

App Dev - Deploying the Application into App Engine Flexible Environment - Java

GSP172



Google Cloud Self-Paced Labs

Overview

An App Engine app is a single application resource with one or more services. Each service can be configured to use different runtimes and to operate with different performance settings. Within each service, you can deploy versions of that service, and each then runs within one or more instances, depending on how much traffic you configured it to handle. For more information, see [an overview of App Engine](#).

App Engine uses either a Standard or Flexible environment. A standard environment runs instances in a sandbox, limiting available CPU options and disc access.

In contrast, a flexible environment runs your application in Docker containers on Compute Engine virtual machines (VMs), which have fewer restrictions. For example, you can use the programming language or library of your choice, write to disk, and even run multiple processes. You also have the choice of Compute Engine machine types for your instances. For more information about App Engine environments, see [The App Engine Standard Environment](#) and [App Engine Flexible Environment](#).

In this lab, you deploy a quiz application into App Engine flexible environment, and leverage App Engine features like versions and traffic splitting.

Objectives

In this lab, you learn how to perform the following tasks:

- Create an app.yaml file to describe the App Engine flexible environment requirements for an application.
- Deploy the quiz application into App Engine flexible environment.
- Employ versions and traffic splitting to perform A/B testing of an application feature.

Setup and Requirements

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:

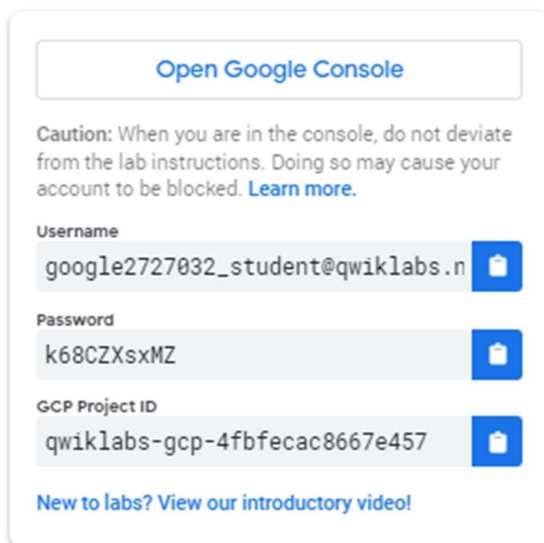
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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.



The screenshot shows a panel with the following content:

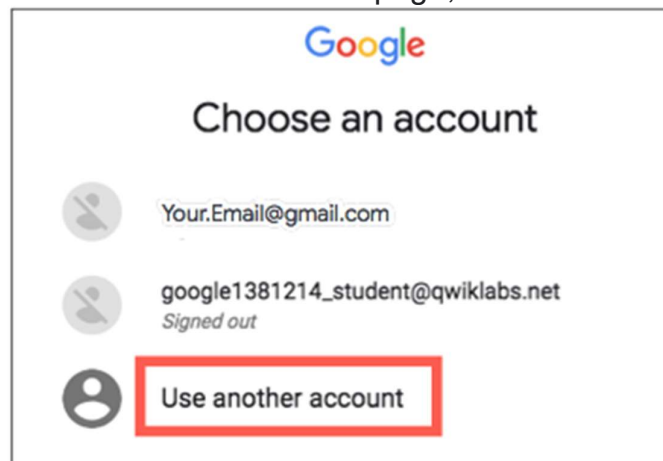
- A button at the top labeled "Open Google Console".
- A caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)"
- Three input fields, each with a copy icon to its right:
 - Username:** google2727032_student@qwiklabs.n
 - Password:** k68CZXsxMZ
 - GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457
- A link at the bottom: "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).

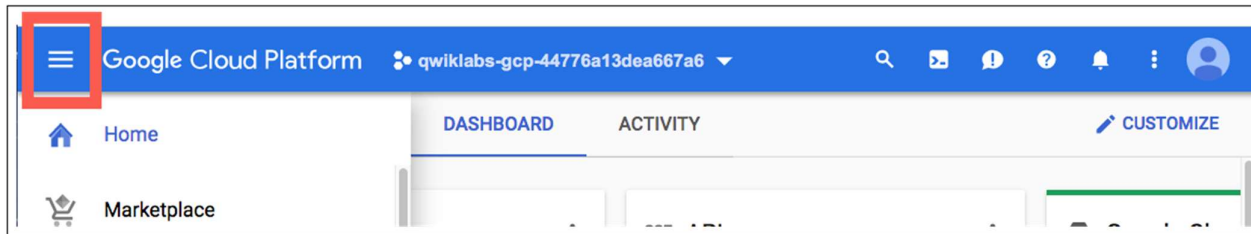
4. Click through the subsequent pages:

- 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-

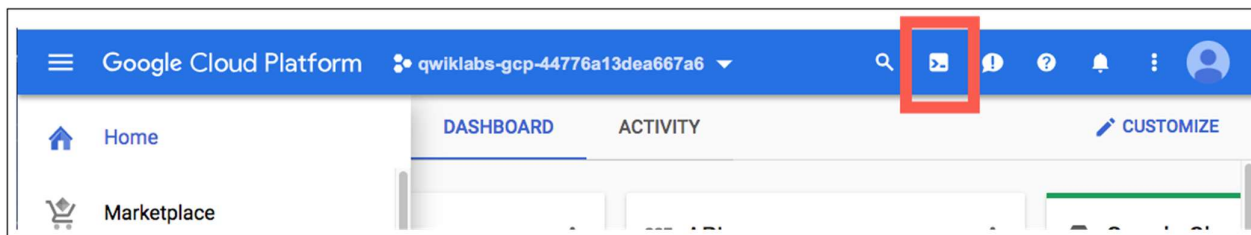
left.



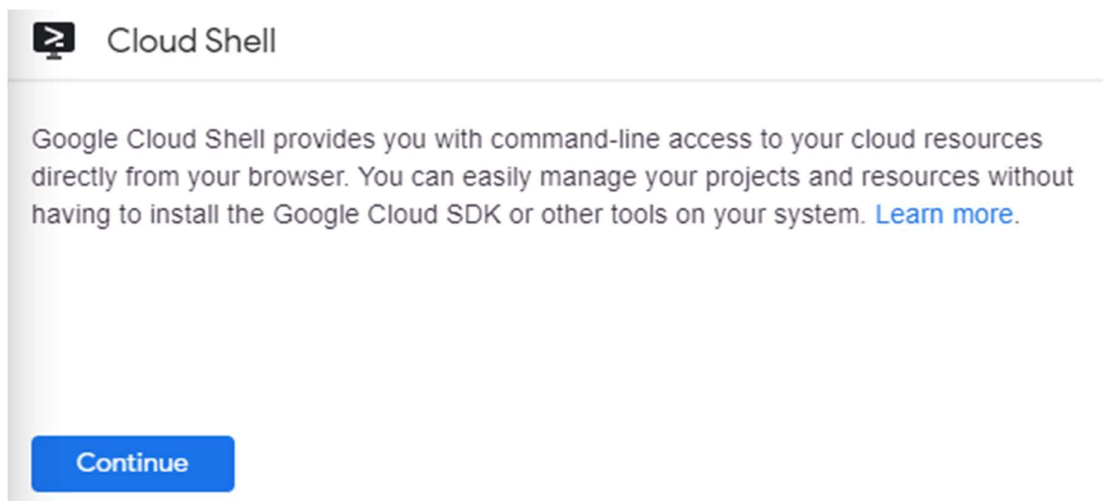
Activate Cloud Shell

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.

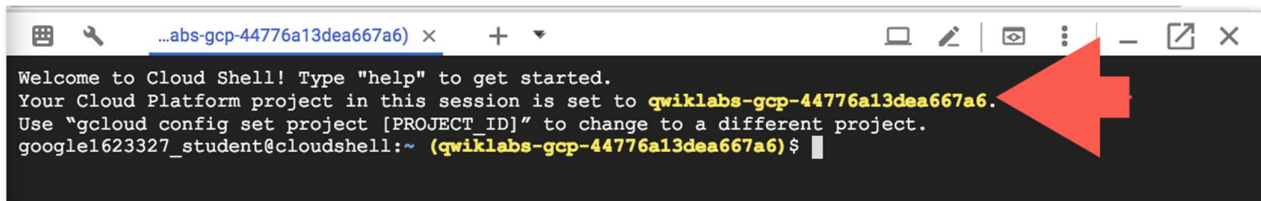
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



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:



```
...abs-gcp-44776a13dea667a6) x + ▾
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](#).

Preparing the Case Study Application

In this section, you access Cloud Shell, clone the git repository that contains the Quiz application, configure environment variables, and run the application.

Clone source code in Cloud Shell

Enter the following command to clone the repository for the lab.

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```

Configure the case study application

Now change the working directory:

```
cd ~/training-data-analyst/courses/developingapps/java/appengine/start
```

You will have to make one small change to your executable file.

Open `prepare_environment.sh` by running the following command:

```
nano prepare_environment.sh
```

Now find the `gcloud beta functions deploy` command near the bottom of the file and remove the `beta` command. Ensure that line of code resembles the following:

```
gcloud functions deploy process-feedback --runtime nodejs8 --trigger-topic feedback --  
source ./function --stage-bucket $G_CLOUD_BUCKET --entry-point subscribe
```

Now exit the nano editor with **CTRL + X > Y > Enter**.

Run the executable file by running the following command:

```
. prepare_environment.sh
```

This script file

- Creates an App Engine application.
- Exports environment variables `G_CLOUD_PROJECT` and `G_CLOUD_BUCKET`.
- Runs `mvn clean install`.
- Creates entities in Google Cloud Datastore.
- Creates a Google Cloud Pub/Sub topic.
- Creates a Cloud Spanner Instance, Database, and Table.
- Prints out the Project ID.

When prompted with `Allow unauthenticated invocations of new function [process-feedback]?`, enter in **Y**.

NOTE: Please re-run the above command, if you get an error as **(gcloud.functions.deploy) OperationError: code=3, message=Failed to retrieve function source code**

Review the code

In this lab you view and edit files. You can use the shell editors that are installed on Cloud Shell, such as nano or vim, or use the Cloud Shell code editor. This lab uses the Cloud Shell code editor.

Launch the Cloud Shell Editor

In the Cloud Platform Console, click **Open editor**. This icon looks like a pencil.



If an error indicates that the code editor could not be loaded because third-party cookies are disabled, click **Open in New Window** and switch to the new tab.

Navigate to **training-data-analyst/courses/developingapps/java/appengine/start**.

The folder structure for the quiz application reflects how it will be deployed in App Engine.

There is a configuration file for App Engine; `app.yaml` in a new folder, `src/main/appengine`.

Click Check my progress to verify the objective.

Preparing Application Code for App Engine Flexible Environment Deployment

In this section, you modify the configuration files for deployment of the quiz application frontend into App Engine flexible environment.

Create the app.yaml file for the frontend

In the Cloud Shell code editor, open `src/main/appengine/app.yaml`.

The following content describes the App Engine configuration. Copy and paste the content into `app.yaml`. Be sure you replace `[G_CLOUD_BUCKET]` with the appropriate bucket name `G_CLOUD_PROJECT-media` from your project.

src/main/appengine/app.yaml

```
runtime: java
env: flex
runtime_config:
  jdk: openjdk8
handlers:
- url: /*
  script: this field is required, but ignored
manual_scaling:
  instances: 1
resources:
  cpu: 1
  memory_gb: 3.75
  disk_size_gb: 10
env_variables:
  G_CLOUD_BUCKET: [G_CLOUD_BUCKET]
```

Save the file.

Deploy the quiz application to App Engine flexible environment

Return to the Cloud Shell window. If the Cloud Shell is not visible, click **Open Terminal**.

In Cloud Shell, enter the following command to deploy the quiz application to App Engine flexible environment.

```
mvn clean compile appengine:deploy
```

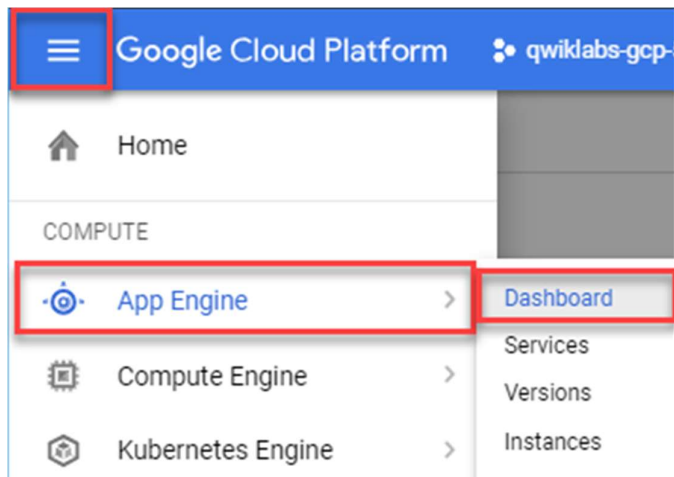
It may take around 10 minutes to complete the deployment.

Maven rebuilds the project and then invokes `gcloud app deploy`. App Engine automatically packages, containerizes, and deploys the code.

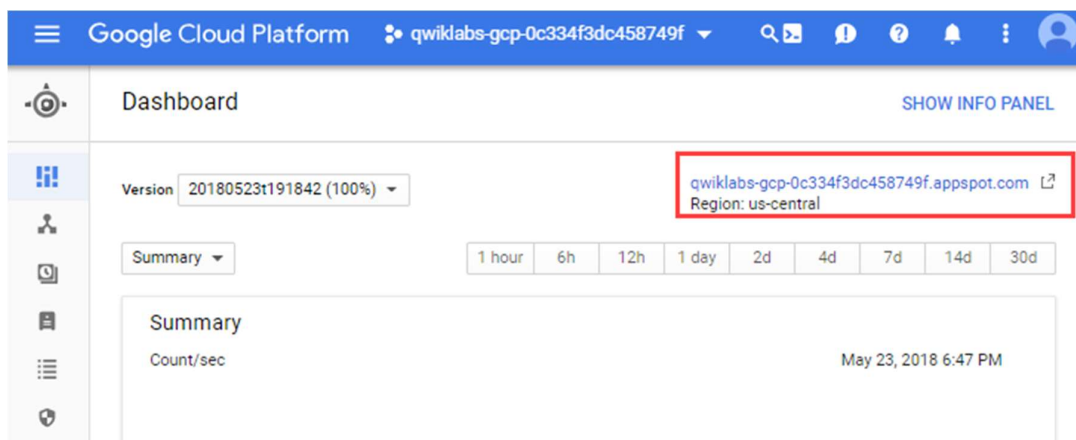
You'll see output similar to the following when the deployment is complete:

```
INFO\] -----
\[INFO\] BUILD SUCCESS
\[INFO\] -----
\[INFO\] Total time: 10:21 min
\[INFO\] Finished at: 2018-05-23T19:28:50-04:00
\[INFO\] Final Memory: 37M/90M
\[INFO\] -----
```

In the Cloud Platform Console, on the **Navigation menu > App Engine > Dashboard**.



Click on the link to your application in the top-right corner of the App Engine Dashboard to see your application, the Quite Interesting Quiz.



Click Check my progress to verify the objective.

Updating an App Engine Flexible Environment Application

In this section, you will modify the application code and then redeploy the application.

Update the quiz application

In the Cloud Shell code editor, open the `src/main/resources/static/index.html` file. Add several exclamation points to the top-level heading.

src/main/resources/static/index.html

```
<!-- This is just a fragment, only add the exclamation points -->
<div class="container">
  <h1>Welcome to the Quite Interesting Quiz!!!!</h1>
  <div class="jumbotron">
    <p>Welcome to the Quite Interesting Quiz where you can create a question, take
a test or review feedback</p>
  </div>
```

This small change stands in for all the changes you might make when updating an application.

Click **File > Save** to save the file.

Deploy the updated application

In **Cloud Shell**, redeploy the App Engine application.

```
mvn clean compile appengine:deploy \
-Dapp.deploy.stopPreviousVersion=False \
-Dapp.deploy.promote=False
```

Notice the two additional flags in the command, which means that the previous version will continue to receive traffic.

It may take around 10 minutes for the deployment to update.

In the Console, click **Navigation menu > App Engine > Dashboard**.

Click on the application URL in the top-right corner of the window. You should see that your application still displays the old title.

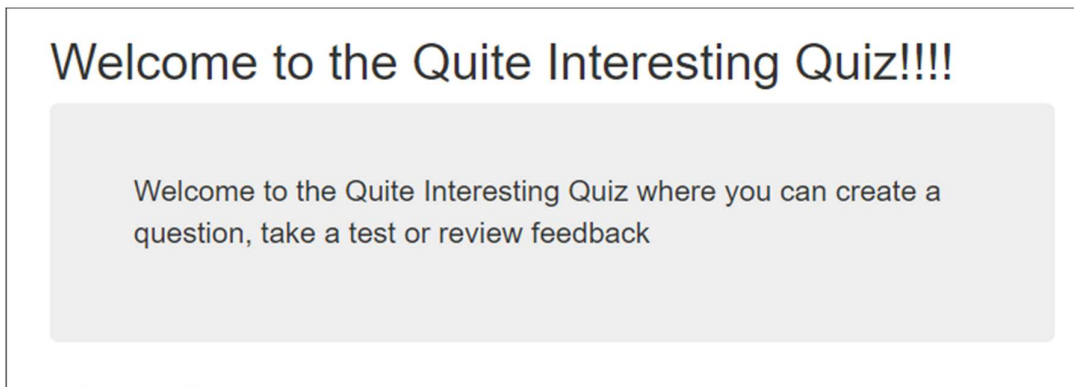
In the **App Engine** window, click **Versions**. You may need to refresh the page until you see two versions of the applications.

App Engine	Versions	REFRESH	DELETE	STOP	START		SHOW INFO PANEL
Dashboard	Filter versions						
Services							
Versions							
Instances							
Task queues							
Security scans							
Firewall rules							

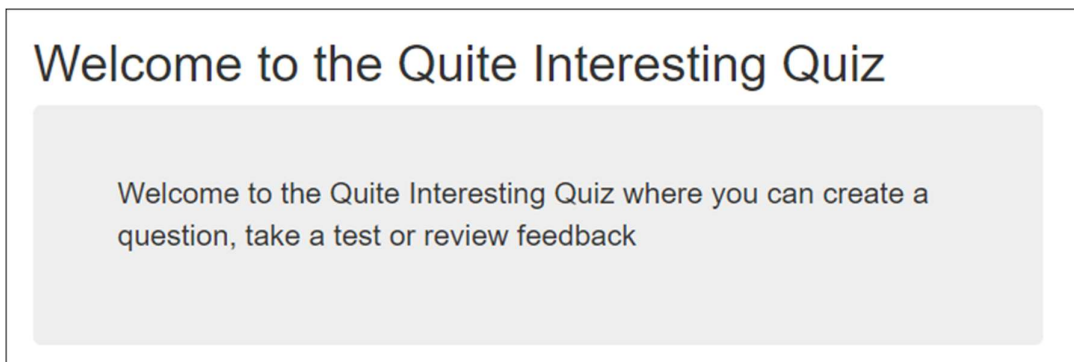
<input type="checkbox"/>	Version	Status	Traffic Allocation	Instances	Runtime	Environment	Size
<input type="checkbox"/>	20180523t195709	Serving	0%	1	java	Flexible	0 B
<input type="checkbox"/>	20180523t191842	Serving	100%	1	java	Flexible	0 B

The version ID is in the form 'yyyymmddthhmmss', so it's easy to see which is the new and which is the old version. Click on both version links to see the new and old version of the quiz application.

New version (notice those exclamation points!):



Old version:



Enable the checkbox for both versions of the application, and click **Split traffic**.

Versions

REFRESH

DELETE

STOP

START

Split traffic

SHOW INFO PANEL

Filter versions

<input checked="" type="checkbox"/>	Version	Status	Traffic Allocation	Instances	Runtime	Environment	Size
<input checked="" type="checkbox"/>	20180523t195709	Serving	0%	1	java	Flexible	0 B
<input checked="" type="checkbox"/>	20180523t191842	Serving	100%	1	java	Flexible	0 B

Select the radio button to deliver versions randomly.

Configure the traffic split to deliver 50% of traffic to the old version, and 50% to the new version.

← Split traffic

You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features [Learn more](#)

Split traffic by

- ☐ IP address
- ☐ Cookie
- ☒ Random

Traffic allocation

20180523t191842	will receive the remaining	50 %	×
20180523t195709		50 %	×

+ Add version

Save Cancel

Click **Save**.

Return to the **Dashboard**, set **Version** to `All versions` and click the application link.

Dashboard [SHOW INFO PANEL](#)

Version `All versions`

qwiklabs-gcp-0c334f3dc458749f.appspot.com [↗](#)
Region: us-central

Open a new browser. If possible the browser should be a different type, for example Google Chrome to Internet Explorer. Copy the application link and paste it in the new browser.

Refresh the homepage a few times.

You should see that the homepage displays the old version approximately half the time, and the new version half the time.

In real-world scenarios, you might start by delivering small amounts of traffic to the new version in a canary release, and would use either a cookie or IP address to ensure that a client viewed a single consistent version of the application.

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

Congratulations!

You created `app.yaml`, a file that describes App Engine flexible environment requirements for an application. You also deployed the application and employed traffic splitting.

Finish your Quest



This lab is part of the [Application Development - Java](#) and [Cloud Development](#) Quests. 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](#).

Next steps

- Learn more about [Java on the Google Cloud](#).
- Get more information about [App Engine Flexible 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 December 01, 2020

Lab last tested December 01, 2020

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.