

Homework #7 Google Cloud Platform

This semester we are allowing all students to explore cloud computing as offered by the Google Cloud Platform. Using the instructions below one can establish a website using Google App Engine. Once established, you will be able to move your PHP program developed for Assignment #6 to your Google App Engine website and have it execute there.

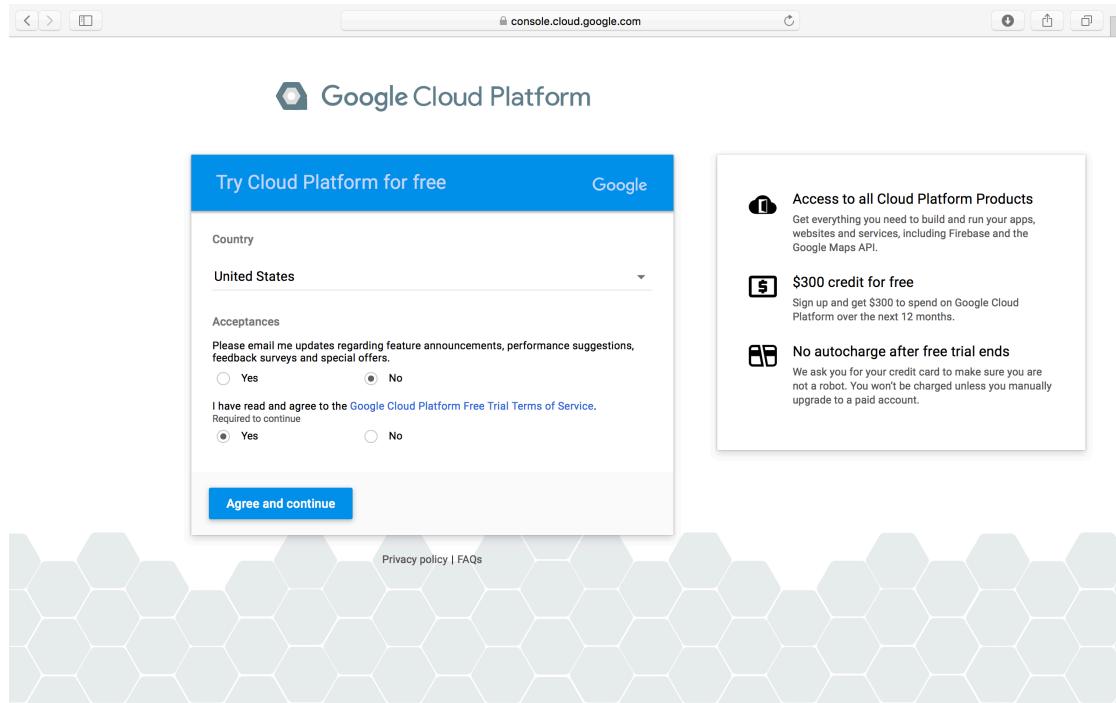
1. Sign up for Google Cloud Platform

To sign up for the Free Trial, with a \$300 credit, you need a credit card. Unfortunately, an American Express or other pre-paid Gift card will not work with Google Cloud.

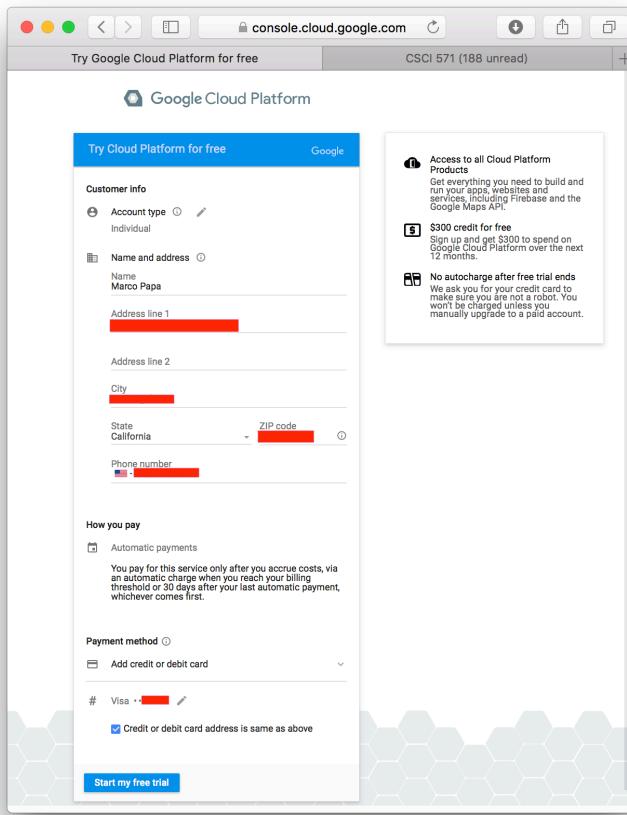
To sign up go to:

<https://console.cloud.google.com/freetrial?pli=1&page=0>

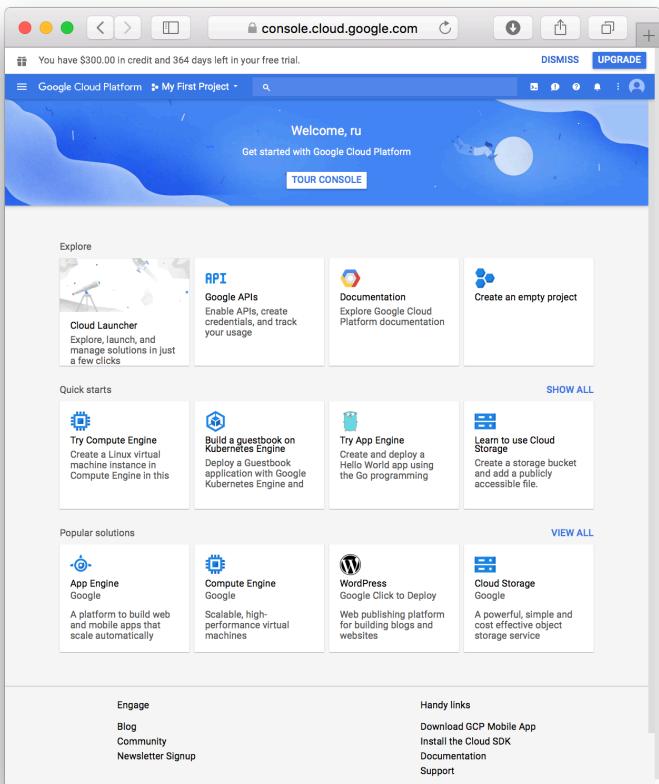
In the Try Cloud Platform for free page, select **Yes** under “I have read and agree to the Google Cloud Platform Free Trial terms of Service” and click on **Agree and continue**.



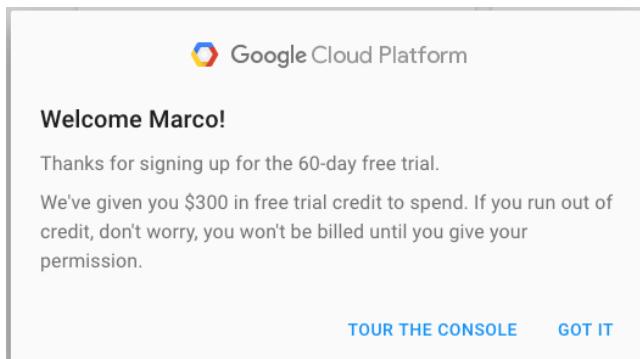
Select **Account type Individual**. Follow the instructions to enter your account data. You should not be using your @usc.edu e-mail account for your primary contact e-mail address, but instead use your @gmail.com address and finish by clicking **Start my free trial**. You will have to provide a credit or debit card.



After you are signed up, you will see the message “Creating project. This may take a few moments.” You will then be redirected to the **Dashboard** of the **Google Developer Console**.



A message will pop up indicating you signed up for the 365-day free trial, and have been given a \$300 free trial credit.



If you previously developed any projects using Google APIs, you will find them listed.

2. Download the App Engine SDK for PHP

The **Google App Engine SDK for PHP** is available for Windows, Mac OS X and Linux platforms.

Go to the URL:

<https://cloud.google.com/appengine/docs/standard/php/download>

Click on “**Download and install the original App Engine SDK for PHP**,” as highlighted in the picture below.

Download the Cloud SDK SEND FEEDBACK

[Python](#) | [Java](#) | **PHP** | [Go](#)

The preferred tooling for managing your App Engine applications in PHP is [Google Cloud SDK](#). Cloud SDK includes a local development server as well as the `gcloud` command-line tooling for deploying and managing your apps.

By downloading, you agree to be bound by the [Terms](#) that govern use of all SDKs for App Engine.

Installing Cloud SDK for PHP

Install Google Cloud SDK along with the corresponding `gcloud` component and other related tools and dependencies.

1. Install and initialize Google Cloud SDK:
DOWNLOAD AND INSTALL
- Note:** Avoid using a package manager such as `apt` or `yum` to install Cloud SDK.
2. Run the following command to install the `gcloud` component that includes the App Engine extension for PHP:
`gcloud components install app-engine-php`
3. Download and install Git for access to code, samples, libraries, and tools in [GitHub](#):
DOWNLOAD AND INSTALL GIT
4. For Linux, install PHP version 5.5.34:
DOWNLOAD AND INSTALL

Updating Cloud SDK for PHP

If you already have the Google Cloud SDK installed, you can run the `gcloud components update` command to update the SDK to the latest version.

1. Review the release notes to see what updates are available:
 - [Google Cloud SDK](#)
 - [PHP runtime in App Engine](#)
2. Run the following command to update all the installed Google Cloud SDK components, including the App Engine extension for PHP:
`gcloud components update`

If you have previously used the original App Engine SDK and need to continue to use the `appcfg` command, you can instead download and install the original App Engine SDK by following the instructions below. Cloud SDK does **not** include the `appcfg` command.

Download and install the original App Engine SDK for PHP.

After you click on the hyperlink, a new section will appear below with LINUX, MAC OS X and WINDOWS tab installations. Select your platform tab.

Optional: Download and install the original App Engine SDK for PHP

LINUX	MAC OS X	WINDOWS
1. Download the App Engine SDK for PHP:		
DOWNLOAD		
Version:	Size:	SHA1 Checksum:
1.9.67 - 2018-02-13	70.5 MB	783d21790145d90346bfd4a4c936b6c93481605

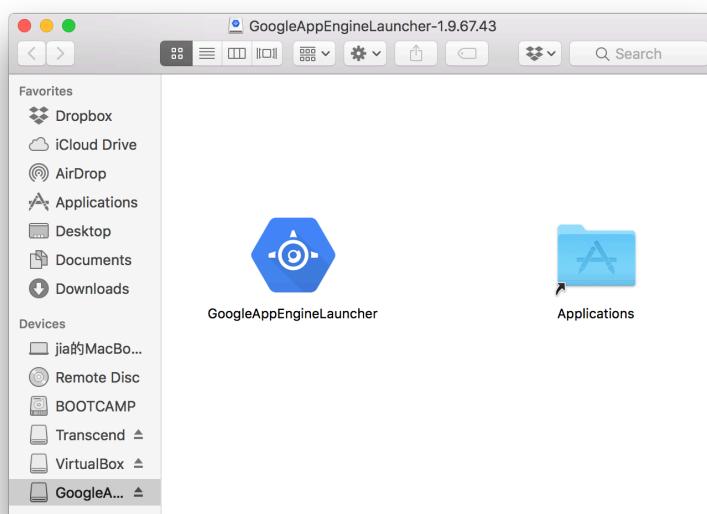
Click the **DOWNLOAD** button to download the package for your Platform to your local machine. The rest of this installation will show the steps needed for installing the SDK on the Mac OS X and Windows platforms.

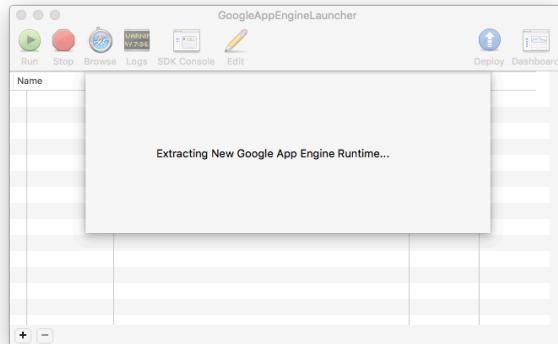
3. Install the App Engine SDK for PHP

Installing on Mac OS X

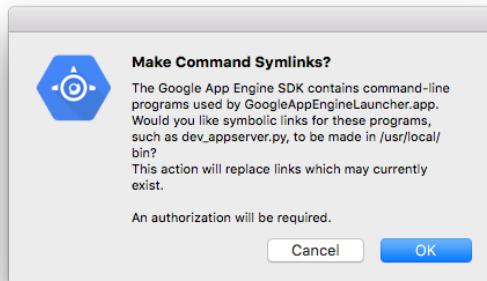
To install the SDK on Mac OS X:

1. In the Finder, click **Go > Downloads** to open the Downloads folder.
2. Double click the App Engine SDK file you downloaded (*GoogleAppEngineLauncher-1.9.67.dmg*) to open it, then drag the **GoogleAppEngineLauncher** icon over to the Applications folder. You can now “eject” the volume *GoogleAppEngineLauncher-1.9.67.43*.
3. Double-click **GoogleAppEngineLauncher** in the Application folder.

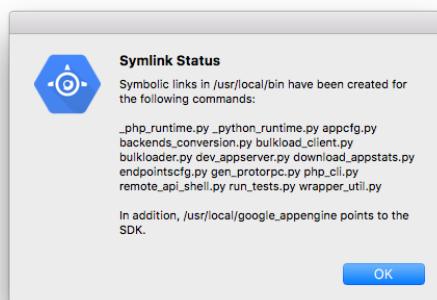




4. When prompted to *Make Command Symlinks*, click **OK**. The symlinks allow you to run important SDK command-line tools in any terminal window.



Important: The GoogleAppEngineLauncher is a convenient UI-based tool for running and deploying App Engine apps, but it *does not* provide all the features you'll need. You will need to use the command-line equivalent, `appcfg.py`, for many of the features you'll want to use.



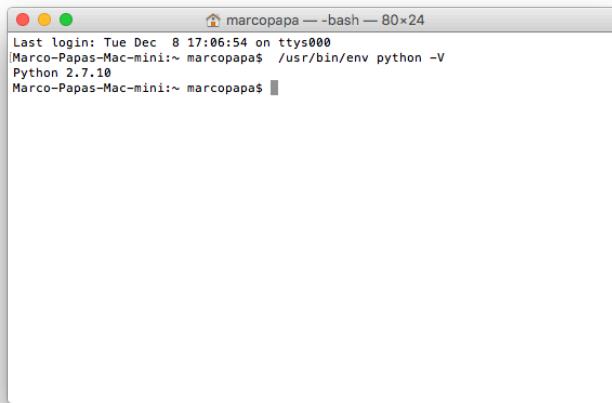
5. Notice that the installation process above unpacks the contents of the App Engine SDK at the location:

```
/usr/local/google_appengine
```

6. The App Engine PHP SDK requires Python 2.7, which is installed by default on Mac OS X 10.6 (Lion) or later. Verify your Mac's Python installation using the following command:

```
/usr/bin/env python -V
```

If the output looks like **Python 2.7.<number>** then you already have the correct Python version installed. Otherwise you can download and install Python 2.7 from [the Python web site](#). If you are using Mac OS X Sierra, you will have Python 2.7.10, as shown below:



A screenshot of a Mac OS X terminal window titled "marcopapa — bash — 80x24". The window shows the following text:
Last login: Tue Dec 8 17:06:54 on ttys000
Marco-Papas-Mac-mini:~ marcopapa\$ /usr/bin/env python -V
Python 2.7.10
Marco-Papas-Mac-mini:~ marcopapa\$

Installing on Windows

To install the SDK on Windows:

1. Double-click the SDK file you downloaded (*GoogleAppEngine-1.9.61.msi*) and follow the prompts to install the SDK.
2. You will need Python 2.7 to use the App Engine PHP SDK, because the [Development Server](#) is a Python application. Download Python 2.7.5 (don't use a higher version) from [the Python web site](#).

Note: The PHP SDK includes binaries for the PHP 5.4 runtime, including all [enabled extensions](#), so there is no need to download PHP separately for the purposes of developing with App Engine -- you just need Python.

4. Create application using GoogleAppEngineLauncher

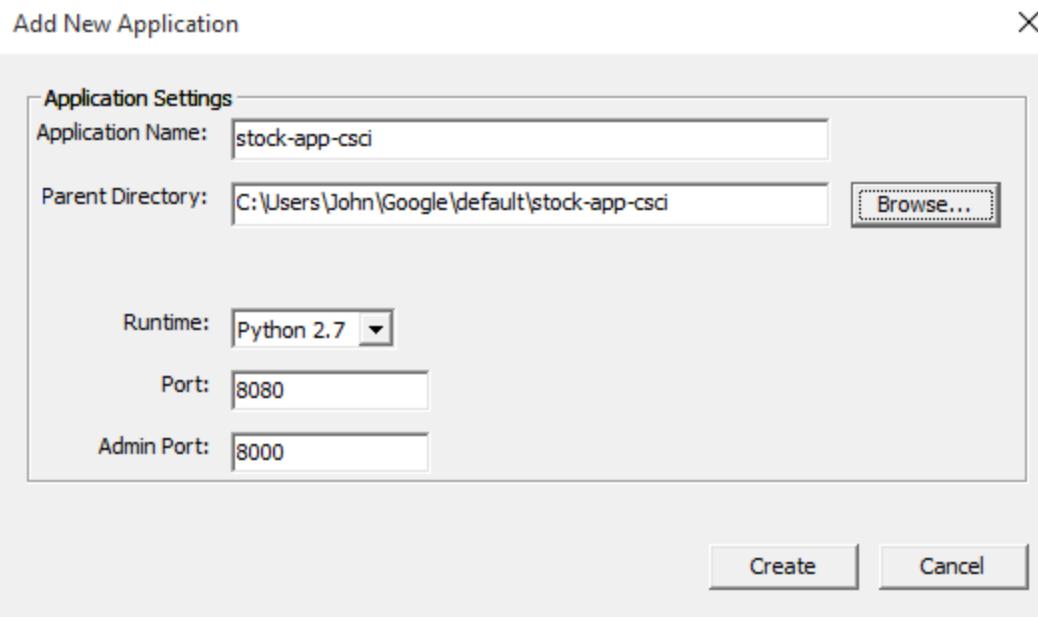
You can use the Google App Engine Launcher, which is installed as part of the App Engine SDK for PHP.

Run the Google App Engine Launcher:

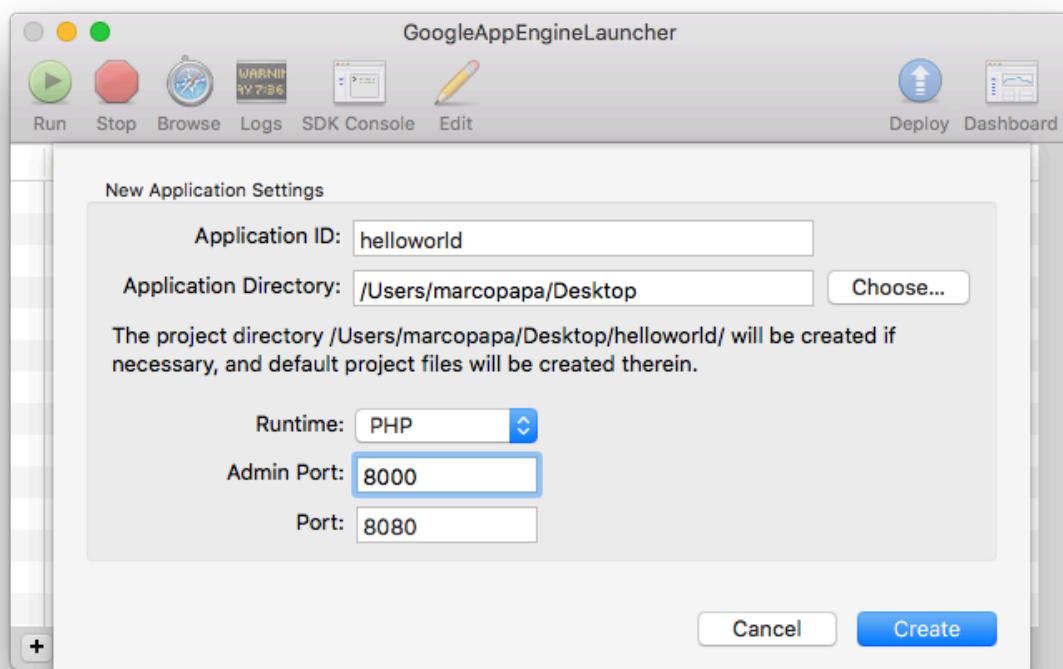


Google App Launcher (Windows)

Invoke File -> Create (Windows) or File -> New Application (OS X). Or use File -> Add Existing Application (OS X) if the app already has been created.



Add New Application (Windows) [Note Runtime should be PHP]

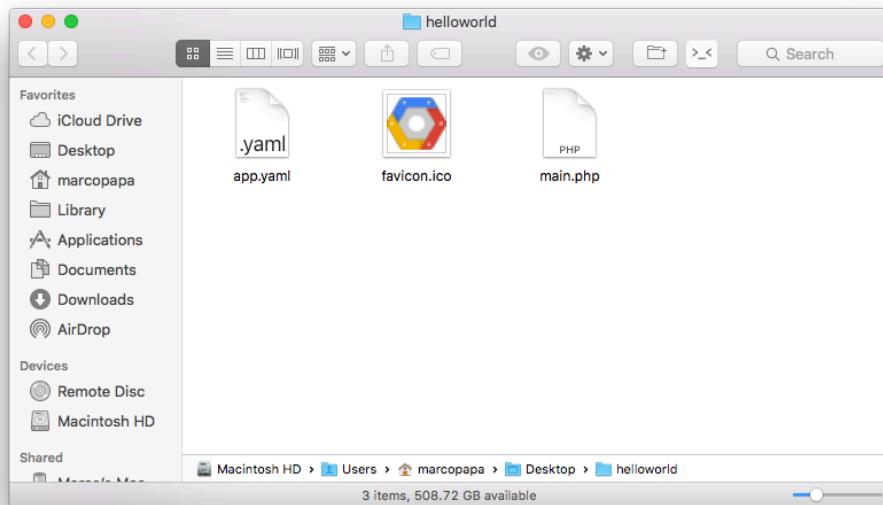


Add New Application (Mac)

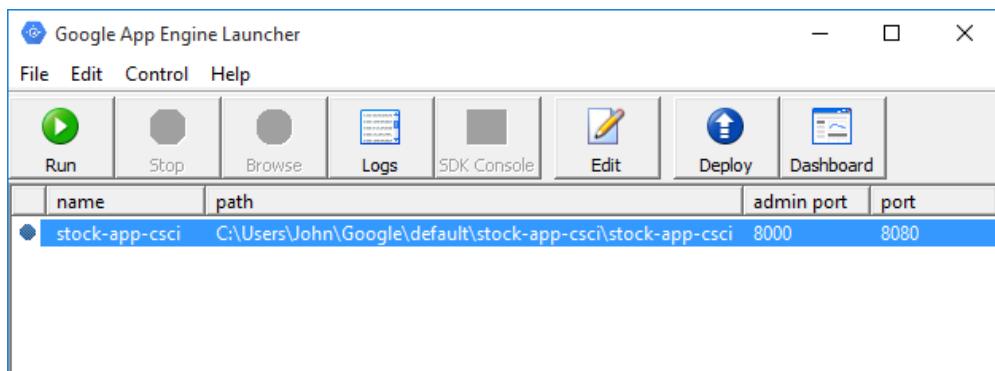
You will need to enter the following fields:

- Application Name or ID: use a new app name, like helloworld (this will be the name of a new folder)
- Parent of Application Directory: use the full path to the parent folder, for example:
 - /Users/yourname/Desktop/ in OS X
 - C:\Users\yourname\Desktop\ in Windows
- Runtime: **should be PHP**
- Port: defaults to 8080
- Admin Port: defaults to 8000

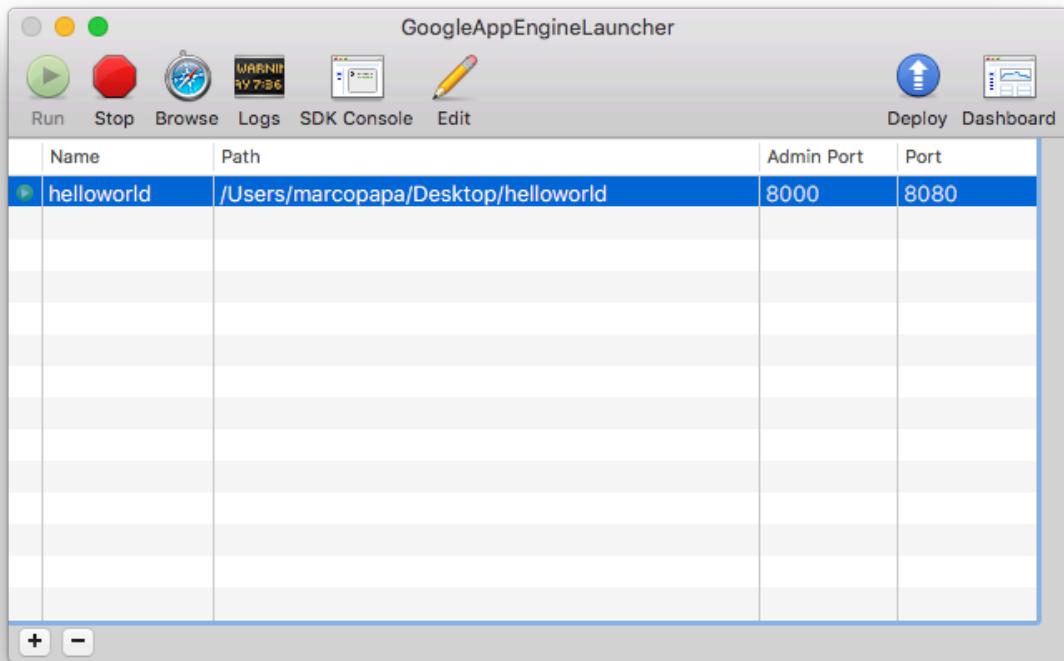
Click **Create**. The folder named helloworld will be created, containing 3 files, as shown below.



Now you can run the application by clicking the **Run** button:



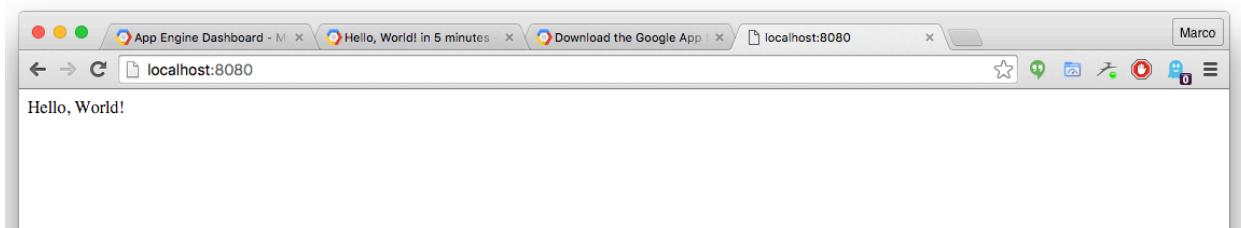
Run application locally (Windows)



Run Application locally (OS X)

The web server is now running, listening for requests on port 8080. You can test the application by visiting the following URL in your web browser:

<http://localhost:8080/>



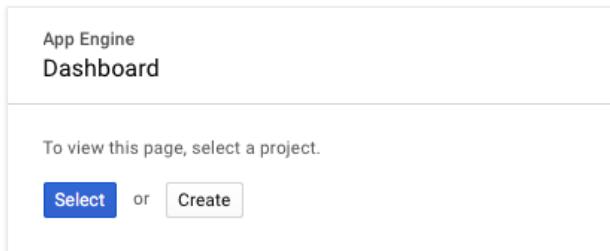
5. Creating a Project and Application

Sign in to App Engine using your Google account. If you do not have a Google account, you can [create a Google account](#) with an email address and password. See section 1.

Sign up for Google Cloud Platform.

1. Before you can deploy your apps to the App Engine standard environment, you typically need to create or set up the following:
 - a. A Cloud Platform project
 - b. An App Engine application
2. To deploy your apps to a Cloud Platform project, you must create a corresponding App Engine application, which defines the location from where you want your App Engine services run. To create a Cloud Platform project and App Engine application:
 - a. Go to the App Engine page:

<https://console.cloud.google.com/projectselector/appengine/create?lang=php&st=true>



- b. Select or Create a Cloud Platform project. If you have not created a Cloud Platform project yet, click **Create**.

New Project

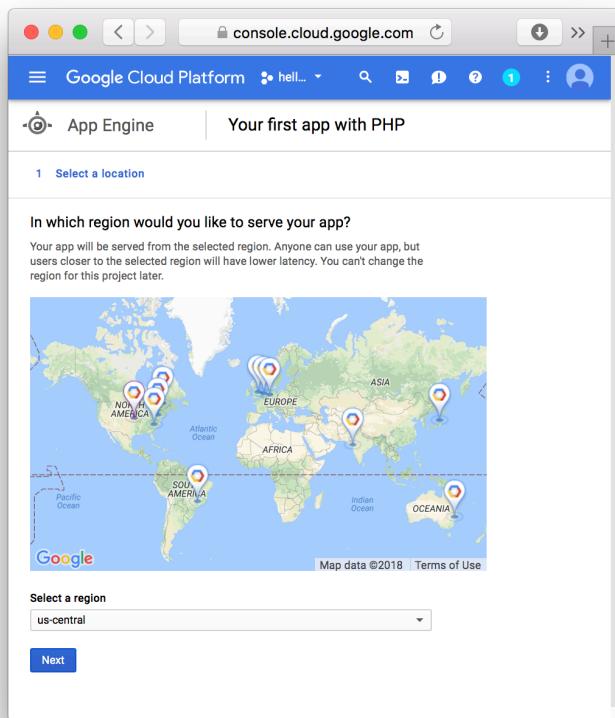
You have 11 projects remaining in your quota. [Learn more.](#)

Project name ?

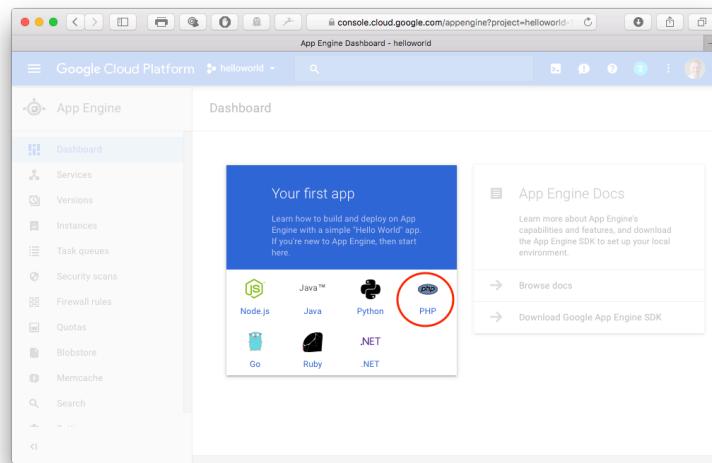
Your project ID will be helloworld-197100 ? [Edit](#)

[Create](#) [Cancel](#)

- c. In the **New Project** dialog enter a Project name. Notice the **project ID**. Accept the generated project ID or supply your own ID by clicking **Edit**. *This project ID is used as the App Engine application ID*. Note that this ID can only be used once: if you subsequently delete your project, you won't be able to re-use the ID in a new project. Click **Create**.



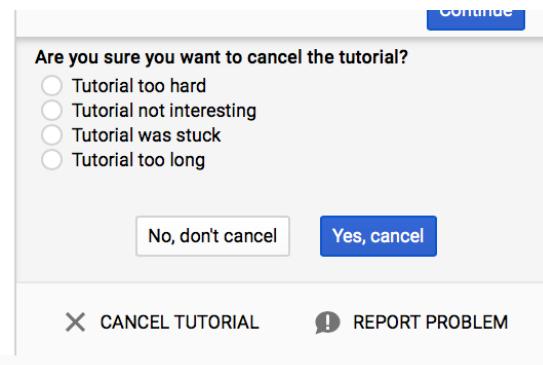
3. You can now **Select a location** for your deployed app. Keep the suggested **us-central**, and click **Next**.
4. The system displays “Preparing you App Engine services.”



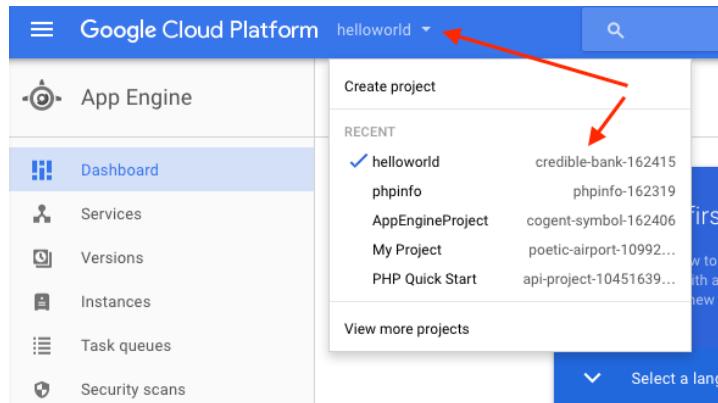
5. Select PHP in the Select a language dropdown.

The screenshot shows the 'App Engine Quickstart' tutorial page. On the left, there's a sidebar with various options like Services, Versions, Instances, etc. In the center, it says 'Let's get started.' with a right-pointing arrow. To the right, there's a detailed description of the tutorial, which includes steps for building and running a 'Hello, world!' app using the gcloud command. At the bottom right of the main content area, there's a blue 'Continue' button.

6. Your project and basic App Engine app (helloworld) have been created. You can start the Tutorial by clicking Continue, or you can simply CANCEL TUTORIAL.



- Click **YES, CANCEL**. If you click in the project dropdown, you will see your Project / app name and its corresponding **project ID** from step 5.3.b. Make a note of it.



- You are now ready to Upload and Deploy your finished application to Google App Engine by invoking the following command. This opens a browser window for you to sign in using your Google account. You'll be providing the **project ID** as the argument for **-A**.

Note: Make sure you run this command in the “parent” directory where you created helloworld. Otherwise, give the full path to the helloworld folder.

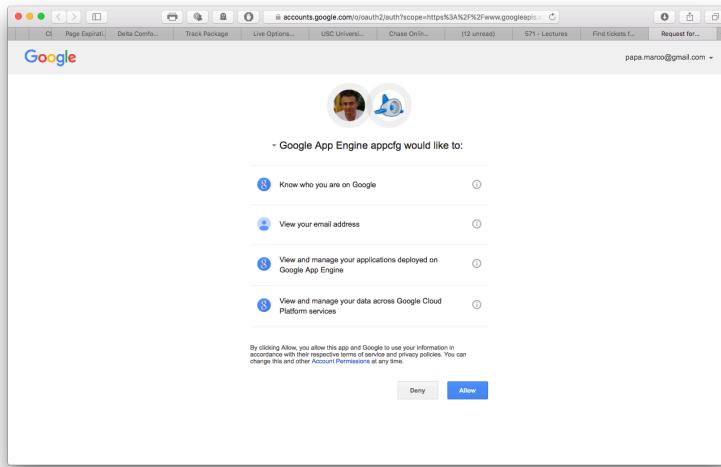
In OS X enter:

```
/usr/local/google_appengine/appcfg.py -A YOUR_PROJECT_ID update helloworld/
```

In Windows enter:

```
python appcfg.py -A YOUR_PROJECT_ID update helloworld/
```

- The first time you execute this script, you may be asked to *Allow* the appcfg.py script, as shown below:



10. You will receive a message in the browser that “The authentication flow has completed.” Your app is now deployed and ready for use! If everything went well, you will receive a message “Deployment successful” on the terminal console, as shown below.

```
Marco-Papas-Mac-mini:Desktop marcopapa$ sudo /usr/local/google_appengine/appcfg.py -A nodal-shift-147415 update helloworld/
Password:
09:05 AM Application: nodal-shift-147415 (was: helloworld); version: 1
09:05 AM Host: appengine.google.com
09:05 AM Starting update of app: nodal-shift-147415, version: 1
09:05 AM Getting current resource limits.
[Your browser has been opened to visit:
 https://accounts.google.com/o/oauth2/auth?scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email&redirect_uri=http%3A%2F%2Flocalhost%3A8090%2F&response_type=code&client_id=550516889912.apps.googleusercontent.com&access_type=offline
If your browser is on a different machine then exit and re-run this
application with the command-line parameter
--noauth_local_webserver
Authentication successful.
09:06 AM Scanning files on local disk.
09:06 AM Cloning 7 application files.
09:06 AM Compilation starting.
09:06 AM Compilation completed.
09:06 AM Starting deployment.
09:07 AM Checking if deployment succeeded.
09:07 AM Deployment successful.
09:07 AM Checking if updated app version is serving.
09:07 AM Completed update of app: nodal-shift-147415, version: 1
Marco-Papas-Mac-mini:Desktop marcopapa$
```

11. The full URL for your application is http://<YOUR_PROJECT_ID>.appspot.com/.

6. GCP Cloud SDK

- As an alternative to the **App Engine SDK for PHP** described in section 3, you can use the **Cloud SDK**, a command line interface for Google Cloud Platform:

<https://cloud.google.com/sdk/>

2. The Cloud SDK is available for Linux, Ubuntu, CentOS, Mac OS X and Windows. Quickstarts for each platform are available here:

<https://cloud.google.com/sdk/docs/quickstarts>

3. Documentation for using PHP on Google Cloud Platform is available here:

<https://cloud.google.com/php/>

4. When using App Engine with PHP make sure you select the “Standard” environment (limited free tier) and not the “Flexible” environment (no free tier). In the Quickstart, select **USE STANDARD ENVIRONMENT**.

PHP

Quickstarts for PHP

SEND FEEDBACK

Google Cloud Platform offers four hosting options for PHP developers. It's important to consider which environment is closest to your needs, but also keep in mind that many apps will need to take advantage of two or more of these hosting options.

Google App Engine is a great place to start. App Engine is a platform-as-a-service for web applications and mobile backends. It can automatically scale your application in response to traffic. You don't have to provision or maintain servers. Alternatively, if you're interested in using virtual machines or containers directly you can read the tutorials on using [Compute Engine](#) or [Container Engine](#).

App Engine has two environments: the [standard environment](#) and the [flexible environment](#). Each environment has tradeoffs. You can read about the general differences between the two environments in the [App Engine documentation](#). For PHP developers, the key differences are:

Standard environment	Flexible environment
Generally available.	Beta.
Limited free tier.	No free tier.
PHP 5.5 only.	PHP 5.6, 7.0 & 7.1.
A subset of PHP extensions are available.	Runtime can be extended to install any PHP extension that works on Debian Linux.
No command-line access to instances.	Command-line access to instances using Cloud Shell and SSH.
Google-managed web server	Configurable NGINX web server.
No filesystem writes.	Writable filesystem.
No ability to run deployment scripts.	Ability to run deployment scripts using <code>composer.json</code> .

USE STANDARD ENVIRONMENT **USE FLEXIBLE ENVIRONMENT**

5. The “*Quickstart for PHP App Engine Standard Environment*” page at:

<https://cloud.google.com/appengine/docs/standard/php/quickstart>

6. Provides all the steps needed to do all the following:

- Create a Cloud Platform project
- Download the Google Cloud SDK and initialize the `gcloud` tool
- Download the Hello World sample app from github

- Test the application locally on your laptop/notebook
- Deploy the same app to App Engine
- View the sample app on a browser
- Clean-up to stop billing

7. You **Create a new Cloud Platform project and App Engine app** like shown before using the Cloud Platform Console:

https://console.cloud.google.com/projectselector/appengine/create?lang=php&st=true&_ga=2.54455777.-810889425.1501011628

When prompted, select the region where you want your App Engine application located. After your App Engine application is created, the **Dashboard** opens.

8. Download and install the **Google Cloud SDK for App Engine** and initialize the `gcloud` tool.

<https://cloud.google.com/appengine/docs/standard/php/download>

9. Install the latest Cloud Tools version for your platform (Mac OS, Windows):

<https://cloud.google.com/sdk/docs/>

Extract the file on your local file system. Run `gcloud init` to initialize the SDK:

```
./google-cloud-sdk 2/bin/gcloud init
```

10. Run a command to install the cloud component that includes the App Engine extension for PHP:

```
./google-cloud-sdk 2/gcloud components install app-engine-php
```

11. Alternatively, install all the components (including the App Engine extension for PHP), by running the command:

```
./google-cloud-sdk 2/install.sh
```

```

Marco-Papas-Mac-mini:google-cloud-sdk 2 marcopapa$ ./install.sh
Welcome to the Google Cloud SDK!
To help improve the quality of this product, we collect anonymized usage data
and associated backtraces when crashes are encountered; additional information
is available at <https://cloud.google.com/sdk/usage-statistics>. You may choose
to opt out of this collection now (by choosing 'N' at the below prompt), or at
any time in the future by running the following command:
  cloud config set disable_usage_reporting true
Do you want to help improve the Google Cloud SDK (Y/n)? n

Your current Cloud SDK version is: 176.0.0
The latest available version is: 176.0.0

Components
+-----+-----+-----+-----+
| Status | Name | ID | Size |
+-----+-----+-----+-----+
| Not Installed | App Engine Go Extensions | app-engine-go | 97.7 MiB |
| Not Installed | Cloud Bigtable Command Line Tool | cbt | 4.0 MiB |
| Not Installed | Cloud Bigtable Emulator | bigtable | 3.5 MiB |
| Not Installed | Cloud DataLab Command Line Tool | databal | < 1 MiB |
| Not Installed | Cloud Datastore Emulator | cloud-datastore-emulator | 17.7 MiB |
| Not Installed | Cloud Datastore Emulator (Legacy) | cloud-datastore-emulator-legacy | 38.0 MiB |
| Not Installed | Cloud Pub/Sub Emulator | pubsub-emulator | 33.2 MiB |
| Not Installed | Emulator Reverse Proxy | emulator-reverse-proxy | 14.5 MiB |
| Not Installed | Google Container Local Builder | container-builder-local | 3.7 MiB |
| Not Installed | Google Container Registry's Docker credential helper | docker-credential-gcr | 2.2 MiB |
| Not Installed | gcloud Alpha Commands | alpha | < 1 MiB |
| Not Installed | gcloud Beta Commands | beta | < 1 MiB |
| Not Installed | gcloud Java Extensions | app-engine-java | 116.0 MiB |
| Not Installed | gcloud app PHP Extensions | app-engine-php | 21.9 MiB |
| Not Installed | gcloud app Python Extensions | app-engine-python | 6.2 MiB |
| Not Installed | kubectl | kubectl | 15.9 MiB |
| Installed | BigQuery Command Line Tool | bq | < 1 MiB |
| Installed | Cloud SDK Core Libraries | core | 5.9 MiB |
| Installed | Cloud Storage Command Line Tool | gsutil | 3.0 MiB |

To install or remove components at your current SDK version [176.0.0], run:
$ gcloud components install COMPONENT_ID
$ gcloud components remove COMPONENT_ID

To update your SDK installation to the latest version [176.0.0], run:
$ gcloud components update

==> Source [/Users/marcopapa/Desktop/google-cloud-sdk/2/completion.bash.inc] in your profile to enable shell command completion for gcloud.
==> Source [/Users/marcopapa/Desktop/google-cloud-sdk/2/path.bash.inc] in your profile to add the Google Cloud SDK command line tools to your $PATH.

For more information on how to get started, please visit:
https://cloud.google.com/sdk/docs/quickstarts

Marco-Papas-Mac-mini:google-cloud-sdk 2 marcopapa$ 

```

12. Download the Hello World app from Github:

```
git clone -b phase0-helloworld
https://github.com/GoogleCloudPlatform/appengine-php-guestbook.
git helloworld
```

13. Test the app on your local machine:

```
dev_appserver.py app.yaml
```

14. Visit <http://localhost:8080> in your browser to view the app.

15. Deploy your app to app engine:

```
gcloud app deploy
```

```

Marco-Papas-Mac-mini:helloworld2 marcopapas$ gcloud app deploy
ERROR: (gcloud.app.deploy) Your current active account [papa.marco@gmail.com] does not have any valid credentials
Please run:

$ gcloud auth login

to obtain new credentials, or if you have already logged in with a
different account:

$ gcloud config set account ACCOUNT

to select an already authenticated account to use.
[Marco-Papas-Mac-mini:helloworld2 marcopapas$ gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&prompt=select_account&
esponse_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2Fwww.googleapis.com%2Fauth%2F
userinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fa
ppengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts
.reauth&access_type=offline

WARNING: 'gcloud auth login' no longer writes application default credentials.
If you need to use ADC, see:
  gcloud auth application-default --help

You are now logged in as [papa.marco@gmail.com].
Your current project is [quickstart-php-183415]. You can change this setting by running:
  $ gcloud config set project PROJECT_ID
[Marco-Papas-Mac-mini:helloworld2 marcopapas$ gcloud app deploy
You are about to deploy the following services:
- quickstart-php-183415/default/20171019t090117 (from [/Users/marcopapa/Desktop/helloworld2/app.yaml])
  Deploying to URL: [https://quickstart-php-183415.appspot.com]

Do you want to continue (Y/n)? Y

Beginning deployment of service [default]...
Some files were skipped. Pass '--verbosity=info' to see which ones.
You may also view the gcloud log file, found at
[/Users/marcopapa/.config/gcloud/logs/2017.10.19/09.01.16.987128.log].
[ Uploading 6 files to Google Cloud Storage

File upload done.
Updating service [default]...done.
Deployed service [default] to [https://quickstart-php-183415.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse
[Marco-Papas-Mac-mini:helloworld2 marcopapas$ gcloud app browse
Opening [https://quickstart-php-183415.appspot.com] in a new tab in your default browser.
Marco-Papas-Mac-mini:helloworld2 marcopapas$ ]

```

16. View your application in the cloud. Launch your browser with the app at [http://\[YOUR_PROJECT_ID\].appspot.com](http://[YOUR_PROJECT_ID].appspot.com), running the command

`gcloud app browse`

17. Clean up. To avoid incurring charges, **delete your Cloud Platform project** to stop billing on all resources.

7. Check PHP Info

To find out the capability and installed options of the PHP component on Google Engine, you need to be able to execute the `phpinfo()` API. Unfortunately, this feature is turned off by default on the Google Cloud Platform.

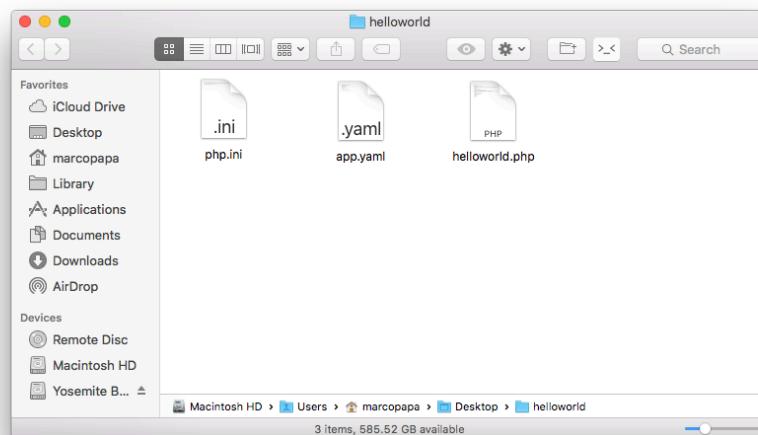
To enable `phpinfo()`, you need to create a file named `php.ini` inside the `helloworld` folder, containing the following:

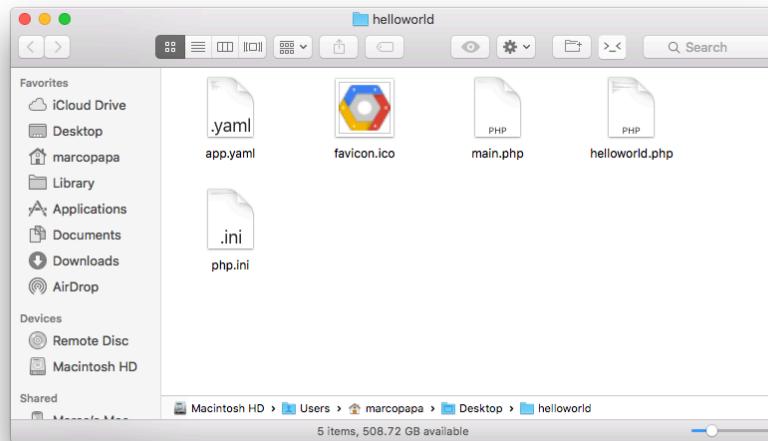
```
google_app_engine.enable_functions = "phpinfo, php_uname, php_sapi_name,  
phpversion"  
  
google_app_engine.enable_curl_lite = "1"
```

Then update your `main.php` script with the code:

```
<html>  
  <head>  
    <title>PHP Test</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  
  </head>  
  <body>  
    <h1>PHP Test</h1>  
    <p>  
      <b>An Example of PHP in Action</b>  
      <?php date_default_timezone_set('America/Los_Angeles');?>  
      <?php echo "The Current Date and Time is: "; echo date("g:i A l, F j Y.");?>  
    </p>  
    <h2>PHP Information</h2>  
    <p>  
      <?php phpversion(); ?>  
      <?php phpinfo(); ?>  
    </p>  
  </body>  
</html>
```

The `helloworld` folder will now contain 3 (or 4) files, as shown below (this assumes you have renamed `main.php`, as `helloworld.php` in `app.yaml`).





You can now “update” your app by invoking the `appcfg.py` script from section 5.4 above. If all goes well, you’ll see the following PHP Info on the new home page:

PHP Version 5.5.38



System	Linux unknown 4.4.0-66-generic #87~14.04.1-Ubuntu SMP Fri Mar 3 17:32:36 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
Build Date	Sep 28 2017 10:04:45
Configure Command	configure
Server API	Google AppEngine PHP Runtime SAPI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	/usr/local/lib
Loaded Configuration File	/base/data/home/apps/s~quickstart-php-183415/20171019t090117.404915797789503662/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API20121212,TS
PHP Extension Build	API20121212,TS
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	ftps, compress.zlib, file, ftp, zip, php, redirect, http, https, gs
Registered Stream Socket Transports	tcp, ssl
Registered Stream Filters	zlib.* , convert.iconv.* , mcrypt.* , mdecrypt.* , string.rot13 , string.toupper , string.tolower , string.strip_tags , convert.* , consumed , dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.5.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies

Powered By



8. Set up Node.js on Google Cloud Platform (GCP)

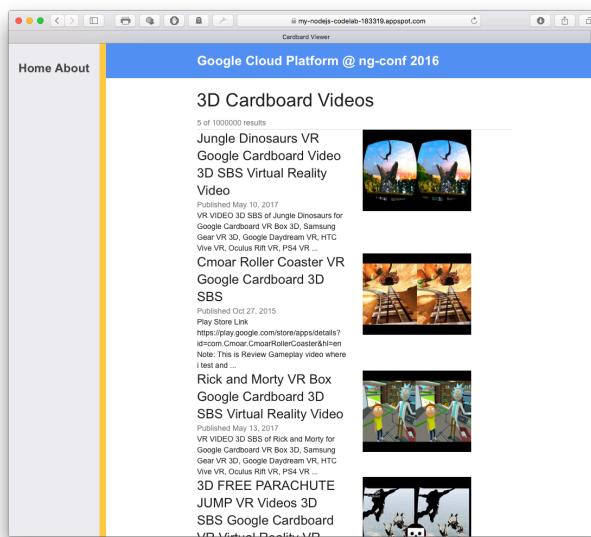
The best way to least how to properly set up Node.js on GCP is to follow the tutorial titled *"Build a Node.js & Angular Web App using Google Cloud Platform,"* available on Google's CodeLabs at:

<https://codelabs.developers.google.com/codelabs/cloud-cardboard-viewer/index.html#0>

Following the tutorial takes about 50 minutes, and it is well worth it.

The screenshot shows the 'Build a Node.js & Angular 2 Web App using Google Cloud Platform' code lab. On the left, a vertical navigation bar lists steps from 1 to 10: Overview, Setup and Requirements, Get the sample code, Try out the sample, Debug, Deploy, Monitor & Debug, Scale, Congratulations!, and Learn More. Step 1 is highlighted. The main content area is titled '1. Overview'. It contains a brief description: 'In this codelab, you'll learn how to deploy, monitor, debug, and scale a Node.js & Angular 2 web application on Google Cloud Platform.' Below this is a section titled 'What you'll learn' with four bullet points: 'How to deploy to App Engine Flexible Environment', 'How to monitor the Node.js app from the Cloud Console', 'How to debug the Node.js app from the Cloud Console', and 'How to automatically scale the Node.js app'. Another section, 'What you'll need', lists requirements: Node.js, NPM, The sample code, and Familiarity with standard Linux text editors such as Vim, EMACs or Nano. A timer in the top right corner shows '48 min remaining'.

Once deployed the app would look this in the following picture.



and will be available at a URL looking like this <PROJECT_ID>.appspot.com:

<https://my-nodejs-codelab-183319.appspot.com>

Additional documentation, including Quickstart, API & Libraries, Tutorials, Code samples are available at the “Node.js on Google Cloud Platform” web page at:

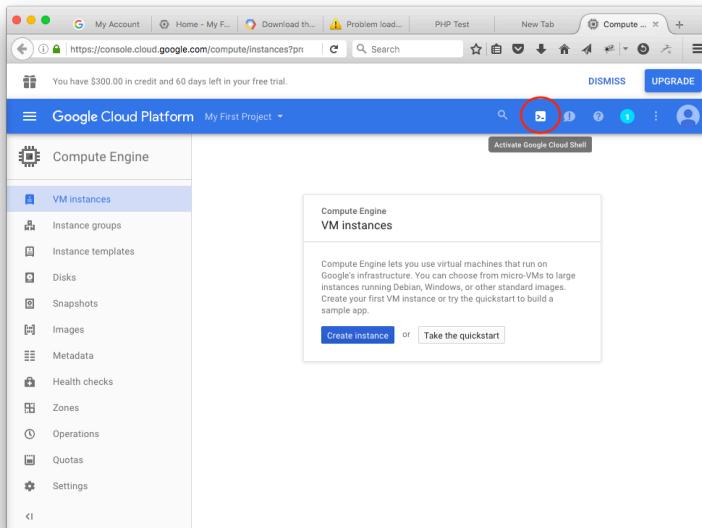
<https://cloud.google.com/nodejs/>

9. Set up Exploring Your instance (Optional)

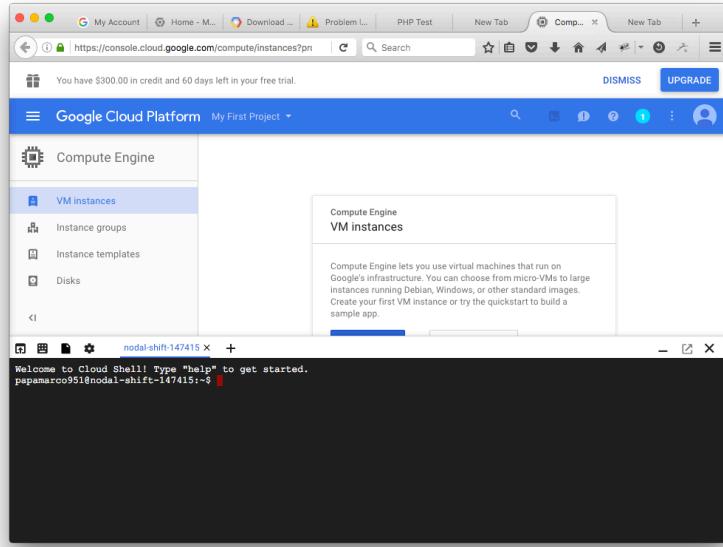
If you want to explore your server instance you can activate the **Google Cloud Shell**.

<https://console.cloud.google.com/compute>

Select the **helloworld** project from the dropdown. Now click on the **Activate Google Cloud Shell** icon next to **helloworld**.



After waiting a few minutes for Google to establish the connection, you will see the shell appear at the bottom of the browser window. You can now use Linux commands to manage your Cloud Platform Console projects and resources.



You can read more about the **Google Cloud Shell** here:

<https://cloud.google.com/cloud-shell/docs/>

10. Monitoring your instance and you Bill

Select Google Cloud Platform and go to the Dashboard. Under **Billing** you will see if you are incurring any charges.

The screenshot shows the Google Cloud Platform Dashboard for a project named "My First Project". The dashboard is divided into several sections:

- Project: My First Project**: ID: nodal-shift-147415 (# 63883682673). Includes a link to "Manage project settings".
- App Engine**: Summary (count/sec) - There is no data for this chart. Includes a link to "Go to the App Engine dashboard".
- Google Cloud status**: All services normal. Includes a link to "Go to Cloud status dashboard".
- Billing**: \$0.00. Approximate charges so far this month. Includes a link to "View detailed charges". This section is highlighted with a red box.
- APIs**: Requests (requests/sec) chart from Oct 24, 1:30 PM to Oct 24, 2:28 PM. Includes a link to "Go to APIs overview".
- Trace**: No trace data from the past 7 days. Includes a link to "Get started with Stackdriver Trace".
- Error Reporting**: No sign of any errors. Have you set up Error Reporting? Includes a link to "Set up Error Reporting".
- Explore other services**: API: Enable APIs and get credentials like keys; Deploy a prebuilt solution; Debug with local source and the new Logs panel; Monitor errors with Error Reporting; Take a VM quickstart; Install the Cloud SDK.
- News**: Now shipping: Windows Server 2016 images on Google Compute Engine 4 hours ago; Production debugging the easy way, with Stackdriver Debugger GA 3 days ago; Announcing new storage classes for Google Cloud Storage.
- Free Trial Support**: Need help getting started with your Free Trial? Send us a question. Includes links to Learn about Compute Engine, Learn about Cloud Storage, and Learn about App Engine.

Have fun exploring Google Cloud Platform!!