Installing the Earth Engine Python API using the conda packaging manager

Table of Contents:

[Installing the conda Package Manager](#_iuy74kr8er9f)

[Mac OSX](#_lfz6ch66otqk)

[Windows](#_2vzpnaul2fdo)

[Linux](#_nuc3ngrghutp)

[Creating a conda Environment for the Earth Engine API](#_ui00wzkycr)

[Authenticating to Earth Engine](#_hm481g6xxeou)

[Adding IPython Notebooks and SciPy to the conda Environment](#_rdx78t45vmjl)

[Configuring App Engine to use the conda Environment](#_3k84tnatgql1)

[Troubleshooting](#_f106xpqtwbjf)

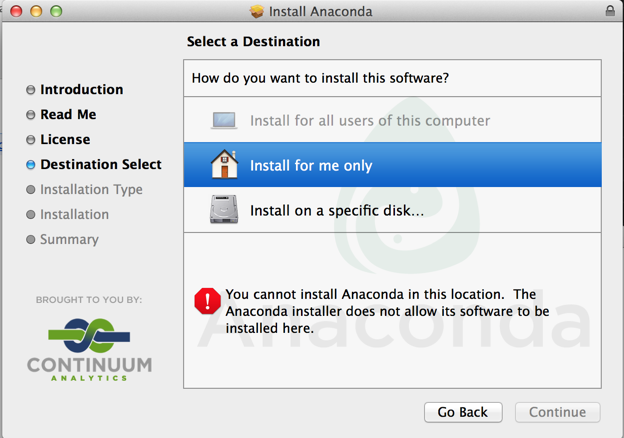
# Installing the conda Package Manager

[**conda**](http://conda.pydata.org/docs/intro.html) is a cross-platform application for installing software packages. Although it is not specific to installing Python packages, it is being adopted by the scientific Python community for it's ability to handle Python packages that have dependencies on shared libraries. conda is open source, and is available as part of the [Anaconda](http://docs.continuum.io/anaconda/index.html) Python distribution, managed by Continuum Analytics.

The easiest way to install conda, is to download and install the Anaconda Python Distribution available. Binary installers are available for various operating systems, and specific instructions/notes for each operating system are listed below.

If you don't want to install the full Anaconda distribution (which is about 350MB), there are alternative methods of installing conda: you can install the minimal conda installation (a.k.a. [miniconda](http://conda.pydata.org/miniconda.html)), install the [conda package from PyPI](https://pypi.python.org/pypi/conda), or to download and install from the [source code](https://github.com/conda/conda).

## Mac OSX

1. Go to <http://continuum.io/downloads> and download the Python 2.7 Graphical Installer.
2. Execute the Graphical Installer package, and follow the instructions.  
   *Note that you may encounter the following message:*  
   *To proceed, first click on the "Install on a specific disk…" option, then click back on the "Install for me only".*
3. After completing the process, Anaconda will be installed in your home directory (i.e. $HOME/anaconda).

## Windows 7

1. Go to <http://continuum.io/downloads> and download the Python 2.7 Graphical Installer for your platform (32 bit or 64 bit).
2. Execute the Graphical Installer package, and follow the instructions.
3. When asked where it should be installed, choose “Just me”. By default, Anaconda will be installed in C:\Users\<username>\AppData\Local\Continuum\Anaconda.
4. Go to your Start Menu > All Programs > Anaconda folder, and start the “Anaconda Command Prompt”.
5. List the system path  
   echo %PATH%  
   If the system path does not start with a path that includes the Anaconda installation location, see the troubleshooting section below.

https://github.com/ContinuumIO/anaconda-issues/issues/41

## Linux

1. Go to <http://continuum.io/downloads> and download the Python 2.7 Installer.
2. Follow the instructions for executing the installer. Note that you should not need administrative (sudo) privileges.
3. During the installation process your shell configuration file (such as ~/.bash\_profile or ~/.bashrc) may be altered so that the Anaconda tools are added to your system path (i.e. by prepending the Anaconda bin directory to the PATH environmental variable). This will cause the version of Python packaged with Anaconda to become your default Python version.  
   # added by Anaconda 2.0.1 installer  
   export PATH="/home/johnsmith/anaconda/bin:$PATH"  
   If you do not want to add the Anaconda tools to your path by default, you instead create an alias that adds Anaconda to the path. For example, you can modify your shell configuration file as follows:  
   # added by Anaconda 2.0.1 installer  
   #export PATH="/home/johnsmith/anaconda/bin:$PATH"  
   alias useconda='export PATH="$HOME/anaconda/bin:$PATH"; unset PYTHONPATH'  
   Then whenever you want to use Anaconda, just manually execute the aliased command by executing:  
   useconda
4. By default all files are installed in a single folder your home directory (~/anaconda). To remove Anaconda, you can simply delete the directory.

## Updating conda

1. Once you have installed conda, run the following to command to update conda to the most recent version.  
   conda update conda  
   To check which version of conda you are running, execute the following command:  
   conda --version

# Creating a conda Environment for the Earth Engine API

This section illustrates how to build a conda environment that allows you to run Python scripts against the Earth Engine API. Note that these instructions were written using conda version 3.5.5.

1. Create a new conda environment that includes Python 2.7.  
   conda create -n ee-python python=2.7  
   A plan for downloading and linking packages will be presented, which you can review before proceeding. Once you proceed, a directory in your anaconda directory will be created.
2. Next activate this environment:  
   [OSX, Linux] source activate ee-python  
   [Windows] activate ee-python  
   Don't do this now, but once you are done using a conda environment, you can deactivate the environment by executing the following code:  
   [OSX, Linux] source deactivate  
   [Windows] deactivate
3. Install pycrypto into the conda environment. This package is used for OAuth2 authentication.  
   conda install pycrypto
4. Install PIP into the conda environment, so that it can be used to install Python packages that are not available as conda packages, but are available in the Python Package Index ([PyPI](https://pypi.python.org)).  
   conda install pip
5. Next install the Earth Engine Python API package. (Note that the "no dependency" flag is used, so that pip does not try to load the pyOpenSSL dependency that is specified in the package's setup.py file. This dependency will be satisfied by installing the conda Crypto package.)  
   pip install oauth2client  
   pip install --no-deps earthengine-api
6. Test that you can successfully import the Earth Engine API package, by running:  
   python -c "import ee; print ee.\_\_version\_\_"  
   **Note for Windows users:** *Make sure that you execute this command in a directory that does not contains python.exe (such as the default Anaconda Command Prompt) , so that the python command correctly executes the python executable in the conda environment and has access to the Python packages that you have installed.*

## Authenticating to Earth Engine

To connect to Earth Engine from Python, you need to:

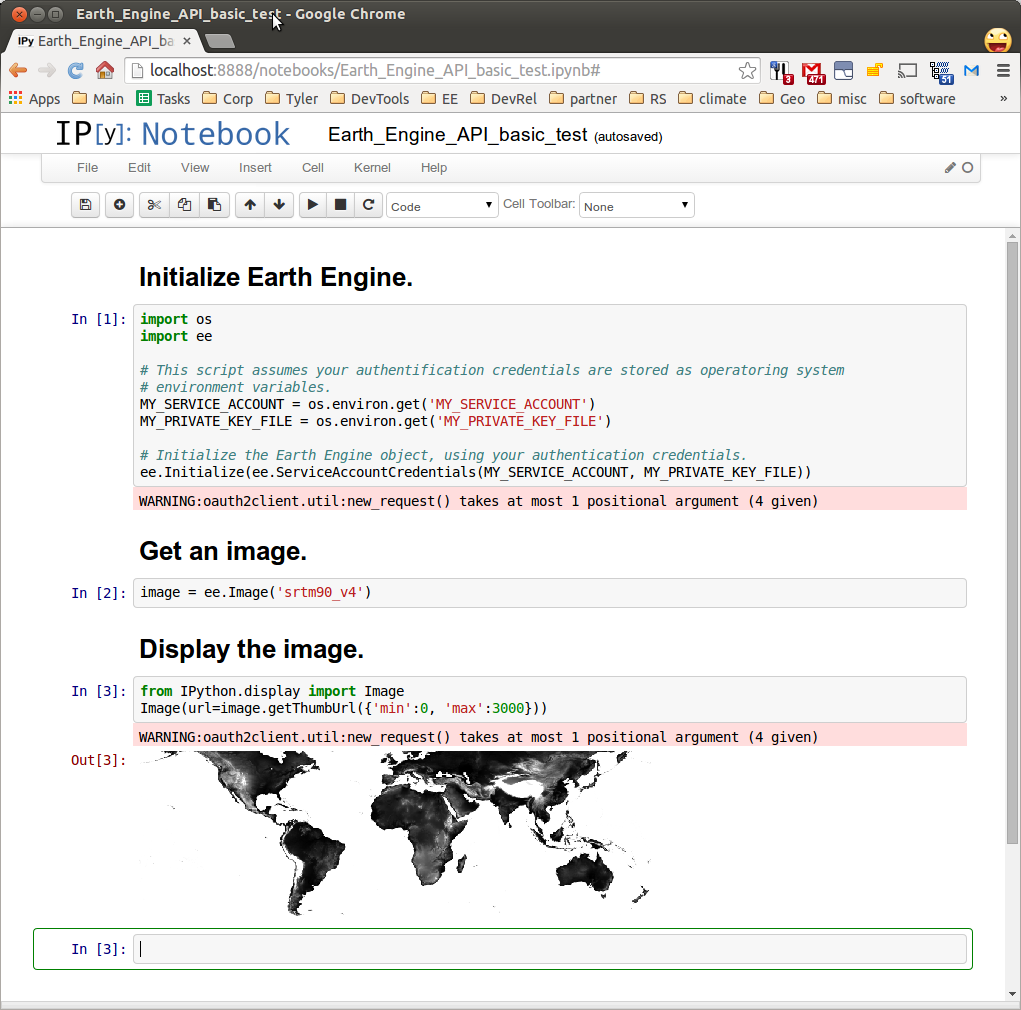
1. Create a service account as described on [this page](https://sites.google.com/site/earthengineapidocs/python-api). At the end of the process you should have a service account email address and a private key file (PKCS12 format, ending in .p12) that you downloaded. Move the private key file to a convenient location, like a folder in your home directory. For example:  
   [OSX, Linux]  
   mkdir ~/.keys  
   mv ~/Downloads/[KEYFILE\_NAME].p12 ~/.keys/privatekey.p12   
   [Windows]  
   mkdir %HOMEPATH%\.keys  
   move %HOMEPATH%\Downloads\\*.p12 %HOMEPATH%\.keys\privatekey.p12
2. Send the service account email address to your contact on the Earth Engine team, so they can whitelist the service account.
3. Convert the private key file (PKCS12 format) into a PEM formatted file using OpenSSL. For Windows, you may need to install OpenSSL from: http://slproweb.com/products/Win32OpenSSL.html  
   For example:  
   [OSX, Linux]  
   openssl pkcs12 -in $HOME/.keys/privatekey.p12 -nodes -nocerts > $HOME/.keys/privatekey.pem  
   [Windows]  
   C:\OpenSSL-Win64\bin\openssl pkcs12 -in %HOMEPATH%\.keys\privatekey.p12 -nodes -nocerts > %HOMEPATH%\.keys\privatekey.pem   
   Note that you may be asked for a password, which was presented to you when you created your key file (“notasecret”). Also note that this assumes you have placed your key files in a subdirectory of your home directory called ".keys", but you can store the key files in other locations.
4. Open up the PEM file with a text editor, and remove any lines that occur **before** "-----BEGIN". (This preamble is not recognized by old versions of oauth2client library.)
5. Set your system environment variables to your service account (which has been whitelisted to access Earth Engine) and your private key file. For example:  
   [OSX, Linux]  
   export MY\_SERVICE\_ACCOUNT=1234567890@developer.gserviceaccount.com  
   export MY\_PRIVATE\_KEY\_FILE=$HOME/.keys/privatekey.pem   
   Note that you may want to add these two commands to your shell configuration file (~/.bashrc or ~/.bash\_profile) so that they are run automatically when you open a new terminal shell.  
   [Windows]  
   set MY\_SERVICE\_ACCOUNT=1234567890@developer.gserviceaccount.com  
   set MY\_PRIVATE\_KEY\_FILE=%HOMEPATH%\.keys\privatekey.pem  
   [Windows 8]  
   setx MY\_SERVICE\_ACCOUNT 1234567890@developer.gserviceaccount.com  
   setx MY\_PRIVATE\_KEY\_FILE %HOMEPATH%\.keys\privatekey.pem
6. Test out whether you can run a Python script that authenticates to Earth Engine. If successful, this script prints out a URL that can be used to view an image generated by Earth Engine.  
   python -c "import os; import ee; MY\_SERVICE\_ACCOUNT = os.environ.get('MY\_SERVICE\_ACCOUNT'); MY\_PRIVATE\_KEY\_FILE = os.environ.get('MY\_PRIVATE\_KEY\_FILE'); ee.Initialize(ee.ServiceAccountCredentials(MY\_SERVICE\_ACCOUNT, MY\_PRIVATE\_KEY\_FILE)); print(ee.Image('srtm90\_v4').getThumbUrl())"

## Adding IPython Notebooks and SciPy to the conda Environment

In many cases, additional tools may make working with the Earth Engine Python API easier. This section will walk through the installation of several useful tools:

* [IPython Notebooks](http://ipython.org/notebook.html) - a browser-based interactive computing environment that can display various types of output
* [SciPy](http://www.scipy.org/scipylib/index.html) library - a collection of efficient numerical routines for scientific analysis, and
* [matplotlib](http://matplotlib.org/) - a 2D plotting library for Python

Instructions for installing in a conda environment:

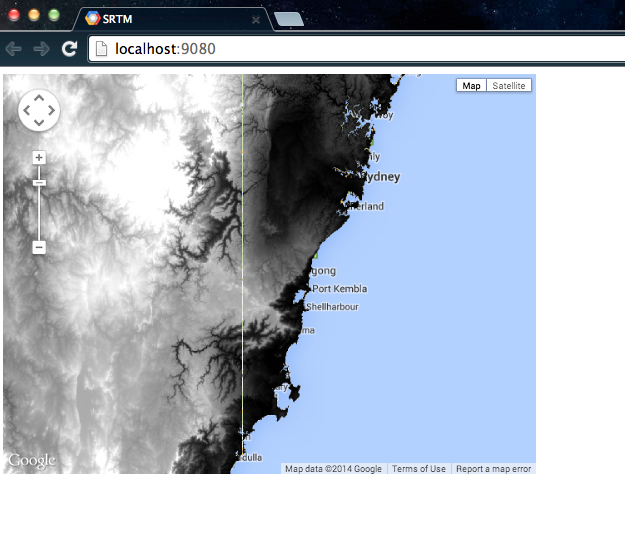
1. Install the IPython Notebook, SciPy and matplotlib packages, using the conda package manager:  
   conda install ipython-notebook scipy matplotlib
2. Create a directory where the IPython Notebooks will live.  
   mkdir -p ~/temp/ipynb  
   cd ~/temp/ipynb  
   [Windows] mkdir %HOMEPATH%\temp\ipynb  
   [Windows] cd %HOMEPATH%\temp\ipynb
3. Download a basic notebook that accesses the Earth Engine Python API using the following command. (Note that despite the line wrapping, this is a single command.)  
   curl -O <https://raw.githubusercontent.com/tylere/IPython-Notebooks/master/ee-demos/basic_tests/Earth_Engine_API_basic_test.ipynb>  
   Curl is not available by default in Windows, so navigate in a browser to: https://github.com/tylere/IPython-Notebooks/blob/master/ee-demos/basic\_tests/Earth\_Engine\_API\_basic\_test.ipynb  
   and then save the "Raw" file link into the temporary directory you previously created..
4. Start the IPython Notebook Server, which will open in a browser and display files and directories below the directory where you start the notebook:  
   ipython notebook
5. Click on the file Earth\_Engine\_API\_Basic\_Test.ipynb, which will open the notebook in a new tab.
6. Look through the commands, and click the play button (near the top-middle of the page) to run each of the notebook cells. Note that IPython Notebooks allow you to run arbitrary code, so do not run untrusted code without first reviewing it!!!  
   If everything is working correctly, you browser window should look similar to the following screenshot.   
   
7. To stop the IPython Notebook server, hit Ctrl-c twice in the terminal where the server was started.
8. If you want to experiment with additional IPython Notebooks, here are a few others to try:  
     
   # IPython In Depth tutorial notebooks  
   curl -L https://github.com/ipython/ipython-in-depth/zipball/master -o ipython-in-depth.zip  
   unzip ipython-in-depth.zip  
     
   # Notebooks presented at SciPy 2014  
   curl -L <https://github.com/tylere/SciPy-2014/zipball/master> -o scipy-ee.zip  
   unzip scipy-ee.zip  
     
   # Exploratory Computing with Python  
   # http://mbakker7.github.io/exploratory\_computing\_with\_python/  
   curl -L <https://github.com/mbakker7/exploratory_computing_with_python/zipball/master> -o exploratory\_computing\_with\_python.zip  
   unzip exploratory\_computing\_with\_python.zip

# Configuring App Engine to use the conda Environment

1. Install Google App Engine for Python, by following the instructions [here](https://developers.google.com/appengine/docs/python/gettingstartedpython27/introduction).
2. Switch to the temporary directory:  
   cd ~/temp
3. Clone the Earth Engine API repository:  
   git clone <https://code.google.com/p/earthengine-api/>
4. Move to the directory for the App Engine sample project.  
   cd earthengine-api/python/examples/AppEngine/hello\_world
5. Update the application name in app.yaml either manually, or by executing the following commands. The application name is arbitrary, and will later be part of the URL that provides access to your app.  
   export APPENGINE\_APP\_ID=test-app  
   mv app.yaml app.yaml.bak  
   sed "s/<your-appid-here>/$APPENGINE\_APP\_ID/g" app.yaml.bak > app.yaml
6. Add a link to your private key file.  
   ln -s $HOME/.keys/privatekey.pem privatekey.pem
7. Update the EE\_ACCOUNT variable in config.py to match your service account email either manually, or by executing the following commands:  
   mv config.py config.py.bak  
   *sed "s/EE\_ACCOUNT = '*[*your-service-account-id@developer.gserviceaccount.com*](mailto:your-service-account-id@developer.gserviceaccount.com)*'/EE\_ACCOUNT = '$MY\_SERVICE\_ACCOUNT'/g" config.py.bak > config.py*
8. Add links to the Python library dependencies.  
   ln -s $HOME/anaconda/envs/ee-python/lib/python2.7/site-packages/ee ee  
   ln -s $HOME/anaconda/envs/ee-python/lib/python2.7/site-packages/oauth2client/ oauth2client  
   ln -s $HOME/anaconda/envs/ee-python/lib/python2.7/site-packages/six.py six.py  
   ln -s $HOME/anaconda/envs/ee-python/lib/python2.7/site-packages/httplib2 httplib2
9. Start up the Google App Engine development server.  
   dev\_appserver.py .
10. [OSX] Start GoogleAppEngineLauncher

Go to Preferences, edit Python path to anaconda/bin/python2.7

1. Open up a browser and navigate to URL where your application is running (typically <http://localhost:8080>). You should see a map with a DEM, you browser window should look similar to the following screenshot.



1. When finished developing, deploy the application  
   appcfg.py update .
2. Browse to the following URL to see the deployed app. [http://APPENGINE\_APP\_ID.appspot.com/](http://appengine_app_id.appspot.com/)

# Troubleshooting

**AttributeError: 'module' object has no attribute 'SignedJwtAssertionCredentials'**

This error is usually caused when the application cannot find the Crypto. Make sure that you can run the following import:

from oauth2client import crypt

**NotImplementedError: PKCS12 format is not supported by the PyCrpto library. Try converting to a "PEM"**

This error can occur if the PEM file does not start with "-----BEGIN". Open the PEM file with a text editor, and remove any lines that occur before "-----BEGIN".

**In Windows, opening up the Anaconda Command Prompt does not set the path correctly.**

As of version 2.0.1 of Anaconda, the batch file that sets the system path does not work correctly. This issue has been reported on the [Anaconda issue tracker](https://github.com/ContinuumIO/anaconda-issues/issues/41). The following code can be used to replace the activate batch files (located in anaconda\Scripts\activate.bat).

@echo off

rem +===========================================================================

rem | Initialisation

rem +===========================================================================

for %%i in ("%~dp0..\envs") do (

set ANACONDA\_ENVS=%%~fi

)

if not "%1" == "" (

if not exist "%ANACONDA\_ENVS%\%1\python.exe" (

echo No environment named "%1" exists in %ANACONDA\_ENVS%

goto :eof

)

set ANACONDA\_ENV\_NAME=%1

set ANACONDA=%ANACONDA\_ENVS%\%1

title Anaconda (%ANACONDA\_ENV\_NAME%^)

) else (

set ANACONDA\_ENV\_NAME=

for %%i in ("%~dp0..") do (

set ANACONDA=%%~fi

)

title Anaconda

)

set ANACONDA\_SCRIPTS=%ANACONDA%\Scripts

set PATH=%ANACONDA%;%ANACONDA\_SCRIPTS%;%PATH%

rem echo Setting path to: %PATH%

if not "%ANACONDA\_ENV\_NAME%" == "" (

echo Activating environment %ANACONDA\_ENV\_NAME%...

set PROMPT=[%ANACONDA\_ENV\_NAME%] $P$G

)

rem vi:set ts=8 sw=4 sts=4 tw=0 expandtab: