

# App Engine: Qwik Start - Ruby

GSP132



# Overview

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

You download a simple Ruby application written with the Sinatra web framework and deployed it to App Engine. Although this sample uses Sinatra, you can use other frameworks such as Ruby on Rails.

Google App Engine Flexible environment applications are easy to create, maintain, and scale as your traffic and data storage changes. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go.

## What you'll learn

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

- Test run an app on your local computer
- Run an application on a web server

## Setup and Requirements

### Qwiklabs 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.

## Cloud Console

### 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.

Open Google Console

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  
k68CZsxMZ

GCP Project ID  
qwiklabs-gcp-4fbfecac8667e457

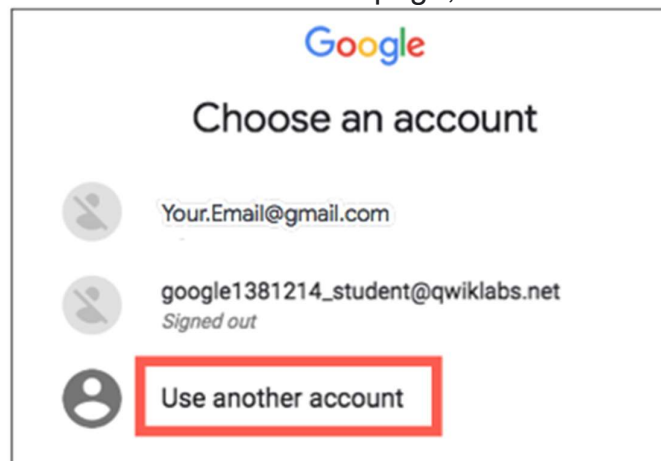
[New to labs? View our introductory video!](#)

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.

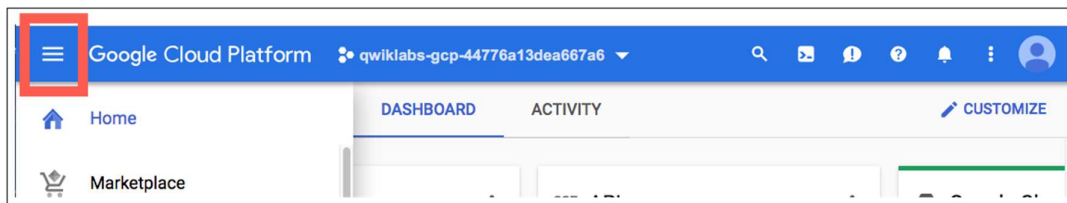
**Important:** You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

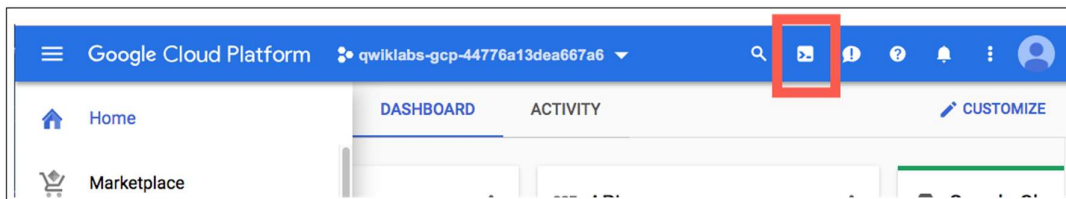


## Google Cloud Shell

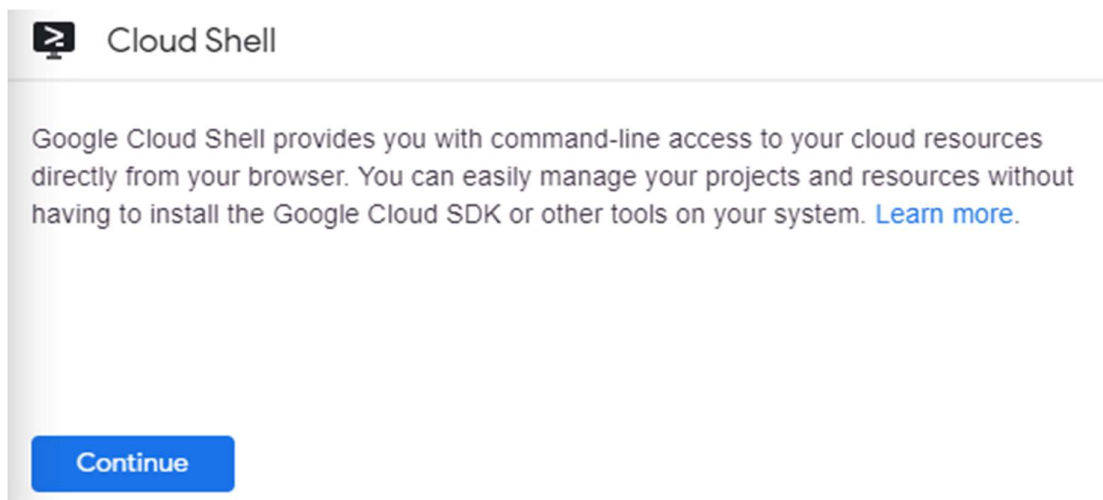
### Activate Cloud Shell

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.

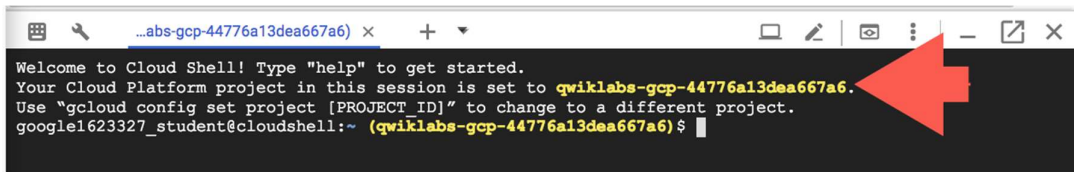
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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>
```

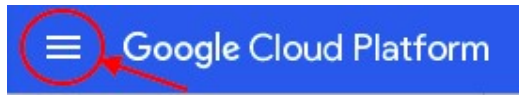
(Example output)

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

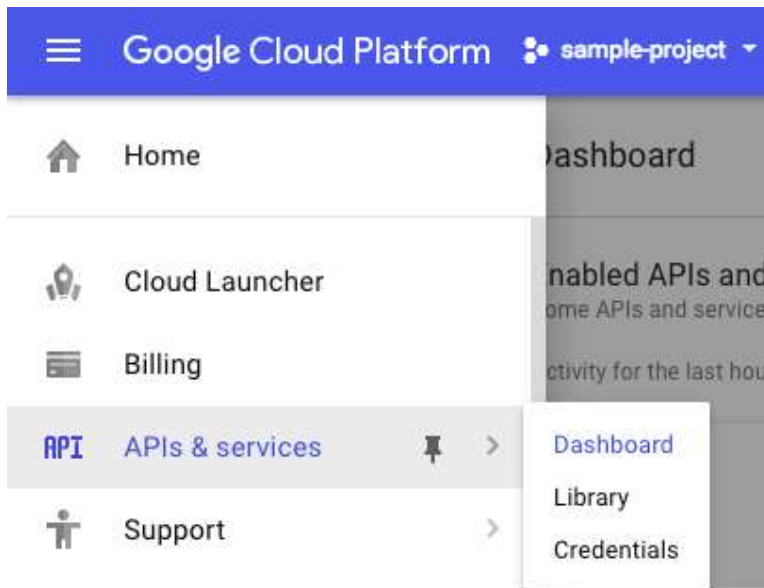
For full documentation of `gcloud` see the [gcloud command-line tool overview](#).

# Enable the App Engine Admin API

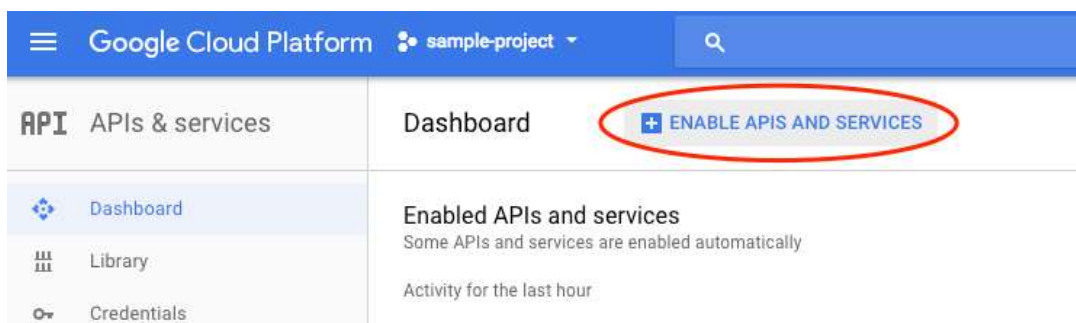
Click on the menu icon in the top left of the screen.



Select **APIs & services** from the drop down and click on **Dashboard**.



Click on **Enable APIs and services**.



Then, search for **App Engine Admin API** in the search box. Click on **App Engine Admin API**, then Click **Enable**.

# Download the Hello World app

We've created a simple Hello World app for Ruby so you can quickly get a feel for deploying an app to the App Engine flexible environment. Follow these steps from the Google Cloud Shell command line to download Hello World to your local machine.

1. Download the sample app and create the app directory:

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

2. Navigate to the directory that contains the sample code:

```
cd ruby-docs-samples/appengine/hello_world
```

## Hello World code review

Hello World is the simplest possible App Engine app, as it contains only one service, has only one version, and all of the code is located within the app's root directory. This section describes each of the app files in detail.

### app.rb

The Hello World app is a very basic one-file Sinatra application. The following shows [appengine/hello\\_world/app.rb](#):

```
require "sinatra"

get "/" do
  "Hello world!"
end
```

### app.yaml

The [app.yaml](#) file describes an application's deployment configuration. The following shows [appengine/hello\\_world/app.yaml](#):

```
runtime: ruby
env: flex
entrypoint: bundle exec ruby app.rb

# This sample incurs costs to run on the App Engine flexible environment.
# The settings below are to reduce costs during testing and are not appropriate
# for production use. For more information, see:
# https://cloud.google.com/appengine/docs/flexible/ruby/configuring-your-app-with-app-
yaml
manual_scaling:
  instances: 1
resources:
  cpu: 1
  memory_gb: 0.5
  disk_size_gb: 10
```

Here, `app.yaml` specifies the runtime used by the app, and sets `env: flex`, specifying that the app uses the [flexible environment](#).

- For more information on how the Ruby runtime works, see [The Ruby Runtime](#).



- For more details about how to design your app to take advantage of versions and services, see [An Overview of App Engine](#).
- For more details about the configuration settings for App Engine, see [Configuring Your App with app.yaml](#).

### Gemfile

[Gemfile](#) is used to specify the application's [RubyGem](#) dependencies, and Bundler is used to declare and install dependencies. The Hello World application requires one Ruby gem: the [Sinatra](#) web framework. The following shows [appengine/hello\\_world/Gemfile](#):

```
source "https://rubygems.org"

gem "sinatra"
```

## Run Hello World on your local computer

Run Hello World on your local computer:

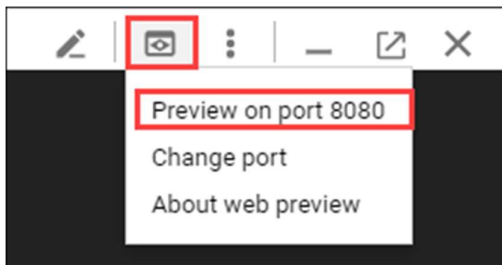
1. Install dependencies for this project by entering the following command:

```
bundle install
```

2. Start a local web server:

```
bundle exec ruby app.rb -p 8080
```

3. Change to the Hello World message displayed in the page by clicking the **Web preview** icon > **Preview on port 8080**.



4. In your terminal window, press **Ctrl+C** to exit the web server. Now you're ready for the next section.

# Deploy and run Hello World on App Engine

To deploy your app to the App Engine flexible environment:

1. Deploy the Hello World app by running the following command from the hello\_world directory:

```
gcloud app deploy
```

2. When prompted, enter the number corresponding to your region, then press Y to confirm the services to deploy. It may take several minutes for the app to deploy.

Learn about [optional flags](#)

3. Run the following command and click the link provided in the output to launch your browser.

```
gcloud app browse
```

The link will be `http://[PROJECT_ID].appspot.com`

This time, the page that displays the Hello World message is delivered by a web server running on an App Engine instance.

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.)

Google app engine flexible environment provides support for various frameworks of Ruby like sinatra and rails.

☐

True

☐

False

# Congratulations!

You've deployed your first Ruby app to App Engine flexible environment!

## Learn More

This lab is 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!

## 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 April 2, 2019

Lab Last Tested September 12, 2018

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.