Installing App Engine with Google Cloud SDK

For Windows 7, 8.1 & 10

This document contains instructions to get Google App Engine up and running via the lastest method that is supported by Google. Previous methods are being phased out and support will soon stop. All of Google's cloud developer products now sit in a command line tool called Google Cloud SDK (the `gcloud` command).

If you still have the old, standalone App Engine Launcher (the graphical program), it is advised to uninstall it and switch to the Google Cloud SDK method, as described in this document.

These instructions are based on Google's quickstart guide, but with missing steps added and clarifications for the Windows platform.

Google Account

If you haven't already got a Google account, you can register for one here.

Create App Engine Project

Next you'll create your first App Engine project. Start by navigating to https://console.cloud.google.com/start, select email contact preference and agree to Google Cloud Platform's terms and conditions.

You can dismiss the banner at the top of the page regarding signing up for a free trial. There is no need to signup further in order to create a project or give any billing details.

Click on the `Projects` drop down in the menu bar at the top of the page and click `Create project`. Type hello-world as the name of your project. Note that a unique project ID is created on the line below. This is the ID that you will use when working with the `gcloud` command.

Wait around 30 seconds for your new project to be created and for its dashboard to load. Your unique project ID can also be found at the end of the URL on this page.

Python

You should have already installed Python on your machine during the Movie Trailer Project. In case you installed Python 3, you should probably uninstall it and install the latest version

of Python 2 (unless you have a real need for Python3 - they can be installed together - see second note below). This is because Google's Python App Engine Standard Environment only supports Python 2.7. You can download the latest Python 2 package here. Once installed, make sure you can run `python` at the command line and you get a Python version starting with 2 (e.g 2.7.12).

Notes

- Anaconda Python 2 has been tested and will work with Google Cloud SDK.
- You can control which Python executable `gcloud` will use with the environment variable `CLOUDSDK_PYTHON`.

Install Google Cloud SDK

Start by doing step one of the these instructions; download the Cloud SDK installer and then follow the on screen prompts. It's probably best to install it as a `Single User`, as this will install everything in your user home directory. So it will get installed to:

```
C:\Users\<YOUR USERNAME>\AppData\Local\Google\Cloud SDK
```

Let it install with the default settings, then leave "`Start Google Cloud SDK Shell`" and "`Run 'gcloud init' to configure the Cloud SDK`" check boxes ticked. Then click the `Finish` button.

A command prompt will appear that will set up Google Cloud SDK (GCS). You will be asked to login to your Google Account (or if you are already logged in, give GCS access to your account).

Since you have only one project, this project will be your default project. As you add more projects, you can either reset the default (see `gcloud topic configurations` to learn more) to another project or specify which project you want to operate on with the `--project <project id>` option to `gcloud` commands.

Location of configurations and credentials

If you ever need to delete your GCS configurations and login credentials, these are stored in the following location:

```
C:\Users\<YOUR USERNAME>\AppData\Roaming\gcloud
```

Install App Engine Python Extensions

Since the `gcloud` command can manage all of Google's cloud development platforms, not everything is installed by default. You need to install the Python version of App Engine that is used throughout this Nanodegree.

You can list currently installed and available components with the command:

```
gcloud components list
```

Go ahead and install the App Engine Python Extensions with the command:

```
gcloud components install app-engine-python
```

To find out more information about the available components, navigate your browser here.

Your first App Engine webapp

If you have Git installed (highly recommended), you can clone Google's Python sample code repository with the command:

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples
```

If not, you can download a zip archive from here:
https://github.com/GoogleCloudPlatform/python-docs-samples/archive/master.zip

Once you have the code, change your current directory with the following command:

```
cd python-docs-samples\appengine\standard\hello_world
```

To run App Engine locally, you use the command `dev_appserver.py` - a Python script program. If this file is in your user PATH environment variable, the next command to run is:

```
dev_appserver.py .
```

Notice the dot (or full stop) on the end of the command. That's very important and tells the command to use the current directory to search for a file called `app.yaml`. This file configures the App Engine webapp. Giving `dev_appserver.py` this file directly as an argument will also work.

If you find the command `dev_appserver.py` is not found, either add it's location to your path (instructions here), or pass the file to Python with the command:

```
python "C:\Users\<YOUR USERNAME>\AppData\Local\Google\Cloud
SDK\google-cloud-sdk\bin\dev_appserver.py" .
```

(all on one line mind. It's only wrapped here as there is limited space).

Firewall notice

You may get a Windows Security Alert asking to allow the Python program through the firewall. This is fine to allow, just make sure only the `Private networks` option is selected and then press the `Allow access` button.

Browse your first app

To look at the web output of the your first app, go to this address in your browser:

`http://localhost:8080`

Each App Engine app also has an admin server at the following address:

`http://localhost:8000`

This has many useful features, but one you will be using a lot is the Datastore Viewer. There's not a lot to see at the moment, but come back to this page when you start putting data into the Datastore:

`http://localhost:8000/datastore`

Make a change

Open up the file `main.py` in a text editor (not Word though). Line 21 contains the text that is currently output to the browser. Change it to something else, like:

```
self.response.write('Goodnight, World!')
```

The reload the page `http://localhost:8080` and you should see the updated message.

Stop the server

To stop the local server, press the key combination `Ctrl+C`.

Deploy your app to the cloud

You can deploy your app to the cloud so anyone in the world can view it with the following command:

```
gcloud app deploy
```

This will deploy your default project in the current configuration. You can also specify another project or version of a project. More details on this options are here.

Once deployed, your app will be available to view and use at the address `http://[YOUR_PROJECT_ID].appspot.com`. Or issue the command:

```
gcloud app browse
```

which will open up your default browser to the public URL for the project. This may not work though, depending on your configuration.

Updating indexes

As you develop your app and it gets more complex, the local `index.yaml` file will get populated with the indexes needed by your app. When you deploy your app, you might encounter errors with the deployed app, as it takes time for the indexes to be built.

A way to speed things up is it manually deploy the `index.yaml` file with:

```
gcloud app deploy index.yaml
```

Further details on updating indexes are available here.

Review the Hello World code

Please review the code explanation of both the `main.py` and `app.yaml` files used in this example, in order to better understand what is going on.

Feedback

If you have any comments, questions or suggestions about this document, please mention @swooding on the Forums.

Credits

Lead author….	Steven Wooding
Additional content….	Renee MacDonald
Reviewed by….	Full Stack Nanodegree Mentors