SSLv2 = 0
PROTOCOL_SSLv23 = 2
PROTOCOL_SSLv3 = 1
PROTOCOL_TLS = 2
PROTOCOL_TLSv1 = 3

PROTOCOL_TLSv1_1 = 4
PROTOCOL_TLSv1_2 = 5
```

The default is `5` (TLS v1.2).

PREFERRED_URL_SCHEME

The preferred scheme that is used to generate URLs. Set this to `https` if HTTPS is configured.

gunicorn

Repository uses Gunicorn. The most commonly used options are `timeout` and `workers`. A complete list of settings can be found in Gunicorn's documentation.

For example:

```
gunicorn:
    timeout: 60
    workers: 5
```

timeout

The number of seconds for which a worker is allowed to process a request, before being forcefully terminated.

Default: 120

workers

The number of workers that Gunicorn spawns to serve Repository. Defaults to 2 × the number of CPUs + 1.

Authentication

AUTH_TYPE

The method Repository uses to authenticate users. Valid choices are `NATIVE`, for built-in authentication, `KERBEROS`, for Kerberos, and `LDAP`.

KRB5_HOSTNAME

See *Kerberos configuration options*.

KRB5_SERVICE_NAME

See *Kerberos configuration options*.

KRB5_KTNAME

See *Kerberos configuration options*.

LDAP

Options for configuring LDAP authentication and group synchronization.

For example:

```
LDAP:
  # Replace with company LDAP server
  URI: 'ldap://<ldap.company.com>'
  # Replace <uid=%(username)s,ou=People,dc=company,dc=com> with your company specific LDAP Bind/Base DN
  # Bind directly to this Base DN.
  BIND_DN: '<uid=%(username)s,ou=People,dc=company,dc=com>'
  # password of the user specified in the BIND_DN
  BIND_AUTH: abc123456

  USER_SEARCH:
      base: cn=Users,dc=example,dc=com
      filter: sAMAccountName=%(username)s
```

(continues on next page)

(continued from previous page)

```
  # Map LDAP keys into application specific keys
  KEY_MAP:
      name: 'cn'
      company: 'o'
      location: 'l'
      email: 'mail'

  OPTIONS:
      OPT_NETWORK_TIMEOUT: 60
      OPT_TIMEOUT: 60
```

NOTE: To use LDAP with SSL, set the `USER_REGEX` and `account_names_filter` options:

```
account_names_filter: false
USER_REGEX: ^[a-z0-9_][a-z0-9_-.]+$
LDAP:
    [configuration continues as above with URI, BIND_DN, and so on]
```

See *Using LDAP and TLS configuration options*.

LOCK_DOWN

Makes all views with the exception of the login form and welcome page, unaccessible to anonymous users.

Email

Repository can be configured to send email for various reasons, including to reset forgotten usernames and passwords. Email can be sent using SMTP protocol, or through Amazon Web Services Simple Email Service (AWS SES).

SMTP_HOST

The hostname of the SMTP server.

SMTP_PORT

The port of the SMTP server.

SMTP_TLS

If set to `true`, Repository attempts an SSL connection to the SMTP server.

SMTP_USERNAME

The username to authenticate against the SMTP server before attempting to send email.

SMTP_PASSWORD

The password to authenticate against the SMTP server before attempting to send email.

USE_SES

If set to true, Repository sends email with AWS SES. To authenticate to AWS, the server should be configured with an appropriate IAM role, or have credentials specified in a Boto configuration file.

RETURN_ADDRESS

The From: email address that Repository uses as sender.

ALLOW_DUPLICATED_EMAILS

If set to true, Repository allows different users to share the same email or secondary email. Defaults to false.

require_email_validation

If set to true, Repository emails new users a unique token to validate their email address before permitting them to log in.

Advanced

AVATAR_METHOD

The method to use to generate the user avatar URL. Valid choices are:

- 'gravatar' to use the gravatar.com service
- 'default' to show a predefined static icon
- 'static' to use a custom static URL

AVATAR_GRAVATAR_URL

A URL for a Gravatar compatible service. Default: https://www.gravatar.com/. This URL is used as the prefix to build a valid gravatar URL.

AVATAR_STATIC_URL

A static URL to use when AVATAR_METHOD is set to static. Defaults to an empty string.

CONSTRUCTOR_TIMEOUT

The timeout in seconds for the call to constructor while building installers, parcels and management packs. Defaults to 60 seconds.

CONSTRUCTOR_TOKEN_TIMEOUT

To provide access to private packages while building an installer, a temporary token is created. It must be valid during the call to `constructor` and it should expire soon after the call completes. `CONSTRUCTOR_TOKEN_TIMEOUT` sets the token's valid lifetime in seconds. Defaults to 60 seconds. This value should be greater than or equal to `CONSTRUCTOR_TIMEOUT`.

CONSTRUCTOR_ALLOWED_OPTIONS

A list of `constructor` option names that are allowed to be included in the installer construction form. The default is `[]` (no options are allowed).

PARCELS_ROOT

The prefix with which Cloudera parcels are generated. Defaults to `/opt/cloudera/parcels`.

PARCEL_DISTRO_SUFFIXES

The distributions for which Cloudera parcels are generated. Defaults to `['el5', 'el6', 'el7', 'lucid', 'precise', 'trusty', 'wheezy', 'jessie', 'squeeze', 'sles11', 'sles12']`.

For example, if you want to support only Ubuntu:

```
PARCEL_DISTRO_SUFFIXES:
    - lucid
    - precise
    - trusty
```

DEFAULT_CHANNELS

The Repository accounts that environments installed with the *bundled Anaconda distributions* pull packages from. Defaults to `['anaconda', 'r-channel']`.

For example, to add an additional `custom` account:

```
DEFAULT_CHANNELS:
    - anaconda
    - r-channel
    - custom
```

CONSTRUCTOR_TMPDIR

When `constructor` builds an installer it stores the configuration in this temporary directory. The default is `None`, which tells `constructor` to create a temporary directory using Python's `tempfile.mkdtemp`.

STANDARD_LABELS

A list of standarized labels. If a user defines a label that is not listed as standard, a warning notice will be shown in the package's page. Defaults to `['main', 'dev', 'alpha', 'beta', 'broken']`.

CONDA_CACHE_SIZE

The maximum size (in bytes) of the `repodata.json` requests cache. Set to `0` to disable `repodata.json` caching. Default: 1 Gb. When the maximum size is reached, the 10 least recently used entries of the cache are evicted.

CACHE_METHOD

The method used for caching repodata info. It can either be `tempfile` (the prior method of caching) or `diskcache`, which uses SQLite as a back-end. Default: `diskcache`.

REMEMBER_COOKIE_ENABLED

Sets whether to use the *remember me* cookie to keep the session alive. If it's set to true the `REMEMBER_COOKIE_DURATION` setting is relevant, and if it's set to false, the `PERMANENT_SESSION_LIFETIME` is relevant. Defaults to `true`.

PERMANENT_SESSION_LIFETIME

An integer that sets how many **minutes** the session will live. Only used when `REMEMBER_COOKIE_ENABLED` is false. Default is 44640 (31 days).

REMEMBER_COOKIE_DURATION

An integer that sets how many **minutes** the session will live when using the *remember me* cookie. Only used when `REMEMBER_COOKIE_ENABLED` is true. Default is 525600 (365 days).

SUPERUSER_ORG_ADMIN

Whether superusers should automatically be granted admin rights on organizations. Default is `false`.

NEXT_URL_WHITELIST

List of hostnames that are marked as safe when redirecting requests due to the presence of a "next" request parameter. It is mainly used under an Anaconda Enterprise Notebooks Single Sign-on Set-up. The default is [] (no external redirects are safe).

NEXT_URL_WHITELIST_REGEXP

A regular expression to match hostnames that are marked as safe when redirecting requests due to the presence of a "next" request parameter. It is mainly used under an Anaconda Enterprise Notebooks Single Sign-on Set-up. The default is '(?!)' which matches nothing, so only local redirects are allowed.

Repository has two installation options:

- *Online*: If you have internet access on the destination server, follow the online instructions.
- *Air gap*: If you have an air gapped system or the destination server does not have internet access, follow the air gap instructions.

Repository provides *advanced configuration options* that can be used to meet site-specific needs.

Repository includes a number of *optional components* that can be installed and used individually.

You may also want to see *Updating Repository* and *Uninstalling Repository*.

User management

Adding a user

New users can navigate in a browser to your local Repository web page and sign themselves up for an account, or you can add them using the command line:

```
anaconda-server-create-user --username jsmith --password
abc123DEF --email jsmith@acme.com
```

NOTE: Replace `jsmith`, `abc123DEF` and `jsmith@acme.com` with the username, password and email address of the new user.

Searching for users

As of Anaconda Repository version 2.33.5, you can search for users by username or email address. Open this URL in a browser:

```
http://your.anaconda.server:port/admin/users
```

NOTE: Replace "your.anaconda.server:port" with the name or IP address and port of your Anaconda server.

Promoting an existing user

As of Anaconda Repository version 2.28, administrators can use the Administration Accounts page to promote users to staff or superuser status.

Viewing rights of staff and superuser

The staff user and superuser can view the following sections of the Administration page:

- Reports.

- User administration.
- License downloads.
- Trial licenses.
- Current LDAP configuration.
- Current server configuration.
- Downloads summary.
- Downloads from a specific address.
- Security feed and security feed details.
- Storage administration.

Privileges of staff and superuser

The staff user and superuser have the following privileges:

- Create licenses.
- Download a CSV of the user database.
- Search for a package in the Administration panel.
- Resend confirmation emails to users.
- Resend password reset emails to users.
- Download a CSV with the users emails.

Additional rights of superuser

In addition to all of the above, a superuser can also view the following on the Administration page:

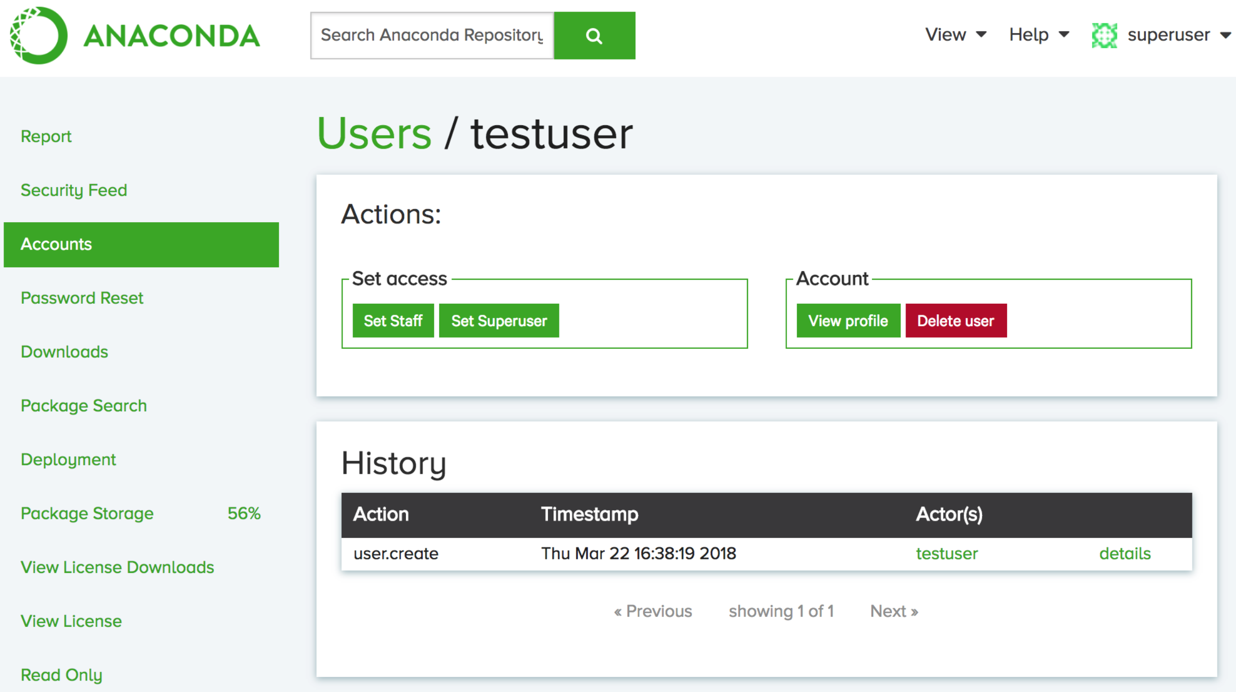
- Plans.
- User details.

To promote an existing user to a staff user or superuser:

1. On the Administration page, in the left navigation pane, select Accounts.
2. Select the username you want to promote.

   Repository displays the user information page:

3. Click the Set Staff button to give the user staff privileges or click the Set Superuser button to give the user superuser privileges.
4. In the dialog box that appears, retype the user’s name.
5. Click the Set button.



Action	Timestamp	Actor(s)	
user.create	Thu Mar 22 16:38:19 2018	testuser	details

Resetting user passwords

If a user forgets their password, you can request a reset link to provide to the user.

To send emails, Repository must have the *email settings* configured.

1. Log into your Repository administrative account.
2. From the top **Tools** menu, select Admin.
3. From the left navigation pane, select Password Reset.
4. Enter the user's email address.

   The Web UI generates a password reset link.
5. Email the link to the user.

You can also reset passwords without sending emails:

```
anaconda-server-admin reset-password jsmith
```

NOTE: Replace `jsmith` with the username whose password you want to reset.

Resending welcome emails to new users

To send emails, Repository must have the *email settings* configured.

If a user reports that they did not receive their welcome email after registering on your local Repository web page, it may have been caught in a spam filter.

Follow the above instructions for resetting user passwords.

Changing a user's storage size or changing their plan to free unlimited

To change a user's storage size or plan:

1. Log into your Repository administrative account.
2. From the top **Tools** menu, select Admin.
3. From the left navigation pane, select Accounts.
4. Select the username of the user whose account you want to change.
5. To update the storage limits click Update Storage.
6. To set the user's plan to free and unlimited, click Set free unlimited plan.

Removing a user

1. Log into your Repository administrative account.
2. From the top **Tools** menu, select Admin.
3. From the left navigation pane, select Accounts.
4. Select the username of the user you want to remove.
5. Click the Delete user button.
6. Optional: Use `anaconda-server-admin clean-storage` to remove files from that user's account.

System management

Recommended workflow

One of the most useful features of Anaconda Repository is its ability to help manage package development and deployment in a seamless fashion. This page describes the development process and channel usage employed by one of our internal teams, to serve as an example of how you can leverage channels for workflow separation.

Multiple channels allow our team to maintain separate package states and easily earmark and control the versions and states of packages that users can install.

Our team created the following channels:

- Master.
- Staging.
- Release.

We have used this workflow through 4 release cycles and it has worked out well for us.

Master

A master is created any time something is merged into our master branch. It is considered the development build of all of the components that make up the software. Code that makes it to this channel should be stable and should have been confirmed independently, but a full QA test has not been run on it yet.

Staging

Once we are ready to start working on a release, we create a staging:X.Y.Z branch. This contains all code that is going to go into a release. No new features should be introduced at this point, just any last minute bug fixes to existing code.

Release

The staging channel gets culled so that only the latest package is maintained in it. Any alpha, beta, or dev packages are removed. After all testing is complete, all issues are resolved, and the channel contains only one version of each package, we copy that package into a release:X.Y.Z channel, then lock that channel.

Performing general maintenance

To maintain a Repository installation, perform all of these tasks regularly:

- Review the error logs at `/var/log/anaconda-server`
- *Back up* the file system and database.
- Update the anaconda-server package with the command:

  ```
  conda update anaconda-server
  ```

Anaconda repository backup and restore procedure

This guide is for backing up and restoring an Anaconda Repository instance that uses local file system storage. If your instance uses Amazon S3 or any other storage provider, please consult their specific documentation on backup and restore procedures.

Before you start

These instructions are for a Repository that is:

- Installed in the directory `/home/anaconda-server/repo` as suggested by the installation guide.
- Owned by the `anaconda-server` user.
- Using the storage directory `/opt/anaconda-server/package-storage`.
- Storing the configuration file in `/etc/anaconda-server`.

If any of these items are different for your instance, modify these instructions accordingly.

- Unless noted, run all shell commands while logged in as the `anaconda-server` user. Using `sudo` privileges, log in as the `anaconda-server` user with this command:

  ```
  sudo su - anaconda-server
  ```

- Execute all commands in the working directory `/home/anaconda-server`:

  ```
  $ pwd
  /home/anaconda-server
  ```

Backup

Before starting the backup process, shut the service down using `supervisorctl`:

```
$ supervisorctl stop all
anaconda-server: stopped
$ supervisorctl status
anaconda-server                   STOPPED    Jul  6 05:05 PM
```

Make a `$VERSION` environment variable and set it to the version of the currently installed Anaconda Repository:

```
$ VERSION=`conda list anaconda-server --json | python -c 'import sys, json; print␣
↪json.load(sys.stdin)[0]["version"]'`
$ echo $VERSION
2.33.27
```

This version string will be used in all backup file names.

It's also useful to add a timestamp to the files, so generate one now:

```
$ TIMESTAMP=`date +%Y-%m-%d`
$ echo $TIMESTAMP
2018-07-30
```

Code/Binaries

Generate a tarfile archive with the installed code, binaries and any dependencies:

```
$ tar -cpsf anaconda-server-repo-$VERSION-$TIMESTAMP.tar --exclude var/run -C /home/
↪anaconda-server repo/
$ sha1sum anaconda-server-repo-$VERSION-$TIMESTAMP.tar > anaconda-server-repo-
↪$VERSION-$TIMESTAMP.tar.sha1
```

Notice that this also generated a SHA1 checksum. This checksum will be used to verify when you restore the archive.

Configuration

This step is necessary only if you stored Anaconda Repository's configuration in a custom location outside of the instance installation folder (usually `/home/anaconda-server/etc/`).

These commands show how to generate the tarfile if the configuration is stored in `/etc/anaconda-server`.

Generate the tarfile with its SHA1 checksum:

```
$ tar -cpsf anaconda-server-etc-$VERSION-$TIMESTAMP.tar /etc/anaconda-server
$ sha1sum anaconda-server-etc-$VERSION-$TIMESTAMP.tar > anaconda-server-etc-$VERSION-
↪$TIMESTAMP.tar.sha1
```

Storage

As before, create a tarfile archive and its checksum with the contents of the package storage location:

```
$ tar -cpsf anaconda-server-package-storage-$VERSION-$TIMESTAMP.tar -C /opt/anaconda-
↪server/ package-storage
$ sha1sum anaconda-server-package-storage-$VERSION-$TIMESTAMP.tar > anaconda-server-
↪package-storage-$VERSION-$TIMESTAMP.tar.sha1
```

Database

Generate a dump of Anaconda Repository's MongoDB database. We recommend you follow MongoDB's guidelines for backup and restore. This guide uses MongoDB tools:

```
$ mongodump --host=127.0.0.1 --port=27017 --archive=anaconda-server-mongodb-$VERSION-
↪$TIMESTAMP.archive
$ sha1sum anaconda-server-mongodb-$VERSION-$TIMESTAMP.archive > anaconda-server-
↪mongodb-$VERSION-$TIMESTAMP.archive.sha1
```

.bashrc

If you chose to let the Anaconda Repository installer update the `.bashrc` file of the user `anaconda-server`, back it up:

```
$ cp /home/anaconda-server/.bashrc anaconda-server-bashrc-$VERSION-$TIMESTAMP.sh
$ sha1sum anaconda-server-bashrc-$VERSION-$TIMESTAMP.sh > anaconda-server-bashrc-
↪$VERSION-$TIMESTAMP.sh.sha1
```

Restore

Before you start

- Verify that the restore environment meets the requirements listed in the *Installation Guide* for Anaconda Repository. You will need:
  - MongoDB (any supported version) installed
  - A user account (usually `anaconda-server`)
  - A storage directory (usually `/opt/anaconda-server/package-storage`) owned by the Anaconda Repository user account. This is only needed if you're using a local filesystem as a storage backend.
- Run all shell commands while logged in as the `anaconda-server` user, as you did when backing up Anaconda Repository. Using `sudo` privileges, log in as the `anaconda-server` user with this command:

  ```
  sudo su - anaconda-server
  ```

- Execute all commands in the working directory `/home/anaconda-server`.

Verify checksums

Verify the integrity of the backup files:

```
$ sha1sum --check *.sha1
anaconda-server-bashrc-2.33.27-2018-07-30.sh: OK
anaconda-server-mongodb-2.33.27-2018-07-30.archive: OK
anaconda-server-package-storage-2.33.27-2018-07-30.tar: OK
anaconda-server-repo-2.33.27-2018-07-30.tar: OK
```

.bashrc

If you backed up the `.bashrc` file of the user `anaconda-server`, restore it:

```
cp anaconda-server-bashrc-$VERSION-$TIMESTAMP.sh /home/anaconda-server/.bashrc
```

After restoring this file, log out and log in as `anaconda-server` again for the changes to take effect.

Database

If you followed the Anaconda Repository *Installation Guide*, MongoDB is up and running and you can use `mongorestore` to restore the database archive:

```
mongorestore --host=127.0.0.1 --port=27017 --db=binstar --archive=anaconda-server-
↪mongodb-$VERSION-$TIMESTAMP.archive
```

Storage

Assuming that the storage directory is `/opt/anaconda-server/package-storage`, restore it with:

```
tar -xpsf anaconda-server-package-storage-$VERSION-$TIMESTAMP.tar -C /opt/anaconda-
↪server/
```

Code/Binaries

Restore the code and binaries:

```
tar -xpsf anaconda-server-repo-$VERSION-$TIMESTAMP.tar -C /home/anaconda-server
```

Restore the `supervisord` configuration:

```
repo/bin/anaconda-server-install-supervisord-config.sh
```

The server should now be up and running. Check the status with `supervisorctl`:

```
$ repo/bin/supervisorctl status
anaconda-server                  RUNNING    pid 8446, uptime 0:03:18
```

Checking for Orphan Files or Packages

You can use the “orphan-check” tool to resynchronize the filesystem and the database if the filesystem and the database get out of sync.

The system can get out of sync when files in the filesystem are not referenced from the database, or when packages in the database do not have a corresponding file in the filesystem.

The orphan-check tool prints on stdout a list of files on the filesystem that are not referenced from the database:

```
anaconda-server-orphan-check --dryrun
```

You can use the *–json* option if you want a JSON representation of the output:

```
anaconda-server-orphan-check --json
```

NOTE: Running `anaconda-server-orphan-check` without arguments is the same as running `anaconda-server-orphan-check --dryrun`.

After you've viewed the list of files without references, "orphan-check –clean" can delete them:

```
anaconda-server-orphan-check --clean
```

You can also check for packages that have missing files:

```
anaconda-server-orphan-check --reverse
```

Then you can delete those file objects from the database:

```
anaconda-server-orphan-check --reverse --clean
```

Using optional components

Anaconda Repository includes a number of components that can be installed and used individually.

This section describes how to install and use two such tools:

- *cas-mirror*.
- *cas-installer*.

As a convention, all packages and commands that are part of the Repository product share the common `cas` prefix, which is short for Continuum Anaconda Server. (Anaconda, Inc. was formerly known as Continuum Analytics, Inc.)

All packages are installed using the `conda` command, which is part of the Miniconda installer. For Repository installation and configuration instructions, see *Installation*.

Using cas-mirror

The cas-mirror tool is a component of the Anaconda Repository Enterprise product.

The cas-mirror tool makes an exact copy of Anaconda's package Repository, or part of it, on a your local Repository server.

For more information about the cas-mirror tool's functionality and configurable options, see *Configuring local mirrors*.

Installing cas-mirror

To install the mirror tool, run:

```
conda install cas-mirror
```

After cas-mirror has been installed, the following commands are available:

```
cas-sync --help
cas-merge --help
cas-sync-api-v4 --help
cas-server --help
```

Using the cas-sync command

The `cas-sync` command brings the local mirror of Repository up-to-date with our remote servers.

To configure the location of the mirror on your file system, check the output of:

```
cas-sync --config
```

If necessary, create a configuration file, either `~/.cas-mirror` or system-wise `/etc/cas-mirror`, which contains the desired location of the local mirror on the file system, the platforms that should be mirrored and an optional blacklist of packages that which should not be mirrored.

EXAMPLE:

```
mirror_dir: /home/data/mirror
remote_url: ""  # where to get miniconda and anaconda installers -- blank to skip
# possible platforms are: linux-64, linux-32, osx-64, win-32, win-64 platforms:
  - linux-64
  - win-32
blacklist:
  - dnspython
  - shapely
  - gdal
```

Once you are satisfied with the mirror directory—which may be the default—run:

```
cas-sync
```

Running this command for the first time takes many hours, because the entire Repository is being downloaded. Subsequent runs take significantly less time.

Using the cas-server command

You need to run `cas-server` as root when you intend to serve on port 80.

To serve repository over HTTP, run:

```
cas-server
```

If needed, use the `--port` option to change the port on which the repository is being served.

Using the “delta” option

If you’ve already downloaded most of the anaconda repository, and you’re only interested in the changes since `cas-sync` was last run, you can use the `delta` configuration option:

```
mirror_dir: /home/data/mirror
remote_url: ""  # where to get miniconda and anaconda installers -- blank to skip
# possible platforms are: linux-64, linux-32, osx-64, win-32, win-64
platforms:
  - linux-64
  - win-32
blacklist:
  - dnspython
  - shapely
  - gdal
delta: true
delta_dir: delta_pkgs
```

Instead of mirroring to the existing local repository, it will record the necessary changes to bring the mirror up to date in a separate directory (delta_pkgs in this case). You can then use this generated directory to update air-gapped mirrors using the cas-merge command.

The cas-merge command takes a delta directory and combines its contents with an existing mirror directory. New packages are added, missing packages are deleted, and the repodata is updated.

If instead of mirroring to a local directory, you want to make the changes directly into an existing Anaconda Repository instance, the cas-sync-api-v4 can be used. You'll need to use the dest_site config option:

```
mirror_dir: /home/data/mirror
remote_url: ""  # where to get miniconda and anaconda installers -- blank to skip
# possible platforms are: linux-64, linux-32, osx-64, win-32, win-64
platforms:
  - linux-64
  - win-32
blacklist:
  - dnspython
  - shapely
  - gdal
dest_site: some_site
```

Make sure that the site is defined in the anaconda config and you're properly logged into it before invoking cas-sync-api-v4.

More extensive information about the cas-mirror tool's functionality and configurable options is available at *Customizing mirrors*.

Using cas-installer

The cas-installer tool makes an environment installer, which is a bash script or Windows executable file that can be run on any machine to install an exact copy of a conda environment and its packages on that machine.

Installing cas-installer

A token from Anaconda is required to install cas-installer, and you should have received it when your organization purchased Repository, Workgroup or Enterprise. If you no longer have access to your token, submit a support ticket or contact us at Enterprise Support. You can also email support at the email address given to you by your sales representative.

When you have the token, run:

```
export TOKEN=<your_anaconda_cloud-token>
conda config --add channels https://conda.anaconda.org/t/$TOKEN/anaconda-server
```

Because this tool allows you to create an installer for a conda environment, it is important that the cas-installer package is installed into the root conda environment, not root user. The following command ensures that this happens:

```
conda install -n root cas-installer=1.3.2
```

Using the cas-installer command

Once installed, the `cas-installer` command is available:

```
cas-installer -h
```

The command takes an installer specification file as its argument, which specifies the name of the installer, the conda channel to pull packages from, the conda packages included in the installer, and so on.

EXAMPLE:

```
# ----------------- required -----------------
# name
name: test

# channels to pull packages from
# The &channels creates a back reference so that it can be reused as
# *channels in the conda_default_channels section below.
channels: &channels
  - https://repo.continuum.io/pkgs/free/

# specifications
specs:
  - python
  - grin

# ----------------- optional -----------------
# platform e.g. linux-32, osx-64, win-32 defaults to current platform
# platform: linux-64

# The conda default channels which are used when running a conda which
# was installed be the cas-installer created: requires conda---3.6.2 or
# greater---in the specifications. The *channels is a YAML reference to
# &channels above. It inserts all the channels from the channels key, so
# that they do not have to be typed twice.

conda_default_channels: *channels

# installer filename
# installer_filename: grin.sh

# default install prefix
default_prefix: /opt/anaconda
```

For Windows, the tool creates nsis-based .exe installers, which can only be created on a Windows platform, although the architecture may be different. For Unix, the tool creates bash-based .sh installer, which can only be created on Unix—Linux or macOS—systems.

Updating Repository

CAUTION: You must have a tested backup of your installation before starting the update process. If updating more than one version, all updates must be performed in sequential order.

Your support representative can provide you with a download URL for an updated Repository installer.

To update to the latest Repository release:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

NOTE: To use a Repository version from 2.33.3 through 2.33.10 and Anaconda Enterprise Notebooks with single sign-on (SSO), you must set `USE_SERVER_BASED_SESSIONS: false` in the Repository configuration. This setting affects the network security properties of AEN and Repository. Specifically, if `USE_SERVER_BASED_SESSIONS` is set to `false`, and if a new cross-site scripting (XSS) vulnerability is discovered, it could expose an additional server fixation vulnerability. Please discuss this with your Anaconda representative and be sure the feature is compatible with your network requirements before setting `USE_SERVER_BASED_SESSIONS: false`.

NOTE: As of Repository 2.33.8, the `fs_storage_root` configuration setting is mandatory for local filesystem storage and the Repository server will not run without it. You can set it with this command:

```
anaconda-server-config --set fs_storage_root /opt/anaconda-server/package-storage
```

You may replace `/opt/anaconda-server/package-storage` with any location owned by the `anaconda-server` user.

Please contact your Professional Support Team contact or sales person if you have any questions or problems regarding the update.

Updating to current and previous versions

Updating to 2.33:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

NOTE: To use a Repository version from 2.33.3 through 2.33.10 and Anaconda Enterprise Notebooks with single sign-on (SSO), you must set `USE_SERVER_BASED_SESSIONS: false` in the Repository configuration. This setting affects the network security properties of AEN and Repository. Specifically, if `USE_SERVER_BASED_SESSIONS` is set to `false`, and if a new cross-site scripting (XSS) vulnerability is discovered, it could expose an additional server fixation vulnerability. Please discuss this with your Anaconda representative and be sure the feature is compatible with your network requirements before setting `USE_SERVER_BASED_SESSIONS: false`.

NOTE: As of Repository 2.33.8, the `fs_storage_root` configuration setting is mandatory for local filesystem storage and the Repository server will not run without it. You can set it with this command:

```
anaconda-server-config --set fs_storage_root /opt/anaconda-server/package-storage
```

You may replace /opt/anaconda-server/package-storage with any location owned by the anaconda-server user.

Updating to 2.32:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.31:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.30:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.29:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.28:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.27:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.26.6:

The logging configuration can be removed. Logs have been moved to `$PREFIX/var/log/anaconda-server/`:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
anaconda-server-config --remove LOGGING
supervisorctl stop all
supervisorctl start all
```

Updating to 2.26:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.25:

```
curl '$INSTALLER_URL' > anaconda_repository.sh
bash anaconda_repository.sh -u
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.24:

```
conda update binstar-server binstar-static anaconda-client
anaconda-server-db-setup --execute
anaconda-server-install-supervisord-config.sh
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.23:

```
conda update binstar-server binstar-static anaconda-client
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.22:

```
conda update binstar-server binstar-static anaconda-client
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.21:

```
conda update binstar-server binstar-static anaconda-client
anaconda-server-db-setup --execute
supervisorctl stop all
```

(continues on next page)

(continued from previous page)

```
supervisorctl reload
supervisorctl start all
```

Updating to 2.20:

```
conda update binstar-server binstar-static anaconda-client
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.19:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl stop all
supervisorctl reload
supervisorctl start all
```

Updating to 2.18:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl stop
supervisorctl reload
supervisorctl start all
```

Updating to 2.17:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl stop
supervisorctl reload
supervisorctl start all
```

Updating to 2.16:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.15:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.14:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.13:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
anaconda-server-config --config-file /etc/binstar/config.yaml --set LABEL_NAME "
↪'channel'"
supervisorctl restart all
```

Updating to 2.12:

```
conda update binstar-server anaconda-client anaconda-build
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.9:

```
conda update binstar-static binstar-server cas-mirror
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.8:

```
conda update binstar-static binstar-server cas-mirror
anaconda-server-db-setup --execute
supervisorctl restart all
```

Updating to 2.6.0:

```
conda update binstar-server
conda install cas-mirror
```

Updating to 2.5.1:

```
conda update binstar-server
```

Updating to 2.3:

```
conda update binstar-server
conda install cas-mirror
```

Updating to 2.2:

```
conda update binstar-server
```

Uninstalling Repository

Before deleting Repository, you may want to make a backup for security reasons. For suggestions on mongo backups, see https://docs.mongodb.org/manual/reference/program/mongodump/ .

To delete Repository:

1. Check the file storage path:

   ```
   anaconda-server-config --get fs_storage_root
   ```

2. Delete the contents of `/home/anaconda-server/repo`:

```
rm -rf /home/anaconda-server/repo
```

3. Delete the appropriate MongoDB database, “binstar.”
4. Delete the contents of `/etc/binstar`:

```
rm -rf /etc/binstar
```

5. Delete the contents of the Repository file storage path.

Troubleshooting

This page provides instructions for troubleshooting issues that may occur with your Anaconda Repository installation.

Cannot connect to the server on port x

This could be because you are behind a firewall. Check if your IPTables rules are blocking your ports:

```
iptables -L -n
```

If a rule blocks a port you want to use, then you must allow the port:

```
sudo iptables -t nat -F
sudo iptables -A INPUT -p tcp -m tcp --dport <PORT> -j ACCEPT
sudo service iptables save
sudo service iptables restart
```

Error: “No environment named ‘search’ exists in. . . ” on Windows

If Anaconda Client is not yet installed and you try to search for a package on Anaconda.org using the `anaconda` command, you may receive the following error message:

```
C:\Users\USERNAME>anaconda search -t conda PACKAGE
No environment named "search" exists in C:\anaconda\envs
```

This error occurs because the Windows version of Anaconda contains an `anaconda.bat` file that is used for setting environment paths and switching environments. If Client is not installed, this batch file is called instead. Once you install Client, the Anaconda `search` command will work:

```
conda install anaconda-client
anaconda search -t conda PACKAGE
```

Anaconda upload fails while behind a reverse proxy

When configuring Client to connect to a Repository behind a reverse proxy, the `anaconda upload` command may appear to try connecting to the internal hostname rather than the external configured one.

This can be corrected in the settings of the reverse proxy, such as NGINX or Apache.

In NGINX, add the setting `proxy_set_header Host $host;` to access the internal host with the external hostname.

In Apache, turn on the option `ProxyPreserveHost`.

Other reverse proxies each have their own settings to handle hostnames correctly.

EXAMPLE: Some other reverse proxies use a settings syntax such as `http_proxy=id:passwd@proxyhost:port`.

Start Repository application as a foreground process

Repository should normally be started as a daemon. For troubleshooting, it can instead be started as a foreground process on a specified port:

```
anaconda-server --port 8080
```

Stop the application with Control-C.

Administrative commands

Many of these actions can be done in the web interface. This command reference is for those administrators who prefer to use command line shortcuts.

In all examples below, replace "jsmith" with the name of the user whose settings you wish to change.

Reset a user's password interactively:

```
anaconda-server-admin reset-password jsmith
```

The above command will prompt you to enter the new password twice. You may also reset the password directly:

```
anaconda-server-admin reset-password --password abcDEF123! jsmith
```

NOTE: Replace "abcDEF123!" with the new password.

Set a user's plan to a free and unlimited plan:

```
anaconda-server-admin free-unlimited-plan jsmith
```

Set all users with a given email domain to a free and unlimited plan:

```
anaconda-server-admin free-unlimited-plan-for-domain yourdomain.com
```

You can do a "dry run" of the command to display what the command will do without changing anything:

```
anaconda-server-admin free-unlimited-plan-for-domain --dry-run yourdomain.com
```

NOTE: Replace "yourdomain.com" with the domain whose users you wish to upgrade.

Give the user the privileges of a superuser or remove them:

```
anaconda-server-admin set-superuser jsmith
anaconda-server-admin unset-superuser jsmith
```

Give the user the privileges of a staff user or remove them:

```
anaconda-server-admin set-staff jsmith
anaconda-server-admin unset-staff jsmith
```

Change a user’s login name (username):

```
anaconda-server-admin move-user old_name new_name
```

NOTE: Replace “old_name” with the current username, and “new_name” with the new username.

Ensure the files recorded in the database exist.

> anaconda-server-admin verify-storage

Ensure that files recorded in the database exist and have the correct checksum:

```
anaconda-server-admin verify-storage --md5
```

List the key names of files with problems:

```
anaconda-server-admin verify-storage --list-files
```

Scan the storage for unused files and delete them:

```
anaconda-server-admin clean-storage
```

Update the bundled installers:

```
anaconda-server-admin update-installers
```

Delete a user:

```
anaconda-server-admin delete-user jsmith
```

Ser or unset “read only” mode:

```
anaconda-server-admin read-only --enable/--disable
```

Convert a regular user account to an organization and add another user to the owners group:

```
anaconda-server-convert-account to-organization --owner some_user regular_user
```

Convert an organization into a regular user account:

```
anaconda-server-convert-account to-user some_organization
```

FAQs

What is Anaconda Repository?

Anaconda Repository is package management server software that makes it easy to find, access, store and share public and private notebooks, projects, installers, environments, and conda and PyPI packages. Repository also makes it easy to stay current with updates made to the packages and environments you are using.

Many enterprises have customized local instances of Repository. Anaconda also makes an instance of Repository available for public use at Anaconda Cloud.

What kind of packages does Repository support?

Repository supports any type of package. It is primarily used for conda, PyPI and R packages, as well as notebooks and environments.

What is Anaconda?

Anaconda is a software development and consulting company of passionate, open source advocates based in Austin, Texas, USA. We are committed to the open source community. We created the Anaconda Python distribution and contribute to many other open source-based data analytics tools. You can find out more about us by reading our story.

How do I get started with Repository?

If you have access to Repository, you can search, download and install hundreds of public packages without having an account.

If you want to upload packages to Repository, you need to sign up for a Repository account, get Anaconda and the Anaconda Client. For more information, see *Creating an account* or ask your system administrator.

What is an organization account, and how is it different from an individual account?

An organization account allows multiple individual users to administer packages and have more control over package access by other users. An individual account is for use by one person.

Who can upload packages to an organization?

Only users who are co-owners of an organization may upload packages to that organization. Administrators who are not co-owners cannot upload packages to the organization. Users who are members of groups with read/write access but who are not co-owners, cannot upload packages to the organization.

Help and support

Your organization receives Professional Support with your purchase of Anaconda Repository. Please contact your system administrator for help.

Joining community support

You are also welcome to join our community support mailing lists for both Anaconda and conda. On these lists you can ask questions, answer questions and discuss ways to use Anaconda. You can also submit requests for new features and make any other comments you may have.

Note that the community support forums cannot provide Anaconda Repository support.

Reporting a bug

Issues with Repository are tracked on GitHub. If you think you have found a bug, search to see if it has been reported, and report it if no one else has.

Release notes

The Anaconda Repository 2.33 release is available to all Anaconda Repository customers as of September 19, 2017.

NOTE: If you have a subscription but do not have a license, contact support to receive that license. Otherwise contact sales to acquire it.

Administrators can update to the new Anaconda Repository release as described in *Updating Repository*.

Please contact your enterprise support representative if you have any questions or problems regarding the release.

Changelog

SEE ALSO: *update instructions for current and past versions*.

2.33.27 - 2018-07-30

User facing changes

- Remove /about/pricing
- Allow the disabling of new Personal and Organization private accounts via Stripe API

2.33.26 - 2018-07-19

User facing changes

- Captcha on organization creation

2.33.25 - 2018-07-19

User facing changes

- Captcha on account creation

2.33.24 - 2018-07-03

User facing changes

- Remove ‘Pricing’ links from header and footer
- Updated compatibility docs

Non visible changes

- Cloudflare cache invalidation
- Fixed pypi simple index mirroring
- Fixed installers and environments downloads on read-only mode
- Raise exception when an invalid USER_REGEX value is used
- Fixed “All labels” file filter
- Fix broken session

2.33.23 - 2018-05-23

User facing changes

- Policy change notice

2.33.22 - 2018-05-21

User facing changes

- Support for ppc64le installers
- Constructor form advanced option validation
- Added explicit Redhat versions to requirements
- Improved API docs regarding the basename of files
- Search includes package summaries
- Bug fixes and broken link fixes
- Added note about compatibility between LDAPS and START_TLS
- Fixed pypi mirror config example

2.33.21 - 2018-05-03

User facing changes

- Added advanced options on installer creation form

- Warn about label `main` when manually editing labels
- Package summary shows summary of latest `main` release

2.33.20 - 2018-04-24

Non visible changes

- Support for serving files from the origin

2.33.19 - 2018-04-19

User facing changes

- Flash warning when label doesn't exist
- Searching using unicode characters
- Notebooks without labels are rendered

Admin facing changes

- Added option to customize constructor temp dir
- Forbid supplying the same account as owner when converting account to org

Non visible changes

- Fix download stats link
- Avoid saving generated zip for installers
- Add quotes on paths in AIC templates
- Add support for unicode characters in version
- Removed server header from responses
- Fixed label validation

2.33.18 - 2018-04-03

User facing changes

- Added icons to the repo files page
- Changed package search placeholder
- Updated notebook upload icon
- Show warning when copying a label onto itself
- Fixed navbar spacing when logged out
- Fixed org feed links

Admin facing changes

- Re-use owners group when converting account to org

Non visible changes

- Fixed redirection after label operations

- Fixed transferring from org to superuser account
- Fixed update of installers and parcels
- Label filters are reset if the label set changes

2.33.17 - 2018-03-09

Non visible changes

- Next URL whitelist

2.33.16 - 2018-03-08

User facing changes

- Add warnings when removing main label
- Update last seen on account change
- Allow signing up with an orgs email
- Show favorites on dropdown menu for orgs
- Show settings tab for collaborators
- Fixed LDAP TLS docs

Admin facing changes

- Allow superusers to be organization admins
- Add billing history

Non visible changes

- Added validation of build number
- Add scheme to AIC templates
- Removed hotjar
- Remove marketo

2.33.15 - 2018-02-27

Non visible changes

- Standardize If-Modified-Since handling

2.33.14 - 2018-02-20

Non visible changes

- Fix HEAD support by stripping quotes from s3's object

2.33.13 - 2018-02-19

User facing changes

- Custom ordering of notebooks and environments
- Added tooltips showing the exact upload date and time of files on the repo page
- New command to convert regular user accounts to organizations and back
- Last upload date on package and installer info pages
- Fixed error message wording when deleting packages on groups
- Fixed error message when deleting packages, environments and notebooks
- Fixed the wording on the empty dashboard cards
- Fixed navbar fonts on IE11
- Fixed file management actions for package collaborators
- Fixed transferring of packages to and from the same user
- Show file actions for collaborating organizations
- Forbid downloads on read-only mode
- Allow collaborators with admin rights to delete ownables

Admin facing changes

- Mirror tools now create organization accounts by default

Non visible changes

- Add custom X-Anaconda-Lockdown and X-Anaconda-Read-Only response headers
- Use database info to construct filenames of conda downloads
- Fixed support for HEAD method on download endpoints
- Added extra validation of the basename on conda package uploads
- Use upserts instead of inserts to stage files

2.33.12 - 2018-02-07

User facing changes

- Updated terms of service

2.33.11 - 2018-02-06

User facing changes

- Added badge for the date of the latest release
- Added badge for platform support
- Show warning if no revision is selected when working with projects
- Updated terms of service
- Favorites are now shown on org dashboards

- A warning message is now shown when all packages are added to a given group

Admin facing changes

- Disable password reset admin option while using LDAP

Non visible changes

- Added proper HEAD support on download endpoints
- Disabled USE_SERVER_BASED_SESSIONS by default
- Disabled database based settings
- Strengthened validation of labels
- Fixed popups for operations when no files or packages are selected
- Fixed deletion of files by collaborators
- Fixed access to static content while on LOCK_DOWN
- Fixed transfer of ownership of items between orgs

2.33.10 - 2018-01-19

User facing changes

- Added “noarch” to the platforms mirrored by anaconda-server-sync-conda

2.33.9 - 2018-01-16

Admin facing changes

- Made fs_storage_root setting mandatory only for local filesystem storage

2.33.8 - 2018-01-15

User facing changes

- Updated LDAP docs
- Hide actions on a user’s repo page when viewing it with an org

Admin facing changes

- Made fs_storage_root setting mandatory
- Block uploading a new license when read-only mode is enabled

Non visible changes

- Fixed exception logging on anaconda-server-sync-conda
- Remove debug code
- Better handling of next url redirect on login link
- Fixed response of repodata endpoint when an invalid If-Valid-Since header is given
- Remove install instructions from label table
- Fix orgs favorites

- Removed suggestions from confirmation dialogs

2.33.7 - 2017-12-11

User facing changes

- Updated support links
- Added activity feed item for installer upload
- Clarified pip install example command
- Added close icon for installer log popup
- Organizations are able to see their email on the profile page

Admin facing changes

- Removed READ_ONLY config option. Added admin cli tool to change read only state

Non visible changes

- Added index in database for package '_name' attribute
- Several fixes on license creation page
- Added proper message to groups permission set
- Fixed actors for some feed items actions
- Validate that users exists when adding a group member
- E-mail confirmation error message for organizations
- Enable read-only option with repo page
- Fixed profile description not being wrapped

2.33.6 - 2017-11-27

Added

- Filter for authenticated packages in search/favorites view
- Updated message for input field when copying label
- Updated support links
- Added email notification when group member is added
- Installation info for R and pypi packages using labels
- Support for defining standard labels
- Added support to remove user using anaconda-server-admin
- Show licence url for packages, environments and notebooks
- Validation for empty fields in credit card info for plan upgrade

Fixed

- Updated instructions to create initial user
- Prevent organizations to have admin access for ownables of its original user

- Fixed package view when a release description is not a string
- Panels for ownables not showing in profile page if there are no packages
- Set limit to installers log height to prevent modal going below the footer
- Validation to prevent organization adding itself to one of his groups
- Set invalid license messages on mirroring script to debug instead of warning
- Generated tokens can be viewed without password prompt if kerberos authentication is used
- Fixed link on R package label page
- Remove password reset option if auth_type is not native
- Replaced urls for R packages sources
- Fixed counts in billing overview page
- Added quotes to install instructions to avoid issues with spaces
- Documentation tooltip in conda packages
- Hide transfer modal if there are no accounts to transfer to
- Use dashboard used instead of current user as actor for feed items
- Fixed feed url links
- Redirection for labels on package files list
- Removed duplicated feed item on package transfer
- Fixed installer version validation message
- Fixed query to retrieve non-private packages
- Fixed dead links to deleted projects on the feed
- Expanded LDAP groups docs
- Handling of duplicate package exceptions on API
- Remove word kapsel from email when collaborator is added

2.33.5 - 2017-11-07

Added

- Added a CLI tool to manage group membership
- Missing tooltips on header and admin section
- Added some missing feed items
- Account search now uses both names and emails
- Filter for authenticated packages

Fixed

- File info modal now works with list attributes that contain dicts
- Validate uploaded environment name
- Improved UX of installer creation form
- Updated feed icon for group collaboration removal

- Show all feed items related to a particular ownable in the History tab for that ownable
- Added main channel to default mirror config
- Generic exceptions during LDAP auth are now caught and logged
- Fixed supervisord script crontab option
- Updated read only rules on projects and installers
- Disable autocomplete suggestions for confirmation input fields
- Added authenticated packages to the billing package limit notice
- Add quotes around conda install help message if label has spaces
- Specify correct package type on tooltip text for label removal
- Updated flask-login-ldap dependency
- Validate name of copied label
- Removed validation of label name on deletion
- Removed duplicate HTTP headers on cached responses
- Do not allow pypi packages in installers
- Updated mirroring docs
- Make installers/projects summary optional
- Replaced some occurrences of word kapsel
- Align upvote icon
- Set package access from packages list
- Only owners can upload installers/environments to its own channel

2.33.4 - 2017-10-24

Added

- Use environment variable to set initial user's password
- Usernames blacklist
- Show projects and installers summaries on header
- Added tooltips to package page buttons
- Instructions to generate tokens for organizations
- Feed items for projects and installers
- Settings for session timeout
- Supervisor script creates folder for extra config
- Updated EULA
- Set private packages and storage to unlimited individually
- Added progress indicator on installer upload
- Command to mirror only latest versions of conda packages

Fixed

- Fixed creation of private packages from the API
- Feed now uses the package database when it doesn't know the package type
- Hidden installers empty panel on profile page
- Infer access attribute from other attributes when adding package
- Incorrect logging of user downloads
- Show 'Set access' options for organizations in packages list
- Flash error messages when an errors occur on LDAP admin page
- Replaced word kapsel with project on flash messages
- Changed s3 content-disposition of anaconda server installers
- Date ranges for stats in admin page
- Transfer projects with the same name as a deleted project
- Remove package groups when package is archived
- No longer is possible to upload expired licenses
- Hide brand from delete user modal if user is an organization
- Package and environment file modal style issue
- Filter public packages from package search in admin
- Allow access to ownable settings to collaborators with 'write' permissions
- Fixed pypi installer tooltip
- In admin user account, prevent setting lower storage than the used storage
- Delete groups when the org is removed
- Removed add-ons page
- Handling missing package after deleting files
- Do not allow to create tokens expiring today
- Fixed redirections to packages on feed items
- Show installer and project feed items in history tab
- Show all collaborators of an organization's package
- Fix issues with package icons on dashboard
- Sorted tabs in group settings
- Always display collaborators tab as 'Collaborators'

2.33.3 - 2017-10-20

Added

- Added support for server based sessions

2.33.2 - 2017-10-10

Fixed

- Exception in admin after updating private packages for a user

2.33.1 - 2017-10-03

Added

- New feed items for group membership and groups collaborations
- Download stats for files API endpoint
- Option to set amount of private packages for a user from admin
- Improved license creation page
- Added distribution_types to downloads feed
- Set packages access as authenticated from packages list
- Added option –authenticated to anaconda-server-sync-conda
- Added conda-build as dependency
- Relaxed expired tokens restrictions for public endpoints
- Add organizations as collaborators for packages, environments and notebooks
- Send email when adding collaborator to a project/installer

Fixed

- Link to docs in packages view
- Catch all exceptions raised when loading environment file
- Return json responses on api calls when an error is encountered
- Error message when uploading an invalid installer file
- Group permissions moved to the settings
- Fix wrong autocomplete using firefox
- Fix typeahead initial suggestions in installers form
- Updated callout in contact us form
- Hiding package access settings for collaborators
- Fixed refresh when closing user menu on the navbar
- Show info about installers downloads on admin interface
- Fixed downloads stats on admin
- Prevent adding package owner as package collaborator
- Storing package_type when API package upload
- Collaborators can now access a package's history page
- API docs are back up again
- Refactored mirroring tools

- Fixed incorrect links from feed items

2.33.0 - 2017-09-19

Added

- SUPERUSER_SEARCH to set superuser status in LDAP
- File format validation on installer upload
- Show which users are admin in users list
- Use simillar settings for typeahead package suggestions
- Require user to be logged in to see user typeahead suggestions
- Cleaned output from test suite
- READ_ONLY mode setting and admin option
- Added tooltips to social media icons on footer
- Hide license download buttons from add ons page
- PAM authentication support
- Added reCAPTCHA to contact us form

Fixed

- Exception in group collaborations list for a package after group delete
- Maintain consistency in redirections after item deletion
- Fix group link in project collaborator view
- Flash message after issues with email validation
- Exception on admin downloads list for a user when package/file was had no owner
- Change dashboard user on item transfer
- Prevent adding current user as a collaborator
- Remove current owner from items ownership transfer options
- Fixed UI issues
- Validate name and version of installers only when full form is submitted
- Fixed some redirections to documentation in Anaconda Cloud
- Hiding delete package for collaborators
- Updated links to slideshare and youtube accounts

2.32.9 - 2017-09-15

Fixed

- Fixed forgot password link

2.32.8 - 2017-09-11

Fixed

- Temporarily disabled contact page for anaconda cloud

2.32.7 - 2017-09-07

Fixed

- Exception in token expiration warning code
- Fixed error when displaying a group that no longer exists

2.32.6 - 2017-09-06

Added

- Separated package groups collaborations in three tabs (packages, notebooks, environments)
- Added icons to all feed items
- Provided more info on feed for uploaded packages/environments/notebooks
- Unicode validation on signup form
- Package api returns builds and adds filter for search platform
- Validation for profile name
- Added option to upload all packages to a group at once
- Added page to see feed for a user
- Added some reserved names for packages
- Semantic versions validation for installers version field
- Warning header when token is about to expire
- Make favorites page public

Fixed

- Allowing anaconda login under lockdown
- Fixed potential exceptions on old cache code
- Changed typeahead environment query to use dashboard user
- Fixed issues with the upload of previously deleted installers
- Fixed Cache-Control headers on old repodata caching code
- Display all packages by default, not just only conda packages
- Show correct label for Groups & Collaborators depending on type of user
- Changed owner of uploaded installer to current dashboard user
- Remove groups permissions from all items after group delete
- Color schemes of some flashing messages
- Prevent project/installer transfer if recipient already has one with the same name

- Empty environment field from installer created from environment if the environment was deleted
- Fixed email validation when other user is logged in
- Maintain consistency on headers from dashboard
- Fixed token generalizations
- Exception on admin downloads list for a user when package/file was missing
- Fixed “View Docs” URL
- Use dashboard user in redirects after file delete

2.32.5 - 2017-08-29

Fixed

- Removed Continuum references

2.32.4 - 2017-08-28

Fixed

- Fixed old continuum links

2.32.3 - 2017-08-24

Fixed

- Navbar logo responsiveness issues

2.32.2 - 2017-08-24

Fixed

- Changed navbar buttons order to the new design schema

2.32.1 - 2017-08-24

Fixed

- Bigger logo on the navbar
- Fixed home page screenshot of the site

2.32.0 - 2017-08-22

Added

- New top bar button structure
- Added option ‘any’ to display all kind of packages in repo view
- Support for custom user avatar methods

- Updated *cas-mirror* docs
- Create installers uploading an environment file
- Parcel and management packs can be created from installers previously created
- SSL protocol version can now be customized
- Fresh design
- Stats API endpoint
- Updated all flash messages colors

Fixed

- Ownership of uploaded environments
- Doc links on the feed
- Styling of installer widget
- Transferred projects now appear in dashboard
- Environment validation
- Environment upload labels
- Show correct package type when transferring package/environment/notebook
- Increased panel sizes in profile and dashboard
- Limited items to display in panels
- Fixed collaborators view
- Potential cache related exception
- Made management packs template order deterministic
- Unsafe redirections
- Removed packages from groups when transferred
- Closed XSS vulnerabilities
- Contact us emails are sent from contact@anaconda.org with a reply-to header
- Fixed typeahead input field to add package collaborators
- Archive items when all their files are deleted
- Return NotFound on item details page when they don't contain any files
- Fixed stats report admin view
- Replaced occurrences of the word "package" for a more appropriate name depending on the package type

2.31.6 - 2017-08-08

Added

- Email notification when added as a collaborator
- Labels regex now distinguishes uppercase characters
- Added site export tool
- Upload option for installers

- Added button to review the build log next to each installer file
- Version specific landing page for packages
- Added group info to site export tool
- Upload option for environments

Fixed

- Layout issues on dashboard and repo pages
- Remove user from groups when the account is removed
- Remove user as a collaborator for installers/packages/projects/notebooks/environments when is deleted
- Show only projects with files in projects list
- CSV export of users
- Typeahead for multi-type packages
- Added user validation to remove collaborator form
- Metadata display on package API
- Cache key generation and diskcache size limit parameter
- Empty packages will no longer appear in search results

2.31.4 - 2017-08-03

Fixed

- Fixed repodata caching

2.31.3 - 2017-07-27

Fixed

- Fixed label validation

2.31.2 - 2017-07-24

Added

- Tool to check a file's checksum

Fixed

- Show conda packages install instructions only for available labels
- Solved exception raise by anaconda-server-admin clean-storage
- Hide empty packages/environments/notebooks from dashboard/profile page
- Solved exception raised creating an installer from an empty environment
- Fixed style issues with dashboard/profile page.

- Added help info in profile page
- Show correct icons in objects page
- Removed extra space from groups breadcrumb
- Fixed license not updating
- Hide License expired message overlay during session when alert is closed
- Fixed email sending on forgot password and forgot username

2.31.1 - 2017-07-13

Added

- Docs for LDAP timeout
- –clean-platforms option for anaconda-server-sync-conda
- Docs for backup and restore procedures based on the default installation
- Added setting for custom installers location
- Updated FontAwesome to 4.7.0

Fixed

- Displays the latest release data on the package page
- Fixed panel size in profile page and added scrollbars on overflow
- Show latest version available for each platform in conda packages
- Custom installers pre-configured to point to repo instance
- Exception when trying to display security log
- Display the correct username on the navigation bar
- Remove unused logging configuration
- Python tags on environment.yaml now parsed on installer creation from environment
- Omitting non-conda dependencies on installer creation from environment
- Updated documentation links
- Added missing R packages icons
- Unicode issue on Contact Us form for anaconda cloud
- Fixed CSS issue on Internet Explorer
- Solved issues with labels containing slashes and spaces
- Updated link to conda documentation
- Force pypi mirroring for a new mirror user
- Fixed validation when setting packages to private in bulk
- Fixed some UI issues with long names
- Fixed org creation on mirroring tools

- Labels link takes you to all type package listing
- Package type filter set to 'all' will not show notebooks/environments anymore
- Confirmation of package delete with username input works also using uppercase
- Correct order of search filters
- Fixed typeahead endpoints
- Fixed duplicated channels on custom installers
- Allowing slashes on token name delete action
- Disabled LDAP referrals by default

2.31.0 - 2017-06-28

Added

- Added USER_REGEX defaults to reference docs
- Added period (.) as a valid character for the default USER_REGEX
- Added diskcache based repodata caching
- Added license url validation for packages
- Option to set storage keyname to full path
- Show error message when attempting to add duplicated collaborators

Fixed

- CSS fixes on top navbar
- CSS fixes on group names
- Added filter for valid packages in installer creation
- Added support for deleted Strip accounts
- Stops adding/updating labels if the validation fails
- Tokens modal is no longer going below the bottom of the page
- Displaying credit card errors correctly
- Handling LDAP login error
- Fixed highlight of proyect settings tab
- Updated tqdm version to stop exception on mirror download
- Environments summary is no longer duplicated
- Display correct package summary after update
- Customized success message on upvote depending on package type
- Updated links to docs for labels
- Updated verbose exception
- Fixed notebook revisions links

- Optimized query to get latest package versions
- Updated all references to docs with correct links for cloud
- Removed top-level domain validation from profile URL
- Improved speed of show_channel endpoint
- Added validation to prevent duplicated packages on installers
- Changed default label filter for packages to 'all'
- Removed distinction of user menu based on username
- CSS issues with long names
- Fixed long project names overlapping

2.30.3 - 2017-06-06

Added

- Added feeds for kapsel creation/removal/new revision

Fixed

- Added some more plural forms for flash messages
- Show project description from latest revision instead of project summary
- Fixed window installer configuration files
- Moved project history to settings
- Projects and installers on the same row
- Fixed installer creation under LOCK_DOWN
- Fixed settings tab highlight on installer admin page
- Fixed handling of empty page param on search page
- Fixed max-age overflow on authentications endpoint
- Fixed project creation time
- Fixed installer form when python package has no releases
- Fixed subscriptions plans link
- Fixed unicode issues on contact form
- Merged LDAP login logic
- CSS clean up for groups with long names

## 2.30.2 - 2017-05-24

- Fixed authentications endpoint

## 2.30.1 - 2017-05-24

Added

- Warning before deleting a package and all of its messages
- Tool to check orphan files and packages
- Added option to lock down all public pages
- Added link to contact us for custom plans.

Fixed

- Display error message when config file is not found
- Fixes resend confirmation email
- Retrying on 502 error while mirroring conda
- Replace non-ascii characters from filename when downloading a file
- Added password validation to password reset form
- Handling 404 on s3 key_exists
- Removed files and packages will appear on package history
- Using user's name and email on contact emails
- Changed Resend Email label to Password Reset in admin page
- Added placeholder to collaborators form
- Added singular form messages in flash notifications
- Fixed popup label for pkg/nbk/env settings
- Fixed scrollbar blocking content in installers documentation
- Added searchbox on navigation header for non-authenticated users
- Added more database indexes for better query performance
- Removed admin monitor page
- Increased request timeout default to 120 seconds
- Updated the mirror configuration examples
- Fixed URL for pricing info
- Fixed token creation API
- Removed outdated mirror documentation
- Added migration to normalize files data
- Added brand as key in api endpoint
- Declined credit card info is no longer stored
- Improved UI for group members page

## 2.30.0 - 2017-05-08

Added

- Added EULA to the installer
- Contact Us form now sends emails to support@anaconda.org
- Create organization instead of user on mirroring tools

Fixed

- Fixed anaconda-server-sync-conda settings message
- Secured web helpers views
- Removed unused test endpoints
- Fixed insecure groups endpoint
- Fixed filename too long exception on type filter for installers
- Server side encryption on S3 storage
- Fixed documentation link in /settings/access API token page
- Year in footer matches current year
- Fixed password restrictions checks
- Replaced binstar-* message for anaconda-server-* on mirroring script
- Added missing instruction to Project upload instructions
- Fixed the way we load the license data from the database
- Added validation for reserved usernames
- Fixed package set-access on firefox
- Fixed encoding error on package information page
- Changed error message on installer creation form

## 2.29.1 - 2017-04-19

Fixed

- Fixed access to LDAP views
- Removed unused remove_user view
- Add support for expired marketo access token
- Fixed exception on installer creation

## 2.29.0 - 2017-04-19

Added

- Visibility on Projects feature
- Sortable account list by package count
- Change plan button for organizations

- Add lead source to marketo requests

Fixed

- Fixed installers downloads
- Set S3 addressing style to "virtual"
- Fixed S3 ETag processing
- Fixed handling of missing arch attribute on search
- Typeahead endpoint access limits
- Changed supervisord runtime files location
- Allowing numeric named installers
- Allows installer creation with environment with url on the channel list
- Fixed typeahead on installer creation form
- Remove unreachable code
- Fixed flake8 findings
- Fixed installer collaborators form
- Fixed remove unlabeled files
- Fixed repo access admin for organizations

## 2.28.1 - 2017-04-03

Added

- Installers can now be created from uploaded environments

Fixed

- Fixed outdated version badges
- Add "jessie" and "sles12" parcel suffixes
- Only allows alphanumeric characters on installers name
- Fix group installers page
- Fix edition of existing installers

2.28 - 2017-03-22

Added

- Add UI to delete and set access of packages on the repo page
- Add UI to delete files in the files section of the repo page
- Add UI to set and unset superuser and staff status on admin page
- Staff users can now access licensing
- Pagination on history for the account admin page

Fixed

- Config set using *anaconda-server-config*
- Username on page titles
- Fixed anaconda-server-sync-conda issue with local repos
- CSV export on emails with special characters
- Package label filtering
- Signup password validation error message
- Exception requesting non existing file url
- Fixed email confirmation for organizations
- Text overflow when username is too long
- Remove all user packages in a single action to avoid filling the queue
- Checking user existence on reset password
- Installer URL shows zip extension

2.27.5 - 2017-03-14

Fixed

- Added boto dependency back

2.27.4 - 2017-03-03

Added

- Support for S3 regions that only use V4 signatures
- Support for S3 server-side encryption
- Support for custom PyPI repo sync
- New releases overrides package's description, summary, license and icon
- Update mongodb to 3.4

Fixed

- API endpoint *DELETE /dist/{owner_login}/{package_name}/{version}/-/{_id}* should delete the file with the associated ID
- Email validation on profile page
- Downloading files with spaces or special characters should result in the correct filename
- Some documentation URLs showed up without styling
- Fixed text overlap in admin deployment page
- */downloads* installers should generate configuration files correctly
- "Not Found" errors are more consistent and clear
- Package search timeout
- Displaying validation in the popup on account's admin page

2.27.3 - 2017-03-02

Fixed

- Add support for string license attribute

2.27.2 - 2017-02-27

Fixed

- Remove pyc from ambari mpack templates folder

2.27.1 - 2017-02-23

Added

- Limit to the cache

Fixed

- Added missing ambari mpack templates folder

2.27.0 - 2017-02-15

Added

- Support for generating custom Hortonworks/Ambari management packs
- License and license url to packages api
- Update email confirmation code to more secure and flexible hash
- Support for multiple users with same email (if option enabled)
- Admin support to remove an account
- Admin support to change storage size or change plan to free unlimited
- Package versions on installers and parcels are now optional
- Specify a configuration file with the environment variable ANACONDA_SERVER_CONFIG

Fixed

- Noarch repodata should not include files that are missing platform and arch
- Fix attribute errors kapsel unit tests
- Mirror configuration *python_versions* should not require quotes
- Add link to package on Favorites page breadcrumbs.
- Improve support for POWER and ARM architectures.
- Provide a useful error page when MongoDB is unreachable.
- Fix notebook and env with same name
- Suppress form errors when adding or removing package/channels
- anaconda-server-config will work on the config file you actually have
- Remove temporary redirects on user settings and org groups settings

- Disable empty as a valid label/channel name
- Fix organization name on group membership view
- Disable Reset Password if it is the only button

2.26.5 - 2017-01-30

- Remove experimental feature from display.

2.26.4 - 2017-01-30

Fixed

- Some files would not be cleaned up correctly when the corresponding user was deleted in the interface. Added a migration to clean up any existing unused files.
- Add *anaconda-server-admin clean-storage* command to clean up unused files.
- Fix csv column order on package info at the admin page.
- Render URLs in the package summary as links.
- Display correct breadcrumb for a selected environment.
- Sort labels on package page alphabetically.
- Pluralize storage information correctly.

2.26.3 - 2017-01-10

- Added AnacondaCON promo to Anaconda Cloud

2.26.2 - 2017-01-06

Added

- Conda repodata is now cached more frequently.
- Constructor installer creation will now be terminated if it takes longer than 60 seconds.

Fixed

- An error that occurred when PyPI packages that were deleted were re-uploaded.

Anaconda repository 2.26.0 - 2016-12-19

- Anaconda Repository has a new mirroring tool with reproducible results, and improved support for delta mirroring.

Anaconda repository 2.25.0 - 2016-11-30

- Anaconda Repository is now distributed as a self-contained installer.

Anaconda repository 2.24.4 - 2016-11-17

- FIX: issues with async workers PR #3120, #3123
- FIX: Issue with sending forgotten username. PR #3120
- LOGGING: log everything to stdout. supervisord script will now log to file instead of syslog PR #3106
- KAPSEL: Remove kapsel uploader PR #3107

Anaconda repository 2.24.0 - 2016-11-09

- Documentation updates
- Fix task queue (removed mtq library)
- Improve performance of PyPI simple index page
- Move licensing code to anaconda_platform.component.licensing
- Add hotjar (cloud)
- Move username regex to check into settings
- Parcels: Add anaconda and python 2.7 as default packages
- Added AIC (Anaconda Installer Configuration) installers
- Remove Kapsel Execution

Anaconda repository 2.23.1 - 2016-10-25

Added

- Gevent in as the server worker_class
- Remove check of key existence in s3 for anaconda.org

Anaconda repository 2.22.0 - 2016-10-18

Fixed

- API: added correct handling when user is deleted
- DOWNLOADS: allow unlimited storage of download stats
- UI: terms and conditions link was incorrectly escaped
- NOTEBOOKS: fix sorting of notebook versions
- REPO: copied package files would sometimes return 404
- UI: added default sorting to more tables
- REPO: usability and functionality fixes for CDH parcel generation

Anaconda repository 2.21.0 - 2016-09-29

Added

- REPO: users can *create custom CDH parcels through Anaconda Repository*
- UI: standardize sorting on tables
- UI: the software version is included in the footer of Anaconda Repository

Fixed

- NOTEBOOKS: added iframe sandboxing to notebooks
- NOTEBOOKS: fixed rendering of thumbnails uploaded by *nb_anacondacloud*
- REPO: copied package files were sometimes incorrectly garbage collected

Changed

- UI: The pages on the *conda.anaconda.org* and *pypi.anaconda.org* domains redirect to *anaconda.org*.

Removed

- BUILD: The deprecated build feature has been removed from Anaconda Repository.

Anaconda repository 2.20.4 - 2016-09-26

Fixed

- NOTEBOOKS: links in the notebook will open in the browser window directly, instead of inside of a frame.

Anaconda repository 2.20.3 - 2016-09-20

Fixed

- Allow numeric usernames

Anaconda repository 2.20.2 - 2016-08-18

Added

- REPO: package types will correctly update from added files (#2492)
- UI: cluster pages now list apps associated with that cluster
- Notebooks larger than 25mb will not be rendered (#2336)
- API: the endpoint `/user/{account}/downloads/{start}--{end}` now provides download activity aggregated by package for an account.

Fixed

- Improve the performance of the security feed (#2335)

Changed

- UI: the reminder to use beta will be hidden for 24 hours when a user clicks the "close" button.
- BUILD: remove welcome to build message for build deprecation notice.
- UI: rename project to kapsel everywhere (except imports) (#2563)

- Collaborators page updated to new groups API (#2512)

Anaconda repository 2.19.5 - 2016-08-04

Fixed

- Fixed generation of URLs to user notebook content server over https

Anaconda repository 2.19.4 - 2016-07-21

Fixed

- DB: improved group migration to handle more corner cases

Anaconda repository 2.19.2 - 2016-07-07

Fixed

- REPO: package types will correctly update from added files (#2492)

Anaconda repository 2.19.1 - 2016-07-07

Added

- Basic Cluster Pages
- Conda Caching - Conda endpoints now use Last-Modified/if-modified-since headers

Anaconda repository 2.18.0 - 2016-06-01

Added

- API: add an endpoint `/user/{account}/downloads/{start}--{end}` that provides an aggregated summary of download activity for an account.
- BUILD: automatically scroll to the bottom of log when new lines are appended
- REPO: improve support for R packages
- WEB: license warning message includes a link to the license configuration page

Fixed

- Users do not need to be logged into GitHub to trigger builds
- BUILD: remote address for workers will be detected correctly when running behind a proxy (#2036)
- API: LDAP users logging in for the first time via *anaconda login* are created correctly.
- PIP v8.1.2 fixed package name lookup

Anaconda repository 2.17.0 - 2016-04-18

Added

- Queue administration page that displays build worker details and history (#1847)
- An additional configuration file can be specified with the environment variable `ANACONDA_SERVER_CONFIG` or the command line argument `--config-file`
- Configuration files in the directory `$PREFIX/etc/anaconda-server/` will now be automatically loaded
- Better logging for login logic
- Failed logins are now recorded in the security log
- `docs.anaconda.org` content is now bundled with Anaconda Repository
- New privacy policy
- Project's API
- Show notebooks with nbpresent metadata as presentations (#1583)
- Can now view different versions of notebooks (#1764)
- Complete list of current settings on /admin/deployment (#1928)
- Decorator to validate params in a requests. (#1970)
- `api.anaconda.org` returns `conda_url`, `pypi_url` and `main_url` (#1984)
- `keyname` is displayed for superusers on the file details modal, allowing an administrator to locate a file on disk (#1985)

Fixed

- Editing package description should not add extra whitespace (#1710)
- Starred packages owned by other users will appear on the dashboard (#1706)
- Notebook output that is too wide will display a scroll-bar (#1581)
- Cleaned up styling on CI settings page (#1713)
- Security log details modal should appear for non-administrator users
- More graceful handling of notebook rendering failure (#1548)
- GitHub OAuth flow in the user settings page (#1931)
- Changed conda install instructions to use short channel name
- Group API exceptions when viewing group members (#1959)
- Fixed error in sample enterprise config file (#1968)

Changed

- Renamed “upvotes” to “favorites” (#1707)
- adjusted helptext for conda install from specific user channel (#1914)

Anaconda repository 2.16.6 - 2016-03-28

- Clean up build workers that have been idle too long (#1749)
- Add SMTP support for sending email (#1747)

- Add remote address of build workers to queue status (#1743)
- Toggleable sections in build log output
- Render progress bars in build log correctly
- Fix organization page redirects
- Improve search performance for “type:pypi” query (#1808)
- Fix duplicated build item when resubmitting via CLI (#1805)
- Fix sorting of file sizes (#1783)
- Fix small issue in package files page

Anaconda repository 2.16.0 - 2016-02-25

- Kerberos Authentication Support
- Several small fixes
- Performance improvements

Anaconda repository 2.15.5 - 2016-02-06

- Minor fixes and improvements
- Made build a separate component from the server
- Added license code
- Improved UI
- Better support for labels
- Improved performance on user profiles / security pages

Anaconda repository 2.14.1 - 2016-01-20

- Re-enabled the anaconda copy command
- Release renaming “channels” to “labels”
- Implemented new UI enhancements that included a new user dashboard
- Performed additional bug fixes

Anaconda repository 2.13.1 - 2016-01-12

- Implemented “My upvotes” page
- Added UI improvements to notebooks
- Implemented error logging fixes
- Performed additional bug fixes

Anaconda repository 2.12.3 - 2015-12-22

- Implemented UI Improvements to align with Anaconda branding, making A-Cloud easier to use
- Added confirmation after sending a message to support from the “contact us” page
- Removed left nav on dashboard
- Moved channel manager to the apps dropdown
- Made it easier for Academic users to access features by adding extended subdomain access for institutions
- Created a landing page for bug reporting to help A-Cloud users better self-select which repo for issue logging

Anaconda repository 2.11 - 2015-12-09

- Implemented UI Improvements
- Fixed minor issues
- Improved user profile
- Improved password validation
- Updated plans and pricing pages

Anaconda repository 2.10 - 2015-11-13

- Implemented UI Improvements

Anaconda repository 2.9 - 2015-09-28

- Implemented Upgrade/Setup script
- Offered free MKL Optimizations and free IOPro Addons for academic use
- Added command line scripts for user name changes
- Allowed port number configuration
- The Anaconda Server will subsequently be referred to as Anaconda repository

Anaconda Server 2.8 - 2015-08-27

- Added support for Jupyter 4.0
- Made passwords configurable
- Supplied better error messages

Anaconda Server 2.7 - 2015-07-28

- Implemented a new environment page
- Offered new channel features

Anaconda Server 2.6 - 2015-07-23

- Added support for conda noarch packages.
- Exposed additional distribution attributes via the API
- Changed Anaconda Server's underlying webserver from tornado to gunicorn

Anaconda Server 2.3 - 2015-04-24

- Increased specificity when mirroring the Anaconda repository including more robust license-blacklisting capacity and new python version-filtering capacity
- Implemented the ability to upload iPython notebooks to your Anaconda Server user account

Anaconda Server 2.2 - 2015-04-17

- Improved the user interface for channel-based interactions, which allowed users to manage multiple package and channel interactions from a single dashboard
- Performed additional unit testing
- **Due to a lack of backwards compatibility, this release locks the following two versions of the dependency packages:**
    - flask-wtf=0.8.4
    - werkzeug=0.9.6

Command reference

Anaconda Client is the command line interface (CLI) to Anaconda Repository. You can use it to log in, log out, manage your account, upload files, generate access tokens, view tokens and other tasks.

The full Client command reference is shown below. You can also view this command reference in a terminal window with the command `anaconda --help` or `anaconda -h`.

See also: Anaconda Cloud API Reference.

anaconda

```
usage: anaconda [-h] [--disable-ssl-warnings] [--show-traceback] [-v] [-q]
                [-V] [-t TOKEN] [-s SITE]
                ...

Anaconda Repository command line manager

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit

output:
  --disable-ssl-warnings
                        Disable SSL warnings (default: False)
  --show-traceback      Show the full traceback for chalmers user errors
                        (default: False)
  -v, --verbose         print debug information ot the console
  -q, --quiet           Only show warnings or errors the console

anaconda-client options:
  -t TOKEN, --token TOKEN
                        Authentication token to use. May be a token or a path
                        to a file containing a token
  -s SITE, --site SITE  select the anaconda-client site to use

Commands:

    auth                Manage Authorization Tokens
    label               Manage your Anaconda Repository labels
    channel             [DEPRECATED in favor of label] Manage your Anaconda
                        Repository channels
    config              Anaconda client configuration
    copy                Copy packages from one account to another
    download            Download notebooks from Anaconda Repository
    groups              Manage Groups
    login               Authenticate a user
    logout              Log out from Anaconda Repository
    notebook            [DEPRECATED in favor of upload/download] Interact
                        with notebooks in anaconda.org
```

(continues on next page)

(continued from previous page)

```
    package             Package utils
    remove              Remove an object from Anaconda Repository. Must refer to
                        the formal package name as it appears in the URL of
                        the package. Also use anaconda show <USERNAME> to see
                        list of package names. Example: anaconda remove
                        continuumio/empty-example-notebook
    search              Search Anaconda Repository
    show                Show information about an object
    upload              Upload packages to Anaconda Repository
    whoami              Print the information of the current user
    build               Anaconda build client for continuous integration,
                        testing and building packages
    worker              Anaconda build client for continuous integration,
                        testing and building packages
```

Authentication

auth

```
usage: anaconda auth [-h] [-n NAME] [-o ORGANIZATION]
                     [--strength {strong,weak}] [--strong] [-w] [--url URL]
                     [--max-age MAX_AGE] [-s SCOPES] [--out OUT]
                     (-x | -l | -r NAME [NAME ...] | -c | -i)

Manage Authorization Tokens

optional arguments:
  -h, --help            show this help message and exit
  -n NAME, --name NAME  A unique name so you can identify this token later.
                        View your tokens at anaconda.org/settings/access
  -o ORGANIZATION, --org ORGANIZATION, --organization ORGANIZATION
                        Set the token owner (must be an organization)

token creation arguments:
  These arguments are only valid with the `--create` action

  --strength {strong,weak}
  --strong              Create a longer token (default)
  -w, --weak            Create a shorter token
  --url URL             The url of the application that will use this token
  --max-age MAX_AGE     The maximum age in seconds that this token will be
                        valid for
  -s SCOPES, --scopes SCOPES
                        Scopes for token. For example if you want to limit
                        this token to conda downloads only you would use
                        --scopes "repo conda:download"
  --out OUT

actions:
  -x, --list-scopes     list all authentication scopes
  -l, --list            list all user authentication tokens
  -r NAME [NAME ...], --remove NAME [NAME ...]
                        remove authentication tokens
  -c, --create          Create an authentication token
```

(continues on next page)

(continued from previous page)

```
  -i, --info, --current-info
                        Show information about the current authentication
                        token

Manage Authentication tokens
```

See also *token*.

login

```
usage: anaconda login [-h] [--hostname HOSTNAME] [--username LOGIN_USERNAME]
                      [--password LOGIN_PASSWORD]

Authenticate a user

optional arguments:
  -h, --help            show this help message and exit
  --hostname HOSTNAME   Specify the host name of this login, this should be
                        unique (default: hq-phone-114.corp.continuum.io)
  --username LOGIN_USERNAME
                        Specify your username. If this is not given, you will
                        be prompted
  --password LOGIN_PASSWORD
                        Specify your password. If this is not given, you will
                        be prompted
```

logout

```
usage: anaconda logout [-h]

Log out from Anaconda Repository

optional arguments:
  -h, --help  show this help message and exit
```

whoami

```
usage: anaconda whoami [-h]

Print the information of the current user

optional arguments:
  -h, --help  show this help message and exit
```

Informational

show

```
usage: anaconda show [-h] spec

Show information about an object

positional arguments:
  spec        Package written as USER[/PACKAGE[/VERSION[/FILE]]]

optional arguments:
  -h, --help  show this help message and exit

Show information about an object
```

EXAMPLE:

```
anaconda show anaconda
anaconda show anaconda/python
anaconda show anaconda/python/2.7.5
anaconda show anaconda/python/2.7.5/linux-64/python-2.7.5-0.tar.bz2
```

search

```
usage: anaconda search [-h] [-t {conda,pypi}]
                       [-p {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-
↪armv6l,linux-armv7l,linux-ppc64le,noarch}]
                       name

Search Anaconda Repository

positional arguments:
  name                  Search string

optional arguments:
  -h, --help            show this help message and exit
  -t {conda,pypi}, --package-type {conda,pypi}
                        only search for packages of this type
  -p {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-armv6l,linux-armv7l,linux-
↪ppc64le,noarch}, --platform {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-
↪armv6l,linux-armv7l,linux-ppc64le,noarch}
                        only search for packages of the chosen platform

Search Anaconda Repository for packages
```

config

```
usage: anaconda config [-h] [--type TYPE] [--set name value] [--get name]
                       [--remove REMOVE] [--show] [-f] [--show-sources] [-u]
                       [-s]

```

(continues on next page)

(continued from previous page)

```
Anaconda client configuration

optional arguments:
  -h, --help            show this help message and exit
  --type TYPE           The type of the values in the set commands

actions:
  --set name value      sets a new variable: name value
  --get name            get value: name
  --remove REMOVE       removes a variable
  --show                show all variables
  -f, --files           show the config file names
  --show-sources        Display all identified config sources

location:
  -u, --user            set a variable for this user
  -s, --system, --site  set a variable for all users on this machine

anaconda-client configuration

Get, Set, Remove or Show the anaconda-client configuration.

###### anaconda-client sites

anaconda-client sites are a mechanism to allow users to quickly switch
between Anaconda Repository instances. This can be used with the on-site Anaconda
Enterprise.

  * Invoke the anaconda command with the `-s/--site` option like this:

        anaconda -s site_name whoami

  * Set a site as the default:

        anaconda config --set default_site site_name
        anaconda whoami

###### Add an anaconda-client site

After installing Anaconda Enterprise
you can add a site named **site_name** like this:

    anaconda config --set sites.site_name.url "http://<anaconda-enterprise-ip>:<port>/
↪api"
    anaconda config --set default_site site_name

###### Site Options VS Global Options

All options can be set as global options that affect all sites
or site options that affect only one site.

By default, options are set globally:

    anaconda config --set OPTION VALUE

If you want the option to be limited to a single site,
prefix the option with `sites.site_name`:
```

(continues on next page)

(continued from previous page)

```
    anaconda config --set sites.site_name.OPTION VALUE

###### Common anaconda-client configuration options

  * `url`: Set the anaconda api url (default: https://api.anaconda.org)
  * `ssl_verify`: Perform ssl validation on the https requests.
    ssl_verify may be `True`, `False` or a path to a root CA pem file.

###### Toggle auto_register when doing anaconda upload

The default is yes, automatically create a new package when uploading.
If no, then an upload will fail if the package name does not already exist on the␣
↪server.

    anaconda config --set auto_register yes|no
```

Package management

package

```
usage: anaconda package [-h]
                        (--add-collaborator user | --list-collaborators | --create)
                        [--summary SUMMARY] [--license LICENSE]
                        [--license-url LICENSE_URL] [--personal | --private]
                        USER/PACKAGE

Anaconda Repository package utilities

positional arguments:
  USER/PACKAGE          Package to operate on

optional arguments:
  -h, --help            show this help message and exit

actions:
  --add-collaborator user
                        username of the collaborator you want to add
  --list-collaborators  list all of the collaborators in a package
  --create              Create a package

metadata arguments:
  --summary SUMMARY     Set the package short summary
  --license LICENSE     Set the package license
  --license-url LICENSE_URL
                        Set the package license url

privacy:
  --personal            Set the package access to personal This package will
                        be available only on your personal registries
  --private             Set the package access to private This package will
                        require authorized and authenticated access to install
```

upload

```
usage: anaconda upload [-h] [-c CHANNELS] [-l LABELS] [--no-progress]
                       [-u USER] [--all] [-p PACKAGE] [-v VERSION]
                       [-s SUMMARY] [-t PACKAGE_TYPE] [-d DESCRIPTION]
                       [--thumbnail THUMBNAIL] [--private]
                       [--no-register | --register] [--build-id BUILD_ID]
                       [-i | -f | --force]
                       files [files ...]

Upload packages to Anaconda Repository

positional arguments:
  files                 Distributions to upload

optional arguments:
  -h, --help            show this help message and exit
  -c CHANNELS, --channel CHANNELS
                        [DEPRECATED] Add this file to a specific channel.
                        Warning: if the file channels do not include "main",
                        the file will not show up in your user channel
  -l LABELS, --label LABELS
                        Add this file to a specific label. Warning: if the
                        file labels do not include "main", the file will not
                        show up in your user label
  --no-progress         Don't show upload progress
  -u USER, --user USER  User account or Organization, defaults to the current
                        user
  --all                 Use conda convert to generate packages for all
                        platforms and upload them
  --no-register         Don't create a new package namespace if it does not
                        exist
  --register            Create a new package namespace if it does not exist
  --build-id BUILD_ID   Anaconda Repository Build ID (internal only)
  -i, --interactive     Run an interactive prompt if any packages are missing
  -f, --fail            Fail if a package or release does not exist (default)
  --force               Force a package upload regardless of errors

metadata options:
  -p PACKAGE, --package PACKAGE
                        Defaults to the package name in the uploaded file
  -v VERSION, --version VERSION
                        Defaults to the package version in the uploaded file
  -s SUMMARY, --summary SUMMARY
                        Set the summary of the package
  -t PACKAGE_TYPE, --package-type PACKAGE_TYPE
                        Set the package type [ipynb, env]. Defaults to
                        autodetect
  -d DESCRIPTION, --description DESCRIPTION
                        description of the file(s)
  --thumbnail THUMBNAIL
                        Notebook's thumbnail image
  --private             Create the package with private access

    anaconda upload CONDA_PACKAGE_1.bz2
    anaconda upload notebook.ipynb
    anaconda upload environment.yml
```

See also:

- *Uploading a conda package*.
- *Uploading PyPI packages*.

download

```
usage: anaconda download [-h] [-f] [-o OUTPUT] handle

Download packages from Anaconda Repository

positional arguments:
  handle                user/notebook

optional arguments:
  -h, --help            show this help message and exit
  -f, --force           Overwrite
  -o OUTPUT, --output OUTPUT
                        Download as

Usage:
    anaconda download notebook
    anaconda download user/notebook
```

remove

```
usage: anaconda remove [-h] [-f] specs [specs ...]

Remove an object from Anaconda Repository

example::

    anaconda remove sean/meta/1.2.0/meta.tar.gz

positional arguments:
  specs        Package written as <user>[/<package>[/<version>[/<filename>]]]

optional arguments:
  -h, --help   show this help message and exit
  -f, --force  Do not prompt removal
```

groups

```
usage: anaconda groups [-h] [--perms {read,write,admin}]
                       {add,show,members,add_member,remove_member,packages,add_
↪package,remove_package}
                       spec

positional arguments:
  {add,show,members,add_member,remove_member,packages,add_package,remove_package}
                        The group management command to execute
```

(continues on next page)

(continued from previous page)

```
  spec                  <organization>/<group_name>/<member>

optional arguments:
  -h, --help            show this help message and exit
  --perms {read,write,admin}
                        The permission the group should provide
```

label

```
usage: anaconda label [-h] [-o ORGANIZATION]
                      (--copy LABEL LABEL | --list | --show LABEL | --lock LABEL | --
↪unlock LABEL | --remove LABEL)

Manage your Anaconda Repository channels

optional arguments:
  -h, --help            show this help message and exit
  -o ORGANIZATION, --organization ORGANIZATION
                        Manage an organizations labels
  --copy LABEL LABEL
  --list                list all labels for a user
  --show LABEL          Show all of the files in a label
  --lock LABEL          Lock a label
  --unlock LABEL        Unlock a label
  --remove LABEL        Remove a label
```

copy

```
usage: anaconda copy [-h] [--to-owner TO_OWNER] [--from-label FROM_LABEL]
                     [--to-label TO_LABEL]
                     spec

Copy packages from one account to another

positional arguments:
  spec                  Package - written as user/package/version[/filename]
                        If filename is not given, copy all files in the
                        version

optional arguments:
  -h, --help            show this help message and exit
  --to-owner TO_OWNER   User account to copy package to (default: your
                        account)
  --from-label FROM_LABEL
                        Label to copy packages from
  --to-label TO_LABEL   Label to put all packages into
```

move

```
usage: anaconda move [-h] [--from-label FROM_LABEL] [--to-label TO_LABEL] spec

Move packages between labels.

positional arguments:
  spec                  Package - written as user/package/version[/filename]
                        If filename is not given, move all files in the
                        version

optional arguments:
  -h, --help            show this help message and exit
  --from-label FROM_LABEL
                        Label to move packages from
  --to-label TO_LABEL   Label to move packages to
```

Glossary

Anaconda

An easy-to-install, free collection of open source packages, including Python and the conda package manager. Over 150 packages are installed with Anaconda. After installing Anaconda, you can install or update over 250 additional open source packages contained in the Anaconda Repository using the `conda install PACKAGE command`.

NOTE: Replace `PACKAGE` with the name of the desired package.

Anaconda Client CLI

The Anaconda Client command line interface (CLI) allows you to log into Anaconda Repository directly from your Terminal window or Anaconda Prompt and manage your account. It is not necessary for downloading or installing packages from Repository.

Anaconda Repository

Repository hosts hundreds of useful Python packages, notebooks and environments for a wide variety of applications. You do not need to be logged in, or even need a Repository account, to search for packages, download and install them.

conda

The conda package manager and environment manager program that installs and updates packages and their dependencies, and lets you easily switch between environments on your local computer.

conda build

The command line interface that lets you build packages for your local operating system.

conda package

A compressed file containing system-level libraries, Python modules, executable programs or other components. The file uses the tarball format.

label

Part of the URLs for Repository where conda looks for packages. Labels are searched only if you specify a label.

The default label is “main,” so packages that are uploaded without specifying a label are automatically labeled “main.” The version labeled main is also downloaded by default, unless a user specifies a different label. So, if a file is labeled main, then the label name may be omitted from the URL.

EXAMPLE: The following repositories are equivalent:

```
https://<your-anaconda-repo>/sean/label/main
https://<your-anaconda-repo>/sean
```

Commands such as `conda install` can be used with a channel, or used with a channel and a label:

```
conda install --channel sean selenium
conda install --channel sean/label/dev selenium
conda install --channel sean/label/stable selenium
```

Using Anaconda Client, *package* developers can create labels such as development `labels/dev`, test `labels/test` or other labels that are searched only if the user specifies the label.

EXAMPLE: The following search examples use a *namespace* of “travis”:

- `https://<your-anaconda-repo>/travis/labels/main`—the label searched by default.
- `https://<your-anaconda-repo>/travis`—same as default label with `main` implicit.
- `https://<your-anaconda-repo>/travis/labels/dev`—contains the packages in development.
- `https://<your-anaconda-repo>/travis/labels/test`—contains packages ready to test.
- `https://<your-anaconda-repo>/travis/labels/any-custom-label`—any label you want to use.

Miniconda

A minimal installer for *conda*. Like *Anaconda*, Miniconda is a software package that includes the conda package manager and Python and its dependencies, but does not include any other packages. Once conda is installed by installing either Anaconda or Miniconda, you can install other software packages directly from the command line using `conda install`.

namespace

Each user and organization has their own location called a “namespace” where they may host packages. You can view the public packages in a user or organization’s namespace by navigating to their user page.

EXAMPLE: The “travis” user namespace located at `https://<your-anaconda-repo>/travis` contains packages that were uploaded and shared by the user whose account is named “travis.”

noarch package

A conda package that contains nothing specific to any system architecture, so it may be installed on any system. When conda searches for packages on any system in a channel, conda always checks both the system-specific subdirectory—such as `linux-64`—and the `noarch` directory.

on-site repository

Repository is powered by Anaconda Server. You can run your own server behind firewalls or in air gapped environments. For more information, contact sales@anaconda.com.

organization account

An organization account is a type of account on Repository that allows multiple individual users to administer packages and control package access to different user groups. It also includes a large amount of storage space.

Use organization accounts to:

- Share packages, environments or notebooks under an organization’s account rather than your personal account.
- Assign multiple account administrators.
- Assign different access permissions to groups of users and customize per-package access by group.

package

All files uploaded to Repository are stored in packages. Each Repository package is visible at its own unique URL based on the name of the user who owns the package and the name of the package.

EXAMPLE: If a user "travis" uploads a test package named "testpkg," it is visible at:

```
https://<your-anaconda-repo>/travis/testpkg
```

Repository packages may contain multiple files, and these files may be data files such as comma separated value (CSV), tab separated value (TSV), or text (TXT), or package files such as conda packages, PyPI packages or R packages.

package manager

A tool that facilitates the process of installing, upgrading, configuring and removing packages on Repository. Repository supports two package managers, *conda* and *PyPI*.

For more information, see *Using package managers*.

project

Anaconda Project is an open source tool created by Anaconda that delivers light-weight, efficient encapsulation and portability of data science projects.

repository

A storage location from which software packages may be retrieved and installed on a computer.

source package

"Source" packages are source code only, not yet built for any specific platform, and might be compatible with all, some or only one of the platforms.

token

An access control token is a random alphanumeric string that is inserted into a URL that you give to another Repository user. The token allows them to download a package or add a channel that you have marked private. Only those users with the correct access token can access the private file. You can use Client to generate tokens to give other users specifically scoped access to packages and collections.

Previous versions

This documentation is provided for the use of our customers who have not yet upgraded to the current version. Your version number is located in the footer.*

Anaconda Repository 2.32

User guide (AER 2.32)

Quickstart (AER 2.32)

Managing Your Account (AER 2.32)

Using Anaconda Repository (AER 2.32)

Parcels, Management Packs, and Installers (AER 2.32)

Reference (AER 2.32)

Command Reference (AER 2.32)

Admin and install guide (AER 2.32)

Install Anaconda Repository (AER 2.32)

Online Installation (AER 2.32)

Offline Installation (AER 2.32)

Advanced Installation Options (AER 2.32)

Troubleshooting your Anaconda repository installation (AER 2.32)

Maintenance and configuration concerns (AER 2.32)

User administration (AER 2.32)

Anaconda repository requirements and verification (AER 2.32)

Client configuration (AER 2.32)

Air gap archive (AER 2.32)

Update/Uninstall Anaconda Repository (AER 2.32)

Anaconda repository backup and restore procedure (AER 2.32)

Anaconda repository command line interface (AER 2.32)

Command line interface (AER 2.32)

Administrative commands (AER 2.32)

Adding a PyPI or Anaconda mirror to your Anaconda repository installation (AER 2.32)

Mirroring an Anaconda repository (AER 2.32)

1.1.2 Anaconda Enterprise 4 Notebooks

Empower the Data Science Team with cross-collaboration

AEN is a browser-based Python data analysis environment and visualization tool from Anaconda®. AEN is a ready-to-use, powerful, fully-configured data analytics environment all in a secure, governed environment.

AEN allows data science team members to create and share private notebooks, manage access, control notebook revisions, compare and identify differences across notebook versions, search notebooks for keywords and packages, use enhanced collaborative notebook features—including revision control and locking—and to access an on-premises and/or cloud collaborative notebook server.

The current version of AEN is 4.2.2, released March 1st, 2018.

User guide

AEN's browser-based management of private packages, notebooks, and environments allows data science team members to:

- Create, share and manage private notebooks.
- Control notebook revisions.
- Compare and identify differences across notebook versions.
- Search notebooks for keywords and packages.
- Use enhanced collaborative notebook features including revision control and locking.
- Access on-premises and/or cloud-based collaborative notebook servers.
- Utilize multiple language kernels like Python and R language in the same notebook.
- Create new notebook environments on the fly without leaving the notebook or entering commands in a prompt.
- Publish results to business stakeholders as interactive visualizations and presentations.

To quickly get up and running with AEN, see *Getting started*.

Download the *Cheat sheet* for easy reference.

Concepts

Projects

AEN users interact with the system predominantly through projects.

A project is a set of conda environments, Jupyter Notebooks, and other files.

Each project has a project drive that all team members can access. The size of the drive is not limited by AEN. Contact your system administrator if you find you do not have sufficent space.

Each project has a separate project directory on the project drive.

The project directory is a directory for project files and data that is separate from the project owner's and team members' home directories, so that team members can share and have equal access.

The path to your project directory is `/projects/<project_owner>/<project_name>`.

For administrative information about projects, directories, and permissions, see *Projects and permissions*.

Team collaboration

Teams collaborate in AEN using projects. Projects allow a team to easily come together by sharing the resources, applications, and environments that are necessary to collaborate effectively.

The AEN project owner and any team members connected to their project will have access to the same:

- Shared files and home directories.
- Shared Python and R environments.
- Shared nodes and hardware.
- Common applications.
- Web user interface.

For more information, see *Working with projects*.

Access control

AEN access controls allow you to:

- Add and remove project access for new team members.
- Limit the access to specific folders and files to members of your project team.
- Use permissions to extend execute access to team members. By default, all of the team members on a project have read and write access to all project assets.

Access control is performed from each project's Workbench application.

For more information, see *Controlling access to your project*.

Sharing projects

AEN supports both public and private sharing.

A project can be "public," which means that anyone with access to the system can view the project assets.

Any content placed in the `public` folder in a project is publicly accessible using its URL.

A project can be "private," which means that only the project owner and team members can view the project assets.

You can also *limit who can access specific files*.

Sharing Jupyter Notebooks

In addition to general project sharing capabilities, you can also publish Jupyter Notebooks to Anaconda Repository. This automatically versions the notebook and allows you to define who can view the notebook.

Project tags

Tags are used to:

- Group similar or related projects.
- Identify your project so that it is easier to find.
- Let others know about your project.

You can *add and remove tags* for any project that you have access to.

Getting started

This section contains information and tasks for first-time AEN users.

In this getting started guide, you will:

1. Download the AEN cheat sheet

Before you start, download and print the *AEN cheat sheet* for easy reference.

2. Access your user home page

After your administrator has set up your server and new Anaconda account, you will receive a welcome email.

1. Click the link in the email to open the AEN login page.

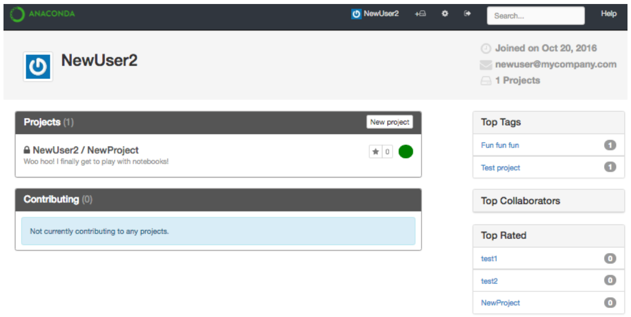
   NOTE: Use the domain name and not the IP address when you connect to AEN. Using the IP address can cause TLS and security certificate errors.

2. Enter your AEN account username and password.

   NOTE: Some administrators allow you to create your own account. If your administrator has allowed this, in the create a new account section, create your own username and password.

3. Click the Login button.

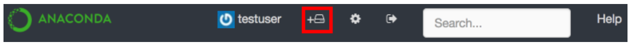
   Your user home page, where all good things happen, is displayed:



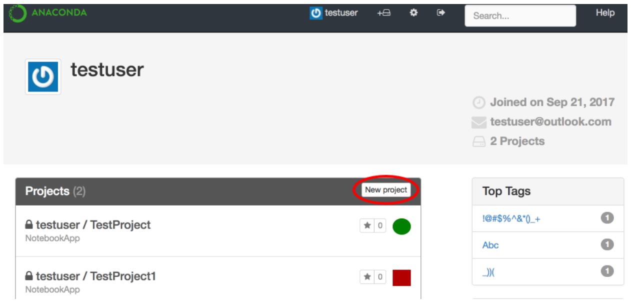
3. Create a new project

1. There are 2 ways to create a new project in AEN:

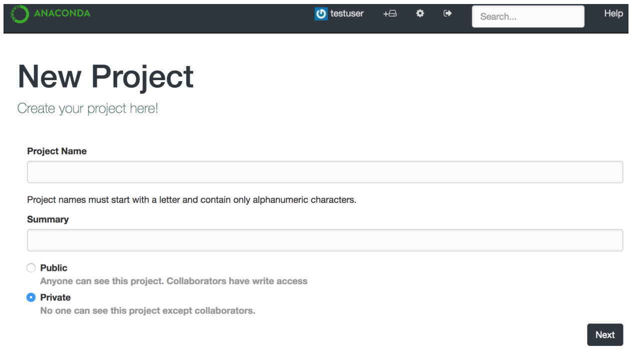
   - On the right side of the AEN task bar, click on the New Project icon:



   - On your home page, click the New project button:



2. On the Project page that is displayed, type a name for your project, such as "Testing."



New Project

Create your project here!

Project Name

Project names must start with a letter and contain only alphanumeric characters.

Summary

○ Public
Anyone can see this project. Collaborators have write access

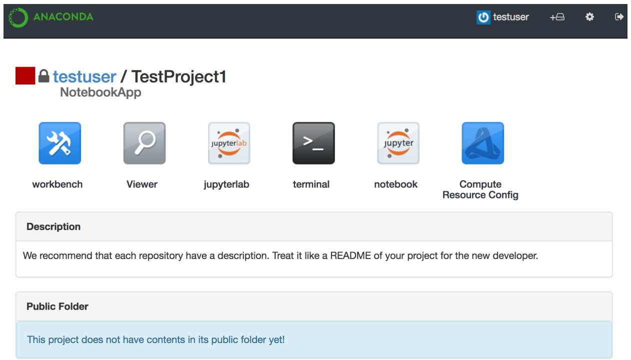
● Private
No one can see this project except collaborators.

3. Type a summary of the project so you can recognize it later.
4. Select whether your project will be public or private.
5. Verify that the default data center is selected.

   TIP: You can update the project summary and description at any time from the **Project** menu in the Project Settings. To return to your project at any time, click the project name.

6. Click the Next button.

   Your new project's home page is displayed:



testuser / TestProject1
NotebookApp

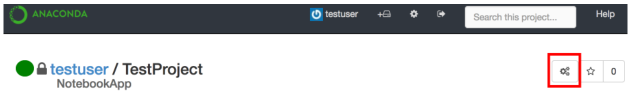
Description

We recommend that each repository have a description. Treat it like a README of your project for the new developer.

Public Folder

This project does not have contents in its public folder yet!

7. To change the project settings, click the Project Settings icon on at the top right.



testuser / TestProject
NotebookApp

8. Modify the summary or add a description of the project.

TIP: A project description is recommended, and may be written in Markdown syntax (plain text valid Markdown).

To see how Markdown will be displayed, in the description area, click the **Preview** tab.

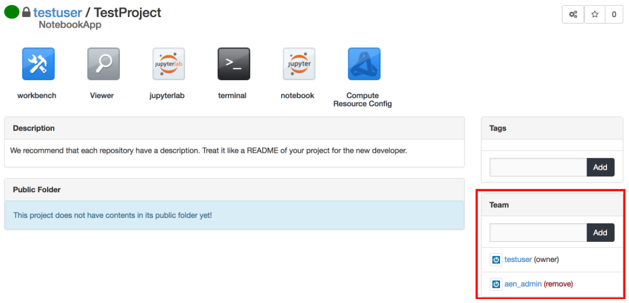
4. Add collaborators

You can add team members to your project as collaborators. Adding team members to your projects makes collaboration easy because they have full access to the project's applications, files and services.

When you add team members, their home directory is mounted in the project. There is no need to download and email data or scripts—team members can work on the same files in the same environment in which you are working.

To add collaborators to your project:

1. From your project home page, in the Team box, begin typing a teammate's username.
2. In the list that is displayed, select the teammate's username.
3. Click the Add button.



We recommend that each repository have a description. Treat it like a README of your project for the new developer.

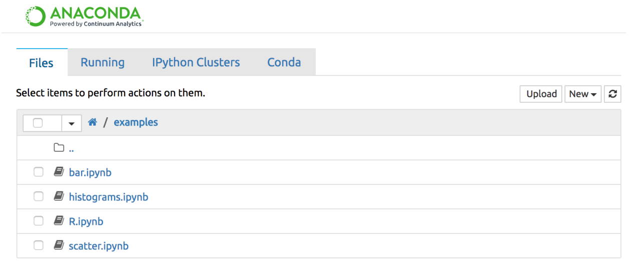
This project does not have contents in its public folder yet!

1. Repeat these steps for each team member you want to add as a collaborator.

TIP: You can add or remove team members any time from the **Team** menu in Project Settings. You can also modify a team member's read, write or execute permissions at any time from the *Using Workbench*.

5a. Open an example notebook, OR

1. From your project home page, click the Jupyter Notebooks icon.
2. On the File View page, click the Examples folder.



Select items to perform actions on them.

1. Select any of the example notebooks.

2. To see the default results of the formulas used in the displayed notebook, in the **Cell** menu, select Run All.
3. To experiment with changing the notebook, edit any of the formulas in the notebook.
4. In the **Cell** menu, select Run All.

Any differences resulting from your edits are displayed.

5b. Create a new environment and notebook

If you are already familiar with creating notebooks, you can easily set up a new environment with the programs you need—like SciPy and NumPy—then open a new notebook and make your edits.

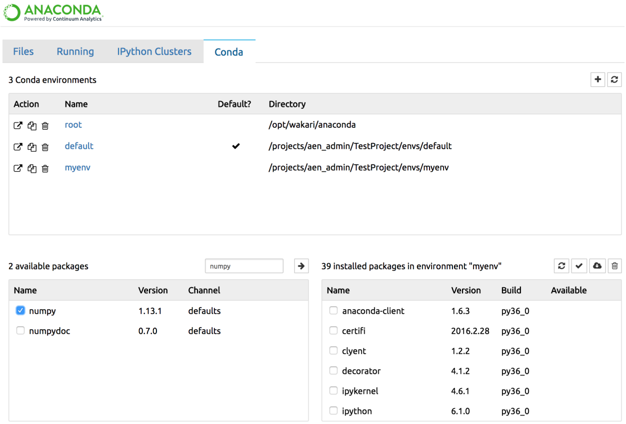
To create a new environment:

1. From your project home page, click the Jupyter Notebooks icon.
2. On the File View page, click the **Conda** tab.
3. To add a new conda environment, on the top right of the **Conda** tab, click the + icon.
4. Type a name for your environment.
5. Select Python 2, Python 3 or R language kernel.
6. Click the Create button.
7. To activate your new environment, click its name.

The packages that are available and installed in your new environment are displayed.

Adding SciPy and Numpy packages

1. In the available packages section, search for the package name `numpy`—all lower case.
2. In the results section, next to numpy, select the checkbox.



3 Conda environments

Action	Name	Default?	Directory
	root		/opt/wakari/anaconda
	default	✓	/projects/aen_admin/TestProject/envs/default
	myenv		/projects/aen_admin/TestProject/envs/myenv

2 available packages

numpy

Name	Version	Channel
☑ numpy	1.13.1	defaults
□ numpydoc	0.7.0	defaults

39 installed packages in environment "myenv"

Name	Version	Build	Available
□ anaconda-client	1.6.3	py36_0	
□ certifi	2016.2.28	py36_0	
□ clyent	1.2.2	py36_0	
□ decorator	4.1.2	py36_0	
□ ipykernel	4.6.1	py36_0	
□ ipython	6.1.0	py36_0	

1. Click the Install icon.

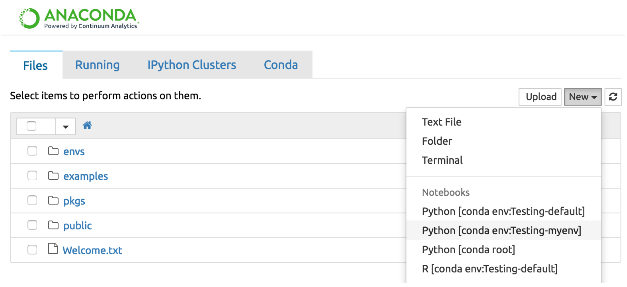
2. To confirm your installation, click the Install button.

Numpy is displayed in the installed packages section—if not, click the Refresh button. Repeat these steps to install the Scipy package—searching for `scipy` in step 1.

TIP: You can return to this screen at any time to add additional packages to this environment.

Creating a new notebook in your environment

1. From the AEN homepage, click the **Files** tab.
2. On the top right of the **Files** tab, click the New button.
3. Under Notebooks, select the Python environment with the name you entered while *creating a new environment*.


ANACONDA
Powered by Continuum Analytics
Files
Running
IPython Clusters
Conda
Select items to perform actions on them.
Upload
New
envs
examples
pkgs
public
Welcome.txt
Text File
Folder
Terminal
Notebooks
Python [conda env:Testing-default]
Python [conda env:Testing-myenv]
Python [conda root]
R [conda env:Testing-default]

> NOTE: If you do not see your new environment listed under Notebooks, next to the New button, click the Refresh button.

A new locked notebook is displayed. Paste or write some code to execute when you are ready.

6. Create checkpoints for version control

Whether you are exploring an existing notebook, or creating a new one, you can easily create checkpoints, return to an earlier version, compare two different versions and save them for reference.

To create a checkpoint, in the **File** menu, select Save and Checkpoint:

To revert your notebook to a previous checkpoint, in the **File** menu, select Revert to Checkpoint.

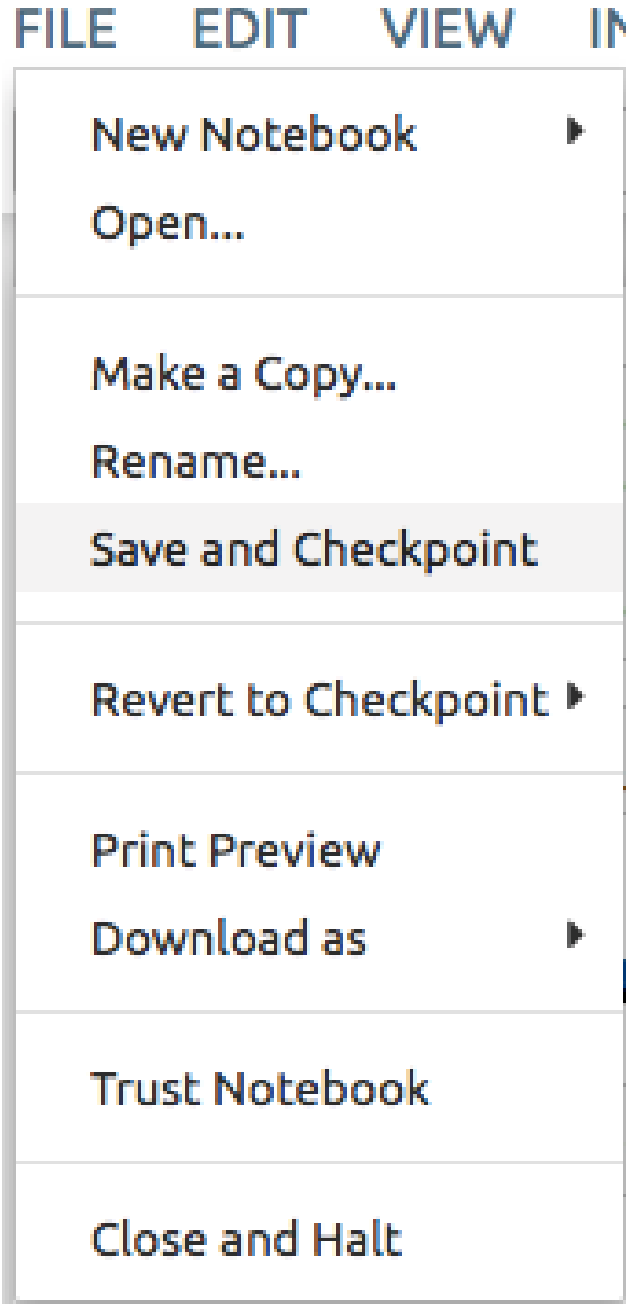
NOTE: For more information about revision control features, including creating commits and comparing differences, see *Using the Revision Control Mechanism extension*.

7. Share your notebook and environment with others

See *Sharing projects and notebooks*.

8. See what to do next

Now that you have completed the Getting Started guide, you are ready to move on to *basic tasks* and *advanced tasks*.


FILE
EDIT
VIEW
IN
New Notebook
Open...
Make a Copy...
Rename...
Save and Checkpoint
Revert to Checkpoint
Print Preview
Download as
Trust Notebook
Close and Halt

Basic tasks

This section contains information and tasks that use the web browser to manage projects and is best-suited for any beginning AEN user:

Working with projects

Almost everything in AEN starts by opening an existing project or creating a new one.

After that, you can set up a special environment with the packages you want, set their access permissions and modify your project settings.

Searching for a project or file

To search for projects and files, use the Search box in the AEN navigation bar. The search provides different results depending on which page you search from:

- On a project home page, search results include any files that match your search criteria within the current project.
- On any other AEN page, search results include any files that match your search criteria within all projects.

TIP: Your search results include only files and projects that you can view: public projects, and private projects to which you have a minimum of view access.

Types of files searched

The following types of files are included in search results:

- `.py`—Python source files.
- `.ipynb`—IPython/Jupyter notebooks.
- `.txt`—plain text files.
- `.md`—Markdown files.

Search indexing

Files that are modified while a project is running are automatically re-indexed shortly after the files are modified. If you create or update a large number of files—such as cloning a git repository or copying a directory—search results may take several minutes to update.

Files that are modified while the project is not running are re-indexed only after the project is started.

Using search constructs

You can use the following search constructs:

- Ordinary words will match the full-text contents of any file.
- Wildcards are permitted.

  EXAMPLE: `John*` will match John and Johnny. These are glob patterns and are similar to their usage in the command line.
- Combine queries using `AND` or `OR`, and group them using parentheses `()`.

Regular expression patterns can be embedded in the query string by wrapping them in forward-slashes (`/`):

```
name:/joh?n(ath[oa]n)/
```

The supported regular expression syntax is explained in the Elasticsearch reference.

NOTE: Wildcards apply inside a regular expression. A query string such as `/.*n/` would force the search to visit every term in the index.

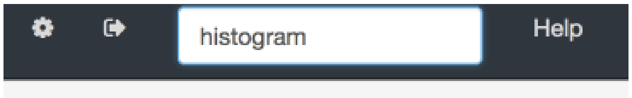
Searching metadata fields

You can search in specific metadata fields:

- `imports:name`—matches files that import the module name.
- `uses:name`—matches files that reference the identifier name. Referenced names include any functions and globals imported from other modules, as well as the names of any methods invoked on any object.
- `defines:name`—matches files that define the identifier name. Defined names include functions defined at global scope, class names, and method names within classes.
- `acl:user`—matches files in which the named user has read access or higher.

Searching a project

1. In the Search box, type a string of text:

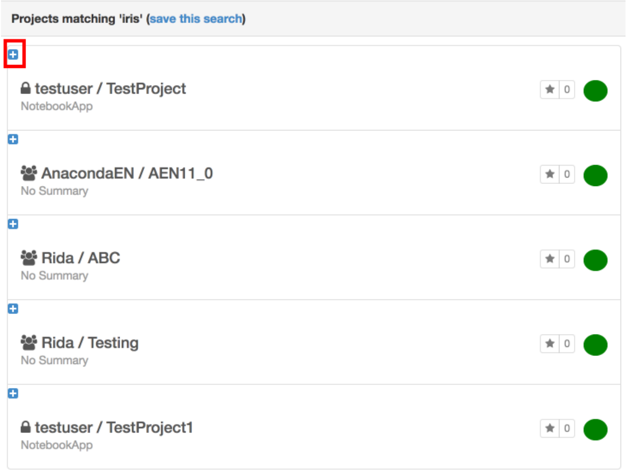
   histogram

   TIP: Search by glob patterns, which are similar to file matching in the command line.

   EXAMPLE: To find projects in the test family that are numbered from 00 to 99, search for `Test-??`. To find all projects whose name ends with "Stats," search for `*Stats`.

2. Press Enter.

3. In the search results, click the plus + icon above a project name to show a list of matching files in the selected project:



Projects matching 'iris' (save this search)

testuser / TestProject
NotebookApp

AnacondaEN / AEN11_0
No Summary

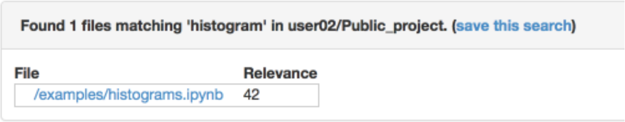
Rida / ABC
No Summary

Rida / Testing
No Summary

testuser / TestProject1
NotebookApp

TIP: Click the project name to open the project's home page.

4. To view a file, click its file name in the matching files list:



Found 1 files matching 'histogram' in user02/Public_project. (save this search)

File	Relevance
/examples/histograms.ipynb	42

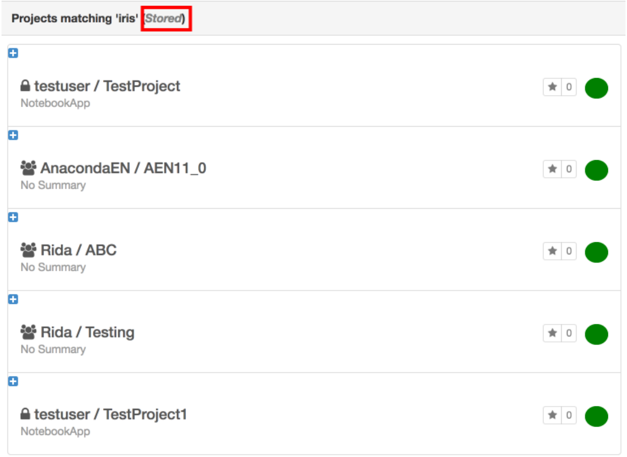
Saving a search

1. At the top of the search results, click Save this search:

The "save this search" text changes to "stored" and your search is saved. Your saved searches are listed on your home page.

Removing a saved search

On your home page, in the Saved searches section, click X next the saved search that you want to remove:



Projects matching 'iris' (Stored)

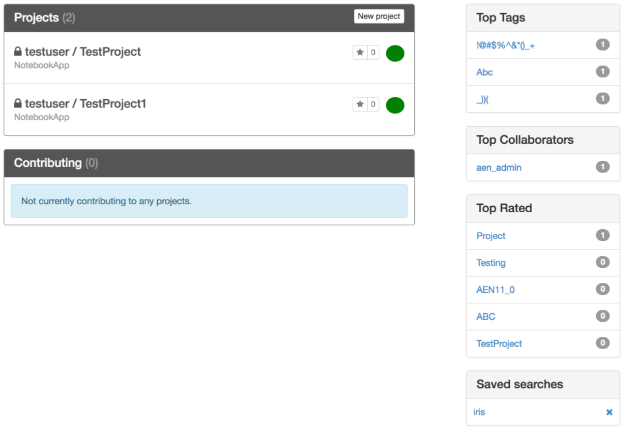
testuser / TestProject ★ 0
NotebookApp

AnacondaEN / AEN11_0 ★ 0
No Summary

Rida / ABC ★ 0
No Summary

Rida / Testing ★ 0
No Summary

testuser / TestProject1 ★ 0
NotebookApp



Projects (2)

New project

testuser / TestProject ★ 0
NotebookApp

testuser / TestProject1 ★ 0
NotebookApp

Contributing (0)

Not currently contributing to any projects.

Top Tags	
!@#$%^&*()_+	1
Abc	1
_))(	1

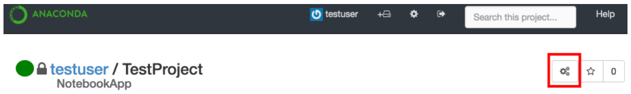
Top Collaborators	
aen_admin	1

Top Rated	
Project	1
Testing	0
AEN11_0	0
ABC	0
TestProject	0

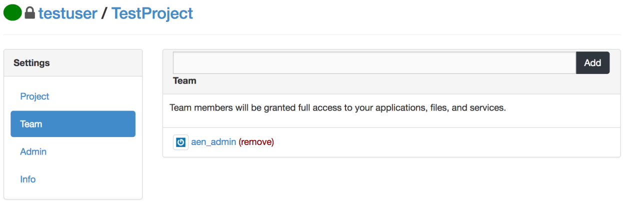
Saved searches	
iris	×

Adding and removing team members on a project

1. On the project home page, click the Project Settings icon to open the Project Settings page.



2. In the **Settings** menu, select Team.



Team members will be granted full access to your applications, files, and services.

Adding a team member

1. In the username box, type in the first few letters of the username for the team member you want to add to the project.
2. In the list of usernames that displays, click the user to add.
3. Click the Add button.

Removing a team member

Click the red Remove link next to the name of the user you want to remove from the project.

Controlling access to your project

- *Controlling team member access*
- *Controlling non-team member access*

Controlling team member access

By default, all of the team members on a project have read and write access permissions for all project assets.

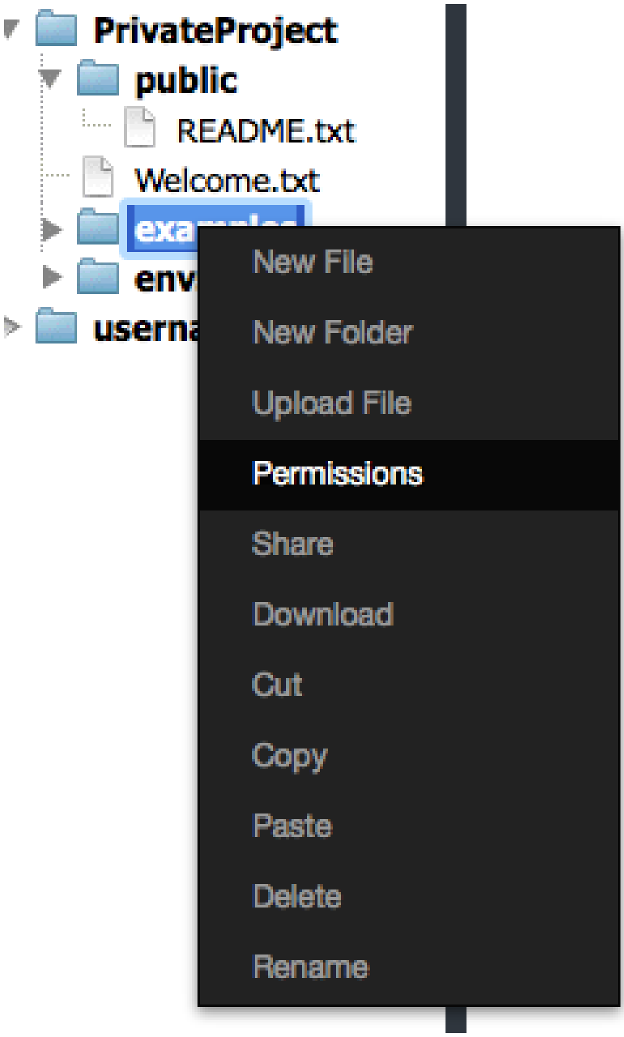
The available permissions are read, write and execute. If you remove all individual or group permissions for a project asset, team members will not be able to access that asset.

To change a project's permissions:

1. Open the project's home page.
2. Click the Workbench icon.
3. In the Workbench app, right-click the file or folder you want to limit access to.

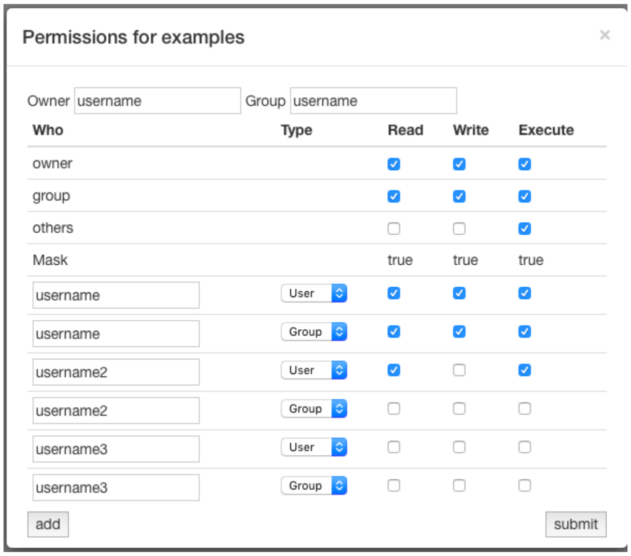
   NOTE: When you change a folder's permissions, the permissions of files and folders inside it do not change. You may change the permissions of those files and folders manually.

4. In the menu that displays, select Permissions:



A list of owners and team members who have access to your project is displayed.

5. Find the team member you want to change access for:



Permissions for examples

Owner username Group username

Who	Type	Read	Write	Execute
owner		☑	☑	☑
group		☑	☑	☑
others		☐	☐	☑
Mask		true	true	true
username	User	☑	☑	☑
username	Group	☑	☑	☑
username2	User	☑	☐	☑
username2	Group	☐	☐	☐
username3	User	☐	☐	☐
username3	Group	☐	☐	☐

6. Next to the team member's name, select or deselect the permissions for that user.

   NOTE: You can add a team member and set their access at the same time by typing their name in a username box, setting their permissions, and then clicking the Add button.

7. Click the Submit button.

   The selected permissions are added, and the deselected permissions are removed.

NOTE: If a team member is in the Workbench application when you give them access, they must refresh their browser window to see their current permissions.

Controlling non-team member access

You can choose to grant file or folder access to someone who is not part of the project team, as long as that person has an AEN account.

Sharing with individuals outside the team is a four step process:

1. *Copy or move the file or folder to your home directory*.
2. *Give the user read and execute access to your home directory*.
3. *Add the user to the file's permissions*.
4. *Have the user add your directory to their workbench*.

Copying a file or folder to your home directory

Your home directory is displayed at the bottom of the File Manager pane in the Workbench.

To protect the other files and folders in your home directory—those you are not providing permissions to a user to access—we recommended that you:

1. Create a sub-folder.
2. Rename the folder with the name of the user you are granting access to.
3. Copy or move the file you want to grant permissions for to the renamed folder.

The file is copied or moved to the new location and is ready for you to update the file permissions.

Granting file access

You must select read and execute access for a user to be able to view, but not edit, the files or folders.

1. Right-click the name of the file or folder you are granting access to.
2. In the menu that is displayed, select Permissions.
3. Click the Add button.
4. Type the username of the user to whom you are granting file access and press Enter.

TIP: If you grant access to a folder instead of a specific file, you only have to set permissions the first time you share the folder with each user, unless you need to update the permissions.

Adding file permissions for a user

Once a user is included in your Permissions list, you must *add the correct permissions* for the user, in the same way as you would for a team member.

Once complete, depending on the access granted, the user will be able to view, read, change, and execute the file.

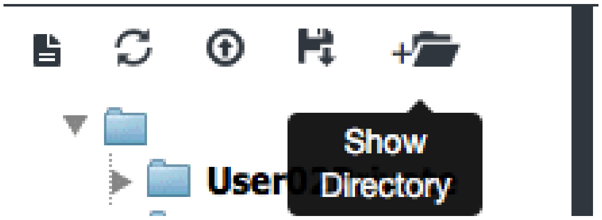
NOTE: If you change permissions for a folder instead of a file, the user will be able to see and access any files within that folder.

Adding a directory to a user's workbench

The user can now add your home directory to their Workbench File Manager.

To add your home directory to another user's workbench, have the other user follow these steps:

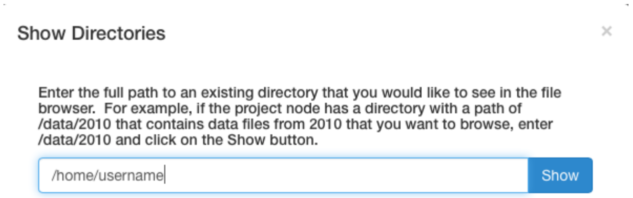
1. Click the Show Directory button at the top of the Workbench File Manager:



The Show Directories dialog box displays.

2. In the text box, type `/home/[yourusername]`.

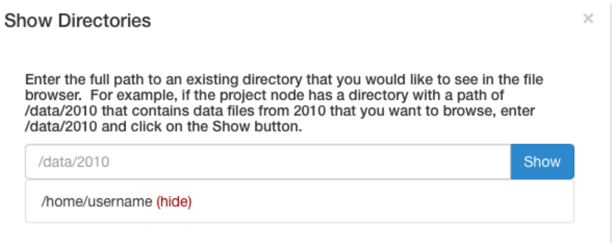
   NOTE: Replace `[yourusername]` with your AEN username.



Enter the full path to an existing directory that you would like to see in the file browser. For example, if the project node has a directory with a path of /data/2010 that contains data files from 2010 that you want to browse, enter /data/2010 and click on the Show button.

/home/username

3. Click the Show button.
4. Verify that the folder is now displayed below the text box:



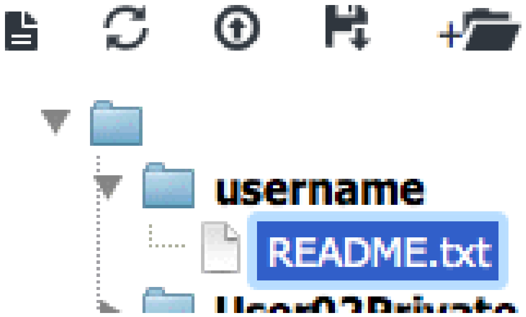
Enter the full path to an existing directory that you would like to see in the file browser. For example, if the project node has a directory with a path of /data/2010 that contains data files from 2010 that you want to browse, enter /data/2010 and click on the Show button.

/data/2010

/home/username (hide)

5. Close the Show Directories dialog box by clicking the X in the upper-right corner or by clicking anywhere outside the box.
6. Click the Refresh button.

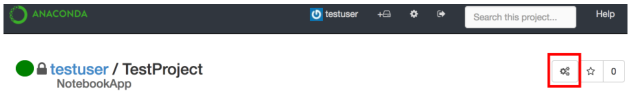
   The shared file is displayed in the File Manager:



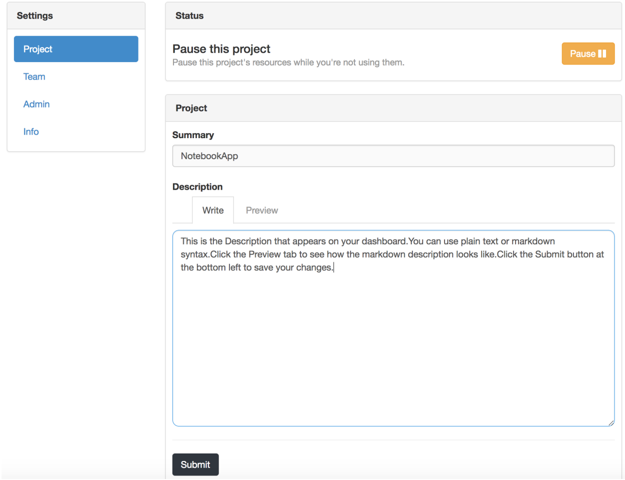
Starting and stopping a project

TIP: Stopping a project stops all the applications launched for that project that use resources when running, such as memory and compute cycles. It is best to stop projects when they are not in use.

1. On the project home page, click the Project Settings icon to open the Project Settings page.



2. In the **Settings** menu, select Project.



Pause this project

Pause this project's resources while you're not using them.

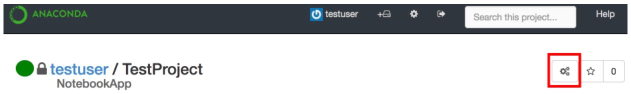
Summary: NotebookApp

Description: This is the Description that appears on your dashboard.You can use plain text or markdown syntax.Click the Preview tab to see how the markdown description looks like.Click the Submit button at the bottom left to save your changes.

3. In the Status section, click the Start or Stop button to toggle between manually starting and stopping your project.

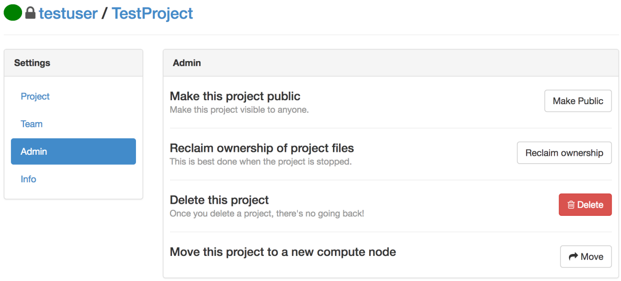
Making a project public or private

1. On the project home page, click the Project Settings icon to open the Project Settings page.



2. In the **Settings** menu, select Admin.

3. Click the Make Public button.



Make this project public
Make this project visible to anyone.

Reclaim ownership of project files
This is best done when the project is stopped.

Delete this project
Once you delete a project, there's no going back!

Move this project to a new compute node

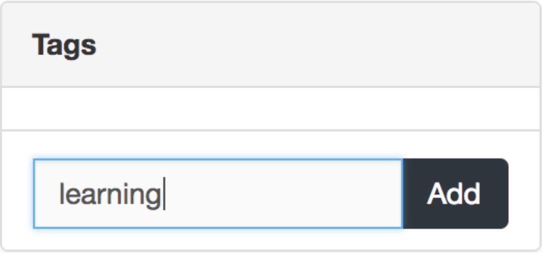
4. If the project is already public and you want to make it private, click the Make Private button.

Tagging a project

Existing tags assigned to a project are listed in the Tags section on the project's home page.

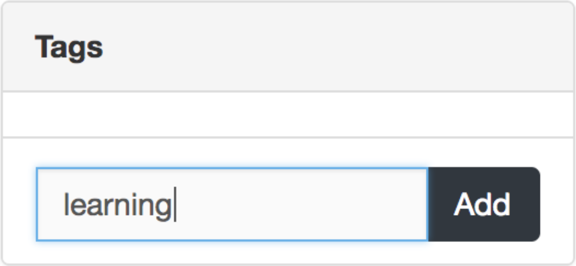
Adding a tag

1. In the Tags box, type the name of the tag you want to add:

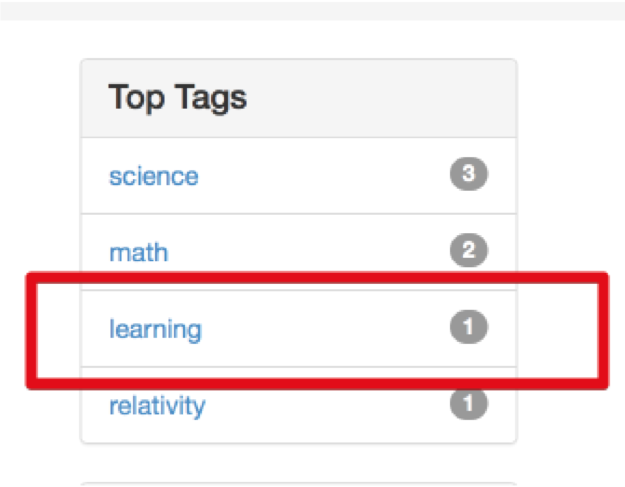


2. Click the Add button.

   The new tag is added to the Tags list:



If the tag was not already in the Top Tags list on your user home page, it is added. If the tag was already listed because another project used it, the number next to the tag is incremented:

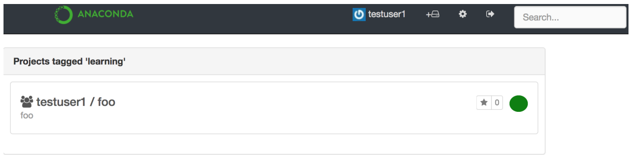


Top Tags

science	3
math	2
learning	1
relativity	1

Removing a tag

1. On your user home page, in the Top Tags list, click the tag name.



Projects tagged 'learning'

testuser1 / foo
foo

1. In the Tags list, click the X button next to tag name.

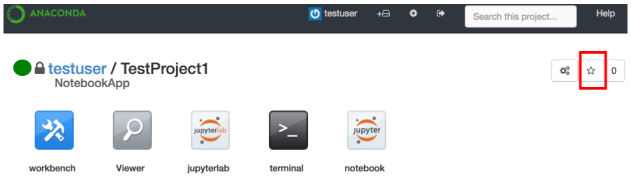
Starring a project (rating)

Starring a project makes it appear on your user home page in the Top Rated list.

Adding or removing stars for a project does not affect the stars added by other users.

1. Open the project that you want to star.
2. On the project home page, click the Star icon at the upper right:

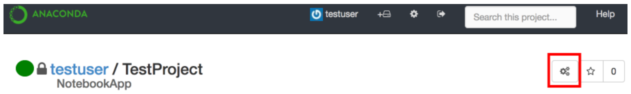
3. To unstar a project, click the Star icon again.



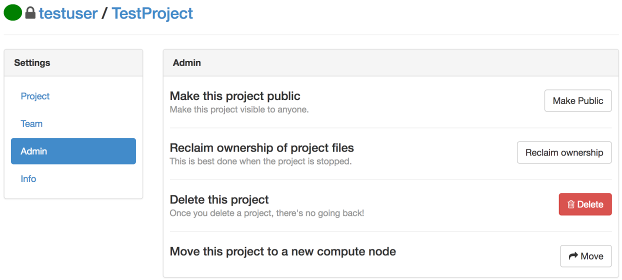
Claim ownership of a project

When you claim ownership of a project, ownership of all files and folders created by the team members on the project is transferred to you. Project files and folders are copied and renamed.

1. *Stop the project* to prevent team members from making changes while you are changing ownership.
2. On the project home page, click the Project Settings icon to open the Project Settings page.



3. In the **Settings** menu, select Admin.



Make this project public

Make this project visible to anyone.

Reclaim ownership of project files

This is best done when the project is stopped.

Delete this project

Once you delete a project, there's no going back!

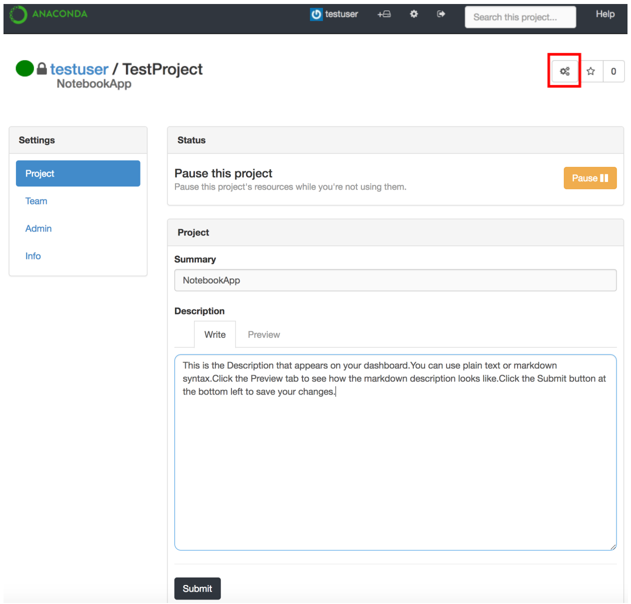
Move this project to a new compute node

4. Click the Reclaim ownership button.

Changing a project's summary or description

1. On the project home page, click the Project Settings icon to open the Project Settings page.

2. In the **Settings** menu, select Project.



Pause this project's resources while you're not using them.

Summary

NotebookApp

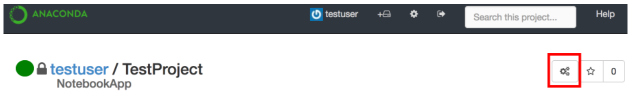
Description

This is the Description that appears on your dashboard.You can use plain text or markdown syntax.Click the Preview tab to see how the markdown description looks like.Click the Submit button at the bottom left to save your changes.

3. Update your project's summary using plain text or its description using Markdown syntax.
4. Click the **Preview** tab to see a preview of the Markdown description.
5. Click the Submit button.

Viewing a project's status

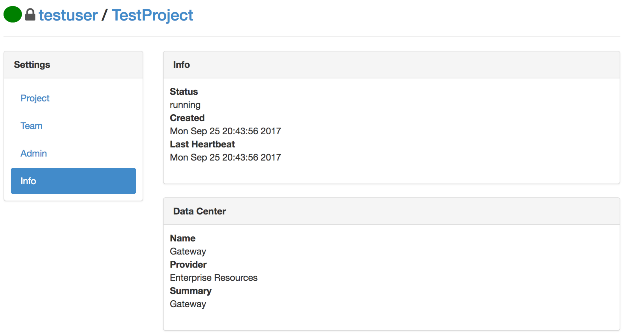
1. On the project home page, click the Project Settings icon to open the Project Settings page.



2. In the **Settings** menu, select Info.

On the Info page, you can see:

- Whether the project is currently running or stopped.
- When the project was created.
- When the project was last accessed.



Info

Status
running
Created
Mon Sep 25 20:43:56 2017
Last Heartbeat
Mon Sep 25 20:43:56 2017

Data Center

Name
Gateway
Provider
Enterprise Resources
Summary
Gateway

- The data center in which the project is running.

Viewing related projects

Related projects are listed on a project's home page.

These are projects that contain fields that are most similar to the current project.

TIP: You will only see projects to which you have been granted access: public projects, and private projects on which you are a team member.

How related projects are identified

To determine which projects should be listed in Related Projects:

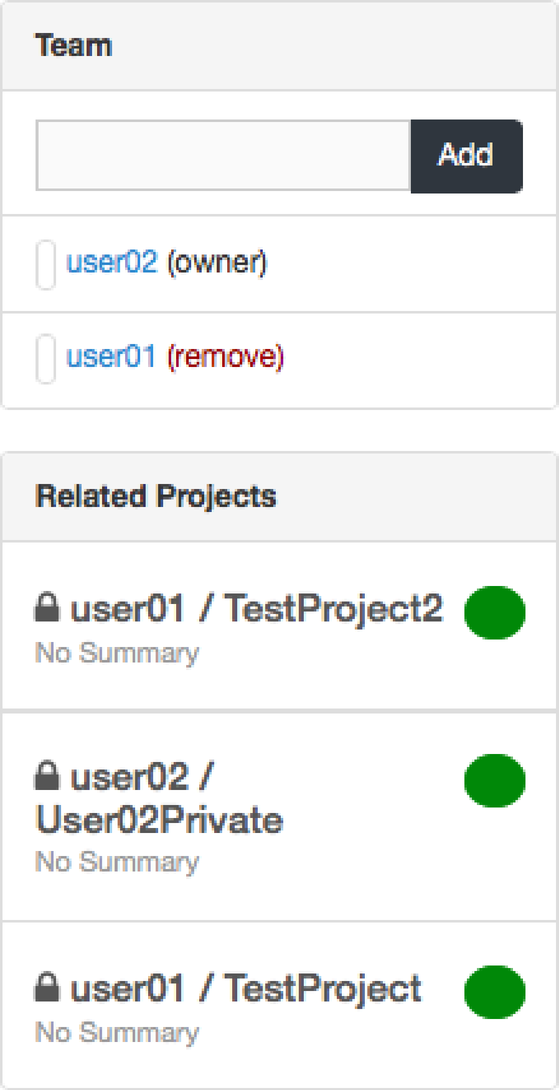
1. The recommendation engine scans the current project's files and weights the terms found to determine which of them to use for the likeness search.
2. The engine performs a search, with extra weight given to the "uses" and "imports" keywords.
3. The engine finds the files and projects that are most similar to the current project and scores the results.
4. The top-scoring matches are displayed in Related Projects. Only public projects and private projects to which you have access are included.

Viewing top-rated projects

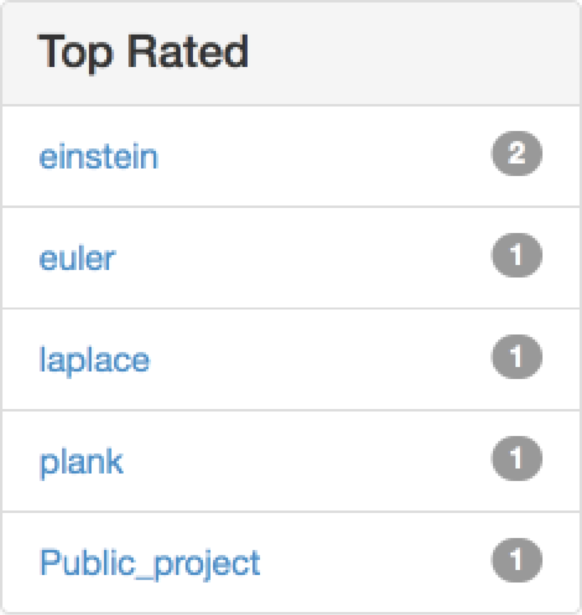
Top-rated projects are listed on your home page:

The number next to a project represents the number of stars that have been given to that project.

Click a project name to view the project's home page.



Team
Add
user02 (owner)
user01 (remove)
Related Projects
user01 / TestProject2
No Summary
user02 / User02Private
No Summary
user01 / TestProject
No Summary

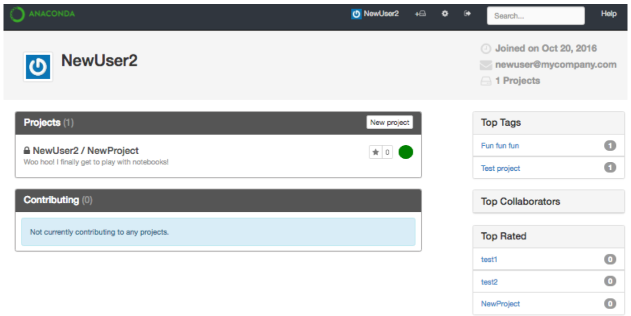


Top Rated

einstein	2
euler	1
laplace	1
plank	1
Public_project	1

Using tags to find a project

The top tags used on your projects are listed on your home page:



NewUser2

Joined on Oct 20, 2016
newuser@mycompany.com
1 Projects

Projects (1)

NewUser2 / NewProject
Woo hoo! I finally get to play with notebooks!

Contributing (0)

Not currently contributing to any projects.

Top Tags

Fun fun fun	1
Test project	1

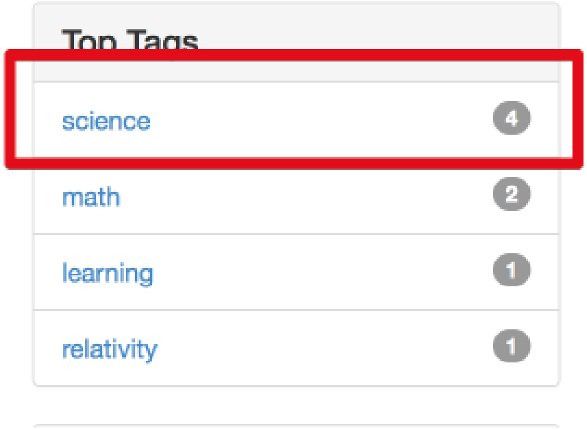
Top Collaborators

Top Rated

test1	0
test2	0
NewProject	0

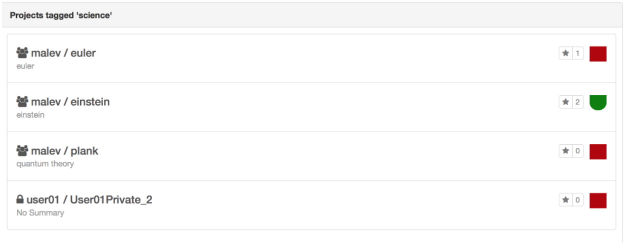
To list all projects that share a specific tag, click the tag name:

A list of projects with the selected tag is displayed:



Top Tags

science	4
math	2
learning	1
relativity	1



Projects tagged 'science'

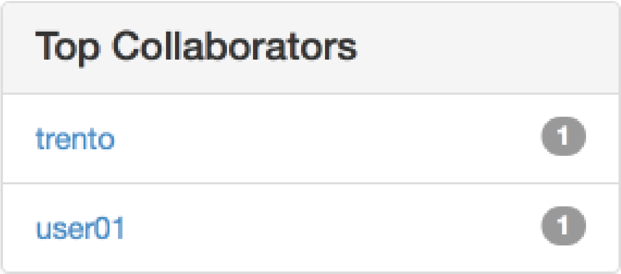
malev / euler euler	★ 1	🟥
malev / einstein einstein	★ 2	🟩
malev / plank quantum theory	★ 0	🟥
🔒 user01 / User01Private_2 No Summary	★ 0	🟥

TIP: The list includes only projects that you have access to: public projects, and private projects on which you are a team member.

Click a project name to open the project's home page.

Viewing your top collaborators

Your top collaborators are listed on your home page:



Top Collaborators

trento	1
user01	1

These are the team members who have the most projects in common with you.

To view a collaborator's home page—where you can see all public projects and the private projects they have shared with you—click the collaborator's name.

Sharing projects and notebooks

For information on sharing projects via the project settings and access control, see *Sharing projects*.

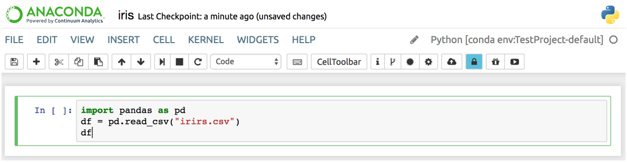
To upload a Jupyter Notebook to Anaconda Repository:

1. Log in to Repository by running the `anaconda login` command or by using the login user interface provided by the *nbextension*.

   CAUTION: If you are not using a secure connection, we strongly recommended that you use the command line to log in.

2. To share your notebook environment, select the Attach conda environment checkbox. This ensures that your team members will have the right environment for your notebook.

3. Click the Upload button to upload your notebook to your local Repository or to Anaconda.org, depending on how your administrator has set up AEN:

   
   ANACONDA
   Powered by Continuum Analytics
   iris Last Checkpoint: a minute ago (unsaved changes)
   FILE
   EDIT
   VIEW
   INSERT
   CELL
   KERNEL
   WIDGETS
   HELP
   Python [conda env:TestProject-default]
   Code
   CellToolbar

   ```
   In [ ]: import pandas as pd
           df = pd.read_csv("irirs.csv")
           df
   ```

   NOTE: If you have not yet logged into Repository or Anaconda Cloud, or have not created an account, you will be asked to do so.

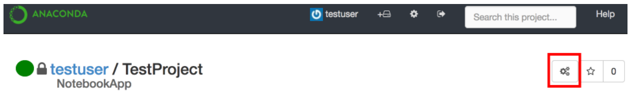
Other ways to share a notebook

- Print—In the **File** menu, select Print.
- Download and share—In the **File** menu, select one of the following options:
  - Download as Notebook.
  - Download as Python.
  - Download as HTML.
  - Download as Markdown.
  - Download as ReStructured Text.
  - Download as PDF.
- Share and control team members' direct access to read, write and/or execute your notebook file or folder. For more information, see *Controlling access to your project*.
- Share and control non-team members' file or folder access. For more information, see *Controlling access to your project*.
- Create a presentation with *NBPresent 4.1*.

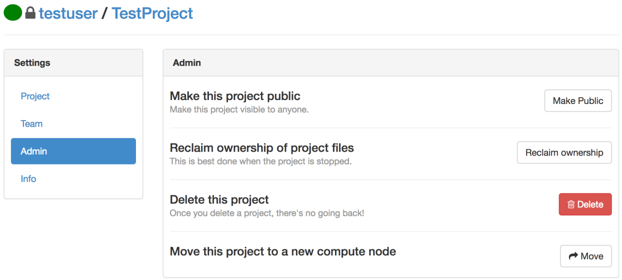
Deleting a project

CAUTION: Deleting a project deletes all project files and information! There is no undo option.

1. Download a copy of any project files that you need to save.
2. On the project home page, click the Project Settings icon to open the Project Settings page.



3. In the **Settings** menu, select Admin.



Make this project public
Make this project visible to anyone.

Reclaim ownership of project files
This is best done when the project is stopped.

Delete this project
Once you delete a project, there's no going back!

Move this project to a new compute node

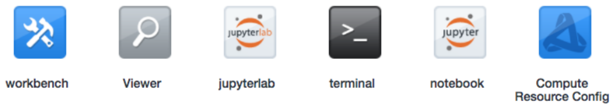
4. Click the Delete button.

Using AEN applications

The applications in your project make it easy for you to interact with your files and data, manage your project's resources and to customize your AEN experience.

To use applications, log into AEN, then select the project you want to work on or create a new project and open it.

On the project home page, the following application icons are displayed:



TIP: Each application opens in a new browser tab. You can run multiple applications at the same time in your project.

For more information on each AEN application, see:

- *Using Workbench*—File viewer and manager, including permissions settings.
- *Using Viewer*—View-only versions of notebooks and other text files.
- *Using JupyterLab*—Alpha preview of the next generation notebook.
- *Using Terminal*—Basic bash shell Terminal.
- *Using Jupyter Notebook*—Jupyter Notebooks with extensions.
- *Using Compute Resource Configuration*—Project information, view and manage applications.

Using Workbench

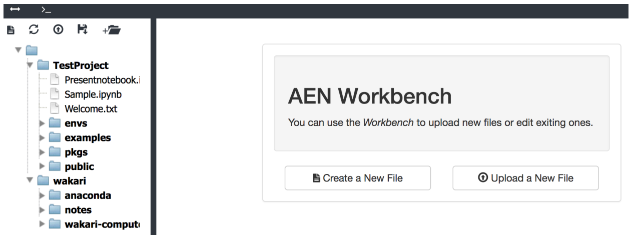
Workbench is a file viewer and manager that includes a file editor and file permissions manager.

You can use Workbench to:

- Upload and download files using the *File Manager*.
- Create new files and folders using the *File Manager*.
- Copy and move files to new locations using the *File Manager*.
- Rename files and/or folders using the *File Manager*.
- Manage the *access permissions* of team members.
- Grant or revoke *access to non-team members*.

Workbench also includes a simple Terminal application, which is convenient because the File Manager is always visible, making navigation simple.

When you first open Workbench, the File Manager is displayed in the left pane, and the Create a New File and Upload a New File buttons are in the right pane:



AEN Workbench

You can use the *Workbench* to upload new files or edit exiting ones.

When you open a file or Workbench Terminal, it is displayed in the right pane. To make the Create or Upload a file options re-appear, refresh your browser window.

Two small icons are displayed in the black navigation bar at the top of the Workbench page. Hovering over them displays tool tips that describe their use:

- The Toggle icon displays or hides the File Manager.
- The Terminal icon opens a simple terminal window.

Opening Workbench

To open Workbench:

1. Log in to AEN.
2. Select the project you want to work on, or create a new project and open it.
3. On the project home page, click the Workbench icon:

Workbench opens in a new browser window.

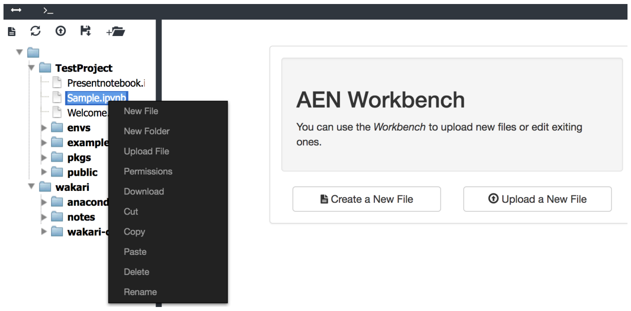
Using File Manager

The File Manager is an intuitive way to interact with your files and folders.

Using the options drop-down menu

To perform any of the actions described below:

1. Right-click on any folder to display the options drop-down menu.
2. Select one of the following options:
   - New File—Create and edit a new file.
   - New Folder—Create a new folder.
   - Upload File—Upload a file to the selected folder. You can also drag a file to the folder.
   - Permissions—*Control access to files and folders*.
   - Cut—Cut the selected file or folder.
   - Copy—Copy the selected file or folder.
   - Paste—Paste a previously cut or copied file or folder.
   - Delete—Delete the highlighted file or folder.
   - Rename—Rename the highlighted file or folder.



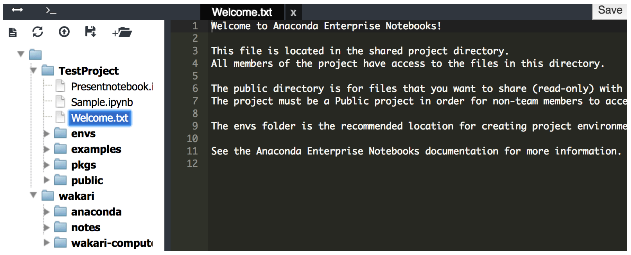
AEN Workbench

You can use the *Workbench* to upload new files or edit exiting ones.

Editing files using the File Editor

1. Double-click any text file in the File Manager.

   The File Editor opens in the right pane:



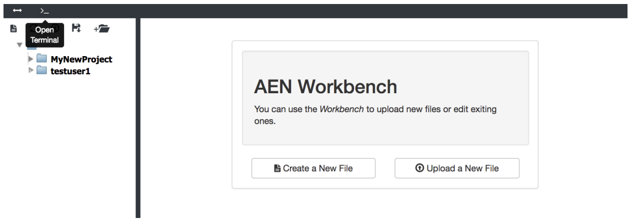
```
Welcome to Anaconda Enterprise Notebooks!

This file is located in the shared project directory.
All members of the project have access to the files in this directory.

The public directory is for files that you want to share (read-only) with
The project must be a Public project in order for non-team members to acce

The envs folder is the recommended location for creating project environme

See the Anaconda Enterprise Notebooks documentation for more information.
```

2. When you finish editing the file, click the Save button.

   NOTE: To close the file without saving, click the X at the top of the page under the file name.

Opening the Workbench terminal

In the navigation bar, click the Open terminal icon:



AEN Workbench

You can use the *Workbench* to upload new files or edit exiting ones.

A Terminal—bash shell—is displayed in the right pane.

TIP: You can open additional terminals by clicking the Open terminal icon again, or by clicking the Plus + icon at the top of an open terminal.

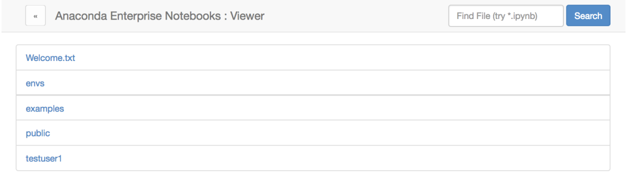
To move between terminal windows, click the **Terminal** tab in the navigation bar, then select the number of the terminal window you want to work in.

Using Viewer

The Viewer application displays a static, view-only version of your notebooks and other text files by rendering the text files directly and using the NBConvert tool to convert notebooks to static HTML.

1. Log in to AEN.
2. Select the project you want to work on, or create a new project and open it.
3. On the project home page, click the Viewer icon.

   Viewer opens in a new browser window:



4. Click any folder to view its contents, or click any filename to view the file.
5. To search for a file or folder name, type text in the Find File box, then press the Enter key. This is not a full-text search, but wildcards are permitted.

Using JupyterLab

JupyterLab is an early alpha-preview of the next generation of the Jupyter Notebook. It is included so that you can take a tour and play with its capabilities.

CAUTION: JupyterLab is experimental. It is not yet intended for production work.

JupyterLab does not include any of the notebook extensions that are available in the *Jupyter Notebook app*.

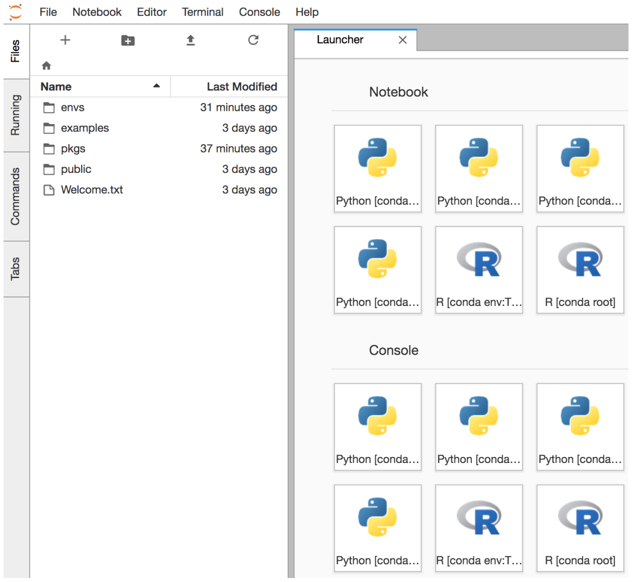
For more information about JupyterLab, see the documentation.

You can also download and print a `Jupyter cheat sheet` on using Jupyter Notebook and the new JupyterLab.

To open JupyterLab:

1. Log in to AEN.
2. Select the project you want to work on, or create a new project and open it.
3. On the project home page, click on the JupyterLab icon.

   JupyterLab opens in a new browser window:



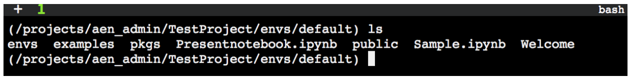
Name	Last Modified
envs	31 minutes ago
examples	3 days ago
pkgs	37 minutes ago
public	3 days ago
Welcome.txt	3 days ago

Experiment with the application on your own, using the **Notebook**, **Editor**, **Terminal** and **Console** menus.

To review a guided tour of all of the features JupyterLab will contain when it is ready for production, click the Take a tour link in the right pane.

Using Terminal

The Terminal application is a simple bash shell terminal that runs in your browser:



```
(/projects/aen_admin/TestProject/envs/default) ls
envs  examples  pkgs  Presentnotebook.ipynb  public  Sample.ipynb  Welcome
(/projects/aen_admin/TestProject/envs/default) █
```

Using Terminal, you can:

- Access your home directory and your project drive.
- Open multiple shells within one instance of Terminal.
- Open multiple instances of Terminal in the same browser window.

1. Log in to AEN.
2. Select a project you want to work on, or create a new project and open it.
3. On the project home page, click the Terminal icon:

Terminal opens the project directory in a new browser window.

By default, the project directory is `/projects/username/project-name`.

EXAMPLE: `/projects/TestUser/MyFirstNotebook`

4. To see the physical path of your directory, run the Print Working Directory command `pwd -P`.

   TIP: The physical path `-P` is important because project attaches data to the beginning of your virtual path to keep your project files together.

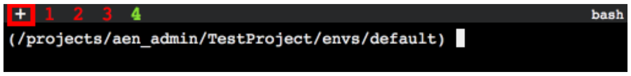
5. To navigate out of your project directory to your home directory, run the command `cd`.
6. To return to your project directory, run the command `cd/projects/username/project-name`.

TIP: If you are new to navigating in a terminal, you may want to use *the Workbench terminal*, which includes a visual navigation tree in the File Manager.

Using multiple Terminals

You can open as many terminals as you want.

To open another shell in the terminal, in the upper left of the pane, click the plus + icon.



```
(/projects/aen_admin/TestProject/envs/default)
```

A corresponding number appears after the plus + icon and 1.

To move to another Terminal, click the corresponding number.

The color of the number tab changes to show which terminal is currently selected.

Using Jupyter Notebook

The Jupyter Notebook application allows you to create and edit documents that display the input and output of a Python or R language script. Once saved, you can share these files with others.

NOTE: Python and R language are included by default, but with customization, Notebook can run several other kernel environments.

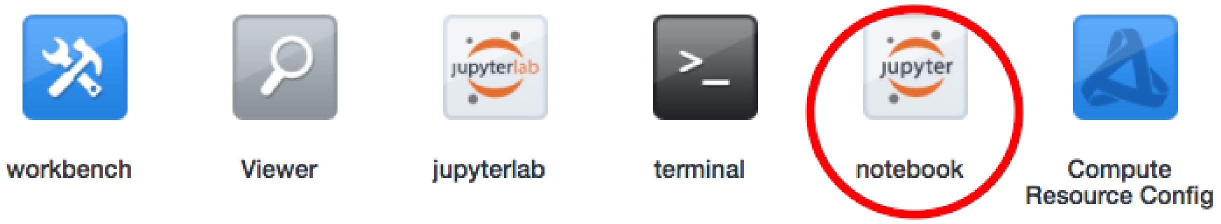
This page provides a brief introduction to Jupyter Notebooks for AEN users.

For the official Jupyter Notebook user instructions, see Jupyter documentation.

For information on the notebook extensions available in AEN, see *Using Jupyter Notebook extensions*.

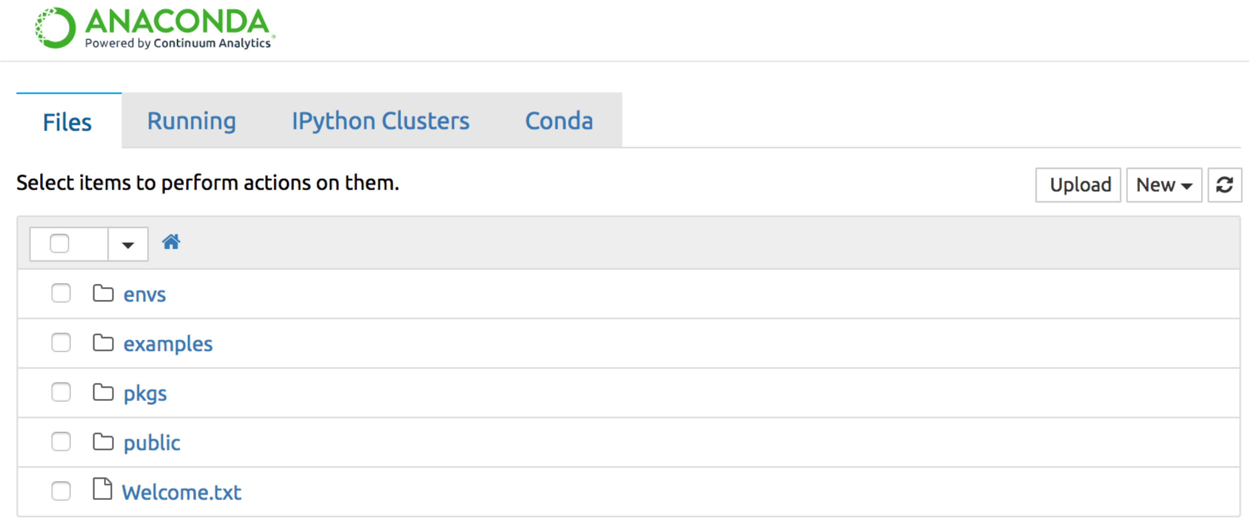
Opening the Jupyter Notebook application

1. Log in to AEN.
2. Select the project you want to work on, or create a new project and open it.
3. On the project home page, click the Jupyter Notebook icon:



Jupyter Notebook opens in a new browser window:

TIP: You can see the same *File Manager* in the Terminal, Workbench, and Viewer applications.



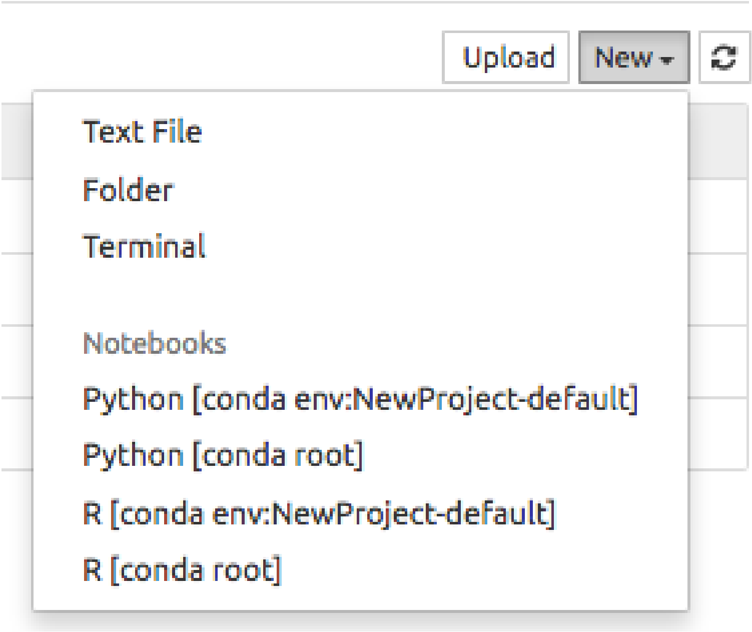
Using example notebooks

The `Examples` folder in Jupyter Notebook contains several types of Notebook examples created in Python—and one with R language—kernel environments.

Open any example notebook to experiment and see how it works.

Creating a new Jupyter Notebook

1. An the top right of the **Files** tab, click the New button.



2. Select the kernel environment to create your new notebook in.

   NOTE: Customizable Python and R Language kernel environments are automatically created for you during project creation.

   - Your project's default conda env kernels are a cloned copy of the root environment. You can customize them and install and delete additional packages.
   - Root environment is managed by your Administrator. You cannot make or save any changes to it.
   - You can switch between Python, R language and any other custom kernels in the notebook as you work in your notebook. For more information, see *Using the Synchronize Environments extension*.

   The new notebook is saved in the related project directory and displayed.

Using Jupyter Notebook extensions

The following extensions are available for use with AEN's Jupyter Notebook application:

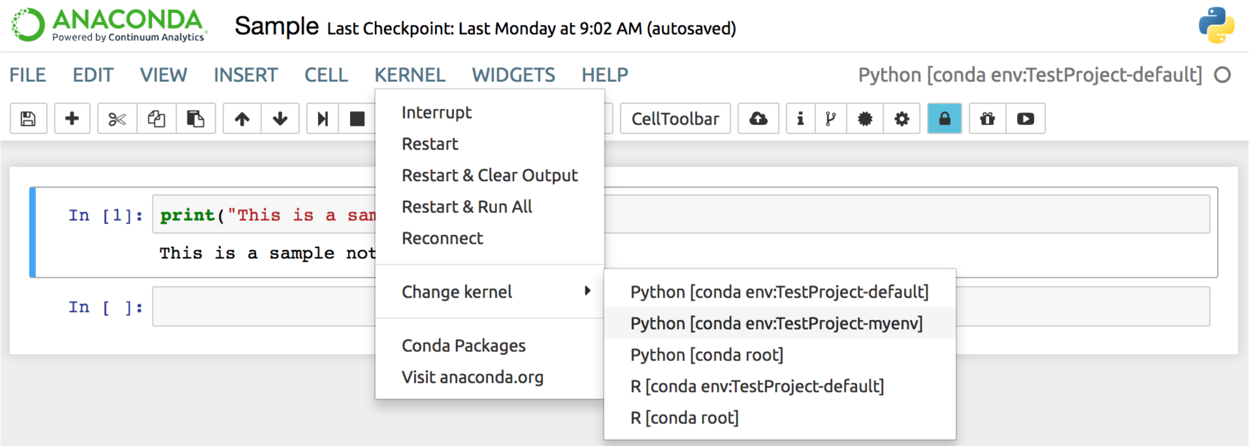
- *Synchronize Environments* with Jupyter from the **Kernel** menu.
- *Locking* adds multi-user capability from the Lock button.
- *Revision Control Mechanism (RCM)* adds Status, Checkout and Commit buttons.
- *Conda environment and package management* tab.
- *Conda notebook* adds conda management inside Notebook from the Kernel > Conda Packages menu option.
- *Anaconda Cloud integration* from the Publish to cloud button.
- *Notebook Present* turns your notebook into a PowerPoint-style presentation.

Using the Synchronize Environments extension

The Synchronize Environments extension allows you to apply a Python, R language or any other custom environment inside your current notebook session, without needing to start up several Notebook instances using each of the selected environments.

To change environments:

1. Open the **Kernel** menu.


ANACONDA
Powered by Continuum Analytics
Sample Last Checkpoint: Last Monday at 9:02 AM (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-default]
CellToolbar
Interrupt
Restart
Restart & Clear Output
Restart & Run All
Reconnect
Change kernel
Conda Packages
Visit anaconda.org
Python [conda env:TestProject-default]
Python [conda env:TestProject-myenv]
Python [conda root]
R [conda env:TestProject-default]
R [conda root]

```
In [1]: print("This is a sam
        This is a sample not
In [ ]:
```

2. Click the Change kernel option.
3. From the list, select the environment to use.

NOTE: In AEN 4.1+ the default kernel for projects is `default`. In versions prior to 4.0, the default kernel for projects is `root Python`.

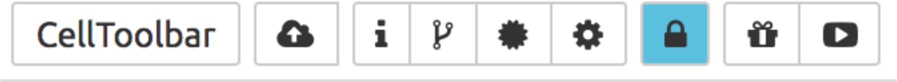
Using the Locking extension

Multi-user capabilities are engaged in AEN when multiple users work in the same notebook file.

The Locking extension allows you to lock a notebook to prevent multiple team members from making changes at the same time. Notebooks are automatically locked when you open them.

If team members open a notebook and make changes while it is locked, their save capability is disabled, and they cannot overwrite the notebook.

To override the lock, they must actively take control of the locked file by clicking the Lock icon in the Notebook menu bar:



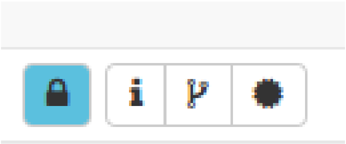
NOTE: This is a soft locking model. Team members can choose to override your lock to save their work. If you give team members write access to your files, confirm that they understand that they should never unlock your file unless they are making meaningful, non-destructive team contributions.

Using the Revision Control Mechanism extension

The Revision Control Mechanism (RCM) Jupyter Notebook extension provides simple version control for notebook files. It uses the internal Jupyter functionality to perform tasks.

On the surface, RCM uses a simple linear model, but beneath that is a more complex git-based branching model. This model uses the latest wins as its main merging strategy to prevent merge conflicts.

The RCM Jupyter Notebook extension adds four buttons:



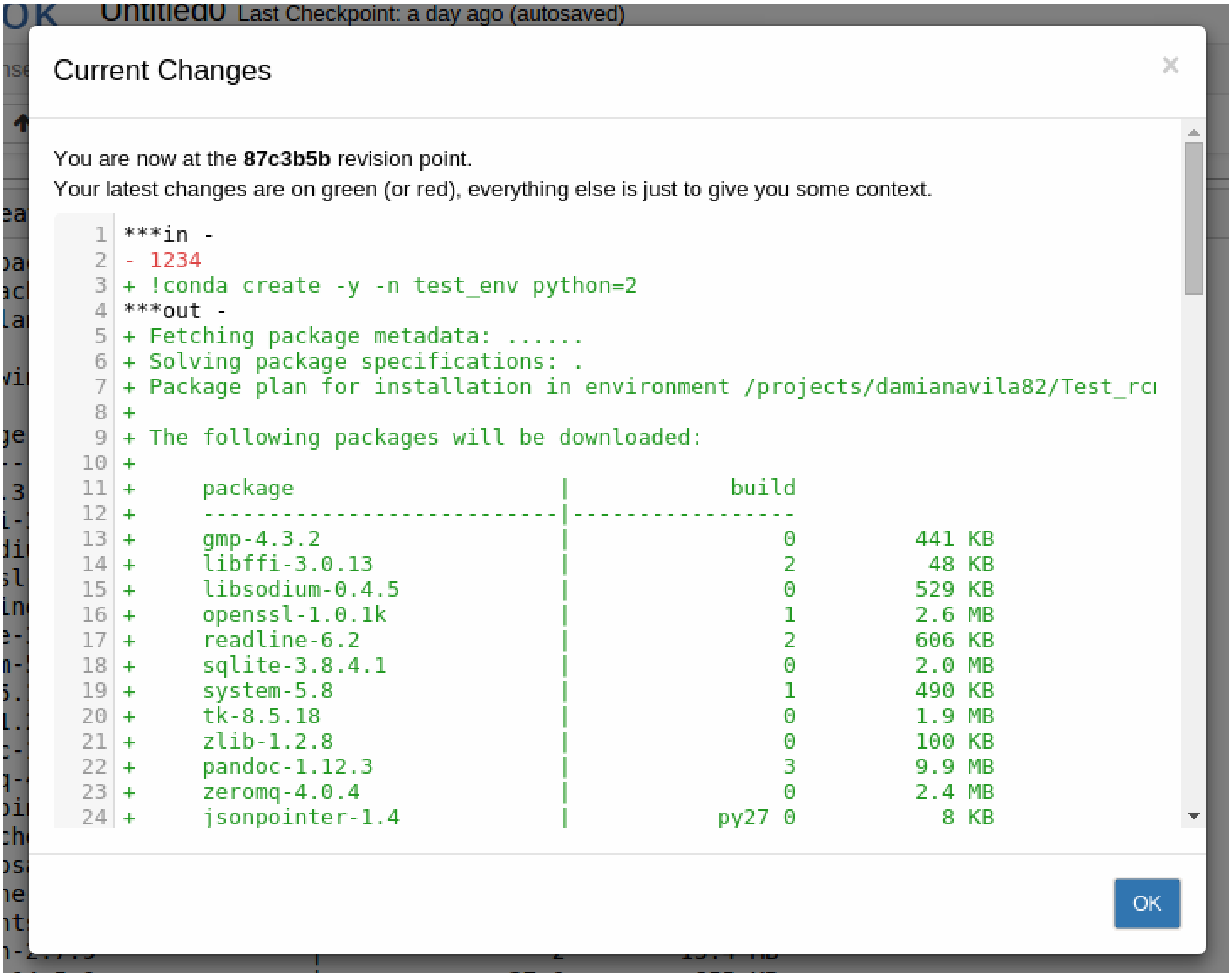
- *Status*.
- *Checkout*.
- *Commit*.
- *Configure git*.

TIP: If you do not see the RCM buttons, see *Setting up RCM for the first time*.

Using the Status button

The Status button allows you to see what revision you are on.

Clicking the Status button displays:



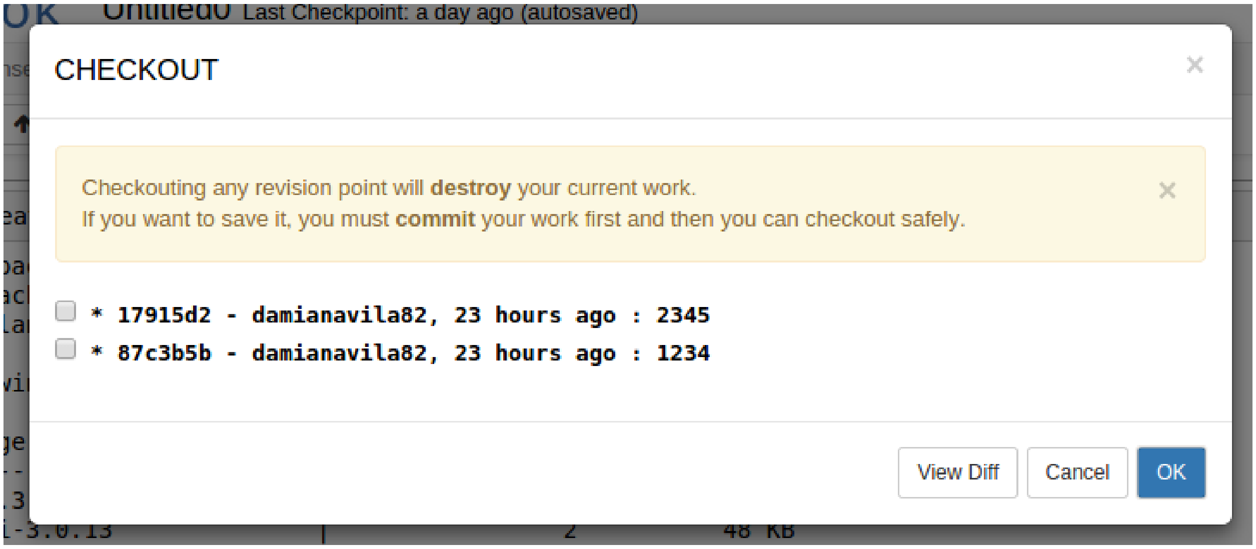
You are now at the **87c3b5b** revision point.
Your latest changes are on green (or red), everything else is just to give you some context.

```
***in -
- 1234
+ !conda create -y -n test_env python=2
***out -
+ Fetching package metadata: ......
+ Solving package specifications: .
+ Package plan for installation in environment /projects/damianavila82/Test_rc
+
+ The following packages will be downloaded:
+
+     package                    |            build
+     ---------------------------|-----------------
+     gmp-4.3.2                  |                0         441 KB
+     libffi-3.0.13              |                2          48 KB
+     libsodium-0.4.5            |                0         529 KB
+     openssl-1.0.1k             |                1         2.6 MB
+     readline-6.2               |                2         606 KB
+     sqlite-3.8.4.1             |                0         2.0 MB
+     system-5.8                 |                1         490 KB
+     tk-8.5.18                  |                0         1.9 MB
+     zlib-1.2.8                 |                0         100 KB
+     pandoc-1.12.3              |                3         9.9 MB
+     zeromq-4.0.4               |                0         2.4 MB
+     jsonpointer-1.4            |           py27 0           8 KB
```

Using the Checkout button

The Checkout button allows you to view a list of the previous revision points, check out a previous revision or compare differences between revisions.

Clicking the Checkout button displays:



CHECKOUT

Checkouting any revision point will **destroy** your current work.
If you want to save it, you must **commit** your work first and then you can checkout safely.

□ * 17915d2 - damianavila82, 23 hours ago : 2345
□ * 87c3b5b - damianavila82, 23 hours ago : 1234

Checking out a previous revision

To checkout a notebook at an earlier revision point:

1. Select the checkbox next to the desired revision point.
2. Click the OK button.

   A copy of the notebook at the selected revision point is displayed.

NOTE: If you have not saved the work in your current project window, checking out a previous revision destroys it. If in doubt, click the Cancel button and save your work before reverting to a previous revision point.

Comparing revisions

To compare 2 previous revision points:

1. Select the checkboxes of the revision points to compare.
2. Click the View Diff button.

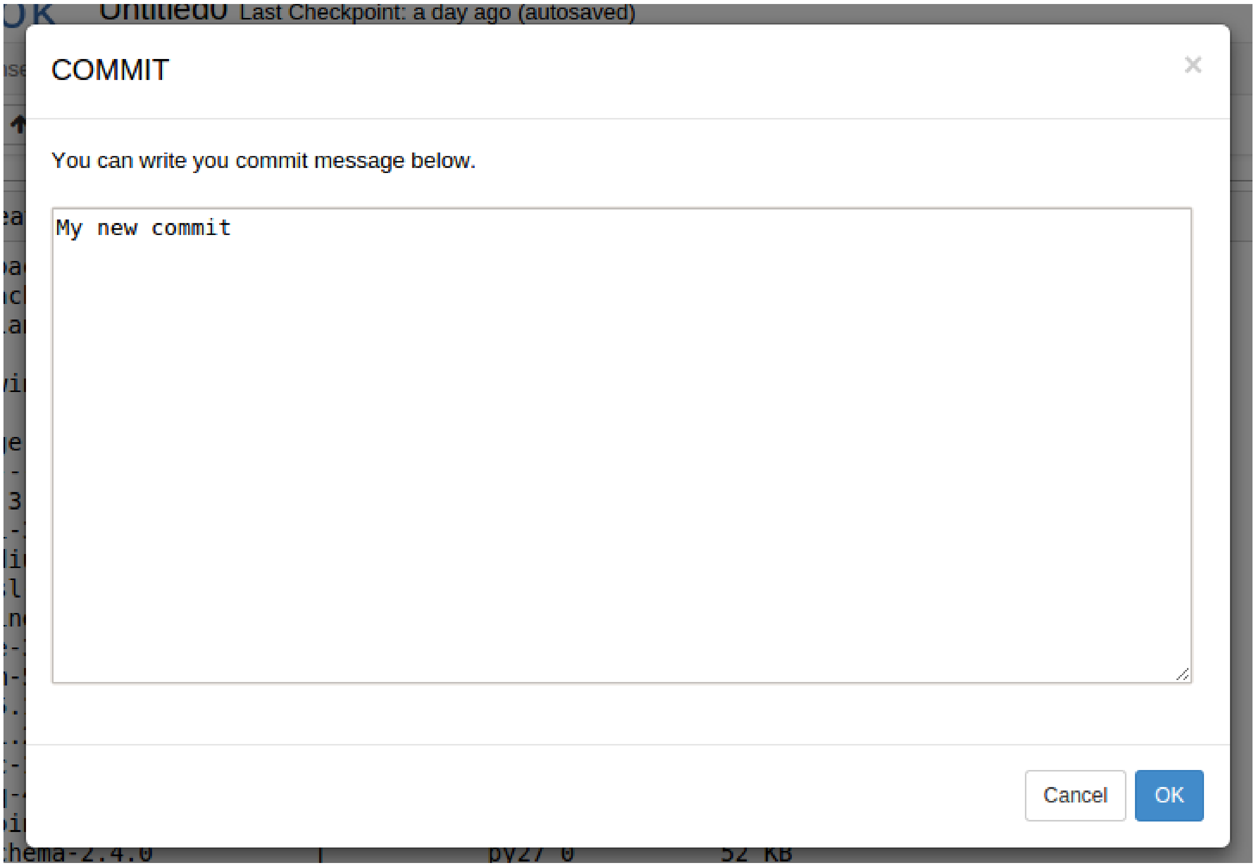
   A side-by-side comparison is displayed.

   Click the Cancel button to close the differences window.

Using the Commit button

The Commit button allows you to save or persist the current changes, keeping a permanent record of any changes that are introduced, so that you do not have to worry about losing important data.

Clicking the Commit button displays:



You can write you commit message below.

```
My new commit
```

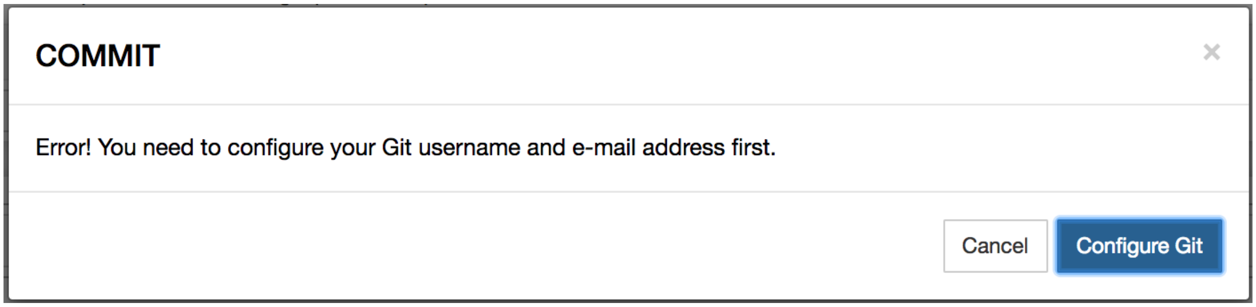
1. Enter a description of the changes in the commit as a reminder in case you need to revert back to it later.
2. Click the OK button.

   Your changes are committed and a revision point is created.

If Git user name and user email are not set, the following window appears:

Configure Git and then try to commit again.

TIP: You can roll back committed changes by *checking out a previous version*.



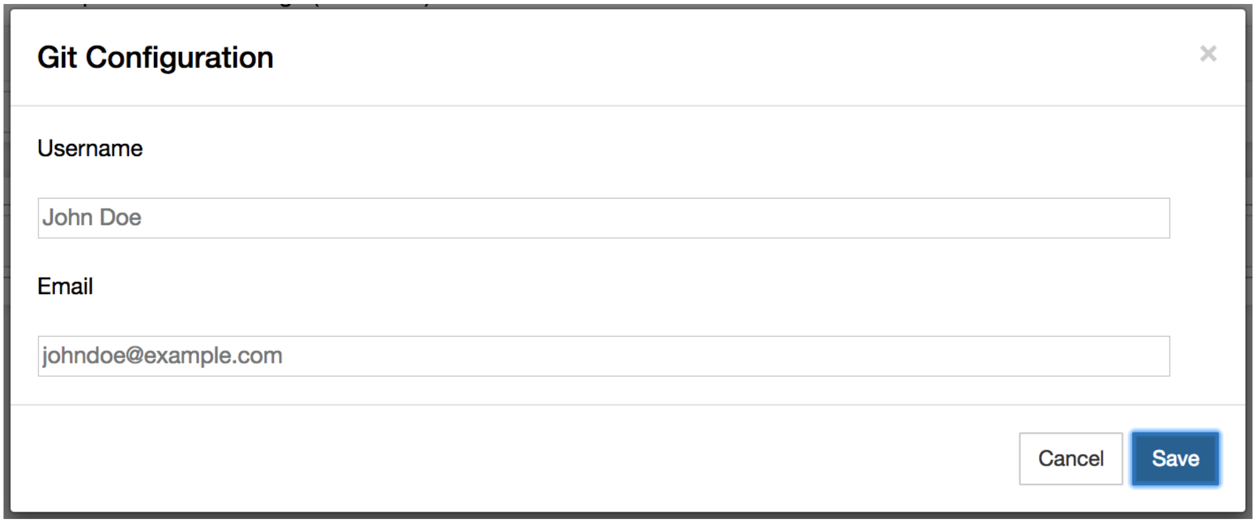
COMMIT

Error! You need to configure your Git username and e-mail address first.

Using the Configure git button

The Configure git button allows you to configure Git user name and email values.

After clicking the Configure Git button, the following window appears:



Git Configuration

Username

John Doe

Email

johndoe@example.com

Enter user name and e-mail address. Click the OK button when finished.

Setting up RCM for the first time

If you do not see the RCM buttons in your notebook:

1. Go to the project home page.
2. Open the Terminal application.
3. In the terminal window, run:

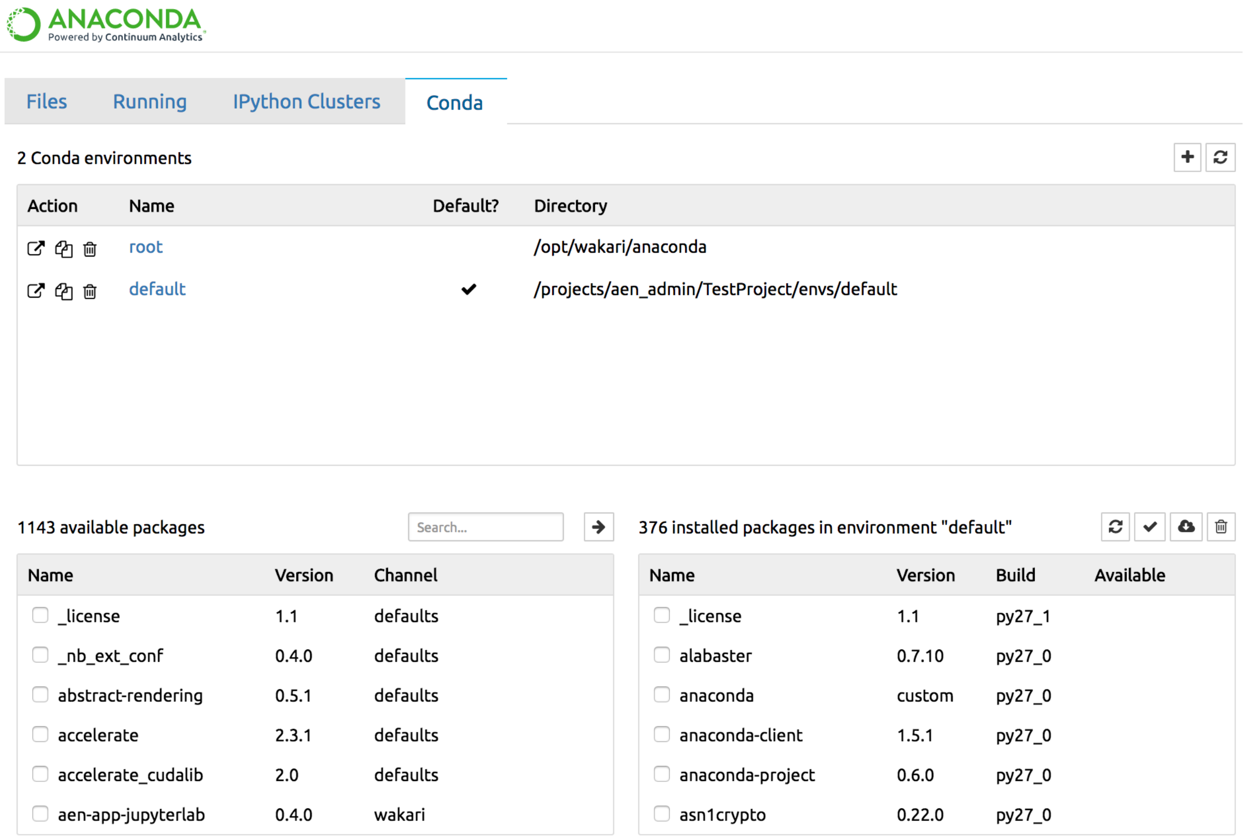
```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

NOTE: Change `you@example.com` to your email address, and `Your Name` to your actual name.

4. Open Jupyter Notebook and refresh the page.

Using the NBConda extension

The NBConda extension adds a Conda tab to your notebook for easy environment and package management from within the notebook.



2 Conda environments

Action	Name	Default?	Directory
	root		/opt/wakari/anaconda
	default	✓	/projects/aen_admin/TestProject/envs/default

1143 available packages

Name	Version	Channel
□ _license	1.1	defaults
□ _nb_ext_conf	0.4.0	defaults
□ abstract-rendering	0.5.1	defaults
□ accelerate	2.3.1	defaults
□ accelerate_cudalib	2.0	defaults
□ aen-app-jupyterlab	0.4.0	wakari

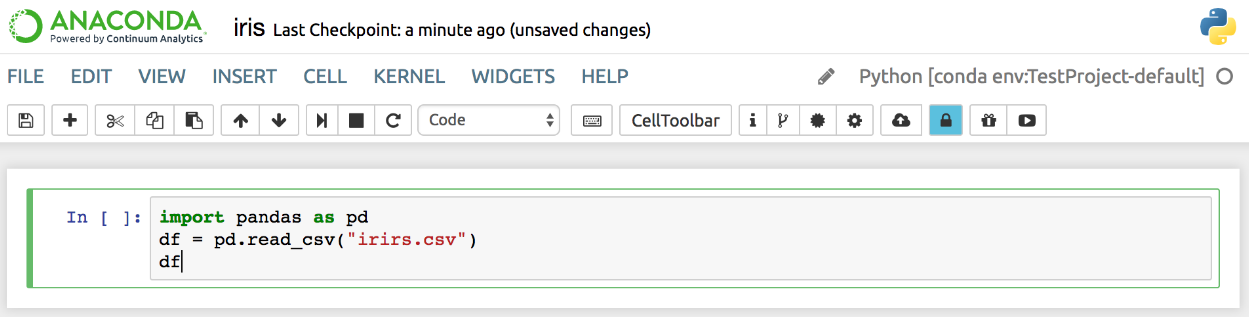
376 installed packages in environment "default"

Name	Version	Build	Available
□ _license	1.1	py27_1	
□ alabaster	0.7.10	py27_0	
□ anaconda	custom	py27_0	
□ anaconda-client	1.5.1	py27_0	
□ anaconda-project	0.6.0	py27_0	
□ asn1crypto	0.22.0	py27_0	

Click the Conda tab in a notebook to display:

- Conda environments list—export, clone or delete an environment in the action column, or create a new environment by clicking the plus + icon. Switch to an environment by clicking it; packages for that environment are displayed below in the installed packages list.
- Conda available packages list—for the selected environment in currently configured channels, search for packages and click a package name to install it.
- Installed packages list—in the selected environment, check for updates, update or delete selected packages.

TIP: While you are in any notebook, you can jump to the NBConda extension for that environment by clicking the **Kernel** menu and selecting Conda Packages:


ANACONDA
Powered by Continuum Analytics
iris Last Checkpoint: a minute ago (unsaved changes)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-default]
Code
CellToolbar

```
In [ ]: import pandas as pd
        df = pd.read_csv("irirs.csv")
        df
```

Using the Conda Notebook extension

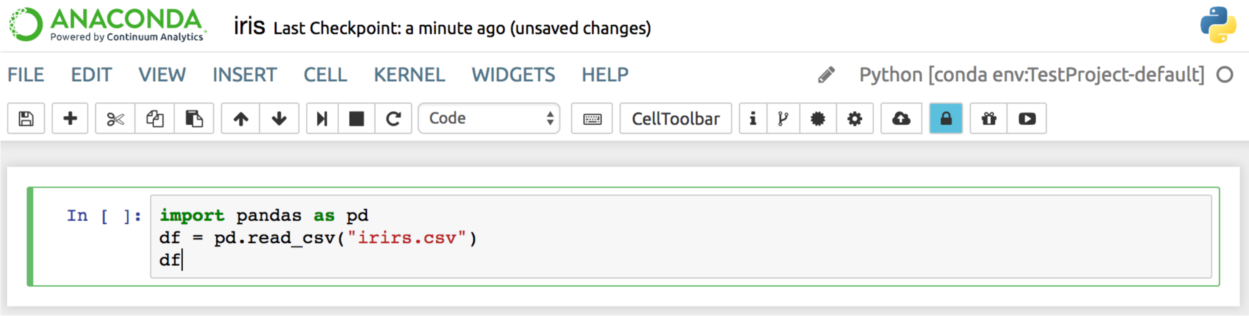
The Conda Notebook extension adds the Conda Packages option to the **Kernel** menu.

Select the Conda Packages option to display a list of all of the Conda packages that are currently used in the environment associated with the running kernel, as well as any available packages.

From the Conda Packages option, you can perform all of the tasks available in the *Conda tab*, but they will only apply to the current environment.

Using the Anaconda Cloud extension

The Anaconda Cloud extension adds the Cloud button to your notebook, allowing you to easily upload your notebook to Cloud:

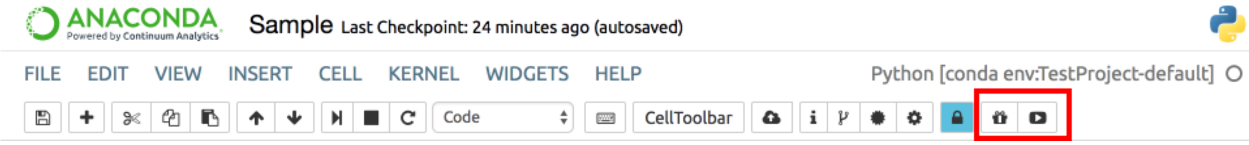

ANACONDA
Powered by Continuum Analytics
iris Last Checkpoint: a minute ago (unsaved changes)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-default]
Code
CellToolbar

```
In [ ]: import pandas as pd
        df = pd.read_csv("irirs.csv")
        df
```

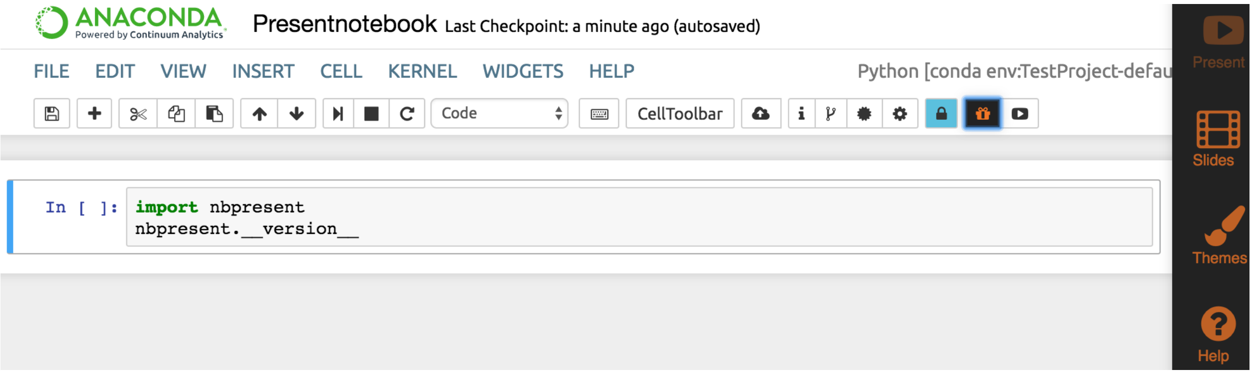
Using the Notebook Present extension

The AEN Notebook Present extension turns your notebook into a Microsoft PowerPoint-style presentation.

The Present extension adds 2 buttons to Notebook's menu bar—Edit Presentation and Show Presentation:


ANACONDA
Powered by Continuum Analytics
Sample Last Checkpoint: 24 minutes ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-default]
Code
CellToolbar

To begin using Notebook Present, click the Edit Presentation button.


ANACONDA
Powered by Continuum Analytics
Presentnotebook Last Checkpoint: a minute ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-defau
Code
CellToolbar
Present
Slides
Themes
Help

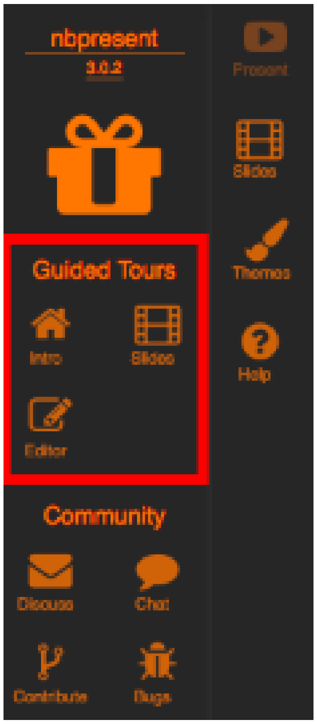
```
In [ ]: import nbpresent
        nbpresent.__version__
```

The Notebook Present sidebar is displayed on the right side of your browser:

Clicking each icon changes the menu and layout of your notebook.

Clicking the Help icon displays 3 tours—demonstrations—of the main features of Present:

- *Intro tour*.
- *Slides tour*.
- *Editor tour*.



Select one of the tours to view a short presentation regarding the specifics of that feature.

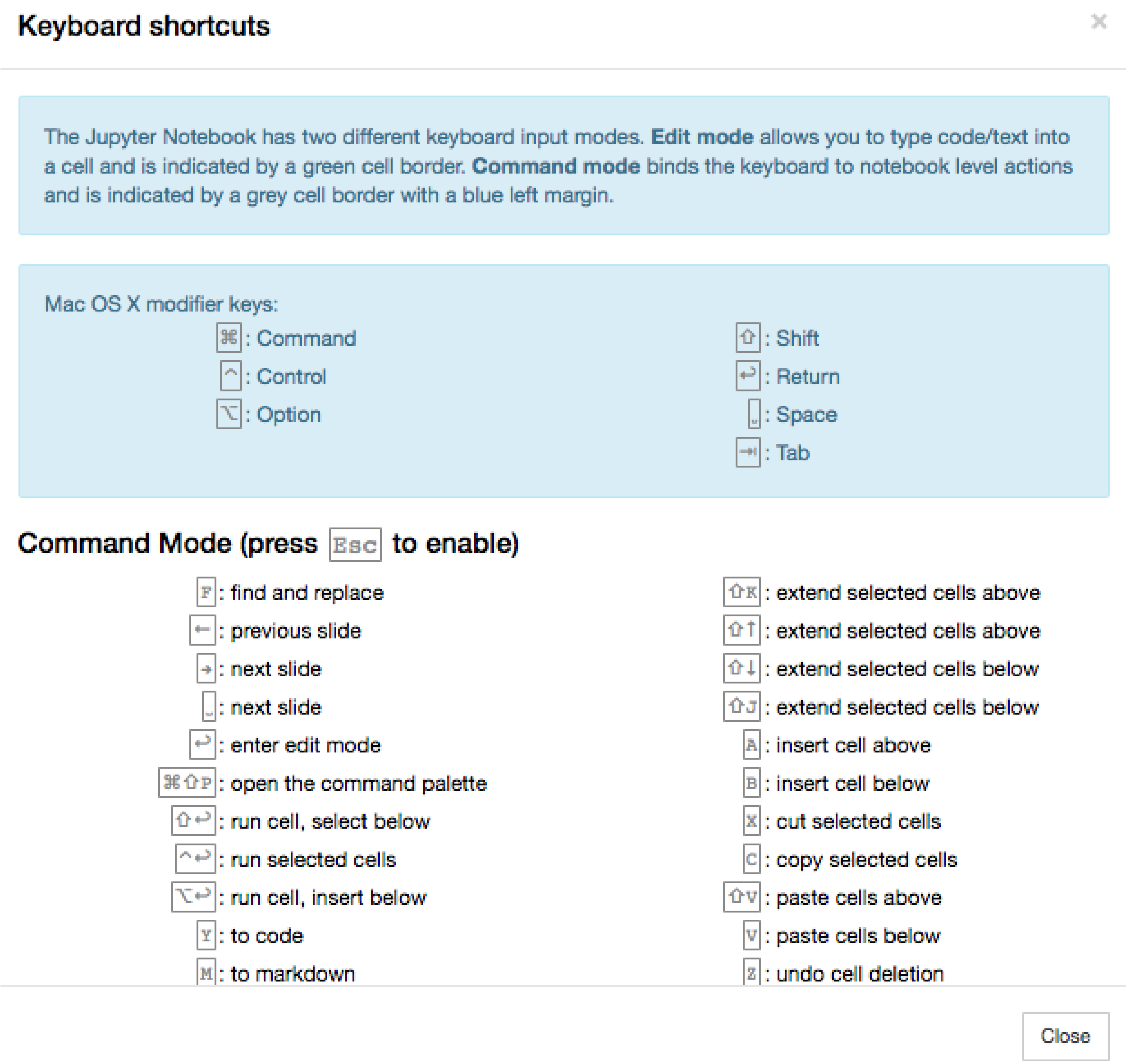
Intro tour

The Intro tour is a 2-minute presentation that explains how to use the main features of Present, including a description of each button's purpose.

NOTE: At any time, you can pause, go back to the previous or move forward to the next slide.

The following information is covered in the Intro tour:

- App Bar—When Authoring, this allows you control the content and style of your presentation. It also can be used to activate several keyboard shortcuts for editing:
- Stop Authoring—Clicking the Edit Presentation button again stops Authoring, and removes all keyboard shortcuts.
- Show Presentation—If you just want to run your presentation without using any Authoring tools, just click the Show Presentation button.
- Presenting/Authoring—Once you've made some slides, start Presenting, where you can use most Notebook functions with the Theme we have defined, as well as customize slides on the fly.
- Slides button—Slides, made of Regions linked to Cell Parts are the bread and butter of any presentation, and can be imported, created, linked, reordered, and edited here.
- Theming—Theming lets you select from existing colors, typography, and backgrounds to make distinctive presentations. The first theme you select will become the default, while you can choose custom themes for a particular slide, like a title.
- Saving—Whenever you save your Notebook, all your presentation data will be stored right in the Notebook .ipynb file.
- Downloading—After you've made a presentation, you can download it as an HTML page by choosing Download → Download As: Presentation (.html) in the menu.
- Help—Activate Help at any time to try other tours, connect with the Present developers and community, and other information.



Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level actions and is indicated by a grey cell border with a blue left margin.

Mac OS X modifier keys:

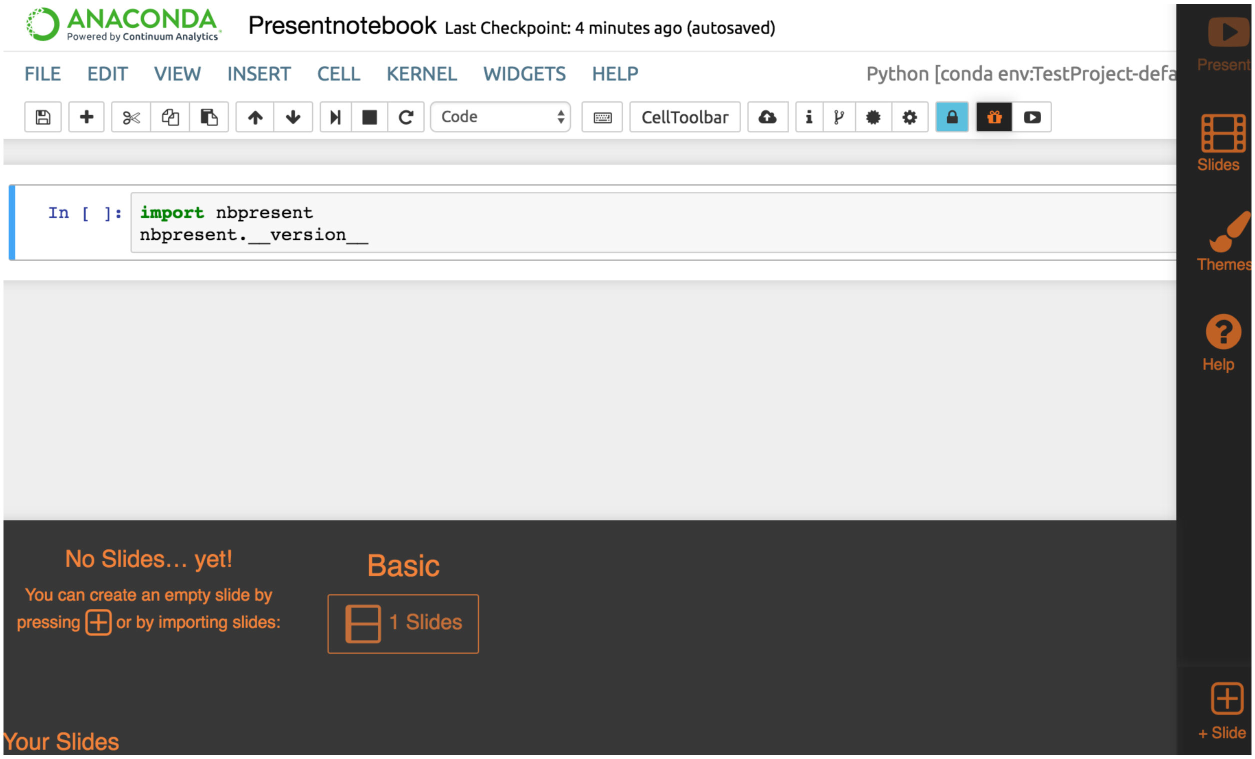
⌘ : Command
^ : Control
⌥ : Option

⇧ : Shift
↩ : Return
␣ : Space
⇥ : Tab

Command Mode (press Esc to enable)

F : find and replace
← : previous slide
→ : next slide
␣ : next slide
↩ : enter edit mode
⌘⇧P : open the command palette
⇧↩ : run cell, select below
^↩ : run selected cells
⌥↩ : run cell, insert below
Y : to code
M : to markdown

⇧K : extend selected cells above
⇧↑ : extend selected cells above
⇧↓ : extend selected cells below
⇧J : extend selected cells below
A : insert cell above
B : insert cell below
X : cut selected cells
C : copy selected cells
⇧V : paste cells above
V : paste cells below
Z : undo cell deletion



Presentnotebook Last Checkpoint: 4 minutes ago (autosaved)

```
In [ ]: import nbpresent
        nbpresent.__version__
```

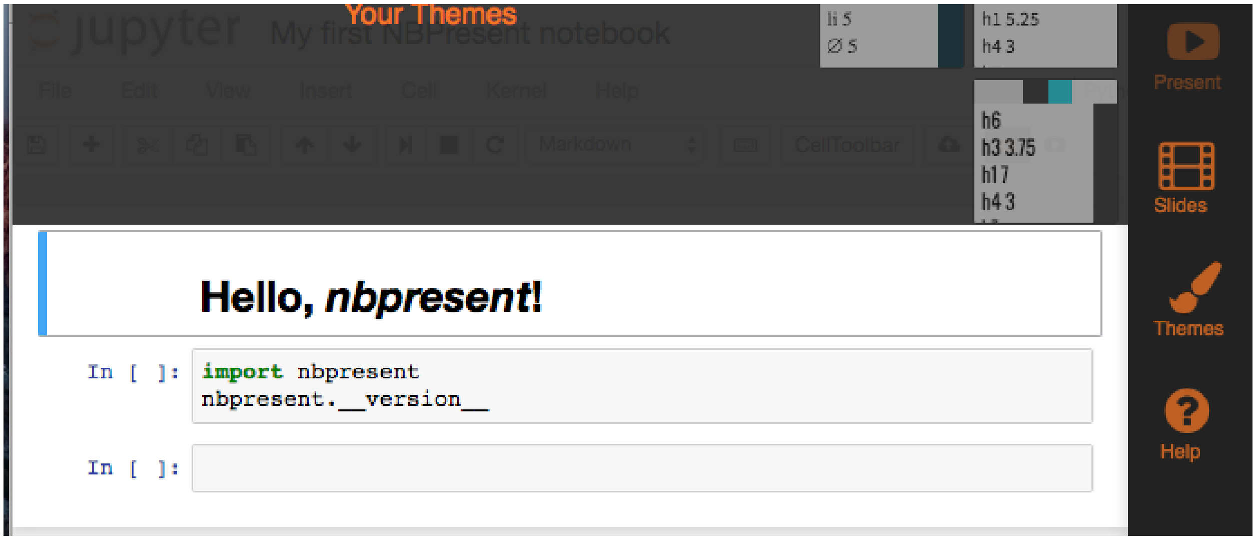
No Slides… yet!

You can create an empty slide by pressing ⊞ or by importing slides:

Basic

1 Slides

Your Slides



Your Themes

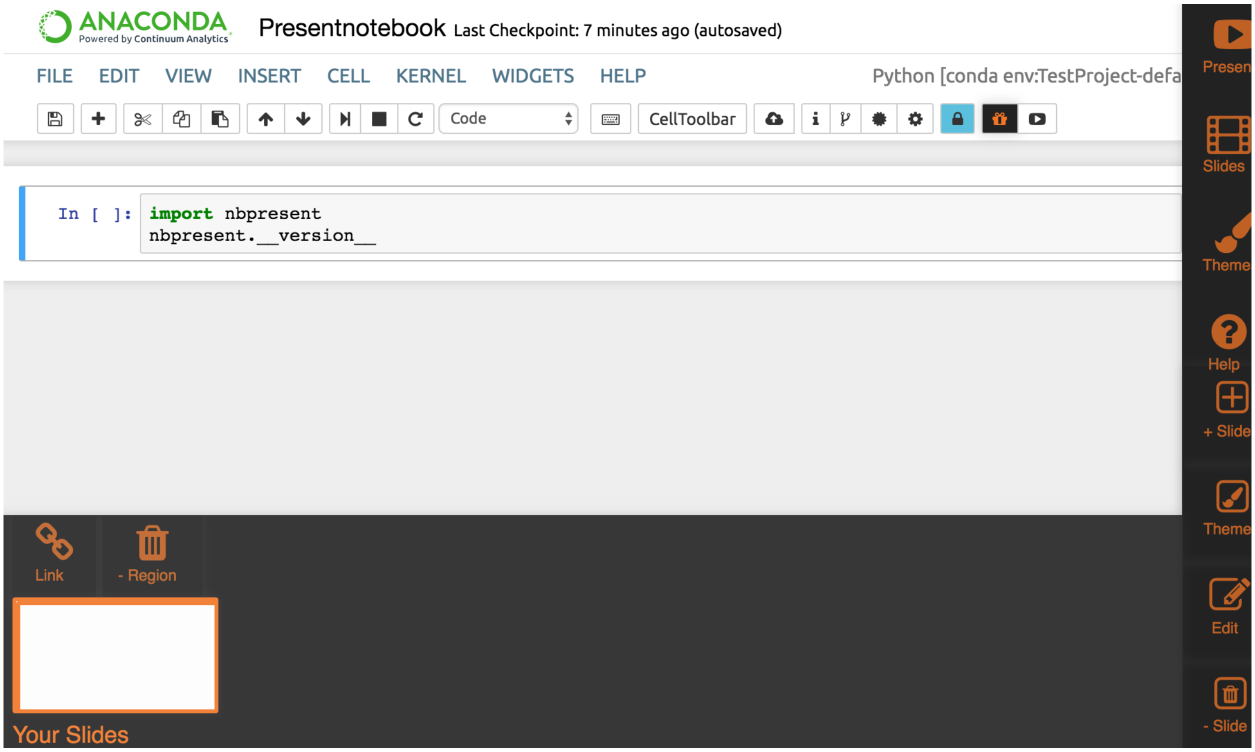
My first NBPresent notebook

Hello, *nbpresent*!

```
In [ ]: import nbpresent
        nbpresent.__version__

In [ ]:
```

Slides tour

Slides make up a presentation. Clicking Slides toggles the sorter view and the Slide Toolbar on and off:



```
In [ ]: import nbpresent
        nbpresent.__version__
```

The Slides tour explains how to create and manage slides, including the following information:

- Slide Toolbar—Create a new slide. Clicking + Slide will offer some choices for creating your new slide.
- Import—The quickest way to create a presentation is to import each cell as a slide. If you've already created slides with the official slideshow cell toolbar or RISE, you can import most of that content.
- Template Library—You can create a presentation from an existing template.
  - Reuse Slide as Template—You can create a presentation based on an existing slide.
  - Simple Template—A common template is the Quad Chart, with four pieces of content arranged in a grid.
- Region—The Quad Chart has four Regions. To select a region, click it.
  - Link a Region to a Cell Part—Each Region can be linked to a single Cell Part using the Link Overlay, which shows all of the parts available.
    - Cell Part: Source (blue)–Source, such as code and Markdown text.
    - Cell Part: Outputs (red)–Outputs, such as rich figures and script results.
    - Cell Part: Widgets (purple)–Jupyter widgets, interactive widgets that provide both visualization and user input.

    - * Cell Part: Whole (orange)–Finally, a Whole Cell, including its Source, Widgets and Outputs can be linked to a single region.
  - Unlink a region from a Cell Part—Unlinking removes the connection between a region and a cell part, without deleting either one.
  - Region: Trashing—Trashing a Region permanently deletes it, without affecting any linked Cell Part.
  - Part Thumbnail—We'll try to draw a part thumbnail. It can only be reliably updated when a linked Cell Part is on-screen when you mouse over it, but you should usually be able to get an idea of what you're seeing. The colors of the regions correspond to the cell types.
- Presenting—Clicking the Present button while editing brings up the Presenter with editing mode still enabled:
  - Linked inputs and widgets are still interactive.
  - Go forward—Click to go to the next slide
  - Go back—Click to go back to the previous slide
  - Go back to the beginning—Click to go back to the first slide
  - My work is done here—Click to go back to the Notebook.

Editor tour

Once you've made a few slides, you'll likely want to customize them. The Editor tour explains how to edit your notebook, including the following information:

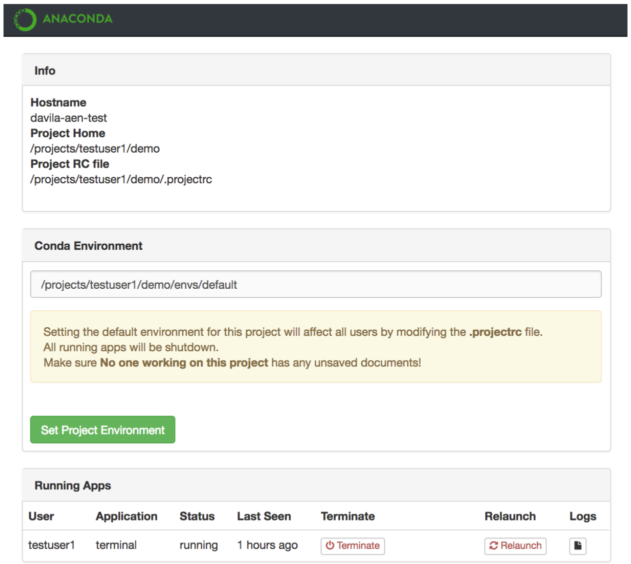
- Editing Slides—Activate the Slide Editor by double-clicking it, or by clicking Edit Slide.
- Region Editor—Click to drag Regions around and resize them.
- Region Tree—Reorder Regions and see the details of how Regions will show their linked Parts.
- Add Region—Add new regions.
- Attribute Editor—Edit the properties of a region.
- Data Layouts—In addition to manually moving regions, you can apply these layouts to automatically fill your slides.
- More Regions—Add more regions—with a weight of 1.
- Tree Weight—Make a Region bigger or smaller, based on its relative weight.
- 12 Grid—A compromise between the Free and Treemap layouts, the 12 Grid option rounds all of the values in a layout to a factor of 12.

Using Compute Resource Configuration

The Compute Resource Configuration (CRC) application displays information about the current project and allows you to set a custom project environment and view and manage your other AEN applications, including stopping, starting, restarting and viewing the logs of each.

The CRC application screen contains 3 sections:

- *Info*.
- *Conda environment*.
- *Running apps*.



Info

Hostname
davila-aen-test
Project Home
/projects/testuser1/demo
Project RC file
/projects/testuser1/demo/.projectrc

Conda Environment

/projects/testuser1/demo/envs/default

Setting the default environment for this project will affect all users by modifying the **.projectrc** file.
All running apps will be shutdown.
Make sure **No one working on this project** has any unsaved documents!

Running Apps

User	Application	Status	Last Seen	Terminate	Relaunch	Logs
testuser1	terminal	running	1 hours ago	Terminate	Relaunch	

Info

The Info section displays:

- Hostname—IP address of the host computer.
- Project Home—File path to the project home.
- Project RC file—File path to the project runtime configuration file `.projectrc`. This file is sourced when a user opens any AEN application. It sets several AEN internal environment variables, sets up the project environment and sets additional user environment variables for the project.

Conda environment

This section displays the path to the default conda environment.

CAUTION: Changing the default environment will affect all users. Be sure that no team members have any unsaved documents before changing the project environment.

To change the default conda environment location:

1. Edit the path to point to your preferred conda environment.
2. Click the Set Project Environment button.

   Your `.projectrc` file is modified.

Running apps

The Running Apps section displays a list of users and the applications that are in use, as well as when the app was last modified.

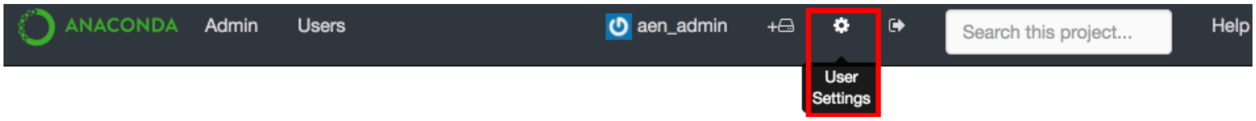
To terminate any individual application, click the Terminate button.

To stop and re-launch any individual application, click the Relaunch button.

To review the run logs of any active application, which may be useful for troubleshooting, click the Logs button.

Managing your account

To access your account information, click the User Settings icon in the AEN navigation bar:



Updating your public profile

Your public profile is made up of a name, a personal URL, your company and location.

1. In the left navigation pane, click the **Public Profile** tab.
2. To update your profile picture, create a Gravatar that is associated with the email address you used to create your AEN account. The gravatar will automatically appear.

Changing your password

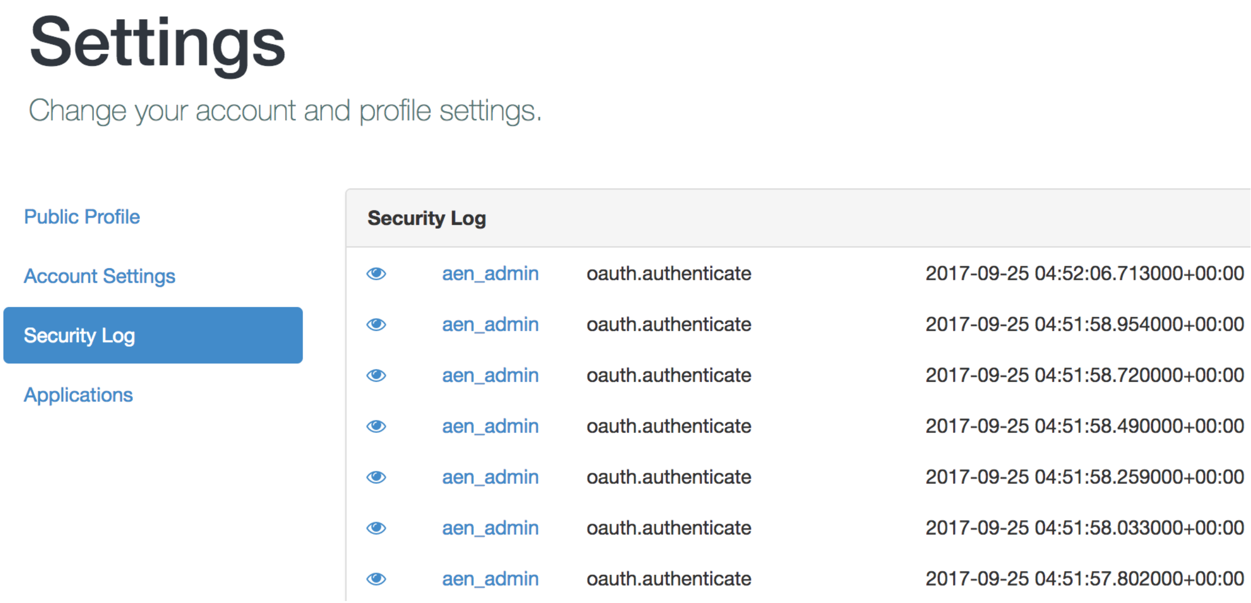
1. In the left navigation pane, click the **Account Settings** tab.

Deleting your AEN account

1. In the left navigation pane, click the **Account Settings** tab.

Viewing account operations

1. In the left navigation pane, click the **Security Log** tab to view a list of operations performed on your account.



Settings

Change your account and profile settings.

Security Log

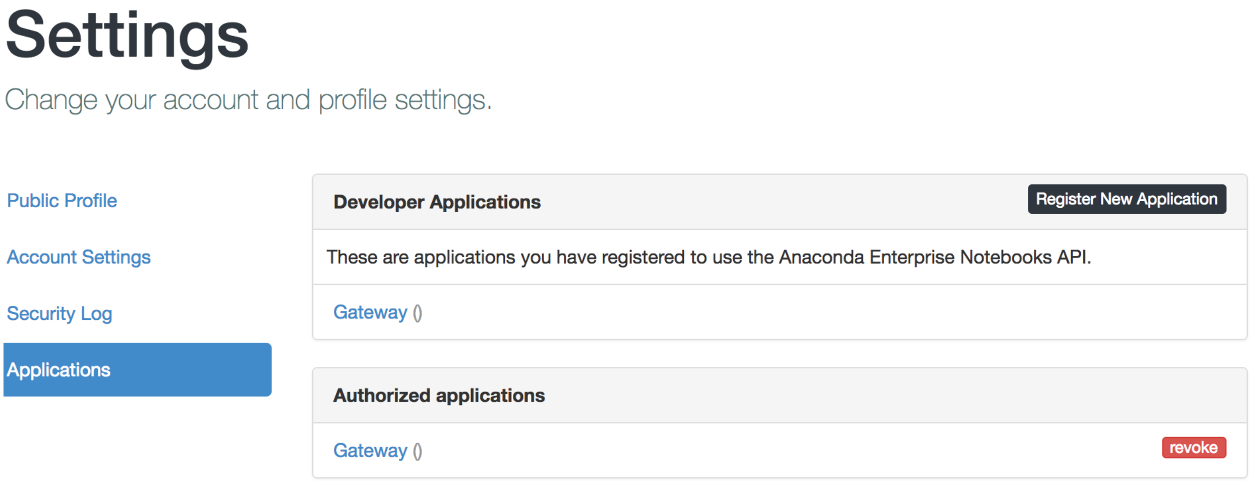
👁	aen_admin	oauth.authenticate	2017-09-25 04:52:06.713000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:58.954000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:58.720000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:58.490000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:58.259000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:58.033000+00:00
👁	aen_admin	oauth.authenticate	2017-09-25 04:51:57.802000+00:00

2. For more information about an operation, click the Eye icon to the left of the the operation name.

Registering an application

If you want to create an application for AEN or have already done so, you must register your application.

1. In the left navigation pane, click the **Applications** tab.



Settings

Change your account and profile settings.

Developer Applications

These are applications you have registered to use the Anaconda Enterprise Notebooks API.

Gateway ()

Authorized applications

Gateway ()

2. Click the Register New Application button to open a form for registering your application.

Advanced tasks

Advanced tasks are best-suited for users who are comfortable working in a Terminal.

Working with environments

AEN runs on conda, a package management system and environment management system for installing multiple versions of software packages and their dependencies and switching easily between them.

A conda environment usually includes 1 version of Python or R language and some packages.

The ability to have a custom project environment is one of the most powerful features of AEN. Your project environment is integrated so that all of your project applications recognize it and all of your team members have access to it.

This section contains information about:

- *Creating a default conda environment using the Jupyter Notebook application*
- *Creating a default conda environment using the Jupyter Notebook application*
- *Using your conda environment in a notebook*
- *Customizing your conda environment*
- *Installing a conda package using Terminal*
- *Installing a conda package using Notebook*
- *Uninstalling a conda package*

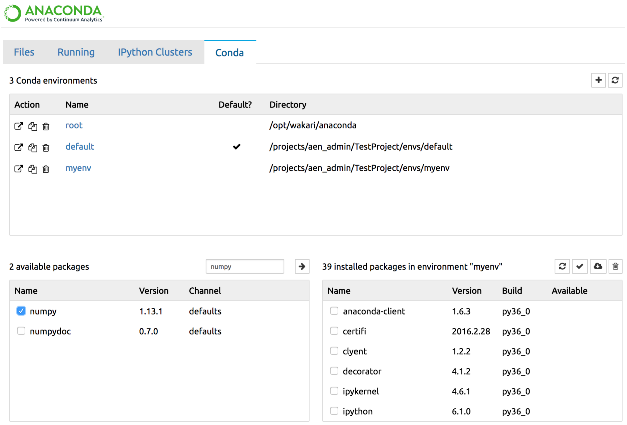
NOTE: This conda environments guide is specific to AEN. For full conda documentation—including cheat sheets, a conda test drive, and command reference—see the conda documentation.

Creating a default conda environment using the Jupyter Notebook application

You can create, activate, and install packages and deactivate environments from within the Notebook menu bar.

To install from the Notebook menu bar:

1. Click the **Conda** tab and select the plus sign icon.
2. Search for `numpy` in the package search box.
3. Select numpy from the search results.



3 Conda environments

Action	Name	Default?	Directory
	root		/opt/wakari/anaconda
	default	✓	/projects/aen_admin/TestProject/envs/default
	myenv		/projects/aen_admin/TestProject/envs/myenv

2 available packages

numpy

Name	Version	Channel
☑ numpy	1.13.1	defaults
□ numpydoc	0.7.0	defaults

39 installed packages in environment "myenv"

Name	Version	Build	Available
□ anaconda-client	1.6.3	py36_0	
□ certifi	2016.2.28	py36_0	
□ clyent	1.2.2	py36_0	
□ decorator	4.1.2	py36_0	
□ ipykernel	4.6.1	py36_0	
□ ipython	6.1.0	py36_0	

1. Click the Install button.

The environment is added to the project's `env` directory.

Creating a default conda environment using Terminal

In AEN, all new environments created with conda automatically include Python, Jupyter Notebooks and pip. You can specify any other packages you want included in your new environment.

TIP: By default, conda creates a new environment in your project's `env` directory—so that all team members have access to the environment. For information about limiting your team member's read, write or execute permissions, see *Workbench*.

To create a new environment within your AEN account, run the command `conda` in a *Terminal* application.

EXAMPLE: To create a new environment named `WeatherModel` that contains Python, NumPy, pip and Jupyter Notebooks in your project's `env` directory:

1. Log in to AEN.
2. Open a project.
3. On the project home page, click the Terminal application icon to open a Terminal.
4. Create the environment:

   ```
   conda create -n WeatherModel numpy
   ```

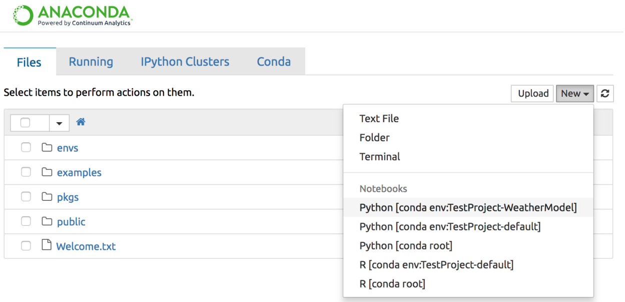
   TIP: Python, pip and Jupyter Notebooks are automatically installed in each new environment. You only need to specify NumPy in this command.

5. Make the new environment your default:

   ```
   source activate WeatherModel
   ```

6. To use your new environment with Jupyter Notebooks, open the Notebook application.
7. Click the New button to open a new notebook. In the drop-down menu under Notebooks, the environment you just created is displayed.
8. To activate that environment, select it.

   The environment is added to the project's `env` directory.



Select items to perform actions on them.

NOTE: You can deactivate the new environment when you are finished with your notebook by opening the Terminal application and running the command `source deactivate`.

Using your conda environment in a notebook

Whether you have created an environment using conda in a terminal, or from the **Conda** tab in a notebook, you can use the conda environment in the same way.

When working in a notebook, to select the environment you have created and want to use with that notebook, in the **Kernel** menu, select Change Kernel.

EXAMPLE: If you have an environment named `my_env` in a project named `test1` that includes NumPy and SciPy and you want to use that environment in your notebook, in the **Kernel** menu, select Python [conda env:test1-my_env].

The notebook code will run in that environment and can import NumPy and SciPy functions.

Customizing your conda environment

If you need a Python package that AEN doesn't include by default, you can install additional packages into your AEN environment.

TIP: You cannot install packages into the default Anaconda environment. You must create your own environment before installing a new package into that environment.

AEN is built on Anaconda, so you can install additional Python packages using conda or pip—both of which are included with Anaconda.

Installing a conda package using Terminal

To install a conda package using the Terminal application:

1. Create and activate the environment using the steps in *Creating a default conda environment using the Jupyter Notebook application*.
2. In your Terminal application, run the command `conda install <packagename>`.

   NOTE: Be sure to specify the Python version you want when using conda to create the environment, or it will use the same version as root.

   EXAMPLE:

   ```
   conda create -n mypy3 python=3 numpy scipy
   ```

   A conda environment named mypy3, running on Python 3 and containing NumPy and SciPy is created. All subsequent packages added to this environment will be the Python 3 compatible versions.

Installing a conda package using Notebook

You can also install the package within your notebook without using the terminal app:

1. From the Notebook application, click the **Conda** tab.
2. Select the environment you wish to use.

3. Search for the package you want to add.
4. Click the Install button.

Uninstalling a conda package

To uninstall a package using this method, run the command `conda remove <packagename>`.

NOTE: Replace `<packagename>` with the name of the package you are uninstalling.

Using visualization packages

AEN supports multiple visualization packages for Python and R language.

For Python, the default environment has *Matplotlib* and *Bokeh* installed.

For R language, the default environment has *r-ggplot2* and r-bokeh installed.

Matplotlib

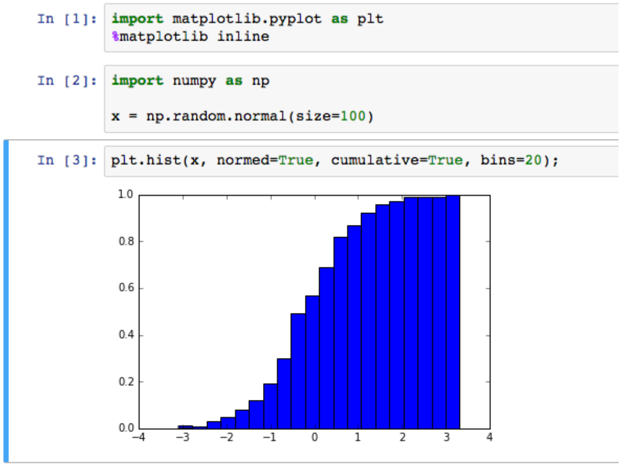
Matplotlib is a Python 2D and 3D plotting and visualization library that produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms.

To display Matplotlib figures in the output cells of a notebook running the default environment, run:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

Any Matplotlib figures in the notebook are displayed in it’s output cells.

EXAMPLE: The following screenshot is of a cumulative density function (CDF) plot using values taken from a normal distribution:



```
In [1]: import matplotlib.pyplot as plt
        %matplotlib inline

In [2]: import numpy as np

        x = np.random.normal(size=100)

In [3]: plt.hist(x, normed=True, cumulative=True, bins=20);
```

For more information, including a gallery, examples, documentation and a list of plotting commands, see the Matplotlib website.

Bokeh

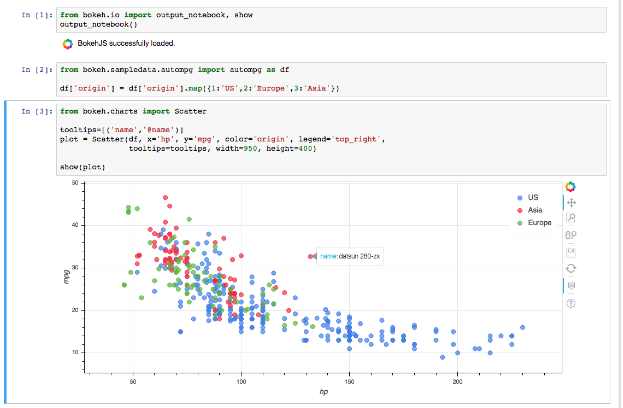
Bokeh is an interactive visualization library that targets modern web browsers to provide elegant, concise construction of novel graphics.

To display Bokeh figures in the output cells of a notebook running the default environment, run:

```
from bokeh.io import output_notebook, show
output_notebook()
```

Any Bokeh figures in the notebook are displayed in its output cells.

The following screenshot is of a scatter plot of miles-per-gallon vs. horsepower for 392 automobiles using the `autompg` sample dataset:



```
In [1]: from bokeh.io import output_notebook, show
        output_notebook()
```

BokehJS successfully loaded.

```
In [2]: from bokeh.sampledata.autompg import autompg as df

        df['origin'] = df['origin'].map({1:'US',2:'Europe',3:'Asia'})
```

```
In [3]: from bokeh.charts import Scatter

        tooltips=[('name','@name')]
        plot = Scatter(df, x='hp', y='mpg', color='origin', legend='top_right',
                       tooltips=tooltips, width=950, height=400)

        show(plot)
```

ggplot2

Ggplot2 is a plotting system for R language which is based on the grammar of graphics. Ggplot2 tries to take only the good parts of base and lattice graphics and none of the bad parts.

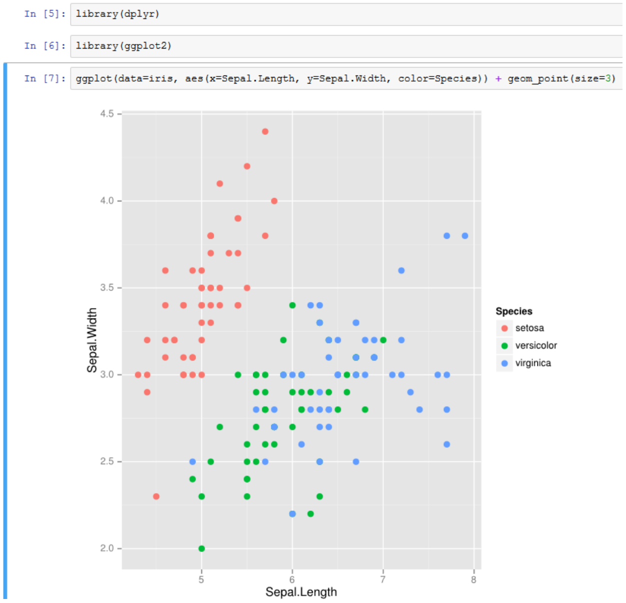
To use ggplot2 with AEN:

1. Open a new Notebook using the R kernel.
2. Load the ggplot2 library with the following code:

   ```
   library(ggplot2)
   ```

The ggplot2 library is loaded and ready for use in AEN.

The following screenshot is of a scatter plot of sepal width vs sepal length using the `iris` dataset provided by the `dplyr` library:



```
In [5]: library(dplyr)

In [6]: library(ggplot2)

In [7]: ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) + geom_point(size=3)
```

Using environment variables

Some Python packages depend on environment variables for correct operation.

EXAMPLE: Theano requires that the directory containing the CUDA compiler is included in the $PATH environment variable in order for GPU acceleration to be enabled.

To change environment variables for all AEN applications, modify the project runtime configuration file `.projectrc`. For more information, see *Using Compute Resource Configuration*.

`.projectrc` sets several AEN internal environment variables, sets up the project environment and can set additional user environment variables for that project. This file is sourced when a user opens any AEN application—including Jupyter Notebook—and Jupyter kernels will be able to read the included environment variables.

Cheat sheet

See the `Anaconda Enterprise Notebooks cheat sheet` PDF (232 KB) for a single-page summary of the most important information about using AEN.

Troubleshooting

This troubleshooting guide provides you with ways to deal with issues that may occur with your AEN installation.

AEN application not working properly

An AEN application is not working as expected.

Cause

There are several reasons an application may not work as expected.

Solution

Most AEN application issues can be resolved by following these steps:

1. Refresh the page.
2. If the issue is not resolved, close and open the application.
3. If the issue is not resolved, *stop and restart your project*.
4. If the issue is not resolved, check that you are using the latest version of your web browser—Chrome, Safari, Edge, or Firefox.
5. Log out of AEN.
6. Restart your browser, and log back in.

If you continue to have issues, then please contact your administrator or enterprise support representative.

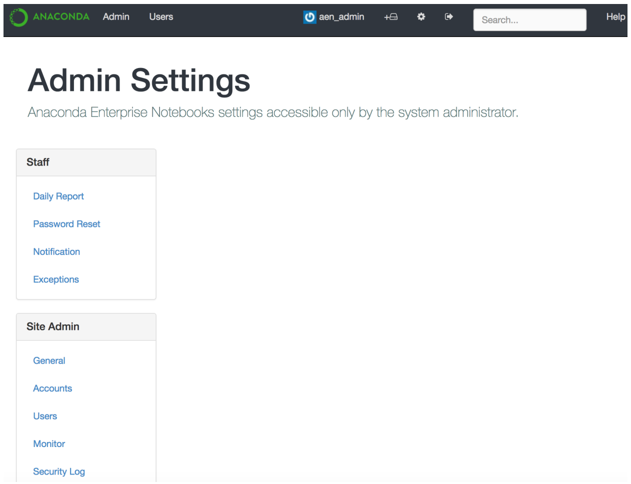
Admin guide

This administrator guide provides information about the administration of an AEN installation.

Most AEN system management is done from the administrative user interface (admin UI). Some advanced tasks are done *using the command line*.

Any AEN user account can be *upgraded to an administrator account* to have both user and administrator privileges.

Administrators see two additional links in the AEN Navigation bar—Admin and Users:



Admin Settings

Anaconda Enterprise Notebooks settings accessible only by the system administrator.

All of the other navigation bar items are the same as for a user account.

Concepts

System overview

The Anaconda Enterprise Notebooks platform consists of 3 main service groups: AEN server, AEN gateway and AEN compute, which are called "nodes":

- *Server node*—The administrative front-end to the system where users login, user accounts are stored, and administrators manage the system.
- *Gateway node(s)*—A reverse proxy that authenticates users and directs them to the proper compute node for their project. Users will not notice this node after installation as it automatically routes them.
- *Compute nodes*—Where projects are stored and run.

These services can be run on a single machine or distributed across multiple servers.

Organizationally, each AEN installation has exactly 1 server instance and 1 or more gateway instances. Each compute node can only be connected to a single gateway. The collection of compute nodes served by a single gateway is called a **data center**. You can add data centers to the AEN installation at any time.

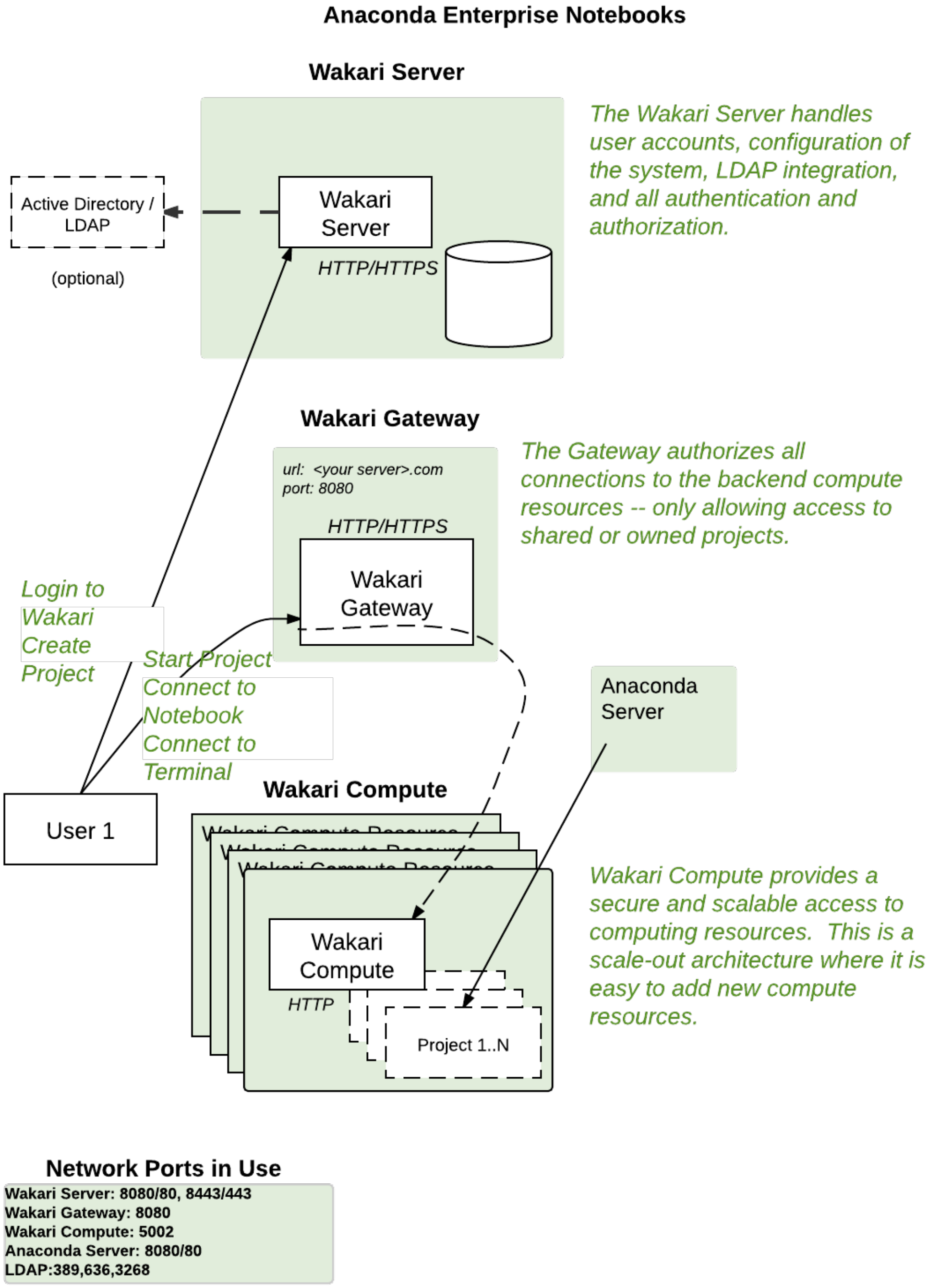
EXAMPLE: An AEN deployment with 2 data centers, where 1 gateway has a cluster of 20 physical computers, and the second gateway has 30 virtual machines, must have the following services installed and running:

- 1 AEN server instance
- 2 AEN gateway instances
- 50 AEN compute instances (20 + 30)

Nodes must be configured and maintained separately.

Server node

The server node controls login, accounts, admin, project creation and management as well as interfacing with the database. It is the main entry point to AEN for all users. The server node handles project setup and ensures that users



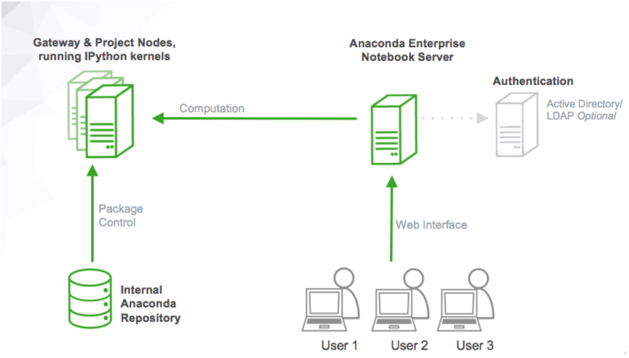
The Wakari Server handles user accounts, configuration of the system, LDAP integration, and all authentication and authorization.

The Gateway authorizes all connections to the backend compute resources -- only allowing access to shared or owned projects.

Wakari Compute provides a secure and scalable access to computing resources. This is a scale-out architecture where it is easy to add new compute resources.

Network Ports in Use

Wakari Server: 8080/80, 8443/443
Wakari Gateway: 8080
Wakari Compute: 5002
Anaconda Server: 8080/80
LDAP:389,636,3268


Gateway & Project Nodes, running IPython kernels
Anaconda Enterprise Notebook Server
Authentication
Active Directory/ LDAP Optional
Computation
Package Control
Web Interface
Internal Anaconda Repository
User 1
User 2
User 3

are sent to the correct project data center.

Since AEN is web-based, it uses the standard HTTP port 80 or HTTPS port 443 on the server.

AEN uses MongoDB for its internal data persistency. It is typically run on the same host as the server but can also be *installed* on a separate host.

Server nodes use NGINX to handle the user-facing AEN web interface. NGINX acts as a request proxy for the actual server web-process which runs on a high numbered port that only listens on localhost. NGINX is also responsible for static content.

Server is installed in the `/opt/wakari/wakari-server` directory.

Server processes

When you *view the status of server processes*, you may see the processes explained below.

supervisord	details
description	Manage `wakari-worker`, multiple processes of `wk-server`.
user	`wakari`
configuration	`/opt/wakari/wakari-server/etc/supervisord.conf`
log	`/opt/wakari/wakari-server/var/log/supervisord.log`
control	`service wakari-server`
ports	none

wk-server	details
description	Handles user interaction and passing jobs on to the wakari gateway. Access to it is managed by NGINX.
user	`wakari`
command	`/opt/wakari/wakari-server/bin/wk-server`
configuration	`/opt/wakari/wakari-server/etc/wakari/`
control	`service wakari-server`
logs	`/opt/wakari/wakari-server/var/log/wakari/server.log`
ports	Not used in versions after 4.1.2 *

* AEN 4.1.2 and earlier use port 5000. This port is used only on localhost. Later versions of AEN use Unix sockets instead. The Unix socket path is: `unix:/opt/wakari/wakari-server/var/run/wakari-server.sock`

wakari-worker	details
description	Asynchronously executes tasks from `wk-server`.
user	`wakari`
logs	`/opt/wakari/wakari-server/var/log/wakari/worker.log`
control	`service wakari-server`

nginx	details
description	Serves static files and acts as proxy for all other requests passed to wk-server process. *
user	nginx
configuration	`/etc/nginx/nginx.conf` `/opt/wakari/wakari-server/etc/conf.d/www.enterprise.conf`
logs	`/var/log/nginx/woc.log` `/var/log/nginx/woc-error.log`
control	`service nginx status`
port	80

* In AEN 4.1.2 and earlier the wk-server process runs on port 5000 on localhost only. In later versions of AEN the wk-server process uses the Unix socket path `unix:/opt/wakari/wakari-server/var/run/wakari-server.sock`.

NGINX runs at least two processes:

- Master process running as root user.
- Worker processes running as nginx user.

Gateway node

The gateway node serves as an access point for a given group of compute nodes. It acts as a proxy service and manages the authorization and mapping of URLs and ports to services that are running on those nodes. The gateway nodes provide a consistent uniform interface for the user.

NOTE: The gateway may also be referred to as a data center because it serves as the proxy for a collection of compute nodes.

You can put a gateway in each data center in a tiered scale-out fashion.

AEN gateway is installed in the `/opt/wakari/wakari-gateway` directory.

Gateway processes

When you *view the status of server processes*, you may see the processes explained below.

supervisord	details
description	Manages the `wk-gateway` process.
user	`wakari`
configuration	`/opt/wakari/wakari-gateway/etc/supervisord.conf`
log	`/opt/wakari/wakari-gateway/var/log/supervisord.log`
control	`service wakari-gateway`
ports	none

wakari-gateway	details
description	Passes requests from the AEN Server to the Compute nodes.
user	`wakari`
configuration	`/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json`
logs	`/opt/wakari/wakari-gateway/var/log/wakari/gateway.application.log` `/opt/wakari/wakari-gateway/var/log/wakari/gateway.log`
working dir	/ (root)
port	8089 (webcache)

Compute node(s)

Compute nodes are where applications such as Jupyter Notebook and Workbench actually run. They are also the hosts that a user sees when using the Terminal app or when using SSH to access a node. Compute nodes contain all user-visible programs.

Compute nodes only need to communicate with a gateway, so they can be completely isolated by a firewall.

Each project is associated with one or more compute nodes that are part of a single data center.

AEN compute nodes are installed in the `/opt/wakari/wakari-compute` directory.

Each compute node in the AEN system requires a compute launcher service to mediate access to the server and gateway.

Compute processes

When you *view the status of server processes*, you may see the processes explained below.

supervisord	details
description	Manages the `wk-compute` process.
user	`wakari`
configuration	`/opt/wakari/wakari-compute/etc/supervisord.conf`
log	`/opt/wakari/wakari-compute/var/log/supervisord.log`
control	`service wakari-compute`
working dir	`/opt/wakari/wakari-compute/etc`
ports	none

wk-compute	details
description	Launches compute processes.
user	`wakari`
configuration	`/opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json` `/opt/wakari/wakari-compute/etc/wakari/scripts/config.json`
logs	`/opt/wakari/wakari-compute/var/log/wakari/compute-launcher.application.log` `/opt/wakari/wakari-compute/var/log/wakari/compute-launcher.log`
working dir	/ (root)
control	`service wakari-compute`
port	5002 (rfe)

Wk-compute loads each of the following configuration files, in this order:

- `/etc/wakari/config.json`.
- `/etc/wakari/compute-launcher-config.json`.
- `./compute-launcher-config.json`.
- Any configuration file specified by the `-c` option.

If an option is specified in multiple files, the last one encountered takes precedence.

Supervisor and supervisord

AEN uses a process control system called "Supervisor" to run its services. Supervisor is run by the AEN Service Account user, usually wakari or aen_admin.

The Supervisor daemon process is called "supervisord". It runs in the background and should rarely need to be restarted.

Service Account

AEN must be installed and executed by a Linux account called the AEN Service Account. The username of the AEN Service Account is called the AEN Functional ID (NFI). The AEN Service Account is created during AEN installation—if it does not exist—and is used to run all AEN services.

The default NFI username is `wakari`. Another popular choice is `aen_admin`.

WARNING: The Service Account should only be used for administrative tasks, and should not be used for operating AEN the way an ordinary user would. If the Service Account creates or starts projects, the permissions on the AEN package cache will be reset to match the Service Account, which will interfere with the normal operation of AEN for all other users.

Anaconda environments

Each project has an associated conda environment containing the packages needed for that project. When a project is first started, AEN clones a default environment with the name "default" into the project directory.

Each release of AEN 4 includes specific tested versions of conda and the conda packages included with AEN. These tested conda packages include Python, R, and other packages, and these tested conda packages include all of the packages in Anaconda.

If you upgrade or install different versions of conda or different versions of any of these conda packages, the new packages will not have been tested as part of the AEN 4 release.

These different packages will usually work, especially if they are newer versions, but they are not tested or guaranteed to work, and in some cases they may break product functionality.

You can use a new conda environment to test a new version of a package before installing it in your existing environments.

If using conda to change the version of a package breaks product functionality, you can use conda to change the version of the package back to the version known to work.

For more information about environments, see *Working with environments*.

Projects and permissions

AEN users interact with the system predominantly through *projects*.

Projects are associated with a single data center within the AEN environment. The team of users includes one owner, which is the user that created the project.

Projects live in the `projectRoot` folder on the compute node—by default, `/projects`.

The project directory is created the first time a project is started. The `start-project` script clones it from `/opt/wakari/wakari-compute/lib/node_modules/wakari-compute-launcher/skeleton`.

Project directory permissions are:

```
owner: rwx, user who created the project
group: rwx, group of the owner
other: --x, to allow access to the Public folder
ACL: rwx for any other team members
```

Files and subdirectories within the project directory have the same permissions as the project directory, except:

- The public folder and everything in it are open to anyone.
- Any files hardlinked into the root anaconda environment—`/opt/wakari/anaconda`—are owned by the root or wakari users.

Project file and directory permissions are maintained by the `start-project` script. All files and directories in the project will have their permissions set when the project is started, except for files owned by root or the AEN_SRVC_ACCT user—by default, wakari or aen_admin.

The permissions set for files owned by root or the AEN_SRVC_ACCT user are not changed to avoid changing the permissions settings of any linked files in the `/opt/wakari/anaconda` directory.

CAUTION: Do not start a project as the AEN_SRVC_ACCT user. The permissions system does not correctly manage project files owned by this user.

Installation

Installation requirements

Hardware requirements

AEN server—At least:

- 2+GB RAM.
- 2+CPU cores.
- 20GB storage.

AEN gateway—At least:

- 2 GB RAM.
- 2 CPU cores.

AEN compute (N-machines)—Configured to meet the needs of the projects. At least:

- 2GB RAM.
- 2 CPU cores.
- 20 GB.

NOTE: We recommend putting `/opt/wakari` and `/projects` on the same filesystem. If the project and conda env directories are on separate filesystems then more disk space will be required on compute nodes and performance will be worse.

Software requirements

- RHEL/CentOS on all nodes. Versions from 6.5 through 7.4 are supported. Other operating systems are supported. However, this document assumes RHEL or CentOS.
- Linux home directories—Jupyter looks in $HOME for profiles and extensions.
- Ability to install in AEN directory `/opt/wakari` with at least 10 GB of storage.
- Ability to install in Projects directory `/projects` with at least 20 GB of storage. Size depends on number and size of projects.

NOTE: To install AEN in a different location see *Installing AEN in a custom location*.

Linux system accounts

Some Linux system accounts (UIDs) are added to the system during installation.

If your organization requires special actions, the following list is available:

- mongod (RHEL) or mongodb (Ubuntu/Debian)—created by the RPM or deb package.
- elasticsearch—created by RPM or deb package.
- nginx—created by RPM or deb package.
- AEN_SRVC_ACCT—created during installation of AEN, and defaults to wakari.
- ANON_USER—An account such as "public" or "anonymous" on the compute node.

  NOTE: If ANON_USER is not found, AEN_SRVC_ACCT will attempt to create it. If it fails, the project(s) will fail to start.
- ACL directories need the filesystem mounted with Posix ACL support (Posix.1e).

  NOTE: You can verify ACL from the command line by running `mount` and `tune2fs -l /path/to/filesystem | grep options`.

Software prerequisites

- AEN server:
  - Mongo—Equal to or higher than version 2.6.8 and lower than version 3.0.
  - NGINX—Equal to or higher than version 1.6.2.
  - Elasticsearch—Equal to or higher than version 1.7.2.
  - Oracle JRE version 7 or 8.
  - bzip2.
- AEN Gateway:
  - bzip2.
- AEN compute:
  - git
  - bzip2
  - bash or zsh
  - X Window System

    NOTE: If you don't want to install the whole X Window System, you must install the following packages to have R plotting support:

    ```
    sudo yum install -y libXrender libXext libXdmcp libSM libICE libXt \
    dejavu-sans-fonts dejavu-serif-fonts dejavu-fonts-common \
    fontpackages-filesystem
    ```

Security requirements

- Root or sudo access.
- File permissions: `umask 0022` is required during the installation.
- SELinux in permissive or disabled mode.

Edit the following file using either root or sudo access:

```
/etc/sysconfig/selinux
```

Edit the following:

```
# This file controls the state of SELinux on the system.
    # SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.

SELINUX=enforcing

# SELINUXTYPE= can take one of these two values:
    #     targeted - Targeted processes are protected,
    #     mls - Multi Level Security protection.

SELINUXTYPE=targeted
```

NOTE: You must reboot for the changes to take effect.

Verify changes with `getenforce`.

Network requirements

TCP Ports:

Direction	Type	Default Port	Protocol	Optional	Configurable	Comments
Inbound	TCP	80	HTTP or HTTPS	No	Yes	Server
Inbound	TCP	8089	HTTP or HTTPS	No	Yes	Gateway
Inbound	TCP	5002	HTTP	No	Yes	Compute

Other requirements

As long as the above requirements are met, there are no additional dependencies for AEN.

See also system requirements for Anaconda Repository and Anaconda Scale.

What’s next

Prepare for installation.

Preparing for installation

Downloading AEN installers

Download the installers and copy them to the corresponding servers.

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.
↪rackcdn.com"
curl -O $RPM_CDN/aen-server-4.2.2-Linux-x86_64.sh
curl -O $RPM_CDN/aen-gateway-4.2.2-Linux-x86_64.sh
curl -O $RPM_CDN/aen-compute-4.2.2-Linux-x86_64.sh
```

NOTE: The current $RPM_CDN server will be confirmed in an email provided by your sales rep.

NOTE: These instructions use *curl* or *wget* to download packages, but you may use other means to move the necessary files into the installation directory.

Gathering IP addresses or FQDNs

AEN is very sensitive to the IP address or domain name used to connect to the server and gateway nodes. If users will be using the domain name, you should install the nodes using the domain name instead of the IP addresses. The authentication system requires the proper hostnames when authenticating users between the services.

Print this page and fill in the domain names or IP addresses of the nodes below and record the user name and auto-generated password for the administrative user account in the box below after installing the AEN server node:

Node | Name or IP address	Port Number	Username | Password	
AEN server |			
AEN gateway |			
AEN compute |			

NOTE: The values of these IP entries or DNS entries are referred to as `<AEN_SERVER_IP>` or `<AEN_SERVER_FQDN>`, particularly in examples of shell commands. Consider actually assigning those values to environment variables with similar names.

Set up variables

Certain variables need to have values assigned to them before you start the installation.

AEN server address

To define an environment variable for the AEN server address—FQDN or IP:

```
export AEN_SERVER=<AEN_SERVER_IP>   # <from table above>
```

NOTE: The address—FQDN or IP—specified for the AEN server must be resolvable by your intended AEN users' web clients.

To verify your hostname, run `echo $AEN_SERVER`.

AEN functional ID

AEN must be installed and executed by a Linux account called the AEN Service Account. The username of the AEN Service Account is called the AEN Functional ID (NFI). The AEN Service Account is created during AEN installation—if it does not exist—and is used to run all AEN services.

The default NFI username is wakari. Another popular choice is aen_admin.

To set the environment variable AEN_SRVC_ACCT to `wakari` or your chosen name before installation, run `export AEN_SRVC_ACCT="aen_admin"`.

This name is now the username of the AEN Service Account and of the AEN administrator account.

When upgrading AEN, set the NFI to the NFI of the current installation.

WARNING: The Service Account should only be used for administrative tasks, and should not be used for operating AEN the way an ordinary user would. If the Service Account creates or starts projects, the permissions on the AEN package cache will be reset to match the Service Account, which will interfere with the normal operation of AEN for all other users.

AEN functional group

The AEN Functional Group (NFG) may be given any name. Most often, it is set to `aen_admin` or `wakari`. This Linux group includes the AEN service account, so all files and directories that have the owner NFI also have the group NFG.

When upgrading AEN, set the NFG to the NFG of the current installation.

To set the NFG before installation, run:

```
export AEN_SRVC_GRP="<NFG>"
```

NOTE: Replace `<NFG>` with your NFG name.

AEN install sudo command

During AEN installation the installers perform various operations that require root level privileges. By default, the installers use the sudo command to perform these operations.

Before installation, set the AEN_SUDO_CMD_INSTALL environment variable to perform root level operations. You can also set it to no command at all if the user running the installer(s) has root privileges and the sudo command is not needed or is not available.

EXAMPLES:

```
export AEN_SUDO_CMD_INSTALL=""
export AEN_SUDO_CMD_INSTALL="sudo2"
```

AEN sudo command

By default the AEN services uses `sudo -u` to perform operations on behalf of other users—including `mkdir`, `chmod`, `cp` and `mv`.

To override the default sudo command when sudo is not available on the system, before installing, set the AEN_SUDO_CMD environment variable.

AEN must have the ability to perform operations on behalf of other users. Therefore, this environment variable cannot be set to an empty string or to `null`.

CAUTION: Any command that replaces AEN_SUDO_CMD must support the `-u` command line parameter—similarly to the sudo command.

EXAMPLE:

```
export AEN_SUDO_CMD="sudo2"
```

The optional environmental variable AEN_SUDO_SH is another way to customize AEN sudo operations. When AEN executes any sudo command, it will include the value of AEN_SUDO_SH, if it is set.

EXAMPLE: If your username is "jsmith" and the values are set as:

```
AEN_SUDO_CMD=sudo
OWNER=jsmith
AEN_SUDO_SH=sudologger
PROJECT_HOME=/projects/jsmith/myproj
```

Then AEN will resolve:

```
$AEN_SUDO_CMD -u ${OWNER} $AEN_SUDO_SH rm -rf $PROJECT_HOME
```

As:

```
sudo -u jsmith sudologger rm -rf /projects/jsmith/myproj
```

In this case the sudologger utility could be a pass-through utility that logs all sudo usage and then executes the remaining parameters.

Post-installation Sudo configuration

While root/sudo privileges are required during installation, root/sudo privileges are not required during normal operations after install, if user accounts are managed outside the software. However root/sudo privileges are required to start the services, thus in the service config files there may still need to be an AEN_SUDO_CMD entry.

For more information, see *Configuring sudo customizations*.

AEN remote database settings

By default AEN server uses a local database. To override the default database location, see *Install AEN connected to a remote Mongo DB instance*.

What's next

Install the AEN server.

Installing the AEN server

The AEN server is the administrative front end to the system. This is where users log in to the system, where user accounts are stored, and where admins can manage the system.

Server is installed in the `/opt/wakari/wakari-server` directory.

Installing the bzip2 package

Be sure you have the *bzip2* package installed. If this package is not installed on your system, install it:

```
sudo yum install bzip2
```

Downloading prerequisite RPMs

To install AEN on a CentOS 6 server:

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.
↪rackcdn.com"
curl -O $RPM_CDN/nginx-1.6.2-1.el6.ngx.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-tools-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-shell-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-server-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-mongos-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/elasticsearch-1.7.2.noarch.rpm
curl -O $RPM_CDN/jre-8u65-linux-x64.rpm
```

To install AEN on a CentOS 7 server:

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.
↪rackcdn.com"
curl -O $RPM_CDN/nginx-1.10.2-1.el7.ngx.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-tools-2.6.12-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-shell-2.6.12-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-server-2.6.12-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-mongos-2.6.12-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-2.6.12-1.x86_64.rpm
curl -O $RPM_CDN/jre-8u112-linux-x64.rpm
curl -O $RPM_CDN/elasticsearch-1.7.6.noarch.rpm
```

Installing prerequisite RPMs

Run:

```
sudo yum install -y *.rpm
sudo service mongod start
sudo chkconfig --add elasticsearch
```

Setting variables and changing permissions

Run:

```
export AEN_SERVER=<FQDN HOSTNAME OR IP ADDRESS> # Use the real FQDN
chmod a+x aen-*.sh                # Set installer to be executable
```

NOTE: Change <FQDN HOSTNAME OR IP ADDRESS> to the actual fully qualified domain hostname or IP address.

Running the AEN server installer

Run:

```
sudo -E ./aen-server-4.2.2-Linux-x86_64.sh -w $AEN_SERVER
<license text>
...
...

PREFIX=/opt/wakari/wakari-server
Logging to /tmp/wakari_server.log
Checking server name
Ready for pre-install steps
Installing miniconda
...
...
Checking server name
Loading config from /opt/wakari/wakari-server/etc/wakari/config.json
Loading config from /opt/wakari/wakari-server/etc/wakari/wk-server-config.json


===================================

Created password '<RANDOM_PASSWORD>' for user 'aen_admin'
```

(continues on next page)

(continued from previous page)

```
===================================


Starting Wakari daemons...
installation finished.
```

After successfully completing the installation script, the installer creates the administrator account—AEN_SRVC_ACCT user—and assigns it a password.

EXAMPLE:

```
Created password '<RANDOM_PASSWORD>' for user 'aen_admin'
```

TIP: Record this password. It will be needed in the following steps. It is also available in the installation log file `/tmp/wakari_server.log`.

Starting NGINX and Elasticsearch

When SELinux is enabled, it blocks NGINX from connecting to the socket created by Gunicorn. If you have SELinux enabled, run these commands to correct these permissions and allow connections between NGINX and Gunicorn:

```
sudo semanage fcontext -a -t httpd_var_run_t "/opt/wakari/wakari-server/var/run/
↪wakari-server.sock"
sudo restorecon -r /opt/wakari/wakari-server/var/run
```

To start NGINX and Elasticsearch to read the new config file:

```
sudo service nginx start
sudo service elasticsearch start
```

TIP: If the AEN web page shows an NGINX 404 error, restart NGINX:

```
sudo nginx -s stop
sudo nginx
```

Testing AEN server installation

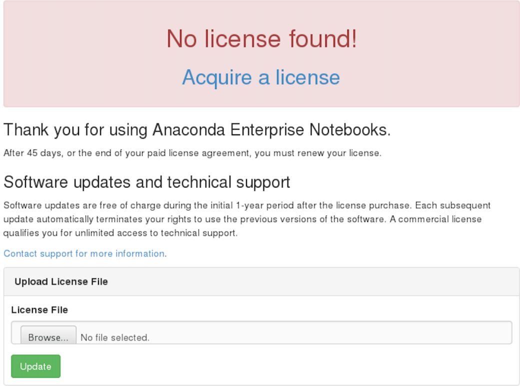
Visit http://\protect\T1\textdollarAEN_SERVER.

The License expired page is displayed.

Updating your license

From the License expired page, follow the onscreen instructions to upload your license file.

After your license is submitted, you will see this page:



No license found!

Acquire a license

Thank you for using Anaconda Enterprise Notebooks.

After 45 days, or the end of your paid license agreement, you must renew your license.

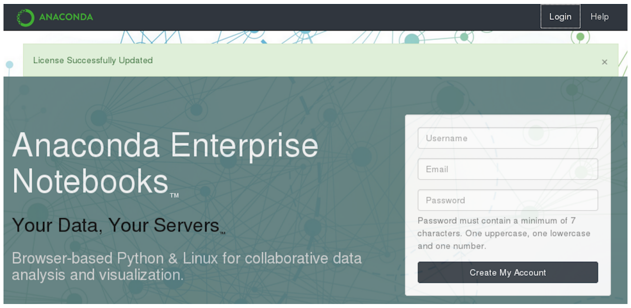
Software updates and technical support

Software updates are free of charge during the initial 1-year period after the license purchase. Each subsequent update automatically terminates your rights to use the previous versions of the software. A commercial license qualifies you for unlimited access to technical support.

Contact support for more information.

Upload License File

License File



License Successfully Updated

Anaconda Enterprise Notebooks™

Your Data, Your Servers™

Browser-based Python & Linux for collaborative data analysis and visualization.

Password must contain a minimum of 7 characters. One uppercase, one lowercase and one number.

What's next

Install the AEN gateway.

Installing the AEN gateway

The gateway is a reverse proxy that authenticates users and automatically directs them to the proper AEN compute node for their project. Users will not notice this node as it automatically routes them.

Gateway is installed in the `/opt/wakari/wakari-gateway` directory.

Setting variables and changing permissions

Run:

```
export AEN_SERVER=<FQDN HOSTNAME OR IP ADDRESS> # Use the real FQDN
export AEN_GATEWAY_PORT=8089
```

(continues on next page)

(continued from previous page)

```
export AEN_GATEWAY=<FQDN HOSTNAME OR IP ADDRESS>  # will be needed shortly
chmod a+x aen-*.sh                # Set installer to be executable
```

NOTE: Change <FQDN HOSTNAME OR IP ADDRESS> to the actual fully qualified domain hostname or IP address.

NOTE: You must perform the entire procedure before closing the terminal to ensure the variable export persists. If the terminal is closed before successful installation, export the variables to continue with the installation.

Running the AEN gateway installer

Run:

```
sudo -E ./aen-gateway-4.2.2-Linux-x86_64.sh -w $AEN_SERVER
<license text>
...
...

PREFIX=/opt/wakari/wakari-gateway
Logging to /tmp/wakari_gateway.log
...
...
Checking server name
Please restart the Gateway after running the following command
to connect this Gateway to the AEN Server
...
```

Registering your gateway

The gateway needs to register with the AEN server.

This needs to be authenticated, so the NFI user’s credentials created during the AEN server install must be used.

To write the configuration file `/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json`, run the following as sudo or root:

```
sudo /opt/wakari/wakari-gateway/bin/wk-gateway-configure \
--server http://$AEN_SERVER --host $AEN_GATEWAY \
--port $AEN_GATEWAY_PORT --name Gateway --protocol http \
--summary Gateway --username $AEN_SRVC_ACCT \
--password '<NFI USER PASSWORD>'
```

NOTE: replace `<NFI USER PASSWORD>` with the password of the NFI user that was generated during *server installation*.

Setting permissions

Run:

```
sudo chown $AEN_SRVC_ACCT /opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json
```

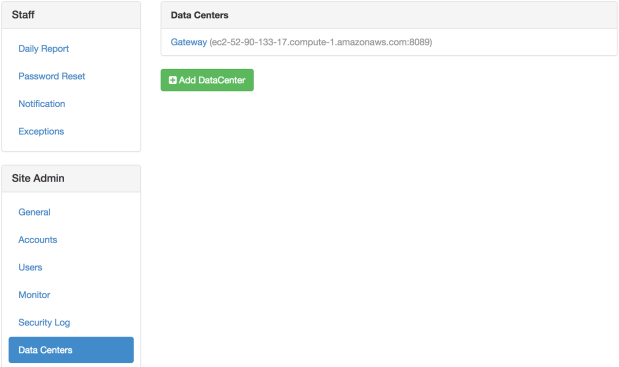
Starting the gateway

Run:

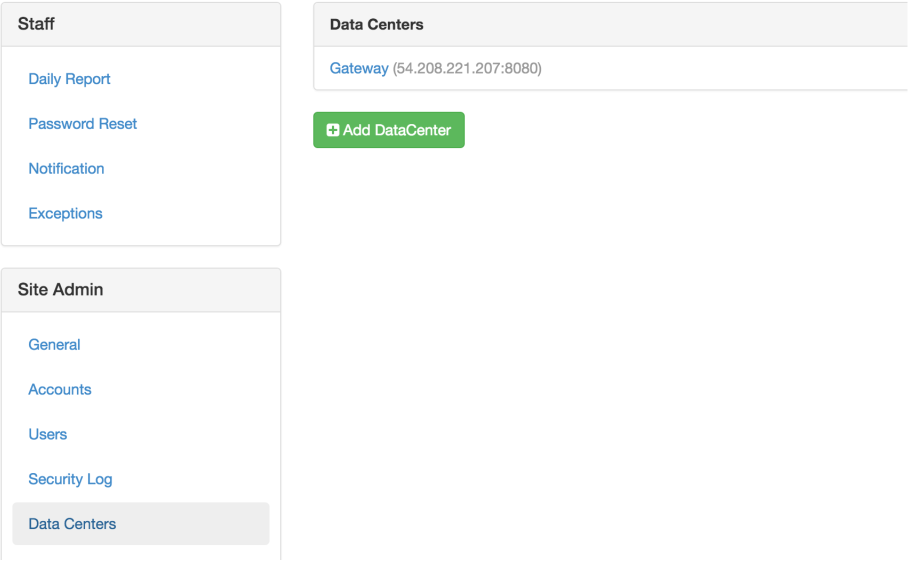
```
sudo service wakari-gateway start
```

Verifying your gateway registration

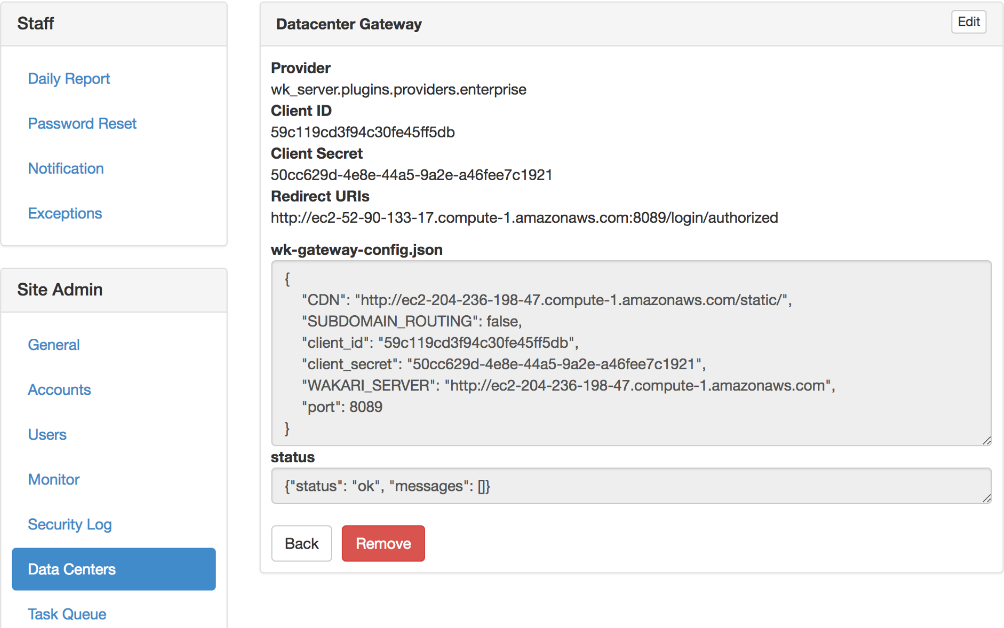
1. Log into the AEN server using the Chrome or Firefox browser and the AEN_SRVC_ACCT user.
2. In the AEN navigation bar, click Admin to open the Admin Settings page.
3. In the **Site Admin** menu, select Data Centers:



4. Click your data center:



5. Verify that your data center is registered and the status is `{"status": "ok", "messages": []}`:



Provider
wk_server.plugins.providers.enterprise
Client ID
59c119cd3f94c30fe45ff5db
Client Secret
50cc629d-4e8e-44a5-9a2e-a46fee7c1921
Redirect URIs
http://ec2-52-90-133-17.compute-1.amazonaws.com:8089/login/authorized

wk-gateway-config.json

```
{
    "CDN": "http://ec2-204-236-198-47.compute-1.amazonaws.com/static/",
    "SUBDOMAIN_ROUTING": false,
    "client_id": "59c119cd3f94c30fe45ff5db",
    "client_secret": "50cc629d-4e8e-44a5-9a2e-a46fee7c1921",
    "WAKARI_SERVER": "http://ec2-204-236-198-47.compute-1.amazonaws.com",
    "port": 8089
}
```

status

```
{"status": "ok", "messages": []}
```

What's next

Install the AEN compute node(s).

Installing the AEN compute node(s)

Compute nodes are where projects are stored and run.

Adding multiple AEN compute machines allows you to scale-out horizontally to increase capacity. Projects can be created on individual compute nodes to spread the load.

Repeat this procedure on each compute machine.

Setting variables and changing permissions

Run:

```
export AEN_SERVER=<FQDN HOSTNAME OR IP ADDRESS> # Use the real FQDN
chmod a+x aen-*.sh                # Set installer to be executable
```

NOTE: Change <FQDN HOSTNAME OR IP ADDRESS> to the actual fully qualified domain hostname or IP address.

NOTE: You must perform the entire procedure before closing the terminal to ensure the variable export persists.

Running the AEN compute installer

Run:

```
sudo -E ./aen-compute-4.2.2-Linux-x86_64.sh -w $AEN_SERVER
...
...
PREFIX=/opt/wakari/wakari-compute
Logging to /tmp/wakari_compute.log
Checking server name
...
...
Initial clone of root environment...
Starting Wakari daemons...
installation finished.
Do you wish the installer to prepend the wakari-compute install location
to PATH in your /root/.bashrc ? [yes|no]
[no] >>> yes
```

Restart the AEN Server

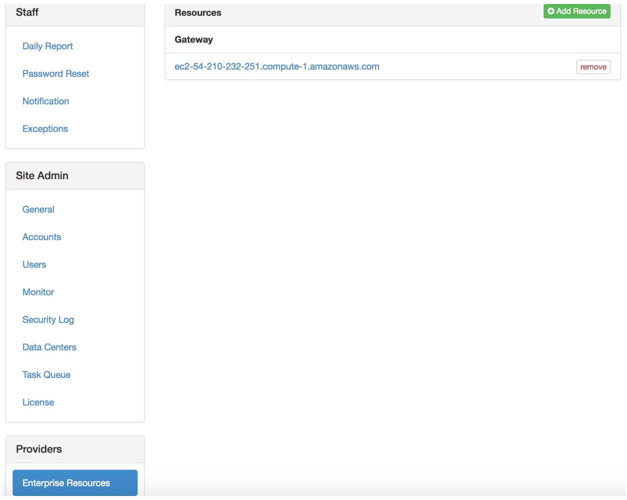
Once configured, restart the AEN server:

```
sudo service wakari-server restart
```

Configuring your compute node(s)

Once installed, you must configure the compute launcher on your server:

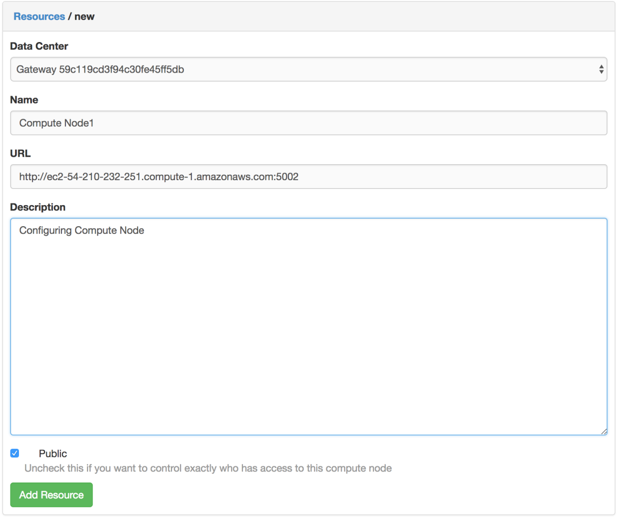
1. In your browser, go to your AEN server.
2. Log in as the AEN_SRVC_ACCT user.
3. In the AEN navigation bar, click Admin to open the Admin Settings page.
4. In the **Providers** menu, select Enterprise Resources:
5. Click the Add Resource button to open the new resource form.
6. Select the data center to associate this compute node with.



Resources

Gateway

ec2-54-210-232-251.compute-1.amazonaws.com



Data Center: Gateway 59c119cd3f94c30fe45ff5db

Name: Compute Node1

URL: http://ec2-54-210-232-251.compute-1.amazonaws.com:5002

Description: Configuring Compute Node

☑ Public

Uncheck this if you want to control exactly who has access to this compute node

7. In the URL box, type: `http://$AEN_COMPUTE:5002`.

   NOTE: If the compute launcher is located on the same box as the gateway, we recommended that you type `http://localhost:5002` instead.

8. Type a Name and Description for the compute node.
9. Click the Add Resource button to save the changes.

Your AEN compute node is configured.

What's next

Configure conda to use your local on-site AEN repository.

Configuring conda to use your local on-site AEN repository

You can configure AEN to use a local on-site Anaconda Repository server instead of Anaconda.org.

To configure AEN to use a local on-site Repository, you must:

1. *Edit condarc on the compute node*.
2. *Configure the Anaconda client*.

Editing condarc on the compute node

NOTE: If there are channels that you haven't mirrored, you must remove them from the configuration.

Edit the file `.condarc` to match the following:

```
#/opt/wakari/anaconda/.condarc
channels:
    - defaults

create_default_packages:
    - anaconda-client
    - ipykernel

# Default channels is needed for when users override the system .condarc
# with ~/.condarc.  This ensures that "defaults" maps to your Anaconda Repository and␣
↪not
# repo.continuum.io
default_channels:
    - http://<your Anaconda Repository name>:8080/conda/anaconda
    - http://<your Anaconda Repository name>:8080/conda/wakari
    - http://<your Anaconda Repository name>:8080/conda/r-channel

# Note:  You must add the "conda" subdirectory to the end
channel_alias: http://<your Anaconda Repository name:8080/conda
```

NOTE: Replace `<your Anaconda Repository name>` with the actual name or IP address of your local Anaconda Repository installation.

Configuring the Anaconda client

Anaconda client lets users work with Repository from the command-line—including searching for packages, logging in, uploading packages, and more.

To set the default configuration of anaconda-client for all users on your compute node:

```
sudo /opt/wakari/anaconda/bin/anaconda config --set url http://<your Anaconda␣
↪Repository>:8080/api -s
```

NOTE: Sudo is required because the configuration file is written to the root file system: `/etc/xdg/binstar/config.yaml`.

NOTE: Replace `<your Anaconda Repository>` with the actual name or IP address of your local Anaconda Repository installation.

What's next

Review the *optional configuration* tasks to see if any apply to your system.

Optional configuration

Using configuration files

The default locations for each component's configuration files are:

- Server—`/opt/wakari/wakari-server/etc/wakari/config.json`.
- Gateway—`/opt/wakari/wakari-gateway/etc/wakari/config.json`.
- Compute—`/opt/wakari/wakari-compute/etc/wakari/config.json`.

Additionally, service-specific configuration files may also be present in the following locations:

- Server—`/opt/wakari/wakari-server/etc/wakari/wk-server-config.json`.
- Gateway—`/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json`.
- Compute—`/opt/wakari/wakari-compute/etc/wakari/wk-compute-config.json`.

Each service loads each of the configuration files in the following order and updates the AEN configuration at each step:

1. `/etc/wakari/config.json`.
2. `/etc/wakari/wk-gateway-config.json`.
3. `/opt/wakari/wakari-SERVICE/etc/wakari/config.json`.
4. `/opt/wakari/wakari-SERVICE/etc/wakari/wk-SERVICE-config.json`.
5. `./config.json`.
6. `./wk-gateway-config.json`.

AEN configuration keys

The following is a list of AEN supported configuration keys:

Table 1: Server Configuration Keys

Key	Default	Description
CDN	$WAKARI_SERVER/static/	The location of static assets.
MONGO_DB	wakari	The name of the AEN database in mongodb.
MONGO_URL	mongodb://localhost/	The URL of your AEN server's mongodb instance. Format: mongodb://<username>:<password>@<host>:<port>/
WAKARI_SERVER		The URL of this AEN server.
DEFAULT_PRIVACY	public	The default project privacy setting—can be either public or private.
SESSION_COOKIE_NAME	wakari.enterprise.session	The Cookie name used to maintain Anaconda Enterprise Notebooks Enterprise login sessions.
PERMANENT_SESSION	True`	Sets cookie session to permanent. This will keep the session open after the browser is closed. The session will still expire after the number of minutes set in the SESSION_LIFETIME key.
SESSION_LIFETIME	120	Time in minutes until the session expires. The counter resets with each request.
USE_SES	false	Sets whether AEN will use Amazon SES to send emails.
SMTP		Sets the SMTP email settings.
- host		A SMTP subkey—the SMTP mail server hostname.
- user		SMTP subkey—the username for SMTP server authentication.
- password		SMTP subkey—the password for SMTP server authentication.
- from_addr		SMTP subkey—the From address for emails sent through SMTP.
verify_gateway_certificate	true	A boolean setting that indicates whether your AEN server should verify the gateway SSL certificate.
accounts	wk_server.plugins.accounts.cloud	The account provider class. For LDAP, this should be set to wk_server.plugins.accounts.ldap_accounts.
uniqueEmail	true	A boolean setting that indicates whether unique user email addresses are required. See *note below* about updating the database when setting uniqueEmail.
has_internet	true	Boolean for retrieving the avatar from the gravatar URL. If false a local default is used instead.
LDAP	389	LDAP configurations.
- SERVER		LDAP subkey—A list of LDAP servers. At least one server name must be listed. The primary server should be listed first. All secondary or fail-over servers should be listed after the primary.
- PORT	389	LDAP subkey—The LDAP port on the LDAP server.
- AUTH_TYPE		LDAP subkey—LDAP Authentication types. simple—no encryption not secure.``TLS``–encrypted secure requires the TLS_CERT to be set.
- TLS_CERT		LDAP subkey—the full path to the TLS certificate file. The certificate file must also be provided by the Enterprise.
- BASEDN		LDAP subkey—the LDAP Base DN value.
- OU		LDAP subkey—a list of Organizational Units. Some Enterprises group users by OUs in their LDAP server records. AEN will loop over the list of OUs when authenticating a user. The OU value is a list of lists to support multiple OUs where each OU is a single name or a hierarchy of names.
ANON_USER	anonymous	Username—such as public or anonymous– assigned users who are not logged in to access projects. To disable public access use the special value disabled. For more information, s *Configuring sudo customizations*.
SEARCH_ENABLED	true	Boolean indicating whether ElasticSearch is enabled
SEARCH_SERVER	'localhost:9200'	IP address or domain name and port of ElasticSearch server

NOTE: If you set uniqueEmail to false, you must drop the existing index in the database. EXAMPLE: If the index name is email_1, run db.users.dropIndex("email_1").

Table 2: Gateway Configuration Keys

Key	Default	Description
WAKARI_SERVER		The URL of the AEN WAKARI_SERVER.
port	8089	The Port number used by the gateway application. Must be a non-privileged port (>= 1024).
client_id		The client ID assigned to this gateway by the server during wk-gateway-configure.
client_secret		The Client secret assigned to this gateway by the server during wk-gateway-configure.
httpTimeout	600	Timeout in seconds. The default is 10 minutes to allow project creation.
logLevel	info	Log verbosity. One of: 'error' 'warn' 'info' 'debug'.
https		Enable SSL encryption. For more information, see *Configuring SSL*.
- key		A https subkey–Path to gateway key.
- cert		A https subkey–Path to gateway cert.
- ca		A https subkey–Required if cert was signed by a private root CA or signed by an intermediate authority. It must contain separate values for the paths to the CA root, any intermediates and the certificate for the Server.
- passhphrase		A https subkey–Passphrase required to decrypt SSL certs.

Table 3: Compute Node Configuration Keys

Key	Default	Description
`WAKARI_SERVER`		The URL of the AEN `WAKARI_SERVER`.
`MANAGE_ACCOUNTS`	`true`	A boolean setting that indicates whether AEN should manage system user accounts. Set to false for LDAP installations.
`identicalGID`	`false`	**To make the AEN compute service create groups with the same uid. Set to `tru`** `/projects` folder resides on an NFSv3 volume. For more information, see *Group and user permissions for NFS*.
`port`	2227	The port number used by the compute-launcher application. Note that individual applications use dynamic ports.
`projectRoot`	`/projects`	The location of project file storage.
`logLevel`	`info`	Log verbosity. One of: 'error' 'warn' 'info' 'debug'
`logMaxSize`	10000000	Max size in bytes of the logfile. Default is 10 MB. If the size is exceeded then a new file is created and a counter will become a suffix of the log file.
`logMaxFiles`	30	Limit the number of files created when the size of the logfile is exceeded
`appIdleTime`	172800000 (48 hours)	The amount of idle time before applications will be auto-terminated (in msec).
`idleCheckInterval`	3600000 (1 hour)	The frequency of idle checks.
`numericUsernames`	`false`	A boolean setting that indicates whether numeric usernames are permitted.
`httpTimeout`	600	The time before a timeout—in seconds. The default is 10 minutes—600 seconds—to allow time for project creation.
`ANON_USER`	`anonymous`	Username such as `public` or `anonymous` for users who are not logged in to access projects. To disable public access use the special value `disabled`. For more information, see *Configuring sudo customizations*.
`projDirsAsHome`	`false`	A boolean setting. When `false` AEN apps use `/home/<username>` as `HOME`. When `true` AEN apps use `/projects/<username>` as `HOME`.

Table 4: Server Internal Configuration Keys - Do not change

Key	Default	Description
`PROVIDERS`	`["wk_server.plugins.providers.enterprise"]`	A list of compute provider classes.
`MONGO_ACTION _LOG_SIZE`	262144000	The size of the Mongo action log in bytes.
`SITE_ADMINS`		A list of site administrator email addresses—used for crash notifications and LDAP password reset requests.
`FROM _EMAIL_ADDR`		The From address for notification emails sent by AEN.
`uniqueUserName`	`true`	A boolean setting that indicates whether unique usernames are required.

Table 5: Gateway Internal Configuration Keys - Do not change

Key	Default	Description
CDN	$WAKARI_SERVER/static/	The location of static assets.
SUBDOMAIN_ROUTING	false	A boolean that indicates whether subdomains are being used.
refreshTokenExpiration	600000	Idle time in milliseconds before the Gateway session expires.

Table 6: Compute Node Internal Configuration Keys - Do not change

Key	Default	Description
CDN	$WAKARI_SERVER/static/	The location of static assets.
USE_SES	false	Sets whether AEN will use Amazon SES to send emails.
multiUser	true	A boolean that indicates whether multi-user support is enabled.
multiProject	true	A boolean that indicates whether multi-project support is enabled.
ANACONDA_ROOT	/opt/wakari/anaconda	The location of your Anaconda installation.
appLogs	/opt/wakari/wakari-compute/var/log/wakari/compute-launcher-apps	The directory where application logs are stored.
appPIDs	/opt/wakari/wakari-compute/var/run/compute-launcher-apps	The directory where application PID files are stored.
applicationLog	/opt/wakari/wakari-compute/var/log/wakari/compute-launcher.application.log	The path to the compute launcher log.
accessLog	opt/wakari/wakari-compute/var/log/wakari/compute-launcher.access.log	Path to compute launcher access log

Checking configuration file syntax

To verify that the configuration file contains valid JSON, run:

```
root@server  # python -m json.tool /opt/wakari/wakari-server/etc/wakari/*.json
root@gateway # python -m json.tool /opt/wakari/wakari-gateway/etc/wakari/*.json
root@compute # python -m json.tool /opt/wakari/wakari-compute/etc/wakari/*.json
```

If the file is correct, the contents are displayed.

If there is a syntax error in the file, a “No JSON object could be decoded” message is displayed instead.

To fix any errors, edit the configuration file and verify that it contains the correct JSON syntax.

Increasing HTTP timeout between gateway and compute nodes

The default HTTP timeout is 600 seconds (10 minutes).

This setting works for HTTP timeout only, not HTTPS.

To modify the HTTP timeout setting:

1. Open the `/opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json` file and modify the httpTimeout key:

   ```
   "httpTimeout": 600
   ```

2. Update the gateway node by modifying the httpTimeout key in the `/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json` file to match the above settings.

3. Restart the AEN compute service:

   ```
   sudo service wakari-compute restart
   ```

Installing AEN in a custom location

To install AEN in a custom location:

1. Make the custom install folder owned by `$AEN_SRVC_ACCT`. EXAMPLE: `/data/aen/`.
2. Make a symlink from `/opt/wakari` to `/data/aen`.
3. Run the installers.
4. Move the folder from `/projects` to your chosen custom location. EXAMPLE: `/data/aen/projects`.
5. Make a symlink from `/projects` to `/data/aen/projects`.

NOTE: We recommend putting `/opt/wakari` and `/projects` on the same filesystem. If the project and conda environment directories are on separate filesystems then more disk space will be required on compute nodes and performance will be worse.

Changing where projects are stored

NOTE: We recommend putting `/opt/wakari` and `/projects` on the same filesystem. If the project and conda env directories are on separate filesystems then more disk space will be required on compute nodes and performance will be worse.

To make aen-compute service use a different directory than `/projects` to store your AEN projects:

1. Modify the `/opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json` file:

   ```
   "projectRoot" : "/nfs/storage/services/wakari/projects",
   ```

   NOTE: The directory `/nfs/storage/services/wakari/projects` specified as `projectRoot` must already exist for this command to resolve properly.

2. Restart the AEN compute service:

```
sudo service wakari-compute restart
```

Group and user permissions for NFS

To install AEN with multiple compute nodes and a /projects folder on an NFSv3 volume, manually pre-create both the anonymous user and the $AEN_SRVC_ACCOUNT user on all nodes. Each of these users must have the same user identity number (UID) and group identity number (GID) on all nodes.

By default AEN creates local users with a different GID on each node. To make the AEN compute service create groups with the same GID:

1. In the /opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json file, change the identicalGID key value to true:

   ```
   , "identicalGID": true
   ```

   If you don't see the identicalGID key, add it.

   NOTE: You must add the comma at the beginning of the line. If you add this line as the last key, you must remove any comma at the end of the line.

2. Restart the AEN compute service:

   ```
   sudo service wakari-compute restart
   ```

Using numeric usernames

1. In the /opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json file, change the numericUsernames key value to true.

   ```
   , "numericUsernames": true
   ```

   If you don't see the numericUsernames key, add it.

   NOTE: You must add the comma at the beginning of the line. If you add this line as the last key, you must remove any comma at the end of the line.

2. Restart the AEN compute service:

   ```
   sudo service wakari-compute restart
   ```

Using project directories as home directories

The projDirsAsHome option changes the AEN home directories from the standard /home/<username> location to the project directories and the location /projects/<username>/<project_name>/<username>/. This ensures that AEN and AEN apps will not be affected by configuration files in a user's home directory, such as .bashrc or configuration files in subdirectories such as .ipython and .jupyter.

Package cache locations

AEN version 4.1.3 stores the cache of packages in /home/<username>, while AEN versions 4.2.0 and higher store the cache of packages in /projects/<username>/<project_name>/<username>/. By moving the package cache to the same filesystem as the project, AEN versions 4.2.0 and higher can use hardlinks and save disk space and time when creating or cloning environments.

These package cache locations are not affected by the `projDirsAsHome` option.

After upgrading from AEN 4.1.3 to AEN 4.2.0 or higher, existing projects will still use the package cache in `/home/<username>`. Do not remove this cache, or the existing projects will break.

When users create new projects or install packages, the newly installed packages will use the new cache location.

If you wish to remove the older package cache in `/home/<username>`:

- Upgrade AEN to 4.2.0 or higher.
- Use `conda remove` to remove every non-default package in every project.
- Use `conda install` to replace them. The replaced packages will link to the new package cache in `/projects/<username>/<project_name>/<username>/`.
- You can now safely remove the older package cache.

Enabling projDirsAsHome

NOTE: The `projDirsAsHome` option should be enabled immediately after performing the installation process and before any users have logged in to AEN. This ensures that users will not have home directories in different places due to some creating their home directories when the option was disabled and others creating their home directories when the option was enabled.

1. In the `/opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json` file, add the `projDirsAsHome` key value and set it to `true`.

   ```
   , "projDirsAsHome": true
   ```

   NOTE: You must add the comma at the beginning of the line. If you add this line as the last key, you must remove any comma at the end of the line.

2. Restart the AEN compute service:

   ```
   sudo service wakari-compute restart
   ```

Setting up a default project environment

AEN includes a full installation of the Anaconda Python distribution—along with several additional packages—located within the root conda environment in `/opt/wakari/anaconda`.

The first time any new AEN project is started, this default project environment is cloned into the new project's workspace.

To configure a different set of packages than the default:

1. Create a new conda environment in the `/opt/wakari/anaconda/envs/default` directory.

   EXAMPLE: Using a Python 3.4 base environment, run:

   ```
   sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
       create -p /opt/wakari/anaconda/envs/default python=3.4
   ```

2. Use `conda` to install any additional packages into the environment.
3. After the environment is created, clone it to ensure that it works correctly:

```
sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
    create -p /opt/wakari/testenv --clone /opt/wakari/anaconda/envs/default
sudo -u $AEN_SRVC_ACCT rm -rf /opt/wakari/testenv
```

For more information and examples about creating a default project environment with Microsoft R Open (MRO), see *Using MRO in AEN*.

Converting an existing project

1. Run the following command to clone the environment:

   ```
   sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
       create -n /projects/owner/project/envs/<ENV_NAME> \
       --clone /opt/wakari/anaconda/envs/default
   ```

   NOTE: Replace `/projects/owner/project/envs/<ENV_NAME>` with the path to the new environment you would like to create within the project.

2. Open the *Compute Resource Configuration application* for your project and set the project environment path there as well.

Using MRO in AEN

In AEN 4.2.2 and higher, you can choose to create environments with the Microsoft R Open (MRO) interpreter by installing the `mro-base` package, or create environments with the R interpreter by installing the `r-base` package. Unless you request a change, conda will continue to use the existing interpreter in each environment. In AEN `r-base` is the default.

EXAMPLE: To create a custom environment called `mro_env` with MRO and R Essentials:

```
.. code-block:: bash

    sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
        create -c https://repo.continuum.io/pkgs/main \
        -n mro_env r-essentials
```

NOTE: Conda 4.4 and higher include the `main` channel by default. Earlier versions of conda do not.

Making a default project environment with MRO

You can also create an environment with MRO and make this the default AEN project environment.

The first time a new project is started, the default project environment is cloned into the new project's workspace.

1. Create a new conda environment in the `/opt/wakari/anaconda/envs/default` directory.

   The command is similar to the one used in the previous example to create a custom environment.

   ```
   sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
       create -c https://repo.continuum.io/pkgs/main \
       -p /opt/wakari/anaconda/envs/default r-essentials
   ```

2. Use `conda` to install any additional packages into the environment.

3. After the environment is created, clone it to check that it works correctly, and then clean up the clone.

```
sudo -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda \
    create -p /opt/wakari/testenv --clone /opt/wakari/anaconda/envs/default
sudo -u $AEN_SRVC_ACCT rm -rf /opt/wakari/testenv
```

NOTE: To convert existing projects, see *Converting an existing project*.

Install AEN connected to a remote Mongo DB instance

To install AEN with a remote database:

1. Connect to the Mongodb instance and create the user for AEN:

```
> user = { user: "<username>",
  pwd: "<super-secure-password>",
  roles: [
    { role: "dbOwner", db: "<db_name>" },
    { role: "dbOwner", db: "<db_name>_mq" }
  ]
}
> db.createUser(user)
Successfully added user: { ... }
```

2. Before installing AEN-server export the database URL and name:

```
$ export MONGO_URL="mongodb://<username>:<password>@<host>:<port>/"
$ export MONGO_DB="<database_name>"
```

3. Continue the installation process: *Install the AEN server*.

Migrate from local to remote MongoDB

To configure your remote database to work with an already installed AEN server:

1. Stop the server, gateway and compute nodes:

```
sudo service wakari-server stop
sudo service wakari-gateway stop
sudo service wakari-compute stop
```

2. Open the `/opt/wakari/wakari-server/etc/wakari/config.json` file and create the MONGO_URL key. For the value parameter, add the database information.

   The final file should read:

```
{
  "MONGO_URL": "mongodb://MONGO-USER:MONGO-PASSWORD@MONGO-URL:MONGO-PORT",
  "MONGO_DB": "MONGO-DB-NAME",
  "WAKARI_SERVER": "http://YOUR-IP",
  "USE_SES": false,
  "CDN": "http://YOUR-IP/static/",
  "ANON_USER": "anonymous"
}
```

   For more information about configuration keys, see *Using configuration files*.

3. Migrate the data from the former database into the new one. For more information, see the MongoDB documentation website.
4. After migration, restart the nodes:

   ```
   sudo service wakari-server start
   sudo service wakari-gateway start
   sudo service wakari-compute start
   ```

Running SELinux in enforcing mode

To run SELinux in Enforcing mode, a few ports must be set up using the `semanage port` command.

The `semange` command relies on `policycoreutils-python`. To install `policycoreutils-python`, if needed, run:

```
sudo yum -y install policycoreutils-python
```

Enable ports 9200 and 9300 for Elasticsearch:

```
sudo semanage port -a -t http_port_t -p tcp 9200
sudo semanage port -a -t http_port_t -p tcp 9300
```

Changing server hostnames

It is possible to change the domain names (hostnames) of the various AEN nodes by updating the configuration files.

NOTE: After the configuration files are updated, the associated nodes need to be restarted.

To edit the information for all of the data centers that you are changing the base domain name for:

1. Go to the Site Admin section of the Admin Settings page.
2. In the Data Centers section, click the Edit button.
3. Make any necessary updates.

   NOTE: This must include the service port if it is different from the default—80 for HTTP and 443 for HTTPS.
4. In the Enterprise Resources sub-section of the Providers section, edit each compute node that has a changed domain name.

   NOTE: These URLs should include the protocol, hostname and port.

Authenticating with LDAP

Anaconda Enterprise Notebooks performs local authentication against accounts in the AEN database by default.

To configure AEN to authenticate against accounts in an LDAP (Lightweight Directory Access Protocol) server, follow the instructions below.

Installing OpenLDAP libraries

The system needs OpenLDAP libraries to be installed and accessible by AEN. AEN uses the OpenLDAP libraries to establish an LDAP connection to your LDAP servers.

To install OpenLDAP on CentOS or Redhat:

```
sudo yum install openldap
```

To install OpenLDAP on Ubuntu or Debian, follow the official OpenLDAP installation instructions.

Configuring OpenLDAP

1. Open the `/opt/wakari/wakari-server/etc/wakari/wk-server-config.json` file.
2. Add the following LDAP settings:

```
{
    "accounts":"wk_server.plugins.accounts.ldap2",
    "LDAP" : {
        "URI": "ldap://openldap.EXAMPLE.COM",
        "BIND_DN": "cn=Bob Jones,ou=Users,DC=EXAMPLE,DC=COM",
        "BIND_AUTH": "secretpass",
        "USER_SEARCH": {"base": "DC=EXAMPLE,DC=COM",
                        "filter": "(| (& (ou=Payroll)
                                         (uid=%(username)s))
                                      (& (ou=Facilities)
                                         (uid=%(username)s)))"
                        },
        "KEY_MAP": {"email": "mail",
                    "name": "cn"
        }
    }
}
```

   - URI—The IP address or hostname of your OpenLDAP server. For SSL/TLS, use the `ldaps://` prefix and specify a TLS_CACERT as described in the SSL/TLS configuration section below.
   - BIND_DN—The full directory path of the user you want AEN server to bind as.
   - BIND_AUTH—The password of the BIND_DN user.
   - USER_SEARCH:
     - base—The level at which you want to start the search.
     - filter—The default is to search for the sAMAccountName attribute, and use its value for the AEN server username field.
   - KEY_MAP—Maps user attributes in AEN server to LDAP user attributes.

     EXAMPLE: The `mail` attribute in LDAP maps to the `email` attribute in AEN server.

3. Restart AEN server to load new settings.
4. Log in with the admin account. This creates the admin user in the local database.
5. As soon as LDAP is installed, LDAP authentication takes over, so you need to add your admin account again:

```
/opt/wakari/wakari-server/bin/wk-server-admin superuser --add "jsmith"
```

Configuring Active Directory

Microsoft Active Directory is a server program that provides directory services and uses the open industry standard Lightweight Directory Access Protocol (LDAP).

To enable Active Directory support:

1. Open the `/opt/wakari/wakari-server/etc/wakari/wk-server-config.json` file.
2. Add the following LDAP settings:

```
{
    "accounts":"wk_server.plugins.accounts.ldap2",
    "LDAP" : {
        "URI": "ldap://<ad.EXAMPLE.COM>",
        "BIND_DN": "CN=Bind User,CN=Users,DC=EXAMPLE,DC=COM",
        "BIND_AUTH": "secretpass",
        "USER_SEARCH": {"base": "CN=Users,DC=EXAMPLE,DC=COM",
                        "filter": "sAMAccountName=%(username)s"
        },
        "KEY_MAP": {"email": "mail",
                    "name": "cn"
        }
    }
}
```

   - URI—The IP address or hostname of your Active Directory server. Replace `<ad.EXAMPLE.COM>` with the actual URI. For SSL/TLS, use the `ldaps://` prefix and specify a TLS_CACERT as described in the SSL/TLS configuration section below.
   - BIND_DN—The full directory path of the user you want AEN server to bind as.
   - BIND_AUTH—The password of the BIND_DN user.
   - USER_SEARCH:
     - base—the level at which you want to start the search.
     - filter—default is to search for the sAMAccountName attribute, and use its value for the AEN server `username` field.
   - KEY_MAP—Maps user attributes in AEN server to LDAP user attributes.

     EXAMPLE: The `mail` attribute in LDAP maps to the `email` attribute in AEN server.

3. Restart AEN server to load new settings.
4. Log in with the admin account. This creates the admin user in the local database.
5. As soon as LDAP is installed, LDAP authentication takes over, so you need to add your admin account again:

```
/opt/wakari/wakari-server/bin/wk-server-admin superuser --add "jsmith"
```

Configuring SSL/TLS

AEN uses system-wide LDAP settings, including SSL/TLS support.

- On Redhat/CentOS systems, these settings are located in the `/etc/openldap/ldap.conf` file.
- On Ubuntu/Debian systems, these settings are located in the `/etc/ldap/ldap.conf` file.

Typically, the only configuration necessary is updating the file to read:

```
TLS_CACERT /path/to/CA.cert
```

NOTE: CA.cert is the Certificate Authority used to sign the LDAP server's SSL certificate. In the case of a self-signed SSL certificate, this is the path to the SSL certificate itself.

Testing LDAP configuration

Test your LDAP configuration using flask-ldap-login-check:

```
/opt/wakari/wakari-server/bin/flask-ldap-login-check \
    wk_server.wsgi:app \
    -u [username] \
    -p [password]
```

NOTE: username is the username of a valid user and password is that user's BIND_AUTH password.

Configuring sudo customizations

If your organization's IT security policy does not allow root access or has restrictions on the use of sudo, after AEN installation, you may customize AEN to meet their requirements.

Your organization may choose to implement any or all of the following:

- *Remove root access* for AEN service account (Note: this restricts AEN from managing user accounts).
- *Configurable sudo command*.
- *Restrict sudo access to all processes*.

These customizations must be done in a terminal window after copying the files to the server node.

Removing all root access from the service account

Because root access is required for useradd, the following process restricts AEN from managing user accounts.

1. Modify the /etc/sudoers.d/wakari_sudo file to read:

   ```
   Defaults:wakari !requiretty, visiblepw
   Runas_Alias    OP = ALL,!root
   wakari ALL=(OP) NOPASSWD: ALL
   ```

   NOTE: If you used a service account name other than wakari, enter that name instead of wakari.

2. Modify the /opt/wakari/wakari-compute/etc/wakari/config.json file to read:

   ```
   "MANAGE_ACCOUNTS": false,
   ```

Using this option means that your IT department must create and manage all user accounts at the OS level.

After an OS-level account exists, you may create on the main AEN web page an AEN account using the same name. The password you choose is not linked in any way to the OS-level password for the account.

Alternatively, you can configure the system to *use LDAP for authenticating users*.

Allowing public users to have access to your AEN projects

A public account is visible to anyone who can access the AEN server. The name of this account can be configured to any name you wish. For example, `public` or `anonymous`. To disable this feature use the special value `disabled`.

1. In the `/opt/wakari/wakari-compute/etc/wakari/wk-compute-launcher-config.json` file, modify the ANON_USER line to read:

   ```
   "ANON_USER": "public"
   ```

2. Restart AEN compute node:

   ```
   sudo service wakari-compute restart
   ```

3. In the `/opt/wakari/wakari-server/etc/wakari/wk-server-config.json` file, modify the ANON_USER line to read:

   ```
   "ANON_USER": "public"
   ```

4. Restart AEN server:

   ```
   sudo service wakari-server restart
   ```

For more information about configuration keys, see *Using configuration files*.

Using a sudo alternative

You can use a sudo alternative as long as it supports the same execution semantics as the original sudo. The alternative must be configured to give the service account permission to run commands on behalf of AEN users.

1. In your terminal window, open the `/opt/wakari/wakari-compute/etc/wakari/config.json` file.
2. Modify the AEN_SUDO_CMD line to read:

   ```
   "AEN_SUDO_CMD": "/path/to/alternative/sudo",
   ```

   NOTE: If the alternate sudo command is available on PATH, then the full path is not required.

Restricting sudo access to a single gatekeeper

By default, sudoers is configured to allow AEN to run any command as a particular user which allows the platform to initiate processes as the logged-in end user. If more restrictive control is required, it should be implemented using a suitable sudoers policy. If that is not possible or practical, it is also possible to route all AEN ID-changing operations through a single gatekeeper.

This gatekeeper wraps the desired executable and provides an alternate way to log, monitor, or control which processes can be initiated by AEN on behalf of a user.

CAUTION: Gatekeeper is a special case configuration and should only be used if required.

To configure an AEN gatekeeper:

1. Modify the `/etc/sudoers.d/wakari_sudo` file to contain:

```
Defaults:wakari !requiretty, visiblepw
Runas_Alias    OP = ALL,!root
wakari ALL=(OP) NOPASSWD: /path/to/gatekeeper
```

2. In the `/opt/wakari/wakari-compute/etc/wakari/config.json` file, modify the AEN_SUDO_SH line to read:

```
"AEN_SUDO_SH": "/path/to/gatekeeper"
```

EXAMPLE: The gatekeeper can be as simple as a script with contents such as:

```
#!/bin/bash
first_cmd=$1
if [ 'bash' == $1 ]; then
    shift
    export HOME=~
    export SHELL=/bin/bash
    export PATH=$PATH:/opt/wakari/anaconda/bin
    bash "$@"
else
    exec $@
fi
```

Configuring SSL

The server node uses NGINX to proxy all incoming http(s) requests to the server running on a local port, and uses NGINX for SSL termination. The default setup uses http—non-SSL—since cert files are required to configure SSL and each enterprise will have their own cert files.

The `www.enterprise.conf` file is the default `nginx.conf` file used for AEN. It is copied to the `/etc/nginx/conf.d` directory during server installation.

NOTE: This section describes setting up SSL after your gateway node has been installed and registered with the server node.

Copying the required files

To configure SSL on AEN, you will need the following files:

- Server certificate and key
- Server CA bundle
- Gateway certificate and key
- Gateway CA bundle

Configure SSL on AEN:

1. Copy the Gateway certificate and key to `/opt/wakari/wakari-gateway/etc/` on the Gateway as `gateway.crt` and `gateway.key`.
2. Copy the Gateway CA bundle to `/opt/wakari/wakari-server/etc/` on the Server.
3. Copy the Server certificate and key to `/etc/nginx` on the Server as `server.crt` and `server.key`.
4. Copy the Server CA bundle to `/opt/wakari/wakari-gateway/etc/` on the Gateway.

If you have a certificate that was signed by a private root CA and/or an intermediate authority:

- The Gateway CA bundle must contain the full chain: root CA, any intermediate authority and the certificate.

```
cat gateway.crt intermediate.crt root.crt >> gatway-crt-int-root.crt
```

- The Server CA bundle must be separated into individual files for the root CA, any intermediate and the certificate.

Configuring SSL on the server node

The `www.enterprise.https.conf` is an NGINX configuration file for SSL. It is set up to use the `server.crt` and `server.key` cert files.

CAUTION: You must change these values to point to the signed cert files for your domain.

NOTE: Self-signed certs or those signed by a private root CA require additional configuration.

Perform the following steps as root:

1. Stop NGINX:

```
service nginx stop
```

2. Move the `/etc/nginx/conf.d/www.enterprise.conf` file to a backup directory.
3. Copy the `/opt/wakari/wakari-server/etc/nginx/conf.d/www.enterprise.https.conf` file to `/etc/nginx/conf.d`.

   NOTE: `/etc/nginx/conf.d` may have `www.enterprise.conf` or `www.enterprise.https.conf` but it may not have both.
4. Edit the `/etc/nginx/conf.d/www.enterprise.https.conf` file and change the `server.crt` and `server.key` values to the names of the real cert and key files if they are different.
5. Restart NGINX by running:

```
service nginx start
```

6. Update the WAKARI_SERVER and CDN settings to use https instead of http in the following configuration files:

```
/opt/wakari/wakari-server/etc/wakari/config.json
/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json
/opt/wakari/wakari-compute/etc/wakari/config.json
```

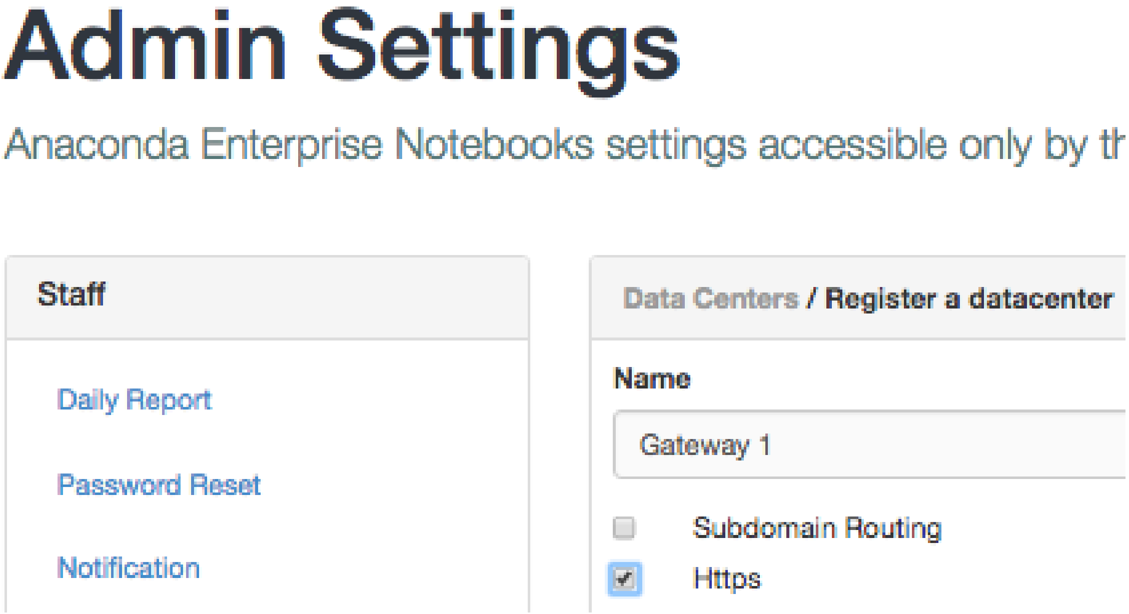
7. Copy the gateway certificate, `gateway.crt` to `/opt/wakari/wakari-server/etc/`.
8. In an editor, open `/opt/wakari/wakari-server/etc/wakari/wk-server-config.json` and add:

```
"verify_gateway_certificate": "/opt/wakari/wakari-server/etc/gateway.crt"
```

9. Restart AEN services on the server by running:

```
service wakari-server restart
```

   NOTE: This step may return an error since the gateway has not yet been configured for SSL.
10. In AEN, verify that the browser uses https. On the Admin Settings page, under Data Centers, click Gateway, then select https:



Admin Settings

Anaconda Enterprise Notebooks settings accessible only by th

Name

Gateway 1

□ Subdomain Routing

☑ Https

Configuring SSL on the gateway

1. For all types of SSL certificates, in `/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json`, add:

```
{
    EXISTING_CONFIGURATION,
    "https": {
        "key": "/opt/wakari/wakari-gateway/etc/gateway.key",
        "cert": "/opt/wakari/wakari-gateway/etc/gateway.crt"
    }
}
```

2. For a server certificate signed by a private root CA or signed by an intermediate authority, add:

```
{
   EXISTING_CONFIGURATION,
   "https": {
       "key": "/opt/wakari/wakari-gateway/etc/gateway.key",
       "cert": "/opt/wakari/wakari-gateway/etc/gateway.crt",
       "ca": ["/opt/wakari/wakari-gateway/etc/server.crt"]
    }
 }
```

NOTE: When the certificate chain has more than one intermediate cert signed by a higher root CA authority, you must manually break up the certs in the chain into individual files, and enumerate them in the `ca` key:

```
{
   EXISTING_CONFIGURATION,
   "https": {
```

(continues on next page)

(continued from previous page)

```
        "key": "/opt/wakari/wakari-gateway/etc/gateway.key",
        "cert": "/opt/wakari/wakari-gateway/etc/gateway.crt",
        "ca": ["/opt/wakari/wakari-gateway/etc/server1.crt",
               "/opt/wakari/wakari-gateway/etc/server2.crt"
               "/opt/wakari/wakari-gateway/etc/server3.crt"]
    }
}
```

3. For a gateway certificate that is encrypted using a passphrase, add:

```
{
   EXISTING_CONFIGURATION,
   "https": {
        "key": "/opt/wakari/wakari-gateway/etc/gateway.key",
        "cert": "/opt/wakari/wakari-gateway/etc/gateway.crt",
        "passphrase": "mysecretpassphrase"
    }
}
```

NOTE: Alternatively, the passphrase can be passed using an environment variable or entered when the wakari-gateway service is manually started.

EXAMPLES:

```
# using an environment variable
AEN_GATEWAY_SSL_PASSPHRASE='mysecretpassphrase' wk-gateway
```

```
# starting wakari-gateway manually
sudo service wakari-gateway start --ask-for-passphrase
Passphrase?
```

4. Restart the gateway:

```
sudo service wakari-gateway restart
```

Configuring SSL on compute nodes

Anaconda Enterprise does not support direct SSL on Compute Nodes. If you need SSL on Compute Nodes, you must install each Compute Node on the same server as a Gateway using `http://localhost:5002` for the URL value while adding it as a resource, and you must use a Gateway for each and every Compute Node.

Security reminder

The permissions on the cert files must be set correctly to prevent them from being read by others. Since NGINX is run by the root user, only the root user needs read access to the cert files.

EXAMPLE: If the cert files are called `server.crt` and `server.key`, then use the root account to set permissions:

```
chmod 600 server.key
chmod 600 server.crt
```

Enabling or disabling the Strict-Transport-Security header

By default, Strict-Transport-Security (STS) is enabled in the www.enterprise.https.conf file:

```
add_header Strict-Transport-Security max-age=31536000;
```

It can remain enabled if either of the following is true:

- The gateway is running on a different host than the server.

  or

- SSL has been enabled for the gateway.

You must comment out this line if both of the following are true:

- The gateway is running on the same host as the server.

  and

- SSL has not been enabled for the gateway.

Leaving STS enabled when these conditions are true will cause a mismatch in protocols between the server and gateway, causing your apps to fail to launch correctly.

Configuring single sign-on

AEN's single sign-on (SSO) capability creates a new authentication provider that defers to your Anaconda Repository for login and authentication cookies.

To enable SSO:

1. Deploy AEN and Repository on the same machine.
2. In the /opt/wakari/wakari-server/etc/wakari/config.json file, add:

   ```
   {
   EXISTING_CONFIGURATION,
   "SECRET_KEY": "<repo signing secret>",
   "REPO_LOGIN_URL":
      "http://example_repo.com:8080/account/login?next=http://example_repo.com/"
   }
   ```

3. Copy the SECRET_KEY from the Repository configuration file.
4. In the /opt/wakari/wakari-server/etc/wakari/wk-server-config.json file, modify:

   ```
   {
   EXISTING_CONFIGURATION,
   "accounts": "wk_server.plugins.accounts.repo",
   }
   ```

5. If you are using Repository version 2.33.3 through 2.33.10, set USE_SERVER_BASED_SESSIONS: false in the Repository configuration.

   This setting affects the network security properties of AEN and Repository. Specifically, if USE_SERVER_BASED_SESSIONS is set to false, and if a new cross-site scripting (XSS) vulnerability is discovered, it could expose an additional server fixation vulnerability. Please discuss this with your Anaconda representative and be sure the feature is compatible with your network requirements before setting USE_SERVER_BASED_SESSIONS: false.

6. To activate the changes restart `wakari-server`:

   ```
   sudo service wakari-server restart
   ```

SSO is enabled.

Adding a third-party extension

Anaconda officially supports and tests functionality of the default environment(s) only for those extensions that ship with AEN.

It is possible to add third-party and custom extensions from conda-forge or pip, but doing so may cause instability in your default project environments or kernels.

CAUTION: Anaconda does not officially support third-party extensions. This section is informational only.

Installing unofficial Jupyter Notebook extensions for AEN

TIP: Always back up and verify your complete system before installing extensions.

The jupyter-contrib-nbextensions extensions are installed on a compute node.

The default conda executable directory for AEN is `/opt/wakari/anaconda/bin/conda`. If you are installing a Jupyter extension, it must be installed in the `wakari-compute` directory.

EXAMPLE: Run:

```
/opt/wakari/anaconda/bin/conda install -p /opt/wakari/wakari-compute/ -c conda-forge␣
↪jupyter_contrib_nbextension
```

For more information, see Unofficial Jupyter Notebook Extensions.

Configure search indexing

For search indexing to work correctly, verify that the AEN Compute node can communicate with the AEN Server.

```
curl -m 5 $AEN_SERVER > /dev/null
```

There must be at least one `inotify` watch available for the number of subdirectories within the project root filesystem. Some Linux distributions default to a low number of watches, which can prevent the search indexer from monitoring project directories for changes.

```
cat /proc/sys/fs/inotify/max_user_watches
```

If necessary, increase the number of max user watches with the following command:

```
echo fs.inotify.max_user_watches=100000 | sudo tee -a /etc/sysctl.conf && sudo sysctl␣
↪-p
```

There must be at least one `inotify` user instance available per project.

```
cat /proc/sys/fs/inotify/max_user_instances
```

If necessary, this can be increased with the following command:

```
echo fs.inotify.max_user_instances=1000 | sudo tee -a /etc/sysctl.conf && sudo sysctl␣
↪-p
```

Create custom Jupyter kernel for Pyspark

These instructions add a custom Jupyter Notebook option to allow users to select PySpark as the kernel.

Install Spark

The easiest way to install Spark is with Cloudera CDH.

You will use YARN as a resource manager. After installing Cloudera CDH, install Spark. Spark comes with a PySpark shell.

Create a notebook kernel for PySpark

You may create the kernel as an administrator or as a regular user. Read the instructions below to help you choose which method to use.

1. As an administrator

Create a new kernel and point it to the root env in each project. To do so create a directory 'pyspark' in */opt/wakari/wakari-compute/share/jupyter/kernels/*.

Create the following kernel.json file:

```
{"argv": ["/opt/wakari/anaconda/bin/python",
 "-m", "ipykernel", "-f", "connection_file}", "--profile", "pyspark"],
 "display_name":"PySpark",  "language":"python" }
```

You may choose any name for the 'display_name'.

This configuration is pointing to the python executable in the root environment. Since that environment is under admin control, users cannot add new packages to the environment. They will need an admin to help update the environment.

2. As an administrator without IPython profile

To have an admin level PySpark kernel without the user .ipython space:

```
{"argv":
["/opt/wakari/wakari-compute/etc/ipython/pyspark.sh", "-f", "{connection_file}"],
"display_name":"PySpark",  "language":"python" }
```

NOTE: The pyspark.sh script is defined in *Without IPython profile* section below.

3. As a regular user

Create a new directory in the user's home directory: *.local/share/jupyter/kernels/pyspark/*. This way the user will be using the default environment and able to upgrade or install new packages.

Create the following kernel.json file:

```
{"argv": ["/projects/<username>/<project_name>/envs/default/bin/python",
 "-m", "ipykernel", "-f", "connection_file}", "--profile", "pyspark"],
 "display_name":"PySpark",  "language":"python" }
```

NOTE: Replace "<username>" with the correct user name and "<project_name>" with the correct project name.

You may choose any name for the 'display_name'.

Create an IPython profile

The above profile call from the kernel requires that we define a particular PySpark profile. This profile should be created for each user that logs in to AEN to use the PySpark kernel.

In the user's home, create the directory and file `~/.ipython/profile_pyspark/startup/00-pyspark-setup.py` with the file contents:

```
import os
import sys

# The place where CDH installed spark, if the user installed Spark locally it can be␣
↪changed here.
# Optionally we can check if the variable can be retrieved from environment.

os.environ["SPARK_HOME"] = "/usr/lib/spark"

os.environ["PYSPARK_PYTHON"] = "/opt/wakari/anaconda/bin/python"

# And Python path
os.environ["PYLIB"] = os.environ["SPARK_HOME"] + "/python/lib"
sys.path.insert(0, os.environ["PYLIB"] +"/py4j-0.9-src.zip")  #10.4-src.zip")
sys.path.insert(0, os.environ["PYLIB"] +"/pyspark.zip")

os.environ["PYSPARK_SUBMIT_ARGS"] = "--name yarn pyspark-shell"
```

Now log in using the user account that has the PySpark profile.

Without IPython profile

If it is necessary to avoid creating a local profile for the users, a script can be made to be called from the kernel. Create a bash script that will load the environment variables:

```
sudo -u $AEN_SRVC_ACCT mkdir /opt/wakari/wakari-compute/etc/ipython
sudo -u $AEN_SRVC_ACCT touch /opt/wakari/wakari-compute/etc/ipython/pyspark.sh
sudo -u $AEN_SRVC_ACCT chmod a+x /opt/wakari/wakari-compute/etc/ipython/pyspark.sh
```

The contents of the file should look like:

```
#!/usr/bin/env bash
# setup environment variable, etc.

export PYSPARK_PYTHON="/opt/wakari/anaconda/bin/python"
export SPARK_HOME="/usr/lib/spark"

```

(continues on next page)

(continued from previous page)

```
# And Python path
export PYLIB=$SPARK_HOME:/python/lib
export PYTHONPATH=$PYTHONPATH:$PYLIB:/py4j-0.9-src.zip
export PYTHONPATH=$PYTHONPATH:$PYLIB:/pyspark.zip

export PYSPARK_SUBMIT_ARGS="--name yarn pyspark-shell"

# run the ipykernel
exec /opt/wakari/anaconda/bin/python -m ipykernel $@
```

Using PySpark

When creating a new notebook in a project, now there will be the option to select PySpark as the kernel. When creating such a notebook you'll be able to import pyspark and start using it:

```
from pyspark import SparkConf
from pyspark import SparkContext
```

NOTE: You can always add those lines and any other command you may use frequently in the PySpark setup file `00-pyspark-setup.py` as shown above.

Upgrading AEN

CAUTION: These instructions are for upgrading AEN to the current version 4.2.2 from 4.2.1 ONLY. Each version must be upgraded iteratively from the previous version. Do not skip versions.

Upgrade instructions for previous versions:

- *AEN 4.2.1 upgrade instructions*
- *AEN 4.2.0 upgrade instructions*
- *AEN 4.1.3 upgrade instructions*
- *AEN 4.1.2 upgrade instructions*
- *AEN 4.1.1 upgrade instructions*.
- *AEN 4.1.0 upgrade instructions*.
- *AEN 4.0.0 upgrade instructions*.

For upgrades from versions before those listed above, please contact your enterprise support representative.

NOTE: Named Service Account functionality is available with AEN 4.0.0+ for new installations only. It is not available for upgraded installations. Contact your enterprise support representative for more information.

An AEN platform update requires that each instance of the 3 node types be upgraded individually:

- AEN Server
- AEN Gateway
- AEN Compute

The upgrade process requires that all AEN service instances be stopped, upgraded, and then restarted.

NOTE: Any commands that call for the root user can also be done using sudo.

If you encounter any difficulty during the upgrade process, see *Troubleshooting* which provides guidance on:

- processes
- configuration files
- log files
- ports

If you are unable to resolve an installation or upgrade problem, please contact your enterprise support representative.

Before you upgrade

CAUTION: Make a tested backup of your installation before starting the upgrade. Upgrading to a higher version of AEN is not reversible. Any errors during the upgrade procedure may result in partial or complete data loss and require restoring data from backups.

CAUTION: Terminate all AEN applications and stop all projects before starting the upgrade process.

Before upgrading each service on each host:

1. Suspend the services on each of the nodes:

```
sudo service wakari-server stop
sudo service wakari-gateway stop
sudo service wakari-compute stop
```

2. Set the AEN Functional ID ("NFI") and AEN Functional Group ("NFG") to the NFI and NFG of the current installation:

```
export AEN_SRVC_ACCT="wakari"
export AEN_SRVC_GRP="wakari"
```

NOTE: The default NFI is `wakari`, but `aen_admin` or any other name may be used instead.

For more information on NFI and NFG, see the *installation instructions*.

3. Install `wget`:

```
yum install wget
```

Upgrading the AEN server node

NOTE: If you are using LDAP-based authentication, back up the `/opt/wakari/wakari-server/etc/wakari/wk-server-config.json` configuration file. After the server has been upgraded, copy that file back into the same location as before the upgrade.

Complete the following steps on the server host:

1. Stop the Elasticsearch service:

   ```
   sudo service elasticsearch stop
   ```

2. Remove any previous index:

   ```
   sudo rm -rf /var/lib/elasticsearch/*
   ```

   NOTE: You can choose to keep the old index, but if you detect any issues with the search capabilities after the upgrade, you will need to run the following to start with a clean index:

   ```
   sudo service wakari-server stop
   sudo service elasticsearch stop
   sudo rm -rf /var/lib/elasticsearch/*
   sudo service elasticsearch start
   sudo service wakari-server start
   ```

3. Upgrade the server:

   ```
   pushd /tmp
   wget http://j.mp/aen-server-update-4_2_2

   sudo -E -u $AEN_SRVC_ACCT /opt/wakari/miniconda/bin/conda install \
                   -p /opt/wakari/wakari-server          \
                   --file aen-server-update-4_2_2

   sudo -E -u $AEN_SRVC_ACCT /opt/wakari/miniconda/bin/conda install \
                    -p /opt/wakari/wakari-server          \
                    --no-deps                             \
                    wakari-enterprise-server-conf-update=2.0.9
   popd
   ```

4. Start Elasticsearch:

   ```
   sudo service elasticsearch start
   ```

   Or, if you do not want to use the search features, edit your server's `/opt/wakari/wakari-server/etc/wakari/config.json` file by adding the line `"SEARCH_ENABLED":  false`.

5. Restart the *NGINX* server:

   AEN server version >= 4.1.3 uses Unix sockets for communication with NGINX. Restart NGINX to load this new configuration:

   ```
   sudo service nginx restart
   ```

   Alternatively, you can restart NGINX with:

   ```
   sudo nginx -s stop
   sudo nginx
   ```

6. Start the server:

   ```
   sudo service wakari-server start
   ```

7. Check that the server is running properly:

   ```
   sudo service wakari-server status
   ```

8. If you see NGINX errors, please check the configuration at /opt/wakari/wakari-server/etc/nginx/conf.d/www.enterprise.conf:18.

9. Connect to AEN server using your web browser with the correct protocol (http or https), hostname and port number.

Upgrading the AEN gateway node

Complete the following steps on each gateway host:

1. Upgrade the gateway:

```
pushd /tmp
wget http://j.mp/aen-gateway-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/miniconda/bin/conda install \
                -p /opt/wakari/wakari-gateway         \
                --file aen-gateway-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/miniconda/bin/conda install \
                -p /opt/wakari/wakari-gateway         \
                --no-deps                             \
                wakari-enterprise-gateway-conf-update=2.0.9
popd
```

2. Start the gateway:

```
sudo service wakari-gateway start
```

3. Check that the gateway is running properly:

```
sudo service wakari-gateway status
```

4. Connect to the gateway using your web browser with the correct http/https, hostname and port number.

Upgrading AEN compute nodes

Complete the following steps on each host where an AEN compute service is running:

1. Check for any wakari-indexer processes running:

```
ps aux | grep wakari-indexer
```

   NOTE: If you stopped all the projects, you will not see any wakari-indexer processes running.

   Terminate any remaining wakari-indexer processes:

```
sudo killall wakari-indexer
```

   NOTE: The processes killed with killall are run by the $AEN_SRVC_ACCT user, so they can be killed as root with sudo killall or killed as the $AEN_SRVC_ACCT user with sudo -u $AEN_SRVC_ACCT killall. Example commands show the sudo killall option.

2. Check for any AEN applications processes running—Workbench, Viewer, Terminal or Notebook:

```
ps aux | grep wk-app-gateone
ps aux | grep wk-app-workbench
ps aux | grep wk-app-viewer
ps aux | grep wk-app-terminal
ps aux | grep jupyter-notebook
```

NOTE: If you stopped all the projects, you will not see any AEN app processes running.

Terminate any remaining AEN application processes by running one or more of the following:

```
sudo killall wk-app-gateone
sudo killall wk-app-workbench
sudo killall wk-app-viewer
sudo killall wk-app-terminal
sudo killall jupyter-notebook
```

3. Verify the contents of `/opt/wakari/anaconda/.condarc`. Modify it to contain the following entries, and possibly others if you customized the `.condarc` file.

   NOTE: Modify the file as the AEN_SRVC_ACCT user (or be sure to keep the same ownership).

```
channels:
  - https://conda.anaconda.org/t/<TOKEN>/anaconda-nb-extensions
  - r
  - https://conda.anaconda.org/wakari
  - defaults

create_default_packages:
  - anaconda-client
  - ipykernel
```

   NOTE: Contact your enterprise support representative to get your token for the Anaconda channel referenced above. Replace `<TOKEN>` with the actual token from your enterprise support representative.

4. Upgrade *Anaconda* in the root environment:

```
pushd /tmp
wget http://j.mp/aen-anaconda-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda install \
              -p /opt/wakari/anaconda              \
              --file aen-anaconda-update-4_2_2
popd
```

5. Upgrade each compute service:

```
pushd /tmp
wget http://j.mp/aen-compute-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda install \
                -p /opt/wakari/wakari-compute        \
                --file aen-compute-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda install \
                --no-deps                            \
                -p /opt/wakari/wakari-compute        \
                wakari-enterprise-compute-conf-update=2.0.13
popd
```

NOTE: When upgrading the wakari-compute environment, you may see ImportError warnings with some nbextensions. As long as the Validating message is OK, the ImportError warnings are harmless—a consequence of the post-link presence on those packages.

6. Initialize the root environment to prime the package cache:

   ```
   sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda create \
                   -p /opt/wakari/testenv \
                   --clone root
   ```

7. Test the offline cloning step:

   ```
   sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda create \
                   -p /opt/wakari/testenvoffline \
                   --clone root --offline
   ```

8. Remove the test environments:

   ```
   sudo rm -rf /opt/wakari/testenv
   sudo rm -rf /opt/wakari/testenvoffline
   ```

9. Install necessary dependencies:

   NOTE: Skip this step if you already have these dependencies installed from previous installations.

   ```
   sudo yum groupinstall "X Window System" -y
   sudo yum install git -y
   ```

   NOTE: If you don't want to install the whole X Window System, you must install the following packages to have R plotting support:

   ```
   sudo yum install -y libXrender libXext libXdmcp libSM libICE libXt \
   dejavu-sans-fonts dejavu-serif-fonts dejavu-fonts-common \
   fontpackages-filesystem
   ```

10. Start the compute service:

    ```
    sudo service wakari-compute start
    ```

11. Verify the compute service is running properly:

    ```
    sudo service wakari-compute status
    ```

12. Restart the AEN Server with:

    ```
    sudo service wakari-server restart
    ```

13. Repeat this upgrade procedure for all compute nodes in your Data Center.

After upgrading

1. Restart the projects and start using AEN applications.

2. If you have a *customized default environment*, you may choose to upgrade it depending on the needs of your users.

   Upgrade the customized default environment at `/opt/wakari/anaconda/envs/default` with the `$AEN_SRVC_ACCT` user:

```
pushd /tmp
wget http://j.mp/aen-anaconda-update-4_2_2

sudo -E -u $AEN_SRVC_ACCT /opt/wakari/anaconda/bin/conda install \
        -p /opt/wakari/anaconda/envs/default \
        --file aen-anaconda-update-4_2_2
popd
```

To upgrade the customized default environments for every user and every project at `/projects/<USER>/<PROJECT>/envs/default`, run these commands for **every** user as that user:

```
pushd /tmp
wget http://j.mp/aen-anaconda-update-4_2_2

sudo -E -u <USER> /opt/wakari/anaconda/bin/conda install \
        -p /projects/<USER>/<PROJECT>/envs/default \
        --file aen-anaconda-update-4_2_2
popd
```

NOTE: Replace `<USER>` with the user's name. Replace `<PROJECT>` with the project name.

NOTE: Upgrading the default environment at `/opt/wakari/anaconda/envs/default` does NOT automatically upgrade the default environment in the users pre-existing projects. For pre-existing projects, the upgrade, if requested, should be done on a per-user basis.

NOTE: These commands update packages listed in `aen-anaconda-update-4_2_2` and do not update any other package.

3. If you did not stop all your projects before upgrading, then the first time you start an application you will see an error page requesting that you restart the application.
4. Restart the application to complete the upgrade.
5. If you still see old applications or icons after restart, reload the page to reset the browser cache.

Uninstalling AEN

Each AEN node must be uninstalled separately.

Begin by setting the AEN Functional ID (NFI). The NFI is the username of the AEN Service Account which is used to run all AEN services and is also the username of the AEN Admin account. The NFI may be any name. The default NFI is `wakari`. The NFI is also often set to `aen_admin`. The NFI (and AEN Functional Group or NFG) are described in *the installation instructions*.

Set the NFI with this command:

```
export AEN_SRVC_ACCT="aen_admin"
```

Replace the name `aen_admin` with the NFI that was set in your installation of Anaconda Enterprise Notebooks.

Uninstalling a server node

To remove a server node, run the following commands as root or sudo on the server node's host system:

1. Stop the server processes:

   ```
   service wakari-server stop
   ```

2. Stop MongoDB:

   ```
   service mongod stop
   ```

3. Remove AEN server software, AEN database files and NGINX configuration:

   ```
   rm -Rf /opt/wakari/wakari-server
   rm -Rf /opt/wakari/miniconda
   rm -Rf /var/lib/mongo/wakari*
   rm -Rf /etc/nginx/conf.d/www.enterprise.conf
   ```

   NOTE: Remove `/etc/nginx/conf.d/www.enterprise.https.conf` if SSL is enabled on the Server node.

4. Restart MongoDB and NGINX:

   ```
   service mongod restart
   service nginx restart
   ```

5. Check for any outstanding server processes and stop them:

   ```
   ps -ef | grep -e wakari-server -e wk-server
   ```

6. Remove the AEN Service Account:

   ```
   userdel $AEN_SRVC_ACCT
   ```

7. Check for and remove any references to "aen" or "wakari" from the root user's `.condarc` file:

   ```
   grep -i aen ~/.condarc
   grep -i wakari ~/.condarc
   ```

Uninstalling a gateway node

To uninstall a gateway node, run the following commands as root or sudo on the gateway host system:

1. Stop the gateway processes:

   ```
   service wakari-gateway stop
   ```

2. Remove gateway software:

   ```
   rm -Rf /opt/wakari/wakari-gateway
   ```

3. Check for any outstanding gateway processes and stop them:

   ```
   ps -ef | grep -e wakari-gateway -e wk-gateway
   ```

4. Remove the AEN Service Account:

```
userdel $AEN_SRVC_ACCT
```

5. Check for and remove any references to “aen” or “wakari” from the root user’s `.condarc` file:

   ```
   grep -i aen ~/.condarc
   grep -i wakari ~/.condarc
   ```

Uninstalling a compute node

To remove a compute node, run the following commands as root or sudo on each compute node host system:

1. Stop the compute processes:

   ```
   service wakari-compute stop
   ```

2. Remove the compute software:

   ```
   rm -Rf /opt/wakari/wakari-compute
   rm -Rf /opt/wakari/miniconda
   rm -Rf /opt/wakari/anaconda
   ```

3. Check for any outstanding compute processes and stop them:

   ```
   ps -ef | grep -e wakari-compute -e wk-compute
   ```

4. Remove the AEN Service Account:

   ```
   userdel $AEN_SRVC_ACCT
   ```

5. Check for and remove any references to “aen” or “wakari” from the root user’s `.condarc` file:

   ```
   grep -i aen ~/.condarc
   grep -i wakari ~/.condarc
   ```

OPTIONAL: Removing projects from compute nodes

CAUTION: This is an extreme measure and is not necessary in most instances. We recommend you create and verify a backup before doing this or any other file removal.

To remove all AEN projects from all of your compute nodes:

```
rm -Rf /projects
```

This is a step-by-step guide to installing an Anaconda Enterprise Notebooks system comprised of a front-end server, a gateway and compute machines.

If you have any questions about these instructions or you encounter any issues while installing AEN, please contact your sales representative or Priority Support team.

When you have completed the installation process, review the *optional configuration tasks* to see if any are appropriate for your system.

Distributed install

In a distributed install the server and gateway run on separate hosts.

Single-box install

In a single-box install, both the server and the gateway need separate external ports since they are independent services that are running on the same host in the single-box installation.

Both port 80 and port 8089 must be open on the firewall for a single-box install.

The compute node only receives connections from the gateway and server nodes and typically runs on port 80 or port 443.

User management

Adding or removing an administrative user

An administrator can make any other user an administrator—or remove their administrator permissions—by using administrator commands in the Terminal application.

A user can also be designated as a superuser or as staff, giving them greater administrative privileges within the system.

Designating a user as an administrator/superuser

To designate a user as an administrator and superuser:

```
/opt/wakari/wakari-server/bin/wk-server-admin superuser --add <username>
```

NOTE: Replace `<username>` with the actual username.

EXAMPLE: To give administrative privileges to the user named “jsmith” and set them as a superuser, run:

```
/opt/wakari/wakari-server/bin/wk-server-admin superuser --add jsmith
```

Removing an administrator/superuser

To remove a user’s administrative privileges:

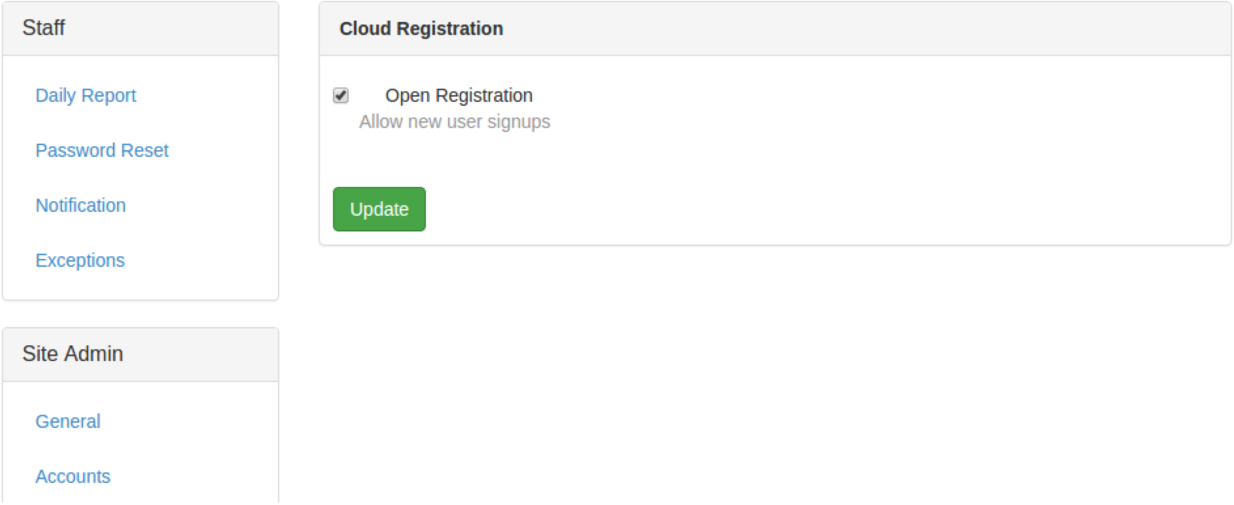
```
/opt/wakari/wakari-server/bin/wk-server-admin superuser --remove <username>
```

NOTE: Replace `<username>` with the actual username.

Allowing and restricting new user registration

When Open Registration is enabled, anyone who has access to the URL of your AEN server can create their own account.

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Accounts.

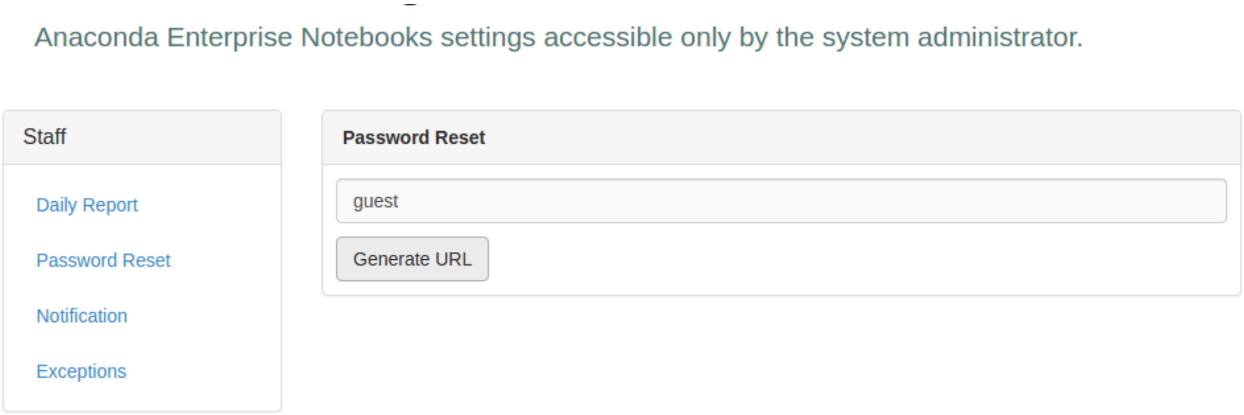


☑ Open Registration
Allow new user signups

3. To open user registration, select the Open Registration checkbox. To close registration, clear the checkbox.
4. Click the Update button.

Resetting a user password

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Staff** menu, select Password Reset:


Anaconda Enterprise Notebooks settings accessible only by the system administrator.

guest

3. Enter the username of the user whose password needs to be reset.
4. Click the Generate URL button.

   A password reset link is generated that you can email to the user.

Alternatively you may use the command line interface:

1. Use ssh to log into the server as root.
2. Run:

   ```
   /opt/wakari/wakari-server/bin/wk-server-admin reset-password -u SOME_USER -p SOME_
   ↪PASSWORD
   ```

   NOTE: Replace SOME_USER with the username and SOME_PASSWORD with the password.
3. Log into AEN as the user.

Managing permissions

This page explains the admin commands used to manage user permissions.

Checking file ownership

To verify that all files in the `/opt/wakari/anaconda` directory are owned by the `wakari` user or group:

```
root@server # find /opt/wakari/anaconda \! -user wakari -print
root@server # find /opt/wakari/anaconda \! -group wakari -print
```

Fixing file ownership settings

To fix the ownership settings of any files that are listed in the output:

```
chown -R wakari:wakari /opt/wakari/anaconda
```

Setting a file owner and permissions

To set a file owner and set its permissions:

```
chown wakari:wakari /opt/wakari/wakari-server/bin/wk-*
chmod 700 /opt/wakari/wakari-server/bin/wk-*
```

Verifying that POSIX ACLs are enabled

The `acl` option must be enabled on the file system that contains the project root directory.

NOTE: By default, the project root directory is `/projects`.

To determine the project root directory where a custom `projectRoot` is configured:

```
root@compute # grep projectRoot /opt/wakari/wakari-compute/etc/wakari/config.json
```

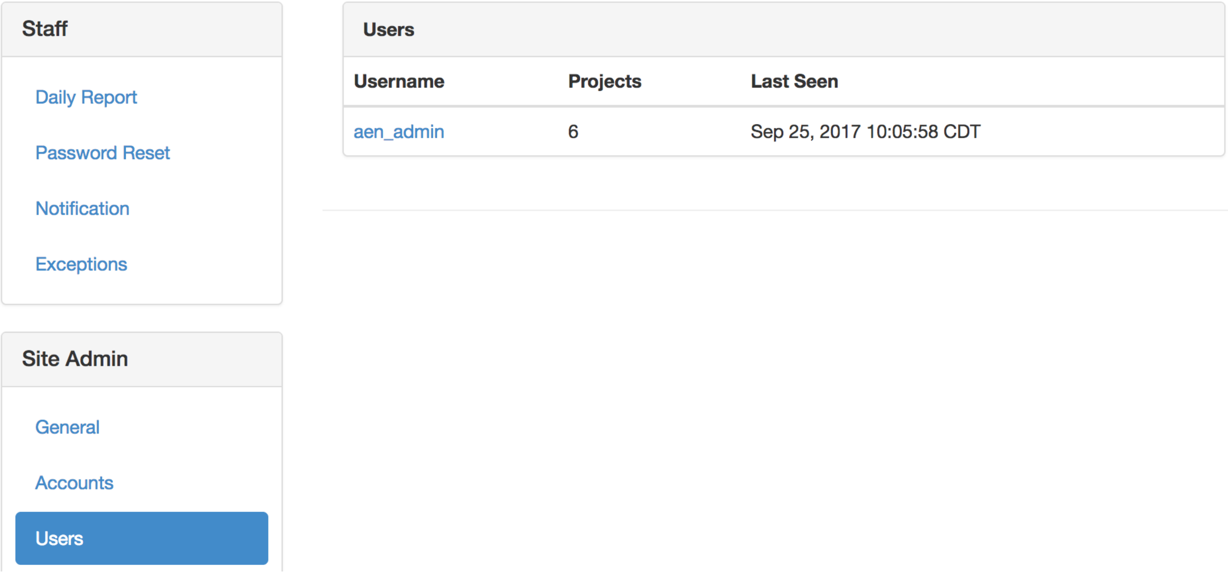
The `mount` options or default options listed by `tune2fs` should indicate that the `acl` option is enabled.

EXAMPLE:

```
root@compute # fs=`df /projects | tail -1 | cut -d " " -f 1`
root@compute # mount | grep $fs
/dev/vda on / type ext4 (rw)
root@compute # tune2fs -l $fs | grep options
Default mount options:    user_xattr acl
```

Viewing a list of users

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Users:



Users

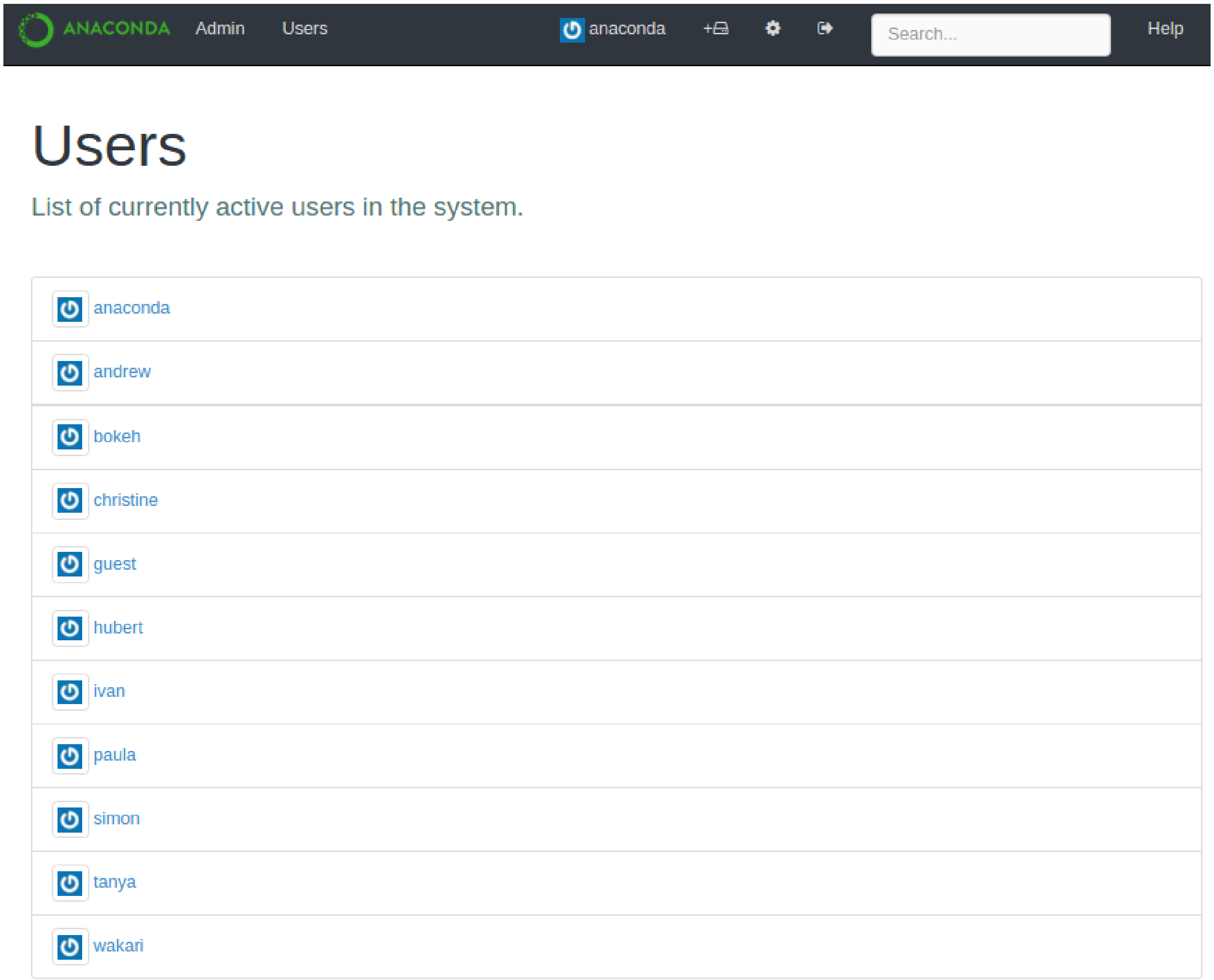
Username	Projects	Last Seen
aen_admin	6	Sep 25, 2017 10:05:58 CDT

The Users section lists the all users who are signed up, the number of projects they have created and the last time they logged on to AEN.

Viewing a list of currently active users

In the AEN navigation bar, click Users.

Click a username to open the user's profile page.



Users

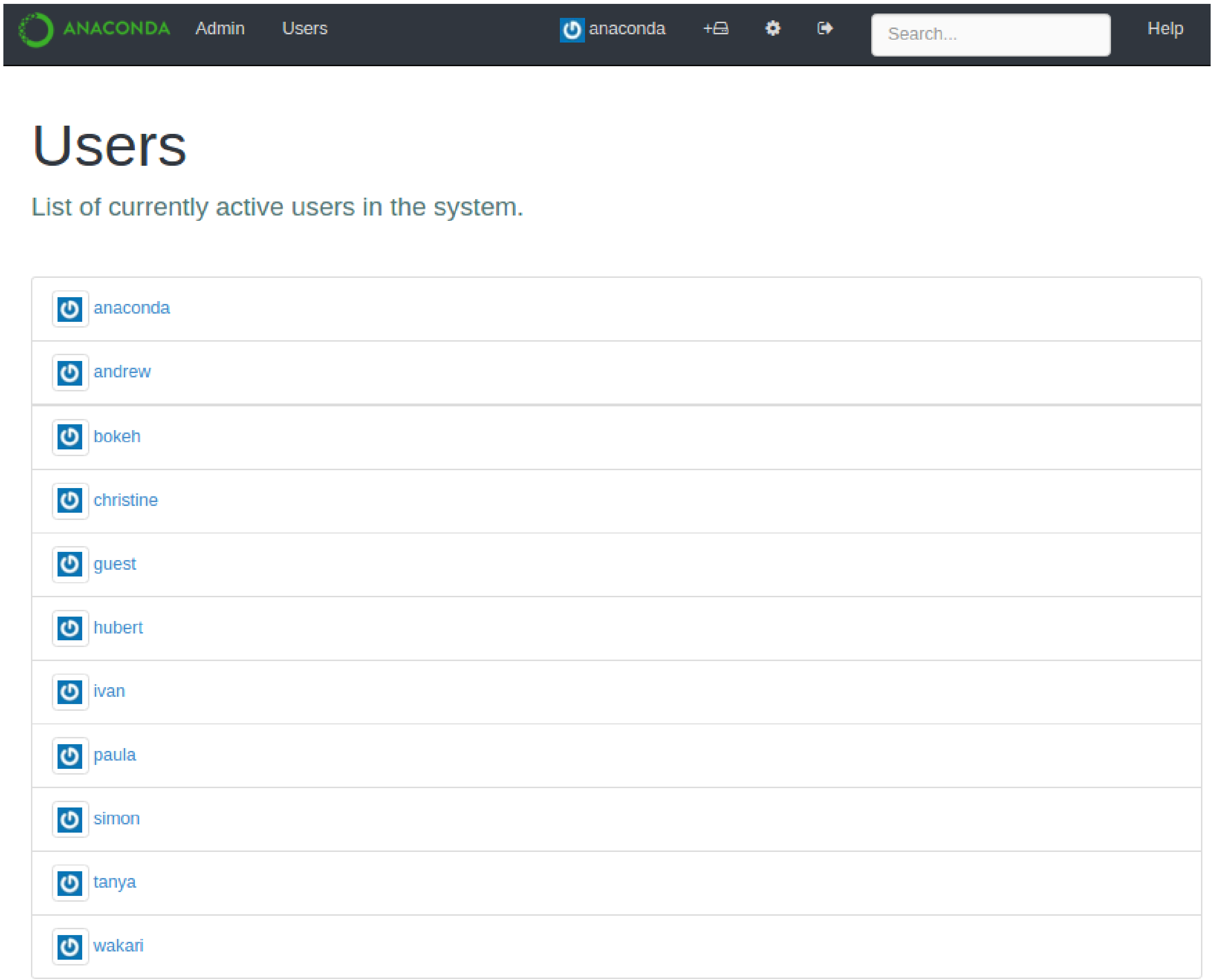
List of currently active users in the system.

- anaconda
- andrew
- bokeh
- christine
- guest
- hubert
- ivan
- paula
- simon
- tanya
- wakari

Viewing a user profile

A user's profile page includes a summary of the projects created by that user and a list of projects on which the user is a team member.

1. In the AEN navigation bar, click Users to see a list of users who are currently logged into the system.
2. On the Users page, click the username of the user whose profile page you want to view.



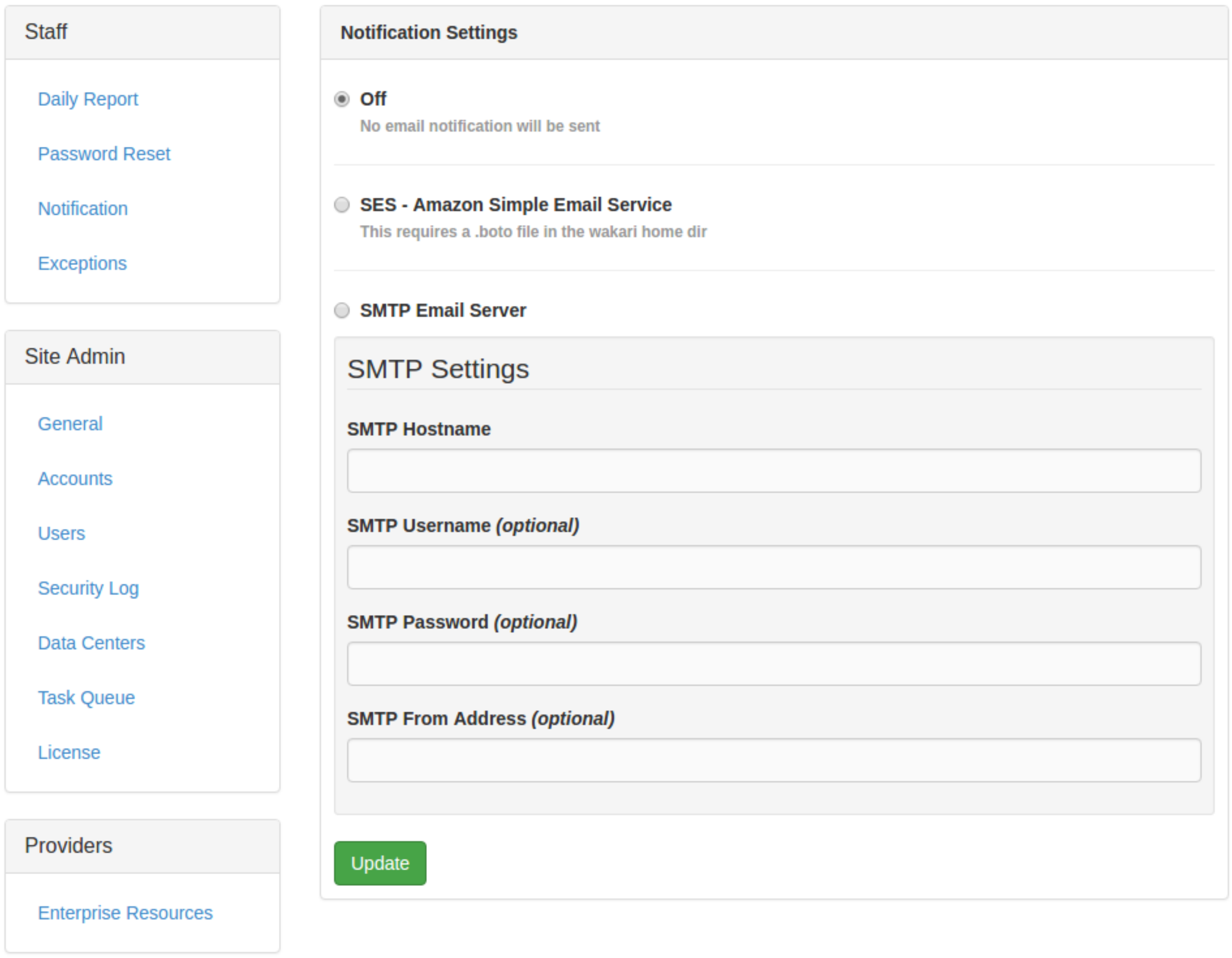
Users

List of currently active users in the system.

anaconda
andrew
bokeh
christine
guest
hubert
ivan
paula
simon
tanya
wakari

Sending a system message

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Staff** menu, select Notification:



Notification Settings

● Off
No email notification will be sent

○ SES - Amazon Simple Email Service
This requires a .boto file in the wakari home dir

○ SMTP Email Server

SMTP Settings

SMTP Hostname

SMTP Username *(optional)*

SMTP Password *(optional)*

SMTP From Address *(optional)*

The Notification Settings section allows you to create a system message that can be relayed to users.

By default, notifications are off.

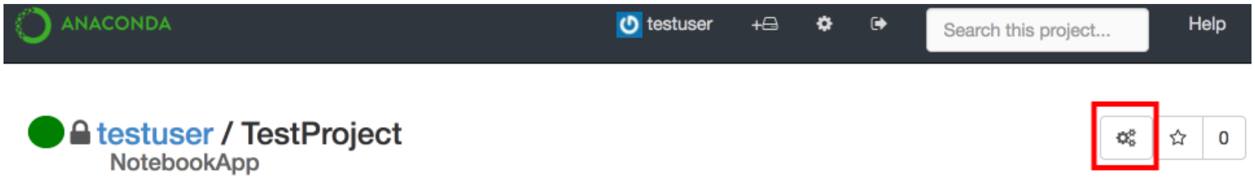
3. To turn on email notifications, select the radio button for the type of email service to use:
   - SES to use Amazon Simple Email Service (SES).
   - SMTP Email Server.
4. If you select SMTP Email Server, complete the SMTP Settings.

NOTE: If you get an error message after changing the SMTP settings, you may need to restart the server.

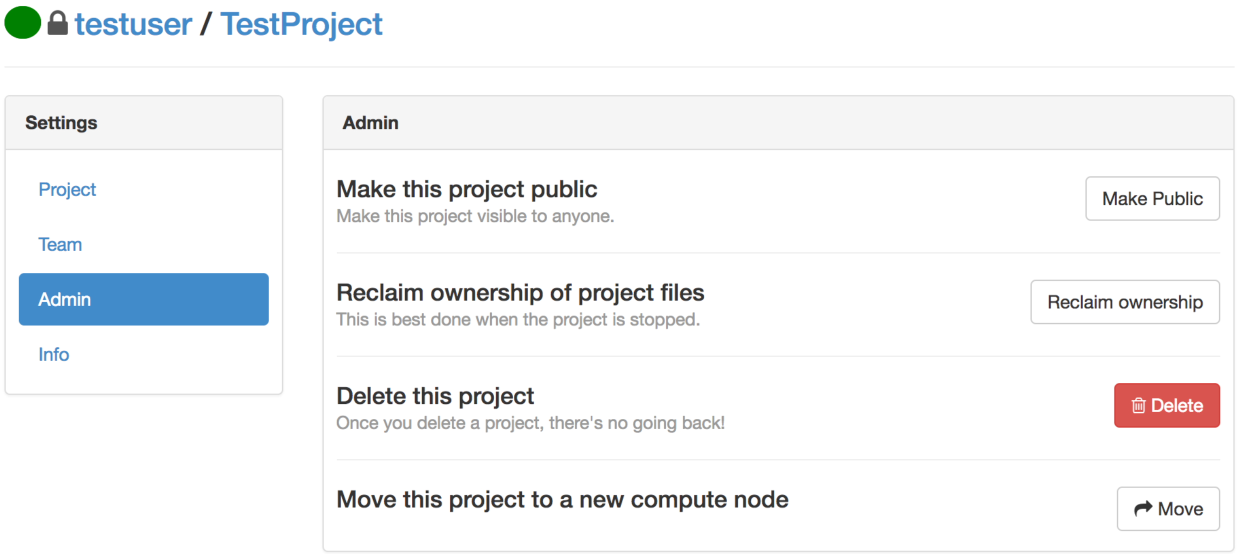
Moving a project to another compute node

If you have multiple compute nodes available and want to move a project from one to another, the project must exist on both nodes.

1. Verify that the project has been created on both compute nodes. You can use `rsync` for this job unless you have a shared file system like nfs.
2. On the project home page, click the Project Settings icon to open the Project Settings page.



3. In the **Settings** menu, select Admin.



Make this project public

Make this project visible to anyone.

Reclaim ownership of project files

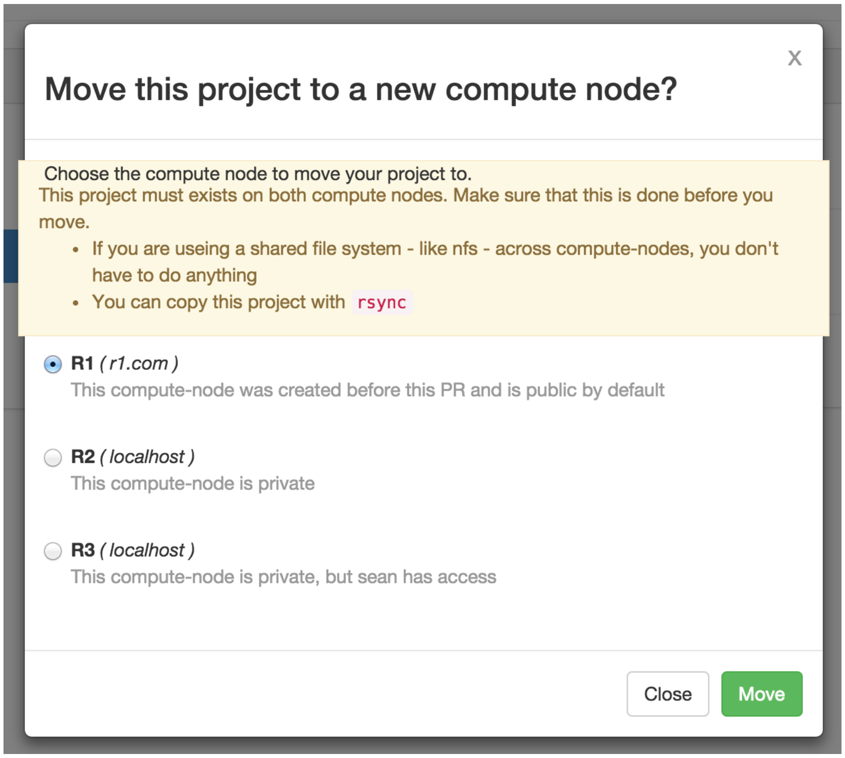
This is best done when the project is stopped.

Delete this project

Once you delete a project, there's no going back!

Move this project to a new compute node

4. Click the Move button.
5. In the move dialog box, click to choose the compute node destination, and click the Move button.



Move this project to a new compute node?

Choose the compute node to move your project to.
This project must exists on both compute nodes. Make sure that this is done before you move.

- If you are useing a shared file system - like nfs - across compute-nodes, you don't have to do anything
- You can copy this project with `rsync`

● **R1** *(r1.com)*
This compute-node was created before this PR and is public by default

○ **R2** *(localhost)*
This compute-node is private

○ **R3** *(localhost)*
This compute-node is private, but sean has access

Deleting a user

To remove a user from the AEN database:

```
/opt/wakari/wakari-server/bin/wk-server-admin remove-user <username>
```

NOTE: Replace `<username>` with the actual username.

NOTE: Changing the owner of a project requires that both the previous owner and the new owner are still AEN users. Before deleting a user, *change the owner* of that user's projects.

Deleting a project

To remove a project from the AEN database:

```
/opt/wakari/wakari-server/bin/wk-server-admin remove-project <username> <projectname>
```

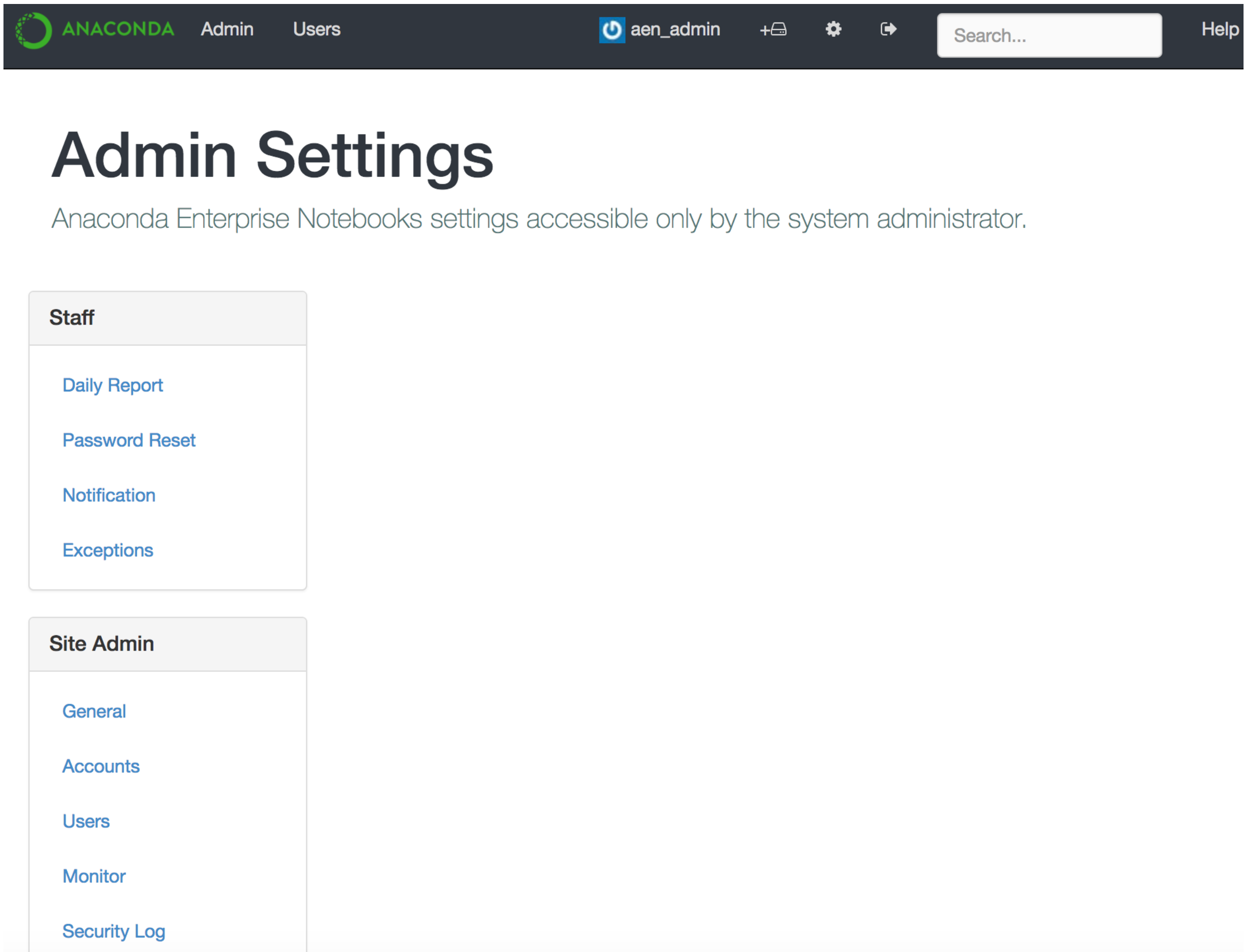
NOTE: Replace `<username>` with the actual username and `<projectname>` with the actual project name you are removing.

System management

Opening the Admin dashboard

If you have administrator privileges, you see two additional links in the AEN navigation bar—Admin and Users:

To open the Admin dashboard, click the Admin link.



Admin Settings

Anaconda Enterprise Notebooks settings accessible only by the system administrator.

Backing up and restoring AEN

Document purpose

This document lays out the steps to backup and restore Anaconda Enterprise Notebooks (AEN) for Disaster Recovery. It is not intended to provide High Availability. Each of the components (Server, Gateway and Compute) has its own instructions and each may be done individually as needed. The steps primarily involve creating tar files of important configuration files and data.

This document is written for a system administrator who is comfortable with basic Linux command line navigation and usage.

To migrate to a new cluster, use these backup and restore instructions to back up the system from the old cluster and restore it to the new cluster.

Important notes

Review the *Concepts* page to become familiar with the different components and how they work together.

Root or sudo access is required for some commands.

CAUTION: All commands MUST be run by `$AEN_SRVC_ACCT` (the account used to run AEN) except for those commands explicitly indicated to run as root or sudo. If the commands are not run by the correct user, the installation will not work, and a full uninstallation and reinstallation will be required!

These instructions assume that the fully qualified domain name (FQDN) has not changed for any of the component nodes. If any of the FQDNs are not the same, additional steps will be needed.

Server component steps

Backup

Mongo database

This will create a single tar file called `aen_mongo_backup.tar` that includes only the database named "wakari" that is used by AEN. It also generates a log of the database backup.

NOTE: These commands must be run by `$AEN_SRVC_ACCT`.

```
mongodump -db wakari -o aen_main >> mongo_backup.log
tar -cvf aen_mongo_backup.tar aen_main
```

AEN Server config files (including License file)

Create a tar file of all of the configuration files, including any license files.

NOTE: This command must be run by $AEN_SRVC_ACCT.

```
tar -cvf aen_server_config.tar -C /opt/wakari/ wakari-server/etc/wakari/
```

Nginx config (if needed)

Make a copy of the nginx configuration file if it has been customized. The default configuration for the AEN server is a symlink.

NOTE: This command must be run by $AEN_SRVC_ACCT.

```
/etc/nginx/conf.d/www.enterprise.conf -> /opt/wakari/wakari-server/etc/nginx/conf.d/
↪www.enterprise.conf
```

SSL certificates (if needed)

Make a copy of the SSL certificates files (certfiles) for the server, including the key file, and a copy of the certfile for the gateway, which is needed for verification if using self-signed or private CA signed certs.

Restore

Reinstall AEN-Server

See *the instructions for installing the current version of AEN-Server*.

It is not necessary to upload the license, because it will be restored with the config files.

NOTE: The new installation will generate a new password for the local $AEN_SRVC_ACCT account.

Restore Mongo database

This assumes that mongo was reinstalled as part of the reinstallation of the server component. Untar the mongo database and restore it.

NOTE: These commands must be run by $AEN_SRVC_ACCT.

```
tar -xvf aen_mongo_backup.tar
mongorestore --drop aen_main
```

NOTE: The --drop option resets the $AEN_SRVC_ACCT user password and restores the database to the exact state it was in at the time of backup. Please see the MongoDB documentation for more information about mongorestore options for Mongo 2.6.

NOTE: AEN uses Mongo 2.6 by default. If you are using a different version, consult the documentation for your version.

AEN Server config files (including License file)

Untar the tar file of all of the configuration files, including any license files.

NOTE: This command must be run by $AEN_SRVC_ACCT.

```
tar -xvf aen_server_config.tar -C /opt/wakari/
```

Make sure the files are in `/opt/wakari/wakari-server/etc/wakari/` and are owned by the `$AEN_SRVC_ACCT`.

Nginx config (if needed)

Make sure any modifications to the nginx configuration are either in `/etc/nginx/conf.d` or in `/opt/wakari/wakari-server/etc/nginx/conf.d/` with a proper symlink.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
/etc/nginx/conf.d/www.enterprise.conf -> /opt/wakari/wakari-server/etc/nginx/conf.d/
↪www.enterprise.conf
```

SSL certificates (if needed)

Move any SSL certificate files to the locations indicated in the config files.

Restart server

Restart the server application.

NOTE: This command must be run as root or with sudo.

```
service wakari-server restart
```

Gateway component steps

Backup

Config files

Create a tar file of all of the configuration files.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -cvf aen_gateway_config.tar -C /opt/wakari/ wakari-gateway/etc/wakari/
```

Custom .condarc file (if needed)

Make a copy of any `/opt/wakari/miniconda/.condarc` if it has been modified.

SSL certificates (if needed)

Make a copy of SSL certificate files for the gateway (including the key file) and the certfile for the server (needed for verification if using self-signed or private CA signed certs).

Restore

Reinstall AEN-Gateway

Setting variables and changing permissions

NOTE: These commands must be run by `$AEN_SRVC_ACCT`.

Run:

```
export AEN_SERVER=<FQDN HOSTNAME OR IP ADDRESS> # Use the real FQDN
export AEN_GATEWAY_PORT=8089
export AEN_GATEWAY=<FQDN HOSTNAME OR IP ADDRESS>  # will be needed shortly
chmod a+x aen-*.sh                # Set installer to be executable
```

NOTE: Change <FQDN HOSTNAME OR IP ADDRESS> to the actual fully qualified domain hostname or IP address.

NOTE: You must perform the entire procedure before closing the terminal to ensure the variable export persists. If the terminal is closed before successful installation, export the variables to continue with the installation.

Running the AEN gateway installer

Run:

```
sudo -E ./aen-gateway-4.2.2-Linux-x86_64.sh -w $AEN_SERVER
<license text>
...
...

PREFIX=/opt/wakari/wakari-gateway
Logging to /tmp/wakari_gateway.log
...
...
Checking server name
Please restart the Gateway after running the following command
to connect this Gateway to the AEN Server
...
```

Config files

Untar the configuration files.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -xvf aen_gateway_config.tar -C /opt/wakari
```

Verify that the files are in `/opt/wakari/wakari-gateway/etc/wakari/` and are owned by the `$AEN_SRVC_ACCT`.

Custom .condarc file (if needed)

Move the custom .condarc file to `/opt/wakari/miniconda/.condarc`.

SSL certificates (if needed)

Move any SSL certificate files to the locations indicated in the config files.

Restart gateway

Restart the gateway application.

NOTE: This command must be run as root or with sudo.

```
service wakari-gateway restart
```

Compute component steps

Backup

Config files

Create a tar file of all of the configuration files.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -cvf aen_compute_config.tar -C /opt/wakari/ wakari-compute/etc/wakari
```

Custom Changes (rare)

Manually backup any custom changes that were applied to the code. One change might be additional files in the skeleton folder:

`/opt/wakari/wakari-compute/lib/node_modules/wakari-compute-launcher/skeleton`

Create user list

AEN uses POSIX access control lists (ACLs) for project sharing, so the backup must preserve the ACL information. This is done with a script that creates a file named `users.lst` containing a list of all users that have access to projects on a given compute node. Download and run the script.

NOTE: These commands must be run by `$AEN_SRVC_ACCT`.

```
wget https://s3.amazonaws.com/continuum-airgap/misc/wk-compute-get-acl-users.py
chmod 755 wk-compute-get-acl-users.py
./wk-compute-get-acl-users.py
```

Project files

Create a tar of the projects directory with ACLs enabled. The default projects base location is `/projects`.

NOTE: This command must be run as root or with sudo.

```
tar --acls -cpvf projects.tar -C <projects base location>/*
```

Full Anaconda (option 1)

If any changes have been made to the default Anaconda installation (additional packages installed or packages removed), it is necessary to backup the entire Anaconda installation.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -cvf aen_anaconda.tar -C /opt/wakari/anaconda/*
```

If no changes have been made to the default installation of Anaconda, you may just backup the `.condarc` file and any custom environments.

Partial Anaconda (option 2)

Custom `.condarc` file

Make a copy of `/opt/wakari/anaconda/.condarc`.

Custom environments (if needed)

Create a tar file of any custom shared environments.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -cvf aen_compute_envs.tar -C /opt/wakari/ anaconda/envs
```

NOTE: If no custom shared environments have been created, the `envs` folder will not be present.

Restore

Reinstall AEN-Compute

Setting variables and changing permissions

NOTE: These commands must be run by `$AEN_SRVC_ACCT`.

Run:

```
export AEN_SERVER=<FQDN HOSTNAME OR IP ADDRESS> # Use the real FQDN
chmod a+x aen-*.sh                   # Set installer to be executable
```

NOTE: Change <FQDN HOSTNAME OR IP ADDRESS> to the actual fully qualified domain hostname or IP address.

NOTE: You must perform the entire procedure before closing the terminal to ensure the variable export persists.

Running the AEN compute installer

Run:

```
sudo -E ./aen-compute-4.2.2-Linux-x86_64.sh -w $AEN_SERVER
...
...
PREFIX=/opt/wakari/wakari-compute
Logging to /tmp/wakari_compute.log
Checking server name
...
...
Initial clone of root environment...
Starting Wakari daemons...
installation finished.
Do you wish the installer to prepend the wakari-compute install location
to PATH in your /root/.bashrc ? [yes|no]
[no] >>> yes
```

Config files

Untar the config files.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
tar -xvf aen_compute_config.tar -C /opt/wakari
```

NOTE: Verify that they are located in `/opt/wakari/wakari-compute/etc/wakari` and are owned by the `$AEN_SRVC_ACCT`.

Custom changes (rare)

Manually restore any custom changes you saved in the backup section. If there are changes in the skeleton directory, these files must be world readable or projects will refuse to start.

Create users

NOTE: Only create users with these instructions if your Linux machine is not bound to LDAP.

In order for the ACLs to be set properly on restore, all users that have permissions to the files must be available on the machine. Ask your system administrator for the proper way to do this for your system, such as using the "useradd" tool. A list of users that are needed was created in the backup process as a file named `users.lst`.

A process similar to the following `useradd` example will be suitable for most Linux systems.

NOTE: This command must be run by `$AEN_SRVC_ACCT`.

```
xargs -0 -n 1 useradd --user-group < users.lst
```

Project files

Create the projects directory in the location specified in projectRoot in `wk-compute-launcher-config.json`.

NOTE: By default this directory is /projects.

Then untar the projects directory with ACLs.

NOTE: This command must be run as root or with sudo:

```
tar --acls -xpvf projects.tar -C <projects base location>
```

Full Anaconda (option 1)

If you did a full backup of the full Anaconda installation, untar this file to /opt/wakari/anaconda.

NOTE: This command must be run by $AEN_SRVC_ACCT.

```
tar -xvf aen_anaconda.tar -C /opt/wakari
```

Partial Anaconda (option 2)

Restore the custom .condarc file.

If you did a partial backup of the Anaconda installation, move the copy of the .condarc file to /opt/wakari/anaconda/.condarc.

Custom environments (if needed)

Untar any custom environments that were created to /opt/wakari/anaconda/envs.

NOTE: This command must be run by $AEN_SRVC_ACCT.

```
tar -xvf aen_compute_envs.tar -C /opt/wakari
```

Restart compute node

Restart the compute-launcher application.

NOTE: This command must be run as root or with sudo.

```
service wakari-compute restart
```

Viewing a list of admin commands

A user who is promoted to administrator can access administrator commands to perform advanced administrator tasks.

NOTE: Utility files are owned by, and should only be executed by, the AEN user who owns the files.

To display a list of all administrator commands:

```
ls -al /opt/wakari/wakari-server/bin/wk-*
```

Viewing help for admin commands

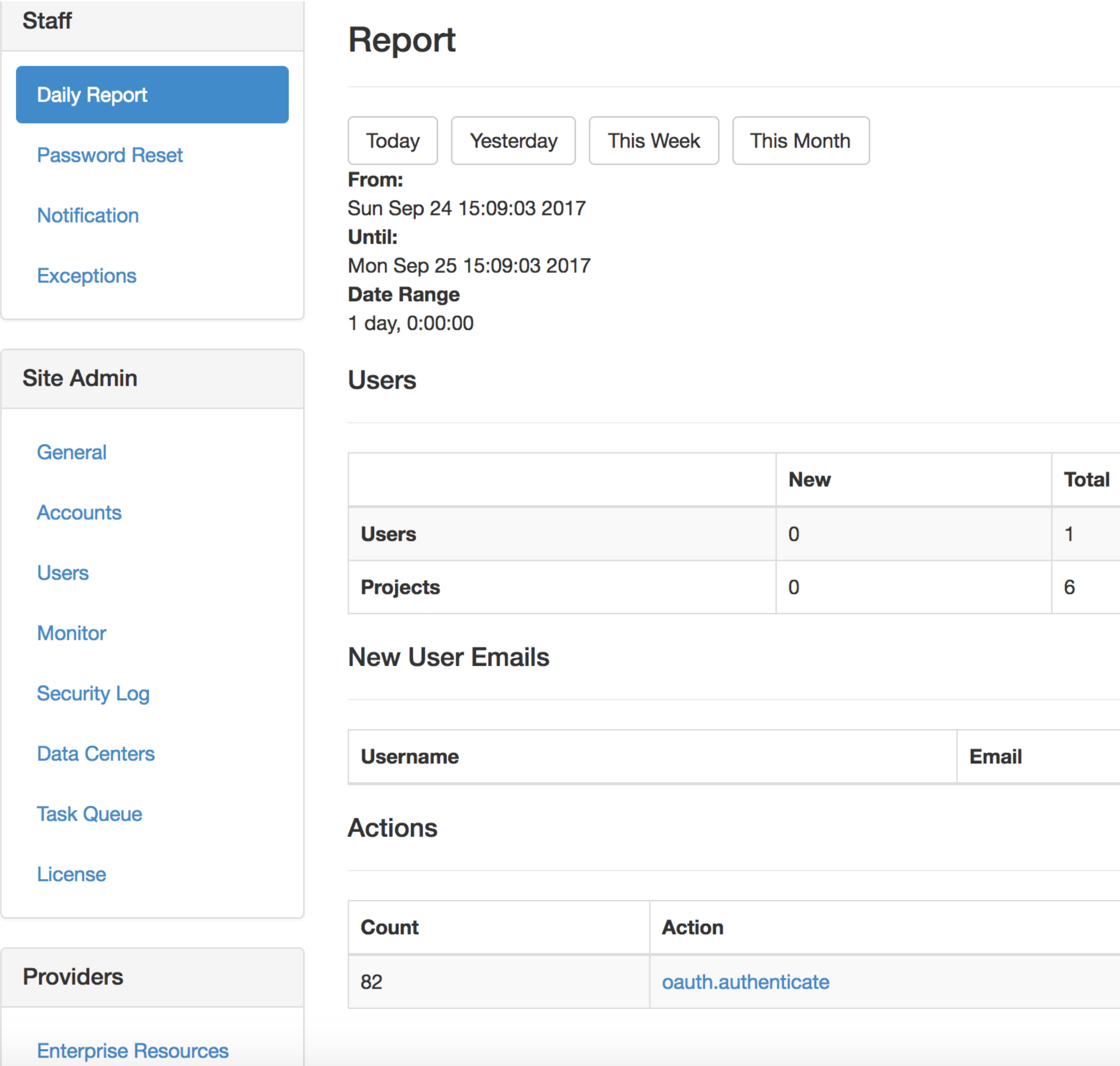
To view help information for command, run the command followed by `-h` or `--help`.

EXAMPLE: To view help for the `remove-user` command:

```
/opt/wakari/wakari-server/bin/wk-server-admin remove-user -h
/opt/wakari/wakari-server/bin/wk-server-admin remove-project -h
```

Running daily reports

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Staff** menu, select Daily Report:



Report

From:
Sun Sep 24 15:09:03 2017
Until:
Mon Sep 25 15:09:03 2017
Date Range
1 day, 0:00:00

Users

	New	Total
Users	0	1
Projects	0	6

New User Emails

Username	Email

Actions

Count	Action
82	oauth.authenticate

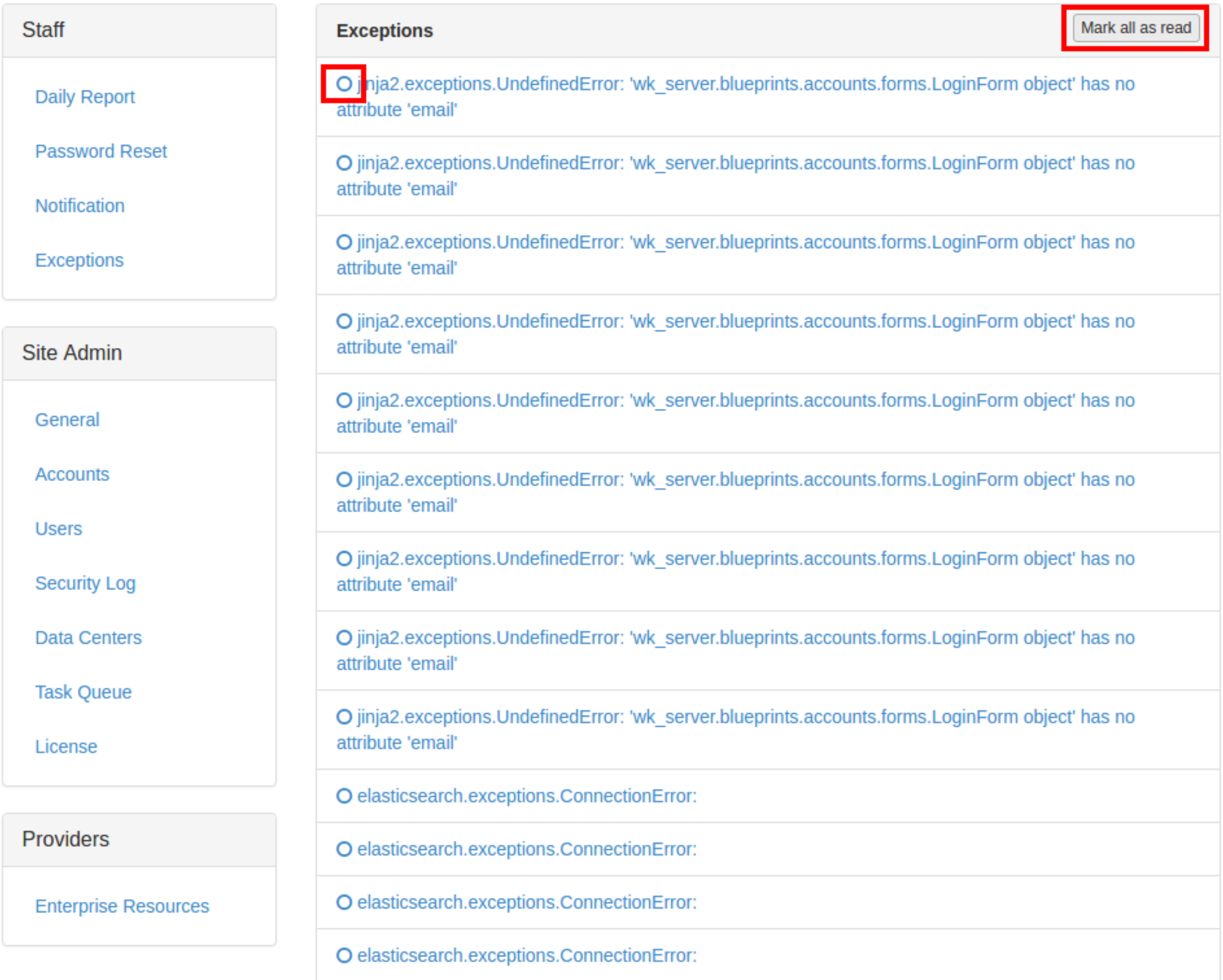
The Report section displays the following:

- Users—The number of users and projects.
- New User Emails—If *open registration is enabled*, the user names and emails for new users.
- Actions—The actions—projects created, projects updated, user authentications and added users—that have occurred in during the selected time frame—today, yesterday, this week, or this month.

Viewing system errors

When an error occurs, a red dot is displayed in the AEN navigation bar next to the Admin link. The red dot is removed when all exceptions are marked as "read."

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Staff** menu, select Exceptions:



○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ jinja2.exceptions.UndefinedError: 'wk_server.blueprints.accounts.forms.LoginForm object' has no attribute 'email'

○ elasticsearch.exceptions.ConnectionError:

○ elasticsearch.exceptions.ConnectionError:

○ elasticsearch.exceptions.ConnectionError:

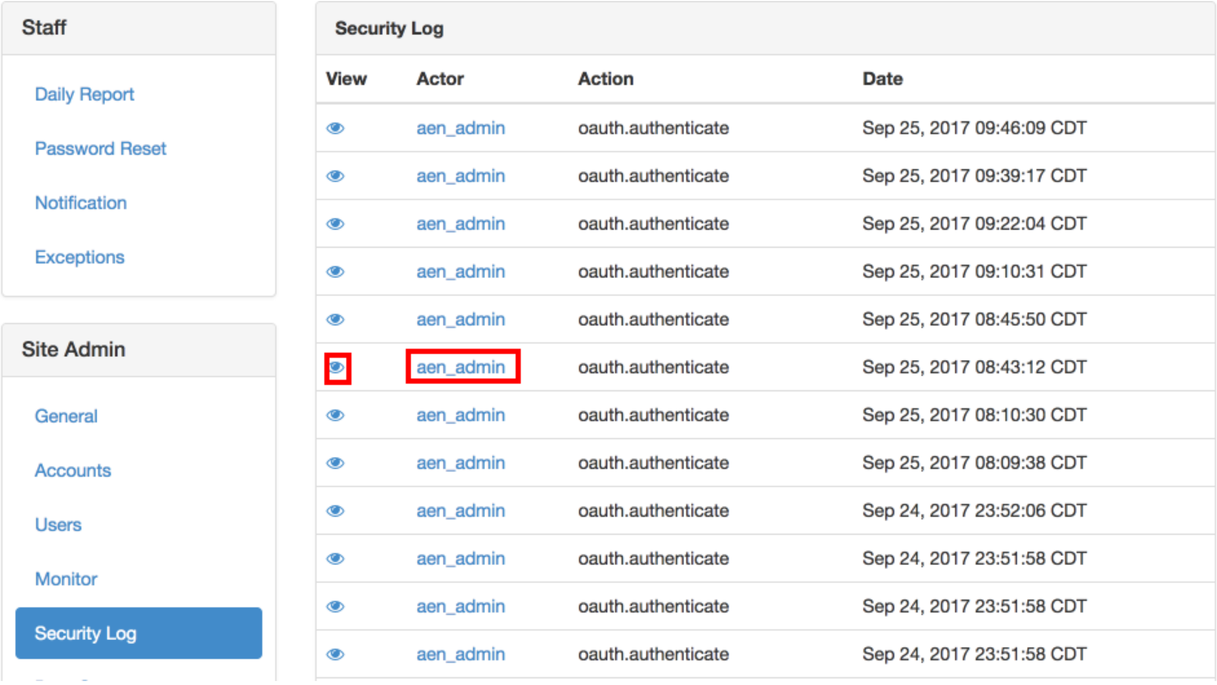
○ elasticsearch.exceptions.ConnectionError:

The Exceptions section lists all errors that have occurred while AEN is running.

3. To see the details of an error, click the radio button next to the error. This also marks the error as "read."
4. To mark all errors as read without reviewing each one, click the Mark all as read button.

Viewing security errors

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Security Log:



Security Log

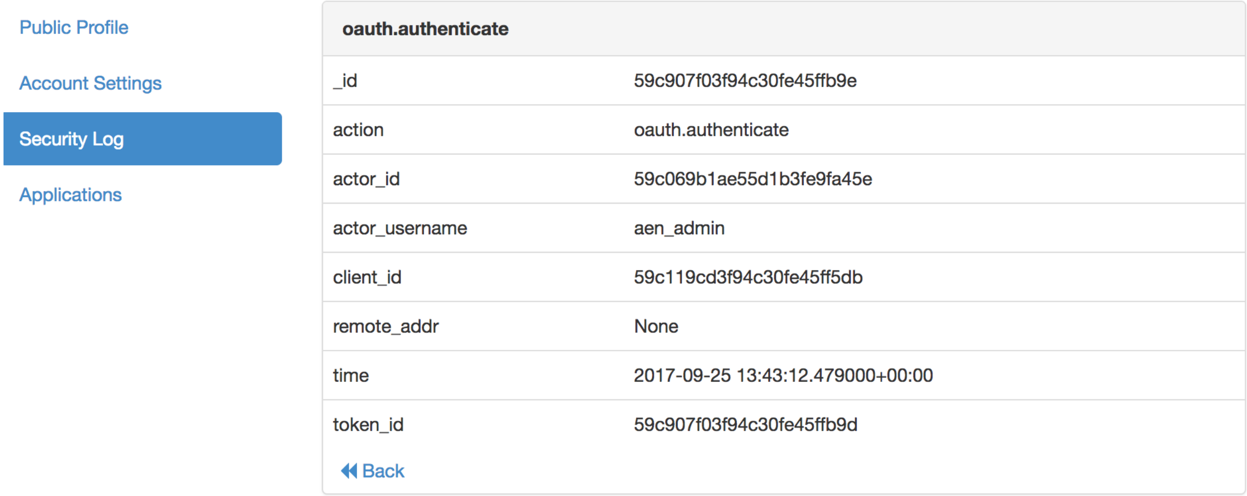
View	Actor	Action	Date
👁	aen_admin	oauth.authenticate	Sep 25, 2017 09:46:09 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 09:39:17 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 09:22:04 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 09:10:31 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 08:45:50 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 08:43:12 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 08:10:30 CDT
👁	aen_admin	oauth.authenticate	Sep 25, 2017 08:09:38 CDT
👁	aen_admin	oauth.authenticate	Sep 24, 2017 23:52:06 CDT
👁	aen_admin	oauth.authenticate	Sep 24, 2017 23:51:58 CDT
👁	aen_admin	oauth.authenticate	Sep 24, 2017 23:51:58 CDT
👁	aen_admin	oauth.authenticate	Sep 24, 2017 23:51:58 CDT

The Security Log section lists all errors that have occurred that could potentially affect AEN security.

3. To view a user's profile page, click their username in the Actor column.
4. To see the details of an error, click the Eye icon next to the error.

   The error details are displayed:

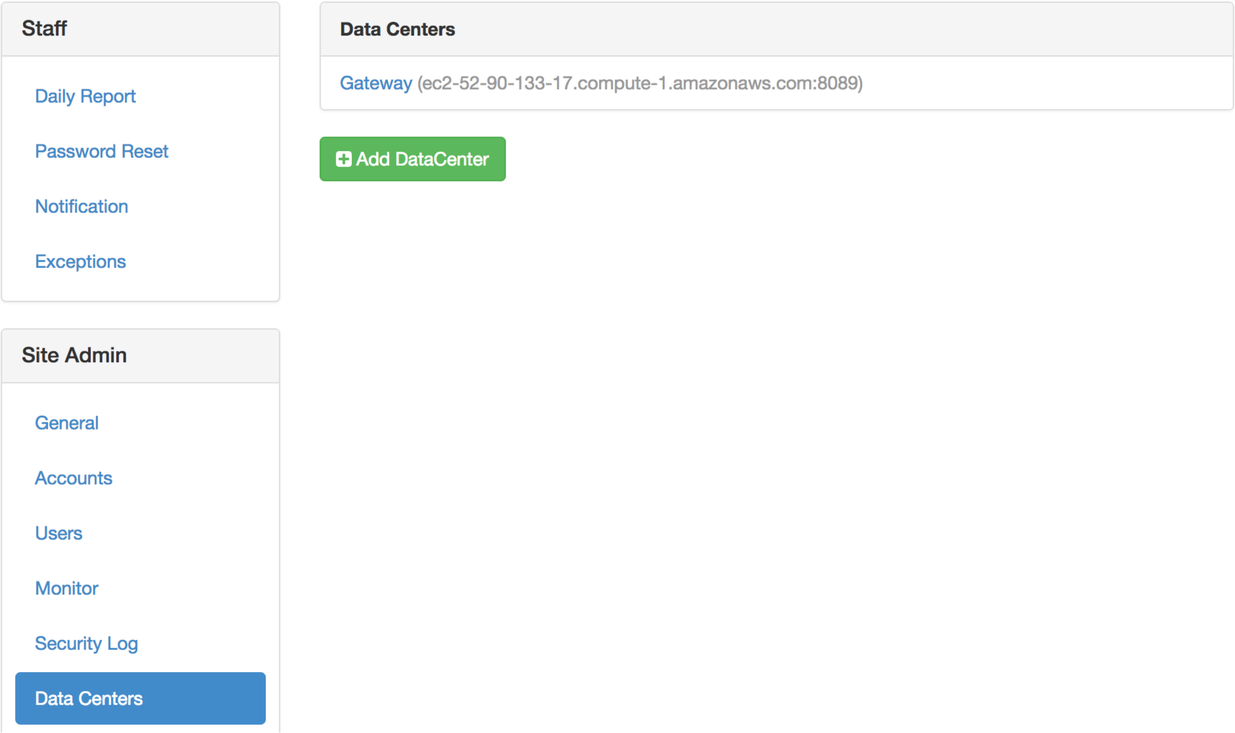
5. To close the error details, click the Back link.



oauth.authenticate	
_id	59c907f03f94c30fe45ffb9e
action	oauth.authenticate
actor_id	59c069b1ae55d1b3fe9fa45e
actor_username	aen_admin
client_id	59c119cd3f94c30fe45ff5db
remote_addr	None
time	2017-09-25 13:43:12.479000+00:00
token_id	59c907f03f94c30fe45ffb9d

Managing data centers

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Data Centers:



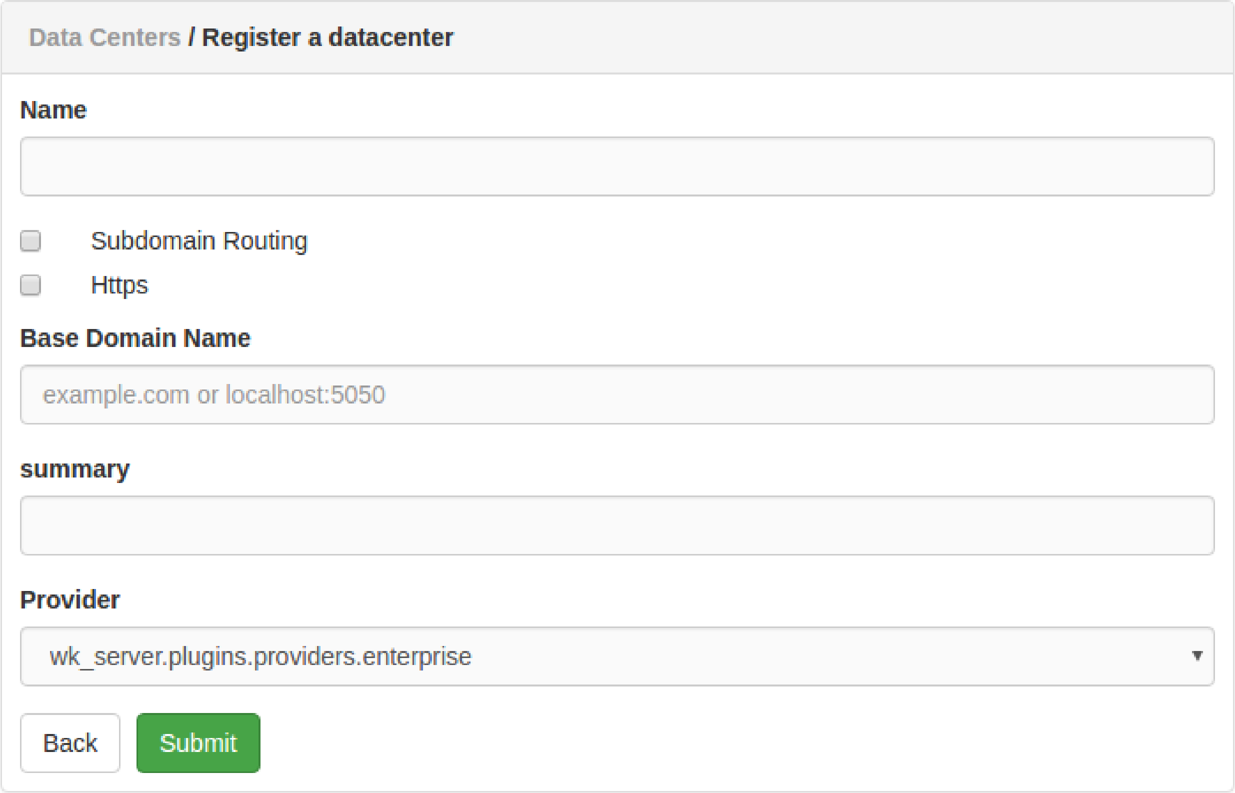
Data Centers

Gateway (ec2-52-90-133-17.compute-1.amazonaws.com:8089)

The Data Centers section displays current data center information.

Adding a data center

1. Click the Add DataCenter button to display the the Register a datacenter form.
2. In the Name box, type a Name for the new data center:



Data Centers / Register a datacenter

Name

□ Subdomain Routing

□ Https

Base Domain Name

example.com or localhost:5050

summary

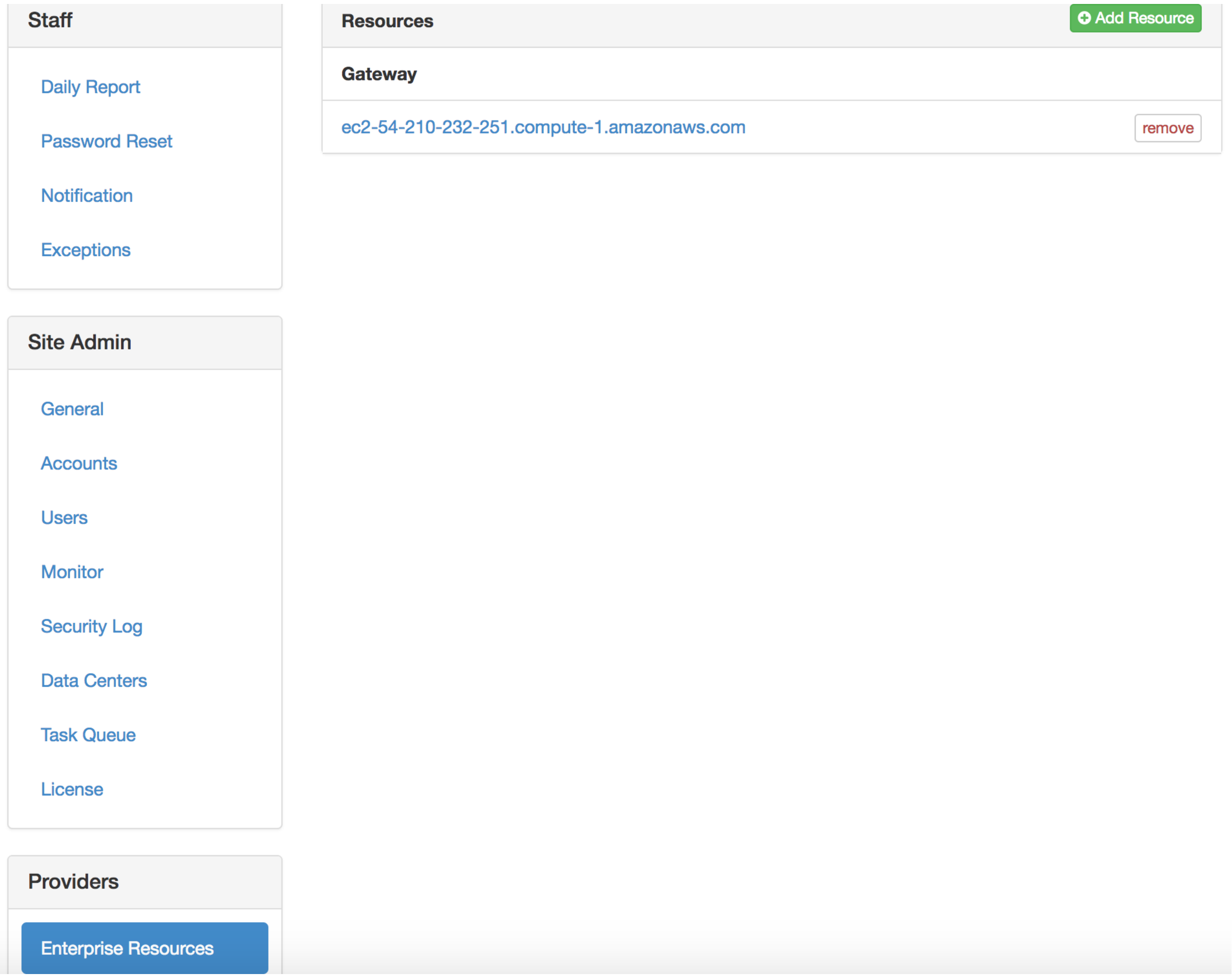
Provider

wk_server.plugins.providers.enterprise

3. Select the Subdomain Routing and/or Https checkboxes.
4. In the Base Domain Name box, type the base domain name.
5. In the Summary box, type a description of the data center.
6. In the Provider list, select a provider.
7. Click the Submit button.

Managing enterprise resources

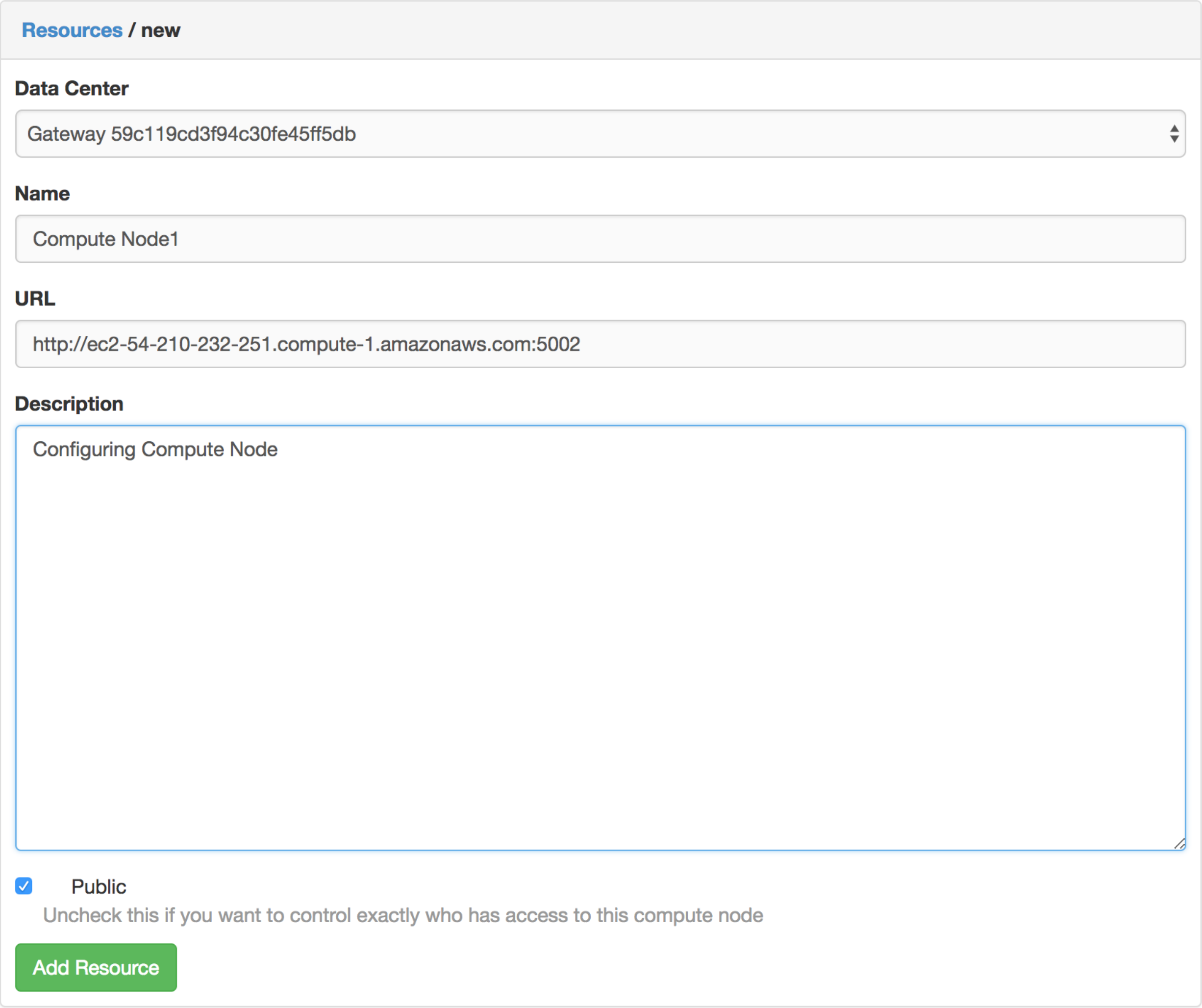
1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Providers** menu, select Enterprise Resources:


Staff
Daily Report
Password Reset
Notification
Exceptions
Site Admin
General
Accounts
Users
Monitor
Security Log
Data Centers
Task Queue
License
Providers
Enterprise Resources
Resources
Add Resource
Gateway
ec2-54-210-232-251.compute-1.amazonaws.com
remove

The Resources section lists your existing cloud and local resources.

Adding a resource

1. Click the Add Resource button to open the new resource form.
2. Complete the form:



Resources / new

Data Center

Gateway 59c119cd3f94c30fe45ff5db

Name

Compute Node1

URL

http://ec2-54-210-232-251.compute-1.amazonaws.com:5002

Description

Configuring Compute Node

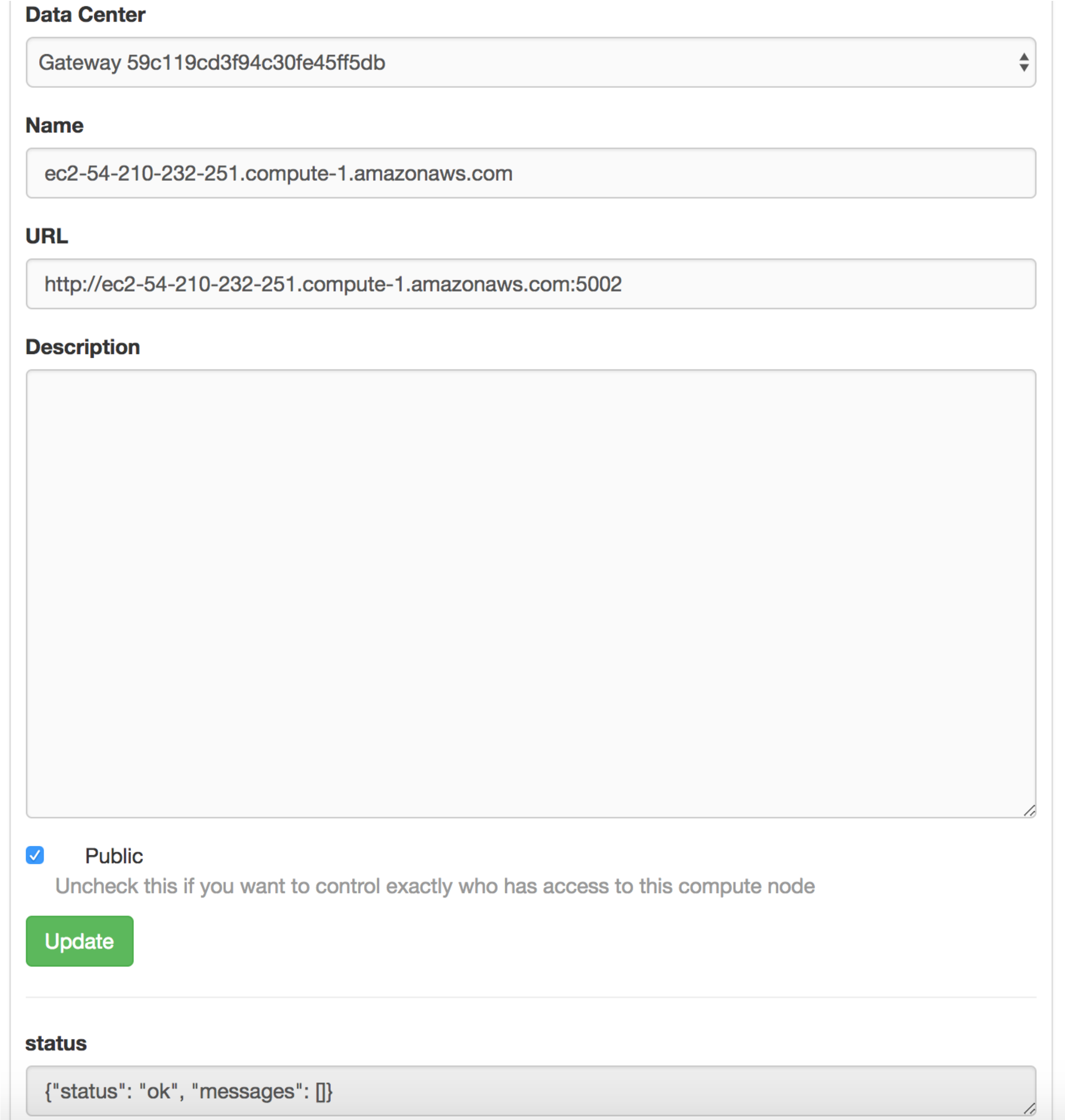
☑ Public

Uncheck this if you want to control exactly who has access to this compute node

3. Click the Add Resource button.

Viewing or changing the resource details

1. Click a resource name to open the Local Resource form.
2. If necessary, change the resource details:



Data Center

Gateway 59c119cd3f94c30fe45ff5db

Name

ec2-54-210-232-251.compute-1.amazonaws.com

URL

http://ec2-54-210-232-251.compute-1.amazonaws.com:5002

Description

☑ Public

Uncheck this if you want to control exactly who has access to this compute node

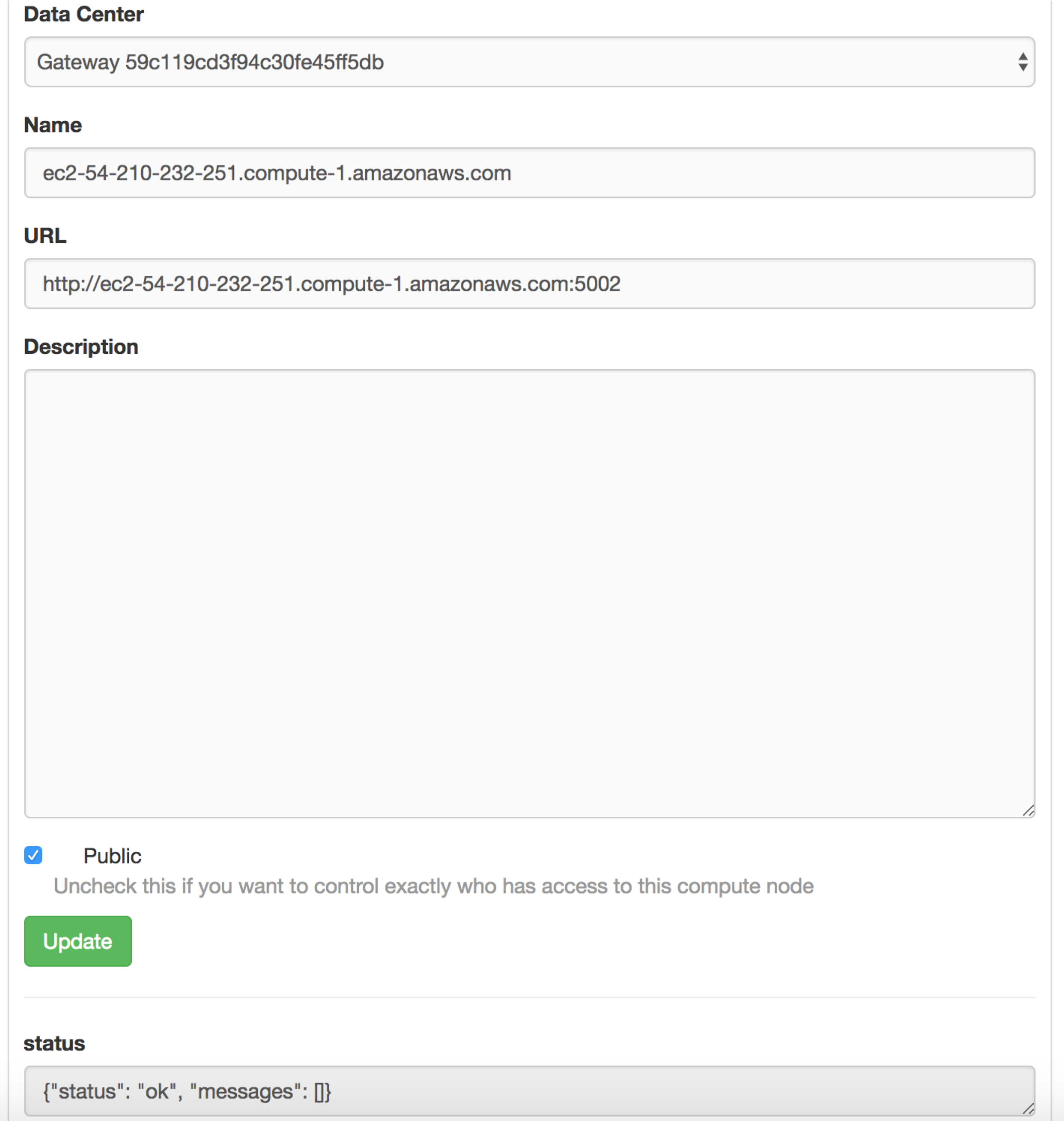
status

{"status": "ok", "messages": []}

3. Click the Update button.

Making a node public or private

1. Click the resource name to open the Local Resource form.
2. Select or clear the Public checkbox:



Data Center

Gateway 59c119cd3f94c30fe45ff5db

Name

ec2-54-210-232-251.compute-1.amazonaws.com

URL

http://ec2-54-210-232-251.compute-1.amazonaws.com:5002

Description

☑ Public

Uncheck this if you want to control exactly who has access to this compute node

status

{"status": "ok", "messages": []}

3. Click the Update button.

Removing a resource

Click the Remove button next to the resource you want to remove.

NOTE: When you remove a resource assigned to a project, the project becomes orphaned. To fix an orphaned project, *move the project to a valid Compute Resource*.

Managing services

The tasks on this page assume that the 3 AEN nodes are installed in the following locations:

- Server–`/opt/wakari/wakari-server/`.
- Gateway–`/opt/wakari/wakari-gateway/`.
- Compute-Launcher–`/opt/wakari/wakari-compute/`.

Checking the status of server node processes

1. Run:

```
# service wakari-server status
wk-server                    RUNNING    pid 20758, uptime 5 days, 0:30:23
worker                       RUNNING    pid 20757, uptime 5 days, 0:30:23
```

OR

```
root@server # ps -Hu wakari
  PID TTY          TIME CMD
20756 ?        00:02:26 .supervisord
20757 ?        00:05:58   mtq-worker
20758 ?        00:00:08   wk-server
```

(continues on next page)

(continued from previous page)

```
20765 ?        00:02:00     wk-server
20766 ?        00:01:55     wk-server
20767 ?        00:02:20     wk-server
20770 ?        00:02:02     wk-server
```

2. Run:

```
root@server # service nginx status
nginx (pid  26303) is running...
```

For more information on server processes, see *Server processes*.

Checking the status of gateway node processes

Run:

```
# service wakari-gateway status
wk-gateway                      RUNNING    pid 1137, uptime 5 days, 1:59:28
```

OR

```
root@gateway # ps -Hu wakari
  PID TTY          TIME CMD
 1136 ?        00:01:59 .supervisord
 1137 ?        00:00:02   wk-gateway
```

For more information on gateway processes, see *Gateway processes*.

Checking the status of compute node processes

Run:

```
# service wakari-compute status
wk-compute                      RUNNING    pid 22050, uptime 3 days, 1:03:19
```

OR

```
root@compute # ps -Hu wakari
  PID TTY          TIME CMD
 1150 ?        00:02:01 .supervisord
 1152 ?        00:00:01   wk-compute
```

For more information on compute node processes, see *Compute processes*.

Starting AEN services

Services should start automatically both when they are first installed and at any point when the system is restarted.

If you need to manually start an AEN service, you must start each node independently, because they may be running on separate machines.

NOTE: The process is basically the same for each node, but the path to the correct commands vary.

To manually start a service:

- On the server node, run:

  ```
  service wakari-server start
  ```

- On the gateway node, run:

  ```
  service wakari-gateway start
  ```

- On a compute node, run:

  ```
  service wakari-compute start
  ```

Verifying that AEN services are set to start with the system

To verify that AEN services are set up to start automatically:

1. Run the following command on each node:

   ```
   chkconfig --list | grep wakari
   ```

2. If services are missing, add them:

   ```
   chkconfig --add [wakari-server|wakari-gateway|wakari-compute]
   ```

3. *Restart the services*.

Stopping AEN services

CAUTION: Do not stop or kill supervisord without first stopping wk-compute and any other processes that use it.

You must stop services on each node independently, because they may be running on separate machines.

To stop an AEN service:

- On the server node, run:

  ```
  service wakari-server stop
  ```

- On the gateway node, run:

  ```
  service wakari-gateway stop
  ```

- On a compute node, run:

  ```
  service wakari-compute stop
  ```

  Compute nodes may have running processes that are not automatically stopped. To stop them, run:

  ```
  sudo /opt/wakari/wakari-compute/bin/wk-compute-apps kill-all
  ```

Restarting AEN services

- On the server node, run:

```
service wakari-server restart
```

- On the gateway node, run:

```
service wakari-gateway restart
```

- On a compute node, run:

```
service wakari-compute restart
```

Identifying extraneous processes

To get a complete list of the processes running under the wakari user account, run `ps -Hu wakari`.

EXAMPLE:

```
root@server # ps -Hu wakari
   PID TTY          TIME CMD
 20756 ?        00:02:26 .supervisord
 20757 ?        00:05:58   mtq-worker
 20758 ?        00:00:08   wk-server
 20765 ?        00:02:00     wk-server
 20766 ?        00:01:55     wk-server
 20767 ?        00:02:20     wk-server
 20770 ?        00:02:02     wk-server

 root@server # ps -f -C nginx
 UID     PID  PPID  C STIME TTY     TIME CMD
 root  26303     1  0 12:18 ?   00:00:00 nginx: master process /usr/sbin/nginx -c /
↪etc/nginx/nginx.conf
 nginx 26305 26303  0 12:18 ?   00:00:00 nginx: worker process

 root@gateway # ps -Hu wakari
   PID TTY          TIME CMD
  1136 ?        00:01:59 .supervisord
  1137 ?        00:00:02   wk-gateway

 root@compute # ps -Hu wakari
   PID TTY          TIME CMD
  1150 ?        00:02:01 .supervisord
  1152 ?        00:00:01   wk-compute
```

- wk-server, wk-gateway and wk-compute should have PIDs reported by supervisorctl.
- The nginx master process should have a PID reported by service nginx status.
- If you have installed more than one AEN node on a single machine, the processes from all of the installed nodes should be displayed for that machine.
- On compute node(s), any AEN applications currently being run by users will be present.

  EXAMPLE:

```
root@compute # ps -Hu wakari
   PID TTY          TIME CMD
  1150 ?        00:00:00 .supervisord
  1152 ?        00:00:00   wk-compute
```

(continues on next page)

(continued from previous page)

```
1340 ?        00:00:00 bash
1341 ?        00:00:00   notebookwrapper
```

Removing extraneous processes

If extra wk-server, wk-gateway, wk-compute, or supervisord processes are present, use the `kill` command to remove them to prevent issues with AEN.

You can safely *restart* any process that you remove in error.

Making sure NGINX and MongoDB are running

In order for AEN to run, the dependencies *mongodb* and *nginx* must be up and running. If either of these fail to start, AEN will not be served on port 80.

Check if *nginx* and *mongod* are both running (RHEL 6x):

```
$ sudo service nginx status
nginx (pid  25956) is running...

$ sudo service mongod status
mongod (pid 25928) is running...
```

If either of these failed to start, tail the log files. The default location of log files is:

```
$ tail -n 50 /var/log/mongodb/mongod.log

# nginx errors reported in error.log
$ tail -n 50 /var/log/nginx/error.log
```

Viewing, terminating, and relaunching applications

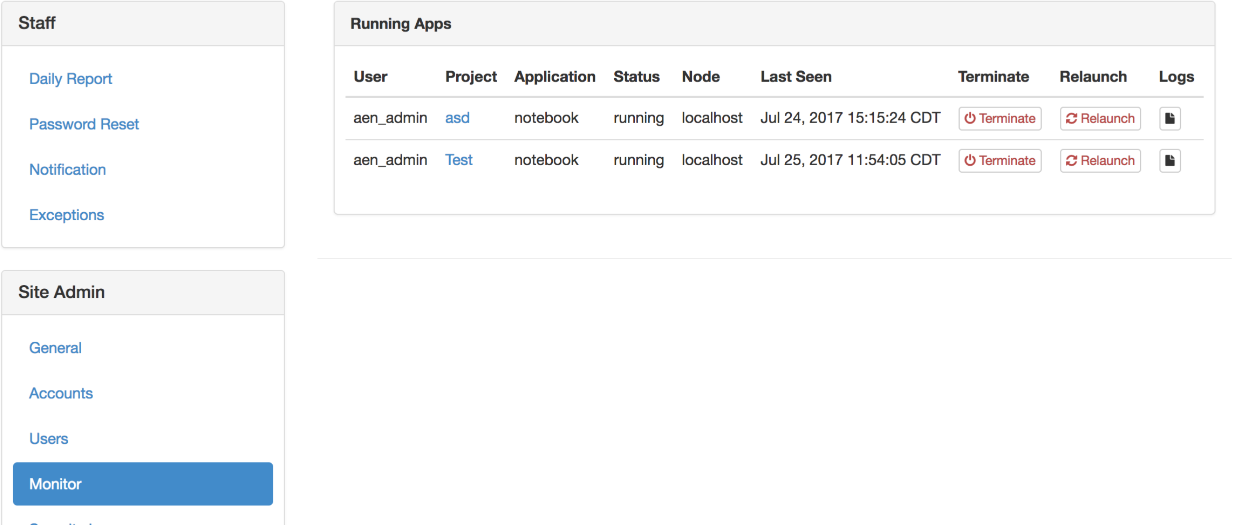
1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Monitor:

   The Monitor menu lists started applications by user and project.

   The list includes columns for the application name, current running status, running node and last seen date.
3. Use the buttons to terminate or relaunch an application.
4. To view an application's logs, click the Logs button with the document icon.

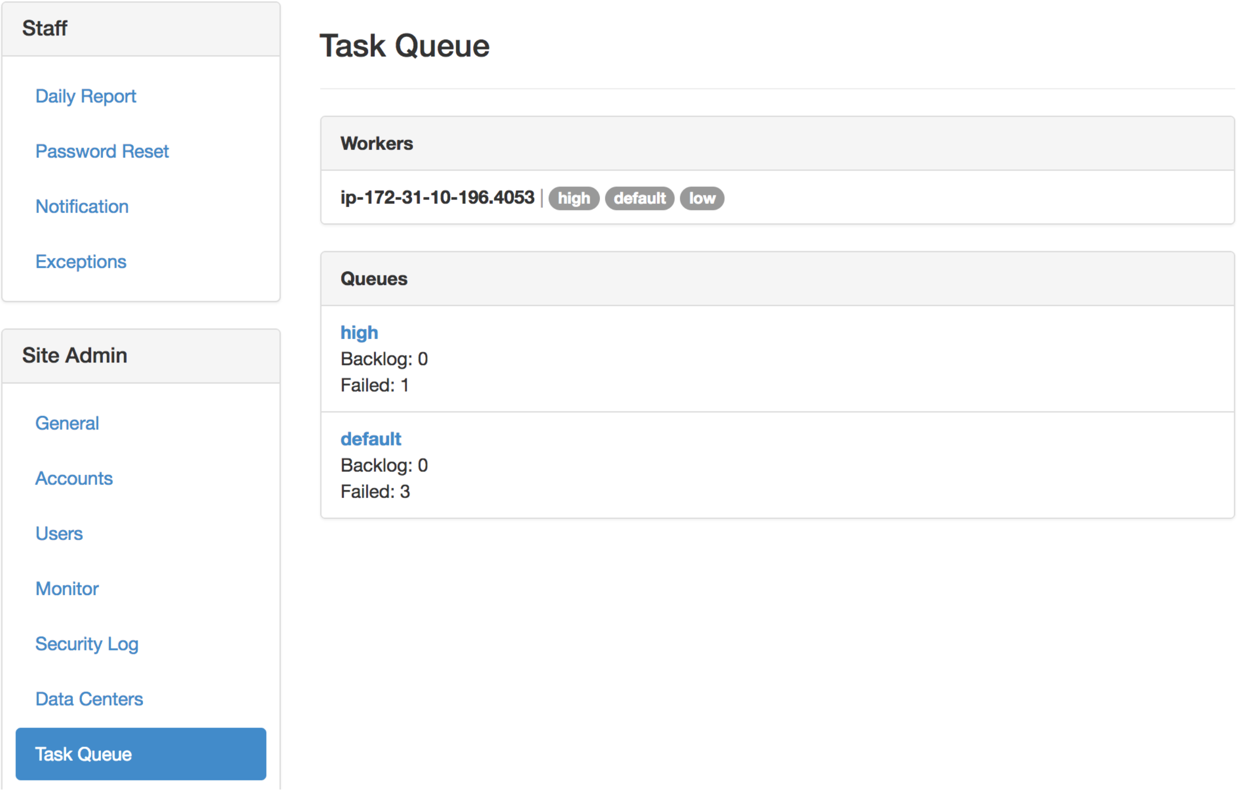
Viewing the task queue

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Task Queue:



Running Apps

User	Project	Application	Status	Node	Last Seen	Terminate	Relaunch	Logs
aen_admin	asd	notebook	running	localhost	Jul 24, 2017 15:15:24 CDT	⏻ Terminate	↻ Relaunch	🗎
aen_admin	Test	notebook	running	localhost	Jul 25, 2017 11:54:05 CDT	⏻ Terminate	↻ Relaunch	🗎



Task Queue

Workers

ip-172-31-10-196.4053 | high default low

Queues

high
Backlog: 0
Failed: 1

default
Backlog: 0
Failed: 3

The Workers section lists the workers in the task queue and whether each worker is set at high, default or low priority.

The Queues section provides information on the default and high priority queues.

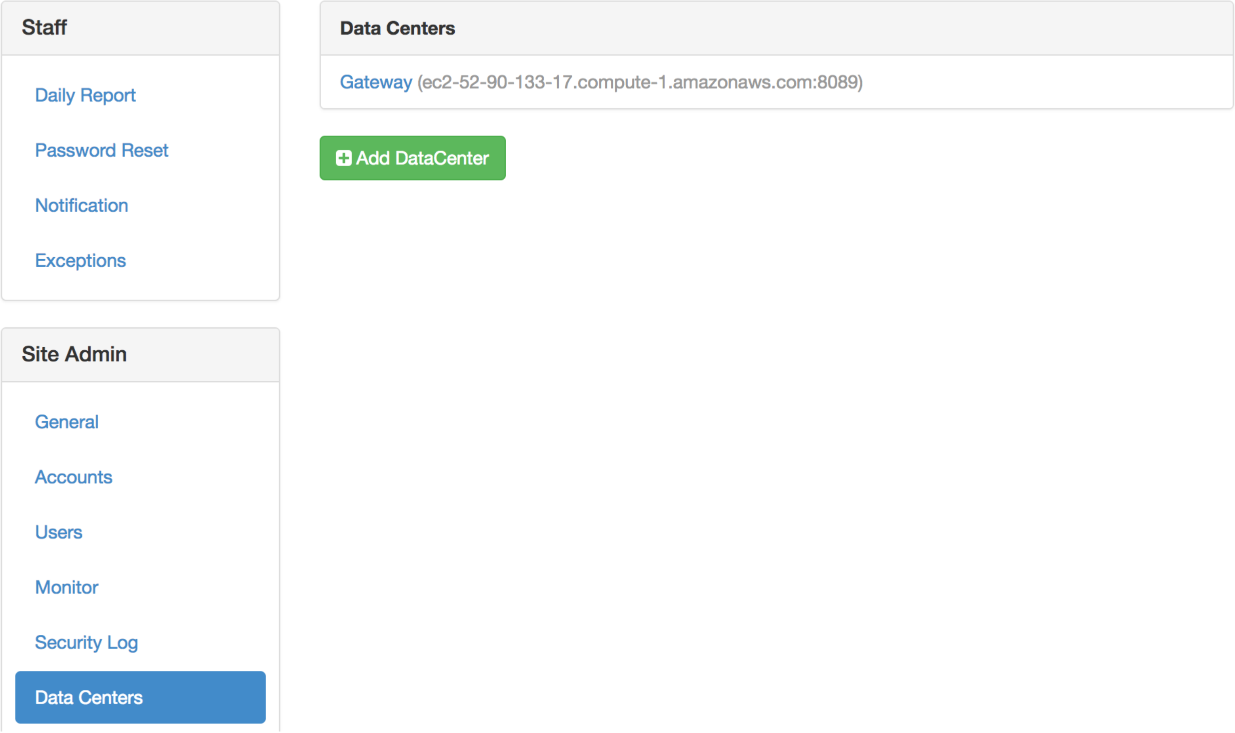
3. To view all the tasks in a particular queue, in the Queues section, click the queue name.

Checking node connections

When the AEN nodes cannot communicate with each other as intended, it can cause issues with you AEN platform installation.

Verifying server to gateway connectivity

1. On the server, in the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select Data Centers:



Data Centers

Gateway (ec2-52-90-133-17.compute-1.amazonaws.com:8089)

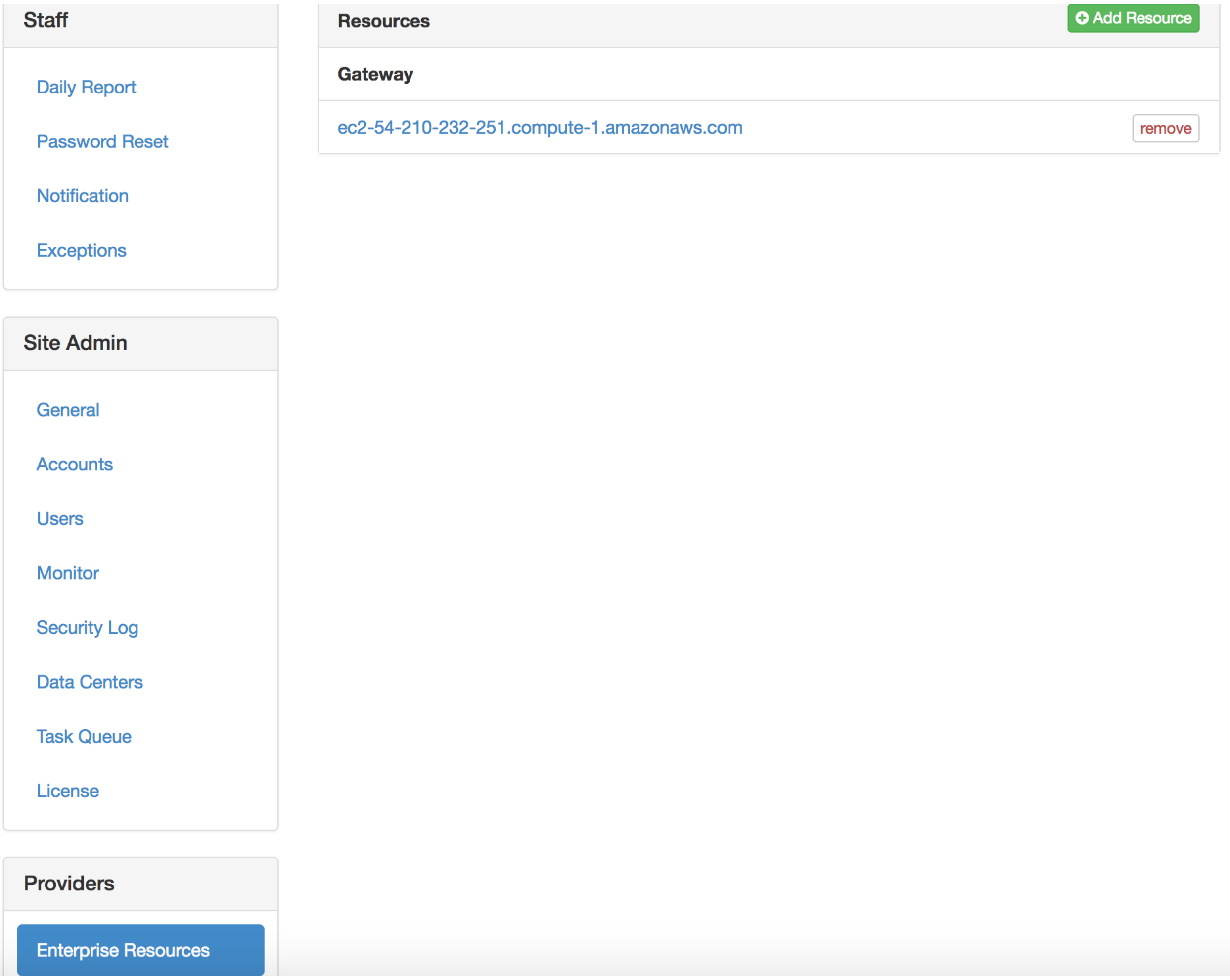
3. For each data center in the list, check connectivity from the server to that gateway.

   EXAMPLE: The gateway in this example is `http://gateway.example.com:8089`:

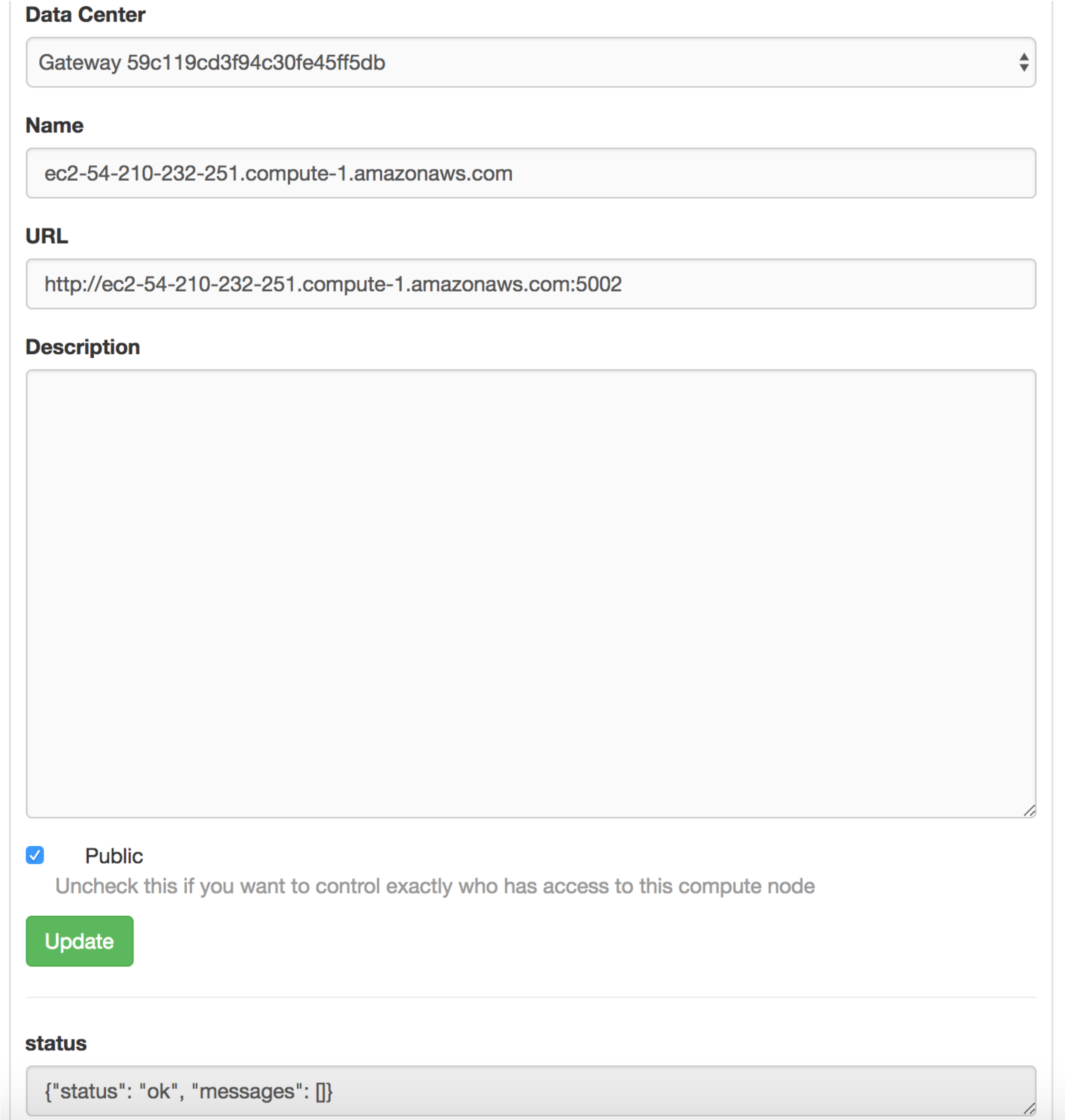
```
root@server # curl --connect-timeout 5 http://gateway.example.com:8089 > /dev/null
```

Verifying gateway to compute node connectivity

1. On the server, in the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Providers** menu, select Enterprise Resources:



3. Open each compute node in the Resources section.
4. Verify that the contents of the URL field begin with either `http` or `https`.



Data Center

Gateway 59c119cd3f94c30fe45ff5db

Name

ec2-54-210-232-251.compute-1.amazonaws.com

URL

http://ec2-54-210-232-251.compute-1.amazonaws.com:5002

Description

☑ Public

Uncheck this if you want to control exactly who has access to this compute node

status

{"status": "ok", "messages": []}

5. Check connectivity to that URL from the corresponding gateway.

   EXAMPLE: The gateway in this example is `http://gateway.example.com:8089`:

   ```
   root@gateway # curl --connect-timeout 5 http://compute.example.com:5002 > /dev/
   ↪null
   ```

Verifying gateway to server connectivity

The gateway-to-server path is used by the gateway configuration command `wk-gateway-configure`.

1. Verify that the gateway is linked to the correct server in the configuration file.
2. Verify that the full server URL is specified.
3. Check connectivity to the server:

   ```
   root@gateway # grep WAKARI_SERVER /opt/wakari/wakari-gateway/etc/wakari/wk-
   ↪gateway-config.json
     "WAKARI_SERVER": "http://wakari.example.com",

   root@gateway # curl --connect-timeout 5 http://wakari.example.com > /dev/null
   root@gateway # curl --connect-timeout 5 http://error.example.com > /dev/null
   curl: (7) Failed to connect to error.example.com port 80: Connection refused
   ```

4. If a connection fails:

   (a) Ensure that gateways (data centers) and compute nodes (Enterprise Resources) are correctly configured on the server.

   (b) Verify that processes are listening on the configured ports:

   ```
   $ sudo netstat -nplt
   Active Internet connections (only servers)
   Proto Recv-Q Send-Q Local Address   Foreign Address State  PID/Program
   tcp        0      0 *:80            *:*             LISTEN 26409/nginx
   tcp        0      0 *:22            *:*             LISTEN 986/sshd
   tcp        0      0 127.0.0.1:25    *:*             LISTEN 1063/master
   tcp        0      0 *:5000          *:*             LISTEN 26192/python
   tcp        0      0 127.0.0.1:27017 *:*             LISTEN 29261/mongod
   tcp        0      0 *:22            *:*             LISTEN 986/sshd
   tcp        0      0 127.0.0.1:25    *:*             LISTEN 1063/master
   ```

   (c) Check the firewall setting and logs on both hosts to ensure that packets are not being blocked or discarded.

Verifying and tuning search indexing

For search indexing to work correctly, a compute node must be able to communicate with the server. To verify this:

1. Run:

   ```
   curl -m 5 $AEN_SERVER > /dev/null
   ```

2. Verify that there are sufficient inotify watches available for the number of subdirectories within the project root file system:

```
cat /proc/sys/fs/inotify/max_user_watches
```

NOTE: Some Linux distributions default to a low number of watches, which may prevent the search indexer from monitoring project directories for changes.

3. If necessary, increase the number of watches:

```
echo fs.inotify.max_user_watches=100000 | sudo tee -a /etc/sysctl.conf && sudo␣
↪sysctl -p
```

4. Verify that there are sufficient inotify user instances available—at least one per project:

```
cat /proc/sys/fs/inotify/max_user_instances
```

5. If necessary, increase the number of inotify user instances:

```
echo fs.inotify.max_user_instances=1000 | sudo tee -a /etc/sysctl.conf && sudo␣
↪sysctl -p
```

Changing the AEN server URL

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select General:

3. In the Wakari Server box, type the main URL where the site can be viewed.
4. Click the Update button.

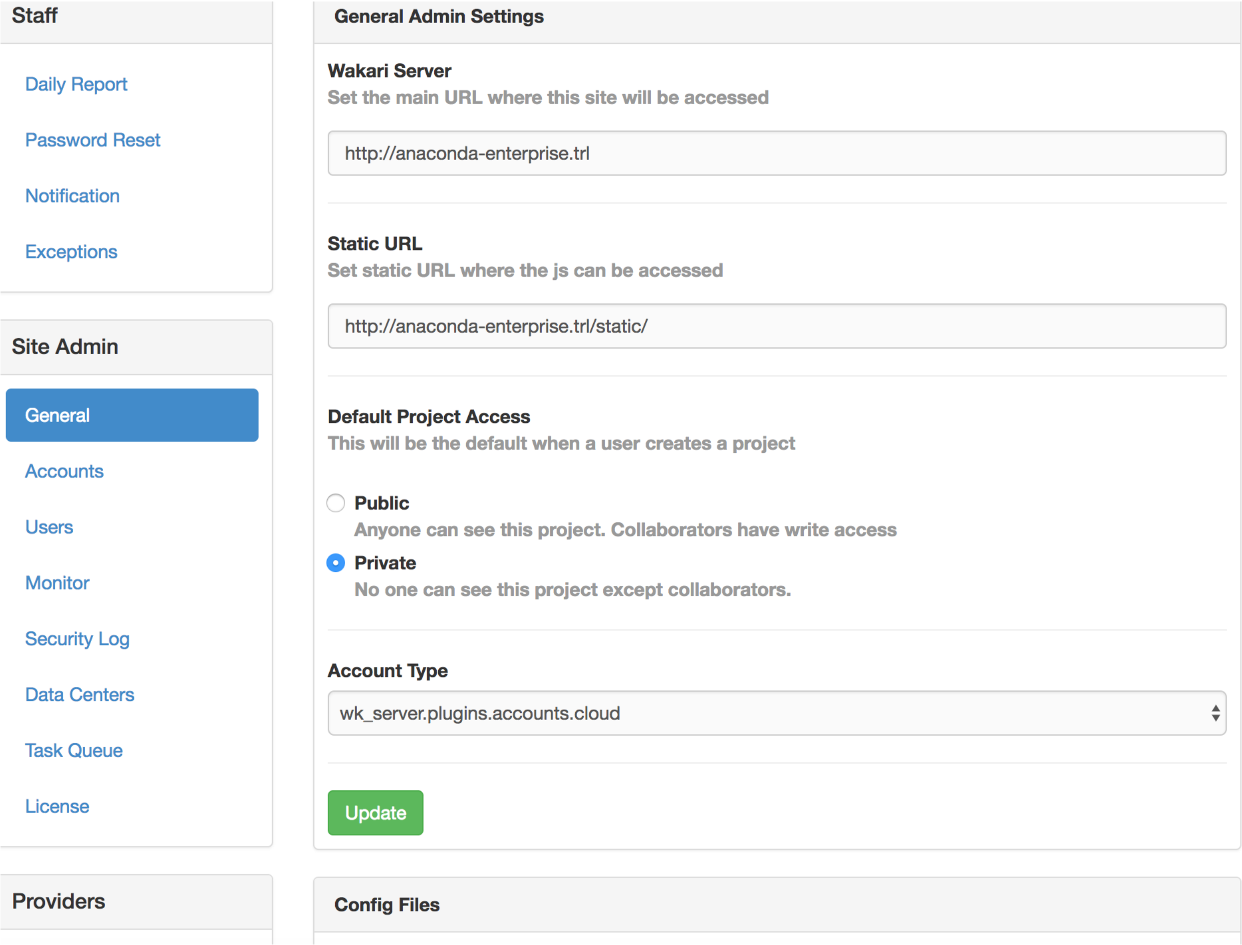
Changing the static URL for JavaScript files

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select General:

3. In the Static URL box, type the static URL where JavaScript files can be accessed.
4. Click the Update button.

Changing the AEN account type

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select General:



General Admin Settings

Wakari Server

Set the main URL where this site will be accessed

http://anaconda-enterprise.trl

Static URL

Set static URL where the js can be accessed

http://anaconda-enterprise.trl/static/

Default Project Access

This will be the default when a user creates a project

○ Public

Anyone can see this project. Collaborators have write access

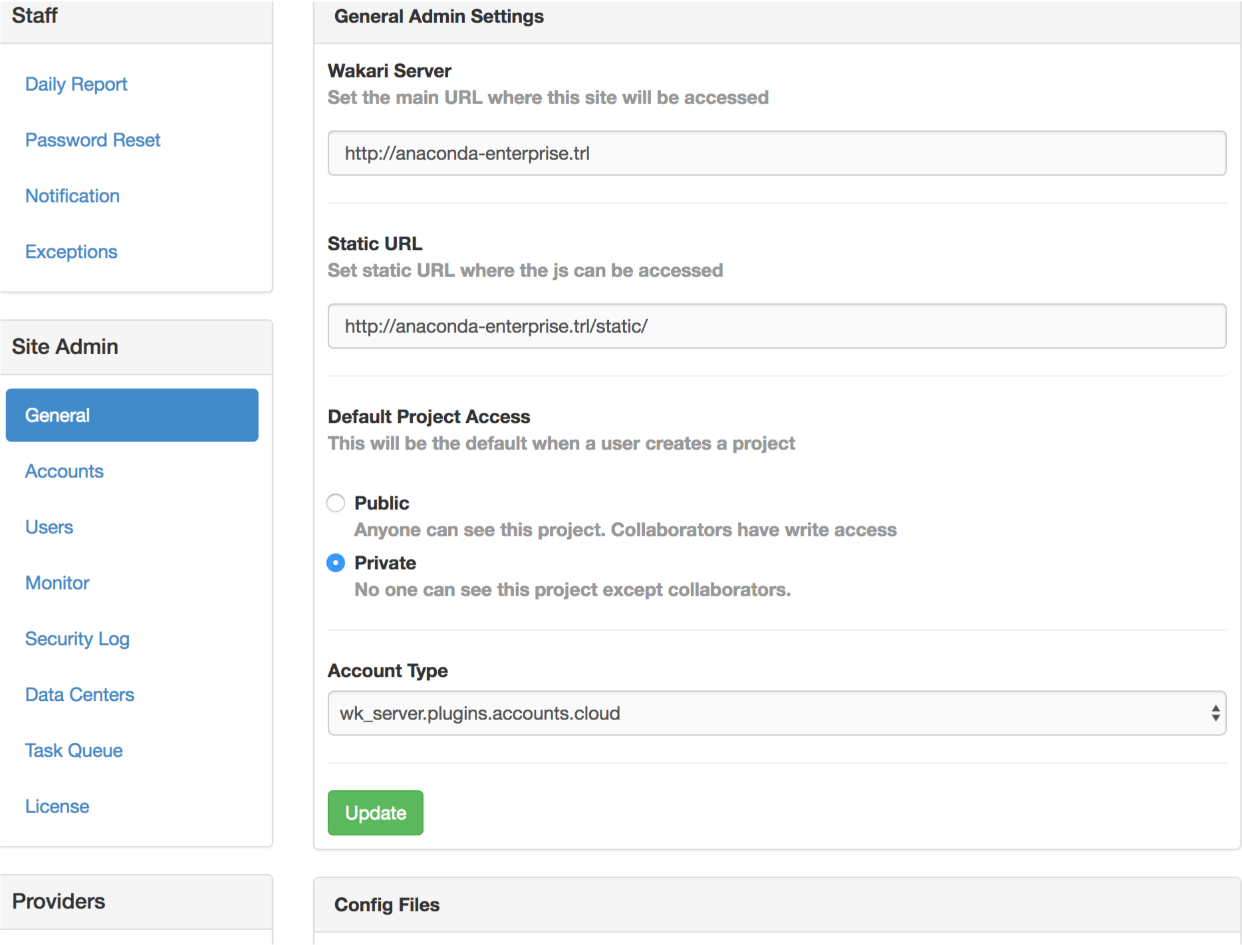
● Private

No one can see this project except collaborators.

Account Type

wk_server.plugins.accounts.cloud

Config Files



General Admin Settings

Wakari Server

Set the main URL where this site will be accessed

http://anaconda-enterprise.trl

Static URL

Set static URL where the js can be accessed

http://anaconda-enterprise.trl/static/

Default Project Access

This will be the default when a user creates a project

○ Public

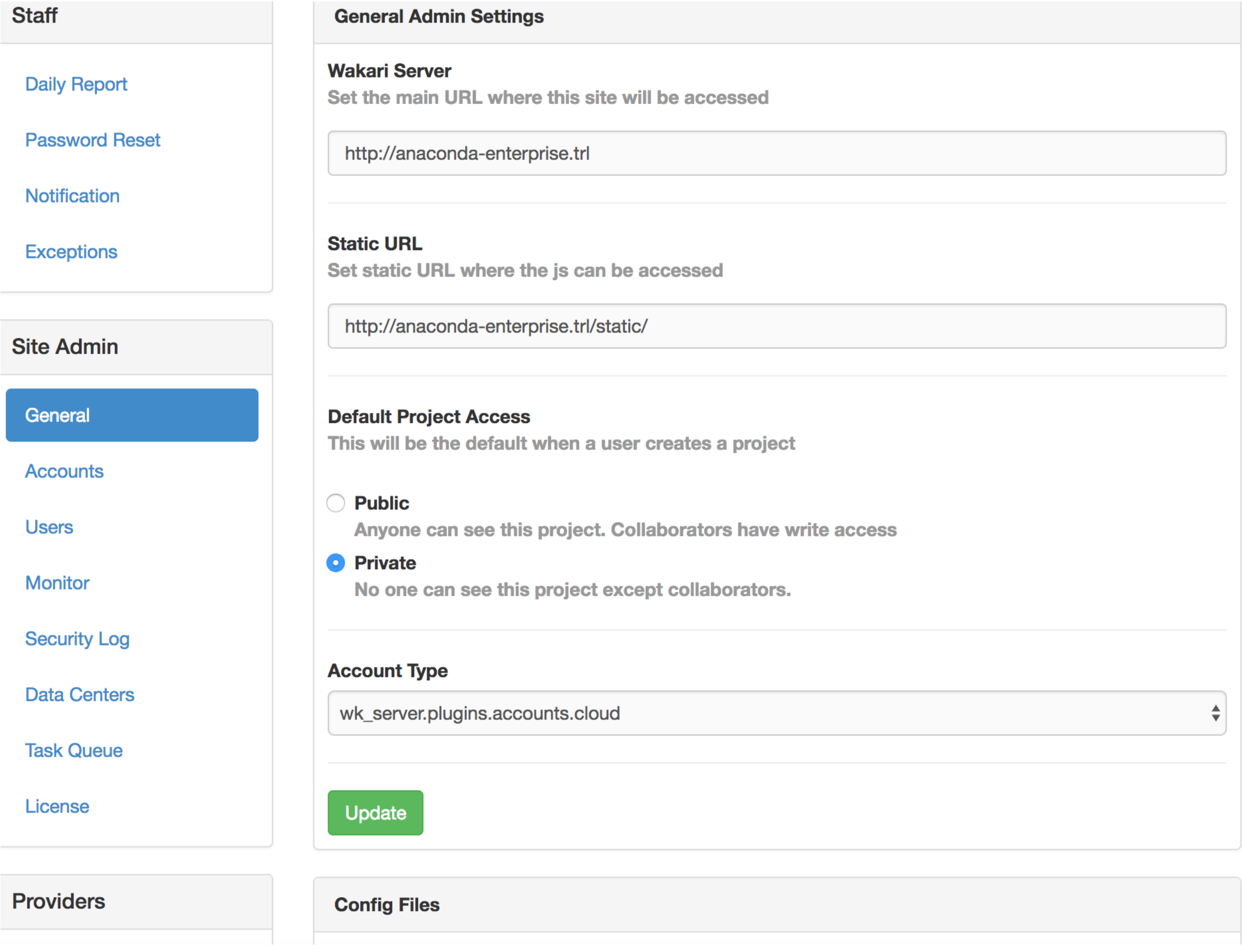
Anyone can see this project. Collaborators have write access

● Private

No one can see this project except collaborators.

Account Type

wk_server.plugins.accounts.cloud



Staff

- Daily Report
- Password Reset
- Notification
- Exceptions

Site Admin

- General
- Accounts
- Users
- Monitor
- Security Log
- Data Centers
- Task Queue
- License

Providers

General Admin Settings

Wakari Server
Set the main URL where this site will be accessed

http://anaconda-enterprise.trl

Static URL
Set static URL where the js can be accessed

http://anaconda-enterprise.trl/static/

Default Project Access
This will be the default when a user creates a project

○ **Public**
Anyone can see this project. Collaborators have write access

● **Private**
No one can see this project except collaborators.

Account Type

wk_server.plugins.accounts.cloud

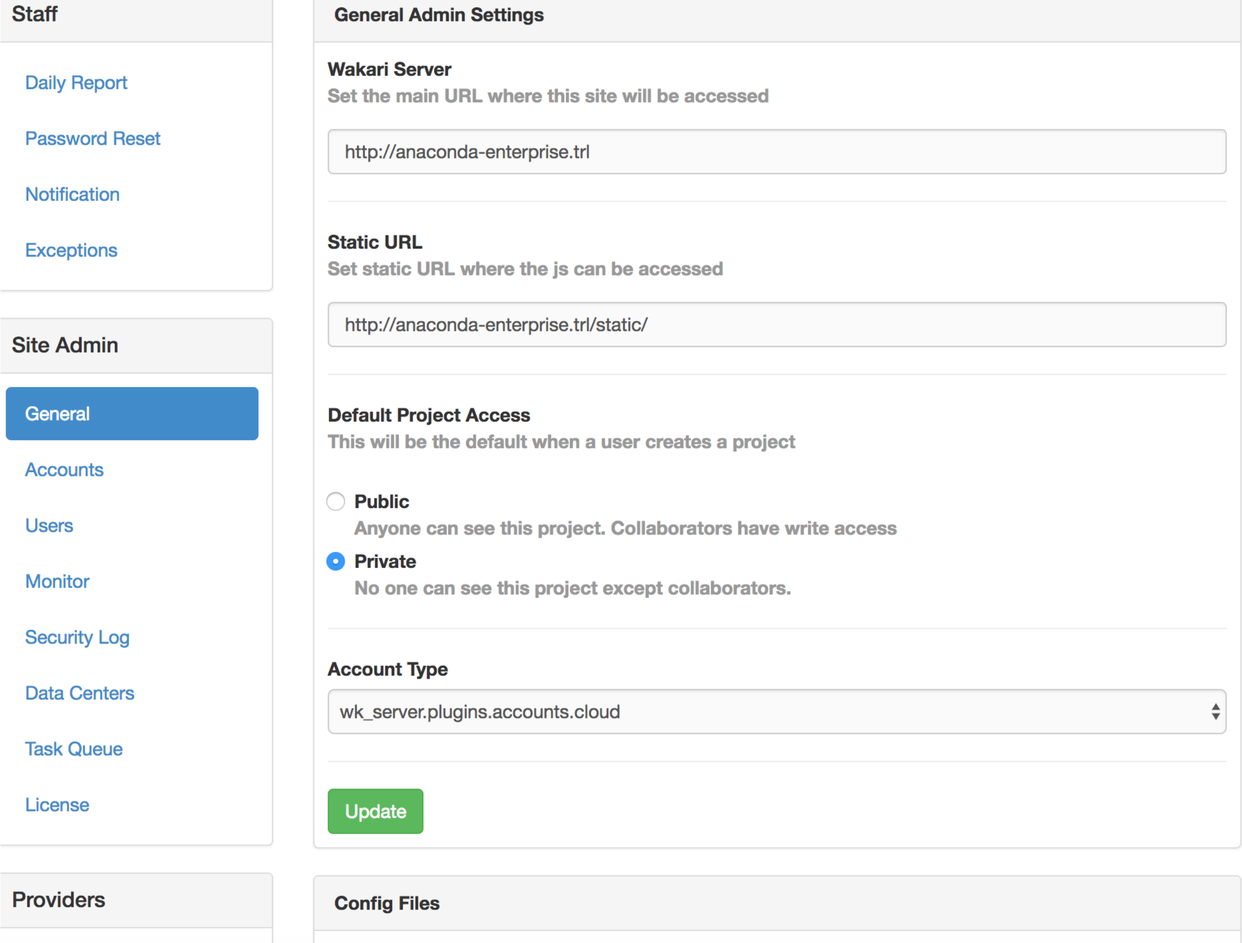
Update

Config Files

3. In the Account Type box, select the account type—cloud or LDAP.
4. Click the Update button.

Changing the default for project access

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select General:



General Admin Settings

Wakari Server
Set the main URL where this site will be accessed

http://anaconda-enterprise.trl

Static URL
Set static URL where the js can be accessed

http://anaconda-enterprise.trl/static/

Default Project Access
This will be the default when a user creates a project

○ Public
Anyone can see this project. Collaborators have write access

● Private
No one can see this project except collaborators.

Account Type

wk_server.plugins.accounts.cloud

3. Under Default Project Access, select the default access type for new projects: Public or Private.
4. Click the Update button.

Changing the owner of a project

To change the owner of a project:

1. Collect the project name, the user name of the previous owner, and the user name of the new owner.
2. Run the `wakari-server` executable command `wk-server-admin`:

```
/opt/wakari/wakari-server/bin/wk-server-admin project-owner --project PROJECT --
↪old OLD_OWNER --new NEW_OWNER --delete --keep-owner
```

- PROJECT: The project name.
- OLD_OWNER: The user name of the previous owner.
- NEW_OWNER: The user name of the new owner.
- `--delete`: An optional flag that deletes the old project directory in the `projects` directory of OLD_OWNER. If this flag is not used, the old project directory is preserved but no longer used.
- `--keep-owner`: An optional flag that makes OLD_OWNER a collaborator of the project after it is transferred to NEW_OWNER. If this flag is not used, OLD_OWNER will no longer have collaborator access to the project.

NOTE: The OLD_OWNER user must still exist when the project's owner is changed. Before deleting any user, be sure to change the owner of the user's projects.

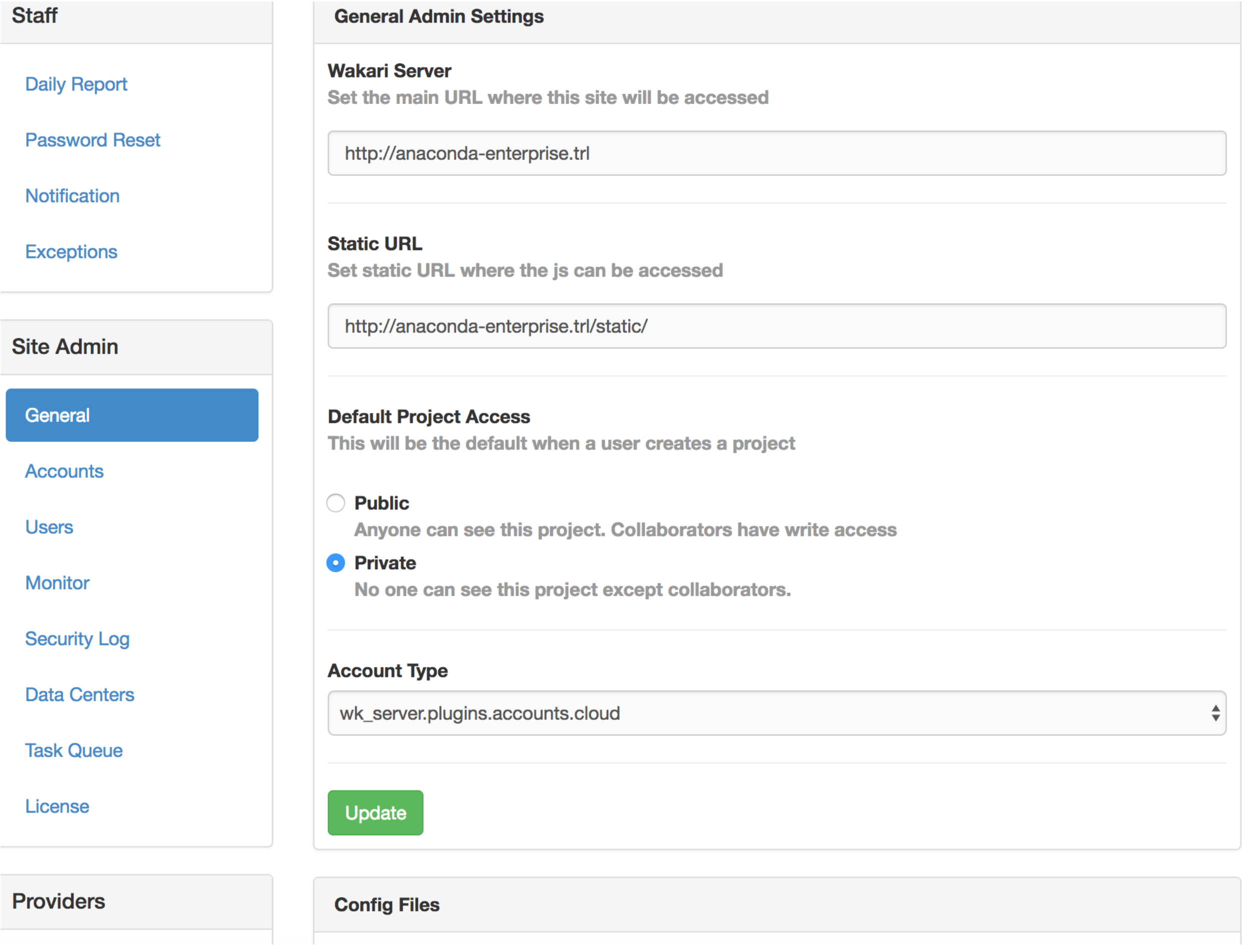
Editing configuration files

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select General.
3. In the Config Files section, change the configuration settings for your AEN installation. For more information on configuration files, see *Using configuration files*.
4. Click the Update button.

Managing your AEN license

1. In the AEN navigation bar, click Admin to open the Admin Settings page.
2. In the **Site Admin** menu, select License:

   The Current License section displays information regarding your AEN license, including the name of the product, vendor, license holder's name, end and issued dates, company name, license type, and contact email.



General Admin Settings

Wakari Server
Set the main URL where this site will be accessed
http://anaconda-enterprise.trl

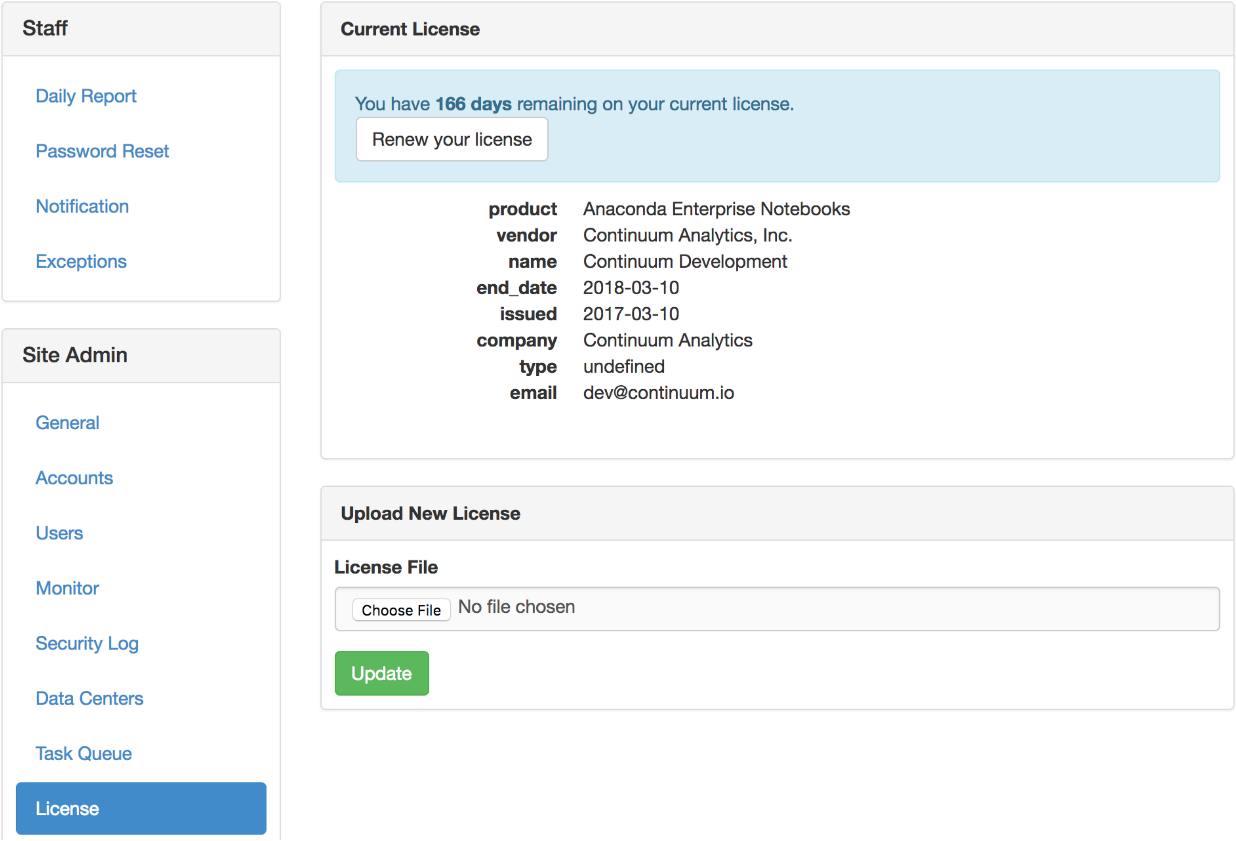
Static URL
Set static URL where the js can be accessed
http://anaconda-enterprise.trl/static/

Default Project Access
This will be the default when a user creates a project

○ Public
Anyone can see this project. Collaborators have write access

● Private
No one can see this project except collaborators.

Account Type
wk_server.plugins.accounts.cloud



Current License

You have **166 days** remaining on your current license.

Renew your license

product	Anaconda Enterprise Notebooks
vendor	Continuum Analytics, Inc.
name	Continuum Development
end_date	2018-03-10
issued	2017-03-10
company	Continuum Analytics
type	undefined
email	dev@continuum.io

Upload New License

License File

Choose File No file chosen

Renewing your AEN license

1. Click the Renew your license button.
2. In the Upload New License section, click the Choose File button.
3. Select the new license file.
4. Click the Open button.
5. Click the Update button.

   Your renewed license information is displayed.

Cheat sheet

The Admin dashboard includes three menus in the left column: **Staff**, **Site Admin** and **Providers**.

Staff menu

- Daily Report—See the number of users and projects.
- Password Reset—Reset a user password.
- Notification—Send system messages to users via SES or SMTP.

- Exceptions—If errors are raised while AEN is running, a red dot appears in the AEN navigation bar. Review errors and mark them as read.

Site Admin menu

- General—Change the configuration settings for your AE Notebook server installation.
- Accounts—Turns on or off Open Registration.
- Users—View usernames, number of projects and last logins.
- Monitor–View status of applications with related data, terminate or restart
- Security Log—View errors that could affect security.
- Data Centers—View current data centers and add a new data center.
- Task Queue—View workers in the task queue and priority.
- License—View current AEN license or upload a new license.

Providers menu

Enterprise Resources—View, add or remove local or cloud services and designate public or private to control access to a compute node.

Troubleshooting

This troubleshooting guide provides you with ways to deal with issues that may occur with your AEN installation.

General troubleshooting steps

1. Clear browser cookies. When you change the AEN configuration or upgrade AEN, cookies remaining in the browser can cause issues. Clearing cookies and logging in again can help to resolve problems.
2. *Make sure NGINX and MongoDB are running*.
3. Make sure that AEN services are *set to start at boot*, on all nodes.
4. *Make sure that services are running* as expected. If any services are not running or are missing, *restart them*.
5. *Check for and remove extraneous processes*.
6. *Check the connectivity between nodes*.
7. *Check the configuration file syntax*.
8. *Check file ownership*.
9. *Verify that POSIX ACLs are enabled*.

Browser error: too many redirects

Cause

Browser cookies are out of date.

Solution

1. Log out.
2. Clear the browser's cookies.
3. Clear the browser cache.
4. Log in.

Browser error: too many redirects when starting project apps

Browser shows "Too many redirects" when the user tries to start an application.

Cause

The project's Compute Resource is invalid or was deleted.

Solution

Move the project to a valid Compute Resource.

Exception: exceptions.TypeError: ‘NoneType’ object has no attribute ‘__getitem__’

This exception appears on the Admin > Exceptions page when a project does not have a Compute Resource assigned.

Cause

The project’s Compute Resource is invalid or was deleted.

Solution

Move the project to a valid Compute Resource.

Error: unix:////opt/wakari/wakari-server/etc/supervisor.sock no such file

This is a supervisorctl error.

Cause

supervisord is not running on the Server.

Solution

Ensure that supervisord is included in the crontab. Then restart supervisord manually.

Error: “Data Center Not Found” when deleting a project

Cause

The data center has been removed.

Solution

As root, run:

```
/opt/wakari/wakari-server/bin/wk-server-admin remove-project --db-only <user>
↪<project>
```

Forgotten administrator password

1. Use ssh to log into the server as root.
2. Run:

   ```
   /opt/wakari/wakari-server/bin/wk-server-admin reset-password -u SOME_USER -p SOME_
   ↪PASSWORD
   ```

NOTE: Replace SOME_USER with the administrator username and SOME_PASSWORD with the password.

3. Log into AEN as the administrator user with the new password.

Alternatively you may add an administrator user:

1. Use ssh to log into the server as root.
2. Run:

```
/opt/wakari/wakari-server/bin/wk-server-admin add-user SOME_USER --admin -p SOME_
↪PASSWORD -e YOUR_EMAIL
```

NOTE: Replace SOME_USER with the username, replace SOME_PASSWORD with the password, and replace YOUR_EMAIL with your email address.

3. Log into AEN as the administrator user with the new password.

Log files being deleted

Log files are being deleted.

NOTE: Locations of AEN log files for each process and application are shown in the node sections in *Concepts*.

Cause

AEN installers log into `/tmp/wakari\_{server,gateway,compute}.log`. If the log files grow too large, they might be deleted.

Solution

To set the logs to be more or less verbose, Jupyter Notebooks uses Application.log_level.

To make the logs less verbose than the default, but still informative, set Application.log_level to ERROR.

Error: This socket is closed

You receive the “This socket is closed” error message when you try to start an application.

Cause

When the supervisord process is killed, information sent to the standard output `stdout` and the standard error `stderr` is held in a pipe that will eventually fill up.

Once full, attempting to start any application will cause the “This socket is closed” error.

Solution

To prevent this issue:

- Follow the instructions in *Managing services* to stop and restart processes.
- Do not stop or kill supervisord without first stopping wk-compute and any other processes that use it.

To resolve the "This socket is closed" error:

1. Stop wk-compute by running `sudo kill -9`.
2. Restart the supervisord and wk-compute processes:

   ```
   sudo /etc/init.d/wakari-compute stop
   sudo /etc/init.d/wakari-compute start
   ```

Service error 502: Cannot connect to the application manager

Gateway node displays "Service Error 502: Can not connect to the application manager."

Cause

A compute node is not responding because the wk-compute process has stopped.

Solution

Stop and then restart the supervisord and wk-compute processes:

```
sudo /etc/init.d/wakari-compute stop
sudo /etc/init.d/wakari-compute start
```

502 communication error on Amazon web services (AWS)

You receive the "502 Communication Error: This gateway could not communicate with the Wakari server" error message.

Cause

An AEN gateway cannot communicate with the Wakari server on AWS. There may be an issue with the IP address of the Wakari server.

Solution

Configure your AEN gateway to use the DNS hostname of the server. On AWS this is the DNS hostname of the Amazon Elastic Compute Cloud (EC2) instance.

Invalid username

Cause

The username does not follow 1 or more of these rules:

- Must be at least 3 characters and no more than 25 characters.
- The first character must be a letter (A-Z) or a digit (0-9).

- Other characters can be a letter, digit, period (.), underscore (_) or hyphen (-).
- The POSIX standard specifies that these characters are the portable filename character set, and that portable usernames have the same character set.

Solution

Follow the above rules for usernames.

Notebook Error: Cannot download notebook as PDF via LaTeX

Cause

LaTeX is not properly installed.

CentOS/6 Solution

1. Install TeXLive from the TUG site. Follow the described steps. The installation may take some time.
2. Add the installation to the `PATH` in the file `/etc/profile.d/latex.sh`. Add the following, replacing the year and architecture as needed:

```
PATH=/usr/local/texlive/2017/bin/x86_64-linux:$PATH
```

3. Restart the compute node.

CentOS/7 Solution

1. Install the missing packages running the command:

```
yum install texlive texlive-xetex texlive-xetexconfig texlive-xetex-def texlive-
↪adjustbox texlive-upquote texlive-ulem
```

Unresponsive `wk-server` thread without error messages

Cause

Two things can cause the `wk-server` thread to freeze without error messages:

- LDAP freezing
- MongoDB freezing

If LDAP or MongoDB are configured with a long timeout, Gunicorn can time out first and kill the LDAP or MongoDB process. Then the LDAP or MongoDB process dies without logging a timeout error.

Solution

1. Check for frozen LDAP or MongoDB server processes.
2. You may also wish to configure the Gunicorn timeout to more than 30 seconds.

Unresponsive `wk-gateway` thread without error messages

Cause

If TLS is configured with a passphrase protected private key, `wk-gateway` will freeze without any error messages.

Solution

Update the TLS configuration so that it does not use a passphrase protected private key.

Error starting projects

Project's status page shows "There was an error starting this project".

Cause

Lack of disk space in compute nodes prevents projects from starting.

Solution

1. Verify that the project node meets the *system requirements*.
2. Check if there is enough free space on the compute node's partition where `/projects` lives:

   ```
   df -h /projects
   ```

3. Free up some disk space to meet the system requirements.
4. Restart the project.

Changes in .condarc file are ignored

Changes applied to `.condarc` are ignored by conda.

Cause

Conda loads its configuration by merging multiple files together.

Solution

Check if you are applying the changes to the correct file.

To show the merged state that conda is currently using:

```
conda config --show
```

To show all config files that conda is currently reading:

```
conda config --show-sources
```

Frequently asked questions

What is AEN?

For information on AEN, see *Anaconda Enterprise 4 Notebooks*.

Can notebooks be shared with anyone?

Yes. When you share a Jupyter Notebook through AEN, it can be viewed and run without the need to install anything special, regardless of what libraries were used to create the notebook. Each notebook also includes the python environment that it needs to run in.

AEN allows users to clone a shared Jupyter Notebook into their AEN account to make whatever changes or modifications they want. The notebook's Python environment is also cloned, so it runs in the same environment as the shared Jupyter Notebook unless it is changed.

Can I disable the option, "publish your notebook to anaconda.org"?

Yes. The upload button in the notebook app executes the option "publish your notebook to anaconda.org". To disable it, log in as the AEN_SRVC_ACCT and run these commands:

```
source activate /opt/wakari/wakari-compute
jupyter-nbextension disable nb_anacondacloud --py --sys-prefix
jupyter-serverextension disable nb_anacondacloud --py --sys-prefix
```

How can I check the version number of my AEN server?

Go to this URL in a browser: `http://$AEN_SERVER/admin/list`

NOTE: Replace `$AEN_SERVER` with the domain name or the domain name and port number of your AEN server.

Can I use AEN to access CSV or Amazon S3 data?

Yes. If your data is in CSV files, upload the CSV files to your AEN account using the upload controls in the File Browser of the Workbench Application or the File Transfer Application.

To access data stored on Amazon S3, use the Boto interface from AEN. See the public data files in AEN for examples of how to use Boto to pull your data from Amazon S3 into AEN. For more information, see Boto documentation.

You can also use IOPro to simplify and optimize the conversion of your data into Python arrays.

Can I install other Python packages?

Yes, by creating a custom environment for your packages within your project.

For more information, see *Using the NBConda extension*.

Can I create a Python environment from the command line?

Yes, you can use the `conda create` command to create custom Python environments with whatever packages you choose. All AEN environments are shared with all the team members of a project.

EXAMPLE: In this example, myenv is a new environment containing the NumPy package.

```
conda create -n myenv numpy
```

NOTE: Python, Jupyter Notebooks and PIP are installed by default in all new AEN environments.

To use your new environment, activate it by running `source activate myenv`.

Can I connect to GitHub with AEN?

Yes, you have full access to GitHub through an AEN Terminal application.

To generate an SSH key from your AEN account and add it to your GitHub account:

1. Generate a GitHub SSH key.
2. Copy your key by running `cat ~/.ssh/id_rsa.pub`.
3. Select and copy the contents of the `id_rsa.pub` file to the clipboard.
4. Follow GitHub's instructions to go to your GitHub account and paste it from your clipboard into the appropriate box in your GitHub settings.

Can I print or print preview my Jupyter Notebooks?

Yes, you can print your notebooks using your browser's regular printing capabilities.

You can also preview the printed page by clicking the **File** menu and selecting Print Preview.

Is there a set amount of storage on AEN?

No, there is no set limit for storage in AEN. You are limited only by the size of the disk where AEN is installed.

If you need more storage, contact your system administrator.

How do I get help, give feedback, suggest features or report a bug?

See *Help and support*.

Help and support

Priority support is included with the purchase of an Anaconda subscription.

Contact your administrator first if you are having problems. Your administrator has a service level agreement where your issue will be responded to within a specific response time, depending on type and severity.

Training and consulting

Training and consulting is available for AEN and any other Anaconda product.

For more information, please contact your account representative or email the sales team.

Providing feedback

Your feedback is very important to us!

Please, send us any product feedback while you are thinking about it.

TIP: Be sure to select AEN as the Platform Component Name.

Submitting feature requests

We'd love to hear your ideas for consideration in future releases!

Your ideas help us build a better product. Your administrator can submit a support ticket for you.

NOTE: You can also request new features by using the product feedback form.

Reporting a bug

If you think you have found a bug, please contact your administrator immediately. They will open a support ticket for your issue.

Additional resources

The following resources are useful for getting started with Jupyter Notebooks:

- Jupyter Notebook quick start guide
- Jupyter Notebook user documentation

- GitHub shows the most popular Jupyter notebooks of the month, week, and day.

Release notes

v4.2.2 March 1, 2018

Administrator-facing changes:

- Add admin command to change project owner
- Server: Add ability to disable public projects
- Gateway: Add support for SSL private key passphrase
- Docs: Add backup and restore runbook to the docs
- Docs: Emphasize backups before upgrading process
- Docs: Recommend putting AEN and projects folder on the same filesystem
- Docs: Add RHEL version 7.4 to supported versions
- Docs: Add troubleshooting instructions to fix problems when downloading notebook as PDF via LaTeX

User-facing changes:

- Upgrade bokeh to version 0.12.7
- Upgrade holoviews to version 1.8.3
- Upgrade numba to version 0.35.0
- Upgrade scikit-learn to version 0.19.0

Internal fixes:

- Fix bug in init scripts when requiretty is enabled
- Fix bugs related to AEN_SUDO_SSH option
- Fix bug in fix_ownership function when directories contain spaces
- Docs: Fix error in Active Directory configuration example
- Server: Fix bug when updating user/group in supervisor configuration files in post-install for server and gateway
- Server: Fix bug Admin reports on user totals are inconsistent
- Server: Fix error in login screen when open registration and LDAP are enabled
- Server: Fix bug in Last seen date
- Server: Fix bug Monitor Report blank
- Server: Load JS files from local CDN
- Server: Fix error when terminating or relaunching an application from Monitor
- Server: Fix error creating projects when using Internet Explorer 11
- Compute: Fix 404 errors when using pivottablesjs
- Remove Wakari Cloud leftovers

v4.2.1 December 18, 2017

Administrator-facing changes:

- None

User-facing changes:

- None

Internal fixes:

- Fix undetected “ca” key when using self-signed certificates signed by a private CA
- Fix login redirects when using SSL
- Add verify gateway SSL certificate for get and post requests

v4.2.0 November 22, 2017

Administrator-facing changes:

- Feature/allow remote MongoDB
- Allow for configuration for login timeout and set default
- Add verbose option to conda create clone
- Avoid duplicate name for resources / compute-nodes
- Allow renaming main and message queue databases
- PAM-based authentication module
- Change wakari logos to Anaconda logos
- Replace ‘wakari’ wording
- New config option to move the user’s home directory into the user’s project directory
- Make logging less verbose in AEN
- Documentation for PySpark kernel installation
- Improve SSL documentation

User-facing changes:

- New config option to move the user’s home directory into the user’s project directory
- Package cache was moved from user’s home directory into the user’s project directory
- Change wakari logos to Anaconda logos
- Fix error for deleting tags to work
- Define shell prompt in `.projectrc` template
- Replace ‘wakari’ wording

Internal fixes:

- Move server unix socket from `/tmp` to `/opt/wakari/wakari-server/var/run`
- Make project deletion synchronous for consistency
- Avoid storing `csrf` token in the user profile

- Expire gateway session when server logs out
- Allow log rotation in the three components
- Fix permissions on static files
- Change log level to debug in gateway
- Do not log private keys in gateway
- Save request remote address when logging action
- Unify logs formatting and timezone in compute nodes with Winston
- Several fixes and documentation improvements

v4.1.3 August 16, 2017

- Upgrade conda to version 4.3.24
- Upgrade anaconda to version 4.4.0
- Admin application monitor
- Block access to package list view
- Add placeholders in password reset form
- Change static content location
- Fix error when checking for package updates in notebook application
- Replace slashes in project tags
- Fix submit errors in password reset form
- Replace/remove "wakari" word from multiple places
- Fix missing commands missing sudo in start-project
- Improve gateway and compute node validators
- Check if bzip2 is installed during server setup process
- Include port number in host header
- Forbid creation of empty tags
- Repair "Create Account" link in login page
- Use UTC for server logs
- Mark datacenters as trusted by default
- Disable heart beating
- Compute resource: Show full path to log file
- Improve init scripts
- Allow deleting all projects
- mtq: Implement exponential backoff on connection error to mongodb
- In the general admin display, do not show the bind password for LDAP
- The accelerate package has been removed from the installation
- Other minor bugfixes

v4.1.2 March 29, 2017

This is mainly a maintenance release improving internal machinery and upgrading the root packages.

- Upgrade conda to version 4.3.14
- Upgrade Anaconda to 4.3.1
- Uprade r-base to 3.2.2
- Fixed AEN nb_conda to be compatible with conda 4.3.x series
- Several documentation fixes
- Other minor bugfixes

v4.1.1 December 15, 2016

- Added CentOS 7 support
- Support dots in usernames
- More usernames validation
- Fixed creation (through nb_conda) of single letter environment names
- Environment names (through nb_conda) validation
- Fixed uploading of notebook using nb_anacondacloud
- Fixed attaching of environments in published notebooks through nb_anacondacloud
- Several documentation fixes
- Other bugfixes

v4.1.0 October 21, 2016

- Added JupyterLab application
- Removed GateOne terminal application
- Included additional notebook extensions (`nbpresent` and `nb_anaconda_theme`)
- Updated to conda 4.2.9 in default project environments
- Added HTTP timeout setting for gateway and compute launcher
- Changed default gateway port to 8089
- Added support for all-numeric usernames
- Add R channel to default conda configuration file
- Other bugfixes

v4.0.0 June 30, 2016

- Customized installation with:
  - AEN Functional ID and Group
  - AEN (installation and run) `sudo` commands

  - Removal of root access from the AEN service account
  - Configurable sudo command
  - Restriction of sudo access to all the processes
- Upgrade Jupyter to 4.2
- Upgrade the anaconda-nb-extensions to the latest versions
- Upgrade Anaconda to 4.0
- Deprecate wakari-publisher
- Security enhancements
- SSL configuration documented between all AEN Server components
- Several bugfixes
- Overall documentation revision and general improvement

v0.10.0 February 2, 2016

- New projects dashboard
- Capability to star and tag a project
- Sticky searches
- New Jupyter Notebook extensions
- Updates to all packages. Highlights: bokeh 0.11, ipython/jupyter 4.1.

v0.9.1 October 19, 2015

- New Search capability to find projects and files within a project.
- Added “Related Projects” list to the project view, based on code similarity.
- New UI for fine-grained access control of project files in the Workbench app
- Viewer app now renders plain text files correctly
- Updated LDAP configuration docs
- Updates to all packages. Highlights: bokeh 0.10, ipython/jupyter 4.0.

Note ElasticSearch, and an Oracle JRE, must be installed on the server in order to use the new search features. Indexing of project files will begin when the project is started (or paused and re-started). If search features are not desired, set `"SEARCH_ENABLED": false` in the server configuration file to avoid errors.

v0.8.0 August 21, 2015

New Features

- Updated packages based on Anaconda 2.3, and removed older packages no longer in Anaconda.
- Updated IPython to version 3.2.1
- Documentation is now installed with the server (use the Help link in the top navigation bar)

- Added the ability for the administrator to define a customized default project environment.
- The server has been updated to use python 2.7.10.
- Init scripts are now provided for each Anaconda Enterprise Notebooks service.
- Added relevant links to some error pages

Problems Resolved in this Release

- Project status indicators (e.g. starting, pausing) now automatically update.
- If an access is unauthorized, the server now returns a 403 (Unauthorized) status code and prompts the user to log in.
- Modified nginx configuration to support running the server on non-standard ports.
- The server installation no longer uses a default password for the wakari user. A random password is generated and displayed during installation.
- Prevent double-click from attempting to create a project twice
- Removed an obsolete script reference that was causes a 404 error to be logged in the browser console when opening the Terminal app.
- The installer scripts no longer fail if the database already contains the 'wakari' user.
- Updated example notebooks to work with latest Bokeh release.
- Fixed terminal app key bindings to allow Mac command key to work normally
- Installers now indicate where the installation logs are stored
- LDAP user attributes containing binary data are now ignored.

Documentation Updates

- Updated and consolidated Troubleshooting guide.
- Simplified some steps in the installation procedure.
- Updated notebooks in the Examples directory for use with the latest IPython Notebook and Bokeh.
- Added a section on project permissions to the Troubleshooting guide.
- Added notes on how to remove a project if the datacenter has already been removed.

v0.7.0 June 12, 2015

New Features

- Updated Bokeh to v0.9
- Ability to list packages installed on the server
- Administrators now have full access to all projects.
- Added automated checking and display of connection status between server, data centers, and compute resources.

- When creating a new project, an environment for the project is automatically created as a clone of the root Anaconda environment.

Problems Resolved in this Release

- Problem with checking in files with revision control extension
- Revision control extension can't handle notebook names with spaces
- Problem moving files form one compute node to another if configured for LDAP
- Should default to UTF-8 encoding and warn user if no locale is detected
- Adding a compute resource via the command line admin tool does not work
- The installer now sets `umask 0022` to ensure correct file permissions

Documentation Updates

- Added a *Troubleshooting* section to the documentation.
- Added notes on how to configure crontab to start the Anaconda Enterprise Notebooks services at startup
- Example SSL config file now has correct log paths
- Added instructions on how to ensure that POSIX ACL support is enabled on the projects directory.
- Fixed syntax problem in sample LDAP config.json
- Added section on how to use self-signed or private CA certificates

v0.6.3 March 27, 2015

- Updated LDAP module
- LDAP user filtering
- Added Notebook locking
- Added Notebook integrated revision control system
- Move projects between compute nodes
- User-specific binding to compute nodes (private compute nodes)
- Improved installation process and dependency checking
- Incorporated support for SSL for Server and Gateway nodes
- Improved Gateway error handling
- Fixed package dependencies for update process
- Documentation updates

Previous versions

Previous version documentation is provided for users who have not yet upgraded to the current version of AEN.

Anaconda Enterprise Notebooks (AEN 4.2.1)

User guide (AEN 4.2.1)

Concepts (AEN 4.2.1)

Getting started (AEN 4.2.1)

Basic tasks (AEN 4.2.1)

Working with projects (AEN 4.2.1)

Searching for a project or file (AEN 4.2.1)

Adding and removing team members on a project (AEN 4.2.1)

Controlling access to your project (AEN 4.2.1)

Starting and stopping a project (AEN 4.2.1)

Making a project public or private (AEN 4.2.1)

Tagging a project (AEN 4.2.1)

Starring a project (rating) (AEN 4.2.1)

Claim ownership of a project (AEN 4.2.1)

Changing a project's summary or description (AEN 4.2.1)

Viewing a project's status (AEN 4.2.1)

Viewing related projects (AEN 4.2.1)

Viewing top-rated projects (AEN 4.2.1)

Using tags to find a project (AEN 4.2.1)

Viewing your top collaborators (AEN 4.2.1)

Sharing projects and notebooks (AEN 4.2.1)

Deleting a project (AEN 4.2.1)

Using AEN applications (AEN 4.2.1)

Using Workbench (AEN 4.2.1)

Using Viewer (AEN 4.2.1)

Using JupyterLab (AEN 4.2.1)

Basic

These topics all use the web browser to manage your projects. They are suited for a beginning user.

Projects (AEN 4.1.2)

Using applications (AEN 4.1.2)

Workbench Application (AEN 4.1.2)

Viewer Application (AEN 4.1.2)

JupyterLab Application (AEN 4.1.2)

Terminal Application (AEN 4.1.2)

Jupyter Notebook Application (AEN 4.1.2)

Compute Resource Config Application (AEN 4.1.2)

Team Collaboration (AEN 4.1.2)

Account Administration (AEN 4.1.2)

Advanced

If you are comfortable entering commands on the command line, there is a lot more you can do with Anaconda Enterprise Notebooks.

Anaconda Environments (AEN 4.1.2)

Data (AEN 4.1.2)

Visualization (AEN 4.1.2)

Project environment variables (AEN 4.1.2)

Help & support

Help and support (AEN 4.1.2)

Frequently Asked Questions (AEN 4.1.2)

Additional Resources (AEN 4.1.2)

Sharing in Anaconda Enterprise Notebooks (AEN 4.1.2)

Administration (AEN 4.1.2)

Configuration Files (AEN 4.1.2)

Start/Shutdown (AEN 4.1.2)

Backup/Restore (AEN 4.1.2)

Third Party Extensions (AEN 4.1.2)

Troubleshooting (AEN 4.1.2)

Compute-nodes (AEN 4.1.2)

Installation (AEN 4.1.2)

Install Steps

Carry out the procedures linked from the table below to perform a complete install of all Anaconda Enterprise Notebooks components.

Installation preparation (AEN 4.1.2)

Install AEN Server (AEN 4.1.2)

Install AEN Gateway (AEN 4.1.2)

Install AEN Compute (AEN 4.1.2)

The following optional install procedures may need to be performed, depending on how you set up your Data Center:

Optional configuration (AEN 4.1.2)

Sudo configuration (AEN 4.1.2)

LDAP configuration (AEN 4.1.2)

SSL (AEN 4.1.2)

wakari_https.conf (AEN 4.1.2)

Single sign on (AEN 4.1.2)

Additional post-install information:

Upgrading Anaconda Enterprise Notebooks (AEN 4.1.2)

Uninstall (AEN 4.1.2)

Release notes (AEN 4.1.2)

Anaconda Enterprise Notebooks (AEN 4.1.1)

Backup/Restore (AEN 4.1.1)

Compute-nodes (AEN 4.1.1)

Configuration Files (AEN 4.1.1)

Administration (AEN 4.1.1)

Start/Shutdown (AEN 4.1.1)

Third Party Extensions (AEN 4.1.1)

Troubleshooting (AEN 4.1.1)

Installation customization (AEN 4.1.1)

Installation Instructions (AEN 4.1.1)

Installation Runbook (AEN 4.1.1)

Updating Anaconda Enterprise Notebooks (AEN 4.1.1)

Configuration (AEN 4.1.1)

Release notes (AEN 4.1.1)

SSL (AEN 4.1.1)

SSO (AEN 4.1.1)

Uninstall (AEN 4.1.1)

wakari_https.conf (AEN 4.1.1)

Additional Resources (AEN 4.1.1)

Account Administration (AEN 4.1.1)

Anaconda Environments (AEN 4.1.1)

Using applications (AEN 4.1.1)

Compute Resource Config Application (AEN 4.1.1)

1.2.1 Installation

TIP: If you don't want the hundreds of packages included with Anaconda, you can install Miniconda, a mini version of Anaconda that includes just conda, its dependencies and Python.

TIP: Looking for Python 3.5? See our *FAQ*.

System requirements

- License: Free use and redistribution under the terms of the *Anaconda End User License Agreement*.
- Operating system: Windows Vista or newer, 64-bit macOS 10.10+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.
- Windows XP supported on Anaconda versions 2.2 and earlier. See *Old package lists*. Download it from our archive.
- System architecture: 64-bit x86, 32-bit x86 with Windows or Linux, Power8 or Power9.
- Minimum 3 GB disk space to download and install.

Detailed installation information

Installing on Windows

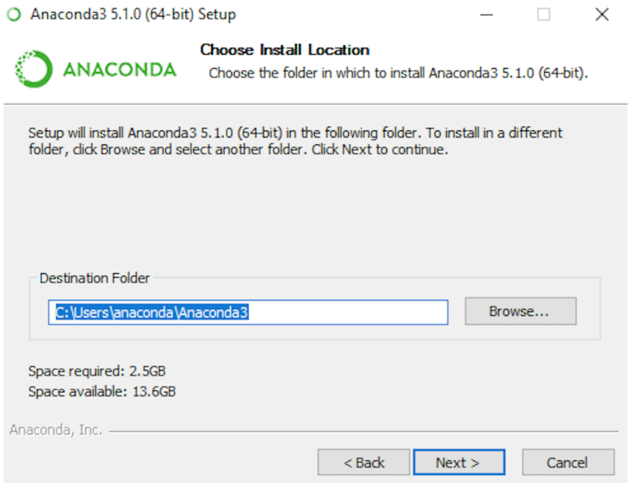
1. Download the Anaconda installer.
2. Optional: *Verify data integrity with MD5 or SHA-256*. More info on hashes
3. Double click the installer to launch.

   NOTE: To prevent permission errors, do not launch the installer from the *Favorites folder*.

   NOTE: If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall Anaconda and re-install it for your user only and try again.
4. Click Next.
5. Read the licensing terms and click "I Agree".
6. Select an install for "Just Me" unless you're installing for all users (which requires Windows Administrator privileges) and click Next.
7. Select a destination folder to install Anaconda and click the Next button. See *FAQ*.

   NOTE: Install Anaconda to a directory path that does not contain spaces or unicode characters.

   NOTE: Do not install as Administrator unless admin privileges are required.
8. Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.



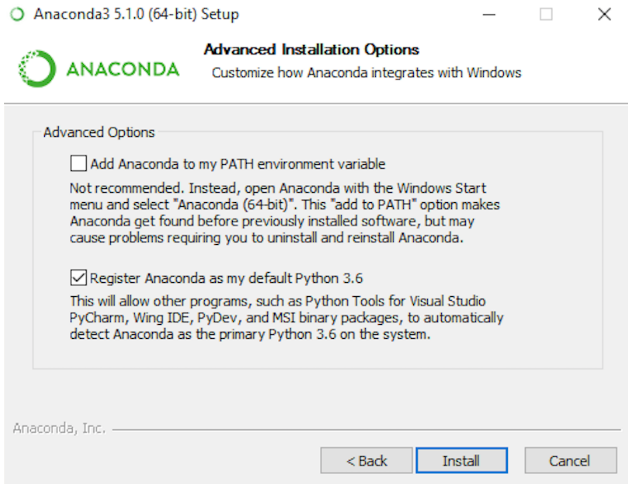
Choose Install Location

Choose the folder in which to install Anaconda3 5.1.0 (64-bit).

Setup will install Anaconda3 5.1.0 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Users\anaconda\Anaconda3

Space required: 2.5GB
Space available: 13.6GB



Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

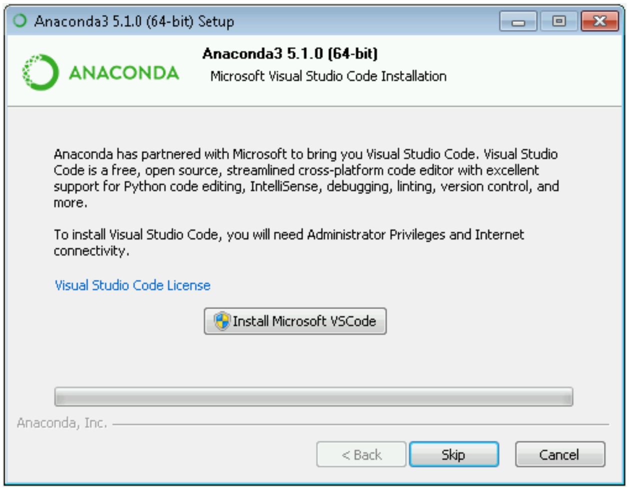
□ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

9. Choose whether to register Anaconda as your default Python 3.6. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, accept the default and leave this box checked.
10. Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
11. Click the Next button.
12. Optional: To *install VS Code*, click the Install Microsoft VS Code button. After the install completes click the Next button.



Anaconda3 5.1.0 (64-bit) Setup

Anaconda3 5.1.0 (64-bit)

Microsoft Visual Studio Code Installation

Anaconda has partnered with Microsoft to bring you Visual Studio Code. Visual Studio Code is a free, open source, streamlined cross-platform code editor with excellent support for Python code editing, IntelliSense, debugging, linting, version control, and more.

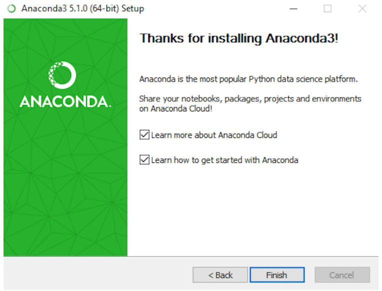
To install Visual Studio Code, you will need Administrator Privileges and Internet connectivity.

Visual Studio Code License

Or to install Anaconda without VS Code, click the Skip button.

NOTE: Installing VS Code with the Anaconda installer requires an internet connection. Offline users may be able to find an offline VS Code installer from Microsoft.

13. After a successful installation you will see the "Thanks for installing Anaconda" dialog box:



Anaconda3 5.1.0 (64-bit) Setup

Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

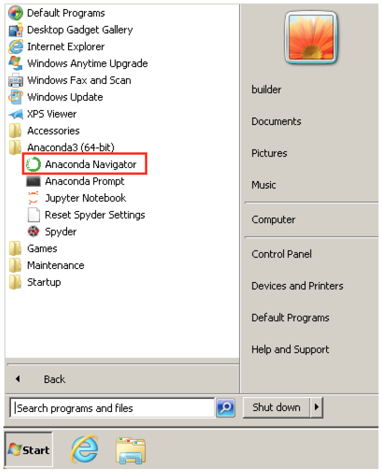
Share your notebooks, packages, projects and environments on Anaconda Cloud!

☑ Learn more about Anaconda Cloud

☑ Learn how to get started with Anaconda

14. If you wish to read more about Anaconda Cloud package management service and Anaconda support, check the boxes "Learn more about Anaconda Cloud" and "Learn how to get started with Anaconda". Click the Finish button.

15. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our *Help page*.



Problems?

See *troubleshooting*.

What's next?

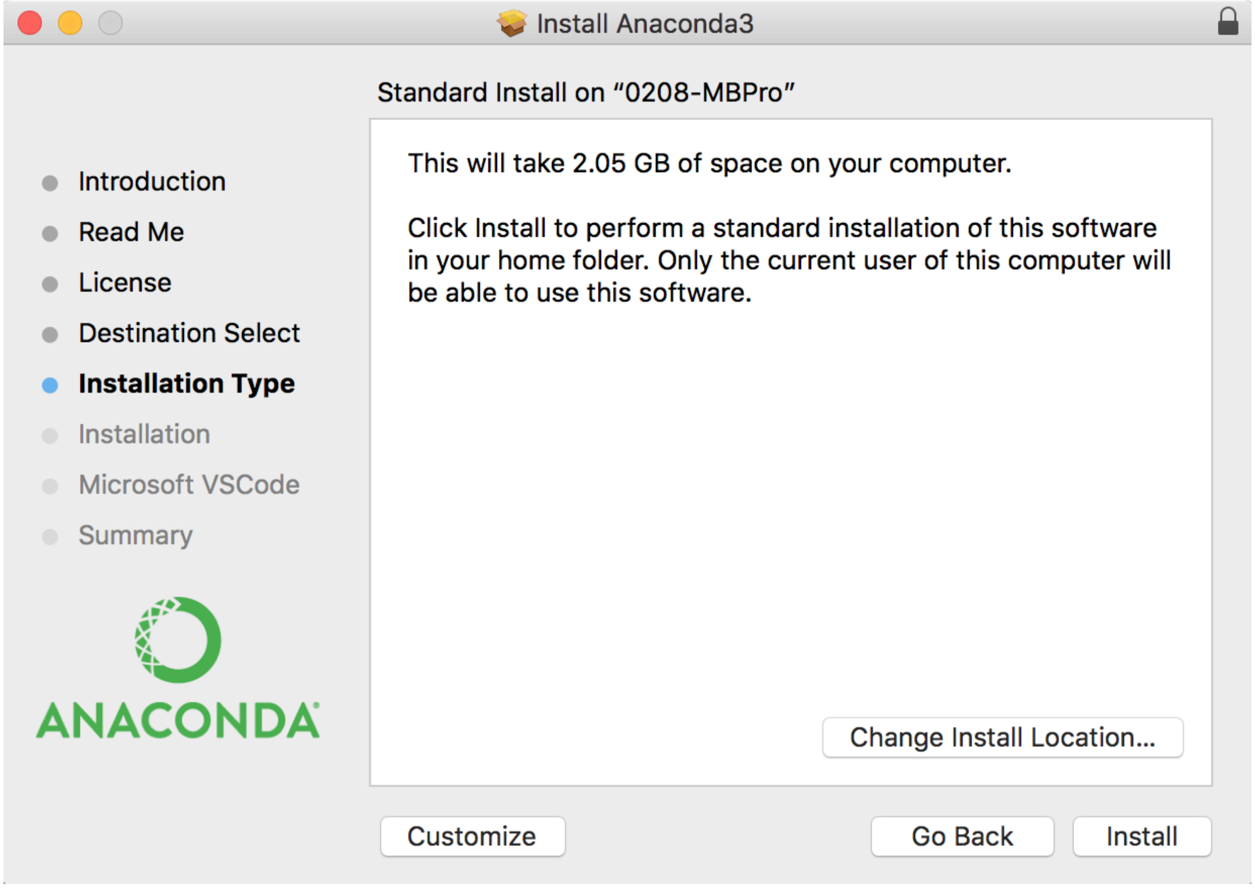
Get started programming quickly with Anaconda in the *Getting started with Anaconda* guide.

Installing on macOS

You can install Anaconda using either the graphical installer ("wizard") or the command line ("manual") instructions below. If you are unsure, choose the graphical install.

macOS graphical install

1. Download the graphical macOS installer for your version of Python.
2. OPTIONAL: *Verify data integrity with MD5 or SHA-256*. For more information on hashes, see What about cryptographic hash verification?.
3. Double-click the downloaded file and click continue to start the installation..
4. Answer the prompts on the Introduction, Read Me and License screens.
5. Click the Install button to install Anaconda in your home user directory (recommended):



Standard Install on “0208-MBPro”

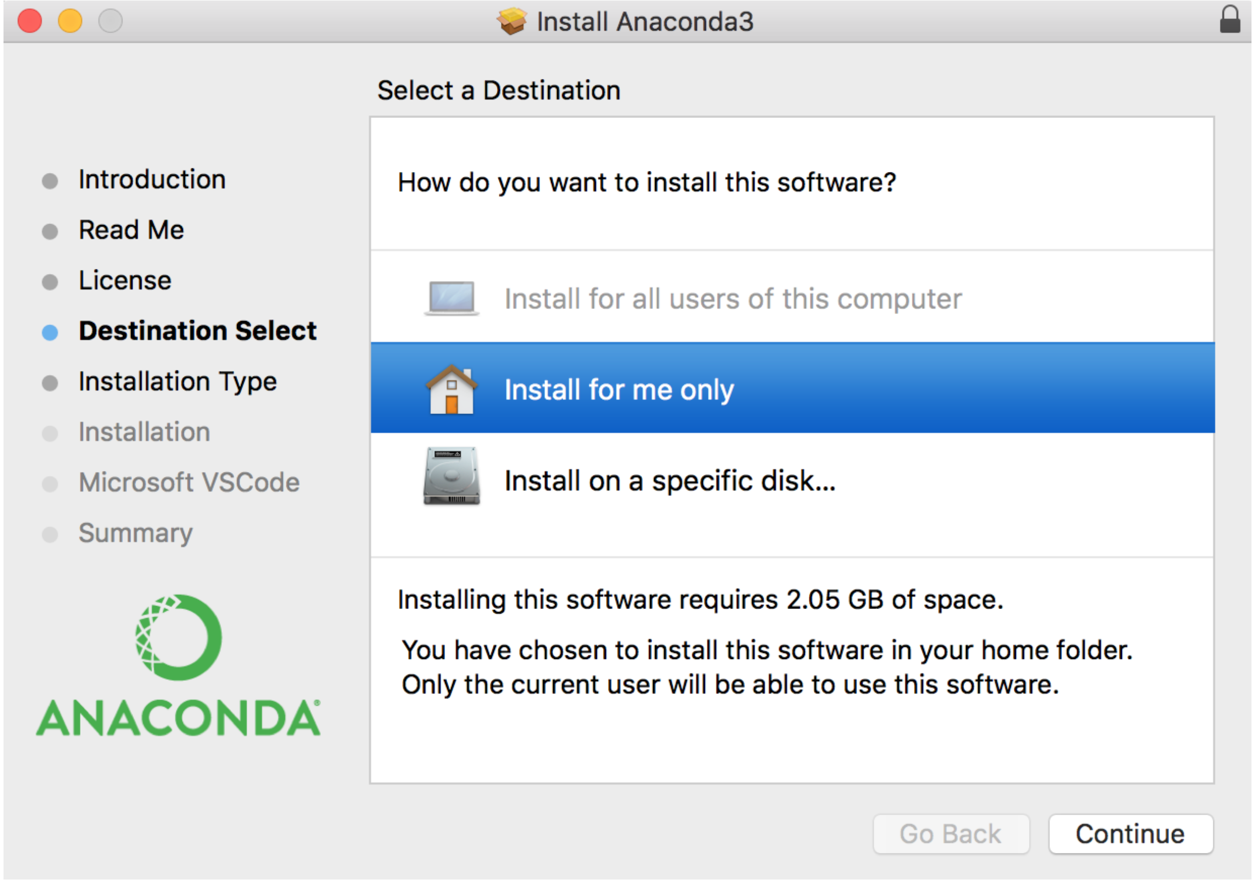
This will take 2.05 GB of space on your computer.

Click Install to perform a standard installation of this software in your home folder. Only the current user of this computer will be able to use this software.

6. OR, click the Change Install Location button to install in another location (not recommended).

   On the Destination Select screen, select Install for me only.

   NOTE: If you get the error message "You cannot install Anaconda in this location," reselect Install for me only.



Select a Destination

How do you want to install this software?

Install for all users of this computer

Install for me only

Install on a specific disk...

Installing this software requires 2.05 GB of space.

You have chosen to install this software in your home folder.
Only the current user will be able to use this software.

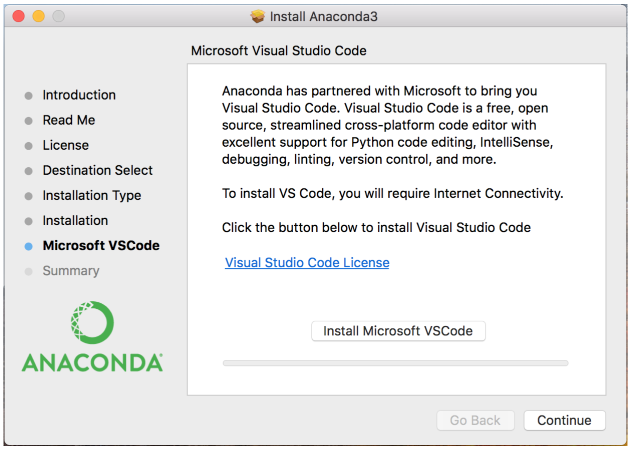
7. Click the continue button.

8. Optional: To *install VS Code*, click the Install Microsoft VS Code button. After the install completes click the Continue button.

   Or to install Anaconda without VS Code, click the Continue button.

   NOTE: Installing VS Code with the Anaconda installer requires an internet connection. Offline users may be able to find an offline VS Code installer from Microsoft.

9. A successful installation displays the following screen:



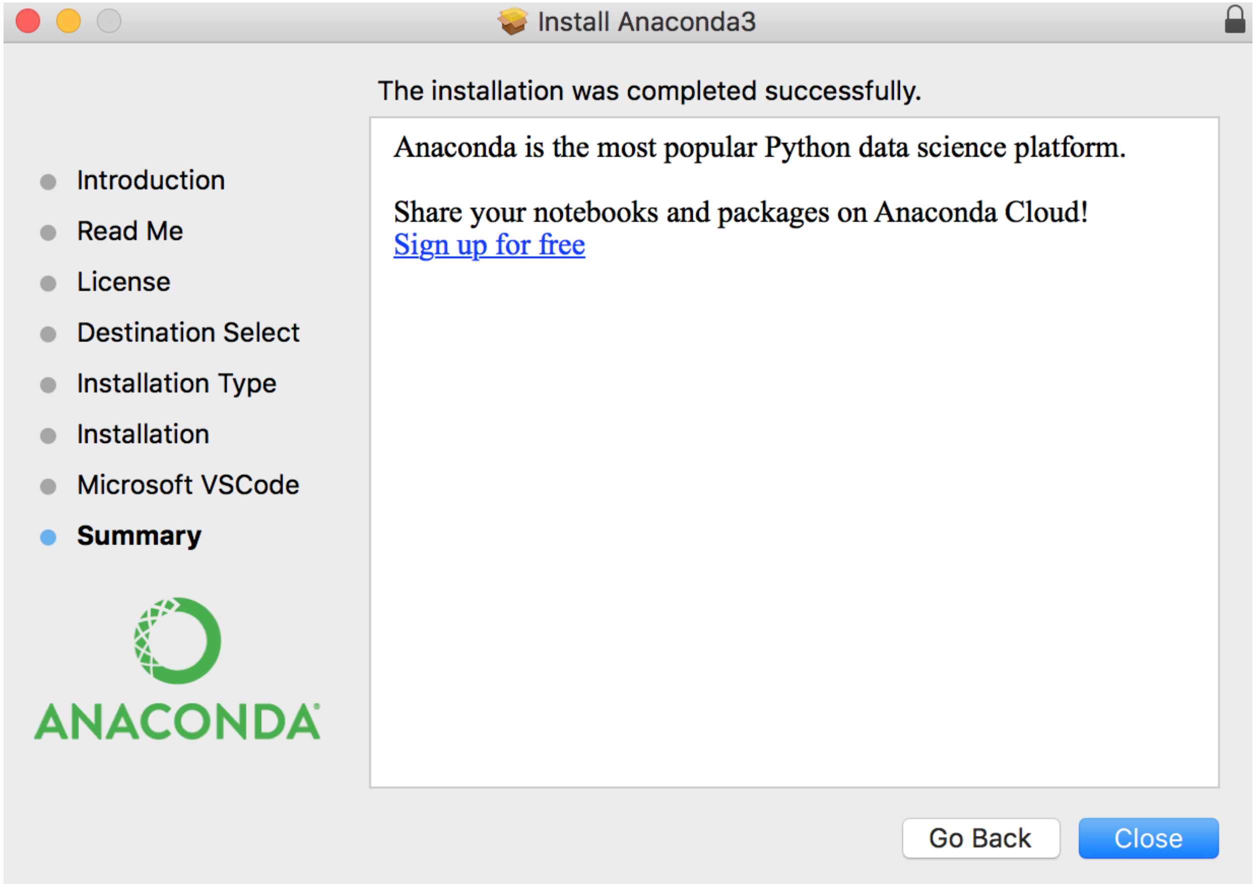
Microsoft Visual Studio Code

Anaconda has partnered with Microsoft to bring you Visual Studio Code. Visual Studio Code is a free, open source, streamlined cross-platform code editor with excellent support for Python code editing, IntelliSense, debugging, linting, version control, and more.

To install VS Code, you will require Internet Connectivity.

Click the button below to install Visual Studio Code

Visual Studio Code License



The installation was completed successfully.

Anaconda is the most popular Python data science platform.

Share your notebooks and packages on Anaconda Cloud!
Sign up for free

10. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from Launchpad, select Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our *Help page*

    TIP: Anaconda Navigator contains Jupyter Notebook and the Spyder IDE. For more information about using Navigator, see *Navigator*.

Using the command-line install

Use this method if you prefer to use a terminal window.

1. In your browser, download the command-line version of the macOS installer for your system.
2. OPTIONAL: *Verify data integrity with MD5 or SHA-256*. For more information on hashes, see What about cryptographic hash verification?
3. Install for Python 3.6 or 2.7:

   - For Python 3.6 enter the following:

     ```
     bash ~/Downloads/Anaconda3-5.2.0-MacOSX-x86_64.sh
     ```

   - For Python 2.7, open the Terminal.app or iTerm2 terminal application and then enter the following:

     ```
     bash ~/Downloads/Anaconda2-5.2.0-MacOSX-x86_64.sh
     ```

   NOTE: Include the `bash` command regardless of whether or not you are using the Bash shell.

   NOTE: Replace `~/Downloads` with your actual path and `Anaconda3-5.2.0-MacOSX-x86_64.sh` with actual name of the file you downloaded.

4. The installer prompts "In order to continue the installation process, please review the license agreement." Click Enter to view license terms.
5. Scroll to the bottom of the license terms and enter yes to agree to them.
6. The installer prompts you to Press Enter to confirm the location, Press CTRL-C to cancel the installation or specify an alternate installation directory. If you accept the default install location, the installer displays "PREFIX=/home/<user>/anaconda<2 or 3>" and continues the installation. It may take a few minutes to complete.

   NOTE: We recommend you accept the default install location.

7. The installer prompts "Do you wish the installer to prepend the Anaconda install location to PATH in your /home/<user>/.bash_profile ?" We recommend "yes".

   NOTE: If you enter "no", specify the path to Anaconda.

   - Add `export PATH="/<path to anaconda>/bin:$PATH"` in .bashrc or .bash_profile.
   - Or set the PATH variable: `export PATH="/<path to anaconda>/bin:$PATH"`

   Replace "<path to anaconda>" with the actual path to your Anaconda installation.

8. The installer finishes and displays "Thank you for installing Anaconda!"
9. The installer describes Microsoft VS Code and asks if you would like to install VS Code. Enter `yes` or `no`. If you selected `yes`, follow the instructions on screen to complete the VS Code installation.

   NOTE: Installing VS Code with the Anaconda installer requires an internet connection. Offline users may be able to find an offline VS Code installer from Microsoft.

10. Close and open your terminal window for the installation to take effect.

11. To verify the installation, see *Verifying your installation*.

To begin using Anaconda, see *Getting started with Anaconda*.

Installing on Linux

For x86 systems.

1. In your browser, download the Anaconda installer for Linux.
2. Optional: *Verify data integrity with MD5 or SHA-256*. (For more information on hashes, see cryptographic hash validation.)

   (a) Run the following:

   ```
   md5sum /path/filename
   ```

   OR:

   ```
   sha256sum /path/filename
   ```

   NOTE: Replace `/path/filename` with the actual path and filename of the file you downloaded.

   (b) Optional: Verify results against *the proper hash page* to make sure the hashes match.

3. Enter the following to install Anaconda for Python 3.6:

   ```
   bash ~/Downloads/Anaconda3-5.2.0-Linux-x86_64.sh
   ```

   OR Enter the following to install Anaconda for Python 2.7:

   ```
   bash ~/Downloads/Anaconda2-5.2.0-Linux-x86_64.sh
   ```

   NOTE: Include the `bash` command regardless of whether or not you are using Bash shell.

   NOTE: If you did not download to your Downloads directory, replace `~/Downloads/` with the path to the file you downloaded.

   NOTE: Choose "Install Anaconda as a user" unless root privileges are required.

4. The installer prompts "In order to continue the installation process, please review the license agreement." Click Enter to view license terms.
5. Scroll to the bottom of the license terms and enter "Yes" to agree.
6. The installer prompts you to click Enter to accept the default install location, CTRL-C to cancel the installation, or specify an alternate installation directory. If you accept the default install location, the installer displays "PREFIX=/home/<user>/anaconda<2 or 3>" and continues the installation. It may take a few minutes to complete.
7. The installer prompts "Do you wish the installer to prepend the Anaconda<2 or 3> install location to PATH in your /home/<user>/.bashrc ?" Enter `Yes`.

   NOTE: If you enter "No", you must manually add the path to Anaconda or conda will not work. See *FAQ*.

8. The installer describes Microsoft VS Code and asks if you would like to install VS Code. Enter `yes` or `no`. If you selected `yes`, follow the instructions on screen to complete the VS Code installation.

   NOTE: Installing VS Code with the Anaconda installer requires an internet connection. Offline users may be able to find an offline VS Code installer from Microsoft.

9. The installer finishes and displays "Thank you for installing Anaconda<2 or 3>!"

10. Close and open your terminal window for the installation to take effect, or you can enter the command `source ~/.bashrc`.

11. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: Open a Terminal window and type `anaconda-navigator`. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our *Help page*.

    TIP: Anaconda Navigator contains Jupyter Notebook and the Spyder IDE. For more information about Navigator, see *Navigator*.

After your install completes, start using Anaconda with the instructions in *Getting started with Anaconda*.

NOTE: If you install multiple versions of Anaconda, the system defaults to the most current version, as long as you haven't altered the default install path.

Installing on Linux POWER8 or POWER9

1. In your browser, download the Anaconda installer for POWER8 and POWER9.

2. Optional: *Verify data integrity with MD5 or SHA-256*. (For more information on hashes, see cryptographic hash validation.)

   (a) Run the following:

   ```
   md5sum /path/filename
   ```

   OR:

   ```
   sha256sum /path/filename
   ```

   NOTE: Replace `/path/filename` with the actual path and name of the downloaded file.

   (b) Optional: Verify results against *the proper hash page* to make sure the hashes match.

3. Enter the following to install Anaconda for Python 3.6:

   ```
   bash ~/Downloads/Anaconda3-5.2.0-Linux-ppc64le.sh
   ```

   OR enter the following to install Anaconda for Python 2.7:

   ```
   bash ~/Downloads/Anaconda2-5.2.0-Linux-ppc64le.sh
   ```

   NOTE: Include the `bash` command regardless of whether or not you are using Bash shell.

   NOTE: If you did not download to your Downloads directory, replace `~/Downloads/` with the path to the file you downloaded.

   NOTE: Choose "Install Anaconda as a user" unless root privileges are required.

4. The installer prompts "In order to continue the installation process, please review the license agreement." Click Enter to view license terms.

5. Scroll to the bottom of the license terms and enter "Yes" to agree to them.

6. The installer prompts you to click Enter to accept the default install location, CTRL-C to cancel the installation, or specify an alternate installation directory. If you accept the default install location, the installer displays "PREFIX=/home/<user>/anaconda<2 or 3>" and continues the installation. It may take a few minutes to complete.

7. The installer prompts "Do you wish the installer to prepend the Anaconda<2 or 3> install location to PATH in your /home/<user>/.bashrc ?" Enter `Yes`.

   NOTE: If you enter "No", you must manually add the path to Anaconda or conda will not work. See *FAQ*.

8. The installer finishes and displays "Thank you for installing Anaconda<2 or 3>!"

9. Close and open your terminal window for the installation to take effect, or you can enter the command `source ~/.bashrc`.

10. Verify your installation by entering `conda list`. If the installation was successful, the terminal window displays a list of installed Anaconda packages.

After your install completes, start using Anaconda with the instructions in *Getting started with Anaconda*.

NOTE: Anaconda on Power8 or Power9 only supports little endian mode.

NOTE: If you install multiple versions of Anaconda, the system defaults to the most current version, as long as you haven't altered the default install path.

Verifying your installation

You can confirm that Anaconda is installed and working with Anaconda Navigator or conda.

Anaconda Navigator

Open Anaconda Navigator, which is automatically installed when you install Anaconda.

- Windows: Click Start - then from the shortcuts, select Anaconda Navigator. If it opens, you have successfully installed Anaconda.
- macOS: Click Launchpad - then select Anaconda Navigator. If it opens, you have successfully installed Anaconda.
- Linux: See next section.

Conda

You can also use conda in an Anaconda prompt (Terminal on Linux or macOS).

To open Anaconda Prompt (or Terminal on Linux or macOS):

- Windows: Open the Anaconda Prompt (Click Start, select Anaconda Prompt)
- macOS: Open Launchpad, then open Terminal or iTerm.
- Linux–CentOS: Open Applications - System Tools - Terminal.
- Linux–Ubuntu: Open the Dash by clicking the upper left Ubuntu icon, then type "terminal".

After opening Anaconda prompt (Terminal on Linux or macOS), choose any of the following methods:

- Enter a command such as `conda list`. If Anaconda is installed and working, this will display a list of installed packages and their versions.
- Enter the command `python`. This command runs the Python shell. If Anaconda is installed and working, the version information it displays when it starts up will include "Anaconda". To exit the Python shell, enter the command `quit()`.

- Open Anaconda Navigator with the command `anaconda-navigator`. If Anaconda is installed properly, the graphical program Anaconda Navigator will open.

If you find any problems, see the *troubleshooting guide* and the *Help and support* page for resources such as free community support and bug reports.

Anaconda installer file hashes

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

	Python 2	Python 3
64-bit Windows	*64-bit Windows, Py2*	*64-bit Windows, Py3*
32-bit Windows	*32-bit Windows, Py2*	*32-bit Windows, Py3*
macOS graphical installer	*macOS, Py2*	*macOS, Py3*
macOS command line installer	*macOS, Py2*	*macOS, Py3*
64-bit Linux	*64-bit Linux, Py2*	*64-bit Linux, Py3*
32-bit Linux	*32-bit Linux, Py2*	*32-bit Linux, Py3*

Information for all files on a single page

Hashes for all files

Name	Size	Time modified	SHA256 hash
Anaconda3-5.2.0-Windows-x86_64.exe	631.3 MiB	2018-05-30 13:04:18	`2672f6537e2c8a79ae9540cf3c49b18bb9`
Anaconda3-5.2.0-Windows-x86.exe	506.3 MiB	2018-05-30 13:04:19	`64305a4c0041aaf4a3fd0fee4466d7b7f2`
Anaconda3-5.2.0-MacOSX-x86_64.sh	523.3 MiB	2018-05-30 13:07:03	`c8089121dc89ffe8f9a0c01205bab75a11`
Anaconda3-5.2.0-MacOSX-x86_64.pkg	613.1 MiB	2018-05-30 13:07:00	`dae8befc73d32b480faef31fa6fb733325`
Anaconda3-5.2.0-Linux-x86_64.sh	621.6 MiB	2018-05-30 13:05:43	`09f53738b0cd3bb96f5b1bac488e5528df`
Anaconda3-5.2.0-Linux-x86.sh	507.3 MiB	2018-05-30 13:05:46	`f3527d085d06f35b6aeb96be2a9253ff9e`
Anaconda3-5.2.0-Linux-ppc64le.sh	288.3 MiB	2018-05-30 13:05:40	`024c811526ffc40ed6fa243a25795fbab5`
Anaconda3-5.1.0-Windows-x86_64.exe	537.1 MiB	2018-02-15 09:10:26	`7d192e58915d7e7fbfd0c987ddc4db38a2`
Anaconda3-5.1.0-Windows-x86.exe	435.5 MiB	2018-02-15 09:10:28	`7a05da21fd592991d181ac8467faac5134`
Anaconda3-5.1.0-MacOSX-x86_64.sh	511.3 MiB	2018-02-15 09:10:24	`be705b3c3a0ca29ee32ce7658890bb5edb`
Anaconda3-5.1.0-MacOSX-x86_64.pkg	594.7 MiB	2018-02-15 09:09:06	`d6bf6309ccafa84314d85ca7421fddc160`
Anaconda3-5.1.0-Linux-x86_64.sh	551.2 MiB	2018-02-15 09:08:57	`7e6785caad25e33930bc03fac4994a434a`
Anaconda3-5.1.0-Linux-x86.sh	449.7 MiB	2018-02-15 09:08:58	`0e940272517d8f8a6f26316a19e4be2bda`
Anaconda3-5.1.0-Linux-ppc64le.sh	285.7 MiB	2018-02-15 09:08:56	`58d1d093450dabefef9279694c9345afed`
Anaconda3-5.0.1-Windows-x86_64.exe	514.8 MiB	2017-10-24 12:37:59	`0b1ec18b7425f8c8518d6dc2fc0bc8ec2f`
Anaconda3-5.0.1-Windows-x86.exe	420.4 MiB	2017-10-24 12:37:10	`9edc3012324c9c8c9aa5257688bd793277`
Anaconda3-5.0.1-MacOSX-x86_64.sh	491.0 MiB	2017-10-23 19:51:10	`f438a0af923bc1edc7bca53f496c59a668`
Anaconda3-5.0.1-MacOSX-x86_64.pkg	568.9 MiB	2017-10-23 20:01:19	`50c28594c785f5828990c9505346848856`
Anaconda3-5.0.1-Linux-x86_64.sh	525.3 MiB	2017-10-23 17:52:55	`55e4db1919f49c92d5abbf27a4be5986ae`

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda3-5.0.1-Linux-x86.sh	431.0 MiB	2017-10-23 18:07:51	991a4b656fcb0236864fbb27ff03bb7f3d
Anaconda3-5.0.0.1-Linux-x86_64.sh	524.0 MiB	2017-10-02 10:50:14	092c92427f44687d789a41922ce8426fbd
Anaconda3-5.0.0.1-Linux-x86.sh	429.8 MiB	2017-10-02 10:50:15	407576899d3aa546bc3c2c4a13cbc18ab5
Anaconda3-5.0.0-Windows-x86_64.exe	510.0 MiB	2017-09-26 14:14:53	53bd80727099b5767b9f20f99e908f9c19
Anaconda3-5.0.0-Windows-x86.exe	415.8 MiB	2017-09-26 16:25:12	a0d5d8e328b1d3a1ed921cadeecda659c4
Anaconda3-5.0.0-MacOSX-x86_64.sh	489.9 MiB	2017-09-26 16:25:11	23df1e3a38a6b4aaa0ab559d0c1e51be76
Anaconda3-5.0.0-MacOSX-x86_64.pkg	567.2 MiB	2017-09-26 16:25:10	06d959384869290845bc61346bb33a18dd
Anaconda3-5.0.0-Linux-x86_64.sh	523.4 MiB	2017-09-26 14:37:22	67f5c20232a3e493ea3f19a8e273e0618a
Anaconda3-5.0.0-Linux-x86.sh	429.3 MiB	2017-09-26 14:48:02	634d2dfa97d19f2cc15e941cb4d059bc83
Anaconda3-5.0.0-Linux-ppc64le.sh	296.3 MiB	2017-09-25 14:39:31	3574d423084e604a9d85a9f38ea481e0fc
Anaconda3-4.4.0.1-Linux-ppc64le.sh	285.6 MiB	2017-07-26 16:08:42	d7c367c9c4fffec37c31c6570218c99448
Anaconda3-4.4.0-Windows-x86_64.exe	437.6 MiB	2017-05-26 17:55:34	ea582602541e748053df55051446042620
Anaconda3-4.4.0-Windows-x86.exe	362.2 MiB	2017-05-26 17:54:21	37afe00b8305cc09b7bd8dd07f65cec3f4
Anaconda3-4.4.0-MacOSX-x86_64.sh	380.4 MiB	2017-05-26 18:35:59	10fe58f09ae524df2548d17b8bb1e75db1
Anaconda3-4.4.0-MacOSX-x86_64.pkg	442.5 MiB	2017-05-26 18:36:17	c5fc645f11505ac3ef710023b4072b7fb2
Anaconda3-4.4.0-Linux-x86_64.sh	499.0 MiB	2017-05-26 18:23:04	3301b37e402f3ff3df216fe0458f1e6a4c
Anaconda3-4.4.0-Linux-x86.sh	428.7 MiB	2017-05-26 18:23:45	b0e492206d43067314b25963bc7d1f0120
Anaconda3-4.4.0-Linux-ppc64le.sh	290.7 MiB	2017-05-17 15:45:50	605251829edecd0c39df8db856d4f09e40
Anaconda3-4.3.1-Windows-x86_64.exe	422.1 MiB	2017-03-06 16:20:48	65ce6d7c09884935fee9eb8d318b30e95f
Anaconda3-4.3.1-Windows-x86.exe	348.1 MiB	2017-03-06 16:19:46	adf322f49542cf509d4f72152cea24e54e
Anaconda3-4.3.1-MacOSX-x86_64.sh	363.4 MiB	2017-03-06 16:26:09	a42267203e207cb5e0f539e0d879ead12e
Anaconda3-4.3.1-MacOSX-x86_64.pkg	424.1 MiB	2017-03-06 16:26:27	ca608d58b1acf77b5c77d10e937b9084e5
Anaconda3-4.3.1-Linux-x86_64.sh	474.3 MiB	2017-03-06 16:12:24	4447b93d2c779201e5fb50cfc45de0ec96
Anaconda3-4.3.1-Linux-x86.sh	399.3 MiB	2017-03-06 16:12:47	7b70bdba282a18ddbdc167afe8131f7532
Anaconda3-4.3.0.1-Windows-x86_64.exe	421.2 MiB	2017-02-02 14:22:10	b5954bf7da9a92d351d905dfdfa0e7bee1
Anaconda3-4.3.0.1-Windows-x86.exe	347.2 MiB	2017-02-02 14:21:02	7f8ffce6b2c3a968ce19171c9dc332dec6
Anaconda3-4.3.0-Windows-x86_64.exe	421.2 MiB	2017-01-27 14:19:41	324568dbef777a6ac8a25c1e8ae1975ffb
Anaconda3-4.3.0-Windows-x86.exe	347.2 MiB	2017-01-27 14:18:45	4a5dfea30b926074b4d6e0f1cea3e97657
Anaconda3-4.3.0-MacOSX-x86_64.sh	362.6 MiB	2017-01-27 14:26:15	c53059b810c5e7a9a5ef9c46a7ed76675d
Anaconda3-4.3.0-MacOSX-x86_64.pkg	423.1 MiB	2017-01-27 14:26:32	f4522ac099ba292940bb47429e8e53eb9a
Anaconda3-4.3.0-Linux-x86_64.sh	473.4 MiB	2017-01-27 14:15:21	e9169c3a5029aa820393ac92704eb9ee07
Anaconda3-4.3.0-Linux-x86.sh	398.4 MiB	2017-01-27 14:14:29	f7ce2eeec3e42c2ba1ee3b9fcd670478fd
Anaconda3-4.2.0-Windows-x86_64.exe	391.4 MiB	2016-09-27 15:57:21	84e30c99833e142a27fc9ee2c748b03f16
Anaconda3-4.2.0-Windows-x86.exe	333.4 MiB	2016-09-27 15:56:30	e7b79a9886da3f840b52882c47ecab3eda
Anaconda3-4.2.0-MacOSX-x86_64.sh	349.5 MiB	2016-09-27 15:50:07	95448921601e1952e01a17ba9767cd3621
Anaconda3-4.2.0-MacOSX-x86_64.pkg	407.1 MiB	2016-10-17 19:33:47	44fe57910aa10967c4afe41ab5663cb492
Anaconda3-4.2.0-Linux-x86_64.sh	455.9 MiB	2016-09-27 15:50:04	73b51715a12b6382dd4df3dd1905b531bd
Anaconda3-4.2.0-Linux-x86.sh	373.9 MiB	2016-09-27 15:50:34	1a8320635f2f06ec9d8610e77d6d0f9cb2
Anaconda3-4.1.1-Windows-x86_64.exe	352.9 MiB	2016-07-08 11:21:17	b4889513dc574f9d6f96db089315d69d29
Anaconda3-4.1.1-Windows-x86.exe	293.8 MiB	2016-07-08 11:21:18	224e3dd90850651ae0d1c9216b4c317d1d
Anaconda3-4.1.1-MacOSX-x86_64.sh	298.7 MiB	2016-07-08 11:21:17	7c3c06e9281c41f1213d357cb5f233fd99
Anaconda3-4.1.1-MacOSX-x86_64.pkg	347.9 MiB	2016-07-08 11:21:15	b5e8cf44958d0aa03a7cc2da15fa835b1e
Anaconda3-4.1.1-Linux-x86_64.sh	406.3 MiB	2016-07-08 11:20:02	4f5c95feb0e7efeadd3d348dcef117d778
Anaconda3-4.1.1-Linux-x86.sh	329.1 MiB	2016-07-08 11:20:02	931626363f4030c7a1e8897549b1d3589d
Anaconda3-4.1.0-Windows-x86_64.exe	351.4 MiB	2016-06-28 11:28:33	9acde60b591233452dba23ac15800f39f2
Anaconda3-4.1.0-Windows-x86.exe	292.6 MiB	2016-06-28 11:28:34	4f444ed9400505e822bb475e986800fac0
Anaconda3-4.1.0-MacOSX-x86_64.sh	297.6 MiB	2016-06-28 11:28:33	4c45c8d75665fa5194ebe4e355d3427f5a
Anaconda3-4.1.0-MacOSX-x86_64.pkg	346.7 MiB	2016-06-28 11:28:32	83772b5fcd3d6deb945316ec96ecc7b0cd
Anaconda3-4.1.0-Linux-x86_64.sh	405.0 MiB	2016-06-28 11:28:31	11d32cf4026603d3b327dc4299863be6b8

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda3-4.1.0-Linux-x86.sh	328.4 MiB	2016-06-28 11:28:32	7764093f337a43e4962b12d01508c1a385
Anaconda3-4.0.0-Windows-x86_64.exe	345.4 MiB	2016-03-29 11:16:22	39bf467cd142c1f8fdb7d673fdea273d87
Anaconda3-4.0.0-Windows-x86.exe	283.1 MiB	2016-03-29 11:16:22	b5a31a9d130a40c3110c0592a6c8fbd43a
Anaconda3-4.0.0-MacOSX-x86_64.sh	292.7 MiB	2016-03-29 11:16:21	704a776c0cf3fcca6e0c5a1e6b60437282
Anaconda3-4.0.0-MacOSX-x86_64.pkg	341.5 MiB	2016-03-29 11:16:08	32a089b1be465a8b03c837041bbfbcb761
Anaconda3-4.0.0-Linux-x86_64.sh	398.4 MiB	2016-03-29 11:15:02	36a558a1109868661a5735f5f32607643f
Anaconda3-4.0.0-Linux-x86.sh	336.9 MiB	2016-03-29 11:15:03	e1469fa0d24de12f33661ce3d7a06d7796
Anaconda3-2.5.0-Windows-x86_64.exe	361.1 MiB	2016-02-03 15:47:40	4728044d77da715e48d4c95d7f2e3c2a02
Anaconda3-2.5.0-Windows-x86.exe	296.3 MiB	2016-02-03 15:46:53	4a3441aaaa269d06f39e1430155f9f25a8
Anaconda3-2.5.0-MacOSX-x86_64.sh	318.3 MiB	2016-02-03 15:41:27	9bb0f926927db210f8c2a8de881213d1a4
Anaconda3-2.5.0-MacOSX-x86_64.pkg	369.8 MiB	2016-02-03 15:56:04	b1a6945f0f025086806624c59de5d92e52
Anaconda3-2.5.0-Linux-x86_64.sh	395.6 MiB	2016-02-03 15:41:27	addadcb927f15cb0b5b6e36890563d3352
Anaconda3-2.5.0-Linux-x86.sh	334.4 MiB	2016-02-03 15:42:07	22ac26c8bde7c4153ea859f6f6d8aca93b
Anaconda3-2.4.1-Windows-x86_64.exe	363.7 MiB	2015-12-08 15:00:55	21d155a4b43805042499b8d008835bcdfd
Anaconda3-2.4.1-Windows-x86.exe	299.1 MiB	2015-12-08 15:00:56	dfe50d13473547b5230f6194dfe6bdff96
Anaconda3-2.4.1-MacOSX-x86_64.sh	213.9 MiB	2015-12-08 15:00:54	22a3267638da9b7d64210d7da90d8762da
Anaconda3-2.4.1-MacOSX-x86_64.pkg	247.6 MiB	2015-12-08 15:00:54	95e9f2d370f7816ed72b862c9413c973ef
Anaconda3-2.4.1-Linux-x86_64.sh	270.7 MiB	2015-12-08 15:00:53	0735e69199fc37135930ea2fd4fb6ad0ad
Anaconda3-2.4.1-Linux-x86.sh	253.2 MiB	2015-12-08 15:00:53	00d13413f5b8129e863dabcc2296a181c6
Anaconda3-2.4.0-Windows-x86_64.exe	392.3 MiB	2015-11-02 16:22:24	beaa1b803dd30022c6aca1c6f05182beae
Anaconda3-2.4.0-Windows-x86.exe	316.4 MiB	2015-11-02 16:22:25	a69a9fe00ce337b0cfd7d024b79ba5141c
Anaconda3-2.4.0-MacOSX-x86_64.sh	233.8 MiB	2015-11-02 16:22:23	f0cd785dbed0bab28dfc08a391c9de1b01
Anaconda3-2.4.0-MacOSX-x86_64.pkg	267.4 MiB	2015-11-02 16:22:23	791f045258bd39bbcdbb4c5425dce082ec
Anaconda3-2.4.0-Linux-x86_64.sh	285.2 MiB	2015-11-02 16:22:22	fb4e480059e991f2fa632b5a9bcdd284c7
Anaconda3-2.4.0-Linux-x86.sh	277.4 MiB	2015-11-02 16:22:22	f6080c6493cefc603cfeb67aaf6c3c4c6b
Anaconda3-2.3.0-Windows-x86_64.exe	336.4 MiB	2015-07-01 13:35:14	bd693b61cf191666ae0473327f3c15bcf3
Anaconda3-2.3.0-Windows-x86.exe	281.1 MiB	2015-07-01 13:35:15	02d5f84da308f96d1a252a6669f3ca91e1
Anaconda3-2.3.0-MacOSX-x86_64.sh	257.4 MiB	2015-07-01 13:35:14	6a0c94a49f41f9fda0138c8e966bd7b0a8
Anaconda3-2.3.0-MacOSX-x86_64.pkg	292.8 MiB	2015-07-01 13:35:13	0b936ab3067bbf32b5a52768f31ff437f0
Anaconda3-2.3.0-Linux-x86_64.sh	336.7 MiB	2015-07-01 13:35:12	3be5410b2d9db45882c7de07c554cf4f10
Anaconda3-2.3.0-Linux-x86.sh	322.6 MiB	2015-07-01 13:35:13	4cc10d65c303191004ada2b6d75562c8ed
Anaconda3-2.2.0-Windows-x86_64.exe	332.6 MiB	2015-03-25 15:32:30	28c5a13b27a9dbd57c7c633316c5f4beb0
Anaconda3-2.2.0-Windows-x86.exe	277.7 MiB	2015-03-25 15:31:19	20c46fff048fb313aaf1a49171c1a7b96a
Anaconda3-2.2.0-MacOSX-x86_64.sh	254.5 MiB	2015-03-25 15:17:44	81a2089ea6127717f146454e99ea0be2bd
Anaconda3-2.2.0-MacOSX-x86_64.pkg	288.8 MiB	2015-03-25 15:28:12	16a5154267d7d52d3e7e0d12ec3405077d
Anaconda3-2.2.0-Linux-x86_64.sh	326.9 MiB	2015-03-25 15:20:14	4aac68743e7706adb93f042f970373a6e7
Anaconda3-2.2.0-Linux-x86.sh	313.3 MiB	2015-03-25 15:20:07	223655cd256aa912dfc83ab24570e47bb3
Anaconda3-2.1.0-Windows-x86_64.exe	363.3 MiB	2014-09-25 11:11:03	ea4059469b1820069f62bd6c256def6259
Anaconda3-2.1.0-Windows-x86.exe	308.7 MiB	2014-09-25 11:09:01	8ffa252aa2b4f63889888ae85a81626ce9
Anaconda3-2.1.0-MacOSX-x86_64.sh	243.5 MiB	2014-09-25 10:53:23	efdb7e9d1e539cbed62dc3874b0de6a141
Anaconda3-2.1.0-MacOSX-x86_64.pkg	277.3 MiB	2014-09-25 11:40:54	2780df02f400e44c0adcd209825fdcf955
Anaconda3-2.1.0-Linux-x86_64.sh	332.8 MiB	2014-09-25 10:50:20	af3225ccbe8df0ffb918939e009aa57740
Anaconda3-2.1.0-Linux-x86.sh	317.7 MiB	2014-09-25 10:50:35	657cb599004c21e37ce693515ea33922e0
Anaconda3-2.0.1-Windows-x86_64.exe	319.8 MiB	2014-06-12 15:00:15	e2b6d3d6a9e378fc0d0dd63342417c02bd
Anaconda3-2.0.1-Windows-x86.exe	265.7 MiB	2014-06-12 15:00:27	b08803296d7439413d590fd1f967b20127
Anaconda3-2.0.1-MacOSX-x86_64.sh	203.3 MiB	2014-06-12 15:00:09	7a08509d4e45efcc7055a6d06d8406a773
Anaconda3-2.0.1-MacOSX-x86_64.pkg	230.7 MiB	2014-06-12 15:00:05	0d53815a83a50bdcfcb5ada686f582730b
Anaconda3-2.0.1-Linux-x86_64.sh	304.8 MiB	2014-06-12 14:59:53	3c3b834793e461f3316ad1d9a9178c6785
Anaconda3-2.0.1-Linux-x86.sh	287.7 MiB	2014-06-12 15:00:00	21293fabbd3d5cfbb1afe0c9a8b39e0bc4

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda3-2.0.0-Windows-x86_64.exe	310.9 MiB	2014-05-27 17:01:42	`a8046fc82da7463ef53cdeaba97c72433c`
Anaconda3-2.0.0-Windows-x86.exe	256.9 MiB	2014-05-27 16:59:13	`37986ce4c104ed3c82838de74b3a4de179`
Anaconda3-2.0.0-MacOSX-x86_64.sh	195.3 MiB	2014-05-27 17:02:53	`776a1cf8a8e898b41bb6558c093632cc92`
Anaconda3-2.0.0-MacOSX-x86_64.pkg	222.9 MiB	2014-05-27 16:30:16	`4d4189ec0c514d344389e216b3ad4eeacd`
Anaconda3-2.0.0-Linux-x86_64.sh	294.4 MiB	2014-05-27 16:26:59	`57ce4f97e300cf94c5724f72d992e9eece`
Anaconda3-2.0.0-Linux-x86.sh	277.5 MiB	2014-05-27 16:35:55	`439761159d5604e182951650a478dd53ca`
Anaconda2-5.2.0-Windows-x86_64.exe	564.0 MiB	2018-05-30 13:04:16	`e5ff95332d08a7b006a5bb723e0a5124c4`
Anaconda2-5.2.0-Windows-x86.exe	443.4 MiB	2018-05-30 13:04:17	`2b81916c477e64db917821bb48a97000fa`
Anaconda2-5.2.0-MacOSX-x86_64.sh	527.1 MiB	2018-05-30 13:05:34	`d7d46e566306da5979cd5632079497fe61`
Anaconda2-5.2.0-MacOSX-x86_64.pkg	616.8 MiB	2018-05-30 13:05:32	`f7695a3571eb8e8ae71fe9f413c36f57c9`
Anaconda2-5.2.0-Linux-x86_64.sh	603.4 MiB	2018-05-30 13:04:33	`cb0d7a08b0e2cec4372033d3269979b4e7`
Anaconda2-5.2.0-Linux-x86.sh	488.7 MiB	2018-05-30 13:05:30	`402758c24767e9eb3b77312c388725a058`
Anaconda2-5.2.0-Linux-ppc64le.sh	269.6 MiB	2018-05-30 13:04:31	`a8fcac3f0884520c35103e76549fcc45d6`
Anaconda2-5.1.0-Windows-x86_64.exe	522.6 MiB	2018-02-15 09:08:54	`3674c8d8c233dbea30842f14dc76cc3fea`
Anaconda2-5.1.0-Windows-x86.exe	419.8 MiB	2018-02-15 09:08:55	`fa78c71d88b01e6367f0c3cbd23da1f82e`
Anaconda2-5.1.0-MacOSX-x86_64.sh	505.9 MiB	2018-02-15 09:08:53	`b686e01aeadb33526d9c154a0ac6f691df`
Anaconda2-5.1.0-MacOSX-x86_64.pkg	588.0 MiB	2018-02-15 09:08:52	`edbe9ef1ee5cfe62e131d7650e07c031ab`
Anaconda2-5.1.0-Linux-x86_64.sh	533.0 MiB	2018-02-15 09:08:50	`5f26ee92860d1dffdcd20910ff2cf75572`
Anaconda2-5.1.0-Linux-x86.sh	431.3 MiB	2018-02-15 09:08:51	`5af0c7a09a5f3aaf3666c0b362246d342d`
Anaconda2-5.1.0-Linux-ppc64le.sh	267.3 MiB	2018-02-15 09:08:49	`ff9baa4d3710bb24bc3a6a40c0f4ef6915`
Anaconda2-5.0.1-Windows-x86_64.exe	499.8 MiB	2017-10-23 21:57:22	`c43f94c51623850b0c1a826710fe9c8e50`
Anaconda2-5.0.1-Windows-x86.exe	403.4 MiB	2017-10-24 12:08:14	`1a50fac8644f2128e318337b218299e53e`
Anaconda2-5.0.1-MacOSX-x86_64.sh	486.5 MiB	2017-10-23 19:51:04	`e3a9a5c84cb89ff079b0781ba773a3433d`
Anaconda2-5.0.1-MacOSX-x86_64.pkg	562.8 MiB	2017-10-23 20:01:12	`22350fe830e6786a263d7727e537f066b1`
Anaconda2-5.0.1-Linux-x86_64.sh	507.7 MiB	2017-10-24 12:13:52	`23c676510bc87c95184ecaeb327c0b2c88`
Anaconda2-5.0.1-Linux-x86.sh	413.2 MiB	2017-10-24 12:13:07	`88c8d698fff16af15862daca10e94a0a46`
Anaconda2-5.0.0.1-Linux-x86_64.sh	506.3 MiB	2017-10-02 10:50:12	`18730808d863a5c194ab3f59dd395c1a63`
Anaconda2-5.0.0.1-Linux-x86.sh	411.9 MiB	2017-10-02 10:50:13	`00fbd979c815ede0bbad48fb4ef62cda33`
Anaconda2-5.0.0-Windows-x86_64.exe	498.2 MiB	2017-09-26 14:30:49	`5fb73395cdf003613f5d44844da9870dbd`
Anaconda2-5.0.0-Windows-x86.exe	402.2 MiB	2017-09-26 16:25:09	`078551cfb0df72779897026724f375671e`
Anaconda2-5.0.0-MacOSX-x86_64.sh	485.3 MiB	2017-09-26 16:25:09	`d85198c63657924fae11b6ea5961f50d81`
Anaconda2-5.0.0-MacOSX-x86_64.pkg	561.3 MiB	2017-09-26 16:25:08	`3ee5cfe80d51685d6f374f83a9b76fa7ec`
Anaconda2-5.0.0-Linux-x86_64.sh	505.7 MiB	2017-09-26 14:37:21	`58a7117f89c40275114bf7e824a613a963`
Anaconda2-5.0.0-Linux-x86.sh	411.4 MiB	2017-09-26 14:48:02	`a3ed8769d20d55a41c04cf7c04e81c9597`
Anaconda2-5.0.0-Linux-ppc64le.sh	282.3 MiB	2017-09-26 16:25:07	`e0512f3c81251e5dcd48fcf02fe2044a66`
Anaconda2-4.4.0.1-Linux-ppc64le.sh	271.4 MiB	2017-07-26 16:10:02	`e14acab146181699e47ca108fc624eceba`
Anaconda2-4.4.0-Windows-x86_64.exe	430.7 MiB	2017-05-26 17:53:22	`7a8ec1a36f385ebf28a1a8cf63b8b03ac0`
Anaconda2-4.4.0-Windows-x86.exe	354.4 MiB	2017-05-26 17:52:15	`0decd861f8839fdf2cbe4fa306c127f69e`
Anaconda2-4.4.0-MacOSX-x86_64.sh	375.4 MiB	2017-05-26 18:35:52	`ab95aef1110c2a385fd39a17e5f11dfbaa`
Anaconda2-4.4.0-MacOSX-x86_64.pkg	438.0 MiB	2017-05-26 18:36:08	`e5acf026892eaebb055e6317af96f295d3`
Anaconda2-4.4.0-Linux-x86_64.sh	485.2 MiB	2017-05-26 18:22:48	`2d30b91ed4d215b6b4a15162a3389e9057`
Anaconda2-4.4.0-Linux-x86.sh	415.0 MiB	2017-05-26 18:23:30	`452aa91ac83d3b6a68b79cea3042170ec5`
Anaconda2-4.4.0-Linux-ppc64le.sh	276.6 MiB	2017-05-17 15:45:20	`c19edfd9a3bd2fcb37ddb0c3aa09339c9e`
Anaconda2-4.3.1-Windows-x86_64.exe	413.7 MiB	2017-03-06 16:18:59	`c0e13a756a856d7b7757b10d65bee577d8`
Anaconda2-4.3.1-Windows-x86.exe	339.0 MiB	2017-03-06 16:18:12	`fc363cea3c321c17b43a0bf2137aa845fe`
Anaconda2-4.3.1-MacOSX-x86_64.sh	358.2 MiB	2017-03-06 16:26:02	`35261360f2b01793f441b29715a94052dc`
Anaconda2-4.3.1-MacOSX-x86_64.pkg	419.4 MiB	2017-03-06 16:26:18	`f5d950451c038f9a7ca80d4036b6a8152c`
Anaconda2-4.3.1-Linux-x86_64.sh	462.0 MiB	2017-03-06 16:12:14	`e9b8f2645df6b1527ba56d61343162e079`
Anaconda2-4.3.1-Linux-x86.sh	387.7 MiB	2017-03-06 16:12:31	`4519ac724d5120d21bb80289c5509c0d1f`

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda2-4.3.0.1-Windows-x86_64.exe	412.8 MiB	2017-02-02 14:20:08	2198e28e9e8e3c43ab72a8371e5b2d0a9a
Anaconda2-4.3.0.1-Windows-x86.exe	338.1 MiB	2017-02-02 14:19:05	863702665aa2b55ede2103a8ca7d26435e
Anaconda2-4.3.0-Windows-x86_64.exe	412.8 MiB	2017-01-27 14:17:59	1117839746a8eabf7ed26ff311fc74e44a
Anaconda2-4.3.0-Windows-x86.exe	338.1 MiB	2017-01-27 14:17:06	a98767acefdeda02fe8d3ef9dadda1a343
Anaconda2-4.3.0-MacOSX-x86_64.sh	357.3 MiB	2017-01-27 14:26:08	834ac0287062929ab5930661735ee617fd
Anaconda2-4.3.0-MacOSX-x86_64.pkg	418.4 MiB	2017-01-27 14:26:23	3e1d1026d2c0b87213a8b4a5f28431060b
Anaconda2-4.3.0-Linux-x86_64.sh	461.1 MiB	2017-01-27 14:15:08	7c52e6e99aabb24a49880130615a48e685
Anaconda2-4.3.0-Linux-x86.sh	386.8 MiB	2017-01-27 14:14:15	b80d471839e8cf7b100e59308720cc13c1
Anaconda2-4.2.0-Windows-x86_64.exe	381.0 MiB	2016-09-27 15:55:47	6254b150edee53000c94e9abfc9c51a2d2
Anaconda2-4.2.0-Windows-x86.exe	324.1 MiB	2016-09-27 15:54:50	a97ca79cb771568d051ef7773d25c0dda4
Anaconda2-4.2.0-MacOSX-x86_64.sh	346.4 MiB	2016-09-27 15:50:02	a8b3ef86233635d9dcc3499dc384980762
Anaconda2-4.2.0-MacOSX-x86_64.pkg	403.9 MiB	2016-10-17 19:33:11	4a74d34c3a3a82df31673ab49497816b03
Anaconda2-4.2.0-Linux-x86_64.sh	446.0 MiB	2016-09-27 15:49:54	beee286d24fb37dd6555281bba39b3deb5
Anaconda2-4.2.0-Linux-x86.sh	365.0 MiB	2016-09-27 15:50:20	618b720f309fe8da4f235415f11b6ce3db
Anaconda2-4.1.1-Windows-x86_64.exe	341.2 MiB	2016-07-08 11:20:01	7be13a69df254b86e47612c726b0b2ba9f
Anaconda2-4.1.1-Windows-x86.exe	286.0 MiB	2016-07-08 11:20:01	4708d73952a0a8040bf1594ea42027a30e
Anaconda2-4.1.1-MacOSX-x86_64.sh	295.8 MiB	2016-07-08 11:20:00	3b2fb323eb26c1c58788f63c41e164c20c
Anaconda2-4.1.1-MacOSX-x86_64.pkg	345.0 MiB	2016-07-08 11:19:59	879385461cc65bd9dbf9639bbf4471ecf1
Anaconda2-4.1.1-Linux-x86_64.sh	399.6 MiB	2016-07-08 11:19:56	9413b1d3ca9498ba6f53913df9c43d685d
Anaconda2-4.1.1-Linux-x86.sh	324.6 MiB	2016-07-08 11:19:57	1ab001c7a469345a90d549ebf4afa3376f
Anaconda2-4.1.0-Windows-x86_64.exe	340.2 MiB	2016-06-28 11:28:30	7a62880ff9bb7f747d70f518f024dfd179
Anaconda2-4.1.0-Windows-x86.exe	285.1 MiB	2016-06-28 11:28:31	c4ad4eefdffb6d838424c62c8b524352d8
Anaconda2-4.1.0-MacOSX-x86_64.sh	295.1 MiB	2016-06-28 11:28:30	8b2c2a32f5e0da75cf8c81c568124cc1ea
Anaconda2-4.1.0-MacOSX-x86_64.pkg	344.2 MiB	2016-06-28 11:28:29	a97840be50d8c86b28caf8be1786bbe748
Anaconda2-4.1.0-Linux-x86_64.sh	398.8 MiB	2016-06-28 11:28:28	3b7e504ca0132fb555d1f10e174cae0700
Anaconda2-4.1.0-Linux-x86.sh	324.4 MiB	2016-06-28 11:28:28	54c06cd1b11cb687db6ba3613df443c057
Anaconda2-4.0.0-Windows-x86_64.exe	334.6 MiB	2016-03-29 11:14:59	213c7d94bdb6f0931edd31bb14ae33ab55
Anaconda2-4.0.0-Windows-x86.exe	281.0 MiB	2016-03-29 11:15:00	f8185ad2fe89356ab001e55a463b663bcb
Anaconda2-4.0.0-MacOSX-x86_64.sh	290.2 MiB	2016-03-29 11:14:59	aa7ba6e1a40e08e672660c00c3151f0124
Anaconda2-4.0.0-MacOSX-x86_64.pkg	339.2 MiB	2016-03-29 11:14:57	242691c7dc9e20143d7620fd9e0cc344fe
Anaconda2-4.0.0-Linux-x86_64.sh	392.5 MiB	2016-03-29 11:14:55	ae312143952ca00e061a656c2080e0e4fd
Anaconda2-4.0.0-Linux-x86.sh	332.3 MiB	2016-03-29 11:14:57	41341c840cea4185ef5bd82520c1de72b4
Anaconda2-2.5.0-Windows-x86_64.exe	348.6 MiB	2016-02-03 15:46:14	4423b43eb23184b4239abc426a564760d6
Anaconda2-2.5.0-Windows-x86.exe	296.2 MiB	2016-02-03 15:45:21	a0336729f0400ff12fe18a7d5e20c3f9b4
Anaconda2-2.5.0-MacOSX-x86_64.sh	316.1 MiB	2016-02-03 15:41:22	e7aa3b41210ee7ccf3c12e5b5ea43190d1
Anaconda2-2.5.0-MacOSX-x86_64.pkg	367.9 MiB	2016-02-03 15:55:31	0f546ed4f388299824e98a31ca9e3fe982
Anaconda2-2.5.0-Linux-x86_64.sh	390.9 MiB	2016-02-03 15:41:18	e10abf459cde4a838bd6fc5ca03023c340
Anaconda2-2.5.0-Linux-x86.sh	330.4 MiB	2016-02-03 15:41:54	4911047df51c46661f551d6022aee21a7e
Anaconda2-2.4.1-Windows-x86_64.exe	354.2 MiB	2015-12-08 15:00:51	cfbe5539cb7f2e5807ec3d2fa2e59db3a4
Anaconda2-2.4.1-Windows-x86.exe	287.8 MiB	2015-12-08 15:00:52	65fb15559b0ddb5055c110ecdb84823a6f
Anaconda2-2.4.1-MacOSX-x86_64.sh	212.0 MiB	2015-12-08 15:00:51	f4bd45a21e0dff106e36d11cfd532f2b50
Anaconda2-2.4.1-MacOSX-x86_64.pkg	245.8 MiB	2015-12-08 15:00:50	1e2445aaf9faf84e801404bf89091fbf4a
Anaconda2-2.4.1-Linux-x86_64.sh	265.0 MiB	2015-12-08 15:00:49	2de682c96edf8cca2852071a84ff860025
Anaconda2-2.4.1-Linux-x86.sh	248.5 MiB	2015-12-08 15:00:50	2388cc714567afe7697bf43b4063ff0ea2
Anaconda2-2.4.0-Windows-x86_64.exe	388.0 MiB	2015-11-02 16:22:21	7a40484e58e91f62d91961c8607de586d3
Anaconda2-2.4.0-Windows-x86.exe	321.4 MiB	2015-11-02 16:22:21	2a05db81a0fe4155bc2dd83a689294d3ac
Anaconda2-2.4.0-MacOSX-x86_64.sh	239.5 MiB	2015-11-02 16:22:20	53c9123c9d508555100805fdb44d984551
Anaconda2-2.4.0-MacOSX-x86_64.pkg	274.3 MiB	2015-11-02 16:22:20	d6842135062f3c3d2f8bd33318133376d4
Anaconda2-2.4.0-Linux-x86_64.sh	288.4 MiB	2015-11-02 16:22:19	49d19834da06b1b82b6fa85bc647d2e78f

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda2-2.4.0-Linux-x86.sh	279.9 MiB	2015-11-02 16:22:19	`478a8fdde3a6e4040a68c57d7bdd6fab1a`
Anaconda-2.3.0-Windows-x86_64.exe	334.7 MiB	2015-07-01 13:35:10	`70b4a84e78c721bd46f3de39c75acb37d1`
Anaconda-2.3.0-Windows-x86.exe	277.4 MiB	2015-07-01 13:35:11	`3b60ddfb84533539e767889706bd64298a`
Anaconda-2.3.0-MacOSX-x86_64.sh	249.9 MiB	2015-07-01 13:35:10	`c4bb59a57bf44dde80612041bbbcfd2e5c`
Anaconda-2.3.0-MacOSX-x86_64.pkg	283.7 MiB	2015-07-01 13:35:09	`f920ae6211d9da3288b5e160100543667c`
Anaconda-2.3.0-Linux-x86_64.sh	323.9 MiB	2015-07-01 13:35:08	`7c02499e9511c127d225992cfe1cd815e8`
Anaconda-2.3.0-Linux-x86.sh	309.6 MiB	2015-07-01 13:35:09	`73fdbbb3e38207ed18e5059f71676d18d4`
Anaconda-2.2.0-Windows-x86_64.exe	331.2 MiB	2015-03-25 15:30:13	`1e01d7e1560668f4c05d1cfafcb59b79da`
Anaconda-2.2.0-Windows-x86.exe	274.2 MiB	2015-03-25 15:28:48	`247e8e7e386224a3df736ffe607596546f`
Anaconda-2.2.0-MacOSX-x86_64.sh	247.1 MiB	2015-03-25 15:17:33	`20570e2f3911e38a78d8f888f3ff445d6c`
Anaconda-2.2.0-MacOSX-x86_64.pkg	279.7 MiB	2015-03-25 15:27:27	`65784323db94b0c297e998bc81db5978e8`
Anaconda-2.2.0-Linux-x86_64.sh	317.3 MiB	2015-03-25 15:20:08	`ca2582cb2188073b0f348ad42207211a2b`
Anaconda-2.2.0-Linux-x86.sh	303.2 MiB	2015-03-25 15:19:54	`6437d5b08a19c3501f2f5dc3ae1ae16f91`
Anaconda-2.1.0-Windows-x86_64.exe	367.0 MiB	2014-09-25 11:07:11	`d9d7c8ed1c914312848407f08fff3d1935`
Anaconda-2.1.0-Windows-x86.exe	310.2 MiB	2014-09-25 11:05:03	`c39193c9018a9c1e9e8f3c1d2692ac6351`
Anaconda-2.1.0-MacOSX-x86_64.sh	241.0 MiB	2014-09-25 10:53:13	`128fd4f53e0895e0d23f33e924ae32e011`
Anaconda-2.1.0-MacOSX-x86_64.pkg	275.0 MiB	2014-09-25 11:33:13	`d8001bae990e7024b81e74c6b06d0f488d`
Anaconda-2.1.0-Linux-x86_64.sh	337.4 MiB	2014-09-25 10:50:15	`191fbf290747614929d0bdd576e330c944`
Anaconda-2.1.0-Linux-x86.sh	321.2 MiB	2014-09-25 10:50:30	`fd70c08719e6b5caae45b7c8402c6975a8`
Anaconda-2.0.1-Windows-x86_64.exe	343.7 MiB	2014-06-12 15:03:07	`5b27e7de356312da711a19ae6a4438c1c8`
Anaconda-2.0.1-Windows-x86.exe	287.3 MiB	2014-06-12 15:03:27	`be5a341bc3f9bf8386c686cfc9ad253f30`
Anaconda-2.0.1-MacOSX-x86_64.sh	214.4 MiB	2014-06-12 15:02:56	`4ecda163c6f46e70cc6a1fe62dece4c6ec`
Anaconda-2.0.1-MacOSX-x86_64.pkg	244.3 MiB	2014-06-12 15:02:50	`d6a0ce0422daa004929a4aef6b485d94f5`
Anaconda-2.0.1-Linux-x86_64.sh	327.9 MiB	2014-06-12 15:02:33	`074204fa26872b4a946123071d15b8390c`
Anaconda-2.0.1-Linux-x86.sh	309.1 MiB	2014-06-12 15:02:41	`e8ffc63f31673b5ce41a95796a1f729ddc`
Anaconda-2.0.0-Windows-x86_64.exe	334.4 MiB	2014-05-28 16:50:53	`60078f8677e62e435e5a53f1084e6f39df`
Anaconda-2.0.0-Windows-x86.exe	278.2 MiB	2014-05-28 16:51:02	`d86cc7100b4c04ec25768267b81798f70a`
Anaconda-2.0.0-MacOSX-x86_64.sh	206.1 MiB	2014-05-28 16:50:45	`ad6271ad21403166bf54d0734ba8c7f7eb`
Anaconda-2.0.0-MacOSX-x86_64.pkg	236.2 MiB	2014-05-28 16:50:41	`e2eb3805451a26235b2ed7f3e63535fc39`
Anaconda-2.0.0-Linux-x86_64.sh	316.9 MiB	2014-05-28 16:50:30	`3aa27ddf4a0ba5046ba52b97da99e20eb0`
Anaconda-2.0.0-Linux-x86.sh	298.4 MiB	2014-05-28 16:50:36	`efb9d3987134d484d88a9d915437b1bd56`
Anaconda-1.9.2-Windows-x86_64.exe	367.3 MiB	2014-04-08 17:41:16	`ef9cfb69c831210fc9000ee5482d2d98ba`
Anaconda-1.9.2-Windows-x86.exe	311.8 MiB	2014-04-08 17:39:25	`fe005aeacd1345b856c73d640856b79ed9`
Anaconda-1.9.2-MacOSX-x86_64.sh	245.4 MiB	2014-04-08 17:34:03	`be4611ca671f80b984fa330d4ecf82244c`
Anaconda-1.9.2-MacOSX-x86_64.pkg	281.0 MiB	2014-04-10 10:05:32	`2fff6dca12507f675b04ed1f303d0ee99d`
Anaconda-1.9.2-Linux-x86_64.sh	484.0 MiB	2014-04-08 17:32:38	`7181d399833a2549a9584255bb477487f2`
Anaconda-1.9.2-Linux-x86.sh	411.8 MiB	2014-04-08 17:33:09	`1f7c850d0b98c011a717b3b757d82077ac`
Anaconda-1.9.1-Windows-x86_64.exe	367.3 MiB	2014-02-20 15:10:34	`d0c3c2faca03b3820ff8fc39688f500bd1`
Anaconda-1.9.1-Windows-x86.exe	311.7 MiB	2014-02-20 15:08:42	`46cbe29a30cfcd56018f7f69a35525708f`
Anaconda-1.9.1-MacOSX-x86_64.sh	245.3 MiB	2014-02-20 13:02:05	`7e4358adbaae2db9e17d1e0e4263b9a017`
Anaconda-1.9.1-MacOSX-x86_64.pkg	280.9 MiB	2014-02-20 15:44:04	`2aa707b162e71d488495085fd13232f8c3`
Anaconda-1.9.1-Linux-x86_64.sh	483.9 MiB	2014-02-20 13:35:16	`f6455e06a72b8cc11c8a96fb88a85518a2`
Anaconda-1.9.1-Linux-x86.sh	411.8 MiB	2014-02-20 13:34:56	`9aa39c05f723fee18c54a9cc1729986193`
Anaconda-1.9.0-Windows-x86_64.exe	365.1 MiB	2014-02-10 10:24:18	`265c7e849688164f7a7fe9df541be01867`
Anaconda-1.9.0-Windows-x86.exe	308.6 MiB	2014-02-10 10:24:52	`2c8c58cf21e537e930535df5a0e8fd4b6d`
Anaconda-1.9.0-MacOSX-x86_64.sh	244.4 MiB	2014-02-10 10:23:47	`722fe4d4406e88c5023e7ee21dc1401bb2`
Anaconda-1.9.0-MacOSX-x86_64.pkg	279.8 MiB	2014-02-10 10:23:46	`b74134e7626f10fc4d86209a3ebbb19de3`
Anaconda-1.9.0-Linux-x86_64.sh	618.8 MiB	2014-02-10 10:23:05	`855f1265e4c0b40d50f5a3a0fe7bae05b1`
Anaconda-1.9.0-Linux-x86.sh	545.3 MiB	2014-02-10 10:23:30	`16471e90b3deb7be1b3d449d8883983d81`

Table 7 – continued from previous page

Name	Size	Time modified	SHA256 hash
Anaconda-1.8.0-Windows-x86_64.exe	342.1 MiB	2013-11-04 13:55:59	434c2b325a368958b66d52cee4cc710f5e
Anaconda-1.8.0-Windows-x86.exe	290.0 MiB	2013-11-04 13:54:14	719bc0987be80b46f9c6b745822777fa1f
Anaconda-1.8.0-MacOSX-x86_64.sh	228.8 MiB	2013-11-04 13:10:16	5844ca595b5930399a1213db64ab53e9b7
Anaconda-1.8.0-MacOSX-x86_64.pkg	263.0 MiB	2013-11-04 13:57:20	fb92afc7750bc58ac12f3cbd65c18ee0f8
Anaconda-1.8.0-Linux-x86_64.sh	465.7 MiB	2013-11-04 15:37:12	69f42966d918f4197040e4dd126d2e3cc3
Anaconda-1.8.0-Linux-x86.sh	393.0 MiB	2013-11-04 15:37:29	2c08a5cd6ccaa9dc84063b0ee9b007aa82
Anaconda-1.7.0-Windows-x86_64.exe	330.1 MiB	2013-09-08 17:12:00	59a3667fd33f8de1ed476d7ff07917d726
Anaconda-1.7.0-Windows-x86.exe	280.6 MiB	2013-09-08 17:10:23	b434776dfeac98f37328c6e538f5a1a531
Anaconda-1.7.0-MacOSX-x86_64.sh	223.3 MiB	2013-09-09 11:52:45	046b592245bc2c11e733acb9700dc50947
Anaconda-1.7.0-MacOSX-x86_64.pkg	256.7 MiB	2013-09-09 12:15:34	d277f7e162c77043e416d03a754389a0d9
Anaconda-1.7.0-Linux-x86_64.sh	452.6 MiB	2013-09-08 17:01:59	6115cfae55a0746b4ae4128be839c99db3
Anaconda-1.7.0-Linux-x86.sh	381.0 MiB	2013-09-08 17:02:26	af372a27a1887e11061485e2a854c53577
Anaconda-1.6.2-Windows-x86_64.exe	289.9 MiB	2013-07-09 13:46:28	641fc25c1d13e49cc030df5f4040170d12
Anaconda-1.6.2-Windows-x86.exe	244.4 MiB	2013-07-09 13:44:58	0873576bbd979e3b7859808bccc2311eda
Anaconda-1.6.1-Windows-x86_64.exe	289.9 MiB	2013-07-02 12:04:26	8cad320d4d6981644fbd1741bd5589d198
Anaconda-1.6.1-Windows-x86.exe	244.4 MiB	2013-07-02 12:02:59	a823dc7688cec49499bb5922783377c262
Anaconda-1.6.1-MacOSX-x86_64.sh	170.0 MiB	2013-07-02 11:59:25	bbc15de34208ce8af5aceedeea1334636f
Anaconda-1.6.1-MacOSX-x86_64.pkg	197.3 MiB	2013-07-02 17:30:12	7c79819dd40a14e52439664c3e88e89ecb
Anaconda-1.6.1-Linux-x86_64.sh	317.6 MiB	2013-07-02 11:57:42	81d1819ba08069343f228b9c819cdba0e4
Anaconda-1.6.1-Linux-x86.sh	247.1 MiB	2013-07-02 11:59:07	745b9452fd18720deefb465a6687c0d66d
Anaconda-1.6.0-Windows-x86_64.exe	290.4 MiB	2013-06-21 14:38:20	6e95dc3612ed430ded28bb48fa1671b32a
Anaconda-1.6.0-Windows-x86.exe	244.9 MiB	2013-06-21 14:36:46	3dc2588557455484b3b38feb14fa95d941
Anaconda-1.6.0-MacOSX-x86_64.sh	169.0 MiB	2013-06-21 14:26:14	e03317888c36c07451a349577b426f435a
Anaconda-1.6.0-Linux-x86_64.sh	309.5 MiB	2013-06-21 14:23:51	20f5b70193af4b0b8f10aa0e66aabca552
Anaconda-1.6.0-Linux-x86.sh	241.6 MiB	2013-06-21 14:23:39	d6aeedfcb39d648fdfb5bd72c4d0b3063a
Anaconda-1.5.1-MacOSX-x86_64.sh	166.2 MiB	2013-05-09 14:26:20	6d3c86a2fdbaeeec2a6c251d5c9034a32b
Anaconda-1.5.0-Windows-x86_64.exe	280.4 MiB	2013-05-08 09:20:08	7edbe2e51b71c69e7e7a1ec01d8d83954a
Anaconda-1.5.0-Windows-x86.exe	236.0 MiB	2013-05-08 09:18:44	d3dd92fa00d999a94638513daf2d4aeed1
Anaconda-1.5.0-MacOSX-x86_64.sh	166.2 MiB	2013-05-08 09:18:44	c69609f0f48f33ca5a12d425a9e4d0fc91
Anaconda-1.5.0-Linux-x86_64.sh	306.7 MiB	2013-05-08 09:18:36	f4cdc194f076e1b438c8a34e7e5f53e70c
Anaconda-1.5.0-Linux-x86.sh	238.8 MiB	2013-05-08 09:18:43	ca7e356dc1b8c8ef27dfb74b32c77563df
Anaconda-1.4.0-Windows-x86_64.exe	241.4 MiB	2013-03-09 16:57:09	6ff0a3bf82fdf5c6f0568d12ff030237ee
Anaconda-1.4.0-Windows-x86.exe	210.1 MiB	2013-03-09 16:55:45	e590e45d36d3f164fcaf58cda6a3cb0925
Anaconda-1.4.0-MacOSX-x86_64.sh	156.4 MiB	2013-03-09 16:46:57	e5d5dae6e93bb7df528abc19f5ed3a69cc
Anaconda-1.4.0-Linux-x86_64.sh	286.9 MiB	2013-03-09 16:46:38	85ae8a0a6e3a41cf7845be3def36ed4058
Anaconda-1.4.0-Linux-x86.sh	220.5 MiB	2013-03-09 16:46:53	065284c5de369c9b89dcae79e7169ce9b7

Anaconda with Python 2 on 64-bit Windows

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703456.9154801
time file was last modified, in human readable format	2018-05-30 13:04:16
exact file size, in bytes	591413392
file size, in human friendly format	564.0 MiB
md5	`595e427e4b625b6eab92623a28dc4e21`
sha256	`e5ff95332d08a7b006a5bb723e0a5124c4c4c9a9e4289afdd05941791a79e`

Hashes for Anaconda2-5.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707334.896789
time file was last modified, in human readable format	2018-02-15 09:08:54
exact file size, in bytes	548001744
file size, in human friendly format	522.6 MiB
md5	`b16d6d6858fc7decf671ac71e6d7cfdb`
sha256	`3674c8d8c233dbea30842f14dc76cc3feaf4badf7d9dfe4145aa5b6679fab`

Hashes for Anaconda2-5.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508813842.7920458
time file was last modified, in human readable format	2017-10-23 21:57:22
exact file size, in bytes	524040968
file size, in human friendly format	499.8 MiB
md5	`b8d9bc02edd61af3f7ece3d07e726e91`
sha256	`c43f94c51623850b0c1a826710fe9c8e50b0d73708874c9cf9b6ef03806ba`

Hashes for Anaconda2-5.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506454249.0
time file was last modified, in human readable format	2017-09-26 14:30:49
exact file size, in bytes	522426032
file size, in human friendly format	498.2 MiB
md5	`8323b1d5f0b1c3fdb5b85efbb099beb0`
sha256	`5fb73395cdf003613f5d44844da9870dbdc2a35cede0f928b02c38b5ee2eb`

Hashes for Anaconda2-4.4.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495839202.0
time file was last modified, in human readable format	2017-05-26 17:53:22
exact file size, in bytes	451651872
file size, in human friendly format	430.7 MiB
md5	`0f60aa52ef3a5d6170aeb6f7e3651f91`
sha256	`7a8ec1a36f385ebf28a1a8cf63b8b03ac0f7744e1531f5d359ce6a6d90391`

Hashes for Anaconda2-4.3.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838739.0
time file was last modified, in human readable format	2017-03-06 16:18:59
exact file size, in bytes	433804976
file size, in human friendly format	413.7 MiB
md5	`bfd41f4de09a690f8b3525d3fb79bd2f`
sha256	`c0e13a756a856d7b7757b10d65bee577d8c9826317050eecec42d2e48d2ea`

Hashes for Anaconda2-4.3.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1486066808.0
time file was last modified, in human readable format	2017-02-02 14:20:08
exact file size, in bytes	432863736
file size, in human friendly format	412.8 MiB
md5	`56b181af1959de40e67fb5ef50612ae2`
sha256	`2198e28e9e8e3c43ab72a8371e5b2d0a9aa6574391aebbcb94bf768a50a57`

Hashes for Anaconda2-4.3.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548279.0
time file was last modified, in human readable format	2017-01-27 14:17:59
exact file size, in bytes	432864904
file size, in human friendly format	412.8 MiB
md5	`2c02e21e542d61760c3e19bf0b3086fe`
sha256	`1117839746a8eabf7ed26ff311fc74e44a58e319555e306f241e04b32363a`

Hashes for Anaconda2-4.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009747.0
time file was last modified, in human readable format	2016-09-27 15:55:47
exact file size, in bytes	399546128
file size, in human friendly format	381.0 MiB
md5	`0a30d509568724dac0ae193e139b9c37`
sha256	`6254b150edee53000c94e9abfc9c51a2d2e5ef3453d8e7cc7ef0a848d6d3b`

Hashes for Anaconda2-4.1.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994801.0
time file was last modified, in human readable format	2016-07-08 11:20:01
exact file size, in bytes	357765440
file size, in human friendly format	341.2 MiB
md5	`1db0244dbf02579f452d1b19ce245144`
sha256	`7be13a69df254b86e47612c726b0b2ba9ffa35c060b4d28edb348004c1f72`

Hashes for Anaconda2-4.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131310.0
time file was last modified, in human readable format	2016-06-28 11:28:30
exact file size, in bytes	356677104
file size, in human friendly format	340.2 MiB
md5	`6c1066a240b28dbb33e9293a97cd40f5`
sha256	`7a62880ff9bb7f747d70f518f024dfd1795a26d4130a20d1ff30043d05ec7`

Hashes for Anaconda2-4.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268099.0
time file was last modified, in human readable format	2016-03-29 11:14:59
exact file size, in bytes	350807856
file size, in human friendly format	334.6 MiB
md5	`6b2ad997c42fbf58bb1b54baa5619e4f`
sha256	`213c7d94bdb6f0931edd31bb14ae33ab557cee52c4ac949300e512397a29e`

Hashes for Anaconda2-2.5.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535974.0
time file was last modified, in human readable format	2016-02-03 15:46:14
exact file size, in bytes	365581384
file size, in human friendly format	348.6 MiB
md5	`57e42190411054333781c1208822659d`
sha256	`4423b43eb23184b4239abc426a564760d6ddf0187ce451468546b88931de4`

Hashes for Anaconda2-2.4.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608451.0
time file was last modified, in human readable format	2015-12-08 15:00:51
exact file size, in bytes	371393960
file size, in human friendly format	354.2 MiB
md5	`733ce916c4c392367c611efd493410b0`
sha256	`cfbe5539cb7f2e5807ec3d2fa2e59db3a419caa1ef8f0497516dd0c861f92`

Hashes for Anaconda2-2.4.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502941.0
time file was last modified, in human readable format	2015-11-02 16:22:21
exact file size, in bytes	406819096
file size, in human friendly format	388.0 MiB
md5	`00a09d300d13c9f4754165920396625d`
sha256	`7a40484e58e91f62d91961c8607de586d3ef14645319c0395683e5f718255`

Hashes for Anaconda-2.3.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775710.0
time file was last modified, in human readable format	2015-07-01 13:35:10
exact file size, in bytes	350951272
file size, in human friendly format	334.7 MiB
md5	`93d3d5d2aae82c175cd9ef4a570c2ab0`
sha256	`70b4a84e78c721bd46f3de39c75acb37d1980a3afa23cf3cef387569606f7`

Hashes for Anaconda-2.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315413.0
time file was last modified, in human readable format	2015-03-25 15:30:13
exact file size, in bytes	347294944
file size, in human friendly format	331.2 MiB
md5	`27230171e315bcdee370ef97ef622158`
sha256	`1e01d7e1560668f4c05d1cfafcb59b79da1b352671dc913a5ec8b766dde12`

Hashes for Anaconda-2.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411661231.0
time file was last modified, in human readable format	2014-09-25 11:07:11
exact file size, in bytes	384818768
file size, in human friendly format	367.0 MiB
md5	a80eea69583fcee6d3d0f6a63a900b2e
sha256	d9d7c8ed1c914312848407f08fff3d19350c20d754c8872d36ef45ce7541c

Hashes for Anaconda-2.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603387.0
time file was last modified, in human readable format	2014-06-12 15:03:07
exact file size, in bytes	360443904
file size, in human friendly format	343.7 MiB
md5	b498d9bf6b266bc09507d2ef9d4b7b55
sha256	5b27e7de356312da711a19ae6a4438c1c857b9c4e357c4aa3275d014db73c

Hashes for Anaconda-2.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313853.0
time file was last modified, in human readable format	2014-05-28 16:50:53
exact file size, in bytes	350647728
file size, in human friendly format	334.4 MiB
md5	9ec65c4cc0d640ff36f89193cb9e7b7d
sha256	60078f8677e62e435e5a53f1084e6f39df7f4874892b77d04fbd819033c44

Hashes for Anaconda-1.9.2-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1396996876.0
time file was last modified, in human readable format	2014-04-08 17:41:16
exact file size, in bytes	385184016
file size, in human friendly format	367.3 MiB
md5	`a610322f6752413c9b02abf72f960ff9`
sha256	`ef9cfb69c831210fc9000ee5482d2d98ba609d4fbe9e05746f93f569045ba`

Hashes for Anaconda-1.9.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392930634.0
time file was last modified, in human readable format	2014-02-20 15:10:34
exact file size, in bytes	385134224
file size, in human friendly format	367.3 MiB
md5	`b8a404c9f5bfd2452316db3710d2b8ef`
sha256	`d0c3c2faca03b3820ff8fc39688f500bd140f207aab7553c50005484ff755`

Hashes for Anaconda-1.9.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049458.0
time file was last modified, in human readable format	2014-02-10 10:24:18
exact file size, in bytes	382872112
file size, in human friendly format	365.1 MiB
md5	`3c5e322e71428167e4d38725e1d92be0`
sha256	`265c7e849688164f7a7fe9df541be018675772f2e91be39d116e6d0e07181`

Hashes for Anaconda-1.8.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383594959.0
time file was last modified, in human readable format	2013-11-04 13:55:59
exact file size, in bytes	358748424
file size, in human friendly format	342.1 MiB
md5	`dccc94b5e1b77e56385a318c5c91b6d1`
sha256	`434c2b325a368958b66d52cee4cc710f5ea40e45657854a9be2b54dd50b9c`

Hashes for Anaconda-1.7.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378678320.0
time file was last modified, in human readable format	2013-09-08 17:12:00
exact file size, in bytes	346131087
file size, in human friendly format	330.1 MiB
md5	`c4e6987a83b00da8d36fc4e559df7d01`
sha256	`59a3667fd33f8de1ed476d7ff07917d726be51de239deaf7ce13ab277bb41`

Hashes for Anaconda-1.6.2-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1373395588.0
time file was last modified, in human readable format	2013-07-09 13:46:28
exact file size, in bytes	303973708
file size, in human friendly format	289.9 MiB
md5	`80bc3fe5f8d2f83110eee775946ed3b8`
sha256	`641fc25c1d13e49cc030df5f4040170d123072e54b439e7097531a61cc385`

Hashes for Anaconda-1.6.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372784666.0
time file was last modified, in human readable format	2013-07-02 12:04:26
exact file size, in bytes	303973712
file size, in human friendly format	289.9 MiB
md5	`3e63a96cc45f665bf53fa38b18491f94`
sha256	`8cad320d4d6981644fbd1741bd5589d198f5e4ca1e1f66a10d57c704ee485`

Hashes for Anaconda-1.6.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1371843500.0
time file was last modified, in human readable format	2013-06-21 14:38:20
exact file size, in bytes	304462009
file size, in human friendly format	290.4 MiB
md5	d215a5aca9515f1875cf131b0c35d78d
sha256	6e95dc3612ed430ded28bb48fa1671b32a185c976eba905796707f9b5b44e

Hashes for Anaconda-1.5.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368022808.0
time file was last modified, in human readable format	2013-05-08 09:20:08
exact file size, in bytes	294062717
file size, in human friendly format	280.4 MiB
md5	058a62bb0fbaf53870b92798453e718a
sha256	7edbe2e51b71c69e7e7a1ec01d8d83954ada9e885e08adfffdd624b9e1b10

Hashes for Anaconda-1.4.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1362869829.0
time file was last modified, in human readable format	2013-03-09 16:57:09
exact file size, in bytes	253175221
file size, in human friendly format	241.4 MiB
md5	7e4ff5278e86cc88852abb5da453ae7a
sha256	6ff0a3bf82fdf5c6f0568d12ff030237ee90825bb0ea60e4cf3833db84753

Anaconda with Python 3 on 64-bit Windows

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703458.8495667
time file was last modified, in human readable format	2018-05-30 13:04:18
exact file size, in bytes	661987080
file size, in human friendly format	631.3 MiB
md5	`62244c0382b8142743622fdc3526eda7`
sha256	`2672f6537e2c8a79ae9540cf3c49b18bb9ba35caec649191b5fa1e759f15b`

Hashes for Anaconda3-5.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707426.6260371
time file was last modified, in human readable format	2018-02-15 09:10:26
exact file size, in bytes	563168960
file size, in human friendly format	537.1 MiB
md5	`83a8b1edcb21fa0ac481b23f65b604c6`
sha256	`7d192e58915d7e7fbfd0c987ddc4db38a22d5fc47a22db71ac5873ef9ba8a`

Hashes for Anaconda3-5.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508866679.6710148
time file was last modified, in human readable format	2017-10-24 12:37:59
exact file size, in bytes	539829832
file size, in human friendly format	514.8 MiB
md5	`3dde7dbbef158db6dc44fce495671c92`
sha256	`0b1ec18b7425f8c8518d6dc2fc0bc8ec2f06ba57f15727aee4731a4f98278`

Hashes for Anaconda3-5.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506453293.0
time file was last modified, in human readable format	2017-09-26 14:14:53
exact file size, in bytes	534742736
file size, in human friendly format	510.0 MiB
md5	`fee3fad608d0006afa5c7bca4de3d02b`
sha256	`53bd80727099b5767b9f20f99e908f9c19cea7572c14f9538dc1c8ca7ab5e`

Hashes for Anaconda3-4.4.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495839334.0
time file was last modified, in human readable format	2017-05-26 17:55:34
exact file size, in bytes	458893576
file size, in human friendly format	437.6 MiB
md5	`aa200a1c059a551e0ba9a5314a9554a5`
sha256	`ea582602541e748053df550514460426202fb4507edf9af4d7d706bc41044`

Hashes for Anaconda3-4.3.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838848.0
time file was last modified, in human readable format	2017-03-06 16:20:48
exact file size, in bytes	442630816
file size, in human friendly format	422.1 MiB
md5	`16f337426454eac463fd0d41c6d2bbb8`
sha256	`65ce6d7c09884935fee9eb8d318b30e95f75f6efe8a8ba221df282cf22c39`

Hashes for Anaconda3-4.3.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1486066930.0
time file was last modified, in human readable format	2017-02-02 14:22:10
exact file size, in bytes	441680784
file size, in human friendly format	421.2 MiB
md5	`07ea8c5a2306ac8fabf3902bd6623787`
sha256	`b5954bf7da9a92d351d905dfdfa0e7bee1cfdf8c74ed0532a29416849a92b`

Hashes for Anaconda3-4.3.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548381.0
time file was last modified, in human readable format	2017-01-27 14:19:41
exact file size, in bytes	441681320
file size, in human friendly format	421.2 MiB
md5	`137043b3f9860519967759fc8ea76514`
sha256	`324568dbef777a6ac8a25c1e8ae1975ffbd95bb621dc91cb3869606cc5924`

Hashes for Anaconda3-4.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009841.0
time file was last modified, in human readable format	2016-09-27 15:57:21
exact file size, in bytes	410431504
file size, in human friendly format	391.4 MiB
md5	`0ca5ef4dcfe84376aad073bbb3f8db00`
sha256	`84e30c99833e142a27fc9ee2c748b03f16c8b1a3ced765024d5db4e68bfbf`

Hashes for Anaconda3-4.1.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994877.0
time file was last modified, in human readable format	2016-07-08 11:21:17
exact file size, in bytes	370055720
file size, in human friendly format	352.9 MiB
md5	a3be394f8274c391148efdfbc63e8ca4
sha256	b4889513dc574f9d6f96db089315d69d293f8b17635da4d2e6eee118dc105

Hashes for Anaconda3-4.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131313.0
time file was last modified, in human readable format	2016-06-28 11:28:33
exact file size, in bytes	368509992
file size, in human friendly format	351.4 MiB
md5	50fe73c084b91e55837f4d090809a35e
sha256	9acde60b591233452dba23ac15800f39f2de9b7a180a89916dfa3b6edb326

Hashes for Anaconda3-4.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268182.0
time file was last modified, in human readable format	2016-03-29 11:16:22
exact file size, in bytes	362171448
file size, in human friendly format	345.4 MiB
md5	a6b7a787c6c574867cee3f2d12ecfc50
sha256	39bf467cd142c1f8fdb7d673fdea273d87ec011af1dbf4b4804c2b0994c61

Hashes for Anaconda3-2.5.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454536060.0
time file was last modified, in human readable format	2016-02-03 15:47:40
exact file size, in bytes	378634984
file size, in human friendly format	361.1 MiB
md5	6572ceba288b6f145e9b3d0c02a5281c
sha256	4728044d77da715e48d4c95d7f2e3c2a02c4ea7de9d2c69acc851bf294500

Hashes for Anaconda3-2.4.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608455.0
time file was last modified, in human readable format	2015-12-08 15:00:55
exact file size, in bytes	381329960
file size, in human friendly format	363.7 MiB
md5	17c562ff74676f004ba8dd029718c09c
sha256	21d155a4b43805042499b8d008835bcdfd9c3a45fe53d1183de9e0a937170

Hashes for Anaconda3-2.4.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502944.0
time file was last modified, in human readable format	2015-11-02 16:22:24
exact file size, in bytes	411312288
file size, in human friendly format	392.3 MiB
md5	bc74a4fb4e8455e8e7c61b7f100e3bac
sha256	beaa1b803dd30022c6aca1c6f05182beaea3cd8a17130f73906851269dfe0

Hashes for Anaconda3-2.3.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775714.0
time file was last modified, in human readable format	2015-07-01 13:35:14
exact file size, in bytes	352774600
file size, in human friendly format	336.4 MiB
md5	ad4abc78581f6fa68b7f7fc342003f6c
sha256	bd693b61cf191666ae0473327f3c15bcf32b7d09961a0aa0284c10e7ea724

Hashes for Anaconda3-2.2.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315550.0
time file was last modified, in human readable format	2015-03-25 15:32:30
exact file size, in bytes	348764152
file size, in human friendly format	332.6 MiB
md5	cd7dae4fd482c94156b4d60bf21d8771
sha256	28c5a13b27a9dbd57c7c633316c5f4beb0cd32cf19b148debd1a81eac86f3

Hashes for Anaconda3-2.1.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411661463.0
time file was last modified, in human readable format	2014-09-25 11:11:03
exact file size, in bytes	380970064
file size, in human friendly format	363.3 MiB
md5	`5d559802f3c699a885c66ea6064f5440`
sha256	`ea4059469b1820069f62bd6c256def6259d801d7382be70523e081c5adbed`

Hashes for Anaconda3-2.0.1-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603215.0
time file was last modified, in human readable format	2014-06-12 15:00:15
exact file size, in bytes	335317304
file size, in human friendly format	319.8 MiB
md5	`139c6d32c484e0886c6cbe530b9fbd4c`
sha256	`e2b6d3d6a9e378fc0d0dd63342417c02bdf6a9676040e05ff8017396d6449`

Hashes for Anaconda3-2.0.0-Windows-x86_64.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401228102.0
time file was last modified, in human readable format	2014-05-27 17:01:42
exact file size, in bytes	325992440
file size, in human friendly format	310.9 MiB
md5	`35fb9536ccb1aca93ec34714e8e69a5b`
sha256	`a8046fc82da7463ef53cdeaba97c72433c37b211c50fa87f1bc19bdfe5163`

Anaconda with Python 2 on 32-bit Windows

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703457.7225163
time file was last modified, in human readable format	2018-05-30 13:04:17
exact file size, in bytes	464889960
file size, in human friendly format	443.4 MiB
md5	`4a3729b14c2d3fccd3a050821679c702`
sha256	`2b81916c477e64db917821bb48a97000fad78cd1041022b343cec3cebf9e9`

Hashes for Anaconda2-5.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707335.7558289
time file was last modified, in human readable format	2018-02-15 09:08:55
exact file size, in bytes	440226936
file size, in human friendly format	419.8 MiB
md5	`a09347a53e04a15ee965300c2b95dfde`
sha256	`fa78c71d88b01e6367f0c3cbd23da1f82e86e02088b0d281437789bfeba59`

Hashes for Anaconda2-5.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508864894.4238315
time file was last modified, in human readable format	2017-10-24 12:08:14
exact file size, in bytes	422964800
file size, in human friendly format	403.4 MiB
md5	`623e8d9ca2270cb9823a897dd0e9bfce`
sha256	`1a50fac8644f2128e318337b218299e53e92ee20ddaf47911ff2be22255c6`

Hashes for Anaconda2-5.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461109.7266462
time file was last modified, in human readable format	2017-09-26 16:25:09
exact file size, in bytes	421720568
file size, in human friendly format	402.2 MiB
md5	`bd3ed48229db828cef4c6b371a8759d1`
sha256	`078551cfb0df72779897026724f375671e12a5fd384cabeaede5cc325cac1`

Hashes for Anaconda2-4.4.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495839135.0
time file was last modified, in human readable format	2017-05-26 17:52:15
exact file size, in bytes	371653096
file size, in human friendly format	354.4 MiB
md5	51f14d30b08b82cd5e44bbb6b0d63349
sha256	0decd861f8839fdf2cbe4fa306c127f69e50b54374e56d7960ba5217087bf

Hashes for Anaconda2-4.3.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838692.0
time file was last modified, in human readable format	2017-03-06 16:18:12
exact file size, in bytes	355485664
file size, in human friendly format	339.0 MiB
md5	4f5ed9917f8c2d2ae2e027e45a85fe8b
sha256	fc363cea3c321c17b43a0bf2137aa845fef349c534fcf511dc285ebb8ae57

Hashes for Anaconda2-4.3.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1486066745.0
time file was last modified, in human readable format	2017-02-02 14:19:05
exact file size, in bytes	354548480
file size, in human friendly format	338.1 MiB
md5	4bff7044ecf0229a0974ba8429520cad
sha256	863702665aa2b55ede2103a8ca7d26435efef614e9d201909c21ec572878f

Hashes for Anaconda2-4.3.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548226.0
time file was last modified, in human readable format	2017-01-27 14:17:06
exact file size, in bytes	354550816
file size, in human friendly format	338.1 MiB
md5	ffd6296dc4b359684c54ce6f3d10e144
sha256	a98767acefdeda02fe8d3ef9dadda1a3439fec110ede9bf5d0e359be76ac1

Hashes for Anaconda2-4.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009690.0
time file was last modified, in human readable format	2016-09-27 15:54:50
exact file size, in bytes	339829096
file size, in human friendly format	324.1 MiB
md5	f4f12af8811759e56464eef5a484963d
sha256	a97ca79cb771568d051ef7773d25c0dda407c63e7ec91305f35dff790aeda

Hashes for Anaconda2-4.1.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994801.0
time file was last modified, in human readable format	2016-07-08 11:20:01
exact file size, in bytes	299852168
file size, in human friendly format	286.0 MiB
md5	`b319d6867c67723ba74aef4f9dd35f82`
sha256	`4708d73952a0a8040bf1594ea42027a30e9bacb4d6760cc5d3e4414b6bfd9`

Hashes for Anaconda2-4.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131311.0
time file was last modified, in human readable format	2016-06-28 11:28:31
exact file size, in bytes	298958864
file size, in human friendly format	285.1 MiB
md5	`ec44ea5c92542ca0112a6be79aff79bf`
sha256	`c4ad4eefdffb6d838424c62c8b524352d8b8e4752382b0a09e9d8b7e9e44b`

Hashes for Anaconda2-4.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268100.0
time file was last modified, in human readable format	2016-03-29 11:15:00
exact file size, in bytes	294659856
file size, in human friendly format	281.0 MiB
md5	`9fb16479d7eb3dd63cf4ad6704622c8a`
sha256	`f8185ad2fe89356ab001e55a463b663bcbc9e7699ab7f7c1775a98d5332db`

Hashes for Anaconda2-2.5.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535921.0
time file was last modified, in human readable format	2016-02-03 15:45:21
exact file size, in bytes	310590880
file size, in human friendly format	296.2 MiB
md5	`506c08af8932332303561e822f285d9b`
sha256	`a0336729f0400ff12fe18a7d5e20c3f9b45f14cd07fe65402 9daa1ec611e2`

Hashes for Anaconda2-2.4.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608452.0
time file was last modified, in human readable format	2015-12-08 15:00:52
exact file size, in bytes	301790720
file size, in human friendly format	287.8 MiB
md5	`0e6cdba39322c28240f4dceaf7bf72f8`
sha256	`65fb15559b0ddb5055c110ecdb84823a6fbc8d3938932fbfa0110bb6d3395`

Hashes for Anaconda2-2.4.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502941.0
time file was last modified, in human readable format	2015-11-02 16:22:21
exact file size, in bytes	337056800
file size, in human friendly format	321.4 MiB
md5	`5a5225bd2f74a5be9ef840ae8e62c82a`
sha256	`2a05db81a0fe4155bc2dd83a689294d3ac7fa1d1a68a5ec6bdafaac9140d4`

Hashes for Anaconda-2.3.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775711.0
time file was last modified, in human readable format	2015-07-01 13:35:11
exact file size, in bytes	290903240
file size, in human friendly format	277.4 MiB
md5	`7efff6446dcb06e4c44607539c953689`
sha256	`3b60ddfb84533539e767889706bd64298a73d07a7bfe944a3c5c7f951e395`

Hashes for Anaconda-2.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315328.0
time file was last modified, in human readable format	2015-03-25 15:28:48
exact file size, in bytes	287557144
file size, in human friendly format	274.2 MiB
md5	32246b48658d4c3faeef425cec64a131
sha256	247e8e7e386224a3df736ffe607596546f4bdd64b44a945fd831db5603782

Hashes for Anaconda-2.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411661103.0
time file was last modified, in human readable format	2014-09-25 11:05:03
exact file size, in bytes	325285048
file size, in human friendly format	310.2 MiB
md5	4b4303ff83c94f6af128fe43c202756b
sha256	c39193c9018a9c1e9e8f3c1d2692ac635133e9b68e72d7864c75841086337

Hashes for Anaconda-2.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603407.0
time file was last modified, in human readable format	2014-06-12 15:03:27
exact file size, in bytes	301248280
file size, in human friendly format	287.3 MiB
md5	579ed15c9599cc49bc073dbbe8870021
sha256	be5a341bc3f9bf8386c686cfc9ad253f3074670c96f130b345b731370ce89

Hashes for Anaconda-2.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313862.0
time file was last modified, in human readable format	2014-05-28 16:51:02
exact file size, in bytes	291661544
file size, in human friendly format	278.2 MiB
md5	`c3b147e0d5f6d708e884ee03d8856040`
sha256	`d86cc7100b4c04ec25768267b81798f70a8563a2bdb6dacfe6ec7e45bfbbb`

Hashes for Anaconda-1.9.2-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1396996765.0
time file was last modified, in human readable format	2014-04-08 17:39:25
exact file size, in bytes	326939720
file size, in human friendly format	311.8 MiB
md5	`7217cfa5c5b45de3e683ff09a10ce35b`
sha256	`fe005aeacd1345b856c73d640856b79ed94a6694245ea8df8cbf94aa7bac7`

Hashes for Anaconda-1.9.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392930522.0
time file was last modified, in human readable format	2014-02-20 15:08:42
exact file size, in bytes	326889840
file size, in human friendly format	311.7 MiB
md5	`5404da4f89dca1a4f5c9efd5ae6fbc5a`
sha256	`46cbe29a30cfcd56018f7f69a35525708f2d14211a613e7344f91ad4171a8`

Hashes for Anaconda-1.9.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049492.0
time file was last modified, in human readable format	2014-02-10 10:24:52
exact file size, in bytes	323587016
file size, in human friendly format	308.6 MiB
md5	`02496bcd853574628adfbe1defe5c40f`
sha256	`2c8c58cf21e537e930535df5a0e8fd4b6d60d4dbe87f13b5964bf2f5f4d27`

Hashes for Anaconda-1.8.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383594854.0
time file was last modified, in human readable format	2013-11-04 13:54:14
exact file size, in bytes	304130696
file size, in human friendly format	290.0 MiB
md5	`3b3bbc639556499d62342f653443553a`
sha256	`719bc0987be80b46f9c6b745822777fa1f0cb7386ff746fa8e71763bfd997`

Hashes for Anaconda-1.7.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378678223.0
time file was last modified, in human readable format	2013-09-08 17:10:23
exact file size, in bytes	294250542
file size, in human friendly format	280.6 MiB
md5	91a6398f63a8cc6fa3db3a1e9195b3bf
sha256	b434776dfeac98f37328c6e538f5a1a53199e0c6ca2ef3a39cb3cd2e64db2

Hashes for Anaconda-1.6.2-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1373395498.0
time file was last modified, in human readable format	2013-07-09 13:44:58
exact file size, in bytes	256262643
file size, in human friendly format	244.4 MiB
md5	5d9ca457b14fe9af5f8f5e338f9db9e2
sha256	0873576bbd979e3b7859808bccc2311edaea3d34f4d6ed1f6b44b1ba1cf1a

Hashes for Anaconda-1.6.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372784579.0
time file was last modified, in human readable format	2013-07-02 12:02:59
exact file size, in bytes	256262655
file size, in human friendly format	244.4 MiB
md5	`3cdf41952ad09f00ab03cca5a289fe50`
sha256	`a823dc7688cec49499bb5922783377c262cbf456830ff8db7c0d4018a2321`

Hashes for Anaconda-1.6.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1371843406.0
time file was last modified, in human readable format	2013-06-21 14:36:46
exact file size, in bytes	256780921
file size, in human friendly format	244.9 MiB
md5	`156a48269ae6b2bfc0bede9c3ff719cc`
sha256	`3dc2588557455484b3b38feb14fa95d941de732e06678365860cd4961c19f`

Hashes for Anaconda-1.5.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368022724.0
time file was last modified, in human readable format	2013-05-08 09:18:44
exact file size, in bytes	247436755
file size, in human friendly format	236.0 MiB
md5	`871f9f4f2321cede8d25ff83f24e70da`
sha256	`d3dd92fa00d999a94638513daf2d4aeed15a387b820eb08b1907387a4f2e8`

Hashes for Anaconda-1.4.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1362869745.0
time file was last modified, in human readable format	2013-03-09 16:55:45
exact file size, in bytes	220256092
file size, in human friendly format	210.1 MiB
md5	`797f4a28462db075de4d21e7977f32a5`
sha256	`e590e45d36d3f164fcaf58cda6a3cb09252a502af5942e0909324b394710f`

Anaconda with Python 3 on 32-bit Windows

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703459.8546116
time file was last modified, in human readable format	2018-05-30 13:04:19
exact file size, in bytes	530914888
file size, in human friendly format	506.3 MiB
md5	`285387e7b6ea81edba98c011922e235a`
sha256	`64305a4c0041aaf4a3fd0fee4466d7b7f238fddd9e44a4c8c10f5fa059e82`

Hashes for Anaconda3-5.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707428.264113
time file was last modified, in human readable format	2018-02-15 09:10:28
exact file size, in bytes	456610264
file size, in human friendly format	435.5 MiB
md5	`7a2291ab99178a4cdec530861494531f`
sha256	`7a05da21fd592991d181ac8467faac51345fb64acca6553609b53c9825e5e`

Hashes for Anaconda3-5.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508866630.5278003
time file was last modified, in human readable format	2017-10-24 12:37:10
exact file size, in bytes	440867256
file size, in human friendly format	420.4 MiB
md5	`9d2ffb0aac1f8a72ef4a5c535f3891f2`
sha256	`9edc3012324c9c8c9aa5257688bd793277ee25bc99cffc8da6610b5f45585`

Hashes for Anaconda3-5.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461112.5017715
time file was last modified, in human readable format	2017-09-26 16:25:12
exact file size, in bytes	436033392
file size, in human friendly format	415.8 MiB
md5	`4a48ded89f15b4a2e39ffa69f3532df2`
sha256	`a0d5d8e328b1d3a1ed921cadeecda659c49b6042980558f5c5f491e884bf8`

Hashes for Anaconda3-4.4.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495839261.0
time file was last modified, in human readable format	2017-05-26 17:54:21
exact file size, in bytes	379794600
file size, in human friendly format	362.2 MiB
md5	`c7a66350b79354773dabbbef6f58a3af`
sha256	`37afe00b8305cc09b7bd8dd07f65cec3f4e1534966c275eb55df7c91fb660`

Hashes for Anaconda3-4.3.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838786.0
time file was last modified, in human readable format	2017-03-06 16:19:46
exact file size, in bytes	365005040
file size, in human friendly format	348.1 MiB
md5	`27fa7486dfe82cf31642eb355b9a184f`
sha256	`adf322f49542cf509d4f72152cea24e54edfb4ff279ba3ab19582a5f27461`

Hashes for Anaconda3-4.3.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1486066862.0
time file was last modified, in human readable format	2017-02-02 14:21:02
exact file size, in bytes	364057456
file size, in human friendly format	347.2 MiB
md5	`5dd0a8b09a5eb6c9d002dc26d6f31492`
sha256	`7f8ffce6b2c3a968ce19171c9dc332dec61741113f7cac4b52953596f9e20`

Hashes for Anaconda3-4.3.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548325.0
time file was last modified, in human readable format	2017-01-27 14:18:45
exact file size, in bytes	364059456
file size, in human friendly format	347.2 MiB
md5	`ae7ec9752cf81c01983fcf0ddf8d7cc2`
sha256	`4a5dfea30b926074b4d6e0f1cea3e9765799fd33532b4347fa0d3d9aaacfe`

Hashes for Anaconda3-4.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009790.0
time file was last modified, in human readable format	2016-09-27 15:56:30
exact file size, in bytes	349560232
file size, in human friendly format	333.4 MiB
md5	`96e5fe052b22d667da9360fb4edce363`
sha256	`e7b79a9886da3f840b52882c47ecab3eda0c97505019c1f8f0c8b7eb15c2d`

Hashes for Anaconda3-4.1.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994878.0
time file was last modified, in human readable format	2016-07-08 11:21:18
exact file size, in bytes	308116424
file size, in human friendly format	293.8 MiB
md5	`39bd047c2169a9d072e98403f487c9e8`
sha256	`224e3dd90850651ae0d1c9216b4c317d1d553d8c118a83c9bc7e315daf85f`

Hashes for Anaconda3-4.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131314.0
time file was last modified, in human readable format	2016-06-28 11:28:34
exact file size, in bytes	306794104
file size, in human friendly format	292.6 MiB
md5	`2f96e23dd2e5f04f9a5059b8ef5d7fd2`
sha256	`4f444ed9400505e822bb475e986800fac058ef6f23298c2b00d285d05df0a`

Hashes for Anaconda3-4.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268182.0
time file was last modified, in human readable format	2016-03-29 11:16:22
exact file size, in bytes	296840248
file size, in human friendly format	283.1 MiB
md5	ae5c9ba0c6f4639fbf94848f81c3d4b4
sha256	b5a31a9d130a40c3110c0592a6c8fbd43a51522e32fdddf20afe15595db80

Hashes for Anaconda3-2.5.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454536013.0
time file was last modified, in human readable format	2016-02-03 15:46:53
exact file size, in bytes	310656088
file size, in human friendly format	296.3 MiB
md5	f7ce22122319026697cc8e7dda80300b
sha256	4a3441aaaa269d06f39e1430155f9f25a8a24122cf48b9fc6bdccb0e96a82

Hashes for Anaconda3-2.4.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608456.0
time file was last modified, in human readable format	2015-12-08 15:00:56
exact file size, in bytes	313632120
file size, in human friendly format	299.1 MiB
md5	78eef53e753cf9a72babe06c374db8ed
sha256	dfe50d13473547b5230f6194dfe6bdff961a8b7f5e3c22edef8c7788194b7

Hashes for Anaconda3-2.4.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502945.0
time file was last modified, in human readable format	2015-11-02 16:22:25
exact file size, in bytes	331748568
file size, in human friendly format	316.4 MiB
md5	0d3b78e2a4747d5975097c47129c0e70
sha256	a69a9fe00ce337b0cfd7d024b79ba5141cd04b1d51982196658df26c0d370

Hashes for Anaconda3-2.3.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775715.0
time file was last modified, in human readable format	2015-07-01 13:35:15
exact file size, in bytes	294752872
file size, in human friendly format	281.1 MiB
md5	8edec318e2957a934bd99a6062ddebd9
sha256	02d5f84da308f96d1a252a6669f3ca91e125c011d1b89ae33f05f6ebe4903

Hashes for Anaconda3-2.2.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315479.0
time file was last modified, in human readable format	2015-03-25 15:31:19
exact file size, in bytes	291166920
file size, in human friendly format	277.7 MiB
md5	`7c49a4e76e1c383038c4a1e8c4ac506f`
sha256	`20c46fff048fb313aaf1a49171c1a7b96a42f5be09e1e1e7052800dcec7ac`

Hashes for Anaconda3-2.1.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411661341.0
time file was last modified, in human readable format	2014-09-25 11:09:01
exact file size, in bytes	323724400
file size, in human friendly format	308.7 MiB
md5	`a2392f068d550bee23673529734ef5d4`
sha256	`8ffa252aa2b4f63889888ae85a81626ce952a1f9ac20d4c065070514acfac`

Hashes for Anaconda3-2.0.1-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603227.0
time file was last modified, in human readable format	2014-06-12 15:00:27
exact file size, in bytes	278631456
file size, in human friendly format	265.7 MiB
md5	`cbf8ff3b86731df7225bd2f7fb2af7f6`
sha256	`b08803296d7439413d590fd1f967b20127916c8d766802a27badc15a3a81b`

Hashes for Anaconda3-2.0.0-Windows-x86.exe

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401227953.0
time file was last modified, in human readable format	2014-05-27 16:59:13
exact file size, in bytes	269399872
file size, in human friendly format	256.9 MiB
md5	`1fd12fc8c5c3defcdf3a0bee6f5129fa`
sha256	`37986ce4c104ed3c82838de74b3a4de17918cc2f408235b9de9d4283d3a55`

Graphical installers for Anaconda with Python 2 on macOS

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703532.6788743
time file was last modified, in human readable format	2018-05-30 13:05:32
exact file size, in bytes	646731039
file size, in human friendly format	616.8 MiB
md5	`2836c839d29be8d9569a715f4c631a3b`
sha256	`f7695a3571eb8e8ae71fe9f413c36f57c92bc8882174c0dbf778e17550ff3`

Hashes for Anaconda2-5.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707332.6606853
time file was last modified, in human readable format	2018-02-15 09:08:52
exact file size, in bytes	616553453
file size, in human friendly format	588.0 MiB
md5	`4f9c197dfe6d3dc7e50a8611b4d3cfa2`
sha256	`edbe9ef1ee5cfe62e131d7650e07c031ab14fd0e8bd12c15a095b73039eb8`

Hashes for Anaconda2-5.0.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508806872.3056438
time file was last modified, in human readable format	2017-10-23 20:01:12
exact file size, in bytes	590135749
file size, in human friendly format	562.8 MiB
md5	`46fc99d1cf1e27f3b2a3eb63fee1a532`
sha256	`22350fe830e6786a263d7727e537f066b13fd9f4646982796bd20248da3f3`

Hashes for Anaconda2-5.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461108.192577
time file was last modified, in human readable format	2017-09-26 16:25:08
exact file size, in bytes	588579426
file size, in human friendly format	561.3 MiB
md5	8a2bbf7eb66290eb0bc82963056fb96c
sha256	3ee5cfe80d51685d6f374f83a9b76fa7ecbf7dc1a20f49a506e963641f2e1

Hashes for Anaconda2-4.4.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841768.0
time file was last modified, in human readable format	2017-05-26 18:36:08
exact file size, in bytes	459233116
file size, in human friendly format	438.0 MiB
md5	d2d5d213764a0c849eb1d53acba0d418
sha256	e5acf026892eaebb055e6317af96f295d39cf1d1a06ce6a1c6ca154ae3cab

Hashes for Anaconda2-4.3.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488839178.0
time file was last modified, in human readable format	2017-03-06 16:26:18
exact file size, in bytes	439742086
file size, in human friendly format	419.4 MiB
md5	`1961c7158bf89f4daf5b7a7d4f265075`
sha256	`f5d950451c038f9a7ca80d4036b6a8152c35d48fee685df3de486729dbae0`

Hashes for Anaconda2-4.3.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548783.0
time file was last modified, in human readable format	2017-01-27 14:26:23
exact file size, in bytes	438746148
file size, in human friendly format	418.4 MiB
md5	`899e90455db3120d584b2d4961c4eede`
sha256	`3e1d1026d2c0b87213a8b4a5f28431060b0cfe5cdc8a368b39248dbb66b53`

Hashes for Anaconda2-4.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1476750791.0
time file was last modified, in human readable format	2016-10-17 19:33:11
exact file size, in bytes	423495522
file size, in human friendly format	403.9 MiB
md5	`cd2ccc991b7f1503335367d80d0317b0`
sha256	`4a74d34c3a3a82df31673ab49497816b03547bab7054525fdd92eef63c8bc`

Hashes for Anaconda2-4.1.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994799.0
time file was last modified, in human readable format	2016-07-08 11:19:59
exact file size, in bytes	361721748
file size, in human friendly format	345.0 MiB
md5	`e88beae19868dc01fae908dd1e067bda`
sha256	`879385461cc65bd9dbf9639bbf4471ecf1713611617eda8d3a05f33a45682`

Hashes for Anaconda2-4.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131309.0
time file was last modified, in human readable format	2016-06-28 11:28:29
exact file size, in bytes	360909420
file size, in human friendly format	344.2 MiB
md5	`b2e2a6ee2fc2436a099ed0a3cc5e8fda`
sha256	`a97840be50d8c86b28caf8be1786bbe7485770b14501911e3e7331d33f2a3`

Hashes for Anaconda2-4.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268097.0
time file was last modified, in human readable format	2016-03-29 11:14:57
exact file size, in bytes	355703551
file size, in human friendly format	339.2 MiB
md5	7c4e4a25a38106d50dc3bc25a7a3009e
sha256	242691c7dc9e20143d7620fd9e0cc344fec7a2a534d1dba5f3b6522f04648

Hashes for Anaconda2-2.5.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454536531.0
time file was last modified, in human readable format	2016-02-03 15:55:31
exact file size, in bytes	385762781
file size, in human friendly format	367.9 MiB
md5	3256a5000b44e4fec1466e509aa641e6
sha256	0f546ed4f388299824e98a31ca9e3fe9823a49a2143d1cbd982caeb536e3d

Hashes for Anaconda2-2.4.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608450.0
time file was last modified, in human readable format	2015-12-08 15:00:50
exact file size, in bytes	257787337
file size, in human friendly format	245.8 MiB
md5	0407becd94e2c67d500700863cccaf82
sha256	1e2445aaf9faf84e801404bf89091fbf4a018709712a3901490fb3f45d44c

Hashes for Anaconda2-2.4.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502940.0
time file was last modified, in human readable format	2015-11-02 16:22:20
exact file size, in bytes	287613909
file size, in human friendly format	274.3 MiB
md5	`1da04d414117e3d5ffdae13a686f300f`
sha256	`d6842135062f3c3d2f8bd33318133376d4f2c789c32818b24f9010ca2240b`

Hashes for Anaconda-2.3.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775709.0
time file was last modified, in human readable format	2015-07-01 13:35:09
exact file size, in bytes	297482814
file size, in human friendly format	283.7 MiB
md5	`1fdb7eb4db925edb48f678c72f70f795`
sha256	`f920ae6211d9da3288b5e160100543667cf1ceb21fa09b16d6cda82f113e8`

Hashes for Anaconda-2.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315247.0
time file was last modified, in human readable format	2015-03-25 15:27:27
exact file size, in bytes	293316812
file size, in human friendly format	279.7 MiB
md5	`53777c4bbba5b6e6f9124e041f024bc9`
sha256	`65784323db94b0c297e998bc81db5978e8951801c44808589e2b9665ad199`

Hashes for Anaconda-2.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411662793.0
time file was last modified, in human readable format	2014-09-25 11:33:13
exact file size, in bytes	288368463
file size, in human friendly format	275.0 MiB
md5	`0632392578c6b4796c9c2a3964f9f2f9`
sha256	`d8001bae990e7024b81e74c6b06d0f488dd8717a9e0779db20d3e8831435b`

Hashes for Anaconda-2.0.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603370.0
time file was last modified, in human readable format	2014-06-12 15:02:50
exact file size, in bytes	256197898
file size, in human friendly format	244.3 MiB
md5	`8c3fa107375b1c4782531b7f6e7eddae`
sha256	`d6a0ce0422daa004929a4aef6b485d94f5e60b67f6d727047719815949fd5`

Hashes for Anaconda-2.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313841.0
time file was last modified, in human readable format	2014-05-28 16:50:41
exact file size, in bytes	247641929
file size, in human friendly format	236.2 MiB
md5	39d1624555ddc087785bc9557ecaa7b7
sha256	e2eb3805451a26235b2ed7f3e63535fc39bf32b46bfa8407f8b6240924cf8

Hashes for Anaconda-1.9.2-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1397142332.0
time file was last modified, in human readable format	2014-04-10 10:05:32
exact file size, in bytes	294660294
file size, in human friendly format	281.0 MiB
md5	432ac816e681c51f0238f30e4207e789
sha256	2fff6dca12507f675b04ed1f303d0ee99d755402c3b2b64c131d93c3b4f14

Hashes for Anaconda-1.9.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392932644.0
time file was last modified, in human readable format	2014-02-20 15:44:04
exact file size, in bytes	294577771
file size, in human friendly format	280.9 MiB
md5	`772b8e5dc385bf5ea3f78cdd21a8ec71`
sha256	`2aa707b162e71d488495085fd13232f8c30ac0f5003e6dd983c99897813d2`

Hashes for Anaconda-1.9.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049426.0
time file was last modified, in human readable format	2014-02-10 10:23:46
exact file size, in bytes	293412173
file size, in human friendly format	279.8 MiB
md5	`e702b99930507a43b59fd258744bd456`
sha256	`b74134e7626f10fc4d86209a3ebbb19de3c4404d0aecf071bedfa184bab22`

Hashes for Anaconda-1.8.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383595040.0
time file was last modified, in human readable format	2013-11-04 13:57:20
exact file size, in bytes	275773973
file size, in human friendly format	263.0 MiB
md5	`2b909458ddc208807efa3516c9ecab2f`
sha256	`fb92afc7750bc58ac12f3cbd65c18ee0f80ec22b80f07e236e739bf5ec5e7`

Hashes for Anaconda-1.7.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378746934.0
time file was last modified, in human readable format	2013-09-09 12:15:34
exact file size, in bytes	269206281
file size, in human friendly format	256.7 MiB
md5	`6e9e2fe69d3c1d847ca162b2f723f7b2`
sha256	`d277f7e162c77043e416d03a754389a0d9fb83636dc78bbb67e7433e29097`

Hashes for Anaconda-1.6.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372804212.0
time file was last modified, in human readable format	2013-07-02 17:30:12
exact file size, in bytes	206839424
file size, in human friendly format	197.3 MiB
md5	`01fe24a1c6605bec8d482dcda9de314a`
sha256	`7c79819dd40a14e52439664c3e88e89ecba29c5f4d2fd72726124d7a138f5`

Graphical installers for Anaconda with Python 3 on macOS

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703620.9068272
time file was last modified, in human readable format	2018-05-30 13:07:00
exact file size, in bytes	642866657
file size, in human friendly format	613.1 MiB
md5	`9c35bf27e9986701f7d80241616c665f`
sha256	`dae8befc73d32b480faef31fa6fb73332579442a524bc68f6d475743f5bb8`

Hashes for Anaconda3-5.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707346.8513427
time file was last modified, in human readable format	2018-02-15 09:09:06
exact file size, in bytes	623585451
file size, in human friendly format	594.7 MiB
md5	`6ed496221b843d1b5fe8463d3136b649`
sha256	`d6bf6309ccafa84314d85ca7421fddc16057ac2d824d698a213ccd597e896`

Hashes for Anaconda3-5.0.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508806879.6149647
time file was last modified, in human readable format	2017-10-23 20:01:19
exact file size, in bytes	596524910
file size, in human friendly format	568.9 MiB
md5	`eef112a1b8cbe8854e189eea1969f699`
sha256	`50c28594c785f5828990c95053468488563c775038b6744951847f9853ed0`

Hashes for Anaconda3-5.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461110.7116904
time file was last modified, in human readable format	2017-09-26 16:25:10
exact file size, in bytes	594734978
file size, in human friendly format	567.2 MiB
md5	`de004893c4d5714e06d4903e0780aabd`
sha256	`06d959384869290845bc61346bb33a18dd02573836f50ba263b72028d2a6a`

Hashes for Anaconda3-4.4.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841777.0
time file was last modified, in human readable format	2017-05-26 18:36:17
exact file size, in bytes	464033256
file size, in human friendly format	442.5 MiB
md5	`c6cd9c30b94c2ba2a5449e6f234d15f5`
sha256	`c5fc645f11505ac3ef710023b4072b7fb24ad31634b48e793e50b067dc301`

Hashes for Anaconda3-4.3.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488839187.0
time file was last modified, in human readable format	2017-03-06 16:26:27
exact file size, in bytes	444660396
file size, in human friendly format	424.1 MiB
md5	`390ba506140e4dfb7e0ab368f6ab18d6`
sha256	`ca608d58b1acf77b5c77d10e937b9084e5997b8706445bac3754459e54c43`

Hashes for Anaconda3-4.3.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548792.0
time file was last modified, in human readable format	2017-01-27 14:26:32
exact file size, in bytes	443649282
file size, in human friendly format	423.1 MiB
md5	`30b108a9cbc5d215a60187c5de89c459`
sha256	`f4522ac099ba292940bb47429e8e53eb9a0fe2ad3421513b2d618d0766337`

Hashes for Anaconda3-4.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1476750827.0
time file was last modified, in human readable format	2016-10-17 19:33:47
exact file size, in bytes	426843208
file size, in human friendly format	407.1 MiB
md5	`51ed7f9af7436a1a23068eb00509d6ad`
sha256	`44fe57910aa10967c4afe41ab5663cb49235bc718a9b7c8912d3ec7f95485`

Hashes for Anaconda3-4.1.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994875.0
time file was last modified, in human readable format	2016-07-08 11:21:15
exact file size, in bytes	364773025
file size, in human friendly format	347.9 MiB
md5	`9d396421683249ae850bd19637577f6e`
sha256	`b5e8cf44958d0aa03a7cc2da15fa835b1e14612b43b9772aef3270079d9b5`

Hashes for Anaconda3-4.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131312.0
time file was last modified, in human readable format	2016-06-28 11:28:32
exact file size, in bytes	363587059
file size, in human friendly format	346.7 MiB
md5	`665bf91beb7df29cfe36e6c135651ff5`
sha256	`83772b5fcd3d6deb945316ec96ecc7b0cdcd58c2c7a23f8f616771704e75a`

Hashes for Anaconda3-4.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268168.0
time file was last modified, in human readable format	2016-03-29 11:16:08
exact file size, in bytes	358139390
file size, in human friendly format	341.5 MiB
md5	b25796c49f9d3b47561c6eac9bbc77f0
sha256	32a089b1be465a8b03c837041bbfbcb761d644893719329ee59b253221456

Hashes for Anaconda3-2.5.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454536564.0
time file was last modified, in human readable format	2016-02-03 15:56:04
exact file size, in bytes	387740293
file size, in human friendly format	369.8 MiB
md5	a3c5835e965b3afb05e4a0472fe36267
sha256	b1a6945f0f025086806624c59de5d92e5234bb39a18b5517d8b1e0dc30b3b

Hashes for Anaconda3-2.4.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608454.0
time file was last modified, in human readable format	2015-12-08 15:00:54
exact file size, in bytes	259674929
file size, in human friendly format	247.6 MiB
md5	`feb6c8b1553b4de35cfa8c8c18c50d34`
sha256	`95e9f2d370f7816ed72b862c9413c973efb2ca17edd4d84550ce33e0d16d8`

Hashes for Anaconda3-2.4.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502943.0
time file was last modified, in human readable format	2015-11-02 16:22:23
exact file size, in bytes	280419790
file size, in human friendly format	267.4 MiB
md5	`64db05cc4c185354453c450ba7551331`
sha256	`791f045258bd39bbcdbb4c5425dce082ecd194074edd99fb401d746ad58da`

Hashes for Anaconda3-2.3.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775713.0
time file was last modified, in human readable format	2015-07-01 13:35:13
exact file size, in bytes	307072618
file size, in human friendly format	292.8 MiB
md5	`51e3b628d2f7580d0753a4dabd46f1d3`
sha256	`0b936ab3067bbf32b5a52768f31ff437f0e01fbcee028698444d1a964209c`

Hashes for Anaconda3-2.2.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427315292.0
time file was last modified, in human readable format	2015-03-25 15:28:12
exact file size, in bytes	302853736
file size, in human friendly format	288.8 MiB
md5	`f6963a1d098dc1aa70b198490cde34cf`
sha256	`16a5154267d7d52d3e7e0d12ec3405077df799c77ce382a3358238352656a`

Hashes for Anaconda3-2.1.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411663254.0
time file was last modified, in human readable format	2014-09-25 11:40:54
exact file size, in bytes	290765346
file size, in human friendly format	277.3 MiB
md5	`c863fb1f7f714917e4cb4dfaffd9d03f`
sha256	`2780df02f400e44c0adcd209825fdcf955559fe42f5b3689d5c46a01bdec2`

Hashes for Anaconda3-2.0.1-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603205.0
time file was last modified, in human readable format	2014-06-12 15:00:05
exact file size, in bytes	241942876
file size, in human friendly format	230.7 MiB
md5	1c22595eedfc62ff18a8786934e19c9c
sha256	0d53815a83a50bdcfcb5ada686f582730bcc93b95295dd04572ee5162724e

Hashes for Anaconda3-2.0.0-MacOSX-x86_64.pkg

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401226216.0
time file was last modified, in human readable format	2014-05-27 16:30:16
exact file size, in bytes	233699227
file size, in human friendly format	222.9 MiB
md5	e59bdfb282ec34a1a54db3eb3bb4eea0
sha256	4d4189ec0c514d344389e216b3ad4eeacd667426d902c5da416ebd7caa54d

Command line installers for Anaconda with Python 2 on macOS

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703534.8069696
time file was last modified, in human readable format	2018-05-30 13:05:34
exact file size, in bytes	552703968
file size, in human friendly format	527.1 MiB
md5	`b1f3fcf58955830b65613a4a8d75c3cf`
sha256	`d7d46e566306da5979cd5632079497fe6103b980e3a089ccf27a9f30cbee8`

Hashes for Anaconda2-5.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707333.7077339
time file was last modified, in human readable format	2018-02-15 09:08:53
exact file size, in bytes	530462553
file size, in human friendly format	505.9 MiB
md5	`e9845ccf67542523c5be09552311666e`
sha256	`b686e01aeadb33526d9c154a0ac6f691dfad135080df96fb44d3ae1e4b128`

Hashes for Anaconda2-5.0.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508806264.3489628
time file was last modified, in human readable format	2017-10-23 19:51:04
exact file size, in bytes	510164626
file size, in human friendly format	486.5 MiB
md5	`17314016dced36614a3bef8ff3db7066`
sha256	`e3a9a5c84cb89ff079b0781ba773a3433d490fe0cfc24042c613a5674748d`

Hashes for Anaconda2-5.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461109.035615
time file was last modified, in human readable format	2017-09-26 16:25:09
exact file size, in bytes	508843477
file size, in human friendly format	485.3 MiB
md5	b8d555fae2b4994f1094c2da85c7e9a4
sha256	d85198c63657924fae11b6ea5961f50d81d09a1185d6f0a9a9d5bc69eb788

Hashes for Anaconda2-4.4.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841752.0
time file was last modified, in human readable format	2017-05-26 18:35:52
exact file size, in bytes	393583324
file size, in human friendly format	375.4 MiB
md5	a57e5c631a7d0c63552519f05ab243a4
sha256	ab95aef1110c2a385fd39a17e5f11dfbaabce25c1a5944598de164d7a2772

Hashes for Anaconda2-4.3.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488839162.0
time file was last modified, in human readable format	2017-03-06 16:26:02
exact file size, in bytes	375651870
file size, in human friendly format	358.2 MiB
md5	`eb1e7f853f64ad8babe1330a068e94e9`
sha256	`35261360f2b01793f441b29715a94052dceaef1137866b7323c76be83c5bc`

Hashes for Anaconda2-4.3.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548768.0
time file was last modified, in human readable format	2017-01-27 14:26:08
exact file size, in bytes	374699540
file size, in human friendly format	357.3 MiB
md5	`80b7958fc805d371d60e133af826752c`
sha256	`834ac0287062929ab5930661735ee617fd379bdfe79f3e0a20aebd614835b`

Hashes for Anaconda2-4.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009402.0
time file was last modified, in human readable format	2016-09-27 15:50:02
exact file size, in bytes	363251366
file size, in human friendly format	346.4 MiB
md5	`52f8b74e0c462575efc297c8f4e6cf14`
sha256	`a8b3ef86233635d9dcc3499dc384980762a0b42d354a318f8307029c399db`

Hashes for Anaconda2-4.1.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994800.0
time file was last modified, in human readable format	2016-07-08 11:20:00
exact file size, in bytes	310125837
file size, in human friendly format	295.8 MiB
md5	f62a0a47a42504e139a5122ad641b40c
sha256	3b2fb323eb26c1c58788f63c41e164c20c417f7f24e30b8057e92ab4d6102

Hashes for Anaconda2-4.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131310.0
time file was last modified, in human readable format	2016-06-28 11:28:30
exact file size, in bytes	309460309
file size, in human friendly format	295.1 MiB
md5	c18a0f560668e9d1215ed600fb64b0cf
sha256	8b2c2a32f5e0da75cf8c81c568124cc1ea701a58cd46b7816133573a7f5b7

Hashes for Anaconda2-4.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268099.0
time file was last modified, in human readable format	2016-03-29 11:14:59
exact file size, in bytes	304288480
file size, in human friendly format	290.2 MiB
md5	a3443b46f99bc6680c77c688af1b1f5a
sha256	aa7ba6e1a40e08e672660c00c3151f0124faa61b598d75bdd07ebe1d24873

Hashes for Anaconda2-2.5.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535682.0
time file was last modified, in human readable format	2016-02-03 15:41:22
exact file size, in bytes	331485310
file size, in human friendly format	316.1 MiB
md5	57c024647fd3a149aa6d787feb35daa2
sha256	e7aa3b41210ee7ccf3c12e5b5ea43190d1811b58eaeca8584ccffa468ac8a

Hashes for Anaconda2-2.4.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608451.0
time file was last modified, in human readable format	2015-12-08 15:00:51
exact file size, in bytes	222326344
file size, in human friendly format	212.0 MiB
md5	1aecf1e5808dbfb9fa81d139abdbeb34
sha256	f4bd45a21e0dff106e36d11cfd532f2b5050d3b792cc0627ab231089341d2

Hashes for Anaconda2-2.4.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502940.0
time file was last modified, in human readable format	2015-11-02 16:22:20
exact file size, in bytes	251172115
file size, in human friendly format	239.5 MiB
md5	`6e39a0b4470f6517c98f6edd21becd35`
sha256	`53c9123c9d508555100805fdb44d9845511c937e7a34f237beb19168d655e`

Hashes for Anaconda-2.3.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775710.0
time file was last modified, in human readable format	2015-07-01 13:35:10
exact file size, in bytes	262015432
file size, in human friendly format	249.9 MiB
md5	`a9c057a22f106748956b708c50f52239`
sha256	`c4bb59a57bf44dde80612041bbbcfd2e5cab8534842209ef456da7a46f919`

Hashes for Anaconda-2.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314653.0
time file was last modified, in human readable format	2015-03-25 15:17:33
exact file size, in bytes	259147994
file size, in human friendly format	247.1 MiB
md5	453ab3de72ee95b7cb7ee5ee7298fbdf
sha256	20570e2f3911e38a78d8f888f3ff445d6c0cf97a2fca40d6956b48d12aaef

Hashes for Anaconda-2.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660393.0
time file was last modified, in human readable format	2014-09-25 10:53:13
exact file size, in bytes	252758049
file size, in human friendly format	241.0 MiB
md5	4d323aea34408b16a316ee5596817d47
sha256	128fd4f53e0895e0d23f33e924ae32e01171c2914b044d2b157a949710810

Hashes for Anaconda-2.0.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603376.0
time file was last modified, in human readable format	2014-06-12 15:02:56
exact file size, in bytes	224812734
file size, in human friendly format	214.4 MiB
md5	85d261fd4e651f891ca5d0df69441e00
sha256	4ecda163c6f46e70cc6a1fe62dece4c6ecd6474845129cc95a1d4e18c42f8

Hashes for Anaconda-2.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313845.0
time file was last modified, in human readable format	2014-05-28 16:50:45
exact file size, in bytes	216067773
file size, in human friendly format	206.1 MiB
md5	`ec288bc9901facac5a1e098ded8c9936`
sha256	`ad6271ad21403166bf54d0734ba8c7f7eb65bb78a70d67c58c15b6874cddc`

Hashes for Anaconda-1.9.2-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1396996443.0
time file was last modified, in human readable format	2014-04-08 17:34:03
exact file size, in bytes	257273472
file size, in human friendly format	245.4 MiB
md5	`9d4bfe3f859718c4ab9c06209c5b8175`
sha256	`be4611ca671f80b984fa330d4ecf82244c388abbdb5c7679a4e6e806b4dca`

Hashes for Anaconda-1.9.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392922925.0
time file was last modified, in human readable format	2014-02-20 13:02:05
exact file size, in bytes	257212033
file size, in human friendly format	245.3 MiB
md5	`6ef81bc54a6ab506f352b5589ea80f81`
sha256	`7e4358adbaae2db9e17d1e0e4263b9a0174394c8f115c89d285c3f0f9206f`

Hashes for Anaconda-1.9.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049427.0
time file was last modified, in human readable format	2014-02-10 10:23:47
exact file size, in bytes	256300639
file size, in human friendly format	244.4 MiB
md5	`ddd474c01696cc02dcaea91da1d72389`
sha256	`722fe4d4406e88c5023e7ee21dc1401bb2a540d6c031d303f0330a95e6013`

Hashes for Anaconda-1.8.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383592216.0
time file was last modified, in human readable format	2013-11-04 13:10:16
exact file size, in bytes	239935643
file size, in human friendly format	228.8 MiB
md5	`9fd7dd485c5f04fb65699a290e69671c`
sha256	`5844ca595b5930399a1213db64ab53e9b7e2fc1c26d8f11769c161fe4f566`

Hashes for Anaconda-1.7.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378745565.0
time file was last modified, in human readable format	2013-09-09 11:52:45
exact file size, in bytes	234119145
file size, in human friendly format	223.3 MiB
md5	16194eb9be2301eeb135f9f01695a566
sha256	046b592245bc2c11e733acb9700dc50947f2eff0f30fec4a4a5bf79368dfa

Hashes for Anaconda-1.6.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372784365.0
time file was last modified, in human readable format	2013-07-02 11:59:25
exact file size, in bytes	178279644
file size, in human friendly format	170.0 MiB
md5	4b60123e71864c447a0adc16398d5386
sha256	bbc15de34208ce8af5aceedeea1334636fe94c578b9890896729f1a61ace5

Hashes for Anaconda-1.6.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1371842774.0
time file was last modified, in human readable format	2013-06-21 14:26:14
exact file size, in bytes	177173725
file size, in human friendly format	169.0 MiB
md5	cccdd0353bfd46d3a93143fc6e47d728
sha256	e03317888c36c07451a349577b426f435a75075d1ee71e204eb9d5dd23936

Hashes for Anaconda-1.5.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368127580.0
time file was last modified, in human readable format	2013-05-09 14:26:20
exact file size, in bytes	174295689
file size, in human friendly format	166.2 MiB
md5	03942512daf1b39eb3ff9016fc7efa0c
sha256	6d3c86a2fdbaeeec2a6c251d5c9034a32b7c68a0437f2fac0b8f25125fe68

Hashes for Anaconda-1.5.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368022724.0
time file was last modified, in human readable format	2013-05-08 09:18:44
exact file size, in bytes	174295630
file size, in human friendly format	166.2 MiB
md5	6fe90601dbcecb29a2afcaf44aeb37f6
sha256	c69609f0f48f33ca5a12d425a9e4d0fc91b2c09d0345a590e1d7772644672

Hashes for Anaconda-1.4.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1362869217.0
time file was last modified, in human readable format	2013-03-09 16:46:57
exact file size, in bytes	163952262
file size, in human friendly format	156.4 MiB
md5	`db8779f0a663e025da1b19755f372a57`
sha256	`e5d5dae6e93bb7df528abc19f5ed3a69cc4bc867836bdc56886c5a3768fcc`

Command line installers for Anaconda with Python 3 on macOS

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703623.1079257
time file was last modified, in human readable format	2018-05-30 13:07:03
exact file size, in bytes	548669102
file size, in human friendly format	523.3 MiB
md5	`b5b789c01e1992de55ee911754c310d4`
sha256	`c8089121dc89ffe8f9a0c01205bab75a112821a13d413152d6690f5eef094`

Hashes for Anaconda3-5.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707424.3679326
time file was last modified, in human readable format	2018-02-15 09:10:24
exact file size, in bytes	536124653
file size, in human friendly format	511.3 MiB
md5	`047e12523fd287149ecd80c803598429`
sha256	`be705b3c3a0ca29ee32ce7658890bb5edb32a9eadedc09dec3d7e3cfbfd23`

Hashes for Anaconda3-5.0.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508806270.7502434
time file was last modified, in human readable format	2017-10-23 19:51:10
exact file size, in bytes	514894862
file size, in human friendly format	491.0 MiB
md5	`3c0f4bf6d9a68d91f6da65051046e106`
sha256	`f438a0af923bc1edc7bca53f496c59a668d1a08b48c768f443ad7f5ea2b8b`

Hashes for Anaconda3-5.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461111.7357368
time file was last modified, in human readable format	2017-09-26 16:25:11
exact file size, in bytes	513706836
file size, in human friendly format	489.9 MiB
md5	`a72e7b22c29f0b4e05579cb8453f89fa`
sha256	`23df1e3a38a6b4aaa0ab559d0c1e51be76eca5d75cb595d473d223c8d17e7`

Hashes for Anaconda3-4.4.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841759.0
time file was last modified, in human readable format	2017-05-26 18:35:59
exact file size, in bytes	398907650
file size, in human friendly format	380.4 MiB
md5	`3958ac6cb84731e560dd833256aa5b15`
sha256	`10fe58f09ae524df2548d17b8bb1e75db17da597a6ec10d695ce01387a2d7`

Hashes for Anaconda3-4.3.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488839169.0
time file was last modified, in human readable format	2017-03-06 16:26:09
exact file size, in bytes	381078558
file size, in human friendly format	363.4 MiB
md5	`fdf4ad01fadb58415bb4c6119153e04a`
sha256	`a42267203e207cb5e0f539e0d879ead12e436311825c7114d0edd880d001b`

Hashes for Anaconda3-4.3.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548775.0
time file was last modified, in human readable format	2017-01-27 14:26:15
exact file size, in bytes	380197908
file size, in human friendly format	362.6 MiB
md5	`e080c503c27d5c072d3e324ee1822641`
sha256	`c53059b810c5e7a9a5ef9c46a7ed76675dfc7183f4ea867b4d81449cbd5a0`

Hashes for Anaconda3-4.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009407.0
time file was last modified, in human readable format	2016-09-27 15:50:07
exact file size, in bytes	366497043
file size, in human friendly format	349.5 MiB
md5	`7cb61e355eb860e342a5e27236e3f375`
sha256	`95448921601e1952e01a17ba9767cd3621c154af7fc52dd6b7f57d462155a`

Hashes for Anaconda3-4.1.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994877.0
time file was last modified, in human readable format	2016-07-08 11:21:17
exact file size, in bytes	313217912
file size, in human friendly format	298.7 MiB
md5	`185aa68d5841869cb7cb3a031bd63936`
sha256	`7c3c06e9281c41f1213d357cb5f233fd99d6d0db6bdba8d9fd7cfad1f1a85`

Hashes for Anaconda3-4.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131313.0
time file was last modified, in human readable format	2016-06-28 11:28:33
exact file size, in bytes	312081344
file size, in human friendly format	297.6 MiB
md5	`262c5c9a12d94a956ceb301d9f258c77`
sha256	`4c45c8d75665fa5194ebe4e355d3427f5aa385f77eb2b5002c0c78d8ae7f2`

Hashes for Anaconda3-4.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268181.0
time file was last modified, in human readable format	2016-03-29 11:16:21
exact file size, in bytes	306950558
file size, in human friendly format	292.7 MiB
md5	`efd870aa3fabc8f4865a1b9567e69b69`
sha256	`704a776c0cf3fcca6e0c5a1e6b6043728229cfac813bff28f003157771824`

Hashes for Anaconda3-2.5.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535687.0
time file was last modified, in human readable format	2016-02-03 15:41:27
exact file size, in bytes	333727463
file size, in human friendly format	318.3 MiB
md5	`7223be67e908fe3db8199129e7253da1`
sha256	`9bb0f926927db210f8c2a8de881213d1a44c7b3d6dbcb93dfa6b99ed4bbd3`

Hashes for Anaconda3-2.4.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608454.0
time file was last modified, in human readable format	2015-12-08 15:00:54
exact file size, in bytes	224240817
file size, in human friendly format	213.9 MiB
md5	`a5831d2a9b7baa9cdd42d7979b32f02c`
sha256	`22a3267638da9b7d64210d7da90d8762da7948234c21c0010a74f2621ee0e`

Hashes for Anaconda3-2.4.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502943.0
time file was last modified, in human readable format	2015-11-02 16:22:23
exact file size, in bytes	245160775
file size, in human friendly format	233.8 MiB
md5	9deaaec2262bbac751a75f8bed4c5ab6
sha256	f0cd785dbed0bab28dfc08a391c9de1b01633422fa317cb8365513a1ae5ae

Hashes for Anaconda3-2.3.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775714.0
time file was last modified, in human readable format	2015-07-01 13:35:14
exact file size, in bytes	269910147
file size, in human friendly format	257.4 MiB
md5	96fb1d4ba62529e5534f23e143ce3967
sha256	6a0c94a49f41f9fda0138c8e966bd7b0a8965d6648fd21ffbd645d1453848

Hashes for Anaconda3-2.2.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314664.0
time file was last modified, in human readable format	2015-03-25 15:17:44
exact file size, in bytes	266868602
file size, in human friendly format	254.5 MiB
md5	793f030f8694659f125399b66123bb78
sha256	81a2089ea6127717f146454e99ea0be2bd595193e4151bb05b4c15749b1d8

Hashes for Anaconda3-2.1.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660403.0
time file was last modified, in human readable format	2014-09-25 10:53:23
exact file size, in bytes	255307129
file size, in human friendly format	243.5 MiB
md5	59e2ffc9366dd32975c2da9e6eb8854a
sha256	efdb7e9d1e539cbed62dc3874b0de6a141f36684e6fbc05018e072b217e24

Hashes for Anaconda3-2.0.1-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603209.0
time file was last modified, in human readable format	2014-06-12 15:00:09
exact file size, in bytes	213128084
file size, in human friendly format	203.3 MiB
md5	65dfe2f379cc14d5c8f7e05a57ce32aa
sha256	7a08509d4e45efcc7055a6d06d8406a773716500bd869a4e85312ff131155

Hashes for Anaconda3-2.0.0-MacOSX-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401228173.0
time file was last modified, in human readable format	2014-05-27 17:02:53
exact file size, in bytes	204782483
file size, in human friendly format	195.3 MiB
md5	ba8d37fdafb2381585ddb24bde34b9ff
sha256	776a1cf8a8e898b41bb6558c093632cc922698dc48486fee35d1e8eae3f60

Anaconda with Python 2 on 64-bit Linux

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703473.7962363
time file was last modified, in human readable format	2018-05-30 13:04:33
exact file size, in bytes	632688935
file size, in human friendly format	603.4 MiB
md5	5c034a4ab36ec9b6ae01fa13d8a04462
sha256	cb0d7a08b0e2cec4372033d3269979b4e72e2353ffd1444f57cb38bc96212

Hashes for Anaconda2-5.2.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703471.0561135
time file was last modified, in human readable format	2018-05-30 13:04:31
exact file size, in bytes	282733770
file size, in human friendly format	269.6 MiB
md5	479633a95906ea6d41056ebe84a4c47b
sha256	a8fcac3f0884520c35103e76549fcc45d64d8806517ba02d8ade4028e1f77

Hashes for Anaconda2-5.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707330.5935898
time file was last modified, in human readable format	2018-02-15 09:08:50
exact file size, in bytes	558878810
file size, in human friendly format	533.0 MiB
md5	5b1b5784cae93cf696e11e66983d8756
sha256	5f26ee92860d1dffdcd20910ff2cf75572c39d2892d365f4e867a611cca2a

Hashes for Anaconda2-5.1.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707329.6225448
time file was last modified, in human readable format	2018-02-15 09:08:49
exact file size, in bytes	280296529
file size, in human friendly format	267.3 MiB
md5	e894dcc547a1c7d67deb04f6bba7223a
sha256	ff9baa4d3710bb24bc3a6a40c0f4ef69150f7608af5be6ada1ff99d01d1be

Hashes for Anaconda2-5.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508865232.4529853
time file was last modified, in human readable format	2017-10-24 12:13:52
exact file size, in bytes	532375438
file size, in human friendly format	507.7 MiB
md5	dc13fe5502cd78dd03e8a727bb9be63f
sha256	23c676510bc87c95184ecaeb327c0b2c88007278e0d698622e2dd8fb14d9f

Hashes for Anaconda2-5.0.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506959412.7513092
time file was last modified, in human readable format	2017-10-02 10:50:12
exact file size, in bytes	530931450
file size, in human friendly format	506.3 MiB
md5	35bea553072ea1f28090e851105c1b00
sha256	18730808d863a5c194ab3f59dd395c1a63cbd769c9bfb1df65efe61ee62fc

Hashes for Anaconda2-5.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506454641.7079751
time file was last modified, in human readable format	2017-09-26 14:37:21
exact file size, in bytes	530296562
file size, in human friendly format	505.7 MiB
md5	2272857fcf773fc75a1bc49f6d507a48
sha256	58a7117f89c40275114bf7e824a613a963da2b0fe63f2ec3c1175fea785b4

Hashes for Anaconda2-5.0.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506461107.2145326
time file was last modified, in human readable format	2017-09-26 16:25:07
exact file size, in bytes	296001230
file size, in human friendly format	282.3 MiB
md5	157890d591c61a9b511f8452476d6d19
sha256	e0512f3c81251e5dcd48fcf02fe2044a66071dc8681269b1375ac5443f971

Hashes for Anaconda2-4.4.0.1-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1501103402.0
time file was last modified, in human readable format	2017-07-26 16:10:02
exact file size, in bytes	284629257
file size, in human friendly format	271.4 MiB
md5	ce166de6f116acd08cd313f9c55c04d6
sha256	e14acab146181699e47ca108fc624ecebaaaac52851312962c649899459d9

Hashes for Anaconda2-4.4.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495840968.0
time file was last modified, in human readable format	2017-05-26 18:22:48
exact file size, in bytes	508722275
file size, in human friendly format	485.2 MiB
md5	`d72add23bc937ccdfc7de4f47deff843`
sha256	`2d30b91ed4d215b6b4a15162a3389e9057b15445a0c02da71bd7bd272e7b8`

Hashes for Anaconda2-4.4.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495053920.0
time file was last modified, in human readable format	2017-05-17 15:45:20
exact file size, in bytes	290045511
file size, in human friendly format	276.6 MiB
md5	`511fdc6f6c29b1c3a702f3792182faf0`
sha256	`c19edfd9a3bd2fcb37ddb0c3aa09339c9e23145269957fac75e9b2abca408`

Hashes for Anaconda2-4.3.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838334.0
time file was last modified, in human readable format	2017-03-06 16:12:14
exact file size, in bytes	484472684
file size, in human friendly format	462.0 MiB
md5	`51336ab38e15ce607b55539c60be2c29`
sha256	`e9b8f2645df6b1527ba56d61343162e0794acc3ee8dde2a6bba353719e2d8`

Hashes for Anaconda2-4.3.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548108.0
time file was last modified, in human readable format	2017-01-27 14:15:08
exact file size, in bytes	483530594
file size, in human friendly format	461.1 MiB
md5	`5f2c3bd60ddbd0e213f7a1fc25bb88b4`
sha256	`7c52e6e99aabb24a49880130615a48e685da444c3c14eb48d6a65f3313bf7`

Hashes for Anaconda2-4.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009394.0
time file was last modified, in human readable format	2016-09-27 15:49:54
exact file size, in bytes	467689464
file size, in human friendly format	446.0 MiB
md5	`a0d1fbe47014b71c6764d76fb403f217`
sha256	`beee286d24fb37dd6555281bba39b3deb5804baec509a9dc5c69185098cf6`

Hashes for Anaconda2-4.1.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994796.0
time file was last modified, in human readable format	2016-07-08 11:19:56
exact file size, in bytes	419038579
file size, in human friendly format	399.6 MiB
md5	`f7bb3c0ccf23c9789bb895335aa68bf3`
sha256	`9413b1d3ca9498ba6f53913df9c43d685dd973440ff10b7fe0c45b1cbdcb5`

Hashes for Anaconda2-4.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131308.0
time file was last modified, in human readable format	2016-06-28 11:28:28
exact file size, in bytes	418188731
file size, in human friendly format	398.8 MiB
md5	`e24d4264205d8d0c8533617db99ff1d3`
sha256	`3b7e504ca0132fb555d1f10e174cae07007f1bc6898cad0f7d416a68aca01`

Hashes for Anaconda2-4.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268095.0
time file was last modified, in human readable format	2016-03-29 11:14:55
exact file size, in bytes	411562823
file size, in human friendly format	392.5 MiB
md5	`31ed3ef07435d7068e1e03be49381b13`
sha256	`ae312143952ca00e061a656c2080e0e4fd3532721282ba8e2978177cad71a`

Hashes for Anaconda2-2.5.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535678.0
time file was last modified, in human readable format	2016-02-03 15:41:18
exact file size, in bytes	409842279
file size, in human friendly format	390.9 MiB
md5	`f8eb687af8c9b4e81968de8c63b0d991`
sha256	`e10abf459cde4a838bd6fc5ca03023c3401b81ad470627acde5a298d56715`

Hashes for Anaconda2-2.4.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608449.0
time file was last modified, in human readable format	2015-12-08 15:00:49
exact file size, in bytes	277827702
file size, in human friendly format	265.0 MiB
md5	`c9317dcb28a2e0c98c34ebc341e0d145`
sha256	`2de682c96edf8cca2852071a84ff860025fbe8c502218e1995acd5ab47e8c`

Hashes for Anaconda2-2.4.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502939.0
time file was last modified, in human readable format	2015-11-02 16:22:19
exact file size, in bytes	302444354
file size, in human friendly format	288.4 MiB
md5	1ab870a0184b84594fccf2027c9be887
sha256	49d19834da06b1b82b6fa85bc647d2e78fa5957d0cbae3ccd6c695a541bef

Hashes for Anaconda-2.3.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775708.0
time file was last modified, in human readable format	2015-07-01 13:35:08
exact file size, in bytes	339594168
file size, in human friendly format	323.9 MiB
md5	c3100392685b5a62c8509c0588ce9376
sha256	7c02499e9511c127d225992cfe1cd815e88fd46cd8a5b3cdf764f3fb4d8d4

Hashes for Anaconda-2.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314808.0
time file was last modified, in human readable format	2015-03-25 15:20:08
exact file size, in bytes	332753610
file size, in human friendly format	317.3 MiB
md5	`3234b2b2d7f7031432c1fd9870d15f58`
sha256	`ca2582cb2188073b0f348ad42207211a2b85c10b244265b5b27bab04481b8`

Hashes for Anaconda-2.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660215.0
time file was last modified, in human readable format	2014-09-25 10:50:15
exact file size, in bytes	353806962
file size, in human friendly format	337.4 MiB
md5	`74a888f8683f67053a030e37d0eae1cf`
sha256	`191fbf290747614929d0bdd576e330c944b22a67585d1c185e0d2b3a3e65e`

Hashes for Anaconda-2.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603353.0
time file was last modified, in human readable format	2014-06-12 15:02:33
exact file size, in bytes	343791922
file size, in human friendly format	327.9 MiB
md5	`ae96da7de52ab1a64d4ed3fa4b43da25`
sha256	`074204fa26872b4a946123071d15b8390c0e5441352c6b65b2abd32511bff`

Hashes for Anaconda-2.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313830.0
time file was last modified, in human readable format	2014-05-28 16:50:30
exact file size, in bytes	332323121
file size, in human friendly format	316.9 MiB
md5	`480ba8864579a457db91cd774bd373c1`
sha256	`3aa27ddf4a0ba5046ba52b97da99e20eb0614273d905bd73e016852451908`

Hashes for Anaconda-1.9.2-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1396996358.0
time file was last modified, in human readable format	2014-04-08 17:32:38
exact file size, in bytes	507498869
file size, in human friendly format	484.0 MiB
md5	`863ee49f52bda17810ab1b94a52f8c95`
sha256	`7181d399833a2549a9584255bb477487f2fde1fda4c7f7215d6034ea2fcfa`

Hashes for Anaconda-1.9.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392924916.0
time file was last modified, in human readable format	2014-02-20 13:35:16
exact file size, in bytes	507437430
file size, in human friendly format	483.9 MiB
md5	`9d973e9ac715ce3241c3785704565971`
sha256	`f6455e06a72b8cc11c8a96fb88a85518a2f7b2a1d6f1065f777d7ab4386f0`

Hashes for Anaconda-1.9.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049385.0
time file was last modified, in human readable format	2014-02-10 10:23:05
exact file size, in bytes	648831310
file size, in human friendly format	618.8 MiB
md5	`52ed5f32f7e36b75b5f951ab58a4bc08`
sha256	`855f1265e4c0b40d50f5a3a0fe7bae05b1cccb0a5301b378a19e0a8f72629`

Hashes for Anaconda-1.8.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383601032.0
time file was last modified, in human readable format	2013-11-04 15:37:12
exact file size, in bytes	488287119
file size, in human friendly format	465.7 MiB
md5	`398d4b7ddc5c0a16c556c415b2444266`
sha256	`69f42966d918f4197040e4dd126d2e3cc3c267bb49869dbf2d6ef277ed5de`

Hashes for Anaconda-1.7.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378677719.0
time file was last modified, in human readable format	2013-09-08 17:01:59
exact file size, in bytes	474606301
file size, in human friendly format	452.6 MiB
md5	`01dc7d6df2ed592e5401ab4fbe3aed4a`
sha256	`6115cfae55a0746b4ae4128be839c99db39d02124160d9c531ca086c4d606`

Hashes for Anaconda-1.6.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372784262.0
time file was last modified, in human readable format	2013-07-02 11:57:42
exact file size, in bytes	333017000
file size, in human friendly format	317.6 MiB
md5	`70a1294c01e3ab5925fc52f2603de159`
sha256	`81d1819ba08069343f228b9c819cdba0e4d15f2142c0c03365759980 8c396`

Hashes for Anaconda-1.6.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1371842631.0
time file was last modified, in human readable format	2013-06-21 14:23:51
exact file size, in bytes	324528041
file size, in human friendly format	309.5 MiB
md5	`207a0b4ebde49bcde67925ac8c72fe37`
sha256	`20f5b70193af4b0b8f10aa0e66aabca552846ec8f4958757ff3f4b79ef7b3`

Hashes for Anaconda-1.5.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368022716.0
time file was last modified, in human readable format	2013-05-08 09:18:36
exact file size, in bytes	321578266
file size, in human friendly format	306.7 MiB
md5	`8319288082262fefbe322451aeae06ce`
sha256	`f4cdc194f076e1b438c8a34e7e5f53e70c2200b411b2d0af719e23fe35c64`

Hashes for Anaconda-1.4.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1362869198.0
time file was last modified, in human readable format	2013-03-09 16:46:38
exact file size, in bytes	300831068
file size, in human friendly format	286.9 MiB
md5	`9be0e7340f0cd2d2cbd5acbe8e988f45`
sha256	`85ae8a0a6e3a41cf7845be3def36ed40582d3dc6e6a50e99063eaf6f1abee`

Anaconda with Python 3 on 64-bit Linux

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703543.880376
time file was last modified, in human readable format	2018-05-30 13:05:43
exact file size, in bytes	651745206
file size, in human friendly format	621.6 MiB
md5	3e58f494ab9fbe12db4460dc152377b5
sha256	09f53738b0cd3bb96f5b1bac488e5528df9906be2480fe61df40e0e0d19e3

Hashes for Anaconda3-5.2.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703540.2002113
time file was last modified, in human readable format	2018-05-30 13:05:40
exact file size, in bytes	302261137
file size, in human friendly format	288.3 MiB
md5	cbd1d5435ead2b0b97dba5b3cf45d694
sha256	024c811526ffc40ed6fa243a25795fbab5b41413372cd5a276aca69a930ef

Hashes for Anaconda3-5.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707337.4719083
time file was last modified, in human readable format	2018-02-15 09:08:57
exact file size, in bytes	577996269
file size, in human friendly format	551.2 MiB
md5	`966406059cf7ed89cc82eb475ba506e5`
sha256	`7e6785caad25e33930bc03fac4994a434a21bc8401817b7efa28f53619fa9`

Hashes for Anaconda3-5.1.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707336.279853
time file was last modified, in human readable format	2018-02-15 09:08:56
exact file size, in bytes	299557404
file size, in human friendly format	285.7 MiB
md5	`47b5b2b17b7dbac0d4d0f0a4653f5b1c`
sha256	`58d1d093450dabefef9279694c9345afed78acf1c334d64a9241bc725f45a`

Hashes for Anaconda3-5.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508799175.9727552
time file was last modified, in human readable format	2017-10-23 17:52:55
exact file size, in bytes	550796553
file size, in human friendly format	525.3 MiB
md5	`c989ecc8b648ab8a64731aaee9ed2e7e`
sha256	`55e4db1919f49c92d5abbf27a4be5986ae157f074bf9f8238963cd4582a40`

Hashes for Anaconda3-5.0.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506959414.4703865
time file was last modified, in human readable format	2017-10-02 10:50:14
exact file size, in bytes	549434488
file size, in human friendly format	524.0 MiB
md5	`614cc8f244e956b41c75417dd1ec96fd`
sha256	`092c92427f44687d789a41922ce8426fbdc3c529cc9d6d4ee6de5b62954b9`

Hashes for Anaconda3-5.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506454642.7570226
time file was last modified, in human readable format	2017-09-26 14:37:22
exact file size, in bytes	548789360
file size, in human friendly format	523.4 MiB
md5	`bb2656314d22aecae6af243ddbbfb32c`
sha256	`67f5c20232a3e493ea3f19a8e273e0618ab678fa14b03b59b178361306214`

Hashes for Anaconda3-5.0.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506368371.0870926
time file was last modified, in human readable format	2017-09-25 14:39:31
exact file size, in bytes	310695077
file size, in human friendly format	296.3 MiB
md5	`8fe5b29ca5be3ff11411621f79babfc2`
sha256	`3574d423084e604a9d85a9f38ea481e0fc9e678923e2d3b9c4ec7340e1644`

Hashes for Anaconda3-4.4.0.1-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1501103322.0
time file was last modified, in human readable format	2017-07-26 16:08:42
exact file size, in bytes	299425582
file size, in human friendly format	285.6 MiB
md5	`fe7c87abd9fd70dc0cb4f83cc22d336f`
sha256	`d7c367c9c4fffec37c31c6570218c9944867c96fde5e9b0249673beda24ba`

Hashes for Anaconda3-4.4.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495840984.0
time file was last modified, in human readable format	2017-05-26 18:23:04
exact file size, in bytes	523283080
file size, in human friendly format	499.0 MiB
md5	50f19b935dae7361978a04d9c7c355cd
sha256	3301b37e402f3ff3df216fe0458f1e6a4ccbb7e67b4d626eae9651de5ea3a

Hashes for Anaconda3-4.4.0-Linux-ppc64le.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495053950.0
time file was last modified, in human readable format	2017-05-17 15:45:50
exact file size, in bytes	304862316
file size, in human friendly format	290.7 MiB
md5	8c663e2056c7c57ac0075774b1f790be
sha256	605251829edecd0c39df8db856d4f09e406454468c3f128c14a7446a4efdf

Hashes for Anaconda3-4.3.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838344.0
time file was last modified, in human readable format	2017-03-06 16:12:24
exact file size, in bytes	497343851
file size, in human friendly format	474.3 MiB
md5	9209864784250d6855886683ed702846
sha256	4447b93d2c779201e5fb50cfc45de0ec96c3804e7ad0fe201ab6b99f73e90

Hashes for Anaconda3-4.3.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548121.0
time file was last modified, in human readable format	2017-01-27 14:15:21
exact file size, in bytes	496412001
file size, in human friendly format	473.4 MiB
md5	dbe2e78adeca1923643be2ecaacd6227
sha256	e9169c3a5029aa820393ac92704eb9ee0701778a085ca7bdc3c57b388ac1b

Hashes for Anaconda3-4.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009404.0
time file was last modified, in human readable format	2016-09-27 15:50:04
exact file size, in bytes	478051940
file size, in human friendly format	455.9 MiB
md5	4692f716c82deb9fa6b59d78f9f6e85c
sha256	73b51715a12b6382dd4df3dd1905b531bd6792d4aa7273b2377a0436d45f0

Hashes for Anaconda3-4.1.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994802.0
time file was last modified, in human readable format	2016-07-08 11:20:02
exact file size, in bytes	425991075
file size, in human friendly format	406.3 MiB
md5	d0dc08d241f83ffc763504db50008e5b
sha256	4f5c95feb0e7efeadd3d348dcef117d7787c799f24b0429e45017008f3534

Hashes for Anaconda3-4.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131311.0
time file was last modified, in human readable format	2016-06-28 11:28:31
exact file size, in bytes	424649707
file size, in human friendly format	405.0 MiB
md5	487d9ba7ae4955e1481ec59de40e51c5
sha256	11d32cf4026603d3b327dc4299863be6b815905ff51a80329085e1bb9f96c

Hashes for Anaconda3-4.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268102.0
time file was last modified, in human readable format	2016-03-29 11:15:02
exact file size, in bytes	417798602
file size, in human friendly format	398.4 MiB
md5	546d1f02597587c685fa890c1d713b51
sha256	36a558a1109868661a5735f5f32607643f6dc05cf581fefb1c10fb8abbe22

Hashes for Anaconda3-2.5.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535687.0
time file was last modified, in human readable format	2016-02-03 15:41:27
exact file size, in bytes	414838933
file size, in human friendly format	395.6 MiB
md5	`02bac549e486be7096070db8d50d0c7f`
sha256	`addadcb927f15cb0b5b6e36890563d3352a8ff6a901ea753d389047d274a2`

Hashes for Anaconda3-2.4.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608453.0
time file was last modified, in human readable format	2015-12-08 15:00:53
exact file size, in bytes	283797156
file size, in human friendly format	270.7 MiB
md5	`45249376f914fdc9fd920ff419a62263`
sha256	`0735e69199fc37135930ea2fd4fb6ad0adef215a2a7ba9fd6b0a0a4daaadb`

Hashes for Anaconda3-2.4.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502942.0
time file was last modified, in human readable format	2015-11-02 16:22:22
exact file size, in bytes	299023674
file size, in human friendly format	285.2 MiB
md5	`48b6d696c73b5f3d573da3300946591d`
sha256	`fb4e480059e991f2fa632b5a9bcdd284c7f0677814cd719c11d524453f96a`

Hashes for Anaconda3-2.3.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775712.0
time file was last modified, in human readable format	2015-07-01 13:35:12
exact file size, in bytes	353018424
file size, in human friendly format	336.7 MiB
md5	`7e10dbd2b620b4aaa360fe90cf5c6790`
sha256	`3be5410b2d9db45882c7de07c554cf4f1034becc274ec9074b23fd37a5c87`

Hashes for Anaconda3-2.2.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314814.0
time file was last modified, in human readable format	2015-03-25 15:20:14
exact file size, in bytes	342778122
file size, in human friendly format	326.9 MiB
md5	`a271fee559b46cf15ba98f21b8549235`
sha256	`4aac68743e7706adb93f042f970373a6e7e087dbf4b02ac467c94ca4ce33d`

Hashes for Anaconda3-2.1.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660220.0
time file was last modified, in human readable format	2014-09-25 10:50:20
exact file size, in bytes	349003566
file size, in human friendly format	332.8 MiB
md5	934cccdf6fa894820d2942ea567dca93
sha256	af3225ccbe8df0ffb918939e009aa57740e35058ebf9dfcf5fec794a77556

Hashes for Anaconda3-2.0.1-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603193.0
time file was last modified, in human readable format	2014-06-12 14:59:53
exact file size, in bytes	319624556
file size, in human friendly format	304.8 MiB
md5	aa7c27b54f710f3004cc17f2db5ff761
sha256	3c3b834793e461f3316ad1d9a9178c67859a9d74aaf7bcade076f04134dd1

Hashes for Anaconda3-2.0.0-Linux-x86_64.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401226019.0
time file was last modified, in human readable format	2014-05-27 16:26:59
exact file size, in bytes	308739435
file size, in human friendly format	294.4 MiB
md5	`c9af4bee8d2da4d74de0d02400ac1c10`
sha256	`57ce4f97e300cf94c5724f72d992e9eecef708fdaa13bc672ae9779773056`

Anaconda with Python 2 on 32-bit Linux

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda2-5.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703530.1317601
time file was last modified, in human readable format	2018-05-30 13:05:30
exact file size, in bytes	512451078
file size, in human friendly format	488.7 MiB
md5	`758e172a824f467ea6b55d3d076c132f`
sha256	`402758c24767e9eb3b77312c388725a058f76e03316464797c3ca404e6eeb`

Hashes for Anaconda2-5.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707331.3896265
time file was last modified, in human readable format	2018-02-15 09:08:51
exact file size, in bytes	452219193
file size, in human friendly format	431.3 MiB
md5	e26fb9d3e53049f6e32212270af6b987
sha256	5af0c7a09a5f3aaf3666c0b362246d342d80e782128ef043998c9ead5ad41

Hashes for Anaconda2-5.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508865187.3479636
time file was last modified, in human readable format	2017-10-24 12:13:07
exact file size, in bytes	433272941
file size, in human friendly format	413.2 MiB
md5	ae155b192027e23189d723a897782fa3
sha256	88c8d698fff16af15862daca10e94a0a46380dcffda45f8d89f5fe03f6bd2

Hashes for Anaconda2-5.0.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506959413.5703459
time file was last modified, in human readable format	2017-10-02 10:50:13
exact file size, in bytes	431941593
file size, in human friendly format	411.9 MiB
md5	5d4b38769f4ee4c33b5cffaa8603356e
sha256	00fbd979c815ede0bbad48fb4ef62cda333c7ad6330184962862a30724792

Hashes for Anaconda2-5.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506455282.2088542
time file was last modified, in human readable format	2017-09-26 14:48:02
exact file size, in bytes	431409105
file size, in human friendly format	411.4 MiB
md5	`a574e495c157d59bf4ec337fa4f72ddd`
sha256	`a3ed8769d20d55a41c04cf7c04e81c95974ea8eb614afab7bbc0c06fa6a52`

Hashes for Anaconda2-4.4.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841010.0
time file was last modified, in human readable format	2017-05-26 18:23:30
exact file size, in bytes	435148039
file size, in human friendly format	415.0 MiB
md5	`b0f8f5ade832b0238357c2f973338b17`
sha256	`452aa91ac83d3b6a68b79cea3042170ec591d468d6966307ff9af18fdbce9`

Hashes for Anaconda2-4.3.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838351.0
time file was last modified, in human readable format	2017-03-06 16:12:31
exact file size, in bytes	406525905
file size, in human friendly format	387.7 MiB
md5	aae1a3192abee1f0abba6c0e1b292cec
sha256	4519ac724d5120d21bb80289c5509c0d1fd9f99c6e9b9a4c6fb352d8bda4a

Hashes for Anaconda2-4.3.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548055.0
time file was last modified, in human readable format	2017-01-27 14:14:15
exact file size, in bytes	405573575
file size, in human friendly format	386.8 MiB
md5	65546028c4a48f4bb582c4ee3e43b893
sha256	b80d471839e8cf7b100e59308720cc13c141deb1ba903a4776c9a05f613e5

Hashes for Anaconda2-4.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009420.0
time file was last modified, in human readable format	2016-09-27 15:50:20
exact file size, in bytes	382758938
file size, in human friendly format	365.0 MiB
md5	e26582ebdf1d982e18efb2bdf52c5ee6
sha256	618b720f309fe8da4f235415f11b6ce3db0a16d702ca67fdceeecf6bec78c

Hashes for Anaconda2-4.1.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994797.0
time file was last modified, in human readable format	2016-07-08 11:19:57
exact file size, in bytes	340385173
file size, in human friendly format	324.6 MiB
md5	8813071788e08e236a323b5f7d337759
sha256	1ab001c7a469345a90d549ebf4afa3376f0f3a57a0df5f042cac7d773b0e0

Hashes for Anaconda2-4.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131308.0
time file was last modified, in human readable format	2016-06-28 11:28:28
exact file size, in bytes	340190685
file size, in human friendly format	324.4 MiB
md5	96e842ef2d5789411c550b0f9bce2314
sha256	54c06cd1b11cb687db6ba3613df443c057f769cdb87693e11674d956d8e5c

Hashes for Anaconda2-4.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268097.0
time file was last modified, in human readable format	2016-03-29 11:14:57
exact file size, in bytes	348392297
file size, in human friendly format	332.3 MiB
md5	f87d5a014499bd9a579ada3939eb22b1
sha256	41341c840cea4185ef5bd82520c1de72b42e7dc43c703fb13b032f04dc0e3

Hashes for Anaconda2-2.5.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535714.0
time file was last modified, in human readable format	2016-02-03 15:41:54
exact file size, in bytes	346405513
file size, in human friendly format	330.4 MiB
md5	aefe284ae4b870ca252da9e46c5d46c9
sha256	4911047df51c46661f551d6022aee21a7e5d31df051d3433b8ff3ea3c2e77

Hashes for Anaconda2-2.4.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608450.0
time file was last modified, in human readable format	2015-12-08 15:00:50
exact file size, in bytes	260583576
file size, in human friendly format	248.5 MiB
md5	2e6983f8fdd5f07025f3a81587c82549
sha256	2388cc714567afe7697bf43b4063ff0ea2150a71b9beb17f75bc7e4879d9b

Hashes for Anaconda2-2.4.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502939.0
time file was last modified, in human readable format	2015-11-02 16:22:19
exact file size, in bytes	293453735
file size, in human friendly format	279.9 MiB
md5	`3fc53407f4a14fe18974d6fb59fc4d3e`
sha256	`478a8fdde3a6e4040a68c57d7bdd6fab1a4f7f6e813948d46dad54867014c`

Hashes for Anaconda-2.3.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775709.0
time file was last modified, in human readable format	2015-07-01 13:35:09
exact file size, in bytes	324643869
file size, in human friendly format	309.6 MiB
md5	`f2459d60a668eb82ff590f97755d93e0`
sha256	`73fdbbb3e38207ed18e5059f71676d18d48fdccbc455a1272eb45a60376cc`

Hashes for Anaconda-2.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314794.0
time file was last modified, in human readable format	2015-03-25 15:19:54
exact file size, in bytes	317885231
file size, in human friendly format	303.2 MiB
md5	e3c3a2dae51a41c5a1cbb959ef68ef2c
sha256	6437d5b08a19c3501f2f5dc3ae1ae16f91adf6bed0f067ef0806a9911b1be

Hashes for Anaconda-2.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660230.0
time file was last modified, in human readable format	2014-09-25 10:50:30
exact file size, in bytes	336767704
file size, in human friendly format	321.2 MiB
md5	3289883a21fdd9fe4fb84748bff677bf
sha256	fd70c08719e6b5caae45b7c8402c6975a8cbc0e3e2a9c4c977554d1784f28

Hashes for Anaconda-2.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603361.0
time file was last modified, in human readable format	2014-06-12 15:02:41
exact file size, in bytes	324151704
file size, in human friendly format	309.1 MiB
md5	0f680aa6dc7150d15123e5490e46eaad
sha256	e8ffc63f31673b5ce41a95796a1f729ddcf4c7db19d6dbe29bedaeaaf8478

Hashes for Anaconda-2.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401313836.0
time file was last modified, in human readable format	2014-05-28 16:50:36
exact file size, in bytes	312856983
file size, in human friendly format	298.4 MiB
md5	`48b6773dacf45e4df0da91cfc149bb23`
sha256	`efb9d3987134d484d88a9d915437b1bd568d065b4fefbd538e0281694bd90`

Hashes for Anaconda-1.9.2-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1396996389.0
time file was last modified, in human readable format	2014-04-08 17:33:09
exact file size, in bytes	431825310
file size, in human friendly format	411.8 MiB
md5	`c8f72746dd5dc68f014d5fccd1f060e8`
sha256	`1f7c850d0b98c011a717b3b757d82077accf0704dd7627f6962267bfb4476`

Hashes for Anaconda-1.9.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392924896.0
time file was last modified, in human readable format	2014-02-20 13:34:56
exact file size, in bytes	431763871
file size, in human friendly format	411.8 MiB
md5	`f1505963a1c7d2bfe7a73c079b22762d`
sha256	`9aa39c05f723fee18c54a9cc1729986193216affedbae125ca5faa0674030`

Hashes for Anaconda-1.9.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1392049410.0
time file was last modified, in human readable format	2014-02-10 10:23:30
exact file size, in bytes	571806071
file size, in human friendly format	545.3 MiB
md5	`11af2251aece5fc4333822dc25f78938`
sha256	`16471e90b3deb7be1b3d449d8883983d81f035dfaa1a3391497de20577de6`

Hashes for Anaconda-1.8.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1383601049.0
time file was last modified, in human readable format	2013-11-04 15:37:29
exact file size, in bytes	412040120
file size, in human friendly format	393.0 MiB
md5	`5028bf0aa7ff8a071d5532b8f8ec924c`
sha256	`2c08a5cd6ccaa9dc84063b0ee9b007aa82e35a75c340fb272b394896de853`

Hashes for Anaconda-1.7.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1378677746.0
time file was last modified, in human readable format	2013-09-08 17:02:26
exact file size, in bytes	399536902
file size, in human friendly format	381.0 MiB
md5	`bbde22bd0346ad9c8932b4d98c0f4000`
sha256	`af372a27a1887e11061485e2a854c535775fd519713e028c38901f90c869c`

Hashes for Anaconda-1.6.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1372784347.0
time file was last modified, in human readable format	2013-07-02 11:59:07
exact file size, in bytes	259053521
file size, in human friendly format	247.1 MiB
md5	`06412ae8de02c87b8de7d7e6d35ed092`
sha256	`745b9452fd18720deefb465a6687c0d66df8f11edceadcee758082dea1b8e`

Hashes for Anaconda-1.6.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1371842619.0
time file was last modified, in human readable format	2013-06-21 14:23:39
exact file size, in bytes	253329362
file size, in human friendly format	241.6 MiB
md5	`7a7f1f53684d38a7aa36935e34af30a3`
sha256	`d6aeedfcb39d648fdfb5bd72c4d0b3063a9d4f4866baf5052aa0645bf5d2c`

Hashes for Anaconda-1.5.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1368022723.0
time file was last modified, in human readable format	2013-05-08 09:18:43
exact file size, in bytes	250369347
file size, in human friendly format	238.8 MiB
md5	`2a75cab6536838635fd38ee7fd3e2411`
sha256	`ca7e356dc1b8c8ef27dfb74b32c77563df704c6ddb39e69cac65ec416ebfe`

Hashes for Anaconda-1.4.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1362869213.0
time file was last modified, in human readable format	2013-03-09 16:46:53
exact file size, in bytes	231260549
file size, in human friendly format	220.5 MiB
md5	`d5826bb10bb25d2f03639f841ef2f65f`
sha256	`065284c5de369c9b89dcae79e7169ce9b734dc3bbe6c409a67a5ec6480cc0`

Anaconda with Python 3 on 32-bit Linux

To verify the file integrity using MD5 or SHA-256, see cryptographic hash verification.

Hashes for Anaconda3-5.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1527703546.172479
time file was last modified, in human readable format	2018-05-30 13:05:46
exact file size, in bytes	531957909
file size, in human friendly format	507.3 MiB
md5	`81d5a1648e3aca4843f88ca3769c0830`
sha256	`f3527d085d06f35b6aeb96be2a9253ff9ec9ced3dc913c8e27e086329f3db`

Hashes for Anaconda3-5.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1518707338.3619494
time file was last modified, in human readable format	2018-02-15 09:08:58
exact file size, in bytes	471561932
file size, in human friendly format	449.7 MiB
md5	`793a94ee85baf64d0ebb67a0c49af4d7`
sha256	`0e940272517d8f8a6f26316a19e4be2bdaea8477a3a32cc2ecee7b48fd0fa`

Hashes for Anaconda3-5.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1508800071.267042
time file was last modified, in human readable format	2017-10-23 18:07:51
exact file size, in bytes	451929576
file size, in human friendly format	431.0 MiB
md5	`d967f023a23698109fe213103a2c07bf`
sha256	`991a4b656fcb0236864fbb27ff03bb7f3d98579205829b76b66f65cfa6734`

Hashes for Anaconda3-5.0.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506959415.21842
time file was last modified, in human readable format	2017-10-02 10:50:15
exact file size, in bytes	450639191
file size, in human friendly format	429.8 MiB
md5	`8b6902d20063e6c3b98ebe70060f3131`
sha256	`407576899d3aa546bc3c2c4a13cbc18ab5bab372c3388ea80087f29b32184`

Hashes for Anaconda3-5.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1506455282.9808888
time file was last modified, in human readable format	2017-09-26 14:48:02
exact file size, in bytes	450106703
file size, in human friendly format	429.3 MiB
md5	`8120fcd072916e4a28d0179be8d29053`
sha256	`634d2dfa97d19f2cc15e941cb4d059bc83a31facedfb9d02a980c4fa74f27`

Hashes for Anaconda3-4.4.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1495841025.0
time file was last modified, in human readable format	2017-05-26 18:23:45
exact file size, in bytes	449473324
file size, in human friendly format	428.7 MiB
md5	`8556e85f81206c08ee2a30b67d1bb707`
sha256	`b0e492206d43067314b25963bc7d1f012096ca0323b7629f4ebcd071b0390`

Hashes for Anaconda3-4.3.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1488838367.0
time file was last modified, in human readable format	2017-03-06 16:12:47
exact file size, in bytes	418659792
file size, in human friendly format	399.3 MiB
md5	`d8986b1503f3b42220be9bfb8a92100e`
sha256	`7b70bdba282a18ddbdc167afe8131f7532076cb1df8d3fbbd13e79ca3afaa`

Hashes for Anaconda3-4.3.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1485548069.0
time file was last modified, in human readable format	2017-01-27 14:14:29
exact file size, in bytes	417717702
file size, in human friendly format	398.4 MiB
md5	`3f173aa1ab2c2b6ab3f8a6bd22827fd7`
sha256	`f7ce2eeec3e42c2ba1ee3b9fcd670478fd30f4be547c6e0a675d183c4ca9`

Hashes for Anaconda3-4.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1475009434.0
time file was last modified, in human readable format	2016-09-27 15:50:34
exact file size, in bytes	392066694
file size, in human friendly format	373.9 MiB
md5	`7aca10e1ea5b9db0a318b4eed5253747`
sha256	`1a8320635f2f06ec9d8610e77d6d0f9cb2c5d11d20a4ff7fcda113e04b0a`

Hashes for Anaconda3-4.1.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467994802.0
time file was last modified, in human readable format	2016-07-08 11:20:02
exact file size, in bytes	345064389
file size, in human friendly format	329.1 MiB
md5	0576a0df8987ca62d5c13491102547d9
sha256	931626363f4030c7a1e8897549b1d3589dc3f429874dc3dd8a79869ecf5c8

Hashes for Anaconda3-4.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1467131312.0
time file was last modified, in human readable format	2016-06-28 11:28:32
exact file size, in bytes	344388621
file size, in human friendly format	328.4 MiB
md5	302fddc310233f5e6f120753ec3e392d
sha256	7764093f337a43e4962b12d01508c1a385f0f62c1ddc006b69af95ae763fc

Hashes for Anaconda3-4.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1459268103.0
time file was last modified, in human readable format	2016-03-29 11:15:03
exact file size, in bytes	353266156
file size, in human friendly format	336.9 MiB
md5	c88cbe27cc8fb4976e6bd38068cc57d6
sha256	e1469fa0d24de12f33661ce3d7a06d77968be8822f366a61a0018a3850ab5

Hashes for Anaconda3-2.5.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1454535727.0
time file was last modified, in human readable format	2016-02-03 15:42:07
exact file size, in bytes	350634167
file size, in human friendly format	334.4 MiB
md5	e1d4e9480b44ea0905cbf39846778f8b
sha256	22ac26c8bde7c4153ea859f6f6d8aca93bbf1e213d800167ad5ea530c6295

Hashes for Anaconda3-2.4.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1449608453.0
time file was last modified, in human readable format	2015-12-08 15:00:53
exact file size, in bytes	265518790
file size, in human friendly format	253.2 MiB
md5	82f1f438ac83ed8b7d36284995f6939b
sha256	00d13413f5b8129e863dabcc2296a181c697056c5ed210739a0aa06454ab7

Hashes for Anaconda3-2.4.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or "checksums") below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1446502942.0
time file was last modified, in human readable format	2015-11-02 16:22:22
exact file size, in bytes	290842015
file size, in human friendly format	277.4 MiB
md5	`423f0300cbec30c206a6c61f7e5dc9bd`
sha256	`f6080c6493cefc603cfeb67aaf6c3c4c6b80a66788f03db48ffd3cfa52017`

Hashes for Anaconda3-2.3.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1435775713.0
time file was last modified, in human readable format	2015-07-01 13:35:13
exact file size, in bytes	338272927
file size, in human friendly format	322.6 MiB
md5	`72b14bfd85f2597089c4372225a96d42`
sha256	`4cc10d65c303191004ada2b6d75562c8ed84e42bf9871af06440dd956077b`

Hashes for Anaconda3-2.2.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1427314807.0
time file was last modified, in human readable format	2015-03-25 15:20:07
exact file size, in bytes	328483185
file size, in human friendly format	313.3 MiB
md5	`fe3681d49ff5b0d755181f553689ed9e`
sha256	`223655cd256aa912dfc83ab24570e47bb3808bc3b0c6bd21b5db0fcf27508`

Hashes for Anaconda3-2.1.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1411660235.0
time file was last modified, in human readable format	2014-09-25 10:50:35
exact file size, in bytes	333141910
file size, in human friendly format	317.7 MiB
md5	`462665c149b14f7c3993bc51e4d10f88`
sha256	`657cb599004c21e37ce693515ea33922e0084fd7c159ef1b96b57c86eed83`

Hashes for Anaconda3-2.0.1-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1402603200.0
time file was last modified, in human readable format	2014-06-12 15:00:00
exact file size, in bytes	301673940
file size, in human friendly format	287.7 MiB
md5	`86213516c4c2e479d8d9834b89c95bed`
sha256	`21293fabbd3d5cfbb1afe0c9a8b39e0bc4d283cd7dbe3c84a60b335481a41`

Hashes for Anaconda3-2.0.0-Linux-x86.sh

All installer files are available at https://repo.continuum.io/archive/ .

You can verify the data integrity of the Anaconda installer files by running a local program to generate their MD5 or SHA-256 cryptographic hashes and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please contact us to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1401226555.0
time file was last modified, in human readable format	2014-05-27 16:35:55
exact file size, in bytes	290973139
file size, in human friendly format	277.5 MiB
md5	860a11c39e58bb574bad5be9d44e2063
sha256	439761159d5604e182951650a478dd53caff52e9dccf17c20ae66689b7b28

Updating from older versions

You can easily update Anaconda to the latest version.

- Windows: Open the Start Menu and choose Anaconda Prompt.
- macOS or Linux: Open a terminal window.

Enter these commands:

```
conda update conda
conda update anaconda
```

Uninstalling Anaconda

To uninstall Anaconda, you can do a simple remove of the program. This will leave a few files behind, which for most users is just fine. See Option A.

If you also want to remove all traces of the configuration files and directories from Anaconda and its programs, you can download and use the Anaconda-Clean program first, then do a simple remove. See Option B.

1. Option A. Use simple remove to uninstall Anaconda:
    - Windows–In the Control Panel, choose Add or Remove Programs or Uninstall a program, and then select Python 3.6 (Anaconda) or your version of Python.
    - macOS–Open the Terminal.app or iTerm2 terminal application, and then remove your entire Anaconda directory, which has a name such as `anaconda2` or `anaconda3`, by entering `rm -rf ~/anaconda3`.
    - Linux–Open a terminal window, and then remove your entire Anaconda directory, which has a name such as `anaconda2` or `anaconda3`, by entering `rm -rf ~/anaconda3`.
2. Option B. Full uninstall using Anaconda-Clean and simple remove.

    NOTE: Anaconda-Clean must be run before simple remove.

    - Install the Anaconda-Clean package from Anaconda Prompt (Terminal on Linux or macOS):

        ```
        conda install anaconda-clean
        ```

    - In the same window, run one of these commands:
        - Remove all Anaconda-related files and directories with a confirmation prompt before deleting each one:

            ```
            anaconda-clean
            ```

        - Or, remove all Anaconda-related files and directories without being prompted to delete each one:

```
anaconda-clean --yes
```

Anaconda-Clean creates a backup of all files and directories that might be removed, such as .bash_profile, in a folder named .anaconda_backup in your home directory. Also note that Anaconda-Clean leaves your data files in the AnacondaProjects directory untouched.

- After using Anaconda-Clean, follow the instructions above in Option A to uninstall Anaconda.

Removing Anaconda path from .bash_profile

If you use Linux or macOS, you may also wish to check the .bash_profile file in your home directory for a line such as:

```
export PATH="/Users/jsmith/anaconda3/bin:$PATH"
```

NOTE: Replace /Users/jsmith/anaconda3/ with your actual path.

This line adds the Anaconda path to the PATH environment variable. It may refer to either Anaconda or Miniconda. After uninstalling Anaconda, you may delete this line and save the file.

General installation information

On Windows, macOS, and Linux, it is best to install Anaconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Anaconda system wide, which does require administrator permissions.

Silent mode install

You can use silent mode to automatically accept default settings and have no screen prompts appear during installation.

Installing Anaconda on a non-networked machine (air gap)

1. Obtain a local copy of the appropriate Anaconda installer for the non-networked machine. You can copy the Anaconda installer to the target machine using many different methods including a portable hard drive, USB drive or CD.
2. After copying the installer to the non-networked machine, follow the detailed installation instructions for your operating system.

1.2.2 Packages

Anaconda package lists

All packages available in the latest release of Anaconda are listed on the pages linked below. These packages may be installed with the command conda install PACKAGENAME and are located in the package repository.

Click the links below to see which packages are available for each version of Python (3.6, 3.5 or 2.7) and each operating system and architecture.

Anaconda is available for 64 and 32 bit Windows, macOS, and 64 and 32 bit Linux on the Intel and AMD x86 and x86-64 CPU architectures, and for 64 bit Linux on the IBM Power CPU architecture.

An RSS feed is updated each time a new package is added to the Anaconda package repository.

To request a package not listed on this page, please create an issue on the Anaconda issues page.

	Python 3.6	Python 3.5	Python 2.7
64-bit Windows	*64-bit Windows, Py3.6*	*64-bit Windows, Py3.5*	*64-bit Windows, Py2.7*
32-bit Windows	*32-bit Windows, Py3.6*	*32-bit Windows, Py3.5*	*32-bit Windows, Py2.7*
macOS	*macOS, Py3.6*	*macOS, Py3.5*	*macOS, Py2.7*
64-bit Linux	*64-bit Linux, Py3.6*	*64-bit Linux, Py3.5*	*64-bit Linux, Py2.7*
32-bit Linux	*32-bit Linux, Py3.6*	*32-bit Linux, Py3.5*	*32-bit Linux, Py2.7*
64-bit Linux on IBM Power CPUs	*Linux on IBM Power, Py3.6*	*Linux on IBM Power, Py3.5*	*Linux on IBM Power, Py2.7*

Packages for 64-bit Windows with Python 3.6

Packages for 64-bit Windows with Python 3.5

Packages for 64-bit Windows with Python 2.7

Packages for 32-bit Windows with Python 3.6

Packages for 32-bit Windows with Python 3.5

Packages for 32-bit Windows with Python 2.7

Packages for macOS with Python 3.6

Packages for macOS with Python 3.5

Packages for macOS with Python 2.7

Packages for 64-bit Linux with Python 3.6

Packages for 64-bit Linux with Python 3.5

Packages for 64-bit Linux with Python 2.7

Packages for 32-bit Linux with Python 3.6

Packages for 32-bit Linux with Python 3.5

Packages for 32-bit Linux with Python 2.7

Packages for 64-bit Linux on IBM Power CPUs with Python 3.6

Packages for 64-bit Linux on IBM Power CPUs with Python 3.5

Packages for 64-bit Linux on IBM Power CPUs with Python 2.7

R language packages for Anaconda

The R language packages available for install from conda are available at http://repo.continuum.io/pkgs/r/ . You can install any of these R language packages into your current environment with the conda command `conda install`

-c r package-name. NOTE: Replace package-name with the name of the package. For example, you can install the package r-acepack with the command conda install -c r r-acepack.

TIP: You can also search for any R package if you know the name, such as conda search -f EXACTNAME. Replace EXACTNAME with the desired R package name.

Many Comprehensive R Archive Network (CRAN) packages have been been rebuilt and made available as conda packages. Anaconda does not provide builds of the entire CRAN repository, so there are some packages in CRAN that are not available as conda packages.

R Essentials bundle

Rather than install each R language package individually, you can get the R Essentials bundle. It includes about 100 of the most popular scientific packages for the R programming language.

You can install the R Essentials bundle with this command:

```
conda install -c r r-essentials
```

More resources

- *Using R language with Anaconda*

List of R packages for Anaconda

Old package lists

You can download previous versions of Anaconda from the Anaconda installer archive.

Older versions of packages can usually be downloaded from the package repository or from https://anaconda.org/anaconda/PackageName.

NOTE: Replace PackageName with the name of the desired package.

EXAMPLE: At https://anaconda.org/anaconda/beautifulsoup4 , previous versions of beautifulsoup4 are shown on the **Files** tab.

You can also search for packages from the command line with conda search PackageName.

NOTE: Replace PackageName with the name of the desired package.

Packages included in previous versions of Anaconda:

Packages included in Anaconda v.1.0

- anaconda launcher
- bitarray 0.8.0
- bitey
- cython 0.16
- dateutil 1.5
- disco 0.4.2 (Linux only)
- erlang (Linux only)
- flask 0.9
- gevent 0.13.7
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet0.4.0
- h5py 2.0.1
- hdf5 1.8.9
- PIL 1.1.7
- iopro 1.0 *
- ipython 0.13
- jinja2 2.6
- llvm 3.1
- llvmpy 0.8.2.dev
- matplotlib 1.1.1
- mpi4py 1.3
- mpich2 1.4.1p1
- networkx 1.7
- nose 1.1.2
- numba 0.1.dev
- numbapro 1.0 *
- numexpr 2.0.1
- numpy 1.7.dev
- opencv 2.4.2
- openssl 1.0.1c
- pandas 0.8.1
- pip 1.1
- pixman 0.26.2
- py2cairo 1.10.0
- pycurl 7.19.0
- pygments 1.5
- pysal 1.4.0
- pysam 0.6
- pytables 2.4.0
- python 2.7.3
- pytz 2012c
- pyyaml 3.10
- pyzmq 2.2.0
- redis 2.4.15 (Linux only)
- redis py-2.4.13
- requests 0.13.5
- scikit-learn 0.11
- scikits-image 0.6.1
- scipy 0.11.0rc2
- sqlalchemy 0.7.8
- sqlite 3.7.13
- statsmodels 0.4.3
- sympy 0.7.1
- theano 0.5.0
- tornado 2.3
- werkzeug 0.8.3
- wiseRF *

*** Included in Anaconda Pro Only**

CONTINUUM®
ANALYTICS

Packages included in Anaconda v.1.1

• anaconda launcher	• iopro 1.0 *	• openssl 1.0.1c	• redis 2.4.15 (Linux only)
• bitarray 0.8.0	• ipython 0.13	• pandas 0.8.1	• redis py-2.4.13
• bitey	• jinja2 2.6	• pip 1.1	• requests 0.13.5
• cython 0.16	• llvm 3.1	• pixman 0.26.2	• scikit-learn 0.11
• dateutil 1.5	• llvmpy 0.8.2.dev	• py2cairo 1.10.0	• scikits-image 0.6.1
• disco 0.4.2 (Linux only)	• matplotlib 1.1.1	• pycurl 7.19.0	• scipy 0.11.0rc2
• erlang (Linux only)	• mpi4py 1.3	• pygments 1.5	• spyder
• flask 0.9	• mpich2 1.4.1p1	• pysal 1.4.0	• sqlalchemy 0.7.8
• gevent 0.13.7	• networkx 1.7	• pysam 0.6	• sqlite 3.7.13
• gevent-websocket 0.3.6	• nose 1.1.2	• pytables 2.4.0	• statsmodels 0.4.3
• gevent_zeromq 0.2.5	• numba 0.1.dev	• python 2.7.3	• sympy 0.7.1
• greenlet0.4.0	• numbapro 1.0 *	• pytz 2012c	• theano 0.5.0
• h5py 2.0.1	• numexpr 2.0.1	• pyyaml 3.10	• tornado 2.3
• hdf5 1.8.9	• numpy 1.7.dev	• pyzmq 2.2.0	• werkzeug 0.8.3
• PIL 1.1.7	• opencv 2.4.2		• wiseRF *

*** Included in Anaconda Pro Only**

Note: Packages may vary on different platforms.

Packages included in Anaconda 1.2.1

- bitarray 0.8.0
- bitey 0.0
- boto 2.6.0
- cairo 1.12.2 *L*
- chaco 4.2.1.dev *M*
- conda 1.2.1
- cython 0.17.1
- dateutil 1.5
- disco 0.4.2 *L*
- distribute 0.6.30
- docutils 0.9.1
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gevent 0.13.7
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.0
- hdf5 1.8.9
- imaging 1.1.7
- iopro 1.2.2 *P*
- ipython 0.13.1
- jinja2 2.6
- jpeg 8d
- libevent 2.0.20
- libpng 1.5.13
- llvm 3.1
- llvmpy 0.9
- matplotlib 1.2.0
- mdp 3.3
- meta 0.4.2.dev
- mingw 4.7 *W*
- mkl 10.3 *LP*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- networkx 1.7
- nltk 2.0.3
- nose 1.1.2
- numba 0.3.2
- numbapro 0.7.3 *P*
- numexpr 2.0.1
- numpy 1.6.2 *W*
- numpy 1.7.0b2 *U*
- opencv 2.4.2 *L*
- pandas 0.9.0
- pip 1.2.1
- psutil 0.6.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pyaudio 0.2.6 *M*
- pycurl 7.19.0
- pyflakes 0.5.0
- pygments 1.5
- pyreadline 1.7.1 *W*
- pysal 1.4.0
- pysam 0.6 *U*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.3
- python 2.7.3
- pytz 2012d
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis 2.4.15 *L*
- redis-py 2.4.13 *L*
- requests 0.13.9
- scikit-learn 0.11
- scikits-image 0.6.1
- scipy 0.11.0
- sphinx 1.1.3
- spyder 2.1.11
- sqlalchemy 0.7.8
- statsmodels 0.4.3
- sympy 0.7.1
- theano 0.5.0 *L*
- tornado 2.3
- werkzeug 0.8.3
- wiserf 1.1 *UP*
- yaml 0.1.4
- zeromq 2.2.0
- zlib 1.2.7

U: Unix - *L*: Linux - *M*: macOS - *W*: Windows - *P*: not in CE

Packages included in Anaconda 1.3.1

- biopython 1.60
- bitarray 0.8.0
- bitey 0.0
- boto 2.7.0
- cairo 1.12.2 *L*
- conda 1.3.5
- cubes 0.10.1
- cython 0.17.4
- dateutil 1.5
- disco 0.4.4 *L*
- distribute 0.6.34
- docutils 0.10
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gdata 2.0.17
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- googlecl 0.9.12
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- iopro 1.3.2 *P*
- ipython 0.13.1
- jinja2 2.6
- libevent 2.0.20
- libnvvm 1.0 *P*
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.10.2
- matplotlib 1.2.0
- mdp 3.3
- meta 0.4.2.dev
- mingw 4.7 *W*
- mkl 10.3 *LP*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- networkx 1.7
- nltk 2.0.4
- nose 1.2.1
- numba 0.6.0
- numbapro 0.8.1 *P*
- numexpr 2.0.1
- numpy 1.6.2 *W*
- numpy 1.7.0rc1 *U*
- opencv 2.4.2 *L*
- pandas 0.10.1
- pip 1.2.1
- ply 3.4
- psutil 0.6.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycrypto 2.6
- pycurl 7.19.0
- pyflakes 0.5.0
- pygments 1.5
- pyparsing 1.5.6
- pysal 1.4.0
- pysam 0.6 *U*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.4
- python 2.7.3
- pytz 2012d
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis 2.6.9 *U*
- redis-py 2.7.2 *U*
- requests 0.13.9
- scikit-learn 0.13
- scikits-image 0.7.1
- scipy 0.11.0
- sphinx 1.1.3
- spyder 2.1.13
- sqlalchemy 0.7.8
- statsmodels 0.4.3
- sympy 0.7.1
- theano 0.5.0 *L*
- tornado 2.4.1
- werkzeug 0.8.3
- wiserf 1.1 *UP*
- zeromq 2.2.0
- zlib 1.2.7

U: Unix - *L*: Linux - *M*: macOS - *W*: Windows - *P*: not in CE

Packages included in Anaconda 1.4.0

Python 2.7 (what is included in the installers):

- astropy 0.2
- biopython 1.60
- bitarray 0.8.0
- bitey 0.0
- boto 2.8.0
- cairo 1.12.2 *L*
- conda 1.4.4
- cubes 0.10.2
- cython 0.18
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.34
- docutils 0.10
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gdata 2.0.17
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- googlecl 0.9.12
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.1
- jinja2 2.6
- libevent 2.0.20
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.1
- lxml 3.0.2
- matplotlib 1.2.0
- mdp 3.3
- menuinst 1.0.0 *W*
- meta 0.4.2.dev
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- networkx 1.7
- nltk 2.0.4
- nose 1.2.1
- numba 0.7.0
- numexpr 2.0.1
- numpy 1.7.0
- opencv 2.4.2 *L*
- pandas 0.10.1
- pip 1.2.1
- ply 3.4
- psutil 0.6.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.6.1
- pygments 1.6
- pyparsing 1.5.6
- pysal 1.5.0
- pysam 0.6 *LM*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.4
- python 2.7.3
- pytz 2012j
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis 2.6.9 *LM*
- redis-py 2.7.2 *LM*
- requests 0.13.9
- scikit-image 0.8.2
- scikit-learn 0.13
- scipy 0.11.0
- six 1.2.0
- sphinx 1.1.3
- spyder 2.1.13
- sqlalchemy 0.7.8
- statsmodels 0.4.3
- sympy 0.7.1
- theano 0.5.0 *L*
- tornado 2.4.1
- werkzeug 0.8.3
- xlrd 0.9.0
- xlwt 0.7.4
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- astropy 0.2
- bitarray 0.8.0
- cython 0.18
- dateutil 2.1
- distribute 0.6.34
- docutils 0.10
- freetype 2.4.10
- greenlet 0.4.0
- ipython 0.13.1
- jinja2 2.6
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.1
- lxml 3.0.2
- matplotlib 1.2.0
- mdp 3.3
- mingw 4.7 *W*
- networkx 1.7
- nose 1.2.1
- numpy 1.7.0
- pandas 0.10.1 *LM*
- pip 1.2.1
- ply 3.4
- psutil 0.6.1
- pycparser 2.9.1
- pycrypto 2.6
- pyflakes 0.6.1
- pygments 1.6
- python 3.3.0
- pytz 2012j
- pyyaml 3.10
- pyzmq 2.2.0.1
- requests 0.13.9
- scikit-image 0.8.2 *LM*
- scipy 0.11.0 *LM*
- six 1.2.0
- sphinx 1.1.3
- sqlalchemy 0.7.8
- tornado 2.4.1
- xlrd 0.9.0
- zlib 1.2.7

Python 2.6 (available through conda):

- argparse 1.2.1
- astropy 0.2
- biopython 1.60
- bitarray 0.8.0
- boto 2.8.0
- cairo 1.12.2 *L*
- cython 0.18
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.34
- docutils 0.10
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gdata 2.0.17
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- googlecl 0.9.12
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.1
- jinja2 2.6
- libevent 2.0.20
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.1
- lxml 3.0.2
- matplotlib 1.2.0 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- networkx 1.7
- nltk 2.0.4
- nose 1.2.1
- numba 0.7.0
- numexpr 2.0.1
- numpy 1.7.0
- opencv 2.4.2 *L*
- pandas 0.10.1
- pip 1.2.1
- ply 3.4
- psutil 0.6.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.6.1
- pygments 1.6
- pyparsing 1.5.6
- pysam 0.6 *LM*
- pytables 2.4.0 *LM*
- pytest 2.3.4
- python 2.6.8
- pytz 2012j
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.7.2 *LM*
- requests 0.13.9
- scikit-image 0.8.2
- scikit-learn 0.13
- scipy 0.11.0
- six 1.2.0
- sphinx 1.1.3
- sqlalchemy 0.7.8
- statsmodels 0.4.3
- sympy 0.7.1 *LM*
- theano 0.5.0 *L*
- tornado 2.4.1
- werkzeug 0.8.3
- xlrd 0.9.0
- xlwt 0.7.4
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.5.0

- astropy 0.2.1
- atom 0.2.3
- biopython 1.61
- bitarray 0.8.1
- boto 2.9.2
- cairo 1.12.2 *L*
- casuarius 1.1
- conda 1.5.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.19
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.36
- docutils 0.10
- dynd-python 0.3.0
- enaml 0.7.6
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.2
- jinja2 2.6
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.2
- lxml 3.2.0
- matplotlib 1.2.1
- mdp 3.3
- menuinst 1.0.0 *W*
- meta 0.4.2.dev
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nltk 2.0.4
- nose 1.3.0
- numba 0.8.1
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.2 *L*
- pandas 0.11.0
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.7.2
- pygments 1.6
- pyparsing 1.5.6
- pysal 1.5.0
- pysam 0.6 *LM*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.4
- python 2.7.4
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis-py 2.7.2 *LM*
- requests 1.2.0
- scikit-image 0.8.2
- scikit-learn 0.13.1
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- spyder 2.2.0
- sqlalchemy 0.8.1
- statsmodels 0.4.3
- sympy 0.7.2
- theano 0.5.0 *L*
- tornado 3.0.1
- werkzeug 0.8.3
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- astropy 0.2.1
- bitarray 0.8.1
- curl 7.30.0 *LM*
- cython 0.19
- dateutil 2.1
- distribute 0.6.36
- docutils 0.10
- dynd-python 0.3.0
- freetype 2.4.10
- greenlet 0.4.0
- hdf5 1.8.9
- ipython 0.13.2
- jinja2 2.6
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.2
- lxml 3.2.0
- matplotlib 1.2.1
- mdp 3.3
- mingw 4.7 *W*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nose 1.3.0
- numba 0.8.1
- numpy 1.7.1
- pandas 0.11.0
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pyflakes 0.7.2
- pygments 1.6
- pyside 1.1.2 *W*
- python 3.3.1
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- requests 1.2.0
- scikit-image 0.8.2
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- sqlalchemy 0.8.1
- sympy 0.7.2
- tornado 3.0.1
- xlrd 0.9.2
- zlib 1.2.7

Python 2.6 (available through conda):

- argparse 1.2.1
- astropy 0.2.1
- atom 0.2.3
- biopython 1.61
- bitarray 0.8.1
- boto 2.9.2
- cairo 1.12.2 *L*
- casuarius 1.1
- curl 7.30.0 *LM*
- cython 0.19
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.36
- docutils 0.10
- dynd-python 0.3.0
- erlang R15B01 *L*
- flask 0.9
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.0
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.2
- jinja2 2.6
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmpy 0.11.2
- lxml 3.2.0
- matplotlib 1.2.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nltk 2.0.4
- nose 1.3.0
- numba 0.8.1
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.2 *L*
- ordereddict 1.1
- pandas 0.11.0
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- py 1.4.12
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.7.2
- pygments 1.6
- pyparsing 1.5.6
- pysam 0.6 *LM*
- pytables 2.4.0 *LM*
- pytest 2.3.4
- python 2.6.8
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis-py 2.7.2 *LM*
- requests 1.2.0
- scikit-image 0.8.2
- scikit-learn 0.13.1
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- sqlalchemy 0.8.1
- statsmodels 0.4.3
- sympy 0.7.2 *LM*
- theano 0.5.0 *L*
- tornado 3.0.1
- werkzeug 0.8.3
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.6.1

- astropy 0.2.3
- atom 0.2.3
- binstar 0.1.2
- biopython 1.61
- bitarray 0.8.1
- boto 2.9.6
- cairo 1.12.2 *L*
- casuarius 1.1
- conda 1.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.10
- dynd-python 0.4.0
- enaml 0.7.6
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.2
- itsdangerous 0.21
- jinja2 2.6
- keyring 1.4
- launcher 0.1.2
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmmath 0.1.0
- llvmpy 0.11.3
- lxml 3.2.1
- matplotlib 1.2.1
- mdp 3.3
- menuinst 1.0.1 *W*
- meta 0.4.2.dev
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nltk 2.0.4
- nose 1.3.0
- numba 0.9.0
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.2 *L*
- pandas 0.11.0
- pep8 1.4.5
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- py 1.4.14
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.7.2
- pygments 1.6
- pyparsing 1.5.6
- pysal 1.5.0
- pysam 0.6 *LM*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.5
- python 2.7.5
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis-py 2.7.2 *LM*
- requests 1.2.3
- rope 0.9.4
- scikit-image 0.8.2
- scikit-learn 0.13.1
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- spyder 2.2.0
- sqlalchemy 0.8.1
- statsmodels 0.4.3
- sympy 0.7.2
- theano 0.5.0 *L*
- tornado 3.1
- werkzeug 0.9.1
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- astropy 0.2.3
- bitarray 0.8.1
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- distribute 0.6.45
- docutils 0.10
- dynd-python 0.4.0
- freetype 2.4.10
- greenlet 0.4.1
- hdf5 1.8.9
- ipython 0.13.2
- jinja2 2.6
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmmath 0.1.0
- llvmpy 0.11.3
- lxml 3.2.1
- matplotlib 1.2.1
- mdp 3.3
- mingw 4.7 *W*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nose 1.3.0
- numba 0.9.0
- numpy 1.7.1
- pandas 0.11.0
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pyflakes 0.7.2
- pygments 1.6
- pyside 1.1.2 *W*
- python 3.3.2
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- requests 1.2.3
- scikit-image 0.8.2
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- sqlalchemy 0.8.1
- sympy 0.7.2
- tornado 3.1
- xlrd 0.9.2
- zlib 1.2.7

Python 2.6 (available through conda):

- argparse 1.2.1
- astropy 0.2.3
- atom 0.2.3
- biopython 1.61
- bitarray 0.8.1
- boto 2.9.6
- cairo 1.12.2 *L*
- casuarius 1.1
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.10
- dynd-python 0.4.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.1.1
- hdf5 1.8.9
- imaging 1.1.7
- ipython 0.13.2
- itsdangerous 0.21
- jinja2 2.6
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13
- llvm 3.2
- llvmmath 0.1.0
- llvmpy 0.11.3
- lxml 3.2.1
- matplotlib 1.2.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.4 *LM*
- networkx 1.7
- nltk 2.0.4
- nose 1.3.0
- numba 0.9.0
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.2 *L*
- ordereddict 1.1
- pandas 0.11.0
- pep8 1.4.5
- pip 1.3.1
- ply 3.4
- psutil 0.7.1
- py 1.4.14
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.7.2
- pygments 1.6
- pyparsing 1.5.6
- pysam 0.6 *LM*
- pytables 2.4.0 *LM*
- pytest 2.3.5
- python 2.6.8
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis-py 2.7.2 *LM*
- requests 1.2.3
- scikit-image 0.8.2
- scikit-learn 0.13.1
- scipy 0.12.0
- six 1.3.0
- sphinx 1.1.3
- sqlalchemy 0.8.1
- statsmodels 0.4.3
- sympy 0.7.2 *LM*
- theano 0.5.0 *L*
- tornado 3.1
- werkzeug 0.9.1
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.7.0

- apptools 4.2.0
- astropy 0.2.4
- atom 0.3.2
- binstar 0.3.1
- biopython 1.61
- bitarray 0.8.1
- bokeh 0.1.1
- boto 2.12.0
- cairo 1.12.2 *L*
- casuarius 1.1
- chaco 4.3.0
- configobj 4.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.4.2
- enable 4.3.0
- enaml 0.7.19
- envisage 4.3.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.2.0
- hdf5 1.8.9
- imaging 1.1.7
- ipython 1.0.0
- itsdangerous 0.23
- jinja2 2.7.1
- keyring 3.0.1
- launcher 0.1.2
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.0
- mayavi 4.3.0
- mdp 3.3
- menuinst 1.0.1 *W*
- meta 0.4.2.dev
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.5 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.10.2
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.6 *L*
- pandas 0.12.0
- patsy 0.2.1
- pep8 1.4.6
- pip 1.4.1
- ply 3.4
- psutil 1.0.1
- py 1.4.14
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyface 4.3.0
- pyflakes 0.7.3
- pygments 1.6
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pysal 1.6.0
- pysam 0.6 *LM*
- pyside 1.1.2
- pytables 2.4.0
- pytest 2.3.5
- python 2.7.5
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.7.4
- redis-py 2.7.2 *LM*
- requests 1.2.3
- rope 0.9.4
- scikit-image 0.8.2
- scikit-learn 0.14.1
- scipy 0.12.0
- six 1.4.1
- sphinx 1.1.3
- spyder 2.2.4
- sqlalchemy 0.8.2
- statsmodels 0.5.0
- sympy 0.7.3
- theano 0.5.0 *L*
- tornado 3.1.1
- traits 4.3.0
- traitsui 4.3.0
- vtk 5.10.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- astropy 0.2.4
- bitarray 0.8.1
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.4.2
- flask 0.10.1
- freetype 2.4.10
- greenlet 0.4.1
- hdf5 1.8.9
- ipython 1.0.0
- itsdangerous 0.23
- jinja2 2.7.1
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.0
- mdp 3.3
- mingw 4.7 *W*
- netcdf4 1.0.5 *LM*
- networkx 1.8.1
- nose 1.3.0
- numba 0.10.2
- numpy 1.7.1
- pandas 0.12.0
- patsy 0.2.1
- pillow 2.1.0
- pip 1.4.1
- ply 3.4
- psutil 1.0.1
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pyflakes 0.7.3
- pygments 1.6
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pyside 1.1.2 *W*
- python 3.3.2
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- requests 1.2.3
- scikit-image 0.8.2
- scipy 0.12.0
- six 1.4.1
- sphinx 1.1.3
- sqlalchemy 0.8.2
- statsmodels 0.5.0
- sympy 0.7.3
- tornado 3.1.1
- werkzeug 0.9.4
- xlrd 0.9.2
- zlib 1.2.7

Python 2.6 (available through conda):

- argparse 1.2.1
- astropy 0.2.4
- atom 0.3.2
- biopython 1.61
- bitarray 0.8.1
- boto 2.12.0
- cairo 1.12.2 *L*
- casuarius 1.1
- configobj 4.7.2
- curl 7.30.0 *LM*
- cython 0.19.1
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.4.2
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.2.0
- hdf5 1.8.9
- imaging 1.1.7
- ipython 1.0.0
- itsdangerous 0.23
- jinja2 2.7.1
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.0 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.5 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.10.2
- numexpr 2.0.1
- numpy 1.7.1
- opencv 2.4.6 *L*
- ordereddict 1.1
- pandas 0.12.0
- patsy 0.2.1
- pep8 1.4.6
- pip 1.4.1
- ply 3.4
- psutil 1.0.1
- py 1.4.14
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6
- pycurl 7.19.0 *LM*
- pyflakes 0.7.3
- pygments 1.6
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pysam 0.6 *LM*
- pytables 2.4.0 *LM*
- pytest 2.3.5
- python 2.6.8
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis-py 2.7.2 *LM*
- requests 1.2.3
- scikit-image 0.8.2
- scikit-learn 0.14.1
- scipy 0.12.0
- six 1.4.1
- sphinx 1.1.3
- sqlalchemy 0.8.2
- statsmodels 0.5.0
- sympy 0.7.3 *LM*
- theano 0.5.0 *L*
- tornado 3.1.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.8.0

Python 2.7 (included in installer):

- apptools 4.2.0
- astropy 0.2.5
- atom 0.3.4
- beautiful-soup 4.3.1
- binstar 0.3.1
- biopython 1.62
- bitarray 0.8.1
- blaze 0.3
- bokeh 0.2
- boto 2.15.0
- cairo 1.12.2 *L*
- casuarius 1.1
- chaco 4.3.0
- colorama 0.2.7
- configobj 4.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.19.2
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.5.0
- enable 4.3.0
- enaml 0.8.3
- envisage 4.3.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.2.0
- hdf5 1.8.9
- imaging 1.1.7
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.1
- keyring 3.2
- launcher 0.1.2
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1
- mayavi 4.3.0
- mdp 3.3
- menuinst 1.0.3 *W*
- meta 0.4.2.dev
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.6 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.11.0
- numexpr 2.2.2
- numpy 1.7.1
- opencv 2.4.6 *L*
- openpyxl 1.6.2
- pandas 0.12.0
- patsy 0.2.1
- pep8 1.4.6
- pip 1.4.1
- ply 3.4
- psutil 1.1.2
- py 1.4.17
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyface 4.3.0
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.1.0
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pysal 1.6.0
- pysam 0.6 *LM*
- pyside 1.2.1
- pytables 3.0.0
- pytest 2.4.2
- python 2.7.5
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.8.5
- redis 2.6.9 *LM*
- redis-py 2.7.2 *LM*
- requests 1.2.3
- rope 0.9.4
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.0
- six 1.4.1
- sphinx 1.1.3
- spyder 2.2.5
- sqlalchemy 0.8.3
- statsmodels 0.5.0
- sympy 0.7.3
- theano 0.5.0 *L*
- tornado 3.1.1
- traits 4.3.0
- traitsui 4.3.0
- vtk 5.10.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- astropy 0.2.5
- beautiful-soup 4.3.1
- bitarray 0.8.1
- colorama 0.2.7
- curl 7.30.0 *LM*
- cython 0.19.2
- dateutil 2.1
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.5.0
- flask 0.10.1
- freetype 2.4.10
- greenlet 0.4.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.1
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1
- mdp 3.3
- mingw 4.7 *W*
- netcdf4 1.0.6 *LM*
- networkx 1.8.1
- nose 1.3.0
- numba 0.11.0
- numexpr 2.2.2
- numpy 1.7.1
- openpyxl 1.6.2
- pandas 0.12.0
- patsy 0.2.1
- pillow 2.1.0
- pip 1.4.1
- ply 3.4
- psutil 1.1.2
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6.1
- pyflakes 0.7.3
- pygments 1.6
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pyside 1.2.1 *W*
- pytables 3.0.0
- python 3.3.2
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- requests 1.2.3
- scikit-image 0.9.3
- scipy 0.13.0
- six 1.4.1
- sphinx 1.1.3
- sqlalchemy 0.8.3
- statsmodels 0.5.0
- sympy 0.7.3
- tornado 3.1.1
- werkzeug 0.9.4
- xlrd 0.9.2
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Python 2.6 (available through conda):

- argparse 1.2.1
- astropy 0.2.5
- atom 0.3.4
- beautiful-soup 4.3.1
- biopython 1.62
- bitarray 0.8.1
- boto 2.15.0
- cairo 1.12.2 *L*
- casuarius 1.1
- colorama 0.2.7
- configobj 4.7.2
- curl 7.30.0 *LM*
- cython 0.19.2
- dateutil 2.1
- disco 0.4.4 *L*
- distribute 0.6.45
- docutils 0.11
- dynd-python 0.5.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- gevent 0.13.8
- gevent-websocket 0.3.6
- gevent_zeromq 0.2.5
- greenlet 0.4.1
- grin 1.2.1
- h5py 2.2.0
- hdf5 1.8.9
- imaging 1.1.7
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.1
- libevent 2.0.20
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmmath 0.1.1
- llvmpy 0.12.0
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.6 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.11.0
- numexpr 2.2.2
- numpy 1.7.1
- opencv 2.4.6 *L*
- ordereddict 1.1
- pandas 0.12.0
- patsy 0.2.1
- pep8 1.4.6
- pip 1.4.1
- ply 3.4
- psutil 1.1.2
- py 1.4.17
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.9.1
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyflakes 0.7.3
- pygments 1.6
- pyparsing 1.5.6
- pyreadline 2.0.dev *W*
- pysam 0.6 *LM*
- pytables 3.0.0 *LM*
- pytest 2.4.2
- python 2.6.9
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.7.2 *LM*
- requests 1.2.3
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.0
- six 1.4.1
- sphinx 1.1.3
- sqlalchemy 0.8.3
- statsmodels 0.5.0
- sympy 0.7.3 *LM*
- theano 0.5.0 *L*
- tornado 3.1.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.9.0

Python 2.7 (included in installer):

- apptools 4.2.1
- argcomplete 0.6.7
- astropy 0.3.0
- atom 0.3.6
- beautiful-soup 4.3.1
- binstar 0.4.4
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.1
- blz 0.6.1
- bokeh 0.4
- boto 2.24.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- chaco 4.4.1
- colorama 0.2.7
- conda 3.0.3
- conda-build 1.1.0
- configobj 4.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.20
- datashape 0.1.0
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.0
- enable 4.3.0
- enaml 0.9.0
- envisage 4.4.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- keyring 3.3
- kiwisolver 0.1.2
- launcher 0.1.2
- libdynd 0.6.0 *LM*
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.2
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1
- mayavi 4.3.1
- mdp 3.3
- menuinst 1.0.3 *W*
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.7 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.0
- numexpr 2.3.0
- numpy 1.8.0
- opencv 2.4.6 *L*
- openpyxl 1.8.2
- openssl 1.0.1c *LM*
- pandas 0.13.0
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyface 4.4.0
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.1.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysal 1.6.0
- pysam 0.6 *LM*
- pyside 1.2.1
- pytables 3.1.0
- pytest 2.5.2
- python 2.7.6
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.8.5
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- rope 0.9.4
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.1
- six 1.5.2
- sphinx 1.2.1
- spyder 2.2.5
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tk 8.5.13 *LM*
- tornado 3.2.0
- traits 4.4.0
- traitsui 4.4.0
- ujson 1.33
- vtk 5.10.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- argcomplete 0.6.7
- astropy 0.3.0
- beautiful-soup 4.3.1
- bitarray 0.8.1
- blaze 0.4.1
- blz 0.6.1
- bokeh 0.4
- cdecimal 2.3
- colorama 0.2.7
- curl 7.30.0 *LM*
- cython 0.20
- datashape 0.1.0
- dateutil 2.1
- docutils 0.11
- dynd-python 0.6.0
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- greenlet 0.4.2
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- libdynd 0.6.0 *LM*
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.2
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- netcdf4 1.0.7 *LM*
- networkx 1.8.1
- nose 1.3.0
- numba 0.12.0
- numexpr 2.3.0
- numpy 1.8.0
- openpyxl 1.8.2
- pandas 0.13.0
- patsy 0.2.1
- pillow 2.1.0
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.1.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pyside 1.1.2 *W*
- pytables 3.1.0
- pytest 2.5.2
- python 3.3.3
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.1
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- tornado 3.2.0
- ujson 1.33
- werkzeug 0.9.4
- xlrd 0.9.2
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Python 2.6 (available through conda):

- argcomplete 0.6.7
- argparse 1.2.1
- astropy 0.3.0
- atom 0.3.6
- beautiful-soup 4.3.1
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.1
- blz 0.6.1
- boto 2.24.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- colorama 0.2.7
- configobj 4.7.2
- curl 7.30.0 *LM*
- cython 0.20
- datashape 0.1.0
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- kiwisolver 0.1.2
- libdynd 0.6.0 *LM*
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.2
- lxml 3.2.3
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.7 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.0
- numexpr 2.3.0
- numpy 1.8.0
- opencv 2.4.6 *L*
- ordereddict 1.1
- pandas 0.13.0
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.1.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysam 0.6 *LM*
- pytables 3.1.0 *LM*
- pytest 2.5.2
- python 2.6.9
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.1
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tornado 3.2.0
- traits 4.4.0
- ujson 1.33
- unittest2 0.5.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.9.1

Python 2.7 (included in installer):

- apptools 4.2.1
- argcomplete 0.6.7
- astropy 0.3.0
- atom 0.3.7
- beautiful-soup 4.3.1
- binstar 0.4.4
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.2
- blz 0.6.1
- bokeh 0.4.1
- boto 2.25.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- chaco 4.4.1
- colorama 0.2.7
- conda 3.0.6
- conda-build 1.2.0
- configobj 4.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.1
- enable 4.3.0
- enaml 0.9.1
- envisage 4.4.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- keyring 3.3
- kiwisolver 0.1.2
- launcher 0.1.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1
- mayavi 4.3.1
- mdp 3.3
- menuinst 1.0.3 *W*
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- opencv 2.4.6 *L*
- openpyxl 1.8.2
- openssl 1.0.1c *LM*
- pandas 0.13.1
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyface 4.4.0
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysal 1.6.0
- pysam 0.6 *LM*
- pyside 1.2.1
- pytables 3.1.0
- pytest 2.5.2
- python 2.7.6
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.8.5
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- rope 0.9.4
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- spyder 2.2.5
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tk 8.5.13 *LM*
- tornado 3.2.0
- traits 4.4.0
- traitsui 4.4.0
- ujson 1.33
- vtk 5.10.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- argcomplete 0.6.7
- astropy 0.3.0
- beautiful-soup 4.3.1
- bitarray 0.8.1
- blaze 0.4.2
- blz 0.6.1
- bokeh 0.4.1
- cdecimal 2.3
- colorama 0.2.7
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- docutils 0.11
- dynd-python 0.6.1
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- greenlet 0.4.2
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- openpyxl 1.8.2
- openssl 1.0.1c *LM*
- pandas 0.13.1
- patsy 0.2.1
- pillow 2.1.0
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pyside 1.2.1 *W*
- pytables 3.1.0
- pytest 2.5.2
- python 3.3.4
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- tk 8.5.13 *LM*
- tornado 3.2.0
- ujson 1.33
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Python 2.6 (available through conda):

- argcomplete 0.6.7
- argparse 1.2.1
- astropy 0.3.0
- atom 0.3.7
- beautiful-soup 4.3.1
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.2 *LM*
- blz 0.6.1
- boto 2.25.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- colorama 0.2.7
- configobj 4.7.2
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.1
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- kiwisolver 0.1.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- opencv 2.4.6 *L*
- openssl 1.0.1c *LM*
- ordereddict 1.1
- pandas 0.13.1
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysam 0.6 *LM*
- pytables 3.1.0 *LM*
- pytest 2.5.2
- python 2.6.9
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tk 8.5.13 *LM*
- tornado 3.2.0
- traits 4.4.0
- ujson 1.33
- unittest2 0.5.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 1.9.2

Python 2.7 (included in installer):

- apptools 4.2.1
- argcomplete 0.6.7
- astropy 0.3.0
- atom 0.3.7
- beautiful-soup 4.3.1
- binstar 0.4.4
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.2
- blz 0.6.1
- bokeh 0.4.1
- boto 2.25.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- chaco 4.4.1
- colorama 0.2.7
- conda 3.4.1
- conda-build 1.3.1
- configobj 4.7.2
- cubes 0.10.2
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.1
- enable 4.3.0
- enaml 0.9.1
- envisage 4.4.0
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- keyring 3.3
- kiwisolver 0.1.2
- launcher 0.1.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1
- mayavi 4.3.1
- mdp 3.3
- menuinst 1.0.3 *W*
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- opencv 2.4.6 *L*
- openpyxl 1.8.2
- openssl 1.0.1g *LM*
- pandas 0.13.1
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pyaudio 0.2.7 *M*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyface 4.4.0
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysal 1.6.0
- pysam 0.6 *LM*
- pyside 1.2.1
- pytables 3.1.0
- pytest 2.5.2
- python 2.7.6
- pytz 2013b
- pywin32 218.4 *W*
- pyyaml 3.10
- pyzmq 2.2.0.1
- qt 4.8.5
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- rope 0.9.4
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- spyder 2.2.5
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tk 8.5.13 *LM*
- tornado 3.2.0
- traits 4.4.0
- traitsui 4.4.0
- ujson 1.33
- vtk 5.10.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Python 3.3 (available through conda):

- argcomplete 0.6.7
- astropy 0.3.0
- beautiful-soup 4.3.1
- bitarray 0.8.1
- blaze 0.4.2
- blz 0.6.1
- bokeh 0.4.1
- cdecimal 2.3
- colorama 0.2.7
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- docutils 0.11
- dynd-python 0.6.1
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- greenlet 0.4.2
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- openpyxl 1.8.2
- openssl 1.0.1g *LM*
- pandas 0.13.1
- patsy 0.2.1
- pillow 2.1.0
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pyside 1.2.1 *W*
- pytables 3.1.0
- pytest 2.5.2
- python 3.3.4
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- tk 8.5.13 *LM*
- tornado 3.2.0
- ujson 1.33
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Python 2.6 (available through conda):

- argcomplete 0.6.7
- argparse 1.2.1
- astropy 0.3.0
- atom 0.3.7
- beautiful-soup 4.3.1
- biopython 1.63
- bitarray 0.8.1
- blaze 0.4.2 *LM*
- blz 0.6.1
- boto 2.25.0
- cairo 1.12.2 *L*
- casuarius 1.1
- cdecimal 2.3
- colorama 0.2.7
- configobj 4.7.2
- curl 7.30.0 *LM*
- cython 0.20.1
- datashape 0.1.1
- dateutil 2.1
- disco 0.4.4 *L*
- docutils 0.11
- dynd-python 0.6.1
- erlang R15B01 *L*
- flask 0.10.1
- freetype 2.4.10
- future 0.11.2
- gevent 1.0
- gevent-websocket 0.9.2
- gevent_zeromq 0.2.5
- greenlet 0.4.2
- grin 1.2.1
- h5py 2.2.1
- hdf5 1.8.9
- ipython 1.1.0
- itsdangerous 0.23
- jinja2 2.7.2
- kiwisolver 0.1.2
- libnetcdf 4.2.1.1 *LM*
- libpng 1.5.13 *LM*
- libsodium 0.4.5 *L*
- libtiff 4.0.2 *LM*
- libxml2 2.9.0 *LM*
- libxslt 1.1.28 *LM*
- llvm 3.3
- llvmpy 0.12.3
- lxml 3.3.1
- markupsafe 0.18
- matplotlib 1.3.1 *LM*
- mdp 3.3
- mingw 4.7 *W*
- mock 1.0.1
- mpi4py 1.3 *L*
- mpich2 1.4.1p1 *L*
- netcdf4 1.0.8 *LM*
- networkx 1.8.1
- nltk 2.0.4
- nose 1.3.0
- numba 0.12.1
- numexpr 2.3.1
- numpy 1.8.0
- opencv 2.4.6 *L*
- openssl 1.0.1g *LM*
- ordereddict 1.1
- pandas 0.13.1
- patsy 0.2.1
- pep8 1.4.6
- pil 1.1.7
- pip 1.5.2
- ply 3.4
- psutil 1.2.1
- py 1.4.20
- py2cairo 1.10.0 *L*
- pycosat 0.6.0
- pycparser 2.10
- pycrypto 2.6.1
- pycurl 7.19.0 *LM*
- pyflakes 0.7.3
- pygments 1.6
- pykit 0.2.0
- pyparsing 2.0.1
- pyreadline 2.0 *W*
- pysam 0.6 *LM*
- pytables 3.1.0 *LM*
- pytest 2.5.2
- python 2.6.9
- pytz 2013b
- pyyaml 3.10
- pyzmq 2.2.0.1
- redis 2.6.9 *LM*
- redis-py 2.9.1 *LM*
- requests 2.2.1
- scikit-image 0.9.3
- scikit-learn 0.14.1
- scipy 0.13.3
- setuptools 2.2
- six 1.5.2
- sphinx 1.2.1
- sqlalchemy 0.9.2
- ssl_match_hostname 3.4.0.2
- statsmodels 0.5.0
- sympy 0.7.4.1
- theano 0.6.0 *L*
- tk 8.5.13 *LM*
- tornado 3.2.0
- traits 4.4.0
- ujson 1.33
- unittest2 0.5.1
- werkzeug 0.9.4
- xlrd 0.9.2
- xlsxwriter 0.5.2
- xlwt 0.7.5
- yaml 0.1.4 *LM*
- zeromq 2.2.0 *LM*
- zlib 1.2.7

Only available on: *L* (Linux) - *M* (macOS) - *W* (Windows)

Packages included in Anaconda 2.0.1 for Python version 2.6

Python version: 2.6

Number of supported packages: 171

Name	Version	License	In Installer
argcomplete	0.6.7	Apache Software License	False
argparse	1.2.1	PSF	False
astroid	1.1.1	LGPL	False
astropy	0.3.2	BSD	False
atom	0.3.7	BSD	False
basemap Linux Mac	1.0.7	PSF	False

Continued on next page

Table 8 – continued from previous page

beautiful-soup	4.3.1	PSF/MIT	False
biopython	1.63	BSD-like	False
bitarray	0.8.1	PSF	False
blaze Linux Mac	0.5.0	BSD	False
blist	1.3.6	BSD	False
blz	0.6.2	BSD	False
boto	2.28.0	MIT	False
bsdiff4	1.1.4	BSD	False
cairo Linux	1.12.2	LGPL 2.1 and MPL 1.1	False
casuarius	1.1	LGPL	False
cdecimal	2.3	BSD	False
cffi Linux Mac	0.8.2	MIT	False
cheetah	2.4.4	MIT	False
chrpath Linux	0.13	GPL	False
colorama	0.2.7	BSD	False
conda-api	1.1.0	BSD	False
configobj	5.0.5	BSD	False
coverage	3.7.1	BSD	False
curl Linux Mac	7.30.0	MIT/X derivate	False
cython	0.20.1	Apache 2.0	False
datashape	0.2.0	BSD	False
dateutil	2.1	BSD	False
decorator	3.4.0	BSD	False
distribute Linux	0.6.45	PSF or ZPL	False
dnspython	1.10.0	as-is	False
docutils	0.11	Public-Domain, PSF, 2-clause BSD, GPL3	False
dynd-python	0.6.2	BSD	False
ecdsa	0.11	MIT	False
faulthandler	2.3	BSD	False
feedparser	5.1.3	MIT	False
flake8	2.1.0	MIT	False
flask	0.10.1	BSD	False
freetype Linux Mac	2.4.10	FreeType License	False
future	0.12.1	MIT	False
futures	2.1.6	BSD	False
gdal	1.10.1	MIT	False
gdata	2.0.18	Apache 2.0	False
geos Linux Mac	3.3.3	LGPL	False
gevent	1.0.1	MIT	False
gevent-websocket	0.9.3	Apache	False
googlecl	0.9.12	Apache 2.0	False
greenlet	0.4.2	MIT	False
grin	1.2.1	BSD	False
gunicorn Linux Mac	18.0	MIT	False
h5py	2.3.0	New BSD	False
hdf5 Linux Mac	1.8.9	BSD-style	False
html5lib	0.999	MIT	False
iopro	1.6.5	proprietary - Continuum Analytics, Inc.	False
itsdangerous	0.24	BSD License	False
jdcal	1.0	BSD	False

Continued on next page

Table 8 – continued from previous page

jinja2	2.7.2	BSD	False
jpeg Linux Mac	8d	Custom free software license	False
kiwisolver	0.1.2	BSD	False
lcms Linux Mac	1.19	MIT	False
libdynd Linux Mac	0.6.2	BSD	False
libffi Linux	3.0.13	MIT	False
libnetcdf Linux Mac	4.2.1.1	MIT	False
libpng Linux Mac	1.5.13	Open Source	False
libsodium Linux Mac	0.4.5	MIT	False
libtiff Linux Mac	4.0.2	as-is	False
libxml2 Linux Mac	2.9.0	MIT	False
libxslt Linux Mac	1.1.28	MIT	False
llvm Linux Mac	3.3	Open Source	False
llvmpy	0.12.6	New BSD License	False
logilab-common	0.61.0	LGPL	False
lxml	3.3.5	BSD	False
markdown Linux Mac	2.4	BSD	False
markupsafe	0.18	BSD	False
mathjax	2.2	Apache	False
matplotlib	1.3.1	PSF-based	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
mercurial	3.0	GPLv2	False
mingw Windows	4.7	GPL	False
mock	1.0.1	BSD	False
mpi4py Linux	1.3	BSD	False
mpich2 Linux	1.4.1p1	mpich license	False
multipledispatch	0.4.3	BSD	False
netcdf4	1.0.8	MIT	False
networkx	1.8.1	BSD	False
nltk	2.0.4	Apache 2.0	False
nose	1.3.3	LGPL	False
numba	0.13.2	numba license	False
numexpr	2.3.1	MIT	False
numpy	1.8.1	BSD	False
numpydoc	0.4	BSD	False
openssl Linux Mac	1.0.1h	Apache-style	False
ordereddict	1.1	MIT	False
pandas	0.14.0	BSD	False
pandasql	0.4.2	BSD	False
paramiko	1.14.0	LGPL	False
pastedeploy Linux Mac	1.5.2	MIT	False
patchelf Linux	0.6	GPL3	False
patsy	0.2.1	BSD License	False
pep8	1.5.6	MIT License	False
pil	1.1.7	PIL license	False
pillow Linux Mac	2.4.0	Standard PIL license	False
pip	1.5.6	MIT	False
pixman Linux	0.26.2	MIT	False
ply	3.4	BSD	False

Continued on next page

Table 8 – continued from previous page

psutil	2.1.1	BSD	False
py	1.4.20	MIT	False
py2cairo $^{\text{Linux}}$	1.10.0	LGPL 2.1 and MPL 1.1	False
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	False
pycparser	2.10	BSD	False
pycrypto	2.6.1	Public Domain	False
pycurl $^{\text{Linux Mac}}$	7.19.3.1	LGPL and MIT/X	False
pyflakes	0.8.1	MIT	False
pygments	1.6	BSD	False
pylint	1.2.1	GPL	False
pymc $^{\text{Linux}}$	2.3.2	Academic Free License	False
pyodbc	3.0.7	MIT	False
pyparsing	2.0.1	MIT	False
pyqt	4.10.4	GPL	False
pyreadline $^{\text{Windows}}$	2.0	BSD	False
pysam $^{\text{Linux Mac}}$	0.6	MIT	False
pytables $^{\text{Linux Mac}}$	3.1.1	BSD	False
pytest	2.5.2	MIT	False
python	2.6.9	PSF	False
pytz	2014.3	MIT	False
pyyaml	3.11	MIT	False
pyzmq	14.3.0	LGPL and BSD	False
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	False
readline $^{\text{Linux Mac}}$	6.2	GPL 3	False
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	False
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	False
requests	2.3.0	ISC	False
rope	0.9.4	GPL	False
scikit-image	0.10.0	Modified BSD	False
scikit-learn	0.14.1	3-clause BSD	False
scipy	0.14.0	BSD	False
setuptools	3.6	PSF or ZPL	False
shapely $^{\text{Linux Mac}}$	1.3.2	BSD	False
sip $^{\text{Linux Mac}}$	4.15.5	GPL	False
six	1.6.1	MIT	False
sphinx	1.2.2	BSD	False
sqlalchemy	0.9.4	MIT	False
sqlite $^{\text{Linux Mac}}$	3.8.4.1	Public Domain	False
sqlparse	0.1.11	BSD	False
ssh	1.8.0	LGPL	False
ssl_match_hostname	3.4.0.2	PSF	False
starcluster $^{\text{Linux}}$	0.93.3	LGPL	False
statsmodels	0.5.0	3-clause Modified BSD	False
sympy	0.7.5	New BSD	False
theano $^{\text{Linux}}$	0.6.0	BSD	False
tk $^{\text{Linux Mac}}$	8.5.15	BSD-style	False
tornado	3.2.1	Apache	False
traits	4.4.0	BSD	False
twisted	14.0.0	MIT	False

Continued on next page

Table 8 – continued from previous page

ujson	1.33	BSD	False
unittest2	0.5.1	BSD	False
unixodbc $^{\text{Linux}}$	2.3.1	???	False
util-linux $^{\text{Linux}}$	2.21	GPL	False
w3lib	1.5	BSD	False
werkzeug	0.9.6	BSD	False
whoosh	2.5.7	BSD	False
workerpool $^{\text{Linux}}$	0.9.2	MIT	False
xlrd	0.9.3	BSD	False
xlsxwriter	0.5.5	BSD	False
xlwt	0.7.5	BSD	False
yaml $^{\text{Linux Mac}}$	0.1.4	MIT	False
zeromq $^{\text{Linux Mac}}$	4.0.4	LGPL	False
zlib $^{\text{Linux Mac}}$	1.2.7	zlib	False
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.0.1 for Python version 2.7

Python version: 2.7

Number of supported packages: 197

Name	Version	License	In Installer
apptools	4.2.1	BSD	False
argcomplete	0.6.7	Apache Software License	True
astroid	1.1.1	LGPL	False
astropy	0.3.2	BSD	True
atom	0.3.7	BSD	True
basemap	1.0.7	PSF	False
beautiful-soup	4.3.1	PSF/MIT	True
binstar	0.5.3	BSD	True
biopython	1.63	BSD-like	False
bitarray	0.8.1	PSF	True
blaze	0.5.0	BSD	True
blist	1.3.6	BSD	False
blz	0.6.2	BSD	True
bokeh	0.4.4	New BSD	True
boto	2.28.0	MIT	True
bsdiff4	1.1.4	BSD	False
cairo $^{\text{Linux}}$	1.12.2	LGPL 2.1 and MPL 1.1	True
casuarius	1.1	LGPL	True
cdecimal	2.3	BSD	True
cffi $^{\text{Linux Mac}}$	0.8.2	MIT	False
chaco	4.4.1	BSD	True
cheetah	2.4.4	MIT	False
chrpath $^{\text{Linux}}$	0.13	GPL	False
colorama	0.2.7	BSD	True
conda	3.5.5	BSD	True
conda-api	1.1.0	BSD	False
conda-build	1.3.5	BSD	True

Continued on next page

Table 9 – continued from previous page

configobj	5.0.5	BSD	True
coverage	3.7.1	BSD	False
cubes	0.10.2	MIT	True
curl $^{\text{Linux Mac}}$	7.30.0	MIT/X derivate	True
cython	0.20.1	Apache 2.0	True
datashape	0.2.0	BSD	True
dateutil	2.1	BSD	True
decorator	3.4.0	BSD	False
distribute $^{\text{Linux Mac}}$	0.6.45	PSF or ZPL	False
dnspython	1.10.0	as-is	False
docutils	0.11	Public-Domain, PSF, 2-clause BSD, GPL3	True
dynd-python	0.6.2	BSD	True
ecdsa	0.11	MIT	False
enable	4.3.0	BSD	True
enaml	0.9.1	BSD	True
envisage	4.4.0	BSD	False
faulthandler	2.3	BSD	False
feedparser	5.1.3	MIT	False
fiona	1.1.4	BSD	False
flake8	2.1.0	MIT	False
flask	0.10.1	BSD	True
freetype $^{\text{Linux Mac}}$	2.4.10	FreeType License	True
future	0.12.1	MIT	True
futures	2.1.6	BSD	False
gdal	1.10.1	MIT	False
gdata	2.0.18	Apache 2.0	False
geos $^{\text{Linux Mac}}$	3.3.3	LGPL	False
gevent	1.0.1	MIT	True
gevent-websocket	0.9.3	Apache	True
googlecl	0.9.12	Apache 2.0	False
greenlet	0.4.2	MIT	True
grin	1.2.1	BSD	True
gunicorn $^{\text{Linux Mac}}$	18.0	MIT	False
h5py	2.3.0	New BSD	True
hdf5 $^{\text{Linux Mac}}$	1.8.9	BSD-style	True
html5lib	0.999	MIT	False
hyde $^{\text{Linux Mac}}$	0.8.5	MIT	False
iopro	1.6.5	proprietary - Continuum Analytics, Inc.	False
ipython	2.1.0	BSD	True
itsdangerous	0.24	BSD License	True
jdcal	1.0	BSD	True
jinja2	2.7.2	BSD	True
jpeg $^{\text{Linux Mac}}$	8d	Custom free software license	True
keyring	3.7	PSF	False
kiwisolver	0.1.2	BSD	True
launcher	0.1.5	proprietary - Continuum Analytics, Inc.	True
lcms $^{\text{Linux Mac}}$	1.19	MIT	True
libdynd $^{\text{Linux Mac}}$	0.6.2	BSD	True
libffi $^{\text{Linux}}$	3.0.13	MIT	False
libnetcdf $^{\text{Linux Mac}}$	4.2.1.1	MIT	False

Continued on next page

Table 9 – continued from previous page

libpng $^{\text{Linux Mac}}$	1.5.13	Open Source	True
libsodium $^{\text{Linux Mac}}$	0.4.5	MIT	True
libtiff $^{\text{Linux Mac}}$	4.0.2	as-is	True
libxml2 $^{\text{Linux Mac}}$	2.9.0	MIT	True
libxslt $^{\text{Linux Mac}}$	1.1.28	MIT	True
llvm $^{\text{Linux Mac}}$	3.3	Open Source	True
llvmpy	0.12.6	New BSD License	True
logilab-common	0.61.0	LGPL	False
lxml	3.3.5	BSD	True
markdown $^{\text{Linux Mac}}$	2.4	BSD	False
markupsafe	0.18	BSD	True
mathjax	2.2	Apache	False
matplotlib	1.3.1	PSF-based	True
mayavi	4.3.1	BSD	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
menuinst $^{\text{Windows}}$	1.0.3	BDF	True
mercurial	3.0	GPLv2	False
mingw $^{\text{Windows}}$	4.7	GPL	True
mock	1.0.1	BSD	True
mpi4py $^{\text{Linux}}$	1.3	BSD	True
mpich2 $^{\text{Linux}}$	1.4.1p1	mpich license	True
multipledispatch	0.4.3	BSD	True
netcdf4	1.0.8	MIT	False
networkx	1.8.1	BSD	True
nltk	2.0.4	Apache 2.0	True
nose	1.3.3	LGPL	True
numba	0.13.2	numba license	True
numexpr	2.3.1	MIT	True
numpy	1.8.1	BSD	True
numpydoc	0.4	BSD	False
openpyxl	1.8.5	MIT/Expat	True
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	True
pandas	0.14.0	BSD	True
pandasql	0.4.2	BSD	False
paramiko	1.14.0	LGPL	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.2.1	BSD License	True
pep8	1.5.6	MIT License	True
pil	1.1.7	PIL license	True
pillow $^{\text{Linux Mac}}$	2.4.0	Standard PIL license	False
pip	1.5.6	MIT	True
pixman $^{\text{Linux}}$	0.26.2	MIT	True
ply	3.4	BSD	True
psutil	2.1.1	BSD	True
py	1.4.20	MIT	True
py2cairo $^{\text{Linux}}$	1.10.0	LGPL 2.1 and MPL 1.1	True
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pyaudio $^{\text{Mac}}$	0.2.7	MIT	True

Continued on next page

Table 9 – continued from previous page

pycosat	0.6.1	MIT	True
pycparser	2.10	BSD	True
pycrypto	2.6.1	Public Domain	True
pycurl $^{\text{Linux Mac}}$	7.19.3.1	LGPL and MIT/X	True
pyface	4.4.0	BSD	True
pyflakes	0.8.1	MIT	True
pygments	1.6	BSD	True
pylint	1.2.1	GPL	False
pymc $^{\text{Linux}}$	2.3.2	Academic Free License	False
pyodbc	3.0.7	MIT	False
pyparsing	2.0.1	MIT	True
pyqt	4.10.4	GPL	True
pyreadline $^{\text{Windows}}$	2.0	BSD	True
pysal	1.6.0	New BSD License	False
pysam $^{\text{Linux Mac}}$	0.6	MIT	False
pytables	3.1.1	BSD	True
pytest	2.5.2	MIT	True
python	2.7.7	PSF	True
pytz	2014.3	MIT	True
pywin32 $^{\text{Windows}}$	218.4	PSF	True
pyyaml	3.11	MIT	True
pyzmq	14.3.0	LGPL and BSD	True
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	True
readline $^{\text{Linux Mac}}$	6.2	GPL 3	True
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	True
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	True
reportlab	3.1.8	BSD	False
requests	2.3.0	ISC	True
rope	0.9.4	GPL	True
runipy	0.1.0	BSD	True
scikit-image	0.10.0	Modified BSD	True
scikit-learn	0.14.1	3-clause BSD	True
scipy	0.14.0	BSD	True
setuptools	3.6	PSF or ZPL	True
shapely $^{\text{Linux Mac}}$	1.3.2	BSD	False
sip $^{\text{Linux Mac}}$	4.15.5	GPL	True
six	1.6.1	MIT	True
sphinx	1.2.2	BSD	True
spyder	2.3.0rc1	MIT	True
sqlalchemy	0.9.4	MIT	True
sqlite $^{\text{Linux Mac}}$	3.8.4.1	Public Domain	True
sqlparse	0.1.11	BSD	False
ssh	1.8.0	LGPL	False
ssl_match_hostname	3.4.0.2	PSF	True
starcluster $^{\text{Linux}}$	0.93.3	LGPL	False
statsmodels	0.5.0	3-clause Modified BSD	True
sympy	0.7.5	New BSD	True
theano $^{\text{Linux}}$	0.6.0	BSD	True
tk $^{\text{Linux Mac}}$	8.5.15	BSD-style	True
tornado	3.2.1	Apache	True

Continued on next page

Table 9 – continued from previous page

traits	4.4.0	BSD	True
traitsui	4.4.0	BSD	True
twisted	14.0.0	MIT	False
ujson	1.33	BSD	True
unixodbc $^{\text{Linux}}$	2.3.1	???	False
util-linux $^{\text{Linux}}$	2.21	GPL	True
vtk	5.10.1	BSD	False
w3lib	1.5	BSD	False
werkzeug	0.9.6	BSD	True
whoosh	2.5.7	BSD	False
workerpool $^{\text{Linux}}$	0.9.2	MIT	False
xlrd	0.9.3	BSD	True
xlsxwriter	0.5.5	BSD	True
xlwings $^{\text{Windows}}$	0.1.0	BSD 3-clause	True
xlwt	0.7.5	BSD	True
yaml $^{\text{Linux Mac}}$	0.1.4	MIT	True
yt $^{\text{Linux Mac}}$	2.6.2	BSD	False
zeromq $^{\text{Linux Mac}}$	4.0.4	LGPL	True
zlib $^{\text{Linux Mac}}$	1.2.7	zlib	True
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.0.1 for Python version 3.3

Python version: 3.3

Number of supported packages: 141

Name	Version	License	In Installer
argcomplete	0.6.7	Apache Software License	False
astroid	1.1.1	LGPL	False
astropy	0.3.2	BSD	False
basemap $^{\text{Linux Mac}}$	1.0.7	PSF	False
beautiful-soup	4.3.1	PSF/MIT	False
binstar	0.5.3	BSD	False
biopython	1.63	BSD-like	False
bitarray	0.8.1	PSF	False
blaze	0.5.0	BSD	False
blist	1.3.6	BSD	False
blz	0.6.2	BSD	False
bokeh	0.4.4	New BSD	False
bsdiff4	1.1.4	BSD	False
cdecimal	2.3	BSD	False
cffi $^{\text{Linux}}$	0.8.2	MIT	False
chrpath $^{\text{Linux}}$	0.13	GPL	False
colorama	0.2.7	BSD	False
conda	3.5.5	BSD	False
conda-api	1.1.0	BSD	False
conda-build	1.3.5	BSD	False
configobj	5.0.5	BSD	False
coverage	3.7.1	BSD	False

Continued on next page

Table 10 – continued from previous page

curl Linux Mac	7.30.0	MIT/X derivate	False
cython	0.20.1	Apache 2.0	False
datashape	0.2.0	BSD	False
dateutil	2.1	BSD	False
docutils	0.11	Public-Domain, PSF, 2-clause BSD, GPL3	False
dynd-python	0.6.2	BSD	False
ecdsa	0.11	MIT	False
feedparser	5.1.3	MIT	False
fiona	1.1.4	BSD	False
flake8	2.1.0	MIT	False
flask	0.10.1	BSD	False
freetype Linux Mac	2.4.10	FreeType License	False
future	0.12.1	MIT	False
futures	2.1.6	BSD	False
gdal	1.10.1	MIT	False
geos Linux Mac	3.3.3	LGPL	False
greenlet	0.4.2	MIT	False
gunicorn Linux Mac	18.0	MIT	False
h5py	2.3.0	New BSD	False
hdf5 Linux Mac	1.8.9	BSD-style	False
html5lib	0.999	MIT	False
ipython	2.1.0	BSD	False
itsdangerous	0.24	BSD License	False
jdcal	1.0	BSD	False
jinja2	2.7.2	BSD	False
jpeg Linux Mac	8d	Custom free software license	False
libdynd Linux Mac	0.6.2	BSD	False
libffi Linux	3.0.13	MIT	False
libnetcdf Linux Mac	4.2.1.1	MIT	False
libpng Linux Mac	1.5.13	Open Source	False
libsodium Linux Mac	0.4.5	MIT	False
libtiff Linux Mac	4.0.2	as-is	False
libxml2 Linux Mac	2.9.0	MIT	False
libxslt Linux Mac	1.1.28	MIT	False
llvm Linux Mac	3.3	Open Source	False
llvmpy	0.12.6	New BSD License	False
logilab-common	0.61.0	LGPL	False
lxml	3.3.5	BSD	False
markdown Linux Mac	2.4	BSD	False
markupsafe	0.18	BSD	False
mathjax	2.2	Apache	False
matplotlib	1.3.1	PSF-based	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
mingw Windows	4.7	GPL	False
mock	1.0.1	BSD	False
multipledispatch	0.4.3	BSD	False
netcdf4	1.0.8	MIT	False
networkx	1.8.1	BSD	False
nose	1.3.3	LGPL	False

Continued on next page

Table 10 – continued from previous page

numba	0.13.2	numba license	False
numexpr	2.3.1	MIT	False
numpy	1.8.1	BSD	False
openpyxl	1.8.5	MIT/Expat	False
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	False
pandas	0.14.0	BSD	False
pandasql	0.4.2	BSD	False
paramiko $^{\text{Linux Mac}}$	1.14.0	LGPL	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.2.1	BSD License	False
pep8	1.5.6	MIT License	False
pillow $^{\text{Linux Mac}}$	2.4.0	Standard PIL license	False
pip	1.5.6	MIT	False
ply	3.4	BSD	False
psutil	2.1.1	BSD	False
py	1.4.20	MIT	False
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	False
pycparser	2.10	BSD	False
pycrypto	2.6.1	Public Domain	False
pyflakes	0.8.1	MIT	False
pygments	1.6	BSD	False
pylint	1.2.1	GPL	False
pyodbc	3.0.7	MIT	False
pyparsing	2.0.1	MIT	False
pyqt	4.10.4	GPL	False
pyreadline $^{\text{Windows}}$	2.0	BSD	False
pytables	3.1.1	BSD	False
pytest	2.5.2	MIT	False
python	3.3.5	PSF	False
pytz	2014.3	MIT	False
pyyaml	3.11	MIT	False
pyzmq	14.3.0	LGPL and BSD	False
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	False
readline $^{\text{Linux Mac}}$	6.2	GPL 3	False
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	False
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	False
reportlab	3.1.8	BSD	False
requests	2.3.0	ISC	False
rope	0.9.4	GPL	False
runipy	0.1.0	BSD	False
scikit-image	0.10.0	Modified BSD	False
scikit-learn	0.14.1	3-clause BSD	False
scipy	0.14.0	BSD	False
setuptools	3.6	PSF or ZPL	False
shapely $^{\text{Linux Mac}}$	1.3.2	BSD	False
sip $^{\text{Linux Mac}}$	4.15.5	GPL	False
six	1.6.1	MIT	False
sphinx	1.2.2	BSD	False

Continued on next page

Table 10 – continued from previous page

sqlalchemy	0.9.4	MIT	False
sqlite $^{\text{Linux Mac}}$	3.8.4.1	Public Domain	False
sqlparse	0.1.11	BSD	False
ssl_match_hostname	3.4.0.2	PSF	False
statsmodels	0.5.0	3-clause Modified BSD	False
sympy	0.7.5	New BSD	False
tk $^{\text{Linux Mac}}$	8.5.15	BSD-style	False
tornado	3.2.1	Apache	False
ujson	1.33	BSD	False
unixodbc $^{\text{Linux}}$	2.3.1	???	False
util-linux $^{\text{Linux}}$	2.21	GPL	False
werkzeug	0.9.6	BSD	False
whoosh	2.5.7	BSD	False
xlrd	0.9.3	BSD	False
xlsxwriter	0.5.5	BSD	False
yaml $^{\text{Linux Mac}}$	0.1.4	MIT	False
zeromq $^{\text{Linux Mac}}$	4.0.4	LGPL	False
zlib $^{\text{Linux Mac}}$	1.2.7	zlib	False
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.0.1 for Python version 3.4

Python version: 3.4

Number of supported packages: 141

Name	Version	License	In Installer
argcomplete	0.6.7	Apache Software License	True
astroid	1.1.1	LGPL	False
astropy	0.3.2	BSD	True
basemap $^{\text{Linux Mac}}$	1.0.7	PSF	False
beautiful-soup	4.3.1	PSF/MIT	True
binstar	0.5.3	BSD	True
biopython	1.63	BSD-like	False
bitarray	0.8.1	PSF	True
blaze	0.5.0	BSD	True
blist	1.3.6	BSD	False
blz	0.6.2	BSD	True
bokeh	0.4.4	New BSD	True
bsdiff4	1.1.4	BSD	False
cdecimal	2.3	BSD	True
cffi $^{\text{Linux}}$	0.8.2	MIT	False
chrpath $^{\text{Linux}}$	0.13	GPL	False
colorama	0.2.7	BSD	True
conda	3.5.5	BSD	True
conda-api	1.1.0	BSD	False
conda-build	1.3.5	BSD	True
configobj	5.0.5	BSD	True
coverage	3.7.1	BSD	False
curl $^{\text{Linux Mac}}$	7.30.0	MIT/X derivate	True

Continued on next page

Table 11 – continued from previous page

cython	0.20.1	Apache 2.0	True
datashape	0.2.0	BSD	True
dateutil	2.1	BSD	True
docutils	0.11	Public-Domain, PSF, 2-clause BSD, GPL3	True
dynd-python	0.6.2	BSD	True
ecdsa	0.11	MIT	False
feedparser	5.1.3	MIT	False
fiona	1.1.4	BSD	False
flake8	2.1.0	MIT	False
flask	0.10.1	BSD	True
freetype Linux Mac	2.4.10	FreeType License	True
future	0.12.1	MIT	True
futures	2.1.6	BSD	False
gdal	1.10.1	MIT	False
geos Linux Mac	3.3.3	LGPL	False
greenlet	0.4.2	MIT	True
gunicorn Linux Mac	18.0	MIT	False
h5py	2.3.0	New BSD	True
hdf5 Linux Mac	1.8.9	BSD-style	True
html5lib	0.999	MIT	False
ipython	2.1.0	BSD	True
itsdangerous	0.24	BSD License	True
jdcal	1.0	BSD	True
jinja2	2.7.2	BSD	True
jpeg Linux Mac	8d	Custom free software license	True
libdynd Linux Mac	0.6.2	BSD	True
libffi Linux	3.0.13	MIT	False
libnetcdf Linux Mac	4.2.1.1	MIT	False
libpng Linux Mac	1.5.13	Open Source	True
libsodium Linux Mac	0.4.5	MIT	True
libtiff Linux Mac	4.0.2	as-is	True
libxml2 Linux Mac	2.9.0	MIT	True
libxslt Linux Mac	1.1.28	MIT	True
llvm Linux Mac	3.3	Open Source	True
llvmpy	0.12.6	New BSD License	True
logilab-common	0.61.0	LGPL	False
lxml	3.3.5	BSD	True
markdown Linux Mac	2.4	BSD	False
markupsafe	0.18	BSD	True
mathjax	2.2	Apache	False
matplotlib	1.3.1	PSF-based	True
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
menuinst Windows	1.0.3	BDF	True
mingw Windows	4.7	GPL	True
mock	1.0.1	BSD	True
multipledispatch	0.4.3	BSD	True
netcdf4	1.0.8	MIT	False
networkx	1.8.1	BSD	True
nose	1.3.3	LGPL	True

Continued on next page

Table 11 – continued from previous page

numba	0.13.2	numba license	True
numexpr	2.3.1	MIT	True
numpy	1.8.1	BSD	True
openpyxl	1.8.5	MIT/Expat	True
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	True
pandas	0.14.0	BSD	True
pandasql	0.4.2	BSD	False
paramiko $^{\text{Linux Mac}}$	1.14.0	LGPL	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.2.1	BSD License	True
pep8	1.5.6	MIT License	False
pillow $^{\text{Linux Mac}}$	2.4.0	Standard PIL license	True
pip	1.5.6	MIT	True
ply	3.4	BSD	True
psutil	2.1.1	BSD	True
py	1.4.20	MIT	True
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	True
pycparser	2.10	BSD	True
pycrypto	2.6.1	Public Domain	True
pyflakes	0.8.1	MIT	True
pygments	1.6	BSD	True
pylint	1.2.1	GPL	False
pyodbc	3.0.7	MIT	False
pyparsing	2.0.1	MIT	True
pyqt	4.10.4	GPL	True
pyreadline $^{\text{Windows}}$	2.0	BSD	True
pytables	3.1.1	BSD	True
pytest	2.5.2	MIT	True
python	3.4.1	PSF	True
pytz	2014.3	MIT	True
pyyaml	3.11	MIT	True
pyzmq	14.3.0	LGPL and BSD	True
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	True
readline $^{\text{Linux Mac}}$	6.2	GPL 3	True
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	True
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	True
reportlab	3.1.8	BSD	False
requests	2.3.0	ISC	True
rope	0.9.4	GPL	True
runipy	0.1.0	BSD	True
scikit-image	0.10.0	Modified BSD	True
scipy	0.14.0	BSD	True
setuptools	3.6	PSF or ZPL	True
shapely $^{\text{Linux Mac}}$	1.3.2	BSD	False
sip $^{\text{Linux Mac}}$	4.15.5	GPL	True
six	1.6.1	MIT	True
sphinx	1.2.2	BSD	True
spyder	2.3.0rc1	MIT	True

Continued on next page

Table 11 – continued from previous page

sqlalchemy	0.9.4	MIT	True
sqlite $^{\text{Linux Mac}}$	3.8.4.1	Public Domain	True
sqlparse	0.1.11	BSD	False
ssl_match_hostname	3.4.0.2	PSF	True
sympy	0.7.5	New BSD	True
tk $^{\text{Linux Mac}}$	8.5.15	BSD-style	True
tornado	3.2.1	Apache	True
ujson	1.33	BSD	True
unixodbc $^{\text{Linux}}$	2.3.1	???	False
util-linux $^{\text{Linux}}$	2.21	GPL	True
werkzeug	0.9.6	BSD	True
whoosh	2.5.7	BSD	False
xlrd	0.9.3	BSD	True
xlsxwriter	0.5.5	BSD	True
yaml $^{\text{Linux Mac}}$	0.1.4	MIT	True
zeromq $^{\text{Linux Mac}}$	4.0.4	LGPL	True
zlib $^{\text{Linux Mac}}$	1.2.7	zlib	True
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.1.0 for Python version 2.6

Python version: 2.6

Number of supported packages: 194

Name	Version	License	In Installer
abstract-rendering $^{\text{Linux Mac}}$	0.5.1	3rd-clause BSD	False
argcomplete	0.8.1	Apache Software License	False
argparse	1.2.1	PSF	False
astroid	1.2.1	LGPL	False
astropy	0.4.2	BSD	False
atom	0.3.9	BSD	False
basemap $^{\text{Linux Mac}}$	1.0.7	PSF	False
bcolz	0.7.1	BSD	False
beautiful-soup	4.3.2	PSF/MIT	False
biopython	1.64	BSD-like	False
bitarray	0.8.1	PSF	False
blaze $^{\text{Linux Mac}}$	0.6.3	BSD	False
blist	1.3.6	BSD	False
blz	0.6.2	BSD	False
boto	2.32.1	MIT	False
bsddb $^{\text{Linux Mac}}$	1.0	PSF	False
bsdiff4	1.1.4	BSD	False
cairo $^{\text{Linux}}$	1.12.2	LGPL 2.1 and MPL 1.1	False
casuarius	1.1	LGPL	False
cdecimal	2.3	BSD	False
cffi	0.8.6	MIT	False
chameleon	2.16	BSD-like	False
cheetah	2.4.4	MIT	False
chrpath $^{\text{Linux}}$	0.13	GPL	False

Continued on next page

Table 12 – continued from previous page

colorama	0.3.1	BSD	False
conda-api	1.1.0	BSD	False
configobj	5.0.6	BSD	False
coverage	3.7.1	BSD	False
cryptography	0.5.4	Apache 2.0	False
cssselect	0.9.1	BSD	False
curl $^{\text{Linux Mac}}$	7.38.0	MIT/X derivate	False
cython	0.21	Apache 2.0	False
cytoolz	0.7.0	BSD	False
datashape	0.3.0	BSD	False
dateutil	2.1	BSD	False
db $^{\text{Linux Mac}}$	5.3.28	AGPLv3	False
decorator	3.4.0	BSD	False
dnspython	1.10.0	as-is	False
docutils	0.12	Public-Domain, PSF, 2-clause BSD, GPL3	False
dynd-python	0.6.5	BSD	False
ecdsa	0.11	MIT	False
ephem	3.7.5.3	LGPL	False
faulthandler	2.3	BSD	False
feedparser	5.1.3	MIT	False
flake8	2.2.3	MIT	False
flask	0.10.1	BSD	False
freetype $^{\text{Linux Mac}}$	2.4.10	FreeType License	False
future	0.13.1	MIT	False
futures	2.1.6	BSD	False
gdal	1.11.0	MIT	False
gdata	2.0.18	Apache 2.0	False
gensim	0.10.2	LGPL	False
geos $^{\text{Linux Mac}}$	3.3.3	LGPL	False
gevent	1.0.1	MIT	False
gevent-websocket	0.9.3	Apache	False
googlecl	0.9.12	Apache 2.0	False
greenlet	0.4.4	MIT	False
grin	1.2.1	BSD	False
gunicorn $^{\text{Linux Mac}}$	19.1.0	MIT	False
h5py	2.3.1	New BSD	False
hdf5 $^{\text{Linux Mac}}$	1.8.13	BSD-style	False
html5lib	0.999	MIT	False
iopro	1.6.7	proprietary - Continuum Analytics, Inc.	False
itsdangerous	0.24	BSD License	False
jdcal	1.0	BSD	False
jinja2	2.7.3	BSD	False
jpeg $^{\text{Linux Mac}}$	8d	Custom free software license	False
kiwisolver	0.1.3	BSD	False
lcms $^{\text{Linux Mac}}$	1.19	MIT	False
libdynd $^{\text{Linux Mac}}$	0.6.5	BSD	False
libffi $^{\text{Linux}}$	3.0.13	MIT	False
libnetcdf $^{\text{Linux Mac}}$	4.3.2	MIT	False
libpng $^{\text{Linux Mac}}$	1.5.13	Open Source	False
libsodium $^{\text{Linux Mac}}$	0.4.5	MIT	False

Continued on next page

Table 12 – continued from previous page

libtiff $^{\text{Linux Mac}}$	4.0.2	as-is	False
libxml2 $^{\text{Linux Mac}}$	2.9.0	MIT	False
libxslt $^{\text{Linux Mac}}$	1.1.28	MIT	False
llvm $^{\text{Linux Mac}}$	3.3	Open Source	False
llvmpy	0.12.7	New BSD License	False
logilab-common	0.62.1	LGPL	False
lxml	3.4.0	BSD	False
markupsafe	0.23	BSD	False
mathjax	2.2	Apache	False
matplotlib	1.4.0	PSF-based	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
mercurial	3.1	GPLv2	False
mingw $^{\text{Windows}}$	4.7	GPL	False
mock	1.0.1	BSD	False
mpi4py $^{\text{Linux}}$	1.3	BSD	False
mpich2 $^{\text{Linux}}$	1.4.1p1	mpich license	False
multimethods	1.0.0	MIT	False
multipledispatch	0.4.7	BSD	False
natsort	3.5.0	MIT	False
netcdf4	1.1.1	MIT	False
networkx	1.9.1	BSD	False
nltk	3.0.0	Apache 2.0	False
nose	1.3.4	LGPL	False
numba	0.14.0	numba license	False
numexpr	2.3.1	MIT	False
numpy	1.9.0	BSD	False
numpydoc	0.4	BSD	False
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	False
ordereddict	1.1	MIT	False
pandas	0.14.1	BSD	False
pandasql	0.6.1	BSD	False
paramiko	1.14.1	LGPL	False
passlib	1.6.2	BSD	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.3.0	BSD License	False
pep381client	1.5	Academic Free License, version 3	False
pep8	1.5.7	MIT License	False
pil	1.1.7	PIL license	False
pillow $^{\text{Linux Mac}}$	2.5.1	Standard PIL license	False
pip	1.5.6	MIT	False
pixman $^{\text{Linux}}$	0.26.2	MIT	False
ply	3.4	BSD	False
psutil	2.1.1	BSD	False
py	1.4.25	MIT	False
py2cairo $^{\text{Linux}}$	1.10.0	LGPL 2.1 and MPL 1.1	False
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	False
pycparser	2.10	BSD	False

Continued on next page

Table 12 – continued from previous page

pycrypto	2.6.1	Public Domain	False
pycurl Linux Mac	7.19.5	LGPL and MIT/X	False
pyflakes	0.8.1	MIT	False
pygments	1.6	BSD	False
pylint	1.3.1	GPL	False
pymc Linux	2.3.3	Academic Free License	False
pymongo	2.7.2	Apache 2.0	False
pyodbc	3.0.7	MIT	False
pyopenssl	0.14	APL2	False
pyparsing	2.0.1	MIT	False
pyqt	4.10.4	GPL	False
pyreadline Windows	2.0	BSD	False
pysam Linux Mac	0.6	MIT	False
pyserial	2.7	PSF	False
pytables Linux Mac	3.1.1	BSD	False
pytest	2.6.3	MIT	False
python	2.6.9	PSF	False
pytz	2014.7	MIT	False
pywin32 Windows	219	PSF	False
pyyaml	3.11	MIT	False
pyzmq	14.3.1	LGPL and BSD	False
qt Linux Mac	4.8.5	LGPL	False
queuelib	1.2.2	BSD	False
readline Linux Mac	6.2	GPL 3	False
redis Linux Mac	2.6.9	3-clause BSD	False
redis-py Linux Mac	2.9.1	MIT	False
repoze.lru	0.6	BSD	False
requests	2.4.1	ISC	False
rope	0.9.4	GPL	False
scikit-image	0.10.1	Modified BSD	False
scikit-learn	0.15.2	3-clause BSD	False
scipy	0.14.0	BSD	False
setuptools	5.8	PSF or ZPL	False
shapely Linux Mac	1.4.1	BSD	False
sip Linux Mac	4.15.5	GPL	False
six	1.8.0	MIT	False
sockjs-tornado	1.0.1	MIT	False
sphinx	1.2.3	BSD	False
sqlalchemy	0.9.7	MIT	False
sqlite Linux Mac	3.8.4.1	Public Domain	False
sqlparse	0.1.12	BSD	False
ssh	1.8.0	LGPL	False
ssl_match_hostname	3.4.0.2	PSF	False
starcluster Linux	0.93.3	LGPL	False
statsmodels	0.5.0	3-clause Modified BSD	False
sympy	0.7.5	New BSD	False
theano Linux	0.6.0	BSD	False
tk Linux Mac	8.5.15	BSD-style	False
toolz	0.7.0	BSD	False
tornado	4.0.2	Apache	False

Continued on next page

Table 12 – continued from previous page

traits	4.4.0	BSD	False
twisted	14.0.2	MIT	False
ujson	1.33	BSD	False
unicodecsv	0.9.4	BSD	False
unittest2	0.5.1	BSD	False
unixodbc Linux	2.3.1	???	False
util-linux Linux	2.21	GPL	False
uuid	1.30	???	False
venusian	1.0	BSD	False
werkzeug	0.9.6	BSD	False
whoosh	2.5.7	BSD	False
workerpool Linux	0.9.2	MIT	False
xlrd	0.9.3	BSD	False
xlsxwriter	0.5.7	BSD	False
xlutils	1.7.1	MIT	False
xlwt	0.7.5	BSD	False
yaml Linux Mac	0.1.4	MIT	False
zeromq Linux Mac	4.0.4	LGPL	False
zlib Linux Mac	1.2.7	zlib	False
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.1.0 for Python version 2.7

Python version: 2.7

Number of supported packages: 224

Name	Version	License	In Installer
abstract-rendering Linux Mac	0.5.1	3rd-clause BSD	True
apptools	4.2.1	BSD	False
argcomplete	0.8.1	Apache Software License	True
astroid	1.2.1	LGPL	False
astropy	0.4.2	BSD	True
atom	0.3.9	BSD	True
basemap	1.0.7	PSF	False
bcolz	0.7.1	BSD	False
beautiful-soup	4.3.2	PSF/MIT	True
binstar	0.7.1	BSD	True
biopython	1.64	BSD-like	False
bitarray	0.8.1	PSF	True
blaze	0.6.3	BSD	True
blist	1.3.6	BSD	False
blz	0.6.2	BSD	True
bokeh	0.6.1	New BSD	True
boto	2.32.1	MIT	True
bsddb Linux Mac	1.0	PSF	False
bsdiff4	1.1.4	BSD	False
cairo Linux	1.12.2	LGPL 2.1 and MPL 1.1	True
casuarius	1.1	LGPL	True
cdecimal	2.3	BSD	True

Continued on next page

Table 13 – continued from previous page

cffi	0.8.6	MIT	True
chaco	4.4.1	BSD	True
chameleon	2.16	BSD-like	False
cheetah	2.4.4	MIT	False
chrpath [Linux]	0.13	GPL	False
colorama	0.3.1	BSD	True
conda	3.7.0	BSD	True
conda-api	1.1.0	BSD	False
conda-build	1.8.2	BSD	True
configobj	5.0.6	BSD	True
coverage	3.7.1	BSD	False
cryptography	0.5.4	Apache 2.0	True
cssselect	0.9.1	BSD	False
cubes	0.10.2	MIT	False
curl [Linux Mac]	7.38.0	MIT/X derivate	True
cython	0.21	Apache 2.0	True
cytoolz	0.7.0	BSD	True
datashape	0.3.0	BSD	True
dateutil	2.1	BSD	True
db [Linux Mac]	5.3.28	AGPLv3	False
decorator	3.4.0	BSD	True
django	1.7	BSD	False
dnspython	1.10.0	as-is	False
docutils	0.12	Public-Domain, PSF, 2-clause BSD, GPL3	True
dynd-python	0.6.5	BSD	True
ecdsa	0.11	MIT	False
enable	4.3.0	BSD	True
enaml	0.9.8	BSD	True
envisage	4.4.0	BSD	False
ephem	3.7.5.3	LGPL	False
faulthandler	2.3	BSD	False
feedparser	5.1.3	MIT	False
fiona	1.1.6	BSD	False
flake8	2.2.3	MIT	False
flask	0.10.1	BSD	True
freetype [Linux Mac]	2.4.10	FreeType License	True
future	0.13.1	MIT	True
futures	2.1.6	BSD	True
gdal	1.11.0	MIT	False
gdata	2.0.18	Apache 2.0	False
gensim	0.10.2	LGPL	False
geos [Linux Mac]	3.3.3	LGPL	False
gevent	1.0.1	MIT	True
gevent-websocket	0.9.3	Apache	True
googlecl	0.9.12	Apache 2.0	False
greenlet	0.4.4	MIT	True
grin	1.2.1	BSD	True
gunicorn [Linux Mac]	19.1.0	MIT	False
h5py	2.3.1	New BSD	True
hdf5 [Linux Mac]	1.8.13	BSD-style	True

Continued on next page

Table 13 – continued from previous page

html5lib	0.999	MIT	False
hyde $^{\text{Linux Mac}}$	0.8.5	MIT	False
iopro	1.6.7	proprietary - Continuum Analytics, Inc.	False
ipython	2.2.0	BSD	True
itsdangerous	0.24	BSD License	True
jdcal	1.0	BSD	True
jinja2	2.7.3	BSD	True
jpeg $^{\text{Linux Mac}}$	8d	Custom free software license	True
kiwisolver	0.1.3	BSD	True
launcher $^{\text{Mac Windows}}$	1.0.0	proprietary - Continuum Analytics, Inc.	True
lcms $^{\text{Linux Mac}}$	1.19	MIT	True
libdynd $^{\text{Linux Mac}}$	0.6.5	BSD	True
libffi $^{\text{Linux}}$	3.0.13	MIT	True
libnetcdf $^{\text{Linux Mac}}$	4.3.2	MIT	False
libpng $^{\text{Linux Mac}}$	1.5.13	Open Source	True
libsodium $^{\text{Linux Mac}}$	0.4.5	MIT	True
libtiff $^{\text{Linux Mac}}$	4.0.2	as-is	True
libxml2 $^{\text{Linux Mac}}$	2.9.0	MIT	True
libxslt $^{\text{Linux Mac}}$	1.1.28	MIT	True
llvm $^{\text{Linux Mac}}$	3.3	Open Source	True
llvmpy	0.12.7	New BSD License	True
logilab-common	0.62.1	LGPL	False
lxml	3.4.0	BSD	True
markdown $^{\text{Linux Mac}}$	2.5	BSD	False
markupsafe	0.23	BSD	True
mathjax	2.2	Apache	False
matplotlib	1.4.0	PSF-based	True
mayavi	4.3.1	BSD	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
menuinst $^{\text{Windows}}$	1.0.4	BDF	True
mercurial	3.1	GPLv2	False
mingw $^{\text{Windows}}$	4.7	GPL	True
mock	1.0.1	BSD	True
mpi4py $^{\text{Linux}}$	1.3	BSD	True
mpich2 $^{\text{Linux}}$	1.4.1p1	mpich license	True
multimethods	1.0.0	MIT	False
multipledispatch	0.4.7	BSD	True
natsort	3.5.0	MIT	False
netcdf4	1.1.1	MIT	False
networkx	1.9.1	BSD	True
nltk	3.0.0	Apache 2.0	True
node-webkit $^{\text{Mac Windows}}$	0.10.1	MIT	True
nose	1.3.4	LGPL	True
numba	0.14.0	numba license	True
numexpr	2.3.1	MIT	True
numpy	1.9.0	BSD	True
numpydoc	0.4	BSD	False
openpyxl	1.8.5	MIT/Expat	True
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	True

Continued on next page

Table 13 – continued from previous page

pandas	0.14.1	BSD	True
pandasql	0.6.1	BSD	False
paramiko	1.14.1	LGPL	False
passlib	1.6.2	BSD	False
pastedeploy Linux Mac	1.5.2	MIT	False
patchelf Linux	0.6	GPL3	False
patsy	0.3.0	BSD License	True
pep381client	1.5	Academic Free License, version 3	False
pep8	1.5.7	MIT License	True
pil	1.1.7	PIL license	True
pillow Linux Mac	2.5.1	Standard PIL license	False
pip	1.5.6	MIT	True
pixman Linux	0.26.2	MIT	True
ply	3.4	BSD	True
psutil	2.1.1	BSD	True
py	1.4.25	MIT	True
py2cairo Linux	1.10.0	LGPL 2.1 and MPL 1.1	True
pyasn1 Linux	0.1.6	BSD	False
pyaudio Mac	0.2.7	MIT	True
pycosat	0.6.1	MIT	True
pycparser	2.10	BSD	True
pycrypto	2.6.1	Public Domain	True
pycurl Linux Mac	7.19.5	LGPL and MIT/X	True
pyface	4.4.0	BSD	True
pyflakes	0.8.1	MIT	True
pygments	1.6	BSD	True
pylint	1.3.1	GPL	False
pymc Linux	2.3.3	Academic Free License	False
pymongo	2.7.2	Apache 2.0	False
pyodbc	3.0.7	MIT	False
pyopenssl	0.14	APL2	True
pyparsing	2.0.1	MIT	True
pyqt	4.10.4	GPL	True
pyreadline Windows	2.0	BSD	True
pysal	1.6.0	New BSD License	False
pysam Linux Mac	0.6	MIT	False
pyserial	2.7	PSF	False
pytables	3.1.1	BSD	True
pytest	2.6.3	MIT	True
python	2.7.8	PSF	True
pytz	2014.7	MIT	True
pywin32 Windows	219	PSF	True
pyyaml	3.11	MIT	True
pyzmq	14.3.1	LGPL and BSD	True
qt Linux Mac	4.8.5	LGPL	True
queuelib	1.2.2	BSD	False
readline Linux Mac	6.2	GPL 3	True
redis Linux Mac	2.6.9	3-clause BSD	True
redis-py Linux Mac	2.9.1	MIT	True
reportlab	3.1.8	BSD	False

Continued on next page

Table 13 – continued from previous page

repoze.lru	0.6	BSD	False
requests	2.4.1	ISC	True
rope	0.9.4	GPL	True
runipy	0.1.1	BSD	True
scikit-bio Linux Mac	0.2.0	BSD	False
scikit-image	0.10.1	Modified BSD	True
scikit-learn	0.15.2	3-clause BSD	True
scipy	0.14.0	BSD	True
scrapy	0.24.4	BSD	False
setuptools	5.8	PSF or ZPL	True
shapely Linux Mac	1.4.1	BSD	False
sip Linux Mac	4.15.5	GPL	True
six	1.8.0	MIT	True
sockjs-tornado	1.0.1	MIT	True
sphinx	1.2.3	BSD	True
spyder	2.3.1	MIT	True
sqlalchemy	0.9.7	MIT	True
sqlite Linux Mac	3.8.4.1	Public Domain	True
sqlparse	0.1.12	BSD	False
ssh	1.8.0	LGPL	False
ssl_match_hostname	3.4.0.2	PSF	True
starcluster Linux	0.93.3	LGPL	False
statsmodels	0.5.0	3-clause Modified BSD	True
sympy	0.7.5	New BSD	True
theano Linux	0.6.0	BSD	True
tk Linux Mac	8.5.15	BSD-style	True
toolz	0.7.0	BSD	True
tornado	4.0.2	Apache	True
traits	4.4.0	BSD	True
traitsui	4.4.0	BSD	True
twisted	14.0.2	MIT	False
ujson	1.33	BSD	True
unicodecsv	0.9.4	BSD	True
unixodbc Linux	2.3.1	???	False
util-linux Linux	2.21	GPL	True
uuid	1.30	???	False
venusian	1.0	BSD	False
vtk	5.10.1	BSD	False
w3lib	1.8.1	BSD	False
werkzeug	0.9.6	BSD	True
whoosh	2.5.7	BSD	False
workerpool Linux	0.9.2	MIT	False
xlrd	0.9.3	BSD	True
xlsxwriter	0.5.7	BSD	True
xlutils	1.7.1	MIT	False
xlwings Windows	0.2.2	BSD 3-clause	True
xlwt	0.7.5	BSD	True
yaml Linux Mac	0.1.4	MIT	True
yt Linux Mac	3.0.1	BSD	False
zeromq Linux Mac	4.0.4	LGPL	True

Continued on next page

Table 13 – continued from previous page

zlib Linux Mac	1.2.7	zlib	True
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.1.0 for Python version 3.3

Python version: 3.3

Number of supported packages: 167

Name	Version	License	In Installer
abstract-rendering Linux Mac	0.5.1	3rd-clause BSD	False
argcomplete	0.8.1	Apache Software License	False
astroid	1.2.1	LGPL	False
astropy	0.4.2	BSD	False
basemap Linux Mac	1.0.7	PSF	False
bcolz	0.7.1	BSD	False
beautiful-soup	4.3.2	PSF/MIT	False
binstar	0.7.1	BSD	False
biopython	1.64	BSD-like	False
bitarray	0.8.1	PSF	False
blaze	0.6.3	BSD	False
blist	1.3.6	BSD	False
blz	0.6.2	BSD	False
bokeh	0.6.1	New BSD	False
boto	2.32.1	MIT	False
bsdiff4	1.1.4	BSD	False
cffi	0.8.6	MIT	False
chameleon	2.16	BSD-like	False
chrpath Linux	0.13	GPL	False
colorama	0.3.1	BSD	False
conda	3.7.0	BSD	False
conda-api	1.1.0	BSD	False
conda-build	1.8.2	BSD	False
configobj	5.0.6	BSD	False
coverage	3.7.1	BSD	False
cryptography	0.5.4	Apache 2.0	False
cssselect Linux Mac	0.9.1	BSD	False
curl Linux Mac	7.38.0	MIT/X derivate	False
cython	0.21	Apache 2.0	False
cytoolz	0.7.0	BSD	False
datashape	0.3.0	BSD	False
dateutil	2.1	BSD	False
decorator	3.4.0	BSD	False
django	1.7	BSD	False
docutils	0.12	Public-Domain, PSF, 2-clause BSD, GPL3	False
dynd-python	0.6.5	BSD	False
ecdsa	0.11	MIT	False
ephem	3.7.5.3	LGPL	False
feedparser	5.1.3	MIT	False
fiona	1.1.6	BSD	False

Continued on next page

Table 14 – continued from previous page

flake8	2.2.3	MIT	False
flask	0.10.1	BSD	False
freetype $^{\text{Linux Mac}}$	2.4.10	FreeType License	False
future	0.13.1	MIT	False
futures	2.1.6	BSD	False
gdal	1.11.0	MIT	False
gensim	0.10.2	LGPL	False
geos $^{\text{Linux Mac}}$	3.3.3	LGPL	False
greenlet	0.4.4	MIT	False
gunicorn $^{\text{Linux Mac}}$	19.1.0	MIT	False
h5py	2.3.1	New BSD	False
hdf5 $^{\text{Linux Mac}}$	1.8.13	BSD-style	False
html5lib	0.999	MIT	False
ipython	2.2.0	BSD	False
itsdangerous	0.24	BSD License	False
jdcal	1.0	BSD	False
jinja2	2.7.3	BSD	False
jpeg $^{\text{Linux Mac}}$	8d	Custom free software license	False
libdynd $^{\text{Linux Mac}}$	0.6.5	BSD	False
libffi $^{\text{Linux}}$	3.0.13	MIT	False
libnetcdf $^{\text{Linux Mac}}$	4.3.2	MIT	False
libpng $^{\text{Linux Mac}}$	1.5.13	Open Source	False
libsodium $^{\text{Linux Mac}}$	0.4.5	MIT	False
libtiff $^{\text{Linux Mac}}$	4.0.2	as-is	False
libxml2 $^{\text{Linux Mac}}$	2.9.0	MIT	False
libxslt $^{\text{Linux Mac}}$	1.1.28	MIT	False
llvm $^{\text{Linux Mac}}$	3.3	Open Source	False
llvmpy	0.12.7	New BSD License	False
logilab-common	0.62.1	LGPL	False
lxml	3.4.0	BSD	False
markdown $^{\text{Linux Mac}}$	2.5	BSD	False
markupsafe	0.23	BSD	False
mathjax	2.2	Apache	False
matplotlib	1.4.0	PSF-based	False
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
mingw $^{\text{Windows}}$	4.7	GPL	False
mock	1.0.1	BSD	False
multimethods	1.0.0	MIT	False
multipledispatch	0.4.7	BSD	False
natsort	3.5.0	MIT	False
netcdf4	1.1.1	MIT	False
networkx	1.9.1	BSD	False
nltk	3.0.0	Apache 2.0	False
nose	1.3.4	LGPL	False
numba	0.14.0	numba license	False
numexpr	2.3.1	MIT	False
numpy	1.9.0	BSD	False
openpyxl	1.8.5	MIT/Expat	False
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	False

Continued on next page

Table 14 – continued from previous page

pandas	0.14.1	BSD	False
paramiko	1.14.1	LGPL	False
passlib	1.6.2	BSD	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.3.0	BSD License	False
pep8	1.5.7	MIT License	False
pillow $^{\text{Linux Mac}}$	2.5.1	Standard PIL license	False
pip	1.5.6	MIT	False
ply	3.4	BSD	False
psutil	2.1.1	BSD	False
py	1.4.25	MIT	False
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	False
pycparser	2.10	BSD	False
pycrypto	2.6.1	Public Domain	False
pycurl $^{\text{Linux Mac}}$	7.19.5	LGPL and MIT/X	False
pyflakes	0.8.1	MIT	False
pygments	1.6	BSD	False
pylint	1.3.1	GPL	False
pymongo	2.7.2	Apache 2.0	False
pyodbc	3.0.7	MIT	False
pyopenssl	0.14	APL2	False
pyparsing	2.0.1	MIT	False
pyqt	4.10.4	GPL	False
pyreadline $^{\text{Windows}}$	2.0	BSD	False
pyserial	2.7	PSF	False
pytables	3.1.1	BSD	False
pytest	2.6.3	MIT	False
python	3.3.5	PSF	False
pytz	2014.7	MIT	False
pywin32 $^{\text{Windows}}$	219	PSF	False
pyyaml	3.11	MIT	False
pyzmq	14.3.1	LGPL and BSD	False
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	False
queuelib	1.2.2	BSD	False
readline $^{\text{Linux Mac}}$	6.2	GPL 3	False
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	False
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	False
reportlab	3.1.8	BSD	False
repoze.lru	0.6	BSD	False
requests	2.4.1	ISC	False
rope	0.9.4	GPL	False
runipy	0.1.1	BSD	False
scikit-bio $^{\text{Linux Mac}}$	0.2.0	BSD	False
scikit-image	0.10.1	Modified BSD	False
scikit-learn	0.15.2	3-clause BSD	False
scipy	0.14.0	BSD	False
setuptools	5.8	PSF or ZPL	False
shapely $^{\text{Linux Mac}}$	1.4.1	BSD	False

Continued on next page

Table 14 – continued from previous page

sip Linux Mac	4.15.5	GPL	False
six	1.8.0	MIT	False
sockjs-tornado	1.0.1	MIT	False
sphinx	1.2.3	BSD	False
sqlalchemy	0.9.7	MIT	False
sqlite Linux Mac	3.8.4.1	Public Domain	False
sqlparse	0.1.12	BSD	False
statsmodels	0.5.0	3-clause Modified BSD	False
sympy	0.7.5	New BSD	False
tk Linux Mac	8.5.15	BSD-style	False
toolz	0.7.0	BSD	False
tornado	4.0.2	Apache	False
twisted	14.0.2	MIT	False
ujson	1.33	BSD	False
unixodbc Linux	2.3.1	???	False
util-linux Linux	2.21	GPL	False
venusian	1.0	BSD	False
w3lib	1.8.1	BSD	False
werkzeug	0.9.6	BSD	False
whoosh	2.5.7	BSD	False
xlrd	0.9.3	BSD	False
xlsxwriter	0.5.7	BSD	False
xz Linux Mac	5.0.5	Public Domain and GPL	False
yaml Linux Mac	0.1.4	MIT	False
zeromq Linux Mac	4.0.4	LGPL	False
zlib Linux Mac	1.2.7	zlib	False
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.1.0 for Python version 3.4

Python version: 3.4

Number of supported packages: 171

Name	Version	License	In Installer
abstract-rendering Linux Mac	0.5.1	3rd-clause BSD	True
argcomplete	0.8.1	Apache Software License	True
astroid	1.2.1	LGPL	False
astropy	0.4.2	BSD	True
basemap Linux Mac	1.0.7	PSF	False
bcolz	0.7.1	BSD	False
beautiful-soup	4.3.2	PSF/MIT	True
binstar	0.7.1	BSD	True
biopython	1.64	BSD-like	False
bitarray	0.8.1	PSF	True
blaze	0.6.3	BSD	True
blist	1.3.6	BSD	False
blz	0.6.2	BSD	True
bokeh	0.6.1	New BSD	True
boto	2.32.1	MIT	True

Continued on next page

Table 15 – continued from previous page

bsdiff4	1.1.4	BSD	False
cffi	0.8.6	MIT	True
chameleon	2.16	BSD-like	False
chrpath Linux	0.13	GPL	False
colorama	0.3.1	BSD	True
conda	3.7.0	BSD	True
conda-api	1.1.0	BSD	False
conda-build	1.8.2	BSD	True
configobj	5.0.6	BSD	True
coverage	3.7.1	BSD	False
cryptography	0.5.4	Apache 2.0	True
cssselect Linux Mac	0.9.1	BSD	False
curl Linux Mac	7.38.0	MIT/X derivate	True
cython	0.21	Apache 2.0	True
cytoolz	0.7.0	BSD	True
datashape	0.3.0	BSD	True
dateutil	2.1	BSD	True
decorator	3.4.0	BSD	True
django	1.7	BSD	False
docutils	0.12	Public-Domain, PSF, 2-clause BSD, GPL3	True
dynd-python	0.6.5	BSD	True
ecdsa	0.11	MIT	False
ephem	3.7.5.3	LGPL	False
feedparser	5.1.3	MIT	False
fiona	1.1.6	BSD	False
flake8	2.2.3	MIT	False
flask	0.10.1	BSD	True
freetype Linux Mac	2.4.10	FreeType License	True
future	0.13.1	MIT	True
futures	2.1.6	BSD	False
gdal	1.11.0	MIT	False
gensim	0.10.2	LGPL	False
geos Linux Mac	3.3.3	LGPL	False
greenlet	0.4.4	MIT	True
gunicorn Linux Mac	19.1.0	MIT	False
h5py	2.3.1	New BSD	True
hdf5 Linux Mac	1.8.13	BSD-style	True
html5lib	0.999	MIT	False
ipython	2.2.0	BSD	True
itsdangerous	0.24	BSD License	True
jdcal	1.0	BSD	True
jinja2	2.7.3	BSD	True
jpeg Linux Mac	8d	Custom free software license	True
launcher Mac Windows	1.0.0	proprietary - Continuum Analytics, Inc.	True
libdynd Linux Mac	0.6.5	BSD	True
libffi Linux	3.0.13	MIT	True
libnetcdf Linux Mac	4.3.2	MIT	False
libpng Linux Mac	1.5.13	Open Source	True
libsodium Linux Mac	0.4.5	MIT	True
libtiff Linux Mac	4.0.2	as-is	True

Continued on next page

Table 15 – continued from previous page

libxml2 $^{\text{Linux Mac}}$	2.9.0	MIT	True
libxslt $^{\text{Linux Mac}}$	1.1.28	MIT	True
llvm $^{\text{Linux Mac}}$	3.3	Open Source	True
llvmpy	0.12.7	New BSD License	True
logilab-common	0.62.1	LGPL	False
lxml	3.4.0	BSD	True
markdown $^{\text{Linux Mac}}$	2.5	BSD	False
markupsafe	0.23	BSD	True
mathjax	2.2	Apache	False
matplotlib	1.4.0	PSF-based	True
mccabe	0.2.1	Expat	False
mdp	3.3	BSD	False
menuinst $^{\text{Windows}}$	1.0.4	BDF	True
mingw $^{\text{Windows}}$	4.7	GPL	True
mock	1.0.1	BSD	True
multimethods	1.0.0	MIT	False
multipledispatch	0.4.7	BSD	True
natsort	3.5.0	MIT	False
netcdf4	1.1.1	MIT	False
networkx	1.9.1	BSD	True
nltk	3.0.0	Apache 2.0	True
node-webkit $^{\text{Mac Windows}}$	0.10.1	MIT	True
nose	1.3.4	LGPL	True
numba	0.14.0	numba license	True
numexpr	2.3.1	MIT	True
numpy	1.9.0	BSD	True
openpyxl	1.8.5	MIT/Expat	True
openssl $^{\text{Linux Mac}}$	1.0.1h	Apache-style	True
pandas	0.14.1	BSD	True
paramiko	1.14.1	LGPL	False
passlib	1.6.2	BSD	False
pastedeploy $^{\text{Linux Mac}}$	1.5.2	MIT	False
patchelf $^{\text{Linux}}$	0.6	GPL3	False
patsy	0.3.0	BSD License	True
pep8	1.5.7	MIT License	False
pillow $^{\text{Linux Mac}}$	2.5.1	Standard PIL license	True
pip	1.5.6	MIT	True
ply	3.4	BSD	True
psutil	2.1.1	BSD	True
py	1.4.25	MIT	True
pyasn1 $^{\text{Linux}}$	0.1.6	BSD	False
pycosat	0.6.1	MIT	True
pycparser	2.10	BSD	True
pycrypto	2.6.1	Public Domain	True
pycurl $^{\text{Linux Mac}}$	7.19.5	LGPL and MIT/X	True
pyflakes	0.8.1	MIT	True
pygments	1.6	BSD	True
pylint	1.3.1	GPL	False
pymongo	2.7.2	Apache 2.0	False
pyodbc	3.0.7	MIT	False

Continued on next page

Table 15 – continued from previous page

pyopenssl	0.14	APL2	True
pyparsing	2.0.1	MIT	True
pyqt	4.10.4	GPL	True
pyreadline $^{\text{Windows}}$	2.0	BSD	True
pyserial	2.7	PSF	False
pytables	3.1.1	BSD	True
pytest	2.6.3	MIT	True
python	3.4.1	PSF	True
pytz	2014.7	MIT	True
pywin32 $^{\text{Windows}}$	219	PSF	True
pyyaml	3.11	MIT	True
pyzmq	14.3.1	LGPL and BSD	True
qt $^{\text{Linux Mac}}$	4.8.5	LGPL	True
queuelib	1.2.2	BSD	False
readline $^{\text{Linux Mac}}$	6.2	GPL 3	True
redis $^{\text{Linux Mac}}$	2.6.9	3-clause BSD	True
redis-py $^{\text{Linux Mac}}$	2.9.1	MIT	True
reportlab	3.1.8	BSD	False
repoze.lru	0.6	BSD	False
requests	2.4.1	ISC	True
rope	0.9.4	GPL	True
runipy	0.1.1	BSD	True
scikit-bio $^{\text{Linux Mac}}$	0.2.0	BSD	False
scikit-image	0.10.1	Modified BSD	True
scikit-learn	0.15.2	3-clause BSD	True
scipy	0.14.0	BSD	True
setuptools	5.8	PSF or ZPL	True
shapely $^{\text{Linux Mac}}$	1.4.1	BSD	False
sip $^{\text{Linux Mac}}$	4.15.5	GPL	True
six	1.8.0	MIT	True
sockjs-tornado	1.0.1	MIT	True
sphinx	1.2.3	BSD	True
spyder	2.3.1	MIT	True
sqlalchemy	0.9.7	MIT	True
sqlite $^{\text{Linux Mac}}$	3.8.4.1	Public Domain	True
sqlparse	0.1.12	BSD	False
statsmodels	0.5.0	3-clause Modified BSD	True
sympy	0.7.5	New BSD	True
tk $^{\text{Linux Mac}}$	8.5.15	BSD-style	True
toolz	0.7.0	BSD	True
tornado	4.0.2	Apache	True
twisted	14.0.2	MIT	False
ujson	1.33	BSD	True
unixodbc $^{\text{Linux}}$	2.3.1	???	False
util-linux $^{\text{Linux}}$	2.21	GPL	True
venusian	1.0	BSD	False
w3lib	1.8.1	BSD	False
werkzeug	0.9.6	BSD	True
whoosh	2.5.7	BSD	False
xlrd	0.9.3	BSD	True

Continued on next page

Table 15 – continued from previous page

xlsxwriter	0.5.7	BSD	True
xz $^{\text{Linux Mac}}$	5.0.5	Public Domain and GPL	True
yaml $^{\text{Linux Mac}}$	0.1.4	MIT	True
zeromq $^{\text{Linux Mac}}$	4.0.4	LGPL	True
zlib $^{\text{Linux Mac}}$	1.2.7	zlib	True
zope.interface	4.1.1	Zope Public License	False

Packages included in Anaconda 2.2.0 for Python version 2.6

Packages included in Anaconda 2.2.0 for Python version 2.7

Packages included in Anaconda 2.2.0 for Python version 3.3

Packages included in Anaconda 2.2.0 for Python version 3.4

Packages included in Anaconda 2.3.0 for Python version 2.6

Packages included in Anaconda 2.3.0 for Python version 2.7

Packages included in Anaconda 2.3.0 for Python version 3.3

Packages included in Anaconda 2.3.0 for Python version 3.4

Packages included in Anaconda 2.4.0 for Python version 2.7

orphan

Packages included in Anaconda 2.4.0 for Python version 3.4

orphan

Packages included in Anaconda 2.4.0 for Python version 3.5

orphan

Packages included in Anaconda 2.4.1 for Python version 2.7

orphan

Packages included in Anaconda 2.4.1 for Python version 3.4

orphan

Packages included in Anaconda 2.4.1 for Python version 3.5

orphan

Packages included in Anaconda 2.5.0 for Python version 2.7

orphan

Packages included in Anaconda 2.5.0 for Python version 3.4

orphan

Packages included in Anaconda 2.5.0 for Python version 3.5

orphan

Packages included in Anaconda 4.0.0 for Python version 2.7

orphan

Packages included in Anaconda 4.0.0 for Python version 3.4

orphan

Packages included in Anaconda 4.0.0 for Python version 3.5

orphan

Packages included in Anaconda 4.1.0 for Python version 2.7

orphan

Packages included in Anaconda 4.1.0 for Python version 3.4

orphan

Packages included in Anaconda 4.1.0 for Python version 3.5

orphan

Packages included in Anaconda 4.1.1 for Python version 2.7

orphan

Packages included in Anaconda 4.1.1 for Python version 3.4

orphan

Packages included in Anaconda 4.1.1 for Python version 3.5

orphan

Packages included in Anaconda 4.2.0 for Python version 2.7

orphan

Packages included in Anaconda 4.2.0 for Python version 3.4

orphan

Packages included in Anaconda 4.2.0 for Python version 3.5

orphan

Packages included in Anaconda 4.3.0 for Python version 2.7

orphan

Packages included in Anaconda 4.3.0 for Python version 3.4

orphan

Packages included in Anaconda 4.3.0 for Python version 3.5

orphan

Packages included in Anaconda 4.3.0 for Python version 3.6

orphan

Packages included in Anaconda 4.3.1 for Python version 2.7

orphan

Packages included in Anaconda 4.3.1 for Python version 3.4

orphan

Packages included in Anaconda 4.3.1 for Python version 3.5

orphan

Packages included in Anaconda 4.3.1 for Python version 3.6

orphan

Packages included in Anaconda 4.4.0 for Python version 2.7

Packages included in Anaconda 4.4.0 for Python version 3.5

Packages included in Anaconda 4.4.0 for Python version 3.6

Packages included in Anaconda 5.0.0 for 32-bit Linux with Python 2.7

Packages included in Anaconda 5.0.0 for 64-bit Linux with Python 2.7

Packages included in Anaconda 5.0.0 for 64-bit Linux on IBM Power CPUs with Python 2.7

Packages included in Anaconda 5.0.0 for macOS with Python 2.7

Packages included in Anaconda 5.0.0 for 32-bit Windows with Python 2.7

Packages included in Anaconda 5.0.0 for 64-bit Windows with Python 2.7

Packages included in Anaconda 5.0.0 for 32-bit Linux with Python 3.5

Packages included in Anaconda 5.0.0 for 64-bit Linux with Python 3.5

Packages included in Anaconda 5.0.0 for 64-bit Linux on IBM Power CPUs with Python 3.5

Packages included in Anaconda 5.0.0 for macOS with Python 3.5

Packages included in Anaconda 5.0.0 for 32-bit Windows with Python 3.5

Packages included in Anaconda 5.0.0 for 64-bit Windows with Python 3.5

Packages included in Anaconda 5.0.0 for 32-bit Linux with Python 3.6

Packages included in Anaconda 5.0.0 for 64-bit Linux with Python 3.6

Packages included in Anaconda 5.0.0 for 64-bit Linux on IBM Power CPUs with Python 3.6

Packages included in Anaconda 5.0.0 for macOS with Python 3.6

Packages included in Anaconda 5.0.0 for 32-bit Windows with Python 3.6

Packages included in Anaconda 5.0.0 for 64-bit Windows with Python 3.6

Packages included in Anaconda 5.0.1 for 32-bit Linux with Python 2.7

Packages included in Anaconda 5.0.1 for 64-bit Linux with Python 2.7

Packages included in Anaconda 5.0.1 for 64-bit Linux on IBM Power CPUs with Python 2.7

Packages included in Anaconda 5.0.1 for macOS with Python 2.7

Packages included in Anaconda 5.0.1 for 32-bit Windows with Python 2.7

is right for you to manage your packages and environments. You can even switch between them, and the work you do with one can be viewed in the other.

Now, try this simple programming exercise two ways, with Navigator and a terminal, to help you decide which approach is right for you.

Your first Python program: Hello, Anaconda!

Write and run a Python program using Anaconda Navigator.

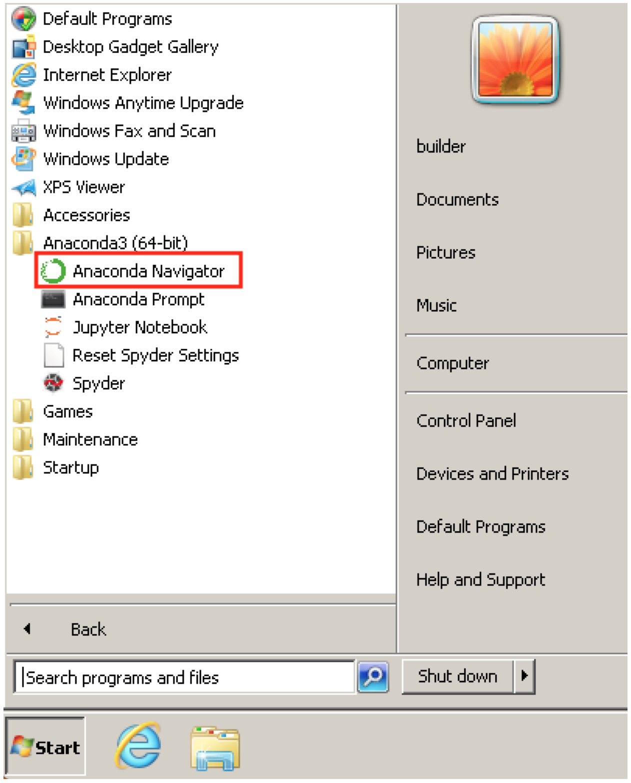
1. Open Navigator

Choose the instructions for your operating system.

- *Windows*
- *macOS*
- *Linux*

Windows

From the Start menu, click the Anaconda Navigator desktop app.



macOS

Open Launchpad, then click the Anaconda Navigator icon.

Linux

Open a Terminal window and type `anaconda-navigator`.

2. Run Python in Spyder IDE (integrated development environment)

TIP: Navigator's Home screen displays several applications for you to choose from. For more information, see links at bottom of this page.

On Navigator's Home tab, in the Applications pane on the right, scroll to the Spyder tile and click the Install button to install Spyder.

NOTE: If you already have Spyder installed, you can jump right to the Launch step.

Launch Spyder by clicking Spyder's Launch button.

In the new file on the left, delete any placeholder text, then type or copy/paste `print("Hello Anaconda")`.

In the top menu, click File - Save As and name your new program `hello.py`.

Run your new program by clicking the triangle Run button.

You can see your program's output in the bottom right Console pane.

3. Close Spyder

From Spyder's top menu bar, select Spyder - Quit Spyder (In macOS, select Python - Quit Spyder).

4. Close Navigator

From Navigator's top menu bar, select Anaconda Navigator - Quit Anaconda-Navigator.

Write a Python program using Anaconda Prompt or Terminal

1. Open Anaconda Prompt

Choose the instructions for your operating system.

- *Windows*
- *macOS*
- *Linux*

Windows

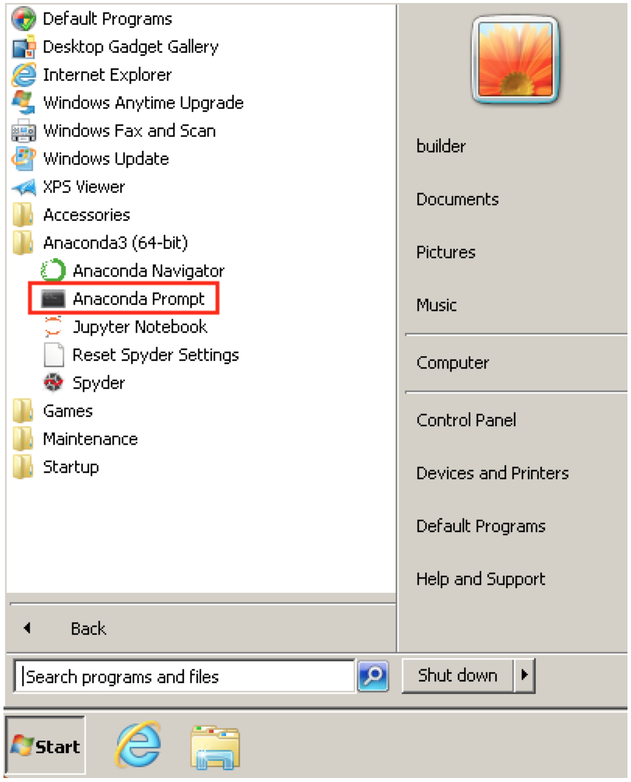
From the Start menu, search for and open "Anaconda Prompt":

macOS

Open Launchpad, then click the Terminal icon.

Linux

Open a Terminal window.



2. Start Python

At Anaconda Prompt (Terminal on Linux or macOS), type `python` and press Enter.

The >>> means you are in Python.

3. Write a Python program

At the >>>, type `print("Hello Anaconda!")` and press Enter.

When you press enter, your program runs. The words "Hello Anaconda!" print to the screen. You're programming in Python!

4. Exit Python

On Windows press CTRL-Z and press Enter. On macOS or Linux type exit() and press Enter.

What's next?

Using Navigator

- Getting started with Navigator (10 minutes)
- Navigator user guide

Using conda

- Getting started with conda (20 minutes)
- Conda cheat sheet (pdf)
- Conda user guide

Links to IDE documentation

- *Eclipse and PyDev*
- *IDLE*
- *Sublime Text*
- *Ninja IDE*
- *Python Tools for Visual Studio (PTVS)*
- *Python for Visual Studio Code*
- *Spyder*
- *Wing IDE*
- *IntelliJ*

Tasks

Installing licenses

You can install licenses for *Accelerate*, *IOPro* or *MKL Optimizations* with the graphical *Anaconda Navigator license manager*, and you can install licenses for these and all other products *from the command line*.

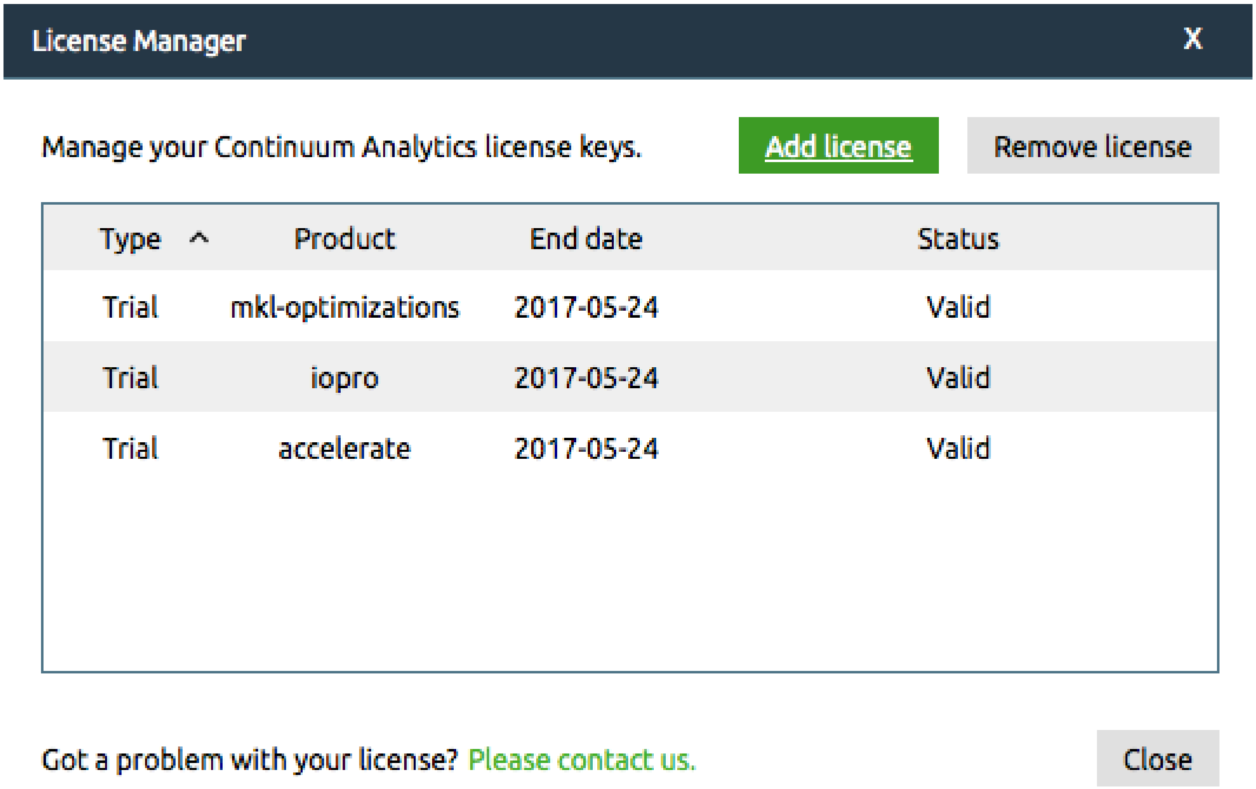
Before you start

- You should have installed conda and received your license by email from your sales representative.
- If want to use Navigator, it needs to be installed. Both Navigator and conda are included in Anaconda.
- Optionally, your products or a free trial may already be installed.
- Save the license file to your `Downloads` folder (or directory). To save a few steps, make a note of the filename now.
- If you need help, contact your sales rep or priority support.

Using Navigator to install a license

The graphical Navigator license manager allows you to add, remove and view details of licensed packages.

1. Open *Navigator*.
2. In the **Help** menu, select License Manager.
3. Click the Add license button, and then select the license file:



Manage your Continuum Analytics license keys.

Type ^	Product	End date	Status
Trial	mkl-optimizations	2017-05-24	Valid
Trial	iopro	2017-05-24	Valid
Trial	accelerate	2017-05-24	Valid

Got a problem with your license? Please contact us.

4. Click the Close button.
5. After you install the license, *update conda and install the product*.
6. *Run the product*.

If you have questions about or problems with a license, click the Got a problem with your license? link, contact your sales rep or contact priority support for assistance.

Using the command line to install a license

Licenses for all Anaconda user products are stored in the home folder in a hidden subfolder named `~/.continuum`. The license directory is hidden to help protect your license files from accidental changes or deletion. The following installation procedures describe how to manage hidden directories for your operating system.

TIP: The conda command `conda info --license` shows you the location of your license files and validates those licenses.

Command line procedures:

- *Windows*.
- *macOS*.
- *Linux*.
- *Updating conda and installing a product–all operating systems*.

- *Running your new packages or products–all operating systems*.
- *If you have problems*.

Windows

Hidden folders appear in Windows Explorer only if the Show hidden files and folders option is enabled. To bypass this issue, use the Anaconda prompt, and if the hidden `.continuum` folder has not been created yet, make one as discussed below. Then copy the license file into the hidden license folder.

1. Open a terminal window.
2. To see the files in your home folder, run the command `dir`.
3. In the output list, look for `.continuum`.

   If you do not see a folder named `.continuum`, make a new one with `mkdir .continuum`.
4. Find the license file in your `Downloads` folder or other folder where you saved the file:

   ```
   dir Downloads
   ```

   NOTE: Replace `Downloads` with the name of the folder where you saved your license file.
5. Copy the license file into the `.continuum` directory:

   ```
   copy Downloads\license_bundle_1234.txt .continuum
   ```

   NOTE: Replace `license_bundle_1234.txt` with the name of your license file.
6. *Update conda and install the product*.
7. *Run the product*.

Your license is now installed on Windows. If you have any questions or problems, please contact your sales rep or priority support for assistance.

macOS

Use Finder to locate the hidden licenses folder. If the folder has not been created yet, make one as discussed below. Then copy and paste the license into the hidden folder.

1. Look for the hidden folder:
   (a) In the bottom row of icons, click the Finder icon.

   (b) In the **Go** menu, select Go to folder.

   (c) In the box that appears, type `~/.continuum`, and then click the Go button.

   (d) If the search results listing is empty, make a new hidden licenses folder by selecting New Folder in the **File** menu and then typing `~/.continuum`.
2. Copy the license file:
   (a) Go to your `Downloads` folder or wherever you saved the license file.

   (b) Highlight the license filename.

   (c) In the **Edit** menu, select Copy File.
3. Return to the licenses directory:

   (a) In the **Go** menu, select Go to folder.

   (b) In the box that appears, type `~/.continuum`, and then click the Go Button.

4. Paste in the license file:

   (a) In the file listing, double-click the `~/.continuum` folder to open the folder.

   (b) In the top menu bar, in the **Edit** menu, select Paste Item.

5. *Update conda and install the product*.

6. *Run the product*.

Your license is now installed on macOS. If you have any questions or problems, please contact your sales rep or priority support for assistance.

Linux

Use the List-All command to display all directories including hidden ones, and if a license directory has not been created yet, make one as discussed below. Then copy the license into the hidden license directory.

1. Open a terminal window.

2. To see the files in your home folder, run the command `ls -a`.

3. In the output list, look for `.continuum`.

   If you do not see a folder named `.continuum`, make a new one with `mkdir .continuum`.

4. Find the license file in your `Downloads` folder or other folder where you saved the file:

   ```
   cd Downloads
   ls
   ```

   NOTE: Replace `Downloads` with the name of the folder where you saved your license file.

   If the list is long run:

   ```
   ls | more
   ```

5. Highlight the filename, and then in the **Edit** menu, select Copy.

6. Copy the license file into the `.continuum` directory:

   ```
   cp license_bundle_1234.txt ~/.continuum
   ```

   NOTE: Replace `license_bundle_1234.txt` with the name of your license file.

   TIP: Instead of typing the file name, you can paste it into the command:

   (a) Type `cp`.

   (b) In the **Edit** menu, select Paste to paste in the license filename.

   (c) Type `~/.continuum`.

   (d) Press Enter.

7. *Update conda and install the product*.

8. *Run the product*.

Your license is now installed on Linux. If you have any questions or problems, please contact your sales rep or priority support for assistance.

Updating conda and installing a product–all operating systems

NOTE: If you previously installed a 30-day trial, you may skip this procedure, but running it again is harmless.

To update conda and install a product:

1. In a terminal window, run:

   ```
   conda update conda
   conda install your_package_name
   ```

   NOTE: Replace `your_package_name` with the actual name of your package or product, such as `accelerate`. Follow the full installation instructions for the product.

Running your new packages or products–all operating systems

1. Start Python:

   ```
   python
   ```

   If you have multiple versions of Python on your computer, make sure that the startup message says it is Anaconda Python.

2. Import your newly licensed packages or products:

   ```
   import your_package_name
   ```

   NOTE: Replace `your_package_name` with the actual name of your package or product, such as `accelerate`. Follow the full installation instructions for the product.

If you have problems

If you have any questions or problems, please contact your sales rep or priority support for assistance.

Installing conda packages

For more information about using the conda package manager in Anaconda Prompt (Terminal on Linux or macOS), see the conda documentation.

You can also use the graphical interface *Anaconda Navigator* to install conda packages with just a few clicks.

Open an Anaconda Prompt (Terminal on Linux or macOS) and follow these instructions.

Installing a conda package

Enter the command:

```
conda install package-name
```

NOTE: Replace `package-name` with your package name.

Installing specific versions of conda packages

Include the desired version number or its prefix after the package name:

```
conda install package-name=2.3.4
```

NOTE: Replace `package-name` with your package name. Replace `2.3.4` with the desired version number.

To specify only a major version, run:

```
conda install package-name=2
```

NOTE: Replace `package-name` with your package name. Replace `2` with the desired version number.

These commands install into the environment that is currently active. To install into a named environment, run:

```
conda install package-name=2.3.4 -n some-environment
```

NOTE: Replace `package-name` with your package name. Replace `2.3.4` with the desired version number. Replace `some-environment` with your environment name.

If the package is specific to a Python version, conda uses the version installed in the current or named environment. For details on versions, dependencies and channels, see Conda FAQ and Conda Troubleshooting.

Installing packages on a non-networked (air gapped) computer

To directly install a conda package from your local computer, run:

```
conda install /package-path/package-filename.tar.bz2
```

NOTE: Replace `package-path` and `package-filename.tar.bz2` with your actual path and filename.

Conda installs packages into the `anaconda/pkgs` directory.

To install a .tar file containing many conda packages, run the following command:

```
conda install /packages-path/packages-filename.tar
```

NOTE: Replace `package-path` and `package-filename.tar` with your actual path and filename.

If conda cannot find the file, try using an absolute path name instead of a relative path name.

NOTE: Installing packages directly from the file does not resolve dependencies. If your installed package does not work, it may have missing dependencies that need to be resolved manually.

Using Jupyter Notebook extensions

You can open Jupyter Notebook by running `jupyter notebook`, or by opening Anaconda Navigator and clicking the Jupyter Notebook icon.

With Anaconda you can download and install four extensions for the Jupyter Notebook that make the notebook easier to use:

- Notebook Present (nbpresent).
- Notebook Anaconda Cloud (nb_anacondacloud).
- Notebook Conda (nb_conda).
- Notebook Conda Kernels (nb_conda_kernels).

Installing any of the four installs all of them. The _nb_ext_conf package is also installed, which activates the extensions.

Obtaining the extensions

To get the extensions using Anaconda Navigator:

1. Install and manage notebook extensions packages like any other packages. See *User guide*.
2. To use the new extensions, on the Navigator **Home** tab, open Jupyter Notebook.

To install all Jupyter Notebook extensions from the command line, run:

```
conda install nb_conda
```

NOTE: These extensions were already installed in Anaconda versions 4.1 and 4.2. If you have Anaconda v4.1 or v4.2 installed, there is no need to install them separately. To begin using them, open a new or existing notebook.

Uninstalling the extensions

To remove all Jupyter Notebook extensions, run:

```
conda remove nb_conda
```

Uninstalling `nb_conda` or any other one of the four extensions uninstalls all four.

To disable Jupyter Notebook extensions individually without uninstalling them, run:

```
python -m nb_conda_kernels.install --disable --prefix=<ENV_PREFIX>

jupyter-nbextension disable nb_conda --py --sys-prefix
jupyter-serverextension disable nb_conda --py --sys-prefix

jupyter-nbextension disable nb_anacondacloud --py --sys-prefix
jupyter-serverextension disable nb_anacondacloud --py --sys-prefix

jupyter-nbextension disable nbpresent --py --sys-prefix
jupyter-serverextension disable nbpresent --py --sys-prefix
```

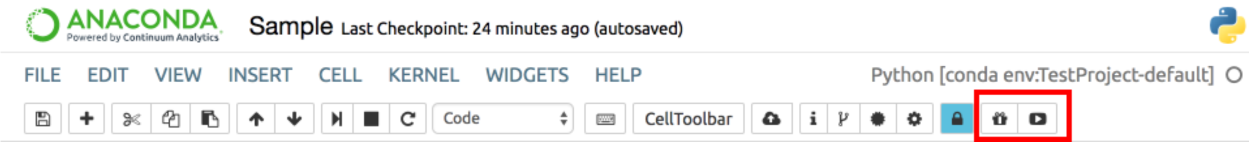
NOTE: Replace `<ENV_PREFIX>` with your root environment or another conda environment where the extensions have been installed.

Notebook Present

This extension turns your notebook into a PowerPoint-style presentation.

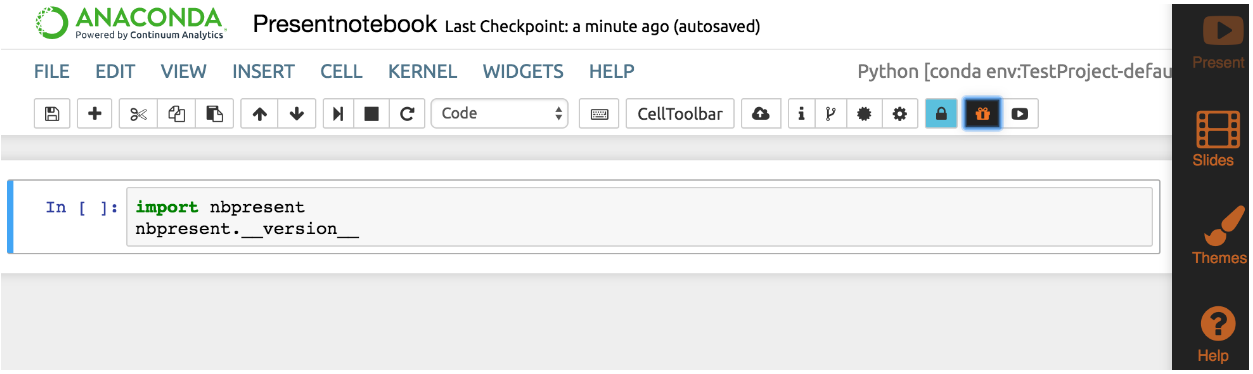
To begin using Notebook Present:

1. In the top menu bar, locate the two buttons Edit Presentation and Show Presentation:


ANACONDA
Powered by Continuum Analytics
Sample Last Checkpoint: 24 minutes ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-default]
Code
CellToolbar

2. Click the Edit Presentation button. The icon looks like a present in a box with wrapping paper and a bow on top.

   A black bar with several icons appears on the right side of your browser. As you click each icon, the layout of your screen changes:


ANACONDA
Powered by Continuum Analytics
Presentnotebook Last Checkpoint: a minute ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-defau
Code
CellToolbar
Present
Slides
Themes
Help

```
In [ ]: import nbpresent
        nbpresent.__version__
```

3. Click the Help icon to view three quick tours of the main features of Notebook Present:

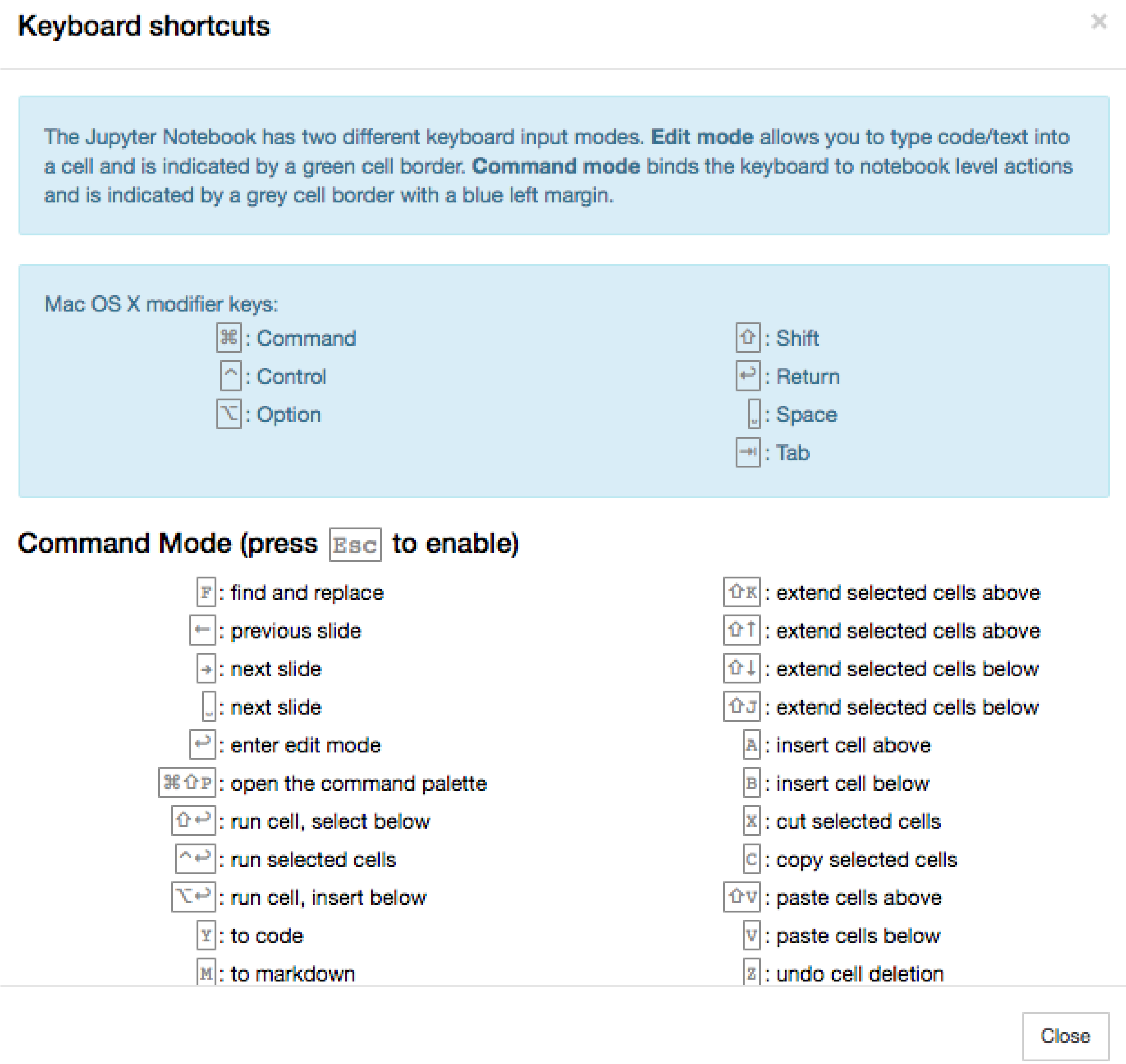
   - Intro Tour.
   - Slides Tour.
   - Editor Tour.

Intro tour

To see a 2-minute presentation on how to use all of the main features, in the **Help** menu, select Intro.

Every button is explained. You can pause, go back to the previous slide or advance to the next slide. Here is a summary of the presentation:

- App Bar–When Authoring, use the App Bar to control the content and style of your presentation. It also activates several special editing keyboard shortcuts:



Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level actions and is indicated by a grey cell border with a blue left margin.

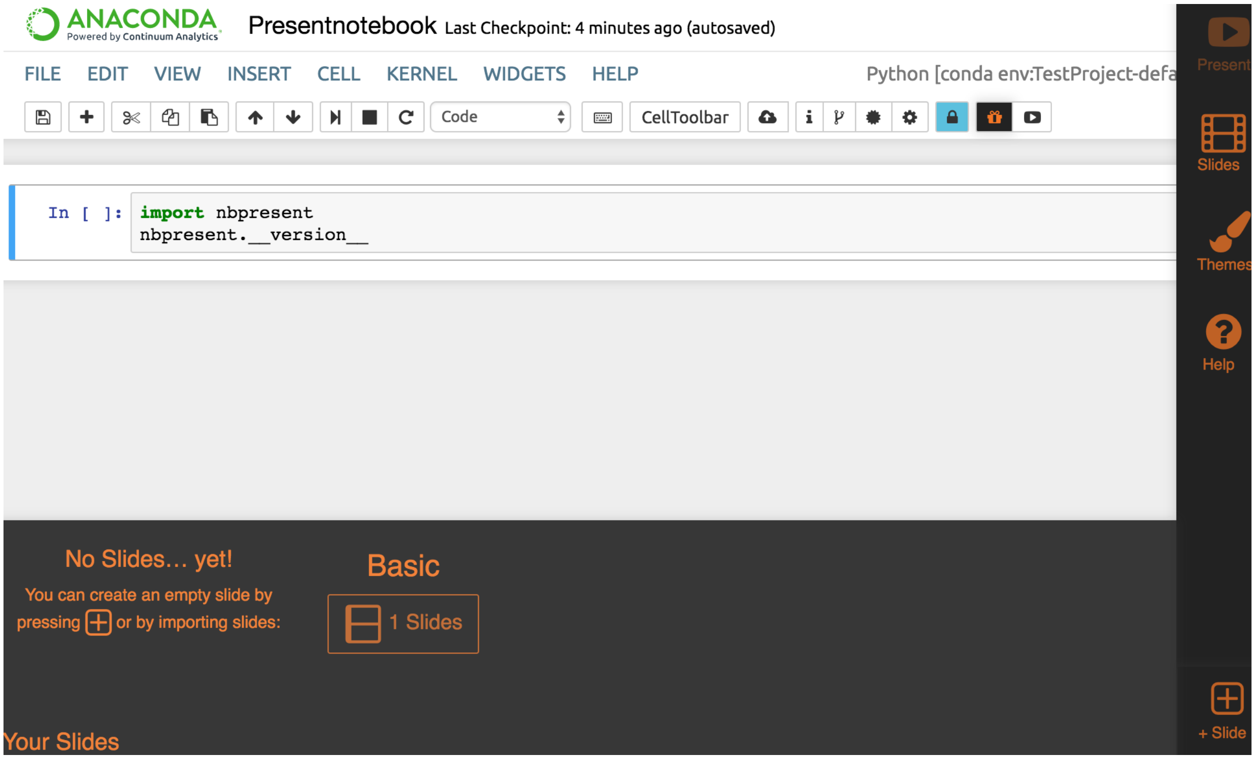
Mac OS X modifier keys:

⌘	: Command	⇧	: Shift
^	: Control	↩	: Return
⌥	: Option	␣	: Space
		⇥	: Tab

Command Mode (press Esc to enable)

F	: find and replace	⇧K	: extend selected cells above
←	: previous slide	⇧↑	: extend selected cells above
→	: next slide	⇧↓	: extend selected cells below
␣	: next slide	⇧J	: extend selected cells below
↩	: enter edit mode	A	: insert cell above
⌘⇧P	: open the command palette	B	: insert cell below
⇧↩	: run cell, select below	X	: cut selected cells
^↩	: run selected cells	C	: copy selected cells
⌥↩	: run cell, insert below	⇧V	: paste cells above
Y	: to code	V	: paste cells below
M	: to markdown	Z	: undo cell deletion

- Stop Authoring–Clicking the Edit Presentation button again stops Authoring and removes all keyboard shortcuts.
- Show Presentation–If you just want to run your presentation without using any Authoring tools, click the Show Presentation button.
- Presenting/Authoring–Once you have made some slides, start Presenting, where you can use most Notebook functions with the Theme we have defined, as well as customize slides on the fly.
- Slides button–Slides, which are made of Regions linked to Cell Parts, and can be imported, created, linked, reordered and edited here:

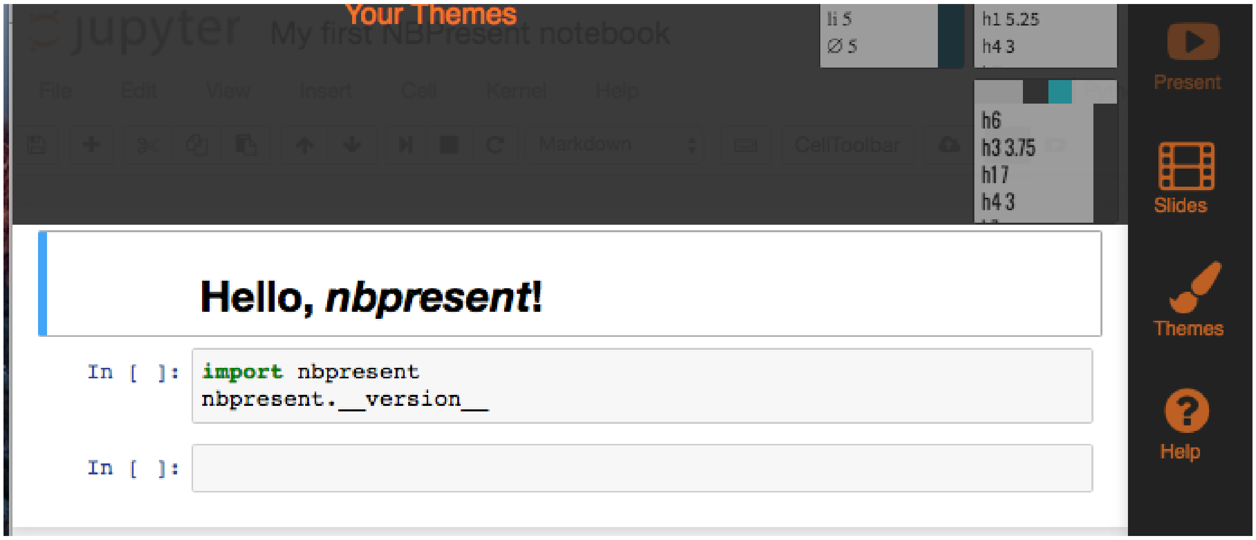

ANACONDA
Powered by Continuum Analytics
Presentnotebook Last Checkpoint: 4 minutes ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
WIDGETS
HELP
Python [conda env:TestProject-defa
Code
CellToolbar
Present
Slides
Themes
Help
+ Slide
Basic
1 Slides
Your Slides

```
In [ ]: import nbpresent
        nbpresent.__version__
```

No Slides… yet!

You can create an empty slide by pressing ⊞ or by importing slides:

- Theming–Theming lets you select from existing colors, typography and backgrounds to make distinctive presentations. The first theme you select becomes the default, while you can choose custom themes for a particular slide, such as a title:
- Saving–Whenever you save your Notebook, all your presentation data will be stored in the Notebook .ipynb file.
- Downloading–After you've made a presentation, you can download it as an HTML page. In the menu, select Download, and then select Download As: Presentation (.html).



Hello, *nbpresent*!

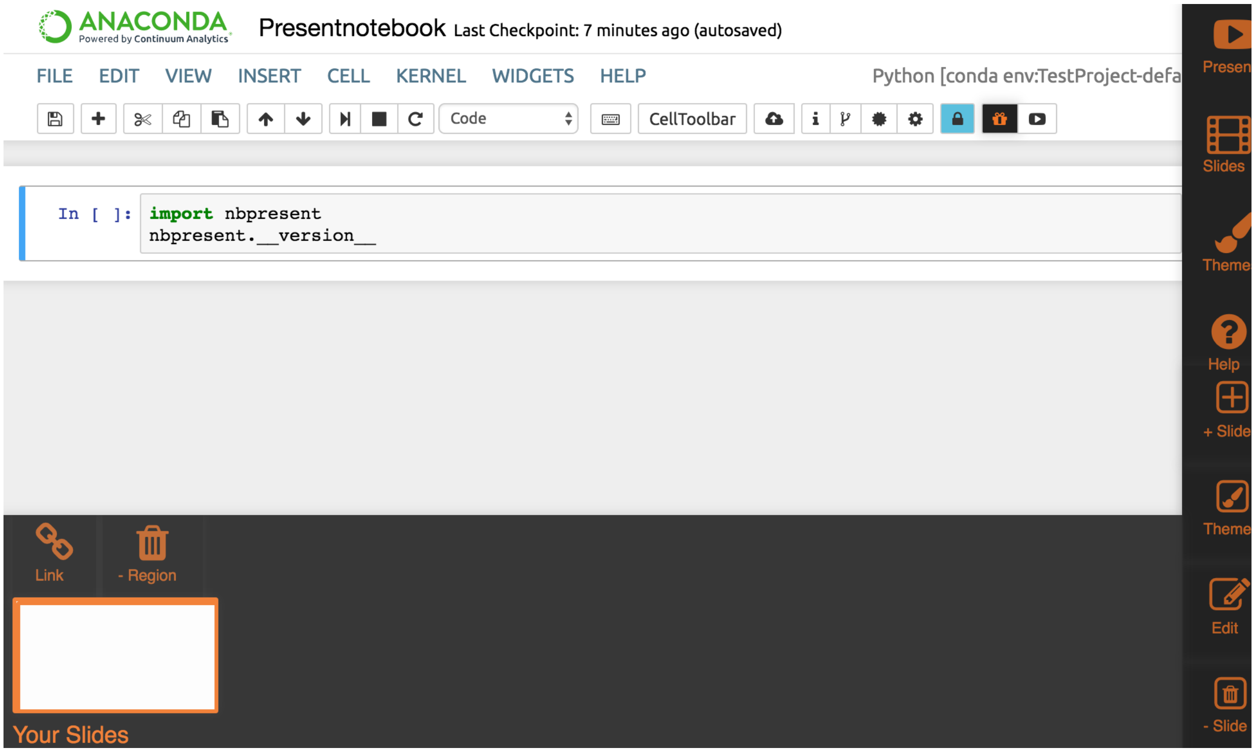
```
In [ ]: import nbpresent
        nbpresent.__version__

In [ ]:
```

- Help–Activate Help at any time to try other tours, obtain other information and connect with the Present community.

Slides Tour

To see a 2-minute presentation on how to create and manage slides, in the **Help** menu, click Slides.

Here is a summary of the presentation:


ANACONDA
Powered by Continuum Analytics
Presentnotebook Last Checkpoint: 7 minutes ago (autosaved)
FILE EDIT VIEW INSERT CELL KERNEL WIDGETS HELP
Python [conda env:TestProject-defa
Code
CellToolbar
Presen
Slides
Theme
Help
+ Slide
Theme
Edit
- Slide
Link
- Region
Your Slides

```
In [ ]: import nbpresent
        nbpresent.__version__
```

- Slides make up a presentation. Clicking Slides toggles the sorter view and the Slide Toolbar.
- Slide Toolbar–Let's create a new slide. Clicking the + Slide offers some choices for creating your new slide.
- Import–The quickest way to create a presentation is to import every complete cell as a slide. If you've already created slides with the official slideshow cell toolbar or RISE, you can import most of the content.
- Template Library–You can also pick from some existing templates.
- Reuse Slide as Template–You can copy an existing slide.
- Simple Template–A common template is the Quad Chart, with four pieces of content arranged in a grid.
- Region–The Quad Chart has four Regions. Select a region by clicking it.
- Linking a Region to a Cell Part–Each Region can be linked to a single Cell Part.
- Link Overlay–The Link Overlay shows all of the parts available.
- Cell Part: Source (blue)–Source, such as code and Markdown text.
- Cell Part: Outputs (red)–Outputs, such as rich figures and script results.
- Cell Part: Widgets (purple)–Jupyter widgets, which are interactive widgets that provide both visualization and user input.
- Cell Part: Whole (orange)–Finally, a Whole Cell (including its Source, Widgets, and Outputs) can be linked to a single region.
- Part Thumbnail–We'll try to draw a part thumbnail. It can be reliably updated only when a linked Cell Part is displayed when you mouse over it, but you should usually be able to get an idea of what you're seeing. The colors of the regions correspond to the cell types.
- Cell Part: Unlinking–Unlinking removes the connection between a region and a cell part, without deleting either one.
- Region: Trashing–Trashing a Region permanently deletes it, without affecting any linked Cell Part.
- Presenting–Clicking the Present button while editing brings up the Presenter with editing mode still enabled.
- It's still a Notebook–Linked inputs and widgets are still interactive.
- Go forward–Click to go to the next slide.
- Go back–Click to go back to the previous slide.
- Go back to the beginning–Click here to go back to the first slide.
- My work is done here–Click to go back to the Notebook.

Editor Tour

To see a 2-minute presentation on editing your notebook, in the **Help** menu, click Editor.

Here is a summary of the presentation:

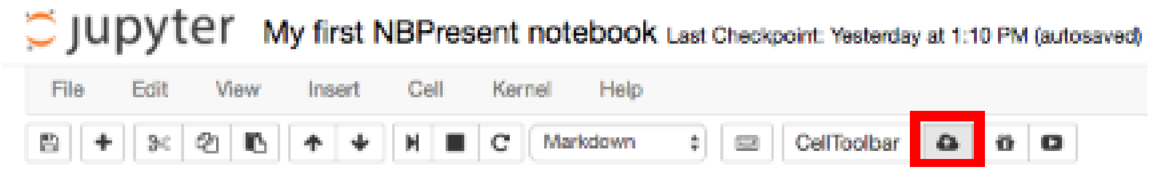
- So You Made Some Slides–Once you've made a few slides, you'll likely need to customize them.
- Editing Slides–Once you have selected a slide, you can activate the Slide Editor by double-clicking the slide or clicking Edit Slide.
- Region Editor–You can click and drag Regions around and resize them.
- Region Tree–You can reorder Regions and see the details of how your Regions show their linked Parts.
- Add Region–You can add new regions.

- Attribute Editor–All of the properties of a region can be edited here.
- Data Layouts–In addition to manually moving regions around, you can use other Layouts, such as a Treemap, which fills the slide.
- More Regions–More regions will be added with a weight of 1.
- Tree Weight–This new value lets you make a Region bigger or smaller based on relative Weight.
- 12 Grid–The Grid is a compromise between Free layout and Treemap layout, and rounds all the values to a factor of 12.

Notebook Anaconda Cloud

You must have an Anaconda Cloud account for this extension to work. You can sign up for a free account at Anaconda Cloud.

1. You can upload your notebook to your Cloud account with a simple button push:


jupyter
My first NBPresent notebook Last Checkpoint: Yesterday at 1:10 PM (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Markdown
CellToolbar

   You can use the Attach conda environment option described below to embed a copy of your conda environment as an `environment.yaml` file in the notebook metadata.

2. Sign in to Cloud:

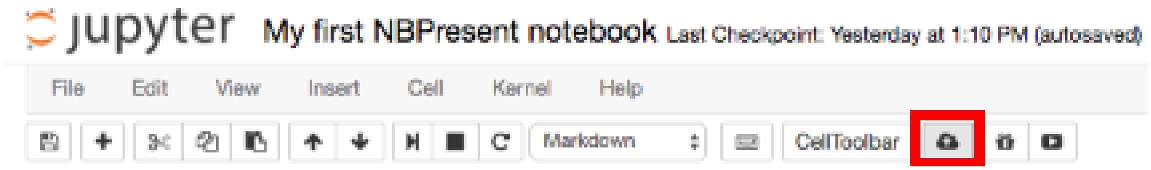
   - If you are not signed in to Cloud, a dialog box appears asking for your Cloud user name and password.
   - You may instead log in at the command line:

   ```
   anaconda login
   ```

   NOTE: This is recommended if you do not have a secure connection.

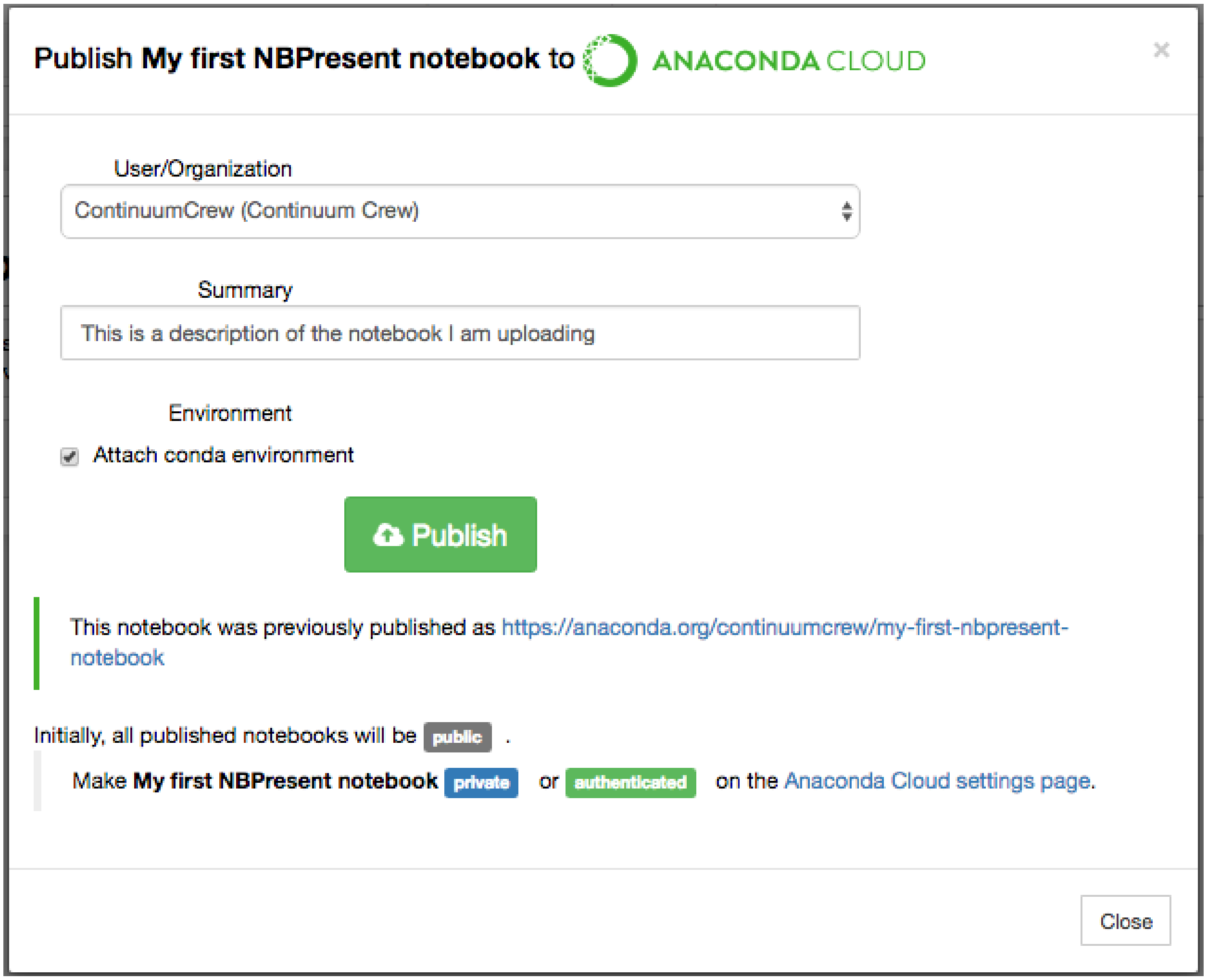
3. Open Jupyter Notebook, then open the notebook you wish to upload to Cloud.

4. In the top navigation bar, click the Publish to Anaconda Cloud button:


jupyter
My first NBPresent notebook Last Checkpoint: Yesterday at 1:10 PM (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Markdown
CellToolbar

5. In the dialog box that appears, select your user name.

6. Type a description of the notebook for display on Cloud:



Publish **My first NBPresent notebook** to ANACONDA CLOUD ×

User/Organization

ContinuumCrew (Continuum Crew)

Summary

This is a description of the notebook I am uploading

Environment

☑ Attach conda environment

Publish

This notebook was previously published as https://anaconda.org/continuumcrew/my-first-nbpresent-notebook

Initially, all published notebooks will be public .

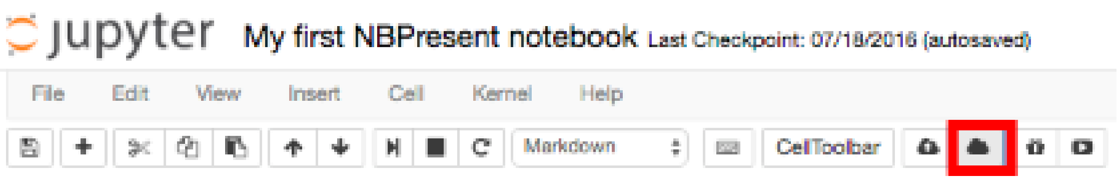
Make **My first NBPresent notebook** private or authenticated on the Anaconda Cloud settings page.

Close

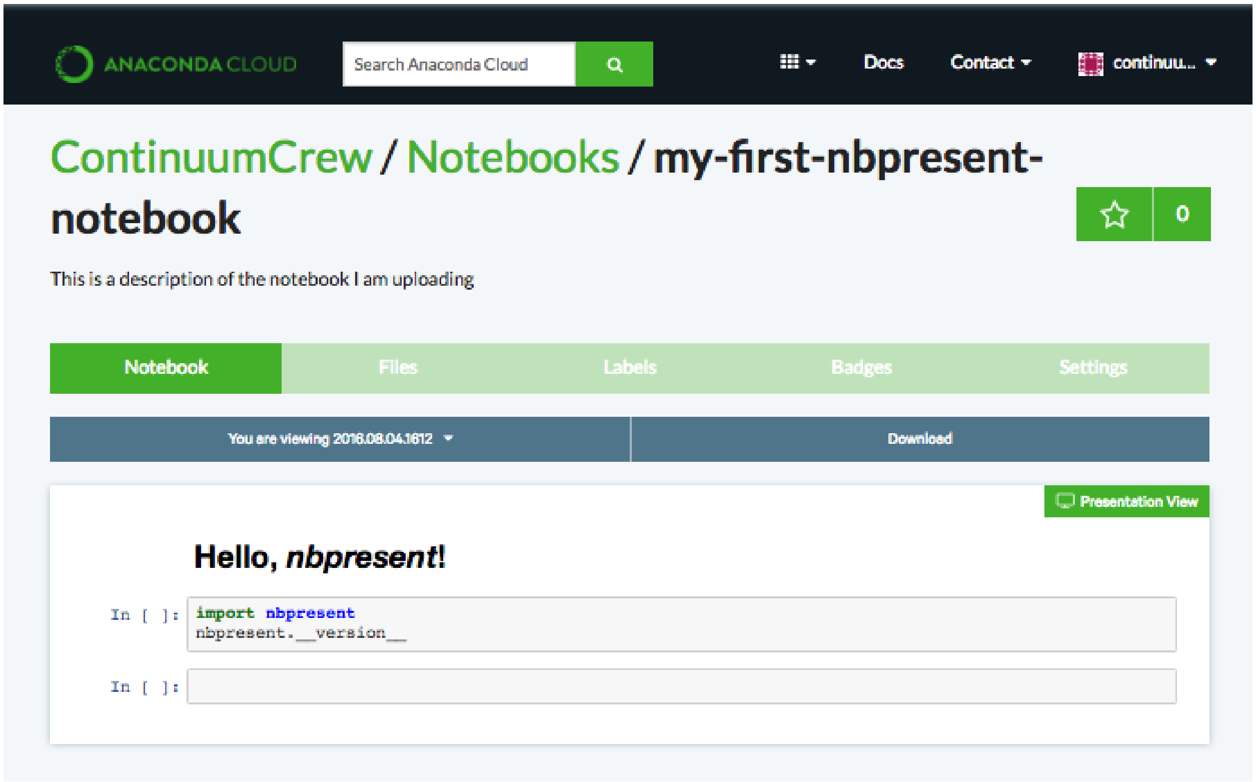
7. If you want the identical environment to be included when the notebook is downloaded and opened, select the Attach conda environment checkbox.

8. Click the Publish button.

9. After publishing, you can view the notebook or play the presentation on Cloud from the top navigation bar by clicking the Cloud button:


jupyter
My first NBPresent notebook Last Checkpoint: 07/18/2016 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Markdown
CellToolbar

Your notebook on Anaconda Cloud will look similar to this one:



ContinuumCrew / Notebooks / my-first-nbpresent-notebook

This is a description of the notebook I am uploading

Hello, *nbpresent*!

```
In [ ]: import nbpresent
        nbpresent.__version__

In [ ]:
```

For more information on Cloud, see *Anaconda Cloud*.

Notebook Conda

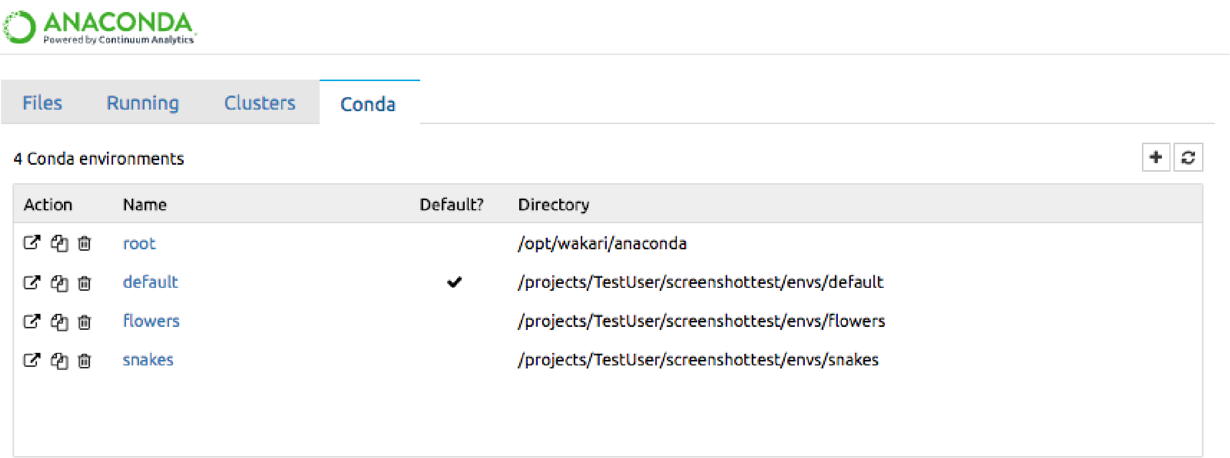
This extension provides conda environment and package access from within Jupyter Notebook.

To manage all environments:

1. While viewing the dashboard file manager, select the **Conda** tab, which shows your current conda environments:

   NOTE: To add a new conda environment, click the + button above the environments list on the right side.

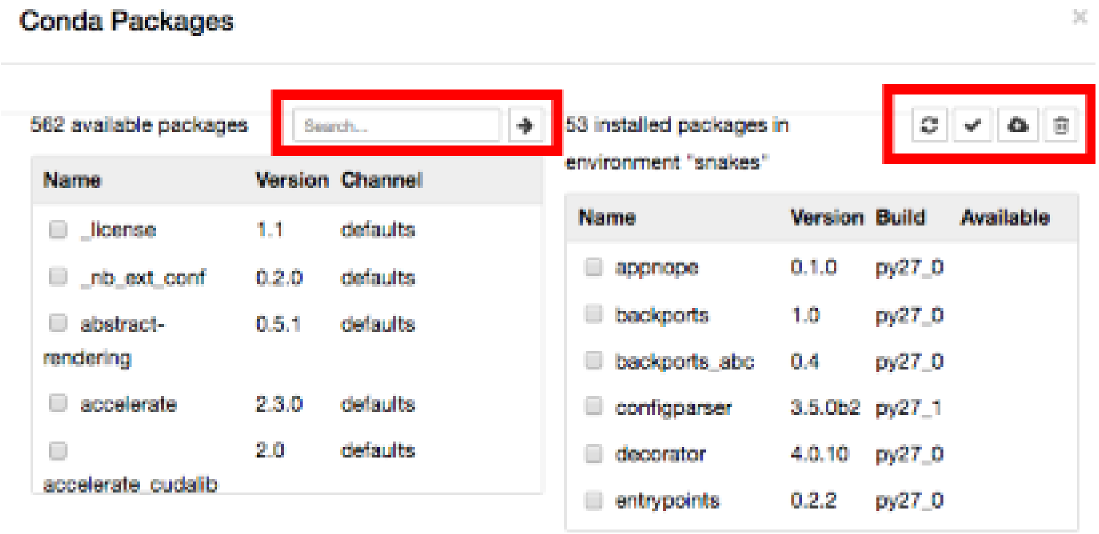
2. Select an environment by clicking its name.
3. In the package management section that displays, the icons from left to right have the following meanings:
   - Search for packages in your current environment.
   - Refresh your packages list.
   - Update selected packages.



4 Conda environments

Action	Name	Default?	Directory
	root		/opt/wakari/anaconda
	default	✓	/projects/TestUser/screenshottest/envs/default
	flowers		/projects/TestUser/screenshottest/envs/flowers
	snakes		/projects/TestUser/screenshottest/envs/snakes

- Remove selected packages.



562 available packages

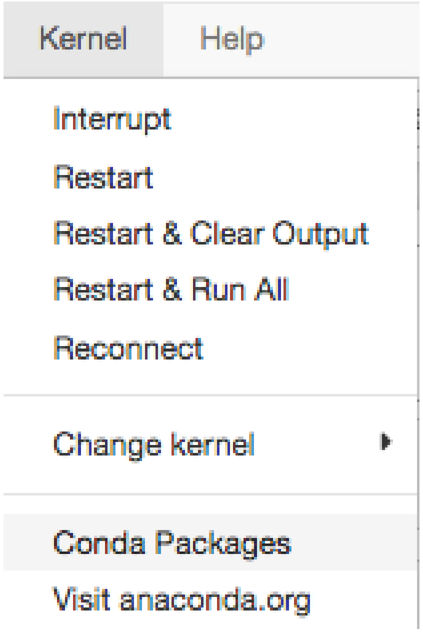
Name	Version	Channel
□ _license	1.1	defaults
□ _nb_ext_conf	0.2.0	defaults
□ abstract-rendering	0.5.1	defaults
□ accelerate	2.3.0	defaults
□ accelerate_cudalib	2.0	defaults

53 installed packages in environment "snakes"

Name	Version	Build	Available
□ appnope	0.1.0	py27_0	
□ backports	1.0	py27_0	
□ backports_abc	0.4	py27_0	
□ configparser	3.5.0b2	py27_1	
□ decorator	4.0.10	py27_0	
□ entrypoints	0.2.2	py27_0	

To manage the current kernel environment, in the **Kernel** menu, select Conda Packages, which displays a list of Conda Packages in the current environment:

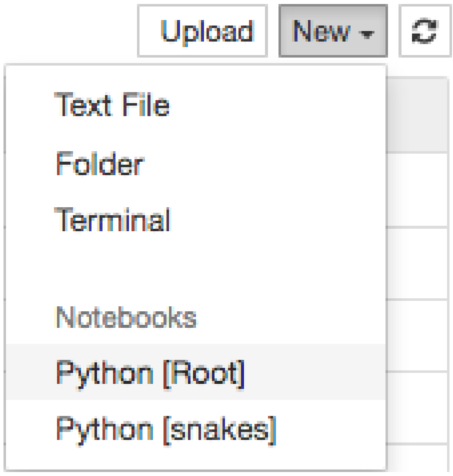
For more information on using and managing conda packages, see Managing packages.



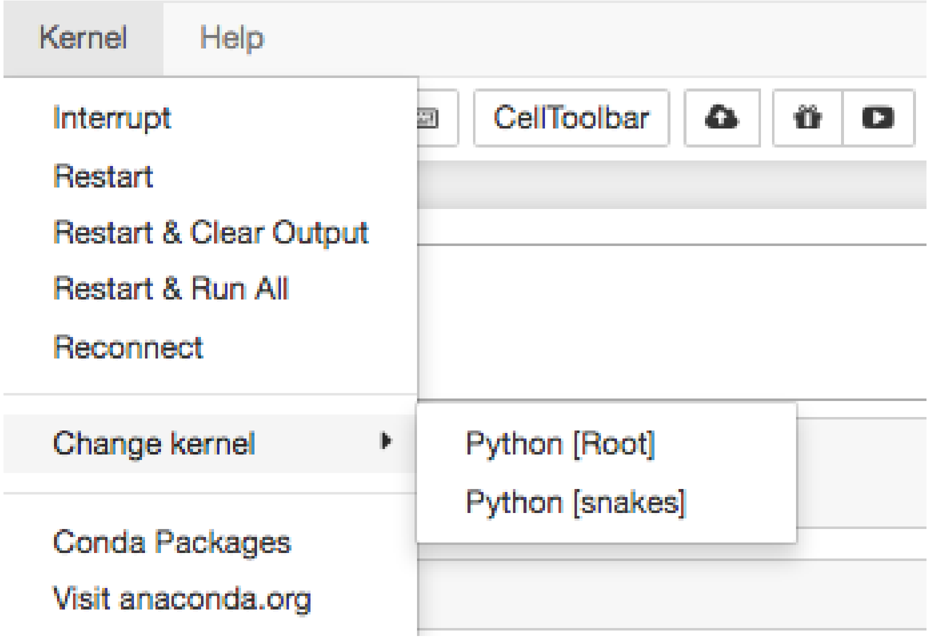
Notebook Conda Kernels

This extension allows you to use conda environment-based kernels from the dashboard and the notebook's **Kernel** menu. It makes the notebook aware of your conda environments, and it is required for Notebook Anaconda Cloud and Notebook Conda.

When creating a new notebook on the **Files** tab, you can pick any of the Python or R language kernels in any of your environments:



You can also change to those kernels on a current notebook:



Using R language with Anaconda

With Anaconda you can easily install the R programming language and over 80 of the most used R packages for data science, and easily create and share your own custom R packages.

The R Essentials bundle contains the IRKernel and more than 80 of the most popular R packages for data science, including dplyr, shiny, ggplot2, tidyr, caret and nnet.

As of early 2018, the default R interpreter installed into new environments is MRO. You can specify the MRO interpreter with the `mro-base` package or the R interpreter with the `r-base` package. Unless you request a change, conda will continue to use the existing interpreter in each environment.

To run the commands below on Windows use Start - Anaconda Prompt. On macOS or Linux open a Terminal.

MRO supported operating systems

- 64-bit systems only for all operating systems—Windows, macOS and Linux.
- Windows 7.0 SP1, Windows 8.1, Windows 10, Windows Server 2008 R2 SP1 and Windows Server 2012.
- Linux—CentOS, Red Hat Enterprise Linux, Debian and Ubuntu.

Anaconda with MRO is not currently supported on macOS but will soon be supported on macOS El Capitan (10.11) and later.

Anaconda with R is supported on macOS Yosemite (10.10) and later.

Creating an environment with MRO and R Essentials

1. Download and *install* Anaconda.

- Create a new conda environment with MRO and all the r-essentials conda packages built from MRAN:

  ```
  conda create -n mro_env r-essentials
  ```

- Activate the environment:

  ```
  conda activate mro_env
  ```

- List the packages in the environment:

  ```
  conda list
  ```

The list shows that the package `mro-base` is installed and `mro` is listed in the build string of the other R packages in the environment.

When using MRO conda packages, starting the R interactive interpreter shows Microsoft R Open in the startup message.

Anaconda Navigator, the Anaconda graphical package manager and application launcher, also creates MRO environments by default. You may instead select R when creating a new conda environment from within Navigator.

Microsoft R Client

Microsoft R Client is a free, community-supported data science tool for high performance analytics built on top of MRO. Additionally, R Client introduces the powerful ScaleR technology and its proprietary functions to benefit from parallelization and remote computing.

Microsoft R Client is now available as a conda package (`r-mrclient`). Users of Windows or RHEL-7/CentOS7/Ubuntu 14.04 and above also have the option to `conda install` the MicrosoftML R package for machine learning (`r-mrclient-mml`) and the pre-trained models for sentiment analysis and image detection (`r-mrclient-mlm`).

Updating R packages

- Update all of the packages and their dependencies with one command:

  ```
  conda update r-essentials
  ```

- If a new version of a package is available in the R channel, you can use `conda update` to update specific packages.

Creating a new environment with R instead of MRO

When creating a new environment, you can use R and not MRO by explicitly including `r-base` in your list of packages. This option will continue to be supported for users who prefer R or use platforms that do not support MRO, including 32-bit operating systems and older versions of macOS.

With conda 4.4:

```
conda create -n r-environment r-essentials r-base
conda activate r-environment
```

Switch an environment from R to MRO

- Activate the environment containing R.
- If you use conda 4.4 or later, run:

```
conda install mro-base
```

- If you use conda 4.3, run:

```
conda remove --force r-base _r-mutex
conda install mro-base
```

The environment's R interpreter will switch from R to MRO.

Switch the default R interpreter from MRO to R

- Run `conda info` and check your version of conda.
- If your version of conda is below 4.4, run `conda update conda` to update conda to the latest version.
- Run:

```
conda config --system --set pinned_packages _r-mutex=*=anacondar*
```

The default R interpreter will switch from MRO to R.

Creating and sharing custom R bundles

Creating and sharing custom R bundles is similar to creating and sharing conda packages.

EXAMPLE: Create a simple custom R bundle metapackage named "Custom-R-Bundle" that contains several popular programs and their dependencies:

```
conda metapackage custom-r-bundle 0.1.0 --dependencies r-irkernel jupyter r-ggplot2 r-
↪dplyr --summary "My custom R bundle"
```

Share the new metapackage by uploading it to your channel on Anaconda Cloud:

```
conda install anaconda-client
anaconda login
anaconda upload custom-r-bundle-0.1.0-0.tar.bz2
```

Anyone can now access your custom R bundle from any computer:

```
conda install -c <your anaconda.org username> custom-r-bundle
```

For more information, see Jupyter and conda for R language.

Mirroring the R channel

Many Anaconda Enterprise customers maintain a local mirror of the R channel.

When mirroring the R channel for the first time after the early 2018 update, clean the existing packages by running the command `anaconda-server-sync-conda` with the option `--clean`.

R packages

- `r-essentials`: The R Essentials bundle contains the IRKernel and more than 80 of the most popular R packages for data science, including dplyr, shiny, ggplot2, tidyr, caret and nnet.
- `mro-basics`: The MRO Basics metapackage contains everything in the Microsoft MRO installers. It does not include `r-mrclient`, `r-mrclient-mml`, or `r-mrclient-mlm`.
- `r-mrclient`: Microsoft R Client is a free, community-supported, data science tool for high performance analytics.
- `r-mrclient-mml`: MicrosoftML provides state-of-the-art fast, scalable machine learning algorithms and transforms for R. Depends on `r-mrclient`.
- `r-mrclient-mlm`: MicrosoftML Machine Learning Models are pre-trained machine learning models for sentiment analysis and image detection. Depends on `r-mrclient-mml`.

Uninstalling R Essentials

To uninstall the R Essentials package, run:

```
conda remove r-essentials
```

NOTE: This removes only R Essentials and disables R Language support. Other R language packages are not removed.

User group

For community help on using conda with MRO, join the conda email group.

Resources

Here are our more popular resources on using Anaconda with the R programming language:

- *R Language packages available for use with Anaconda*–There are hundreds of R language packages now available, and several ways to get them.
- *Navigator tutorial*–Use the R programming language with Anaconda Navigator. The Anaconda Navigator graphical interface (GUI) makes it easy for even new users to use and run the R language in a Jupyter Notebook.
- *Using R packages with Anaconda and Cloudera CDH*–Anaconda Scale provides resource management tools to easily deploy Anaconda across a cluster. It helps you manage multiple conda environments and packages, including Python and R language, on bare-metal or cloud-based clusters.
- Blog post: Jupyter and conda for R–The many benefits that Jupyter, the IRKernel and conda can provide for data scientists working with the R programming language.
- Blog post: Anaconda for R users: SparkR and rBokeh– Data Scientist Christine Doig presents two projects for the R programming language that are powered by Anaconda. rBokeh allows you to create beautiful interactive visualizations. Scale your predictive models with SparkR through Anaconda's cluster management capabilities.
- Using Anaconda with Hadoop: Distributed language processing with PySpark– This notebook example shows how Anaconda for cluster management makes it easy to manage packages, including Python and R, on a Hadoop cluster with PySpark.
- Webinar: Predict. Share. Deploy.–Download the webinar video to:

  - Build predictive models in Python with Anaconda using Python packages such as pandas and scikit-learn in Jupyter Notebooks.
  - Use modern open data science languages including Python and R together in your analysis.
  - Share your results with your entire data science team.
- Webinar: Anaconda for R Users–Download the slides from the webinar to see how Anaconda makes package, dependency and environment management easy with R language and other Open Data Science languages.

Switching between Python 2 and Python 3 environments

You can easily maintain separate environments for Python 2 programs and Python 3 programs on the same computer, without worrying about the programs interacting with each other. Switching to an environment is called activating it.

Summary

1. Create a Python 2 environment
2. Create a Python 3 environment
3. Activate and use the Python 2 environment
4. Deactivate the Python 2 environment
5. Activate and use the Python 3 environment
6. Deactivate the Python 3 environment

Create a Python 2 environment named py2, install Python 2.7:

```
conda create --name py2 python=2.7
```

Create a new environment named py3, install Python 3.5:

```
conda create --name py3 python=3.5
```

Now you have two environments to work with. You can install packages and run programs as desired in either one.

Activate and use the Python 2 environment

WINDOWS:

```
activate py2
```

LINUX, macOS:

```
source activate py2
```

Use your py2 environment to install packages and run programs as desired. When finished, deactivate the environment

WINDOWS:

```
deactivate
```

macOS, LINUX:

```
source deactivate
```

Activate and use the Python 3 environment

WINDOWS:

```
activate py3
```

LINUX, macOS:

```
source activate py3
```

Use the py3 environment to install and run programs as desired. When finished, deactivate the environment

WINDOWS:

```
deactivate
```

macOS, LINUX:

```
source deactivate
```

Moving Anaconda from one directory to another

Anaconda environment management requires known paths, and Anaconda must know the exact path where it is installed at installation time.

If you simply copy the Anaconda files to a new directory, Anaconda will not work.

To move Anaconda from one directory to another:

1. OPTIONAL: Save your environments using the conda managing environments instructions.
2. *Uninstall Anaconda*.
3. Go to the new directory and install it there following the *Anaconda installation instructions*.
4. OPTIONAL: Restore your environments using the conda managing environments instructions.

Integrations

Data science is a team sport, so we have built the Anaconda platform to be language-agnostic as well as extensible. With Anaconda integrations, you can continue to use your favorite IDEs, cloud computing platforms, virtual machine platforms, and more.

cloudera

Cloudera provides Apache Hadoop-based software, support and services, as well as training to business customers. Their open source Apache Hadoop distribution, CDH (Cloudera Distribution Including Apache Hadoop), targets enterprise-class deployments of that technology.

Amazon Web Services (AWS) is a collection of cloud-computing services that make up a cloud-computing platform offered by Amazon.com.

Docker is an open platform for developers and system administrators to build, ship and run distributed applications, whether on laptops, data center virtual machines or the cloud.

Using IDEs (Integrated Development Environments) with Anaconda

You can use the following IDEs with Anaconda:

- *Eclipse and PyDev*
- *IDLE*
- *IntelliJ*
- *Ninja IDE*
- *Python Tools for Visual Studio (PTVS)*
- *Python for Visual Studio Code*
- *Spyder*
- *Sublime Text*
- *Microsoft Visual Studio Code (VS Code)*
- *Wing IDE*

NOTE: IDEs often require you to specify the path to your Python interpreter. See *Finding your Anaconda Python interpreter path*.

Microsoft Excel–Anaconda on Windows comes ready to bring the power of Python into Excel. Use one of the many packages included in Anaconda or connect to an outside integration tool.

Cloudera CDH

There are two methods of using Anaconda on an existing cluster with Cloudera CDH, Cloudera's distribution including Apache Hadoop:

- Use the Anaconda parcel for Cloudera CDH. The following procedure describes how to install the Anaconda parcel on a CDH cluster using Cloudera Manager. The Anaconda parcel provides a static installation of Anaconda, based on Python 2.7, that can be used with Python and PySpark jobs on the cluster.

- Use Anaconda Scale, which provides additional functionality, including the ability to manage multiple conda environments and packages, including Python and R, alongside an existing CDH cluster. For more information, see *Using Anaconda with Cloudera CDH*.

See the blog post Self-service Open Data Science: Custom Anaconda parcels for Cloudera.

To install the Anaconda parcel:

1. In the Cloudera Manager Admin Console, in the top navigation bar, click the Parcels icon.
2. At the top right of the parcels page, click the Edit Settings button.
3. In the Remote Parcel Repository URLs section, click the plus symbol, and then add the following repository URL for the Anaconda parcel:

   ```
   https://repo.continuum.io/pkgs/misc/parcels/
   ```

4. At the top of the page, click the Save Changes button.
5. In the top navigation bar, click the Parcels icon to return to the list of available parcels, where you should see the latest version of the Anaconda parcel that is available.
6. To the right of the Anaconda parcel listing, click the Download button.
7. After the parcel is downloaded, click the Distribute button to distribute the parcel to all of the cluster nodes.
8. After the parcel is distributed, click the Activate button to activate the parcel on all of the cluster nodes.
9. When prompted, confirm the activation.

   After the parcel is activated, Anaconda is available on all of the cluster nodes.

You can submit Spark jobs along with the `PYSPARK_PYTHON` environment variable that refers to the location of Anaconda. For example, enter the following command all on one line:

```
PYSPARK_PYTHON=/opt/cloudera/parcels/Anaconda/bin/
python spark-submit pyspark_script.py
```

NOTE: The line break in the example above is for readability only. Enter the command all on one line.

NOTE: The repository URL shown above installs the most recent version of the Anaconda parcel. To install an older version of the Anaconda parcel, add the following repository URL to the Remote Parcel Repository URLs in Cloudera manager, and then follow the above steps with your desired version of the Anaconda parcel.

```
https://repo.continuum.io/pkgs/misc/parcels/archive/
```

Anaconda builds new Cloudera parcels at least once a year each spring and also offers custom parcel creation for our enterprise customers. The Anaconda parcel provided at the repository URL shown above is based on Python 2.7. To use the Anaconda parcel with other versions of Python or with additional packages, contact sales@anaconda.com for more information about custom Anaconda parcel builds or other enterprise solutions for using Anaconda with cluster computing.

Anaconda Workgroup and Anaconda Enterprise subscribers can also use Anaconda Repository to *create and distribute their own custom Anaconda parcels for Cloudera Manager*.

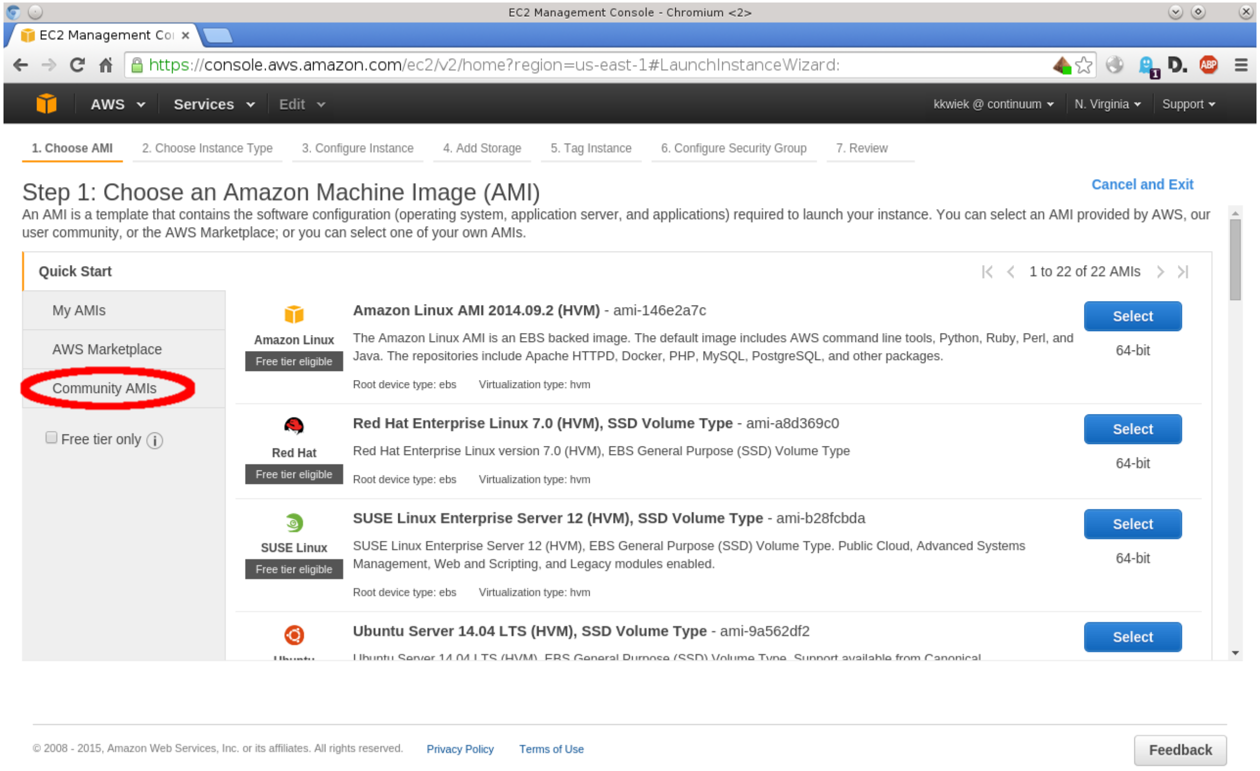
For more information about managing Cloudera parcels, see the Cloudera documentation.

Amazon AWS

This procedure describes how to create a new Amazon AWS virtual machine with Anaconda. To add Anaconda to an existing AWS virtual machine using the command line, see *Linux installation*.

To create a new Amazon AWS:

1. Create an Amazon EC2 instance. EC2 (Elastic Compute Cloud) is Amazon's virtualization service.
2. Open your EC2 console and click the Launch instance button.
3. On the Choose an Amazon Machine Image (AMI) page, select Community AMIs:



EC2 Management Console - Chromium <2>

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2014.09.2 (HVM) - ami-146e2a7c

The Amazon Linux AMI is an EBS backed image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Apache HTTPD, Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

Red Hat Enterprise Linux 7.0 (HVM), SSD Volume Type - ami-a8d369c0

Red Hat Enterprise Linux version 7.0 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-b28fcbda

SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9a562df2

Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

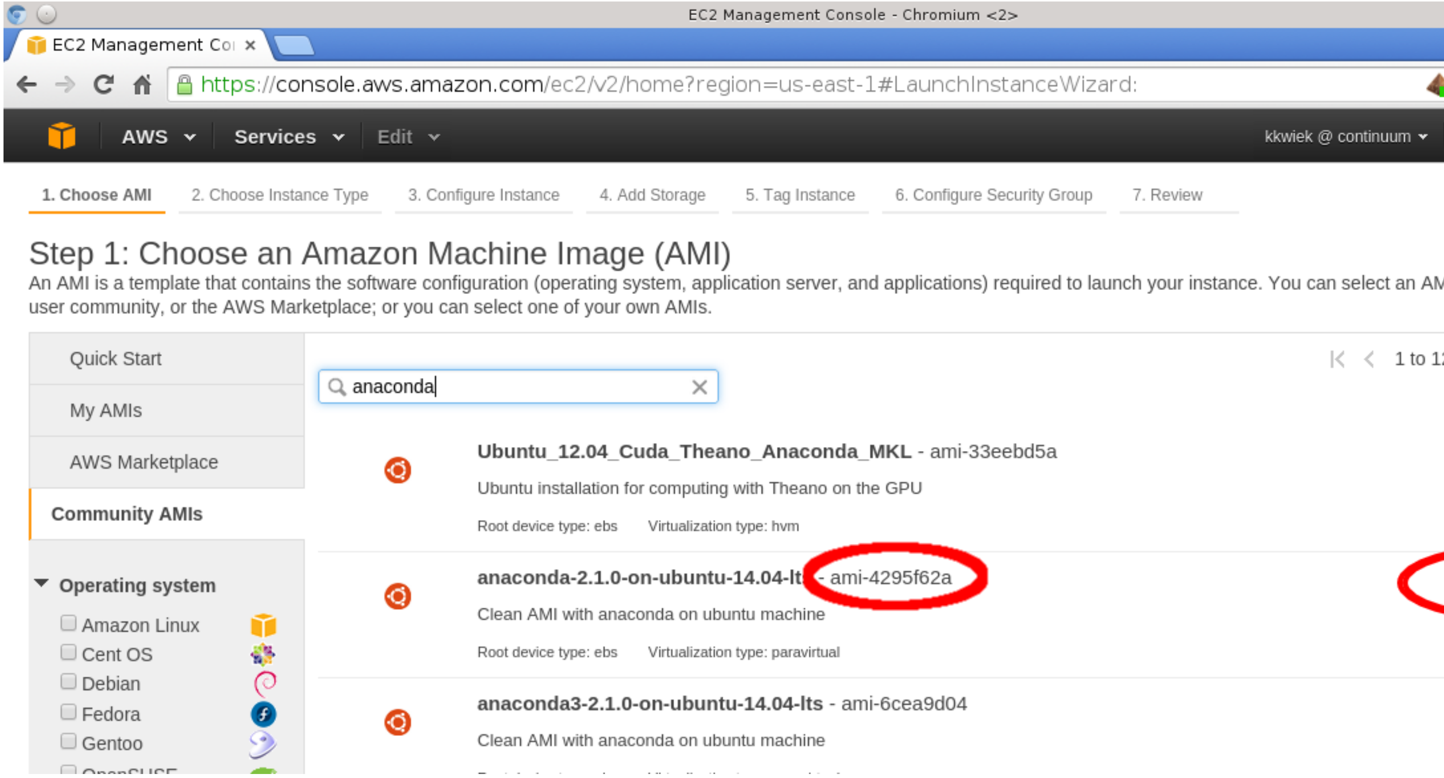
4. Search the Community Amazon Machine Images for the Anaconda or Miniconda AMI.
5. On the search results page, click the Select button:

6. To be sure that you are using an official Anaconda® image, verify its AMI ID by matching it against the *official list of Anaconda images*.
7. Install your AMI with Amazon's Launch an EC2 Instance instructions.

Official Anaconda images

Docker

Read the official Docker documentation and specifically the information related to Docker images.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AM user community, or the AWS Marketplace; or you can select one of your own AMIs.

anaconda

Ubuntu_12.04_Cuda_Theano_Anaconda_MKL - ami-33eebd5a
Ubuntu installation for computing with Theano on the GPU
Root device type: ebs Virtualization type: hvm

anaconda-2.1.0-on-ubuntu-14.04-lts - ami-4295f62a
Clean AMI with anaconda on ubuntu machine
Root device type: ebs Virtualization type: paravirtual

anaconda3-2.1.0-on-ubuntu-14.04-lts - ami-6cea9d04
Clean AMI with anaconda on ubuntu machine

Begin by browsing the available Anaconda images on our Docker profile.

To obtain a fully working Anaconda image:

1. In a terminal window, run this command to display a list of available images:

```
docker search continuumio
```

2. Pull the desired image:

```
docker pull continuumio/miniconda
```

3. Create a container using the image:

```
docker run -t -i continuumio/miniconda /bin/bash
```

   This gives you direct access to the container where the conda tool is already available.

4. Test the container:

```
conda info
```

   You now have a fully working Anaconda image.

To install and launch the Jupyter Notebook, execute the following command all on one line from the host machine:

```
docker run -i -t -p 8888:8888 continuumio/miniconda /bin/bash
-c "/opt/conda/bin/conda install jupyter -y --quiet && mkdir
/opt/notebooks && /opt/conda/bin/jupyter notebook
--notebook-dir=/opt/notebooks --ip='*' --port=8888
--no-browser"
```

NOTE: Line breaks in the example above are for readability only. Enter the command all on one line.

To access the Jupyter notebook open http://localhost:8888 in your browser, or open http://<DOCKER-MACHINE-IP>:8888 if you are using a Docker Machine VM.

NOTE: Replace `<DOCKER-MACHINE-IP>` with your Docker Machine VM IP address.

Eclipse and PyDev

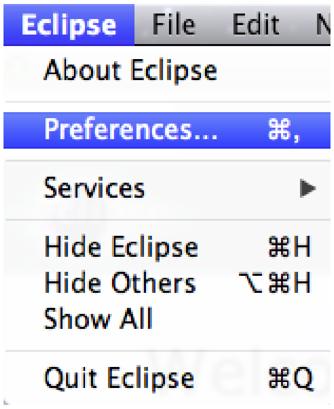
Eclipse is an open source platform that provides an array of convenient and powerful code-editing and debugging tools. PyDev is a Python IDE that runs on top of Eclipse.

If you do not already have Eclipse and PyDev installed:

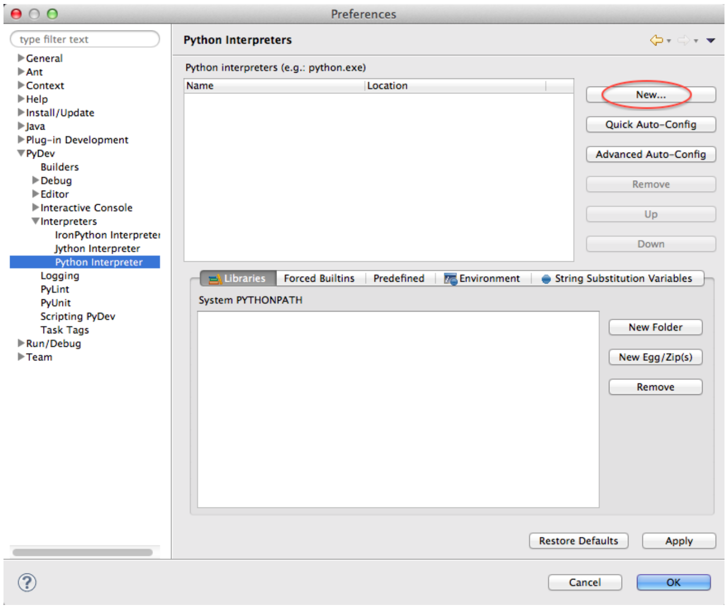
- Download Eclipse and read the Eclipse resources.
- You may need to install the latest Java JDK/JRE before installing PyDev in order for PyDev to show up in the Eclipse Preferences menu after PyDev installation.
- Install PyDev.

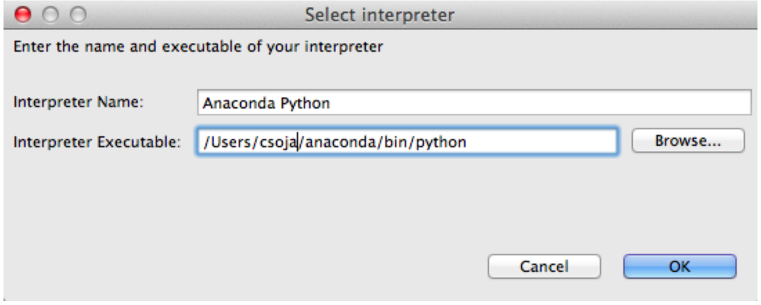
After you have Eclipse, PyDev, and Anaconda installed, set Anaconda Python as your default:

1. Open the Eclipse Preferences window:



2. In the PyDev list, select Interpreters, and then select Python Interpreter.
3. Click the New button:
4. In the Interpreter Name box, type `Anaconda Python`.
5. Browse to your *Anaconda Python interpreter path*.
6. Click the OK button:





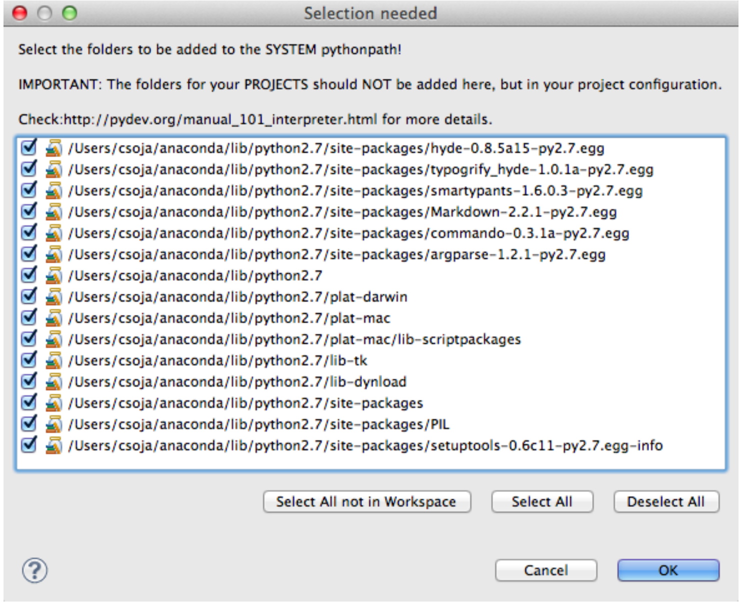
Enter the name and executable of your interpreter

Interpreter Name: Anaconda Python

Interpreter Executable: /Users/csoja/anaconda/bin/python

7. In the next window, select the folders to add to the SYSTEM python path:

   (a) Select all the folders:



Select the folders to be added to the SYSTEM pythonpath!

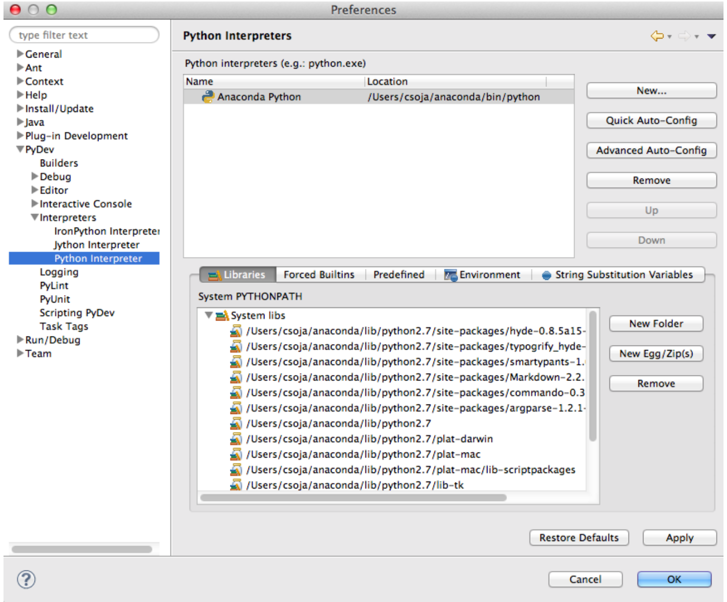
IMPORTANT: The folders for your PROJECTS should NOT be added here, but in your project configuration.

Check:http://pydev.org/manual_101_interpreter.html for more details.

☑ /Users/csoja/anaconda/lib/python2.7/site-packages/hyde-0.8.5a15-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/typogrify_hyde-1.0.1a-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/smartypants-1.6.0.3-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/Markdown-2.2.1-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/commando-0.3.1a-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/argparse-1.2.1-py2.7.egg
☑ /Users/csoja/anaconda/lib/python2.7
☑ /Users/csoja/anaconda/lib/python2.7/plat-darwin
☑ /Users/csoja/anaconda/lib/python2.7/plat-mac
☑ /Users/csoja/anaconda/lib/python2.7/plat-mac/lib-scriptpackages
☑ /Users/csoja/anaconda/lib/python2.7/lib-tk
☑ /Users/csoja/anaconda/lib/python2.7/lib-dynload
☑ /Users/csoja/anaconda/lib/python2.7/site-packages
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/PIL
☑ /Users/csoja/anaconda/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg-info

   (b) Click the OK button.

   The Python Interpreters window now displays Anaconda Python:



Python Interpreters

Python interpreters (e.g.: python.exe)

Name	Location
Anaconda Python	/Users/csoja/anaconda/bin/python

System PYTHONPATH

System libs
/Users/csoja/anaconda/lib/python2.7/site-packages/hyde-0.8.5a15-
/Users/csoja/anaconda/lib/python2.7/site-packages/typogrify_hyde-
/Users/csoja/anaconda/lib/python2.7/site-packages/smartypants-1.
/Users/csoja/anaconda/lib/python2.7/site-packages/Markdown-2.2.
/Users/csoja/anaconda/lib/python2.7/site-packages/commando-0.3
/Users/csoja/anaconda/lib/python2.7/site-packages/argparse-1.2.1-
/Users/csoja/anaconda/lib/python2.7
/Users/csoja/anaconda/lib/python2.7/plat-darwin
/Users/csoja/anaconda/lib/python2.7/plat-mac
/Users/csoja/anaconda/lib/python2.7/plat-mac/lib-scriptpackages
/Users/csoja/anaconda/lib/python2.7/lib-tk

(c) Click the OK button.

You are now ready to use Anaconda Python with your Eclipse and PyDev installation.

IDLE

IDLE is a very small and simple cross-platform IDE that is included free with Python and is released under the open source Python Software Foundation License.

Anaconda and Miniconda include IDLE.

To use IDLE:

1. Find the IDLE program file:

   EXAMPLES for Miniconda with Python 3 and user name "jsmith":

   - On macOS, the full path may be `/Users/jsmith/miniconda3/bin/idle3.5`.
   - On Windows, the full path may be `C:\Users\jsmith\Miniconda3\Scripts\idle`.

2. Run IDLE from the file explorer, or run it from the command line by entering the full path.

TIP: You can make a shortcut to the IDLE program file on your desktop or add an item to the main operating system menu.

IntelliJ

IntelliJ IDEA Community Edition is the open source version of IntelliJ IDEA, an IDE (Integrated Development Environment) for Java, Groovy and other programming languages such as Scala or Clojure. It is made by JetBrains, maker of PyCharm Python IDE.

Before you start

You should have both Miniconda and IntelliJ installed and working.

Set up IntelliJ using Miniconda

Find location of Miniconda Python executable:

```
which python
```

The system responds with your path to Python. You need this in the next step.

Within IntelliJ

Go to `File > Project Structure > Platform Settings > SDKs`

Click the Add (+) icon

Choose `Python SDK`

Enter location of Miniconda Python executable `/Users/Me/miniconda3/bin/python`

NOTE: Substitute your actual path to Miniconda that you found in the previous step.

Go to `File > Project Settings > Project > Project SDK`

Select from the drop-down: Python 3.6.0 (~/miniconda3/bin/python. . .)

Go to `Run > Debug`

Click the Add (+) icon

Select `Python` and then enter the following:

Name: MyProject Script: /Users/Me/MyProject/my_file.py Use Specified Interpreter: Python 3.6.0 (~/miniconda3/bin/python. . .)

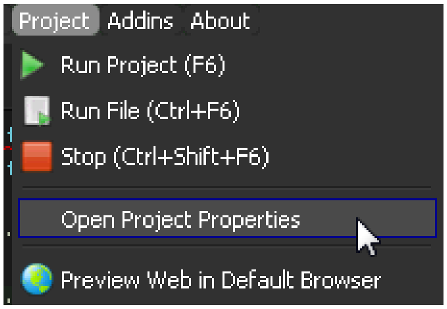
For more information, see the IntelliJ IDEA documentation.

Ninja IDE

Ninja IDE is a cross-platform, free, open source IDE specially designed for Python application development.

To use Anaconda Python with the Ninja IDE:

1. In the **Project** menu, select Open Project Properties:



2. On the **Project Execution** tab, in the Python Custom Interpreter box, enter *the Anaconda Python interpreter path* to select Anaconda Python:

3. Verify the setup with a test script:

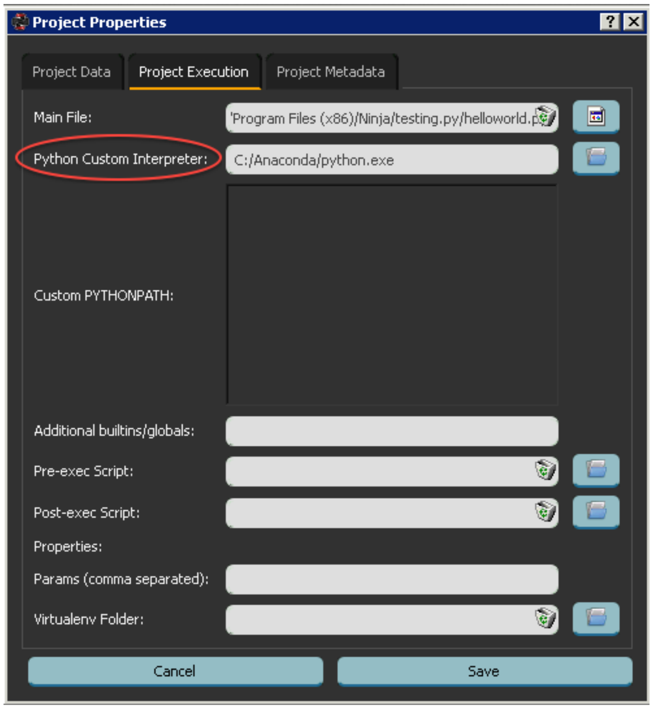
   (a) To create the test script, enter this code:

```
import sys
print(sys.version)
```

   (b) Run your test script.

   In the command output, your Anaconda Python version is listed:

For more information, see the Ninja-IDE documentation.



Main File: 'Program Files (x86)/Ninja/testing.py/helloworld.p

Python Custom Interpreter: C:/Anaconda/python.exe

Custom PYTHONPATH:

Additional builtins/globals:

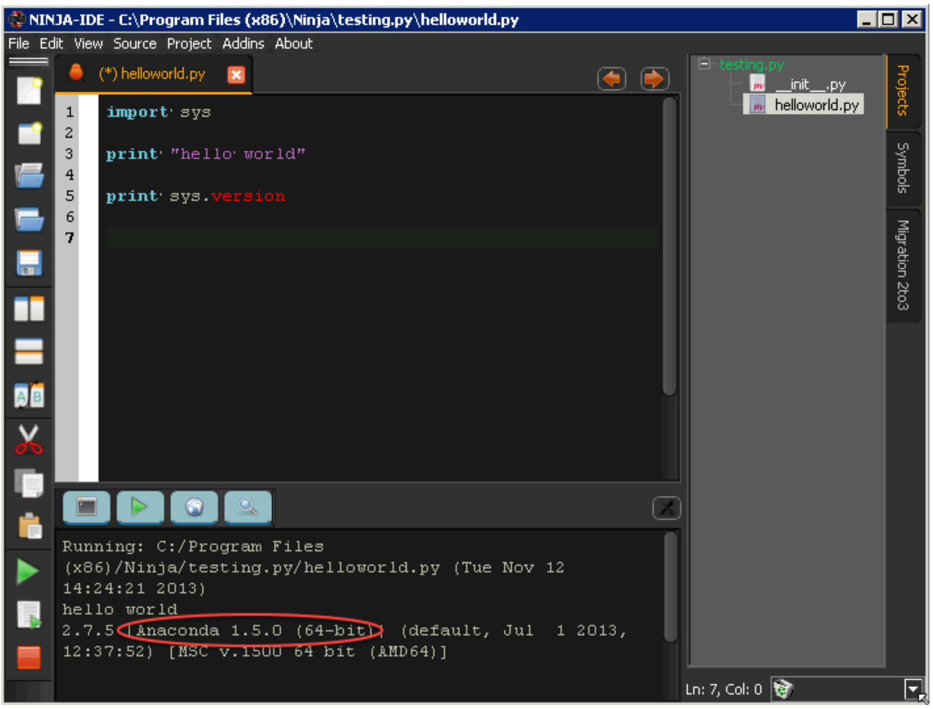
Pre-exec Script:

Post-exec Script:

Properties:

Params (comma separated):

Virtualenv Folder:



```
import sys

print "hello world"

print sys.version
```

```
Running: C:/Program Files
(x86)/Ninja/testing.py/helloworld.py (Tue Nov 12
14:24:21 2013)
hello world
2.7.5 |Anaconda 1.5.0 (64-bit)| (default, Jul  1 2013,
12:37:52) [MSC v.1500 64 bit (AMD64)]
```

Python Tools for Visual Studio (PTVS)

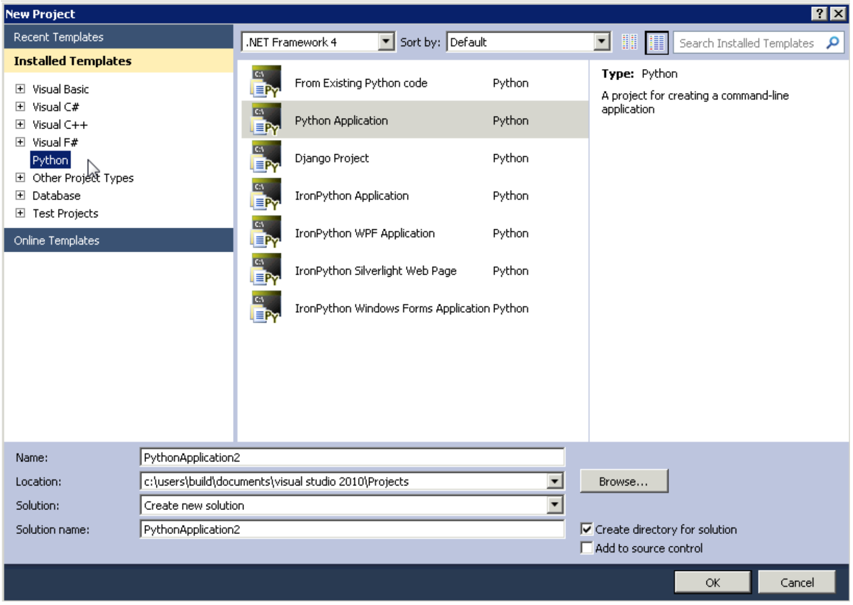
Python Tools for Visual Studio is a free, open source plugin that turns Visual Studio into a Python IDE.

If you have installed Anaconda as your default Python installation and installed PTVS, your Visual Studio installation is already set to use Anaconda's Python interpreter in PTVS.

To verify this, create a new Python project, and then check whether Anaconda is the Python that it uses.

To create a new Python project:

1. In the **File** menu, select New, and then select Project. The keyboard shortcut is Ctrl-Shift-N.
2. Select Python Application:


New Project
Recent Templates
Installed Templates
Visual Basic
Visual C#
Visual C++
Visual F#
Python
Other Project Types
Database
Test Projects
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
From Existing Python code Python
Python Application Python
Django Project Python
IronPython Application Python
IronPython WPF Application Python
IronPython Silverlight Web Page Python
IronPython Windows Forms Application Python
Browse...
OK
Cancel

Type: Python

A project for creating a command-line application

Name: PythonApplication2

Location: c:\users\build\documents\visual studio 2010\Projects

Solution: Create new solution

Solution name: PythonApplication2

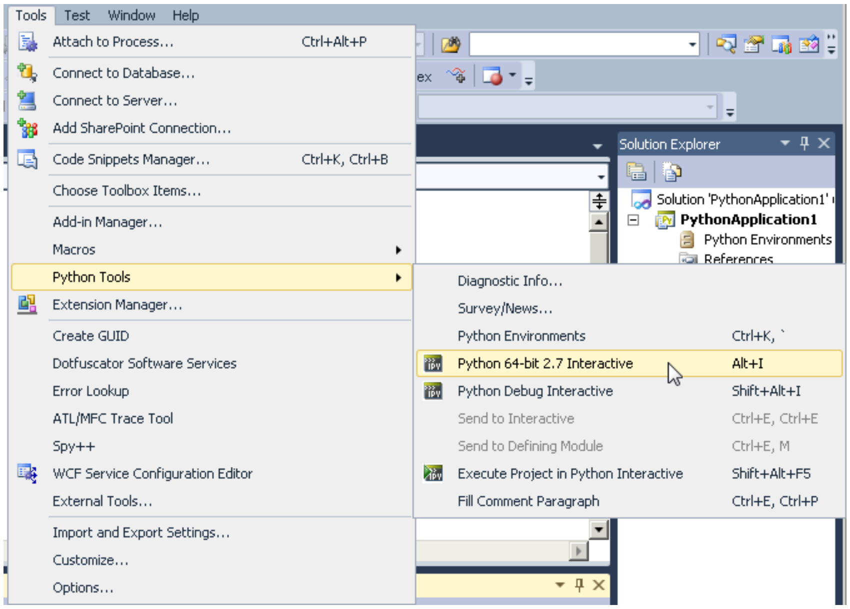
☑ Create directory for solution

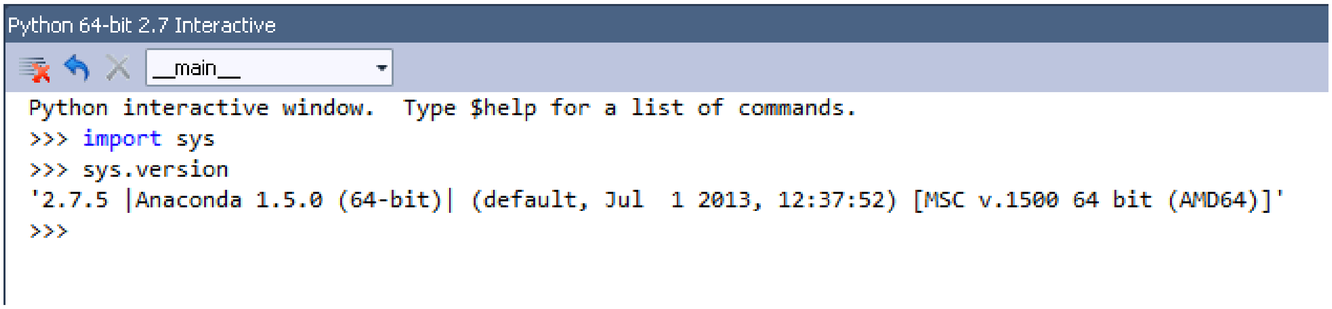
☐ Add to source control

To check which Python the project uses:

1. From the **Tools** menu, select Python Tools, and then select Python Interactive. The keyboard shortcut is Alt+I:
2. In the **Python Interactive** window, type `import sys`, and then press Enter.
3. Type `sys.version`, and then press Enter.

If PTVS is using Anaconda, your Anaconda Python version is displayed. In the example below, it is Anaconda 1.5.0 (64-bit):


Tools
Test
Window
Help
Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Add SharePoint Connection...
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
Macros
Python Tools
Extension Manager...
Create GUID
Dotfuscator Software Services
Error Lookup
ATL/MFC Trace Tool
Spy++
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...
Diagnostic Info...
Survey/News...
Python Environments Ctrl+K, `
Python 64-bit 2.7 Interactive Alt+I
Python Debug Interactive Shift+Alt+I
Send to Interactive Ctrl+E, Ctrl+E
Send to Defining Module Ctrl+E, M
Execute Project in Python Interactive Shift+Alt+F5
Fill Comment Paragraph Ctrl+E, Ctrl+P
Solution Explorer
Solution 'PythonApplication1'
PythonApplication1
Python Environments
References



```
Python interactive window.  Type $help for a list of commands.
>>> import sys
>>> sys.version
'2.7.5 |Anaconda 1.5.0 (64-bit)| (default, Jul  1 2013, 12:37:52) [MSC v.1500 64 bit (AMD64)]'
>>>
```

If PTVS does not automatically discover Anaconda, see the official PTVS documentation, especially the section on Selecting and Installing Python Interpreters and the PTVS installation instructions.

Python for Visual Studio Code

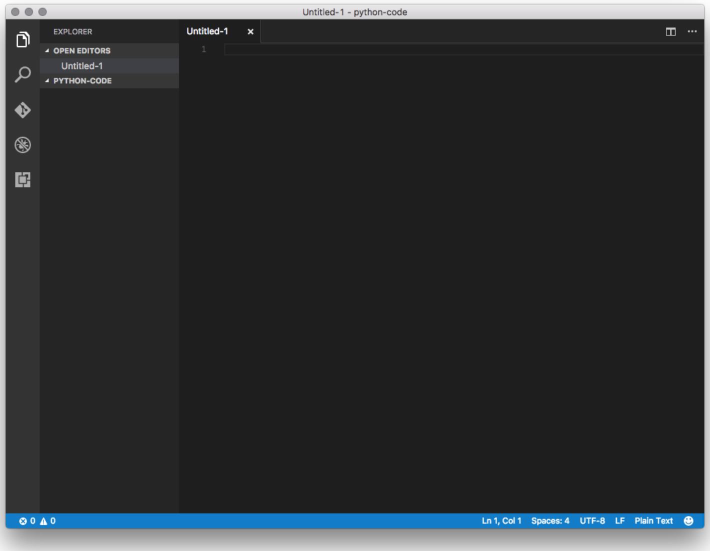
Visual Studio Code (VSC) is a free cross-platform source code editor. The Python for Visual Studio Code extension allows VSC to connect to Python distributions installed on your computer.

If you've installed Anaconda as your default Python installation and installed Python for Visual Studio Code, your VSC installation is already set to use Anaconda's Python interpreter.

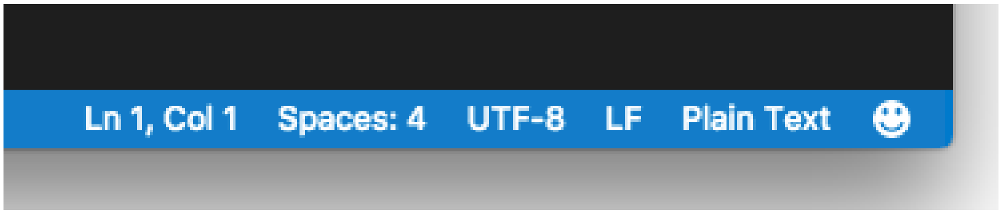
1. Create a new Python source code file:

    (a) To choose a directory to place the code, in the **File** menu, select Open.

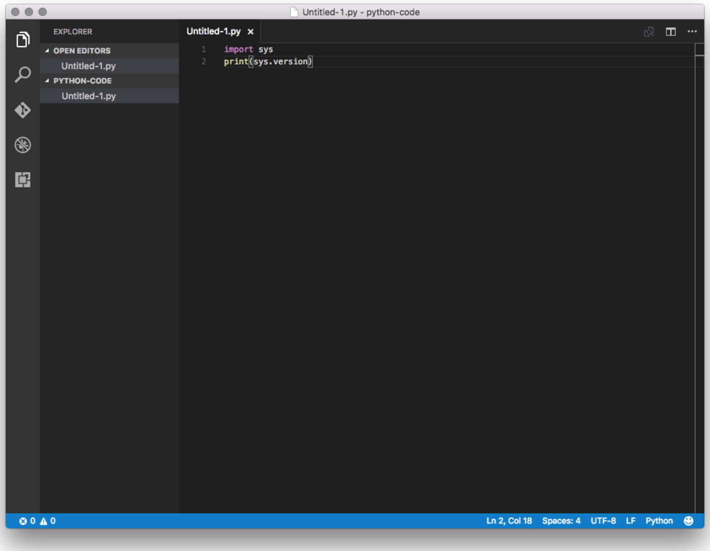
    (b) In the **File** menu, select New File. Your screen will now look like this:



2. To associate the new file with the Python interpreter, at the bottom of the window, click Plain Text:

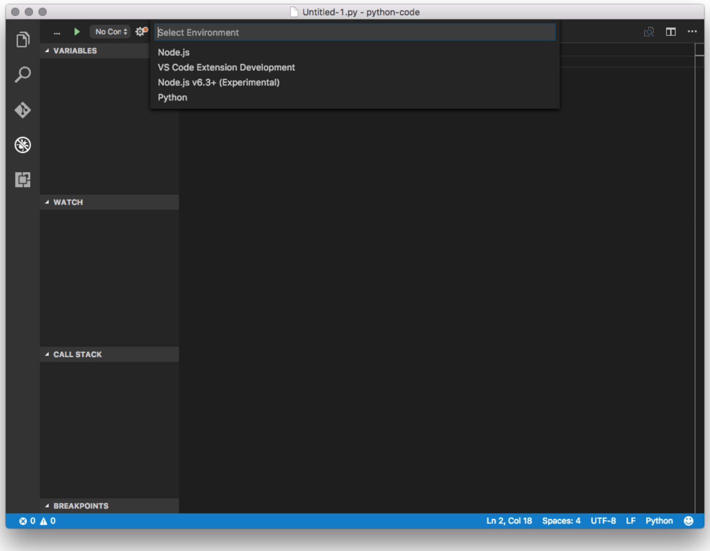


3. In the menu that displays, type or select Python.
4. In the pane on the right, add source code:



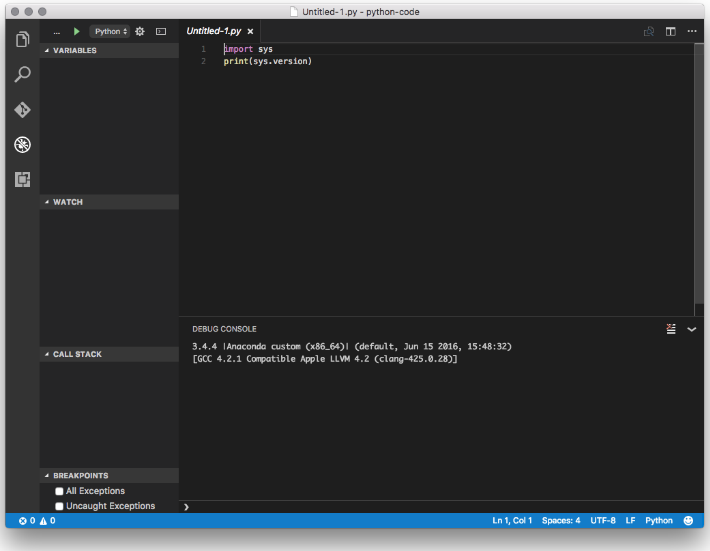
```
import sys
print(sys.version)
```

5. To save the file, in the **File** menu, select Save.
6. To open the Debug pane, click the bug icon. Click the gear icon, and then select Python:


Untitled-1.py - python-code
No Con
VARIABLES
Select Environment
Node.js
VS Code Extension Development
Node.js v6.3+ (Experimental)
Python
WATCH
CALL STACK
BREAKPOINTS
0 0
Ln 2, Col 18 Spaces: 4 UTF-8 LF Python

7. At the top-right, click the green run arrow next to Python.

   The source code is run using your Anaconda Python interpreter:



```
import sys
print(sys.version)
```

```
3.4.4 |Anaconda custom (x86_64)| (default, Jun 15 2016, 15:48:32)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)]
```

Spyder

Spyder, the Scientific PYthon Development EnviRonment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging and introspection features.

After you have installed Anaconda, start Spyder on Windows, macOS or Linux by running the command `spyder`.

Spyder is also pre-installed in *Anaconda Navigator*, included in Anaconda. On the Navigator **Home** tab, click the Spyder icon.

For more information about Spyer, see the Spyder web page or the Spyder documentation.

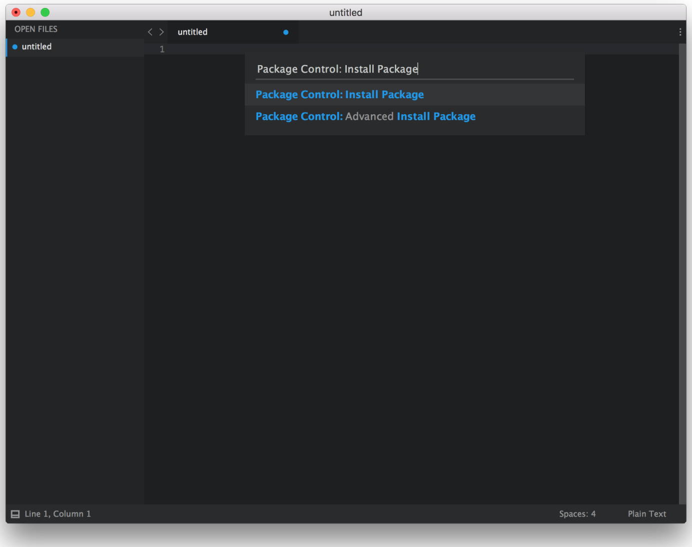
Sublime Text

Sublime Text is a cross-platform text editor for code, markup and prose.

- Download and Install Sublime Text.

To use your Anaconda installation with Sublime Text:

- Download Package control.
- Open the Sublime Text command palette by pressing ctrl+shift+p (Windows, Linux) or cmd+shift+p (macOS).
- All Package Control commands begin with Package Control: , so start by typing Package. Select Package Control : Install Package.

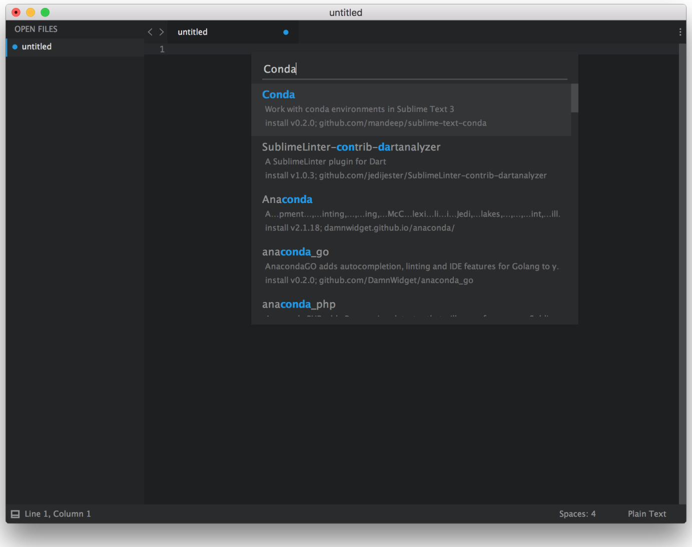


Package Control: Install Package

Package Control: Install Package

Package Control: Advanced Install Package

- Search for Conda in the command palette and select the Conda plugin. When the plugin is installed, a Package Control Message will open in the Sublime Text window.



Conda

Conda
Work with conda environments in Sublime Text 3
install v0.2.0; github.com/mandeep/sublime-text-conda

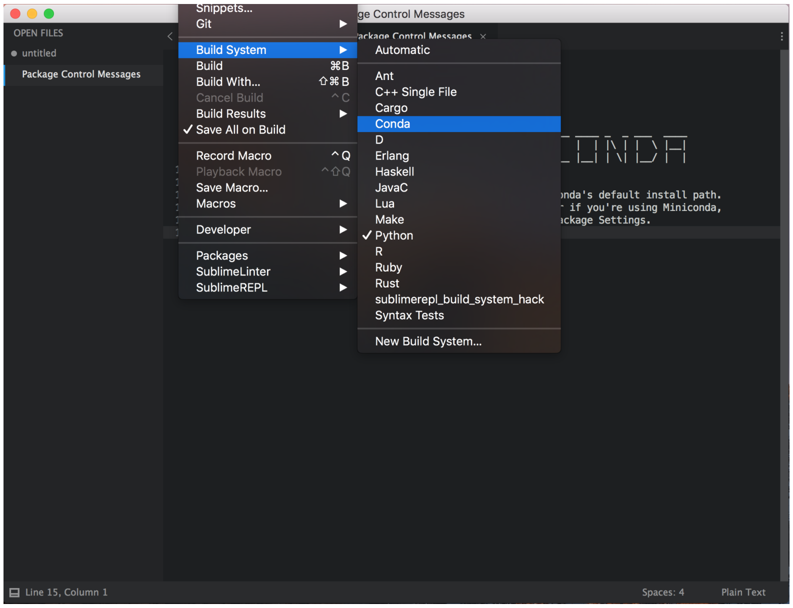
SublimeLinter-contrib-dartanalyzer
A SublimeLinter plugin for Dart
install v1.0.3; github.com/jedijester/SublimeLinter-contrib-dartanalyzer

Anaconda
A...pment...,...inting,...,...ing,...McC...lexi...li...i...Jedi,...lakes,...,...,...int,...ill.
install v2.1.18; damnwidget.github.io/anaconda/

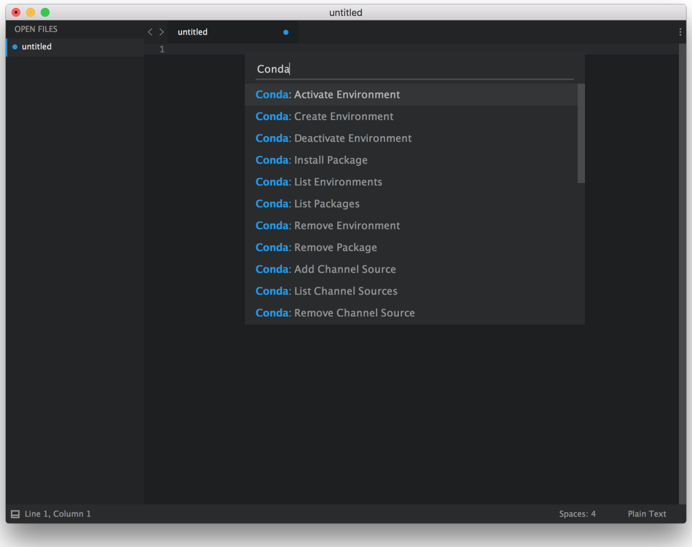
anaconda_go
AnacondaGO adds autocompletion, linting and IDE features for Golang to y.
install v0.2.0; github.com/DamnWidget/anaconda_go

anaconda_php

- Change the current Build System to Conda by accessing Tools -> Build System -> Conda in the menu bar.



- Access the Conda Commands with the Command Palette by searching for Conda.


untitled
OPEN FILES
untitled
Conda
Conda: Activate Environment
Conda: Create Environment
Conda: Deactivate Environment
Conda: Install Package
Conda: List Environments
Conda: List Packages
Conda: Remove Environment
Conda: Remove Package
Conda: Add Channel Source
Conda: List Channel Sources
Conda: Remove Channel Source
Line 1, Column 1
Spaces: 4
Plain Text

Microsoft Visual Studio Code (VS Code)

Anaconda Distribution offers Visual Studio Code (VS Code), Microsoft's lightweight and fast open source code editor.

VS Code is free for both private and commercial use, runs on Windows, macOS and Linux, and includes support for linting, debugging, task running, version control and Git integration, IntelliSense code completion and conda environments.

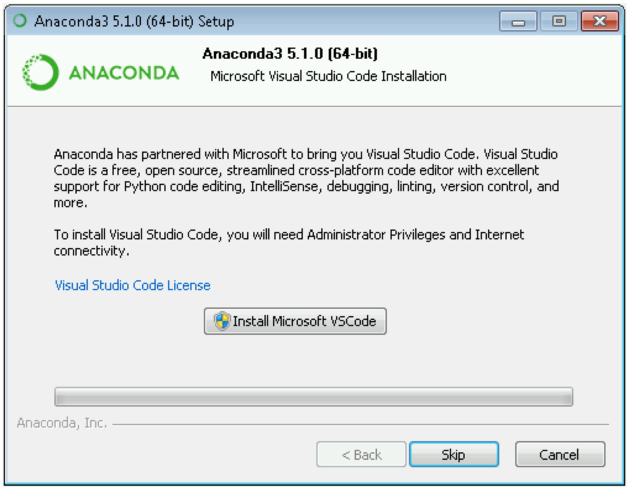
VS Code is openly extensible and many extensions are available. When you install VS Code with Anaconda it comes with the Python Extension for Visual Studio Code for editing Python code, and the RedHat YAML extension for editing YAML files such as conda configuration files.

When installing Anaconda, you may select whether to install VS Code. VS Code is not offered during silent mode

Anaconda installs, but you can easily install VS Code after the Anaconda install completes.

The Anaconda Linux installer will offer to install VS Code if your Linux distribution is in this list and at or above these versions:

Distribution	Version
CentOS	7
Debian	8
Fedora	23
openSUSE	42.1
RHEL	7
SLES	12
Ubuntu	14.04



Anaconda3 5.1.0 (64-bit) Setup

Anaconda3 5.1.0 (64-bit)

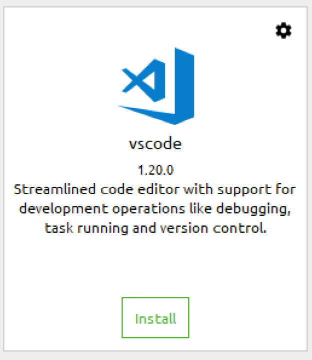
Microsoft Visual Studio Code Installation

Anaconda has partnered with Microsoft to bring you Visual Studio Code. Visual Studio Code is a free, open source, streamlined cross-platform code editor with excellent support for Python code editing, IntelliSense, debugging, linting, version control, and more.

To install Visual Studio Code, you will need Administrator Privileges and Internet connectivity.

Visual Studio Code License

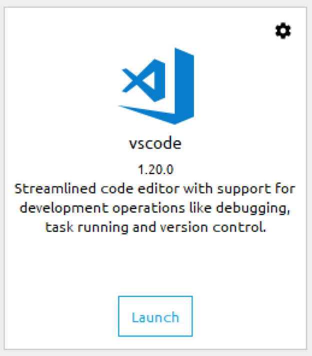
In Anaconda Navigator version 1.7 or higher, use the VS Code tile on the home screen to install or launch VS Code.



vscode

1.20.0

Streamlined code editor with support for development operations like debugging, task running and version control.



vscode

1.20.0

Streamlined code editor with support for development operations like debugging, task running and version control.

When you launch VS Code from Navigator, VS Code is configured to use the Python interpreter in the currently selected environment.

In addition to VS Code, Anaconda fully supports *Spyder*, Jupyter Notebook, and other IDEs.

Wing IDE

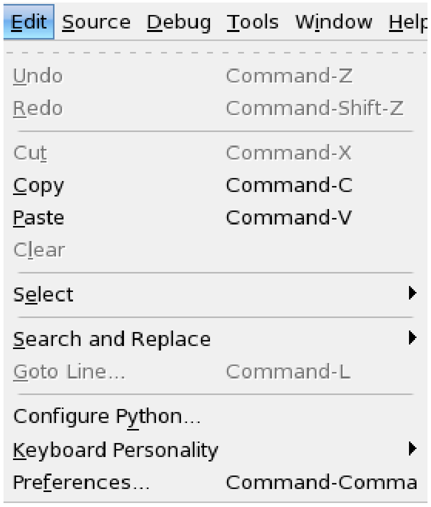
Wing IDE is an IDE designed specifically for the Python programming language. Wing IDE is offered in a paid Pro version and in free Personal and 101 versions.

To set up your Wing IDE installation to use Anaconda:

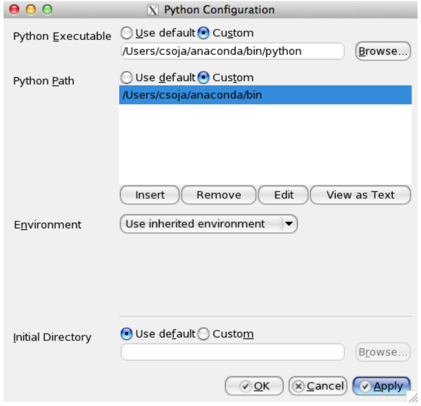
1. Navigate to the Python configuration window:

   - In Wing 101, in the **Edit** menu, select Configure Python.
   - In Wing Personal and Wing Pro, in the **Project** menu, select Project Properties.

   The Wing 101 menu looks like this:



2. Next to Python Executable, click the Custom radio button.
3. Click the Browse button and navigate to your *the Anaconda Python interpreter path*.
4. For Python Path, click the Custom radio button.
5. Click the Insert button and browse to your *Anaconda Python interpreter path*.
6. Click the OK button:



Python Configuration

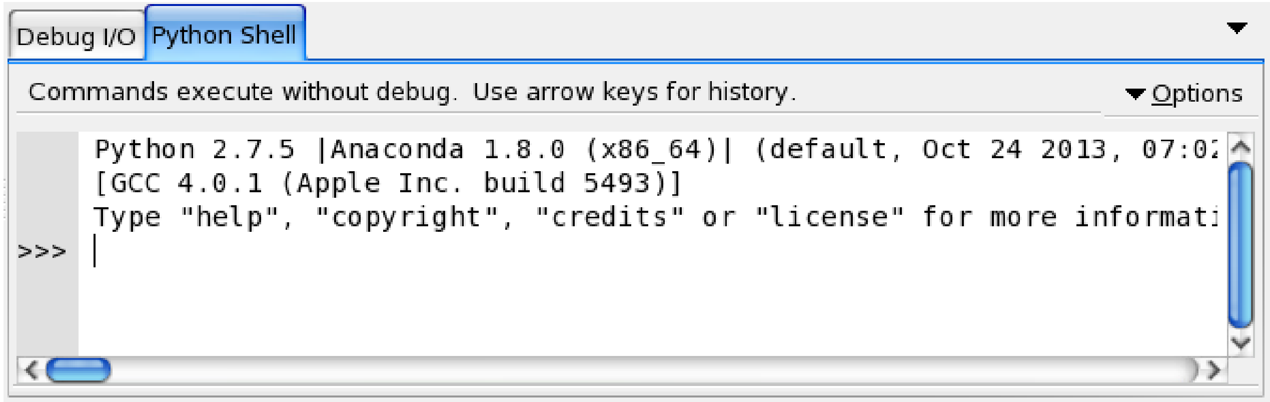
Python Executable ○ Use default ● Custom
/Users/csoja/anaconda/bin/python

Python Path ○ Use default ● Custom
/Users/csoja/anaconda/bin

Environment: Use inherited environment

Initial Directory ● Use default ○ Custom

7. If you are prompted to reload your Python shell, do so. After the reload, Anaconda is displayed on the **Python Shell** tab:



Commands execute without debug. Use arrow keys for history.

```
Python 2.7.5 |Anaconda 1.8.0 (x86_64)| (default, Oct 24 2013, 07:0
[GCC 4.0.1 (Apple Inc. build 5493)]
Type "help", "copyright", "credits" or "license" for more informati
>>>
```

Wing IDE is now set up to Anaconda's Python.

For more information, see the Wing IDE documentation.

Finding your Anaconda Python interpreter path

IDEs often require you to specify the path to your Python interpreter.

This path varies according to which operating system version and which Anaconda version you use, so you will need to search your file system to find the correct path to your Python interpreter.

You can search for the Python interpreter with your operating system's file manager, such as File Explorer on Windows, Finder on macOS or Nautilus on Ubuntu Linux.

You can also use the command line to show the location of the Python interpreter in the active conda environment.

Windows:

1. From the Start Menu open the Anaconda Prompt.
2. If you want the location of a Python interpreter for a conda environment other than the root conda environment, run `activate environment-name`.

   NOTE: Replace `environment-name` with the name of the environment you wish to activate.
3. Run `where python`.

macOS and Linux:

1. Open a Terminal window.
2. If you want the location of a Python interpreter for a conda environment other than the root conda environment, run `source activate environment-name`.

   NOTE: Replace `environment-name` with the name of the environment you wish to activate.
3. Run `which python`.

EXAMPLES:

- Windows 10 with Anaconda3 and username "jsmith"– `C:\Users\jsmith\Anaconda3\python.exe`.

  The Python image in a conda environment called "my-env" might be in a location such as `C:\Users\jsmith\Anaconda3\envs\my-env\python.exe`.
- macOS– `~/anaconda/bin/python` or `/Users/jsmith/anaconda/bin/python`.
- Linux– `~/anaconda/bin/python` or `/home/jsmith/anaconda/bin/python`.

Instead of `anaconda`, the folder in your home directory might be named one of the following:

- `anaconda2`.
- `anaconda3`.

If you have installed Miniconda instead of Anaconda, the folder might be named:

- `miniconda`.
- `miniconda2`.
- `miniconda3`.

Excel plug-ins for Anaconda

Anaconda on Windows comes ready to interact with Microsoft Excel–quickly, intuitively and powerfully. You can use one of the many included packages in Anaconda or you can connect to an outside integration tool to bring the power of Python into Excel.

The packages described here are available only on Windows unless otherwise noted. Like most Anaconda packages, this software is written by third-party open source development teams around the world. Anaconda collects and curates these programs, builds them into conda packages, and distributes them through the Anaconda platform so our users can enjoy the benefits of easy installation, version control, package management and environment management.

What tool should I use?

For a versatile, all-purpose tool, Anaconda includes xlwings, which incorporates the following features:

- Drives Excel interactively from an IPython Session/Notebook.
- Performs one-line conversion to and from a NumPy array or pandas DataFrame.
- Uses Python as Excel's computation backend by wrapping Python function calls in VBA macros.
- Easily shares Python-integrated Excel workbooks with collaborators who are also running Anaconda, with no additional installation needed.

NOTE: The xlwings package is in active development. Additional features may be added in the future. This package is available for Windows and macOS platforms.

To export data from a Python object into Excel or import the contents of an Excel spreadsheet to perform calculations or visualizations in Python, Anaconda includes the following libraries and modules:

- openpyxl–Read/write Excel 2007 xlsx/xlsm files.
- xlrd– Extract data from Excel spreadsheets–.xls and .xlsx, versions 2.0 onwards–on any platform.
- xlsxwriter–Write files in the Excel 2007+ XLSX file format.
- xlwt–Generate spreadsheet files that are compatible with Excel 97/2000/XP/2003, OpenOffice.org Calc and Gnumeric.

To determine which one best suits your needs, see the documentation for each library or module.

Python-Excel tools not included in Anaconda

- DataNitro–Offers full-featured Python-Excel integration, including user-defined functions (UDFs). Trial downloads are available, but users must purchase a license.
- ExcelPython–A free, open-source library that lets you write UDFs and macros in Python, as well as load Python modules, call methods and manipulate objects from VBA without modifying the original Python code.
- XLLoop–Open source software that implements UDFs that are hosted from a server in a variety of languages, including Python, Java, C++ and R. Installation requires multiple steps to set up the provided Excel add-in and configure the UDF server.
- ExPy–Freely available demonstration software that is simple to install. Once installed, Excel users have access to built-in Excel functions that wrap Python code. Documentation and examples are provided at the site.
- PyXLL–A widely used tool that is free for personal or educational use. It implements UDFs written in Python as add-in functions for Excel.

Using default repositories

When you use a conda command that involves looking for a package to install or upgrade, by default conda searches the default repository located at https://repo.continuum.io/pkgs.

Main channel

https://repo.continuum.io/pkgs/main

Added Sept 26, 2017 with the release of Anaconda 5.0 and conda 4.3.27, the main channel includes packages built by Anaconda, Inc. with the new compiler stack. The majority of all new Anaconda, Inc. package builds are hosted here. This is the top priority channel in conda's default channel list.

More info

Utilizing the New Compilers in Anaconda Distribution 5

Free channel

https://repo.continuum.io/pkgs/free

The `free` channel contains packages built without the new compiler stack. It includes packages built as far back as Fall 2012. The majority of these packages are compatible with the packages in /main. These are included in conda's default channels.

R Language channel

https://repo.continuum.io/pkgs/r

Mirror: https://anaconda.org/r

Microsoft R Open conda packages and Anaconda, Inc.'s R conda packages. This channel is included in conda's "defaults" channel. When creating new environments, MRO is now chosen as the default R implementation.

MRO channel

https://repo.continuum.io/pkgs/mro

Mirror: https://anaconda.org/mro

As of Dec. 19, 2017, this is an empty channel. Packages in this channel have been moved to pkgs/mro-archive. New MRO packages are in the pkgs/r channel.

Pro channel

https://repo.continuum.io/pkgs/pro

Now deprecated, though still available in conda's default channels. Packages in this channel were once sold commercially, but are now open source and available without charge. The last package was updated Feb. 2017. Includes MKL Optimizations, IOPro and Accelerate.

More info

Open sourcing Anaconda Accelerate Working Efficiently with Big Data in Text Formats Using Free Software Anaconda 2.5 Release – now with MKL Optimizations

Anaconda-Extras channel

https://anaconda.org/anaconda-extras

This channel contains packages custom built for customers by Anaconda, Inc.

Archive channel

https://repo.continuum.io/pkgs/archive

Sometimes a package that is released onto one of the other channels has an issue that forces Anaconda, Inc. to remove it from the channel. In these cases, the package is archived to this channel for anyone who still needs it.

MSYS2 channel

https://repo.continuum.io/pkgs/msys2

Mirror: https://anaconda.org/msys2

Windows only - included in conda's default channels. Necessary for Anaconda, Inc.'s R conda packages and some others in /main and /free. It provides a bash shell, Autotools, revision control systems and the like for building native Windows applications using MinGW-w64 toolchains.

Anaconda channel on anaconda.org

https://anaconda.org/anaconda

The Anaconda channel on anaconda.org is a mirror of the packages available in https://repo.continuum.io/pkgs/main, https://repo.continuum.io/pkgs/free, and https://repo.continuum.io/pkgs/pro.

Working with GPU packages

The Anaconda Distribution includes several packages that use the GPU as an accelerator to increase performance, sometimes by a factor of five or more. These packages can dramatically improve machine learning and simulation use cases, especially deep learning.

While both AMD and NVIDIA are major vendors of GPUs, NVIDIA is currently the most common GPU vendor for machine learning and cloud computing. The information on this page applies only to NVIDIA GPUs. As of August 27th, 2018, experimental AMD GPU packages for Anaconda are in progress but not yet officially supported.

GPU compatibility

GPU acceleration requires the author of a project such as TensorFlow to implement GPU-specific code paths for algorithms that can be executed on the GPU. A GPU-accelerated project will call out to NVIDIA-specific libraries for standard algorithms or use the NVIDIA GPU compiler to compile custom GPU code. Only the algorithms specifically modified by the project author for GPU usage will be accelerated, and the rest of the project will still run on the CPU.

For most packages, GPU support is either a compile-time or run-time choice, allowing a variant of the package to be available for CPU-only usage. When GPU support is a compile-time choice, Anaconda will typically need to build two versions of the package, to allow the user to choose between the "regular" version of the project that runs on CPU only and the "GPU-enabled" version of the project that runs on GPU.

Due to the different ways that CUDA support is enabled by project authors, there is no universal way to detect GPU support in a package. For many GPU-enabled packages, there is a dependency on the `cudatoolkit` package. Other packages such as Numba do not have a `cudatoolkit` dependency, because they can be used without the GPU.

Hardware requirements

NVIDIA released the CUDA API for GPU programming in 2006, and all new NVIDIA GPUs released since that date have been CUDA-capable regardless of market. Although any NVIDIA GPU released in the last 5 years will technically work with Anaconda, these are the best choices for machine learning and specifically model training use cases:

- Tesla P100 or V100
- Titan Xp or V
- GeForce 1080 or 1080 Ti
- Various recent Quadro models

Deployed models do not always require a GPU. When a GPU is required for a deployed model, there are other Tesla GPU models that are more optimized for inference than training, such as the Tesla M4, M40, P4 and P40.

Cloud and on-premise data center deployments require Tesla cards, whereas the GeForce, Quadro, and Titan options are suitable for use in workstations.

Most users will have an Intel or AMD 64-bit CPU. We recommend having at least two to four times more CPU memory than GPU memory, and at least 4 CPU cores to support data preparation before model training. There are a limited number of Anaconda packages with GPU support for IBM POWER 8/9 systems as well.

Software requirements

The best performance and user experience for CUDA is on Linux systems, and Windows is also supported. No Apple computers have been released with an NVIDIA GPU since 2014, so they generally lack the memory for machine learning applications and only have support for Numba on the GPU.

Anaconda requires that the user has installed a recent NVIDIA driver that meets the version requirements in the table below. Anaconda does not require the installation of the CUDA SDK.

Ubuntu and some other Linux distributions ship with a third party open source driver for NVIDIA GPUs called Nouveau. CUDA requires replacing the Nouveau driver with the official closed source NVIDIA driver.

All other CUDA libraries are supplied as conda packages.

GPU-enabled packages are built against a specific version of CUDA. Currently supported versions include CUDA 8, 9.0 and 9.2. The NVIDIA drivers are designed to be backward compatible to older CUDA versions, so a system with NVIDIA driver version 384.81 can support CUDA 9.0 packages and earlier. As a result, if a user is not using the latest NVIDIA driver, they may need to manually pick a particular CUDA version by selecting the version of the `cudatoolkit` conda package in their environment. To select a `cudatoolkit` version, add a selector such as `cudatoolkit=8.0` to the version specification.

Required NVIDIA driver versions, excerpted from the NVIDIA CUDA Toolkit Release Notes:

CUDA Version	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 8.0 (8.0.61 GA2)	>= 375.26	>= 376.51
CUDA 9.0 (9.0.76)	>= 384.81	>= 385.54
CUDA 9.2 (9.2.88)	>= 396.26	>= 397.44
CUDA 9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26

Sometimes specific GPU hardware generations have a minimum CUDA version. As of August 27th, 2018, the only relevant constraint is that the Tesla V100 and Titan V (using the "Volta" GPU architecture) require CUDA 9 or later.

Available packages

TensorFlow

TensorFlow is a general machine learning library, but most popular for deep learning applications. There are three supported variants of the `tensorflow` package in Anaconda, one of which is the NVIDIA GPU version. This is selected by installing the meta-package `tensorflow-gpu`:

```
conda install tensorflow-gpu
```

Other packages such as Keras depend on the generic `tensorflow` package name and will use whatever version of TensorFlow is installed. This makes it easy to switch between variants in an environment.

PyTorch

PyTorch is another machine learning library with a deep learning focus. PyTorch detects GPU availability at run-time, so the user does not need to install a different package for GPU support.

```
conda install pytorch
```

Caffe

Caffe was one of the first popular deep learning libraries.

```
conda install caffe-gpu
```

Chainer/CuPy (Linux only)

Chainer is a deep learning library that uses NumPy or CuPy for computations.

```
conda install chainer
```

Chainer's companion project CuPy is a GPU-accelerated clone of the NumPy API that can be used as a drop-in replacement for NumPy with a few changes to user code. When CuPy is installed, Chainer is GPU-accelerated. CuPy can also be used on its own for general array computation.

```
conda install cupy
```

XGBoost

XGBoost is a machine learning library that implements gradient-boosted decision trees. Training several forms of trees is GPU-accelerated.

```
conda install py-xgboost-gpu
```

MXNet

MXNet is a machine learning library supported by various industry partners, most notably Amazon. Like TensorFlow, it comes in three variants, with the GPU variant selected by the `mxnet-gpu` meta-package.

```
conda install mxnet-gpu
```

Numba

Numba is a general-purpose JIT compiler for Python functions. It provides a way to implement custom GPU algorithms in purely Python syntax when the `cudatoolkit` package is present.

```
conda install numba cudatoolkit
```

GPU support in Anaconda Enterprise

GPU-enabled conda packages can be used in AE 5 projects when the cluster has resource profiles which include GPUs. For more details see the GPU support section of the AE 5 FAQ.

Cheatsheet

Download the `Anaconda Cheat Sheet` (48 KB PDF) for a quick guide to using Anaconda.

Troubleshooting

If you have a troubleshooting issue that is not listed here, obtain free support for Anaconda through the Anaconda mailing list. For Anaconda installation or technical support options, visit our support offerings page.

You may also wish to see the *Anaconda Navigator Troubleshooting guide*.

Anaconda installer download problems

Cause

The Anaconda installer files are large (over 300 MB), and some users have problems with errors and interrupted downloads when downloading large files.

Solution

One option is to download and install the smaller Miniconda (under 60MB) and then use the command `conda install anaconda` to download and install all the remaining packages in Anaconda. If the package downloads are interrupted, just run `conda install anaconda` again. Conda only downloads the packages that were not finished in any previous attempts.

A second option is to download the large Anaconda installer file, and restart it if the download is interrupted or you need to pause it.

Windows

If you use Internet Explorer:

1. Click the Settings icon.
2. Click "View Downloads" to open the Download Manager.
3. Click on the "Resume" button next to the stopped download to restart downloading. The download resumes at the point where it stopped.

If you use Edge browser:

1. In Windows Explorer, open your downloads folder. There will be temporary files there associated with the partial downloads. Delete all of the temporary files except for the download you want to resume.

2. In Edge, click the file to download it again. Pause the download but do not cancel it.
3. In Windows Explorer, open your downloads folder. You will see two files: the partially downloaded file from earlier, and the paused download you just started. Copy the name of the file you just started, delete this file, and rename the other file with the copied name.
4. In Edge, resume the download.

If you use Chrome browser:

Download the plugin for Chrome called Chrono Download manager. In your Chrome browser, go to https://chrome.google.com/webstore/category/extensions, search on “Chrono Download” and select, “Add to Chrome.”

To resume the download using Chrono Download, from your top browser menu, click on the Chrome menu button, then click “Downloads.” Select the filename, then click “Resume” to restart your download.

macOS and Linux

- In your terminal window, download the file with the command `curl -O FILENAME`.

  NOTE: Replace FILENAME with the full path and name of the file, including `http://` or `https://`.
- To pause the download, use `CTRL-c`.

  NOTE: While a download is paused, you can shut down or restart your computer.
- When ready to resume your download, use `curl -O -C FILENAME`.

  Where “-C” is the option for “continue”. You can pause and restart a download as many times as you wish.

Cannot open Anaconda Prompt after installation

I get an error message that says “activate.bat is not a recognized file or command”.

Cause

Anaconda 5.0.1 sometimes does not install completely on Windows.

Solution

Until a new version is released, you can install Miniconda, and then use conda to install the rest of the packages in Anaconda with these instructions:

Open the command prompt (Windows key + the R key on your keyboard) which brings up the Run. . . dialog box. Enter `cmd.exe` and then press enter)

Copy the following text:

```
cd %UserProfile%
powershell -command "& { (New-Object Net.WebClient).DownloadFile('https://repo.
↪continuum.io/miniconda/Miniconda3-latest-Windows-x86_64.exe', 'mc3.exe') }"
start /wait "" mc3.exe /InstallationType=JustMe /AddToPath=0 /RegisterPython=0 /
↪NoRegistry=0 /S /D=%UserProfile%\anaconda3
%UserProfile%\anaconda3\Scripts\activate.bat
conda install -y anaconda=5.0.1 conda-build _ipyw_jlab_nb_ext_conf
```

Then paste it into the command prompt window.

NOTE: This installs to a subdirectory in your User directory named `anaconda3`. If you use a different directory, replace `anaconda3` with the actual name.

Cannot see Anaconda menu shortcuts after installation on Windows

After installing on Windows, in the Windows Start menu I cannot see Anaconda prompt, Anaconda Cloud and Navigator shortcuts.

Cause

This is caused by the way Windows updates the Start menu.

Solution

If start menu shortcuts are missing, Microsoft recommends rebooting your computer or restarting Windows Explorer.

Windows error: Failed to create Anaconda menus or Failed to add Anaconda to the system PATH

During installation on a Windows system, a dialog box appears that says "Failed to create Anaconda menus, Abort Retry Ignore" or "Failed to add Anaconda to the system PATH." There are many possible Windows causes for this.

Solution

Try these solutions, in order:

- Do not install on a PATH longer than 1024 characters
- Turn off anti-virus programs during install, then turn back on
- Uninstall all previous Python installations
- Clear all PATHs related to Python in sysdm.cpl file
- Delete any previously set up Java PATHs
- If JDK is installed, uninstall it.

I'm having trouble with the Anaconda installer on Windows. How can I debug my issue?

Cause

The cause could be any number of issues.

Solution

Anaconda 4.4 added a feature to the Windows installer so that the "verbose" install information is printed out to a special debug stream via the Win32 API function OutputDebugStream. To see these messages, during installation you need to run the Microsoft utility https://technet.microsoft.com/en-us/sysinternals/debugview.aspx. This may provide useful clues for troubleshooting or submitting bug reports.

Cannot get conda to run after installing

You may get "conda not found" or "conda is not recognized as an internal or external command" or a similar message, and you cannot execute conda in a terminal window regardless of what path you are on.

Cause

Most likely when you were installing Anaconda or Miniconda, you answered "NO" to the question whether or not to prepend the conda prompt to your path.

Solution

Uninstall and then reinstall Anaconda or Miniconda, answering "YES" to the question about prepending the conda prompt.

Or, you can manually edit your .bashrc file to prepend the Anaconda or Miniconda install location. Open a text editor and in your home directory, locate the hidden file .bashrc. Add this line to it and save:

```
export PATH=/Users/your-username/anaconda3/bin:$PATH
```

NOTE: Replace "your-username" with your username.

Close your terminal window and re-open before running a conda command.

Insecure Platform Warning

Cause

"InsecurePlatformWarning" appears only when the installed version of Python is older than version 2.7.9. This message warns only that the validity of the SSL connection is not being verified. It should not affect your package downloads.

Solution

To resolve this on Windows, install the updated package `ndg-httpsclient`:

```
conda install ndg-httpsclient
```

NOTE: When initially installing this package, you receive the SSL warning again. Once it is installed, the package will prevent the warnings.

Error message: Conda: command not found, on macOS or Linux

Cause

Problems with the PATH environment variable can cause "conda: command not found" errors or a failure to load the correct versions of python or ipython.

Solution

1. Find the location of your Anaconda binary directory.
2. In your home directory, in the `.bashrc` file, add a line to add that location to your PATH.
3. Close and then re-open your terminal windows.

EXAMPLE: A user with the user name "jsmith" on a Linux machine whose Anaconda binary directory is `~/anaconda` would add this line to the `.bashrc` file:

```
export PATH="/home/jsmith/anaconda/bin:$PATH"
```

Anaconda interfering with other software on Windows

Cause

If a user chooses to add Anaconda to the Windows PATH, this can cause programs to use the new Anaconda versions of software such as Python and not the versions that were already in place. In some cases this can cause incompatibility and errors.

Solution

We recommend not adding Anaconda to the Windows PATH. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

MKL Trial warning is displayed even though MKL-linked packages are now free and installed by default

See Dismissing MKL Trial warnings.

Error message: Permission denied, when loading MKL shared libraries

See Resolving MKL shared library permission denied errors.

Error message on Miniconda install: Already installed

Cause

This situation can occur if you are getting a conda error and you want to reinstall Miniconda to fix it.

Solution

For macOS and Linux, download and install the appropriate Miniconda for your operating system from the Miniconda download page using the force or -f option:

```
bash Miniconda3-latest-MacOSX-x86_64.sh -f
```

NOTE: For `Miniconda3-latest-MacOSX-x86_64`, substitute the appropriate filename and version for your operating system.

NOTE: Be sure that you install to the same location as your existing install so it overwrites the core conda files and does not install a duplicate in a new folder.

Conda update anaconda command does not install the latest version of Anaconda

Cause

For users who have installed packages that are not compatible with the latest version of the Anaconda metapackage, running `conda update anaconda` updates the Anaconda metapackage to the latest compatible version, but this may not be the latest version.

Solution

Obtain a list of the conflicting packages by running `conda update anaconda` or `conda install anaconda=5.2`.

NOTE: Replace `5.2` with the latest version number.

Once you know which packages are conflicting, you can update all current packages without upgrading to the latest version of Anaconda, or you can remove the conflicting packages and then upgrade to the latest version of Anaconda.

To update all current packages without upgrading to the latest version of Anaconda:

1. Use `conda remove anaconda` to remove the Anaconda metapackage itself. (This will not remove any of the packages included with Anaconda.)
2. Use `conda update --all` to update all currently installed packages.

To remove the conflicting packages and upgrade to the latest version of Anaconda:

1. Remove the conflicting packages by running `conda remove package-name` for each one.

   NOTE: Replace `package-name` with the name of the package.
2. Run `conda update anaconda`.

Linking problems when Python extensions are compiled with gcc

Cause

When compiling Python extensions with gcc on Windows, linking problems may result.

Solution

To resolve these linking problems, use the mingw import library–the conda package libpython–which Anaconda builds and includes with the Anaconda Distribution.

Error message: Unable to remove files

When trying to update or install packages with conda, you may see an error message such as:

```
Error: Unable to remove files for package: <package-name>
Please close all processes running code from conda and try again.
```

Cause

This may be caused by a file lock issue.

Solution

Before updating or installing any packages with conda, be sure to terminate any running Anaconda processes such as Spyder or IPython.

You can also force the installation of the package: `conda install -f package-name`.

NOTE: Replace `package-name` with the name of the package that you want to install.

Files left behind after uninstalling Anaconda on Windows

Cause

Some users may need to keep settings files and other users may need to delete them, so Anaconda leaves some settings files in place when it is uninstalled. Specifically, the directories `.spyder2`, `.ipython`, `.matplotlib`, and `.astropy` remain. Depending on your version of Windows these may be in `C:\Documents and Settings\Your_User_Name` or in `C:\Users\Your_User_Name`.

NOTE: Replace `Your_User_Name` with your Windows user name as it appears in the `Documents and Settings` or `Users` folder.

Solution

Manually delete any unneeded settings files.

Spyder errors or failure to launch on Windows

Cause

This may be caused by errors in the Spyder setting and configuration files.

Solution

1. Close and relaunch Spyder and see if the problem remains.
2. On the menu, select Start, then select Reset Spyder Settings and see if the problem remains.
3. Close Spyder and relaunch it from the Anaconda Prompt:

   (a) From the Start menu, open the Anaconda Prompt.

   (b) At the Anaconda Prompt, enter Spyder.

   (c) See if the problem remains.

4. Delete the directory .spyder2 and then repeat the previous steps from Step 1. Depending on your version of Windows, .spyder2 may be in C:\Documents and Settings\Your_User_Name or in C:\Users\Your_User_Name.

   NOTE: Replace Your_User_Name, with your Windows user name as it appears in the *Documents and Settings* folder.

Problems running Anaconda on macOS 10.12.2

Cause

Some installations of Anaconda on macOS 10.12.2 experienced incorrect file and directory permissions, which caused a range of errors with Navigator and other parts of Anaconda.

Solution

We recommend that any users with Anaconda on macOS 10.12.2 follow these steps:

1. Uninstall Anaconda. Open the Terminal.app or iTerm2 terminal application and remove your Anaconda directory, which will have a name such as "anaconda2" or "anaconda3", by entering a command such as this: rm -rf ~/anaconda3
2. Use a text editor such as TextEdit to open the file named .bash_profile in your home directory. If you see a line that adds Anaconda or Miniconda to your PATH environment variable, remove this line, and then save and close the file. For example, if you see a line such as export PATH="/Users/jsmith/anaconda3/bin:$PATH", remove that line.
3. Update to macOS 10.12.3 or later.
4. Reinstall Anaconda.

"execution error: localhost doesn't understand the "open location" message. (-1708)" when opening a Jupyter notebook on macOS 10.12.5

Cause

This version of macOS seems to have a bug affecting some of the ways for a program to open a web page in a browser.

Solution

Several possible workarounds have been found for this bug.

You can explicitly set the browser in ~/.jupyter/jupyter_notebook_config.py with a line such as this:

```
c.NotebookApp.browser = u'Safari'
```

Or you can copy the Jupyter notebook URL from the log messages on the command line and paste it into your browser.

Or you can set the BROWSER environment variable: `export BROWSER=/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome`

Further information is available at the Jupyter bug tracker, the Python bug tracker, and this blog post.

Missing libgfortran on Power8

Cause

Anaconda 4.4.0.0 for Power8 did not include libgfortran.

Solution

Anaconda 4.4.0.1 and later for Power8 do include libgfortran.

Upgrade to the latest version of Anaconda:

```
conda update anaconda
```

Anaconda 4.4.0.0 users who do not wish to upgrade may instead install libgfortran with this command:

```
conda install libgfortran
```

Missing libgomp on Power8

If the Python command “import numpy” fails, the system is likely missing the libgomp system library.

Cause

Most Power8 Linux distributions include libgomp, but some may not.

Solution

Check whether the system is missing libgomp with this command:

```
conda inspect linkages -n root numpy
```

If libgomp.so.1 is listed in the “not found:” section, it must be installed.

Install libgomp on Ubuntu with this command:

```
apt install libgomp1
```

Install libgomp on Red Hat Enterprise Linux (RHEL) or CentOS with this command:

```
yum install libgomp
```

Anaconda on Power8 reports "can not execute binary file"

Cause

Anaconda on Power8 only supports little endian mode. The little endian Python binary will not execute on a big endian operating system.

Solution

Install Anaconda on Power8 on a little endian Linux installation or VM.

Uninstaller requests admin privileges on Windows

Cause

After installing Anaconda or Miniconda as a non-administrative user on Windows, uninstalling may prompt for administrative privileges.

This occurs when running the uninstaller by choosing Control Panel, System, Apps & features, Python x.x.x (Miniconda3 4.3.xx 64-bit), Uninstall.

Solution

Open the Anaconda or Miniconda installation folder and run the `.exe` file uninstaller from that location. Uninstallation will complete without prompting for administrative privileges.

EXAMPLE: If you installed Miniconda3, the uninstall file will be `Uninstall-Miniconda3.exe`. Users who installed Miniconda2 or Anaconda will find a similar file with the appropriate name.

Windows permission errors when installing from Favorites folder

Cause

The Windows Favorites folder has unusual permissions and may cause permission errors with installers of any software. If you try launching the installer from the Favorites folder you may see errors such as "Setup was unable to create the directory", "Access is denied", or "Error opening file for writing".

Solution

Move the installer to a different folder and run the installer from the new folder.

1.2.4 Anaconda Navigator

Desktop Portal to Data Science

Installation

System requirements

Operating systems: Navigator supports the same operating systems that the Anaconda Distribution supports. These include:

- Windows Vista and newer, 32-bit or 64-bit.
- macOS, 64-bit.
- Ubuntu 14.04, 32-bit or 64-bit.

Python: Versions 2.7, 3.4, 3.5 and 3.6.

Installing Navigator

Navigator is automatically installed when you install *Anaconda* version 4.0.0 or higher.

If you have Miniconda or an older version of Anaconda installed, you can install Navigator from an Anaconda Prompt by running the command `conda install anaconda-navigator`.

To start Navigator, see *Getting Started*.

Uninstalling Navigator

To uninstall Anaconda Navigator, open the Anaconda Prompt (Terminal on macOS or Linux), and enter this command:

```
conda remove anaconda-navigator
```

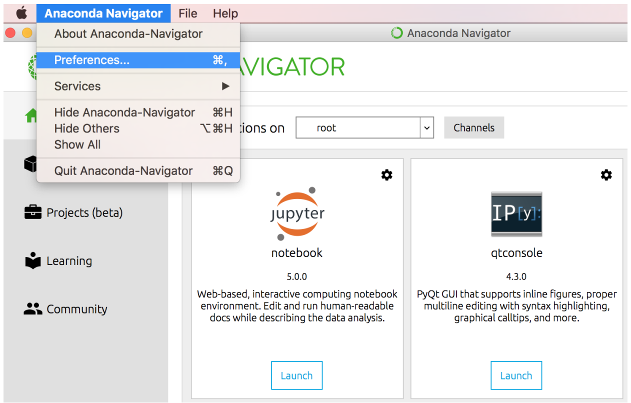
To uninstall all of Anaconda see *Uninstalling Anaconda*.

Configuring Navigator to work with a local Anaconda Repository

If you're an Anaconda Enterprise subscriber, you can configure Navigator to search for packages in your local Anaconda Repository instead of searching on Anaconda.org.

To configure Navigator to search in a local Anaconda Repository:

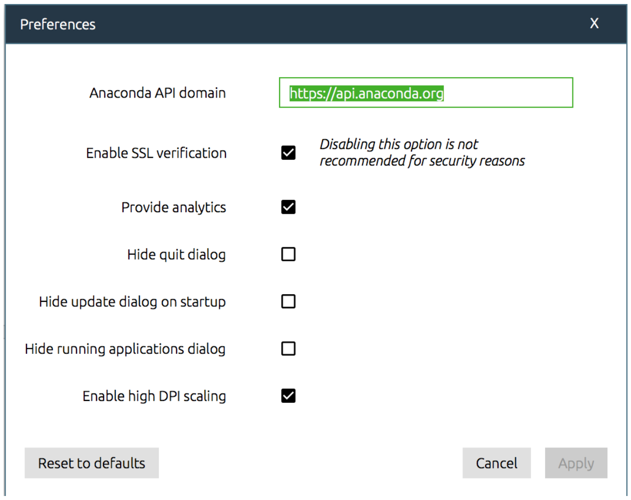
1. In the top menu bar's **Anaconda Navigator** menu (on macOS it's **File**), select Preferences.



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

2. In the Anaconda API domain field, type the address of your local Anaconda Repository.



Anaconda API domain https://api.anaconda.org

Enable SSL verification ☑ *Disabling this option is not recommended for security reasons*

Provide analytics ☑

Hide quit dialog □

Hide update dialog on startup □

Hide running applications dialog □

Enable high DPI scaling ☑

3. Click the Apply button.
4. At the top of the Navigator window, click the Sign in to Anaconda Cloud button.
5. In the Username and Password fields, type your Anaconda Repository username and password.

6. Click the Login button.

Updating Navigator

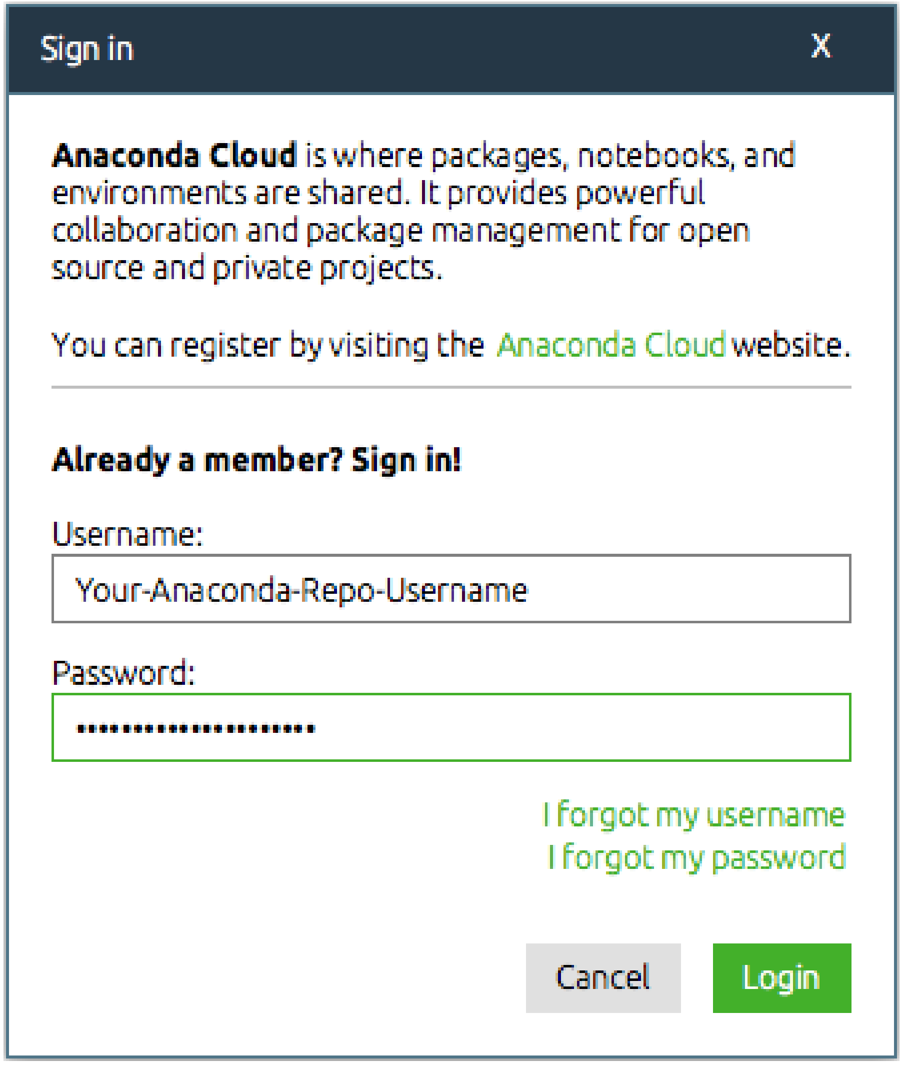
Every time Navigator starts, it checks whether a new version is available. If one is available, a dialog box is displayed that allows you to upgrade to a new Navigator version or keep your current version.

Alternate method

If you prefer, you may update Navigator manually.

Open the Anaconda prompt (Terminal on Linux or macOS).

Run this command:



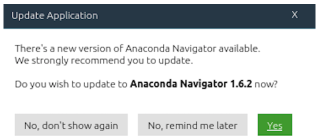
Anaconda Cloud is where packages, notebooks, and environments are shared. It provides powerful collaboration and package management for open source and private projects.

You can register by visiting the Anaconda Cloud website.

Already a member? Sign in!

Username: Your-Anaconda-Repo-Username

Password: ••••••••••••••••••••



There's a new version of Anaconda Navigator available.
We strongly recommend you to update.

Do you wish to update to **Anaconda Navigator 1.6.2** now?

```
conda update anaconda-navigator
```

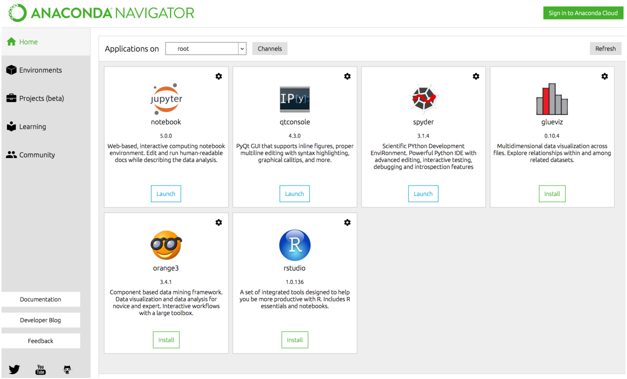
NOTE: We recommend that you keep Navigator updated to the latest version.

User guide

The Navigator user guide.

Overview

This page describes the tabs, menus, and buttons in the Anaconda Navigator window.



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

The tabs in the left column represent the main components in Navigator. Click a tab to open it.

TIP: To learn more about terms used in Anaconda, see the *Glossary*.

Home tab

The **Home** tab, shown in the image above, displays all of the available applications that you can manage with Navigator.

The first time you open Navigator, the following popular graphical Python applications are already installed or are available to install:

- Jupyter notebook
- Orange data visualization
- Qt Console
- Spyder IDE
- Glueviz multidimensional data visualization
- R Studio IDE

You can also build your own Navigator applications.

In each application box, you can:

- Launch the application–Click its Launch button.
- Install an application–Click its Install button.
- Update, remove or install a specific version of an application–Click the gear icon in the top right corner of the application box.

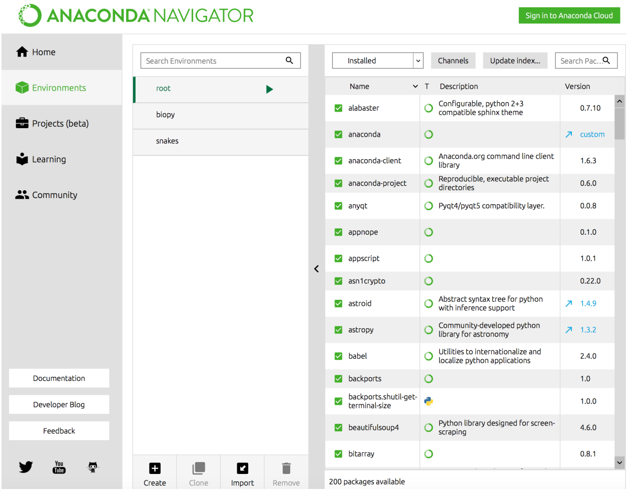
Applications are installed in the active environment, which is displayed in the "Applications on" list. To install an application in a specific environment, first select the environment in the list, then click the application's Install button. You can also create a new environment on the **Environments** tab, then return to the **Home** tab to install packages in the new environment.

Licensed applications

Some applications require licenses. To see the status of all licensed applications and add new licenses, on the **Help** menu, select License Manager. For more information, see *Managing application licenses*.

Environments tab

The **Environments** tab allows you to manage installed *environments*, *packages* and *channels*.



	Name	T	Description	Version
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.10
☑	anaconda	○		↗ custom
☑	anaconda-client	○	Anaconda.org command line client library	1.6.3
☑	anaconda-project	○	Reproducible, executable project directories	0.6.0
☑	anyqt	○	Pyqt4/pyqt5 compatibility layer.	0.0.8
☑	appnope	○		0.1.0
☑	appscript	○		1.0.1
☑	asn1crypto	○		0.22.0
☑	astroid	○	Abstract syntax tree for python with inference support	↗ 1.4.9
☑	astropy	○	Community-developed python library for astronomy	↗ 1.3.2
☑	babel	○	Utilities to internationalize and localize python applications	2.4.0
☑	backports	○		1.0
☑	backports.shutil-get-terminal-size			1.0.0
☑	beautifulsoup4	○	Python library designed for screen-scraping	4.6.0
☑	bitarray	○		0.8.1

200 packages available

The left column lists your environments. Click an environment to activate it.

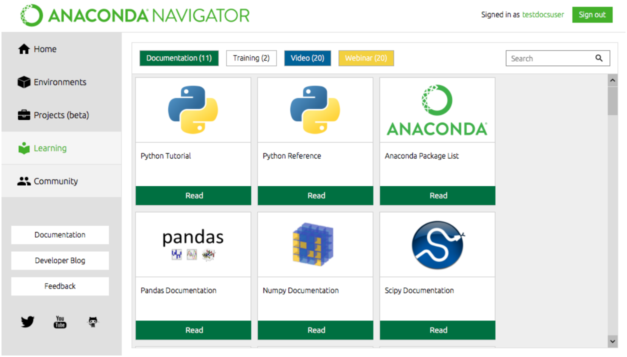
With Navigator, like with conda, you can create, export, list, remove and update environments that have different versions of Python and/or packages installed. Switching or moving between environments is called activating the environment. Only one environment is active at any point in time. For more information, see *Managing environments*.

The right column lists packages in the current environment. The default view is Installed packages. To change which packages are displayed, click the arrow next to the list, then select Not Installed, Upgradeable or All packages. For more information, see *Managing packages*.

Channels are locations where Navigator or conda looks for packages. Click the Channels button to open the Channels Manager. For more information, see *Managing channels*.

Learning tab

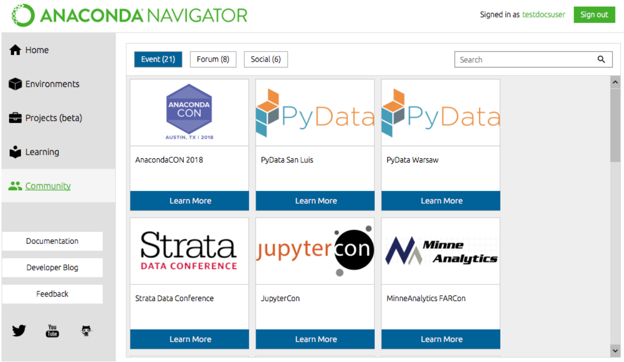
On the **Learning** tab you can learn more about Navigator, the Anaconda platform and open data science. Click the Webinars, Documentation, Video, or Training buttons, then click any item to open it in a browser window.



Community tab

On the **Community** tab you can learn more about events, free support forums and social networking relating to Navigator. Click the Events, Forum or Social buttons, then click any item to open it in a browser window.

TIP: To get help with Anaconda and Navigator from the community, join the Anaconda forum.

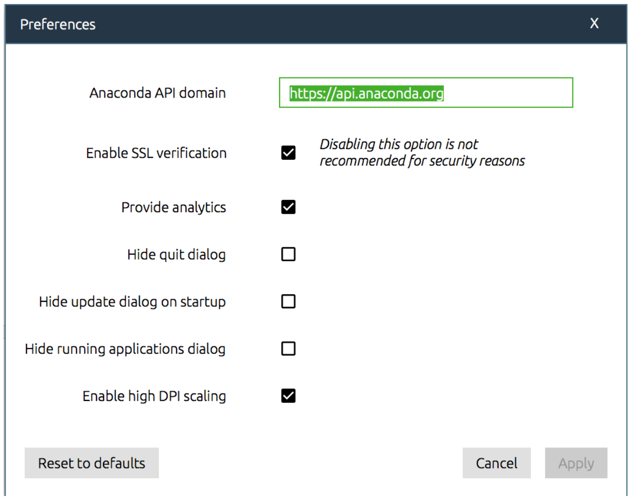


File menu

The Navigator **File** menu (on macOS **Anaconda Navigator**) contains the following options:

- About–displays information about Navigator, including a link for bug reports and feature requests. (In Linux this is in the Help menu.)
- Preferences–allows you to set your Navigator preferences. In the Preferences window you can:
  - Reset the Anaconda API domain to point to a local Anaconda repository instead of Anaconda Cloud.
  - Enable or disable SSL.
  - Toggle the option to provide personally non-identifiable information to help improve the product.
  - Hide the Quit dialog box when exiting the program.
  - Hide the Update dialog box when starting the program.
  - Hide the Close running applications dialog, which normally displays when exiting the program if there are still running applications that were launched from Navigator.
  - Modify Navigator's display with Enable High DPI scaling option. This option can be useful if Navigator isn't displaying correctly on some high DPI screens.

  TIP: Click the Reset to defaults button to change preferences back to their default values.



Anaconda API domain: https://api.anaconda.org

Enable SSL verification ☑ *Disabling this option is not recommended for security reasons*

Provide analytics ☑

Hide quit dialog □

Hide update dialog on startup □

Hide running applications dialog □

Enable high DPI scaling ☑

- Services (macOS only)–links to your computer's system preferences menu.
- Hide Anaconda Navigator (macOS only)–hides the Navigator window.
- Hide Others (macOS only)–hides all windows except the Navigator window.
- Show All (macOS only)–shows all windows including the Navigator window.
- Quit Anaconda Navigator–exits Navigator.

Help menu

The **Help** menu contains the following options:

- Search–links to your computer's Help (Windows and macOS only).
- Online Documentation–links to this documentation, which you can read in any web browser. You can also open the documentation by clicking the Documentation button at the bottom left of the Navigator window.
- License manager–allows you to add and remove licenses, and view details of licensed packages. For more information, see *Managing application licenses*.
- Logs viewer–allows you to review the logs of all actions performed in Navigator in the current session. This option displays a list of log files, including `navigator.log`, which contains Navigator application logs, and `condamanager.log`, which contains logs written by the conda-manager component.

  NOTE: A new log file is created every time you run Navigator, with a sequential number appended to the file name. More recent log files have higher numbers.

Navigator window buttons

- Sign in to Anaconda Cloud–displayed at the top right. Click to sign into Anaconda Cloud (Anaconda.org) and enable searching for packages on it. After you have logged in, the button label changes to "Signed in as [username]."
- Documentation–displayed at the bottom left. Click to open Navigator documentation in a browser.
- Developer Blog–displayed at the bottom left. Click to read what our developers have to say about Navigator development.
- Feedback–displayed at the bottom left. Click to send us feedback about Navigator and help us make it better.

Getting started with Navigator

Anaconda Navigator is a graphical user interface to the conda package and environment manager.

This 10-minute guide to Navigator will have you navigating the powerful conda program in a web-like interface without having to learn command line commands.

SEE ALSO: Getting started with conda to learn how to use conda. Compare the Getting started guides for each to see which program you prefer.

Before you start

You should have already *installed Anaconda* .

Contents

- *Starting Navigator* on Windows, macOS or Linux. 1 MINUTE
- *Managing Navigator*. Verify that Anaconda is installed and check that Navigator is updated to the current version. 1 MINUTE
- *Managing environments*. Create environments and move easily between them. 3 MINUTES
- *Managing Python*. Create an environment that has a different version of Python. 2 MINUTES

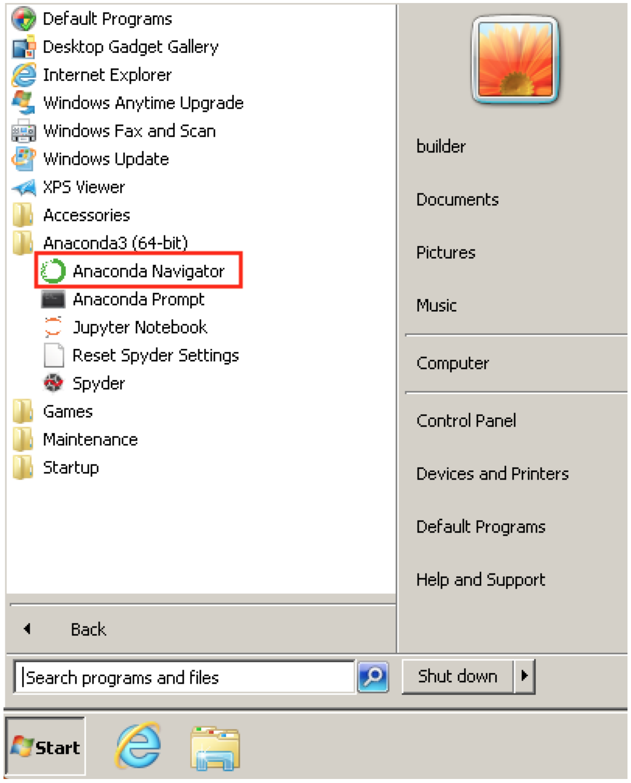
- *Managing packages*. Find packages available for you to install. Install packages. 3 MINUTES

TOTAL TIME: 10 MINUTES

Starting Navigator

Windows

- From the Start menu, click the Anaconda Navigator desktop app.
- Or from the Start menu, search for and open "Anaconda Prompt" and type the command `anaconda-navigator`.



MacOS

- Open Launchpad, then click the Anaconda-Navigator icon.
- Or open Launchpad and click the Terminal icon. Then in Terminal, type `anaconda-navigator`.

Linux

- Open a Terminal window and type `anaconda-navigator`.

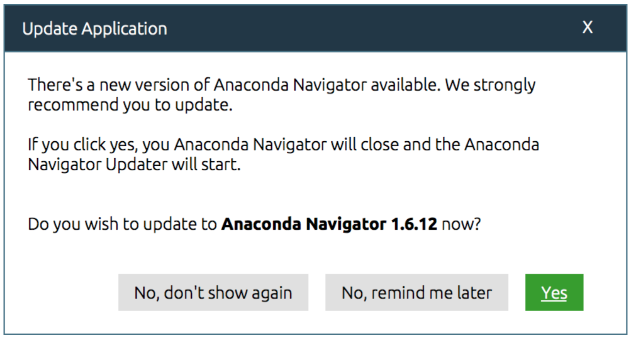
Managing Navigator

Verify that Anaconda is installed and running on your system.

- When Navigator starts up, it verifies that Anaconda is installed.
- If Navigator does not start up, go back to Anaconda installation and make sure you followed all the steps.

Check that Navigator is updated to the current version.

- When you start Navigator, it automatically checks for a new version. If Navigator finds a new version, you will see a dialog box like this:



Update Application

There's a new version of Anaconda Navigator available. We strongly recommend you to update.

If you click yes, you Anaconda Navigator will close and the Anaconda Navigator Updater will start.

Do you wish to update to **Anaconda Navigator 1.6.12** now?

Click the "Yes" button to update Navigator to the current version.

TIP: We recommend that you always keep Navigator updated to the latest version.

Managing Environments

Navigator uses conda to create separate environments containing files, packages and their dependencies that will not interact with other environments.

1. Create a new environment named snowflakes and install a package in it:

   In Navigator, click the **Environments** tab, then click the Create button.

   The Create new environment dialog box appears.

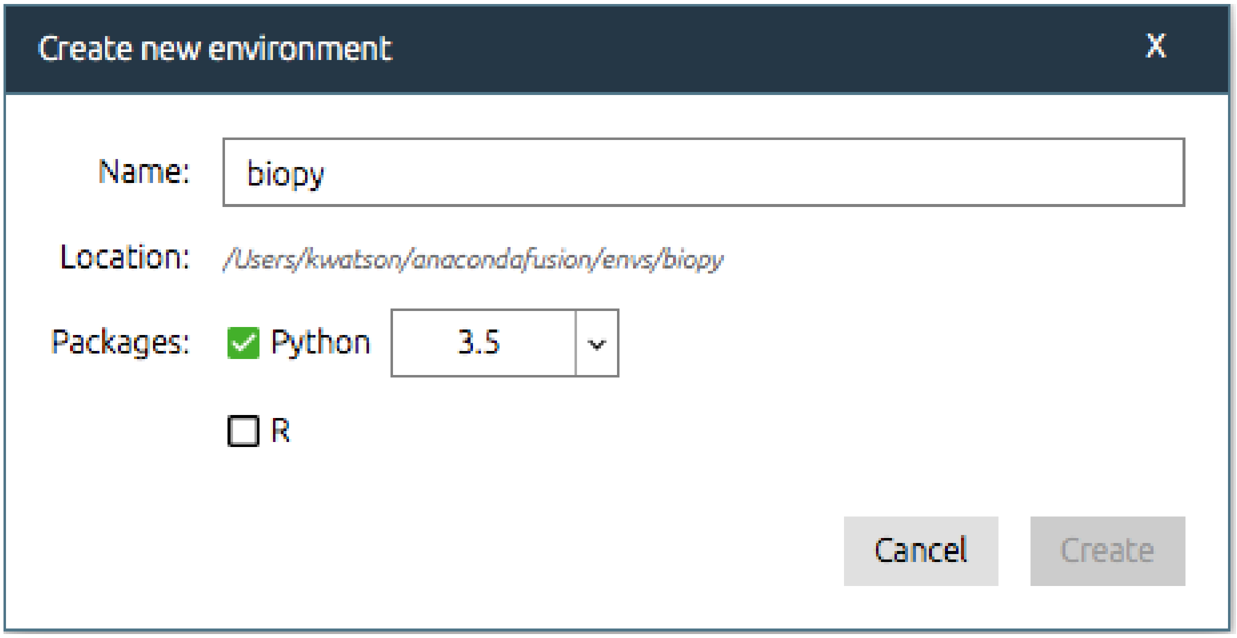
   In the Environment name field, type a descriptive name for your environment:

   Click the Create button.

   Navigator creates the new environment and activates it:

   Now you have two environments, base (the default) and snowflakes. Switch between them (activate and deactivate environments) by clicking the name of the environment you want to use.

   TIP: The active environment is the one with the arrow next to its name.

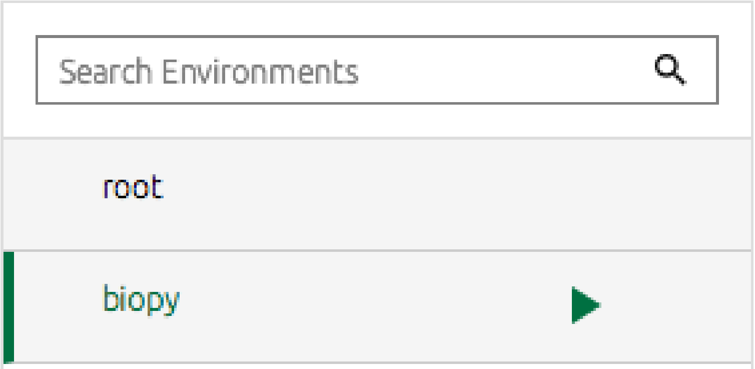


Name: biopy

Location: /Users/kwatson/anacondafusion/envs/biopy

Packages: ☑ Python 3.5

□ R



Return to the other environment by clicking its name.

Managing Python

When you create a new environment, Navigator installs the same Python version you used when you downloaded and installed Anaconda. If you want to use a different version of Python, for example Python 3.5, simply create a new environment and specify the version of Python that you want in that environment.

Create a new environment named "snakes" that contains Python 3.5:

1. In Navigator, click the **Environments** tab, then click the Create button.

   The Create new environment dialog box appears.

2. In the Environment name field, type the descriptive name "snakes" and select the version of Python you want to use from the Python Packages box (3.6, 3.5 or 2.7). Select a different version of Python than is in your other environments, base or snowflakes.

3. Click the Create button.

4. Activate the version of Python you want to use by clicking the name of that environment.

Managing packages

In this section, you check which packages you have installed, check which are available and look for a specific package and install it.

To find a package you have already installed, click the name of the environment you want to search. The installed packages are displayed in the right pane.

You can change the selection of packages displayed in the right pane at any time by clicking the drop-down box above it and selecting Installed, Not Installed, Updateable, Selected, or All.

Check to see if a package you have not installed named "beautifulsoup4" is available from the Anaconda repository (must be connected to the Internet):

On the Environments tab, in the Search Packages box, type `beautifulsoup4`, and from the Search Subset box select All or Not Installed.

To install the package into the current environment:

Check the checkbox next to the package name, then click the bottom Apply button.

The newly installed program is displayed in your list of installed programs.

More information

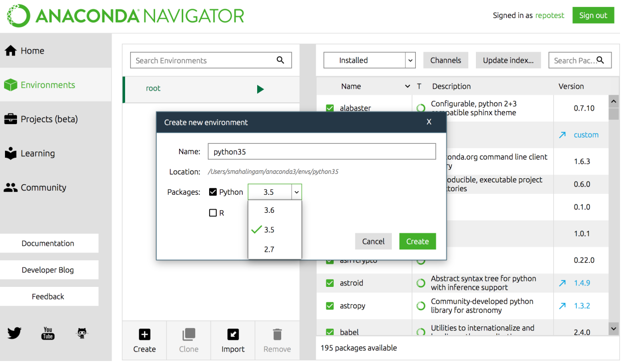
- Full documentation—https://docs.anaconda.com/anaconda/navigator/
- Free community support—https://groups.google.com/a/anaconda.com/forum/#!forum/anaconda
- Paid support options—https://www.anaconda.com/support/
- Training—https://www.anaconda.com/training/

Tutorials

General tasks:

Creating a Python 3.5 environment from Anaconda2 or Anaconda3

1. In Navigator, click the Environments tab, then click the Create button. The Create new environment dialog box appears.
2. In the Environment name field, type a descriptive name for your environment.



Name: python35

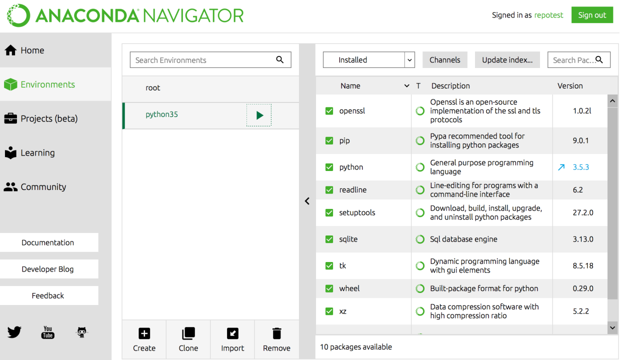
Location: /Users/smahalingam/anaconda3/envs/python35

Packages: ☑ Python 3.5

□ R

	Name	T	Description	Version
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.10
☑	astroid	○	Abstract syntax tree for python with inference support	1.4.9
☑	astropy	○	Community-developed python library for astronomy	1.3.2
☑	babel	○	Utilities to internationalize and	2.4.0

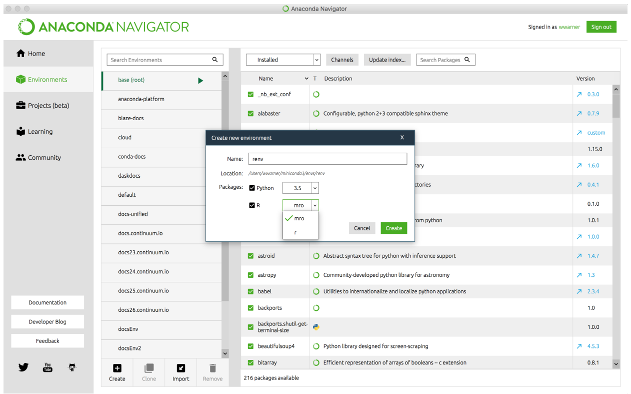
3. In the Packages list select "Python" and in the Python version list select "3.5".
4. Click the Create button.
5. Navigator creates the new environment and activates it, as shown by the highlighted green bar. All actions take place in the active environment.



	Name	T	Description	Version
☑	openssl	○	Openssl is an open-source implementation of the ssl and tls protocols	1.0.2l
☑	pip	○	Pypa recommended tool for installing python packages	9.0.1
☑	python	○	General purpose programming language	3.5.3
☑	readline	○	Line-editing for programs with a command-line interface	6.2
☑	setuptools	○	Download, build, install, upgrade, and uninstall python packages	27.2.0
☑	sqlite	○	Sql database engine	3.13.0
☑	tk	○	Dynamic programming language with gui elements	8.5.18
☑	wheel	○	Built-package format for python	0.29.0
☑	xz	○	Data compression software with high compression ratio	5.2.2

Creating an R environment and running RStudio

1. In Navigator, click the Environments tab, then click the Create button. The Create new environment dialog box appears.
2. In the Environment name field, type a descriptive name for your environment.
3. In the Packages list, select "R".
4. On the drop-down menu, select `mro` to install Microsoft R Open with the packages `mro-base` and `r-essentials`. If you prefer, you may instead select `r` to install R with the packages `r-base` and `r-essentials`. For more information please see *Using R language with Anaconda*.
5. Click the Create button.
6. Navigator creates the new environment and activates it, as shown by the highlighted green bar. All actions take place in the active environment.



Name: renv

Location: /Users/wwarner/miniconda3/envs/renv

Packages: ☑ Python 3.5

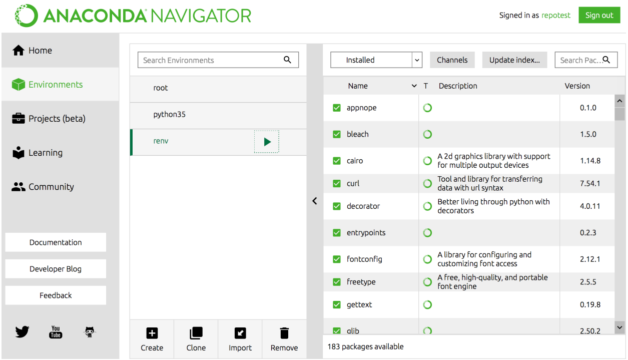
☑ R mro

✓ mro

r

Name	T	Description	Version
☑ _nb_ext_conf	○		0.3.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.9
			custom
			1.15.0
		rary	1.6.0
		ctories	0.4.1
			0.1.0
		om python	1.0.1
			1.0.0
☑ astroid	○	Abstract syntax tree for python with inference support	1.4.7
☑ astropy	○	Community-developed python library for astronomy	1.3
☑ babel	○	Utilities to internationalize and localize python applications	2.3.4
☑ backports	○		1.0
☑ backports.shutil-get-terminal-size			1.0.0
☑ beautifulsoup4	○	Python library designed for screen-scraping	4.5.3
☑ bitarray	○	Efficient representation of arrays of booleans -- c extension	0.8.1

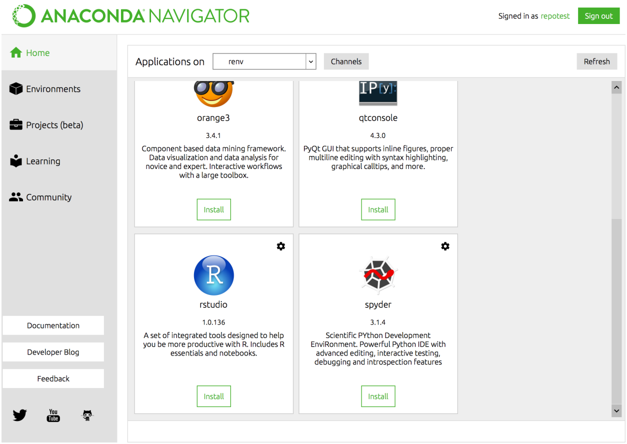
216 packages available



Name	T	Description	Version
☑ appnope	○		0.1.0
☑ bleach	○		1.5.0
☑ cairo	○	A 2d graphics library with support for multiple output devices	1.14.8
☑ curl	○	Tool and library for transferring data with url syntax	7.54.1
☑ decorator	○	Better living through python with decorators	4.0.11
☑ entrypoints	○		0.2.3
☑ fontconfig	○	A library for configuring and customizing font access	2.12.1
☑ freetype	○	A free, high-quality, and portable font engine	2.5.5
☑ gettext	○		0.19.8
☑ glib	○		2.50.2

183 packages available

7. With the new environment active, click the Home button and then click the Install button on the RStudio application.



orange3 3.4.1: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

qtconsole 4.3.0: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

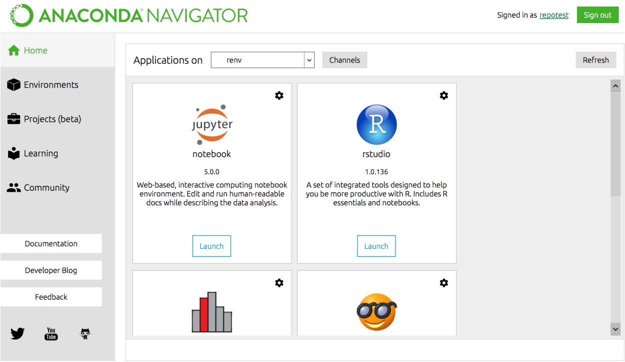
rstudio 1.0.136: A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

spyder 3.1.4: Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

8. Click Launch to run RStudio from Navigator.

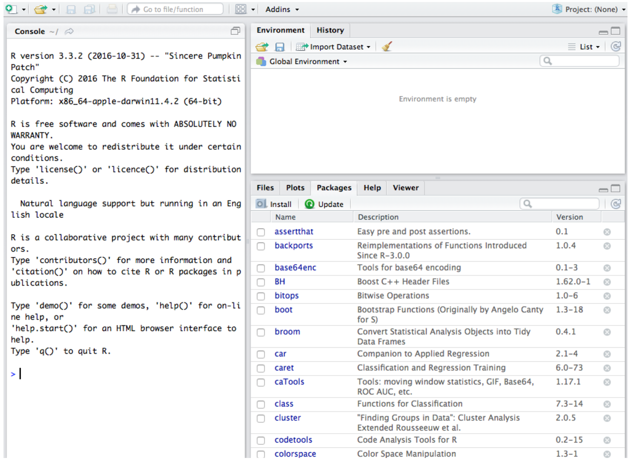
9. You will see RStudio running from Navigator.

Managing environments



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.



```
R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin
Patch"
Copyright (C) 2016 The R Foundation for Statisti
cal Computing
Platform: x86_64-apple-darwin11.4.2 (64-bit)

R is free software and comes with ABSOLUTELY NO
WARRANTY.
You are welcome to redistribute it under certain
conditions.
Type 'license()' or 'licence()' for distribution
details.

  Natural language support but running in an Eng
lish locale

R is a collaborative project with many contribut
ors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in p
ublications.

Type 'demo()' for some demos, 'help()' for on-li
ne help, or
'help.start()' for an HTML browser interface to
help.
Type 'q()' to quit R.

> |
```

	Name	Description	Version
□	assertthat	Easy pre and post assertions.	0.1
□	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.0.4
□	base64enc	Tools for base64 encoding	0.1-3
□	BH	Boost C++ Header Files	1.62.0-1
□	bitops	Bitwise Operations	1.0-6
□	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-18
□	broom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.1
□	car	Companion to Applied Regression	2.1-4
□	caret	Classification and Regression Training	6.0-73
□	caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
□	class	Functions for Classification	7.3-14
□	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.5
□	codetools	Code Analysis Tools for R	0.2-15
□	colorspace	Color Space Manipulation	1.3-1

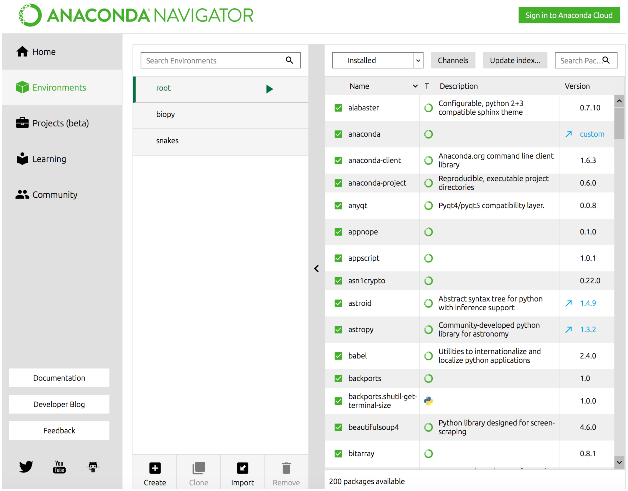
On the **Environments** tab, the left column displays your environments.

Searching for an environment

In the Search Environments box, type all or part of the environment name.

Creating a new environment

1. At the bottom of the environments list, click the Create button.
2. In the Create Environment dialog box, type a descriptive name for the new environment.



	Name	T	Description	Version
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.10
☑	anaconda	○		↗ custom
☑	anaconda-client	○	Anaconda.org command line client library	1.6.3
☑	anaconda-project	○	Reproducible, executable project directories	0.6.0
☑	anyqt	○	Pyqt4/pyqt5 compatibility layer.	0.0.8
☑	appnope	○		0.1.0
☑	appscript	○		1.0.1
☑	asn1crypto	○		0.22.0
☑	astroid	○	Abstract syntax tree for python with inference support	↗ 1.4.9
☑	astropy	○	Community-developed python library for astronomy	↗ 1.3.2
☑	babel	○	Utilities to internationalize and localize python applications	2.4.0
☑	backports	○		1.0
☑	backports.shutil-get-terminal-size			1.0.0
☑	beautifulsoup4	○	Python library designed for screen-scraping	4.6.0
☑	bitarray	○		0.8.1

200 packages available

3. Select Python, R or both for a mixed environment.
4. If applicable, choose a Python version for the new environment.
5. Click the Create button.

When you install an R language environment, the https://anaconda.org/r channel is automatically added and made active, and a new channel automatically appears in condarc.

Using an environment

1. In the environments list, click the environment name.
2. Click the arrow button next to the name. The activation options dialog appears.
3. Select one of the following options for opening the environment: Terminal, Python interpreter, IPython Console or Jupyter Notebook.

   NOTE: Some of these options may not be available if they were not installed in the environment.

Cloning an environment

1. Click the name of the environment you want to clone.
2. At the bottom of the environments list, click the Clone button.
3. Type a descriptive name for the new environment.

Importing an environment

Each environment has a YAML-formatted configuration file. If someone has given you an environment file that you want to use, for example my-environment-file.yml, and you have saved it to your computer, you can import it into Navigator.

1. At the bottom of the environments list, click the Import button.
2. In the Import dialog box, type a descriptive name for the new environment.
3. Click the file folder icon to browse to the YAML file, or type the file name, including its path.

4. Click the Import button.

Removing an environment

1. In the environments list, click the environment you want to remove.
2. At the bottom of the list, click the Remove button.

Advanced environment management

Navigator provides a convenient graphical interface for managing conda environments, channels and packages. But if you're comfortable working with Anaconda prompt (Terminal on Linux or macOS), you can access additional, advanced management features. To learn more, see Managing environments with conda.

Managing packages

On the Navigator **Environments** tab, the packages table in the right column lists the packages included in the environment selected in the left column.

NOTE: Packages are managed separately for each environment. Changes you make to packages only apply to the active environment.

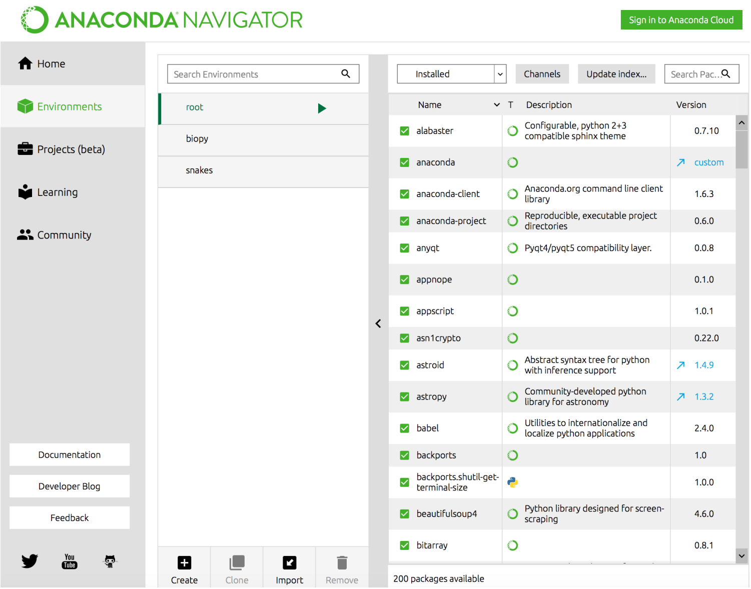
TIP: Click a column heading in the table to sort the table by package name, description, or version.

TIP: The Update Index button updates the packages table with all packages that are available in any of the enabled channels.

Filtering the packages table

By default, only Installed packages are shown in the packages table. To filter the table to show different packages, click the arrow next to Installed, then select which packages to display: Installed, Not Installed, Upgradable or All.

NOTE: Selecting the Upgradable filter lists packages that are installed and have upgrades available.



	Name	T	Description	Version
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.10
☑	anaconda	○		↗ custom
☑	anaconda-client	○	Anaconda.org command line client library	1.6.3
☑	anaconda-project	○	Reproducible, executable project directories	0.6.0
☑	anyqt	○	Pyqt4/pyqt5 compatibility layer.	0.0.8
☑	appnope	○		0.1.0
☑	appscript	○		1.0.1
☑	asn1crypto	○		0.22.0
☑	astroid	○	Abstract syntax tree for python with inference support	↗ 1.4.9
☑	astropy	○	Community-developed python library for astronomy	↗ 1.3.2
☑	babel	○	Utilities to internationalize and localize python applications	2.4.0
☑	backports	○		1.0
☑	backports.shutil-get-terminal-size			1.0.0
☑	beautifulsoup4	○	Python library designed for screen-scraping	4.6.0
☑	bitarray	○		0.8.1

200 packages available

Finding a package

In the Search Packages box, type the name of the package.

Installing a package

1. Select the Not Installed filter to list all packages that are available in the environment's channels but are not installed.

   NOTE: Only packages that are compatible with your current environment are listed.

2. Select the name of the package you want to install, or in the Version column, click the blue up arrow.
3. Click the Apply button.

TIP: If after installing a new package it doesn't appear in the packages table, select the **Home** tab, then click the Refresh button to reload the packages table.

Upgrading a package

1. Select the Upgradable filter to list all installed packages that have upgrades available.
2. Click the checkbox next to the package you want to upgrade, then in the menu that appears select Mark for Upgrade.

   OR

   In the Version column, click the blue up arrow.

3. Click the Apply button.

Installing a different package version

1. Click the checkbox next to the package whose version you want to change.

2. In the menu that appears, select Mark for specific version installation.

   If other versions are available for this package, they are displayed in a list.

3. Click the package version you want to install.
4. Click the Apply button.

Removing a package

1. Click the checkbox next to the package you want to remove.
2. In the menu that appears, select Mark for Removal.
3. Click the Apply button.

Advanced package management

Navigator provides a convenient graphical interface for managing conda environments, channels and packages. But if you're comfortable working with Anaconda prompt (Terminal on Linux or macOS), you can access additional, advanced management features. To learn more, see Managing packages with conda.

Managing channels

Channels are locations where Navigator and conda look for packages.

You can search and browse packages and channels on Anaconda Cloud.

NOTE: Navigator and conda only search for packages in active channels. You can temporarily disable a channel by making it inactive.

EXAMPLE: Let's say you want to look for packages on the "Milagro" channel of Anaconda Cloud. All three of these channel variations will look for packages in the same place on Cloud:

- By URL–https://anaconda.org/milagro.
- By conda URL–https://conda.anaconda.org/milagro.
- By name–milagro.

On the **Environment** tab, click the Channels button to open the Channels Manager.

Adding a channel

1. Click the Add button.
2. Type or paste the channel name, the URL, or the conda URL.

   NOTE: A URL can also contain an access token parameter and value. A URL will automatically be transformed to a conda URL.

3. Click the Save button.

Making a channel active or inactive

Active channels are included when you search for packages and other assets.

All channels that are active in the conda configuration file `.condarc` are active by default in Navigator. So if you're not used to thinking about channels, there's no need to change the default.

Active channels show a check mark to the left of the channel name.

To make a channel active or inactive:

1. Select or clear the checkbox next to the channel name.
2. Click the Update Channels button.

Deleting a channel

1. Click the trash can icon next to the channel name.
2. Click the Update Channels button.

Managing application licenses

To add an application license:

1. In the top navigation bar, click the **Help** menu, then select License Manager.
2. In the License Manager window, click the Add License button:
3. Browse to and select the license file on your computer.
4. Click the Open button.

If you have a problem, click the "Got a problem with your license? Please contact us." link.

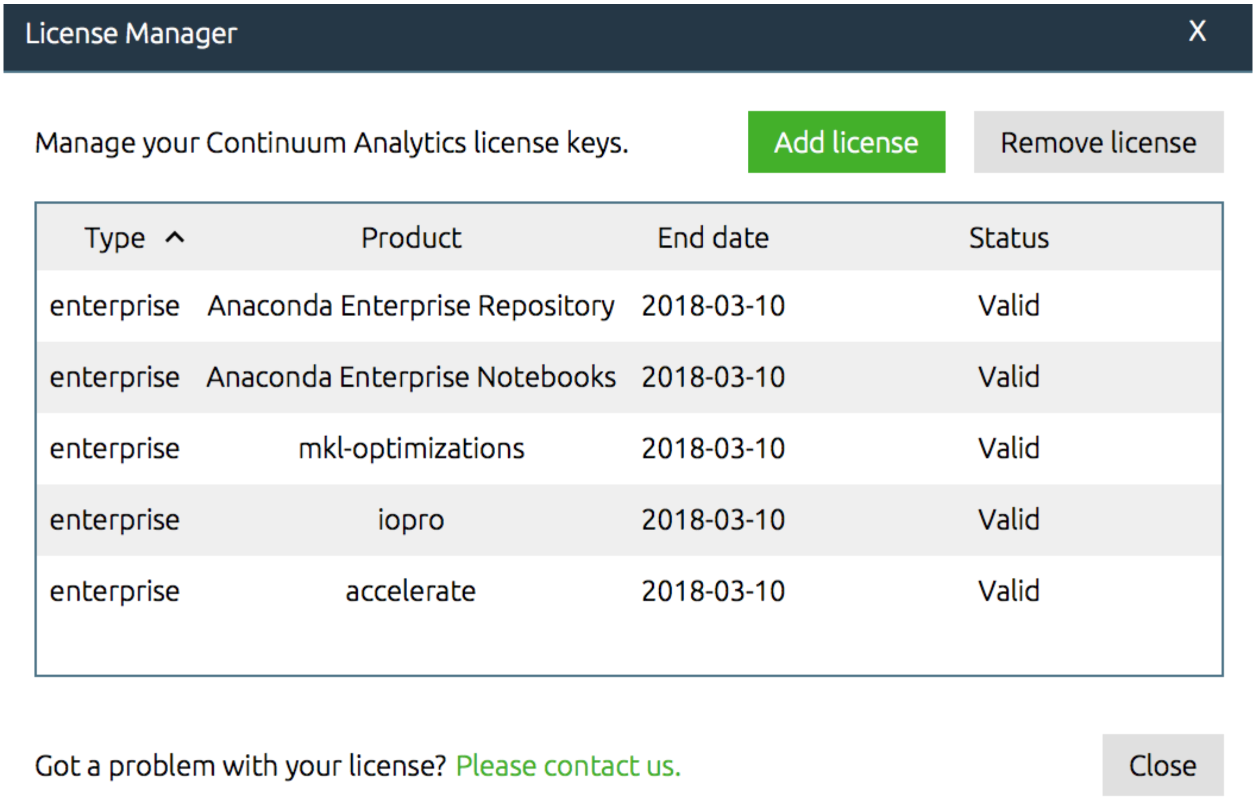
For more information, see *License installation for Anaconda*.

Using multiple versions of Python with Navigator

With Navigator you can have different versions of Python on your computer and easily switch between them by keeping them in different **environments**.

To use a different version of Python than the one you are currently using, set up a new environment:

1. In Navigator, click the Environments tab, then click the Create button. The Create new environment dialog box appears.
2. In the Environment name field, type a descriptive name for your environment.
3. In the Packages list select "Python" and in the Python version list select the version you want to use.
4. Click the Create button.

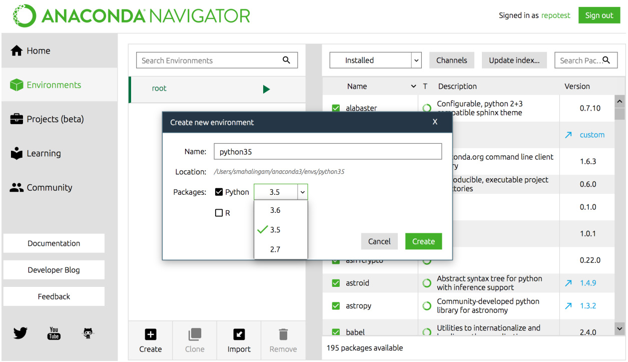


License Manager

Manage your Continuum Analytics license keys.

Type	Product	End date	Status
enterprise	Anaconda Enterprise Repository	2018-03-10	Valid
enterprise	Anaconda Enterprise Notebooks	2018-03-10	Valid
enterprise	mkl-optimizations	2018-03-10	Valid
enterprise	iopro	2018-03-10	Valid
enterprise	accelerate	2018-03-10	Valid

Got a problem with your license? Please contact us.



Create new environment

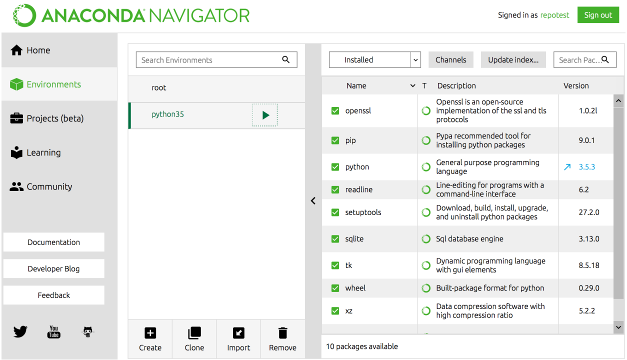
Name: python35

Location: /Users/smahalingam/anaconda3/envs/python35

Packages: ☑ Python 3.5

☐ R

5. Navigator creates the new environment and activates it, as shown by the highlighted green bar. All actions take place in the active environment.



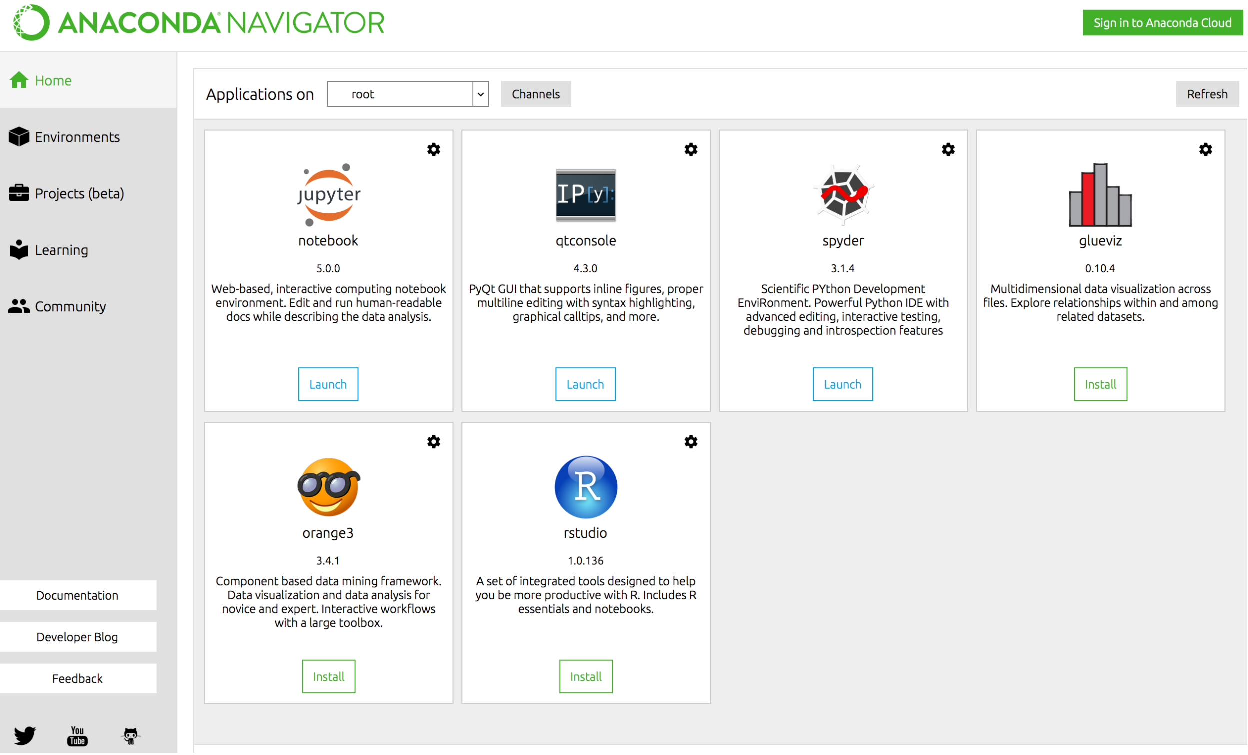
Name	T	Description	Version
☑ openssl	○	Openssl is an open-source implementation of the ssl and tls protocols	1.0.2l
☑ pip	○	Pypa recommended tool for installing python packages	9.0.1
☑ python	○	General purpose programming language	↗ 3.5.3
☑ readline	○	Line-editing for programs with a command-line interface	6.2
☑ setuptools	○	Download, build, install, upgrade, and uninstall python packages	27.2.0
☑ sqlite	○	Sql database engine	3.13.0
☑ tk	○	Dynamic programming language with gui elements	8.5.18
☑ wheel	○	Built-package format for python	0.29.0
☑ xz	○	Data compression software with high compression ratio	5.2.2

10 packages available

Searching Anaconda Cloud

To search Anaconda Cloud using Anaconda Navigator:

1. If you haven't already, sign into Anaconda Cloud by clicking the top right sign-in button. If you don't yet have an account, you can create one.
2. In Navigator, click the left Environments tab.



notebook

5.0.0

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

qtconsole

4.3.0

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

spyder

3.1.4

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

glueviz

0.10.4

Multidimensional data visualization across files. Explore relationships within and among related datasets.

orange3

3.4.1

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

rstudio

1.0.136

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

3. In the list at the top left of the packages area, select All to search All packages on Anaconda Cloud.

In the Search Packages box, type the name of the package you want to search for on Anaconda Cloud.

All packages with that text string are displayed in the Search Results.

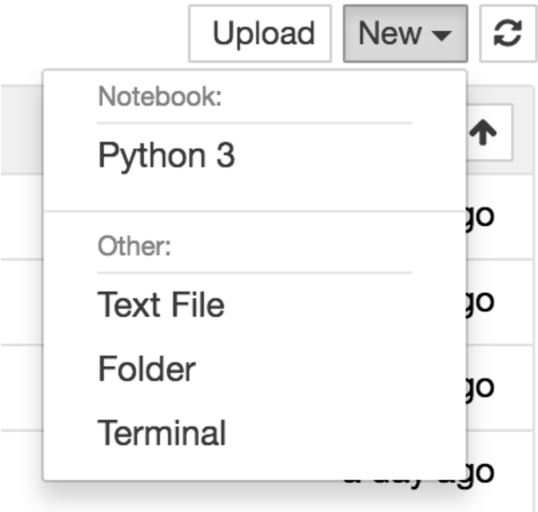
For more information on installing and using the packages you find on Anaconda Cloud, see *Package installation for Anaconda*.

Specific examples:

Displaying BioPython help text

After installing a package, you can confirm that the package was installed correctly by opening a Jupyter Notebook in that environment, importing the package and displaying its help text. In this example we will use the BioPython package.

1. *Start Navigator*.
2. To install the BioPython package, follow the instructions in *Managing packages*.
3. Click the icon to open a Jupyter Notebook.
4. In the Jupyter Notebook, click the New button and select your installed Python version.



5. Copy and paste the following code into the first cell:

```
import Bio
help(Bio)
```

6. To run the code, in the menu bar, click Cell then select Run Cells, or use the keyboard shortcut Ctrl-Enter.

   The BioPython help text is displayed.

Using the R programming language in Jupyter Notebook

R is a popular programing language for statistics.

To install and run R in a Jupyter Notebook:

1. *Start Navigator*.
2. To install the R language and r-essentials packages, follow the instructions in *Managing packages*.
3. *Open the environment with the R package* using the Open with Jupyter Notebook option.
4. To create a new notebook for the R language, in the Jupyter Notebook menu, select New, then select R.
5. Copy and paste the following code into the first cell:

```
library(dplyr)
iris
```

6. To run the code, in the menu bar, click Cell then select Run Cells, or use the keyboard shortcut Ctrl-Enter.
7. The iris data table is displayed.
8. To plot the data, click + to open a second cell, then copy and paste the following code into the second cell:

```
library(ggplot2)
ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) + geom_
↪point(size=3)
```

9. To run the code, in the menu bar, click Cell then select Run Cells, or use the keyboard shortcut Ctrl-Enter.

For more resources on using R with Anaconda, see *Using R language with Anaconda*.

TIP: For more information and to see the example code, see the Anaconda Developer Blog.

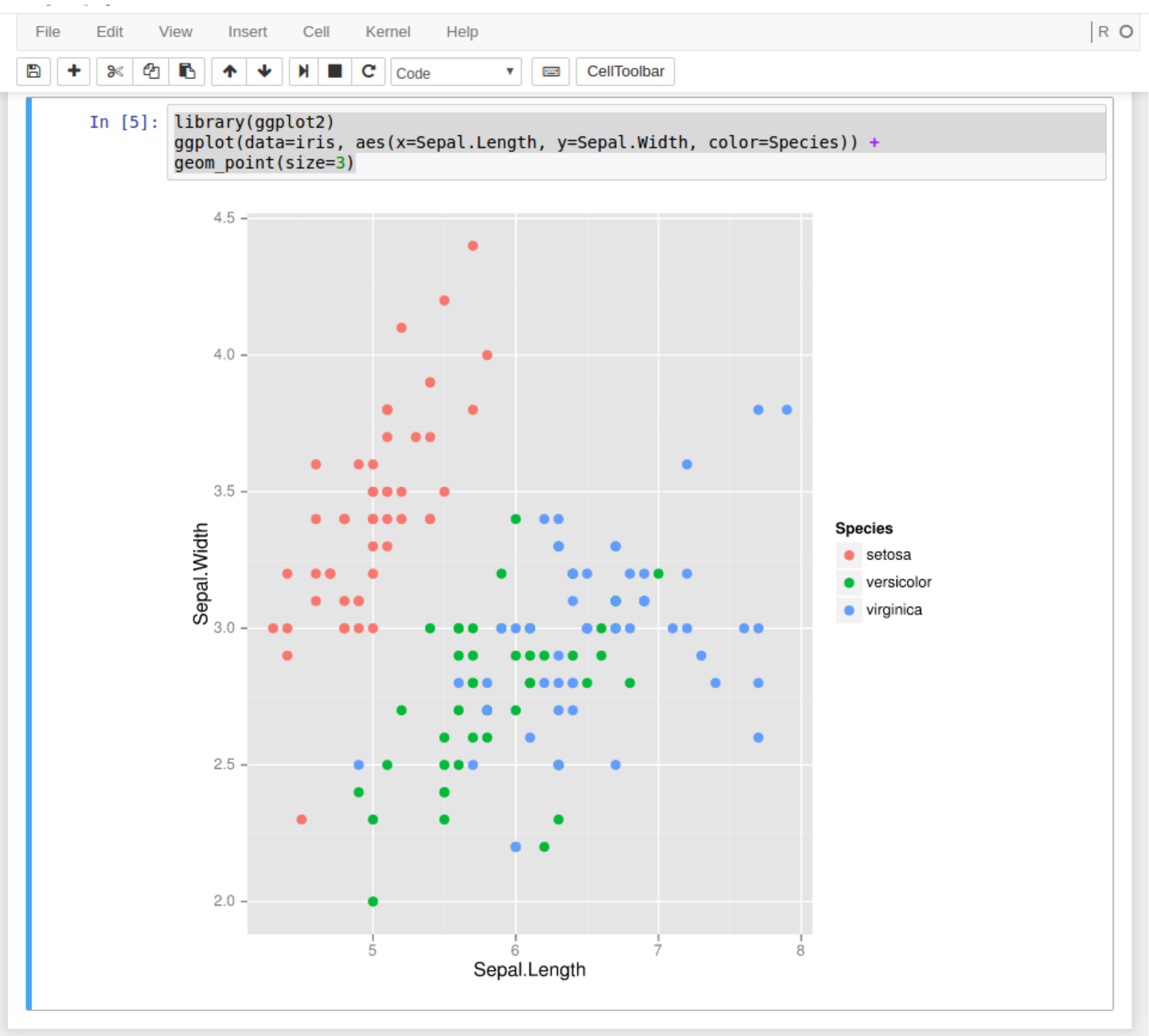
Installing and running Pandas

Pandas is a common Python tool for data manipulation and analysis. This task explains how to use Navigator to set up and begin working with Pandas in your choice of Terminal, Python, IPython or Jupyter Notebook. The steps are similar for *installing and opening* nearly any package.

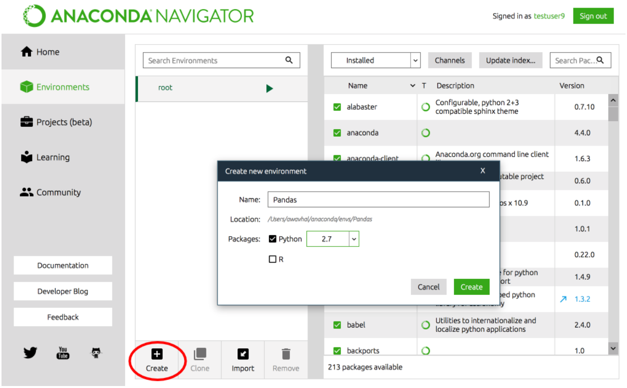
1. *Start Navigator*.
2. Click the **Environments** tab.
3. Click the Create button. When prompted, enter a descriptive name for the environment, such as “Pandas.”
4. Select a Python version to run in the environment.
5. Click OK.

   The new environment appears in the environments list.
6. Click the name of the new environment to activate it.

   The environment is highlighted with a green background.
7. In the list above the packages table, select All to filter the table to show all packages in all channels.



```
In [5]: library(ggplot2)
        ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
        geom_point(size=3)
```



Create new environment

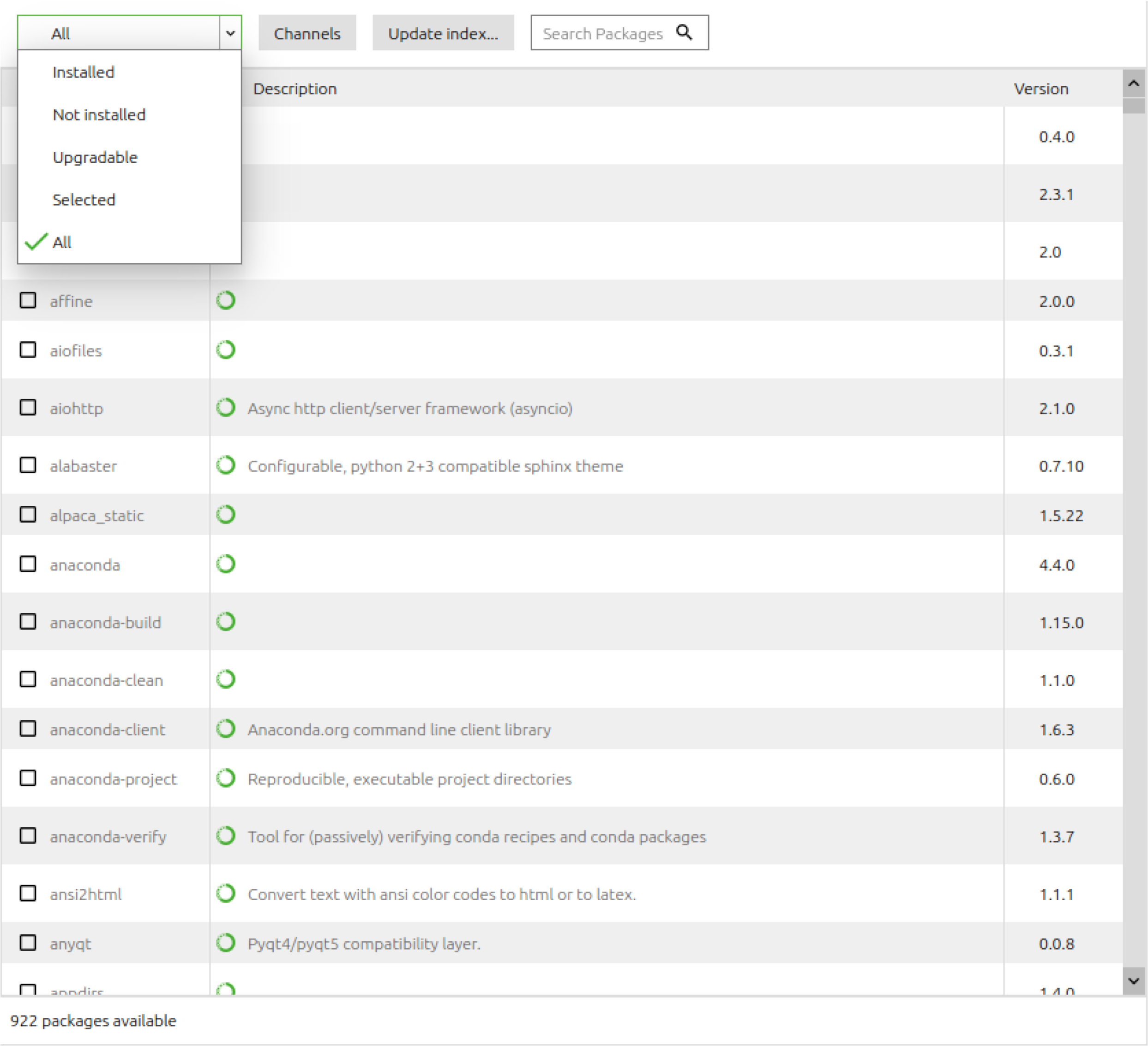
Name: Pandas

Location: /Users/awavhal/anaconda/envs/Pandas

Packages: ☑ Python 2.7

□ R

Cancel Create

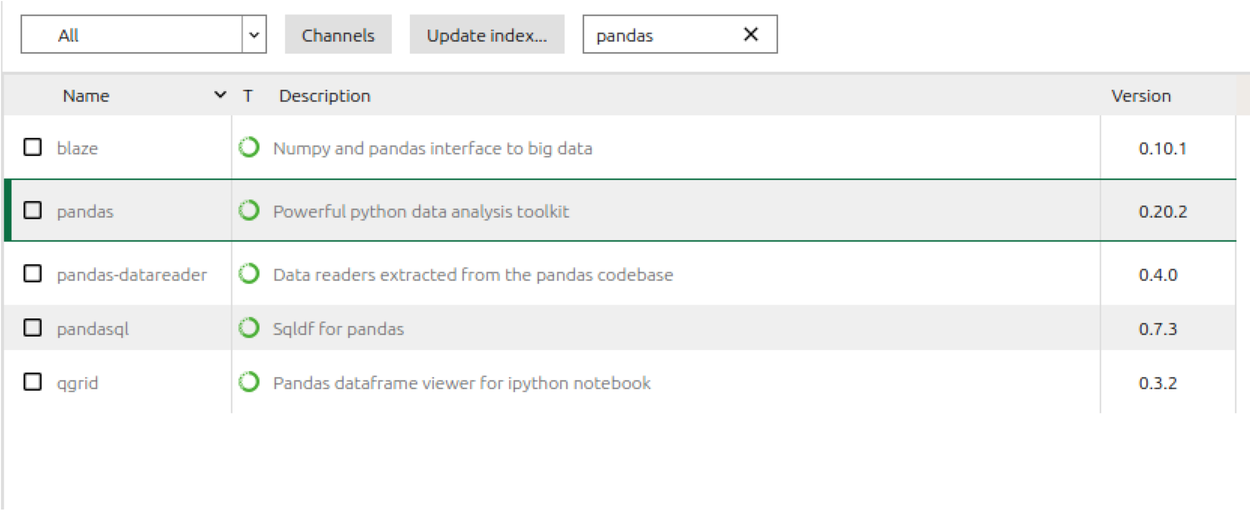


		Description	Version
			0.4.0
			2.3.1
			2.0
□	affine	○	2.0.0
□	aiofiles	○	0.3.1
□	aiohttp	○ Async http client/server framework (asyncio)	2.1.0
□	alabaster	○ Configurable, python 2+3 compatible sphinx theme	0.7.10
□	alpaca_static	○	1.5.22
□	anaconda	○	4.4.0
□	anaconda-build	○	1.15.0
□	anaconda-clean	○	1.1.0
□	anaconda-client	○ Anaconda.org command line client library	1.6.3
□	anaconda-project	○ Reproducible, executable project directories	0.6.0
□	anaconda-verify	○ Tool for (passively) verifying conda recipes and conda packages	1.3.7
□	ansi2html	○ Convert text with ansi color codes to html or to latex.	1.1.1
□	anyqt	○ Pyqt4/pyqt5 compatibility layer.	0.0.8
□	appdirs	○	1.4.0

922 packages available

8. In the Search Packages box, type `Pandas`.

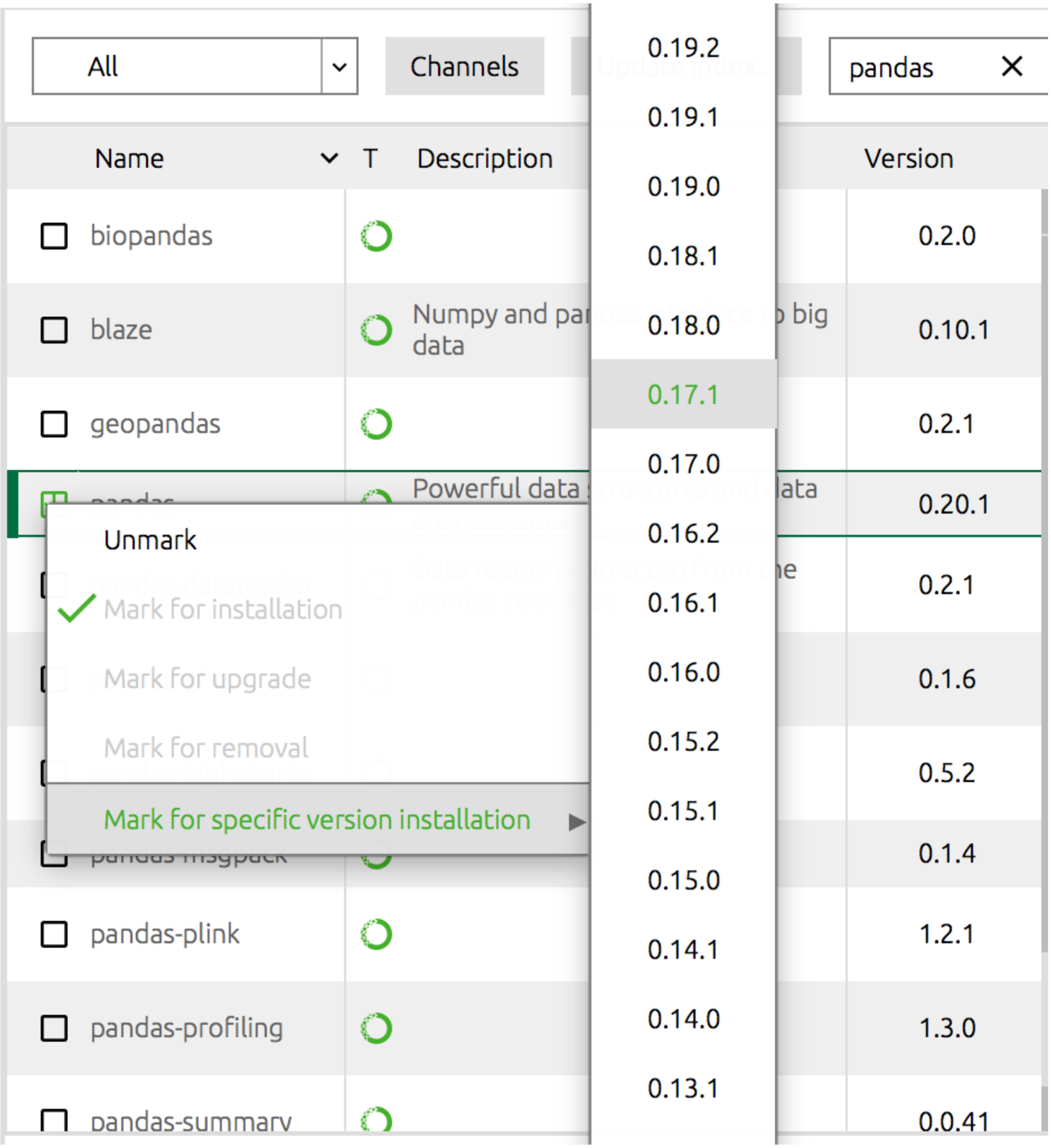
   Pandas appears as a package available for installation.



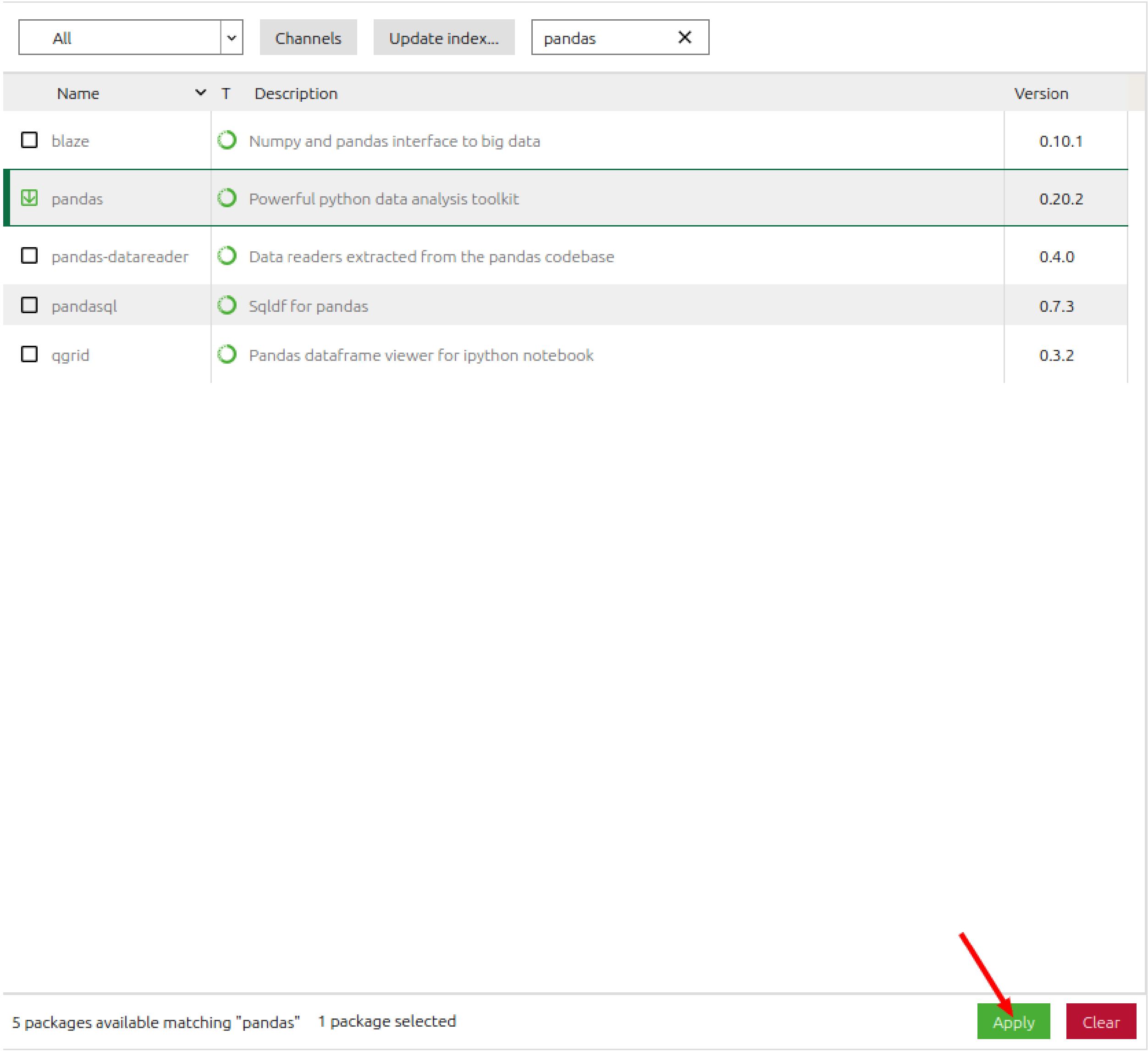
Name	T	Description	Version
☐ blaze	○	Numpy and pandas interface to big data	0.10.1
☐ pandas	○	Powerful python data analysis toolkit	0.20.2
☐ pandas-datareader	○	Data readers extracted from the pandas codebase	0.4.0
☐ pandasql	○	Sqldf for pandas	0.7.3
☐ qgrid	○	Pandas dataframe viewer for ipython notebook	0.3.2

9. Select the checkbox in front of the Pandas package name.
10. In the menu that appears, select Mark for specific version installation.
11. In the list that appears, select the Pandas version you want to install.
12. Click the Apply button.

    A progress bar appears below the Packages pane while Pandas and its dependencies are installed.
13. To begin using your new environment, click the **Environments** tab.
14. Click the arrow button next to the Pandas environment name.
15. In the list that appears, select the tool to use to open Pandas: Terminal, Python, IPython or Jupyter Notebook.

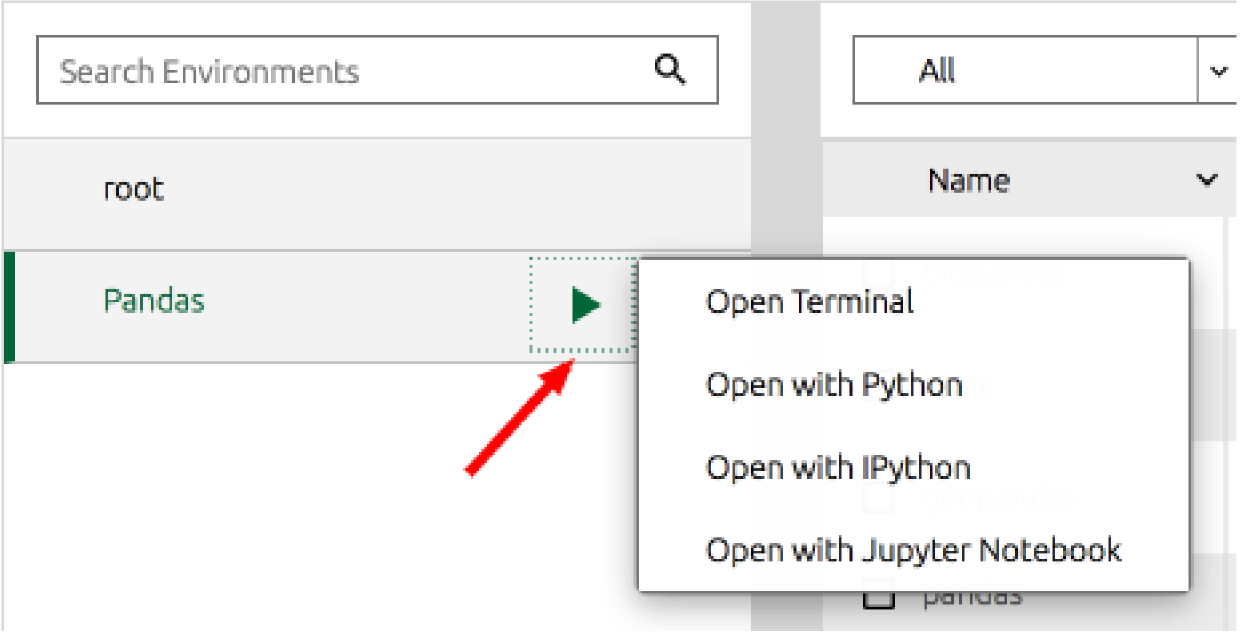


Name	T	Description	Version
□ biopandas	○		0.2.0
□ blaze	○	Numpy and pandas interface to big data	0.10.1
□ geopandas	○		0.2.1
☑ pandas	○	Powerful data structures and data analysis tools	0.20.1
□ pandas-datareader	○	Data readers extracted from the pandas codebase	0.2.1
□	○		0.1.6
□	○		0.5.2
□ pandas-msgpack	○		0.1.4
□ pandas-plink	○		1.2.1
□ pandas-profiling	○		1.3.0
□ pandas-summary	○		0.0.41



	Name	T	Description	Version
□	blaze	○	Numpy and pandas interface to big data	0.10.1
☑	pandas	○	Powerful python data analysis toolkit	0.20.2
□	pandas-datareader	○	Data readers extracted from the pandas codebase	0.4.0
□	pandasql	○	Sqldf for pandas	0.7.3
□	qgrid	○	Pandas dataframe viewer for ipython notebook	0.3.2

5 packages available matching "pandas"   1 package selected



Learn more about Pandas

- 10 Minutes to Pandas
- Pandas Cookbook

Troubleshooting

Navigator 1.3.2 fails to start correctly

Try editing the `.condarc` file in your home directory and removing custom channels from the channels list. Then restart Navigator.

RStudio fails to install in Navigator 1.6.2 if special characters are in install path

This is a known issue in Navigator 1.6.2 on Windows 7 64-bit.

Help and support

Anaconda Navigator is copyright 2016-2017 Anaconda®, Inc. It may be copied and distributed freely only as part of an Anaconda or Miniconda installation.

For free community support for Anaconda and Navigator, join the Anaconda Mailing List.

For Anaconda installation or technical support options, visit our support offerings page.

Training and consulting

Training and consulting is available for all Anaconda products, including Navigator. For more information, please contact your account representative or email sales@anaconda.com.

Reporting bugs

Please report bugs on the Anaconda GitHub issue tracker.

Release notes

1.8 – February 28, 2018

- **Remove "Projects" tab.**
- Support environment names with spaces on Windows.
- Improve switching between Anaconda API domains.
- Better integration with Visual Studio Code.
- Numerous bug fixes and performance enhancements.

1.7 – February 13, 2018

- Add R and MRO documentation tiles.
- Add custom spinner.
- Better handling of Access Denied errors.
- Better integration with Visual Studio Code.
- Numerous bug fixes and performance enhancements.

1.6 – May 19, 2017

- Better logging for application launches
- Package list can be filtered by "to be installed"
- Selecting an environment on the Home or Environments tab now selects it on both
- Numerous bug fixes and performance enhancements

1.5 – March 2, 2017

- New Projects tab for working with Anaconda Projects, still in beta status
- Added Navigator Updater and increased visibility of available updates
- Numerous bug fixes and performance enhancements

1.4 – January 31, 2017

- Navigator is out of beta status.

  The word "Beta" no longer appears in the title bar.

- R Studio available from the Navigator **Home** tab.

  R Studio is an open-source integrated development environment (IDE) for the R programming language. R Studio has been added to the R channel and can now be installed and launched from the Navigator **Home** tab. To install and launch R Studio, in the R Studio pane, click the Install button, then click the Launch button.

- Orange app returns to the **Home** tab.

  Orange app was removed from version 1.3 due to compatibility issues that have since been resolved. Orange app is pre-installed starting from version 1.4, so you can click the Launch button to launch it, with no need to install it first.

- Option to disable or enable SSL verification.

  From the top menu bar, select Preferences, select or clear the Enable SSL verification checkbox, then click the Apply button.

- Support for Anaconda Fusion licenses.

  Navigator now supports packages that require a license, including Anaconda Fusion. Installing Anaconda Fusion now automatically downloads a Fusion trial license from Anaconda Cloud. This license is generated once per user, so if a user uninstalls Fusion and reinstalls it, the user receives the original license. Trial licenses require users to register for a free account on Anaconda Cloud.

- Correct display of "no arch" packages.

  When Navigator installs "no arch" packages, they now appear correctly on the **Environments** tab.

- Friendly warning about updating channels for long-term users.

  If users have created `.condarc` files that try to access binstar.org–the original name of anaconda.org–Navigator now warns that binstar.org channels are outdated.

1.3 – September 27, 2016

- License Manager dialog.

  You can now easily *upload licenses* for applications that require them.

- Applications per environment.

  You can install applications to any environment. To do this, on the **Home** tab, click the **Applications** menu, then select the environment in which you want to install an application. If an environment is not selected, applications are installed by default in the root environment.

- One-click updates and installs.

  You can install a new package or update an installed package in the active environment. See *Managing packages*.

- Clearer indication of option to delete a search filter.

  On the **Environments** tab, when the packages are filtered by a user-defined search string, hovering over the X button now turns the cursor into an arrow and the X turns red, to emphasize that clicking the X will clear the search string.

- Channels now support tokens.

  The Channel manager now supports security tokens. See *Adding a channel*.

- Channel name copy and paste support.

- Navigator in a specific environment.

Installing Navigator in an environment other than root now makes that environment the default upon Navigator startup, both for the **Home** tab and the **Environments** tab.

- Table pre-filtering according to Python version.

  To avoid unnecessary conflicts, the table that displays a filtered view of packages–according to the Installed, Not Installed, Upgradeable and Downgradable filter values–does not display packages that are incompatible with the installed version of Python.

- Dialog size restore.

  Upon restart, the application window's last size and location are preserved, instead of the window's being maximized upon startup.

1.2 – August 1, 2016

- Improved condarc support.

  Navigator now reads from and writes to the conda configuration file (`.condarc`). In previous releases, updates made within Navigator to the conda configuration were not reflected in the `.condarc` file, and vice versa. Starting from version 1.2, when you add channels inside Navigator, the `.condarc` file is updated, and if `.condarc` is updated outside Navigator, Navigator correctly reads the available channels.

- Improved channels support.

  You can now *manage the channels* on Anaconda Cloud that you want Navigator to consider active. Active channels are included when you search for packages and other assets.

- R environment creation support.

  Navigator now allows you to *install a standalone R language environment or a mixed Python and R environment*.

- Conda environment file import.

  Navigator now allows you to *import environment.yml files*.

- Updates for Enterprise users.

  NOTE: You cannot use an Enterprise version of Anaconda Cloud with Navigator.

  - Simplified preferences.

    Preference options are simplified so that only one URL is needed to set up the API. Changing this value affects the URL setting of the anaconda-client configuration.

    NOTE: Changing the *URL setting in the CLI* affects the value displayed in the preferences.

  - Proxy.

    When using Navigator behind a proxy, the settings must be manually set inside the conda configuration file. See the conda proxy documentation.

Glossary

Anaconda

A downloadable, free, open source, high-performance, optimized Python and R distribution with 100+ packages and the option to easily install an additional 620+ popular open source packages for data science, including advanced and scientific analytics. Anaconda includes *Anaconda Navigator* and *conda* management utilities. Available for Windows, macOS and Linux. All versions are supported by the development community.

Anaconda Cloud

A web-based repository hosting service in the cloud. *Packages* created locally can be published to your *channel* on Anaconda Cloud to be shared with the public. Paid subscriptions to Anaconda Cloud can designate packages as private, to be shared only with authorized users. Anaconda Cloud is Anaconda's repository product made available to the public. Anaconda repository is also available for purchase by companies that wish to maintain their own on-site version of Anaconda Cloud. See Anaconda business subscriptions.

Anaconda Enterprise

Anaconda version that includes enterprise technical support, indemnification for a select number of open source packages, collaborative notebooks, high performance scalability, Hadoop, interactive visualization, governance and security. See Anaconda business subscriptions.

Anaconda Navigator

A desktop graphical user interface (GUI) included in all versions of *Anaconda* that allows you to easily manage *conda packages*, *environments*, *channels* and notebooks without the need to use the *command line interface (CLI)*.

Anaconda Repository

A private enterprise server on your network where open source and proprietary packages may be stored and retrieved for installation on a local computer. Anaconda Repository is different from Anaconda Cloud or the default conda repository. The Anaconda Repository is used to govern access to data science assets including packages and notebooks.

Channels

The locations of the repositories where conda looks for packages, often on Anaconda Cloud (Anaconda.org). Channels may also go to a private location on a remote or local repository that you or your organization manage.

Command line interface (CLI)

A program in which commands are entered as text, one line at a time, for a computer to execute. This is done in the Anaconda Prompt in Windows, and in a terminal in macOS and Linux. Conda is executed in a CLI. Contrast with *Graphical User Interface (GUI)*.

Conda

A package and environment manager program that is packaged with *Anaconda* and run in a CLI. Using conda, you can install and update *conda packages* and their dependencies, and switch between *conda environments* on your local computer. Contrast to *Anaconda Navigator*.

Conda environment

A folder or directory that contains a specific collection of *conda packages* and their dependencies, so they can be maintained and run independently without interfering with each other. Environments in Anaconda Navigator are conda environments.

EXAMPLE: It can be useful to create one environment for packages that run on Python 2, and another for packages that run on Python 3.

Changes to one environment do not affect other environments.

EXAMPLE: If you upgrade a program in one environment, this will not upgrade the same program in another environment.

Conda package

An archive file that contains everything that a software program needs in order to be installed and run, so you do not have to manually find and install each dependency separately. This can include system-level libraries, Python modules, executable programs and other components. Managed with *conda* or *Anaconda Navigator*. Packages in Anaconda Navigator are conda packages.

Conda repository

A cloud-based repository that contains 720+ open source certified packages that are easily installed locally via the `conda install` command. Can be viewed directly at https://repo.continuum.io/pkgs/ and used within *Anaconda Navigator* when downloading and installing packages from their Environments tab, or by using *conda* commands in a *Command Line Interface*.

Graphical user interface (GUI)

A program with graphic images, icons and menus in which commands are entered by clicking with a mouse and/or entering text in edit boxes. *Anaconda Navigator* is a graphical user interface that overlays the conda utility.

Miniconda

A minimal installer for conda that is run from a command line interface (CLI). Like Anaconda, Miniconda is a free software package that includes the conda package and environment manager, but Miniconda does not include Anaconda, Anaconda Navigator, or any packages other than those dependencies needed to install it. After Miniconda is installed, additional conda packages may be installed directly from the command line interface with the command `conda install`. See also *Anaconda* and *conda*.

Package manager

A collection of software tools that automates the process of installing, upgdating, configuring, and removing computer programs on a computer. Also known as a package management system. *Anaconda Navigator* includes the conda package manager with a GUI overlay for ease of use.

Packages

Software program files and information about the software, such as its name, the specific version and a description, bundled into a file that can be installed and managed by a package manager.

R packages

Conda packages that install and run the R computer language. Examples include R Essentials, a bundle of 80 popular open source software programs written in the R computer language. To learn more, see http://conda.pydata.org/docs/r-with-conda.html.

Repository

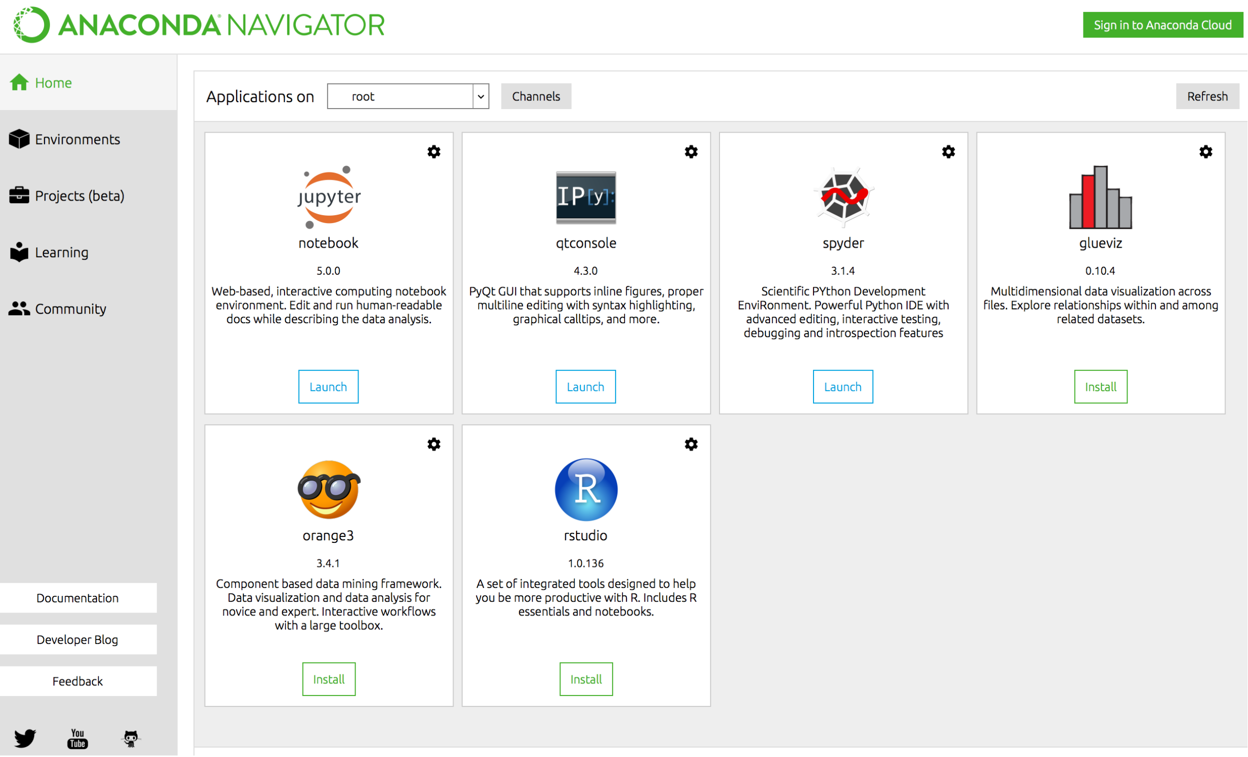
Any storage location from which software or software assets may be retrieved and installed on a local computer. See also: *Anaconda Repository* and *conda repository*.

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux.

To get Navigator, get the `Navigator Cheat Sheet` and *install Anaconda*.

The Navigator *Getting started with Navigator* section shows how to start Navigator from the shortcuts or from a terminal window.



notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

qtconsole
4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

spyder
3.1.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

glueviz
0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.

orange3
3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

rstudio
1.0.136
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

What applications can I access using Navigator?

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QTConsole
- Spyder
- VSCode
- Glueviz
- Orange 3 App
- Rodeo
- RStudio

Advanced conda users can also build your own Navigator applications.

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

Navigator User Guide

- *Overview*
- *Getting started with Navigator*
- *Tutorials*
- *Troubleshooting*

What's new in 1.8?

- Support environment names with spaces on Windows.
- Improve switching between Anaconda API domains.
- Better integration with Visual Studio Code.
- Numerous bug fixes and performance enhancements.

For details on changes in this and previous versions, see *Release notes*.

To see what version of Navigator you have, on Linux or Windows, from the **Help** on the Help menu, select About. On macOS, from the **Anaconda Navigator** menu, select About Anaconda-Navigator.

1.2.5 Frequently asked questions

How do I get Anaconda with Python 3.5?

There are three ways to get Anaconda with Python 3.5:

- We recommend that you download the latest version of Anaconda and then make a Python 3.5 environment.
- Or download the latest version of Anaconda and run this command to install Python 3.5 in the root environment: `conda install python=3.5`
- Or download the most recent Anaconda that included Python 3.5 by default, Anaconda 4.2.0. You can download it from our archive. Scroll down the page until you find version 4.2.0 for your platform.

How do I get previous versions of Anaconda or older versions of packages?

You can download previous versions of Anaconda from the Anaconda installer archive.

The installation procedures for previous versions are the same as the installation for the current version. For graphical installs, double click the downloaded installer file and follow the prompts. For command line installs, run "bash filename.sh" and follow the prompts. The installer filenames and hashes for previous versions will be different.

Older versions of packages can usually be downloaded from the package repository or from `https://anaconda.org/anaconda/PackageName`.

NOTE: Replace `PackageName` with the name of the desired package.

EXAMPLE: At https://anaconda.org/anaconda/beautifulsoup4 , previous versions of beautifulsoup4 are shown on the **Files** tab.

You can also search for packages from the command line with `conda search PackageName`.

NOTE: Replace `PackageName` with the name of the desired package.

I am behind a firewall. How can I download Anaconda?

If your corporate security settings do not allow you to download a Windows .exe executable file, download our zipped file.

How can I install Anaconda on an air gapped computer?

After you have the file, it's the same as any other install. Save a local copy of the appropriate Anaconda installer for the non-networked computer. You can copy the Anaconda installer using many different methods including a portable hard drive, USB drive or CD.

After copying the installer to the air gapped machine, follow the installation instructions for your operating system.

In what folder should I install Anaconda on Windows?

We recommend installing Anaconda or Miniconda into a directory that contains only 7-bit ASCII characters and no spaces, such as `C:\anaconda`. Do not install into paths that contain spaces such as `C:\Program Files` or that include `Unicode` characters outside the 7-bit ASCII character set. This helps ensure correct operation and no errors when using any open source tools in either Python 3 or Python 2 conda environments.

Should I add Anaconda to the Windows PATH?

When installing Anaconda, we recommend that you do not add Anaconda to the Windows PATH because this can interfere with other software. Instead, open Anaconda with the Start Menu and select Anaconda Prompt, or use Anaconda Navigator (Start Menu - Anaconda Navigator).

Should I add Anaconda to the macOS or Linux PATH?

When installing Anaconda, we recommend that you do add Anaconda to the PATH if you answer "No" during installation to the question "Do you wish the installer to prepend the Anaconda<2 or 3>install location to PATH in your `/home/<user>/.bashrc`?". Conda will not work until you add the PATH manually.

To add the PATH manually, open a text editor and open the file .bashrc or .bash_profile from your home directory. Add the line `export PATH="/<path to anaconda>/bin:$PATH"`.

NOTE: Replace <path-to-anaconda> with the actual path of your installed anaconda file.

Save the file. If you have any terminal windows open, close them all then open a new one. You may need to restart your computer for the PATH change to take effect.

What is the default path for installing Anaconda?

If you accept the default option to install Anaconda on the "default path" Anaconda is installed in your user home directory:

- Windows 10: `C:\Users\<your-username>\Anaconda3\`
- macOS: `/Users/<your-username>/anaconda3`
- Linux: `/home/<your-username>/anaconda3`

NOTE: If your username includes spaces, as is common on Windows systems, you should not accept the default path. See *In what folder should I install Anaconda on Windows?*.

How can I use Anaconda on older Macintosh systems?

- For macOS 10.5 and macOS 10.6, use the command line installers for Anaconda versions 1.8 and earlier. NOTE: These installer files end in `.sh`, not `.pkg`. Download from our archive.
- For macOS 10.7 and macOS 10.8, use the command line installers for Anaconda versions 4.2 and earlier. Download from our archive.
- For macOS 10.9, use the command line or graphical installers for Anaconda versions 5.1 and earlier. Download from our archive. Qt and other packages released after Anaconda Distribution 5.1's release date of February 15th, 2018 may not work on macOS 10.9, so it may be necessary not to update certain packages beyond this point.

How can I configure or opt out of the Intel Math Kernel Library (MKL)?

For information on configuring and uninstalling MKL, see the *Anaconda MKL documentation*.

How can I use TKinter?

Make sure the conda package `tk` is installed:

```
conda list tk
```

If it is not installed, run:

```
conda install tk
```

Python programs can use TKinter with `import Tkinter` on Python 2 or `import tkinter` on Python 3.

How can I use Cython on macOS?

Cython needs a C compiler, so you need to install Xcode.

How can I use Theano?

Theano requires gcc for acceleration.

To install Theano with acceleration:

- On Windows, run `conda install theano`.
- On Linux and macOS run `conda install gcc theano`.

NOTE: Theano is available from the default Anaconda channels for Windows, macOS and Linux with the command `conda install theano`.

How can I use GPUs with Anaconda?

See *Working with GPU packages*.

How is CPython compiled?

- Python 2.6 and 2.7 were compiled with Visual Studio 2008.
- Python 3.3 and 3.4 were compiled with VS 2010.
- Python 3.5 was compiled with VS 2015.

How do I cite Anaconda in an academic paper?

To cite Anaconda in an academic paper, use the recommended format. Example:

Anaconda Software Distribution. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. <https://anaconda.com>.

1.2.6 Help and support

The following help and support resources are available:

- Free community support is available from the Anaconda Support Group.
- Anaconda offers support and training.
- Anaconda also offers custom package builds through our consulting services.
- Package creators or maintainers may be able to help you with installing a package or building and publishing an updated version. Package creators may also contact Anaconda to submit their package for consideration to be included in Anaconda.
- Report bugs on the Anaconda GitHub issue tracker.

1.2.7 Release notes

Anaconda 5.2.0 (May 30, 2018)

User-facing changes

- Windows installers disallow the characters `! % ^ =` in installation path to prevent later usability issues
- Improved Windows multi-user installations by providing more dynamic shortcut working directory behavior

- Default channels point to repo.anaconda.com instead of repo.continuum.io

Backend improvements (non-visible changes)

- Security fixes for more than 20 packages based on a deep-dive of CVE vulnerabilities
- Windows installer uses a trimmed down value for `PATH` env var, to avoid DLL hell with existing software
- History file in the conda-meta directory is populated correctly to improve behavior of `--prune`
- Developer certificate for macOS pkg installers has been updated to Anaconda, Inc.

Changes for all x86 platforms

Updated:

- anaconda-client 1.6.9 -> 1.6.14
- astroid 1.6.1 -> 1.6.3
- astropy 2.0.3 -> 3.0.2
- attrs 17.4.0 -> 18.1.0
- backports.functools_lru_cache 1.4 -> 1.5
- bleach 2.1.2 -> 2.1.3
- bokeh 0.12.13 -> 0.12.16
- ca-certificates 2017.8.26 -> 2018.3.7
- certifi 2018.1.18 -> 2018.4.16
- cffi 1.11.4 -> 1.11.5
- cloudpickle 0.5.2 -> 0.5.3
- cryptography 2.1.4 -> 2.2.2
- curl 7.58.0 -> 7.60.0
- cython 0.27.3 -> 0.28.2
- cytoolz 0.9.0 -> 0.9.0.1
- dask 0.16.1 -> 0.17.5
- dask-core 0.16.1 -> 0.17.5
- decorator 4.2.1 -> 4.3.0
- distributed 1.20.2 -> 1.21.8
- filelock 2.0.13 -> 3.0.4
- flask 0.12.2 -> 1.0.2
- flask-cors 3.0.3 -> 3.0.4
- gevent 1.2.2 -> 1.3.0
- greenlet 0.4.12 -> 0.4.13
- hdf5 1.10.1 -> 1.10.2
- imageio 2.2.0 -> 2.3.0
- imagesize 0.7.1 -> 1.0.0

- ipaddress 1.0.19 -> 1.0.22
- ipykernel 4.8.0 -> 4.8.2
- ipython 6.2.1 -> 6.4.0
- ipywidgets 7.1.1 -> 7.2.1
- isort 4.2.15 -> 4.3.4
- jdcal 1.3 -> 1.4
- jedi 0.11.1 -> 0.12.0
- jupyter_client 5.2.2 -> 5.2.3
- jupyterlab 0.31.5 -> 0.32.1
- jupyterlab_launcher 0.10.2 -> 0.10.5
- libcurl 7.58.0 -> 7.60.0
- libxml2 2.9.7 -> 2.9.8
- llvmlite 0.21.0 -> 0.23.1
- lxml 4.1.1 -> 4.2.1
- matplotlib 2.1.2 -> 2.2.2
- mkl 2018.0.1 -> 2018.0.2
- msgpack-python 0.5.1 -> 0.5.6
- multipledispatch 0.4.9 -> 0.5.0
- nltk 3.2.5 -> 3.3.0
- notebook 5.4.0 -> 5.5.0
- numba 0.36.2 -> 0.38.0
- numexpr 2.6.4 -> 2.6.5
- numpy 1.14.0 -> 1.14.3
- numpydoc 0.7.0 -> 0.8.0
- openpyxl 2.4.10 -> 2.5.3
- openssl 1.0.2n -> 1.0.2o
- packaging 16.8 -> 17.1
- pandas 0.22.0 -> 0.23.0
- parso 0.1.1 -> 0.2.0
- path.py 10.5 -> 11.0.1
- pathlib2 2.3.0 -> 2.3.2
- pillow 5.0.0 -> 5.1.0
- pip 9.0.1 -> 10.0.1
- pkginfo 1.4.1 -> 1.4.2
- ply 3.10 -> 3.11
- psutil 5.4.3 -> 5.4.5

- py 1.5.2 -> 1.5.3
- pycodestyle 2.3.1 -> 2.4.0
- pylint 1.8.2 -> 1.8.4
- pyodbc 4.0.22 -> 4.0.23
- pyopenssl 17.5.0 -> 18.0.0
- pyqt 5.6.0 -> 5.9.2
- pysocks 1.6.7 -> 1.6.8
- pytables 3.4.2 -> 3.4.3
- pytest 3.3.2 -> 3.5.1
- python 3.6.4 -> 3.6.5
- python-dateutil 2.6.1 -> 2.7.3
- pytz 2017.3 -> 2018.4
- pyzmq 16.0.3 -> 17.0.0
- qt 5.6.2 -> 5.9.5
- qtpy 1.3.1 -> 1.4.1
- scandir 1.6 -> 1.7
- scipy 1.0.0 -> 1.1.0
- send2trash 1.4.2 -> 1.5.0
- setuptools 38.4.0 -> 39.1.0
- sip 4.18.1 -> 4.19.8
- sortedcollections 0.5.3 -> 0.6.1
- sortedcontainers 1.5.9 -> 1.5.10
- sphinx 1.6.6 -> 1.7.4
- spyder 3.2.6 -> 3.2.8
- sqlalchemy 1.2.1 -> 1.2.7
- sqlite 3.22.0 -> 3.23.1
- statsmodels 0.8.0 -> 0.9.0
- subprocess32 3.2.7 -> 3.5.0
- tornado 4.5.3 -> 5.0.2
- typing 3.6.2 -> 3.6.4
- wheel 0.30.0 -> 0.31.1
- widgetsnbextension 3.1.0 -> 3.2.1
- xlsxwriter 1.0.2 -> 1.0.4

Added:

- backcall 0.1.0
- blas 1.0

- blosc 1.14.3
- kiwisolver 1.0.1
- mkl_fft 1.0.1
- mkl_random 1.0.1
- more-itertools 4.1.0
- numpy-base 1.14.3
- pytest-arraydiff 0.2
- pytest-astropy 0.3.0
- pytest-doctestplus 0.1.3
- pytest-openfiles 0.3.0
- pytest-remotedata 0.2.1
- snappy 1.1.7

More changes specific to linux-64

Updated:

- dbus 1.12.2 -> 1.13.2
- fontconfig 2.12.4 -> 2.12.6
- glib 2.53.6 -> 2.56.1
- graphite2 1.3.10 -> 1.3.11
- gst-plugins-base 1.12.4 -> 1.14.0
- gstreamer 1.12.4 -> 1.14.0
- harfbuzz 1.7.4 -> 1.7.6
- libedit 3.1 -> 3.1.20170329
- libsodium 1.0.15 -> 1.0.16
- libxcb 1.12 -> 1.13
- ncurses 6.0 -> 6.1
- pcre 8.41 -> 8.42
- pexpect 4.3.1 -> 4.5.0
- unixodbc 2.3.4 -> 2.3.6
- xz 5.2.3 -> 5.2.4
- zeromq 4.2.2 -> 4.2.5

More changes specific to linux-32

Updated:

- dbus 1.12.2 -> 1.13.2

- fontconfig 2.12.4 -> 2.12.6
- glib 2.53.6 -> 2.56.1
- graphite2 1.3.10 -> 1.3.11
- gst-plugins-base 1.12.4 -> 1.14.0
- gstreamer 1.12.4 -> 1.14.0
- harfbuzz 1.7.4 -> 1.7.6
- libedit 3.1 -> 3.1.20170329
- libsodium 1.0.15 -> 1.0.16
- libxcb 1.12 -> 1.13
- ncurses 6.0 -> 6.1
- pcre 8.41 -> 8.42
- pexpect 4.3.1 -> 4.5.0
- unixodbc 2.3.4 -> 2.3.6
- xz 5.2.3 -> 5.2.4
- zeromq 4.2.2 -> 4.2.5

More changes specific to linux-ppc64le

Updated:

- anaconda-client 1.6.6 -> 1.6.14
- astroid 1.6.1 -> 1.6.2
- bokeh 0.12.13 -> 0.12.15
- cairo 1.14.10 -> 1.14.12
- cffi 1.11.2 -> 1.11.5
- conda 4.4.9 -> 4.5.4
- conda-build 3.4.1 -> 3.10.5
- cython 0.27.3 -> 0.28.1
- cytoolz 0.8.2 -> 0.9.0.1
- dask 0.16.0 -> 0.17.5
- dask-core 0.16.0 -> 0.17.5
- fontconfig 2.12.4 -> 2.12.6
- glib 2.53.6 -> 2.56.1
- ipaddress 1.0.18 -> 1.0.22
- ipywidgets 7.1.1 -> 7.2.0
- libsodium 1.0.15 -> 1.0.16
- libxcb 1.12 -> 1.13
- libxml2 2.9.4 -> 2.9.8

- libxslt 1.1.29 -> 1.1.32
- mistune 0.8.1 -> 0.8.3
- msgpack-python 0.4.8 -> 0.5.6
- ncurses 6.0 -> 6.1
- pcre 8.41 -> 8.42
- pytest 3.2.5 -> 3.5.1
- setuptools 36.5.0 -> 39.1.0
- sortedcontainers 1.5.7 -> 1.5.10
- sphinx 1.6.3 -> 1.7.4
- sqlite 3.21.0 -> 3.23.1
- tornado 4.5.2 -> 5.0.2
- unixodbc 2.3.4 -> 2.3.6
- xz 5.2.3 -> 5.2.4
- zeromq 4.2.2 -> 4.2.3

Added:

- attrs 18.1.0
- pluggy 0.6.0

Removed:

- gmpy2
- mpc
- mpfr

More changes specific to win-32

Updated:

- menuinst 1.4.11 -> 1.4.14
- pywin32 222 -> 223
- pywinpty 0.5 -> 0.5.1
- xlwings 0.11.5 -> 0.11.8

Added:

- libsodium 1.0.16
- m2w64-gcc-libgfortran 5.3.0
- m2w64-gcc-libs 5.3.0
- m2w64-gcc-libs-core 5.3.0
- m2w64-gmp 6.1.0
- m2w64-libwinpthread-git 5.0.0.4634.697f757
- msys2-conda-epoch 20160418

- zeromq 4.2.5

More changes specific to win-64

Updated:

- menuinst 1.4.11 -> 1.4.14
- pywin32 222 -> 223
- pywinpty 0.5 -> 0.5.1
- xlwings 0.11.5 -> 0.11.8

Added:

- libsodium 1.0.16
- m2w64-gcc-libgfortran 5.3.0
- m2w64-gcc-libs 5.3.0
- m2w64-gcc-libs-core 5.3.0
- m2w64-gmp 6.1.0
- m2w64-libwinpthread-git 5.0.0.4634.697f757
- msys2-conda-epoch 20160418
- zeromq 4.2.5

Anaconda 5.1.0 (Feb 15, 2018)

User-facing changes

- Microsoft Visual Studio Code added as an install option
- Anaconda Navigator has install and launch options for VS Code
- The installer support link has been replaced with the *Getting Started* page

Backend improvements (non-visible changes)

- Power packages are built with same recipes as rest of distribution
- Fixed some incomplete Windows installations due to interactions with antivirus software
- Fixed spaces in paths problems on Windows
- Anaconda Navigator was removed from the anaconda metapackage (but not the Anaconda installer)
- Installers present warnings when executed on the wrong platform

Changes for all x86 platforms

Updated:

- anaconda-client 1.6.5 -> 1.6.9
- anaconda-project 0.8.0 -> 0.8.2
- anaconda-navigator 1.6.9 -> 1.7.0

- asn1crypto 0.22.0 -> 0.24.0
- astroid 1.5.3 -> 1.6.1
- astropy 2.0.2 -> 2.0.3
- babel 2.5.0 -> 2.5.3
- bleach 2.0.0 -> 2.1.2
- bokeh 0.12.10 -> 0.12.13
- certifi 2017.7.27.1 -> 2018.1.18
- cffi 1.10.0 -> 1.11.4
- cloudpickle 0.4.0 -> 0.5.2
- conda 4.3.27 -> 4.4.10
- conda-build 3.0.27 -> 3.4.1
- cryptography 2.0.3 -> 2.1.4
- curl 7.55.1 -> 7.58.0
- cython 0.26.1 -> 0.27.3
- cytoolz 0.8.2 -> 0.9.0
- dask 0.15.3 -> 0.16.1
- dask-core 0.15.3 -> 0.16.1
- decorator 4.1.2 -> 4.2.1
- distributed 1.19.1 -> 1.20.2
- filelock 2.0.12 -> 2.0.13
- futures 3.1.1 -> 3.2.0
- glob2 0.5 -> 0.6
- h5py 2.7.0 -> 2.7.1
- html5lib 0.999999999 -> 1.0.1
- ipaddress 1.0.18 -> 1.0.19
- ipykernel 4.6.1 -> 4.8.0
- ipython 6.1.0 -> 6.2.1
- ipywidgets 7.0.0 -> 7.1.1
- jedi 0.10.2 -> 0.11.1
- jinja2 2.9.6 -> 2.10
- jupyter_client 5.1.0 -> 5.2.2
- jupyter_core 4.3.0 -> 4.4.0
- jupyterlab_launcher 0.4.0 -> 0.10.2
- libpng 1.6.32 -> 1.6.34
- libtiff 4.0.8 -> 4.0.9
- libxml2 2.9.4 -> 2.9.7

- libxslt 1.1.29 -> 1.1.32
- llvmlite 0.20.0 -> 0.21.0
- lxml 4.1.0 -> 4.1.1
- matplotlib 2.1.0 -> 2.1.2
- mistune 0.7.4 -> 0.8.3
- mkl 2018.0.0 -> 2018.0.1
- mpmath 0.19 -> 1.0.0
- msgpack-python 0.4.8 -> 0.5.1
- networkx 2.0 -> 2.1
- nltk 3.2.4 -> 3.2.5
- notebook 5.0.0 -> 5.4.0
- numba 0.35.0 -> 0.36.2
- numexpr 2.6.2 -> 2.6.4
- numpy 1.13.3 -> 1.14.0
- olefile 0.44 -> 0.45.1
- openpyxl 2.4.8 -> 2.4.10
- openssl 1.0.2l -> 1.0.2n
- pandas 0.20.3 -> 0.22.0
- path.py 10.3.1 -> 10.5
- patsy 0.4.1 -> 0.5.0
- pep8 1.7.0 -> 1.7.1
- pillow 4.2.1 -> 5.0.0
- psutil 5.4.0 -> 5.4.3
- py 1.4.34 -> 1.5.2
- pycosat 0.6.2 -> 0.6.3
- pycurl 7.43.0 -> 7.43.0.1
- pylint 1.7.4 -> 1.8.2
- pyodbc 4.0.17 -> 4.0.22
- pyopenssl 17.2.0 -> 17.5.0
- pytest 3.2.1 -> 3.3.2
- python 3.6.3 -> 3.6.4
- pytz 2017.2 -> 2017.3
- pyzmq 16.0.2 -> 16.0.3
- rope 0.10.5 -> 0.10.7
- ruamel_yaml 0.11.14 -> 0.15.35
- scikit-image 0.13.0 -> 0.13.1

- scipy 0.19.1 -> 1.0.0
- seaborn 0.8.0 -> 0.8.1
- setuptools 36.5.0 -> 38.4.0
- sortedcontainers 1.5.7 -> 1.5.9
- sphinx 1.6.3 -> 1.6.6
- spyder 3.2.4 -> 3.2.6
- sqlalchemy 1.1.13 -> 1.2.1
- sqlite 3.20.1 -> 3.22.0
- toolz 0.8.2 -> 0.9.0
- tornado 4.5.2 -> 4.5.3
- werkzeug 0.12.2 -> 0.14.1
- wheel 0.29.0 -> 0.30.0
- widgetsnbextension 3.0.2 -> 3.1.0

Added:

- attrs 17.4.0
- libcurl 7.58.0
- parso 0.1.1
- pluggy 0.6.0
- send2trash 1.4.2

More changes specific to win-64

Updated:

- comtypes 1.1.2 -> 1.1.4
- jupyterlab 0.27.0 -> 0.31.4
- menuinst 1.4.10 -> 1.4.11
- pywin32 221 -> 222
- xlwings 0.11.4 -> 0.11.5

Added:

- backports.shutil_which 3.5.1
- pywinpty 0.5
- terminado 0.8.1
- winpty 0.4.3

Removed:

- cachecontrol
- distlib
- lockfile

- progress

More changes specific to win-32

Updated:

- comtypes 1.1.2 -> 1.1.4
- jupyterlab 0.27.0 -> 0.31.5
- menuinst 1.4.10 -> 1.4.11
- pywin32 221 -> 222
- xlwings 0.11.4 -> 0.11.5

Added:

- backports.shutil_which 3.5.1
- pywinpty 0.5
- terminado 0.8.1
- winpty 0.4.3

Removed:

- cachecontrol
- distlib
- lockfile
- progress

More changes specific to osx-64

Updated:

- dbus 1.10.22 -> 1.12.2
- expat 2.2.4 -> 2.2.5
- jupyterlab 0.27.0 -> 0.31.5
- libsodium 1.0.13 -> 1.0.15
- pexpect 4.2.1 -> 4.3.1
- terminado 0.6 -> 0.8.1
- xlwings 0.11.4 -> 0.11.5

More changes specific to linux-64

Updated:

- cairo 1.14.10 -> 1.14.12
- dbus 1.10.22 -> 1.12.2
- expat 2.2.4 -> 2.2.5

- gst-plugins-base 1.12.2 -> 1.12.4
- gstreamer 1.12.2 -> 1.12.4
- harfbuzz 1.5.0 -> 1.7.4
- jupyterlab 0.27.0 -> 0.31.5
- libsodium 1.0.13 -> 1.0.15
- pango 1.40.11 -> 1.41.0
- pexpect 4.2.1 -> 4.3.1
- pycairo 1.13.3 -> 1.15.4
- terminado 0.6 -> 0.8.1

More changes specific to linux-32

Updated:

- cairo 1.14.10 -> 1.14.12
- dbus 1.10.22 -> 1.12.2
- expat 2.2.4 -> 2.2.5
- gst-plugins-base 1.12.2 -> 1.12.4
- gstreamer 1.12.2 -> 1.12.4
- harfbuzz 1.5.0 -> 1.7.4
- jupyterlab 0.27.0 -> 0.31.5
- libsodium 1.0.13 -> 1.0.15
- pango 1.40.11 -> 1.41.0
- pexpect 4.2.1 -> 4.3.1
- pycairo 1.13.3 -> 1.15.4
- terminado 0.6 -> 0.8.1

Changes for linux-ppc64le

Updated:

- anaconda-client 1.6.3 -> 1.6.6
- anaconda-project 0.6.0 -> 0.8.2
- asn1crypto 0.22.0 -> 0.23.0
- astroid 1.5.3 -> 1.6.1
- astropy 2.0.1 -> 2.0.3
- bleach 1.5.0 -> 2.1.2
- bokeh 0.12.7 -> 0.12.13
- cairo 1.14.8 -> 1.14.10

- certifi 2016.2.28 -> 2018.1.18
- cffi 1.10.0 -> 1.11.2
- cloudpickle 0.4.0 -> 0.5.2
- cryptography 1.8.1 -> 2.1.4
- curl 7.52.1 -> 7.55.1
- cython 0.26 -> 0.27.3
- dask 0.15.2 -> 0.16.0
- distributed 1.18.1 -> 1.20.2
- expat 2.1.0 -> 2.2.5
- filelock 2.0.7 -> 2.0.13
- fontconfig 2.12.1 -> 2.12.4
- freetype 2.5.5 -> 2.8
- futures 3.1.1 -> 3.2.0
- glob2 0.5 -> 0.6
- h5py 2.7.0 -> 2.7.1
- hdf5 1.8.17 -> 1.10.1
- html5lib 0.9999999 -> 1.0.1
- ipykernel 4.6.1 -> 4.8.0
- ipython 5.3.0 -> 5.4.1
- ipython 6.1.0 -> 6.2.1
- ipywidgets 6.0.0 -> 7.1.1
- jedi 0.10.2 -> 0.11.0
- jinja2 2.9.6 -> 2.10
- jupyter_client 5.1.0 -> 5.2.2
- jupyter_core 4.3.0 -> 4.4.0
- libpng 1.6.30 -> 1.6.32
- libsodium 1.0.10 -> 1.0.15
- libtiff 4.0.6 -> 4.0.9
- lxml 3.7.3 -> 4.1.1
- matplotlib 2.0.2 -> 2.1.2
- mistune 0.7.4 -> 0.8.1
- mpmath 0.19 -> 1.0.0
- nbconvert 5.2.1 -> 5.3.1
- networkx 1.11 -> 2.1
- nltk 3.2.4 -> 3.2.5
- notebook 5.0.0 -> 5.4.0

- numexpr 2.6.2 -> 2.6.4
- numpy 1.13.1 -> 1.13.3
- olefile 0.44 -> 0.45.1
- openblas 0.2.19 -> 0.2.20
- openpyxl 2.4.8 -> 2.4.10
- openssl 1.0.2l -> 1.0.2n
- pandas 0.20.3 -> 0.22.0
- path.py 10.3.1 -> 10.5
- pcre 8.39 -> 8.41
- pexpect 4.2.1 -> 4.3.0
- pillow 4.2.1 -> 5.0.0
- psutil 5.2.2 -> 5.4.1
- py 1.4.34 -> 1.5.2
- pycairo 1.10.0 -> 1.13.3
- pycosat 0.6.2 -> 0.6.3
- pylint 1.7.2 -> 1.8.2
- pyodbc 4.0.16 -> 4.0.22
- pyopenssl 17.0.0 -> 17.5.0
- pytest 3.2.1 -> 3.2.5
- python 2.7.13 -> 2.7.14
- python 3.6.2 -> 3.6.4
- pytz 2017.2 -> 2017.3
- pyzmq 16.0.2 -> 16.0.3
- requests 2.14.2 -> 2.18.4
- ruamel_yaml 0.11.14 -> 0.15.35
- scandir 1.5 -> 1.6
- scikit-image 0.13.0 -> 0.13.1
- scikit-learn 0.19.0 -> 0.19.1
- scipy 0.19.1 -> 1.0.0
- seaborn 0.8 -> 0.8.1
- setuptools 36.4.0 -> 36.5.0
- six 1.10.0 -> 1.11.0
- sqlalchemy 1.1.13 -> 1.2.1
- sqlite 3.13.0 -> 3.21.0
- terminado 0.6 -> 0.8.1
- tk 8.5.18 -> 8.6.7

- wheel 0.29.0 -> 0.30.0
- widgetsnbextension 3.0.2 -> 3.1.0
- xlsxwriter 0.9.8 -> 1.0.2
- yaml 0.1.6 -> 0.1.7
- zeromq 4.1.5 -> 4.2.2
- zict 0.1.2 -> 0.1.3

Added:

- backports.functools_lru_cache 1.4
- backports.shutil_get_terminal_size 1.0.0
- bzip2 1.0.6
- ca-certificates 2017.08.26
- conda 4.4.9
- conda-build 3.4.1
- conda-env 2.6.0
- conda-verify 3.0.0
- dask-core 0.16.0
- glib 2.53.6
- gmp 6.1.2
- gmpy2 2.0.8
- icu 58.2
- imageio 2.2.0
- libedit 3.1.20170329
- libgcc-ng 7.2.0
- libgfortran-ng 7.2.0
- libopenblas 0.2.20
- libssh2 1.8.0
- libstdcxx-ng 7.2.0
- libxcb 1.12
- lzo 2.10
- mccabe 0.6.1
- mpc 1.0.3
- mpfr 3.1.5
- ncurses 6.0
- openblas-devel 0.2.20
- pandoc 2.0.0.1
- parso 0.1.1

- pysocks 1.6.7
- readline 7.0
- send2trash 1.4.2
- typing 3.6.2
- urllib3 1.22
- webencodings 0.5.1

Removed:

- functools_lru_cache
- libgfortran
- libiconv

Anaconda 5.0.1 (Oct 25, 2017)

The changes detailed here are based on an upgrade from Anaconda 5.0.0.

- R has been updated to version 3.4.2. All R packages (including RStudio) have been rebuilt to be compatible with the new Anaconda 5.0 compilers.
- Updated many packages, including Python, Numpy, Spyder, Navigator, and Bokeh.
- The MKL library load path has been modified to address issue for Julia users.
- Fixed an OpenSSL issue with WSL on Windows.
- Fixed Anaconda Installer Configuration (AIC) feature for Unix installers.
- Re-enabled spaces in installation paths on Windows (temporarily disabled in 5.0.0).

Changes for all x86 platforms

Updated:

- anaconda-navigator 1.6.8 -> 1.6.9
- bokeh 0.12.7 -> 0.12.10
- conda 4.3.27 -> 4.3.30
- conda-build 3.0.22 -> 3.0.27
- dask 0.15.2 -> 0.15.3
- dask-core 0.15.2 -> 0.15.3
- distributed 1.18.3 -> 1.19.1
- lxml 3.8.0 -> 4.1.0
- matplotlib 2.0.2 -> 2.1.0
- networkx 1.11 -> 2.0
- numpy 1.13.1 -> 1.13.3
- psutil 5.2.2 -> 5.4.0
- pyflakes 1.5.0 -> 1.6.0

- pylint 1.7.2 -> 1.7.4
- python 2.7.13 -> 2.7.14
- python 3.6.2 -> 3.6.3
- scandir 1.5 -> 1.6
- scikit-learn 0.19.0 -> 0.19.1
- six 1.10.0 -> 1.11.0
- spyder 3.2.3 -> 3.2.4
- xlsxwriter 0.9.8 -> 1.0.2
- zict 0.1.2 -> 0.1.3

More changes specific to win-64

Updated:

- menuinst 1.4.8 -> 1.4.10

Added:

- lzo 2.10

More changes specific to win-32

Updated:

- menuinst 1.4.8 -> 1.4.10

Added:

- lzo 2.10

More changes specific to macOS-64

Added:

- bzip2 1.0.6
- lzo 2.10

More changes specific to linux-64

Added:

- bzip2 1.0.6

More changes specific to linux-32

Added:

- bzip2 1.0.6

Anaconda 5.0.0.1 (Oct 2, 2017)

- Fixes Python & C compiler fallback path for all cases on x86/x86_64 Linux. Without this fix, people were required to use our new compilers, which is not something we want to enforce at this time. This was affecting travis-ci builds and pip installs of packages that require compilation for extensions.

What's new in Anaconda 5.0?

Anaconda 5.0 was released on Sept 26, 2017.

- Over 100 packages updated and added. MKL is updated to 2018.0.0. JupyterLab alpha preview 0.27.0 is included.
- All new compilers on macOS and Linux, giving substantial security and performance improvements.
- Where possible, all build recipes use conda-forge as a base, using https://github.com/AnacondaRecipes .
- A new channel, `pkgs/main`, has been added to `defaults`. The new channel is given top priority within `defaults` and holds packages built with the new compiler stack.
- Continuum Analytics has been renamed to Anaconda, Inc. See this blog post for more.
- Spaces are no longer allowed in the installation path on Windows.
- Transitioned to more flexible dependency pinning of numpy packages, giving wider ranges of compatibility.

Changes for all x86 platforms

Updated:

- anaconda-client 1.6.3 -> 1.6.5
- anaconda-navigator 1.6.2 -> 1.6.8
- anaconda-project 0.6.0 -> 0.8.0
- astroid 1.4.9 -> 1.5.3
- astropy 1.3.2 -> 2.0.2
- babel 2.4.0 -> 2.5.0
- blaze 0.10.1 -> 0.11.3
- bleach 1.5.0 -> 2.0.0
- bokeh 0.12.5 -> 0.12.7
- boto 2.46.1 -> 2.48.0
- chardet 3.0.3 -> 3.0.4
- cloudpickle 0.2.2 -> 0.4.0
- conda 4.3.21 -> 4.3.27
- cryptography 1.8.1 -> 2.0.3
- curl 7.52.1 -> 7.55.1
- cython 0.25.2 -> 0.26.1
- dask 0.14.3 -> 0.15.2
- decorator 4.0.11 -> 4.1.2

- distributed 1.16.3 -> 1.18.3
- docutils 0.13.1 -> 0.14
- entrypoints 0.2.2 -> 0.2.3
- flask-cors 3.0.2 -> 3.0.3
- freetype 2.5.5 -> 2.8
- gevent 1.2.1 -> 1.2.2
- html5lib 0.999 -> 0.999999999
- idna 2.5 -> 2.6
- ipywidgets 6.0.0 -> 7.0.0
- isort 4.2.5 -> 4.2.15
- jupyter_client 5.0.1 -> 5.1.0
- jupyter_console 5.1.0 -> 5.2.0
- lazy-object-proxy 1.2.2 -> 1.3.1
- libpng 1.6.27 -> 1.6.32
- libtiff 4.0.6 -> 4.0.8
- llvmlite 0.18.0 -> 0.20.0
- lxml 3.7.3 -> 3.8.0
- markupsafe 0.23 -> 1.0
- mkl 2017.0.1 -> 2018.0.0
- nbconvert 5.1.1 -> 5.3.1
- nbformat 4.3.0 -> 4.4.0
- nltk 3.2.3 -> 3.2.4
- numba 0.33.0 -> 0.35.0
- numpy 1.12.1 -> 1.13.1
- numpydoc 0.6.0 -> 0.7.0
- odo 0.5.0 -> 0.5.1
- openpyxl 2.4.7 -> 2.4.8
- pandas 0.20.1 -> 0.20.3
- pandocfilters 1.4.1 -> 1.4.2
- pathlib2 2.2.1 -> 2.3.0
- pillow 4.1.1 -> 4.2.1
- prompt_toolkit 1.0.14 -> 1.0.15
- py 1.4.33 -> 1.4.34
- pycparser 2.17 -> 2.18
- pylint 1.6.4 -> 1.7.2
- pyodbc 4.0.16 -> 4.0.17

- pyopenssl 17.0.0 -> 17.2.0
- pyparsing 2.1.4 -> 2.2.0
- pytest 3.0.7 -> 3.2.1
- python-dateutil 2.6.0 -> 2.6.1
- qtconsole 4.3.0 -> 4.3.1
- qtpy 1.2.1 -> 1.3.1
- requests 2.14.2 -> 2.18.4
- rope 0.9.4 -> 0.10.5
- scikit-learn 0.18.1 -> 0.19.0
- scipy 0.19.0 -> 0.19.1
- seaborn 0.7.1 -> 0.8.0
- setuptools 27.2.0 -> 36.5.0
- sip 4.18 -> 4.18.1
- sphinx 1.5.6 -> 1.6.3
- spyder 3.1.4 -> 3.2.3
- sqlalchemy 1.1.9 -> 1.1.13
- sympy 1.0 -> 1.1.1
- testpath 0.3 -> 0.3.1
- tk 8.5.18 -> 8.6.7
- tornado 4.5.1 -> 4.5.2
- widgetsnbextension 2.0.0 -> 3.0.2
- wrapt 1.10.10 -> 1.10.11
- xlrd 1.0.0 -> 1.1.0
- xlsxwriter 0.9.6 -> 0.9.8
- zlib 1.2.8 -> 1.2.11

Added:

- backports.shutil_get_terminal_size 1.0.0
- bkcharts 0.2
- ca-certificates 2017.08.26
- certifi 2017.7.27.1
- conda-build 3.0.22
- dask-core 0.15.2
- filelock 2.0.12
- glob2 0.5
- imageio 2.2.0
- intel-openmp 2018.0.0

- jupyterlab 0.27.0
- jupyterlab_launcher 0.4.0
- libssh2 1.8.0
- mccabe 0.6.1
- pkginfo 1.4.1
- pycodestyle 2.3.1
- pysocks 1.6.7
- sphinxcontrib 1.0
- sphinxcontrib-websupport 1.0.1
- typing 3.6.2
- urllib3 1.22
- webencodings 0.5.1

Removed:

- _license

More changes specific to win-64

Updated:

- hdf5 1.8.15.1 -> 1.10.1
- icu 57.1 -> 58.2
- ipython 5.3.0 -> 5.4.1
- ipython 5.3.0 -> 6.1.0
- menuinst 1.4.7 -> 1.4.8
- pytables 3.2.2 -> 3.4.2
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- pywin32 220 -> 221
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- vs2008_runtime 9.00.30729.5054 -> 9.00.30729.1
- xlwings 0.10.4 -> 0.11.4
- xlwt 1.2.0 -> 1.3.0

Added:

- backports.functools_lru_cache 1.4
- cachecontrol 0.12.3
- distlib 0.2.5
- icc_rt 2017.0.4
- libiconv 1.15

- libxml2 2.9.4
- libxslt 1.1.29
- lockfile 0.12.2
- pandoc 1.19.2.1
- progress 1.3
- sqlite 3.20.1
- vc 14
- vc 9
- win_inet_pton 1.0.1
- wincertstore 0.2
- yaml 0.1.7

More changes specific to win-32

Updated:

- hdf5 1.8.15.1 -> 1.10.1
- icu 57.1 -> 58.2
- ipython 5.3.0 -> 5.4.1
- ipython 5.3.0 -> 6.1.0
- menuinst 1.4.7 -> 1.4.8
- pytables 3.2.2 -> 3.4.2
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- pywin32 220 -> 221
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- vs2008_runtime 9.00.30729.5054 -> 9.00.30729.1
- xlwings 0.10.4 -> 0.11.4
- xlwt 1.2.0 -> 1.3.0

Added:

- backports.functools_lru_cache 1.4
- cachecontrol 0.12.3
- distlib 0.2.5
- icc_rt 2017.0.4
- libiconv 1.15
- libxml2 2.9.4
- libxslt 1.1.29
- lockfile 0.12.2

- pandoc 1.19.2.1
- progress 1.3
- sqlite 3.20.1
- vc 14
- vc 9
- win_inet_pton 1.0.1
- wincertstore 0.2
- yaml 0.1.7

More changes specific to macOS-64

Updated:

- hdf5 1.8.17 -> 1.10.1
- icu 54.1 -> 58.2
- ipython 5.3.0 -> 5.4.1
- ipython 5.3.0 -> 6.1.0
- libiconv 1.14 -> 1.15
- ptyprocess 0.5.1 -> 0.5.2
- pytables 3.3.0 -> 3.4.2
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- python.app 1.2 -> 2
- readline 6.2 -> 7.0
- sqlite 3.13.0 -> 3.20.1
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- xlwings 0.10.4 -> 0.11.4
- xz 5.2.2 -> 5.2.3
- yaml 0.1.6 -> 0.1.7

Added:

- backports.functools_lru_cache 1.4
- dbus 1.10.22
- expat 2.2.4
- gettext 0.19.8.1
- glib 2.53.6
- gmp 6.1.2
- gmpy2 2.0.8
- libcxx 4.0.1

- libcxxabi 4.0.1
- libedit 3.1
- libffi 3.2.1
- libgfortran 3.0.1
- libsodium 1.0.13
- mpc 1.0.3
- mpfr 3.1.5
- ncurses 6.0
- pandoc 1.19.2.1
- pcre 8.41
- zeromq 4.2.2

More changes specific to linux-64

Updated:

- cairo 1.14.8 -> 1.14.10
- dbus 1.10.10 -> 1.10.22
- expat 2.1.0 -> 2.2.4
- fontconfig 2.12.1 -> 2.12.4
- glib 2.50.2 -> 2.53.6
- gst-plugins-base 1.8.0 -> 1.12.2
- gstreamer 1.8.0 -> 1.12.2
- harfbuzz 0.9.39 -> 1.5.0
- hdf5 1.8.17 -> 1.10.1
- icu 54.1 -> 58.2
- ipython 5.3.0 -> 5.4.1
- ipython 5.3.0 -> 6.1.0
- libsodium 1.0.10 -> 1.0.13
- libtool 2.4.2 -> 2.4.6
- pango 1.40.3 -> 1.40.11
- pcre 8.39 -> 8.41
- ptyprocess 0.5.1 -> 0.5.2
- pycairo 1.10.0 -> 1.13.3
- pytables 3.3.0 -> 3.4.2
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- readline 6.2 -> 7.0

- sqlite 3.13.0 -> 3.20.1
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- xlwt 1.2.0 -> 1.3.0
- xz 5.2.2 -> 5.2.3
- yaml 0.1.6 -> 0.1.7
- zeromq 4.1.5 -> 4.2.2

Added:

- backports.functools_lru_cache 1.4
- gmp 6.1.2
- gmpy2 2.0.8
- graphite2 1.3.10
- libedit 3.1
- libgcc-ng 7.2.0
- libgfortran-ng 7.2.0
- libstdcxx-ng 7.2.0
- lzo 2.10
- mpc 1.0.3
- mpfr 3.1.5
- ncurses 6.0
- pandoc 1.19.2.1
- patchelf 0.9

Removed:

- libgcc
- libgfortran
- libiconv

More changes specific to linux-32

Updated:

- cairo 1.14.8 -> 1.14.10
- dbus 1.10.10 -> 1.10.22
- expat 2.1.0 -> 2.2.4
- fontconfig 2.12.1 -> 2.12.4
- glib 2.50.2 -> 2.53.6
- gst-plugins-base 1.8.0 -> 1.12.2
- gstreamer 1.8.0 -> 1.12.2
- harfbuzz 0.9.39 -> 1.5.0

- hdf5 1.8.17 -> 1.10.1
- icu 54.1 -> 58.2
- ipython 5.3.0 -> 5.4.1
- ipython 5.3.0 -> 6.1.0
- libsodium 1.0.10 -> 1.0.13
- libtool 2.4.2 -> 2.4.6
- pango 1.40.3 -> 1.40.11
- pcre 8.39 -> 8.41
- ptyprocess 0.5.1 -> 0.5.2
- pycairo 1.10.0 -> 1.13.3
- pytables 3.3.0 -> 3.4.2
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- readline 6.2 -> 7.0
- sqlite 3.13.0 -> 3.20.1
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- xlwt 1.2.0 -> 1.3.0
- xz 5.2.2 -> 5.2.3
- yaml 0.1.6 -> 0.1.7
- zeromq 4.1.5 -> 4.2.2

Added:

- backports.functools_lru_cache 1.4
- gmp 6.1.2
- gmpy2 2.0.8
- graphite2 1.3.10
- libedit 3.1
- libgcc-ng 7.2.0
- libgfortran-ng 7.2.0
- libstdcxx-ng 7.2.0
- lzo 2.10
- mpc 1.0.3
- mpfr 3.1.5
- ncurses 6.0
- pandoc 1.15.0.6
- patchelf 0.9

Removed:

- libgcc
- libgfortran
- libiconv

Changes for linux-ppc64le

Updated:

- astroid 1.4.9 -> 1.5.3
- astropy 1.3.2 -> 2.0.1
- babel 2.4.0 -> 2.5.0
- bokeh 0.12.5 -> 0.12.7
- boto 2.46.1 -> 2.48.0
- chardet 3.0.3 -> 3.0.4
- cloudpickle 0.2.2 -> 0.4.0
- cython 0.25.2 -> 0.26
- dask 0.14.3 -> 0.15.2
- decorator 4.0.11 -> 4.1.2
- distributed 1.16.3 -> 1.18.1
- docutils 0.13.1 -> 0.14
- entrypoints 0.2.2 -> 0.2.3
- flask-cors 3.0.2 -> 3.0.3
- gevent 1.2.1 -> 1.2.2
- html5lib 0.999 -> 0.9999999
- idna 2.5 -> 2.6
- ipython 5.3.0 -> 6.1.0
- isort 4.2.5 -> 4.2.15
- jupyter_client 5.0.1 -> 5.1.0
- jupyter_console 5.1.0 -> 5.2.0
- lazy-object-proxy 1.2.2 -> 1.3.1
- libpng 1.6.27 -> 1.6.30
- markupsafe 0.23 -> 1.0
- nbconvert 5.1.1 -> 5.2.1
- nbformat 4.3.0 -> 4.4.0
- nltk 3.2.3 -> 3.2.4
- numpy 1.12.1 -> 1.13.1
- numpydoc 0.6.0 -> 0.7.0
- odo 0.5.0 -> 0.5.1

- openpyxl 2.4.7 -> 2.4.8
- pandas 0.20.1 -> 0.20.3
- pandocfilters 1.4.1 -> 1.4.2
- pathlib2 2.2.1 -> 2.3.0
- pillow 4.1.1 -> 4.2.1
- prompt_toolkit 1.0.14 -> 1.0.15
- ptyprocess 0.5.1 -> 0.5.2
- py 1.4.33 -> 1.4.34
- pycparser 2.17 -> 2.18
- pyflakes 1.5.0 -> 1.6.0
- pylint 1.6.4 -> 1.7.2
- pyparsing 2.1.4 -> 2.2.0
- pytables 3.2.2 -> 3.4.2
- pytest 3.0.7 -> 3.2.1
- python 3.5.3 -> 3.5.4
- python 3.6.1 -> 3.6.2
- python-dateutil 2.6.0 -> 2.6.1
- scikit-learn 0.18.1 -> 0.19.0
- scipy 0.19.0 -> 0.19.1
- seaborn 0.7.1 -> 0.8
- setuptools 27.2.0 -> 36.4.0
- sphinx 1.5.6 -> 1.6.3
- sqlalchemy 1.1.9 -> 1.1.13
- ssl_match_hostname 3.4.0.2 -> 3.5.0.1
- sympy 1.0 -> 1.1.1
- testpath 0.3 -> 0.3.1
- tornado 4.5.1 -> 4.5.2
- widgetsnbextension 2.0.0 -> 3.0.2
- wrapt 1.10.10 -> 1.10.11
- xlrd 1.0.0 -> 1.1.0
- xlsxwriter 0.9.6 -> 0.9.8
- xlwt 1.2.0 -> 1.3.0
- xz 5.2.2 -> 5.2.3
- zlib 1.2.8 -> 1.2.11

Added:

- bkcharts 0.2

- certifi 2016.2.28
- filelock 2.0.7
- functools_lru_cache 1.4
- glob2 0.5
- jedi 0.10.2
- patchelf 0.9
- pkginfo 1.4.1
- pycodestyle 2.3.1
- sphinxcontrib 1.0
- sphinxcontrib-websupport 1.0.1
- typing 3.6.2

What's new in Anaconda 4.4?

Anaconda 4.4 was released on May 31, 2017 and includes the following:

- Support added for the "ppc64le" machine type, for the POWER8 LE architecture used by IBM Power Systems and OpenPOWER servers.
- On Windows, the PATH environment variable is no longer changed by default, as this can cause trouble with other software. Instead, use Anaconda Navigator or the Anaconda Prompt in the Start Menu under "Anaconda" to use Anaconda software. If a user does choose to change the PATH variable, Anaconda is no longer appended to the PATH in system mode, and is now always added to the front of PATH in either system mode or user mode.
- Python 3.5 is updated from 3.5.2 to 3.5.3 and Python 3.6 from 3.6.0 to 3.6.1. Anaconda 4.4 supports Python 2.7, 3.5, and 3.6. Anaconda 4.3 was the last release to support Python 3.4.
- Minimum supported version of CentOS is now CentOS 6. Anaconda 4.3 was the last release to support CentOS 5.
- Applied pycrypto patch for CVE-2013-7439.
- Improved cp_acp support for install paths with non-ASCII characters on Windows.
- conda is updated from 4.3.14 to 4.3.21.
- Navigator is updated from 1.5.0 to 1.6.2.
- Project is updated from 0.4.1 to 0.6.0.
- Added distributed and pyodbc to the installers.
- Updated EULA.
- Conda packages with "mkl" in the package name now contain a file license.txt with a copy of the Intel Simplified Software License that applies to the Intel Math Kernel Library (MKL).
- Over 90 packages are updated or added.

2017-05-31 4.4.0:

Highlights:

- add support for the ppc64le (POWER8 LE used by IBM Power Systems and OpenPOWER servers) machine types

Other changes:

- On Windows, the PATH environment variable is no longer changed by default, as this can cause trouble with other software. The recommended approach is to instead use Anaconda Navigator or the Anaconda Prompt (located in the Start Menu under "Anaconda") when you wish to use Anaconda software. Also, Anaconda will always be added to the front of PATH, for either system or user mode. (Previously it was appended to the system path.)
- improve cp_acp support for install path on Windows
- updated 80 packages in the installer (and their dependencies)
- added distributed and pyodbc to the installers
- apply pycrypto patch for CVE-2013-7439
- end support for CentOS 5. CentOS 6 is now the minimum supported version.

Updates:

- alabaster from 0.7.9 to 0.7.10
- anaconda-client from 1.6.0 to 1.6.3
- anaconda-navigator from 1.5.0 to 1.6.2
- anaconda-project from 0.4.1 to 0.6.0
- astropy from 1.3 to 1.3.2
- babel from 2.3.4 to 2.4.0
- beautifulsoup4 from 4.5.3 to 4.6.0
- bokeh from 0.12.4 to 0.12.5
- boto from 2.45.0 to 2.46.1
- bottleneck from 1.2.0 to 1.2.1
- cffi from 1.9.1 to 1.10.0
- chardet from 2.3.0 to 3.0.3
- colorama from 0.3.7 to 0.3.9
- conda from 4.3.14 to 4.3.21
- contextlib2 from 0.5.4 to 0.5.5
- cryptography from 1.7.1 to 1.8.1
- dask from 0.13.0 to 0.14.3
- flask from 0.12 to 0.12.2
- futures from 3.0.5 to 3.1.1
- greenlet from 0.4.11 to 0.4.12
- h5py from 2.6.0 to 2.7.0
- hdf5 from 1.8.15.1 to 1.8.17
- idna from 2.2 to 2.5
- ipykernel from 4.5.2 to 4.6.1
- ipython from 5.1.0 to 5.3.0

- ipython_genutils from 0.1.0 to 0.2.0
- ipywidgets from 5.2.2 to 6.0.0
- jedi from 0.9.0 to 0.10.2
- jinja2 from 2.9.4 to 2.9.6
- jsonschema from 2.5.1 to 2.6.0
- jupyter_client from 4.4.0 to 5.0.1
- jupyter_console from 5.0.0 to 5.1.0
- jupyter_core from 4.2.1 to 4.3.0
- llvmlite from 0.15.0 to 0.18.0
- lxml from 3.7.2 to 3.7.3
- matplotlib from 2.0.0 to 2.0.2
- menuinst from 1.4.4 to 1.4.7
- mistune from 0.7.3 to 0.7.4
- nbconvert from 4.2.0 to 5.1.1
- nbformat from 4.2.0 to 4.3.0
- nltk from 3.2.2 to 3.2.3
- notebook from 4.3.1 to 5.0.0
- numba from 0.30.1 to 0.33.0
- numpy from 1.11.3 to 1.12.1
- numexpr from 2.6.1 to 2.6.2
- openpyxl from 2.4.1 to 2.4.7
- openssl from 1.0.2k to 1.0.2l
- pandas from 0.19.2 to 0.20.1
- partd from 0.3.7 to 0.3.8
- path.py from 10.0 to 10.3.1
- pathlib2 from 2.2.0 to 2.2.1
- pillow from 4.0.0 to 4.1.1
- ply from 3.9 to 3.10
- prompt_toolkit from 1.0.9 to 1.0.14
- psutil from 5.0.1 to 5.2.2
- py from 1.4.32 to 1.4.33
- pycosat from 0.6.1 to 0.6.2
- pygments from 2.1.3 to 2.2.0
- pyopenssl from 16.2.0 to 17.0.0
- pytables from 3.2.2 to 3.3.0
- pytest from 3.0.5 to 3.0.7

- python 3.5 from 3.5.2 to 3.5.3
- python 3.6 from 3.6.0 to 3.6.1
- pytz from 2016.10 to 2017.2
- qtawesome from 0.4.3 to 0.4.4
- qtconsole from 4.2.1 to 4.3.0
- requests from 2.12.4 to 2.14.2
- scandir from 1.4 to 1.5
- scikit-image from 0.12.3 to 0.13.0
- scipy from 0.18.1 to 0.19.0
- sphinx from 1.5.1 to 1.5.6
- spyder from 3.1.2 to 3.1.4
- sqlalchemy from 1.1.5 to 1.1.9
- statsmodels from 0.6.1 to 0.8.0
- tornado from 4.4.2 to 4.5.1
- traitlets from 4.3.1 to 4.3.2
- werkzeug from 0.11.15 to 0.12.2
- widgetsnbextension from 1.2.6 to 2.0.0
- wrapt from 1.10.8 to 1.10.10
- xlwings from 0.10.2 to 0.10.4

Added:

- asn1crypto 0.22.0
- bleach 1.5.0
- distributed 1.16.3
- html5lib 0.999
- msgpack-python 0.4.8
- navigator-updater 0.1.0
- olefile 0.44
- packaging 16.8
- pandocfilters 1.4.1
- pyodbc 4.0.16
- pywavelets 0.5.2
- sortedcollections 0.5.3
- sortedcontainers 1.5.7
- tblib 1.3.2
- testpath 0.3
- zict 0.1.2

Removed (from installers only):

- argcomplete
- chest
- configobj
- dill
- pyasn1
- redis
- redis-py
- sockjs-tornado

2017-03-10 4.3.1:

This patch release fixes problems with Anaconda Navigator not starting correctly on some versions of Mac OS X when using the GUI installers.

Fixes:

- removed creation of `~/.continuum` folder during install process on all platforms
- fixed '//' showing up in prefix when installing system wide on Mac OS using the GUI installer
- fixed OpenSSL not being installable into a path which contains spaces
- allow Unicode characters in install path on Windows (cp_acp fix)

Updates:

- anaconda-navigator from 1.4.3 to 1.5.0
- conda from 4.3.8 to 4.3.14

Added:

- anaconda-project 0.4.1

2017-02-03 4.3.0.1:

In this "mirco" patch release, we fixed a problem with the Windows installers which was causing problems with Qt applications when the install prefix exceeds 30 characters. No new Anaconda meta-packages correspond to this release (only new Windows installers).

2017-01-31 4.3.0:

Highlights:

- The Anaconda3 installers are based on Python 3.6. Anaconda 4.3 supports Python 2.7, 3.4, 3.5 and 3.6. Anaconda 4.3 will be the last release which supports Python 3.4. We will discontinue regular Python 3.4 package updates in the next release.
- The Intel Math Kernel Library (MKL) is updated from 11.3.3 to 2017.0.1.
- Over 90 packages are updated.
- seaborn is now installed by default.

Other changes:

- Updates jpeg and libpng to increase compatibility with conda-forge.
- Warns about possible errors if installing on Windows into an install path with spaces, and does not allow installation if the install path contains unicode characters.
- Fixes many Windows menu uninstallation issues and some other often reported uninstallation issues on Windows.
- Anaconda 4.2 is the last release that supports macOS 10.7 and macOS 10.8. Anaconda 4.3 supports macOS versions from 10.9 through the current version 10.12.
- conda-build, anaconda-clean and the Jupyter Notebook extensions are no longer installed by default but can be installed with a single conda command.

Updates:

- anaconda-client from 1.5.1 to 1.6.0
- anaconda-navigator from 1.3.1 to 1.4.3
- astroid from 1.4.7 to 1.4.9
- astropy from 1.2.1 to 1.3
- backports_abc from 0.4 to 0.5
- beautifulsoup4 from 4.5.1 to 4.5.3
- bokeh from 0.12.2 to 0.12.4
- boto from 2.42.0 to 2.45.0
- bottleneck from 1.1.0 to 1.2.0
- cairo from 1.12.18 to 1.14.8
- cffi from 1.7.0 to 1.9.1
- click from 6.6 to 6.7
- cloudpickle from 0.2.1 to 0.2.2
- conda from 4.2.9 to 4.3.8
- contextlib2 from 0.5.3 to 0.5.4
- cryptography from 1.5 to 1.7.1
- curl from 7.49.0 to 7.52.1
- cython from 0.24.1 to 0.25.2
- cytoolz from 0.8.0 to 0.8.2
- dask from 0.11.0 to 0.13.0
- datashape from 0.5.2 to 0.5.4
- decorator from 4.0.10 to 4.0.11
- docutils from 0.12 to 0.13.1
- flask from 0.11.1 to 0.12
- flask-cors from 2.1.2 to 3.0.2
- fontconfig from 2.11.1 to 2.12.1

- gevent from 1.1.2 to 1.2.1
- glib from 2.43.0 to 2.50.2
- greenlet from 0.4.10 to 0.4.11
- hdf5 from 1.8.15.1 to 1.8.17
- idna from 2.1 to 2.2
- ipaddress from 1.0.16 to 1.0.18
- ipykernel from 4.5.0 to 4.5.2
- jdcal from 1.2 to 1.3
- jinja2 from 2.8 to 2.9.4
- jpeg from 8d to 9b
- jupyter_core from 4.2.0 to 4.2.1
- lazy-object-proxy from 1.2.1 to 1.2.2
- libpng from 1.6.22 to 1.6.27
- libxml2 from 2.9.2 to 2.9.4
- libxslt from 1.1.28 to 1.1.29
- llvmlite from 0.13.0 to 0.15.0
- lxml from 3.6.4 to 3.7.2
- matplotlib from 1.5.3 to 2.0.0
- menuinst from 1.4.1 to 1.4.4
- mkl from 11.3.3 to 2017.0.1
- multipledispatch from 0.4.8 to 0.4.9
- nbformat from 4.1.0 to 4.2.0
- nltk from 3.2.1 to 3.2.2
- notebook from 4.2.3 to 4.3.1
- numba from 0.28.1 to 0.30.1
- numpy from 1.11.1 to 1.11.3
- openpyxl from 2.3.2 to 2.4.1
- openssl from 1.0.2j to 1.0.2k
- pandas from 0.18.1 to 0.19.2
- partd from 0.3.6 to 0.3.7
- path.py from 8.2.1 to 10.0
- pathlib2 from 2.1.0 to 2.2.0
- pexpect from 4.0.1 to 4.2.1
- pillow from 3.3.1 to 4.0.0
- pip from 8.1.2 to 9.0.1
- pixman from 0.32.6 to 0.34.0

- prompt_toolkit from 1.0.3 to 1.0.9
- psutil from 4.3.1 to 5.0.1
- py from 1.4.31 to 1.4.32
- pycparser from 2.14 to 2.17
- pyflakes from 1.3.0 to 1.5.0
- pylint from 1.5.4 to 1.6.4
- pyopenssl from 16.0.0 to 16.2.0
- pytables from 3.2.2 to 3.3.0
- pytest from 2.9.2 to 3.0.5
- python from 2.7.12 to 2.7.13
- python-dateutil from 2.5.3 to 2.6.0
- pytz from 2016.6.1 to 2016.10
- pyzmq from 15.4.0 to 16.0.2
- qt from 5.6.0 to 5.6.2
- qtawesome from 0.3.3 to 0.4.3
- qtpy from 1.1.2 to 1.2.1
- requests from 2.11.1 to 2.12.4
- scikit-learn from 0.17.1 to 0.18.1
- sphinx from 1.4.6 to 1.5.1
- spyder from 3.0.0 to 3.1.2
- sqlalchemy from 1.0.13 to 1.1.5
- toolz from 0.8.0 to 0.8.2
- tornado from 4.4.1 to 4.4.2
- traitlets from 4.3.0 to 4.3.1
- werkzeug from 0.11.11 to 0.11.15
- wrapt from 1.10.6 to 1.10.8
- xlsxwriter from 0.9.3 to 0.9.6
- xlwings from 0.10.0 to 0.10.2
- xlwt from 1.1.2 to 1.2.0
- zeromq from 4.1.4 to 4.1.5

Added:

- chardet 2.3.0
- isort 4.2.5
- libiconv 1.14
- numpydoc 0.6.0
- pcre 8.39 (on Linux)

- scandir 1.4
- seaborn 0.7.1
- subprocess32 3.2.7 (Python 2)

Removed (from installer only):

- anaconda-clean
- dynd-python
- filelock
- libdynd
- nb_anacondacloud
- nb_conda
- nb_conda_kernels
- nbpresent
- patchelf
- pkginfo

2016-09-28 4.2.0:

Highlights:

- updated Qt from major version 4 to 5
- updated IPython from 4.2 to 5.1
- added anaconda-clean, a tool for cleaning up Anaconda related configuration files and directories

Fixes:

- fixed Windows Outlook crash in silent install mode
- updated OpenSSL to 1.0.2j which contains important security fixes

Updates:

- alabaster from 0.7.8 to 0.7.9
- anaconda-client from 1.4.0 to 1.5.1
- anaconda-navigator from 1.2.1 to 1.3.1
- babel from 2.3.3 to 2.3.4
- beautifulsoup4 from 4.4.1 to 4.5.1
- bokeh from 0.12.0 to 0.12.2
- boto from 2.40.0 to 2.42.0
- cffi from 1.6.0 to 1.7.0
- conda from 4.1.4 to 4.2.9
- conda-build form 1.21.2 to 2.0.2
- configparser from 3.5.0b2 to 3.5.0
- cryptography from 1.4 to 1.5

- cython from 0.24 to 0.24.1
- dask from 0.10.0 to 0.11.0
- gevent from 1.1.1 to 1.1.2
- hdf5 from 1.8.15.1 to 1.8.17
- ipykernel from 4.3.1 to 4.5.0
- ipython from 4.2.0 to 5.1.0
- ipywidgets from 4.1.1 to 5.2.2
- jupyter_client from 4.3.0 to 4.4.0
- jupyter_console from 4.1.1 to 5.0.0
- jupyter_core from 4.1.0 to 4.2.0
- llvmlite from 0.11.0 to 0.13.0
- lxml from 3.6.0 to 3.6.4
- matplotlib from 1.5.1 to 1.5.3
- mistune from 0.7.2 to 0.7.3
- nb_anacondacloud from 1.1.0 to 1.2.0
- nb_conda from 1.1.0 to 2.0.0
- nb_conda_kernels from 1.0.3 to 2.0.0
- nbformat from 4.0.1 to 4.1.0
- notebook from 4.2.1 to 4.2.3
- numba from 0.26.0 to 0.28.1
- numexpr from 2.6.0 to 2.6.1
- openssl from 1.0.2h to 1.0.2j
- partd from 0.3.4 to 0.3.6
- pickleshare from 0.7.2 to 0.7.4
- pillow from 3.2.0 to 3.3.1
- ply from 3.8 to 3.9
- psutil from 4.3.0 to 4.3.1
- pyflakes from 1.2.3 to 1.3.0
- pyopenssl from 0.16.0 to 16.0.0
- pyqt from 4.11.4 to 5.6.0
- pytables from 3.2.2 to 3.2.3.1
- pytz from 2016.4 to 2016.6.1
- pyyaml from 3.11 to 3.12
- pyzmq from 15.2.0 to 15.4.0
- qt from 4.8.7 to 5.6.0
- qtpy from 1.0.2 to 1.1.2

- requests from 2.10.0 to 2.11.1
- ruamel_yaml from 0.11.7 to 0.11.14
- scipy from 0.17.1 to 0.18.1
- setuptools from 23.0.0 to 27.2.0
- sip from 4.16.9 to 4.18
- sphinx from 1.4.1 to 1.4.6
- spyder from 2.3.9 to 3.0.0
- tornado from 4.3 to 4.4.1
- traitlets from 4.2.1 to 4.3.0
- werkzeug from 0.11.10 to 0.11.11
- xlsxwriter from 0.9.2 to 0.9.3
- xlwings from 0.7.2 to 0.10.0

Added:

- anaconda-clean 1.0.0
- astroid 1.4.7
- dbus 1.10.10 (Linux)
- expat 2.1.0 (Linux)
- filelock 2.0.6
- glib 2.43.0 (Linux)
- gst-plugins-base 1.8.0 (Linux)
- gstreamer 1.8.0 (Linux)
- harfbuzz 0.9.39 (Linux)
- icu 57.1
- lazy-object-proxy 1.2.1
- libgcc 4.8.5 (Linux)
- libxcb 1.12 (Linux)
- pkginfo 1.3.2
- prompt_toolkit 1.0.3
- pylint 1.5.4
- qtawesome 0.3.3
- wcwidth 0.1.7
- widgetsnbextension 1.2.6
- win_unicode_console 0.5 (Windows)
- wrapt 1.10.6

Removed (from installer only):

- pyreadline

- sphinx_rtd_theme
- conda-env (now part of conda itself)

2016-07-08 4.1.1:

Fixes:

- Running the shell installer on some older system, would print out (harmless) tracebacks during the install process, see: https://github.com/ContinuumIO/anaconda-issues/issues/860
- We added blaze 0.10.1 back into the installer, which was accidentally missing in 4.1.0

Updates:

- bokeh from 0.11.1 to 0.12.0
- bottleneck from 1.0.0 to 1.1.0
- conda from 4.1.4 to 4.1.6
- conda-build from 1.21.2 to 1.21.3
- numpy from 1.11.0 to 1.11.1
- Python 2.7 from 2.7.11 to 2.7.12
- Python 3.4 from 3.4.4 to 3.4.5
- Python 3.5 from 3.5.1 to 3.5.2

2016-06-28 4.1.0:

Highlights:

- added Jupyter Notebook Extensions
- Windows installation: silent mode fixes & now compatible with SCCM (System Center Configuration Manager)
- updated MKL to 11.3.3, numpy to 1.11.0, as well as over 80 other updates, see below
- update Navigator from 1.1 to 1.2, in particular it no longer installs a desktop shortcut on macOS
- conda-recipes used to build the vast majority of the packages in the Anaconda installer have been published at: https://github.com/ContinuumIO/anaconda-recipes

Updates:

- alabaster from 0.7.7 to 0.7.8
- anaconda-navigator from 1.1.0 to 1.2.1
- astropy from 1.1.2 to 1.2.1
- babel from 2.2.0 to 2.3.3
- boto from 2.39.0 to 2.40.0
- cffi from 1.5.2 to 1.6.0
- cloudpickle from 0.1.1 to 0.2.1
- clyent from 1.2.1 to 1.2.2
- conda from 4.0.5 to 4.1.4

- conda-build from 1.20.0 to 1.21.2
- conda-env from 2.4.5 to 2.5.1
- cryptography from 1.3 to 1.4
- curl from 7.45.0 to 7.49.0
- cython from 0.23.4 to 0.24
- cytoolz from 0.7.5 to 0.8.0
- dask from 0.8.1 to 0.10.0
- datashape from 0.5.1 to 0.5.2
- decorator from 4.0.9 to 4.0.10
- dill from 0.2.4 to 0.2.5
- enum34 from 1.1.2 to 1.1.6
- flask from 0.10.1 to 0.11.1
- funcsigs from 0.4 to 1.0.2
- futures from 3.0.3 to 3.0.5
- gevent from 1.1.0 to 1.1.1
- greenlet from 0.4.9 to 0.4.10
- h5py from 2.5.0 to 2.6.0
- hdf5 from 1.8.15.1 to 1.8.16
- idna from 2.0 to 2.1
- ipaddress from 1.0.14 to 1.0.16
- ipython from 4.1.2 to 4.2.0
- jsonschema from 2.4.0 to 2.5.1
- jupyter_client from 4.2.2 to 4.3.0
- libffi from 3.0.13 to 3.2.1
- libgfortran from 3.0 to 3.0.0
- libpng from 1.6.17 to 1.6.22
- libsodium from 1.0.3 to 1.0.10
- llvmlite from 0.9.0 to 0.11.0
- menuinst from 1.3.2 to 1.4.1
- mkl from 11.3.1 to 11.3.3
- nbconvert from 4.1.0 to 4.2.0
- nltk from 3.2 to 3.2.1
- notebook from 4.1.0 to 4.2.1
- numba from 0.24.0 to 0.26.0
- numexpr from 2.5 to 2.6.0
- numpy from 1.10.4 to 1.11.0

- odo from 0.4.2 to 0.5.0
- openssl from 1.0.2g to 1.0.2h
- pandas from 0.18.0 to 0.18.1
- partd from 0.3.2 to 0.3.4
- patchelf from 0.8 to 0.9
- path.py from 8.1.2 to 8.2.1
- patsy from 0.4.0 to 0.4.1
- pickleshare from 0.5 to 0.7.2
- pillow from 3.1.1 to 3.2.0
- pip from 8.1.1 to 8.1.2
- psutil from 4.1.0 to 4.3.0
- ptyprocess from 0.5 to 0.5.1
- pycurl from 7.19.5.3 to 7.43.0
- pyflakes from 1.1.0 to 1.2.3
- pygments from 2.1.1 to 2.1.3
- pyopenssl from 0.15.1 to 0.16.0
- pyparsing from 2.0.3 to 2.1.4
- pytest from 2.8.5 to 2.9.2
- python-dateutil from 2.5.1 to 2.5.3
- pytz from 2016.2 to 2016.4
- qtconsole from 4.2.0 to 4.2.1
- qtpy from 1.0 to 1.0.2
- redis from 2.6.9 to 3.2.0
- redis-py from 2.10.3 to 2.10.5
- requests from 2.9.1 to 2.10.0
- scipy from 0.17.0 to 0.17.1
- setuptools from 20.3 to 23.0.0
- sockjs-tornado from 1.0.1 to 1.0.3
- sphinx from 1.3.5 to 1.4.1
- spyder from 2.3.8 to 2.3.9
- sqlalchemy from 1.0.12 to 1.0.13
- sqlite from 3.9.2 to 3.13.0
- terminado from 0.5 to 0.6
- toolz from 0.7.4 to 0.8.0
- werkzeug from 0.11.4 to 0.11.10
- xlrd from 0.9.4 to 1.0.0

- xlsxwriter from 0.8.4 to 0.9.2
- xlwings from 0.7.0 to 0.7.2
- xlwt from 1.0.0 to 1.1.2
- xz from 5.0.5 to 5.2.2
- zeromq from 4.1.3 to 4.1.4

Added:

- click 6.6
- configparser 3.5.0b2
- contextlib2 0.5.3
- entrypoints 0.2.2
- functools32 3.2.3.2
- get_terminal_size 1.0.0
- imagesize 0.7.1
- nb_anacondacloud 1.1.0
- nb_conda 1.1.0
- nb_conda_kernels 1.0.3
- nbpresent 3.0.2
- pathlib2 2.1.0
- ruamel_yaml 0.11.7

Removed:

- conda-manager
- qtawesome

2016-03-29 4.0.0:

The reason for jumping the Anaconda version from 2.5 to 4.0 is to avoid any possible confusion with the versions of Python included in Anaconda.

Highlights:

- this release of Anaconda includes the new Navigator, which is a graphical tool developed by Continuum Analytics to manage conda environments, applications and much more.

Enhancements:

- much improved package resolving in the new conda 4.0

Fixes:

- updated OpenSSL to 1.0.2g which contains important security fixes

Updates:

- anaconda-client from 1.2.2 to 1.4.0
- astropy from 1.1.1 to 1.1.2
- blaze from 0.9.0 to 0.9.1

- bokeh from 0.11.0 to 0.11.1
- cffi from 1.2.1 to 1.5.2
- clyent from 1.2.0 to 1.2.1
- colorama from 0.3.6 to 0.3.7
- conda from 3.19.1 to 4.0.5
- conda-build from 1.19.0 to 1.20.0
- cryptography from 1.0.2 to 1.3
- cycler from 0.9.0 to 0.10.0
- datashape from 0.5.0 to 0.5.1
- decorator from 4.0.6 to 4.0.9
- dynd-python from 0.7.1 to 0.7.2
- gevent from 1.0.2 to 1.1.0
- ipykernel from 4.2.2 to 4.3.1
- ipython from 4.0.3 to 4.1.2
- jupyter_client from 4.1.1 to 4.2.2
- jupyter_console from 4.1.0 to 4.1.1
- jupyter_core from 4.0.6 to 4.1.0
- libdynd from 0.7.1 to 0.7.2
- libgfortran from 1.0 to 3.0
- llvmlite from 0.8.0 to 0.9.0
- lxml from 3.5.0 to 3.6.0
- mistune from 0.7.1 to 0.7.2
- nltk from 3.1 to 3.2
- numba from 0.23.1 to 0.24.0
- numexpr from 2.4.6 to 2.5
- odo from 0.4.0 to 0.4.2
- openssl from 1.0.2f to 1.0.2g
- pandas from 0.17.1 to 0.18.0
- pexpect from 3.3 to 4.0.1
- pillow from 3.1.0 to 3.1.1
- pip from 8.0.2 to 8.1.1
- psutil from 3.4.2 to 4.1.0
- pyflakes from 1.0.0 to 1.1.0
- pygments from 2.1 to 2.1.1
- python-dateutil from 2.4.2 to 2.5.1
- pytz from 2015.7 to 2016.2

- pywin32 from 219 to 220
- qtconsole from 4.1.1 to 4.2.0
- scikit-image from 0.11.3 to 0.12.3
- scikit-learn from 0.17 to 0.17.1
- setuptools from 19.6.2 to 20.3
- sqlalchemy from 1.0.11 to 1.0.12
- sympy from 0.7.6.1 to 1.0
- traitlets from 4.1.0 to 4.2.1
- werkzeug from 0.11.3 to 0.11.4
- wheel from 0.26.0 to 0.29.0
- xlwings from 0.6.4 to 0.7.0

Added:

- anaconda-navigator 1.1.0
- chest 0.2.3
- cloudpickle 0.1.1
- conda-manager 0.3.1
- dask 0.8.1
- dill 0.2.4
- flask-cors 2.1.2
- heapdict 1.0.0
- locket 0.2.0
- mpmath 0.19
- partd 0.3.2
- qtawesome 0.3.2
- qtpy 1.0

Removed (from installer only, the packages are still available):

- abstract-rendering
- gevent-websocket
- launcher
- node-webkit

2016-02-05 2.5.0:

Highlights:

- add MKL (runtime, version 11.3.1) and make it the default backend for numpy, scipy, scikit-learn and numexpr on all platforms

Enhancements:

- added Windows debug information files, more precisely program database (.pdb files) files for Python by default
- added NoRegistry option to Windows installers, passing /NoRegistry=1 makes the installer not touch the registry

Fixes:

- in some cases start menu items were not created on Windows, due to a race condition, which we fixed in menuinst
- fixed the -f option of the Unix bash installers
- updated OpenSSL to 1.0.2f which contains important security fixes

Updates:

- alabaster from 0.7.6 to 0.7.7
- anaconda-client from 1.2.1 to 1.2.2
- astropy from 1.0.6 to 1.1.1
- babel from 2.1.1 to 2.2.0
- blaze-core from 0.8.3 to 0.9.0
- bokeh from 0.10.0 to 0.11.0
- boto from 2.38.0 to 2.39.0
- colorama from 0.3.3 to 0.3.6
- conda from 3.18.8 to 3.19.1
- conda-build from 1.18.2 to 1.19.0
- cytoolz from 0.7.4 to 0.7.5
- datashape from 0.4.7 to 0.5.0
- decorator from 4.0.4 to 4.0.6
- dynd-python from 0.7.0 to 0.7.1
- enum34 from 1.0.4 to 1.1.2
- gevent from 1.0.1 to 1.0.2
- gevent-websocket from 0.9.3 to 0.9.5
- ipykernel from 4.1.1 to 4.2.2
- ipython from 4.0.1 to 4.0.3
- ipywidgets from 4.1.0 to 4.1.1
- jdcal from 1.0 to 1.2
- jupyter_console from 4.0.3 to 4.1.0
- libdynd from 0.7.0 to 0.7.1
- lxml from 3.4.4 to 3.5.0
- matplotlib from 1.5.0 to 1.5.1
- menuinst from 1.3.1 to 1.3.2
- nbconvert from 4.0.0 to 4.1.0

- networkx from 1.10 to 1.11
- notebook from 4.0.6 to 4.1.0
- numba from 0.22.1 to 0.23.1
- numexpr from 2.4.4 to 2.4.6
- numpy from 1.10.1 to 1.10.4
- odo from 0.3.4 to 0.4.0
- openpyxl from 2.2.6 to 2.3.2
- openssl from 1.0.2d to 1.0.2f
- patchelf from 0.6 to 0.8
- pep8 from 1.6.2 to 1.7.0
- pillow from 3.0.0 to 3.1.0
- pip from 7.1.2 to 8.0.2
- psutil from 3.3.0 to 3.4.2
- py from 1.4.30 to 1.4.31
- pycurl from 7.19.5.1 to 7.19.5.3
- pygments from 2.0.2 to 2.1
- pytest from 2.8.1 to 2.8.5
- python 3.4 from 3.4.3 to 3.4.4
- pyzmq from 14.7.0 to 15.2.0
- requests from 2.8.1 to 2.9.1
- scipy from 0.16.0 to 0.17.0
- setuptools from 18.5 to 19.6.2
- snowballstemmer from 1.2.0 to 1.2.1
- sphinx from 1.3.1 to 1.3.5
- sphinx_rtd_theme from 0.1.7 to 0.1.9
- sqlalchemy from 1.0.9 to 1.0.11
- sqlite from 3.8.4.1 to 3.9.2
- traitlets from 4.0.0 to 4.1.0
- werkzeug from 0.11.2 to 0.11.3
- xlsxwriter from 0.7.7 to 0.8.4
- xlwings from 0.5.0 to 0.6.4

Added:

- et_xmlfile 1.0.1
- futures 3.0.3
- mkl 11.3.1
- mkl-service 1.1.1

Removed (from installer only, the packages are still maintained and available):

- openblas 0.2.14 (Linux)
- theano 0.7.0 (Linux)
- ujson 1.33

2015-12-08 2.4.1:

Fixes:

- added missing Windows process elevation when creating menu items
- added libdynd and dynd-python, which was missing in the last release
- fixed Cython on Mac OS X reporting missing libgcc_s.10.5.dylib
- fixed default channels being shown correctly in "conda list" after installing using Anaconda installer

Updates:

- anaconda-client from 1.1.0 to 1.2.1
- astropy from 1.0.5 to 1.0.6
- clyent from 0.4.0 to 1.2.0
- conda from 3.18.3 to 3.18.8
- conda-build 1.18.1 to 1.18.2
- conda-env from 2.4.4 to 2.4.5
- ipython from 4.0.0 to 4.0.1
- llvmlite from 0.7.0 to 0.8.0
- matplotlib from 1.4.3 to 1.5.0
- menuinst from 1.2.1 to 1.3.1
- numba from 0.21.0 to 0.22.1
- pandas from 0.17.0 to 0.17.1
- pixman from 0.26.2 to 0.32.6
- psutil from 3.2.2 to 3.3.0
- python 2.7 from 2.7.10 to 2.7.11
- python 3.5 from 3.5.0 to 3.5.1
- pytz from 2015.6 to 2015.7
- qtconsole from 4.1.0 to 4.1.1
- scikit-learn from 0.16.1 to 0.17
- setuptools from 18.4 to 18.5
- spyder from 2.3.7 to 2.3.8
- tornado from 4.2.1 to 4.3
- werkzeug from 0.10.4 to 0.11.2
- xlwings from 0.4.1 to 0.5.0

Added:

- backports_abc 0.4
- cycler 0.9.0
- libdynd and dynd-python 0.7.0
- jbig 2.1
- pycairo 1.10.0

2015-11-02 2.4.0:

Highlights:

- add Python 3.5 support
- updated NumPy to 1.10
- added OpenBLAS support on Linux
- made drastic speed improvements to conda
- moved from IPython to Jupyter
- improved Start Menus on Windows
- updated Qt to 4.8.7 on all platforms
- updates to more than 60 other packages

Known issues:

- numba and llvmlite are missing for Python 3.5 (because they don't support this Python version yet)
- numpy 1.10 has performance regression for record array access, see https://github.com/numpy/numpy/issues/6467
- Python 3.5 does not work on Windows XP

Updates:

- alabaster from 0.7.3 to 0.7.6
- argcomplete from 0.8.9 to 1.0.0
- astropy from 1.0.3 to 1.0.5
- babel from 1.3 to 2.1.1
- blaze-core from 0.8.0 to 0.8.3
- bokeh from 0.9.0 to 0.10.0
- cffi from 1.1.0 to 1.2.1
- clyent from 0.3.4 to 0.4.0
- cryptography from 0.9.1 to 1.0.2
- curl from 7.43.0 to 7.45.0
- conda from 3.14.1 to 3.18.3
- conda-build from 1.14.1 to 1.18.1
- conda-env from 2.2.3 to 2.4.4

- cython from 0.22.1 to 0.23.4
- cytoolz from 0.7.3 to 0.7.4
- datashape from 0.4.5 to 0.4.7
- decorator from 3.4.2 to 4.0.4
- freetype from 2.5.2 to 2.5.5
- greenlet from 0.4.7 to 0.4.9
- ipaddress from 1.0.7 to 1.0.14
- ipython from 3.2.0 to 4.0.0
- ipython-notebook from 3.2.0 to 4.0.4
- ipython-qtconsole from 3.2.0 to 4.0.1
- jedi from 0.8.1 to 0.9.0
- jinja2 from 2.7.3 to 2.8
- libsodium from 0.4.5 to 1.0.3
- libtiff from 4.0.2 to 4.0.6
- llvmlite from 0.5.0 to 0.7.0
- menuinst from 1.0.4 to 1.2.1
- mistune from 0.5.1 to 0.7.1
- multipledispatch from 0.4.7 to 0.4.8
- networkx from 1.9.1 to 1.10
- nltk from 3.0.3 to 3.1
- numba from 0.19.1 to 0.21.0
- numexpr from 2.4.3 to 2.4.4
- numpy from 1.9.2 to 1.10.1
- odo from 0.3.2 to 0.3.4
- openpyxl from 1.8.5 to 2.2.6
- openssl from 1.0.1k to 1.0.2d
- pandas from 0.16.2 to 0.17.0
- patsy from 0.3.0 to 0.4.0
- pillow from 2.8.2 to 3.0.0
- pip from 7.0.3 to 7.1.2
- ply from 3.6 to 3.8
- psutil from 2.2.1 to 3.2.2
- ptyprocess from 0.4 to 0.5
- py from 1.4.27 to 1.4.30
- pyasn1 from 0.1.7 to 0.1.9
- pyflakes from 0.9.2 to 1.0.0

- pyqt from 4.11.3 to 4.11.4
- pytables from 3.2.0 to 3.2.2
- pytest from 2.7.1 to 2.8.1
- pytz from 2015.4 to 2015.6
- qt from 4.8.6 to 4.8.7
- requests from 2.7.0 to 2.8.1
- scipy from 0.15.1 to 0.16.0
- setuptools from 17.1.1 to 18.4
- sip from 4.16.5 to 4.16.9
- six from 1.9.0 to 1.10.0
- spyder from 2.3.5.2 to 2.3.7
- spyder-app from 2.3.5.2 to 2.3.7
- sqlalchemy from 1.0.5 to 1.0.9
- sympy from 0.7.6 to 0.7.6.1
- toolz from 0.7.2 to 0.7.4
- tornado from 4.2 to 4.2.1
- unicodecsv from 0.9.4 to 0.14.1
- xlrd from 0.9.3 to 0.9.4
- xlsxwriter from 0.7.3 to 0.7.7
- xlwings from 0.3.5 to 0.4.1
- zeromq from 4.0.5 to 4.1.3

Added:

- anaconda-client 1.1.0
- beautifulsoup4 4.4.1
- ipykernel 4.1.1
- ipython_genutils 0.1.0
- ipywidgets 4.1.0
- jupyter 1.0.0
- jupyter_client 4.1.1
- jupyter_console 4.0.3
- jupyter_core 4.0.6
- libgfortran 1.0
- nbconvert 4.0.0
- nbformat 4.0.1
- notebook 4.0.6
- openblas 0.2.14

- patchelf 0.6
- path.py 8.1.2
- pexpect 3.3
- pickleshare 0.5
- qtconsole 4.1.0
- simplegeneric 0.8.1
- singledispatch 3.4.0.3
- traitlets 4.0.0
- wheel 0.26.0

Removed (from installer only, the packages are still maintained and available):

- bcolz
- blz
- certifi
- dynd-python
- libdynd
- mock
- runipy

2015-07-02 2.3.0:

Highlights:

- updates to 60 packages, including Python 2.7.10 and Pandas 0.16.2
- support for signed packages in conda

Fixes:

- fixed the extra space in IPython terminal being created when typing the tab key on Linux
- added missing zope.interface.common sub-package
- fixed Sphinx package being included in Spyder package

Enhancements:

- added support for signed packages in conda
- added curl on Windows, and kerberos authentication support
- added Windows support for libnetcdf and hdf5
- split gdal into libgdal and gdal (python-bindings)

Known issues:

- when opening some HDF5 files, pytables will crash on Windows and Python 3

Updates:

- argcomplete from 0.8.4 to 0.8.9
- astropy from 1.0.1 to 1.0.3

- bcolz from 0.8.1 to 0.9.0
- binstar from 0.10.1 to 0.11.0
- blaze-core from 0.7.3 to 0.8.0
- bokeh from 0.8.1 to 0.9.0
- boto from 2.36.0 to 2.38.0
- cffi from 0.9.2 to 1.1.0
- cryptography from 0.8 to 0.9.1
- conda from 3.10.0 to 3.14.1
- conda-build from 1.11.0 to 1.14.1
- conda-env from 2.1.3 to 2.2.3
- curl from 7.38.0 to 7.43.0
- cython from 0.22 to 0.22.1
- cytoolz from 0.7.2 to 0.7.3
- datashape from 0.4.4 to 0.4.5
- decorator from 3.4.0 to 3.4.2
- greenlet from 0.4.5 to 0.4.7
- h5py from 2.4.0 to 2.5.0
- hdf5 from 1.8.14 to 1.8.15.1
- ipython from 3.0.0 to 3.2.0
- libpng from 1.5.13 to 1.6.17
- libxml2 from 2.9.0 to 2.9.2
- llvmlite from 0.2.2 to 0.5.0
- lxml from 3.4.2 to 3.4.4
- nltk from 3.0.2 to 3.0.3
- nose from 1.3.4 to 1.3.7
- numba from 0.17.0 to 0.19.1
- numexpr from 2.3.1 to 2.4.3
- odo from 0.3.1 to 0.3.2
- pandas from 0.15.2 to 0.16.2
- pillow from 2.7.0 to 2.8.2
- pip from 6.0.8 to 7.0.3
- ply from 3.4 to 3.6
- py from 1.4.26 to 1.4.27
- pycparser from 2.10 to 2.14
- pyflakes from 0.8.1 to 0.9.2
- pyopenssl from 0.14 to 0.15.1

- pytables from 3.1.1 to 3.2.0
- pytest from 2.6.4 to 2.7.1
- python from 2.7.9 to 2.7.10
- python-dateutil from 2.4.1 to 2.4.2
- pytz from 2015.2 to 2015.4
- pyzmq from 14.5.0 to 14.7.0
- requests from 2.6.0 to 2.7.0
- scikit-image from 0.11.2 to 0.11.3
- scikit-learn from 0.15.2 to 0.16.1
- setuptools from 14.3 to 17.1.1
- sphinx from 1.2.3 to 1.3.1
- spyder from 2.3.4 to 2.3.5.2
- sqlalchemy from 0.9.9 to 1.0.5
- theano from 0.6.0 to 0.7.0
- toolz from 0.7.1 to 0.7.2
- tornado from 4.1 to 4.2
- werkzeug from 0.10.1 to 0.10.4
- xlsxwriter from 0.6.7 to 0.7.3
- xlwt from 0.7.5 to 1.0.0
- yaml from 0.1.4 to 0.1.6
- zeromq from 4.0.4 to 4.0.5

Added to Anaconda installers:

- alabaster 0.7.3
- babel 1.3
- bottleneck 1.0.0
- idna 2.0
- ipaddress 1.0.7
- snowballstemmer 1.2.0
- sphinx_rtd_theme 0.1.7

Removed (from installer only, the packages are still maintained and available):

- futures

Added (repository) support for:

- ansi2html
- azure
- blockspring
- boost

- btrees
- cloudpickle
- chest
- cligj
- csvkit
- dbf
- dill
- essbasepy
- flask-login
- heapdict
- holoviews
- ldap3
- line_profiler
- locket
- lockfile
- markdown
- markdown2
- meld3
- msgpack-python
- mysql-connector-python
- nano (Unix)
- param
- partd
- plac
- pyopengl
- pywget
- rasterio
- sas7bdat
- seaborn
- semantic_version
- snuggs
- spacy
- stripe
- supervisor (Unix)
- thinc
- unxutils (Windows)

- xray

2015-03-31 2.2.0:

Highlights:

- updates to 61 packages, including: Python, NumPyBokeh, pandas and blaze
- added 16 new packages
- added HTTPS support for default conda packages repo

Fixes:

- fixed cython command on Windows
- fixed untgz NSIS plugin to install files with exactly 100 characters

Enhancements:

- added https support for default conda packages repo
- renamed dateutil to python-dateutil to reflect the offical name
- added HDF5 and netcdf support for GDAL
- switched to using Pillow instead of PIL
- changed ipython-notebook to start directory to home directory on Windows

Updates:

- argcomplete from 0.8.1 to 0.8.4
- astropy from 0.4.2 to 1.0.1
- binstar from 0.7.1 to 0.10.1
- blaze from 0.6.3 to 0.7.3
- bokeh from 0.6.1 to 0.8.1
- boto from 2.32.1 to 2.36.0
- cairo from 1.12.2 to 1.12.18
- cffi from 0.8.6 to 0.9.2
- colorama from 0.3.1 to 0.3.3
- conda from 3.7.0 to 3.10.0
- conda-build from 1.8.2 to 1.11.0
- cryptography from 0.5.4 to 0.8
- cython from 0.21 to 0.22
- cytoolz from 0.7.0 to 0.7.2
- datashape from 0.3.0 to 0.4.4
- freetype from 2.4.10 to 2.5.2
- futures from 2.1.6 to 2.2.0
- greenlet from 0.4.4 to 0.4.5
- h5py from 2.3.1 to 2.4.0

- hdf5 from 1.8.13 to 1.8.14
- ipython from 2.2.0 to 3.0.0
- ipython-notebook from 2.2.0 to 3.0.0
- ipython-qtconsole from 2.2.0 to 3.0.0
- lxml from 3.4.0 to 3.4.2
- matplotlib from 1.4.0 to 1.4.3
- nltk from 3.0.0 to 3.0.2
- numba from 0.14.0 to 0.17.0
- numpy from 1.9.0 to 1.9.2
- openssl from 1.0.1h to 1.0.1k
- pandas from 0.14.1 to 0.15.2
- pep8 from 1.5.7 to 1.6.2
- pip from 1.5.6 to 6.0.8
- psutil from 2.1.1 to 2.2.1
- py from 1.4.25 to 1.4.26
- pycurl from 7.19.5 to 7.19.5.1
- pygments from 1.6 to 2.0.2
- pyparsing from 2.0.1 to 2.0.3
- pyqt from 4.10.4 to 4.11.3
- pytest from 2.6.3 to 2.6.4
- python from 2.7.8 to 2.7.9
- python-dateutil from 2.1 to 2.4.1
- pytz from 2014.7 to 2015.2
- pyzmq from 14.3.1 to 14.5.0
- qt from 4.8.5 to 4.8.6
- redis-py from 2.9.1 to 2.10.3
- requests from 2.4.1 to 2.6.0
- runipy from 0.1.1 to 0.1.3
- scikit-image from 0.10.1 to 0.11.2
- scipy from 0.14.0 to 0.15.1
- setuptools from 5.8 to 14.3
- sip from 4.15.5 to 4.16.5
- six from 1.8.0 to 1.9.0
- spyder from 2.3.1 to 2.3.4
- spyder-app from 2.3.1 to 2.3.4
- sqlalchemy from 0.9.7 to 0.9.9

- statsmodels from 0.5.0 to 0.6.1
- sympy from 0.7.5 to 0.7.6
- tk from 8.5.15 to 8.5.18
- toolz from 0.7.0 to 0.7.1
- tornado from 4.0.2 to 4.1
- werkzeug from 0.9.6 to 0.10.1
- xlsxwriter from 0.5.7 to 0.6.7
- zlib from 1.2.7 to 1.2.8

Added:

- bcolz 0.8.1
- certifi 14.05.14
- clyent 0.3.4
- enum34 1.0.4 (on Python 2.6, 2.7 and 3.3)
- fastcache 1.0.2
- fontconfig 2.11.1
- funcsigs 0.4
- jedi 0.8.1
- jsonschema 2.4.0
- llvmlite 0.2.2
- mistune 0.5.1
- odo 0.3.1
- pillow 2.7.0
- ptyprocess 0.4
- pyasn1 0.1.7
- terminado 0.5

Removed (from installer only, the packages are still maintained and available):

- atom
- casuarius
- chaco
- enable
- enaml
- future
- kiwisolver
- llvmpy
- mingw (on Windows)
- mpi4py

- pil (in favor of pillow)
- pyface
- traits
- traitsui

2014-09-30 2.1.0:

Fixes:

- fixed the ability to compile C extensions in Python 3 using MinGW
- added missing lzma module to Python 3.3 and 3.4 (links to xz)
- added missing werkzeug/debug/shared package data to Werkzeug package

Enhancements:

- added statsmodel support for Python 3.4
- added LZO support for pytables on Linux
- added scikit-learn support for Python 3.4
- added Windows cffi support (in repository)
- added bsddb conda package on Linux to support Berkeley DB

Updates:

- argcomplete from 0.6.7 to 0.8.1
- astropy from 0.3.2 to 0.4.2
- atom from 0.3.7 to 0.3.9
- beautiful-soup from 4.3.1 to 4.3.2
- binstar from 0.5.3 to 0.7.1
- blaze from 0.5.0 to 0.6.3
- bokeh from 0.4.4 to 0.6.1
- boto from 0.28.0 to 2.32.1
- conda from 3.5.2 to 3.7.0
- conda-build from 1.3.3 to 1.8.2
- configobj from 5.0.5 to 5.0.6
- colorama from 0.2.7 to 0.3.1
- curl from 7.30.0 to 7.38.0
- cython from 0.20.1 to 0.20.2
- datashape from 0.2.0 to 0.3.0
- docutils from 0.11 to 0.12
- dynd from 0.6.2 to 0.6.5
- enaml from 0.9.1 to 0.9.8
- future from 0.12.1 to 0.13.1

- greenlet from 0.4.2 to 0.4.4
- h5py from 2.3.0 to 2.3.1
- hdf5 from 1.8.9 to 1.8.13
- ipython from 2.1.0 to 2.2.0
- jinja2 from 2.7.2 to 2.7.3
- kiwisolver from 0.1.2 to 0.1.3
- launcher from 0.1.5 to 1.0.0
- libnetcdf from 4.2.1.1 to 4.3.2
- llvmpy from 0.12.6 to 0.12.7
- lxml from 3.3.5 to 3.4.0
- markupsafe from 0.18 to 0.23
- matplotlib from 1.3.1 to 1.4.0
- multipledispatch from 0.4.3 to 0.4.7
- networkx from 1.8.1 to 1.9.1
- nltk from 2.0.4 to 3.0.0
- nose from 1.3.3 to 1.3.4
- numba from 0.13.2 to 0.14.0
- numpy from 1.8.2 to 1.9.0
- pandas from 0.14.0 to 0.14.1
- patsy from 0.2.1 to 0.3.0
- pep8 from 1.5.6 to 1.5.7
- py from 1.4.20 to 1.4.25
- pycurl from 7.19.3.1 to 7.19.5
- pytest from 2.5.2 to 2.6.3
- python from 2.7.7 to 2.7.8
- pytz from 2014.3 to 2014.7
- pyzmq from 14.3.0 to 14.3.1
- requests from 2.3.0 to 2.4.1
- runipy from 0.1.0 to 0.1.1
- scikit-image from 0.10.0 to 0.10.1
- scikit-learn from 0.14.1 to 0.15.2
- setuptools from 3.6 to 5.8
- six 1.6.1 to 1.8.0
- sphinx from 1.2.2 to 1.2.3
- spyder from 2.3.0rc1 to 2.3.1
- sqlalchemy from 0.9.4 to 0.9.7

- tornado from 3.2.1 to 4.0.2
- xlsxwriter from 0.5.5 to 0.5.7

Added:

- abstract-rendering 0.5.1 (on Unix and Python 2)
- cffi 0.8.6
- cryptography 0.5.4
- cytoolz 0.7.0
- decorator 3.4.0
- futures-2.1.6 (for Python 2)
- pyopenssl 0.14
- sockjs-tornado 1.0.1
- toolz 0.7.0
- unicodecsv 0.9.4
- xz 5.0.5

2014-06-12 2.0.1:

Fixes:

- added missing libpython (the so-called MinGW import library) for Python 3.4
- in order to make pandas.io.excel work, we downgraded openpyxl from 2.0.2 to 1.8.5
- added missing idle script on Windows

Updates:

- conda from 3.5.2 to 3.5.5
- conda-build from 1.3.3 to 1.3.5
- numba from 0.13.1 to 0.13.2
- openssl from 1.0.1g to 1.0.1h
- pandas from 0.13.1 to 0.14.0
- python from 2.7.6 to 2.7.7
- scikit-image from 0.9.3 to 0.10.0
- werkzeug from 0.9.4 to 0.9.6

2014-05-28: 2.0.0:

Fixes:

- on Windows the /D option of the executable installer is now working
- added missing TkAgg matplotlib backend on Windows
- added missing osgeo._gdal_array extension to GDAL
- fixed rope on Python 3

Known issues:

- even though the matplotlib tkagg backend is now supported on all platforms and with all Python versions now, there might be problems on Windows with Python 3.4
- the Windows file association ('.py'-files being executed by the Anaconda Python interpreter) is not working

Enhancements:

- switched from using PySide to PyQt as the default Qt binding
- added lcms to PIL on Unix
- added ability to copy files upon install when filesystem fails to create hard links
- added netCDF4 support on 64-bit Windows
- on macOS, Tk in now linked to Cocoa (instead of X11)

Updated:

- astropy from 0.3.0 to 0.3.2
- binstar from 0.4.4 to 0.5.3
- blz from 0.6.1 to 0.6.2
- bokeh from 0.4.1 to 0.4.4
- boto from 2.25.0 to 2.28.0
- conda from 3.0.6 to 3.5.2
- conda-build from 1.2.0 to 1.3.3
- configobj from 4.7.2 to 5.0.5
- datashape from 0.1.1 to 0.2.0
- dynd-python gtom 0.6.1 to 0.6.2
- future from 0.11.2 to 0.12.1
- gevent from 1.0 to 1.0.1
- gevent-websocket from 0.9.2 to 0.9.3
- h5py from 2.2.1 to 2.3.0
- ipython from 1.1.0 to 2.1.0
- itsdangerous from 0.23 to 0.24
- launcher from 0.1.2 to 0.1.5
- llvmpy from 0.12.3 to 0.12.4
- lxml from 3.3.1 to 3.3.5
- nose from 1.3.0 to 1.3.3
- numba from 0.12.1 to 0.13.1
- numpy from 1.8.0 to 1.8.1
- openpyxl 1.8.2 to 2.0.2
- openssl from 1.0.1c to 1.0.1g
- pep8 gtom 1.4.6 to 1.5.6

- pip from 1.5.2 to 1.5.6
- psutil from 1.2.1 to 2.1.1
- pycosat from 0.6.0 to 0.6.1
- pycurl from 7.19.0 to 7.19.3.1
- pyflakes from 0.7.3 to 0.8.1
- pytables from 3.1.0 to 3.1.1
- pytz from 2013b to 2014.3
- pyyaml 3.10 to 3.11
- pyzmq from 2.2.0.1 to 14.3.0
- requests from 2.2.1 to 2.3.0
- scipy from 0.13.3 to 0.14.0
- setuptools from 2.2 to 3.6
- six from 1.5.2 to 1.6.1
- sphinx form 1.2.1 to 1.2.2
- spyder from 2.2.5 to 2.3.0rc1
- sqlalchemy from 0.9.2 to 0.9.4
- sqlite from 3.7.13 to 3.8.4.1
- sympy from 0.7.4.1 to 0.7.5
- tk from 8.5.13 to 8.5.15
- tornado from 3.2.0 to 3.2.1
- xlrd from 0.9.2 to 0.9.3
- xlsxwriter from 0.5.2 to 0.5.5
- zeromq from 2.2.0 to 4.0.4

Added:

- conda-launch 0.1
- jdcal 1.0
- multipledispatch 0.4.0
- python 3.4.1
- pyqt 4.10.4
- runipy 0.1.0
- sip 4.15.5
- xlwings 0.1.0 (Windows only)

Removed (from installer, packages are still supported and available in repo):

- apptools
- biopython
- envisage

- disco and erlang (from 64-bit Linux)
- gevent_zeromq
- keyring
- mayavi
- mdp
- netcdf4
- pykit
- pysal
- pysam
- vtk

2014-04-09: 1.9.2:

Fixes:

- updated to openssl 1.0.1g on Unix to fix the "Heartbleed bug" of the TLS Heartbeat Extension problem (reported in the news)
- fixed /D option in silent mode for Windows installer

Updates:

- openssl from 1.0.1c to 1.0.1g
- conda from 3.0.6 to 3.4.1
- conda-build from 1.2.0 to 1.3.1

2014-02-20: 1.9.1:

Fixes:

- openpyxl to depend on up-to-date version of lxml
- added missing MSVCP (both for 2008 and 2010) DLLs to Windows installers which fixes issues with the user install mode on some systems

Updates:

- atom from 0.3.6 to 0.3.7
- blaze from 0.4.1 to 0.4.2
- bokeh from 0.4 to 0.4.1
- boto 2.24.0 to 2.25.0
- conda from 3.0.3 to 3.0.6
- conda-build from 1.1.0 to 1.2.0
- cython from 0.20 to 0.20.1
- datashape from 0.1.0 to 0.1.1
- dynd-python 0.6.0 to 0.6.1

- enaml from 0.9.0 to 0.9.1
- llvmpy from 0.12.2 to 0.12.3
- lxml from 3.2.3 to 3.3.1
- netcdf4 from 1.0.7 to 1.0.8
- numba from 0.12.0 to 0.12.1
- numexpr from 2.3.0 to 2.3.1
- pandas from 0.13.0 to 0.13.1
- pykit from 0.1.0 to 0.2.0
- python from 3.3.3 to 3.3.4
- setuptools from 2.1 to 2.2

Added:

- added Py3k support for h5py
- netcdf4 on 32-bit Windows
- xlsxwriter 0.5.2

2014-02-10: 1.9.0:

Highlights:

- NumPy updated to version 1.8
- updates to about 35 packages, including Bokeh and Blaze
- added 10 new packages

Enhancements:

- use MSVC 2010 to compile Python 3 and C extensions on Windows
- remove distribute in favor of setuptools
- enable threadsafe when building HDF5 on Linux
- renamed the Python Imaging Library (PIL) from imaging to pil
- updated EULA

Fixes:

- ssl bindings in Python standard library on macOS
- Windows installers not working properly when PYTHONHOME is set

Updated:

- apptools from 4.2.0 to 4.2.1
- astropy from 0.2.5 to 0.3.0
- atom from 0.3.4 to 0.3.6
- binstar from 0.3.1 to 0.4.4
- biopython from 1.62 to 1.63
- blaze from 0.3.0 to 0.4.1

- boto from 2.15.0 to 2.24.0
- bokeh from 0.2 to 0.4
- chaco from 4.3.0 to 4.4.1
- conda from 2.0.4 to 3.0.3
- cython from 0.19.2 to 0.20
- dynd-python 0.5.0 to 0.6.0
- enaml from 0.8.3 to 0.9.0
- envisage from 4.3.0 to 4.4.1
- gevent from 0.13.8 to 1.0
- gevent-websocket from 0.3.6 to 0.9.2
- greenlet from 0.4.1 to 0.4.2
- h5py from 2.2.0 to 2.2.1
- jinja2 from 2.7.1 to 2.7.2
- keyring from 3.2 to 3.3
- llvmmath from 0.1.1 to 0.1.2
- llvmpy from 0.12.0 to 0.12.1
- netcdf4 from 1.0.6 to 1.0.7
- numexpr from 2.2.2 to 2.3.0
- numba from 0.11.0 to 0.12.0
- numpy from 1.7.1 to 1.8.0
- openpyxl from 1.6.2 to 1.8.2
- pandas from 0.12.0 to 0.13.0
- pip from 1.4.1 to 1.5.2
- psutil from 1.1.2 to 1.2.1
- py from 1.4.17 to 1.4.20
- pycparser from 2.9.1 to 2.10
- pyface from 4.3.0 to 4.4.0
- pyparsing from 1.5.6 to 2.0.1
- pyreadline from 2.0.dev to 2.0
- pytables from 3.0.0 to 3.1.0
- pytest from 2.4.2 to 2.5.2
- python from 2.7.5 to 2.7.6, and 3.3.2 to 3.3.3
- redis-py from 2.7.2 to 2.9.1
- requests 1.2.3 to 2.2.1
- scipy from 0.13.0 to 0.13.2
- six from 1.4.1 to 1.5.2

- sphinx from 1.1.3 to 1.2.1
- sqlalchemy from 0.8.3 to 0.9.2
- sympy from 0.7.3 to 0.7.4.1
- tornado from 3.1.1 to 3.2.0
- theano from 0.5.0 to 0.6.0
- traits from 4.3.0 to 4.4.0
- traitsui from 4.3.0 to 4.4.0

Added:

- argcomplete 0.6.7
- blz 0.6.1
- cdecimal 2.3
- conda-build 1.1.0
- datashape 0.1.0
- future 0.11.2
- kiwisolver 0.1.2
- mock 1.0.1
- setuptools 2.1
- ssl_match_hostname 3.4.0.2
- ujson 1.33
- scikit-learn and redis-py support for Python 3
- added traits and unittest2 support for Python 2.6

Removed:

- distribute
- llvmmath
- meta

2013-11-05: 1.8.0:

Enhancements:

- on Windows the installer now allows installing in user mode without having system administrator privileges
- use RPATH (instead of RUNPATH) on Linux, which fixes link problems when LD_LIBRARY_PATH is set
- added ipython-notebook meta package for all supported Python versions
- removed curl binary on macOS, to avoid superseding system curl

Fixes:

- wrong location of VTK libraries in VTKTargets-debug.cmake on Linux

- fix python.app problem related to adding the symlink to lib, which also appears to create problems with %%cython magic in ipython
- apply patch to Mayavi, which foxes running mlab scripts

Updated:

- astropy from 0.2.4 to 0.2.5
- atom from 0.3.2 to 0.3.4
- biopython from 1.61 to 1.62
- boto from 2.12.0 t 2.15.0
- bokeh from 0.1.1 to 0.2
- conda from 1.9.1 to 2.0.4
- cython from 0.19.1 to 0.19.2
- dynd from 0.4.2 to 0.5.0
- enaml from 0.7.19 to 0.8.3
- ipython 1.0.0 to 1.1.0
- keyring from 3.0.1 to 3.2
- matplotlib from 1.3.0 to 1.3.1
- netcdf4 from 1.0.5 to 1.0.6
- numba from 0.10.2 to 0.11.0
- numexpr from 2.0.1 to 2.2.2
- psutil from 1.0.1 to 1.1.2
- pycrypto from 2.6 to 2.6.1
- pytables from 2.4.0 to 3.0.0
- python 2.6.8 to 2.6.9
- pytest from 2.3.5 to 2.4.2
- scikit-image from 0.8.2 to 0.9.3
- scipy from 0.12.0 to 0.13.0
- sqlalchemy from 0.8.2 to 0.8.3
- sypder from 2.2.4 to 2.2.5

Added:

- blaze 0.3
- pykit 0.1
- beautiful-soup 4.3.1
- openpyxl 1.6.2

2013-09-10: 1.7.0:

Highlights:

- added VTK, Mayavi and Bokeh
- updated to new ipython 1.0 and matplotlib 1.3, and llvm 3.3
- many improvements and bug fixes to conda

Enhancements:

- removed conda as a direct anaconda dependency
- add MinGW runtime DLLs to Scripts directory on Windows
- fixed python-dateutil for Python 2

Updated:

- astropy from 0.2.3 to 0.2.4
- atom from 0.2.3 to 0.3.2
- binstar from 0.1.2 to 0.3.1
- boto from 2.9.6 to 2.12.0
- conda from 1.7.2 to 1.9.1
- docutils from 0.10 to 0.11
- enaml from 0.7.6 to 0.7.19
- h5py from 2.1.1 to 2.2.0
- ipython from 0.13.2 to 1.0.0
- itsdangerous from 0.21 to 0.23
- jinja2 from 2.6 to 2.7.1
- libdynd and dynd-python from 0.4.0 to 0.4.2
- llvm from 3.2 to 3.3
- llvmmath from 0.1.0 to 0.1.1
- llvmpy from 0.11.3 to 0.12.0
- lxml from 3.2.1 to 3.2.3
- keyring from 1.4 to 3.0.1
- matplotlib from 1.2.1 to 1.3.0
- netcdf4 from 1.0.4 to 1.0.5
- networkx fomr 1.7 to 1.8.1
- numba from 0.9.0 to 0.10.2
- opencv from 2.4.2 to 2.4.6
- pandas from 0.11.0 to 0.12.0
- pep8 from 1.4.5 to 1.4.6
- pip from 1.3.1 to 1.4.1
- psutil from 0.7.1 to 1.0.1

- pyflakes from 0.7.2 to 0.7.3
- pysal 1.5.0 to 1.6.0
- pyside from 1.1.2 to 1.2.1 (not macOS)
- qt from 4.7.4 to 4.8.5 (not macOS)
- scikit-learn from 0.13.1 to 0.14.1
- six from 1.3.0 to 1.4.1
- spyder from 2.2.0 to 2.2.4
- sqlalchemy 0.8.1 to 0.8.2
- statsmodels from 0.4.3 to 0.5.0
- sympy from 0.7.2 to 0.7.3
- tornado form 3.1 to 3.1.1
- werkzeug from 0.9.1 to 0.9.4

Added:

- bokeh 0.1.1
- chaco 4.3.0
- configobj 4.7.2
- markupsafe 0.18
- mayavi 4.3.0
- patsy 0.2.1
- traits 4.3.0
- vtk 5.10.1

2013-07-09: 1.6.2: (Windows only)

- Fixed path being incorrectly appended on Windows during install

2013-07-03: 1.6.1:

- fixed .pkg installer
- updated binstar from 0.1.1 to 0.1.2 (fixes upload on Windows)
- updated conda from 1.7.1 to 1.7.2 (fixes updating python.app on macOS)
- updated dynd from 0.3.0 to 0.4.0
- updated launcher from 0.1.1 to 0.1.2
- added missing pydoc command on Unix
- macOS: fix hard coded build prefix in .dylib files for may other libraries
- Windows: append instead of prepend path on Windows

2013-06-21: 1.6.0:

Highlights:

- the “Launcher”, which allows users to start up applications
- conda has now the ability to build conda packages, which can be uploaded to binstar.org
- conda now uses a SAT solver to solve the install dependency problem

Enhancements:

- added a .pkg installer for macOS, such that installing Anaconda does not require command line access
- create copy of .bashrc (.bash_profile on macOS) before appending the PATH setting upon install
- Windows Menu items should are now installed system wide

Fixes:

- removed Lib/test/testbz2_bigmem.bz2 from Windows, this (harmless) file was triggering Trojan-ArcBomb.BZip.Agent warning, see: http://bugs.python.org/issue17843
- json.decoder in Python 2.7 standard library not using .decode(‘hex’), this was (mysteriously) causing install problems on macOS
- fixed PIL._imagetk on Linux

Updated:

- python from 2.7.4 to 2.7.5, and 3.3.1 to 3.3.2
- astropy from 0.2.1 to 0.2.3
- boto from 2.9.2 to 2.9.6
- conda from 1.5.2 to 1.7.1
- cython from 0.19 to 0.19.1
- distribute from 0.6.36 to 0.6.45
- flask from 0.9 to 0.10.1
- greenlet from 0.4.0 to 0.4.1
- llvmpy from 0.11.2 to 0.11.3
- lxml form 3.2.0 to 3.2.1
- numba from 0.8.1 to 0.9.0
- py from 1.4.12 to 1.4.14
- pytest from 2.3.4 to 2.3.5
- requests from 1.2.0 to 1.2.3
- tornado from 3.0.1 to 3.1
- werkzeug from 0.8.3 to 0.9.1

Added:

- launcher 0.1.1
- binstar 0.1.1
- itsdangerous 0.21

- keyring 1.4
- llvmmath 0.1.0
- pep8 1.4.5
- pywin32 218.4 on Windows

2013-05-09: 1.5.1: (macOS only)

- fixed pip command
- replace spaces by underscore in user install location
- shared libraries now link to /usr/lib/libgcc_s.1.dylib (instead of /usr/local/lib/libgcc_s.1.dylib)

2013-05-08: 1.5.0:

Highlights:

- updates to all important packages: python, numpy, scipy, ipython, matplotlib, pandas, cython
- added netCDF4 (with OpenDAP support) on Linux and macOS

Fixed:

- Cython on macOS on Python 2.6, due to a problem with Pyhton's config/Makefile
- py.test command (all platforms)
- python-config command not using correct interpreter on macOS
- added missing MSVC 2008 and 2010 runtime to Windows installers
- removed hard-coded build location from identification name of dynamic shared libraries on macOS, to allow easier building against libraries in Anaconda

Enhancements:

- **The Windows installers has a new dialog box with the following options:**
  - "Add Anaconda to the System Path"
  - "Register Anaconda as default Python version of the system"
- The Unix installers have an option to add Anaconda to the path in the ~/.bashrc (on Linux) or ~/.bash_profile (on macOS). On macOS the default is 'yes', on Linux the default is 'no'.
- added armv6l supports (only Python 2.7 and selected packages)
- link pycurl and libnetcdf dynamically against curl library on Unix
- add configure option –with-pgm for zeromq on Unix

Added:

- netCDF4 (with OpenDAP support) 1.0.4 on Linux and macOS
- ordereddict 1.1 (on Python 2.6)
- pycosat 0.6.0
- atom 0.2.3
- enaml 0.7.6

- casuarius 1.1
- dynd-python 0.3.0
- rope 0.9.4

Updates:

- python from 2.7.3 to 2.7.4, and 3.3.0 to 3.3.1
- numpy from 1.7.0 to 1.7.1
- scipy from 0.11.0 to 0.12.0
- ipython from 0.13.1 to 0.13.2
- matplotlib from 1.2.0 to 1.2.1
- astropy from 0.2 to 0.2.1
- biopython from 1.60 to 1.61
- bitarray from 0.8.0 to 0.8.1
- boto from 2.8.0 to 2.9.2
- conda from 1.4.4 to 1.5.2
- curl from 7.26.0 to 7.30.0
- cython from 0.18 to 0.19
- distribute from 0.6.34 to 0.6.36
- llvmpy from 0.11.1 to 0.11.2
- lxml from 3.0.2 to 3.2.0
- nose from 1.2.1 to 1.3.0
- numba from 0.7.0 to 0.8.1
- pandas from 0.10.1 to 0.11.0
- psutil from 0.6.1 to 0.7.1
- pyflakes from 0.6.1 to 0.7.2
- pytz from 2012j to 2013b
- requests from 0.13.9 to 1.2.0
- scikit-learn from 0.13 to 0.13.1
- six from 1.2.0 to 1.3.0
- spyder from 2.1.13 to 2.2.0
- sqlalchemy from 0.7.8 to 0.8.1
- sympy from 0.7.1 to 0.7.2
- tornado from 2.4.1 to 3.0.1
- xlrd from 0.9.0 to 0.9.2
- xlwt from 0.7.4 to 0.7.5

Removed (from installer only, still available with conda):

- bitey

- gdata (conflicts with atom (which is required by enaml))
- googlecl (requires gdata)

2013-03-12: 1.4.0:

Many more packages are now supported on Python 3.3. In addition, we have added Python 3 support on Windows, such that now Python 2.6, 2.7 and 3.3 is supported across all platforms. In addition, we have redesigned the Windows installer, which was suffering from a number of problems, in particular the lack of uninstall functionality and the GUI not working on some systems.

added:

- astropy 0.2
- lxml 3.0.2
- pycparser 2.9.1
- six 1.2.0
- xlrd 0.9.0
- xlwt 0.7.4

updated:

- boto from 2.7.0 to 2.8.0
- conda from 1.3.5 to 1.4.2
- cubes from 0.10.1 to 0.10.2
- cython from 0.17.4 to 0.18
- dateutil from 1.5 to 2.1
- llvmpy from 0.10.2 to 0.11.1
- numba from 0.6.0 to 0.7.0
- numpy from 1.7rc1 to 1.7.0
- pyflakes from 0.5.0 to 0.6.1
- pygments from 1.5 to 1.6
- pysal from 1.4.0 to 1.5.0
- pyreadline from 1.7.1 to 2.0.dev
- pytz from 2012d to 2012j
- scikit-image from 0.7.1 to 0.8.2

fixed:

- pytables on Windows

other notes:

- Linux: a ATLAS package is now available (conda install atlas). In addition the site.cfg in the numpy.distutils has been updated, such that other packages (e.g. scipy) which use the atlas build configuration can be build against the (non-MKL linked) numpy in Anaconda.

2013-02-06: 1.3.1:

added:

- Python 2.6 support for iopro, numba and numbapro
- Python 2.6 support on Windows (Python 2.6 is now supported on all platforms)
- added pythonw (the command to run Python GUI applications) on macOS
- added chaco on Windows

updates:

- conda from 1.3.2 to 1.3.5
- iopro from 1.3.0 to 1.3.2
- llvmpy from 0.10.0 to 0.10.2
- numba from 0.5.0 to 0.6.0
- numbapro from 0.8.1 to 0.8.1

other changes:

- removed anaconda-launcher

2013-01-23: 1.3.0:

fixes:

- fixed missing Grammar.txt in Sphinx
- recompiled llvm and llvmpy using gcc 4.4.6 on Linux, this fixes a problem with the mandel.py example in numba
- made Windows installer dialog box resizeable
- fixed problem that importing numbapro or iopro removed sys from the namespace

updates:

- LLVM from 3.1 to 3.2
- llvmpy from 0.9 to 0.10.0
- numba from 0.3.2 to 0.5.0
- numbapro from 0.7.3 to 0.8.0
- iopro from 1.2.3 to 1.3.0
- conda from 1.2.1 to 1.3.0
- pandas from 0.9.0 to 0.10.1
- cython from 0.17.1 to 0.17.4
- iopro from 1.2.2 to 1.2.3
- spyder form 2.1.11 to 2.1.13
- h5py from 2.1.0 to 2.1.1
- distribute form 0.6.30 to 0.6.34
- nose from 1.1.2 to 1.2.1

- tornado from 2.3 to 2.4.1
- docutils form 0.9.1 to 0.10
- nltk from 2.0.3 to 2.0.4
- gevent from 0.13.7 to 0.13.8
- numpy from 1.7.0b2 to 1.7.0rc1
- boto from 2.6.0 to 2.7.0
- scikit-learn from 0.11 to 0.12.1
- scikits-image from 0.6.1 to 0.7.1
- pyaudio from 0.2.6 to 0.2.7
- pytest from 2.3.3 to 2.3.4
- redis from 2.4.15 to 2.6.9
- redis-py from 2.4.13 to 2.7.2
- disco from 0.4.2 to 0.4.4

added:

- Tkinter support on all platforms
- redis support on macOS
- cubes 0.10.1
- ply 3.4
- pycrypto 2.6
- pyparsing 1.5.6
- googlecl 0.9.12
- gdata 2.0.17
- biopython 1.60

2012-11-21: 1.2.1:

- pucurl on macOS
- anaconda-launcher envs by updating to conda 1.2.1
- add missing pyodbc numpy_tests in iopro/tests/pyodbc
- updated wiseRF to version 1.1
- add creation of .continuum directory (if not created yet) on Windows
- minor fixes in numba, numbapro, and iopro
- fixed version of “py” package
- add missing Windows manifest to Windows executable installer
- fixed Windows Menu install and making Anaconda the default Python
- on 2012-12-06 we released a 32-bit Linux version

2012-11-13: 1.2.0:

- performance and feature enhancements to Numba Pro
- performance and feature enhancements to IOPro
- improved conda command (package management)
- added Qt to the Linux Version (Qt is now on all platforms)
- added MDP, NLTK and py, pytest
- update matplotlib from 1.1.1 to 1.2.0
- update h5py from 2.0.1 to 2.1.0
- update IOPro to 1.2.1
- update libpng to 1.5.13

2012-10-05: 1.1.0:

- add GUI to Windows installer
- IDE Spyder (Qt) for Mac Version
- add conda 1.0
- update llvmpy to 0.8.3
- add MinGW on Windows

2012-09-06: 1.0.0:

- add Windows support
- installer can now install into different locations
- enable building free and permium version
- enable termcap in erlang
- add MKL support to permium version
- add networkx, pysal, pycurl, gevent_zeromq, requests, pip, distribute
- add iopro to permium version
- update scipy from 0.11.0b1 to 0.11.0rc2
- update scikits-image form 0.6 to 0.6.1
- update pytables from 2.4.0b1 to 2.4.0
- update pandas from 0.8.0 to 0.8.1
- add patch to disco to always use the anaconda erl
- remove useless files (Python) from being installed

2012-08-21: 0.9.0:

- add macOS (x86_64 10.5 or higher) support
- add bitey and other packages
- update several other packages

2012-07-18: 0.8.3:

- update changes to etc/init.d/disco script
- add patch, fixes disco and ddfs listdir misfeature
- add –packages option to anaconda command
- add missing h5py
- improve ease of testing

2012-07-18: 0.8.2:

- fixed theano.sparse
- removed (broken) scikits namespace
- add disco config and setup files
- add anaconda command, for version information

2012-07-17: 0.8.1:

- fixed libm.so ctypes error in scipy tests
- added import tests to all C extension modules
- fixed lzo support in pytables

2012-07-17: 0.8.0:

- initial release

1.2.8 Glossary

Anaconda Cloud A web-based repository hosting service in the cloud. Packages created locally can be published to the cloud to be shared with others. Free accounts on Cloud can publish packages to be shared publicly. Paid subscriptions to Cloud can designate packages as private to be shared with authorized users. Cloud is Anaconda's *Anaconda Repository* product made available to the public. Repository is also available for purchase by companies that wish to power their own on-premise version of Cloud. See product comparison.

Anaconda Distribution A downloadable, free, open source, high-performance, optimized Python and R distribution with 200+ packages plus easy access to an additional 620+ popular open source packages for data science including advanced and scientific analytics. It also includes *conda*, an open source package, dependency and environment manager. Thousands more open source packages can be installed with the `conda` command. Available for Windows, macOS and Linux, all versions are supported by the community.

Anaconda Navigator A desktop graphical user interface (GUI) to conda that is included in all versions of *Anaconda Distribution*. Navigator allows you to easily manage *conda packages*, *environments*, *channels* and notebooks without the need for the *command line interface*.

Anaconda Enterprise Anaconda with enterprise technical support for a specific number of users, indemnification for a select number of open source packages, collaborative notebooks, high-performance scalability, Hadoop, interactive visualization, governance and security. See the subscriptions page for more details.

Anaconda Enterprise Notebook A browser-based Python data analysis environment and visualization tool in a secure, governed environment. Allows data science team members to create and share private notebooks, manage access, control notebook revisions, compare and identify differences across notebook versions, search notebooks for keywords and packages, use enhanced collaborative notebook features including revision control and locking—and to access an on-premises and/or cloud collaborative notebook server.

Anaconda Enterprise Repository An enterprise server on your network where open source and proprietary packages may be stored, retrieved and installed on a local computer. Anaconda Repository is different from Anaconda Cloud or the default conda repository. The repository is used to govern access to data science assets including packages and notebooks.

Anaconda repository A cloud-based repository that contains 1,000+ open source certified packages that are easily installed locally with the `conda install` command. Can be accessed by anyone using *conda* commands or viewed directly at https://repo.continuum.io/pkgs/ .

Channels The locations of the repositories where conda looks for packages. Channels may point to a Cloud repository or a private location on a remote or local repository that you or your organization created. The `conda channel` command has a default set of channels to search beginning with https://repo.continuum.io/pkgs/ . You may override the default channels, for example to maintain a private or internal channel. In conda commands and in the `.condarc` file, these default channels are referred to by the channel name `defaults`.

Command Line Interface (CLI) A program in which commands are entered as text, one line at a time, for a computer to execute. Also referred to as a terminal. Contrast with *Graphical User Interface (GUI)*.

Conda A package and environment manager program bundled with *Anaconda Distribution* that finds, installs and updates *conda packages* and their dependencies. Also lets you easily switch between *conda environments* on your local computer.

Conda build A program that assembles the necessary components to create a *conda package* using *conda* commands. A conda build is done locally with your own (optional) clusters.

Conda environment A folder or directory that contains a specific collection of *conda packages* and their dependencies, so that they can be maintained and run separately without interference from each other. For example, you may use one conda environment for only Python 2.7 and Python 2.7 packages, and maintain another conda environment with only Python 3.5 and Python 3.5 packages.

Conda package A compressed file that contains everything that a software program needs in order to be installed and run, including system-level libraries, Python modules, executable programs and other components. With a conda package, you do not have to manually find and install each dependency separately. Managed with *conda*.

Environment manager A software tool that maintains a separation of computer programs so they do not interfere with each other. Also known as an environment management system. Conda is an example of an environment manager.

Graphical User Interface (GUI) A program with graphic images, icons and menus in which commands are entered by clicking with a mouse and/or entering text in form boxes. It is an easy-to-use overlay to a program that is run using a *Command Line Interface (CLI)*. *Anaconda Navigator* is a GUI for the conda program.

Miniconda A minimal installer for conda. Like *Anaconda Distribution*, Miniconda is a free software package that includes the Anaconda Distribution and the *conda* package and environment manager, but Miniconda does not include any packages other than those dependencies needed to install it. After installing Miniconda, you can install additional conda packages directly from the command line with `conda install`.

Microsoft R Open (MRO) An enhanced distribution of the R language from Microsoft Corporation. It is a complete open source platform for statistical analysis and data science.

Package manager A collection of software tools that automates the process of installing, updating, configuring and removing computer programs for a computer's operating system. Also known as a package management system. Conda is an example of a package manager.

Packages Software files and information about the software, such as its name, the specific version and a description, that are bundled into a file that can be installed and managed by a package manager.

R packages *Conda packages* that install and run the R computer language. Examples include R Essentials, a bundle of 80 popular, open source software programs written in the R computer language. See http://conda.io/docs/r-with-conda.html.

Repository Any storage location from which software or software assets may be retrieved and installed on a local computer. See also *Anaconda repository* and *Anaconda Enterprise Repository*.

Terminal A program in which commands are entered as text, one line at a time, for a computer to execute. Also referred to as a Command Line Interface. Contrast with *Graphical User Interface (GUI)*.

1.2.9 Anaconda End User License Agreement

Notice of Third Party Software Licenses

Anaconda Distribution contains open source software packages from third parties. These are available on an "as is" basis and subject to their individual license agreements. These licenses are available in Anaconda Distribution or at http://docs.anaconda.com/anaconda/pkg-docs. Any binary packages of these third party tools you obtain via Anaconda Distribution are subject to their individual licenses as well as the Anaconda license. Anaconda, Inc. reserves the right to change which third party tools are provided in Anaconda Distribution.

In particular, Anaconda Distribution contains re-distributable, run-time, shared-library files from the Intel(TM) Math Kernel Library ("MKL binaries"). You are specifically authorized to use the MKL binaries with your installation of Anaconda Distribution. You are also authorized to redistribute the MKL binaries with Anaconda Distribution or in the conda package that contains them. Use and redistribution of the MKL binaries are subject to the licensing terms located at https://software.intel.com/en-us/license/intel-simplified-software-license. If needed, instructions for removing the MKL binaries after installation of Anaconda Distribution are available at http://www.anaconda.com.

Anaconda Distribution also contains cuDNN software binaries from NVIDIA® Corporation ("cuDNN binaries"). You are specifically authorized to use the cuDNN binaries with your installation of Anaconda Distribution. You are also authorized to redistribute the cuDNN binaries with an Anaconda Distribution package that contains them. If needed, instructions for removing the cuDNN binaries after installation of Anaconda Distribution are available at http://www.anaconda.com.

Anaconda Distribution also contains Visual Studio Code software binaries from Microsoft Corporation ("VS Code"). You are specifically authorized to use VS Code with your installation of Anaconda Distribution. Use of VS Code is subject to the licensing terms located at https://code.visualstudio.com/License.

Cryptography Notice

This distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See the Wassenaar Arrangement http://www.wassenaar.org/ for more information.

Anaconda, Inc. has self-classified this software as Export Commodity Control Number (ECCN) 5D992b, which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries. The Intel(TM) Math Kernel Library contained in Anaconda, Inc.'s software is classified by Intel(TM) as ECCN 5D992b with no license required for export to non-embargoed countries and Microsoft's Visual Studio Code software is classified by Microsoft as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages are included in this distribution that relate to cryptography:

openssl The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols as well as a full-strength general purpose cryptography library.

pycrypto A collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.).

pyopenssl A thin Python wrapper around (a subset of) the OpenSSL library.

kerberos (krb5, non-Windows platforms) A network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.

cryptography A Python library which exposes cryptographic recipes and primitives.

Anaconda® is a package manager, an environment manager, a Python distribution, and a collection of *over 1,000+ open source packages*. It is free and easy to install, and it offers free community support.

Get the `Anaconda Cheat Sheet` and then download Anaconda.

Don't want the huge collection of 1,000+ software packages? Get Miniconda.

1.2.10 Anaconda Navigator or conda?

After you install Anaconda or Miniconda, if you prefer a desktop graphical user interface (GUI) then use Navigator. If you prefer to use Anaconda prompt (or Terminal on Linux or macOS), then use conda. You can also switch between them.

You can install, remove or update any Anaconda package with a few clicks in Navigator, or with a single conda command in Anaconda Prompt (Terminal on Linux or macOS).

To try Navigator, after installing Anaconda, click the Navigator icon on your operating system's program menu, or in Anaconda prompt (or Terminal on Linux or macOS), run the command `anaconda-navigator`.

To try conda, after installing Anaconda or Miniconda, take the 30-minute conda test drive, download a conda cheat sheet and see Using conda for fun things to do with conda.

1.2.11 Packages available in Anaconda

- Over *150 packages* are automatically installed with Anaconda.
- Over 250 additional open source packages can be individually installed from the Anaconda repository with the `conda install` command.
- Thousands of other packages are available from Anaconda Cloud.
- You can download other packages using the `pip install` command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together.
- You can also make your own custom packages using the `conda build` command, and you can share them with others by uploading them to Anaconda Cloud, PyPi or other repositories.

1.2.12 Previous versions

Previous versions of Anaconda are available in the archive. For a list of packages included in each previous version, see *Old package lists*.

Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.6. However, it does not matter which one you download, because you can create new environments that include any version of Python packaged with conda. See Managing Python with conda.

1.2.13 Product archive

The following Anaconda products have been replaced and are no longer available to new users. Documentation for these products is available for existing users only:

- *Anaconda Launcher* (replaced by *Anaconda Navigator*).
- *Anaconda for Cluster Management* (replaced by *Anaconda Scale*).
- *NumbaPro* (replaced by Numba).
- *Anaconda Accelerate* (replaced by Numba).
- *MKL Optimizations* (now freely available in Anaconda.)

Also, the former Anaconda product *IOPro* was open sourced.

Older versions of some of these components required *license installation*.

NumbaPro

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it.

NumbaPro has been deprecated, and its code generation features have been moved into open-source Numba. The CUDA library functions have been moved into Accelerate, along with some Intel MKL functionality. High-level functions and access to additional native library implementations will be added in future releases of Accelerate, and there will be no further updates to NumbaPro.

A NumbaPro compatibility layer (listed as release 0.22.0 of NumbaPro) provides access to the new Accelerate packages through the old NumbaPro package names. This avoids the need to change any existing code immediately for use with Accelerate. A warning will be generated upon import of the compatibility layer, to highlight the deprecation of the NumbaPro package.

Archived NumbaPro documentation is still available.

NumbaPro

Get started with the NumbaPro Quick Start [pdf]

NumbaPro is an enhanced version of Numba which adds premium features and functionality that allow developers to rapidly create optimized code that integrates well with NumPy.

With NumbaPro, Python developers can define NumPy ufuncs and generalized ufuncs (gufuncs) in Python, which are compiled to machine code dynamically and loaded on the fly. Additionally, NumbaPro offers developers the ability to target multicore and GPU architectures with Python code for both ufuncs and general-purpose code.

For targeting the GPU, NumbaPro can either do the work automatically, doing its best to optimize the code for the GPU architecture. Alternatively, CUDA-based API is provided for writing *CUDA* code specifically in Python for ultimate control of the hardware (with thread and block identities).

Getting Started

Let’s start with a simple function to add together all the pairwise values in two NumPy arrays. Asking NumbaPro to compile this Python function to vectorized machine code for execution on the CPU is as simple as adding a single line of code (invoked via a decorator on the function):

Similarly, one can instead target the GPU for execution of the same Python function by modifying a single line in the above example:

```
@vectorize(['float32(float32, float32)'], target='gpu')
```

Targeting the GPU for execution introduces the potential for numerous GPU-specific optimizations so as a starting point for more complex scenarios, one can also target the GPU with NumbaPro via its *Just-In-Time* (JIT) compiler:

Features

Here’s a list of highlighted features:

- Portable data-parallel programming through ufuncs and gufuncs for single core CPU, multicore CPU and GPU
- Bindings to CUDA libraries: cuRAND, cuBLAS, cuFFT
- Python CUDA programming for maximum control of hardware resources

User Guide

New users should first read the installation manual:

Installation

NumbaPro is part of the *Anaconda Accelerate* product. Please refer to the Anaconda Accelerate Documentation for instructions on downloading and installation.

Purchasing a License

Accelerate is included with Anaconda Workgroup and Anaconda Enterprise subscriptions.

To start a 30-day free trial just download and install the Anaconda Accelerate package.

If you already have Anaconda (free Python distribution) installed:

```
conda update conda
conda install accelerate
```

If you do not have Anaconda installed, you can download it here.

Anaconda Accelerate can also be installed into your own (non-Anaconda) Python environment. For more information about Accelerate please contact sales@anaconda.com.

Update Instructions

With Anaconda Accelerate already installed, first update the conda package management tool to the latest version, then use conda to update the NumbaPro module. In the terminal:

```
$ conda update conda
$ conda update numbapro
```

CUDA GPUs Setup

CUDA Driver

NumbaPro does not ship the CUDA driver. It is users responsibility to ensure their systems are using the **latest driver**. Currently, users should use the driver shipped with CUDA 5.5 SDK.

CUDA Support & Detection

NumbaPro GPU support currently requires NVIDIA CUDA GPUs with **compute-capability 2.0 or above**. Users should check their hardware with the following:

```
import numbapro
numbapro.check_cuda()
```

A sample output looks like:

```
------------------------------libraries detection-------------------------------
Finding cublas
    located at /Users/.../lib/libcublas.dylib
Finding cusparse
    located at /Users/.../lib/libcusparse.dylib
```

(continues on next page)

(continued from previous page)

```
Finding cufft
    located at /Users/.../lib/libcufft.dylib
Finding curand
    located at /Users/.../lib/libcurand.dylib
Finding nvvm
    located at /Users/.../lib/libnvvm.dylib
In /Users/.../lib
    finding libdevice.compute_20.bc         ok
    finding libdevice.compute_30.bc         ok
    finding libdevice.compute_35.bc         ok
-------------------------------hardware detection-------------------------------
Found 1 CUDA devices
id 0         GeForce GT 650M                              [SUPPORTED]
                      compute capability: 3.0
                           pci device id: 0
                              pci bus id: 1
Summary:
    1/1 devices are supported
```

This performs CUDA library and GPU detection. Discovered GPUs are listed with information for compute capability and whether it is supported by NumbaPro.

The message “cuda disabled by user” means that either the environment variable NUMBA_DISABLE_CUDA is set to 1 and must be set to 0, or the system is 32-bit. CUDA does not support 32-bit systems.

Basics:

Quick Start

Numba/NumbaPro uses decorators extensively to annotate function for compilation. This document explains the major decorators: `jit`, `autojit`, `vectorize` and `guvectorize`.

Types

Numba/NumbaPro decorators specializes generic python function to typed native function. All decorators except `autojit` requires type information to be supplied. Here is a table of common Numba type objects:

Kind	Numba Types
signed integer	int8, int16, int32, int64
unsigned integer	uint8, uint16, uint32, uint64
float-points	float32, float64
complex numbers	complex64, complex128
boolean	bool_
others	void

Compiler Target `target="..."`

All decorators, `jit`, `autojit`, `vectorize` and `guvectorize`, have a target keyword argument to select the code generation target. User provides a string to name the target. Numba supports only the `"cpu"` target. NumbaPro adds `"parallel"` and `"gpu"`. The `"parallel"` target is only available for `vectorize`, which will distributes the work across CPU threads. The “gpu” offloads the computation to a Nvidia CUDA GPU.

High-level APIs for CPU/GPU:

Universal Functions

With NumbaPro, universal functions (ufuncs) can be created by applying the *vectorize* decorator on to simple scalar functions. A ufunc can operates on scalars or NumPy arrays. When used on arrays, the ufunc apply the core scalar function to every group of elements from each arguments in an element-wise fashion. NumPy Broadcasting is applied to every argument with mismatching dimensions.

Example: Basic

Here is a simple example to perform element-wise addition:

The ufunc is compiled to operate on *float32* and *float64* arrays. It is used to compute element-wise addition of array *a* and *b* which are arrays of *numpy.float64* with 10 elements. The output

Usage

A generalization of the usage of the *vectorize* decorator is described in this section.

vectorize(*type_signatures*[, *target='cpu'*])

Returns a vectorizer object to be applied to python functions.

Parameters

- **type_signatures** – an iterable of type signatures, which are either function type object or a string describing the function type.
- **target** – a string for hardware target; e.g. "cpu", "parallel", "gpu".

Returns a vectorizers object.

To use multithreaded version, change the target to "parallel":

For CUDA target, use "gpu" for target: <div class="highlight-default"><div class="highlight"><pre><span></span><span class="kn">from</span> <span class="nn">numbapro</span> <span class="k">import</span> <span class="n">vectorize</span>

Performance Guideline

A general guideline is to choose different targets for different data sizes and algorithms. The "cpu" target works well for small data sizes (approx. less than 1KB) and low compute intensity algorithms. It has the least amount of overhead. The "parallel" target works well for medium data sizes (approx. less than 1MB). Threading adds a small delay. The "gpu" target works well for big data sizes (approx. greater than 1MB) and high compute intensity algorithms. Transfering memory to and from the GPU adds significant overhead.

Universal Function Targets

There are several vectorizer versions available. The different options are listed below:

Target	Description
cpu	Single-threaded CPU
parallel	Multi-core CPU
stream	Optimize for CPU cache **Note:** Experimental. Computation speeds may vary.
gpu	CUDA GPU **Note:** This creates an *ufunc-like* object. See documentation for CUDA ufunc for detail.

Generalized Ufuncs

GPU Reduction

Python CUDA Programming

CUDA Programming Introduction

NumbaPro provides multiple entry points for programmers of different levels of expertise on CUDA. For expert CUDA-C programmers, NumbaPro provides a Python dialect <CUDAJit.html>‘_ for low-level programming on the CUDA hardware. It provides full control over the hardware for fine tunning the performance of CUDA kernels. For new CUDA programmers, the high-level API such as the universal functions (ufunc) and generalized ufuncs (gufunc) are the easiest way to write array operations for the GPU.

A Very Brief Introduction to CUDA

A CUDA GPU contains one or more *streaming multiprocessors* (SMs). Each SM is a manycore processor that is optimized for high throughput. The *manycore* architecture is very different from the common multicore CPU architecture. Instead of having a large cache and complex logic for instruction level optimization, a manycore processor achieves high throughput by executing many threads in parallel on many simpler cores. It overcomes latency due to cache miss or long operations by using zero-cost context switching. It is common to launch a CUDA kernel with hundreds or thousands of threads to keep the GPU busy.

The CUDA programming model is similar to the SIMD vector model in modern CPUs. A CUDA SM schedules the same instruction from a *warp* of 32-threads at each issuing cycle. The advantage of CUDA is that the programmer does not need to handle the divergence of execution path in a warp, whereas a SIMD programmer would be required to properly mask and shuffle the vectors. The CUDA model decouples the data structure from the program logic.

To know more about CUDA, please refer to NVIDIA CUDA-C Programming Guide.

CUDA Ufuncs and Generalized Ufuncs

This page describes the CUDA ufunc-like object.

To support the programming pattern of CUDA programs, CUDA Vectorize and GUVectorize cannot produce a conventional ufunc. Instead, a ufunc-like object is returned. This object is a close analog but not fully compatible with a regular NumPy ufunc. The CUDA ufunc adds support for passing intra-device arrays (already on the GPU device) to reduce traffic over the PCI-express bus. It also accepts a *stream* keyword for launching in asynchronous mode.

Example: Calling Device Functions

All CUDA ufunc kernels have the ability to call other CUDA device functions:

```
from numbapro import vectorize, cuda

# define a device function
@cuda.jit('float32(float32, float32, float32)', device=True, inline=True)
def cu_device_fn(x, y, z):
    return x ** y / z

# define a ufunc that calls our device function
@vectorize(['float32(float32, float32, float32)'], target='gpu')
def cu_ufunc(x, y, z):
    return cu_device_fn(x, y, z)
```

Generalized CUDA ufuncs

Generalized ufuncs may be executed on the GPU using CUDA, analogous to the CUDA ufunc functionality. This may be accomplished as follows:

```
from numbapro import guvectorize

@guvectorize(['void(float32[:,:], float32[:,:], float32[:,:])'],
             '(m,n),(n,p)->(m,p)', target='gpu')
def matmulcore(A, B, C):
    ...
```

There are times when the gufunc kernel uses too many of a GPU's resources, which can cause the kernel launch to fail. The user can explicitly control the maximum size of the thread block by setting the *max_blocksize* attribute on the compiled gufunc object.

```
from numbapro import guvectorize

@guvectorize(..., target='gpu')
def very_complex_kernel(A, B, C):
    ...

very_complex_kernel.max_blocksize = 32  # limits to 32 threads per block
```

Writing CUDA-Python

CUDA Libraries Host API

CUDA Device Management

CUDA Support

Note: Platforms supported: Linux, Windows and Mac OSX 32/64-bit. Only support CUDA devices with **compute capability 2.0 and above**. Please see CUDA GPUs for a list of CUDA GPUs and their compute capability.

Note: As of version 0.12.2, Anaconda is distributing a subset of CUDA toolkit 5.5 libraries. The following information is no longer necessary for most users.

NumbaPro searches in the system shared library path for the CUDA drivers and CUDA libraries (e.g. cuRAND, cuBLAS). Users can set environment variable **LD_LIBRARY_PATH** to the directory of the CUDA drivers to ensure that NumbaPro can find them. The instruction to do so is printed at the end of the CUDA SDK installation.

User can override the search path with the following environment variables:

- **NUMBAPRO_CUDA_DRIVER** path to CUDA driver shared library file
- **NUMBAPRO_NVVM** path to CUDA libNVVM shared library file
- **NUMBAPRO_LIBDEVICE** path to CUDA libNVVM libdevice directory which contains .bc files.

CUDA Python Specification (v0.2)

(This documents reflects the implementation of CUDA Python in NumbaPro 0.12. In time, we may refine the specification.)

As usage of Python on CUDA GPUs is becoming more mature, it has become necessary to define a formal specification for a dialect and its mapping to the PTX ISA. There are places where the semantic of this dialect differs from the Python semantic. The change in semantic is necessary for us to generate high-performance code that is otherwise hard to achieve.

No-Python Mode (NPM)

CUDA Python is a superset of the *No-Python mode* (NPM). NPM is a statically typed subset of the Python language. It does not use the Python runtime; thus, it only supports lower level types; such as booleans, ints, floats, complex numbers and arrays. It does not support Python objects. Since we drop the support for objects entirely, many basic language constructs must be handled differently. For instance, a simple for-loop is:

```python
for i in range(10):
    ...
```

where range returns an iteratable. NPM restricts the language so that only `range` or `xrange` can be used.

For array support, NPM models the NumPy ndarray. An array is a structure with a pointer to the data, an array of shape and an array of strides. Valid attributes are `shape`, `strides`, `size` and `ndim`. Arrays cannot be unpacked.

The only way to access array elements is through the __getitem__ and __setitem__ operators (e.g. ary[i, j]). Slicing is not supported. When indexing into an array, a N-dimension array must be provided with N indices.

Tuples are minimally supported for unpacking array shape and strides attributes and some return value of calls.

In time, we aim to enhance NPM to expand the supported subset and recognize more idiomatic Python patterns.

Summary:

- no object;
- no exception;
- for-loop only works on range or xrange;
- supported types: ints, floats, complex numbers, and arrays.

Type Inference

The type inference algorithm for CUDA Python differs from Numba as we recognize that its users require stronger typing to better predict code performance. This is a summary of the type inference rules:

- Implicit coercion for all ints and floats only.
- Variable type is assigned at definition but a variable can be redefined; thus its type can be modified at the next assignment.
- Inside a loop, the variable type remains unchanged even at redefinition. The type assigned at the preloop block (the dominator of the basic-block) is assumed. This greatly differs from Python semantic.

User can force the type of any value by using the type object defined in numbapro namespace:

```
from numbapro import cuda, int16, float32

@cuda.autojit
def a_cuda_kernel(arg):
    must_be_int16 = int16(123)
    must_be_float32 = float32(321)
```

Basic Arithmetic Operations

For binary operators + - *, the operands are coerced to the most generic type of the two before the computation. The result type is the coerced type.

For floordiv //, the coercion rule on operands for basic binary operators applies. But, the result type is always coerced to an integer of the same bitwidth and at least has 32-bits.

For truediv /, the operands are promoted to a floating point representation with bitwidth equals to the maximum of the two operands before the computation and at least has 32-bits. The result type is the coerced type.

For binary bitwise & | ^ >> <<, the operands must be of integer types and they are coerced to the most generic type of the two before the computation. The result type is the coerced type.

For complex numbers, only + - * are defined.

Please refer to the CUDA-C Programming Guide: Arithmetic Instructions for the precision each operation.

Intp

`intp` is used to represent the integer whose width equals the address width.

Array Operations

Array attributes are read-only:

- `shape` contains the number of elements for each dimension. It can be indexed or unpacked like a tuple. It is a tuple of intp.
- `strides` contains the number of bytes skip to move forward to the next element for a given dimension. It can be indexed or unpacked like a tuple. It is a tuple of intp.
- `size` contains the number of elements in the array but may not be correspond to the actual size of the data buffer since strides can be zero or negative. It is an intp.
- `ndim` contains the number of dimension in the array. It is an intp.

`__getitem__` returns the element at the given index. Slicing or fancy indexing are not supported. The result type is always the same as the element type of the array.

`__setitem__` stores a value into the array at an index. The value is coerced if necessary.

CUDA Intrinsics

All intrinsics are defined under the `numbapro.cuda` namespace.

Thread ID intrinsics:

- `cuda.threadIdx.x`, `cuda.threadIdx.y`, `cuda.threadIdx.z` are the X, Y and Z IDs of the thread.
- `cuda.blockIdx.x`, `cuda.blockIdx.y` are the X and Y ID of the block.
- `cuda.blockDim.x`, `cuda.blockDim.y`, `cuda.blockDim.z` are the X, Y and Z width of the thread block.
- `cuda.gridDim.x`, `cuda.gridDim.y` are the X and Y width of the grid.

Barrier intrinsics:

- `cuda.syncthreads()` equivalent to `__syncthreads()` in CUDA-C. It is a thread block level barrier.

Shared memory intrinsics

- `cuda.shared.array(shape, dtype)` constructs a statically allocated array in the shared memory of kernel. `dtype` argument must be a type object defined in the `NumbaPro` namespace. It must be declared in the entry block of the kernel.

Math

Python Dialect for CUDA translates math functions defined in the math module of the Python standard library. All the functions use the semantic of the CUDA-C definition. Please refer to the CUDA-C Programming Guide: Math Function.

Supported functions:

```
math.acos
math.asin
math.atan
math.arctan
math.acosh
math.asinh
math.atanh
math.cos
math.sin
math.tan
math.cosh
math.sinh
math.tanh
math.atan2
math.exp
math.expm1              # not available in Python 2.6
math.fabs
math.log
math.log10
math.log1p
math.sqrt
math.pow
math.ceil
math.floor
math.copysign
math.fmod
math.isnan
math.isinf
```

Fast Math

coming soon in the next release…

Learn by Examples

The developer team maintains a public GitHub repository of examples. Many examples are designed to show off the potential performance gain by using GPUs.

Requirements

Base:

- Python 2.6, 2.7, 3.3, 3.4
- LLVM 3.3

For CUDA GPU features:

- Latest NVIDIA CUDA driver
- CUDA Toolkit 5.5 or above
- At least one CUDA GPU with compute capability 2.0 or above

Python modules:

- llvmpy 0.12.7 or above
- numba 0.14.0 or above

Release Notes

Version (0.21.0)

NumbaPro will be deprecated with most code generation features moved into the opensource Numba and the CUDA bindings moved into a new commerical package called "Accelerate". The new package will feature more high-level API functions from the CUDA libraries as well as MKL.

The next release of NumbaPro will provide aliases to the features that are moved to Numba and Accelerate. There will not be any new feature added to NumbaPro. In the future, there maybe bug fix releases for maintaining the aliases to the moved features.

Changes:

- Depends on numba 0.21.0
- Fix auto thread-per-block tuning support for CUDA CC 3.7 devices
- `Blas.dotu` is deprecated. A warning is generated when it is used. `Blas.dot` is an alias to it and is preferred.

Version (0.20.0)

This release depends on numba 0.20, which has upgraded to CUDA 7 for GPU support. CUDA 7 has deprecated the support for all 32-bit platforms. The oldest supported Windows version is Windows 7.0. This does not affect CPU features.

Version (0.19.0)

- Depends on numba 0.19
- Fixes issue with GPU ufunc broadcasting
- Improves GPU ufunc implementation

Version (0.18.0)

- Depends on numba 0.18.1
- **Improve CUDA gufunc implementation**
  - Simplified code generation
  - Smarter blocksize selection

Version (0.17.1)

- Depends on numba 0.17.0
- Warns about incompatible numba version at import time
- Fixes some CUDA library APIs on windows

Version (0.17.0)

- Depends on numba 0.16.0
- Replaces llvmpy with llvmlite, which also upgrades to llvm3.5
- Update occupancy autotuner for CC 5.0 and CC 5.2 devices
- Fix handling of empty array in GPU reduction
- Fix occupancy autotuner that may pick invalid blocksize

Version (0.16.0)

- Add *numbapro.cuda.reduce* for autogeneration of CUDA reduce kernels and driver.
- Fix device to host auto transfer logic in some ufunc function.
- Upgrades to Numba 0.15

Version (0.15.0)

- **Add *numbapro.cudalib.sorting*:**
  - Added GPU radixsort and radixselect using implementation from http://nvlabs.github.io/cub
  - Added GPU segmentedsort from http://nvlabs.github.io/moderngpu
- Fix GPU *print()* when there are multiple arguments

Version (0.14.3)

- CUDA driver is initialized lazily
- Improved stability of CUDA ufunc machinery
- Improved stability of parallel ufunc

Version (0.14.2)

- Unify numba.cuda and numbapro.cuda backend
- Enable Python 3 support
- Fixes workqueue module import for embedded python usecase

Version (0.14.1)

Fixes:

- UnboundReferenceError due to mishandling of incompatible driver (pre CUDA5.5 driver). The fix relaxes the driver requirement by allowing some features to fail on use.
- numbapro.cuda.* symbols are still exported when CUDA is not available. They would raise execption on use.

Version (0.14.0)

Features:

- Add cuSparse API
- Improve CUDA driver and resource management
- Some of CUDA-python language feature is now opensourced as numba.cuda

Fixes:

- New CUDA driver system prevents freezing OSX on kernel launch error

Version (0.13.2)

- Fix problem with numpy 18 array scalar contiguousness
- Fix CUDA target auto initialization on *import numbapro*
- Fix an access violation error on Windows 8 due to mishandling by LLVM.
- Add non-public API for profiler control.

Version (0.13.1)

- Guard error due to mishandling of interleaved memory buffer (#60)
- Update to use Numba 0.12.1
- Fix powi bug

Version (0.13)

- Add print statement for strings and scalar numeric types for debugging on GPU
- Add constant and local memory array allocation on GPU
- Add debug mode for GPU
- Allow raising exception classes on GPU
- Update CUDA toolkit libraries
- Fix boolean mapping

Version (0.12.7)

- Fix major bug that mistreats py2 division as inplace floor-division for real numbers.
- Fix using of array as argument of a CUDA device function.
- Delay initialization the CUDA subsystem upon first import of the cuda package.
- Add docstrings.

Version (0.12.6)

- Fix major bug that mistreats py2 division as floor-division for real numbers.

Version (0.12.5)

- Update to Numba 0.10.2
- Update to LLVM 3.3
- Various bug fixes

Version (0.12.4)

- Update to Numba 0.10.0
- Minor bug fixes

Version (0.12.3)

- Accept older driver by defering driver error to first use of specific API
- Report incompatible GPU at context creation
- Improve device information reporting
- Autotuning base on compiler info and occupancy calculator
- Add basic support for ravel and reshape

Version (0.12.2)

- Distribute CUDA toolkit in Anaconda
- Better error message
- Fix gufunc signature parsing to accept trailing comma.
- Fix CUDA driver log info bug
- Support JIT linking

Version (0.12.1)

- Fix libNVVM search path (now accept directory path)
- Fix sign-extension error in forloop precondition
- Fix support for true-division

Version (0.12.0)

- Use CUDA 5.5rc
- Expand math support through CUDA NVVM libdevice
- Rewritten nopython mode for CUDA-Python
- Removed experimental CU API
- Removed minivectorize

Version (0.11.0)

- Add cuBlas binding
- Improve CUDA ndarray and memory managment
- Add CUDA mapped host memory
- Add CUDA event

Version (0.10.1)

- Fix CU memory leak
- Fix CU hanging on some GPU
- Improve error message for unsupported GPU devices
- Add cuFFT

Version (0.10)

- Added Compute Unit (CU) API
- Added cuRAND binding
- Added CUDA device array
- Various improvements to CUDA support

Version (0.9)

- Improve CUDA driver discovery.

Version 0.8

- Update for SSA types inference in Numba
- Allow user to select CUDA device
- Add support for pinned and mapped CUDA memory
- Improvement on small memory allocation in CUDA
- Default to use libNVVM from Anaconda

- Bug fixes

Version 0.7

- Prange: parallel for-range
- Array slicing
- Refactor CUDA dispatch mechanisms
- Migrate to NVVM instead of PTX for CUDA codegen

Version 0.6 and earlier

- Array expressions
- Fast ufuncs and generalized-ufunc (gufunc) with single-core, multi-core and CUDA
- CUDA JIT.

License Agreement

NumbaPro EULA

END USER LICENSE AGREEMENT

NumbaPro (“the Software Product”) and accompanying documentation is licensed and not sold. The Software Product is protected by copyright laws and treaties, as well as laws and treaties related to other forms of intellectual property. Continuum Analytics, Inc. (dba Anaconda, Inc.) or its subsidiaries, affiliates, and suppliers (collectively “Continuum”) own intellectual property rights in the Software Product. The Licensee’s (“you” or “your”) license to download, use, copy, or change the Software Product is subject to these rights and to all the terms and conditions of this End User License Agreement (“Agreement”).

Acceptance

YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY SELECTING THE “ACCEPT” OPTION AND DOWNLOADING THE SOFTWARE PRODUCT OR BY INSTALLING, USING, OR COPYING THE SOFTWARE PRODUCT. YOU MUST AGREE TO ALL OF THE TERMS OF THIS AGREEMENT BEFORE YOU WILL BE ALLOWED TO DOWNLOAD THE SOFTWARE PRODUCT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST SELECT “DECLINE” AND YOU MUST NOT INSTALL, USE, OR COPY THE SOFTWARE PRODUCT.

Trial Period

You have the right to use NumbaPro on a single computer or group of computers for 30 days with no license. After 30 days you must purchase an appropriate license to use the software on one or more machines or stop using the software and remove it from all of your machines on which you installed the software.

License Grant

This Agreement entitles you to install and use one copy of the Software Product on as many machines as you will personally use. The Software Product is licensed to a particular user. Only the user to whom the software is licensed may use the software. You must obtain a license for as many users as you wish to use the software. In addition, you may make archival copies of the Software Product Installer.

Restrictions on Transfer

Without first obtaining the express written consent of Continuum, you may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software Product.

Restrictions on Use

You may not use, copy, or install the Software Product on any system where more than one user will be able to use the software unless you have purchased a license for each user of the system. You may not decompile, "reverse-engineer", disassemble, or otherwise attempt to derive the source code for the Software Product.

Restrictions on Alteration

You may not modify the Software Product or create any derivative work of the Software Product or its accompanying documentation. Derivative works include but are not limited to translations. You may not alter any files or libraries in any portion of the Software Product.

Restrictions on Copying

You may not copy any part of the Software Product except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make archival copies of the Software Product installer.

Limited Software Product Warranty

For a period of 60 days from the date of shipment or from the date that you download the Software Product, as applicable, Continuum warrants that when properly installed and used under normal conditions, the Software Product will perform substantially as advertised.

Disclaimer of Warranties and Limitation of Liability

UNLESS OTHERWISE EXPLICITLY AGREED TO IN WRITING BY CONTINUUM, CONTINUUM MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE PRODUCT.

Continuum makes no warranty that the Software Product will meet your requirements or operate under your specific conditions of use. Continuum makes no warranty that operation of the Software Product will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE PRODUCT SUFFICIENTLY

MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE PRODUCT TO MEET YOUR REQUIREMENTS. CONTINUUM WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE. UNDER NO CIRCUMSTANCES SHALL CONTINUUM, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE PRODUCT, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF CONTINUUM OR ANY OTHER PARTY, EVEN IF CONTINUUM IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS CONTINUUM'S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

Limitation of Remedies and Damages

Your remedy for a breach of this Agreement or of any warranty included in this Agreement is the correction or replacement of the Software Product. Selection of whether to correct or replace shall be solely at the discretion of Continuum. Continuum reserves the right to substitute a functionally equivalent copy of the Software Product as a replacement. If Continuum is unable to provide a replacement or substitute Software Product or corrections to the Software Product, your sole alternate remedy shall be a refund of the purchase price for the Software Product exclusive of any costs for shipping and handling. Any claim must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by Continuum to have been caused by you. All limited warranties on the Software Product are granted only to you and are non-transferable. You agree to indemnify and hold Continuum harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

Governing Law, Jurisdiction and Costs

This Agreement is governed by the laws of Texas, without regard to Texas's conflict or choice of law provisions.

Severability

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

Third Party Dependencies

The following sections enumerate NumbaPro thirdparty dependencies and their license information.

CUDA Toolkit Redistributable Libraries

The `cudatoolkit` conda package contains redistributable binaries from the CUDA toolkit. See CUDA toolkit EULA at http://docs.nvidia.com/cuda/eula/index.html

CUB

The `nbpro_radixsort.so` library (shipped in `numbapro_cudalib` conda package) uses CUB (http://nvlabs.github.io/cub) which has the following license:

```
Copyright (c) 2011, Duane Merrill.  All rights reserved.
Copyright (c) 2011-2014, NVIDIA CORPORATION.  All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
   Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.
   Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
   Neither the name of the NVIDIA CORPORATION nor the
      names of its contributors may be used to endorse or promote products
      derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL NVIDIA CORPORATION BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

ModernGPU

The `nbpro_segsort.so` library (shipped in `numbapro_cudalib` conda package) uses ModernGPU (http://nvlabs.github.io/moderngpu) which has the following license:

```
Copyright (c) 2013, NVIDIA CORPORATION.  All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
    * Neither the name of the NVIDIA CORPORATION nor the
      names of its contributors may be used to endorse or promote products
      derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL NVIDIA CORPORATION BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

MKL Optimizations

Developed specifically for science, engineering, and financial computations, Intel™ Math Kernel Library (MKL) is a set of threaded and vectorized math routines that work to accelerate various math functions and applications. Anaconda has packaged MKL-powered binary versions of some of the most popular numerical/scientific Python libraries into MKL Optimizations for improved performance.

MKL Optimizations includes:

- Speed-boosted NumPy, SciPy, scikit-learn, and NumExpr
- The packaging of MKL with redistributable binaries in Anaconda for easy access to the MKL runtime library.
- Python bindings to the low level MKL service functions, which allow for the modification of the number of threads being used during runtime.

Included Packages

Packages upgraded to use MKL

- NumPy
- NumExpr
- SciPy
- Scikit-Learn

mkl-service

This package exposes a few functions which are declared in mkl_service.h. The main purpose of the package is to allow the user to change the number of CPU's MKL is using at runtime.

Example:

In this small example, we first query how many CPU are being used, and then change this number.

```
>>> import mkl
>>> mkl.get_max_threads()
2
>>> mkl.set_num_threads(1)
>>> mkl.get_max_threads()
1
```

Reference

mkl service functions:

`mkl.get_cpu_clocks()` -> int Return the CPU clocks as an integer.

`mkl.get_cpu_frequency()` -> float Return CPU frequency in GHz as a float.

`mkl.get_max_threads()` -> int Return the number of threads Intel MKL is targeting for parallelism.

`mkl.get_version_string()` -> str Return the MKL library version information as a string.

mkl.mem_stat(n) -> int Returns an amount of memory, allocated by the MKL Memory Allocator.

mkl.set_num_threads(n) Set the number of threads MKL should use. This is only a hint, and no guaranteed is made this number of threads will actually be used. This function takes precedence over the environment variable MKL_NUM_THREADS.

Installation

In Anaconda versions 2.5 and later, MKL is freely available by default in Anaconda.

If you already have the free Anaconda Python distribution installed, get MKL by upgrading to the latest version:

```
conda update conda
conda update anaconda
```

If you do not have Anaconda installed, you can download it here.

Existing MKL licenses can be viewed and removed with the graphical Anaconda Navigator license manager or manually with your operating system. For more information please see the *License installation* page.

Configuration

Anaconda now also includes a small utility package called mkl-service which provides a Python interface to some useful MKL functions that are declared in `mkl_service.h`, such as setting the number of threads to use.

Uninstalling MKL

MKL takes roughly 100MB and some use cases do not need it, so users can opt out of MKL and instead use OpenBLAS for Linux or the native Accelerate Framework for MacOSX. To opt out, run `conda install nomkl` and then use `conda install` to install packages that would normally include MKL or depend on packages that include MKL, such as `scipy`, `numpy`, and `pandas`. Conda will install the non-MKL versions of these packages together with their dependencies. If you are using OS X or Linux, have already installed these packages or already installed all of Anaconda, and wish to switch away from MKL, use the command `conda install nomkl numpy scipy scikit-learn numexpr` followed by `conda remove mkl mkl-service`.

Updating

If you already have the free Anaconda Python distribution installed and wish to update MKL:

```
conda update conda
conda update mkl
```

Dismissing MKL Trial warnings

Because past versions of Anaconda did not include MKL linked binaries by default, some users who have used `conda update --all` may see an MKL Trial warning or a license expiration error, even though MKL linked packages are now free and installed by default. A license expiration error message may read, “You cannot run MKL without a license any longer.”

To resolve this, set your installation to use the mkl-linked libraries that do not require a license:

```
conda remove mkl-rt
conda install -f mkl
```

Then run conda install with the specific packages you choose:

```
conda install numpy scipy scikit-learn numexpr
```

or with all of Anaconda:

```
conda install anaconda
```

Resolving MKL shared library "permission denied" errors

On Linux platforms that have SELinux enabled, you may encounter security errors like the following:

```
error while loading shared libraries: <libmkl_intel.so>: cannot restore segment prot␣
↪after reloc: Permission denied
```

This is because MKL requires text relocation permissions, which SELinux denies by default. This prevents MKL from being loaded by Numpy, preventing Numpy from being imported.

There are two known solutions to this issue:

1. Does not require root privileges. Replace MKL with OpenBLAS by issuing the command:

   ```
   conda install nomkl numpy scipy scikit-learn numexpr
   ```

   You may revert back to the MKL default versions at any time by using:

   ```
   conda remove nomkl
   conda install mkl
   ```

2. Requires root privileges. Temporarily disable SELinux enforcement. From a root-privileged terminal enter:

   ```
   /usr/sbin/setenforce 0
   ```

   NOTE: If you prefer to make this change permanent, in the file `/etc/selinux/config` change "enforcing" to "disabled" and then reboot.

License Agreement

The MKL Optimizations are included in the free Anaconda python distribution, and have been made available by the terms of the *Anaconda End User License Agreement*.

Past versions of MKL Optimizations were not freely available in Anaconda, and have been made available separately by the terms of the *MKL Optimizations End User License Agreement*.

MKL Optimizations End User License Agreement

MKL Optimizations ("the Software Product" or "MKL Optimizations") and accompanying documentation is licensed and not sold. The Software Product is protected by copyright laws and treaties, as well as laws and treaties related to other forms of intellectual property. Continuum Analytics, Inc. (dba Anaconda, Inc.) or its subsidiaries, affiliates, and suppliers (collectively "Continuum") own intellectual property rights in the Software Product. The Licensee's ("you" or “your”) license to download, use, copy, or change the Software Product is subject to these rights and to all the terms and conditions of this End User License Agreement (“Agreement”).

Acceptance

YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY SELECTING THE “ACCEPT” OPTION AND DOWNLOADING THE SOFTWARE PRODUCT OR BY INSTALLING, USING, OR COPYING THE SOFTWARE PRODUCT. YOU MUST AGREE TO ALL OF THE TERMS OF THIS AGREEMENT BEFORE YOU WILL BE ALLOWED TO DOWNLOAD THE SOFTWARE PRODUCT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST SELECT “DECLINE” AND YOU MUST NOT INSTALL, USE, OR COPY THE SOFTWARE PRODUCT.

Other Vendor Tools

In addition to Continuum’s proprietary software, the Software product may contain other software packages from other sources (“Other Vendor Tools”). These are available on an “as is” basis and subject to their individual license agreements. These licenses are available either in the package distributed with the Software Product or via request to info@anaconda.com. Any binary packages of Other Vendor Tools you obtain via MKL Optimizations are subject to their individual licenses as well as the terms of this EULA. Continuum reserves the right to change which Other Vendor Tools are provided in MKL Optimizations.

Trial Period

You have the right to use MKL Optimizations on a single computer or group of computers for 30 days with no license. After 30 days you must purchase an appropriate license to use the software on one or more machines or stop using the software and remove it from all of your machines on which you installed the software.

License Grant

This Agreement entitles you to install and use one copy of the Software Product on as many machines as you will personally use. The Software Product is licensed to a particular user. Only the user to whom the software is licensed may use the software. You must obtain a license for as many users as you wish to use the software. In addition, you may make archival copies of the Software Product Installer.

Right to Updates

This license entitles you to updates to the Software Product for one year from the time of payment. The Software Product will continue to function and you may continue to use The Software Product and any updates you have received for as long as you would like however you will no longer be able to receive updates from Continuum unless this License is renewed. Please contact sales@anaconda.com with any questions or concerns.

Restrictions on Transfer

Without first obtaining the express written consent of Continuum, you may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software Product.

Restrictions on Use

You may not use, copy, or install the Software Product on any system where more than one user will be able to use the software unless you have purchased a license for each user of the system. You may not decompile, “reverse-engineer”, disassemble, or otherwise attempt to derive the source code for the Software Product.

Restrictions on Alteration

You may not modify the Software Product or create any derivative work of the Software Product or its accompanying documentation. Derivative works include but are not limited to translations. You may not alter any files or libraries in any portion of the Software Product.

Restrictions on Copying

You may not copy any part of the Software Product except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make archival copies of the Software Product installer.

Limited Software Product Warranty

For a period of 60 days from the date of shipment or from the date that you download the Software Product, as applicable, Continuum warrants that when properly installed and used under normal conditions, the Software Product will perform substantially as advertised.

Disclaimer of Warranties and Limitation of Liability

UNLESS OTHERWISE EXPLICITLY AGREED TO IN WRITING BY CONTINUUM, CONTINUUM MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE PRODUCT.

Continuum makes no warranty that the Software Product will meet your requirements or operate under your specific conditions of use. Continuum makes no warranty that operation of the Software Product will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE PRODUCT SUFFICIENTLY MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE PRODUCT TO MEET YOUR REQUIREMENTS. CONTINUUM WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE. UNDER NO CIRCUMSTANCES SHALL CONTINUUM, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE PRODUCT, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF CONTINUUM OR ANY OTHER PARTY, EVEN IF CONTINUUM IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS CONTINUUM’S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

Limitation of Remedies and Damages

Your remedy for a breach of this Agreement or of any warranty included in this Agreement is the correction or replacement of the Software Product. Selection of whether to correct or replace shall be solely at the discretion of Continuum. Continuum reserves the right to substitute a functionally equivalent copy of the Software Product as a replacement. If Continuum is unable to provide a replacement or substitute Software Product or corrections to the Software Product, your sole alternate remedy shall be a refund of the purchase price for the Software Product exclusive of any costs for shipping and handling. Any claim must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by Continuum to have been caused by you. All limited warranties on the Software Product are granted only to you and are non-transferable. You agree to indemnify and hold Continuum harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

Governing Law, Jurisdiction and Costs

This Agreement is governed by the laws of Texas, without regard to Texas's conflict or choice of law provisions.

Export Regulations

Any use or distribution of MKL Optimizations is made under conditions that the user and/or distributor is in full compliance with all export and other governing laws of the United States of America, including full and ongoing compliance with the Export Administration Regulations (EAR) of the United States Department of Commerce. See www.commerce.gov/ and http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear. Use or distribution of Continuum software products to any persons, entities or countries currently under US sanctions is strictly prohibited. MKL Optimizations is classified with an ECCN of 5D992 with no license required for export to non-embargoed countires.

The United States currently has embargoes against Cuba, Iran, North Korea, Sudan and Syria. The exportation, re-exportation, sale or supply, directly or indirectly, from the United States, or by a U.S. person wherever located, of any Continuum software to any of these countries is strictly prohibited without prior authorization by the United States Government By accepting this Agreement, you represent to Continuum that you will comply with all applicable export regulations for MKL Optimizations.

Severability

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

IOPro

IOPro loads NumPy arrays (and Pandas DataFrames) directly from files, SQL databases, and NoSQL stores–including ones with millions of rows–without creating millions of temporary, intermediate Python objects, or requiring expensive array resizing operations.

IOPro provides a drop-in replacement for the NumPy functions `loadtxt()` and `genfromtxt()`, but dramatically improves performance and reduces memory overhead.

The current (and final) version of IOPro 1.9.1 was released on March 1, 2017.

The open source successors to IOPro

All of the code in IOPro was released in 2017 under a BSD open source license. As part of that release, the code was refactored into its several component libraries, and version numbers of each were set to 2.0.0 to indicate they supersede the code under the IOPro name.

The several libraries are available on GitHub at:

- TextAdapter (CSV, JSON, etc)
- DBAdapter (SQL derivatives)
- PostgresAdapter (PostgreSQL)
- AccumuloAdapter (Apache Accumulo)
- MongoAdapter (MongoDB)

These various packages are or will be available as conda packages (using lowercase versions of their names), mostly in the conda-forge channel. Maintenance, improvements, and packaging will be a community-led project.

How to get IOPro

IOPro is included with Anaconda Workgroup and Anaconda Enterprise subscriptions.

To start a 30-day free trial just download and install the IOPro package.

If you already have Anaconda (free Python platform) or Miniconda installed:

```
conda update conda
conda install -c anaconda-extras iopro=1.9.1
```

If you do not have Anaconda installed, you can download it.

For more information about IOPro please contact sales@anaconda.com.

Requirements

- Python 2.7 or 3.4+
- NumPy 1.10+

Optional Python modules:

- Boto (for S3 support)
- Pandas (to use DataFrames)

What’s new in version 1.9?

The documentation has been substantially updated for version 1.9.0. Numba has been removed and the code has been cleaned up, but no other features were added or removed. Some refactoring was done that didn’t change functionality. We recommend that users not use older versions. See *Release notes* for additional detail.

Getting started

Some of the basic usage patterns look like these. Create TextAdapter object for data source:

```
>>> import iopro
>>> adapter = iopro.text_adapter('data.csv', parser='csv')
```

Define field dtypes (example: set field 0 to unsigned int and field 4 to float):

```
>>> adapter.set_field_types({0: 'u4', 4:'f4'})
```

Parse text and store records in NumPy array using slicing notation:

```
>>> # read all records
>>> array = adapter[:]

>>> # read first ten records
>>> array = adapter[0:10]

>>> # read last record
>>> array = adapter[-1]

>>> # read every other record
>>> array = adapter[::2]
```

User guide

Installation

If you do not already have Anaconda installed, please download it via the downloads page and install it.

IOPro is included with Anaconda Workgroup and Anaconda Enterprise subscriptions.

To start a 30-day free trial just download and install the IOPro package.

If you already have Anaconda (free Python platform) or *Miniconda <http://conda.io/miniconda.html>* installed:

```
conda update conda
conda install iopro
```

If you do not have Anaconda installed, you can download it.

For more information about IOPro please contact sales@anaconda.com.

IOPro Update Instructions

If you have Anaconda (free Python platform) installed, first update the conda package management tool to the latest version, then use conda to update the IOPro product installation:

```
conda update conda
conda update iopro
```

Uninstall

To uninstall using conda:

```
conda remove iopro
```

Installing license

The IOPro license can be installed with the graphical Anaconda Navigator license manager or manually with your operating system. In your organization this may be handled by your site administrator or IT department. Both installation methods are explained in the *License installation* page.

TextAdapter First Steps

Basic Usage

IOPro works by attaching to a data source, such as a local CSV file. Before we get started, let's create a sample CSV file to work with:

```
from random import random, randint, shuffle
import string

NUMROWS = 10
with open('data/table.csv','w') as data:
    # Header
    for n in range(1,5):
        print("f%d" % n, end=",", file=data)
    print("comment", file=data)

    # Body
    letters = list(string.ascii_letters)
    for n in range(NUMROWS):
        shuffle(letters)
        s = "".join(letters[:randint(5,20)])
        vals = (n, randint(1000,2000), random(), random()*100, s)
        print("%d,%d,%f,%f,%s" % vals, file=data)
```

Let's read in the local CSV file created here. Because this small file easily fits in memory it would work to use the `csv` or `pandas` modules, but we will demonstrate the interfaces and capabilities that will apply to much larger data.

```
>>> import iopro
>>> adapter = iopro.text_adapter('data/table.csv', parser='csv')
>>> adapter.get_field_names()
['f1', 'f2', 'f3', 'f4', 'comment']
```

We can specify the data types for values in the columns of the CSV file being read; but first we look at the ability of IOPro's TextAdapter to auto-discover the data types used.

We can ask IOPro's TextAdapter to parse text and return records in NumPy arrays from selected portions of the CSV file using slicing notation:

```
>>> # the inferred datatypes
>>> array = adapter[:]
```

(continues on next page)

(continued from previous page)

```
>>> array.dtype
dtype([('f1', '<u8'), ('f2', '<u8'), ('f3', '<f8'), ('f4', '<f8'),
       ('comment', 'O')])
```

Let's define field dtypes (example: set field 0 to a 16-bit unsigned int and field 3 to a 32-bit float).

Massage the datatypes:

```
>>> adapter.set_field_types({0: 'u2', 3:'f4'})
>>> array = adapter[:]
>>> array.dtype
dtype([('f1', '<u2'), ('f2', '<u8'), ('f3', '<f8'), ('f4', '<f4'),
       ('comment', 'O')])
```

The first five records:

```
>>> array = adapter[0:5]
>>> print(array)
[(0, 1222, 0.926116, 84.44437408447266, 'MlzvBRyquns')
 (1, 1350, 0.553585, 81.03726959228516, 'ikgEauJeTZvd')
 (2, 1932, 0.710919, 31.59865951538086, 'uUQmHJFZhnirecAvx')
 (3, 1494, 0.622391, 57.90607452392578, 'iWQBAZodkfHODtI')
 (4, 1981, 0.820246, 40.848018646240234, 'igxeXdBpqE')]
```

Read last five records:

```
>>> array = adapter[-5:]
>>> print(array)
[(5, 1267, 0.694631, 6.999039173126221, 'bRSrwitHeY')
 (6, 1166, 0.37465, 38.7022705078125, 'qzbMgVThXtHpfDNrd')
 (7, 1229, 0.390566, 55.338134765625, 'hyarmvWi')
 (8, 1816, 0.201106, 59.74718475341797, 'DcHymelRusO')
 (9, 1416, 0.725697, 42.50992965698242, 'QMUGRAwe')]
```

Read every other record:

```
>>> array = adapter[::2]
>>> print(array)
[(0, 1222, 0.926116, 84.44437408447266, 'MlzvBRyquns')
 (2, 1932, 0.710919, 31.59865951538086, 'uUQmHJFZhnirecAvx')
 (4, 1981, 0.820246, 40.848018646240234, 'igxeXdBpqE')
 (6, 1166, 0.37465, 38.7022705078125, 'qzbMgVThXtHpfDNrd')
 (8, 1816, 0.201106, 59.74718475341797, 'DcHymelRusO')]
```

Read first and second, third fields only:

```
>>> array = adapter[[0,1,2]][:]
>>> list(array)
[(0, 1222, 0.926116),
 (1, 1350, 0.553585),
 (2, 1932, 0.710919),
 (3, 1494, 0.622391),
 (4, 1981, 0.820246),
 (5, 1267, 0.694631),
 (6, 1166, 0.37465),
 (7, 1229, 0.390566),
```

(continues on next page)

(continued from previous page)

```
 (8, 1816, 0.201106),
 (9, 1416, 0.725697)]
```

Read fields named ‘f2’ and ‘comment’ only:

```
>>> array = adapter[['f2','comment']][:]
>>> list(array)
[(1222, 'MlzvBRyquns'),
 (1350, 'ikgEauJeTZvd'),
 (1932, 'uUQmHJFZhnirecAvx'),
 (1494, 'iWQBAZodkfHODtI'),
 (1981, 'igxeXdBpqE'),
 (1267, 'bRSrwitHeY'),
 (1166, 'qzbMgVThXtHpfDNrd'),
 (1229, 'hyarmvWi'),
 (1816, 'DcHymelRusO'),
 (1416, 'QMUGRAwe')]
```

JSON Support

Text data in JSON format can be parsed by specifying ‘json’ for the parser argument:

Content of file `data/one.json`:

```
{"id":123, "name":"xxx"}
```

Single JSON object:

```
>>> adapter = iopro.text_adapter('data/one.json', parser='json')
>>> adapter[:]
array([(123, 'xxx')],
      dtype=[('id', '<u8'), ('name', 'O')])
```

Currently, each JSON object at the root level is interpreted as a single NumPy record. Each JSON object can be part of an array, or separated by a newline. Examples of valid JSON documents that can be parsed by IOPro, with the NumPy array result:

Content of file `data/two.json`:

```
[{"id":123, "name":"xxx"}, {"id":456, "name":"yyy"}]
```

Array of two JSON objects:

```
>>> iopro.text_adapter('data/two.json', parser='json')[:]
array([(123, 'xxx'), (456, 'yyy')],
      dtype=[('id', '<u8'), ('name', 'O')])
```

Content of file `data/three.json`:

```
{"id":123, "name":"xxx"}
{"id":456, "name":"yyy"}
```

Two JSON objects separated by newline:

```
>>> iopro.text_adapter('data/three.json', parser='json')[:]
array([(123, 'xxx'), (456, 'yyy')],
      dtype=[('id', '<u8'), ('name', 'O')])
```

Massaging data in the adapter

A custom function can be used to modify values as they are read.

```
>>> import iopro, io, math
>>> stream = io.StringIO('3,abc,3.3\n7,xxx,9.9\n4,,')
>>> adapter = iopro.text_adapter(stream, parser='csv', field_names=False)
```

Override default converter for first field:

```
>>> adapter.set_converter(0, lambda x: math.factorial(int(x)))
>>> adapter[:]
array([(6, 'abc', 3.3), (5040, 'xxx', 9.9), (24, '', nan)],
      dtype=[('f0', '<u8'), ('f1', 'O'), ('f2', '<f8')])
```

We can also force data types and set fill values for missing data.

Apply data types to columns:

```
>>> stream = io.StringIO('3,abc,3.3\n7,xxx,9.9\n4,,')
>>> adapter = iopro.text_adapter(stream, parser='csv', field_names=False)
>>> adapter.set_field_types({1:'S3', 2:'f4'})
>>> adapter[:]
array([(3, b'abc', 3.299999952316284), (7, b'xxx', 9.899999618530273),
       (4, b'', nan)],
      dtype=[('f0', '<u8'), ('f1', 'S3'), ('f2', '<f4')])
```

Set fill value for missing values in each field:

```
>>> adapter.set_fill_values({1:'ZZZ', 2:999.999})
>>> adapter[:]
array([(3, b'abc', 3.299999952316284), (7, b'xxx', 9.899999618530273),
       (4, b'ZZZ', 999.9990234375)],
      dtype=[('f0', '<u8'), ('f1', 'S3'), ('f2', '<f4')])
```

Combining regular expressions and typecasting

A later section discusses regular expressions in more detail. This example is a quick peek into using them with IOPro.

Content of the file `data/transactions.csv`:

```
$2.56, 50%, September 20 1978
$1.23, 23%, April 5 1981
```

Combining features:

```
>>> import iopro
>>> regex_string = '\$(\d)\.(\d{2}),\s*([0-9]+)\%,\s*([A-Za-z]+)'
>>> adapter = iopro.text_adapter('data/transactions.csv',
...                              parser='regex',
```

(continues on next page)

(continued from previous page)

```
...                                     regex_string=regex_string,
...                                     field_names=False,
...                                     infer_types=False)
```

Set dtype of fields and their names:

```
>>> adapter.set_field_types({0:'i2', 1:'u2', 2:'f4', 3:'S10'})
>>> adapter.set_field_names(['dollars', 'cents', 'percentage', 'month'])
>>> adapter[:]
array([(2, 56, 50.0, b'September'), (1, 23, 23.0, b'April')],
       dtype=[('dollars', '<i2'), ('cents', '<u2'),
              ('percentage', '<f4'), ('month', 'S10')])
```

Advanced TextAdapter

`iopro.loadtext()` versus `iopro.genfromtxt()`

Within IOPro there are two closely related functions. `loadtext()`, which we have been looking at, makes a more optimistic assumption that your data is well-formatted. `genfromtxt()` has a number of arguments for handling messier data, and special behaviors for dealing with missing data.

`loadtext()` is already highly configurable for dealing with data under many CSV and other delimited formats. `genfromtxt()` contains a superset of these arguments.

Gzip Support

IOPro can decompress gzip'd data on the fly, simply by indicating a `compression` keyword argument.

```
>>> adapter = iopro.text_adapter('data.gz', parser='csv', compression='gzip')
>>> array = adapter[:]
```

As well as being able to store and work with your compressed data without having to decompress it first, you also do not need to sacrifice any performance in doing so. For example, with one test 419 MB CSV file of numerical data, and a 105 MB file of the same data compressed with gzip, the following are run times on a test machine for loading the entire contents of each file into a NumPy array. Exact performance will vary between machines, especially between machines with HDD and SSD architecture.:

```
-   uncompressed: 13.38 sec
-   gzip compressed: 14.54 sec
```

In the test, the compressed file takes slightly longer, but consider having to uncompress the file to disk before loading with IOPro:

- uncompressed: 13.38 sec
- gzip compressed: 14.54 sec
- gzip compressed (decompress to disk, then load): 21.56 sec

Indexing CSV Data

One of the most useful features of IOPro is the ability to index data to allow for fast random lookup.

For example, to retrieve the last record of the compressed 109 MB dataset we used above:

```
>>> adapter = iopro.text_adapter('data.gz', parser='csv', compression='gzip')
>>> array = adapter[-1]
```

Retrieving the last record into a NumPy array takes 14.82 sec. This is about the same as the time to read the entire array, because the entire dataset has to be parsed to get to the last record.

To make seeking faster, we can build an index:

```
>>> adapter.create_index('index_file')
```

The above method creates an index in memory and saves it to disk, taking 9.48 sec. Now when seeking to and reading the last record again, it takes a mere 0.02 sec.

Reloading the index only takes 0.18 sec. If you build an index once, you get near instant random access to your data forever (assuming the data remains static):

```
>>> adapter = iopro.text_adapter('data.gz', parser='csv',
...                              compression='gzip',
...                              index_name='index_file')
```

Let's try it with a moderate sized example. You can download this data from the Exoplanets Data Explorer site.

```
>>> adapter = iopro.text_adapter('data/exoplanets.csv.gz',
...                              parser='csv', compression='gzip')
>>> print(len(adapter[:]), "rows")
>>> print(', '.join(adapter.field_names[:3]),
...       '...%d more...\n    ' % (adapter.field_count-6),
...       ', '.join(adapter.field_names[-3:]))
2042 rows
name, mass, mass_error_min ...73 more...
    star_teff, star_detected_disc, star_magnetic_field
```

```
>>> adapter.field_types
{0: dtype('O'),
 1: dtype('float64'),
 2: dtype('float64'),
 3: dtype('O'),
 4: dtype('float64'),
 5: dtype('float64'),
 6: dtype('float64'),
 7: dtype('float64'),
 8: dtype('O'),
 9: dtype('float64'),
 [... more fields ...]
 69: dtype('float64'),
 70: dtype('float64'),
 71: dtype('float64'),
 72: dtype('float64'),
 73: dtype('float64'),
 74: dtype('O'),
 75: dtype('float64'),
 76: dtype('float64'),
 77: dtype('O'),
 78: dtype('uint64')}
```

Do some timing (using an IPython magic):

```
>>> %time row = adapter[-1]
CPU times: user 35 ms, sys: 471 µs, total: 35.5 ms
Wall time: 35.5 ms
```

```
>>> %time adapter.create_index('data/exoplanets.index')
CPU times: user 15.7 ms, sys: 3.35 ms, total: 19.1 ms
Wall time: 18.6 ms
```

```
>>> %time row = adapter[-1]
CPU times: user 18.3 ms, sys: 1.96 ms, total: 20.3 ms
Wall time: 20.1 ms
```

```
>>> new_adapter = iopro.text_adapter('data/exoplanets.csv.gz', parser='csv',
...                                  compression='gzip',
...                                  index_name='data/exoplanets.index')
```

```
>>> %time row = new_adapter[-1]
CPU times: user 17.3 ms, sys: 2.12 ms, total: 19.4 ms
Wall time: 19.4 ms
```

Regular Expressions

> Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems. –Jamie Zawinski

IOPro supports using regular expressions to help parse messy data. Take for example the following snippet of actual NASDAQ stock data found on the Internet:

The content of the file `data/stocks.csv`:

```
Name,Symbol,Exchange,Range
Apple,AAPL,NasdaqNM,363.32 - 705.07
Google,GOOG,NasdaqNM,523.20 - 774.38
Microsoft,MSFT,NasdaqNM,24.30 - 32.95
```

The first three fields are easy enough: name, symbol, and exchange. The fourth field presents a bit of a problem. Let's try IOPro's regular expression based parser:

```
>>> regex_string = '([A-Za-z]+),([A-Z]{1,4}),([A-Za-z]+),'\
...                '(\d+.\.\d{2})\s*\-\s*(\d+.\.\d{2})'
>>> adapter = iopro.text_adapter('data/stocks.csv', parser='regex',
...                              regex_string=regex_string)
```

```
>>> # Notice that header does not now match the regex
>>> print(adapter.field_names)
['Name,Symbol,Exchange,Range', '', '', '', '']
```

```
>>> # We can massage the headers to reflect our match pattern
>>> info = adapter.field_names[0].split(',')[:3]
>>> adapter.field_names =  info + ["Low", "High"]
>>> adapter[:]
array([('Apple', 'AAPL', 'NasdaqNM', 363.32, 705.07),
       ('Google', 'GOOG', 'NasdaqNM', 523.2, 774.38),
```

(continues on next page)

(continued from previous page)

```
       ('Microsoft', 'MSFT', 'NasdaqNM', 24.3, 32.95)],
       dtype=[('Name', 'O'), ('Symbol', 'O'),
              ('Exchange', 'O'), ('Low', '<f8'), ('High', '<f8')])
```

Regular expressions are compact and often difficult to read, but they are also very powerful. By using the above regular expression with the grouping operators '(' and ')', we can define exactly how each record should be parsed into fields. Let's break it down into individual fields:

- `([A-Za-z]+)` defines the first field (stock name) in our output array
- `([A-Z]{1-4})` defines the second (stock symbol)
- `([A-Za-z]+)` defines the third (exchange name)
- `(\d+.\.\d{2})` defines the fourth field (low price)
- `\s*\-\s*` is skipped because it is not part of a group
- `(\d+.\.\d{2})` defines the fifth field (high price)

The output array contains five fields: three string fields and two float fields. Exactly what we want.

S3 Support

IOPro can parse CSV data stored in Amazon's S3 cloud storage service. In order to access S3 files, you need to specify some credentials along with the resource you are accessing.

The first two parameters are your AWS access key and secret key, followed by the S3 bucket name and key name. The S3 CSV data is downloaded in 128K chunks and parsed directly from memory, bypassing the need to save the entire S3 data set to local disk.

Let's take a look at what we have stored from the Health Insurance Marketplace data. There's a little bit of code with BeautifulSoup just to prettify the raw XML query results.

```
>>> import urllib.request
>>> url = 'http://s3.amazonaws.com/product-training/'
>>> xml = urllib.request.urlopen(url).read()
```

```
>>> import bs4, re
>>> r = re.compile(r'^(\s*)', re.MULTILINE)
>>> def display(bs, encoding=None, formatter="minimal", indent=4):
...     print(r.sub(r'\1' * indent, bs.prettify(encoding, formatter)))
>>> display(bs4.BeautifulSoup(xml, "xml"))
<?xml version="1.0" encoding="utf-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Name>
        product-training
    </Name>
    <Prefix/>
    <Marker/>
    <MaxKeys>
        1000
    </MaxKeys>
    <IsTruncated>
        false
    </IsTruncated>
    <Contents>
```

(continues on next page)

(continued from previous page)

```
        <Key>
            BusinessRules.csv
        </Key>
        <LastModified>
            2016-06-25T00:03:20.000Z
        </LastModified>
        <ETag>
            "a565ebede6a7e6e060cd4526a7ae4345"
        </ETag>
        <Size>
            8262590
        </Size>
        <StorageClass>
            STANDARD
        </StorageClass>
    </Contents>
    <Contents>
        [... more files ...]
    </Contents>
</ListBucketResult>
```

In simple form, we see details about some S3 resources. Let's access one of them. Note that you will need to fill in your actual AWS access key and secret key.

```
>>> user_name = "class1"
>>> aws_access_key = "ABCD"
>>> aws_secret_key = "EFGH/IJK"
>>> bucket = 'product-training'
>>> key_name = 'BusinessRules.csv' # 21k lines, 8MB
>>> # key_name = 'PlanAttributes.csv' # 77k lines, 95MB
>>> # key_name = 'Rate.csv.gzip' # 13M lines, 2GB raw, 110MB compressed
>>> adapter = iopro.s3_text_adapter(aws_access_key, aws_secret_key,
...                                 bucket, key_name)
>>> # Don't try with the really large datasets, works with the default one
>>> df = adapter.to_dataframe()
>>> df.iloc[:6,:6]
```

IOPro can also build an index for S3 data just as with disk based CSV data, and use the index for fast random access lookup. If an index file is created with IOPro and stored with the S3 dataset in the cloud, IOPro can use this remote index to download and parse just the subset of records requested. This allows you to generate an index file once and share it on the cloud along with the data set, and does not require others to download the entire index file to use it.

iopro.pyodbc First Steps

iopro.pyodbc extends pyodbc with methods that allow data to be fetched directly into numpy containers. These functions are faster than regular fetch calls in pyodbc, providing also the convenience of being returned in a container appropriate to fast analysis.

This notebook is intended to be a tutorial on iopro.pyodbc. Most of the material is applicable to pyodbc (and based on pyodbc tutorials). There will be some examples specific to iopro.pyodbc. When that's the case, it will be noted.

Concepts

In pyodbc there are two main classes to understand:

- connection
- cursor

A connection is, as its name says, a connection to a datasource. A datasource is your database. It may be a database handled by a DBMS or just a plain file. A cursor allows you to interface with statements. Interaction with queries and other commands is performed through a cursor. A cursor is associated to a connection and commands over a cursor are performed over that connection to the datasource. In order to use iopro.pyodbc you must import it:

```
>>> import iopro.pyodbc as pyodbc
```

Connection to a datasource

In order to operate with pyodbc you need to connect to a datasource. Typically this will be a database. This is done by creating a connection object. To create a connection object you need a connection string. This string describes the datasource to use as well as some extra parameters. You can learn more about connection strings here.:

```
>>> connection_string = '''DSN=SQLServerTest;DATABASE=Test'''
>>> connection = pyodbc.connect(connection_string)
```

pyodbc.connect supports a keyword parameter autocommit. This controls the way the connection is handle. The default value (False) means that the commands that modify the database statements need to be committed explicitly. All commands between commits will form a single transaction. If autocommit is enabled every command will be issued and committed. It is also possible to change autocommit status after the connection is established.:

```
>>> connection.autocommit = True #enable autocommit
>>> connection.autocommit = False # disable autocommit
```

When not in autocommit mode, you can end a transaction by either commiting it or rolling it back.:

```
In[6]: connection.commit() # commit the transaction
In[7]: connection.rollback() # rollback the transaction
```

Note that commit/rollback is always performed at the connection level. pyodbc provides a commit/rollback method in the cursor objects, but they will act on the associated connection.

Working with cursors

Command execution in pyodbc is handled through cursors. You can create a cursor from a connection using the cursor() method. The first step is creating a cursor:

```
In[8]: cursor = connection.cursor()
```

With a cursor created, we can start issuing SQL commands using the execute method.

Creating a sample table

First, create a sample table in the database. The following code will create a sample table with three columns of different types.:

```
>>> def create_test_table(cursor):
...     try:
...         cursor.execute('drop table test_table')
```

(continues on next page)

(continued from previous page)

```
...    except:
...        pass
...    cursor.execute('''create table test_table (
...                                    name varchar(10),
...                                    fval float(24),
...                                    ival int)''')
...    cursor.commit()

>>> create_test_table(cursor)
```

Filling the sample table with sample data

After creating the table, rows can be inserted by executing insert into the table. Note you can pass parameters by placing a ? into the SQL statement. The parameters will be taken in order for the sequence appears in the next parameter.:

```
>>> cursor.execute('''insert into test_table values (?,?,?)''', ('foo', 3.0, 2))
>>> cursor.rowcount
1
```

Using executemany a sequence of parameters to the SQL statement can be passed and the statement will be executed many times, each time with a different parameter set. This allows us to easily insert several rows into the database so that we have a small test set::

```
>>> cursor.executemany('''insert into test_table values (?,?,?)''', [
...                         ('several', 2.1, 3),
...                         ('tuples', -1.0, 2),
...                         ('can', 3.0, 1),
...                         ('be', 12.0, -3),
...                         ('inserted', 0.0, -2),
...                         ('at', 33.0, 0),
...                         ('once', 0.0, 0)
...                         ])
```

Remember that if autocommit is turned off the changes won't be visible to any other connection unless we commit.:

```
>>> cursor.commit() # remember this is a shortcut to connection.commit() method
```

Querying the sample data from the sample table

Having populated our sample database, we can retrieve the inserted data by executing select statements::

```
>>> cursor.execute('''select * from test_table''')
<pyodbc.Cursor at 0x6803510>
```

After calling execute with the select statement we need to retrieve the data. This can be achieved by calling fetch methods in the cursor fetchone fetches the next row in the cursor, returning it in a tuple:

```
>>> cursor.fetchone()
('foo', 3.0, 2)
```

fetchmany retrieves several rows at a time in a list of tuples:

```
>>> cursor.fetchmany(3)
[('several', 2.0999999046325684, 3), ('tuples', -1.0, 2), ('can', 3.0, 1)]
```

fetchall retrieves all the remaining rows in a list of tuples:

```
>>> cursor.fetchall()
[('be', 12.0, -3), ('inserted', 0.0, -2), ('at', 33.0, 0), ('once', 0.0, 0)]
```

All the calls to any kind of fetch advances the cursor, so the next fetch starts in the row after the last row fetched. execute returns the cursor object. This is handy to retrieve the full query by chaining fetchall. This results in a one-liner::

```
>>> cursor.execute('''select * from test_table''').fetchall()
[('foo', 3.0, 2),
 ('several', 2.0999999046325684, 3),
 ('tuples', -1.0, 2),
 ('can', 3.0, 1),
 ('be', 12.0, -3),
 ('inserted', 0.0, -2),
 ('at', 33.0, 0),
 ('once', 0.0, 0)]
```

iopro.pyodbc extensions

When using iopro.pyodbc it is possible to retrieve the results from queries directly into numpy containers. This is accomplished by using the new cursor methods fetchdictarray and fetchsarray.

fetchdictarray

fetchdictarray fetches the results of a query in a dictionary. By default fetchdictarray fetches all remaining rows in the cursor.:

```
>>> cursor.execute('''select * from test_table''')
>>> dictarray = cursor.fetchdictarray()
>>> type(dictarray)
dict
```

The keys in the dictionary are the column names::

```
>>> dictarray.keys()
['ival', 'name', 'fval']
```

Each column name is mapped to a numpy array (ndarray) as its value::

```
>>> ', '.join([type(dictarray[i]).__name__ for i in dictarray.keys()])
'ndarray, ndarray, ndarray'
```

The types of the numpy arrays are infered from the database column information. So for our columns we get an appropriate numpy type. Note that in the case of name the type is a string of 11 characters even if in test_table is defined as varchar(10). The extra parameter is there to null-terminate the string::

```
>>> ', '.join([repr(dictarray[i].dtype) for i in dictarray.keys()])
"dtype('int32'), dtype('|S11'), dtype('float32')"
```

The numpy arrays will have a shape containing a single dimension with the number of rows fetched::

```
>>> ', '.join([repr(dictarray[i].shape) for i in dictarray.keys()])
'(8L,), (8L,), (8L,)'
```

The values in the different column arrays are index coherent. So in order to get the values associated to a given row it suffices to access each column using the appropriate index. The following snippet shows this correspondence::

```
>>> print '\n'.join(
... [', '.join(
...     [repr(dictarray[i][j]) for i in dictarray.keys()])
...         for j in range(dictarray['name'].shape[0])])
2, 'foo', 3.0
3, 'several', 2.0999999
2, 'tuples', -1.0
1, 'can', 3.0
-3, 'be', 12.0
-2, 'inserted', 0.0
0, 'at', 33.0
0, 'once', 0.0
```

Having the results in numpy containers makes it easy to use numpy to analyze the data::

```
>>> import numpy as np
>>> np.mean(dictarray['fval'])
6.5124998092651367
```

fetchdictarray accepts an optional parameter that places an upper bound to the number of rows to fetch. If there are not enough elements left to be fetched in the cursor the arrays resulting will be sized accordingly. This way it is possible to work with big tables in chunks of rows.:

```
>>> cursor.execute('''select * from test_table''')
>>> dictarray = cursor.fetchdictarray(6)
>>> print dictarray['name'].shape
(6L,)
>>> dictarray = cursor.fetchdictarray(6)
>>> print dictarray['name'].shape
(2L,)
```

fetchsarray

fetchsarray fetches the result of a query in a numpy structured array.:

```
>>> cursor.execute('''select * from test_table''')
>>> sarray = cursor.fetchsarray()
>>> print sarray
[('foo', 3.0, 2) ('several', 2.0999999046325684, 3) ('tuples', -1.0, 2)
 ('can', 3.0, 1) ('be', 12.0, -3) ('inserted', 0.0, -2) ('at', 33.0, 0)
 ('once', 0.0, 0)]
```

The type of the result is a numpy array (ndarray)::

```
>>> type(sarray)
numpy.ndarray
```

The dtype of the numpy array contains the description of the columns and their types::

```
>>> sarray.dtype
dtype([('name', '|S11'), ('fval', '&lt;f4'), ('ival', '&lt;i4')])
```

The shape of the array will be one-dimensional, with cardinality equal to the number of rows fetched::

```
>>> sarray.shape
(8L,)
```

It is also possible to get the shape of a column. In this way it will look similar to the code needed when using dictarrays:

```
>>> sarray['name'].shape
(8L,)
```

In a structured array it is as easy to access data by row or by column::

```
>>> sarray['name']
array(['foo', 'several', 'tuples', 'can', 'be', 'inserted', 'at', 'once'],
      dtype='|S11')








>>> sarray[0]
('foo', 3.0, 2)
```

It is also very easy and efficient to feed data into numpy functions::

```
>>> np.mean(sarray['fval'])
6.5124998092651367
```

fetchdictarray vs fetchsarray

Both methods provide ways to input data from a database into a numpy-friendly container. The structured array version provides more flexibility extracting rows in an easier way. The main difference is in the memory layout of the resulting object. An in-depth analysis of this is beyond the scope of this notebook. Suffice it to say that you can view the dictarray laid out in memory as an structure of arrays (in fact, a dictionary or arrays), while the structured array would be laid out in memory like an array of structures. This can make a lot of difference performance-wise when working with large chunks of data.

iopro.pyodbc Enhanced Capabilities

Demo code showing the enhanced capabilities of iopro.pyodbc submodule

This demo shows the basic capabilities for the iopro.pyodbc module. It first will connect with the database of your choice by ODBC, create and fill a new table (market) and then retrieve data with different methods (fetchall(), fetchdictarray() and fetchsarray()).

Author: Francesc Alted, Anaconda

```
>>> import iopro.pyodbc as pyodbc
>>> # Open the database (use the most appropriate for you)
>>> connect_string = 'DSN=odbcsqlite;DATABASE=market.sqlite'  # SQLite
>>> #connect_string = 'Driver={SQL Server};SERVER=MyWinBox;DATABASE=Test;USER=Devel;
↪PWD=XXX'  # SQL Server
>>> #connect_string = 'DSN=myodbc3;UID=devel;PWD=XXX;DATABASE=test'  # MySQL
>>> #connect_string = 'DSN=PSQL;UID=devel;PWD=XXX;DATABASE=test'   # PostgreSQL
>>> connection = pyodbc.connect(connect_string)
>>> cursor = connection.cursor()
```

Create the test table (optional if already done)

```
>>> try:
...     cursor.execute('drop table market')
... except:
...     pass
>>> cursor.execute('create table market (symbol_ varchar(5), open_ float, low_ float,␣
↪high_ float, close_ float, volume_ int)')
```

Fill the test table (optional if already done)

```
>>> from time import time
>>> t0 = time()
>>> N = 1000*1000
>>> for i in xrange(N):
...     cursor.execute(
...         "insert into market(symbol_, open_, low_, high_, close_, volume_)"
...         " values (?, ?, ?, ?, ?, ?)",
...         (str(i), float(i), float(2*i), None, float(4*i), i))
>>> cursor.execute("commit")             # not supported by SQLite
>>> t1 = time() - t0
>>> print "Stored %d rows in %.3fs" % (N, t1)
```

Do the query in the traditional way

```
>>> # Query of the full table using the traditional fetchall
>>> query = "select * from market"
>>> cursor.execute(query)
>>> %time all = cursor.fetchall()
CPU times: user 5.23 s, sys: 0.56 s, total: 5.79 s
Wall time: 7.09 s
```

Do the query and get a dictionary of NumPy arrays

```
>>> # Query of the full table using the fetchdictarray (retrieve a dictionary of␣
↪arrays)
>>> cursor.execute(query)
>>> %time dictarray = cursor.fetchdictarray()
```

(continues on next page)

(continued from previous page)

```
CPU times: user 0.92 s, sys: 0.10 s, total: 1.02 s
Wall time: 1.44 s
```

Peek into the retrieved data

```
>>> dictarray.keys()
['high_', 'close_', 'open_', 'low_', 'volume_', 'symbol_']
>>> dictarray['high_']
array([ nan,  nan,  nan, ...,  nan,  nan,  nan])
>>> dictarray['symbol_']
array(['0', '1', '2', ..., '99999', '99999', '99999'], dtype='|S6')
```

Do the query and get a NumPy structured array

```
>>> # Query of the full table using the fetchsarray (retrieve a structured array)
>>> cursor.execute(query)
>>> %time sarray = cursor.fetchsarray()
CPU times: user 1.08 s, sys: 0.11 s, total: 1.20 s
Wall time: 1.99 s
```

Peek into retrieved data

```
>>> sarray.dtype
dtype([('symbol_', 'S6'), ('open_', '&lt;f8'), ('low_', '&lt;f8'), ('high_', '&lt;f8
↪'), ('close_', '&lt;f8'), ('volume_', '&lt;i4')])
>>> sarray[0:10]
array([('0', 0.0, 0.0, nan, 0.0, 0), ('1', 1.0, 2.0, nan, 4.0, 1),
       ('2', 2.0, 4.0, nan, 8.0, 2), ('3', 3.0, 6.0, nan, 12.0, 3),
       ('4', 4.0, 8.0, nan, 16.0, 4), ('5', 5.0, 10.0, nan, 20.0, 5),
       ('6', 6.0, 12.0, nan, 24.0, 6), ('7', 7.0, 14.0, nan, 28.0, 7),
       ('8', 8.0, 16.0, nan, 32.0, 8), ('9', 9.0, 18.0, nan, 36.0, 9)],
      dtype=[('symbol_', 'S6'), ('open_', '&lt;f8'), ('low_', '&lt;f8'), ('high_', '&
↪lt;f8'), ('close_', '&lt;f8'), ('volume_', '&lt;i4')])
>>> sarray['symbol_']
array(['0', '1', '2', ..., '99999', '99999', '99999'], dtype='|S6')
```

iopro.pyodbc Cancelling Queries

Starting with version 1.5, the pyodbc submodule of IOPro makes it possible to cancel operations. This is done by exposing the SQLCancel ODBC function as a cancel method in the Cursor object.

A Simple Example

A very simple example would be:

```
conn = iopro.pyodbc.connect(conn_str)
cursor = conn.cursor()
cursor.execute('SELECT something FROM sample_table')
result = cursor.fetchone()
cursor.cancel()
```

This is not very interesting, and it doesn't add much to the functionality of pyodbc.

What makes the cancel method more interesting is that it is possible to cancel running statements that are blocking another thread.

A Sample With Threading

Having access to the cancel method it is possible to stop running queries following different criteria. For example, it would be possible to execute queries with a time-out. If the time runs out, the query gets cancelled.

```
import iopro.pyodbc
import time
import threading

def query_with_time_out(conn, query, timeout):
    def watchdog(cursor, time_out):
        time.sleep(wait_time)
        cursor.cancel()

    cursor = conn.cursor()

    t = threading.Thread(target=watchdog, args=(cursor, timeout))
    t.start()
    try:
        cursor.execute(query)

        result = cursor.fetchall()
    except iopro.pyodbc.Error:
        result = 'timed out'

    return result
```

This is just one possibility. As cursor exposes directly the SQLCancel, many oportunities open in implementing policies to cancel running queries.

Finishing notes

In order for this to work, the underlying ODBC driver must support SQLCancel.

The pyodbc submodule of IOPro releases the Python GIL when it calls ODBC, so while queries are being executed other Python threads continue to execute while the thread that performed the query is blocked. This allows for cancel to be called by another thread. Coupled with threading, the cancel method is a very useful primitive.

IOPro END USER LICENSE AGREEMENT

IOPro ("the Software Product") and accompanying documentation is licensed and not sold. The Software Product is protected by copyright laws and treaties, as well as laws and treaties related to other forms of intellectual property. Continuum Analytics, Inc. (dba Anaconda, Inc.) or its subsidiaries, affiliates, and suppliers (collectively "Continuum") own intellectual property rights in the Software Product. The Licensee's ("you" or "your") license to download, use, copy, or change the Software Product is subject to these rights and to all the terms and conditions of this End User License Agreement ("Agreement").

Acceptance

YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY SELECTING THE "ACCEPT" OPTION AND DOWNLOADING THE SOFTWARE PRODUCT OR BY INSTALLING, USING, OR COPYING THE SOFTWARE PRODUCT. YOU MUST AGREE TO ALL OF THE TERMS OF THIS AGREEMENT BEFORE YOU WILL BE ALLOWED TO DOWNLOAD THE SOFTWARE PRODUCT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST SELECT "DECLINE" AND YOU MUST NOT INSTALL, USE, OR COPY THE SOFTWARE PRODUCT.

Trial Period

You have the right to use IOPro on a single computer or group of computers for 30 days with no license. After 30 days you must purchase an appropriate license to use the software on one or more machines or stop using the software and remove it from all of your machines on which you installed the software.

License Grant

This Agreement entitles you to install and use one copy of the Software Product on as many machines as you will personally use. The Software Product is licensed to a particular user. Only the user to whom the software is licensed may use the software. You must obtain a license for as many users as you wish to use the software. In addition, you may make archival copies of the Software Product Installer.

Right to Updates

This license entitles you to updates to the Software Product for one year from the time of payment. The Software Product will continue to function and you may continue to use The Software Product and any updates you have received for as long as you would like however you will no longer be able to receive updates from Continuum unless this License is renewed. Please contact sales@anaconda.com with any questions or concerns.

Restrictions on Transfer

Without first obtaining the express written consent of Continuum, you may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software Product.

Restrictions on Use

You may not use, copy, or install the Software Product on any system where more than one user will be able to use the software unless you have purchased a license for each user of the system. You may not decompile, "reverse-engineer", disassemble, or otherwise attempt to derive the source code for the Software Product.

Restrictions on Alteration

You may not modify the Software Product or create any derivative work of the Software Product or its accompanying documentation. Derivative works include but are not limited to translations. You may not alter any files or libraries in any portion of the Software Product.

Restrictions on Copying

You may not copy any part of the Software Product except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make archival copies of the Software Product installer.

Limited Software Product Warranty

For a period of 60 days from the date of shipment or from the date that you download the Software Product, as applicable, Continuum warrants that when properly installed and used under normal conditions, the Software Product will perform substantially as advertised.

Disclaimer of Warranties and Limitation of Liability

UNLESS OTHERWISE EXPLICITLY AGREED TO IN WRITING BY CONTINUUM, CONTINUUM MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE PRODUCT.

Continuum makes no warranty that the Software Product will meet your requirements or operate under your specific conditions of use. Continuum makes no warranty that operation of the Software Product will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE PRODUCT SUFFICIENTLY MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE PRODUCT TO MEET YOUR REQUIREMENTS. CONTINUUM WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE. UNDER NO CIRCUMSTANCES SHALL CONTINUUM, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE PRODUCT, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF CONTINUUM OR ANY OTHER PARTY, EVEN IF CONTINUUM IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS CONTINUUM'S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

Limitation of Remedies and Damages

Your remedy for a breach of this Agreement or of any warranty included in this Agreement is the correction or replacement of the Software Product. Selection of whether to correct or replace shall be solely at the discretion of Continuum. Continuum reserves the right to substitute a functionally equivalent copy of the Software Product as a replacement. If Continuum is unable to provide a replacement or substitute Software Product or corrections to the Software Product, your sole alternate remedy shall be a refund of the purchase price for the Software Product exclusive of any costs for shipping and handling. Any claim must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by Continuum to have been caused by you. All limited warranties on the Software Product are granted only to you and are non-transferable. You agree to indemnify and hold Continuum harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

Governing Law, Jurisdiction and Costs

This Agreement is governed by the laws of Texas, without regard to Texas's conflict or choice of law provisions.

Export Regulations

Any use or distribution of IOPro is made under conditions that the user and/or distributor is in full compliance with all export and other governing laws of the United States of America, including full and ongoing compliance with the Export Administration Regulations (EAR) of the United States Department of Commerce. See www.commerce.gov/ and http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear. Use or distribution of Continuum software products to any persons, entities or countries currently under US sanctions is strictly prohibited. IOPro is classified with an ECCN of 5D992 with no license required for export to non-embargoed countires.

The United States currently has embargoes against Cuba, Iran, North Korea, Sudan and Syria. The exportation, re-exportation, sale or supply, directly or indirectly, from the United States, or by a U.S. person wherever located, of any Continuum software to any of these countries is strictly prohibited without prior authorization by the United States Government By accepting this Agreement, you represent to Continuum that you will comply with all applicable export regulations for IOPro.

Severability

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

IOPro Release Notes

2016-07-30: 1.9.0

- Remove warnings and documentation for unsupported Numba use
- Rewrite most documentation for clarity and accuracy
- Improve unit tests

2016-04-05: 1.8.0:

- Add PostgresAdapter for reading data from PostgreSQL databases
- Add AccumuloAdapter for reading data from Accumulo databases

2015-10-09: 1.7.2:

- Fix an issue with pyodbc where result NumPy arrays could return uninitialized data after the actual data null character. Now it pads the results with nulls.

2015-05-04: 1.7.1

- Properly cache output string objects for better performance

2015-03-02: 1.7.0

- Add Python 3 support
- Add support for parsing utf8 text files
- Add ability to set/get field types in MongoAdapter

2015-02-02: 1.6.11

- Fix issue with escape char not being parsed correctly inside quoted strings

2014-12-17: 1.6.10

- Fix issue with using field filters with json parser

2014-12-02: 1.6.9

- Fix issue with json field names getting mixed up

2014-11-20: 1.6.8

- Fix issue with return nulls returning wrong "null" for large queries (more than 10000 rows) in some circumstances.
- Fix issue with reading slices of json data
- Change json parser so that strings fields of numbers do not get converted to number type by default
- Allow json field names to be specified with field_names constructor argument
- If user does not specify json field names, use json attribute names as field names in array result

2014-07-03: 1.6.7

- Fix issue when reading more than 10000 rows containing unicode strings in platform where ODBC uses UTF-16/UCS2 encoding (notably Windows and unixODBC). The resulting data could be corrupt.

2014-06-16: 1.6.6

- Fix possible segfault when dealing with unicode strings in platforms where ODBC uses UTF-16/UCS2 encoding (notably Windows and unixODBC)
- Add iopro_set_text_limit function to iopro. It globally limits the size of text fields read by fetchdictarray and fetchsarray. By default it is set to 1024 characters.
- Fix possible segfault in fetchdictarray and fetchsarray when failing to allocate some NumPy array. This could notably happen in the presence of "TEXT" fields. Now it will raise an OutOfMemory error.
- Add lazy loading of submodules in IOPro. This reduces upfront import time of IOPro. Features are imported as they are used for the first time.

2014-05-07: 1.6.5

- Fix crash when building textadapter index

2014-04-29: 1.6.4

- Fix default value for null strings in IOPro/pyodbc changed to be an empty string instead of 'NA'. NA was not appropriate as it can collide with valid data (Namibia country code is 'NA', for example), and it failed with single character columns.
- Ignore SQlRowCount when performing queries with fetchsarray and fetchdictarray, since SQLRowCount sometimes returns incorrect number of rows.

2014-03-25: 1.6.3

- Fix SQL TINYINT is now returned as an unsigned 8 bit integer in fetchdictarray/fetchsarray. This is to match the range specified in SQL (0. . . 255). It was being returned as a signed 8 bit integer before (range -128. . . 127)
- Add Preliminary unicode string support in fetchdictarray/fetchsarray.

2014-02-12: 1.6.2

- Disable Numba support for version 0.12 due to lack of string support.

2014-01-30: 1.6.1

- Fix a regression that made possible some garbage in string fields when using fetchdictarray/fetchsarray.
- Fix a problem where heap corruption could happen in IOPro.pyodbc fetchdictarray/fetchsarray related to nullable string fields.
- Fix the allocation guard debugging code: iopro.pyodbc.enable_mem_guards(True|False) should no longer crash.
- Merge Vertica fix for cancelling queries

2013-10-30: 1.6.0

- Add JSON support
- Misc bug fixes
- Fix crash in IOPro.pyodbc when dealing with nullable datetimes in fetch_dictarray and fetch_sarray.

2013-06-12: 1.5.5

- Fix issue parsing negative ints with leading whitespace in csv data.

2013-06-10: 1.5.4

- Allow delimiter to be set to None for csv files with single field.
- Fill in missing csv fields with fill values.
- Fill in blank csv lines with fill values for pandas dataframe output.
- Allow list of field names for TextAdapter field_names parameter.
- Change default missing fill value to empty string for string fields.

2013-06-05: 1.5.3

- Temporary fix for IndexError exception in TextAdapter.__read_slice method.

2013-05-28: 1.5.2

- Add ability to specify escape character in csv data

2013-05-23: 1.5.1

- fixed coredump when using datetime with numpy < 1.7

2013-05-22: 1.5.0

- Added a cancel method to the Cursor object in iopro.pyodbc. This method wraps ODBC SQLCancel.
- DECIMAL and NUMERIC types are now working on iopro.pyodbc on regular fetch functions. They are still unsupported in fetchsarray and fetchdict and fetchsarray
- Add ftp support
- Performance improvements to S3 support
- Misc bug fixes

2013-04-05: 1.4.3

- Update loadtxt and genfromtxt to reflect numpy versions' behavior for dealing with whitespace (default to any whitespace as delimiter, and treat multiple whitespace as one delimiter)
- Add read/write field_names property
- Add support for pandas dataframes as output
- Misc bug fixes

Reference guide

TextAdapter

Contents

The TextAdapter module reads CSV data and produces a NumPy array containing the parsed data. The following features are currently implemented:

- The TextAdapter engine is written in C to ensure text is parsed as fast as data can be read from the source. Text is read and parsed in small chunks instead of reading entire data into memory at once, which enables very large files to be read and parsed without running out of memory.
- Python slicing notation can be used to specify a subset of records to be read from the data source, as well as a subset of fields.
- Fields can be specified in any one of three ways: by a delimiter character, using fixed field widths, or by a regular expression. This enables a larger variety of CSV-like and other types of text files to be parsed.
- A gzipped file can be parsed without having to uncompress it first. Parsing speed is about the same as an uncompressed version of same file.
- An index of record offsets in a file can be built to allow fast random access to records. This index can be saved to disk and loaded again later.
- Converter functions can be specified for converting parsed text to proper dtype for storing in NumPy array.
- The TextAdapter engine has automatic type inference so the user does not have to specify dtypes of the output array. The user can still specify dtypes manually if desired.
- Remote data stored in Amazon S3 can be read. An index can be built and stored with S3 data. Index can be read remotely, allowing for random access to S3 data.

Methods

The TextAdapter module contains the following factory methods for creating TextAdapter objects:

text_adapter (source, parser='csv', compression=None, comment='#',

quote='"', num_records=0, header=0, field_names=True, indexing=False, index_name=None, encoding='utf-8')

Create a text adapter for reading CSV, JSON, or fixed width
text files, or a text file defined by regular expressions.

source - filename, file object, StringIO object, BytesIO object, S3 key, http url, or python generator
parser - Type of parser for parsing text. Valid parser types are 'csv', 'fixed width', 'regex', and 'json'.
encoding - type of character encoding (current ascii and utf8 are supported)
compression - type of data compression (currently only gzip is supported)
comment - character used to indicate comment line
quote - character used to quote fields
num_records - limits parsing to specified number of records; defaults to all records
header - number of lines in file header; these lines are skipped when parsing
footer - number of lines in file footer; these lines are skipped when parsing
indexing - create record index on the fly as characters are read
index_name - name of file to write index to
output - type of output object (numpy array or pandas dataframe)

If parser is set to 'csv', additional parameters include:

delimiter - Delimiter character used to define fields in data source. Default is ','.

If parser is set to 'fixed_width', additional parameters include:

field_widths - List of field widths

If parser is set to 'regex', additional parameters include:

regex - Regular expression used to define records and fields in data source. See the regular expression example in the Advanced Usage section.

s3_text_adapter (access_key, secret_key, bucket_name, key_name, remote_s3_index=False)

parser='csv', compression=None, comment='#', quote='"', num_records=0, header=0, field_names=True, indexing=False, index_name=None, encoding='utf-8')

Create a text adapter for reading a text file from S3. Text file can be
CSV, JSON, fixed width, or defined by regular expressions

In addition to the arguments described for the text_adapter function above, the s3_text_adapter function also has the following parameters:

access_key - AWS access key
secret_key - AWS secret key
bucket_name - name of S3 bucket
key_name - name of key in S3 bucket
remote_s3_index - use remote S3 index (index name must be key name + '.idx' extension)

The TextAdapter object returned by the text_adapter factory method contains the following methods:

set_converter (field, converter)

Set converter function for field

field - field to apply converter function

converter - python function object

set_missing_values (missing_values)

Set strings for each field that represents a missing value

missing_values - dict of field name or number, and list of missing value strings

Default missing values: 'NA', 'NaN', 'inf', '-inf', 'None', 'none', ''

set_fill_values (fill_values, loose=False)

Set fill values for each field

fill_values - dict of field name or number, and fill value

loose - If value cannot be converted, and value does not match any of the missing values, replace with fill value anyway.

Default fill values for each data type: | int - 0 | float - numpy.nan | char - 0 | bool - False | object - numpy.nan | string - numpy.nan

create_index (index_name=None, density=1)

Create an index of record offsets in file

index_name - Name of file on disk used to store index. If None, index will be created in memory but not saved.

density - density of index. Value of 1 will index every record, value of 2 will index every other record, etc.

to_array ()

Parses entire data source and returns data as NumPy array object

to_dataframe ()

Parses entire data source and returns data as Pandas DataFrame object

The TextAdapter object contains the following properties:

size (readonly)

Number of records in data source. This value is only set if entire data source has been read or indexed, or number of recods was specified in text_adapter factory method when creating object.

field_count (readonly)

Number of fields in each record

field_names

Field names to use when creating output NumPy array. Field names can be set here before reading data or in text_adapter function with field_names parameter.

field_types

NumPy dtypes for each field, specified as a dict of fields and associated dtype. (Example: {0:'u4', 1:'f8', 2:'S10'})

field_filter

Fields in data source to parse, specified as a list of field numbers or names (Examples: [0, 1, 2] or ['f1', 'f3', 'f5']). This filter stays in effect until it is reset to empty list, or is overridden with array slicing (Example: adapter[[0, 1, 3, 4]][:]).

See the NumPy data types documentation for more details: http://docs.continuum.io/anaconda/numpy/reference/arrays.dtypes.html

The TextAdapter object supports array slicing:

Read all records: adapter[:]

Read first 100 records: adapter[0:100]

Read last record (only if data has been indexed or entire dataset has been read once before): adapter[-1]

Read first field in all records by specifying field number: adapter[0][:]

Read first field in all records by specifying field name: adapter['f0'][:]

Read first and third fields in all records: adapter[[0, 2]][:]

Basic Usage

Create TextAdapter object for data source:

```
>>> import iopro
>>> adapter = iopro.text_adapter('data.csv', parser='csv')
```

Parse text and store records in NumPy array using slicing notation:

```
>>> # read all records
>>> array = adapter[:]

>>> # read first ten records
>>> array = adapter[0:10]

>>> # read last record
>>> array = adapter[-1]

>>> # read every other record
>>> array = adapter[::2]
```

Advanced Usage

user defined converter function for field 0:

```
>>> import iopro
>>> import io

>>> data = '1, abc, 3.3\n2, xxx, 9.9'
>>> adapter = iopro.text_adapter(io.StringIO(data), parser='csv', field_names=False)

>>> # Override default converter for first field
>>> adapter.set_converter(0, lambda x: int(x)*2)
>>> adapter[:]
array([(2L, ' abc', 3.3), (4L, ' xxx', 9.9)],
          dtype=[('f0', '<u8'), ('f1', 'S4'), ('f2', '<f8')])
```

overriding default missing and fill values:

```
>>> import iopro
>>> import io

>>> data = '1,abc,inf\n2,NA,9.9'
>>> adapter = iopro.text_adapter(io.StringIO(data), parser='csv', field_names=False)

>>> # Define field dtypes (example: set field 1 to string object and field 2 to float)
>>> adapter.field_types = {1:'O', 2:'f4'}

>>> # Define list of strings for each field that represent missing values
>>> adapter.set_missing_values({1:['NA'], 2:['inf']})

>>> # Set fill value for missing values in each field
>>> adapter.set_fill_values({1:'xxx', 2:999.999})
>>> adapter[:]
array([(' abc', 999.9990234375), ('xxx', 9.899999618530273)],
          dtype=[('f0', 'O'), ('f1', '<f4')])
```

creating and saving tuple of index arrays for gzip file, and reloading indices:

```
>>> import iopro
>>> adapter = iopro.text_adapter('data.gz', parser='csv', compression='gzip')

>>> # Build index of records and save index to disk.
>>> adapter.create_index(index_name='index_file')

>>> # Create new adapter object and load index from disk.
>>> adapter = iopro.text_adapter('data.gz', parser='csv', compression='gzip',␣
↪indexing=True, index_name='index_file')

>>> # Read last record
>>> adapter[-1]
array([(100, 101, 102)],dtype=[('f0', '<u4'), ('f1', '<u4'), ('f2', '<u4')])
```

Use regular expression for finer control of extracting data:

```
>>> import iopro
>>> import io

```

(continues on next page)

(continued from previous page)

```
>>> # Define regular expression to extract dollar amount, percentage, and month.
>>> # Each set of parentheses defines a field.
>>> data = '$2.56, 50%, September 20 1978\n$1.23, 23%, April 5 1981'
>>> regex_string = '([0-9]\.[0-9][0-9]+)\,\s ([0-9]+)\%\,\s ([A-Za-z]+)'
>>> adapter = iopro.text_adapter(io.StringIO(data), parser='regex', regex_
↪string=regex_string, field_names=False, infer_types=False)

>>> # set dtype of field to float
>>> adapter.field_types = {0:'f4', 1:'u4', 2:'S10'}
>>> adapter[:]
array([(2.56, 50L, 'September'), (1.23, 23L, 'April')],
    dtype=[('f0', '<f8'), ('f1', '<u8'), ('f2', 'S9')])
```

iopro.pyodbc

This project is an enhancement of the Python database module for ODBC that implements the Python DB API 2.0 specification. You can see the original project here:

homepage http://code.google.com/p/pyodbc

source http://github.com/mkleehammer/pyodbc

source http://code.google.com/p/pyodbc/source/list

The enhancements are documented in this file. For general info about the pyodbc package, please refer to the original project documentation.

This module enhancement requires:

- Python 2.4 or greater
- ODBC 3.0 or greater
- NumPy 1.5 or greater (1.7 is required for datetime64 support)

The enhancements in this module consist mainly in the addition of some new methods for fetching the data after a query and put it in a variety of NumPy containers.

Using NumPy as data containers instead of the classical list of tuples has a couple of advantages:

1) The NumPy container is much more compact, and hence, it requires much less memory, than the original approach.

2) As a NumPy container can hold arbitrarily large arrays, it requires much less object creation than the original approach (one Python object per datum retrieved).

This means that this enhancements will allow to fetch data out of relational databases in a much faster way, while consuming significantly less resources.

API additions

Variables

- *pyodbc.npversion* The version for the NumPy additions

Methods

Cursor.fetchdictarray (size=cursor.arraysize)

This is similar to the original *Cursor.fetchmany(size)*, but the data is returned in a dictionary where the keys are the names of the columns and the values are NumPy containers.

For example, it a SELECT is returning 3 columns with names 'a', 'b' and 'c' and types *varchar(10)*, *integer* and *timestamp*, the returned object will be something similar to:

```
{'a': array([...], dtype='S11'),
 'b': array([...], dtype=int32),
 'c': array([...], dtype=datetime64[us])}
```

Note that the *varchar(10)* type is translated automatically to a string type of 11 elements ('S11'). This is because the ODBC driver needs one additional space to put the trailing '0' in strings, and NumPy needs to provide the room for this.

Also, it is important to stress that all the *timestamp* types are translated into a NumPy *datetime64* type with a resolution of microseconds by default.

Cursor.fetchsarray (size=cursor.arraysize)

This is similar to the original *Cursor.fetchmany(size)*, but the data is returned in a NumPy structured array, where the name and type of the fields matches to those resulting from the SELECT.

Here it is an example of the output for the SELECT above:

```
array([(...),
       (...)],
      dtype=[('a', '|S11'), ('b', '<i4'), ('c', ('<M8[us]', {}))])
```

Note that, due to efficiency considerations, this method is calling the *fetchdictarray()* behind the scenes, and then doing a conversion to get an structured array. So, in general, this is a bit slower than its *fetchdictarray()* counterpart.

Data types supported

The new methods listed above have support for a subset of the standard ODBC. In particular:

- String support (SQL_VARCHAR) is supported.
- Numerical types, be them integers or floats (single and double precision) are fully supported. Here it is the complete list: SQL_INTEGER, SQL_TINYINT, SQL_SMALLINT, SQL_FLOAT and SQL_DOUBLE.
- Dates, times, and timestamps are mapped to the *datetime64* and *timedelta* NumPy types. The list of supported data types are: SQL_DATE, SQL_TIME and SQL_TIMESTAMP,
- Binary data is not supported yet.
- Unicode strings are not supported yet.

NULL values

As there is not (yet) a definitive support for missing values (NA) in NumPy, this module represents NA data as particular values depending on the data type. Here it is the current table of the particular values:

```
int8: -128 (-2**7)
uint8: 255 (2**8-1)
int16: -32768 (-2**15)
uint16: 65535 (2**16-1)
int32: -2147483648 (-2**31)
uint32: 4294967295 (2**32-1)
int64: -9223372036854775808 (-2**63)
uint64: 18446744073709551615 (2**64-1)
float32: NaN
float64: NaN
datetime64: NaT
timedelta64: NaT (or -2**63)
string: 'NA'
```

Improvements for 1.1 release

- The rowcount is not trusted anymore for the *fetchdict()* and *fetchsarray()* methods. Now the NumPy containers are built incrementally, using realloc for a better use of resources.
- The Python interpreter does not exit anymore when fetching an exotic datatype not supported by NumPy.
- The docsctrings for *fetchdict()* and *fetchsarray()* have been improved.

MongoAdapter

Contents

The MongoAdapter module reads data from a Mongo database collection and produces a NumPy array containing the loaded. The following features are currently implemented:

- The MongoAdapter engine is written in C to ensure data is loaded fast with minimal memory usage.
- Python slicing notation can be used to specify the subset of records to be read from the data source.
- The MongoAdapter engine has automatic type inference so the user does not have to specify dtypes of the output array.

Methods

The MongoAdapter module contains the follwowing constructor for creating MongoAdapter objects:

MongoAdapter (host, port, database, collection)

MongoAdapter contructor

host - Host name where Mongo database is running.
port - Port number where Mongo database is running.

database - Mongo database to connect to

collection - Mongo database collection

set_field_names (names)

Set field names to read when creating output NumPy array.

get_field_names ()

Returns names of fields that will be read when reading data from Mongo database.

set_field_types (types=None)

Set NumPy dtypes for each field, specified as a dict of field names/indices and associated dtype. (Example: {0:'u4', 1:'f8', 2:'S10'})

get_field_types ()

Returns dict of field names/indices and associated NumPy dtype.

The MongoAdapter object contains the following properties:

size (readonly)

Number of documents in the Mongo database + collection specified in constructor.

Basic Usage

1. Create MongoAdapter object for data source

```
>>> import iopro
>>> adapter = iopro.MongoAdapter('localhost', 27017, 'database_name', 'collection_
↪name')
```

2. Load Mongo collection documents into NumPy array using slicing notation

```
>>> # read all records for 'field0' field
>>> array = adapter['field0'][:]
```

```
>>> # read first ten records for 'field0' and 'field1' fields
>>> array = adapter[['field0', 'field1']][0:10]
```

```
>>> # read last record
>>> array = adapter['field0'][-1]
```

```
>>> # read every other record
>>> array = adapter['field0'][::2]
```

Accumulo Adapter

Contents

The AccumuloAdapter module reads data from Accumulo key/value stores and produces a NumPy array containing the parsed values.

- The AccumuloAdapter engine is written in C to ensure returned data is parsed as fast as data can be read from the source. Data is read and parsed in small chunks instead of reading entire data into memory at once.
- Python slicing notation can be used to specify a subset of records to be read from the data source.

Adapter Methods

Accumulo Adapter Constructor:

AccumuloAdapter (server='localhost', port=42424, username='', password='', table=None, field_type='f8', start_key=None, stop_key=None, start_key_inclusive=True, stop_key_inclusive=False, missing_values=None, fill_value=None):

> Create an adaptor for connecting to an Accumulo key/value store.
>
> server: Accumulo server address
> port: Accumulo port
> username: Accumulo user name
> password: Accumulo user password
> table: Accumulo table to read data from
> field_type: str, NumPy dtype to interpret table values as
> start_key: str, key of record where scanning will start from
> stop_key: str, key of record where scanning will stop at
> start_key_inclusive: If True, start_key is inclusive (default is True)
> stop_key_inclusive: If True, stop_key is inclusive (default is False)
> missing_values: list, missing value strings. Any values in table equal to one of these strings will be replaced with fill_value.
> fill_value: fill value used to replace missing value when scanning

close ()

> Close connection to the database.

The AccumuloAdapter object supports array slicing:

> Read all records: adapter[:]
>
> Read first ten records: adapter[0:10]
>
> Read last record: adapter[-1]
>
> Read every other record: adapter[::2]

Adapter Properties

field_type (readonly)

Get dtype of output NumPy array

start_key

Get/set key of record where reading/scanning will start.
The start_key_inclusive property specifies whether this key is inclusive
(default is inclusive).

stop_key

Get/set key of record where reading/scanning will stop.
The stop_key_inclusive property specifies whether this key is inclusive
(default is exclusive).

start_key_inclusive

Toggle whether start key is inclusive. Default is true.

stop_key_inclusive

Toggle whether stop key is inclusive. Default is False.

missing_values

Get/Set missing value strings. Any values in Accumulo table matching one
of these strings will be replaced with fill_value.

fill_value

Fill value used to replace missing_values. Fill value type should match
specified output type.

Basic Usage

Create AccumuloAdapter object for data source:

```
>>> import iopro
>>> adapter = iopro.AccumuloAdapter(server='172.17.0.1',
                                    port=42424,
                                    username='root',
                                    password='password',
                                    field_type='f4',
                                    table='iopro_tutorial_data')
```

IOPro adapters use slicing to retrieve data. To retrieve records from the table or query, the standard NumPy slicing notation can be used:

```
>>> # read all records
>>> array = adapter[:]
array([ 0.5,   1.5,   2.5,   3.5,   4.5], dtype=float32)
```

```
>>> # read first three records
>>> array = adapter[0:3]
array([ 0.5,   1.5,   2.5], dtype=float32)
```

```
>>> # read every other record from the first four records
>>> array = adapter[:4:2]
array([ 0.5,  2.5], dtype=float32)
```

The Accumulo adapter does not support seeking from the last record.

The field_types property can be used to see what type the output NumPy array will have:

```
>>> adapter.field_type
'f4'
```

Since Accumulo is essentially a key/value store, results can be filtered based on key. For example, a start key using the start_key property. This will retrieve all values with a key equal to or greater than the start key.

```
>>> adapter.start_key = 'row02'
>>> adapter[:]
array([ 1.5,  2.5,  3.5,  4.5], dtype=float32)
```

Likewise, a stop key. This will retrieve all values with a key less than the stop key but equal to or greater than the start key.

```
>>> adapter.stop_key = 'row04'
>>> adapter[:]
array([ 1.5,  2.5], dtype=float32)
```

By default, the start key is inclusive. This can be changed by setting the start_key_inclusive property to False.

```
>>> adapter.start_key_inclusive = False
>>> adapter[:]
array([ 2.5], dtype=float32)
```

By default, the stop key is exclusive. This can be changed by setting the stop_key_inclusive property to True.

```
>>> adapter.stop_key_inclusive = True
>>> adapter[:]
array([ 2.5,  3.5], dtype=float32)
```

The Accumulo adapter can handle missing values. If it is known that the strings 'NA' and 'nan' signify missing float values, the missing_values property can be used to tell the adapter to treat these strings as missing values: Also, the fill_value property can be used to specify what value to replace missing values with.

```
>>> adapter = iopro.AccumuloAdapter('172.17.0.1', 42424, 'root', 'password', 'iopro_
↪tutorial_missing_data', field_type='S10')
>>> adapter[:]
array([b'NA', b'nan'], dtype='|S10')
```

```
>>> adapter = iopro.AccumuloAdapter('172.17.0.1', 42424, 'root', 'secret', 'iopro_
↪tutorial_missing_data', field_type='f8')
>>> adapter.missing_values = ['NA', 'nan']
>>> adapter.fill_value = np.nan
>>> adapter[:]
array([ nan,  nan])
```

Close database connection:

```
>>> adapter.close()
```

PostgresAdapter, PostGIS, and GreenPlum

Contents

The PostgresAdapter module reads data from PostresSQL based databases and produces a NumPy array or a Pandas Dataframe containing the parsed data. The PostgresAdapter can be used to access data from PostgresSQL and GreenPlum, and has enhancements to support PostGIS points, lines, multilines, polygons, and multipolygons. The following features are currently implemented:

- The PostgresAdapter engine is written in C to ensure returned data is parsed as fast as data can be read from the source. Data is read and parsed in small chunks instead of reading entire data into memory at once.
- Python slicing notation can be used to specify a subset of records to be read from the data source.
- A subset of columns can be specified to be returned instead of returning all columns for the records.

Adapter Methods

PostgreSQL Adapter Constructor:

PostgresAdapter (connection_uri, table=None, query=None, field_filter=None, dataframe=False, field_names=None, field_types=None, field_shapes=None):

Create an adaptor for connecting to a PostgreSQL based database.

connection_uri: string URI describing how to connect to database

table: string, name of table to read records from. Only table

parameter or query parameter can be set, but not both.

query: string, custom query to use for reading records. Only query parameter or table parameter can be set, but not both. field_filter parameter cannot be set when query parameter is set (since it is trivial to specify fields in query string).

field_filter: names of fields include in query (only valid when table parameter is specified)

dataframe: bool, return results as dataframe instead of array

field_names: list, names of fields in output array or dataframe. Defaults to database table column names.

field_types: list, NumPy dtype for each field in output array or dataframe. Defaults to database table column types.

field_shapes: list, shape of each field value for geometry field types with variable length data. For example, for a 'path' database column with 2d points, the points of the path will be stored in a list object by default. If a field shape of '10' is specified, the points will be stored in a 2x10 float subarray (2 floats per point* 10 points max). A field shape of the form (x,y) should be specifed for types like multipolygon where x is the max number of polygons and y is the max length of each polygon (the size of the point is inferred).

close ()

Close connection to the database.

The PostgresAdapter object supports array slicing:

Read all records: adapter[:]

Read first 100 records: adapter[0:100]

Read last record: adapter[-1]

Read every other record: adapter[::2]

Adapter Properties

num_records (readonly)

Get number of records that will be returned from table or custom query.

num_fields (readonly)

Get number of fields in records that will be returned from table or custom query

field_names

Get/set names of fields in final array or dataframe. Field names can be set by specifying a list of names, or dict mapping of field number to field name. If names is a list, the length of list must match the number of fields in data set. If names is a dict, the field name from the database will be used if no name in dict is specified for that field.

field_types

Get/set field types in final array or dataframe. Field types can be set by specifying a list of NumPy dtypes, or a dict mapping of field number or name to field type. If types is a list, the length of list must match the number of fields in data set. If types is a dict, the field type from the database will be used if type is not specified in dict.

field_shapes

Get/set field shapes for variable length fields. Field shapes can be set by specifying a list of shape tuples (or a single integer if shape has one dimension), or a dict mapping of field number or name to field shape. If shapes is a list, the length of the list must match the number of fields in data set. A value of None or zero for a field, or an unspecified shape, means that an infinite length value will be allowed for field, and value will be stored in Python list object if field is PostgreSQL geometry type, or as Well Known Text string objects if field is PostGIS type.

Basic Usage

Create PostgresAdapter object for data source:

```
>>> import iopro
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=db_name user=user␣
↪table=table_name')
```

IOPro adapters use slicing to retrieve data. To retrieve records from the table or query, the standard NumPy slicing notation can be used:

```
>>> # read all records
>>> array = adapter[:]
```

```
>>> # read first ten records
>>> array = adapter[0:10]
```

```
>>> # read last record
>>> array = adapter[-1]
```

```
>>> # read every other record
>>> array = adapter[::2]
```

The PostgreSQL adapter has a few properties that we can use to find out more about our data. To get the number of records in our dataset:

```
>>> adapter.num_records
5
```

or the number of fields:

```
>>> adapter.num_fields
5
```

To find the names of each field:

```
>>> adapter.field_names
['field1', 'real', 'name', 'point2d', 'multipoint3d']
```

These names come from the names of the columns in the database and are used by default for the field names in the NumPy array result. These names can be changed by setting the field names property using a list of field names:

```
>>> adapter.field_names = ['field1', 'field2', 'field3', 'field4', 'field5']
>>> adapter[:].dtype
dtype([('field1', '<i4'), ('field2', '<f4'), ('field3', '<U10'), ('field4', '<f8', (2,
↪)), ('field5', 'O')])
```

Individual fields can also be set by using a dict, where the key is the field number and the value is the field name we want:

```
>>> adapter.field_names = {1: 'AAA'}
>>> adapter[:].dtype
dtype([('integer', '<i4'), ('AAA', '<f4'), ('string', '<U10'), ('point2d', '<f8', (2,
↪)), ('multipoint3d', 'O')])
```

To find out the NumPy dtype of each field:

```
>>> adapter.field_types
['i4', 'f4', 'U10', 'f8', 'O']
```

Similar to the field names property, the types property can be set using a list or dict to force a field to be cast to a specific type:

```
>>> adapter.field_types = {0: 'f4', 1: 'i4', 2: 'U3', 4: 'O'}
```

To filter the fields returned by passing a list of field names to the constructor:

```
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=db_name user=user',
                                     table='data',
                                     field_filter=['field1', 'field2'])
```

For fields like path or multipoint3d with a variable length, the adapter will return values as a list of tuples containing the float components of each point (if a PostgreSQL geometric type) or as string objects in Well Known Text format (if a PostGIS type). For improved performance, a field shape can be specified which will set the max dimensions of the field values. For example, a multipoint3d field can be set to have a maximum of two points so that each set of 3d points will be stored in a 2x3 subarray of floats:

```
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=db_name user=user'
                                    table='data',
                                    field_filter=['multipoint3d'],
                                    field_shapes={'multipoint3d': 2})
>>> adapter[:]
array([([[0.0, 1.0, 2.0], [3.0, 4.0, 5.0]],),
       ([[6.0, 7.0, 8.0], [9.0, 10.0, 11.0]],),
       ([[12.0, 13.0, 14.0], [15.0, 16.0, 17.0]],),
       ([[18.0, 19.0, 20.0], [21.0, 22.0, 23.0]],),
       ([[24.0, 25.0, 26.0], [27.0, 28.0, 29.0]],)],
      dtype=[('multipoint3d', '<f8', (2, 3))])
```

For more advanced queries, a custom select query can be passed to the constructor. Either a table name or a custom query can be passed to the constructor, but not both.

```
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=db_name user=user',
                                     query='select integer, string from data where␣
↪data.integer > 2')
>>> adapter[:]
```

Data can also be returned as a pandas dataframe using the adapter constructor's dataframe' argument:

```
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=iopro_tutorial user=jayvius
↪',
                                    table='data',
                                    dataframe=True)
```

To retrieve some PostGIS data that falls within a given bounding box:

```
>>> adapter = iopro.PostgresAdapter('host=localhost dbname=db_name user=user',
                                    query='select integer, point2d from data '
                                          'where data.point2d @ ST_MakeEnvelope(0, 0,␣
↪4, 4)')
```

Close database connection:

```
>>> adapter.close()
```

iopro.loadtxt

Load data from a text file.

Each row in the text file must have the same number of values.

Parameters

fname [file or str] File, filename, or generator to read. If the filename extension is `.gz` or `.bz2`, the file is first decompressed. Note that generators should return byte strings for Python 3k.

dtype [data-type, optional] Data-type of the resulting array; default: float. If this is a record data-type, the resulting array will be 1-dimensional, and each row will be interpreted as an element of the array. In this case, the number of columns used must match the number of fields in the data-type.

comments [str, optional] The character used to indicate the start of a comment; default: '#'.

delimiter [str, optional] The string used to separate values. By default, this is any whitespace.

converters [dict, optional] A dictionary mapping column number to a function that will convert that column to a float. E.g., if column 0 is a date string: `converters = {0: datestr2num}`. Converters can also be used to provide a default value for missing data (but see also *iopro.genfromtxt*): `converters = {3: lambda s: float(s.strip() or 0)}`. Default: None.

skiprows [int, optional] Skip the first *skiprows* lines; default: 0.

usecols [sequence, optional] Which columns to read, with 0 being the first. For example, `usecols = (1,4,5)` will extract the 2nd, 5th and 6th columns. The default, None, results in all columns being read.

unpack [bool, optional] If True, the returned array is transposed, so that arguments may be unpacked using `x, y, z = iopro.loadtxt(...)`. When used with a record data-type, arrays are returned for each field. Default is False.

ndmin [int, optional] The returned array will have at least *ndmin* dimensions. Otherwise mono-dimensional axes will be squeezed. Legal values: 0 (default), 1 or 2. .. versionadded:: 1.6.0

Returns

out [ndarray] Data read from the text file.

See Also

iopro.genfromtxt : Load data with missing values handled as specified.

Examples

simple parse of StringIO object data

```
>>> import iopro
>>> from io import StringIO   # StringIO behaves like a file object
>>> c = StringIO("0 1\\n2 3")
>>> iopro.loadtxt(c)
>>> array([[ 0.,  1.],
       [ 2.,  3.]])
```

set dtype of output array

```
>>> d = StringIO("M 21 72\\nF 35 58")
>>> iopro.loadtxt(d, dtype={'names': ('gender', 'age', 'weight'),
...                         'formats': ('S1', 'i4', 'f4')})
>>> array([('M', 21, 72.0), ('F', 35, 58.0)],
      dtype=[('gender', '|S1'), ('age', '<i4'), ('weight', '<f4')])
```

set delimiter and columns to parse

```
>>> c = StringIO("1,0,2\\n3,0,4")
>>> x, y = iopro.loadtxt(c, delimiter=',', usecols=(0, 2), unpack=True)
>>> x
>>> array([ 1.,  3.])
>>> y
>>> array([ 2.,  4.])
```

iopro.genfromtxt

Load data from a text file, with missing values handled as specified.

Each line past the first *skip_header* lines is split at the *delimiter* character, and characters following the *comments* character are discarded.

Parameters

fname [file or str] File, filename, or generator to read. If the filename extension is *.gz* or *.bz2*, the file is first decompressed. Note that generators must return byte strings in Python 3k.

dtype [dtype, optional] Data type of the resulting array. If None, the dtypes will be determined by the contents of each column, individually.

comments [str, optional] The character used to indicate the start of a comment. All the characters occurring on a line after a comment are discarded

delimiter [str, int, or sequence, optional] The string used to separate values. By default, any consecutive whitespaces act as delimiter. An integer or sequence of integers can also be provided as width(s) of each field.

skip_header [int, optional] The numbers of lines to skip at the beginning of the file.

skip_footer [int, optional] The numbers of lines to skip at the end of the file

converters [variable, optional] The set of functions that convert the data of a column to a value. The converters can also be used to provide a default value for missing data: `converters = {3: lambda s: float(s or 0)}`.

missing_values [variable, optional] The set of strings corresponding to missing data.

filling_values [variable, optional] The set of values to be used as default when the data are missing.

usecols [sequence, optional] Which columns to read, with 0 being the first. For example, `usecols = (1, 4, 5)` will extract the 2nd, 5th and 6th columns.

names [{None, True, str, sequence}, optional] If *names* is True, the field names are read from the first valid line after the first *skip_header* lines. If *names* is a sequence or a single-string of comma-separated names, the names will be used to define the field names in a structured dtype. If *names* is None, the names of the dtype fields will be used, if any.

excludelist [sequence, optional] A list of names to exclude. This list is appended to the default list ['return','file','print']. Excluded names are appended an underscore: for example, *file* would become *file_*.

deletechars [str, optional] A string combining invalid characters that must be deleted from the names.

defaultfmt [str, optional] A format used to define default field names, such as "f%i" or "f_%02i".

autostrip [bool, optional] Whether to automatically strip white spaces from the variables.

replace_space [char, optional] Character(s) used in replacement of white spaces in the variables names. By default, use a '_'.

case_sensitive [{True, False, 'upper', 'lower'}, optional] If True, field names are case sensitive. If False or 'upper', field names are converted to upper case. If 'lower', field names are converted to lower case.

unpack [bool, optional] If True, the returned array is transposed, so that arguments may be unpacked using `x, y, z = loadtxt(...)`

usemask [bool, optional] If True, return a masked array. If False, return a regular array.

invalid_raise [bool, optional] If True, an exception is raised if an inconsistency is detected in the number of columns. If False, a warning is emitted and the offending lines are skipped.

Returns

out [ndarray] Data read from the text file. If *usemask* is True, this is a masked array.

See Also

iopro.loadtxt : equivalent function when no data is missing.

Notes

- When spaces are used as delimiters, or when no delimiter has been given as input, there should not be any missing data between two fields.
- When the variables are named (either by a flexible dtype or with *names*, there must not be any header in the file (else a ValueError exception is raised).
- Individual values are not stripped of spaces by default. When using a custom converter, make sure the function does remove spaces.

Examples

```
>>> import iopro
>>> from io import StringIO
```

Comma delimited file with mixed dtype

```
>>> s = StringIO("1,1.3,abcde")
>>> data = iopro.genfromtxt(s, dtype=[('myint','i8'),('myfloat','f8'),
... ('mystring','S5')], delimiter=",")
>>> data
array((1, 1.3, 'abcde'),
      dtype=[('myint', '<i8'), ('myfloat', '<f8'), ('mystring', '|S5')])
```

Using dtype = None

```
>>> s.seek(0) # needed for StringIO example only
>>> data = iopro.genfromtxt(s, dtype=None,
... names = ['myint','myfloat','mystring'], delimiter=",")
>>> data
```

(continues on next page)

(continued from previous page)

```
array((1, 1.3, 'abcde'),
      dtype=[('myint', '<i8'), ('myfloat', '<f8'), ('mystring', '|S5')])
```

Specifying dtype and names

```
>>> s.seek(0)
>>> data = iopro.genfromtxt(s, dtype="i8,f8,S5",
... names=['myint','myfloat','mystring'], delimiter=",")
>>> data
array((1, 1.3, 'abcde'),
      dtype=[('myint', '<i8'), ('myfloat', '<f8'), ('mystring', '|S5')])
```

An example with fixed-width columns

```
>>> s = StringIO("11.3abcde")
>>> data = iopro.genfromtxt(s, dtype=None, names=['intvar','fltvar','strvar'],
...     delimiter=[1,3,5])
>>> data
array((1, 1.3, 'abcde'),
      dtype=[('intvar', '<i8'), ('fltvar', '<f8'), ('strvar', '|S5')])
```

Previous Versions

This documentation is provided for the use of our customers who have not yet upgraded to the current version.

NOTE: We recommend that users not use older versions of IOPro.

IOPro

Installation

Advanced TextAdapter

TextAdapter First Steps

iopro.pyodbc First Steps

iopro.pyodbc Enhanced Capabilities

iopro.pyodbc Cancelling Queries

TextAdapter

iopro.pyodbc

MongoAdapter

iopro.loadtxt

iopro.genfromtxt

IOPro Release Notes

IOPro END USER LICENSE AGREEMENT

1.3 Anaconda Cloud

What is Anaconda Cloud?

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, and conda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using. Cloud hosts hundreds of useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in, or even to have a Cloud account, to search for public packages, download and install them.

You can build new packages using the Anaconda Client command line interface (CLI), then manually or automatically upload the packages to Cloud to quickly share with others or access yourself from anywhere.

For developers, Cloud is designed to make software development, release and maintenance easy by providing broad package management support. Cloud allows for free public package hosting, as well as package channels, providing a flexible and scalable service for groups and organizations of all sizes.

Hosting of freely available packages always remains free for individuals and organizations hosting up to 3 GB of packages.

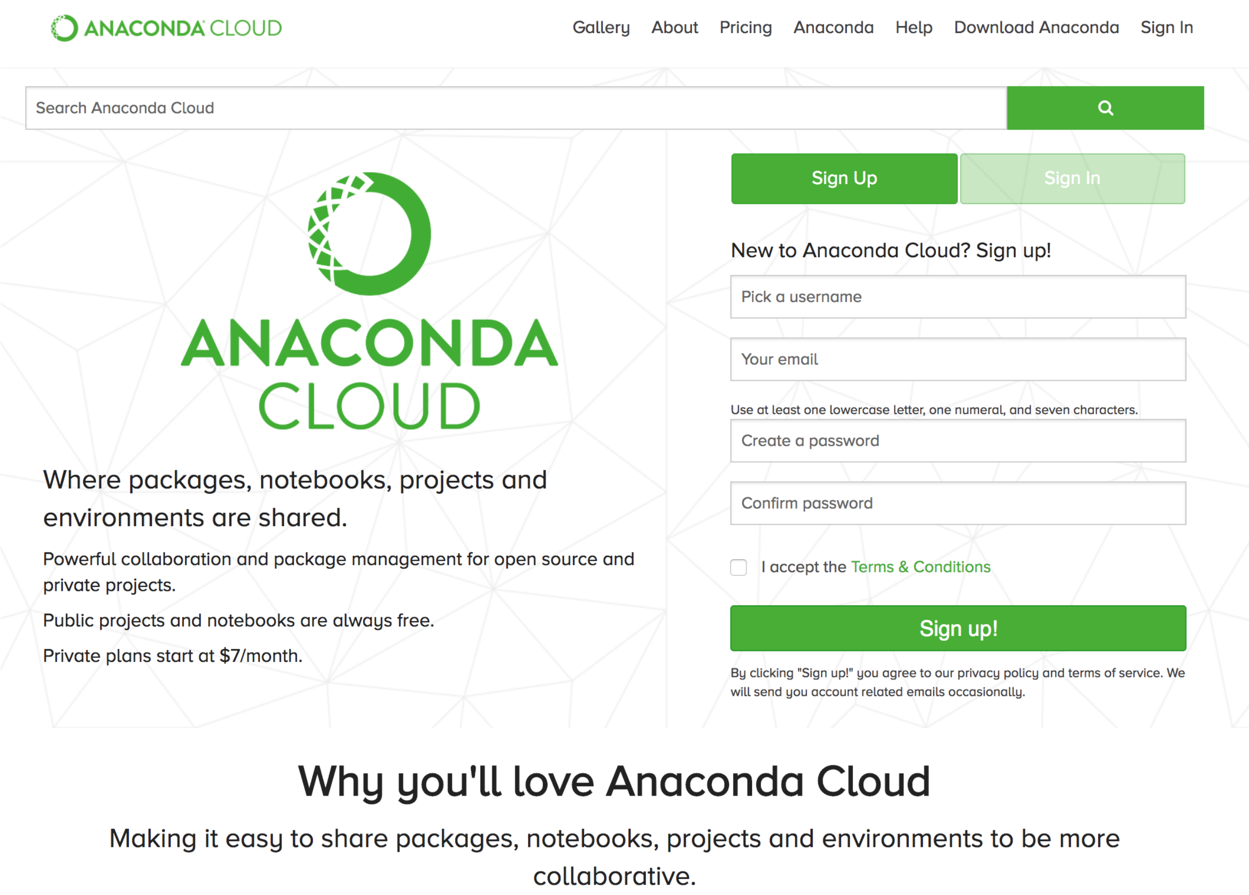
To use Cloud, you should first:

- Download Anaconda.

- Become familiar with using conda. A good place to start is the conda cheat sheet and the conda test drive.

1.3.1 User guide

Anaconda Cloud is a package management service that makes it easy to find, access, store and share public notebooks, environments, and conda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using.



Where packages, notebooks, projects and environments are shared.

Powerful collaboration and package management for open source and private projects.

Public projects and notebooks are always free.

Private plans start at $7/month.

New to Anaconda Cloud? Sign up!

Pick a username

Your email

Use at least one lowercase letter, one numeral, and seven characters.

Create a password

Confirm password

□ I accept the Terms & Conditions

Sign up!

By clicking "Sign up!" you agree to our privacy policy and terms of service. We will send you account related emails occasionally.

Why you'll love Anaconda Cloud

Making it easy to share packages, notebooks, projects and environments to be more collaborative.

To begin using Cloud, read *Getting started*, then the remaining sections of the user guide.

Getting started

- *Sharing notebooks*
- *Sharing environments*

Installing Anaconda Client

You can use Anaconda Client command line interface (CLI) to:

- Connect to and manage your Anaconda Cloud account.
- Upload *packages* you have created.
- Generate access *tokens* to allow access to private packages.

NOTE: Client is not necessary to search for and download packages.

This tool can be installed using an Anaconda Prompt or the Terminal application in three ways: with conda, with pip or with pip from source. We recommend using conda.

Option 1, conda:

```
conda install anaconda-client
```

Option 2, pip:

```
pip install anaconda-client
```

Option 3, installing with pip from source:

```
pip install git+https://github.com/Anaconda-Platform/anaconda-client
```

After installing, view the complete list of Client tasks with this command:

```
anaconda -h
```

Finding, downloading and installing packages

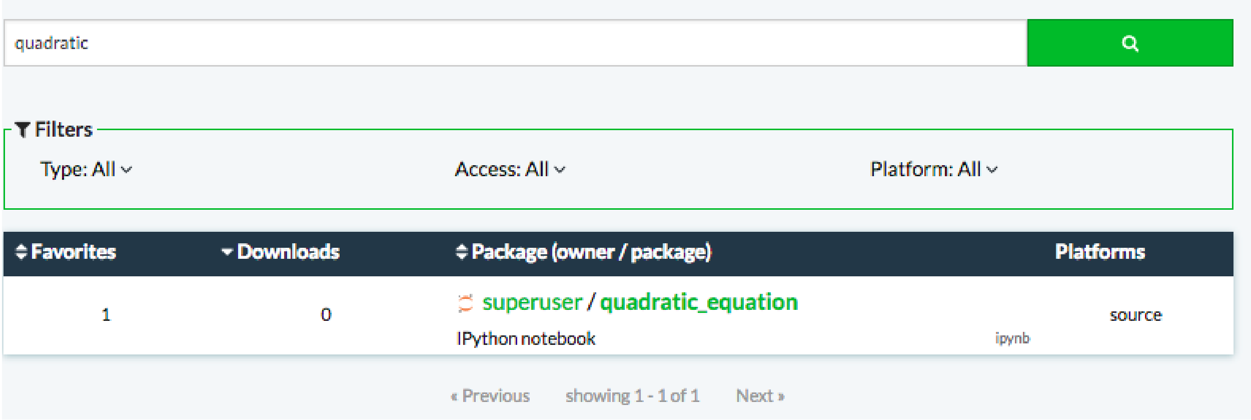
Searching for public packages

Cloud hosts hundreds of useful Python packages for a wide variety of applications. You do not need to be logged in, or even need a Cloud account, to search for public packages, download and install them. You need an account only to access *private packages* without a *token* or to share your packages with others.

To search for packages:

1. In the top Search box, type part or all of the name of a program you are searching for, and then press Enter.
2. Packages that match your search string are displayed. To see more information, click the package name.

For more information, see *Working with packages*.



Favorites	Downloads	Package (owner / package)		Platforms
1	0	superuser / quadratic_equation IPython notebook	ipynb	source

Refining your search results

You can filter search results using 3 filter controls:

- `Type`: All, conda only or PyPI only.
- `Access`: All, Public and/or Private–only available if you are logged in.
- `Platform`: All, source, linux-32, linux-64, linux-armv61, linux-armv71, linux-ppc641e, noarch, osx-32, osx-64, win-32, win-64.

`Source` packages are source code only, not yet built for any specific platform.

`Noarch` packages are built to work on all platforms.

Downloading and installing packages from Cloud

You can download and install packages using *Anaconda Navigator*, the graphical user interface for Anaconda. Advanced users may prefer a terminal window or command prompt.

Using Navigator

Navigator is automatically installed when you install Anaconda®.

To download and install a package into its own environment:

1. Start Navigator by clicking its program icon on your desktop or in your programs menu.
2. Sign Navigator into Cloud so you can search for packages marked as private. Click the top right Sign in to Anaconda Cloud button and type your Cloud username and password, then click the Login button.
3. On the **Environments** tab, in the far-right Search packages box, enter the name of the desired package.
4. In the list to the left of Channels, select either Not installed or All, then click the Search button.
5. Select the checkbox of the package you want to install, then click the Apply button.

For more information, see *Navigator*.

Using conda in a Terminal window or Anaconda Prompt

To download and install a package into its own environment:

1. Locate a package on Cloud that you want to download, then click on the package name.
2. A detail page displays specific installation instructions for the current operating system. Copy and paste the full command into your terminal window.

For example, the command could be structured as:

```
conda install -c username packagename
```

NOTE: For the following examples to work, you need to have conda downloaded and installed.

TIP: Conda expands `username` to a URL such as https://anaconda.org/username, based on the settings in the `.condarc` file.

Building and uploading packages

Open a Cloud account to upload packages or to access private packages without a token.

To build and upload packages, install the Client command line interface (CLI). For more information, see *Installing Anaconda Client*.

Use the Terminal window or an Anaconda Prompt to perform the following steps.

1. Install Anaconda Client:

   ```
   conda install anaconda-client
   ```

2. Log into your Cloud account:

   ```
   anaconda login
   ```

   At the prompt, enter your Cloud username and password.

3. Choose the package you would like to build. For this example, download our public test package:

   ```
   git clone https://github.com/Anaconda-Platform/anaconda-client
   cd anaconda-client/example-packages/conda/
   ```

4. To build your test package, first install `conda-build` and turn off automatic Client uploading, then run the `conda build` command:

   ```
   conda install conda-build
   conda config --set anaconda_upload no
   conda build .
   ```

5. Find the path to where the newly-built file was placed so you can use it in the next step:

   ```
   conda build . --output
   ```

6. Upload your newly-built test package to your Cloud account:

   ```
   anaconda login
   anaconda upload /your/path/conda-package.tar.bz2
   ```

   NOTE: Replace `/your/path/` with the actual path that you found in the previous step.

For more information, see *conda packages*.

Sharing notebooks

To share a Jupyter notebook–formerly IPython notebook–on Cloud:

1. To Upload your notebook to Cloud, open the Terminal or an Anaconda Prompt and enter:

   ```
   anaconda upload my-notebook.ipynb
   ```

   NOTE: Replace `my-notebook` with the actual name of your notebook.

2. You can see an HTML version of your notebook stored at:

   ```
   http://notebooks.anaconda.org/<USERNAME>/my-notebook
   ```

   NOTE: Replace `<USERNAME>` with your username, and `my-notebook` with the actual name of your notebook.

3. Anyone who has access to Cloud can download your notebook. To download the notebook, open the Terminal or an Anaconda Prompt and enter:

   ```
   anaconda download username/my-notebook
   ```

   NOTE: Replace `username` with your username, and `my-notebook` with the actual name of your notebook.

Sharing environments

To share an environment on Cloud:

1. See the conda user guide to create and save a conda environment. Open the Terminal or an Anaconda Prompt and enter:

   ```
   conda env export -n my-environment -f my-environment.yml
   ```

   NOTE: Replace `my-environment` with the actual name of your environment.

2. Upload it to Cloud either using the web interface or the `anaconda upload` command.

   - Using the web interface:

     Go to `https://anaconda.org/USERNAME/environments`.

     NOTE: Replace `USERNAME` with your username.

     In the top right corner use the Upload button to upload your environment.

   - Using the `anaconda upload` command from the Terminal window or an Anaconda Prompt:

     ```
     anaconda upload my-environment.yml
     ```

     NOTE: Replace `my-environment` with the actual name of your environment.

3. You can see a list of your uploaded environments at:

   ```
   http://envs.anaconda.org/<USERNAME>
   ```

   NOTE: Replace `<USERNAME>` with your username.

4. Anyone can download and install your environment from Cloud.

- Using the web interface:

  Go to `https://anaconda.org/USERNAME/environments`.

  NOTE: Replace `USERNAME` with the username.

  Select the environment, click the Files tab, and under the Names field click the file to download.

- Using the Terminal or an Anaconda Prompt:

  ```
  conda env create user/my-environment
  source activate my-environment
  ```

NOTE: Replace `my-environment` with the actual name of your environment.

How to. . .

- *Use packages*
- *Use the Anaconda Client CLI*
- *Build packages*

Use packages

Finding a package

In your browser, you can search Anaconda Cloud for packages by package name. From the top navigation bar of any page, enter the package name in the search box. You can filter your searches to specify only conda or PyPI packages, and you can sort results by number of favorites or number of downloads by clicking the search results column heading.

Downloading and installing a package from Cloud

To install a conda package, in your Terminal window or Anaconda Prompt run:

```
conda install -c username packagename
```

Conda expands `username` to a URL such as https://anaconda.org/username or https://conda.anaconda.org/username based on the settings in the `.condarc` file.

NOTE: Replace `username` with your username, and `packagename` with the actual name of the package.

Downloading and installing a PyPI package from Cloud

To install a PyPI package, in your Terminal window or Anaconda Prompt run:

```
pip install --index-url pypi.anaconda.org/USERNAME/simple packagename
```

NOTE: Replace `USERNAME` with your username, and `packagename` with the actual name of the package.

Use the Anaconda Client CLI

Installing Client

For installation and setup instructions, see *Install Anaconda Client*.

Finding my Client login credentials

Your credentials for Client are those you used to create an account on Cloud.

To get help:

1. Go to https://anaconda.org.
2. Select the **Sign In** tab.
3. Click either `I forgot my password` or `I forgot my username`.

Logging into Client

After you have downloaded and configured Client, open a Terminal window or an Anaconda Prompt and run:

```
anaconda login
```

Displaying a list of Client commands

From a Terminal window or an Anaconda Prompt, run:

```
anaconda --help
```

Finding out more about a Client command

From a Terminal window or an Anaconda Prompt, run:

```
anaconda COMMANDNAME -h
```

NOTE: Replace `COMMANDNAME` with the name of the command about which you want more information.

Listing all available Client configuration files

From a Terminal window or an Anaconda Prompt, run:

```
anaconda config --files
```

Listing all of your Client configuration variables

From a Terminal window or an Anaconda Prompt, run:

```
anaconda config --show
```

Finding out more about Client

You can learn more about Client using the help command, documentation or community support email group.

Build packages

Building and uploading a package

For a quick example, see *Building and uploading packages* in *Getting started*.

Testing a built package

In your Terminal window or Anaconda Prompt run:

```
conda create --use-local -n test PACKAGENAME
```

Specify the `--use-local` option.

NOTE: Replace PACKAGENAME with the actual name of the package.

Uploading a package to Cloud

In a Terminal window or Anaconda Prompt, run:

```
anaconda upload PACKAGENAME
```

NOTE: Replace `PACKAGENAME` with the actual name of the package.

Finding help for uploading packages

You can obtain a complete list of upload options, including:

- Package channel.
- Label.
- Availability to other users.
- Metadata.

To list the options, in a Terminal window or Anaconda Prompt run:

```
anaconda upload -h
```

Tutorial

Using labels in the development cycle

Anaconda Cloud *labels* can be used to facilitate a development cycle and organize the code that is in development, in testing and in production, without affecting non-development users. With labels you can upload a file to a specific label, so only users who put that label in the URL they search are able to find it.

Using Anaconda Client, *package* developers can create additional labels such as development `labels/dev`, test `labels/test` or other labels that are searched only if the user specifies the label. The following search examples use a *namespace* of `travis`:

- https://anaconda.org/travis/labels/main – the label searched by default.
- https://anaconda.org/travis – same as default label with `main` implicit.
- https://anaconda.org/travis/labels/dev – contains the packages in development.
- https://anaconda.org/travis/labels/test – contains packages ready to test.
- https://anaconda.org/travis/labels/any-custom-label – any label you want to use.

In this example, we show you how to use a `test` label, so that you can upload files without affecting your production-quality packages. Without a `--label` argument the default label is `main`.

Use the Terminal window or an Anaconda Prompt to perform the following steps:

1. Let us start with a conda package. If you do not have one, use our example conda package. Before you build the package, edit the version in the `meta.yaml` file in `anaconda-client/example-packages/conda/` to be `2.0`:

```
git clone https://github.com/Anaconda-Platform/anaconda-client/
cd anaconda-client/example-packages/conda/
nano meta.yaml # Bump version to 2.0
conda config --set anaconda_upload no
conda build .
```

2. Upload your test package to Cloud using the Client *upload* command.

   Adding the `--label` option tells Cloud to make the upload visible only to users who specify that label:

```
anaconda upload /path/to/conda-package-2.0.tar.bz2 --label test
```

   NOTE: Replace `/path/to/` with the actual path where you stored the package.

3. You now can see that even when you search conda `main`, you do not see the `2.0` version of the test package. This is because you need to tell conda to look for your new `test` label.

4. The `--override` argument tells conda not to use any channels in your `~/.condarc` file.

   No `2.0` results:

```
conda search --override -c USERNAME conda-package
```

   NOTE: Replace USERNAME with your username.

   Your `2.0` package is here:

```
conda search --override -c USERNAME/label/test conda-package
```

   NOTE: Replace `USERNAME` with your username.

5. You can give the label `USERNAME/label/test` to your testers.

   NOTE: Replace `USERNAME` with your username.

6. Once they finish testing, you may then want to copy the `test` packages back to your `main` label:

   ```
   anaconda label --copy test main
   ```

   You can also manage your package labels from your dashboard: `https://anaconda.org/USERNAME/conda-package`.

   Your version `2.0` is now in `main`:

   ```
   conda search --override -c USERNAME conda-package
   ```

   NOTE: Replace `USERNAME` with your username.

If you use `anaconda-client` 1.7 or higher, you can use `anaconda move` to move packages from one label to another:

```
anaconda move --from-label OLD --to-label NEW SPEC
```

Replace OLD with the old label, NEW with the new label, and SPEC with the package to move. SPEC can be either "user/package/version/file", or "user/package/version" in which case it moves all files in that version.

Tasks

Working with accounts

Personal accounts

Overview

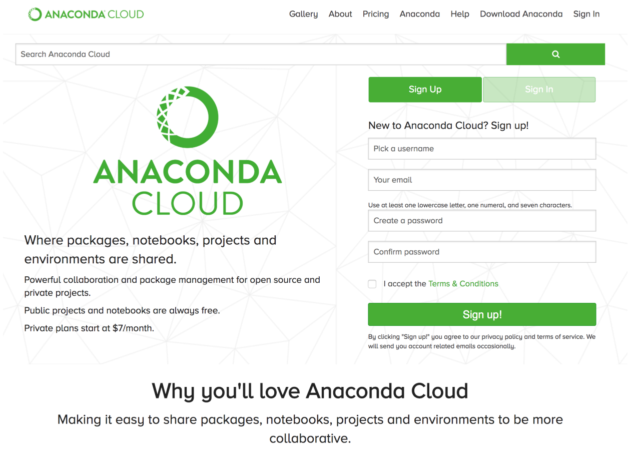
This section explains how to create a free, personal account, log in and out, and access the settings and features of different types of accounts.

Creating a free account

All Anaconda Cloud users can find, download and use packages without having a Cloud account.

However, you need to create a Cloud account to:

- Author packages.
- Upload packages, notebooks and environments.
- Access shared, private packages.
- Create organizations.



Where packages, notebooks, projects and environments are shared.

Powerful collaboration and package management for open source and private projects.

Public projects and notebooks are always free.

Private plans start at $7/month.

New to Anaconda Cloud? Sign up!

Use at least one lowercase letter, one numeral, and seven characters.

□ I accept the Terms & Conditions

By clicking "Sign up!" you agree to our privacy policy and terms of service. We will send you account related emails occasionally.

Why you'll love Anaconda Cloud

Making it easy to share packages, notebooks, projects and environments to be more collaborative.

To sign up for a free Cloud account:

1. In a browser, go to Anaconda Cloud.
2. Make sure the **Sign Up** tab is active.

   NOTE: There is also a **Sign In** tab for existing users.
3. Select a username.
4. Enter your email address.

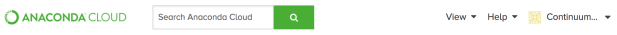
   NOTE: Users who register with an .edu email are granted some additional features.
5. Create a password.

   NOTE: The password must be at least 7 characters long.
6. Enter the password again to confirm it.
7. Read and accept the Terms and Conditions.
8. Click the Sign up button.

   The system creates your free account, logs you in and displays your personal dashboard.

TIP: Cloud displays your profile photo if the email address you used to register on Cloud is associated with a Gravatar account. To associate your email address or to change your Gravatar profile photo, go to gravatar.com.

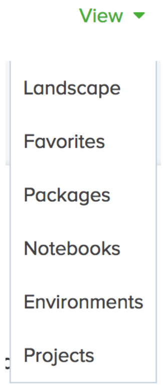
When you are logged into Cloud, the Profile list appears at the top right of every page. This indicates the name of the currently active user or organization.



The View menu contains these options:

- `Landscape`: Your home page.

- `Favorites`: Packages from other users you have starred.
- `Packages`: Only packages you have created.
- `Notebooks`: Only notebooks you have created.
- `Environments`: Only environments you have created.
- `Projects`: Only projects you have uploaded.



Packages, notebooks and environments that you have created with this account appear on your dashboard. For more information, see *packages*.

Resetting your password

The **Sign In** tab provides two links to help regain access to your account:

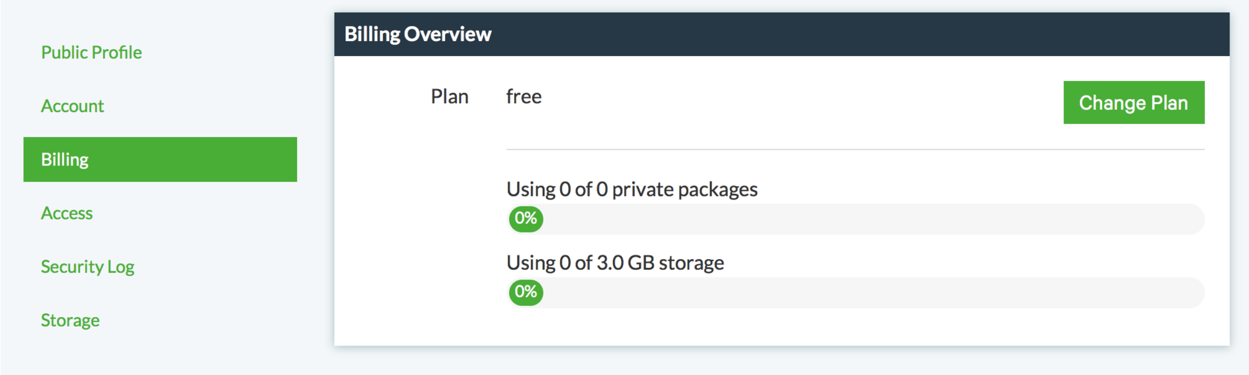
- `I forgot my username.` Click this link to have the username emailed to the email address of record.
- `I forgot my password.` Click this link to have a reset password link sent to the email address of record.

NOTE: The reset password link expires within 24 hours. If you no longer have access to the email account, you can create a new account or email support@anaconda.com for assistance.

Upgrading or downgrading your plan

To change a Cloud plan:

1. Log in to the Cloud account that you want to change.
2. At the top right of the Cloud interface, in the Profile list, select Settings.
3. Select the Billing option.
4. Click the Change Plan button.



5. Select the desired plan.

   If you are moving from a free to a charged plan, enter your credit card information, and then click the OK button.

NOTE: If you need more private packages or storage space than is included in a personal plan, contact Anaconda so we can customize a plan for you.

NOTE: If you need assistance with billing questions, you can contact Anaconda.

Creating access tokens

The best way to manage access or make packages private is to create *organizations* or *groups*, which allow you to set separate permissions per package, notebook or environment.

You can also control access with the *token* system. You can use tokens to control access to private repositories, collections or packages on Cloud. Additionally, the degree of access a token grants is completely configurable at the time of generation. You can generate multiple tokens to control which groups of users have access to certain features if they have the appropriate token.

Generating tokens

You can generate tokens using the Web UI or Anaconda Client.

NOTE: By default, tokens expire after one year.

To generate a token using the Web UI:

1. Navigate to:

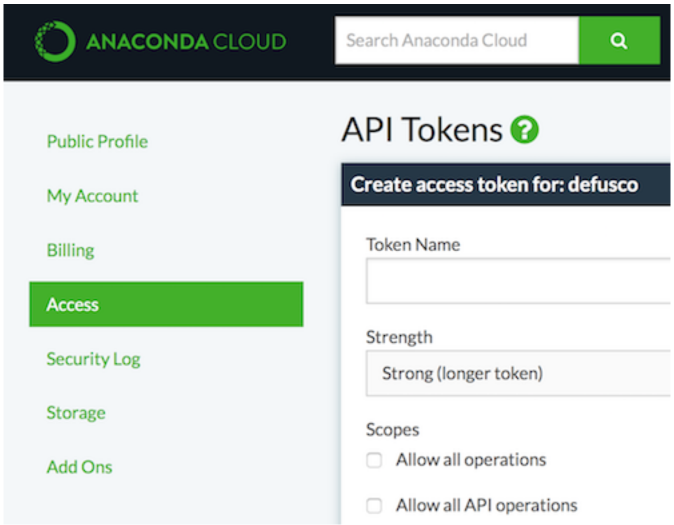
```
https://anaconda.org/<channel>/settings/access
```

   NOTE: Replace `<channel>` with the actual channel name.

2. From here you can select a name for the token and set the required scopes.

To generate a token with Client:

1. In your Terminal window or Anaconda Prompt run:



API Tokens

Create access token for: defusco

Token Name

Strength

Strong (longer token)

Scopes

□ Allow all operations

□ Allow all API operations

```
anaconda auth --create --name YOUR-TOKEN-NAME --scopes 'repos conda:download'
```

NOTE: Replace `YOUR-TOKEN-NAME` with the token name you want.

2. This generates a random alphanumeric token string, which you can then distribute to fellow Cloud users to enable them to download a package that you have marked private. The token produced in this example provides access to download any of your private conda repositories.

3. It can be enabled with the `conda config` command. In your Terminal window or Anaconda Prompt run:

```
   conda config --add channels https://conda.anaconda.org/t/<token>/<channel>

Add a channel with a token and a label::

   conda config --add channels https://conda.anaconda.org/t/<token>/<channel>/
↪label/<labelname>


NOTE: Replace ``<token>`` with your token string,``<channel>`` with the channel␣
↪name,
``<labelname>`` with the label name used in :ref:`cloud-accounts-generate`.
```

NOTE: If you lose the random alphanumeric string, you need to *revoke the token* and create a new one.

Scopes are provided as a space-separated, quoted list. The available scopes are:

- `all`: Allow all operations.
- `api`: Allow all API operations.
- `api:modify-group`: Allow addition and modification of groups.
- `api:read`: Allow read access to the API site.
- `api:write`: Allow write access to the API site.
- `conda`: Allow all operations on Conda repositories.
- `conda:download`: Allow private downloads from Conda repositories.
- `pypi`: Allow all operations on PyPI repositories.
- `pypi:download`: Allow private downloads from PyPI repositories.

- `pypi:upload`: Allow uploads to PyPI repositories.
- `repos`: Allow access to all package repositories.

NOTE: Tokens provide access to all packages within a specified channel. Separate permissions per package, notebook or environment may be better handled with *organizations* and *groups*.

Revoking tokens

You can revoke tokens using the Web UI or Anaconda Client.

To revoke a token using the Web UI, navigate to the following URL:

```
https://anaconda.org/<channel>/settings/access
```

NOTE: Replace `<channel>` with the actual channel name.

To revoke a token using Client, in your Terminal window or Anaconda Prompt run:

```
anaconda auth -r YOUR-TOKEN-NAME
```

NOTE: Replace `YOUR-TOKEN-NAME` with the token name you want to revoke.

Organization accounts

Overview

You can create a *Cloud organization* to:

- Share packages, environments or notebooks under an organization's account rather than your personal account.
- Assign multiple account administrators.
- Assign different access permissions to groups of users and customize per-package access by group.
- Host more, larger packages.

Working with organizations

Organizations enable you to maintain group-owned repositories and set access levels for various users in the group.

Creating an organization

To create an organization:

1. Log in to Anaconda Cloud.
2. In the Tools list, select Organizations, and then scroll to the bottom of the page.
3. Supply an organization name.

   NOTE: Organization names cannot include spaces or special characters.

4. Supply an email address, then click the Create Organization button.

   The system displays the dashboard for the new organization.

As the creator and owner of an organization, you have automatic administrative access to this organization and any packages associated with the organization.

In the Tools list, the Group Membership option shows a list of all organizations to which you belong.

Deleting an organization

To delete an organization you administer and erase all data associated with it:

1. At the top right of the Cloud interface, in the Profile list, select Switch To.
2. Select the organization you want to delete.
3. In the Profile list, select Settings.
4. Select the Account option. You may be asked to verify your password.
5. In the Delete Account section, click the Delete button.

   A confirmation page then requests that you provide the full name of the organization.

Customizing users and groups

To add, remove, or edit group and user access for an organization you administer:

1. At the top right of the Cloud interface, in the Profile list, select the desired organization profile.
2. In the Tools list, select Groups.
3. You can review and edit the current group and user access for an organization, as well as add new groups and users at this address: https://anaconda.org/<OrgName>/groups/

   NOTE: Replace `<OrgName>` with the organization name.

NOTE: Users receive a dashboard notification when you add them to an organization.

Uploading packages to an organization

To upload a package to an organization, use the `-u/--user` option and in the Terminal window or an Anaconda Prompt, run:

```
anaconda upload --user ORGANIZATION package.tar.bz2
```

NOTE: Replace `ORGANIZATION` with the organization name, and `package.tar.bz2` with the actual name of the package.

NOTE: Only the co-owners of an organization may upload packages to the organization.

Creating groups for differing access levels

Within an organization, you can create a group to customize access for a group of users. There are four types of permissions you can grant:

- Read: Provides access only to the packages. Users in a read-only group can see the list of files within a package and can install them through conda.
- Read-Write: Provides access to upload new versions of an existing package, delete files of a package and manage the individual labels of files. Users in a read-write group cannot upload files for non-existing packages, cannot delete or transfer the package and cannot lock and unlock labels.
- Admin: Provides access do everything except uploading new packages (admin users can still upload to an existing package) and lock and unlock labels.
- Owners: The user has full control over the organization and group.

To create a group and set access levels:

1. Click the +New Group button.
2. Give the group a name, and assign the desired permissions– Read-Only, Read-Write or Administration.
3. Add the desired members by username in the Members box.
4. Add the packages which this group can access in the Packages box.
5. Click the Save Group button.

Working with packages

Overview

All files uploaded to Anaconda Cloud are stored in packages. Each Cloud package is visible at its own unique URL based on the name of the user who owns the package and the name of the package.

Users can create a Cloud package and then upload files into it.

For more information, see *package*.

Namespaces

A namespace is the part of Cloud where a user or organization may host packages. For example, the user namespace https://anaconda.org/travis contains packages that were uploaded and shared by a user named `travis`.

For more information, see *namespace*.

Labels

A label is part of the URLs on Cloud where conda looks for packages.

Each file within a package may be tagged with one or more labels, or not tagged at all to accept the default label of `main`.

For more information, see *label*.

Using package managers

Cloud supports two package managers, *conda* and *PyPI*. To work with conda or PyPI packages, you must use their corresponding subdomains:

- To install conda packages from the user `travis`, use the repository URL `https://conda.anaconda.org/travis`
- To install PyPI packages from the user `travis`, use the repository URL `https://pypi.anaconda.org/travis`

Conda packages

Uploading conda packages

This example shows how to build and upload a conda package to Cloud using `conda build`.

Use the Terminal window or an Anaconda Prompt to perform the following steps:

1. Before you start, install `anaconda-client` and `conda-build`:

   ```
   conda install anaconda-client conda-build
   ```

2. Choose the repository for which you would like to build the package. In this example, we use a simple public conda test package:

   ```
   git clone https://github.com/Anaconda-Platform/anaconda-client
   cd anaconda-client/example-packages/conda/
   ```

In this directory, there are two required files, meta.yaml and build.sh.

macOS and Linux systems are Unix systems. Packages built for Unix systems require a `build.sh` file, packages built for Windows require a `bld.bat` file, and packages built for both Windows and Unix systems require both a `build.sh` file and a `bld.bat` file. All packages require a `meta.yaml` file.

3. To build the package, turn off automatic Client uploading and then run the `conda build` command:

```
conda config --set anaconda_upload no
conda build .
```

   All packages built in this way are placed in a subdirectory of *Anaconda's* `conda-bld` directory.

4. You can check where the resulting file was placed with the `--output` option:

```
conda build . --output
```

5. You can upload the test package to Cloud with the *anaconda upload* command:

```
anaconda login
anaconda upload /path/to/conda-package.tar.bz2
```

   NOTE: Replace `/path/to/` with the actual path where you stored the package.

For more information on conda's overall build framework, you may also want to read the articles Building conda packages and Tutorials on conda build.

Installing conda packages

You can install conda packages from Cloud by adding channels to your conda configuration.

Use the Terminal window or an Anaconda Prompt to perform the following steps:

1. Because conda knows how to interact with Cloud, specifying the channel `sean` translates to https://anaconda.org/sean:

```
conda config --add channels sean
```

2. You can now install public conda packages from Sean's Cloud account. Try installing the `testci` package at https://anaconda.org/sean/testci:

```
conda install testci
```

3. You can install a package from a channel with a token and a label:

```
conda install -c https://conda.anaconda.org/t/<token>/<channel>/label/<labelname>
↪<package>
```

   NOTE: Replace `<token>` with the provided token,‘‘<channel>‘‘ with a user channel, `<labelname>` with the label name and <package> with the package name you want to install.

PyPI packages

Uploading PyPI packages

We can test PyPI package uploading with a small public example package saved in the anaconda-client repository.

Use the Terminal window or an Anaconda Prompt to perform the following steps:

1. Begin by cloning the repository from the command line:

   ```
   git clone git@github.com:Anaconda-Platform/anaconda-client.git
   cd anaconda-client/example-packages/pypi/
   ```

2. You can now create your PyPI package with the `setup.py` script:

   ```
   python setup.py sdist
   ```

3. The package has now been built as a source tarball and is ready to be uploaded:

   ```
   anaconda upload dist/*.tar.gz
   ```

   Your package is now available at `http://anaconda.org/USERNAME/PACKAGE`.

   NOTE: Replace `USERNAME` with your username, and `PACKAGE` with the package name.

Installing PyPI packages

The best way to install a PyPI package is using `pip`. For the following command, we use the package we authored in the examples above. In your Terminal window or an Anaconda Prompt, run:

```
pip install --extra-index-url https://pypi.anaconda.org/USERNAME/simple pypi-test-
↪package
```

Installing private PyPI packages

The best way to manage access or make PyPI and other packages private is to create *organizations* or *groups*, which allow you to set separate permissions per package, notebook or environment.

You can also control access with the token system. All Cloud URLs can be prefixed with `/t/$TOKEN` to access private packages.

In your Terminal window or an Anaconda Prompt, run:

```
TOKEN=$(anaconda auth --create --name YOUR-TOKEN-NAME)
pip install --index-url https://pypi.anaconda.org/t/$TOKEN/USERNAME/simple test-
↪package
```

NOTE: Replace `YOUR-TOKEN-NAME` with the name of the token you created, `USERNAME` with your username and `simple test-package` with the actual test-package name.

Uploading packages

1. To upload package files to Cloud, use the Terminal window or an Anaconda Prompt and the *upload* command:

   ```
   anaconda login
   anaconda upload PACKAGENAME
   ```

   NOTE: Replace `PACKAGENAME` with the actual package name.

   Cloud automatically detects packages and notebooks, package or notebook types, and their versions.

2. Your package is now available at:

```
https://anaconda.org/<USERNAME>/<PACKAGENAME>
```

NOTE: Replace `<USERNAME>` with your username, and `<PACKAGENAME>` with the package name.

3. Your package also can be downloaded by anyone using Client from the Terminal window or an Anaconda Prompt:

```
anaconda download USERNAME/PACKAGENAME
```

NOTE: Replace `<USERNAME>` with your username, and `<PACKAGENAME>` with the package name.

Using private packages

By default, all packages, notebooks and environments uploaded to Cloud are accessible to anyone who has access to the repository.

Packages uploaded to your user channel on Cloud can be marked as private using the Web UI:

1. Select the desired package.
2. Select the **Settings** tab.
3. Select Admin in the sidebar.
4. Alternatively, you can reach this page with the following URL:

```
https://anaconda.org/<username>/<package>/settings/admin
```

NOTE: Replace `<username>` with your username, and `<package>` with the package name.

NOTE: Jupyter notebooks and conda environments can also be marked private using this procedure and URL.

NOTE: Other Cloud users may access your private packages either with tokens or by logging in.

Private packages with tokens

To make your private packages available to be accessed with tokens:

1. First create an access *token* that includes the following scope for Client:

```
conda:download
```

Or, in the Web UI with:

```
Allow private downloads from conda repositories
```

The token is a random alphanumeric string and this is used to install a package or add a channel from which you want to install private packages.

2. Using the provided token, a user channel can be added to `config` from the Terminal window or an Anaconda Prompt with:

```
conda config --add channels https://conda.anaconda.org/t/<token>/<channel>
```

NOTE: Replace `<token>` with the provided token, and `<channel>` with a user channel.

3. The token can also be used to install packages without first adding the channel. In the Terminal window or an Anaconda Prompt, run:

```
  conda install -c https://conda.anaconda.org/t/<token>/<channel> <package>

To install a package from a channel using token and label name::

  conda install -c https://conda.anaconda.org/t/<token>/<channel>/label/
↪<labelname> <package>
```

NOTE: Replace `<token>` with the provided token, `<channel>` with a user channel, `<labelname>` with the label name and `<package>` with a package name you want to install.

4. Private PyPI packages can also be installed in the Web UI:

```
https://pypi.anaconda.org/t/<token>/<channel>
```

NOTE: Replace `<token>` with the provided token, and `<channel>` with a user channel.

Private packages with login

To make your private packages available to users who have logged in:

1. Create an *organization*.
2. Create a group in that organization, which may be a read-only group.
3. Add to the group the users that you want to grant access to.
4. Upload the package to the organization, or transfer an existing package to the organization.

After you grant them access, other users can download and install your package using the Web UI or Client.

To download a package:

1. In a browser, navigate to the desired channel.
2. If the organization name is `OrgName` and the package name is `conda-package`, use these commands in the Terminal window or an Anaconda Prompt:

```
conda install anaconda-client
anaconda login
conda install -c OrgName conda-package
```

Or instead:

```
conda install anaconda-client
anaconda login
conda install -c https://conda.anaconda.org/OrgName conda-package
```

Removing a previous version of a package

To remove a previous version of one of your packages from Cloud:

1. Select the package name.
2. Select the **Files** tab.
3. Select the checkbox to the left of the version you want to remove.
4. In the **Actions** menu, select Remove.

You may instead use the Terminal window or an Anaconda Prompt:

1. Run:

   ```
   anaconda remove jsmith/testpack/0.2
   ```

   NOTE: Replace `jsmith` with your username, `testpack` with the package name and `0.2` with the desired version.

2. You can now see the change on your profile page:

   ```
   https://anaconda.org/<USERNAME>/<PACKAGE>
   ```

   NOTE: Replace `<USERNAME>` with your username, and `<PACKAGE>` with the package name.

Adding a collaborator to a package

You can add other users that are not part of an organization to collaborate on your packages. You will need the usernames of the other users.

1. From your dashboard, select the package by clicking on its name.
2. To display the package settings, select the Settings option.
3. To display the current collaborators, select the Collaborators option.
4. Type the username of the person you want to add as a collaborator, and then click the Add button.

NOTE: All collaborators are given full read/write permissions to the package, even private packages.

Removing a collaborator from a package

To revoke package access previously granted to a collaborator:

1. From your dashboard select the package by clicking on its name.
2. To display the package settings, select the Settings option.
3. To display the current collaborators, select the Collaborators option.
4. Click the red X button next to a collaborator to revoke their access.

Transferring a package to a new owner

By default, when you create or add packages, they are attached to your individual profile. You can transfer ownership to another owner account you control, such as an organization profile you manage.

To transfer a package to a new owner:

1. From your dashboard–or the dashboard of an organization you administer–select the package for which you want to transfer ownership.

   The system displays options for that package.

2. To display the package settings, select the Settings option.
3. Select the Admin option.
4. Under Transfer this package to a new owner, click the Transfer button.
5. Select the organization name for the new owner.
6. Click the Transfer Ownership button.

Copying a package

To copy a package from the channel `conda-forge` to a personal channel such as `jsmith`:

```
anaconda copy conda-forge/glueviz/0.10.4 --to-owner jsmith
```

`conda-forge/glueviz/0.10.4` is a "spec" and can match either of two formats: `user/package/version` or `user/package/version/filename`.

Previously labels were called "channels", and the `anaconda copy` command has deprecated options `from-channel` and `to-channel` that expect to operate on labels. These deprecated options should not be used. If you attempt to run `anaconda copy --from-channel conda-forge --to-channel jsmith glueviz`, you will receive an error that `Label conda-forge does not exist`.

Deleting a package

To delete a package from Cloud, including all of its versions:

1. Select the package name.
2. Select the Settings option.
3. In the left sidebar, select Admin.
4. Click Delete.

You may instead use the Terminal window or an Anaconda Prompt:

1. Run:

   ```
   anaconda remove jsmith/testpak
   ```

   NOTE: Replace `jsmith` with your user name, and `testpak` with the package name.

2. You can now see the change on your profile page:

   ```
   https://anaconda.org/<USERNAME>
   ```

   NOTE: Replace `<USERNAME>` with your username.

Working with notebooks

To begin working with Jupyter Notebooks, see the Official Jupyter Notebook documentation.

To share a Jupyter notebook on Cloud:

1. Save a `notebook`.
2. Download and check out the handy `JupyterLab and Jupyter Notebook cheat sheet` to create and save a notebook.
3. To upload your notebook to Cloud, open Anaconda Prompt or Terminal and enter:

   ```
   anaconda upload my-notebook.ipynb
   ```

   NOTE: Replace `my-notebook` with the actual name of your notebook.

4. You can see an HTML version of your notebook stored at:

```
http://notebooks.anaconda.org/<USERNAME>/my-notebook
```

NOTE: Replace `<USERNAME>` with your username, and `my-notebook` with the actual name of your notebook.

5. Anyone who has access to Cloud can download your notebook. To download the notebook, open Anaconda Prompt or Terminal and enter:

```
anaconda download username/my-notebook
```

NOTE: Replace `username` with your username, and `my-notebook` with the actual name of your notebook.

Working with environments

To share an environment on Anaconda Cloud:

1. See the conda user guide to create and save a conda environment. Open the Terminal or an Anaconda Prompt and enter:

```
conda env export -n my-environment -f my-environment.yml
```

NOTE: Replace `my-environment` with the actual name of your environment.

2. Upload it to Cloud either using the web interface or the `anaconda upload` command.

   - Using the web interface:

     Go to `https://anaconda.org/USERNAME/environments`.

     NOTE: Replace `USERNAME` with your username.

     In the top right corner use the Upload button to upload your environment.

   - Using the `anaconda upload` command from the Terminal or an Anaconda Prompt:

```
anaconda upload my-environment.yml
```

     NOTE: Replace `my-environment` with the actual name of your environment.

3. You can see a list of your uploaded environments at:

```
http://envs.anaconda.org/USERNAME
```

NOTE: Replace `USERNAME` with your username.

4. Anyone can download and install your environment from Cloud.

   - Using the web interface:

     Go to `https://anaconda.org/USERNAME/environments`.

     NOTE: Replace `USERNAME` with the username.

     Select the environment, go to Files tab and click the file to download under Names field.

   - Using the Terminal or an Anaconda Prompt:

```
conda env create user/my-environment
source activate my-environment
```

     NOTE: Replace `my-environment` with the actual name of your environment.

Working with other file types

In addition to packages and notebooks, Anaconda Cloud can be used to store and share data science files of any type.

Uploading other file types

You can upload any type of file with the Anaconda Client command line interface (CLI) by using the steps below.

PyPI package files, conda package files and notebook files are automatically detected. There is no auto-detect for other types of files, so you must explicitly specify the `package`, `package-type` and `version` fields.

In the following example, we upload a spreadsheet named `baby-names` in comma separated value (CSV) format.

Use the Terminal window or an Anaconda Prompt to upload the spreadsheet:

1. Create a new package, which creates a *namespace* that can hold multiple files:

```
anaconda login
anaconda package --create jsmith/baby-names
```

2. Upload the file to the new namespace:

```
anaconda upload --user jsmith --package baby-names --package-type file --version␣
↪1 baby-names1.csv
```

NOTE: In this example, the user–or organization–name is `jsmith`, the package name is `baby-names`, the package type is `file`, the version is `1` and the full filename is `baby-names1.csv`.

Downloading other file types

Files, such as the one created above, are available at:

```
https://anaconda.org/<USERNAME>/<PACKAGENAME>
```

Anyone can download these files using Client from the Terminal window or an Anaconda Prompt:

```
anaconda download <USERNAME>/<PACKAGENAME>
```

NOTE: Replace `<USERNAME>` with the desired username, and `<PACKAGENAME>` with the desired package name.

If the repository has multiple files with the same name and different extensions, `anaconda download` will download all of them by default. If you use `anaconda-client` 1.7 or higher, you can use `anaconda download` with the option `--package-type` or `-t` to specify only one of these files. This option can work with the values `pypi`, `conda`, `ipynb`, and `env`.

1.3.2 FAQs

What is Anaconda Cloud?

Anaconda Cloud is a package management service by Anaconda. For more information, see *Anaconda Cloud*.

What kind of packages does Cloud support?

Cloud supports any type of package. Today, it is primarily used for conda and PyPI packages, as well as notebooks and environments.

Who can find and install my packages?

If you have a free account, all of your packages are public. After you upload them to Cloud, anyone can search for and download them.

What is Anaconda, Inc.?

Anaconda is a software development and consulting company of passionate open source advocates based in Austin, Texas, USA. We are committed to the open source community. We created the Anaconda Python distribution and contribute to many other open source-based data analytics tools. You can find out more about us by reading our story.

What are Cloud's terms of service?

Our terms of service are available at https://anaconda.org/about/legal/terms . For any additional questions, contact us by email.

How much does Cloud cost?

Cloud is free for downloading and uploading public packages.

How do I get started with Cloud?

You can search, download and install hundreds of public packages without having an account. If you want to upload packages, you need to sign up for a Cloud account. For more information, see *sign up for a free Cloud account*.

What kind of account do I have?

By default your account is a personal, free account. All packages you upload to Cloud are public, and you are the only person with administrative access to your account.

What is included in the free version of Cloud?

The free plan allows you to search for, create and host public packages, and provides up to 3 GB storage space.

What is an organization account, and how is it different from an individual account?

An organization account allows multiple individual users to administer packages and have more control of package access by other users. An individual account is for use by one person.

1.3.3 Help and support

For support issues, see Anaconda Cloud Support.

Joining community support

We invite you to join our community support mailing lists for both Anaconda and conda. On these lists you can ask questions, answer questions and discuss ways to use Cloud. You also can submit requests for new features and make any other comments you may have.

Reporting a bug

Please use the Anaconda Cloud page Report a Bug to find the correct point of contact to report the bug you are experiencing. When reporting bugs on GitHub, please search to see if anyone else has reported it, and make a new issue if no one else has.

1.3.4 Command Reference

See also: API Reference

Anaconda client is the command line interface (CLI) to Anaconda Cloud, and can be used for logging in, logging out, managing your account, uploading files, generating access tokens, viewing tokens, and other tasks as shown by running:

```
anaconda -h
```

Full command reference:

anaconda

```
usage: anaconda [-h] [--disable-ssl-warnings] [--show-traceback] [-v] [-q]
                [-V] [-t TOKEN] [-s SITE]
                ...

Anaconda Cloud command line manager

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit

output:
```

(continues on next page)

(continued from previous page)

```
  --disable-ssl-warnings
                        Disable SSL warnings (default: False)
  --show-traceback      Show the full traceback for chalmers user errors
                        (default: False)
  -v, --verbose         print debug information ot the console
  -q, --quiet           Only show warnings or errors the console

anaconda-client options:
  -t TOKEN, --token TOKEN
                        Authentication token to use. May be a token or a path
                        to a file containing a token
  -s SITE, --site SITE  select the anaconda-client site to use

Commands:

    auth                Manage Authorization Tokens
    label               Manage your Anaconda Cloud labels
    channel             [DEPRECATED in favor of label] Manage your Anaconda
                        Cloud channels
    config              Anaconda client configuration
    copy                Copy packages from one account to another
    download            Download notebooks from Anaconda Cloud
    groups              Manage Groups
    login               Authenticate a user
    logout              Log out from Anaconda Cloud
    notebook            [DEPRECATED in favor of upload/download] Interact
                        with notebooks in anaconda.org
    package             Package utils
    remove              Remove an object from Anaconda Cloud. Must refer to
                        the formal package name as it appears in the URL of
                        the package. Also use anaconda show <USERNAME> to see
                        list of package names. Example: anaconda remove
                        continuumio/empty-example-notebook
    search              Search Anaconda Cloud
    show                Show information about an object
    upload              Upload packages to Anaconda Cloud
    whoami              Print the information of the current user
    build               Anaconda build client for continuous integration,
                        testing and building packages
    worker              Anaconda build client for continuous integration,
                        testing and building packages
```

Authentication

auth

```
usage: anaconda auth [-h] [-n NAME] [-o ORGANIZATION]
                     [--strength {strong,weak}] [--strong] [-w] [--url URL]
                     [--max-age MAX_AGE] [-s SCOPES] [--out OUT]
                     (-x | -l | -r NAME [NAME ...] | -c | -i)

Manage Authorization Tokens

optional arguments:
```

(continues on next page)

(continued from previous page)

```
  -h, --help            show this help message and exit
  -n NAME, --name NAME  A unique name so you can identify this token later.
                        View your tokens at anaconda.org/settings/access
  -o ORGANIZATION, --org ORGANIZATION, --organization ORGANIZATION
                        Set the token owner (must be an organization)

token creation arguments:
  These arguments are only valid with the `--create` action

  --strength {strong,weak}
  --strong              Create a longer token (default)
  -w, --weak            Create a shorter token
  --url URL             The url of the application that will use this token
  --max-age MAX_AGE     The maximum age in seconds that this token will be
                        valid for
  -s SCOPES, --scopes SCOPES
                        Scopes for token. For example if you want to limit
                        this token to conda downloads only you would use
                        --scopes "repo conda:download"
  --out OUT

actions:
  -x, --list-scopes     list all authentication scopes
  -l, --list            list all user authentication tokens
  -r NAME [NAME ...], --remove NAME [NAME ...]
                        remove authentication tokens
  -c, --create          Create an authentication token
  -i, --info, --current-info
                        Show information about the current authentication
                        token

Manage Authentication tokens
```

See also: *Using Anaconda Cloud Tokens*

login

```
usage: anaconda login [-h] [--hostname HOSTNAME] [--username LOGIN_USERNAME]
                      [--password LOGIN_PASSWORD]

Authenticate a user

optional arguments:
  -h, --help            show this help message and exit
  --hostname HOSTNAME   Specify the host name of this login, this should be
                        unique (default: hq-phone-114.corp.continuum.io)
  --username LOGIN_USERNAME
                        Specify your username. If this is not given, you will
                        be prompted
  --password LOGIN_PASSWORD
                        Specify your password. If this is not given, you will
                        be prompted
```

logout

```
usage: anaconda logout [-h]

Log out from Anaconda Cloud

optional arguments:
  -h, --help  show this help message and exit
```

whoami

Print the information of the current user

```
usage: anaconda whoami [-h]

Print the information of the current user

optional arguments:
  -h, --help  show this help message and exit
```

Informational

show

```
usage: anaconda show [-h] spec

Show information about an object

positional arguments:
  spec        Package written as USER[/PACKAGE[/VERSION[/FILE]]]

optional arguments:
  -h, --help  show this help message and exit

Show information about an object

Examples:

    anaconda show continuumio
    anaconda show continuumio/python
    anaconda show continuumio/python/2.7.5
    anaconda show sean/meta/1.2.0/meta.tar.gz
```

search

```
usage: anaconda search [-h] [-t {conda,pypi}]
                       [-p {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-
↪armv6l,linux-armv7l,linux-ppc64le,noarch}]
                       name

Search Anaconda Cloud
```

(continues on next page)

(continued from previous page)

```
positional arguments:
  name                  Search string

optional arguments:
  -h, --help            show this help message and exit
  -t {conda,pypi}, --package-type {conda,pypi}
                        only search for packages of this type
  -p {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-armv6l,linux-armv7l,linux-
↪ppc64le,noarch}, --platform {osx-32,osx-64,win-32,win-64,linux-32,linux-64,linux-
↪armv6l,linux-armv7l,linux-ppc64le,noarch}
                        only search for packages of the chosen platform

Search Anaconda Cloud for packages
```

config

```
usage: anaconda config [-h] [--type TYPE] [--set name value] [--get name]
                       [--remove REMOVE] [--show] [-f] [--show-sources] [-u]
                       [-s]

Anaconda client configuration

optional arguments:
  -h, --help            show this help message and exit
  --type TYPE           The type of the values in the set commands

actions:
  --set name value      sets a new variable: name value
  --get name            get value: name
  --remove REMOVE       removes a variable
  --show                show all variables
  -f, --files           show the config file names
  --show-sources        Display all identified config sources

location:
  -u, --user            set a variable for this user
  -s, --system, --site  set a variable for all users on this machine

anaconda-client configuration

Get, Set, Remove or Show the anaconda-client configuration.

###### anaconda-client sites

anaconda-client sites are a mechanism to allow users to quickly switch
between Anaconda Cloud instances. This is primarily used for testing
the anaconda alpha site. But also has applications for the
on-site [Anaconda Enterprise](http://continuum.io/anaconda-server).

anaconda-client comes with two pre-configured sites `alpha` and
`binstar` you may use these in one of two ways:

  * Invoke the anaconda command with the `-s/--site` option
```

(continues on next page)

(continued from previous page)

```
    e.g. to use the alpha testing site:

        anaconda -s alpha whoami

  * Set a site as the default:

        anaconda config --set default_site alpha
        anaconda whoami

###### Add an anaconda-client site

After installing Anaconda Enterprise
you can add a site named **site_name** like this:

    anaconda config --set sites.site_name.url "http://<anaconda-enterprise-ip>:<port>/
↪api"
    anaconda config --set default_site site_name

###### Site Options VS Global Options

All options can be set as global options - affecting all sites,
or site options - affecting only one site

By default options are set globally e.g.:

    anaconda config --set OPTION VALUE

If you want the option to be limited to a single site,
prefix the option with `sites.site_name` e.g.

    anaconda config --set sites.site_name.OPTION VALUE

###### Common anaconda-client configuration options

  * `url`: Set the anaconda api url (default: https://api.anaconda.org)
  * `ssl_verify`: Perform ssl validation on the https requests.
    ssl_verify may be `True`, `False` or a path to a root CA pem file.

###### Toggle auto_register when doing anaconda upload

The default is yes, automatically create a new package when uploading.
If no, then an upload will fail if the package name does not already exist on the␣
↪server.

    anaconda config --set auto_register yes|no
```

Managing Packages

package

```
usage: anaconda package [-h]
                        (--add-collaborator user | --list-collaborators | --create)
                        [--summary SUMMARY] [--license LICENSE]
                        [--license-url LICENSE_URL] [--personal | --private]
```

(continues on next page)

(continued from previous page)

```
                        USER/PACKAGE

Anaconda Cloud package utilities

positional arguments:
  USER/PACKAGE          Package to operate on

optional arguments:
  -h, --help            show this help message and exit

actions:
  --add-collaborator user
                        username of the collaborator you want to add
  --list-collaborators  list all of the collaborators in a package
  --create              Create a package

metadata arguments:
  --summary SUMMARY     Set the package short summary
  --license LICENSE     Set the package license
  --license-url LICENSE_URL
                        Set the package license url

privacy:
  --personal            Set the package access to personal This package will
                        be available only on your personal registries
  --private             Set the package access to private This package will
                        require authorized and authenticated access to install
```

upload

```
usage: anaconda upload [-h] [-c CHANNELS] [-l LABELS] [--no-progress]
                       [-u USER] [--all] [-p PACKAGE] [-v VERSION]
                       [-s SUMMARY] [-t PACKAGE_TYPE] [-d DESCRIPTION]
                       [--thumbnail THUMBNAIL] [--private]
                       [--no-register | --register] [--build-id BUILD_ID]
                       [-i | -f | --force]
                       files [files ...]

Upload packages to Anaconda Cloud

positional arguments:
  files                 Distributions to upload

optional arguments:
  -h, --help            show this help message and exit
  -c CHANNELS, --channel CHANNELS
                        [DEPRECATED] Add this file to a specific channel.
                        Warning: if the file channels do not include "main",
                        the file will not show up in your user channel
  -l LABELS, --label LABELS
                        Add this file to a specific label. Warning: if the
                        file labels do not include "main", the file will not
                        show up in your user label
  --no-progress         Don't show upload progress
```

(continues on next page)

(continued from previous page)

```
  -u USER, --user USER  User account or Organization, defaults to the current
                        user
  --all                 Use conda convert to generate packages for all
                        platforms and upload them
  --no-register         Don't create a new package namespace if it does not
                        exist
  --register            Create a new package namespace if it does not exist
  --build-id BUILD_ID   Anaconda Cloud Build ID (internal only)
  -i, --interactive     Run an interactive prompt if any packages are missing
  -f, --fail            Fail if a package or release does not exist (default)
  --force               Force a package upload regardless of errors

metadata options:
  -p PACKAGE, --package PACKAGE
                        Defaults to the package name in the uploaded file
  -v VERSION, --version VERSION
                        Defaults to the package version in the uploaded file
  -s SUMMARY, --summary SUMMARY
                        Set the summary of the package
  -t PACKAGE_TYPE, --package-type PACKAGE_TYPE
                        Set the package type [ipynb, env]. Defaults to
                        autodetect
  -d DESCRIPTION, --description DESCRIPTION
                        description of the file(s)
  --thumbnail THUMBNAIL
                        Notebook's thumbnail image
  --private             Create the package with private access

    anaconda upload CONDA_PACKAGE_1.bz2
    anaconda upload notebook.ipynb
    anaconda upload environment.yml
```

See Also:

- *Uploading a Conda Package*
- *Uploading a PyPI Package*

download

```
usage: anaconda download [-h] [-f] [-o OUTPUT] handle

Download packages from Anaconda Cloud

positional arguments:
  handle                user/notebook

optional arguments:
  -h, --help            show this help message and exit
  -f, --force           Overwrite
  -o OUTPUT, --output OUTPUT
                        Download as

Usage:
    anaconda download notebook
    anaconda download user/notebook
```

remove

```
usage: anaconda remove [-h] [-f] specs [specs ...]

Remove an object from Anaconda Cloud

example::

    anaconda remove sean/meta/1.2.0/meta.tar.gz

positional arguments:
  specs         Package written as <user>[/<package>[/<version>[/<filename>]]]

optional arguments:
  -h, --help    show this help message and exit
  -f, --force   Do not prompt removal
```

groups

```
usage: anaconda groups [-h] [--perms {read,write,admin}]
                       {add,show,members,add_member,remove_member,packages,add_
↪package,remove_package}
                       spec

positional arguments:
  {add,show,members,add_member,remove_member,packages,add_package,remove_package}
                        The group management command to execute
  spec                  <organization>/<group_name>/<member>

optional arguments:
  -h, --help            show this help message and exit
  --perms {read,write,admin}
                        The permission the group should provide
```

label

```
usage: anaconda label [-h] [-o ORGANIZATION]
                      (--copy LABEL LABEL | --list | --show LABEL | --lock LABEL | --
↪unlock LABEL | --remove LABEL)

Manage your Anaconda Cloud channels

optional arguments:
  -h, --help            show this help message and exit
  -o ORGANIZATION, --organization ORGANIZATION
                        Manage an organizations labels
  --copy LABEL LABEL
  --list                list all labels for a user
  --show LABEL          Show all of the files in a label
  --lock LABEL          Lock a label
  --unlock LABEL        Unlock a label
  --remove LABEL        Remove a label
```

copy

```
usage: anaconda copy [-h] [--to-owner TO_OWNER] [--from-label FROM_LABEL]
                     [--to-label TO_LABEL]
                     spec

Copy packages from one account to another

positional arguments:
  spec                  Package - written as user/package/version[/filename]
                        If filename is not given, copy all files in the
                        version

optional arguments:
  -h, --help            show this help message and exit
  --to-owner TO_OWNER   User account to copy package to (default: your
                        account)
  --from-label FROM_LABEL
                        Label to copy packages from
  --to-label TO_LABEL   Label to put all packages into
```

move

```
usage: anaconda move [-h] [--from-label FROM_LABEL] [--to-label TO_LABEL] spec

Move packages between labels.

positional arguments:
  spec                  Package - written as user/package/version[/filename]
                        If filename is not given, move all files in the
                        version

optional arguments:
  -h, --help            show this help message and exit
  --from-label FROM_LABEL
                        Label to move packages from
  --to-label TO_LABEL   Label to move packages to
```

1.3.5 Glossary

Anaconda

An easy-to-install, free collection of open source packages, including Python and the conda package manager, with free community support. Over 150 packages are installed with Anaconda. After installing Anaconda, you can install or update over 250 additional open source packages contained in the Anaconda repository using the `conda install PACKAGENAME` command.

NOTE: Replace `PACKAGENAME` with the name of the desired package.

Anaconda Cloud

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, and conda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using.

Cloud hosts hundreds of useful Python packages, notebooks and environments for a wide variety of applications. You do not need a Cloud account, or to be logged in, to search for public packages, download and install them.

For more information, see the *introduction to Anaconda Cloud*.

Anaconda Client CLI

The Anaconda Client command line interface (CLI) allows you to log into Cloud directly from your terminal window and manage your account. It is not necessary for downloading or installing packages from Cloud.

Binstar

Binstar was an early project name for Cloud. You may still see the term Binstar in certain command and directory names.

conda

The conda package manager and environment manager program that installs and updates packages and their dependencies, and lets you easily switch between environments on your local computer.

conda build

The command line interface that lets you build packages for your local operating system.

conda package

A compressed file containing system-level libraries, Python modules, executable programs or other components. The file uses the tarball format.

label

Part of the URLs on Cloud where conda looks for packages. Labels are searched only if you specify a label.

The default label is `main`, so packages that are uploaded without specifying a label are automatically labeled `main`. The version labeled `main` is also downloaded by default, unless a user specifies a different label.

So, if a file is labeled `main`, then the label name may be omitted from the URL. For example, the following repositories are equivalent:

```
https://anaconda.org/sean/labels/main
https://anaconda.org/sean
```

Commands such as `conda install` can be used with a channel or used with a channel and a label:

```
conda install --channel sean selenium
conda install --channel sean/label/dev selenium
conda install --channel sean/label/stable selenium
```

Miniconda

A minimal installer for *conda*. Like *Anaconda*, Miniconda is a software package that includes the conda package manager and Python and its dependencies. However, Miniconda does not include any other packages. Once conda is installed by installing either Anaconda or Miniconda, you can install other software packages directly from the command line using `conda install`.

namespace

Each user and organization has their own location called a “namespace” where they may host packages. You can view the public packages in a user or organization’s namespace by navigating to their user page.

EXAMPLE: The `travis` user namespace located at `https://anaconda.org/travis` contains packages that were uploaded and shared by the user whose account is named `travis`.

noarch package

A conda package that contains nothing specific to any system architecture, so it may be installed on any system. When conda searches for packages on any system in a channel, conda always checks both the system-specific subdirectory, for example, `linux-64` and the `noarch` directory.

on-site repository

Cloud is powered by Anaconda Repository by Anaconda, Inc. You can run your own private repository behind firewalls or in air-gapped environments. For more information, contact sales@anaconda.com.

organization account

An organization account is a type of account on Cloud that allows multiple individual users to administer packages and control package access to different user groups. It also includes a large amount of storage space.

package

All files uploaded to Cloud are stored in packages. Each Cloud package is visible at its own unique URL based on the name of the user who owns the package and the name of the package.

For example, if a user `travis` uploads a test package named `testpkg`, it is visible at:

```
https://anaconda.org/travis/testpkg
```

Cloud packages may contain multiple files, and these files may be data files such as comma separated value (CSV), tab separated value (TSV), or text (TXT), or package files such as conda packages, PyPI packages or R packages.

NOTE: All packages are public if uploaded by users of free accounts. Packages may be designated as private by upgrading to a paid account.

package manager

A package manager is a tool that facilitates the process of installing, upgrading, configuring and removing packages, including the packages on Cloud. Cloud supports two package managers, *conda* and *PyPI*.

For more information, see *using package managers*.

repository

A storage location from which software packages may be retrieved and installed on a computer.

source package

"Source" packages are source code only, not yet built for any specific platform, and might be compatible with all, some or only one of the platforms.

token

A token–or authentication token–is the mechanism by which anonymous users can download private packages without using a Cloud account. It is an alpha-numeric code that is inserted into a URL that allows access by anyone who has the URL. You can use Client to generate new tokens to give other users specifically scoped access to packages and collections.

1.4 Archive

This is the archive of Anaconda products that are no longer in production. This documentation is provided as a courtesy for customers who still use them.

1.4.1 Anaconda Accelerate

High Performance Computing

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it. SEE https://www.anaconda.com/blog/developer-blog/open-sourcing-anaconda-accelerate/

Anaconda Accelerate is a package that provides the Anaconda® platform access to several numerical libraries that are optimized for performance on Intel CPUs and NVidia GPUs.

The current version, 2.3.1, was released on January 30, 2017.

Anyone can now use the functionality from Accelerate without purchasing a license!

On July 27, 2017, Accelerate was split into the Intel Distribution for Python and the open source Numba project's sub-projects pyculib, pyculib_sorting and data_profiler. Numba and all its sub-projects are available under a BSD license.

More information is available in the post Open Sourcing Anaconda Accelerate.

Features

- **Bindings to the following *CUDA libraries*:**
  - *cuBLAS*
  - *cuFFT*
  - *cuSPARSE*
  - *cuRAND*
  - *CUDA Sorting* algorithms from the CUB and Modern GPU libraries
- Speed-boosted linear algebra operations in NumPy, SciPy, scikit-learn and NumExpr libraries using Intel's Math Kernel Library (*MKL*).
- Accelerated variants of Numpy's built-in *UFuncs*.
- Increased-speed Fast Fourier Transformations (*FFT*) in NumPy.

Installation

This section contains information related to:

Requirements

- 64-bit operating system–Windows, macOS or Linux
- **Supported Python and Numpy combinations:**

  - Python 2.7 with Numpy 1.9, 1.10 or 1.11
  - Python 3.4 with Numpy 1.9, 1.10 or 1.11
  - Python 3.5 with Numpy 1.9, 1.10 or 1.11
  - Python 3.6 with Numpy 1.11
- Numba 0.30

CUDA feature requirements

- NVidia driver version 349.00 or later
- CUDA toolkit 7.0
- At least one CUDA GPU with compute capability 2.0 or above

Installing Accelerate

Accelerate is included with Anaconda Workgroup and Anaconda Enterprise subscriptions.

To start a 30-day free trial, download and install the Anaconda Accelerate package.

Installing with Anaconda

If you already have the *Anaconda free Python distribution <http://anaconda.com/downloads.html>*, take the following steps to install Accelerate:

1. Run the command `conda update conda`.
2. Run the command `conda install accelerate`.

If you do not have Anaconda installed, see Downloads.

NOTE: Accelerate can also be installed into your own non-Anaconda Python environment.

For more information, contact sales@anaconda.com.

Updating Accelerate

To update Accelerate, take the following steps:

1. Run the command `conda update conda`.
2. Run the command `conda update accelerate`.

Updating NumbaPro

If you have NumbaPro installed, you must manually upgrade NumbaPro to install the NumbaPro compatibility layer by taking the following steps:

1. Run the command `conda update conda`.
2. Run the command `conda update numbapro`.

Managing your Accelerate license

You can install, view or remove your Accelerate license with the graphical Anaconda Navigator license manager, or manually with your operating system.

For more information, see *Installing licenses*.

User guide

This section contains information related to:

CUDA libraries

The following CUDA libraries have bindings and algorithms that are available for use with Accelerate:

cuBLAS

Provides basic linear algebra building blocks. See NVIDIA cuBLAS.

The cuBLAS binding provides an interface that accepts NumPy arrays and Numba's CUDA device arrays. The binding automatically transfers NumPy array arguments to the device as required. This automatic transfer may generate some unnecessary transfers, so optimal performance is likely to be obtained by the manual transfer for NumPy arrays into device arrays and using the cuBLAS to manipulate device arrays where possible.

No special naming convention is used to identify the data type, unlike in the BLAS C and Fortran APIs. Arguments for array storage information which are part of the cuBLAS C API are also not necessary since NumPy arrays and device arrays contain this information.

All functions are accessed through the `accelerate.cuda.blas.Blas` class:

BLAS Level 1

`accelerate.cuda.blas.Blas.`**`nrm2`**`(x)`

Computes the L2 norm for array *x*. Same as *numpy.linalg.norm(x)*.

Parameters **x** (*python.array*) – input vector

Returns resulting norm.

`accelerate.cuda.blas.Blas.`**`dot`**`(x, y)`

Compute the dot product of array *x* and array *y*. Same as *np.dot(x, y)*.

Parameters

- **x** (*python.array*) – vector
- **y** (*python.array*) – vector

Returns dot product of *x* and *y*

`accelerate.cuda.blas.Blas.`**`dotc`**`(x, y)`

Uses the conjugate of the element of the vectors to compute the dot product of array *x* and array *y* for complex dtype only. Same as *np.vdot(x, y)*.

Parameters

- **x** (*python.array*) – vector

- **y** (*python.array*) – vector

Returns dot product of *x* and *y*

`accelerate.cuda.blas.Blas.`**`scal`**(*alpha*, *x*)

Scale *x* inplace by alpha. Same as *x = alpha * x*

Parameters

- **alpha** – scalar
- **x** (*python.array*) – vector

`accelerate.cuda.blas.Blas.`**`axpy`**(*alpha*, *x*)

Compute *y = alpha * x + y* inplace.

Parameters

- **alpha** – scalar
- **x** (*python.array*) – vector

`accelerate.cuda.blas.Blas.`**`amax`**(*x*)

Find the index of the first largest element in array *x*. Same as *np.argmax(x)*

Parameters **x** (*python.array*) – vector

Returns index (start from 0).

`accelerate.cuda.blas.Blas.`**`amin`**(*x*)

Find the index of the first largest element in array *x*. Same as *np.argmin(x)*

Parameters **x** (*python.array*) – vector

Returns index (start from 0).

`accelerate.cuda.blas.Blas.`**`asum`**(*x*)

Compute the sum of all element in array *x*.

Parameters **x** (*python.array*) – vector

Returns *x.sum()*

`accelerate.cuda.blas.Blas.`**`rot`**(*x*, *y*, *c*, *s*)

Apply the Givens rotation matrix specified by the cosine element *c* and the sine element *s* inplace on vector element *x* and *y*.

Same as *x, y = c * x + s * y, -s * x + c * y*

Parameters

- **x** (*python.array*) – vector
- **y** (*python.array*) – vector

`accelerate.cuda.blas.Blas.`**`rotg`**(*a*, *b*)

Constructs the Givens rotation matrix with the column vector (a, b).

Parameters

- **a** – first element of the column vector
- **b** – second element of the column vector

Returns

a tuple (r, z, c, s)

r – *r = a**2 + b**2*

z – Use to reconstruct *c* and *s*. Refer to cuBLAS documentation for detail.

c – The consine element.

s – The sine element.

`accelerate.cuda.blas.Blas.`**`rotm`**(*x*, *y*, *param*)

Applies the modified Givens transformation inplace.

Same as:

```
param = flag, h11, h21, h12, h22
x[i] = h11 * x[i] + h12 * y[i]
y[i] = h21 * x[i] + h22 * y[i]
```

Refer to the cuBLAS documentation for the use of *flag*.

Parameters

- **x** (*python.array*) – vector
- **y** (*python.array*) – vector

`accelerate.cuda.blas.Blas.`**`rotmg`**(*d1*, *d2*, *x1*, *y1*)

Constructs the modified Givens transformation *H* that zeros out the second entry of a column vector *(d1 * x1, d2 * y1)*.

Parameters

- **d1** – scaling factor for the x-coordinate of the input vector
- **d2** – scaling factor for the y-coordinate of the input vector
- **x1** – x-coordinate of the input vector
- **y1** – y-coordinate of the input vector

Returns A 1D array that is usable in *rotm*. The first element is the flag for *rotm*. The rest of the elements corresponds to the *h11, h21, h12, h22* elements of *H*.

BLAS Level 2

All level 2 routines follow the following naming convention for all arguments:

- **A, B, C, AP – (2D array) Matrix argument.** *AP* implies packed storage for banded matrix.
- x, y, z – (1D arrays) Vector argument.
- alpha, beta – (scalar) Can be floats or complex numbers depending.
- m – (scalar) Number of rows of matrix *A*.
- **n – (scalar) Number of columns of matrix *A*. If *m* is not needed,** *n* also means the number of rows of the matrix *A*; thus, implying a square matrix.
- **trans, transa, transb – (string)** Select the operation *op* to apply to a matrix:
  - ‘N’: *op(X)* = *X*, the identity operation;
  - ‘T’: *op(X)* = *X**T*, the transpose;
  - ‘C’: *op(X)* = *X**H*, the conjugate transpose.

  trans only applies to the only matrix argument. *transa* and *transb* apply to matrix *A* and matrix *B*, respectively.

- **uplo – (string) Can be 'U' for filling the upper trianglar matrix; or 'L' for** filling the lower trianglar matrix.
- diag – (boolean) Whether the matrix diagonal has unit elements.
- **mode – (string) 'L' means the matrix is on the left side in the equation.** 'R' means the matrix is on the right side in the equation.

Note: The last array argument is always overwritten with the result.

`accelerate.cuda.blas.Blas.`**`gbmv`**(*trans*, *m*, *n*, *kl*, *ku*, *alpha*, *A*, *x*, *beta*, *y*)

banded matrix-vector multiplication *y = alpha * op(A) * x + beta * y* where *A* has *kl* sub-diagonals and *ku* super-diagonals.

`accelerate.cuda.blas.Blas.`**`gemv`**(*trans*, *m*, *n*, *alpha*, *A*, *x*, *beta*, *y*)

matrix-vector multiplication *y = alpha * op(A) * x + beta * y*

`accelerate.cuda.blas.Blas.`**`trmv`**(*uplo*, *trans*, *diag*, *n*, *A*, *x*)

triangular matrix-vector multiplication *x = op(A) * x*

`accelerate.cuda.blas.Blas.`**`tbmv`**(*uplo*, *trans*, *diag*, *n*, *k*, *A*, *x*)

triangular banded matrix-vector *x = op(A) * x*

`accelerate.cuda.blas.Blas.`**`tpmv`**(*uplo*, *trans*, *diag*, *n*, *AP*, *x*)

triangular packed matrix-vector multiplication *x = op(A) * x*

`accelerate.cuda.blas.Blas.`**`trsv`**(*uplo*, *trans*, *diag*, *n*, *A*, *x*)

Solves the triangular linear system with a single right-hand-side. *op(A) * x = b*

`accelerate.cuda.blas.Blas.`**`tpsv`**(*uplo*, *trans*, *diag*, *n*, *AP*, *x*)

Solves the packed triangular linear system with a single right-hand-side. *op(A) * x = b*

`accelerate.cuda.blas.Blas.`**`tbsv`**(*uplo*, *trans*, *diag*, *n*, *k*, *A*, *x*)

Solves the triangular banded linear system with a single right-hand-side. *op(A) * x = b*

`accelerate.cuda.blas.Blas.`**`symv`**(*uplo*, *n*, *alpha*, *A*, *x*, *beta*, *y*)

symmetric matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`hemv`**(*uplo*, *n*, *alpha*, *A*, *x*, *beta*, *y*)

Hermitian matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`sbmv`**(*uplo*, *n*, *k*, *alpha*, *A*, *x*, *beta*, *y*)

symmetric banded matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`hbmv`**(*uplo*, *n*, *k*, *alpha*, *A*, *x*, *beta*, *y*)

Hermitian banded matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`spmv`**(*uplo*, *n*, *alpha*, *AP*, *x*, *beta*, *y*)

symmetric packed matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`hpmv`**(*uplo*, *n*, *alpha*, *AP*, *x*, *beta*, *y*)

Hermitian packed matrix-vector multiplication *y = alpha * A * x + beta * y*

`accelerate.cuda.blas.Blas.`**`ger`**(*m*, *n*, *alpha*, *x*, *y*, *A*)

the rank-1 update *A := alpha * x * y ** T + A*

`accelerate.cuda.blas.Blas.`**`geru`**(*m*, *n*, *alpha*, *x*, *y*, *A*)

the rank-1 update *A := alpha * x * y ** T + A*

`accelerate.cuda.blas.Blas.`**`gerc`**(*m*, *n*, *alpha*, *x*, *y*, *A*)

the rank-1 update *A := alpha * x * y ** H + A*

accelerate.cuda.blas.Blas.**syr**(*uplo*, *n*, *alpha*, *x*, *A*)
symmetric rank 1 operation *A := alpha * x * x ** T + A*

accelerate.cuda.blas.Blas.**her**(*uplo*, *n*, *alpha*, *x*, *A*)
hermitian rank 1 operation *A := alpha * x * x ** H + A*

accelerate.cuda.blas.Blas.**spr**(*uplo*, *n*, *alpha*, *x*, *AP*)
the symmetric rank 1 operation *A := alpha * x * x ** T + A*

accelerate.cuda.blas.Blas.**hpr**(*uplo*, *n*, *alpha*, *x*, *AP*)
hermitian rank 1 operation *A := alpha * x * x ** H + A*

accelerate.cuda.blas.Blas.**syr2**(*uplo*, *n*, *alpha*, *x*, *y*, *A*)
symmetric rank-2 update *A* = *alpha * x * y ** T + y * x ** T + A*

accelerate.cuda.blas.Blas.**her2**(*uplo*, *n*, *alpha*, *x*, *y*, *A*)
Hermitian rank-2 update *A* = *alpha * x * y ** H + alpha * y * x ** H + A*

accelerate.cuda.blas.Blas.**spr2**(*uplo*, *n*, *alpha*, *x*, *y*, *A*)
packed symmetric rank-2 update *A* = *alpha * x * y ** T + y * x ** T + A*

accelerate.cuda.blas.Blas.**hpr2**(*uplo*, *n*, *alpha*, *x*, *y*, *A*)
packed Hermitian rank-2 update *A* = *alpha * x * y ** H + alpha * y * x ** H + A*

BLAS Level 3

All level 3 routines follow the same naming convention for arguments as in level 2 routines.

accelerate.cuda.blas.Blas.**gemm**(*transa*, *transb*, *m*, *n*, *k*, *alpha*, *A*, *B*, *beta*, *C*)
matrix-matrix multiplication *C* = *alpha * op(A) * op(B) + beta * C*

accelerate.cuda.blas.Blas.**syrk**(*uplo*, *trans*, *n*, *k*, *alpha*, *A*, *beta*, *C*)
symmetric rank- k update *C* = *alpha * op(A) * op(A) ** T + beta * C*

accelerate.cuda.blas.Blas.**herk**(*uplo*, *trans*, *n*, *k*, *alpha*, *A*, *beta*, *C*)
Hermitian rank- k update *C* = *alpha * op(A) * op(A) ** H + beta * C*

accelerate.cuda.blas.Blas.**symm**(*side*, *uplo*, *m*, *n*, *alpha*, *A*, *B*, *beta*, *C*)
symmetric matrix-matrix multiplication:

```
if  side == 'L':
    C = alpha * A * B + beta * C
else:   # side == 'R'
    C = alpha * B * A + beta * C
```

accelerate.cuda.blas.Blas.**hemm**(*side*, *uplo*, *m*, *n*, *alpha*, *A*, *B*, *beta*, *C*)
Hermitian matrix-matrix multiplication:

```
if  side == 'L':
    C = alpha * A * B + beta * C
else:    #  side == 'R':
    C = alpha * B * A + beta * C
```

accelerate.cuda.blas.Blas.**trsm**(*side*, *uplo*, *trans*, *diag*, *m*, *n*, *alpha*, *A*, *B*)
Solves the triangular linear system with multiple right-hand-sides:

```
if  side == 'L':
    op(A) * X = alpha * B
else:        # side == 'R'
    X * op(A) = alpha * B
```

accelerate.cuda.blas.Blas.**trmm**(*side*, *uplo*, *trans*, *diag*, *m*, *n*, *alpha*, *A*, *B*, *C*)

triangular matrix-matrix multiplication:

```
if  side == ':'
    C = alpha * op(A) * B
else:    # side == 'R'
    C = alpha * B * op(A)
```

accelerate.cuda.blas.Blas.**dgmm**(*side*, *m*, *n*, *A*, *x*, *C*)

matrix-matrix multiplication:

```
if  mode == 'R':
    C = A * x * diag(X)
else:        # mode == 'L'
    C = diag(X) * x * A
```

accelerate.cuda.blas.Blas.**geam**(*transa*, *transb*, *m*, *n*, *alpha*, *A*, *beta*, *B*, *C*)

matrix-matrix addition/transposition *C = alpha * op(A) + beta * op(B)*

cuSPARSE

Provides basic linear algebra operations for sparse matrices. See NVIDIA cuSPARSE for an in-depth description of the cuSPARSE library and its methods and data types. All functions are accessed through the accelerate.cuda.sparse.Sparse class:

Similarly to the cuBLAS interface, no special naming convention is used for functions to operate on different datatypes - all datatypes are handled by each function, and dispatch of the corresponding library function is handled by Accelerate. However, it is often necessary to provide a *matrix descriptor* to functions, which provides some information about the format and properties of a matrix. A matrix descriptor can be obtained from the *accelerate.cuda.sparse.Sparse.matdescr()* method:

accelerate.cuda.sparse.Sparse.**matdescr**(*indexbase*, *diagtype*, *fillmode*, *matrixtype*)

Creates a matrix descriptor that describes a matrix with the given *indexbase*, *diagtype*, *fillmode*, and *matrixtype*. Note that not all of these options are relevant to every matrix storage format.

Parameters

- **indexbase** – Optional. 0 for 0-based indexing, or 1 for 1-based indexing. If not specified, the default given to the accelerate.cuda.sparse.Sparse constructor is used instead.
- **diagtype** – Optional. Defaults to *'N'*. *'N'* signifies that the matrix diagonal has non-unit elements. *'U'* signifies that the matrix diagonal only contains unit elements.
- **fillmode** – Optional. Defaults to *'L'*. *'L'* indicates that the lower triangular part of the matrix is stored. *'U'* indicates that the upper triangular part of the matrix is stored.
- **matrixtype** – Optional. Defaults to *'G'*. *'S'* indicates that the matrix is symmetric. *'H'* indicates that the matrix is Hermitian. *'T'* indicates that the matrix is triangular. *'G'* is used for a *general* matrix, which is not symmetric, Hermitian, or triangular.

Returns A matrix descriptor.

Many of the methods of the accelerate.cuda.sparse.Sparse class accept the individual data structures that make up a sparse representation of a matrix (for example the values, the row pointers and the column indices for a CSR format matrix). However, some methods (such as *accelerate.cuda.sparse.Sparse.csrgemm_ez()*), accept an instance of the *accelerate.cuda.sparse.CudaSparseMatrix* class:

class accelerate.cuda.sparse.**CudaSparseMatrix**

Base class for a representation of a sparse matrix on a CUDA device. The constructor takes no arguments.

from_host_matrix(*matrix*, *stream*)

Initialise the matrix structure and values from an instance of a matrix on the host. The host matrix must be of the corresponding host type, which is documented for each subclass below.

copy_to_host(*stream*)

Create an instance of the corresponding host matrix type and copy the matrix structure and data into it from the device. See subclass documentation for an indication of the corresponding matrix type.

Subclasses of the sparse matrix type are:

class accelerate.cuda.sparse.**CudaBSRMatrix**

CUDA sparse matrix for which the corresponding type is a `scipy.sparse.bsr_matrix`.

class accelerate.cuda.sparse.**CudaCSRMatrix**

CUDA sparse matrix for which the corresponding type is a `scipy.sparse.csr_matrix`.

class accelerate.cuda.sparse.**CudaCSCMatrix**

CUDA sparse matrix for which the corresponding type is a `scipy.sparse.csc_matrix`.

There are also some convenience methods for constructing CUDA sparse matrices in a similar manner to Scipy sparse matrices:

BLAS Level 1

accelerate.cuda.sparse.Sparse.**axpyi**(*alpha*, *xVal*, *xInd*, *y*)

Multiplies the sparse vector *x* by *alpha* and adds the result to the dense vector *y*.

Parameters

- **alpha** – scalar
- **xVal** – vector of non-zero values of *x*
- **xInd** – vector of indices of non-zero values of *x*
- **y** – dense vector

Returns dense vector

accelerate.cuda.sparse.Sparse.**doti**(*xVal*, *xInd*, *y*)

Computes the dot product of the sparse vector *x* and dense vector *y*.

Parameters

- **xVal** – vector of non-zero values of *x*
- **xInd** – vector of indices of non-zero values of *x*
- **y** – dense vector

Returns scalar

accelerate.cuda.sparse.Sparse.**dotci**(*xVal*, *xInd*, *y*)

Computes the dot product of the complex conjugate of the sparse vector *x* and the dense vector *y*.

Parameters

- **xVal** – vector of non-zero values of *x*
- **xInd** – vector of indices of non-zero values of *x*
- **y** – dense vector

Returns scalar

`accelerate.cuda.sparse.Sparse.`**`gthr`**(*y*, *xVal*, *xInd*)

Gathers the elements of *y* at the indices *xInd* into the array *xVal*

Parameters

- **`xVal`** – vector of non-zero values of *x*
- **`xInd`** – vector of indices of non-zero values of *x*
- **`y`** – dense vector

Returns None

`accelerate.cuda.sparse.Sparse.`**`gthrz`**(*y*, *xVal*, *xInd*)

Gathers the elements of *y* at the indices *xInd* into the array *xVal* and zeroes out the gathered elements of *y*.

Parameters

- **`xVal`** – vector of non-zero values of *x*
- **`xInd`** – vector of indices of non-zero values of *x*
- **`y`** – dense vector

Returns None

`accelerate.cuda.sparse.Sparse.`**`sctr`**(*xVal*, *xInd*, *y*)

Scatters the elements of the sparse vector *x* into the dense vector *y*. Elements of *y* whose indices are not listed in *xInd* are unmodified.

Parameters

- **`xVal`** – vector of non-zero values of *x*
- **`xInd`** – vector of indices of non-zero values of *x*
- **`y`** – dense vector

Returns None

BLAS Level 2

All level 2 routines follow the following naming convention for the following arguments:

- alpha, beta – (scalar) Can be real or complex numbers.
- descr, descrA, descrB – (descriptor) Matrix descriptor. An appropriate descriptor may be obtained by calling *`accelerate.cuda.sparse.Sparse.matdescr()`*. *descr* only applies to the only matrix argument. *descrA* and *descrB* apply to matrix *A* and matrix *B*, respectively.
- dir – (string) Can be *'C'* to indicate column-major block storage or *'R'* to indicate row-major block storage.
- **trans, transa, transb – (string)** Select the operation *op* to apply to a matrix:
  - *'N'*: *op(X)* = *X*, the identity operation;
  - *'T'*: *op(X)* = *X**T*, the transpose;
  - *'C'*: *op(X)* = *X**H*, the conjugate transpose.

  trans only applies to the only matrix argument. *transa* and *transb* apply to matrix *A* and matrix *B*, respectively.

`accelerate.cuda.sparse.Sparse.`**`bsrmv_matrix`**(*dir*, *trans*, *alpha*, *descr*, *bsrmat*, *x*, *beta*, *y*)

Matrix-vector multiplication *y = alpha * op(A) * x + beta * y* with a BSR-format matrix.

Parameters

- **dir** – block storage direction
- **trans** – operation to apply to the matrix
- **alpha** – scalar
- **descr** – matrix descriptor
- **bsrmat** – the matrix *A*
- **x** – dense vector
- **beta** – scalar
- **y** – dense vector

Returns None

`accelerate.cuda.sparse.Sparse.`**`bsrmv`**(*dir*, *trans*, *mb*, *nb*, *nnzb*, *alpha*, *descr*, *bsrVal*, *bsrRowPtr*, *bsrColInd*, *blockDim*, *x*, *beta*, *y*)

Matrix-vector multiplication *y = alpha * op(A) * x + beta * y* with a BSR-format matrix. This function accepts the individual arrays that make up the structure of a BSR matrix - if a `accelerate.cuda.sparse.CudaBSRMatrix` instance is to hand, it is recommended to use the `bsrmv_matrix()` method instead.

Parameters

- **dir** – block storage direction
- **trans** – operation to apply to the matrix
- **mb** – Number of block rows of the matrix
- **nb** – Number of block columns of the matrix
- **nnzb** – Number of nonzero blocks of the matrix
- **alpha** – scalar
- **descr** – matrix descriptor
- **bsrVal** – vector of nonzero values of the matrix
- **bsrRowPtr** – vector of block row pointers of the matrix
- **bsrColInd** – vector of block column indices of the matrix
- **blockDim** – block dimension of the matrix
- **x** – dense vector
- **beta** – scalar
- **y** – dense vector

Returns None

`accelerate.cuda.sparse.Sparse.`**`bsrxmv`**(*dir*, *trans*, *sizeOfMask*, *mb*, *nb*, *nnzb*, *alpha*, *descr*, *bsrVal*, *bsrMaskPtr*, *bsrRowPtr*, *bsrEndPtr*, *bsrColInd*, *blockDim*, *x*, *beta*, *y*)

Matrix-vector multiplication similar to `bsrmv()`, but including a mask operation: *y(mask) = (alpha * op(A) * x + beta * y)(mask)*. The blocks of y to be updated are specified in *bsrMaskPtr*. Blocks whose indices are not specified in *bsrMaskPtr* are left unmodified.

Parameters

- **dir** – block storage direction
- **trans** – operation to apply to the matrix
- **sizeOfMask** – number of updated blocks of rows of *y*
- **mb** – Number of block rows of the matrix
- **nb** – Number of block columns of the matrix
- **nnzb** – Number of nonzero blocks of the matrix
- **alpha** – scalar
- **descr** – matrix descriptor
- **bsrVal** – vector of nonzero values of the matrix
- **bsrMaskPtr** – vector of indices of the block elements to be updated
- **bsrRowPtr** – vector of block row pointers of the matrix
- **bsrEndPtr** – vector of pointers to the end of every block row plus one
- **bsrColInd** – vector of block column indices of the matrix
- **blockDim** – block dimension of the matrix
- **x** – dense vector
- **beta** – scalar
- **y** – dense vector

Returns None

accelerate.cuda.sparse.Sparse.**csrmv**(*trans*, *m*, *n*, *nnz*, *alpha*, *descr*, *csrVal*, *csrRowPtr*, *csrColInd*, *x*, *beta*, *y*)

Matrix-vector multiplication *y = alpha * op(A) * x + beta * y* with a CSR-format matrix.

Parameters

- **trans** – operation to apply to the matrix
- **m** – Number of rows of the matrix
- **n** – Number of columns of the matrix
- **nnz** – Number of nonzeroes of the matrix
- **alpha** – scalar
- **descr** – matrix descriptor
- **csrVal** – vector of nonzero values of the matrix
- **csrRowPtr** – vector of row pointers of the matrix
- **csrColInd** – vector of column indices of the matrix
- **x** – dense vector
- **beta** – scalar
- **y** – dense vector

Returns None

accelerate.cuda.sparse.Sparse.**csrsv_analysis**(*trans*, *m*, *nnz*, *descr*, *csrVal*, *csrRowPtr*, *csrColInd*)

Performs the analysis phase of the solution of the sparse triangular linear system *op(A) * y = alpha * x*. This needs to be executed only once for a given matrix and operation type.

Parameters

- **trans** – operation to apply to the matrix
- **m** – number of rows of the matrix
- **nnz** – number of nonzeroes of the matrix
- **descr** – matrix descriptor
- **csrVal** – vector of nonzero values of the matrix
- **csrRowPtr** – vector of row pointers of the matrix
- **csrColInd** – vector of column indices of the matrix

Returns the analysis result, which can be used as input to the solve phase

accelerate.cuda.sparse.Sparse.**csrsv_solve**(*trans*, *m*, *alpha*, *descr*, *csrVal*, *csrRowPtr*, *csrColInd*, *info*, *x*, *y*)

Performs the analysis phase of the solution of the sparse triangular linear system *op(A) * y = alpha * x*.

Parameters

- **trans** – operation to apply to the matrix
- **m** – number of rows of the matrix
- **alpha** – scalar
- **descr** – matrix descriptor
- **csrVal** – vector of nonzero values of the matrix
- **csrRowPtr** – vector of row pointers of the matrix
- **csrColInd** – vector of column indices of the matrix
- **info** – the analysis result from *csrsv_analysis()*
- **x** – dense vector
- **y** – dense vector into which the solve result is stored

Returns None

BLAS Level 3

accelerate.cuda.sparse.Sparse.**csrmm**(*transA*, *m*, *n*, *k*, *nnz*, *alpha*, *descrA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *B*, *ldb*, *beta*, *C*, *ldc*)

Matrix-matrix multiplication *C = alpha * op(A) * B + beta * C* where *A* is a sparse matrix in CSR format and *B* and *C* are dense matrices.

Parameters

- **transA** – operation to apply to *A*
- **m** – number of rows of *A*
- **n** – number of columns of *B* and *C*
- **k** – number of columns of *A*

- **nnz** – number of nonzeroes in *A*
- **alpha** – scalar
- **descrA** – matrix descriptor
- **csrValA** – vector of nonzero values of *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **B** – dense matrix
- **ldb** – leading dimension of *B*
- **beta** – scalar
- **C** – dense matrix
- **ldc** – leading dimension of *C*

Returns None

accelerate.cuda.sparse.Sparse.**csrmm2**(*transA*, *transB*, *m*, *n*, *k*, *nnz*, *alpha*, *descrA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *B*, *ldb*, *beta*, *C*, *ldc*)

Matrix-matrix multiplication *C = alpha * op(A) * op(B) + beta * C* where *A* is a sparse matrix in CSR format and *B* and *C* are dense matrices.

Parameters

- **transA** – operation to apply to *A*
- **transB** – operation to apply to *B*
- **m** – number of rows of *A*
- **n** – number of columns of *B* and *C*
- **k** – number of columns of *A*
- **nnz** – number of nonzeroes in *A*
- **alpha** – scalar
- **descrA** – matrix descriptor
- **csrValA** – vector of nonzero values of *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **B** – dense matrix
- **ldb** – leading dimension of *B*
- **beta** – scalar
- **C** – dense matrix
- **ldc** – leading dimension of *C*

Returns None

accelerate.cuda.sparse.Sparse.**csrsm_analysis**(*transA*, *m*, *nnz*, *descrA*, *csrValA*, *csrRowPtrA*, *csrColIndA*)

Performs the analysis phase of the solution of a sparse triangular linear system *op(A) * Y = alpha * X* with multiple right-hand sides where *A* is a sparse matrix in CSR format, and *X* and *Y* are dense matrices.

Parameters

- **transA** – operation to apply to *A*
- **m** – number of rows of *A*
- **nnz** – number of nonzeroes in *A*
- **descrA** – matrix descriptor
- **csrValA** – vector of nonzero values of *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*

Returns the analysis result

accelerate.cuda.sparse.Sparse.**csrsm_solve**(*transA*, *m*, *n*, *alpha*, *descrA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *info*, *X*, *ldx*, *Y*, *ldy*)

Performs the analysis phase of the solution of a sparse triangular linear system *op(A) * Y = alpha * X* with multiple right-hand sides where *A* is a sparse matrix in CSR format, and *X* and *Y* are dense matrices.

Parameters

- **transA** – operation to apply to *A*
- **m** – number of rows of *A*
- **n** – number of columns of *B* and *C*
- **alpha** – scalar
- **descrA** – matrix descriptor
- **csrValA** – vector of nonzero values of *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **info** – the analysis result from *csrsm_analysis()*
- **X** – dense matrix
- **ldx** – leading dimension of *X*
- **Y** – dense matrix
- **ldy** – leading dimension of *Y*

Returns None

Extra Functions

accelerate.cuda.sparse.Sparse.**XcsrgeamNnz**(*m*, *n*, *descrA*, *nnzA*, *csrRowPtrA*, *csrColIndA*, *descrB*, *nnzB*, *csrRowPtrB*, *csrColIndB*, *descrC*, *csrRowPtrC*)

Set up the sparsity pattern for the matrix operation *C* = *alpha * A + beta * B* where *A*, *B*, and *C* are all sparse matrices in CSR format.

Parameters

- **m** – number of rows of all matrices
- **n** – number of columns of all matrices

- **descrA** – matrix descriptor for *A*
- **nnzA** – number of nonzeroes in *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **descrB** – matrix descriptor for *B*
- **nnzB** – number of nonzeroes in *B*
- **csrRowPtrB** – vector of row pointers of *B*
- **csrColIndB** – vector of column indices of *B*
- **descrC** – matrix descriptor for *B*
- **csrRowPtrC** – vector of row pointers of *C*, written to by this method

Returns number of nonzeroes in *C*

accelerate.cuda.sparse.Sparse.**csrgeam**(*m*, *n*, *alpha*, *descrA*, *nnzA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *beta*, *descrB*, *nnzB*, *csrValB*, *csrRowPtrB*, *csrColIndB*, *descrC*, *csrValC*, *csrRowPtrC*, *csrColIndC*)

Performs the the matrix operation *C* = *alpha* * *A* + *beta* * *B* where *A*, *B*, and *C* are all sparse matrices in CSR format.

Parameters

- **m** – number of rows of all matrices
- **n** – number of columns of all matrices
- **alpha** – scalar
- **descrA** – matrix descriptor for *A*
- **nnzA** – number of nonzeroes in *A*
- **csrValA** – vector of nonzero values of *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **beta** – scalar
- **descrB** – matrix descriptor for *B*
- **nnzB** – number of nonzeroes in *B*
- **csrValB** – vector of nonzero values of *B*
- **csrRowPtrB** – vector of row pointers of *B*
- **csrColIndB** – vector of column indices of *B*
- **descrC** – matrix descriptor for *B*
- **csrValC** – vector of nonzero values of *C*
- **csrRowPtrC** – vector of row pointers of *C*
- **csrColIndC** – vector of column indices of *C*

Returns None

accelerate.cuda.sparse.Sparse.**XcsrgemmNnz**(*transA*, *transB*, *m*, *n*, *k*, *descrA*, *nnzA*, *csrRowPtrA*, *csrColIndA*, *descrB*, *nnzB*, *csrRowPtrB*, *csrColIndB*, *descrC*, *csrRowPtrC*)

Set up the sparsity pattern for the matrix operation *C = op(A) * op(B)* where *A*, *B*, and *C* are all sparse matrices in CSR format.

Parameters

- **transA** – operation to apply to *A*
- **transB** – operation to apply to *B*
- **m** – number of rows of *A* and *C*
- **n** – number of columns of *B* and *C*
- **k** – number of columns/rows of *A*/*B*
- **descrA** – matrix descriptor for *A*
- **nnzA** – number of nonzeroes in *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **descrB** – matrix descriptor for *B*
- **nnzB** – number of nonzeroes in *B*
- **csrRowPtrB** – vector of row pointers of *B*
- **csrColIndB** – vector of column indices of *B*
- **descrC** – matrix descriptor for *C*
- **csrRowPtrC** – vector of row pointers of *C*, written by this function

Returns number of nonzeroes in *C*

accelerate.cuda.sparse.Sparse.**csrgemm**(*transA*, *transB*, *m*, *n*, *k*, *descrA*, *nnzA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *descrB*, *nnzB*, *csrValB*, *csrRowPtrB*, *csrColIndB*, *descrC*, *csrValC*, *csrRowPtrC*, *csrColIndC*)

Perform the matrix operation *C = op(A) * op(B)* where *A*, *B*, and *C* are all sparse matrices in CSR format.

Parameters

- **transA** – operation to apply to *A*
- **transB** – operation to apply to *B*
- **m** – number of rows of *A* and *C*
- **n** – number of columns of *B* and *C*
- **k** – number of columns/rows of *A*/*B*
- **descrA** – matrix descriptor for *A*
- **nnzA** – number of nonzeroes in *A*
- **csrValA** – vector of nonzero values in *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **descrB** – matrix descriptor for *B*

- **nnzB** – number of nonzeroes in *B*
- **csrValB** – vector of nonzero values in *B*
- **csrRowPtrB** – vector of row pointers of *B*
- **csrColIndB** – vector of column indices of *B*
- **descrC** – matrix descriptor for *C*
- **csrValC** – vector of nonzero values in *C*
- **csrRowPtrC** – vector of row pointers of *C*
- **csrColIndC** – vector of column indices of *C*

Returns None

accelerate.cuda.sparse.Sparse.**csrgemm_ez**(*A*, *B*, *transA='N'*, *transB='N'*, *descrA=None*, *descrB=None*, *descrC=None*)

Performs the matrix operation *C = op(A) * op(B)* where *A*, *B* and *C* are all sparse matrices in CSR format. This function accepts and returns *accelerate.cuda.sparse.CudaCSRMatrix* matrices, and makes calls to *XcsrgemmNnz()* and *csrgemm()*.

Parameters

- **A** – *accelerate.cuda.sparse.CudaCSRMatrix*
- **B** – *accelerate.cuda.sparse.CudaCSRMatrix*
- **transA** – optional, operation to apply to *A*
- **transB** – optional, operation to apply to *B*
- **descrA** – optional, matrix descriptor for *A*
- **descrB** – optional, matrix descriptor for *B*
- **descrC** – optional, matrix descriptor for *C*

Returns *accelerate.cuda.sparse.CudaCSRMatrix*

Preconditioners

accelerate.cuda.sparse.Sparse.**csric0**(*trans*, *m*, *descr*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *info*)

Computes incomplete Cholesky factorization of a sparse matrix in CSR format with 0 fill-in and no pivoting: *op(A) = R**T * R*. This method must follow a call to *csrsv_analysis()*. The matrix *A* is overwritten with the upper or lower triangular factors *R* or *R**T*.

Parameters

- **trans** – operation to apply to the matrix
- **m** – number of rows and columns of the matrix
- **descr** – matrix descriptor
- **csrValA** – vector of nonzero values in *A*
- **csrRowPtrA** – vector of row pointers of *A*
- **csrColIndA** – vector of column indices of *A*
- **info** – analysis result

Returns None

`accelerate.cuda.sparse.Sparse.`**`csrilu0`**(*trans*, *m*, *descr*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *info*)

Computes incomplete-LU factorization of a sparse matrix in CSR format with 0 fill-in and no pivoting: *op(A) = L * U*. This method must follow a call to `csrsv_analysis()`. The matrix *A* is overwritten with the lower and upper triangular factors *L* and *U*.

Parameters

- **`trans`** – operation to apply to the matrix
- **`m`** – number of rows and columns of the matrix
- **`descr`** – matrix descriptor
- **`csrValA`** – vector of nonzero values in *A*
- **`csrRowPtrA`** – vector of row pointers of *A*
- **`csrColIndA`** – vector of column indices of *A*
- **`info`** – analysis result

Returns None

`accelerate.cuda.sparse.Sparse.`**`gtsv`**(*m*, *n*, *dl*, *d*, *du*, *B*, *ldb*)

Computes the solution of a tridiagonal linear system with multiple right-hand sides: *A * Y = alpha * X*.

Parameters

- **`m`** – the size of the linear system
- **`n`** – the number of right-hand sides in the system
- **`dl`** – dense vector storing the lower-diagonal elements
- **`d`** – dense vector storing the diagonal elements
- **`du`** – dense vector storing the upper-diagonal elements
- **`B`** – dense matrix holding the right-hand sides of the system
- **`ldb`** – the leading dimension of *B*

Returns None

`accelerate.cuda.sparse.Sparse.`**`gtsv_nopivot`**(*m*, *n*, *dl*, *d*, *du*, *B*, *ldb*)

Similar to `gtsv()`, but computes the solution without performing any pivoting.

Parameters

- **`m`** – the size of the linear system
- **`n`** – the number of right-hand sides in the system
- **`dl`** – dense vector storing the lower-diagonal elements
- **`d`** – dense vector storing the diagonal elements
- **`du`** – dense vector storing the upper-diagonal elements
- **`B`** – dense matrix holding the right-hand sides of the system
- **`ldb`** – the leading dimension of *B*

Returns None

`accelerate.cuda.sparse.Sparse.`**`gtsvStridedBatch`**(*m*, *dl*, *d*, *du*, *x*, *batchCount*, *batchStride*)

Computes the solution of *i* tridiagonal linear systems: *A(i) * y(i) = alpha * x(i)*.

Parameters

- **m** – the size of the linear systems
- **dl** – stacked dense vector storing the lower-diagonal elements of each system
- **d** – stacked dense vector storing the diagonal elements of each system
- **du** – stacked dense vector storing the upper-diagonal elements of each system
- **x** – dense matrix holding the right-hand sides of the systems
- **batchCount** – number of systems to solve
- **batchStride** – number of elements separating the vectors of each system

Returns None

Format Conversion

accelerate.cuda.sparse.Sparse.**bsr2csr**(*dirA*, *mb*, *nb*, *descrA*, *bsrValA*, *bsrRowPtrA*, *bsrColIndA*, *blockDim*, *descrC*, *csrValC*, *csrRowPtrC*, *csrColIndC*)

Convert the sparse matrix *A* in BSR format to CSR format, stored in *C*.

Parameters

- **dirA** – row ('R') or column ('C') orientation of block storage
- **mb** – number of block rows of *A*
- **nb** – number of block columns of *A*
- **descrA** – matrix descriptor for *A*
- **bsrValA** – vector of nonzero values of *A*
- **bsrRowPtrA** – vector of block row pointers of *A*
- **bsrColIndA** – vector of block column indices of *A*
- **blockDim** – block dimension of *A*
- **descrC** – matrix descriptor for *C*
- **csrValA** – vector of nonzero values in *C*
- **csrRowPtrA** – vector of row pointers of *C*
- **csrColIndA** – vector of column indices of *C*

Returns None

accelerate.cuda.sparse.Sparse.**Xcoo2csr**(*cooRowInd*, *nnz*, *m*, *csrRowPtr*)

Converts an array containing uncompressed row indices corresponding to the COO format into into an array of compressed row pointers corresponding to the CSR format.

Parameters

- **cooRowInd** – integer array of uncompressed row indices
- **nnz** – number of nonzeroes
- **m** – number of matrix rows
- **csrRowPtr** – vector of row pointers to be written to

Returns None

accelerate.cuda.sparse.Sparse.**csc2dense**(*m*, *n*, *descrA*, *cscValA*, *cscRowIndA*, *cscColPtrA*, *A*, *lda*)

Convert the sparse matrix *A* in CSC format into a dense matrix.

Parameters

- **m** – number of rows of *A*
- **n** – number of columns of *A*
- **descrA** – matrix descriptor for *A*
- **cscValA** – values in the CSC representation of *A*
- **cscRowIndA** – row indices in the CSC representation of *A*
- **cscColPtrA** – column pointers in the CSC representation of *A*
- **A** – dense matrix representation of *A* to be written by this function.
- **lda** – leading dimension of *A*

Returns None

accelerate.cuda.sparse.Sparse.**Xcsr2bsrNnz**(*dirA*, *m*, *n*, *descrA*, *csrRowPtrA*, *csrColIndA*, *blockDim*, *descrC*, *bsrRowPtrC*)

Performs the analysis necessary for converting a matrix in CSR format into BSR format.

Parameters

- **dirA** – row (‘R’) or column (‘C’) orientation of block storage
- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **descrA** – matrix descriptor for input matrix *A*
- **csrRowPtrA** – row pointers of matrix
- **csrColIndA** – column indices of matrix
- **blockDim** – block dimension of output matrix *C*
- **descrC** – matrix descriptor for output matrix *C*

Returns number of nonzeroes of matrix

accelerate.cuda.sparse.Sparse.**csr2bsr**(*dirA*, *m*, *n*, *descrA*, *csrValA*, *csrRowPtrA*, *csrColIndA*, *blockDim*, *descrC*, *bsrValC*, *bsrRowPtrC*, *bsrColIndC*)

Performs conversion of a matrix from CSR format into BSR format.

Parameters

- **dirA** – row (‘R’) or column (‘C’) orientation of block storage
- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **descrA** – matrix descriptor for input matrix *A*
- **csrValA** – nonzero values of matrix
- **csrRowPtrA** – row pointers of matrix
- **csrColIndA** – column indices of matrix
- **blockDim** – block dimension of output matrix *C*

- **descrC** – matrix descriptor for output matrix *C*
- **bsrValC** – nonzero values of output matrix *C*
- **bsrRowPtrC** – block row pointers of output matrix *C*
- **bsrColIndC** – block column indices of output matrix *C*

Returns number of nonzeroes of matrix

accelerate.cuda.sparse.Sparse.**Xcsr2coo**(*csrRowPtr*, *nnz*, *m*, *cooRowInd*)

Converts an array of compressed row pointers corresponding to the CSR format into an array of uncompressed row indices corresponding to the COO format.

Parameters

- **csrRowPtr** – vector of row pointers
- **nnz** – number of nonzeroes
- **m** – number of rows of matrix
- **cooRowInd** – vector of uncompressed row indices written by this function

Returns None

accelerate.cuda.sparse.Sparse.**csr2csc**(*m*, *n*, *nnz*, *csrVal*, *csrRowPtr*, *csrColInd*, *cscVal*, *cscRowInd*, *cscColPtr*, *copyValues*)

Converts a sparse matrix in CSR format into a sparse matrix in CSC format.

Parameters

- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **nnz** – number of nonzeroes of the matrix
- **csrVal** – values in the CSR representation
- **csrRowPtr** – row indices in the CSR representation
- **csrColInd** – column pointers in the CSR representation
- **cscVal** – values in the CSC representation
- **cscRowInd** – row indices in the CSC representation
- **cscColPtr** – column pointers in the CSC representation
- **copyValues** – *'N'* or *'S'* for symbolic or numeric copy of values

Returns None

accelerate.cuda.sparse.Sparse.**csr2dense**(*m*, *n*, *descr*, *csrVal*, *csrRowPtr*, *csrColInd*, *A*, *lda*)

Convert a sparse matrix in CSR format into dense format.

Parameters

- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **descr** – matrix descriptor
- **csrVal** – values in the CSR representation
- **csrRowPtr** – row indices in the CSR representation
- **csrColInd** – column pointers in the CSR representation

- **A** – the dense representation, written to by this function
- **lda** – leading dimension of the matrix

Returns None

`accelerate.cuda.sparse.Sparse.`**`dense2csc`**(*m*, *n*, *descrA*, *A*, *lda*, *nnzPerCol*, *cscVal*, *cscRowInd*, *cscColPtr*)

Convert a dense matrix into a sparse matrix in CSC format. The *nnzPerCol* parameter may be computed with a call to `nnz()`.

Parameters

- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **descrA** – matrix descriptor
- **A** – the matrix in dense format
- **lda** – leading dimension of the matrix
- **nnzPerCol** – array containing the number of nonzero elements per column
- **cscVal** – values in the CSC representation
- **cscRowInd** – row indices in the CSC representation
- **cscColPtr** – column pointers in the CSC representation

Returns None

`accelerate.cuda.sparse.Sparse.`**`dense2csr`**(*m*, *n*, *descrA*, *A*, *lda*, *nnzPerRow*, *csrVal*, *csrRowPtr*, *csrColInd*)

Convert a dense matrix into a sparse matrix in CSR format. The *nnzPerRow* parameter may be computed with a call to `nnz()`.

Parameters

- **m** – number of rows of matrix
- **n** – number of columns of matrix
- **descrA** – matrix descriptor
- **A** – the matrix in dense format
- **lda** – leading dimension of the matrix
- **nnzPerRow** – array containing the number of nonzero elements per row
- **csrVal** – values in the CSR representation
- **csrRowPtr** – row indices in the CSR representation
- **csrColInd** – column pointers in the CSR representation

Returns None

`accelerate.cuda.sparse.Sparse.`**`nnz`**(*dirA*, *m*, *n*, *descrA*, *A*, *lda*, *nnzPerRowCol*)

Computes the number of nonzero elements per row or column of a dense matrix, and the total number of nonzero elements in the matrix.

Parameters

- **dirA** – *'R'* for the number of nonzeroes per row, or *'C'* for per column.
- **m** – number of rows of matrix

- **n** – number of columns of matrix
- **descrA** – matrix descriptor
- **A** – the matrix
- **lda** – leading dimension of the matrix
- **nnzPerRowCol** – array to contain the number of nonzeroes per row or column

Returns total number of nonzeroes in the matrix

cuFFT

Provides FFT and inverse FFT for 1D, 2D and 3D arrays. See NVIDIA cuFFT.

Note: cuFFT only supports FFT operations on numpy.float32, numpy float64, numpy.complex64, numpy.complex128 with C-contiguous datalayout.

Forward FFT

accelerate.cuda.fft.**fft**(*ary*, *out*[, *stream*])

accelerate.cuda.fft.**fft_inplace**(*ary*[, *stream*])

Parameters

- **ary** – The input array. The inplace version stores the result in here.
- **out** – The output array for non-inplace versions.
- **stream** – The CUDA stream in which all operations will take place.

Inverse FFT

accelerate.cuda.fft.**ifft**(*ary*, *out*[, *stream*])

accelerate.cuda.fft.**ifft_inplace**(*ary*[, *stream*])

Parameters

- **ary** – The input array. The inplace version stores the result in here.
- **out** – The output array for non-inplace versions.
- **stream** – The CUDA stream in which all operations will take place.

FFTPlan

cuRAND

Provides *pseudo-random number generator* (PRNG) and *quasi-random generator* (QRNG). See NVIDIA cuRAND.

CUDA Sorting

Accelerate provides routines for sorting arrays on CUDA GPUs.

Sorting Large Arrays

The `accelerate.cuda.sorting.RadixSort` class is recommended for sorting large (approx. more than 1 million items) arrays of numeric types.

Sorting Many Small Arrays

Using `accelerate.cuda.sorting.RadixSort` on small (approx. less than 1 million items) arrays has significant overhead due to multiple kernel launches.

A better alternative is to use `accelerate.cuda.sorting.segmented_sort()`-which launches a single kernel for sorting a batch of many small arrays.

Math Kernel Library

The Math Kernel Library provides BLAS, LAPACK, and other math routines. For more information, see Intel's site.

The following functions, FFT variants and accelerated UFuncs are available for use with Accelerate:

MKL utility functions

The *accelerate.mkl* module contains a set of functions to configure and retrieve information about the underlying Intel MKL library.

Reference

`accelerate.mkl.`**`set_num_threads`**`(`*n*`)`

Set the number of threads Intel MKL should use. This is only a hint, and no guarantee is made this number of threads will actually be used. This function takes precedence over the environment variable MKL_NUM_THREADS.

This function wraps the Intel MKL function `mkl_set_num_threads`.

`accelerate.mkl.`**`get_max_threads`**`()` → int

Return the number of threads Intel MKL is targeting for parallelism.

This function wraps the Intel MKL function `mkl_get_max_threads`.

`accelerate.mkl.`**`get_version_string`**`()` → str

Return the Intel MKL version information as a string.

This function wraps the Intel MKL function `mkl_get_version_string`.

`accelerate.mkl.`**`mem_stat`**`()` → int, int

Return a (bytes, blocks) tuple of memory usage statistics about the underlying Intel MKL memory allocator. The return tuple contains:

- The total number of bytes allocated (bytes).
- The number of allocated blocks (blocks).

This function wraps the Intel MKL function `mkl_mem_stat`.

`accelerate.mkl.`**`get_cpu_clocks`**`()` → int

Return elapsed CPU clocks as an integer. This may be useful when timing short intervals with high resolution. Note the result is limited to unsigned 64 bit integers, so wrapping of elapsed time is possible.

This function wraps the Intel MKL function `mkl_get_cpu_clocks`.

`accelerate.mkl.`**`get_cpu_frequency`**`()` → int

Return CPU frequency in GHz as a float. Note the result may vary from run to run as it returns the current frequency. That frequency can change in some systems due to several factors (i.e. power management).

This function wraps the Intel MKL function `mkl_get_cpu_frequency`.

MKL BLAS functions

The *accelerate.mkl.blas* module contains a subset of BLAS functions implemented by means of the underlying Intel MKL library.

Reference

`accelerate.mkl.blas.`**`dot`**(*x*, *y*)

Compute and return the vector dot product of *x* and *y*.

Parameters

- **`x`** – one-dimensional array
- **`y`** – one-dimensional array

Return type result

Note the input arguments may be copied to adjust their types.

Example

```
alpha = 1.+1.j
A = np.arange(16, dtype=np.float64).reshape(4,4)
x = np.arange(4, dtype=np.float64)
beta = 0.
y = np.arange(4, dtype=np.float64)
result = blas.gemv('N', alpha, A, x, beta, y)
```

MKL FFT

The speed-boosted variants of NumPy's FFT operations are accessible in the `numpy.fft` package, and the `accelerate.mkl.fftpack` package. The following functions in these packages are accelerated using MKL:

Function	Description
`fft(a, n=None, axis=-1)`	1-dimensional forward transform
`ifft(a, n=None, axis=-1)`	1-dimensional inverse transform
`rfft(a, n=None, axis=-1)`	1-dimensional forward transform of purely real data
`irfft(a, n=None, axis=-1)`	1-dimensional inverse transform of purely real data
`hfft(a, n=None, axis=-1)`	Hermite transform
`ihfft(a, n=None, axis=-1)`	Inverse Hermite transform
`fftn(a, s=None, axes=None)`	N-dimensional forward transform
`ifftn(a, s=None, axes=None)`	N-dimensional inverse transform
`fft2(a, s=None, axes=(-2, -1))`	2-dimensional forward transform
`ifft2(a, s=None, axes=(-2, -1))`	2-dimensional inverse transform
`rfftn(a, s=None, axes=None)`	N-dimensional forward transform of purely real data
`rfft2(a, s=None, axes=(-2, -1))`	2-dimensional forward transform of purely real data
`irfftn(a, s=None, axes=None)`	N-dimensional inverse transform of purely real data
`irfft2(a, s=None, axes=(-2, -1))`	2-dimensional inverse transform of purely real data

For further information on these functions, please refer to the Numpy documentation: `numpy.fft`.

Accelerated UFuncs

Variants of some Numpy UFuncs that use Intel's Vector Math Library (VML) are found in the `accelerate.mkl` package in the `ufuncs` object. In comparison to Numpy's built-in UFuncs, Accelerated UFuncs have the following properties:

Performance Because Accelerated UFuncs call functions from VML, which is a library optimised for high performance using multiple threads and SIMD instructions, performance may be increased when calling Accelerated UFuncs in comparison to calling Numpy UFuncs. The performance increase will only be obtained for contiguous arguments. For non-contiguous arguments, performance comparable to Numpy's will be observed.

Accuracy

- Accelerated UFuncs produce similar results to their Numpy equivalents for the range of finite values, up to a given relative tolerance.
- The tolerance varies between functions and data types, and is specified for each combination in the following section.
- For the range of infinite and NaN values, the majority of results computed by Accelerated UFuncs will be equal to those computed by their Numpy counterparts. However, this is not guaranteed; for example, an input for which the Numpy UFunc produces a result of `x + inf*j`, the equivalent Accelerated UFunc may produce a result of `y + inf*j`, where `x != y`.
- Towards the edge of the domain of a data type (e.g. near `3.4e+38` for `float` and `1.79e+308` for `double` and values of similar magnitude at the negative end of the domain) some Accelerated UFuncs may produce results which differ from Numpy UFuncs, or may raise `FloatingPointError` or `ZeroDivisionError` exceptions. These functions are marked *domain edge warning* in the *Accuracy* column of the tables in the following sections.
- Some numpy implementations of functions that operate in the complex domain have branch cuts which differ from those in MKL, where this has been identified in testing these functions are marked *branch cut warning* in the *Accuracy* column of the tables in the following sections. This warning may also be present in the case of implementations which contain incorrect branch cuts.
- Denormal input values may be treated as zero by Accelerated UFuncs, and denormal output values may be flushed to zero.

- The accuracy reported is approximate and related to the least accurate value encountered when testing with contiguous input arguments. Non-contiguous arguments are more directly exposed to system math library implementations and so their accuracy is not commented on.

Exception handling For the range of finite values, exceptions will not be raised by Accelerated UFuncs, just as they would not by Numpy. For the infinite and NaN ranges, Accelerated UFuncs may raise `FloatingPointError` or `ZeroDivisionError` exceptions in cases when Numpy would not, and vice-versa.

Supported functions are described in the following sections.

Arithmetic Functions

Function	Type	Accuracy
`add(x, y)`	`f4`	`rtol=1.0e-7`
	`f8`	`rtol=1.0e-15`
	`c8`	`rtol=1.0e-7`
	`c16`	`rtol=1.0e-15`
`subtract(x, y)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-7`
	`c16`	`rtol=1.e-15`
`square(x)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
`multiply(x, y)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-6`
	`c16`	`rtol=1.e-15`
`absolute(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`

Power and Root Functions

Function	Type	Accuracy
`reciprocal(x)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
`true_divide(x, y)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-6`
	`c16`	`rtol=1.e-15`
`sqrt(x, y)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`power(x, y)`	`f4`	`rtol=1.e-7`
	`f8`	`rtol=1.e-15`
`hypot(x, y)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`

Exponential and Logarithmic Functions

Function	Type	Accuracy
`exp(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`expm1(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
`log(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-5`
	`c16`	`rtol=1.e-13`
`log10(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-5`, domain edge warning
	`c16`	`rtol=1.e-13`, domain edge warning
`log1p(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`

Trigonometric Functions

Function	Type	Accuracy
`cos(x)`	`f4`	`rtol=1.e-6`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`sin(x)`	`f4`	`rtol=1.e-5`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`tan(x)`	`f4`	`rtol=1.e-6`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=5.e-4`, domain edge warning
	`c16`	`rtol=5.e-14`, domain edge warning
`arccos(x)`	`f4`	`rtol=1.e-6`, branch cut warning
	`f8`	`rtol=1.e-15`, branch cut warning
	`c8`	**`rtol=1.e-5`, domain edge warning,** branch cut warning
	`c16`	**`rtol=1.e-14`, domain edge warning,** branch cut warning
`arcsin(x)`	`f4`	`rtol=1.e-6`, branch cut warning
	`f8`	`rtol=1.e-15`, branch cut warning
	`c8`	**`rtol=1.e-3`, domain edge warning,** branch cut warning
	`c16`	**`rtol=1.e-12`, domain edge warning,** branch cut warning
`arctan(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-6`
	`c16`	`rtol=1.e-15`
`arctan2(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`

Hyperbolic Functions

Function	Type	Accuracy
`cosh(x)`	`f4`	`rtol=1.e-6`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`sinh(x)`	`f4`	`rtol=1.e-6`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`tanh(x)`	`f4`	`rtol=1.e-6`, domain edge warning
	`f8`	`rtol=1.e-15`, domain edge warning
	`c8`	`rtol=1.e-6`, domain edge warning
	`c16`	`rtol=1.e-15`, domain edge warning
`arccosh(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=1.e-5`, domain edge warning
	`c16`	`rtol=1.e-14`, domain edge warning
`arcsinh(x)`	`f4`	`rtol=1.e-6`
	`f8`	`rtol=1.e-15`
	`c8`	`rtol=5.e-5`, domain edge warning
	`c16`	`rtol=1.e-13`, domain edge warning
`arctanh(x)`	`f4`	`rtol=1.e-5`, branch cut warning
	`f8`	`rtol=5.e-11`, branch cut warning
	`c8`	**`rtol=1.e-4`, domain edge warning,** branch cut warning
	`c16`	**`rtol=5.e-14`, domain edge warning,** branch cut warning

Profiling

The Python standard library includes code profiling functionality. When active, function invocations and the time spent on them are recorded.

The `accelerate.profiler` module extends this functionality by also recording the functions' signatures, which are useful because often the precise control flow–and thus function performance– depends on the argument types. For NumPy array types, this includes not only the `dtype` attribute, but also the array's shape.

Profiling example

The following code defines a simple dot function and profiles it without signatures to match the behaviour of the Python standard library profile module:

```
from accelerate import profiler
import numpy as np

def dot(a, b):
  sum=0
  for i in range(len(a)):
      sum += a[i]*b[i]
  return sum

a = np.arange(16, dtype=np.float32)
b = np.arange(16, dtype=np.float32)

p = profiler.Profile(signatures=False)
p.enable()
dot(a, b)
p.disable()
p.print_stats()
```

This generates the following output:

```
      3 function calls in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     1    0.000    0.000    0.000    0.000 builtins.len
     1    0.000    0.000    0.000    0.000 dot.py:7(dot)
     1    0.000    0.000    0.000    0.000 {method 'disable' of 'prof.Profiler'␣
↪objects}
```

NOTE: By default, the `Profile` constructor's `signature` flag is set to `True`, resulting in the following output:

```
      3 function calls (2 primitive calls) in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     1    0.000    0.000    0.000    0.000 dot.py:1(disable())
   2/1    0.000    0.000    0.000    0.000 dot.py:7(dot(a:ndarray(dtype=float32,␣
↪shape=(16,)), b:ndarray(dtype=float32, shape=(16,))))
```

In the results for more realistic code, the call graph–and thus the table of function calls–is much bigger, making working with data in tabular form inconvenient.

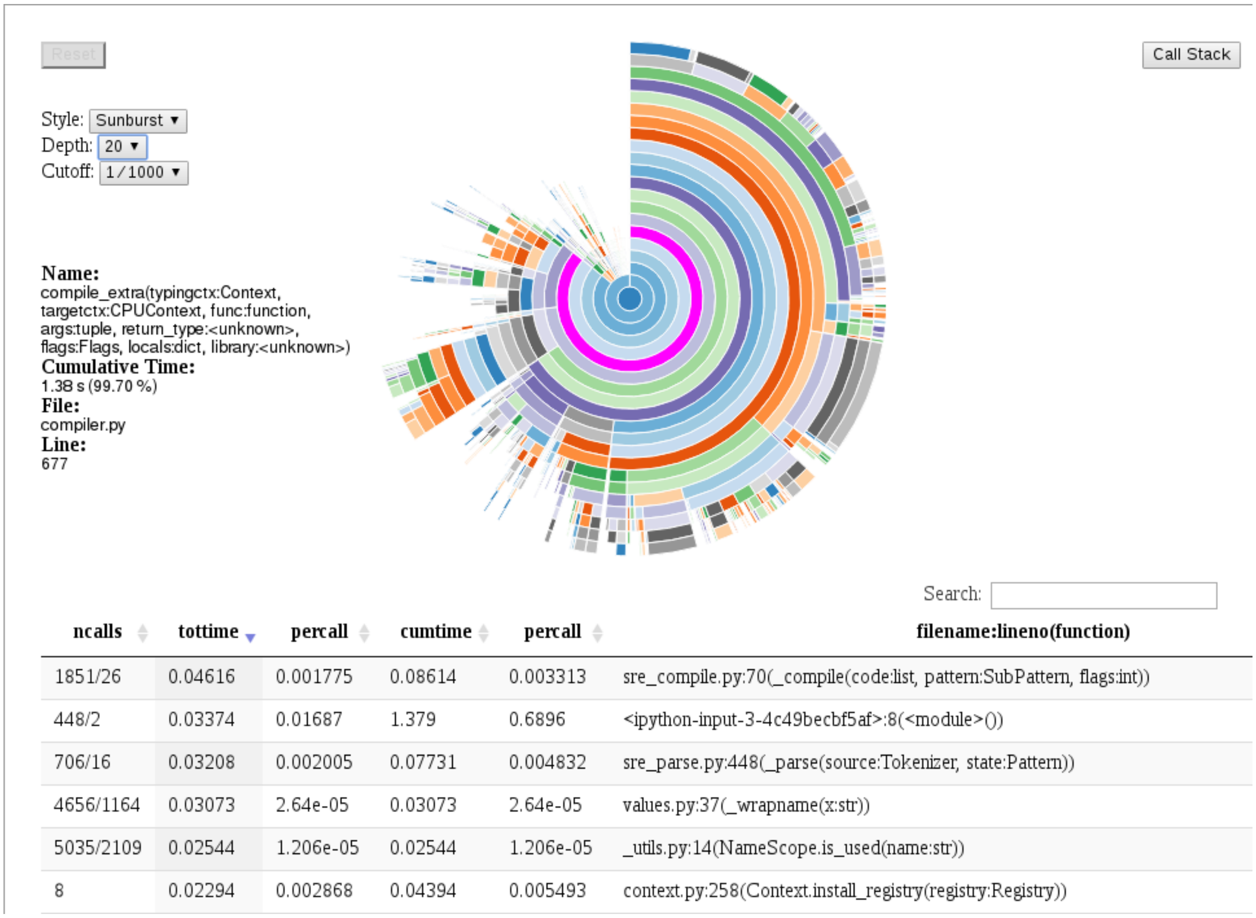
To solve this, the `accelerate.profiler` module also provides functionality to visualize the data. Instead of calling the `print_stats()` method, call the `accelerate.profiler.plot()` function.

NOTE: At this time, the `accelerate.profiler.plot()` function may only be called from inside a notebook.

If the code above was executed from inside a notebook, run the following code:

```
In [3]: profiler.plot(p)
```

Which generates the following output:



Style: Sunburst
Depth: 20
Cutoff: 1/1000

Name:
compile_extra(typingctx:Context, targetctx:CPUContext, func:function, args:tuple, return_type:<unknown>, flags:Flags, locals:dict, library:<unknown>)
Cumulative Time:
1.38 s (99.70 %)
File:
compiler.py
Line:
677

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1851/26	0.04616	0.001775	0.08614	0.003313	sre_compile.py:70(_compile(code:list, pattern:SubPattern, flags:int))
448/2	0.03374	0.01687	1.379	0.6896	<ipython-input-3-4c49becbf5af>:8(<module>())
706/16	0.03208	0.002005	0.07731	0.004832	sre_parse.py:448(_parse(source:Tokenizer, state:Pattern))
4656/1164	0.03073	2.64e-05	0.03073	2.64e-05	values.py:37(_wrapname(x:str))
5035/2109	0.02544	1.206e-05	0.02544	1.206e-05	_utils.py:14(NameScope.is_used(name:str))
8	0.02294	0.002868	0.04394	0.005493	context.py:258(Context.install_registry(registry:Registry))

Accelerate.profiler API

Environment variables

`ACCELERATE_WARNINGS`

If set to anything but 0 (zero), Accelerate may issue performance warnings, such as when input arguments need to be copied to adjust their data layout, or types, to match particular backend requirements.

License agreement

Anaconda Accelerate End User License Agreement

Anaconda Accelerate (“the Software Product” or “Accelerate”) and accompanying documentation is licensed and not sold. The Software Product is protected by copyright laws and treaties, as well as laws and treaties related to other forms of intellectual property. Continuum Analytics, Inc. (dba Anaconda, Inc.) or its subsidiaries, affiliates, and suppliers (collectively “Continuum”) own intellectual property rights in the Software Product. The Licensee’s (“you” or “your”) license to download, use, copy, or change the Software Product is subject to these rights and to all the terms and conditions of this End User License Agreement (“Agreement”).

Acceptance

YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY SELECTING THE “ACCEPT” OPTION AND DOWNLOADING THE SOFTWARE PRODUCT OR BY INSTALLING, USING, OR COPYING THE SOFTWARE PRODUCT. YOU MUST AGREE TO ALL OF THE TERMS OF THIS AGREEMENT BEFORE YOU WILL BE ALLOWED TO DOWNLOAD THE SOFTWARE PRODUCT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST SELECT “DECLINE” AND YOU MUST NOT INSTALL, USE, OR COPY THE SOFTWARE PRODUCT.

Other Vendor Tools

In addition to Continuum’s proprietary software, the Software product may contain other software packages from other sources (“Other Vendor Tools”). These are available on an “as is” basis and subject to their individual license agreements. These licenses are available either in the package distributed with the Software Product or via request to info@anaconda.com. Any binary packages of Other Vendor Tools you obtain via Accelerate are subject to their individual licenses as well as the terms of this EULA. Continuum reserves the right to change which Other Vendor Tools are provided in Accelerate.

Trial Period

You have the right to use Accelerate on a single computer or group of computers for 30 days with no license. After 30 days you must purchase an appropriate license to use the software on one or more machines or stop using the software and remove it from all of your machines on which you installed the software.

License Grant

This Agreement entitles you to install and use one copy of the Software Product on as many machines as you will personally use. The Software Product is licensed to a particular user. Only the user to whom the software is licensed may use the software. You must obtain a license for as many users as you wish to use the software. In addition, you may make archival copies of the Software Product Installer.

Right to Updates

This license entitles you to updates to the Software Product for one year from the time of payment. The Software Product will continue to function and you may continue to use The Software Product and any updates you have received for as long as you would like however you will no longer be able to receive updates from Continuum unless this License is renewed. Please contact sales@anaconda.com with any questions or concerns.

Restrictions on Transfer

Without first obtaining the express written consent of Continuum, you may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software Product.

Restrictions on Use

You may not use, copy, or install the Software Product on any system where more than one user will be able to use the software unless you have purchased a license for each user of the system. You may not decompile, "reverse-engineer", disassemble, or otherwise attempt to derive the source code for the Software Product.

Restrictions on Alteration

You may not modify the Software Product or create any derivative work of the Software Product or its accompanying documentation. Derivative works include but are not limited to translations. You may not alter any files or libraries in any portion of the Software Product.

Restrictions on Copying

You may not copy any part of the Software Product except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make archival copies of the Software Product installer.

Limited Software Product Warranty

For a period of 60 days from the date of shipment or from the date that you download the Software Product, as applicable, Continuum warrants that when properly installed and used under normal conditions, the Software Product will perform substantially as advertised.

Disclaimer of Warranties and Limitation of Liability

UNLESS OTHERWISE EXPLICITLY AGREED TO IN WRITING BY CONTINUUM, CONTINUUM MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE PRODUCT.

Continuum makes no warranty that the Software Product will meet your requirements or operate under your specific conditions of use. Continuum makes no warranty that operation of the Software Product will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE PRODUCT SUFFICIENTLY MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE PRODUCT TO MEET YOUR REQUIREMENTS. CONTINUUM WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE. UNDER NO CIRCUMSTANCES SHALL CONTINUUM, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE PRODUCT, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF CONTINUUM OR ANY OTHER PARTY, EVEN IF CONTINUUM IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS CONTINUUM'S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

Limitation of Remedies and Damages

Your remedy for a breach of this Agreement or of any warranty included in this Agreement is the correction or replacement of the Software Product. Selection of whether to correct or replace shall be solely at the discretion of Continuum. Continuum reserves the right to substitute a functionally equivalent copy of the Software Product as a replacement. If Continuum is unable to provide a replacement or substitute Software Product or corrections to the Software Product, your sole alternate remedy shall be a refund of the purchase price for the Software Product exclusive of any costs for shipping and handling. Any claim must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by Continuum to have been caused by you. All limited warranties on the Software Product are granted only to you and are non-transferable. You agree to indemnify and hold Continuum harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

Governing Law, Jurisdiction and Costs

This Agreement is governed by the laws of Texas, without regard to Texas's conflict or choice of law provisions.

Export Regulations

Any use or distribution of Accelerate is made under conditions that the user and/or distributor is in full compliance with all export and other governing laws of the United States of America, including full and ongoing compliance with the Export Administration Regulations (EAR) of the United States Department of Commerce. See www.commerce.gov/ and http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear. Use or distribution of Continuum software products to any persons, entities or countries currently under US sanctions is strictly prohibited. Accelerate is classified with an ECCN of 5D992 with no license required for export to non-embargoed countries.

The United States currently has embargoes against Cuba, Iran, North Korea, Sudan and Syria. The exportation, re-exportation, sale or supply, directly or indirectly, from the United States, or by a U.S. person wherever located, of any Continuum software to any of these countries is strictly prohibited without prior authorization by the United States Government. By accepting this Agreement, you represent to Continuum that you will comply with all applicable export regulations for Accelerate.

Severability

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

Release notes

Release notes

Version 2.3.1

This is a minor update to bump the required version of Numba to 0.30 and to support Python 3.6.

Version 2.3.0

This release of Accelerate is a feature release, containing these additions:

- A new BLAS wrapper API on top of MKL.
- MKL 11.3.3 is now supported.
- Numba 0.26 is supported-and is required.

Version 2.2.0

This release of Accelerate is a feature release, containing two additions:

- Expanded the support of Intel MKL accelerated ufuncs with trigonometric and hyperbolic functions.
- Numba 0.25 is supported, and is a requirement.

Version 2.1.0

This release of Accelerate is a feature release, containing 2 additions:

- Profiling tools
  - An enhanced version of the Python profiler that captures function arguments, including shapes and dtypes of Numpy arrays.
  - Integration of tools for visualising profiles in Jupyter notebooks, allowing interactive experimentation.
- Accelerated UFuncs, which provide a speed improvement over Numpy's built-in UFuncs by using Intel MKL.

Version 2.0.2

This update adds support for updated versions of Numba and MKL:

- Numba 0.23 is now supported, and is a requirement.
- MKL 11.3.1, standard in Anaconda 2.5, is now supported.

Version 2.0.1

This minor update to Accelerate adds no functional changes, but provides additional clarification of the the relationship between the Accelerate and NumbaPro conda packages upon installation.

Version 2.0

NumbaPro has been deprecated, and its code generation features have been moved into open-source Numba. The CUDA library functions have been moved into Accelerate, along with some Intel MKL functionality. High-level functions and access to additional native library implementations will be added in future releases of Accelerate, and there will be no further updates to NumbaPro.

A NumbaPro compatibility layer (listed as release 0.22.0 of NumbaPro) provides access to the new Accelerate packages through the old NumbaPro package names. This avoids the need to change any existing code immediately for use with Accelerate. A warning will be generated upon import of the compatibility layer, to highlight the deprecation of the NumbaPro package.

CUDA libraries

CUDA library functionality is equivalent to that in NumbaPro 0.21, with the following packages renamed:

NumbaPro Package	Accelerate package
`numbapro.cudalib.cublas`	`accelerate.cuda.blas`
`numbapro.cudalib.cufft`	`accelerate.cuda.fft`
`numbapro.cudalib.curand`	`accelerate.cuda.rand`
`numbapro.cudalib.cusparse`	`accelerate.cuda.sparse`
`numbapro.cudalib.sorting`	`accelerate.cuda.sorting`

Code generation

The `vectorize` targets `parallel` and `cuda` can now be accessed with Numba, as can the `cuda.reduce` decorator. Printing of integers and floating point values from CUDA kernels is also possible in Numba, and no longer requires NumbaPro or Accelerate to be imported.

Intel MKL

The MKL FFT implementation is available in the package `accelerate.mkl.fftpack`, and MKL service functions are available in the `accelerate.mkl` package.

Previous versions

The following documentation is provided for users who have not yet upgraded to the current version of Accelerate:

Anaconda Accelerate

cuBLAS

CUDA Libraries

cuFFT

cuRAND

cuSPARSE

Anaconda Accelerate END USER LICENSE AGREEMENT

MKL

MKL utility functions

MKL FFT

Accelerated UFuncs

Profiling

Release Notes

CUDA Sorting

Anaconda Accelerate

cuBLAS

CUDA Libraries

cuFFT

cuRAND

cuSPARSE

Anaconda Accelerate END USER LICENSE AGREEMENT

MKL

MKL utility functions

MKL FFT

Accelerated UFuncs

Profiling

1.4.2 Anaconda Adam

Server & Cluster Installer & Manager

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it.

Anaconda Adam is a command line interface (CLI) utility for Linux that lets you:

- Install and manage these Anaconda® platform components:

Platform component	Description
Anaconda Enterprise Notebooks	Collaborative multi-user notebooks and projects
Anaconda Repository	On-premise repository for conda packages and environments
Anaconda Scale	Distributed package management and computation framework

- Create, provision and remove clusters.
- Perform administration tasks on Amazon Elastic Compute Cloud (EC2) servers.
- Manage Anaconda packages.
- Execute commands on multiple nodes.

You must have an Anaconda Enterprise license to install Anaconda platform components. If you would like to purchase Anaconda platform, please contact sales@anaconda.com.

You can use Adam with:

- Distributed computation frameworks such as Spark or Dask.
- Enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP.
- HPC clusters.
- A collection of virtual machines or containers.

Installation

Installation requirements

Make sure your system meets the requirements below before installing and using Anaconda Adam.

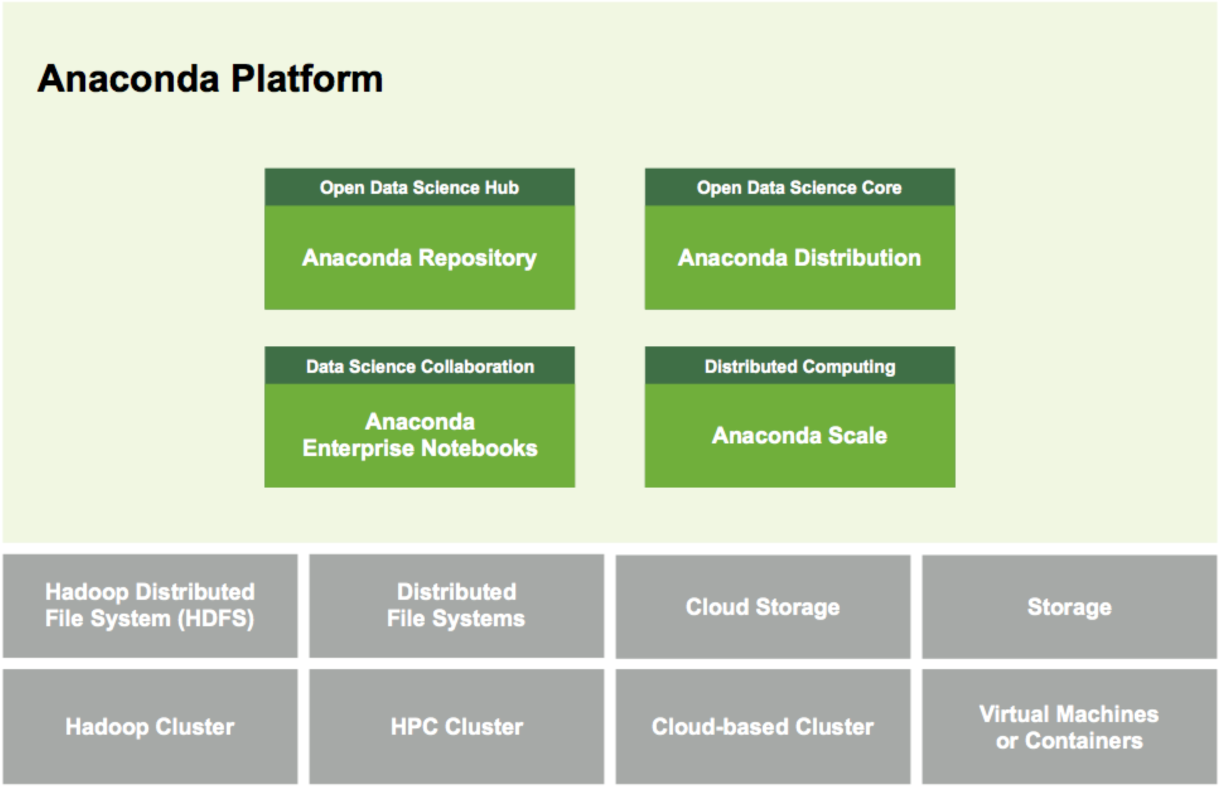
Hardware and operating system

To install Adam, you must have one or more 64-bit Linux machines with one of the following Linux distributions:

- RHEL/CentOS 6 or 7
- Ubuntu 14.04 or 16.04

and the following commands installed:

- `ssh`
- `openssl`
- `curl`



- `md5sum`

License

To install Adam, you must have a valid Anaconda Enterprise license.

Account access and security

To install Adam, you need one of the following access options on the target machines:

- Access to the root user with the root password.
- Access to the root user with an SSH key pair.
- Passwordless SSH/sudo enabled for a user account.
- Password-based SSH and passwordless sudo enabled for a user account.

The root or user credentials used for installation must be the same on all of the target nodes.

Network

You must configure network and security settings to allow access to and from each node–for example, using IPTables or SELinux.

Anaconda platform components use the default network ports listed below. You can configure these ports when you define a cluster profile.

Platform Component	Port
Anaconda Adam (SSH)	22
Anaconda Adam (Salt)	14505
Anaconda Adam (Salt)	14506
Anaconda Adam (Salt REST API)	18000
Anaconda Enterprise Notebooks (UI)	80
Anaconda Enterprise Notebooks (Gateway)	8089
Anaconda Repository (UI)	8080
Anaconda Scale (Dask/Distributed)	8786
Anaconda Scale (Dask/Distributed UI)	8787
Anaconda Scale (Jupyter Notebook UI)	8888

Network ports with a browser user interface (UI) must be accessible by machines outside of the network of Anaconda platform nodes.

Network ports without a browser UI must be accessible between the other Anaconda platform nodes.

Platform components

See the documentation below for information on CPU, memory and disk space requirements for Anaconda platform components.

- *Anaconda Enterprise Notebooks*.
- *Anaconda Repository*.
- *Anaconda Scale*.

Python considerations

Adam and Anaconda platform components do not affect existing framework installations of Python. Adam operates independently from the system installation of Python, while Anaconda platform components install their own versions of Python via Anaconda or Miniconda.

Standard installation

Downloading the Adam installer

To download the installer, run one of the commands below on the host machine where you want to manage the Anaconda platform:

```
wget https://anaconda.org/anaconda-adam/adam-installer/4.4.0/download/adam-installer-
↪4.4.0-Linux-x86_64.sh
```

Or:

```
curl -LO https://anaconda.org/anaconda-adam/adam-installer/4.4.0/download/adam-
↪installer-4.4.0-Linux-x86_64.sh
```

Installing Adam

1. Run the installer, specifying the installation directory:

   ```
   bash adam-installer-4.4.0-Linux-x86_64.sh -b -p ~/adam
   ```

   NOTE: Replace `~/adam` with the actual installation directory.

2. Add the `adam` command to your path by modifying the `~/.bashrc` file:

   ```
   echo -e '\n# Anaconda Adam\nexport PATH=~/adam/bin:$PATH' >> ~/.bashrc
   ```

3. To activate the changes, source your `~/.bashrc` file:

   ```
   source ~/.bashrc
   ```

Verifying the installation

To verify that the installation succeeded and to create configuration files in `~/.continuum/adam`, run the `adam` command, which outputs the Adam help text:

```
$ adam
Usage: adam [OPTIONS] COMMAND [ARGS]...

Options:
  --version                    Show the version and exit.
  -l, --log-level [info|debug|error]
                               Logging level [default: debug]
  -h, --help                   Show this message and exit.

Commands:
  cmd                   Execute a command on the nodes
  describe              Describe an Adam cluster or profile
  ec2                   EC2 options
  enterprise-notebooks  Anaconda Enterprise Notebooks options
  info                  Display Adam version, system, and
                          license information
  jupyter               Jupyter options
  kubernetes            Kubernetes options
  list                  List Adam clusters and profiles
  platform              Anaconda Platform options
  provision             Provision options
  remove                Remove a cluster
  repository            Anaconda Repository options
  salt                  Execute a Salt module
  scale                 Anaconda Scale options
  ssh                   SSH to one of the nodes (0-based
                          index)
  up                    Create a cluster from a profile
```

Installing your Anaconda Enterprise license file

Copy your Anaconda Enterprise license file to any of the following directories:

- `~/adam/licenses/`.

- `~/.continuum/`.
- Any other directory shown by the command `~/adam/bin/conda info --license`.

Adam searches all of these directories for license files named `license*.txt`.

EXAMPLE: `~/adam/licenses/license_bundle_20170428044737.txt`.

Defining a cluster profile

1. Switch to root user and create a file with this name and path `~/.continuum/adam/profile.d/profile.yaml` and the following contents:

```
name: profile
provider: bare

bare:
  username: centos
  \# password: password  \# Optional
  port: 22
  keypair: ~/.ssh/my-private-key
  nodes:
  - host: node1.company.com
  - host: node2.company.com
  - host: node3.company.com
  - host: node4.company.com

security:
  flush_iptables: false
  selinux_context: false
  selinux_permissive: false
```

2. Replace `centos` with an actual username. Verify that the username has sudo privileges.
3. Replace `22` with the appropriate *SSH port number*.
4. Replace `~/.ssh/my-private-key` with the location of the private SSH key (key pair) to be used. Verify that you have SSH access to all of the cluster nodes.
5. Edit the IP addresses or hostnames of the nodes in the cluster. Each host must be able to connect to the other hosts using the specified FQDN or IP address.

   TIP: We recommend using the nodes' FQDN that matches the hostname on each machine.

   NOTE: The first node in the list of nodes is the head node. The remaining nodes are compute nodes.

By default, the `anaconda` user/group is created across the cluster and owns the directory and files located in the root installation directory. If desired, you can *customize the cluster profile* to change these and other settings.

Creating a cluster

To create a cluster, run:

```
adam up -n [cluster-name] [profile-name]
```

EXAMPLE: To create a cluster named "cluster" from a cluster profile named "profile":

```
adam up -n cluster profile
```

Installing platform components

After creating or *provisioning a cluster*, you can install platform components by running:

```
adam [platform-component] -n [cluster-name] install
```

EXAMPLE: To install Repository, Enterprise Notebooks, and Scale on a cluster named "cluster":

```
$ adam repository -n cluster install
$ adam enterprise-notebooks -n cluster install
$ adam scale -n cluster cluster install
$ adam scale -n cluster dask install
```

Opening a platform component

To open a platform component's user interface in your browser, run:

```
$ adam [platform-component] -n [cluster-name] open
```

EXAMPLE: To open Repository and Enterprise Notebooks on a cluster named "cluster":

```
$ adam repository -n cluster open
$ adam enterprise-notebooks -n cluster open
```

For more information see *Working with platform components*.

Installing into an air gapped environment

Currently, Adam can install the remote conda management functionality in an *air gapped* environment using artifacts from Repository. In future versions of Adam, air gapped installations of all Anaconda platform components will be fully supported.

1. *Install Repository*.
2. Configure Repository for the Adam installation on the nodes:
   (a) Mirror the `anaconda` and `anaconda-adam` channels onto your local Repository.
   (b) Copy the latest versions of the Anaconda and Miniconda installers to the `/static/extras/` folder on the Repository web server.
3. Switch to root user and create a cluster profile at `~/.continuum/adam/profile.d/profile.yaml` with the following contents:

```
name: profile
provider: bare

bare:
  username: centos
  # password: password  # Optional
  port: 22
  keypair: ~/.ssh/my-private-key
  nodes:
  - host: node1.company.com
  - host: node2.company.com
```

(continues on next page)

(continued from previous page)

```
  - host: node3.company.com
  - host: node4.company.com

plugins:
  conda:
    anaconda_hash: md5=d72add23bc937ccdfc7de4f47deff843
    anaconda_url: http://local.anaconda.repository.address:8080/static/extras/
↪Anaconda2-4.4.0-Linux-x86_64.sh
    channel_alias: http://local.anaconda.repository.address:8080/conda/
    channels:
    - http://local.anaconda.repository.address:8080/conda/anaconda
    - http://local.anaconda.repository.address:8080/conda/anaconda-adam
    enabled: true
    miniconda_hash: md5=7097150146dd3b83c805223663ebffcc
    miniconda_url: http://local.anaconda.repository.address:8080/static/extras/
↪Miniconda2-4.3.21-Linux-x86_64.sh
    rootdir: /opt/continuum
    ssl_verify: False

security:
  flush_iptables: false
  selinux_context: false
  selinux_permissive: false
```

4. Replace `centos` with an actual username. Verify that the username has sudo privileges.
5. Replace `22` with the appropriate *SSH port number*.
6. Replace `~/.ssh/my-private-key` with the location of the private SSH key (`keypair`) to be used. Verify that you have SSH access to all of the cluster nodes.
7. Edit the port and IP addresses or hostnames of the nodes in the cluster.

   NOTE: The first node in the list of nodes is the head node. The remaining nodes are compute nodes.
8. Replace the `anaconda_hash` value with the hash of your Anaconda installer.
9. Replace the `miniconda_hash` value with the hash of your Miniconda installer.
10. Replace `local.anaconda.repository.address` with the address of your Repository.

Installing into a cloud environment

Adam can be used to create cloud-based instances and install Anaconda platform components on the nodes.

Currently, Adam fully supports Amazon Elastic Compute Cloud (EC2).

To use Adam with instances that already exist on Amazon EC2, or with other cloud providers, follow the *primary installation instructions*.

Refer to the *Installation requirements* page for more information about system requirements, including networking and security requirements.

Downloading the Adam installer

To download the installer, run one of the commands below on the host machine where you want to manage the Anaconda platform:

```
wget https://anaconda.org/anaconda-adam/adam-installer/4.4.0/download/adam-installer-
↪4.4.0-Linux-x86_64.sh
```

Or:

```
curl -LO https://anaconda.org/anaconda-adam/adam-installer/4.4.0/download/adam-
↪installer-4.4.0-Linux-x86_64.sh
```

Installing Adam

1. Run the installer, specifying the installation directory:

   ```
   bash adam-installer-4.4.0-Linux-x86_64.sh -b -p ~/adam
   ```

   NOTE: Replace `~/adam` with the actual installation directory.

2. Add the `adam` command to your path by modifying the `~/.bashrc` file:

   ```
   echo -e '\n# Anaconda Adam\nexport PATH=~/adam/bin:$PATH' >> ~/.bashrc
   ```

3. To activate the changes, source your `~/.bashrc` file:

   ```
   source ~/.bashrc
   ```

Verifying the installation

To verify that the installation succeeded and to create configuration files in `~/.continuum/adam`, run the `adam` command, which outputs the Adam help text:

```
$ adam
Usage: adam [OPTIONS] COMMAND [ARGS]...

Options:
  --version                     Show the version and exit.
  -l, --log-level [info|debug|error]
                                Logging level [default: debug]
  -h, --help                    Show this message and exit.

```

(continues on next page)

(continued from previous page)

```
Commands:
  cmd                     Execute a command on the nodes
  describe                Describe an Adam cluster or profile
  ec2                     EC2 options
  enterprise-notebooks    Anaconda Enterprise Notebooks options
  info                    Display Adam version, system, and
                            license information
  jupyter                 Jupyter options
  kubernetes              Kubernetes options
  list                    List Adam clusters and profiles
  platform                Anaconda Platform options
  provision               Provision options
  remove                  Remove a cluster
  repository              Anaconda Repository options
  salt                    Execute a Salt module
  scale                   Anaconda Scale options
  ssh                     SSH to one of the nodes (0-based
                            index)
  up                      Create a cluster from a profile
```

Installing your Anaconda Enterprise license file

Copy your Anaconda Enterprise license file to any of the following directories:

- `~/adam/licenses/`.
- `~/.continuum/`.
- Any other directory shown by the command `~/adam/bin/conda info --license`.

Adam searches all of these directories for license files named `license*.txt`.

EXAMPLE: `~/adam/licenses/license_bundle_20170428044737.txt`.

Creating an EC2 cluster

1. Create a file named `~/.aws/credentials` that contains your AWS credentials:

   ```
   [default]
   aws_access_key_id = your-access-key
   aws_secret_access_key = your-secret-key
   region = us-east-1
   ```

   NOTE: Replace `your-access-key`, `your-secret-key`, and `us-east-1` with your actual AWS access key ID, your AWS secret access key, and your default region. Region is optional. For more information about configuring your AWS credentials, see the Boto documentation.

2. Create a cluster on EC2 by running:

   ```
   adam ec2 -n mycluster up --keyname my_keypair --keypair ~/.ssh/my_keypair.pem
   ```

   NOTE: Replace “mycluster” with the name you wish to give the new cluster. Replace “my_keypair” with the keyname on the EC2 console. Replace `~/.ssh/my_keypair.pem` with the path to the keypair that matches the keyname.

   `keyname` and `keypair` are required.

OPTIONAL: You can also specify the AMI, number of nodes, instance types and more. To see the available options, run `adam ec2 up --help`:

```
$ adam ec2 -n cluster up --help
Usage: adam ec2 up [OPTIONS]

Options:
  --keyname TEXT                Keyname on EC2 console
                                  [required]
  --keypair PATH                Path to the keypair that
                                  matches the keyname
                                  [required]
  --vpc-id TEXT                 EC2 VPC ID
  --subnet-id TEXT              EC2 Subnet ID on the VPC
  --region-name TEXT            AWS region  [default:
                                  us-east-1]
  --ami TEXT                    EC2 AMI  [default:
                                  ami-d05e75b8]
  --username TEXT               User to SSH to the AMI
                                  [default: ubuntu]
  --type TEXT                   EC2 Instance Type
                                  [default: m4.xlarge]
  --count INTEGER               Number of nodes
                                  [default: 4]
  --security-group TEXT         Security Group Name
                                  [default: adam-default]
  --volume-type TEXT            Root volume type
                                  [default: gp2]
  --volume-size INTEGER         Root volume size (GB)
                                  [default: 500]
  -t, --tag TEXT                Extra tags to add to the
                                  instances
  --check-ami / --no-check-ami  Whether or not to check
                                  the AMI  [default: True]
  --provision / --no-provision  Provision salt on the
                                  nodes  [default: True]
  -y, --yes                     Answers yes to questions
  -h, --help                    Show this message and exit.
```

Installing platform components

After creating or *provisioning a cluster*, you can install platform components by running:

```
adam [platform-component] -n [cluster-name] install
```

EXAMPLE: To install Repository, Enterprise Notebooks, and Scale on a cluster named “cluster”:

```
$ adam repository -n cluster install
$ adam enterprise-notebooks -n cluster install
$ adam scale -n cluster cluster install
$ adam scale -n cluster dask install
```

Opening a platform component

To open a platform component’s user interface in your browser, run:

```
adam [platform-component] -n [cluster-name] open
```

EXAMPLE: To open Repository and Enterprise Notebooks on a cluster named “cluster”:

```
$ adam repository -n cluster open
$ adam enterprise-notebooks -n cluster open
```

For more information see *Working with platform components*.

Removing an EC2 cluster

To remove an EC2 cluster and terminate the corresponding instances:

1. Run:

   ```
   adam ec2 -n [cluster-name] destroy
   ```

2. When prompted, type `y`.

EXAMPLE: To remove the cluster named “cluster”:

```
$ adam ec2 -n cluster destroy
Are you sure you want to destroy the cluster cluster?
[y/N]: y
```

The installation process described in the pages above is appropriate for most environments and use cases. However, if you want to customize Anaconda platform settings related to security, network, installation paths, or other settings, please see *Customizing a cluster profile*.

User guide

Concepts

Anaconda platform components run on clusters, which are groups of machines listed in a cluster profile.

Cluster profile

A cluster profile defines the cluster’s name, its machine nodes and other options such as security and network settings.

One are more cluster profiles are defined during Adam *installation*, just before creating the cluster.

Cluster profiles are located in the `~/.continuum/adam/profile.d/` directory.

Cluster definition

A cluster definition identifies information about a running cluster, including the cluster name, nodes in the cluster and other platform component options. Cluster definitions are located in the `~/.continuum/adam/cluster.d/` directory.

Air gap installation

Installation in an environment without connectivity to the internet for security reasons.

Getting started

Running the `adam` command shows the options and subcommands that you can use:

```
$ adam
Usage: adam [OPTIONS] COMMAND [ARGS]...

Options:
  --version                     Show the version and exit.
  -l, --log-level [info|debug|error]
                                Logging level  [default: debug]
  -h, --help                    Show this message and exit.

Commands:
  cmd                     Execute a command on the nodes
  describe                Describe an Adam cluster or profile
  ec2                     EC2 options
  enterprise-notebooks    Anaconda Enterprise Notebooks options
  info                    Display Adam version, system, and
                            license information
  list                    List Adam clusters and profiles
  provision               Provision options
  remove                  Remove a cluster
  repository              Anaconda Repository options
  salt                    Execute a Salt module
  scale                   Anaconda Scale options
  ssh                     SSH to one of the nodes
                            (0-based index)
  up                      Create a cluster from a profile
```

To see more information about a subcommand, add the `-h` or `--help` option after the subcommand.

EXAMPLE: `adam up --help`.

For details on using the various subcommands, see *Tasks*.

For information on installing Adam, creating a cluster, and installing and managing platform components, see the *installation guide*.

Tasks

Working with cluster profiles

After installing Anaconda Adam and before creating clusters, you must define one or more cluster profiles.

NOTE: An *air gapped machine* needs additional setup and a special cluster profile.

Customizing a cluster profile

Although the default cluster profile described in the *installation guide* is appropriate for most environments and use cases, you can customize profile settings as needed.

Cluster profiles are used to define information about a cluster before it is created, including the cluster name, nodes in the cluster, and other platform component options. Cluster files are located in the `~/.continuum/adam/profile.d/` directory.

Cluster definitions are used to identify information about a running cluster, including the cluster name, nodes in the cluster, and other platform component options. Cluster definitions are located in the `~/.continuum/adam/cluster.d/` directory.

Sample profile showing all settings

A cluster profile located at `~/.continuum/adam/profile.d/cluster.yaml` with all configurable settings is shown below:

```
name: cluster
provider: bare

bare:
  username: centos
  # password: anaconda  # Optional
  port: 22
  keypair: ~/.ssh/my-private-key
  nodes:
  - host: node1.company.com
  - host: node2.company.com
  - host: node3.company.com
  - host: node4.company.com

security:
  flush_iptables: false
  selinux_context: false
  selinux_permissive: false

network:
  http_proxy: http://server:port/
  # https_proxy: http://server:port/

system:
  tmp_dir: /tmp

plugins:
  conda:
    anaconda_hash: md5=d72add23bc937ccdfc7de4f47deff843
    anaconda_url: http://repo.continuum.io/archive/Anaconda2-4.4.0-Linux-x86_64.sh
```

(continues on next page)

(continued from previous page)

```
    channel_alias: https://conda.anaconda.org/
    channels:
    - defaults
    - anaconda-adam
    conda_canary: false
    enabled: true
    miniconda_hash: md5=7097150146dd3b83c805223663ebffcc
    miniconda_url: http://repo.continuum.io/miniconda/Miniconda2-4.3.21-Linux-x86_64.
↪sh
    rootdir: /opt/continuum
    ssl_verify: False

  dask:
    bokeh_port: 8787
    bokeh_whitelist: '''*'''
    enabled: false
    host: 0.0.0.0
    http_port: 9786
    nprocs: 1
    port: 8786

  enterprise-notebooks:
    admin_email: admin@yourdomain.com
    admin_password: anaconda
    admin_user: wakari
    directory: /opt/wakari
    elasticsearch_fn: elasticsearch-1.7.2.noarch.rpm
    enabled: false
    enterprise_notebooks_version: 4.0.0
    gateway_port: 8089
    java_fn: jre-8u65-linux-x64.rpm
    mongodb_fn: mongodb-org-2.6.8-1.x86_64.rpm
    mongodb_mongos_fn: mongodb-org-mongos-2.6.8-1.x86_64.rpm
    mongodb_server_fn: mongodb-org-server-2.6.8-1.x86_64.rpm
    mongodb_shell_fn: mongodb-org-shell-2.6.8-1.x86_64.rpm
    mongodb_tools_fn: mongodb-org-tools-2.6.8-1.x86_64.rpm
    mongodb_version: 2.6.8
    nginx_fn: nginx-1.6.2-1.el6.ngx.x86_64.rpm
    root_download_url: https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.
↪ssl.cf1.rackcdn.com/
    user: wakari

  jupyter:
    dashboards_server:
      enabled: false
      ip: 0.0.0.0
      port: 3000
      user: anaconda
    enabled: false
    kernel_gateway:
      enabled: false
      ip: 0.0.0.0
      port: 7000
      user: anaconda
    notebook:
      directory: ~/notebooks
      enabled: false
```

(continues on next page)

(continued from previous page)

```
      ip: 0.0.0.0
      password: 'anaconda'
      port: 8888
      user: anaconda

  repository:
    binstar_server_version: 2.21.0
    channel: main
    email: youremail@anaconda.com
    enabled: false
    mongodb_fn: mongodb-org-2.6.8-1.x86_64.rpm
    mongodb_mongos_fn: mongodb-org-mongos-2.6.8-1.x86_64.rpm
    mongodb_server_fn: mongodb-org-server-2.6.8-1.x86_64.rpm
    mongodb_shell_fn: mongodb-org-shell-2.6.8-1.x86_64.rpm
    mongodb_tools_fn: mongodb-org-tools-2.6.8-1.x86_64.rpm
    mongodb_version: 2.6.9
    password: anaconda
    port: 8080
    root_download_url: https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.
↪ ssl.cf1.rackcdn.com/
    server_user: anaconda-server
    superuser: superuser
    token: qu-a49e2e69-1047-4eab-a879-a2ee9c198381

  salt_settings:
    acl:
    - anaconda
    job_pub_port: 14505
    minion_pub_port: 14510
    minion_pull_port: 14511
    minion_ret_port: 14506
    rest_port: 18000
    salt_groupname: anaconda
    salt_password: anaconda
    salt_username: anaconda
    service_scripts: false
```

Cluster name

The `name` setting specifies the name of the cluster and is used by the `-n` option in Anaconda Adam commands.

EXAMPLE: To create a new cluster named “cluster” from a profile named “profile”:

```
$ adam up -n cluster profile
```

SSH authentication

To connect to remote machines via SSH, Adam requires a valid username, port–default is 22–and an authentication method. Adam supports the following SSH authentication methods:

- Password.
- Key pair–full path to your private key.

- Encrypted key pair–full path to your public key and ssh-agent. The private key must first be added to a running ssh-agent. Use the `agent_pubkey` setting to specify the full path to the associated public key.

For more information about system and account requirements, see *Account access and security*.

EXAMPLE: To use SSH with a username and password:

```
bare:
  username: centos
  password: anaconda
```

EXAMPLE: To use SSH with a key pair:

```
bare:
  username: centos
  keypair: /full/path/to/key.rsa
```

EXAMPLE: To use SSH with an encrypted key pair:

```
bare:
  username: centos
  agent_pubkey: /full/path/to/key.pub
```

Node specification

The `nodes` setting defines the nodes that exist within the cluster. By default, the first node in the list of nodes is defined as the head node and the remaining nodes are defined as compute nodes.

Each host must be able to connect to the other hosts via the specified FQDN or IP address.

TIP: We recommended using the node's FQDN that matches the hostname on each machine:

EXAMPLE:

```
bare:
  nodes:
  - host: node1.company.com
  - host: node2.company.com
  - host: node3.company.com
  - host: node4.company.com
```

Security settings

The `security` settings let you configure IPTables and SELinux.

To flush iptables, reset `flush_iptables` to `true`. Default value: `false`.

```
security:
  flush_iptables: false
```

To set SELinux to permissive, reset `selinux_permissive` to `true`. Default value: `false`.

```
security:
  selinux_permissive: false
```

To set SELinux contexts, reset `selinux_context` to `true`. Default value: `false`.

```
security:
  selinux_context: false
```

Network settings–proxy configuration

The `network` settings let you install Adam to nodes that access the internet via an HTTP or HTTPS proxy.

To specify an HTTP proxy:

```
network:
  http_proxy: http://server:port/
```

NOTE: Replace `server:port` with the actual server and port.

To specify an HTTPS proxy:

```
network:
  https_proxy: http://server:port/
```

NOTE: Replace `server:port` with the actual server and port.

Conda settings

These are the default settings for configuring conda:

```
conda:
    anaconda_hash: md5=d72add23bc937ccdfc7de4f47deff843
    anaconda_url: http://repo.continuum.io/archive/Anaconda2-4.4.0-Linux-x86_64.sh
    channel_alias: https://conda.anaconda.org/
    channels:
    - defaults
    - anaconda-adam
    enabled: true
    miniconda_hash: md5=7097150146dd3b83c805223663ebffcc
    miniconda_url: http://repo.continuum.io/miniconda/Miniconda2-4.3.21-Linux-x86_64.
↪sh
    rootdir: /opt/continuum
    ssl_verify: False
```

You can set ssl_verify to:

- `False`–no SSL verification–default.
- `True`–SSL verification is used and conda verifies certificates for SSL connections.
- `[cert path]`–the string path to a certificate to be used to verify SSL connections.

Salt settings

Salt is the configuration management system used by Adam. The `salt_settings` let you configure the network options, access control list and API user credentials used by Salt.

Use the settings below to specify the ports used by the Salt master, minions and REST API:

```
plugins:
  salt_settings:
    job_pub_port: 14505
    minion_pub_port: 14510
    minion_pull_port: 14511
    minion_ret_port: 14506
    rest_port: 18000
```

Use the `acl` setting to specify an access control list that defines non-root system users who can execute Salt commands:

```
plugins:
  salt_settings:
    acl:
    - anaconda
```

Use the settings below to specify the username, password and group that gets created across the cluster and owns the directory and files located in the root installation directory–default: `/opt/continuum`:

```
plugins:
  salt_settings:
    salt_username: anaconda
    salt_password: anaconda
    salt_groupname: anaconda
```

If you do not want to store a clear text password in the cluster profile or cluster definition file, set `salt_password` to an empty string:

```
plugins:
  salt_settings:
    salt_username: anaconda
    salt_password: ''
    salt_groupname: anaconda
```

In this case, Adam prompts for a password in the CLI and only stores the password in memory for the duration of the command or job.

System settings

The `tmp_dir` setting lets you specify the download directory where temporary installers are downloaded–default: `/tmp`:

```
system:
  tmp_dir: /tmp
```

Viewing cluster profile information

To view a list of cluster profiles, run `adam list profiles`:

```
$ adam list profiles
profile
second-profile
third-profile
```

To view more detailed information about a profile, run:

```
$ adam describe -p [profile-name]
```

You may also use the `--verbose` flag for increased detail.

EXAMPLE: To view verbose information about the profile named "profile":

```
$ adam describe -p profile --verbose
bare:
  keypair: ~/.ssh/my_keypair.pem
  nodes:
  - host: 192.168.1.1
  - host: 192.168.1.2
  - host: 192.168.1.3
  - host: 192.168.1.4
  port: 22
  username: centos
name: cluster
provider: bare
security:
  flush_iptables: false
  selinux_context: false
  selinux_permissive: false
```

Working with clusters

Defining one or more cluster profiles and creating clusters are described in the *installation guide*.

Provisioning a cluster

To provision or reprovision an existing cluster, run:

```
$ adam provision -n [cluster-name]
```

EXAMPLE: To provision the cluster named "cluster":

```
$ adam provision -n cluster
```

Customizing the provision command–advanced

By default, the `adam provision` command:

- Checks SSH connectivity.
- Creates a default user.
- Installs Salt.
- Syncs formulas.
- Provisions enabled plugins and security settings that are defined in the cluster's profile.

To run only a portion of this provisioning process, use `adam provision` options and subcommands to control what happens during provisioning.

Run `adam provision --help` to see the available options and subcommands:

```
$ adam provision --help
Usage: adam provision [OPTIONS] COMMAND [ARGS]...

  Execute all the provisioning steps for a cluster

  See subcommands help for more info. Usage: adam provision -n

Options:
  -n, --cluster TEXT    Cluster name  [required]
  --flush-iptables      Flush IPTables  [default: False]
  --selinux-permissive  Set SELinux to permissive  [default:
                          False]
  --selinux-context     Set SELinux contexts  [default: False]
  --plugins             Install enabled plugins on provision
                          [default: False]
  -h, --help            Show this message and exit.

Commands:
  anaconda-user       Create the default anaconda user in all
                        the nodes
  check-ssh           Check SSH connection to the nodes
  flush-iptables      Flush IPTables
  plugins             Provision enabled plugins
  roles               Set roles on the cluster nodes
  salt                Provision Salt
  security-settings   Execute the security settings
  selinux-context     Set SELinux contexts
  selinux-permissive  Set SELinux to permissive
  sync                Sync formulas and settings
```

EXAMPLE: To check SSH connectivity to all of the nodes in an existing cluster:

```
$ adam provision -n cluster check-ssh
Checking SSH connection to nodes
2016-10-03 22:10:49,327 - adam.models - DEBUG - Checking ssh
connection for 54.88.20.164
2016-10-03 22:10:50,905 - adam.models - DEBUG - SSH
connection to 54.88.20.164: OK
2016-10-03 22:10:51,008 - adam.models - DEBUG - Checking ssh
connection for 52.90.168.130
2016-10-03 22:10:51,996 - adam.models - DEBUG - SSH
connection to 52.90.168.130: OK
+---------------+---------------+
| Node IP       | SSH Available |
+---------------+---------------+
| 54.88.20.164  | True          |
| 52.90.168.130 | True          |
+---------------+---------------+
| All nodes     | True          |
+---------------+---------------+
```

Viewing cluster information

To view a list of running clusters, run `adam list`:

```
$ adam list
cluster
second-cluster
third-cluster
```

To view detailed information about a cluster, run:

```
$ adam describe -n [cluster-name]
```

EXAMPLE: To view information about the cluster named “cluster”:

```
$ adam describe -n cluster
conda:
  anaconda_hash: md5=7097150146dd3b83c805223663ebffcc
  anaconda_url: https://repo.continuum.io/miniconda/Miniconda2-4.3.21-Linux-x86_64.sh
↪#adam-ci
  enabled: true
  rootdir: /opt/continuum
name: cluster
nodes:
- host: ci_head_1
  keypair: ~/.ssh/my_private_key
  port: 22
  username: root
- host: ci_compute_1
  keypair: ~/.ssh/my_private_key
  port: 22
  username: root
```

Adding nodes to an existing cluster

1. In the existing cluster profile, add `host` lines to specify the additional node(s).

   EXAMPLE: You have this existing cluster profile, `~/.continuum/adam/profile.d/profile.yaml`:

   ```
   name: profile
   provider: bare

   bare:
     username: centos
     # password: password  # Optional
     port: 22
     keypair: ~/.ssh/my-private-key
     nodes:
     - host: node1.company.com
     - host: node2.company.com
     - host: node3.company.com
     - host: node4.company.com
   ```

   You want to add the following nodes:

   - node10.company.com
   - node11.company.com
   - node12.company.com

   Assuming these nodes use the same authentication mode as the existing nodes, add a `host` line for each node:

```
name: profile
provider: bare

bare:
  username: centos
  # password: password  # Optional
  port: 22
  keypair: ~/.ssh/my-private-key
  nodes:
  - host: node1.company.com
  - host: node2.company.com
  - host: node3.company.com
  - host: node4.company.com
  - host: node10.company.com
  - host: node11.company.com
  - host: node12.company.com
```

2. Run the `adam up` command to provision the cluster.

   EXAMPLE: To use the profile named "profile" to provision the cluster named "cluster", run:

   ```
   $ adam up -n cluster profile
   ```

   You are prompted to overwrite the cluster definition file located at `~/.continuum/adam/cluster.d/cluster.yaml` to include the newly defined nodes.

   During the provisioning process, Adam will install on the new nodes and will not reinstall on the previously defined nodes.

Managing Anaconda across a cluster

Using Anaconda Scale, you can create and manage conda packages, including Python and R, in multiple conda environments across your nodes.

For more information about installing and managing Anaconda across a cluster, see *Anaconda Scale*.

Using SSH to connect to nodes

To SSH to the head node in a cluster, run:

```
$ adam ssh -n [cluster-name]
```

EXAMPLE: To SSH to the head node in a cluster named "cluster":

```
$ adam ssh -n cluster
Last login: Wed Jun 29 03:08:49 2016 from 172.18.0.1
[root@192.168.1.1 ~]#
```

To connect to a specific node, include the node number after the cluster name. Node 0 is the head node, node 1 is the first compute node, and so on.

EXAMPLE: To SSH to the second compute node in a cluster named "cluster":

```
$ adam ssh -n cluster 2
Last login: Wed Jun 29 03:10:14 2016 from 172.18.0.1
[root@192.168.1.3 ~]#
```

Running remote commands

To execute remote commands on a cluster, run:

```
$ adam cmd -n [cluster-name] -t [target-nodes] [command]
```

EXAMPLE: To run the `date` command on all nodes in a cluster named “cluster”:

```
$ adam cmd -n cluster date
192.168.1.1: Wed Jun 29 03:08:10 UTC 2016
192.168.1.2: Wed Jun 29 03:08:10 UTC 2016
192.168.1.3: Wed Jun 29 03:08:10 UTC 2016
192.168.1.4: Wed Jun 29 03:08:10 UTC 2016
```

Excecuting Salt modules

To execute Salt modules, run:

```
$ adam salt -n [cluster-name] [nodes] [module-name]
```

EXAMPLE: To execute the “test.ping” Salt module on all nodes in a cluster named “cluster”:

```
$ adam salt -n cluster '*' test.ping
192.168.1.1: true
192.168.1.2: true
192.168.1.3: true
192.168.1.4: true
```

Removing a cluster

To remove the cluster definition file for an existing bare-metal cluster:

1. Run:

   ```
   $ adam remove -n [cluster-name]
   ```

2. When prompted, type `y`.

EXAMPLE: To remove the cluster named “cluster”:

```
$ adam remove -n cluster
Are you sure you want to remove the cluster cluster? [y/N]: y
```

To remove an EC2 cluster, see the removal instructions at the end of the *cloud installation guide*.

Working with platform components

Overview

Using Anaconda Adam, you can install and manage the Anaconda platform components listed below. An Adam command for managing an Anaconda component has the format:

```
$ adam [platform-component] [command]
```

Platform component	Adam Command Prefix
Anaconda Enterprise Notebooks	`adam enterprise-notebooks`
Anaconda Repository	`adam repository`
Anaconda Scale	`adam scale`

You can also use Adam to install and manage the Anaconda platform on *Amazon EC2 cloud-based nodes* and *air gapped machines*.

Default usernames and passwords

The default administrator usernames and passwords for each platform component are:

Platform component	Username	Password
Anaconda Enterprise Notebooks	wakari	anaconda
Anaconda Repository	superuser	anaconda
Jupyter Notebook	-	anaconda

Platform component documentation

For more information on a platform component, see:

- *Anaconda Enterprise Notebooks*.
- *Anaconda Repository*.
- *Anaconda Scale*.

Viewing a platform component's status

To view the status of a platform component, run:

```
$ adam [platform-component] -n [cluster-name] status
```

EXAMPLE: To view the status of Repository on a cluster named "cluster":

```
$ adam repository -n cluster status
```

Stopping services for a platform component

To stop services related to a platform component, run:

```
$ adam [platform-component] -n [cluster-name] stop
```

EXAMPLE: To stop Repository services on a cluster named “cluster”:

```
$ adam repository -n cluster stop
```

Restarting services for a platform component

To restart services related to a platform component, run:

```
$ adam [platform-component] -n [cluster-name] restart
```

EXAMPLE: To restart Repository services on a cluster named “cluster”:

```
$ adam repository -n cluster restart
```

Viewing your Adam version and license

To view version and license information, run `adam info`:

```
$ adam info
Adam version: 4.1.0
Plugins:
  adam_enterprise_notebooks: 4.1.0
  adam_repository: 4.1.0
  adam_notebook: 4.1.0
  adam_dask: 4.1.0

Platform: Darwin-15.6.0-x86_64
Darwin Kernel Version 15.6.0: Thu Jun 23 18:25:34 PDT 2016;
root:xnu-3248.60.10~1/RELEASE_X86_64
Python version: 3.5.1 |Continuum Analytics, Inc.| (default,
Dec  7 2015, 11:24:55)
[GCC 4.2.1 (Apple Inc. build 5577)]
Processor: i386
Byte-ordering: little

License information:
Number of nodes currently in use: 14
Number of licensed nodes: 16
Number of managed clusters: 2

Valid platform component licenses:
1: Anaconda Repository Enterprise. End date: 2018-05-24
2: Anaconda Cluster. End date: 2018-05-24
3: Wakari Enterprise. End date: 2018-05-24
4: MKL Optimizations. End date: 2018-05-24
5: IOPro. End date: 2018-05-24
6: Accelerate. End date: 2018-05-24
```

Updating Adam

To update Anaconda Adam to a newer version:

1. Delete the existing Adam installation directory. By default, this is `~/adam`.

   NOTE: *Cluster profile* and *cluster definition* files are stored in `~/.continuum/adam`, which is preserved during the update process.

2. *Re-install Adam*.

Troubleshooting

Errors when creating or provisioning a cluster

You get errors when you run `adam up` to create a cluster or `adam provision` to provision a cluster.

Cause

These errors may be caused by an invalid SSH private key or incorrect settings in a cluster profile.

Solution

Verify the following:

- The contents of your SSH private key are correct and, on Mac/Linux, set to `600` permissions.
- The user name in the cluster profile, such as `username:  centos`, is defined correctly.
- The cluster profile settings are defined correctly in `~/.continuum/adam/profile.d/<profile_name>.yaml`.

Error: unsupported or unrecognized field

When running Adam commands, you get an error similar to the following:

```
2016-12-05 11:09:50,120 - adam.config - ERROR - {'notebook':
'Rogue field'}
Error: One or more fields is not supported or recognized by
this version of Anaconda Adam. Correct or remove the
unsupported field in the cluster YAML profile or cluster
definition YAML file, and try the operation again. See the
above errors for more details: {'notebook': 'Rogue field'}
```

Cause

There is an unsupported or unrecognized field in a cluster profile or definition file. If you have updated to a newer version of Adam, some fields in the cluster profile or cluster definition file may have been renamed or removed between versions.

Solution

1. In the cluster profile file in the `~/.continuum/adam/profile.d` directory or the cluster definition file in the `~/.continuum/adam/cluster.d` directory, remove or rename the field shown in the error.
2. Retry the command.

FAQs

Does Anaconda Adam work with a cluster that already has a managed Spark/Hadoop stack?

Yes, you can install Adam alongside existing Hadoop distributions such as Cloudera CDH or Hortonworks HDP. You can use Adam to manage Python and R conda packages and environments across a cluster.

Which cloud providers does Adam support?

Currently, Adam offers full support for Amazon Elastic Compute Cloud (EC2).

Can I use Adam with a different cloud provider?

Yes, you can manually create instances on another cloud provider and then provision the nodes. For more information, see *Tasks*.

Which network ports need to be accessible from the client machine and cluster nodes?

For communication between the cluster nodes, ports 22, 4505 and 4506 are used to provision the cluster via SSH and Salt.

Help and support

Priority support is included with the purchase of an Anaconda subscription. Visit the support section of our website for documentation and contact information for support.

Training and consulting

Training and consulting are available for the Anaconda platform and all Anaconda platform components. For more information, contact sales@anaconda.com.

Release notes

Below is a summary of platform component updates, features, bug fixes, and backend improvements for each Anaconda Adam release.

Anaconda Adam 4.4.0

Released: June 20, 2017

Platform Component	Version	Plugin Version
Anaconda/Miniconda	4.4.0/4.3.21	-
Anaconda Enterprise Notebooks	4.1.2	4.2.14
Anaconda Repository	2.30.3	4.4.0
Anaconda Scale (Dask/Distributed)	0.15.0/1.17.1	4.4.0
Anaconda Scale (Jupyter Notebook)	5.0.0	4.2.14

Download Anaconda Adam:

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.4.0/download/adam-installer-4.4.0-Linux-x86_64.sh
- MD5: 64cc02dcaaf7829144cd82db786ab775
- SHA256: 663e2f93177830badadf5dea6647cb0e3d5376770999b85bb02d535d13cfac6d

Platform Component Updates:

- Updated to Anaconda 4.4.0 (conda 4.3.21)
- Updated to Miniconda 4.3.21 (conda 4.3.21)
- Updated to Anaconda Repository 2.30.3
- Updated to Dask 0.15.0 and Distributed 1.17.1

Anaconda Adam 4.2.14

Released: April 28, 2017

Platform Component	Version	Plugin Version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Enterprise Notebooks	4.1.2	4.2.14
Anaconda Repository	2.29.1	4.2.14
Anaconda Scale (Dask/Distributed)	0.14.1/1.16.1	4.2.14
Anaconda Scale (Jupyter Notebook)	5.0.0	4.2.14

Download Anaconda Adam:

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.14/download/adam-installer-4.2.14-Linux-x86_64.sh
- MD5: 3fd72336d12ee635323d3f3541cce1c0
- SHA256: 43275084838a698fee7329651d55b20b3087037cc14bec04e27d6d985d402fbd

Platform Component Updates:

- Updated to Anaconda Enterprise Notebooks 4.1.2
- Updated to Anaconda Repository 2.29.1
- Updated to Dask 0.14.1 and Distributed 1.16.1
- Updated to Jupyter Notebook 5.0.0

Anaconda Adam 4.2.13

Released: March 2, 2017

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Enterprise Notebooks	4.1.1	4.2.11
Anaconda Repository	2.27.2	4.2.13
Anaconda Scale (Dask/Distributed)	0.14.0/1.16.0	4.2.13
Anaconda Scale (Jupyter Notebook)	4.3.1	4.2.11

Platform component updates

- Updated to Anaconda Repository 2.27.2.
- Updated to Dask 0.14.0 and Distributed 1.16.0.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.13/download/adam-installer-4.2.13-Linux-x86_64.sh.
- MD5: a49e1a0c16b00deaaff5d33b8663e8a5.
- SHA256: 4d38ea7fca4e8d872bd64bfa80ecd641cfeaf9aac342d00d3e90b46914268c29.

Anaconda Adam 4.2.12

Released: February 3, 2017

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Enterprise Notebooks	4.1.1	4.2.11
Anaconda Repository	2.26.3	4.2.11
Anaconda Scale (Dask/Distributed)	0.13.0/1.15.0	4.2.11
Anaconda Scale (Jupyter Notebook)	4.3.1	4.2.11

Bug fixes

Fixed Anaconda/Miniconda download links to be compatible with new `repo.continuum.io` CDN.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.12/download/adam-installer-4.2.12-Linux-x86_64.sh.
- MD5: b119e123b563b3c6ad659835afeec9f1.
- SHA256: fd595b2eee304be5949a22e52d5cf2ee7fd41f8dcded6700ecf525746a846e19.

Anaconda Adam 4.2.11

Released: January 16, 2017

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Enterprise Notebooks	4.1.1	4.2.11
Anaconda Repository	2.26.3	4.2.11
Anaconda Scale (Dask/Distributed)	0.13.0/1.15.0	4.2.11
Anaconda Scale (Jupyter Notebook)	4.3.1	4.2.11

Features

Implemented default password in Jupyter Notebooks.

Backend improvements

- Disable automatic updates of conda.
- Updates for compatibility with conda 4.3.x features.

Platform component updates

- Updated to Anaconda Enterprise Notebooks 4.1.1.
- Updated to Anaconda Repository 2.26.3.
- Updated to Dask 0.13.0 and Distributed 1.15.0.
- Updated to Jupyter Notebook 4.3.1.
- Removed Anaconda Accelerate as a standalone platform component (now installed as part of Anaconda Scale).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.11/download/adam-installer-4.2.11-Linux-x86_64.sh.
- MD5: a6e4406c1552ff1e8ee91aff98ed9b3d.
- SHA256: 6a49fa7e6f3d9912bbc3b3986a7b4516a98d2544e6d032242256e6630aea9f28.

Anaconda Adam 4.2.10

Released: December 5, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.7
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Repository	2.25.1	4.2.10
Anaconda Scale (Dask/Distributed)	0.12.0/1.14.3	4.2.10
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.7

Platform component updates

- Updated to Anaconda Repository 2.25.1.
- Updated to Dask 0.12.0 and Distributed 1.14.3.
- Removed Anaconda Mosaic.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.10/download/adam-installer-4.2.10-Linux-x86_64.sh.
- MD5: 4ec7e5aae0a6913ab8f14498ba657b7b.
- SHA256: 15a843ec115f413167bad2ef6fe1b4aa7752e74ad8816a0f911042a5626c239a.

Anaconda Adam 4.2.9

Released: November 29, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.7
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.7
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.7
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.7

Bug fixes

Fixed scripts that were not included in the release bundle (required for `service_scripts` setting).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.9/download/adam-installer-4.2.9-Linux-x86_64.sh.
- MD5: c1a5c61c0ae9e37686f9d3fb3670aa18.
- SHA256: 90cb740f04ec5ac78b65c4574b0d4238eb5c8e913ddaec49c27dcb820665bd00.

Anaconda Adam 4.2.8

Released: November 29, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.7
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.7
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.7
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.7

Bug fixes

Fixed issue with cluster profile settings not being migrated to cluster definition file.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.8/download/adam-installer-4.2.8-Linux-x86_64.sh.
- MD5: f1083e30500b281e8c1d668c02ae8e3c.
- SHA256: 5a4b4391b1743cfc03d7bfa2e5693cc064cbd609f4c4fea7438846f2fa8d27e8.

Anaconda Adam 4.2.7

Released: November 16, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.7
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.7
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.7
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.7

Features

Added configuration settings for `salt_username`, `salt_groupname` and `salt_password`, for a user that is created across the cluster and owns the directory and files located in the root installation directory.

Backend improvements

Added retries for Salt minion/API connection.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.7/download/adam-installer-4.2.7-Linux-x86_64.sh.
- MD5: e1efa2a6652ba8bc9cf974a07dbc1df2.
- SHA256: 71f375ca4e7ab92072af71c970080950292e3b663fd3ca42c681651a154d682d.

Anaconda Adam 4.2.6

Released: November 9, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.6
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.3
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Bug fixes

Fixed installation issue with Anaconda Mosaic related to conda environments.

Backend improvements

Reverted to `requests` for file downloads via Salt HTTP module backend.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.6/download/adam-installer-4.2.6-Linux-x86_64.sh.
- MD5: 035ee07ef6add9d60e23509bd91bebdf.
- SHA256: 3a4ca7ce75a8781fac9d7f38ec768b40c39b52091d39267dd898ec33ef299724.

Anaconda Adam 4.2.5

Released: November 9, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.2.12	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.1
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.3
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Platform component updates

Updated to Miniconda 4.2.12 (conda 4.2.12).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.5/download/adam-installer-4.2.5-Linux-x86_64.sh.
- MD5: 2614d2f8239aa0f3b8d070bd2a1ea83e.
- SHA256: 5355646e8e143cc5929061422427ee6e1f2fa143c5898865ccd0122803b72fa5.

Anaconda Adam 4.2.4

Released: November 8, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.1.11	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.1.0	4.2.4
Anaconda Mosaic	1.3.1	4.2.1
Anaconda Repository	2.23.1	4.2.4
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.3
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Features

- Added support for creating and updating environments using conda env YAML files and listing and removing environments with `adam scale env`.
- Added `tmp_dir` setting to cluster profile to configure temporary download location.
- Added support for removing conda environments using `adam scale conda remove -n my-env --all`.

Backend improvements

- Added support for downloads from CDN by enabling location header redirects when using `curl`.
- Improved handling and catching for errors returned from Salt.

Bug fixes

- Fixed handling of proxy settings (including `.condarc` files) in cluster configuration/profile.
- Fixed handling of security settings in cluster definition when using Amazon EC2 instances.
- Fixed ownership for license directory in Anaconda Scale `rootdir`; default: `/opt/continuum/anaconda/licenses`.

Platform component updates

- Updated to Anaconda Enterprise Notebooks 4.1.0.
- Updated to Anaconda Repository 2.23.1.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.4/download/adam-installer-4.2.4-Linux-x86_64.sh.
- MD5: 67df6905007d892aa6597c06d3fe0693.
- SHA256: 2630798205a215ce0021db50591e8507c20cd36164eda27d74dcf1eb9a4d58fd.

Anaconda Adam 4.2.3

Released: October 24, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.1.11	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.0.0	4.2.1
Anaconda Mosaic	1.3.1	4.2.1
Anaconda Repository	2.22.0	4.2.3
Anaconda Scale (Dask/Distributed)	0.11.1/1.13.3	4.2.3
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Backend improvements

- Updated `ssl_verify` setting to handle various inputs (true, false, or string).
- Switched to `curl` for downloading Anaconda/Miniconda in environments with SSL/proxies.

Platform component updates

- Updated to Anaconda Repository 2.22.0.
- Updated to Dask 0.11.1 and Distributed 1.13.3.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.3/download/adam-installer-4.2.3-Linux-x86_64.sh.
- MD5: a484a933afcebb99bafbc5bde4e63081.
- SHA256: 8beb36f619f9d6177f0815090fa1ba7d41c769e47978d86c0b0c60164d63b996.

Anaconda Adam 4.2.2

Released: October 19, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.1.11	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.0.0	4.2.1
Anaconda Mosaic	1.3.1	4.2.1
Anaconda Repository	2.21.0	4.2.1
Anaconda Scale (Dask/Distributed)	0.11.0/1.12.2	4.2.1
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Bug fixes

Implemented various fixes related to conda configuration file (`.condarc`) and environment variable (`CONDARC`).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.2/download/adam-installer-4.2.2-Linux-x86_64.sh.
- MD5: b0a7cc3061283c0c2b8b86dc8643190c.
- SHA256: f6ac6a6a1a6434b039d1ead09ddb056a09e07598859eb04eab33e24e9ad6d85c.

Anaconda Adam 4.2.1

Released: October 14, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.1.11	-
Anaconda Accelerate	2.3.0	4.2.1
Anaconda Enterprise Notebooks	4.0.0	4.2.1
Anaconda Mosaic	1.3.1	4.2.1
Anaconda Repository	2.21.0	4.2.1
Anaconda Scale (Dask/Distributed)	0.11.0/1.12.2	4.2.1
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.1

Features

- Improved output when checking node status (using Salt `test.ping`).
- Added configuration settings to cluster profile related to Salt API username, password, and group.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.1/download/adam-installer-4.2.1-Linux-x86_64.sh.
- MD5: f8b01e94cf6b114b4d9df0f2074206a2.
- SHA256: 3ce27577757eb576ed7e3863ad35c5ca37755cdcb0de8f634ca6215ee4b6c67a.

Anaconda Adam 4.2.0

Released: October 6, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.2.0/4.1.11	-
Anaconda Accelerate	2.3.0	4.2.0
Anaconda Enterprise Notebooks	4.0.0	4.2.0
Anaconda Mosaic	1.3.1	4.2.0
Anaconda Repository	2.21.0	4.2.0
Anaconda Scale (Dask/Distributed)	0.11.0/1.12.2	4.2.0
Anaconda Scale (Jupyter Notebook)	4.2.3	4.2.0

Features

- Improved error messaging for missing/incompatible cluster profile fields.
- Improved handling of state for installed/enabled platform components.
- Simplified `adam describe` ouput to hide platform components that are not installed.

Backend improvements

- Renamed `salt` to `salt_settings` in cluster profile and added documentation.
- Added conda canary builds to test framework.

Bug fixes

- Fixed user/group ownership for platform component license directories.
- Improved validation of SSH key pairs, passwords, or agent public keys.
- Fixed check for existing Anaconda/Supervisor installations.
- Improved hash check for Miniconda downloads.
- Improved retry functionality for commands executed in parallel across multiple nodes.
- Improved handling of returned states from remote conda commands.
- Print URL when opening/viewing platform components if no browser is available.

Platform component updates

- Updated to Anaconda 4.2.0 (conda 4.2.9).
- Updated to Anaconda Repository 2.21.0.
- Updated to Dask 0.11.0 and Distributed 1.12.2.
- Updated to Jupyter Notebook 4.2.3.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.2.0/download/adam-installer-4.2.0-Linux-x86_64.sh.
- MD5: 4c6e685615d58486911f06c264bc1990.
- SHA256: 0556ee1b7e8bcd379faf6c2089e7fb525d2845187361a2a96d99a281ba207108.

Anaconda Adam 4.1.2

Released: August 24, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.1.11	-
Anaconda Accelerate	2.3.0	4.1.0
Anaconda Enterprise Notebooks	4.0.0	4.1.1
Anaconda Mosaic	1.3.1	4.1.0
Anaconda Repository	2.19.7	4.1.1
Anaconda Scale (Dask/Distributed)	0.10.2/1.11.3	4.1.0
Anaconda Scale (Jupyter Notebook)	4.2.2	4.1.0

Features

Added support for installations via proxy (HTTP and HTTPS).

Bug fixes

Fixed issue with the creation of a custom home directory for the anaconda user.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.1.2/download/adam-installer-4.1.2-Linux-x86_64.sh.
- MD5: 672611f7cdc5902fad40c36549b9e214.
- SHA256: db6b78338e126e017fcc6082add46f5d87a2bb48a21e5fd6191277ecb700031e.

Anaconda Adam 4.1.1

Released: August 18, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.1.11	-
Anaconda Accelerate	2.3.0	4.1.0
Anaconda Enterprise Notebooks	4.0.0	4.1.0
Anaconda Mosaic	1.3.1	4.1.0
Anaconda Repository	2.19.7	4.1.0
Anaconda Scale (Dask/Distributed)	0.10.2/1.11.3	4.1.0
Anaconda Scale (Jupyter Notebook)	4.2.2	4.1.0

Bug fixes

- Fixed issue with Salt API job responses.
- Fixed issue with invoke_shell buffer size.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.1.1/download/adam-installer-4.1.1-Linux-x86_64.sh.
- MD5: 48a6fda5c16842ea2f138fe59723f425.
- SHA256: c4c6f14d3c4ed0d85d7eb8eaa1faa2d5959cecfbad87d0025cfcfa2f521ed7db.

Anaconda Adam 4.1.0

Released: August 17, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.1.11	-
Anaconda Accelerate	2.3.0	4.1.0
Anaconda Enterprise Notebooks	4.0.0	4.1.0
Anaconda Mosaic	1.3.1	4.1.0
Anaconda Repository	2.19.7	4.1.0
Anaconda Scale (Dask/Distributed)	0.10.2/1.11.3	4.1.0
Anaconda Scale (Jupyter Notebook)	4.2.2	4.1.0

Bug fixes

- Fixed issue when attempting to disable SELinux.
- Fixed licensing check when recreating an existing cluster.
- Fixed use of `host` during installation of Anaconda Enterprise Notebooks.

Backend improvements

- Renamed `fqdn` to `host` in cluster profile.
- Changed default AEN Gateway port to 8089 in Anaconda Enterprise Notebooks.
- Changed default system account to `anaconda-server` for Anaconda Repository.
- Added diagnostic SSH checks to initial cluster provision.
- Implemented asynchronous Salt jobs for cluster-wide tasks.
- Added support for DSA and ECDSA SSH keys.
- Added optional `invoke_shell` mode for installations with restricted `sudo` functionalty.
- Updated to Salt 2016.3.1.

Platform component updates

- Updated to Miniconda 4.1.11 (conda 4.1.11).
- Updated to Anaconda Repository 2.19.7.
- Updated to Jupyter Notebook 4.2.2.
- Initial release of Anaconda Scale platform component (replaces Anaconda for cluster management).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.1.0/download/adam-installer-4.1.0-Linux-x86_64.sh.
- MD5: 063b75a7352e711626dd6e4eabef5228.
- SHA256: 6dd4b344bec10ed6933a04d1d014260e94a6afb1b3acda4ca8093c60e80b6d3f.

Anaconda Adam 4.0.1

Released: July 23, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	4.0.0
Anaconda Enterprise Notebooks	4.0.0	4.0.1
Anaconda Mosaic	1.3.1	4.0.0
Anaconda Repository	2.19.2	4.0.0
Dask/Distributed	0.10.1/1.11.2	4.0.0
Jupyter Notebook	4.2.1	4.0.0

Backend improvements

- Store Anaconda Adam API certificates and keys in isolated location.
- Disabled `requiretty` setting in `/etc/sudoers` when installing Anaconda Enterprise Notebooks.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.0.1/download/adam-installer-4.0.1-Linux-x86_64.sh.
- MD5: 90027abca5f16917946df2c9018a443f.
- SHA256: df1e21a03745f60dba60ce99124275526deac64013f6a6d1c03b52c01eccb1b0.

Anaconda Adam 4.0.0

Released: July 22, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	4.0.0
Anaconda Enterprise Notebooks	4.0.0	4.0.0
Anaconda Mosaic	1.3.1	4.0.0
Anaconda Repository	2.19.2	4.0.0
Dask/Distributed	0.10.1/1.11.2	4.0.0
Jupyter Notebook	4.2.1	4.0.0

Features

- Sync Anaconda Adam version with Anaconda Platform version.
- Fixed password setting for Jupyter Notebook.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/4.0.0/download/adam-installer-4.0.0-Linux-x86_64.sh.
- MD5: 37662cf0d58955fc46205f2059372cd9.
- SHA256: 57781eabbae9269171f3360dfff56afd9e016480ce43cfd907d62151ef380c12.

Anaconda Adam 1.6.2

Released: July 21, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	1.6.0
Anaconda Enterprise Notebooks	4.0.0	1.6.0
Anaconda Mosaic	1.3.1	1.6.2
Anaconda Repository	2.19.2	1.6.0
Dask/Distributed	0.10.1/1.11.2	1.6.1
Jupyter Notebook	4.2.1	1.6.0

Features

Added conda `channel_alias` setting for global Anaconda installation.

Platform component updates

- Updated to Dask 0.10.0 and Distributed 1.11.0.
- Updated to Anaconda Mosaic 1.3.1.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.6.2/download/adam-installer-1.6.2-Linux-x86_64.sh.
- MD5: 859d1b4b759cc0562866ee58a8ceeb35.
- SHA256: 5d378bc7cb681f80f0b80435467d1f889788dd5efc61b5be0f1a07c5993d131e.

Anaconda Adam 1.6.0

Released: July 19, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	1.6.0
Anaconda Enterprise Notebooks	4.0.0	1.6.0
Anaconda Mosaic	1.2.3	1.6.0
Anaconda Repository	2.19.2	1.6.0
Dask/Distributed	0.10.0/1.11.0	1.6.0
Jupyter Notebook	4.2.1	1.6.0

Features

- Improvements to air gapped installation process for Adam core library.
- Added optional `sudo_su` mode for installations with restricted `sudo` functionalty.
- A `licenses` directory is now created in the root of the `adam` installation location.

Backend improvements

Improved checks for failed remote commands and conda commands.

Bug fixes

Fixed hash key/value check for Anaconda/Miniconda installers.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.6.0/download/adam-installer-1.6.0-Linux-x86_64.sh.
- MD5: ae7a4614b8e1312812baa9402f462aac.
- SHA256: 996e66e0ca956aaa05cc089cf66ebaf76bb1d8532d69900abb765f9246d5bf60.

Anaconda Adam 1.5.5

Released: July 14, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	1.5.2
Anaconda Enterprise Notebooks	4.0.0	1.5.2
Anaconda Mosaic	1.2.3	1.5.2
Anaconda Repository	2.19.2	1.5.4
Dask/Distributed	0.10.0/1.11.0	1.5.1
Jupyter Notebook	4.2.1	1.5.2

Bug fixes

Fixed password-based SSH commands used during provision/installation.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.5.5/download/adam-installer-1.5.5-Linux-x86_64.sh.
- MD5: e28887409326f3539743eda9beeff0b6.
- SHA256: 9e6eb3854ed2bff57aab6573804478ec05f8586e8b4ecd8ee6d0cd5faa44d570.

Anaconda Adam 1.5.4

Released: July 13, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.1/4.0.5	-
Anaconda Accelerate	2.3.0	1.5.2
Anaconda Enterprise Notebooks	4.0.0	1.5.2
Anaconda Mosaic	1.2.3	1.5.2
Anaconda Repository	2.19.2	1.5.4
Dask/Distributed	0.10.0/1.11.0	1.5.1
Jupyter Notebook	4.2.1	1.5.2

Features

- Added conda `channels` setting for global Anaconda installation.
- Support for multiple conda channels in conda commands.
- Added explicit Anaconda and platform component installation steps to `adam provision`, `adam ec2`, and `adam up`.
- Added customizable platform component settings to cluster definition.
- Ability to configure security settings for IPTables and SELinux.
- Improved confirmation/details when destroying Amazon EC2 instances.

Bug fixes

Fixed password-based SSH logins for cluster nodes.

Platform component updates

- Updated to Anaconda 4.1.1 (conda 4.1.6).
- Updated to Anaconda Repository 2.19.2.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.5.4/download/adam-installer-1.5.4-Linux-x86_64.sh.
- MD5: 348928b0816c93a9ccb20a32fa6a6b20.
- SHA256: a3f76295b61b02cda3d2f31c24dfd08fdb4739fb7aff3312753c5ba87ca3eaff.

Anaconda Adam 1.5.3

Released: July 11, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.1.0/4.0.5	-
Anaconda Accelerate	2.3.0	1.5.1
Anaconda Enterprise Notebooks	4.0.0	1.5.1
Anaconda Mosaic	1.2.3	1.5.1
Anaconda Repository	2.19.1	1.5.2
Dask/Distributed	0.10.0/1.11.0	1.5.0
Jupyter Notebook	4.2.1	1.5.1

Features

Changed default admin username and password for Anaconda Repository.

Bug fixes

Added `setuptools` as dependency for `Adam`.

Platform component updates

Updated to Anaconda 4.1.0 (conda 4.1.4).

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.5.3/download/adam-installer-1.5.3-Linux-x86_64.sh.
- MD5: 4e30813039ec5a7010324a3cf577c621.
- SHA256: 83b03a7704b448bda9eec05b1cf6aa0c7b384cb4c4401a820559e077c2c104d6.

Anaconda Adam 1.5.1

Released: July 8, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.0.0/4.0.5	-
Anaconda Accelerate	2.3.0	1.5.1
Anaconda Enterprise Notebooks	4.0.0	1.5.1
Anaconda Mosaic	1.2.3	1.5.1
Anaconda Repository	2.19.1	1.5.1
Dask/Distributed	0.10.0/1.11.0	1.5.0
Jupyter Notebook	4.2.1	1.5.0

Features

- Changed default AEN Gateway port to 8088 in Anaconda Enterprise Notebooks.
- Added profile settings for Anaconda Repository.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.5.1/download/adam-installer-1.5.1-Linux-x86_64.sh.
- MD5: 8dfe05136a4f022c9611092b1cfc2373.
- SHA256: 89b500affce32c0fb2df78b553d2c96277f0d7852037553710838ed4e82df36f.

Anaconda Adam 1.5.0

Released: July 7, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.0.0/4.0.5	-
Anaconda Accelerate	2.3.0	1.5.1
Anaconda Enterprise Notebooks	4.0.0	1.5.0
Anaconda Mosaic	1.2.3	1.5.1
Anaconda Repository	2.19.1	1.5.0
Dask/Distributed	0.10.0/1.11.0	1.5.0
Jupyter Notebook	4.2.1	1.5.0

Backend improvements

Updated license checks to be compatible with new platform component names.

Platform component updates

- Updated to Anaconda Repository 2.19.1.
- Updated to Anaconda Accelerate 2.3.0.
- Updated to Anaconda Mosaic 1.2.3.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.5.0/download/adam-installer-1.5.0-Linux-x86_64.sh.
- MD5: b162d5425b0d41c96da6b65e45f841e7.
- SHA256: 00d23cae15fc55fea1c82645f79d10d34e085d05c105d4a556402138730d7ae4.

Anaconda Adam 1.4.2

Released: July 6, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.0.0/4.0.5	-
Anaconda Accelerate	2.2.0	1.4.1
Anaconda Enterprise Notebooks	4.0.0	1.4.1
Anaconda Mosaic	1.2.2	1.4.1
Anaconda Repository	2.18.1	1.4.1
Dask/Distributed	0.10.0/1.11.0	1.4.2
Jupyter Notebook	4.2.1	1.4.2

Features

Added `adam describe` command to get detailed information about clusters and profiles.

Bug fixes

Fixed issue with compute node registration in Anaconda Enterprise Notebooks.

Platform component updates

- Updated to Dask 0.10.0 and Distributed 1.11.0.
- Updated to Jupyter 1.4.2.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.4.2/download/adam-installer-1.4.2-Linux-x86_64.sh.
- MD5: c8c906b1623bf109813c588a126d7407.
- SHA256: 0c17e670e13da18dc2d64e09980ce790bc0812754802f6937cc9ecc44efd0da0.

Anaconda Adam 1.4.1

Released: June 28, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.0.0/4.0.5	-
Anaconda Accelerate	2.2.0	1.4.1
Anaconda Enterprise Notebooks	4.0.0	1.4.0
Anaconda Mosaic	1.2.2	1.4.1
Anaconda Repository	2.18.1	1.4.1
Dask/Distributed	0.9.0/1.10.2	1.4.1
Jupyter Notebook	-	1.4.1

Features

Updated license checks in Dask and Notebook platform components.

Bug fixes

Fixed issue with license check.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.4.1/download/adam-installer-1.4.1-Linux-x86_64.sh.
- MD5: 5ec1faec73d8cf53c3fdb6742f86daf5.
- SHA256: 8f61c01029eaff31a1910bf56fbc3b33a0a4a845fcd86ddc12c673706dd1e06e.

Anaconda Adam 1.4.0

Released: June 28, 2016

Platform component	Version	Plugin version
Anaconda/Miniconda	4.0.0/4.0.5	-
Anaconda Accelerate	2.2.0	1.4.1
Anaconda Enterprise Notebooks	4.0.0	1.4.0
Anaconda Mosaic	1.2.2	1.4.1
Anaconda Repository	2.18.1	1.4.1
Dask/Distributed	0.9.0/1.10.2	1.4.0
Jupyter Notebook	-	1.4.1

Features

- Added `adam info` command with license information.
- Added `adam remove` command to remove cluster definition file.
- License files are now uploaded to nodes for each platform component.
- Improved handling of license issues.
- Installers are now automatically built and uploaded.

Backend improvements

Updated Salt to 2016.3.0.

Bug fixes

- Fixed Salt API timeout issue.
- Fixed issue with Salt master and minions starting up.

Download Adam

- Link: https://anaconda.org/anaconda-adam/adam-installer/1.4.0/download/adam-installer-1.4.0-Linux-x86_64.sh.
- MD5: 7d9d2ebfb02e33977fb41d1dcf1240fc.
- SHA256: a2f9c98338d4ba4e224ea815bb338d45ee719cc002af0f5c7d18a8f593a33d54.

1.4.3 Anaconda for Cluster Management

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it.

NOTE: Anaconda for cluster management has been replaced by Anaconda Scale. This documentation is made available for existing installations only.

To use Anaconda with a cluster, please use Anaconda Scale and refer to the Anaconda Scale documentation.

Anaconda for cluster management provides resource management tools to easily deploy Anaconda across a cluster. It helps you manage multiple conda environments and packages (including Python and R) on bare-metal or cloud-based clusters. Supported platforms include Amazon EC2, bare-metal clusters, or even a collection of virtual machines.

Anaconda for cluster management can be used with distributed computation frameworks such as Spark or Dask and works alongside enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP.

Anaconda for cluster management is freely available via Anaconda Cloud for unlicensed, unsupported use with up to 4 cluster nodes. Anaconda Workgroup and Enterprise include licenses for 8 and 16 nodes, respectively. If you would like to use Anaconda for cluster management with additional nodes on a bare-metal, on-premises, or cloud-based cluster, please contact sales@anaconda.com.

Features of Anaconda for cluster management:

- Easily install Python and R packages across multiple cluster nodes
- Manage multiple conda environments across a cluster
- Push local conda environments to all cluster nodes
- Manage both cloud-based and bare-metal clusters
- Remotely SSH and upload/download files to and from cluster nodes

Typical configuration of Anaconda for cluster management:

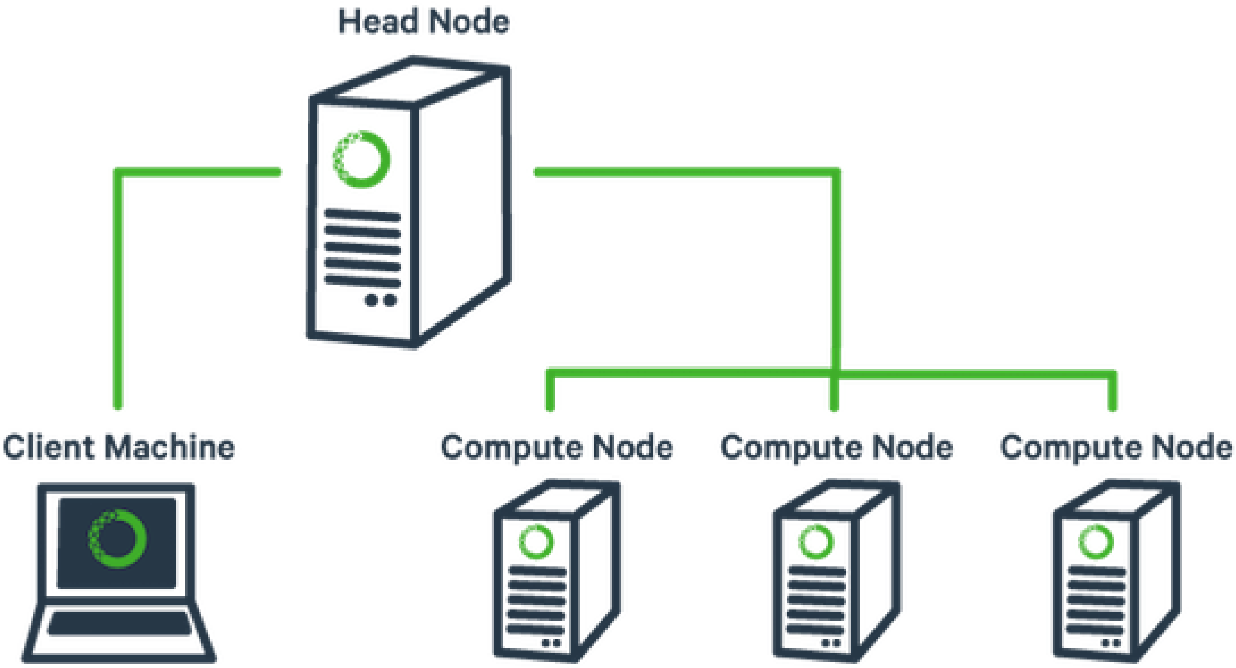


Table of contents:

Quickstart

This quickstart provides a walkthrough of Anaconda for cluster management using Amazon Web Services (AWS) Elastic Compute Cloud (EC2). The steps covered in this quickstart include defining and launching a cloud-based cluster on Amazon EC2, managing conda packages on the cluster nodes, and installing plugins.

Installation

Install Anaconda for cluster management on your local machine following the instructions on the installation page.

Create a provider

A sample providers file (shown below) is included with a new installation of Anaconda for cluster management and is located within the `~/.acluster/providers.yaml` file.

```
aws_east:
  cloud_provider: ec2
  keyname: my-private-key
  location: us-east-1
  private_key: ~/.ssh/my-private-key.pem
  secret_id: AKIAXXXXXX
  secret_key: XXXXXXXXXX
```

Edit this file and replace the settings and credentials with your information.

Refer to the *Provider settings* page for more details about provider settings, including security groups.

You can list the providers with the command:

```
$ acluster list providers
```

Create a profile

A sample profile is included with a new installation of Anaconda for cluster management and is located in the `~/.acluster/profiles.d/` directory. The sample profile named `aws_profile_sample` is shown below:

```
name: aws_profile_sample
provider: aws_east
num_nodes: 4
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
node_type: m3.large
user: ubuntu
```

You can use this profile to create a 4-node cluster based on Ubuntu 14.04.

Refer to the *Profile settings* page for more details about profile settings.

You can list the profiles with the command:

```
$ acluster list profiles
```

Create the cluster

After the provider and profile files are defined, you can create a cluster using the command:

```
$ acluster create demo_cluster --profile aws_profile_sample
```

This will create your new cluster on Amazon EC2 and provision the cluster nodes, which typically requires between 5 and 10 minutes. You will see updates as the tasks and initialization steps are completed.

Install conda packages

Now that you have a cluster running, you can install conda packages using the `acluster conda` command. The `acluster` command can be prepended to most of the `conda` commands.

To install numpy, scipy, and pandas on all of the cluster nodes, use the following command:

```
$ acluster conda install numpy scipy pandas
```

Note: Refer to the *Conda management* page for a full list of remote conda commands.

Install plugins

Anaconda for cluster management supports multiple plugins such as Apache Spark, Hadoop Distributed File System (HDFS), the Jupyter Notebook, and more. These plugins can be installed on the cluster by using the `acluster install` command.

For example, the following command can be used to install IPython Notebook on the cluster:

```
$ acluster install notebook
```

The notebook will be available on `http://{{ HEAD_NODE_IP }}:8888`. You can open the respective URLs for many of these applications in your browser using the `acluster open` command:

```
$ acluster open notebook
```

Run the `acluster open` command to view a complete list of supported applications.

Destroy the cluster

When you are finished, the following command can be used to destroy the cluster and terminate all instances in it. It will prompt for confirmation before destroying the cluster.

```
$ acluster destroy demo_cluster
```

Further information

Refer to the *Python with Spark How-tos* page for more example use cases for use-cases and example scripts.

Creating a cluster

Requirements

- **Client machine:**
  - Windows, Mac, or Linux, 64-bit
  - Anaconda for cluster management client library installed
- **Cluster Nodes:**
  - One or more 64-bit Linux machines with 8+ GB RAM, 8+ cores (recommended)
  - Access to ports 22, 4505, and 4506 from the client machine to the cluster nodes to provision the cluster via SSH and Salt

Supported operating systems

For the cluster nodes, we support and test Anaconda for cluster management with the following Linux distributions:

- CentOS 6 and CentOS 7
- RHEL 6 and RHEL 7
- Ubuntu 12.04 and Ubuntu 14.04

Cloud-based or bare-metal cluster creation

For more information about configuring and creating a cloud-based or bare-metal cluster, refer to the following documentation:

Cloud-based Cluster Setup

Anaconda for cluster management can launch and bootstrap clusters on a variety of cloud services. We currently support Amazon EC2. Other providers such as Microsoft Azure, Rackspace, Google Cloud Platform, and others are on our roadmap. If you are interested in using a cloud provider that is not listed here, please contact us at sales@anaconda.com.

Cloud configuration involves the use of `profiles` and `providers`. Splitting configurations allows users to easily share and distribute specific cluster configurations (`profiles`), while retaining private authentication credentials (`providers`).

1. Provider Setup

Authentication and cloud definitions are managed in a single provider file on the client machine: `~/.acluster/providers.yaml`. A provider file can contain multiple providers with different settings and credentials.

A sample providers file is included with a new installation of Anaconda for cluster management and is located within the `~/.acluster/providers.yaml` file on the client machine. You can edit the contents of this file to reflect the settings and credentials for your cloud provider.

An example `~/.acluster/providers.yaml` file with a provider named `aws_east` that is configured for Amazon EC2 is shown below:

```
aws_east:
  cloud_provider: ec2
  keyname: my-private-key
  location: us-east-1
  private_key: ~/.ssh/my-private-key.pem
  secret_id: AKIAXXXXXX
  secret_key: XXXXXXXXXX
```

Note that you will need access to ports 22, 4505, and 4506 from the client machine to the cluster nodes to provision the cluster via SSH and Salt.

Refer to the *Provider settings* page for more details about provider settings, including security groups.

Linux or Mac: You should set the permissions of `providers.yaml` file to `0600`.

2. Profile Setup

The settings for each cluster are managed in a profile file.

A sample profile is included with a new installation of Anaconda for cluster management and is located in the `~/.acluster/profiles.d/` directory on the client machine. You can edit the contents of this file to reflect the settings for your cluster.

An example profile located at `~/.acluster/profiles.d/profile_name.yaml` and named `profile_name` that is configured to use the `aws_east` provider is shown below:

```
name: profile_name
provider: aws_east
num_nodes: 4
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
node_type: m3.large
user: ubuntu
```

Anaconda for cluster management supports and is tested with the following Linux-based AMIs on Amazon EC2 (us-east-1 region), which can be set as the `node_id`:

OS	AMI	User
Ubuntu 12.04	ami-08faa660	ubuntu
Ubuntu 14.04	ami-d05e75b8	ubuntu
CentOS 6.6	ami-d89fb7b0	root

For more information on AMIs that are available in other Amazon EC2 regions, refer to the Amazon EC2 documentation or the Ubuntu Amazon EC2 AMI Locator.

Refer to the *Profile settings* page for more details about profile settings.

3. Create Cluster

Use the following command to launch a cluster with the specified profile:

```
$ acluster create demo_cluster --profile profile_name
```

Note: Replace `demo_cluster` with the name of your cluster, and `profile_name` with the name of your profile.

After this command is executed, the cloud-based cluster will be created, and you will see updates as the tasks and initialization steps are completed.

A typical launch takes between 5 and 10 minutes, and has output similar to the following:

```
$ acluster create demo_cluster --profile profile_name
Creating cluster
Number of existing nodes: 0
Number of requested nodes: 4
Licensed nodes: 16
License is valid for the current number of nodes.
INFO: Creating new cluster "demo_cluster" with profile "profile_name"
INFO: Creating 4 instances
INFO: Instances configuration:
INFO:   Name: demo_cluster
INFO:   Security Group: ['all-open']
INFO:   Number of Nodes: 4
INFO:   Type: <NodeSize: id=m3.large, name=Large Instance, ram=7168
        disk=32000 bandwidth=None price=0.14 driver=Amazon EC2 ...>
INFO:   Location: us-east-1
INFO:   Additional Tags {'billingProject': 'anaconda-cluster'}
INFO: Successfully created instances
INFO: Cluster info: {'ips': ['54.81.228.35', '54.167.198.242',
      '54.145.107.208', '54.166.207.40'], 'user': 'ubuntu',
      'ids': ['i-29818d89', 'i-2c818d8c', 'i-2e818d8e', 'i-2f818d8f'],
      'name': u'demo_cluster'}
Saving cluster file
Cluster "demo_cluster": 4 nodes
Number of existing nodes: 4
Number of requested nodes: 0
Licensed nodes: 16
License is valid for the current number of nodes.
Checking ssh connection
INFO: Checking SSH connection
Checking sudo
```

(continues on next page)

(continued from previous page)

```
Bootstraping conda
INFO: Installing miniconda
INFO: Anaconda URL: https://repo.continuum.io/miniconda/Miniconda-latest-Linux-x86_64.
↪sh
INFO: Checking conda installation
INFO: Checking conda installation
Bootstraping salt
INFO: Setting up Salt master and Minions
INFO: Creating conda env for salt
INFO: Installing salt
INFO: Settings grains/roles
INFO: Head roles: []
INFO: Compute roles: []
Starting salt
INFO: Flushing iptables rules
INFO: Setting SELinux to permissive
INFO: Starting salt-master daemon
INFO: Starting salt-minion daemon
Checking salt connection
Uploading formulas
INFO: Uploading formulas to head
INFO: Uploading profile
Syncing formulas
Done
```

The cluster is now ready for use.

View the *Troubleshooting FAQ* if you encounter errors while creating a cluster.

4. Destroy Cluster

If you want to destroy a cluster, use the following command:

```
$ acluster destroy demo_cluster
```

Note: Replace the name `demo_cluster` above with the actual name of your cluster.

After issuing the above command, you will be prompted to confirm termination of the specified cluster.

Bare-metal Cluster Setup

Anaconda for cluster management can be used to manage pre-existing clusters, such as bare-metal machines, collections of virtual machines (Vagrant, Docker, etc.), or previously instantiated cloud instances. These types of installations are collectively referred to as *bare-metal installations.*

Additional bare-metal cluster requirements:

- **Client machine:**
  - Passwordless SSH access to all cluster nodes
- **Cluster Nodes:**
  - Matching user account/credentials on all nodes with passwordless sudo enabled

Provisioning instructions

1. Provider Setup

A sample providers file is included with a new installation of Anaconda for cluster management and is located within the `~/.acluster/providers.yaml` file on the client machine. You can edit the contents of this file to reflect the settings for your bare metal provider.

An example `~/.acluster/providers.yaml` file with a provider named `bare_metal` is shown below:

```
bare_metal:
  cloud_provider: none
  private_key: ~/.ssh/my-private-key
```

Refer to the *Provider settings* page for more details about provider settings.

2. Profile Setup

Create a file on the client machine located at `~/.acluster/profiles.d/profile_name.yaml` that defines the following information:

```
name: profile_name
provider: bare_metal
node_id: bare_metal
node_type: bare_metal
user: ubuntu
num_nodes: 4
machines:
  head:
    - 192.168.1.1
  compute:
    - 192.168.1.2
    - 192.168.1.3
    - 192.168.1.4

# Optional for Anaconda Server
# (note that the ports might be different in your configuration)
anaconda_url: http://[your-anaconda-server-ip]:9000/Miniconda-latest-Linux-x86_64.sh
conda_channels:
  - http://[your-anaconda-server-ip]:8080/conda/anaconda-cluster
  - http://[your-anaconda-server-ip]:8080/conda/anaconda
  - defaults
```

The example profile shown above is named `profile_name` and is configured for a 4-node bare-metal cluster.

Refer to the *Profile settings* page for more details about profile settings.

3. Configure Cluster

Use the following command to configure a bare-metal cluster with the specified profile:

```
$ acluster create demo_cluster --profile profile_name
```

Note: Replace `demo_cluster` with the name of your cluster, and `profile_name` with the name of your profile.

After this command is executed, the bare-metal cluster will be configured, and you will see updates as the tasks and initialization steps are completed.

A typical bare-metal cluster configuration takes a few minutes, and has output similar to the following:

```
$ acluster create demo_cluster --profile profile_name
Creating cluster
Number of existing nodes: 0
Number of requested nodes: 4
Licensed nodes: 16
License is valid for the current number of nodes.
INFO: Creating new cluster "demo_cluster" with profile "profile_name"
INFO: Creating 4 instances
INFO: Successfully created instances
INFO: Cluster info: {'ips': ['54.84.227.194', '52.23.192.232',
      '54.84.184.193', '54.88.116.203'], 'user': 'ubuntu',
      'name': u'demo_cluster'}
Saving cluster file
Cluster "demo_cluster": 4 nodes
Number of existing nodes: 4
Number of requested nodes: 0
Licensed nodes: 16
License is valid for the current number of nodes.
Checking ssh connection
INFO: Checking SSH connection
Checking sudo
Bootstraping conda
INFO: Installing miniconda
INFO: Anaconda URL: https://repo.continuum.io/miniconda/Miniconda-latest-Linux-x86_64.
↪sh
INFO: Checking conda installation
INFO: Checking conda installation
Bootstraping salt
INFO: Setting up Salt master and Minions
INFO: Creating conda env for salt
INFO: Installing salt
Starting salt
INFO: Flushing iptables rules
INFO: Setting SELinux to permissive
INFO: Starting salt-master daemon
INFO: Starting salt-minion daemon
Checking salt connection
Uploading formulas
INFO: Uploading formulas to head
INFO: Uploading profile
Setting Roles
INFO: Settings grains/roles
Syncing formulas
Done
```

The cluster is now ready for use.

View the *Troubleshooting FAQ* if you encounter errors while creating a cluster.

Configuration

After installing Anaconda for cluster management and running any `acluster` command, the `~/.acluster` directory is created, along with a sample `provider` file located in `~/.acluster/providers.yaml` and a sample profile file located in ~/.acluster/profiles.d/.

An example of an ~/.acluster directory with multiple clusters and profiles is shown below:

```
/home/user/.acluster
├── clusters.d
│   ├── cluster1.yaml
│   ├── cluster2.yaml
│   ├── cluster3.yaml
├── profiles.d
│   ├── profile1.yaml
│   ├── profile2.yaml
└── providers.yaml
```

The profiles.d directory contains information about cluster setups, including the number and type of nodes, plugins, and other settings.

The providers.yaml file contains information about cloud providers.

The clusters.d directory contains information about clusters that are currently running.

For more information about profile and provider settings, refer to the following pages:

Profile settings

The following profile settings can be used to define cluster configurations for cloud-based or bare-metal clusters. Each named profile file should contain information for a single cluster definition.

Profile settings

anaconda_url (optional)

Set a custom location to download Anaconda/Miniconda from (default: Miniconda with Python 2).

```
anaconda_url: http://localhost/miniconda/Miniconda-latest-Linux-x86_64.sh
```

aws (optional)

Set AWS options for all instances:

- `tags` are optional key/value metadata that are assigned to each instance.
- `termination_protection` prevents your instances from being accidentally terminated (default: true).
- `use_private_ip` uses the private IP addresses of the AWS nodes instead of the public IP addresses. This is useful if you are using a custom security group (default: false).

```
aws:
  tags:
    - billingProject: anaconda-cluster
    - cluster_version: production
    - ...
  termination_protection: true
  use_private_ip: true
```

conda_channels (optional)

Set custom channels to download conda packages from on the cluster nodes. This setting will overwrite all of the configured conda channels, including the default channels. If you are using this setting, you will need to explicitly include the two default channels, `defaults` and `anaconda-cluster`, followed by your custom channels.

```
conda_channels:
  - defaults
  - anaconda-cluster
  - blaze
  - pypi
  - username
  - https://conda.anaconda.org/username/
```

default_channels (optional)

Set default_channels to download the default conda packages from the URL defined. This setting will overwrite the default conda channels: http://repo.continuum.io.

```
default_channels: http://localhost/conda/anaconda
```

channel_alias (optional)

Set channel_alias to instruct conda to look for channels in a local repository This setting will overwrite the default: http://conda.anaconda.org.

```
channel_alias: http://localhost/conda/
```

name

Name of the profile.

```
name: profile_name
```

node_id

Image to configure on each node. For bare-metal clusters, a dummy value such as `bare-metal` can be used.

```
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
```

node_type

The type of node to launch. For bare-metal clusters, a dummy value such as `bare-metal` can be used.

```
node_type: m3.large
```

num_nodes

Number of nodes to launch (cloud-based) or manage (bare-metal).

```
num_nodes: 4
```

machines (optional)

IP addresses of head and compute nodes in the cluster. Note that only IP addresses should be used with this setting, not hostnames or FQDNs.

This is used along with the *Bare-metal Cluster Setup* and can refer to bare-metal nodes, cloud-based nodes that were manually launched outside of Anaconda for cluster management, or a collection of virtual machines. If SSH is running on a port other than 22, you can optionally include the SSH port number (e.g., `192.168.1.1:2222`).

```
machines:
  head:
    - 192.168.1.1
  compute:
    - 192.168.1.2
    - 192.168.1.3
    - 192.168.1.4
```

plugins (optional)

Install plugins upon cluster creation or provisioning. Some plugins also have configruable settings. Refer to the *Plugin Settings* documentation for more information.

```
plugins:
  - notebook
  - dask
```

provider

Name of the provider to use to launch instances for a cloud-based or bare-metal cluster. For more information, refer to the *Provider settings* documentation.

```
provider: aws_east
```

root_size (optional)

Size of the root volume (GB). Currently only used for Amazon EBS volumes.

```
root_size: 200
```

security (optional)

Security settings to enable/disable SELinux or flush iptables rules (default: true).

```
security:
  disable_selinux: true
  flush_iptables: true
```

user

User to SSH to the cluster nodes as. This user must have passwordless sudo access.

```
user: ubuntu
```

Sample cloud-based profile file

Below is a sample cloud-based profile named `profile_name` located in the `~/.acluster/profiles/profile_name.yaml` file that is configured with all required and optional settings.

```
name: profile_name
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
node_type: m3.large
num_nodes: 4
provider: aws_east
root_size: 50
user: ubuntu
```

(continues on next page)

(continued from previous page)

```
anaconda_url: http://localhost/miniconda/Miniconda-latest-Linux-x86_64.sh

aws:
  tags:
    - billingProject: anaconda-cluster

plugins:
  - notebook
  - dask

default_channels: http://localhost/conda/anaconda

channel_alias: http://localhost/conda/

conda_channels:
  - defaults
  - anaconda-cluster
  - blaze
  - pypi
  - username
  - https://conda.anaconda.org/username/

security:
  disable_selinux: true
  flush_iptables: true
```

Sample bare-metal profile file

Below is a sample bare-metal profile named `profile_name` located in the `~/.acluster/profiles/profile_name.yaml` file that is configured with all required and optional settings.

```
name: profile_name
node_id: bare_metal
node_type: bare_metal
num_nodes: 4
provider: bare_metal
user: ubuntu
anaconda_url: http://localhost/miniconda/Miniconda-latest-Linux-x86_64.sh

machines:
  head:
    - 192.168.1.1
  compute:
    - 192.168.1.2
    - 192.168.1.3
    - 192.168.1.4

plugins:
  - notebook
  - dask

default_channels: http://localhost/conda/anaconda

channel_alias: http://localhost/conda/
```

(continues on next page)

(continued from previous page)

```
conda_channels:
  - defaults
  - anaconda-cluster
  - blaze
  - pypi
  - username
  - https://conda.anaconda.org/username/

security:
  disable_selinux: false
  flush_iptables: false
```

Provider settings

The following provider settings can be used to define settings for creating cloud-based or bare-metal clusters. The `providers.yaml` file can contain credentials and settings for multiple providers.

Provider settings

cloud_provider

Name of the cloud provider (`ec2`, `none`). Use `none` for bare-metal clusters.

```
cloud_provider: ec2
```

keyname (optional)

Name of the key to authenticate with the cluster nodes. This setting is used with a cloud provider.

```
keyname: my-private-key
```

location (optional)

Region or location to use with a cloud provider.

```
location: us-east-1
```

private_key

Path to the private SSH key on the client machine.

```
private_key: ~/.ssh/my-private-key.pem
```

secret_id (optional)

Secret ID to authenticate with a cloud provider.

```
secret_id: AKIA***************
```

secret_key (optional)

Secret key to authenticate with a cloud provider.

```
secret_key: RXE*************************************
```

security_group (optional)

Security Group to use with a cloud provider. Note that this is the “Group Name” in the AWS Console, not the “Name” or “Group ID”. Note that you will need access to ports 22, 4505, and 4506 from the client machine to the cluster nodes to provision the cluster via SSH and Salt. If this parameter is not specified, then a default Security Group will be created for you, called `anaconda-cluster-default`, with all ports open.

```
security_group: my-security-group
```

Sample providers file

Below is a sample providers file located in the `~/.acluster/providers.yaml` file that is configured with all required and optional settings. The `providers.yaml` file can contain credentials and settings for multiple providers. The sample providers file shown below defines two providers named `aws_east` and `bare_metal` for cloud-based and bare-metal clusters, respectively.

```
aws_east:
  cloud_provider: ec2
  keyname: my-private-key
  location: us-east-1
  private_key: ~/.ssh/my-private-key.pem
  secret_id: AKIAXXXXXX
  secret_key: XXXXXXXXXX
  security_group: my-security-group
```

(continues on next page)

(continued from previous page)

```
bare_metal:
  cloud_provider: none
  private_key: ~/.ssh/my-private-key
```

Plugins

List of available plugins

Anaconda for cluster management can install and manage the following plugins on a cluster:

- Anaconda distribution
- Dask/Distributed
- Ganglia
- IPython Parallel
- Jupyter Notebook

- Salt

The following plugins are unsupported and should only by used for prototyping/experimental purposes:

- Elasticsearch
- HDFS
- Hive
- Impala
- Kibana
- Logstash
- Spark (standalone mode or YARN)
- Storm
- YARN
- ZooKeeper

If you're interested in using Anaconda with production Hadoop clusters, Anaconda for cluster management works with enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP.

On clusters with existing enterprise Hadoop installations, Anaconda for cluster management can manage packages (e.g., for PySpark, SparkR, or Dask) and can install and manage the Jupyter Notebook and Dask plugins.

Installing Plugins

Plugins can be installed using two methods:

1. Using the cluster `profile.yaml` file when creating or provisioning a cluster.
2. Using the `acluster install` command after the cluster is created.

1. Install plugins with the profile.yaml file

When creating or provisioning a cluster, create or edit the file `~/.acluster/profiles.d/profile_name.yaml` as shown in the example below:

```
name: profile_name
provider: aws
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
user: ubuntu
node_type: m3.large
num_nodes: 5
plugins:
  - notebook
  - dask
```

Optionally, you can configure some plugin settings from within your profile. For example, you can specify a password to protect a Jupyter Notebook using:

```
name: profile_name
provider: aws
node_id: ami-d05e75b8  # Ubuntu 14.04, us-east-1 region
user: ubuntu
node_type: m3.large
num_nodes: 5
```

(continues on next page)

(continued from previous page)

```
plugins:
  - notebook:
      password: secret
```

Refer to the *Plugin Settings* section for more details on configuring plugin settings.

2. Install plugins using the acluster install command

After the cluster has been created, you can view a list of available plugins with the command `acluster install` as shown in the example below:

```
$ acluster install
Usage: acluster install [OPTIONS] COMMAND [ARGS]...

  Install plugins across the cluster

Options:
  -h, --help  Show this message and exit.

Commands:
  conda               Install (mini)conda
  dask                Install Dask/Distributed
  elasticsearch       Install Elasticsearch
  ganglia             Install Ganglia
  hdfs                Install HDFS
  hive                Install Hive
  impala              Install Impala
  ipython-parallel    Install IPython Parallel
  kibana              Install Kibana
  notebook            Install Jupyter Notebook
  logstash            Install Logstash
  salt                Install Salt
  spark-standalone    Install Spark (standalone)
  spark-yarn          Install Spark (YARN)
  storm               Install Storm
  yarn                Install YARN
  zookeeper           Install Zookeeper
```

All of the above subcommands can optionally receive a `--cluster/-x` option to specify a cluster. This option is not required if you only have one cluster running.

Note: Salt is the base for all plugins except Conda. Therefore, the Salt plugin needs to be installed before the other plugins can be installed. Salt is installed by default when you create or provision a cluster using the `acluster create` or `acluster provision` command.

Plugin Settings

The following plugins support custom settings that can be defined within a profile.

Conda

Configure conda packages and environments to be installed upon cluster creation or provisioning:

```
name: profile_name
plugins:
  - conda:
      environments:
        root:
          - numpy
        py27:
          - python=2.7
          - scipy
          - numba
        py34:
          - python=3.4
          - pandas
          - nltk
```

Configure the installation location of conda (default: `/opt/anaconda`):

```
name: profile_name
plugins:
  - conda:
      install_prefix: /opt/another_anaconda
```

Set `conda_sh` to false to disable the creation of `/etc/profile.d/conda.sh` on cluster nodes (default: true):

```
name: profile_name
plugins:
  - conda:
      conda_sh: false
```

Set `conda_group_name` to any valid unix group (default: anaconda):

```
name: profile_name
plugins:
  - conda:
      conda_group_name: anaconda
```

Set `conda_acl` to a list of users who will be given access for Anaconda Cluster admin capabilities. Note: When using this setting at least one user must have sudo access during the provisioning phase. Typically, this will include the user previously set above as `user`.

```
name: profile_name
plugins:
  - conda:
      conda_acl:
        - user1
        - user2
```

Set `ssl_verify` to a custom SSL path or to `False` to disable SSL verification for `conda`. Refer to the conda documentation for more information.

```
name: profile_name
plugins:
  - conda:
      ssl_verify: False
```

Dask

You can optionally set the number of processes (`nprocs`) to use for the Dask/Distributed workers and the `host` to access the Dask/Distributed UI via a browser. Refer to the Dask distributed scheduler documentation for more information about these settings.

```
name: profile_name
plugins:
  - dask:
      nprocs: 8
      host: dask-ui.com
```

HDFS

By default, the HDFS plugin is configured to use the following directories: `/data/dfs/nn` on the namenode and `/data/dfs/dn` on the datanodes. When multiple drives are available, the same directories are used on all non-root drives.

You can optionally set custom directories for the HDFS namenode and datanode. For example, the following settings can be used if you want to utilize a large root volume (e.g., using the `root_volume` profile setting with Amazon EC2):

```
name: profile_name
plugins:
  - hdfs:
      namenode_dirs:
        - /data/dfs/nn
      datanode_dirs:
        - /data/dfs/dn
```

Jupyter Notebook

Set a custom password to protect a Jupyter Notebook (default: `acluster`):

```
name: profile_name
plugins:
  - notebook:
      password: acluster
```

Set a custom port for the Jupyter Notebook server (default: `8888`):

```
name: profile_name
plugins:
  - notebook:
      port: 8888
```

Set a custom directory for Jupyter Notebooks (default: `/opt/notebooks`):

```
name: profile_name
plugins:
  - notebook:
      directory: /opt/notebooks
```

Custom download settings for plugins

Most plugins are installed from the standard package management repositories for your Linux distribution. Some plugins are downloaded directly from their source/project website. You can override the default download settings for the following plugins:

```
name: profile_name
plugins:
  - elasticsearch:
      download_url: https://download.elasticsearch.org/elasticsearch/release/org/
↪elasticsearch/distribution/tar/elasticsearch/2.1.0/elasticsearch-2.1.0.tar.gz
      download_hash: sha1=b6d681b878e3a906fff8c067b3cfe855240bffbb
      version: elasticsearch-2.1.0
  - logstash:
      download_url: https://download.elastic.co/logstash/logstash/logstash-2.1.1.tar.
↪gz
      download_hash: sha1=d71a6e015509030ab6012adcf79291994ece0b39
      version: logstash-2.1.1
  - kibana:
      download_url: https://download.elastic.co/kibana/kibana/kibana-4.3.0-linux-x64.
↪tar.gz
      download_hash: sha1=d64e1fc0ddeaaab85e168177de6c78ed82bb3a3b
      version: kibana-4.3.0-linux-x64
  - storm:
      source_url: http://apache.arvixe.com/storm/apache-storm-0.9.5/apache-storm-0.9.
↪5.tar.gz
      version_name: apache-storm-0.9.5
```

Managing Plugins

The following commands can be used to manage your plugins:

Plugin Status

Check the status of the plugins by using the `acluster status` command:

```
$ acluster status
Usage: acluster status [OPTIONS] COMMAND [ARGS]...

Options:
  -x, --cluster TEXT                Cluster name
  -l, --log-level [info|debug|error]
                                    Library logging level
  -h, --help                        Show this message and exit.

Commands:
  conda              Check Conda status
  dask               Check Dask/Distributed status
  elasticsearch      Check Elasticsearch status
  ganglia            Check Ganglia status
  hdfs               Check HDFS status
  hive               Check Hive status
  impala             Check Impala status
  ipython-parallel   Check IPython Parallel status
```

(continues on next page)

(continued from previous page)

```
  kibana            Check Kibana status
  notebook          Check IPython/Jupyter Notebook status
  salt              Check Salt package status
  salt-conda        Check Salt Conda module status
  salt-key          Check salt-minion keys status
  spark-standalone  Check Spark (standalone) status
  ssh               Check SSH status
  storm             Check Storm status
  yarn              Check YARN status
```

Example:

```
$ acluster status conda
Checking status of conda on cluster: demo_cluster
54.81.187.22: True
54.91.219.252: True
54.163.26.229: True
54.145.10.211: True
Status all: True

$ acluster status salt
54.167.128.130: True
54.205.160.114: True
50.16.32.99: True
50.16.34.82: True
54.163.143.34: True
Status all: True

$ acluster status salt-key
ip-10-156-23-215.ec2.internal: True
ip-10-147-47-235.ec2.internal: True
ip-10-225-181-251.ec2.internal: True
ip-10-237-145-2.ec2.internal: True
ip-10-156-30-32.ec2.internal: True
Status all: True

$ acluster status salt-conda
ip-10-156-23-215.ec2.internal: True
ip-10-237-145-2.ec2.internal: True
ip-10-225-181-251.ec2.internal: True
ip-10-147-47-235.ec2.internal: True
ip-10-156-30-32.ec2.internal: True
Status all: True
```

The multiple salt status checks perform different checks. The `acluster status salt` command verifies that the salt package is installed, the `acluster status salt-key` command verifies that the salt minions are connected to the head node, and the `acluster status salt-conda` command verifies that the salt conda module is distributed across the cluster and is operational.

All of the above checks should return a successful status after executing `acluster install salt`.

Open Plugin UIs

Some of the plugins provide a browser UI that can be displayed to the user. The `acluster open` command is a utility command that opens a browser window corresponding to each plugin. The command:

```
$ acluster open notebook
```

will open a browser window to the Jupyter Notebook (port 8888).

Use the `--no-browser` option to print the URL for the plugin interface without opening a browser window.

```
$ acluster open notebook --no-browser
notebook: http://54.172.82.53:8888
```

Restart Plugins

If a plugin is not working correctly, restart the processes by using the `acluster restart` command.

Stop Plugins

To stop a plugin, use the `acluster stop` command.

Plugin Notes and Network Ports

HDFS

Requires: `salt`

Distributed file system used by many of the distributed analytics engines such as `Impala`, `Hive`, and `Spark`.

Service	Location	Port
NameNode UI	Head node	50070
HDFS Master	Head node	9000
WebHDFS	Head node	14000

Jupyter Notebook

Requires: `salt`

Web-based interactive computational environment for Python.

Service	Location	Port
Notebook	Head node	8888

Notebooks are saved in the directory `/opt/notebooks`. You can upload and download notebooks from the cluster by using the `put` and `get` commands, respectively.

```
$ acluster put mynotebook.ipynb /opt/notebooks/mynotebook.ipynb
$ acluster get /opt/notebooks/mynotebook.ipynb mynotebook.ipynb
```

Miniconda

Python distribution from Anaconda – ships with Python libraries for large-scale data processing, predictive analytics, and scientific computing.

Salt

Requires: `conda`

Configuration management system.

Service	Location	Port
Salt Master/Minion	All nodes	4505
Salt Master/Minion	All nodes	4506

Spark

Computing framework and analytics engine written in Java/Scala with a Python interface (PySpark).

Service	Location	Port
Spark UI (standalone)	Head node	8080

YARN

Resource manager in which all Hadoop and Hadoop-like jobs are run.

Service	Location	Port
Resource UI	Head node	9026
NodeManager	Compute node(s)	9035

Cluster management

Overview

Anaconda for cluster management provides functionality to easily interact with clusters, including:

- Secure shell (SSH) to cluster nodes

- Executing commands on cluster nodes
- Uploading (put) and downloading (get) files from cluster nodes
- Executing salt module commands: http://docs.saltstack.com/en/latest/ref/modules/all/

For a concise overview of all of the available commands, view the `Anaconda for cluster management Cheat Sheet`.

For the `acluster` commands shown below, you can optionally specify a `--cluster/-x` option to specify a target cluster to operate on. This option is not required if only one cluster is running.

SSH access

The command `acluster ssh` can be used to SSH into any node within the cluster.

The first (optional) argument is a zero-indexed number that refers to a node, where `0` is the head node, `1` is the first compute node, and so on.

Note: This functionality requires SSH. **Windows users:** Install an SSH client such as PuTTY.

Executing commands

The `acluster cmd` command allows commands to be executed across multiple nodes on the cluster. This functionality is performed via salt (default) or fabric (using the `--ssh/-s` flag).

Salt provides faster execution times (especially in large clusters) of remote commands, assuming salt is installed and running on the cluster. Salt is installed by default on cloud providers and can be installed on bare-metal clusters using the command `acluster install salt`.

Fabric provides remote command execution based on SSH, which can be slow in big clusters but does not require salt to be installed on the cluster. Using fabric, you can also target specific nodes using the `--target/-t` option. The `--target/-t` option can receive the keywords 'head' and 'compute', a number (zero-indexed, where 0 is the head node and 1 is the first compute node), or two numbers separated by a comma to target a group of consecutive nodes.

Example:

```
$ acluster cmd 'date'
Executing command "date" target: "*" cluster: "demo_cluster"
4 nodes response: Thu May 14 19:21:48 UTC 2015
1 nodes response: Thu May 14 19:21:44 UTC 2015
1 nodes response: Thu May 14 19:21:47 UTC 2015

$ acluster cmd 'date' --ssh
54.196.149.0: Thu May 14 19:22:06 UTC 2015
54.159.15.86: Thu May 14 19:22:03 UTC 2015
107.21.159.226: Thu May 14 19:22:06 UTC 2015
107.20.60.246: Thu May 14 19:22:06 UTC 2015
54.237.177.104: Thu May 14 19:22:06 UTC 2015
54.227.91.24: Thu May 14 19:22:06 UTC 2015

$ acluster cmd 'date' --ssh -t head
107.21.159.226: Thu May 14 19:23:02 UTC 2015

$ acluster cmd 'date' --ssh -t 3,6
54.196.149.0: Thu May 14 19:23:29 UTC 2015
54.159.15.86: Thu May 14 19:23:26 UTC 2015
54.227.91.24: Thu May 14 19:23:29 UTC 2015
```

Upload/download files

The commands `acluster put` and `acluster get` allow files to be uploaded to and downloaded from any node in the cluster. By default, the `put` and `get` commands target the head node.

Both of the `put` and `get` commands have the same `--target/-t` option, which has the same functionality as `cmd` with the `-ssh/-s` flag.

Example:

```
$ acluster put environment.yml /tmp/env.yml
Uploading file "environment.yml" to "/tmp/env.yml" - cluster: "demo_cluster" - hosts:
"['107.21.159.226']"

$ acluster put environment.yml /tmp/env.yml --all
Uploading file "environment.yml" to "/tmp/env.yml" - cluster: "demo_cluster" - hosts:
"['107.21.159.226', '107.20.60.246', '54.237.177.104', '54.159.15.86', '54.196.149.0',
'54.227.91.24']"

$ acluster put environment.yml /tmp/env.yml -t ,3
Uploading file "environment.yml" to "/tmp/env.yml" - cluster: "demo_cluster" - hosts:
"['107.21.159.226', '107.20.60.246', '54.237.177.104']"

$ acluster put environment.yml /tmp/env.yml -t compute
Uploading file "environment.yml" to "/tmp/env.yml" - cluster: "demo_cluster" - hosts:
"['107.20.60.246', '54.237.177.104', '54.159.15.86', '54.196.149.0', '54.227.91.24']"
```

The `get` command has similar behavior. Although the `-target/-t` option is also available, it might not make sense to download the same file repeatedly since it will get overwritten.

Example:

```
$ acluster get /tmp/env.yml env2.yml
Downloading file "/tmp/env.yml" to "env2.yml" - cluster: "demo_cluster" - hosts:
"['107.21.159.226']"

$ acluster get /tmp/env.yml env2.yml --all
Downloading file "/tmp/env.yml" to "env2.yml" - cluster: "demo_cluster" - hosts:
"['107.21.159.226', '107.20.60.246', '54.237.177.104', '54.159.15.86', '54.196.149.0',
 '54.227.91.24']"
Warning: Local file /tmp/env2.yml already exists and is being overwritten.
Warning: Local file /tmp/env2.yml already exists and is being overwritten.
```

Command history

The `acluster history` command displays commands that were previously executed using `acluster`. By default, the ten most recent commands are shown.

```
$ acluster history
2015-06-09 23:39:22,515: acluster create demo_cluster --profile profile_name
2015-06-10 00:00:46,151: acluster ssh
2015-06-10 00:01:35,304: acluster acluster conda install numpy scipy
2015-06-10 00:02:01,029: acluster open notebook
```

Centralized logging

You can install the Elasticsearch, Logstash, and Kibana (ELK) plugins and use them to collect, search, and analyze data. When you install the ELK plugins, they come preconfigured for centralized logging of other Anaconda for cluster management plugins. For more information about plugins, refer to the *Plugins* documentation.

To install the ELK plugins, insert the following in your cluster profile before you create or provision a cluster:

```
plugins:
  - elasticsearch
  - logstash
  - kibana
```

or, use the following command to install the ELK plugins on an active cluster:

```
$ acluster install elasticsearch logstash kibana
```

After the ELK plugins are installed, they will begin collecting logs for various services, and the log data can be searched using Elasticsearch and visualized using Kibana. You can open the Kibana UI in your browser using the following command:

```
$ acluster open kibana
```

By default, the following index patterns are available in Kibana:

```
YYYY-MM-DD
salt
hdfs
yarn
spark_yarn
spark_standalone
zookeeper
impala
hive
hbase
hadoop-httpfs
mapreduce
syslog
```

Note that you can use the ELK plugins in a number of different workflows once they are installed on the cluster, including:

- Viewing and searching logs for all cluster-related plugins
- Viewing logs for Spark or YARN jobs for centralized job monitoring and troubleshooting
- Utilizing centralized logging capabilities from your application

Anaconda for cluster management client information

The `acluster info` command displays version information related to the Anaconda for cluster management client installation.

```
$ acluster info
anaconda-cluster version: 1.0
Platform: Darwin-14.3.0-x86_64
Processor: i386
```

(continues on next page)

(continued from previous page)

```
Byte-ordering: little
Python version: 2.7.10 |Continuum Analytics, Inc.| (default, May 28 2015, 17:04:42)
[GCC 4.2.1 (Apple Inc. build 5577)]
apache-libcloud version: 0.16.0
yaml version: 3.11
```

Advanced: Running Salt modules

By default, salt provides quite a bit of functionality in its default modules. All of this functionality is available by using the `acluster function` command.

For example, you can ping all of the cluster nodes to see that they are available:

```
$ acluster function test.ping
ip-10-153-156-45.ec2.internal:
    True
ip-10-169-57-125.ec2.internal:
    True
ip-10-181-18-153.ec2.internal:
    True
ip-10-144-199-24.ec2.internal:
    True
ip-10-113-145-230.ec2.internal:
    True
ip-10-136-78-206.ec2.internal:
    True
```

Advanced functions are also available, such as listing all of the IP address associated with a node:

```
$ acluster function network.ipaddrs
ip-10-169-57-125.ec2.internal:
    - 10.169.57.125
ip-10-144-199-24.ec2.internal:
    - 10.144.199.24
ip-10-181-18-153.ec2.internal:
    - 10.181.18.153
ip-10-153-156-45.ec2.internal:
    - 10.153.156.45
ip-10-136-78-206.ec2.internal:
    - 10.136.78.206
ip-10-113-145-230.ec2.internal:
    - 10.113.145.230
```

Another common salt command can be used to install packages using the system package manager:

```
 $ acluster function pkg.install httpd
 ip-10-136-78-206.ec2.internal:
     ...
     httpd:
         ----------
         new:
             2.2.15-39.el6.centos
         old:
     ...
```

(continues on next page)

(continued from previous page)

```
 ip-10-144-199-24.ec2.internal:
     ...
     httpd:
         ----------
         new:
             2.2.15-39.el6.centos
         old:
     ...

...
```

For more information on salt commands, see http://docs.saltstack.com/en/latest/ref/modules/all/

Advanced: Syncing Salt formulas

The `acluster sync` command is an advanced feature that synchronizes the Salt formulas from the client to the head node and then across the cluster.

Conda management

Overview

One of the primary features of Anaconda for cluster management is the remote deployment and management of Anaconda environments across a cluster.

Prepending `acluster` to core `conda` commands will execute those commands across all of the cluster nodes.

For example, to view the conda environments on all of the cluster nodes:

```
$ acluster conda info -e
All nodes (x3) response:
# conda environments:
```

(continues on next page)

(continued from previous page)

```
#
root                  *  /opt/anaconda
```

To install numpy on all of the cluster nodes:

```
$ acluster conda install numpy
Installing (u'numpy',) on cluster "demo_cluster"
Node "ip-10-234-8-208.ec2.internal":
    Successful actions: 1/1
Node "ip-10-170-59-28.ec2.internal":
    Successful actions: 1/1
Node "ip-10-232-42-58.ec2.internal":
    Successful actions: 1/1
```

For a concise overview of all of the available commands, view the `Anaconda for cluster management Cheat Sheet`.

Installing packages from channels

You can install packages from Anaconda Cloud or from an Anaconda Repository installation by adding the `--channel/-c` option. For example, to install the apache-libcloud package from the *anaconda-cluster* channel:

```
$ acluster conda install -c https://conda.anaconda.org/anaconda-cluster apache-
↪libcloud
Installing (u'apache-libcloud',) on cluster "demo_cluster"
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1
$ acluster conda list | grep apache-libcloud
- 'apache-libcloud: 0.16.0 (py27_0)'
```

List of remote conda commands

The following conda commands are available in Anaconda for cluster management:

- `acluster conda install` – Install package(s)
- `acluster conda update` – Update package(s)
- `acluster conda remove` – Remove package(s)
- `acluster conda list` – List package(s)
- `acluster conda create` – Create conda environment
- `acluster conda info` – Display information about current conda install
- `acluster conda push` – Push conda environment to cluster

For more information about conda, refer to the conda documentation.

Example remote conda commands

Install conda packages

Any conda package can be installed on a cluster managed by Anaconda for cluster management:

```
$ acluster conda install numpy

Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1
```

Verify that the package was installed using the `list` command:

```
$ acluster conda list

All nodes (x2) response:
...
- 'numpy: 1.9.2 (py27_0)'
...
```

You may also install multiple conda packages with a single command:

```
$ acluster conda install scipy pandas scikit-learn
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 3/3
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 3/3

$ acluster conda list
All nodes (x2) response:
...
- 'pandas: 0.16.1 (np19py27_0)'
...
- 'scikit-learn: 0.16.1 (np19py27_0)'
- 'scipy: 0.15.1 (np19py27_0)'
...
```

Note: It is recommended that you install all of the packages you want at once. Installing one package at a time can result in dependency conflicts.

List conda packages

A useful conda command to run on a cluster is to list the packages that are available on the nodes (by default all of the nodes should have the same packages):

```
$ acluster conda list

All nodes (x2) response:
- 'libsodium: 0.4.5 (0)'
- 'sqlite: 3.8.4.1 (1)'
- 'conda-env: 2.1.4 (py27_0)'
- 'python: 2.7.9 (3)'
...
```

Update conda packages

You can also specify which versions of conda packages to install or update:

```
$ acluster conda install pandas==0.13
Installing (u'pandas==0.13',) on cluster "demo_cluster"
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1
$ acluster conda list | grep pandas
- 'pandas: 0.13.0 (np18py27_0)'

$ acluster conda update pandas
Updating (u'pandas',) on cluster "demo_cluster"
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1
$ acluster conda list | grep pandas
- 'pandas: 0.16.1 (np19py27_0)'
```

Remove conda packages

You can remove conda packages across a cluster:

```
$ acluster conda remove pandas
Removing (u'pandas',) on cluster "demo_cluster"
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1

$ acluster conda list | grep pandas
... NO OUTPUT ...
```

Create conda environments

You can also manage conda environments across a cluster with Anaconda for cluster management.

To create a new conda environment that contains Python and numpy, use the command `conda create -n test_env numpy`.

On a cluster, use the same command and simply prepend `acluster` as shown:

```
$ acluster conda create -n test_env numpy
All nodes (x2) response:
Conda environment "test_env" created
```

Once the environment is created, refer to that named environment by adding the `-n` name option to `conda` commands:

```
$ acluster conda list -n test_env
All nodes (x2) response:
- 'sqlite: 3.8.4.1 (1)'
```

(continues on next page)

(continued from previous page)

```
- 'python: 2.7.9 (3)'
- 'zlib: 1.2.8 (0)'
- 'openssl: 1.0.1k (1)'
- 'system: 5.8 (2)'
- 'tk: 8.5.18 (0)'
- 'setuptools: 15.2 (py27_0)'
- 'pip: 6.1.1 (py27_0)'
- 'readline: 6.2 (2)'
- 'numpy: 1.9.2 (py27_0)'

$ acluster conda install -n test_env requests
Installing (u'requests',) on cluster "d" - target: "*"
Node "ip-10-136-80-92.ec2.internal":
    Successful actions: 1/1
Node "ip-10-63-173-62.ec2.internal":
    Successful actions: 1/1

$ acluster conda list -n test_env
All nodes (x2) response:
- 'sqlite: 3.8.4.1 (1)'
- 'python: 2.7.9 (3)'
- 'zlib: 1.2.8 (0)'
- 'openssl: 1.0.1k (1)'
- 'system: 5.8 (2)'
- 'tk: 8.5.18 (0)'
- 'setuptools: 15.2 (py27_0)'
- 'pip: 6.1.1 (py27_0)'
- 'readline: 6.2 (2)'
- 'numpy: 1.9.2 (py27_0)'
- 'requests: 2.7.0 (py27_0)'
```

Push conda environments

You can also push conda environments from the client machine to the cluster by using a conda `environment.yml` file:

```
$ acluster conda push ./environment.yml
['ip-10-234-8-208.ec2.internal'] nodes response:
Conda environment with "/tmp/anaconda-cluster/environment.yml" created
...
```

List conda environments

To verify that the environment has been pushed to all nodes, use the `info` command:

```
$ acluster conda info -e
All nodes (x3) response:
# conda environments:
#
stats                    /opt/anaconda/envs/stats
root                  *  /opt/anaconda
```

Python with Spark How-tos

These how-tos will show you how to run Python tasks on a Spark cluster using the PySpark module. These how-tos will also show you how to interact with data stored within HDFS on the cluster.

While these how-tos are not dependent on each other and can be accomplished in any order it is recommended that you begin with the *Overview of Spark, YARN and HDFS* first.

Overview of Spark, YARN and HDFS

Spark is an analytics engine and framework that is capable of running queries 100 times faster than traditional MapReduce jobs written in Hadoop. In addition to the performance boost, developers can write Spark jobs in Scala, Python and Java if they so desire. Spark can load data directly from disk, memory and other data storage technologies such as Amazon S3, Hadoop Distributed File System (HDFS), HBase, Cassandra, etc.

You can install Spark using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

Submitting Spark Jobs

Spark scripts are often developed interactively and can be written as a script file or as a Jupyter notebook file.

A Spark script can be submitted to a Spark cluster using various methods:

- Running the script directly on the head node.
- Using the spark-submit script in either Standalone mode or with the YARN resource manager
- Interactively in an IPython shell or Jupyter Notebook on the cluster

To run a script on the head node, simply execute `python example.py` on the cluster.

Note: that in order to launch Jupyter Notebook on the cluster, the plugin must already be installed. See the *Plugins* documentation for more information.

Working with Data in HDFS

Moving data in and around HDFS can be difficult. If you need to move data from your local machine to HDFS, from Amazon S3 to HDFS, from Amazon S3 to Redshift, from HDFS to Hive and so on, we recommend using odo, which is part of the Blaze ecosystem. `Odo` efficiently migrates data from the source to the target through a network of conversions.

If you are unfamiliar with Spark and/or SQL, we recommend using Blaze to express selections, aggregations, groupbys, etc. in a dataframe-like style. Blaze provides Python users with a familiar interface to query data that exists in different data storage systems.

How to Run a Spark Standalone Job

Overview

This is a minimal Spark script that imports PySpark, initializes a SparkContext and performs a distributed calculation on a Spark cluster in standalone mode.

Who is this for?

This how-to is for users of a Spark cluster that has been configured in standalone mode who wish to run Python code.

Spark Standalone Summary

1. *Before you start*
2. *Modifying the script*
3. *Running the job*
4. *Troubleshooting*
5. *Further information*

Before you start

To execute this example, download the `cluster-spark-basic.py example script` to the cluster node where you submit Spark jobs.

For this example, you'll need Spark running with the standalone scheduler. You can install Spark using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP. Some additional configuration might be necessary to use Spark in standalone mode.

Modifying the script

After downloading the `cluster-spark-basic.py example script` open the file in a text editor on your cluster. Replace `HEAD_NODE_HOSTNAME` with the hostname of the head node of the Spark cluster.

```
# cluster-spark-basic.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('spark://HEAD_NODE_HOSTNAME:7077')
conf.setAppName('spark-basic')
sc = SparkContext(conf=conf)

def mod(x):
    import numpy as np
    return (x, np.mod(x, 2))

rdd = sc.parallelize(range(1000)).map(mod).take(10)
print rdd
```

Let's analyze the contents of the `spark-basic.rst example script`. The first code block contains imports from PySpark.

The second code block initializes the SparkContext and sets the application name.

The third code block contains the analysis code that calculates the modulus of a range of numbers up to 1000 using the NumPy package and returns/prints the first 10 results.

Note: you may have to install NumPy with `acluster conda install numpy`.

Running the job

You can run this script by submitting it to your cluster for execution using spark-submit or by running this command

```
python cluster-spark-basic.py
```

The output from the above command shows the first ten values that were returned from the `cluster-spark-basic.py` script.

```
16/05/05 22:26:53 INFO spark.SparkContext: Running Spark version 1.6.0

[...]

16/05/05 22:27:03 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID␣
↪0, localhost, partition 0,PROCESS_LOCAL, 3242 bytes)
16/05/05 22:27:04 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory␣
↪on localhost:46587 (size: 2.6 KB, free: 530.3 MB)
16/05/05 22:27:04 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID␣
↪0) in 652 ms on localhost (1/1)
16/05/05 22:27:04 INFO cluster.YarnScheduler: Removed TaskSet 0.0, whose tasks have␣
↪all completed, from pool
16/05/05 22:27:04 INFO scheduler.DAGScheduler: ResultStage 0 (runJob at PythonRDD.
↪scala:393) finished in 4.558 s
16/05/05 22:27:04 INFO scheduler.DAGScheduler: Job 0 finished: runJob at PythonRDD.
↪scala:393, took 4.951328 s
[(0, 0), (1, 1), (2, 0), (3, 1), (4, 0), (5, 1), (6, 0), (7, 1), (8, 0), (9, 1)]
```

Troubleshooting

If something goes wrong consult the *FAQ / Known issues* page.

Further information

See the Spark and PySpark documentation pages for more information.

How to Run with the YARN resource manager

Overview

This script runs on the Spark cluster with the YARN resource manager and returns the hostname of each node in the cluster.

Who is this for?

This how-to is for users of a Spark cluster who wish to run Python code using the YARN resource mananger.

Spark YARN Summary

1. *Before you start*

Before you start

To execute this example, download the `cluster-spark-yarn.py example script` to your cluster.

For this example, you'll need Spark running with the YARN resource manager. You can install Spark and YARN using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

Running the Job

Here is the complete script to run the Spark + YARN example in PySpark:

```
# cluster-spark-yarn.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-yarn')
sc = SparkContext(conf=conf)


def mod(x):
    import numpy as np
    return (x, np.mod(x, 2))

rdd = sc.parallelize(range(1000)).map(mod).take(10)
print rdd
```

Note: you may have to install NumPy with `acluster conda install numpy`.

Run the script on the Spark cluster with spark-submit. The output shows the first ten values that were returned from the `cluster-spark-basic.py` script.

```
16/05/05 22:26:53 INFO spark.SparkContext: Running Spark version 1.6.0

[...]

16/05/05 22:27:03 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID␣
↪0, localhost, partition 0,PROCESS_LOCAL, 3242 bytes)
16/05/05 22:27:04 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory␣
↪on localhost:46587 (size: 2.6 KB, free: 530.3 MB)
16/05/05 22:27:04 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID␣
↪0) in 652 ms on localhost (1/1)
16/05/05 22:27:04 INFO cluster.YarnScheduler: Removed TaskSet 0.0, whose tasks have␣
↪all completed, from pool
16/05/05 22:27:04 INFO scheduler.DAGScheduler: ResultStage 0 (runJob at PythonRDD.
↪scala:393) finished in 4.558 s
16/05/05 22:27:04 INFO scheduler.DAGScheduler: Job 0 finished: runJob at PythonRDD.
↪scala:393, took 4.951328 s
[(0, 0), (1, 1), (2, 0), (3, 1), (4, 0), (5, 1), (6, 0), (7, 1), (8, 0), (9, 1)]
```

Troubleshooting

If something goes wrong consult the *FAQ / Known issues* page.

Further information

See the Spark and PySpark documentation pages for more information.

How to perform a word count on text data in HDFS

Overview

This example counts the number of words in text files that are stored in HDFS.

Who is this for?

This how-to is for users of a Spark cluster who wish to run Python code using the YARN resource manager that reads and processes files stored in HDFS.

Spark Wordcount Summary

1. *Before you start*
2. *Load HDFS data*
3. *Running the Job*
4. *Troubleshooting*
5. *Further information*

Before you start

To execute this example, download the `cluster-spark-wordcount.py example script` and the `cluster-download-wc-data.py script`.

For this example, you'll need Spark running with the YARN resource manager and the Hadoop Distributed File System (HDFS). You can install Spark, YARN, and HDFS using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

You will also need valid Amazon Web Services (AWS) credentials.

Load HDFS data

First, we will load the sample text data into the HDFS data store. The following script will transfer sample text data (approximately 6.4 GB) from a public Amazon S3 bucket to the HDFS data store on the cluster.

Download the `cluster-download-wc-data.py script` to your cluster and Insert your Amazon AWS credentials in the `AWS_KEY` and `AWS_SECRET` variables.

```
import subprocess

AWS_KEY = ''
AWS_SECRET = ''

s3_path = 's3n://{0}:{1}@blaze-data/enron-email'.format(AWS_KEY, AWS_SECRET)
cmd = ['hadoop', 'distcp', s3_path, 'hdfs:///tmp/enron']
subprocess.call(cmd)
```

Note: The `hadoop distcp` command might cause HDFS to fail on smaller instance sizes due to memory limits.

Run the `cluster-download-wc-data.py` script on the Spark cluster.

```
python cluster-download-wc-data.py
```

After a few minutes, the text data will be in the HDFS data store on the cluster and ready for analysis.

Running the Job

Download the `cluster-spark-wordcount.py example script` to your cluster. This script will read the text files downloaded in step 2 and count all of the words.

```
# cluster-spark-wordcount.py
from pyspark import SparkConf
from pyspark import SparkContext

HDFS_MASTER = 'HEAD_NODE_IP'

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-wordcount')
conf.set('spark.executor.instances', 10)
sc = SparkContext(conf=conf)

distFile = sc.textFile('hdfs://{0}:9000/tmp/enron/*/*.txt'.format(HDFS_MASTER))

nonempty_lines = distFile.filter(lambda x: len(x) > 0)
print 'Nonempty lines', nonempty_lines.count()

words = nonempty_lines.flatMap(lambda x: x.split(' '))

wordcounts = words.map(lambda x: (x, 1)) \
                  .reduceByKey(lambda x, y: x+y) \
                  .map(lambda x: (x[1], x[0])).sortByKey(False)

print 'Top 100 words:'
print wordcounts.take(100)
```

Replace the `HEAD_NODE_IP` text with the IP address of the head node.

Run the script on your Spark cluster using spark-submit The output shows the top 100 words from the sample text data that were returned from the Spark script.

```
54.237.100.240: Using Spark's default log4j profile: org/apache/spark/log4j-defaults.
↪properties
15/06/13 04:58:42 INFO SparkContext: Running Spark version 1.4.0
```

(continues on next page)

(continued from previous page)

```
[...]

15/06/26 04:32:03 INFO YarnScheduler: Removed TaskSet 7.0, whose tasks have all␣
↪completed, from pool
15/06/26 04:32:03 INFO DAGScheduler: ResultStage 7 (runJob at PythonRDD.scala:366)␣
↪finished in 0.210 s
15/06/26 04:32:03 INFO DAGScheduler: Job 3 finished: runJob at PythonRDD.scala:366,␣
↪took 18.124243 s
[(288283320, ''), (22761900, '\t'), (19583689, 'the'), (13084511, '\t0'), (12330608,
↪'-'),
(11882910, 'to'), (11715692, 'of'), (10822018, '0'), (10251855, 'and'), (6682827, 'in
↪'),
(5463285, 'a'), (5226811, 'or'), (4353317, '/'), (3946632, 'for'), (3695870, 'is'),
(3497341, 'by'), (3481685, 'be'), (2714199, 'that'), (2650159, 'any'), (2444644,
↪'shall'),
(2414488, 'on'), (2325204, 'with'), (2308456, 'Gas'), (2268827, 'as'), (2265197, 'this
↪'),
(2180110, '$'), (1996779, '\t$0'), (1903157, '12:00:00'), (1823570, 'The'), (1727698,
↪'not'),
(1626044, 'such'), (1578335, 'at'), (1570484, 'will'), (1509361, 'has'), (1506064,
↪'Enron'),
(1460737, 'Inc.'), (1453005, 'under'), (1411595, 'are'), (1408357, 'from'), (1334359,
↪'Data'),
(1315444, 'have'), (1310093, 'Energy'), (1289975, 'Set'), (1281998, 'Technologies,'),
(1280088, '***********'), (1238125, '\t-'), (1176380, 'all'), (1169961, 'other'),␣
↪(1166151, 'its'),
(1132810, 'an'), (1127730, '&'), (1112331, '>'), (1111663, 'been'), (1098435, 'This'),
(1054291, '0\t0\t0\t0\t'), (1021797, 'States'), (971255, 'you'), (971180, 'which'),␣
↪(961102, '.'),
(945348, 'I'), (941903, 'it'), (939439, 'provide'), (902312, 'North'), (867218,
↪'Subject:'),
(851401, 'Party'), (845111, 'America'), (840747, 'Agreement'), (810554, '#N/A\t'),␣
↪(807259, 'may'),
(800753, 'please'), (798382, 'To'), (771784, '\t$-'), (753774, 'United'), (740472, 'if
↪'),
(739731, '\t0.00'), (723399, 'Power'), (699294, 'To:'), (697798, 'From:'), (672727,
↪'Date:'),
(661399, 'produced'), (652527, '2001'), (651164, 'format'), (650637, 'Email'),␣
↪(646922, '3.0'),
(645078, 'licensed'), (644200, 'License'), (642700, 'PST'), (641426, 'cite'), (640441,
↪ 'Creative'),
(640089, 'Commons'), (640066, 'NSF'), (639960, 'EML,'), (639949, 'Attribution'),
(639938, 'attribution,'), (639936, 'ZL'), (639936, '(http://www.zlti.com)."'),␣
↪(639936, '"ZL'),
(639936, 'X-ZLID:'), (639936, '<http://creativecommons.org/licenses/by/3.0/us/>'),␣
↪(639936, 'X-SDOC:')]
```

Troubleshooting

If something goes wrong consult the *FAQ / Known issues* page.

Further information

See the Spark and PySpark documentation pages for more information.

How to do Natural Language Processing

Overview

This example provides a simple PySpark job that utilizes the NLTK library. NLTK is a popular Python package for natural language processing. This example will demonstrate the installation of Python libraries on the cluster, the usage of Spark with the YARN resource manager and execution of the Spark job.

Who is this for?

This how-to is for users of a Spark cluster who wish to run Python code using the YARN resource manager. This how-to will show you how to integrate thrid-party Python libraries with Spark.

Spark NLTK summary

1. *Before you start*
2. *Install NLTK*
3. *Running the Job*
4. *Troubleshooting*
5. *Further information*

Before you start

To execute this example, download the: `cluster-spark-nltk.py example script` or `cluster-spark-nltk.ipynb example notebook`.

For this example you'll need Spark running with the YARN resource manager. You can install Spark and YARN using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

Install NLTK

If you have permission to install packages with `acluster` you can install NLTK as a conda package.

```
acluster conda install nltk
```

You should see output similar to this from each node, which indicates that the package was successfully installed across the cluster:

```
Node "ip-10-140-235-89.ec2.internal":
    Successful actions: 1/1
Node "ip-10-154-10-144.ec2.internal":
    Successful actions: 1/1
Node "ip-10-31-96-152.ec2.internal":
    Successful actions: 1/1
```

In order to use the full NLTK library, you will need to download the data for the NLTK project. You can download the data on all cluster nodes by using the `acluster cmd` command.

```
acluster cmd 'sudo /opt/anaconda/bin/python -m nltk.downloader -d /usr/share/nltk_
↪data all'
```

After a few minutes you should see output similar to this.

```
Execute command "sudo /opt/anaconda/bin/python -m nltk.downloader -d /usr/share/nltk_
↪data all" target: "*" cluster: "d"
    All nodes (x3) response: [nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /usr/share/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /usr/share/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /usr/share/nltk_data...

....

[nltk_data]    |   Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /usr/share/nltk_data...
[nltk_data]    |   Unzipping models/word2vec_sample.zip.
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

Running the Job

Here is the complete script to run the Spark + NLTK example in PySpark.

```
# cluster-spark-nltk.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-nltk')
sc = SparkContext(conf=conf)

data = sc.textFile('file:///usr/share/nltk_data/corpora/state_union/1972-Nixon.txt')

def word_tokenize(x):
    import nltk
    return nltk.word_tokenize(x)

def pos_tag(x):
```

(continues on next page)

(continued from previous page)

```
    import nltk
    return nltk.pos_tag([x])

words = data.flatMap(word_tokenize)
print words.take(10)

pos_word = words.map(pos_tag)
print pos_word.take(5)
```

Let's walk through the above code example. First, we will create a SparkContext. Note that Anaconda for cluster management will not create a SparkContext by default. In this example, we use the YARN resource manager.

```
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-nltk')
sc = SparkContext(conf=conf)
```

After a SparkContext is created, we can load some data into Spark. In this case, the data file is from one of the example documents provided by NLTK.

```
data = sc.textFile('file:///usr/share/nltk_data/corpora/state_union/1972-Nixon.txt')
```

Next, we write a function that imports `nltk` and calls `nltk.word_tokenize`. The function is mapped to the text file that was read in the previous step.

```
def word_tokenize(x):
    import nltk
    return nltk.word_tokenize(x)

words = data.flatMap(word_tokenize)
```

We can confirm that the `flatMap` operation worked by returning some of the words in the dataset.

```
print words.take(10)
```

Finally, NTLK's POS-tagger can be used to find the part of speech for each word.

```
def pos_tag(x):
    import nltk
    return nltk.pos_tag([x])

pos_word = words.map(pos_tag)
print pos_word.take(5)
```

Run the script on the Spark cluster using the spark-submit script. The output shows the words that were returned from the Spark script, including the results from the `flatMap` operation and the `POS-tagger`.

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
15/06/13 05:14:29 INFO SparkContext: Running Spark version 1.4.0

[...]

['Address',
```

(continues on next page)

(continued from previous page)

```
 'on',
 'the',
 'State',
 'of',
 'the',
 'Union',
 'Delivered',
 'Before',
 'a']

[...]

[[('Address', 'NN')],
 [('on', 'IN')],
 [('the', 'DT')],
 [('State', 'NNP')],
 [('of', 'IN')]]
```

Troubleshooting

If something goes wrong consult the *FAQ / Known issues* page.

Further information

See the Spark and PySpark documentation pages for more information.

For more information on NLTK see the *NLTK book <http://www.nltk.org/book/>*.

How to do Image Processing with GPUs

Overview

To demonstrate the capability of running a distributed job in PySpark using a GPU, this example uses NumbaPro and the CUDA platform for image analysis. This example executes 2-dimensional FFT convolution on images in grayscale and compares the execution time of CPU-based and GPU-based calculations.

Who is this for?

This how-to is for users of a Spark cluster who wish to run Python code using the YARN resource manager. This how-to will show you how to integrate third-party Python libraries with Spark.

Image processing summary

1. *Before you start*
2. *Install dependencies*
3. *Load data into HDFS*
4. *Running the Job*

5. *Troubleshooting*
6. *Further information*

Before you start

For this example, you'll need Spark running with the YARN resource manager. You can install Spark and YARN using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

You will also need valid Amazon Web Services (AWS) credentials in order to download the example data.

For this example, we recommend the use of the GPU-enabled AWS instance type `g2.2xlarge` and the AMI `ami-12fd8178` (us-east-1 region), which has CUDA 7.0 and the NVIDIA drivers pre-installed. An example profile (to be placed in `~/.acluster/profiles.d/gpu_profile.yaml`) is shown below:

```
name: gpu_profile
node_id: ami-12fd8178  # Ubuntu 14.04, Cuda 7.0, us-east-1 region
node_type: g2.2xlarge
num_nodes: 4
provider: aws_east
user: ubuntu
```

To execute this example, download the: `spark-numbapro.py example script` or `spark-numbapro.ipynb example notebook`.

If you wish to use the `spark-numbapro.ipynb example notebook` the Jupyter Notebook plugin can be installed on the cluster using the following command:

```
acluster install notebook
```

Once the Jupyter Notebook plugin is installed, you can view Jupyter Notebook in your browser using the following command:

```
acluster open notebook
```

Install dependencies

If you have permission to install packages with `acluster` you can install the required packages on all nodes using the following command.

```
acluster conda install scipy matplotlib numbapro PIL
```

Load data into HDFS

First, we will load the sample text data into the HDFS data store. The following script will transfer sample image data (approximately 1.1 GB) from a public Amazon S3 bucket to the HDFS data store on the cluster.

Download the `cluster-download-data.py script` to your cluster and insert your Amazon AWS credentials in the `AWS_KEY` and `AWS_SECRET` variables.

```
import subprocess

AWS_KEY = 'YOUR_AWS_KEY'
AWS_SECRET = 'YOUR_AWS_SECRET'
```

(continues on next page)

(continued from previous page)

```
s3_path = 's3n://{0}:{1}@blaze-data/dogs-cats-img/images'.format(AWS_KEY, AWS_SECRET)
cmd = ['hadoop', 'distcp', s3_path, 'hdfs:///tmp/dogs-cats']
subprocess.call(cmd)
```

Note: The `hadoop distcp` command might cause HDFS to fail on smaller instance sizes due to memory limits.

Run the `cluster-download-data.py` script on the cluster.

```
python cluster-download-data.py
```

After a few minutes, the image data will be in the HDFS data store on the cluster and ready for analysis.

Running the Job

Run the `spark-numbapro.py` script on the Spark cluster using spark-submit. The output shows the image processing execution times for the CPU-based vs. GPU-based calculations.

```
54.164.123.31: Using Spark's default log4j profile: org/apache/spark/log4j-defaults.
↪properties
15/11/09 02:33:21 INFO SparkContext: Running Spark version 1.5.1

[...]

15/11/09 02:33:45 INFO TaskSetManager: Finished task 6.0 in stage 1.0 (TID 13)
in 106 ms on ip-172-31-9-24.ec2.internal (7/7)
15/11/09 02:33:45 INFO YarnScheduler: Removed TaskSet 1.0, whose tasks have
all completed, from pool
15/11/09 02:33:45 INFO DAGScheduler: ResultStage 1
(collect at /tmp/anaconda-cluster/spark-numbapro.py:106) finished in 4.844 s
15/11/09 02:33:45 INFO DAGScheduler: Job 1 finished:
collect at /tmp/anaconda-cluster/spark-numbapro.py:106, took 4.854970 s

10 images
CPU: 6.91735601425
GPU: 4.88133311272

[...]

15/11/09 02:34:27 INFO TaskSetManager: Finished task 255.0 in stage 3.0 (TID 525)
 in 139 ms on ip-172-31-9-24.ec2.internal (256/256)
15/11/09 02:34:27 INFO YarnScheduler: Removed TaskSet 3.0, whose tasks have
all completed, from pool
15/11/09 02:34:27 INFO DAGScheduler: ResultStage 3
(collect at /tmp/anaconda-cluster/spark-numbapro.py:126) finished in 19.340 s
15/11/09 02:34:27 INFO DAGScheduler: Job 3 finished:
collect at /tmp/anaconda-cluster/spark-numbapro.py:126, took 19.400670 s

500 images
CPU: 22.1282501221
GPU: 19.8209779263
```

Troubleshooting

If something goes wrong consult the *FAQ / Known issues* page.

Further information

See the Spark and PySpark documentation pages for more information.

For more information on NumbaPro see the NumbaPro documentation.

Using Anaconda with Cloudera CDH

NOTE: This page is superseded, please see https://docs.continuum.io/anaconda-scale/cloudera-cdh

There are two methods of using Anaconda on an existing cluster with Cloudera CDH, Cloudera's Distribution Including Apache Hadoop: 1) the Anaconda parcel for Cloudera CDH, and 2) Anaconda for cluster management. The instructions below describe how to uninstall the Anaconda parcel on a CDH cluster and transition to Anaconda for cluster management.

Uninstalling the Anaconda parcel

If the Anaconda parcel is installed on the CDH cluster, use the following steps to uninstall the parcel. Otherwise, you can skip to the next section.

1. From the Cloudera Manager Admin Console, click the Parcels indicator in the top navigation bar.
2. Click the `Deactivate` button to the right of the Anaconda parcel listing.
3. Click `OK` on the Deactivate prompt to deactive the Anaconda parcel and restart Spark and related services.
4. Click the arrow to the right of the Anaconda parcel listing and choose `Remove From Hosts`, which will prompt with a confirmation dialog.
5. The Anaconda parcel has been removed from the cluster nodes.

For more information about managing Cloudera parcels, refer to the Cloudera documentation.

Using Anaconda for cluster management

Anaconda for cluster management provides additional functionality, including the ability to manage multiple conda environments and packages (including Python and R) alongside an existing CDH cluster.

1. Configure the nodes with Anaconda for cluster management using the Bare-metal Cluster Setup instructions.
2. During this process, you will create a profile and provider that describes the cluster.
3. Provision the cluster using the following command, replacing `cluster-cdh` with the name of your cluster and `profile-cdh` with the name of your profile:

```
$ acluster create cluster-cdh -p profile-cdh
```

4. You can submit Spark jobs along with the `PYSPARK_PYTHON` environment variable that refers to the location of Anaconda, for example:

```
$ PYSPARK_PYTHON=/opt/anaconda/bin/python spark-submit pyspark_script.py
```

Support

Priority support

Support is included with the purchase of an Anaconda subscription. Please contact Anaconda for cluster management support at ac_support@anaconda.com.

Report a bug

If you think you may have encountered a bug, please contact Anaconda for cluster management support at ac_support@anaconda.com.

Additional nodes

With the unlicensed, unsupported version of Anaconda for cluster management, you can create and manage up to four cluster nodes. To request a license to manage more cluster nodes, please contact your account representative or sales@anaconda.com.

Training and consulting

Training and consulting is available for Anaconda for cluster management and all Anaconda products. For more information, please contact your account representative or sales@anaconda.com.

FAQ / Known issues

Product FAQ

Does Anaconda for cluster management work with a cluster that already has a managed Spark/Hadoop stack?

Yes, Anaconda for cluster management can be installed alongside enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP and can be used to manage Python and R conda packages and environments across a cluster.

Does Anaconda for cluster management offer integration with Anaconda Repository or Anaconda Cloud?

Yes, conda packages can be installed from Anaconda Repository or Anaconda Cloud by using channel specifications. Refer to the documentation regarding *Installing packages from channels* for more details.

Which cloud providers does Anaconda for cluster management support?

Currently Anaconda for cluster management offers full support for Amazon Elastic Compute Cloud (EC2). Other providers such as Microsoft Azure, Google Cloud Platform, Rackspace, and others are on our roadmap. If you are interested in a cloud provider that is not listed here, please contact us at sales@anaconda.com.

Can I use Anaconda for cluster management with a different cloud provider?

Yes, you can manually create instances on another cloud provider, then provision the nodes as if you were using a bare-metal cluster. Refer to the *Bare-metal Cluster Setup* documentation for more information.

Does Anaconda for cluster management support Amazon EMR?

Anaconda for cluster management does not support Amazon Elastic MapReduce (EMR), which provides a managed Hadoop framework in the cloud. Anaconda for cluster management can be used to manage conda packages and environments across a cluster. Anaconda for cluster management does support Amazon EC2.

Technical FAQ

Which versions of Python does Anaconda for cluster management support?

Some of the plugins have dependencies that require the use of Python 2. Therefore, by default, Anaconda for cluster management installs Miniconda with Python 2 into the root conda environment.

However, you can easily create a new conda environment on the cluster with Python 3, or you can specify a different root version of Anaconda/Miniconda in the profile.

Which network ports need to be accessible from the client machine and cluster nodes?

From the client machine to the cluster nodes, you will need access to ports 22, 4505, and 4506 to provision the cluster via SSH and Salt. For communication between the cluster nodes, Salt uses ports 4505 and 4506.

Can I use Anaconda for cluster management with iptables and SElinux?

Yes, you can customize the security behavior of Anaconda for cluster management when creating or provisioning a cluster using the Security settings in the profile. Refer to the *Profile settings* documentation for more information.

Troubleshooting FAQ

Errors when creating or provisioning a cluster

Verify the following:

- The contents of your SSH private key are correct (and set to `600` permissions on Mac/Linux)
- The user name in your profile (e.g., `user: ubuntu`) is defined correctly
- Profile settings are defined correctly in `~/.acluster/profiles.d/<profile_name>.yaml`
- Provider settings are defined correctly in `~/.acluster/providers.yaml`

Known Issues

No matching SLS error (Salt)

When installing a plugin during cluster creation/provisioning or using the `acluster install` command, you might receive an error similar to:

```
============================= Standard output =============================

ip-10-144-206-102.ec2.internal:
- No matching sls found for 'cdh5.hdfs' in env 'base'

===========================================================================
```

(continues on next page)

(continued from previous page)

```

Fatal error: One or more hosts failed while executing task 'parallel_sudo'

Aborting.
One or more hosts failed while executing task 'parallel_sudo'
```

This is a known issue with Salt that periodically occurs. You can reprovision or reinstall the plugin that failed, and the installation should succeed.

Theading error when creating a cluster

When creating/provisioning a cluster, you might receive an error similar to:

```
Uploading formulas
INFO: Uploading formulas to head
Syncing formulas
INFO: Syncing formulas across the cluster
Done
Exception in thread Thread-6 (most likely raised during interpreter shutdown):
Exception in thread Thread-8 (most likely raised during interpreter shutdown):
```

This is a known issue with Paramiko/Fabric. You can safely disregard this message. This issue was resolved in Anaconda for cluster management 1.3.1.

Release notes

Below is a summary of features, bug fixes, and improvements for each Anaconda for cluster management release.

Anaconda for cluster management 1.4.4

Note: This is the last feature release of Anaconda for cluster management. Anaconda for cluster management has been replaced by Anaconda Scale.

Backend improvements:

- Use full paths for system commands

Anaconda for cluster management 1.4.3

Backend improvements:

- Fix license check and support information

Anaconda for cluster management 1.4.2

Features:

- Updates to examples in documentation

Backend improvements:

- Improved group configuration/detection on RHEL/CentOS
- Enabled configuration of `anaconda` group name

Anaconda for cluster management 1.4.1

Features:

- Renamed `distributed` plugin to `dask`
- Enabled `notebook` and `dask` plugins to be installed alongside existing Hadoop installations

- Added conda profile plugin setting for ssl_verify

Anaconda for cluster management 1.4.0

Features:

- Added setting for channel_alias for improved integration with Anaconda Repository

Bug Fixes:

- Fixed provisioning on Windows client machines
- Fixed issue with profile setting for conda access control lists
- Fixed issue with certain plugin status checks
- Fixed issue with Storm plugin installation

Backend improvements:

- Improved management and configurability of distributed plugin
- Updated version of Apache Libcloud dependency (0.20.1)
- Updated to accept new license format (backwards compatible with existing licenses)
- Removed deprecated acluster submit command due to inconsistencies with environments
- Removed acluster defaultenv and acluster setenv commands due to inconsistencies with environments
- Consolidated sample profile to a single profile

Anaconda for cluster management 1.3.3

Bug Fixes:

- Fixed SSH private key permissions check

Backend improvements:

- Implemented Salt ACL functionality
- Implemented Salt PAM authentication
- Removed sudo requirement after install

Anaconda for cluster management 1.3.2

Bug Fixes:

- Fixed installation prefix and default channel documentation and tests

Backend improvements:

- Improved detection of plugin installations and plugin status messages

Anaconda for cluster management 1.3.1

Features:

- Updated plugin: Spark 1.6
- Added option to disable creation of /etc/profile.d/conda.sh on cluster nodes

Bug Fixes:

- Fixed R path for Jupyter Notebook
- Fixed issue with Amazon EC2 block device mapping

- Fixed issue with configuration of default conda channels
- Improved warnings for misconfigured SSH keys
- Improved error message when AWS Key Pair is missing

Backend improvements:

- Added configurable download URLs for plugins (used for airgap installations): Elasticsearch, Logstash, Kibana, Storm
- Added deprecated warning for `acluster submit` command due to inconsistencies with environments
- Added detection for enterprise Hadoop installations and warns before installing conflicting services
- Configured HDFS for short-circuit reads by default
- Pinned versions of dependencies in conda recipe

Anaconda for cluster management 1.3.0

Features:

- Simplified bare-metal cluster setup using profile/provider instead of clusters.d
- Added ability to specify conda packages and environments in profile plugin settings
- Switched to system-wide .condarc file for users on cluster nodes
- Added centralized logging via Elasticsearch, Logstash, and Kibana plugins
- Added `--no-browser` option for `acluster open` command
- Added license information to `acluster info`
- Removed optional `security_group` setting from `providers.yaml` template and documentation
- Added `vpc_id` setting for Amazon EC2 to use with default security group
- Improved documentation for bare-metal and cloud-based cluster creation
- Added more information about requirements, FAQs, and known issues to documentation

Bug Fixes:

- Improved error messages for cloud providers, security groups, and when no clusters are defined
- Fixed error when command output contained Unicode characters
- Fixed repeated prompts when installing multiple plugins

Backend improvements:

- Added test for SSH connectivity during bare-metal cluster creation
- Added detection/removal of orphaned clusters upon `acluster destroy`
- Added detection/installation of missing sudo/bzip2 packages upon cluster creation

Anaconda for cluster management 1.2.2

Features:

- Moved bare metal cluster setup to profile and provider files
- Ability to specify private IPs for VPC clusters on Amazon EC2

Anaconda for cluster management 1.2.1

Features:

- Updated plugin: Spark 1.5.1
- Consistent output when listing clusters and profiles
- New blog post and updated documentation

Bug Fixes:

- Removed duplicate error messages
- Fixed notebook plugin formulas

Anaconda for cluster management 1.2.0

Features:

- Dedicated salt/supervisor environments
- New cluster profile (no nesting)
- Documentation updates
- Improvements to CLI output

Bug Fixes:

- Storm plugin
- Notebook plugin
- Logstash plugin
- Kibana plugin
- Elasticsearch plugin
- Windows Fixes: push/submit

Backend improvements:

- Refactor status/stop/restart and plugin loading
- Refactor connection object
- Multiple backends (initial work)

Anaconda for cluster management 1.1.0

- Improved Jenkins testing
- Added Storm plugin
- Added Elasticsearch/Logstash/Kibana plugins (ELK stack)
- Added Dask plugin
- Fixes for single-node installations
- Profile updates
- Add setting for AWS Tags
- Added security options (SELinux/iptables)
- Added setting for notebook password

Anaconda for cluster management 1.0.0

- CLI rewrite
- Aggregated and improved CLI messaging

- Backend library refactoring
- Salt-packaging refactoring
- Easier bare-metal installation
- Added option for streaming output
- Fixed issue with Ganglia on Ubuntu
- Windows/libcloud Compatibility
- Added licensing with new CLI
- Conda remote fixes
- Install Anaconda/Miniconda from custom url

Glossary

Anaconda A downloadable free, open source, high performance, optimized Python and R distribution with 100+ packages plus access to easily installing an additional 620+ popular open source packages for data science including advanced and scientific analytics. It also includes `conda`, an open source package, dependency and environment manager. Thousands more open source packages can be installed with the `conda` command. Available for Windows, OS X and Linux, all versions are supported by the community.

Bare-metal cluster On-site or in-house machines, collections of virtual machines (Vagrant, Docker, etc.), or previously instantiated cloud nodes.

Cloud-based cluster A cluster that consists of machines in a cloud provider such as Amazon EC2.

Cluster A group of computers that work in parallel to perform a single task. Also called “parallel computing” since the compute nodes can perform their operations in parallel.

Client machine The laptop or workstation that contains Anaconda for cluster management and manages the cluster nodes.

Cluster file The file that defines the configuration including the location of the head node, compute nodes, and authentication/configuration information.

Compute node The machines managed by the head node that all work together to complete a single task.

Conda Conda is an open source package management system and environment management system for installing multiple versions of software packages and their dependencies and switching easily between them. It works on Windows, Mac, and Linux. It was created for Python packages, but is able to package and distribute any software. Conda is included in all versions of Anaconda, Anaconda Server, and Miniconda.

Head node A system configured to act as the intermediary between the cluster and the outside network. Can also be referred to as the master or edge node.

Miniconda A minimal or “bootstrap” version of Anaconda. Installs only what you need to get conda running, including Python, conda, and its dependencies.

PEM key Privacy enhanced electronic mail file, originally for email and now a general file format for cryptographic keys. In Anaconda for cluster management, this is used for cloud-based clusters and can be obtained from the cloud provider.

Plugin In Anaconda for cluster management, plugins are analytics engines and management services that can be installed on a cluster.

Profile A configuration file that defines how a cluster should be configured. It contains information about the number and types of cluster nodes, plugins, and other settings.

Provider Configuration file that defines settings for cloud or bare metal providers. A provider is referenced by a profile and used to provision resources.

1.4.4 Anaconda Launcher

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it.

Anaconda Launcher is now deprecated

As of Anaconda 4.0, Anaconda Launcher has been replaced by *Anaconda Navigator*. This section has been left in place as a reference for users of earlier versions of Anaconda. Some users who have upgraded from an earlier version may have both Navigator and Launcher installed, which is harmless.

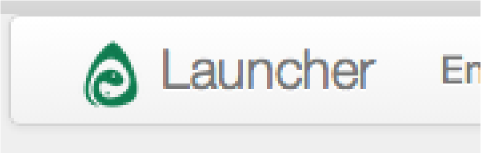
Launcher can be uninstalled with a single command:

```
conda remove launcher
```

Anaconda Launcher commands

Refresh Application List

Click on **Launcher** to refresh the apps display.



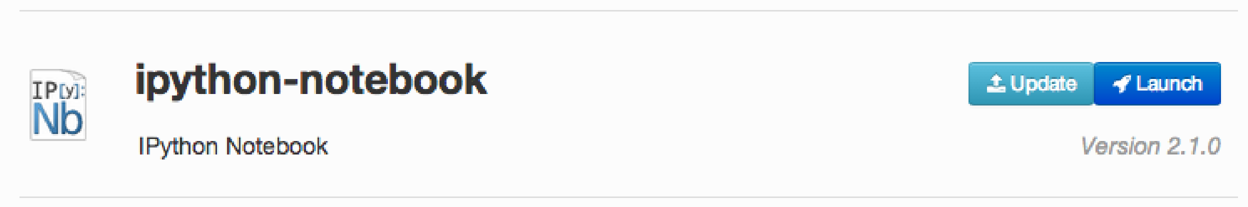
Install Listed Application

Click on the green **Install** button to install the desired app.

img/appInstall.png

Launch Application

Click on the **Launch** button to launch the application.

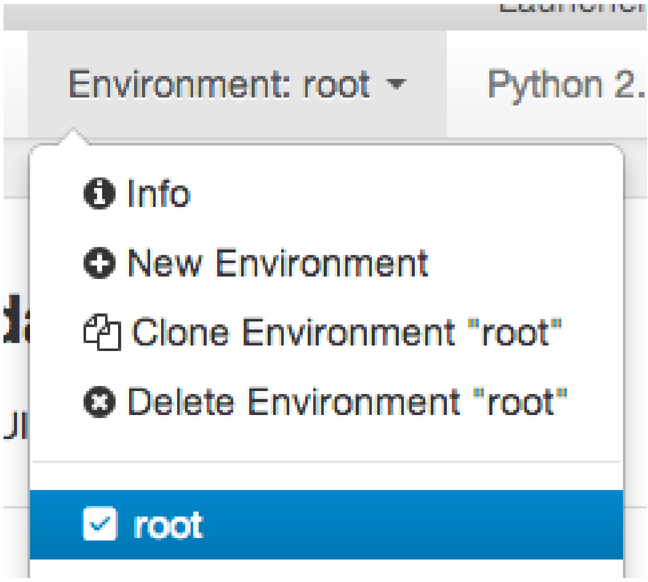


Update Application

Click on the **Update** button to update the application.

Change Environments

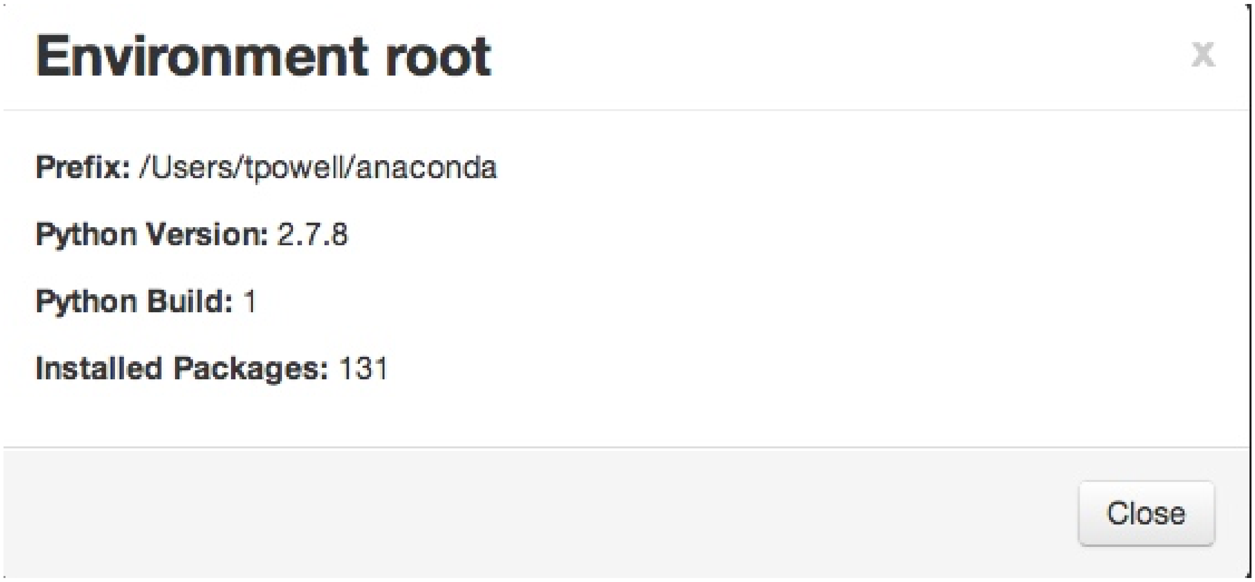
To change the current Environment, click on the **Environment:** pull down menu item and select the desired environment to make it active.



Environment Information

To get a summary of the current environment, click on the **Environment:** menu item and select info.

You can also get a summary of the environment by click on the **Python version** menu item.


Environment root

Prefix: /Users/tpowell/anaconda

Python Version: 2.7.8

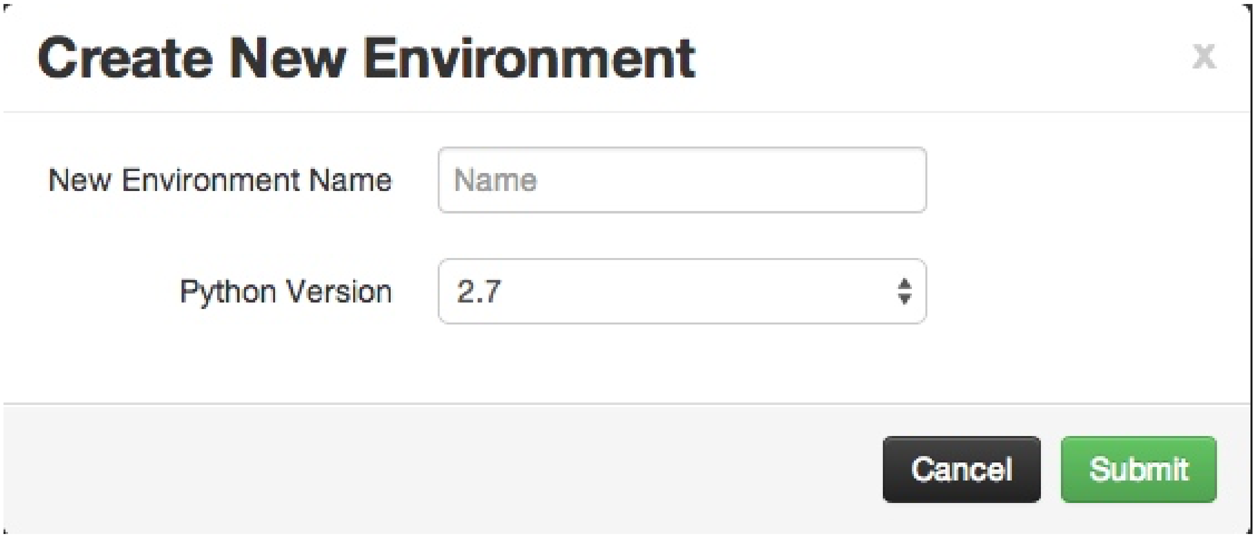
Python Build: 1

Installed Packages: 131

Create New Environment

To create a new environment, click on the **Environment:** menu item and select **New Environment**.

When the **Create New Environment** dialog appears, enter the name of the new environment, select the desired **Python Version**, and click **Submit**.


Create New Environment

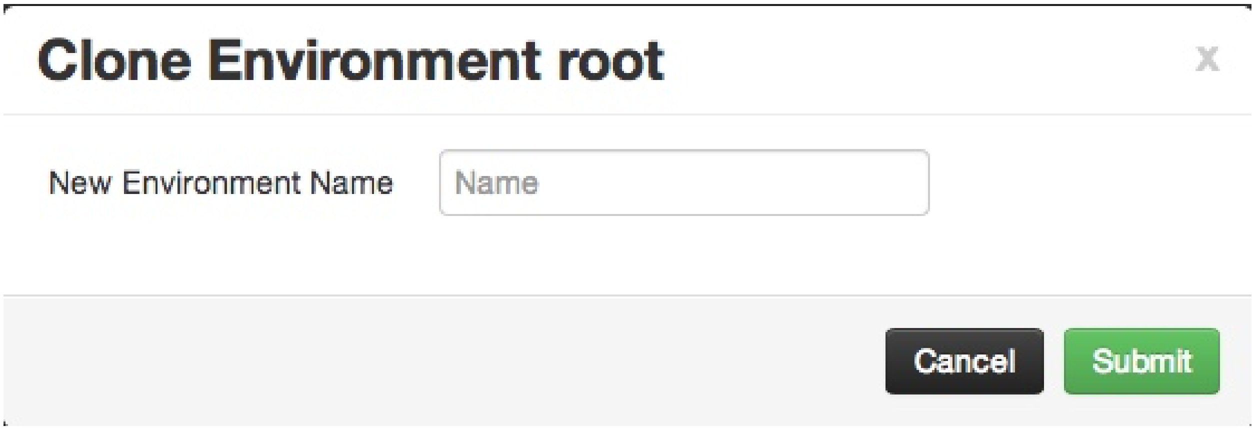
New Environment Name: Name

Python Version: 2.7

Clone Environment

To clone an environment, click on the **Environment:** menu item and select **Clone Environment**.

When the **Clone Environment** dialog appears, enter the name of the new environment and click **Submit**.



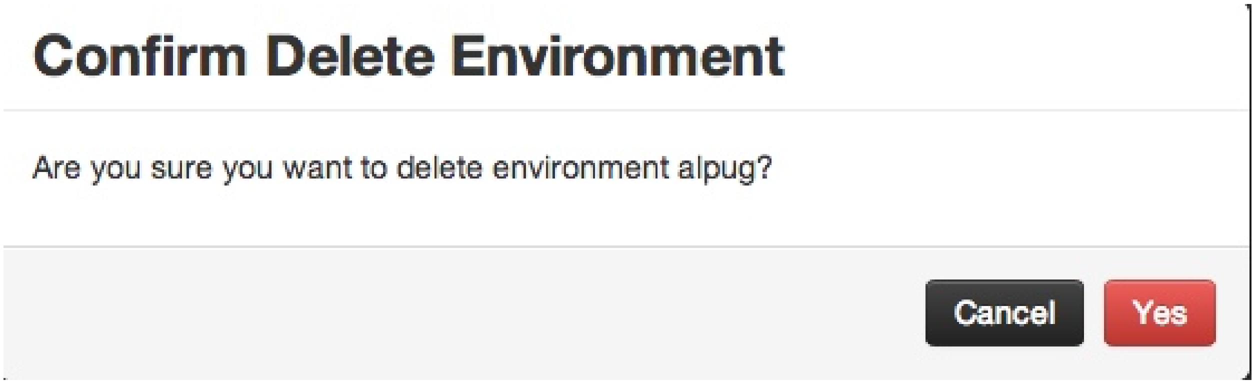
Clone Environment root

New Environment Name

Delete Environment

To delete an environment, click on the **Environment:** menu item and select **Delete Environment**.

When the **Confirm Delete Environment** dialog appears, click **Yes** to delete the environment.

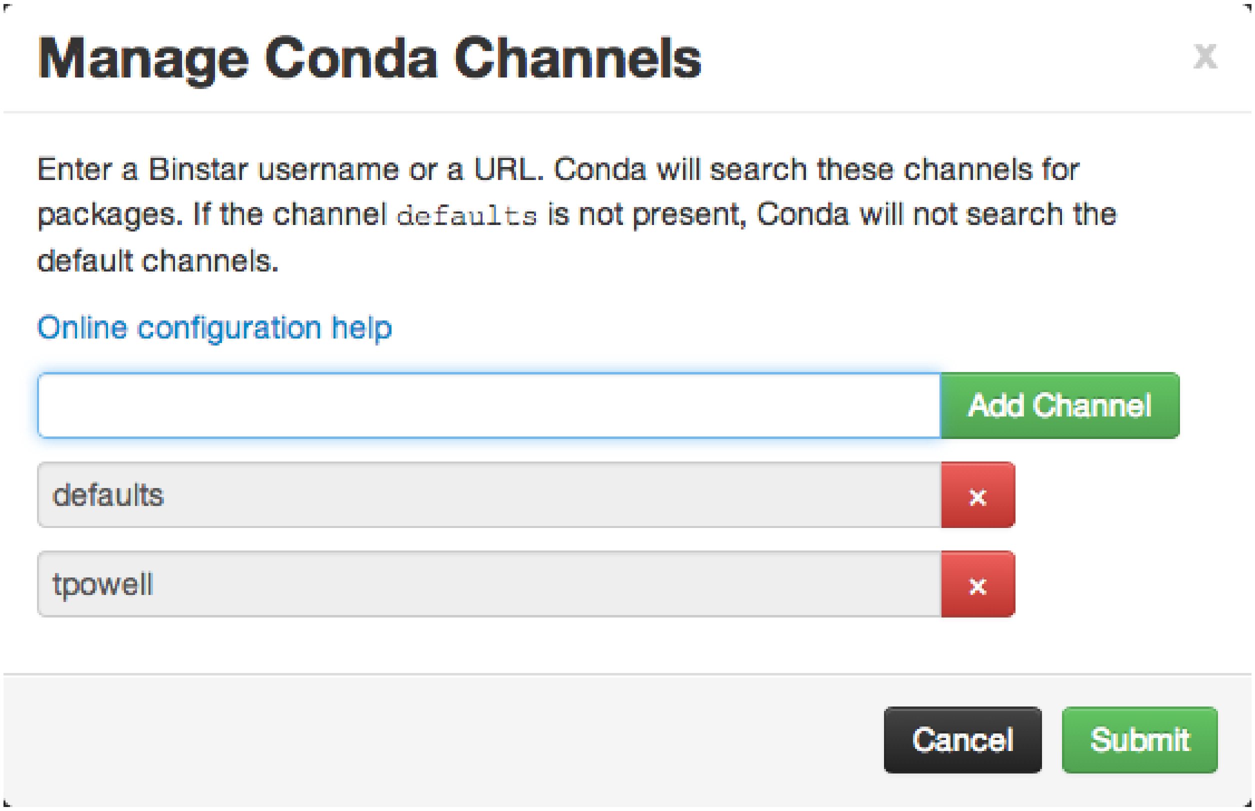


Confirm Delete Environment

Are you sure you want to delete environment alpug?

Manage Channels

To manage your current package channels, click on the **Manage Channels** menu item.



Manage Conda Channels

Enter a Binstar username or a URL. Conda will search these channels for packages. If the channel `defaults` is not present, Conda will not search the default channels.

Online configuration help

defaults

tpowell

Add Channel

Enter your channel into the text field to the left of the **Aadd Channel** button.

Click on the **Add Channel** to add your channel to your .condarc file.

Remove Channel

Click on the **x** button to the right of the channel name you would like to remove.

Click **Cancel** to removed the changes to your channels.

Click **Submit** to save the channel updates.

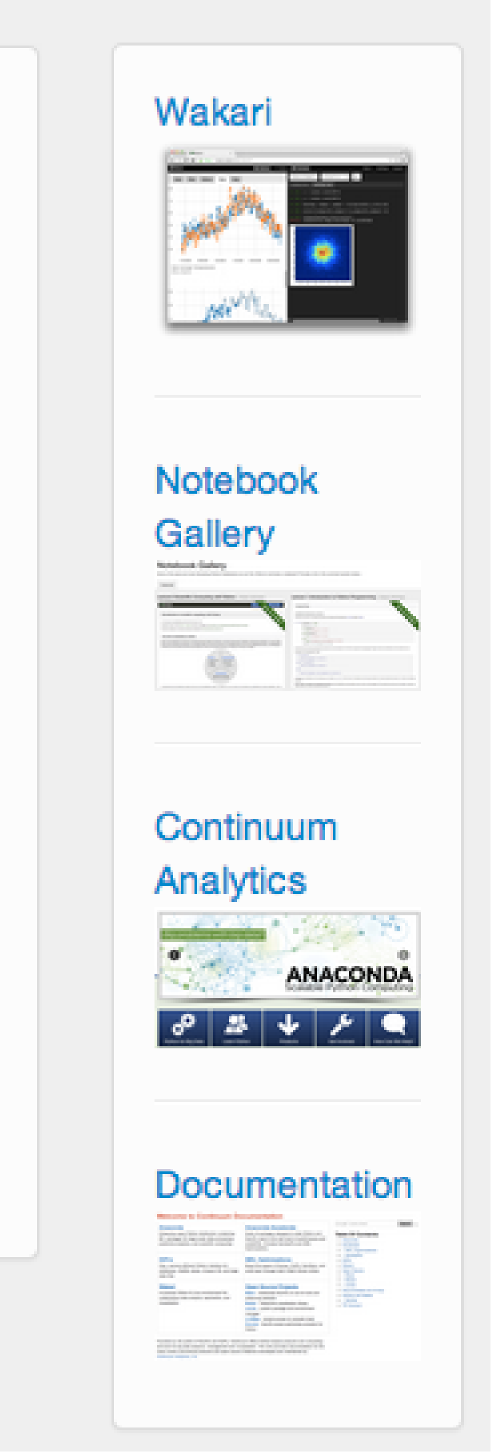
Anaconda Launcher Info

For Anaconda Launcher details, click on the version number in the upper right hand corner of the Anaconda Launcher Window.

img/launcherInfo.png

Other Helpful Links

Along the right side of the launcher are some helpful links to additional documentation and various Anaconda and Anaconda Sponsored websites.



1.4.5 Anaconda Scale

Distributed Computing

Note: This product is discontinued. This archived copy of the product documentation is provided for those customers who are still using it.

Anaconda Scale provides functionality to easily deploy Anaconda® packages and a distributed computation framework across a cluster. It helps you manage multiple conda environments and packages–including Python and R–on cluster nodes.

Anaconda Scale provides different options for deploying Anaconda on a cluster, including:

- Centrally managed installation of Anaconda, including multiple environments such as Python and R
- Anaconda parcel for Cloudera CDH, including custom-generated parcels
- Deployment of conda packages and environments with Spark jobs

Features

- Easily install Anaconda–including *Anaconda Accelerate*–across multiple cluster nodes
- Provision distributed compute services with Dask
- Perform interactive, distributed computations with single-user Jupyter Notebook
- Easily launch and configure a cloud-based cluster on Amazon EC2

Compatibility

Anaconda Scale can be used with distributed computation frameworks such as Spark or Dask and works alongside enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP. Anaconda Scale has been tested with the following Hadoop distributions and Spark versions:

- Cloudera CDH 5.3.x through 5.11.x
- Hortonworks HDP 2.2.x through 2.6.x (with Apache Ambari 2.2.x and 2.4.x)
- Spark 1.3.x through 2.0.x

License

Anaconda Scale is available with Anaconda Enterprise. If you would like to use Anaconda Scale with a cluster on a bare-metal, on-premises, or cloud-based cluster, please contact us.

Using Scale

Installation

Anaconda Scale is installed and managed using the Anaconda platform installer, Anaconda Adam. For more information on installation, security, account, network and other Anaconda Scale requirements, see the *system requirements for Anaconda Adam*. For more information about the installation process, see *the Anaconda Adam installation instructions*.

After installing Anaconda Adam and provisioning a cluster, you can install a centrally managed version of Anaconda–including *Anaconda Accelerate*–on all of the cluster nodes using:

```
$ adam scale -n <cluster-name> cluster install
```

To install Dask and Distributed and then start Distributed workers on all of the cluster nodes, use:

```
$ adam scale -n <cluster-name> dask install
```

To install a single-user Jupyter Notebook server on the head node of the cluster, use:

```
$ adam scale -n <cluster-name> notebook install
```

User guide

Using Anaconda Scale with Anaconda Adam, you can create and manage conda packages, including Python and R, in multiple conda environments across your nodes. You can prepend `adam scale` to a subset of familiar conda commands to install and manage conda packages and environments across a cluster.

```
$ adam scale -n cluster conda --help
Usage: adam conda [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help          Show this message and exit.

Commands:
  create   Create conda environment
  info     Get information about conda installation
  install  Install package(s)
  list     List conda packages
  remove   Remove package(s)
  update   Update package(s)
```

By default, the root conda environment is installed in `/opt/continuum/anaconda` on all of the cluster nodes, which can be configured for a different location. You can manage multiple conda environments across the same cluster, such as Python 2, Python 3 and R, without affecting the framework installation of Python on the cluster nodes.

Getting started

You can get started with Anaconda Scale by using Anaconda Adam, the Anaconda platform installer. See the *Anaconda Adam documentation* for more information.

With Anaconda Adam, you can install Anaconda Scale on a bare-metal cluster or existing cloud-based cluster. You can also use Anaconda Scale to easily launch and provision a cloud-based cluster with Anaconda. See the *Anaconda Adam installation instructions* for more information.

Tasks

- *General functions*
- *Using Anaconda with Cloudera CDH*
- *Configuring Anaconda with Spark*

Cheatsheet

Download the `Scale cheat sheet` (48 KB PDF) for a quick guide to using Anaconda Scale.

General functions

Installing conda packages

Install conda packages on the cluster nodes using `adam scale conda install`.

EXAMPLE: To install NumPy and pandas and their dependencies into the root conda environment on a cluster named `cluster`:

```
$ adam scale -n cluster conda install numpy pandas
All nodes (x4) response:
{
  "actions": {
    "LINK": [
      "mkl-11.3.3-0 /opt/continuum/anaconda/pkgs 1",
      "openssl-1.0.2h-1 /opt/continuum/anaconda/pkgs 1",
      "sqlite-3.13.0-0 /opt/continuum/anaconda/pkgs 1",
      "zlib-1.2.8-3 /opt/continuum/anaconda/pkgs 1",
      "python-2.7.11-5 /opt/continuum/anaconda/pkgs 1",
      "conda-env-2.5.1-py27_0 /opt/continuum/anaconda/pkgs 1",
      "numpy-1.11.0-py27_2 /opt/continuum/anaconda/pkgs 1",
      "pycosat-0.6.1-py27_1 /opt/continuum/anaconda/pkgs 1",
      "pytz-2016.4-py27_0 /opt/continuum/anaconda/pkgs 1",
      "pyyaml-3.11-py27_4 /opt/continuum/anaconda/pkgs 1",
      "requests-2.10.0-py27_0 /opt/continuum/anaconda/pkgs 1",
      "ruamel_yaml-0.11.7-py27_0 /opt/continuum/anaconda/pkgs 1",
      "setuptools-23.0.0-py27_0 /opt/continuum/anaconda/pkgs 1",
      "six-1.10.0-py27_0 /opt/continuum/anaconda/pkgs 1",
      "wheel-0.29.0-py27_0 /opt/continuum/anaconda/pkgs 1",
      "conda-4.1.4-py27_0 /opt/continuum/anaconda/pkgs 1",
      "pip-8.1.2-py27_0 /opt/continuum/anaconda/pkgs 1",
      "python-dateutil-2.5.3-py27_0 /opt/continuum/anaconda/pkgs 1",
      "pandas-0.18.1-np111py27_0 /opt/continuum/anaconda/pkgs 1"
      ],
      "PREFIX": "/opt/continuum/anaconda",
    },
    "success": true
  }
```

Creating conda environments

Create multiple conda environments on the cluster nodes using `adam scale conda create`.

EXAMPLE: To create a conda environment called `py3` with Python 3 on a cluster named `cluster`:

```
$ adam scale -n cluster conda create -n py3 python=3
All nodes (x4) response:
{
  "actions": {
    "LINK": [
```

(continues on next page)

(continued from previous page)

```
      "openssl-1.0.2h-1 1 None",
      "readline-6.2-2 1 None",
      "sqlite-3.13.0-0 1 None",
      "tk-8.5.18-0 1 None",
      "xz-5.2.2-0 1 None",
      "zlib-1.2.8-3 1 None",
      "python-3.5.1-5 1 None",
      "setuptools-23.0.0-py35_0 1 None",
      "wheel-0.29.0-py35_0 1 None",
      "pip-8.1.2-py35_0 1 None"
    ],
    "PREFIX": "/opt/continuum/anaconda/envs/py3",
    "SYMLINK_CONDA": [
      "/opt/continuum/anaconda"
    ],
  },
  "success": true
}
```

Updating conda packages

Update conda packages on the cluster nodes using `adam scale conda update`.

EXAMPLE: To update pandas and its dependencies in the root conda environment on a cluster named `cluster`:

```
$ adam scale -n cluster conda update pandas
All nodes (x4) response:
{
  "actions": {
    "LINK": [
      "pandas-0.18.1-np111py27_0 1 None"
    ],
    "PREFIX": "/opt/continuum/anaconda",
    "SYMLINK_CONDA": [
      "/opt/continuum/anaconda"
    ],
    "UNLINK": [
      "pandas-0.18.0-np111py27_0"
    ],
  },
  "success": true
}
```

Removing conda packages or environments

Remove conda packages on the cluster nodes using `adam scale conda remove`.

EXAMPLE: To remove pandas from the root conda environment on a cluster named `cluster`:

```
$ adam scale -n cluster conda remove pandas
All nodes (x4) response:
{
  "actions": {
    "PREFIX": "/opt/continuum/anaconda",
```

(continues on next page)

(continued from previous page)

```
    "UNLINK": [
      "pandas-0.18.1-np111py27_0"
    ],
  },
  "success": true
}
```

Remove conda environments on the cluster nodes using `adam scale conda remove`.

EXAMPLE: To remove an environment called `py3` on a cluster named `cluster`:

```
$ adam scale -n cluster conda remove -n py3 --all
All nodes (x4) response:
{
  "actions": {
    "PREFIX": "/opt/continuum/anaconda/envs/py3",
    "UNLINK": [
      "openssl-1.0.2j-0",
      "pip-8.1.2-py35_0",
      "python-3.5.2-0",
      "readline-6.2-2",
      "setuptools-27.2.0-py35_0",
      "sqlite-3.13.0-0",
      "tk-8.5.18-0",
      "wheel-0.29.0-py35_0",
      "xz-5.2.2-0",
      "zlib-1.2.8-3"
    ]
  },
  "success": true
}
```

Listing conda packages

List conda packages on the cluster nodes using `adam scale conda list`.

EXAMPLE: To list conda packages in the root conda environment on a cluster named `cluster`:

```
$ adam scale -n cluster conda list
All nodes (x4) response:
+------------------+---------+--------+
| Package          | Version | Build  |
+------------------+---------+--------+
| numpy            | 1.11.0  | py27_2 |
| conda-env        | 2.5.1   | py27_0 |
| tk               | 8.5.18  | 0      |
| pip              | 8.1.2   | py27_0 |
| readline         | 6.2     | 2      |
| setuptools       | 23.0.0  | py27_0 |
| openssl          | 1.0.2h  | 1      |
| pycosat          | 0.6.1   | py27_1 |
| python-dateutil  | 2.5.3   | py27_0 |
| sqlite           | 3.13.0  | 0      |
| pytz             | 2016.4  | py27_0 |
| mkl              | 11.3.3  | 0      |
```

(continues on next page)

(continued from previous page)

```
| yaml             | 0.1.6   | 0      |
| ruamel_yaml      | 0.11.7  | py27_0 |
| zlib             | 1.2.8   | 3      |
| pyyaml           | 3.11    | py27_4 |
| requests         | 2.10.0  | py27_0 |
| wheel            | 0.29.0  | py27_0 |
| python           | 2.7.11  | 5      |
| conda            | 4.1.4   | py27_0 |
| pycrypto         | 2.6.1   | py27_0 |
| six              | 1.10.0  | py27_0 |
+------------------+---------+--------+
```

Getting conda information

View information about the conda installation on the cluster nodes using `adam scale conda info`.

EXAMPLE: To view information about the root conda environment on a cluster named `cluster`:

```
$ adam scale -n cluster conda info
All nodes (x4) response:
Current conda install:

             platform : linux-64
        conda version : 4.1.4
    conda-env version : 2.5.1
  conda-build version : not installed
       python version : 2.7.11.final.0
     requests version : 2.10.0
     root environment : /opt/continuum/anaconda  (writable)
  default environment : /opt/continuum/anaconda
     envs directories : /opt/continuum/anaconda/envs
        package cache : /opt/continuum/anaconda/pkgs
         channel URLs : https://repo.continuum.io/pkgs/free/linux-64/
                        https://repo.continuum.io/pkgs/free/noarch/
                        https://repo.continuum.io/pkgs/pro/linux-64/
                        https://repo.continuum.io/pkgs/pro/noarch/
          config file : None
         offline mode : False
    is foreign system : False
```

Or to view a list of conda environments on the cluster nodes:

```
$ adam scale -n cluster conda info -e
All nodes (x4) response:
# conda environments:
#
py27                     /opt/continuum/anaconda/envs/py27
py35                     /opt/continuum/anaconda/envs/py35
r                        /opt/continuum/anaconda/envs/r
root                  *  /opt/continuum/anaconda
```

Managing conda with environment definition files

Manage conda environments on the cluster nodes using `adam scale env`:

```
$ adam scale -n cluster env
Usage: adam scale env [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  create  Create conda environment
  list    List conda environments
  remove  Remove conda environment
  update  Update conda environment
```

This functionality uses conda-env and conda environment environment definition–YAML–files.

EXAMPLE: The following `environment.yml` file describes a conda environment named `my-env` with Python 2.7 and additional packages:

```
name: my-env
channels:
- defaults
dependencies:
- python=2.7.*
- bokeh
- dask
- distributed
- pandas
```

Use `adam scale env create` to create a conda environment based on the above conda environment definition file `environment.yml`:

```
$ adam scale -n cluster env create -f environment.yml
All nodes (x4) response:
{
  "actions": {
    "EXTRACT": [
      "msgpack-python-0.4.8-py27_0",
      "numpy-1.11.2-py27_0",
      "psutil-5.0.0-py27_0",
      "pytz-2016.7-py27_0",
      "tblib-1.3.0-py27_0",
      "python-dateutil-2.6.0-py27_0",
      "zict-0.1.0-py27_0",
      "pandas-0.19.1-np111py27_0",
      "tornado-4.4.2-py27_0",
      "bokeh-0.12.3-py27_0",
      "dask-0.12.0-py27_0",
      "distributed-1.14.1-py27_0"
    ],
    "FETCH": [
      "msgpack-python-0.4.8-py27_0",
      "numpy-1.11.2-py27_0",
      "psutil-5.0.0-py27_0",
      "pytz-2016.7-py27_0",
      "tblib-1.3.0-py27_0",
      "python-dateutil-2.6.0-py27_0",
      "zict-0.1.0-py27_0",
      "pandas-0.19.1-np111py27_0",
```

(continues on next page)

(continued from previous page)

```
      "tornado-4.4.2-py27_0",
      "bokeh-0.12.3-py27_0",
      "dask-0.12.0-py27_0",
      "distributed-1.14.1-py27_0"
    ],
    "LINK": [
      "mkl-11.3.3-0 1",
      "openssl-1.0.2j-0 1",
      "readline-6.2-2 1",
      "sqlite-3.13.0-0 1",
      "tk-8.5.18-0 1",
      "yaml-0.1.6-0 1",
      "zlib-1.2.8-3 1",
      "python-2.7.12-1 1",
      "backports-1.0-py27_0 1",
      "backports_abc-0.4-py27_0 1",
      "click-6.6-py27_0 1",
      "cloudpickle-0.2.1-py27_0 1",
      "futures-3.0.5-py27_0 1",
      "heapdict-1.0.0-py27_1 1",
      "locket-0.2.0-py27_1 1",
      "markupsafe-0.23-py27_2 1",
      "msgpack-python-0.4.8-py27_0 1",
      "numpy-1.11.2-py27_0 1",
      "psutil-5.0.0-py27_0 1",
      "pytz-2016.7-py27_0 1",
      "pyyaml-3.12-py27_0 1",
      "requests-2.11.1-py27_0 1",
      "setuptools-27.2.0-py27_0 1",
      "six-1.10.0-py27_0 1",
      "tblib-1.3.0-py27_0 1",
      "toolz-0.8.0-py27_0 1",
      "wheel-0.29.0-py27_0 1",
      "chest-0.2.3-py27_0 1",
      "jinja2-2.8-py27_1 1",
      "partd-0.3.6-py27_0 1",
      "pip-9.0.1-py27_0 1",
      "python-dateutil-2.6.0-py27_0 1",
      "singledispatch-3.4.0.3-py27_0 1",
      "ssl_match_hostname-3.4.0.2-py27_1 1",
      "zict-0.1.0-py27_0 1",
      "pandas-0.19.1-np111py27_0 1",
      "tornado-4.4.2-py27_0 1",
      "bokeh-0.12.3-py27_0 1",
      "dask-0.12.0-py27_0 1",
      "distributed-1.14.1-py27_0 1"
    ],
    "PREFIX": "/opt/continuum/anaconda/envs/my-env",
    "SYMLINK_CONDA": [
      "/opt/continuum/anaconda"
    ],
    "op_order": [
      "RM_FETCHED",
      "FETCH",
      "RM_EXTRACTED",
      "EXTRACT",
      "UNLINK",
```

(continues on next page)

(continued from previous page)

```
      "LINK",
      "SYMLINK_CONDA"
    ]
  },
  "success": true
}
```

NOTE: The command `adam scale env create` is compatible only with conda dependencies and ignores any pip or other non-conda dependencies that are specified in the conda environment definition file.

Use `adam scale env list` to list conda environments on the cluster:

```
$ adam scale -n cluster env list
All nodes (x4) response:
# conda environments:
#
my-env                   /opt/continuum/anaconda/envs/my-env
root                  *  /opt/continuum/anaconda
```

Use `adam scale env update` to update packages in an existing conda environment based on a conda environment definition file:

```
$ adam scale -n cluster env update -f environment.yml
    All nodes (x4) response:
    {
      "message": "All requested packages already installed.",
      "success": true
    }
```

Use `adam scale env remove` to remove a conda environment from the cluster:

```
$ adam scale -n cluster env remove -n my-env
All nodes (x4) response:
{
  "actions": {
    "PREFIX": "/opt/continuum/anaconda/envs/my-env",
    "UNLINK": [
      "mkl-11.3.3-0",
      "numpy-1.11.2-py27_0",
      "openssl-1.0.2j-0",
      "pandas-0.18.0-np111py27_0",
      "pip-9.0.1-py27_0",
      "python-2.7.12-1",
      "python-dateutil-2.6.0-py27_0",
      "pytz-2016.7-py27_0",
      "readline-6.2-2",
      "setuptools-27.2.0-py27_0",
      "six-1.10.0-py27_0",
      "sqlite-3.13.0-0",
      "tk-8.5.18-0",
      "wheel-0.29.0-py27_0",
      "zlib-1.2.8-3"
    ]
  },
  "success": true
}
```

Using Anaconda with Cloudera CDH

There are different methods of using Anaconda Scale on a cluster with Cloudera CDH:

- The freely available *Anaconda parcel for Cloudera CDH*.
- Custom Anaconda parcels for Cloudera CDH
- A dynamic, managed version of Anaconda on all of the nodes using Anaconda Scale

The freely available Anaconda parcel is based on Python 2.7 and includes the default conda packages that are available in the free Anaconda distribution.

Anaconda Enterprise users can also leverage Anaconda Repository to *create and distribute their own custom Anaconda parcels for Cloudera Manager*.

If you need more dynamic functionality than the Anaconda parcels offer, Anaconda Scale lets you dynamically install and manage multiple conda environments–such as Python 2, Python 3, and R environments–and packages across a cluster.

Using the Anaconda parcel

For more information about installing the Anaconda parcel on a CDH cluster using Cloudera Manager, see the *Anaconda parcel documentation*.

Transitioning to the dynamic, managed version of Anaconda Scale

To transition from the Anaconda parcel for CDH to the dynamic, managed version of Anaconda Scale, follow the instructions below to uninstall the Anaconda parcel on a CDH cluster and then transition to a centrally managed version of Anaconda.

Uninstalling the Anaconda parcel

If the Anaconda parcel is installed on the CDH cluster, uninstall the parcel:

1. From the Cloudera Manager Admin Console, in the top navigation bar, click the Parcels indicator.
2. To the right of the Anaconda parcel listing, click the Deactivate button.
3. When prompted, click OK to deactivate the Anaconda parcel and restart Spark and related services.
4. Click the arrow to the right of the Anaconda parcel listing and select Remove From Hosts.
5. In the confirmation dialog box, confirm removal of the Anaconda parcel from the cluster nodes.

For more information about managing Cloudera parcels, see the Cloudera documentation.

Transitioning to a centrally managed Anaconda installation

Once you've uninstalled the Anaconda parcel, see the *Anaconda Scale installation instructions* for more information about installing a centrally managed version of Anaconda.

Using Anaconda with Spark

Apache Spark is an analytics engine and parallel computation framework with Scala, Python and R interfaces. Spark can load data directly from disk, memory and other data storage technologies such as Amazon S3, Hadoop Distributed File System (HDFS), HBase, Cassandra and others.

Anaconda Scale can be used with a cluster that already has a managed Spark/Hadoop stack. Anaconda Scale can be installed alongside existing enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP and can be used to manage Python and R conda packages and environments across a cluster.

To run a script on the head node, simply execute `python example.py` on the cluster. Alternatively, you can install Jupyter Notebook on the cluster using Anaconda Scale. See the *Installation* documentation for more information.

Different ways to use Spark with Anaconda

You can develop Spark scripts interactively, and you can write them as Python scripts or in a Jupyter Notebook.

You can submit a PySpark script to a Spark cluster using various methods:

- Run the script directly on the head node by executing python example.py on the cluster.
- Use the spark-submit command either in Standalone mode or with the YARN resource manager.
- Submit the script interactively in an IPython shell or Jupyter Notebook on the cluster. For information on using Anaconda Scale to install Jupyter Notebook on the cluster, see *Installation*.

You can also use Anaconda Scale with enterprise Hadoop distributions such as Cloudera CDH or Hortonworks HDP.

Using Anaconda Scale with Spark

The topics listed below describe how to:

- Use Anaconda and Anaconda Scale with Apache Spark and PySpark
- Interact with data stored within the Hadoop Distributed File System (HDFS) on the cluster

While these tasks are independent and can be performed in any order, we recommend that you begin with *Configuring Anaconda with Spark*.

Configuring Anaconda with Spark

You can configure Anaconda to work with Spark jobs in three ways: *with the "spark-submit" command*, or *with Jupyter Notebooks and Cloudera CDH*, or *with Jupyter Notebooks and Hortonworks HDP*.

After you configure Anaconda with one of those three methods, then you can *create and initialize a SparkContext*.

Configuring Anaconda with the `spark-submit` command

You can submit Spark jobs using the `PYSPARK_PYTHON` environment variable that refers to the location of the Python executable in Anaconda.

EXAMPLE:

```
PYSPARK_PYTHON=/opt/continuum/anaconda/bin/python spark-submit pyspark_script.py
```

Configuring Anaconda with Jupyter Notebooks and Cloudera CDH

Configure Jupyter Notebooks to use Anaconda Scale with Cloudera CDH using the following Python code at the top of your notebook:

```
import os
import sys
os.environ["PYSPARK_PYTHON"] = "/opt/continuum/anaconda/bin/python"
os.environ["JAVA_HOME"] = "/usr/java/jdk1.7.0_67-cloudera/jre"
os.environ["SPARK_HOME"] = "/opt/cloudera/parcels/CDH/lib/spark"
os.environ["PYLIB"] = os.environ["SPARK_HOME"] + "/python/lib"
sys.path.insert(0, os.environ["PYLIB"] +"/py4j-0.9-src.zip")
sys.path.insert(0, os.environ["PYLIB"] +"/pyspark.zip")
```

The above configuration was tested with Cloudera CDH 5.11 and Spark 1.6. Depending on the version of Cloudera CDH that you have installed, you might need to customize these paths according to the location of Java, Spark and Anaconda on your cluster.

If you've installed a custom Anaconda parcel, the path for `PYSPARK_PYTHON` will be `/opt/cloudera/parcels/PARCEL_NAME/bin/python`, where `PARCEL_NAME` is the name of the custom parcel you created.

Configuring Anaconda with Jupyter Notebooks and Hortonworks HDP

Configure Jupyter Notebooks to use Anaconda Scale with Hortonworks HDP using the following Python code at the top of your notebook:

```
import os
import sys
os.environ["PYSPARK_PYTHON"] = "/opt/continuum/anaconda/bin/python"
os.environ["SPARK_HOME"] = "/usr/hdp/current/spark-client"
os.environ["PYLIB"] = os.environ["SPARK_HOME"] + "/python/lib"
sys.path.insert(0, os.environ["PYLIB"] +"/py4j-0.9-src.zip")
sys.path.insert(0, os.environ["PYLIB"] +"/pyspark.zip")
```

The above configuration was tested with Hortonworks HDP 2.6, Apache Ambari 2.4 and Spark 1.6. Depending on the version of Hortonworks HDP that you have installed, you might need to customize these paths according to the location of Spark and Anaconda on your cluster.

If you've installed a custom Anaconda management pack, the path for `PYSPARK_PYTHON` will be `/opt/continuum/PARCEL_NAME/bin/python`, where `PARCEL_NAME` is the name of the custom parcel you created.

Creating a SparkContext

Once you have configured the appropriate environment variables, you can initialize a SparkContext–in `yarn-client` client mode in this example–using:

```
from pyspark import SparkConf
from pyspark import SparkContext
conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('anaconda-pyspark')
sc = SparkContext(conf=conf)
```

For more information about configuring Spark settings, see the PySpark documentation.

Once you've initialized a SparkContext, you can start using Anaconda with Spark jobs. For examples of Spark jobs that use libraries from Anaconda, see *Using Anaconda with Spark*.

Running PySpark as a Spark standalone job

This example runs a minimal Spark script that imports PySpark, initializes a SparkContext and performs a distributed calculation on a Spark cluster in standalone mode.

Who is this for?

This example is for users of a Spark cluster that has been configured in standalone mode who wish to run a PySpark job.

Before you start

Download the `spark-basic.py example script` to the cluster node where you submit Spark jobs.

You need Spark running with the standalone scheduler. You can install Spark using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP. Some additional configuration might be necessary to use Spark in standalone mode.

Modifying the script

After downloading the `spark-basic.py example script`, open the file in a text editor on your cluster. Replace `HEAD_NODE_HOSTNAME` with the hostname or IP address of the Spark master as defined in your Hadoop configuration.

```python
# spark-basic.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('spark://HEAD_NODE_HOSTNAME:7077')
conf.setAppName('spark-basic')
sc = SparkContext(conf=conf)

def mod(x):
    import numpy as np
    return (x, np.mod(x, 2))

rdd = sc.parallelize(range(1000)).map(mod).take(10)
print rdd
```

Examine the contents of the `spark-basic.py` example script.

The first code block contains imports from PySpark.

The second code block initializes the SparkContext and sets the application name.

The third code block contains the analysis code that uses the NumPy package to calculate the modulus of a range of numbers up to 1000, then returns and prints the first 10 results.

The fourth code block runs the calculation on the Spark cluster and prints the results. The code uses the NumPy library from Anaconda on each Spark worker.

NOTE: You may need to install NumPy on the cluster nodes using `adam scale -n cluster conda install numpy`.

Running the job

Run the script by submitting it to your cluster for execution using spark-submit or by running this command:

```
$ python spark-basic.py
```

The output from the above command shows the first 10 values returned from the `spark-basic.py` script:

```
16/05/05 22:26:53 INFO spark.SparkContext: Running Spark version 1.6.0

[...]

16/05/05 22:27:03 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID␣
↪0, localhost, partition 0,PROCESS_LOCAL, 3242 bytes)
16/05/05 22:27:04 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory␣
↪on localhost:46587 (size: 2.6 KB, free: 530.3 MB)
16/05/05 22:27:04 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID␣
↪0) in 652 ms on localhost (1/1)
16/05/05 22:27:04 INFO cluster.YarnScheduler: Removed TaskSet 0.0, whose tasks have␣
↪all completed, from pool
16/05/05 22:27:04 INFO scheduler.DAGScheduler: ResultStage 0 (runJob at PythonRDD.
↪scala:393) finished in 4.558 s
16/05/05 22:27:04 INFO scheduler.DAGScheduler: Job 0 finished: runJob at PythonRDD.
↪scala:393, took 4.951328 s
[(0, 0), (1, 1), (2, 0), (3, 1), (4, 0), (5, 1), (6, 0), (7, 1), (8, 0), (9, 1)]
```

Troubleshooting

If something goes wrong, see *Help and support*.

Further information

See the Spark and PySpark documentation:

Running PySpark with the YARN resource manager

This example runs a script on the Spark cluster with the YARN resource manager and returns the hostname of each node in the cluster.

Who is this for?

This example is for users of a Spark cluster who wish to run a PySpark job using the YARN resource manager.

Before you start

Download the `spark-yarn.py example script` to your cluster.

You need Spark running with the YARN resource manager. You can install Spark and YARN using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

Running the job

This code is almost the same as the code on the page *Running PySpark as a Spark standalone job*, which describes the code in more detail.

Here is the complete script to run the Spark + YARN example in PySpark:

```
# spark-yarn.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-yarn')
sc = SparkContext(conf=conf)


def mod(x):
    import numpy as np
    return (x, np.mod(x, 2))

rdd = sc.parallelize(range(1000)).map(mod).take(10)
print rdd
```

NOTE: You may need to install NumPy on the cluster nodes using `adam scale -n cluster conda install numpy`.

Run the script on the Spark cluster with spark-submit. The output shows the first 10 values that were returned from the `spark-basic.py` script.

```
16/05/05 22:26:53 INFO spark.SparkContext: Running Spark version 1.6.0

[...]

16/05/05 22:27:03 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID␣
↪0, localhost, partition 0,PROCESS_LOCAL, 3242 bytes)
16/05/05 22:27:04 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory␣
↪on localhost:46587 (size: 2.6 KB, free: 530.3 MB)
16/05/05 22:27:04 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID␣
↪0) in 652 ms on localhost (1/1)
16/05/05 22:27:04 INFO cluster.YarnScheduler: Removed TaskSet 0.0, whose tasks have␣
↪all completed, from pool
16/05/05 22:27:04 INFO scheduler.DAGScheduler: ResultStage 0 (runJob at PythonRDD.
↪scala:393) finished in 4.558 s
16/05/05 22:27:04 INFO scheduler.DAGScheduler: Job 0 finished: runJob at PythonRDD.
↪scala:393, took 4.951328 s
[(0, 0), (1, 1), (2, 0), (3, 1), (4, 0), (5, 1), (6, 0), (7, 1), (8, 0), (9, 1)]
```

Troubleshooting

If something goes wrong, see *Help and support*.

Further information

See the Spark and PySpark documentation pages for more information.

Performing a word count on text data in HDFS

This example counts the number of words in text files that are stored in HDFS.

Who is this for?

This example is for users of a Spark cluster who wish to run a PySpark job–with the YARN resource manager–that reads and processes text files stored in HDFS.

Before you start

Download the `spark-wordcount.py example script` and the `download-data.py script`.

You need Spark running with the YARN resource manager and the Hadoop Distributed File System (HDFS). You can install Spark, YARN and HDFS using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

You also need valid Amazon Web Services (AWS) credentials.

Load HDFS data

Load the sample text data into HDFS. The following script transfers sample text data (approximately 6.4 GB) from a public Amazon S3 bucket to the HDFS data store on the cluster.

Download the `download-data.py script` to your cluster and insert your Amazon AWS credentials in the `AWS_KEY` and `AWS_SECRET` variables.

```
import subprocess

AWS_KEY = ''
AWS_SECRET = ''

s3_path = 's3n://{0}:{1}@blaze-data/enron-email'.format(AWS_KEY, AWS_SECRET)
cmd = ['hadoop', 'distcp', s3_path, 'hdfs:///tmp/enron']
subprocess.call(cmd)
```

NOTE: The `hadoop distcp` command might fail to run on smaller Amazon EC2 instance sizes due to memory limits.

Run the `download-data.py` script on the Spark cluster:

```
$ python download-data.py
```

After a few minutes, the text data is loaded into HDFS and ready for analysis.

Running the job

The following script reads the text files downloaded in the previous step and counts all of the words. Download the `spark-wordcount.py example script` to your cluster, and then replace `HEAD_NODE_IP` with the IP address of the head node.

```
# spark-wordcount.py
from pyspark import SparkConf
from pyspark import SparkContext

HDFS_MASTER = 'HEAD_NODE_IP'

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-wordcount')
conf.set('spark.executor.instances', 10)
sc = SparkContext(conf=conf)

distFile = sc.textFile('hdfs://{0}:9000/tmp/enron/*/*.txt'.format(HDFS_MASTER))

nonempty_lines = distFile.filter(lambda x: len(x) > 0)
print 'Nonempty lines', nonempty_lines.count()

words = nonempty_lines.flatMap(lambda x: x.split(' '))

wordcounts = words.map(lambda x: (x, 1)) \
                  .reduceByKey(lambda x, y: x+y) \
                  .map(lambda x: (x[1], x[0])).sortByKey(False)

print 'Top 100 words:'
print wordcounts.take(100)
```

Run the script on your Spark cluster using spark-submit. The output shows the 100 most frequently occurring words from the sample text data:

```
54.237.100.240: Using Spark's default log4j profile: org/apache/spark/log4j-defaults.
↪properties
15/06/13 04:58:42 INFO SparkContext: Running Spark version 1.4.0

[...]

15/06/26 04:32:03 INFO YarnScheduler: Removed TaskSet 7.0, whose tasks have all␣
↪completed, from pool
15/06/26 04:32:03 INFO DAGScheduler: ResultStage 7 (runJob at PythonRDD.scala:366)␣
↪finished in 0.210 s
15/06/26 04:32:03 INFO DAGScheduler: Job 3 finished: runJob at PythonRDD.scala:366,␣
↪took 18.124243 s
[(288283320, ''), (22761900, '\t'), (19583689, 'the'), (13084511, '\t0'), (12330608,
↪'-'),
(11882910, 'to'), (11715692, 'of'), (10822018, '0'), (10251855, 'and'), (6682827, 'in
↪'),
(5463285, 'a'), (5226811, 'or'), (4353317, '/'), (3946632, 'for'), (3695870, 'is'),
(3497341, 'by'), (3481685, 'be'), (2714199, 'that'), (2650159, 'any'), (2444644,
↪'shall'),
(2414488, 'on'), (2325204, 'with'), (2308456, 'Gas'), (2268827, 'as'), (2265197, 'this
↪'),
(2180110, '$'), (1996779, '\t$0'), (1903157, '12:00:00'), (1823570, 'The'), (1727698,
↪'not'),
```

(continues on next page)

(continued from previous page)

```
(1626044, 'such'), (1578335, 'at'), (1570484, 'will'), (1509361, 'has'), (1506064,
↪'Enron'),
(1460737, 'Inc.'), (1453005, 'under'), (1411595, 'are'), (1408357, 'from'), (1334359,
↪'Data'),
(1315444, 'have'), (1310093, 'Energy'), (1289975, 'Set'), (1281998, 'Technologies,'),
(1280088, '***********'), (1238125, '\t-'), (1176380, 'all'), (1169961, 'other'),␣
↪(1166151, 'its'),
(1132810, 'an'), (1127730, '&'), (1112331, '>'), (1111663, 'been'), (1098435, 'This'),
(1054291, '0\t0\t0\t0\t'), (1021797, 'States'), (971255, 'you'), (971180, 'which'),␣
↪(961102, '.'),
(945348, 'I'), (941903, 'it'), (939439, 'provide'), (902312, 'North'), (867218,
↪'Subject:'),
(851401, 'Party'), (845111, 'America'), (840747, 'Agreement'), (810554, '#N/A\t'),␣
↪(807259, 'may'),
(800753, 'please'), (798382, 'To'), (771784, '\t$-'), (753774, 'United'), (740472, 'if
↪'),
(739731, '\t0.00'), (723399, 'Power'), (699294, 'To:'), (697798, 'From:'), (672727,
↪'Date:'),
(661399, 'produced'), (652527, '2001'), (651164, 'format'), (650637, 'Email'),␣
↪(646922, '3.0'),
(645078, 'licensed'), (644200, 'License'), (642700, 'PST'), (641426, 'cite'), (640441,
↪ 'Creative'),
(640089, 'Commons'), (640066, 'NSF'), (639960, 'EML,'), (639949, 'Attribution'),
(639938, 'attribution,'), (639936, 'ZL'), (639936, '(http://www.zlti.com)."'),␣
↪(639936, '"ZL'),
(639936, 'X-ZLID:'), (639936, '<http://creativecommons.org/licenses/by/3.0/us/>'),␣
↪(639936, 'X-SDOC:')]
```

Troubleshooting

If something goes wrong, see *Help and support*.

Further information

See the Spark and PySpark documentation:

Distributed natural language processing

This example provides a simple PySpark job that utilizes the NLTK library. NLTK is a popular Python package for natural language processing. This example shows you how to integrate third-party Python libraries with Spark. This example demonstrates the installation of Python libraries on the cluster, the usage of Spark with the YARN resource manager and execution of the PySpark job.

Who is this for?

This example is for users of a Spark cluster who wish to run a PySpark job with the YARN resource manager.

Before you start

Download the `spark-nltk.py example script` or `spark-nltk.ipynb example notebook`.

You need Spark running with the YARN resource manager. You can install Spark and YARN using an enterprise Hadoop distribution such as Cloudera CDH or Hortonworks HDP.

Install NLTK

Install NLTK on all of the cluster nodes using the `adam scale` command:

```
$ adam scale -n cluster conda install nltk
```

You should see output similar to this from each node, which indicates that the package was successfully installed across the cluster:

```
All nodes (x4) response:
{
  "actions": {
    "EXTRACT": [
      "conda-env-2.5.2-py27_0",
      "conda-4.1.11-py27_0"
    ],
    "FETCH": [
      "conda-env-2.5.2-py27_0",
      "conda-4.1.11-py27_0"
    ],
    "LINK": [
      "conda-env-2.5.2-py27_0 1 None",
      "conda-4.1.11-py27_0 1 None"
    ],
    "PREFIX": "/opt/continuum/anaconda",
    "SYMLINK_CONDA": [
      "/opt/continuum/anaconda"
    ],
    "UNLINK": [
      "conda-4.1.6-py27_0",
      "conda-env-2.5.1-py27_0"
    ],
    "op_order": [
      "RM_FETCHED",
      "FETCH",
      "RM_EXTRACTED",
      "EXTRACT",
      "UNLINK",
      "LINK",
      "SYMLINK_CONDA"
    ]
  },
  "success": true
}
```

For this example, you need to download the NLTK sample data. Download the data on all cluster nodes by using the `adam cmd` command:

```
$ adam cmd 'sudo /opt/continuum/anaconda/bin/python -m nltk.downloader -d /usr/share/
↪nltk_data all'
```

The sample data is downloaded over a few minutes. After the download completes, you should see output similar to:

```
All nodes (x4) response: [nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /usr/share/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /usr/share/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /usr/share/nltk_data...

....

[nltk_data]    |   Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /usr/share/nltk_data...
[nltk_data]    |   Unzipping models/word2vec_sample.zip.
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

Running the job

Here is the complete script to run the Spark + NLTK example in PySpark:

```python
# spark-nltk.py
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-nltk')
sc = SparkContext(conf=conf)

data = sc.textFile('file:///usr/share/nltk_data/corpora/state_union/1972-Nixon.txt')

def word_tokenize(x):
    import nltk
    return nltk.word_tokenize(x)

def pos_tag(x):
    import nltk
    return nltk.pos_tag([x])

words = data.flatMap(word_tokenize)
print words.take(10)

pos_word = words.map(pos_tag)
print pos_word.take(5)
```

Examine the above code example. First, it imports PySpark and creates a SparkContext:

```python
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
conf.setMaster('yarn-client')
conf.setAppName('spark-nltk')
sc = SparkContext(conf=conf)
```

After a SparkContext is created, we can load some data into Spark. In this case, the data file is from one of the example documents provided by NLTK.

NOTE: You could also copy the data to HDFS and load it from Spark.

```
data = sc.textFile('file:///usr/share/nltk_data/corpora/state_union/1972-Nixon.txt')
```

Next is a function called `word_tokenize` that imports `nltk` on the Spark worker nodes and calls `nltk.word_tokenize`. The function is mapped to the text file that was read in the previous step:

```
def word_tokenize(x):
    import nltk
    return nltk.word_tokenize(x)

words = data.flatMap(word_tokenize)
```

You can confirm that the `flatMap` operation worked by returning some of the words in the dataset:

```
print words.take(10)
```

Finally, you can use NTLK's part-of-speech tagger to attach the part of speech to each word in the data set:

```
def pos_tag(x):
    import nltk
    return nltk.pos_tag([x])

pos_word = words.map(pos_tag)
print pos_word.take(5)
```

Run the script on the Spark cluster using the spark-submit script. The output shows the words that were returned from the Spark script, including the results from the `flatMap` operation and the `POS-tagger`:

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
15/06/13 05:14:29 INFO SparkContext: Running Spark version 1.4.0

[...]

['Address',
 'on',
 'the',
 'State',
 'of',
 'the',
 'Union',
 'Delivered',
 'Before',
 'a']

[...]

[[('Address', 'NN')],
 [('on', 'IN')],
 [('the', 'DT')],
 [('State', 'NNP')],
 [('of', 'IN')]]
```

Troubleshooting

If something goes wrong, consult *Help and support*.

Further information

See the Spark and PySpark documentation:

For more information on NLTK, see the NLTK book.

Help and support

Support is included with the purchase of an Anaconda subscription. Visit the support section of our website for documentation and contact information for priority support.

Training and consulting

Training and consulting is available for Anaconda Scale. For more information, please contact us.

Index