

CONDA

**CHRIS CALLOWAY FOR
TRIANGLE PYTHON USERS GROUP
AT CAKTUS GROUP
DECEMBER 14, 2017**



WHAT IS CONDA

- Cross-platform
- Language Agnostic
- Package Manager
- Dependency Manager
- Environment Manager
- Package Creator
- Command Line Interface to the Conda Community

 CONDA



WHAT IS CONDA

- Cross-platform:
 - Linux
 - OSX
 - Windows

The word "CONDA" is written in a bold, green, sans-serif font. The letter 'C' is stylized with a white, grid-like pattern inside its upper curve.



WHAT IS CONDA

- Language Agnostic:
 - Python
 - R
 - Ruby
 - Lua
 - Scala
 - Java
 - JavaScript
 - C/ C++
 - FORTRAN

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.



WHAT IS CONDA

- Package Manager:
 - Replaces pip
 - Pip still works with Conda

 CONDA



WHAT IS CONDA

- Dependency Manager:
 - Replaces setuptools
 - Setuptools is still useful for making Conda-installable packages

 CONDA



WHAT IS CONDA

- Environment Manager:
 - Replaces virtualenv and venv
 - **Virtualenv and venv don't work with Conda**

 CONDA



WHAT IS CONDA

- Package Creator:
 - Conda recipes create Conda packages
 - Conda recipes can wrap a setup.py

CONDA



WHAT IS CONDA

- Command Line Interface to the Conda community
 - <https://anaconda.org/>
 - A package repository
 - A social network
 - A directory of package “channels”
 - An Anaconda channel
 - A community conda-forge channel
 - Your channel
 - Your organization's channel

 CONDA



WHAT IS CONDA

- Open source
- Created by Continuum Analytics
 - Creators of the Anaconda Python distribution

CONDA



WHAT IS CONDA

- Huge uptake in the scientific and data analytics communities
 - By virtue of inclusion in the Anaconda Python distribution
 - SciPy community
 - PyData community

The word "CONDA" is written in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.



WHAT IS CONDA

- Less uptake in the web community although:
 - Django
 - Flask
 - CherryPy
 - Pyramid
 - Etc.

Are all available through Conda

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

INSTALLING CONDA

- Conda comes with Anaconda
 - Download at <https://www.anaconda.com/download/>
 - Choose platform:
 - Linux
 - OSX
 - Windows
 - Choose architecture:
 - 64 bit
 - 32 bit
 - Choose Python version:
 - 3.6
 - 2.7

 CONDA



INSTALLING CONDA

- Chose installer type:
 - Graphical
 - Command Line
- Choose:
 - Full Anaconda (~500 packages in addition to the Python Standard Library)
 - Miniconda (Just enough to get started)
 - <https://conda.io/miniconda.html>
- Prepend to PATH
 - Optional at conclusion of installer
 - `export PATH="/Users/cbc/anaconda/bin:$PATH"`



CONDA



MANAGING ENVIRONMENTS

- Parent environment:

```
$ python
```

```
Python 3.6.0 |Anaconda 4.3.1 (x86_64)| (default, Dec  
23 2016, 13:19:00)
```

```
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-  
600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for  
more information.
```

```
>>>
```

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

MANAGING ENVIRONMENTS

- Parent environment:

```
$ ipython
```

```
Python 3.6.0 |Anaconda 4.3.1 (x86_64)| (default, Dec 23  
2016, 13:19:00)
```

```
Type "copyright", "credits" or "license" for more  
information.
```

```
IPython 5.1.0 -- An enhanced Interactive Python.
```

```
? -> Introduction and overview of IPython's  
features.
```

```
%quickref -> Quick reference.
```

```
help -> Python's own help system.
```

```
object? -> Details about 'object', use 'object??' for  
extra details.
```

```
In [1]:
```



MANAGING ENVIRONMENTS

- Parent environment:

```
$ jupyter notebook
[I 20:29:16.190 NotebookApp] Serving notebooks from local
directory: /Users/cbc
[I 20:29:16.190 NotebookApp] 0 active kernels[I
20:29:16.190 NotebookApp] The Jupyter Notebook is running
at:
http://localhost:8888/?token=7e10099bd7e3b1857b4b491b9509
db3d49789174ff456404
[I 20:29:16.190 NotebookApp] Use Control-C to stop this
server and shut down all kernels (twice to skip
confirmation).

...
[I 20:29:16.402 NotebookApp] Accepting one-time-token-
authenticated connection from ::1
```



MANAGING ENVIRONMENTS

- Parent environment:

```
$ which python  
/Users/cbc/anaconda/bin/python  
$
```

CONDA



MANAGING ENVIRONMENTS

- Create a child environment:

```
$ conda create -n example anaconda
```

- Generates a ton of output while also prompting for acceptance of package list
- The conda command takes subcommands

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

MANAGING ENVIRONMENTS

- List environments:

```
$ conda env list
# conda environments
#
example      /Users/cbc/anaconda/envs/example
root        * /Users/cbc/anaconda
$
```

- Conda subcommands may take their own subcommands
- The splat indicates the currently "active" environment

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

MANAGING ENVIRONMENTS

- List subcommands:

```
$ conda -h
```

- info
- help
- list
- search
- create
- install
- update
- upgrade
- remove
- uninstall
- config
- clean

- Includes a brief explanation of each subcommand

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

MANAGING ENVIRONMENTS

- List subcommands of subcommands:

```
$ conda env -h
```

- attach
- create
- export
- list
- remove
- update
- upload

- Shows both options and further subcommands and brief explanations of both

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

MANAGING ENVIRONMENTS

- List further subcommands and options

```
$ conda env export -h
```

- Show both short and long form options

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

ACTIVATING ENVIRONMENTS

```
$ which python
/Users/cbc/anaconda/bin/python
$ source activate example
(example) $ which python
/Users/cbc/anaconda/envs/example/bin/python
(example) $
```

- Notice the command prompt changes to include the name of the active child environment in parentheses

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white and green pattern resembling a DNA double helix or a network structure.



ACTIVATING ENVIRONMENTS

- On Windows, simply:

```
$ activate example  
(example) $
```

- `activate` is a batch file on Windows

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white and green pattern resembling a DNA double helix or a snake's head.

ACTIVATING ENVIRONMENTS

- All activation really does is munge your PATH environment variable so that the Conda environment is searched first:

```
(example) $ echo $PATH
```

```
/Users/cbc/anaconda/envs/example/bin:
```

```
/Users/cbc/anaconda/bin:
```

```
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
(example) $ which python
```

```
/Users/cbc/anaconda/envs/example/bin/python
```

```
(example) $
```



DEACTIVATING ENVIRONMENTS

```
(example) $ source deactivate
```

```
$ echo $PATH
```

```
/Users/cbc/anaconda/bin:
```

```
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
$ which python
```

```
/Users/cbc/anaconda/bin/python
```

```
$
```

- Notice the command prompt changes back to not include a child environment name
- Simply **deactivate** on Windows

The logo for CONDA, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

INSTALLING TO ENVIRONMENTS

- Installing packages is the primary reason you want to have a child environment
- Each environment has an **isolated** set of installed packages
- Multiple environments may have different versions of the same package without interfering with one another

The word "CONDA" is written in a bold, green, sans-serif font. The letter 'C' is stylized with a white, grid-like pattern inside its upper curve.



INSTALLING TO ENVIRONMENTS

- If the packages installed in an environment become problematic, you can simply remove the environment
- It's a good idea not to install additional packages to your parent environment
- Environments can be recreated by exporting the list of installed packages to a file

The word "CONDA" is written in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.



INSTALLING TO ENVIRONMENTS

- Let's create a new child environment and activate it:

```
$ conda create -n example1 python=2
$ source activate example1
(example1) $
```

- We've specified a particular version of package
- Installed packages can be binaries, not just Python packages
- Even a different Python binary than the parent Python
- Installing just Python and its dependencies is like installing miniconda



INSTALLING TO ENVIRONMENTS

- Easy to manage multiple versions of Python

```
(example1) $ python
Python 2.7.13 |Continuum Analytics, Inc.| (default, Dec 20 2016, 23:05:08)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>>
```

The word "CONDA" is written in a large, bold, green sans-serif font. The letter "C" is stylized with a white and green pattern resembling a snake's head or a DNA helix.



INSTALLING TO ENVIRONMENTS

- Let's install Node.js in our child environment:

```
(example1) $ which node
(example1) $ conda install nodejs
... lots of output ...
(example1) $ which node
/Users/cbc/anaconda/envs/example1/bin/node
(example1) $ node
> .exit
(example1) $
```

- Conda is language agnostic

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

INSTALLING TO ENVIRONMENTS

- Your package installations to your child environment are isolated to your child environment:

```
(example1) $ source deactivate  
$ which node  
$
```

- This is what makes Conda a superior alternative to other tools such as homebrew, ports, fink, yum, apt-get, and virtualenv

The word "CONDA" is written in a bold, green, sans-serif font. The letter 'C' is stylized with a white, grid-like pattern inside its upper curve.

UNINSTALLING

- Let's uninstall Node.js from our child environment:

```
$ source activate example1  
(example1) $ conda uninstall nodejs  
... lots of output ...  
(example1) $ which node  
(example1) $
```

- Make sure you know which environment is activated!



REMOVING AN ENVIRONMENT

- Or we could remove the entire child environment:

```
(example1) $ source deactivate
$ conda env remove -n example1
$ which python
/Users/cbc/anaconda/bin/python
$
```

- Make sure the environment is deactivated!



CONDA CHANNELS



**ANACONDA
CLOUD**

CONDA



CONDA CHANNELS

ANACONDA CLOUD

Search Anaconda Cloud



View

Help



cbcunc

My Anaconda Landscape

Packages

View all (0)

Get more information on how to create a Package.

Notebooks

View all (0)

Get more information on how to create a Notebook.

Environments

View all (0)

Get more information on how to create an Environment.

Projects

View all (0)

No projects yet, [upload one here](#).

Favorites

View all (0)

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

Activity Feed

View more

cbcunc created the organization **renci** 2 years and 8 months ago

Welcome to Anaconda Cloud! 3 years and 2 months ago

Anaconda Cloud allows you to create or distribute software packages.

Getting started: [Installing your first package](#)

Getting started: [Distributing your first package](#)

CONDA



CONDA CHANNELS

 ANACONDA CLOUD

Search Anaconda Cloud



View ▾

Help ▾



cbcunc ▾

Profile

cbcunc



Contributor since Sep 18, 2014

cbc@chrisalloway.org

Organizations



Looks like user **cbcunc** has nothing to share!

CONDA



CONDA CHANNELS

 ANACONDA CLOUD

Search Anaconda Cloud



Profile

conda-forge













Organization created on Apr 11, 2015

<https://conda-forge.github.io>

Packages

View all (3550)

-  [perl-app-cpanminus](#) 1 hour and 5 minutes ago
-  [jupyter_sphinx](#) 1 hour and 20 minutes ago
-  [progress_reporter](#) 1 hour and 27 minutes ago
-  [r-rspectra](#) 2 hours and 37 minutes ago
-  [r-splus2r](#) 2 hours and 50 minutes ago
-  [r-gee](#) 2 hours and 56 minutes ago
-  [pymultinest](#) 2 hours and 59 minutes ago
-  [fmpy](#) 3 hours and 1 minute ago
-  [fann2](#) 3 hours and 4 minutes ago
-  [ipython_unittest](#) 3 hours and 19 minutes ago

CONDA



CONDA CHANNELS

- Create a child environment entirely of packages from the conda-forge channel:

```
$ conda create -n example2 -c conda-forge pandas
Fetching package metadata .....
Solving package specifications: .
Package plan for installation in environment
/Users/cbc/anaconda/envs/example2:
The following NEW packages will be INSTALLED:
  ca-certificates: 2017.11.5-0          conda-forge
  zlib: 1.2.11-0                        conda-forge
Proceed ([y]/n)? Y
ca-certificate 100% |#####| Time: 0:00:00    2.58 MB/s
.....
zlib-1.2.11-0. 100% |#####| Time: 0:00:00   19.44 MB/s
$.....
$
```



CONDA CHANNELS

- Dependencies will come from the default channels if not in the specified channel:

```
$ source activate example2
(example2) $ conda list
# packages in environment at /Users/cbc/anaconda/envs/example2:
#
ca-certificates      2017.11.5          0         conda-forge
certifi              2017.11.5        py36_0         conda-forge
.....

numpy                 1.13.1           py36_0
.....
(example2) $
```

- Notice that numpy is coming from the default Anaconda channel, not conda-forge



CONDA CHANNELS

- The installed package list can be piped to a file:

```
(example2) $ conda list --explicit > example2.txt  
(example2) $ cat example2.txt  
(example2) $
```

- Similar to a requirements.txt file
- The `--explicit` option outputs URLs to the exact
 - Channel
 - Versionof each package



CONDA CHANNELS

- The package list file can be used to create a new environment:

```
(example2) $ conda create -n example3 --file  
example2.txt  
(example2) $ source activate example3  
(example3) $ conda list
```

- You don't have to deactivate one environment before creating or even activating another
- This makes conda also a replacement for much of virtualenvwrapper
- The package list file can be committed to your version control system



CONDA CHANNELS

- You don't need a package list file to clone an environment:

```
(example3) $ conda create -n example4 --clone example3  
(example3) $ source activate example4  
(example4) $ conda list
```

The logo for Conda, featuring the word "CONDA" in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.

CONDA CHANNELS

- You may search default channels by keyword:

```
(example4) $ conda search node
Fetching package metadata .....
node-webkit.      0.10.1      0      defaults
nodejs            4.2.6       0      defaults
                  6.10.3      0      defaults
```

(example4) \$

- There are currently two supported node versions in the Anaconda default channel



CONDA CHANNELS

- You may add specific channels to your search:

```
(example4) $ conda search -c conda-forge node
Fetching package metadata .....
node-webkit      0.10.1      @          defaults
nodebook         0.2.1       py_0       conda-forge
nodeenv          1.2.0       py_0       conda-forge
nodejs           4.2.6       @          defaults
                 6.10.3      @          defaults
                 4.4.1       @          conda-forge
                 4.4.7       @          conda-forge
                 4.5.0       @          conda-forge
                 6.6.0       @          conda-forge
                 6.10.2      @          conda-forge
                 6.11.0      @          conda-forge
                 6.12.0      @          conda-forge
                 8.8.1       @          conda-forge
```

(example4) \$

- Other channels may have many other versions of particular packages



CONDA CHANNELS

- You may add channels to be included in all commands:

```
(example4) $ conda config --add channels conda-forge
```

```
(example4) $ cat ~/.condarc
```

```
channels:
```

- conda-forge
- defaults

```
(example4) $
```

- Using conda config creates a .condrc hidden YAML file in your home directory
- Conda config is much like git config



CONDA CHANNELS

- You may see all your configuration:

```
(example4) $ conda config --show
```

```
... lots of output ...
```

```
(example4) $ conda config --describe
```

```
... lots of output ...
```

```
(example4)
```

- You may also see descriptions of all the available configuration keys



CONDA CHANNELS

- If you know a package name, you may show a lot of information about all the available versions of the package:

```
(example4) $ conda info iris  
... lots of output ...  
(example4) $
```

- Now that conda-forge was added to defaults in .condarc, we don't need to add `-c channel` to our conda commands to use the conda-forge channel
- Channels are searched in the order they appear in .condarc



CONDA CHANNELS

- If you know a package name, you may show a lot of information about all the available versions of the package:

```
(example4) $ conda info iris  
... lots of output ...  
(example4) $
```

- Iris is a meteorological package not in the Anaconda channel
- Many discipline specific packages are available in conda-forge
- Packages added to conda-forge are community reviewed and moderated



CONDA RECIPES

- Conda packages are built by a new conda command you install:

```
(example4) $ source deactivate  
$ conda install conda-build  
$ conda update conda  
$ conda update conda-build  
$
```

- The conda-build package adds several new conda subcommands



CONDA RECIPES

- Skeleton is one of those new subcommands:

```
$ mkdir Recipes  
$ cd Recipes  
$ conda skeleton pypi primer  
$ ls primer  
$
```

- Skeleton's purpose is to create boilerplate for your Conda recipe
- The recipe a directory with at least a meta.yaml file
- The simplest way to create a pure Python recipe is from a distribution on the Python package index (pypi)
- Primer is an example Python distribution



CONDA RECIPES

- It is normal to edit the meta.yaml file created by skeleton:

```
$ emacs primer/meta.yaml  
$
```

- Meta.yaml is metadata for your conda recipe
- The recipe directory may contain other files
- But only meta.yaml is required
- Meta.yaml keys are described in the online Conda docs
- Only a few are required

The word "CONDA" is written in a bold, green, sans-serif font. The letter "C" is stylized with a white, jagged, sawtooth-like pattern on its left side.

CONDA RECIPES

- The build subcommand creates a distribution from the recipe:

```
$ conda build primer  
... lots of output ...  
$ ls ~/anaconda/conda-bld  
$
```

- Distributions are compressed file bundles
- Distributions are placed in your Anaconda parent's conda-bld subdirectory



CONDA RECIPES

- The anaconda command manages Anaconda Cloud:

```
$ anaconda upload \  
~/anaconda/conda-bld/osx-64/primer-1.4.4-py36_0.tar.bz2  
... lots of output ...  
$
```

- Upload is one of the subcommands of the anaconda command
- You will need to authenticate with anaconda.org



CONDA RECIPES

ANACONDA CLOUD

Search Anaconda Cloud



View Help



cbcunc

My Anaconda Landscape

Packages

View all (1)

primer 3 minutes and a few seconds ago

Notebooks

View all (0)

Get more information on how to create a Notebook.

Environments

View all (0)

Get more information on how to create an Environment.

Projects

View all (0)

No projects yet, [upload one here](#).

Favorites

View all (0)

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

Activity Feed

View more

cbcunc created package [cbcunc/primer](#) 3 hours and 24 minutes ago

cbcunc created the organization [renci](#) 2 years and 8 months ago

Welcome to [Anaconda Cloud](#)! 3 years and 2 months ago

Anaconda Cloud allows you to create or distribute software packages.

Getting started: [Installing your first package](#)

Getting started: [Distributing your first package](#)

CONDA



CONDA RECIPES

 ANACONDA CLOUD

Search Anaconda Cloud



[Gallery](#)

[About](#)

[Pricing](#)

[Anaconda](#)

[Help](#)

[Download Anaconda](#)

[Sign In](#)

Profile

cbcunc




Contributor since Sep 18, 2014

Organizations



Packages

[View all \(1\)](#)

 primer 9 minutes and a few seconds ago

CONDA



CONDA RECIPES

- Let's create an environment in which to install the new distribution:

```
$ cd ~
$ conda create -n primer python
... lots of output ...
$ source activate primer
(primer) $ python
>>> import primer
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'primer'
>>>
```



CONDA RECIPES

- And let's install our new distribution into the new environment:

```
>>> exit()
(primer) $ conda install -c cbcunc primer
... lots of output ...
(primer) $ conda list
...
primer          1.4.4          py36_0          cbcunc
...
$
```



CONDA RECIPES

- The primer distribution is coming from a personal channel (cbcunc)
- You can install from public packages in anyone's channel
- Package may also be private or require authentication

The word "CONDA" is written in a bold, green, sans-serif font. The letter "C" is stylized with a white, grid-like pattern inside its upper curve.



CONDA RECIPES

- Awesome sauce:

```
(primer) $ python
>>> import primer
>>> primer.primes(10)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
>>>
```

- Get more information at <https://conda.io/>

