App Engine: Qwik Start - Java

GSP068

Google Cloud Self-Paced Labs

Overview

App Engine allows developers to focus on doing what they do best, writing code. The App Engine standard environment is based on container instances running on Google's infrastructure. Containers are preconfigured with one of several available runtimes (Java 8, Python 3.7, Go and PHP). Each runtime also includes libraries that support App Engine Standard APIs. For many applications, the standard environment runtimes and libraries might be all you need.
The App Engine standard environment makes it easy to build and deploy an application that runs reliably even under heavy load and with large amounts of data. It includes the following features:

- Persistent storage with queries, sorting, and transactions.
- Automatic scaling and load balancing.
- Asynchronous task queues for performing work outside the scope of a request.
- Scheduled tasks for triggering events at specified times or regular intervals.
- Integration with other Google cloud services and APIs.

Applications run in a secure, sandboxed environment, allowing App Engine standard environment to distribute requests across multiple servers, and scaling servers to meet traffic demands. Your application runs within its own secure, reliable environment that is independent of the hardware, operating system, or physical location of the server.

This hands-on lab shows you how to create a small App Engine application that displays a short message.

What you'll do

In this lab you will learn how to:

- Download starter code from a GitHub repository.
- Run your application locally from a Cloud Shell session.
- Deploy your application with Google App Engine.

Setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

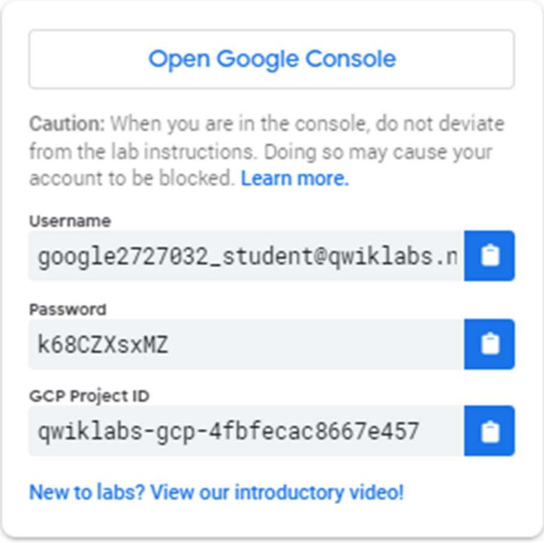
To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.
  Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

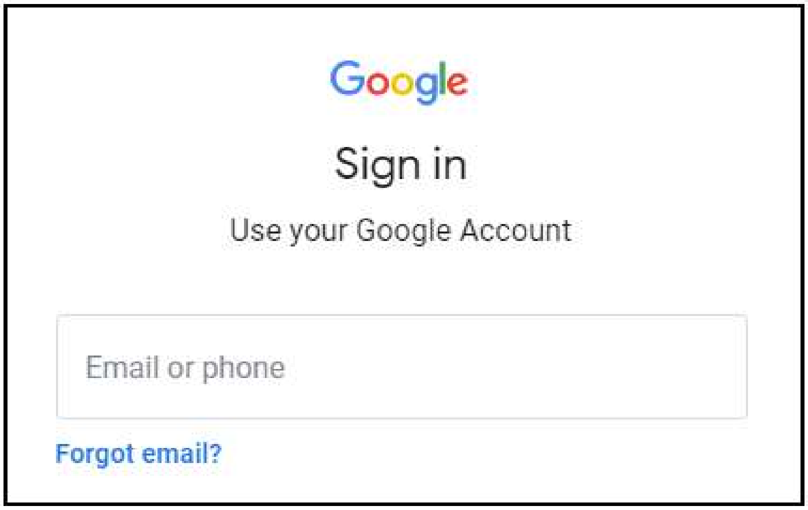
Username: google2727032_student@qwiklabs.n

Password: k68CZXsxMZ

GCP Project ID: qwiklabs-gcp-4fbfecac8667e457

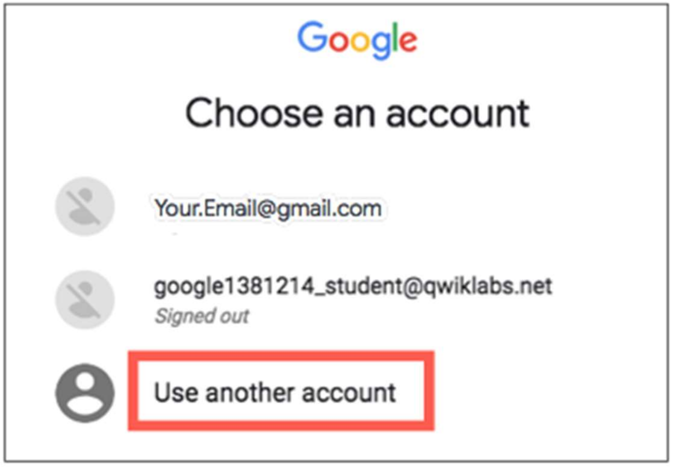
New to labs? View our introductory video!

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another Account**.



3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

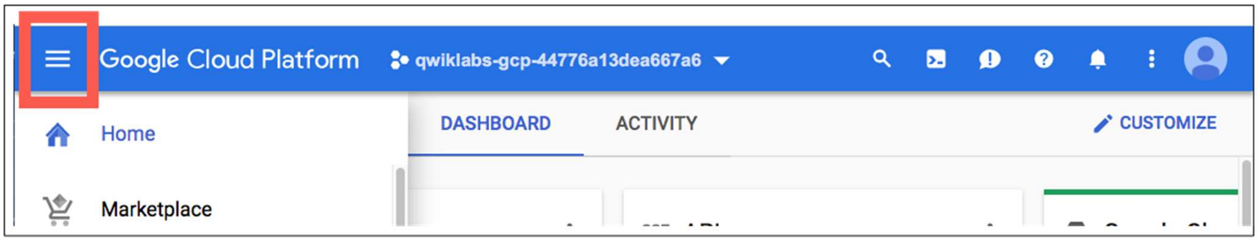
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:

   - Accept the terms and conditions.
   - Do not add recovery options or two-factor authentication (because this is a temporary account).
   - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

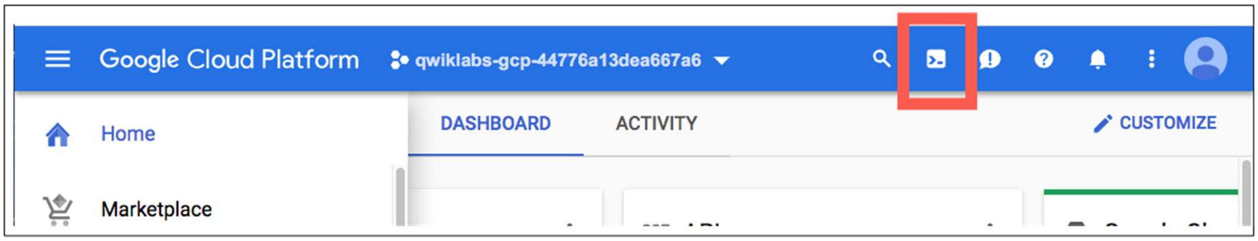
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



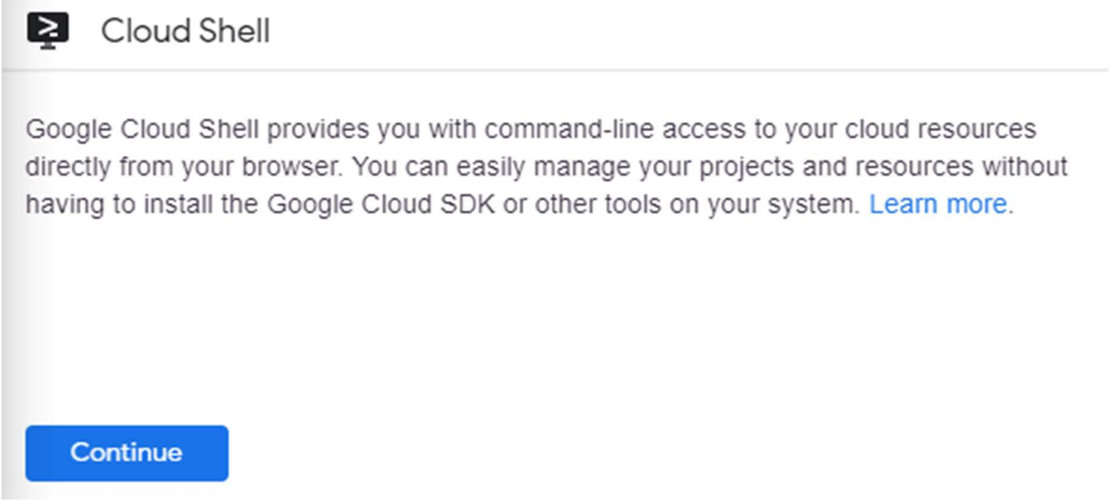
Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



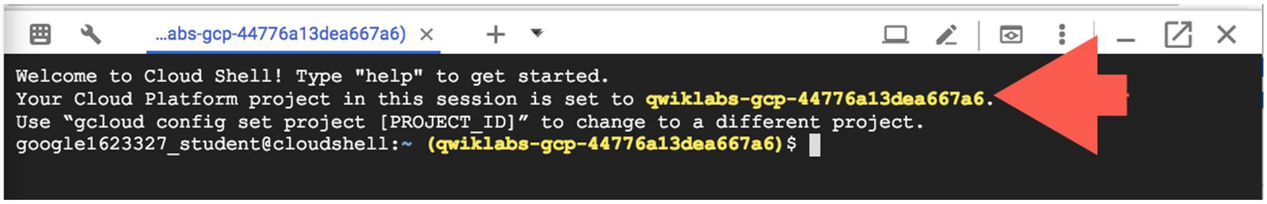
Click **Continue**.



Cloud Shell

Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. You can easily manage your projects and resources without having to install the Google Cloud SDK or other tools on your system. Learn more.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:
 - <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
project = <project_ID>
```

(Example output)

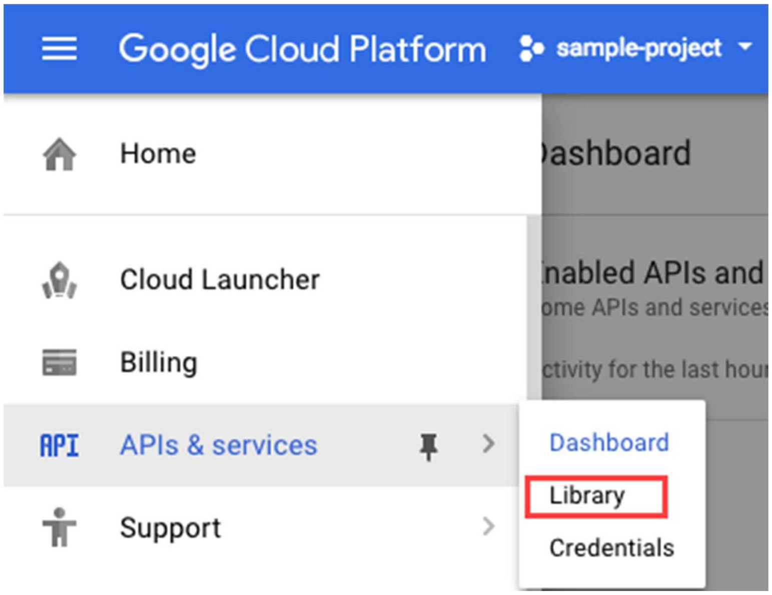
```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the gcloud command-line tool overview.

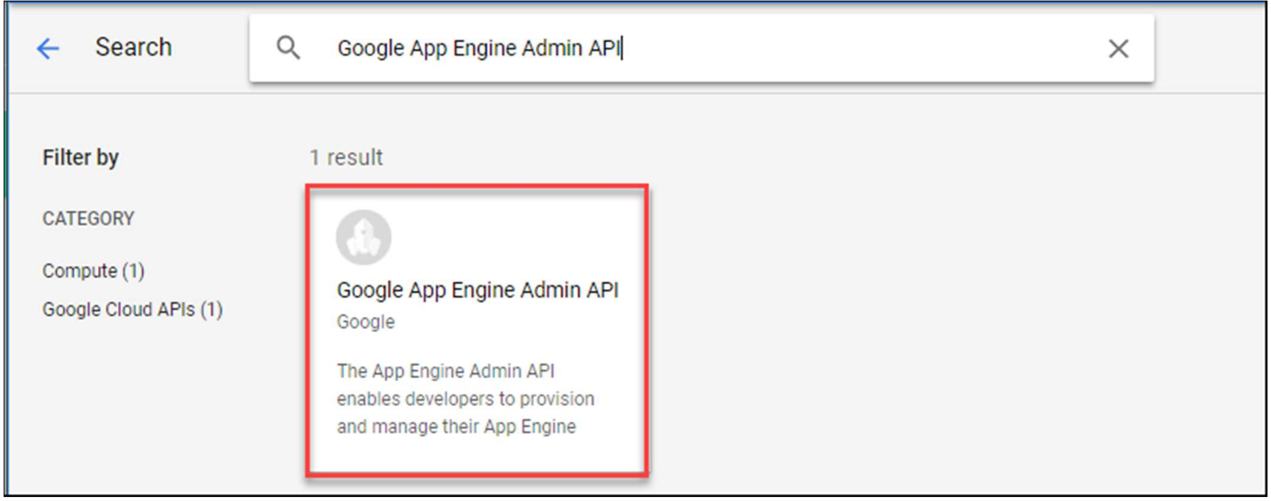
Enable Google App Engine Admin API

The App Engine Admin API enables developers to provision and manage their App Engine Applications.

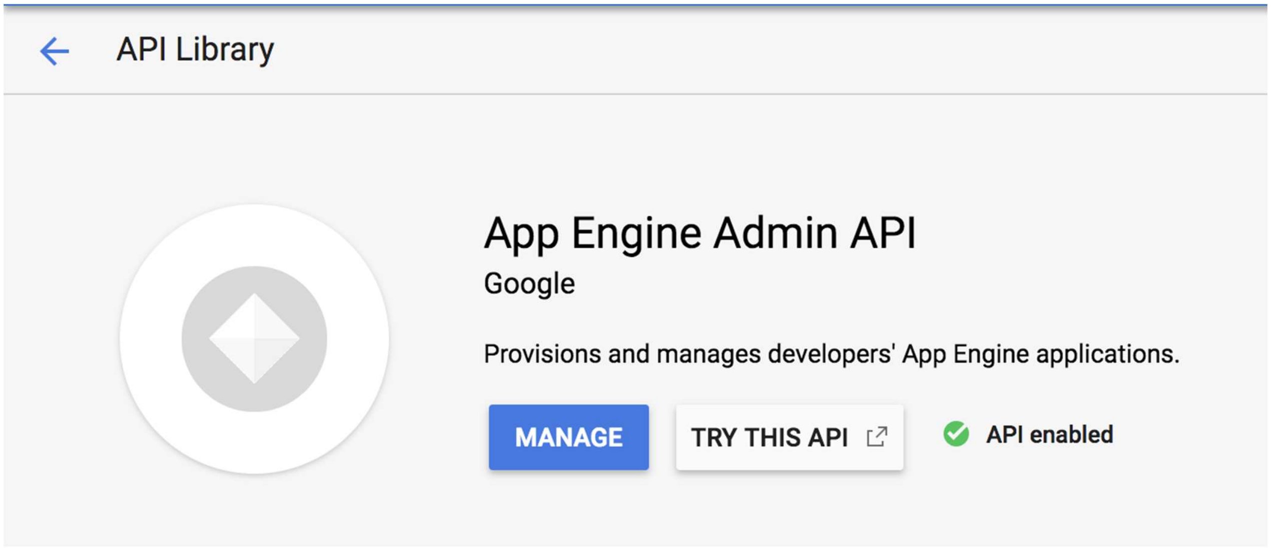
1. In the left-hand menu click on **APIs & Services** > **Library**.



2. Type "App Engine Admin API" in the search box.
3. Click **App Engine Admin API**.


Search
Google App Engine Admin API
Filter by
CATEGORY
Compute (1)
Google Cloud APIs (1)
1 result
Google App Engine Admin API
Google
The App Engine Admin API enables developers to provision and manage their App Engine

4. Click **Enable** if it isn't already set. Your page should now resemble the following:


API Library

App Engine Admin API

Google

Provisions and manages developers' App Engine applications.

Download the Hello World app

We've created a simple `Hello World` app written in Java so you can quickly get a feel for deploying an application to the Google Cloud. Follow these steps to download `Hello World` to your temporary Google Cloud shell environment.

1. Open a Cloud Shell session and run the following command to clone the Hello World sample app repository:

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

Output:

```
Cloning into 'getting-started-java'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 7608 (delta 17), reused 23 (delta 14), pack-reused 7567
Receiving objects: 100% (7608/7608), 50.79 MiB | 24.17 MiB/s, done.
Resolving deltas: 100% (4166/4166), done.
```

2. Then go to the directory that contains the sample code:

```
cd java-docs-samples/appengine-java8/helloworld
```

In this folder you will find the `src` directory that contains a package called `com.example.appengine.helloworld` that implements a simple HTTP Servlet.

Test the application using the development server

To get the development server running, you'll download Maven to manage compiling your app and starting the development server.

1. Run the following commands to configure your Maven environment:

```
mvn clean
mvn package
```

2. Enter the following Maven command to download and install Maven and run the app:

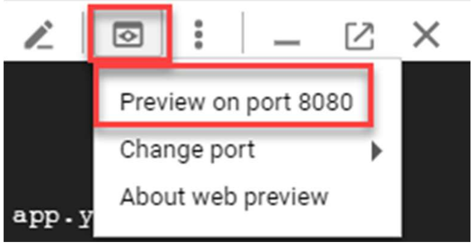
```
mvn appengine:run
```

It will take a few minutes for Maven to download and install.

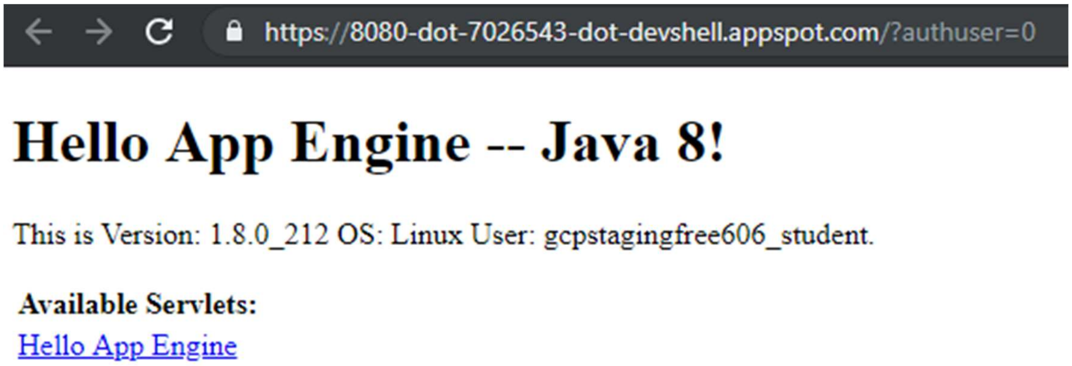
The development server is listening for requests on port 8080 when you see the following last line of output:

```
[INFO] GCLOUD: INFO: Dev App Server is now running
```

3. View the app by clicking the **Web preview** button > **Preview on port 8080**:



Your page should resemble the following:


https://8080-dot-7026543-dot-devshell.appspot.com/?authuser=0

Hello App Engine -- Java 8!

This is Version: 1.8.0_212 OS: Linux User: gcpstagingfree606_student.

Available Servlets:
Hello App Engine

In your terminal window, press `Ctrl+C` to stop the development server.

Deploy your app

1. Now you'll create an application on an App Engine with the following command:

```
gcloud app create
```

When prompted, enter your choice of region. You should receive the following output soon after:

```
Success! The app is now created. Please use `gcloud app deploy` to deploy your first
app.
```

2. Open the `pom.xml` file with the following command:

```
nano pom.xml
```

3. Key down towards the bottom of the page until you find this section:

```
<version>2.2.0</version>
<configuration>
  <!-- can be set w/ -DprojectId=myProjectId on command line -->
  <projectId>myProjectId</projectId>
  <!-- set the GAE version or use "GCLOUD_CONFIG" for an autogenerated GAE version -->
  <version>GCLOUD_CONFIG</version>
</configuration>
```

4. Change `myProjectId` to your Qwiklabs Project ID. Ensure this section resembles the following before moving on:

```
<configuration>
  <!-- can be set w/ -DprojectId=myProjectId on command line -->
  <projectId>qwiklabs-gcp-02-47242f3ecbf9</projectId>
  <!-- set the GAE version or use "GCLOUD_CONFIG" for an autogenerated GAE version -->
  <version>GCLOUD_CONFIG</version>
</configuration>
```

5. Now exit nano and save the file with **CTRL** + **X** --> **Y** --> **Enter**.

6. **DO NOT** use the `gcloud app deploy` command as stated in the output to deploy your app. Instead, run the following command to deploy your application:

```
mvn package appengine:deploy
```

You should receive the following output:

```
------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  38.848 s
[INFO] Finished at: 2019-01-15T17:42:49-08:00
[INFO] ------------------------------------------------------------------------
```

View your application

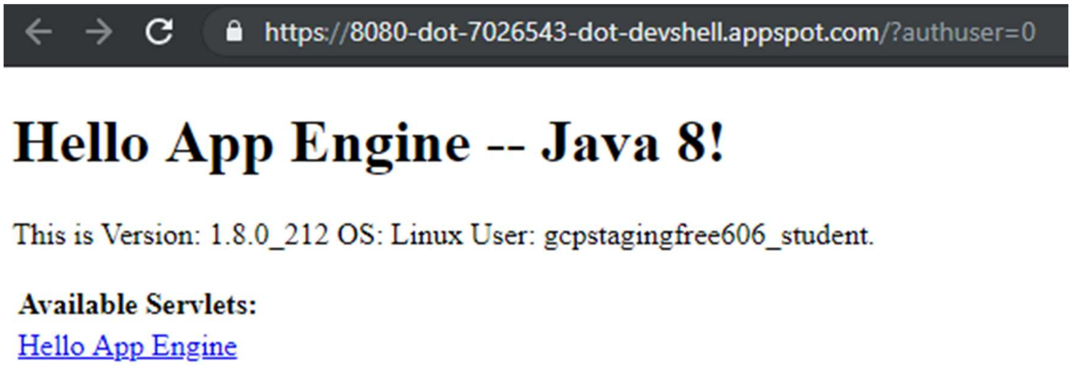
To launch your browser, enter the following command then click on the link it provides.

```
gcloud app browse
```

Example output; your link will be different:

```
Did not detect your browser. Go to this link to view your app:
https://qwiklabs-gcp-5c823ee0b4c7fa19.appspot.com
```

Your application is deployed and you can read the short message in your browser:



https://8080-dot-7026543-dot-devshell.appspot.com/?authuser=0

Hello App Engine -- Java 8!

This is Version: 1.8.0_212 OS: Linux User: gcpstagingfree606_student.

Available Servlets:
Hello App Engine

Click **Check my progress** to verify the objective.

Test your knowledge

Test your knowledge about Google cloud Platform by taking our quiz. (Please select multiple correct options if necessary.)

□ ch of the following are feature of App Engine?

□ nchronous task queues

□ omatic scaling and load balancing.

□ sistent storage with queries, sorting, and transactions.

□ eduled tasks for triggering events at specified times

All of them

Submit

Congratulations!

Finish Your Quest

Continue your Quest with Baseline: Deploy & Develop. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in this Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take Your Next Lab

This lab is also part of a series of labs called Qwik Starts. These labs are designed to give you a little taste of the many features available with Google Cloud. Search for "Qwik Starts" in the lab catalog to find the next lab you'd like to take!

Next Steps /Learn More

- Lean more about an App Engine with An Overview Of App Engine
- Try something else with an App Engine with Getting Started with Flask on App Engine Standard Environment

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.
Manual Last Updated June 05, 2020
Lab Last Tested June 05, 2020