## Lab 4-01: App Engine

### Lab Prerequisites

* Familiarity with basic Google Cloud Computing concepts and terminology.
* A Google account with an active subscription.

### Service Introduction

Google Cloud Platform (GCP) provides a platform-as-a-service (PaaS) offering called App Engine, which allows developers to deploy and manage web applications on a fully managed serverless platform.

App Engine provides a number of features, such as automatic scaling, load balancing, and automatic security patching, which allows developers to focus on writing code rather than managing infrastructure.

One key feature of App Engine is its versioning system. App Engine allows developers to create multiple versions of their application, each with its own unique URL and resources, such as memory and CPU limits. This allows for A/B testing, rollbacks, and easy deployment of new features.

### Case Study Food Delivering Platform – QuickFood

Background

QuickFood is a food delivery website that allows users to order food from local restaurants and have it delivered to their location. The website is built using a combination of front-end and back-end technologies and is deployed locally on its own data centers. Their business was booming until competitors arrived. Their customers started moving towards their rivals, with the main complaint being that their service (i.e., app and website) was slow and unreliable. QuickFood needs to modernize its IT infrastructure on numerous fronts in order to accomplish its demanding goals. The business desired a more flexible development and deployment procedure. It also needs a more highly available solution than its on-premises data centers could provide.

The Company has decided to move its operations to the Google Cloud Platform (GCP) in order to take advantage of its scalability, reliability, and security. Therefore, they have hired you, a Cloud Architect, to move its operations to GCP.

### Business Challenge

QuickFood wants to hire a Cloud Architect to regularly test and deploy the updated versions of their website. The website will be regularly updated because they want to reach the peak they were once at.

### Proposed Solution

You have been hired as a full-time Cloud Architect for QuickFood. You have suggested the use of App Engine and its version testing to launch the website on GCP.

Lab Diagram

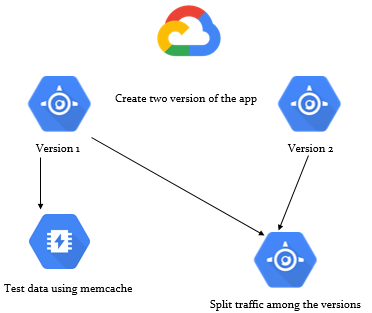


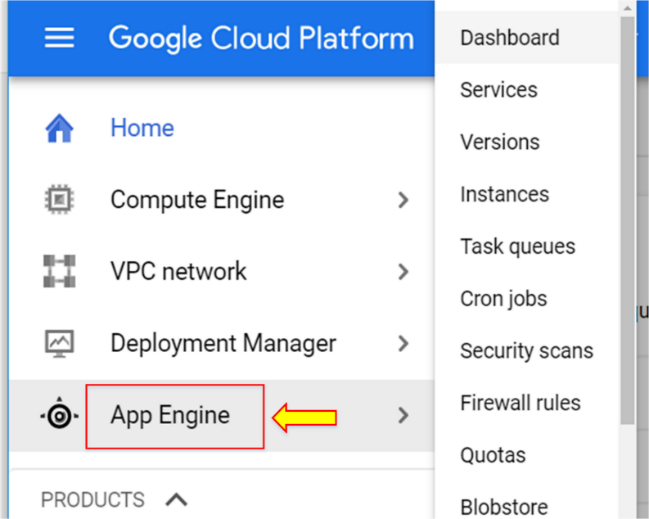
Figure 4-01: Lab diagram

Implementation Steps

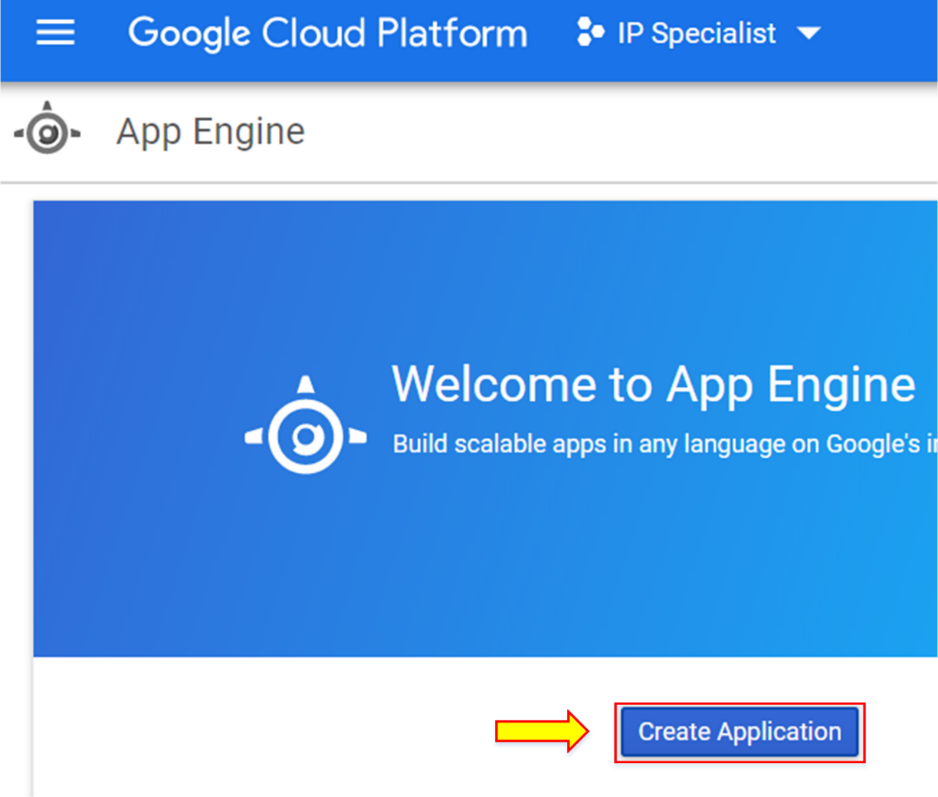
1. Create and deploy an App Engine application.
2. Change the memchache service level.
3. Test the memchache.
4. Create a new version of the app.
5. Split traffic among the versions.

Solution

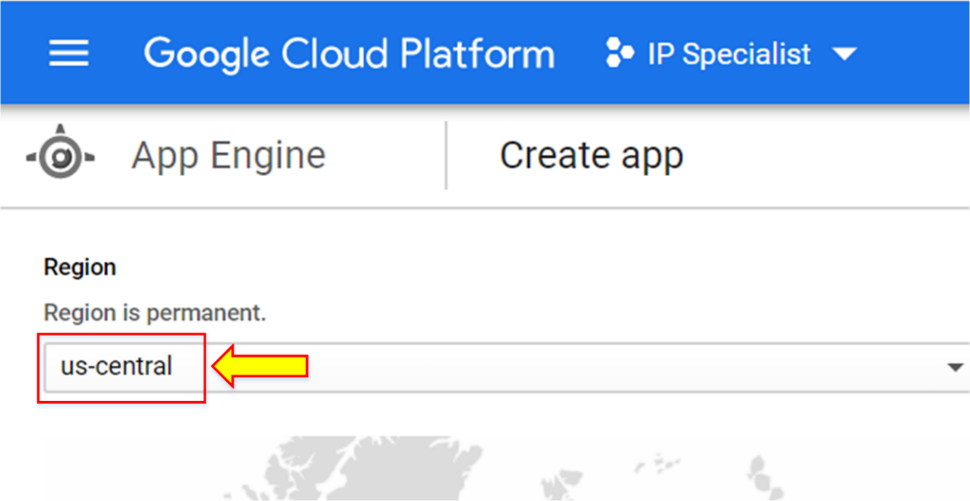
1. Go to “App Engine”.



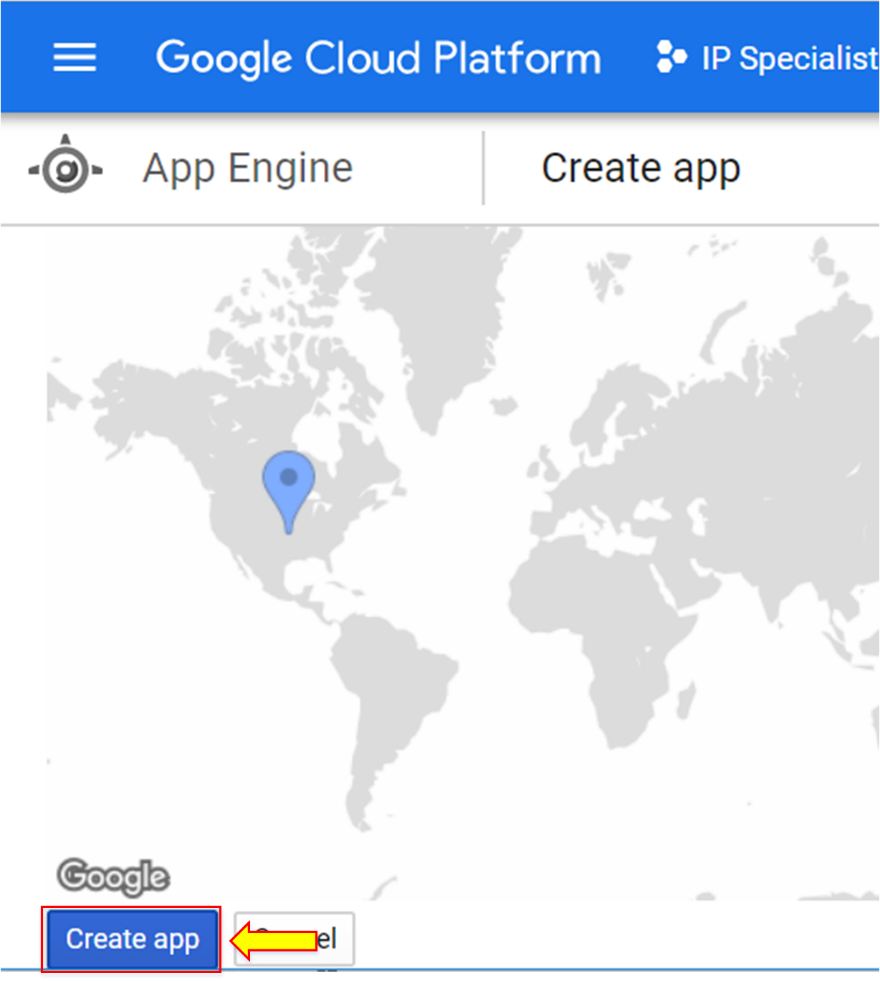
1. Click “Create Application”.



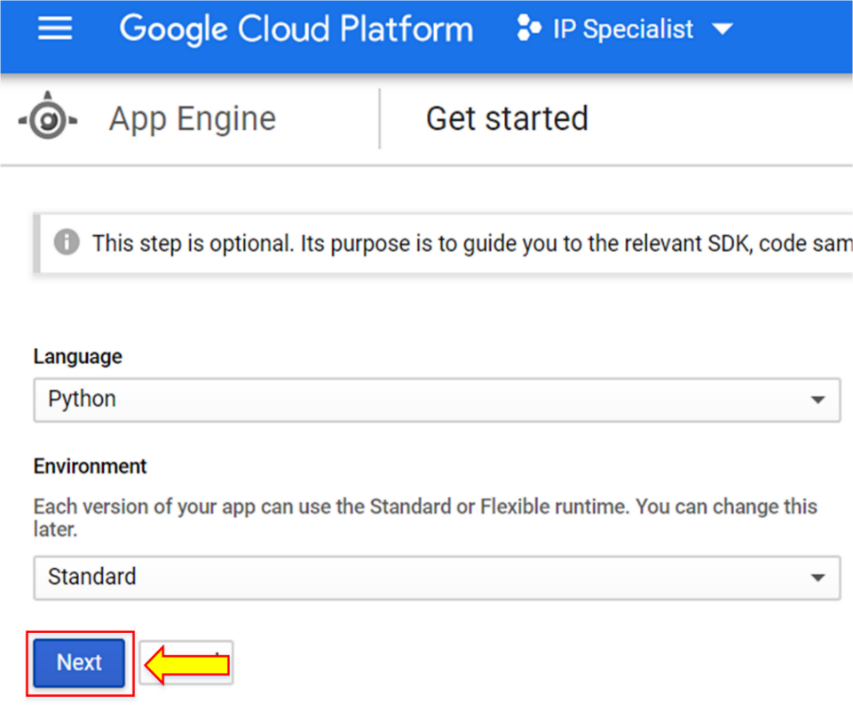
1. Select any region.



1. Click “Create app”.

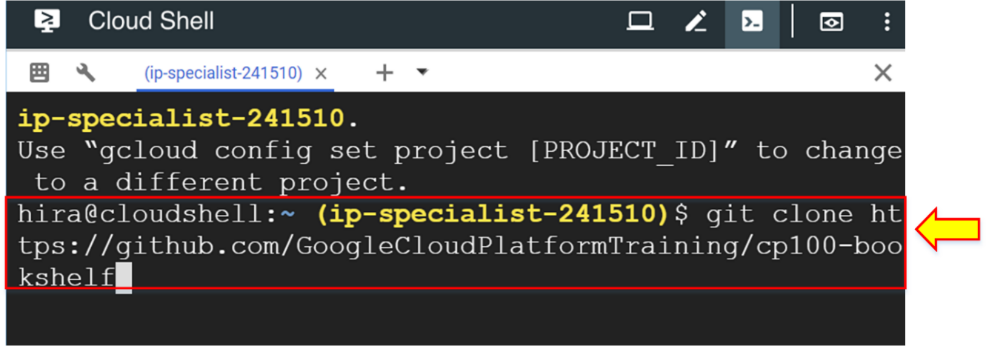


1. Click “Next”.

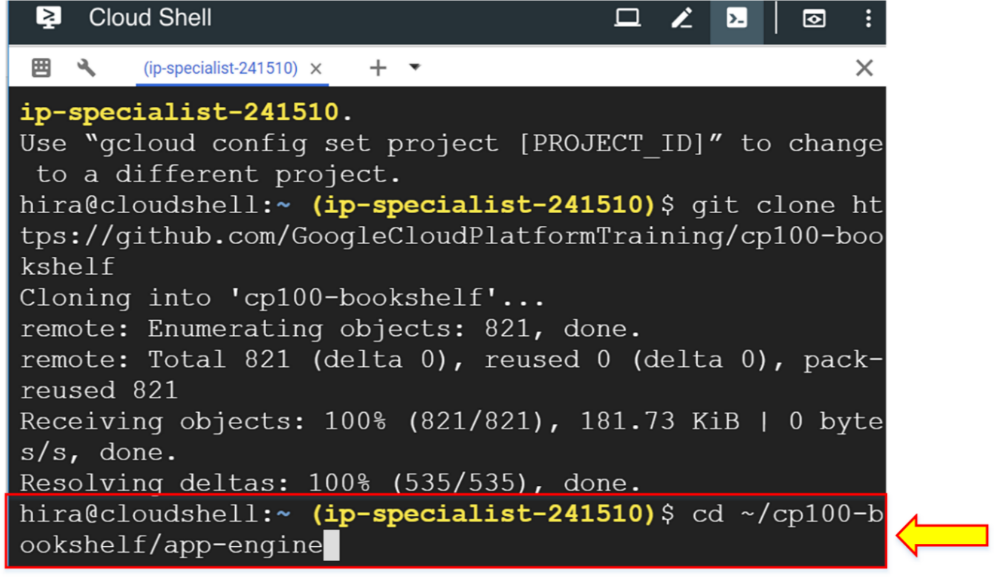


1. Go to “Cloud Shell” and enter the command “git clone <https://github.com/GoogleCloudPlatformTraining/cp100-bookshelf>”.

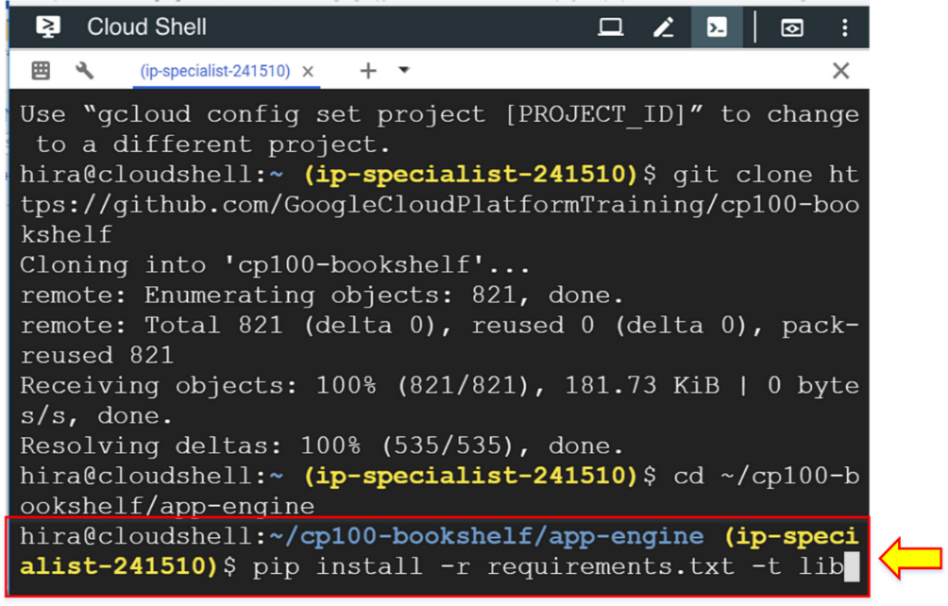
“cd ~/cp100-bookshelf/app-engine”. “cd ~/cp100-bookshelf/app-engine”.



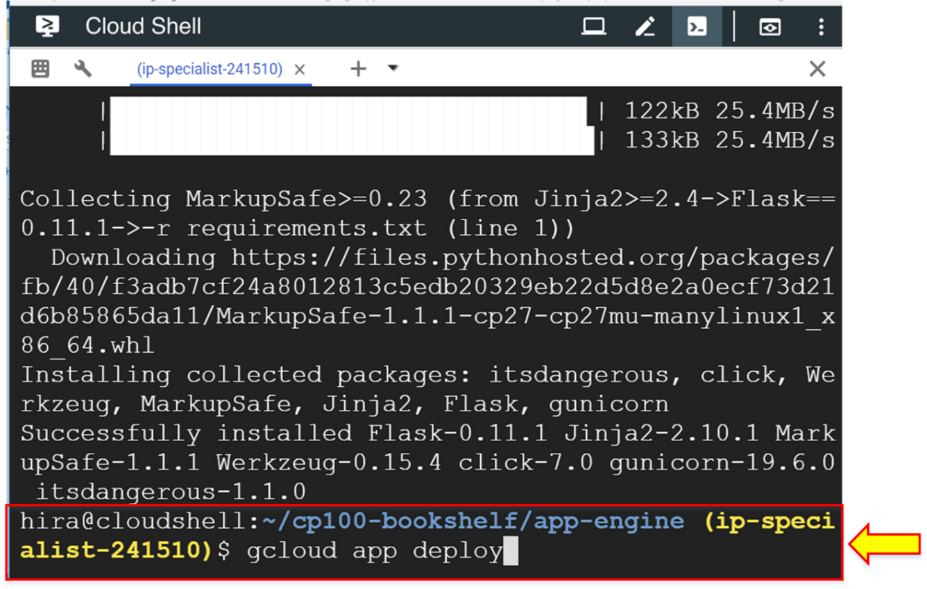
1. Enter the command “cd ~/cp100-bookshelf/app-engine” to move to the App engine directory.



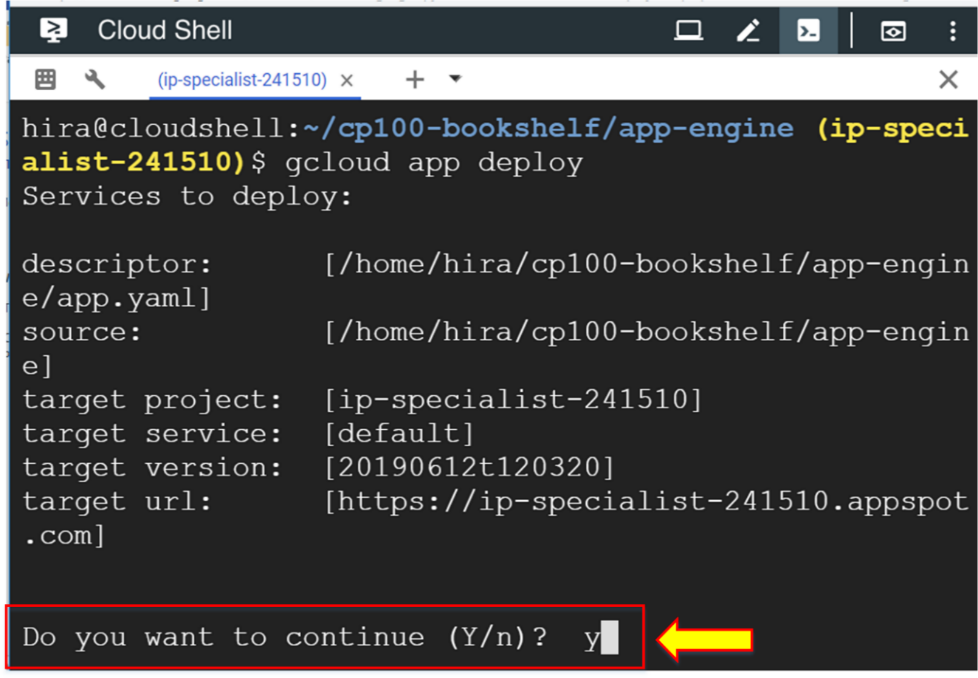
1. Enter the command “pip install -r requirements.txt -t lib”.



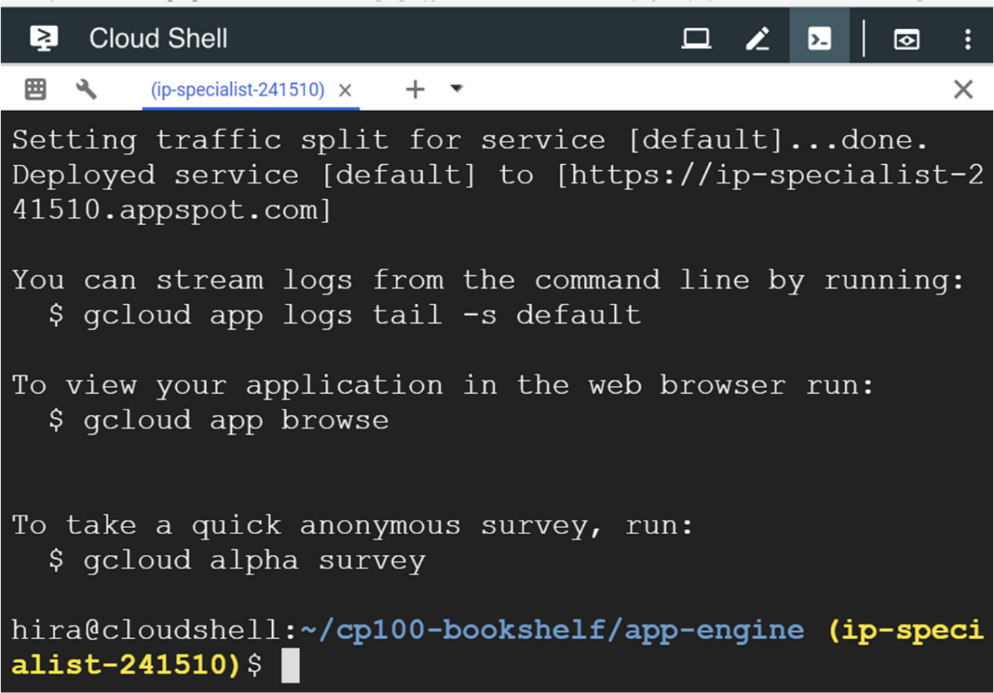
1. To deploy the application, enter the command “gcloud app deploy”.



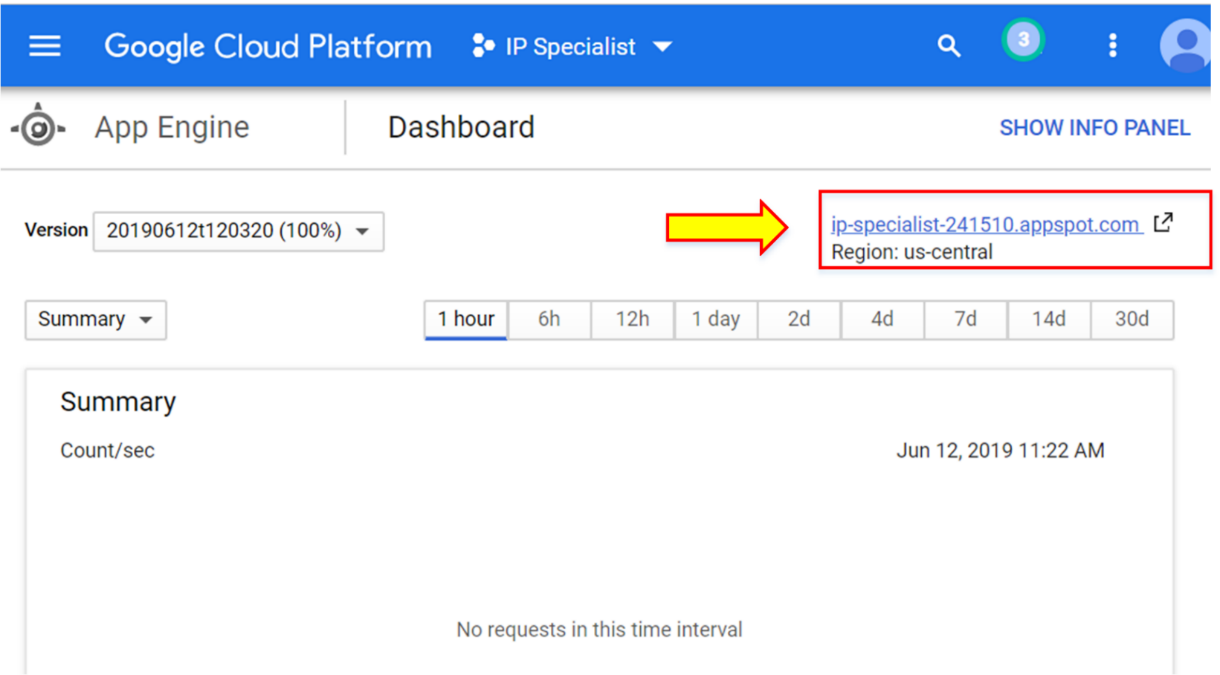
1. Enter “Y”.



The application has been deployed.



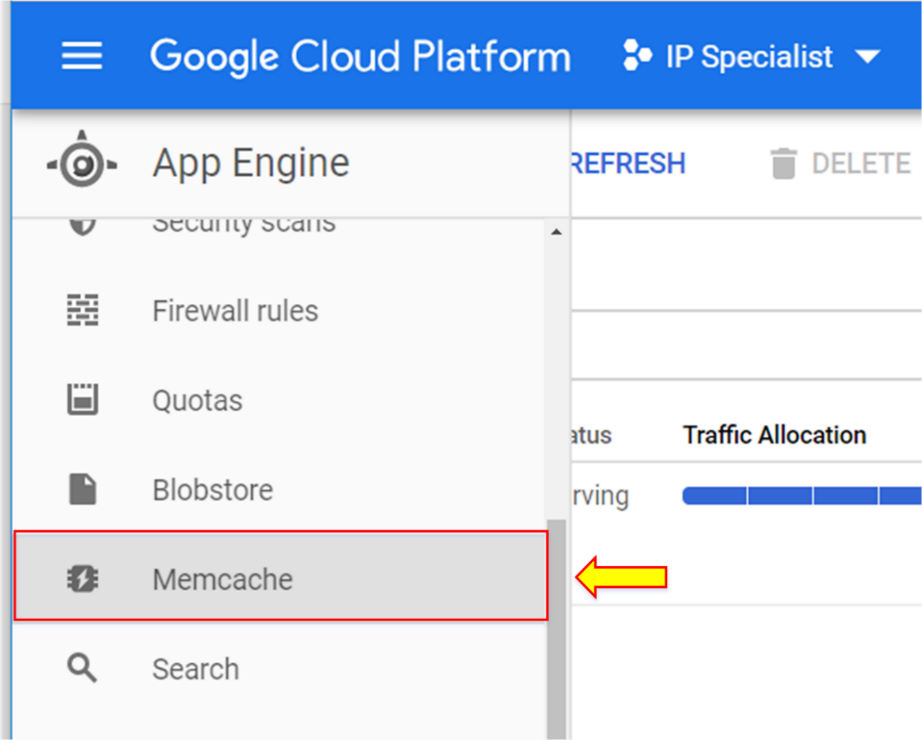
1. Click the link of the application in the “App Engine” dashboard.



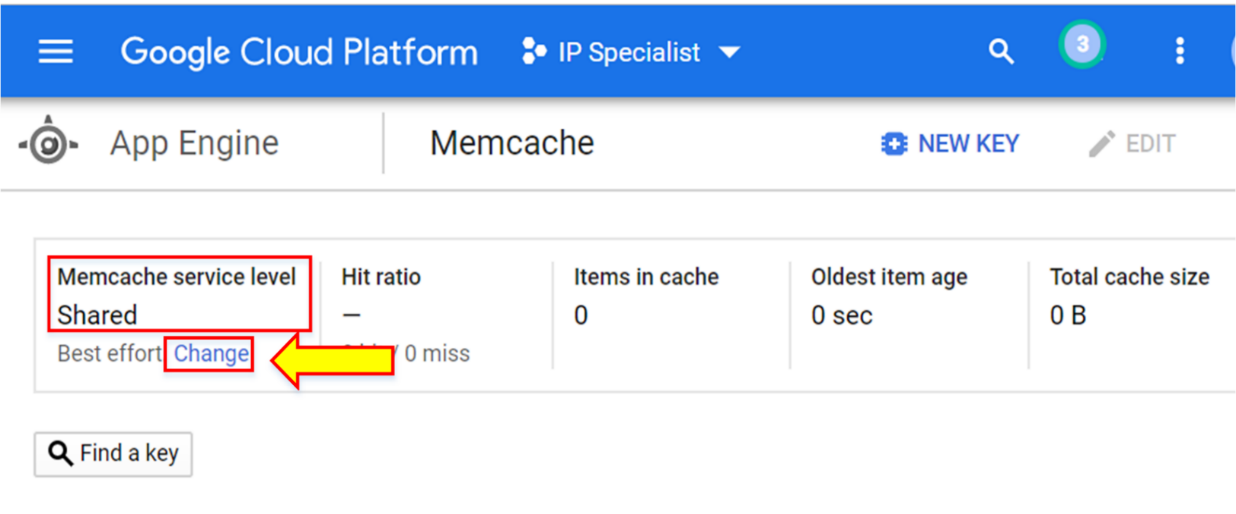
The application is now running.



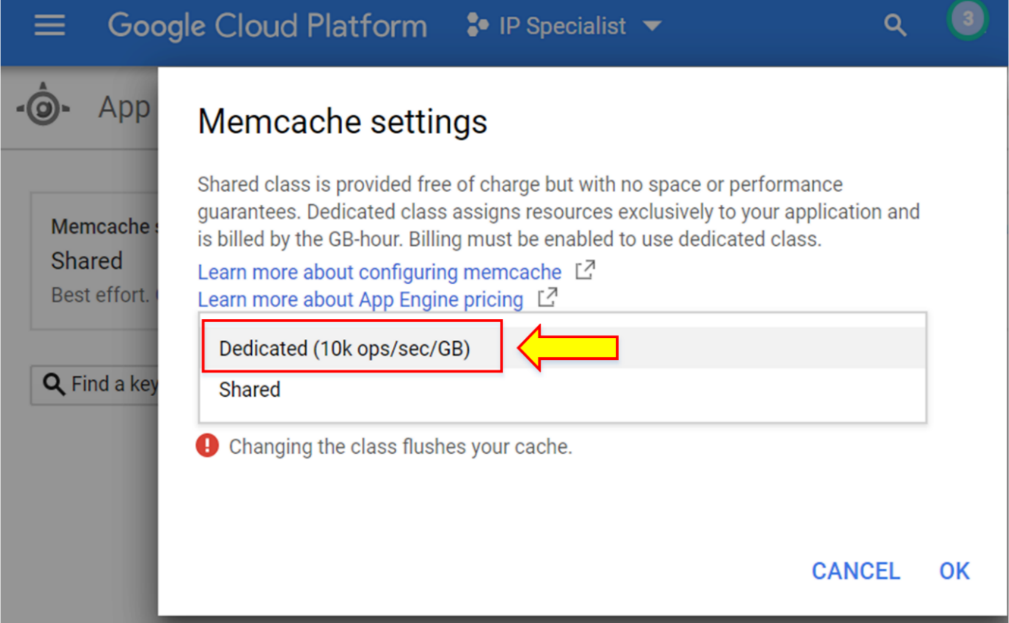
1. Select “Memcache”.



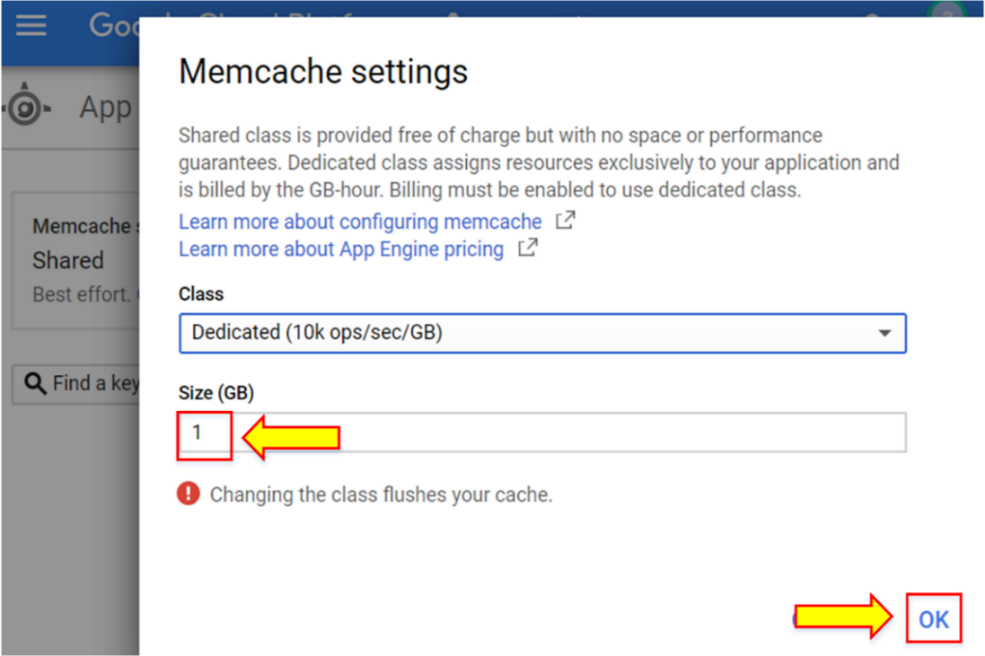
1. You will notice that Memcache is service level shared. Click “Change”.



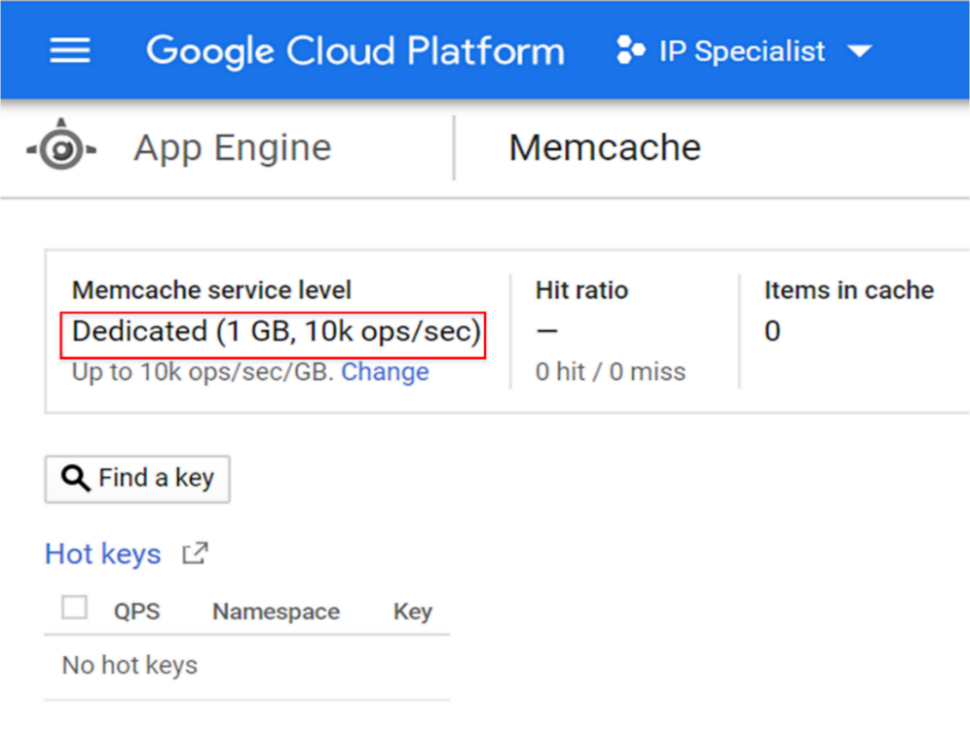
You can change the service level to “Dedicated”.



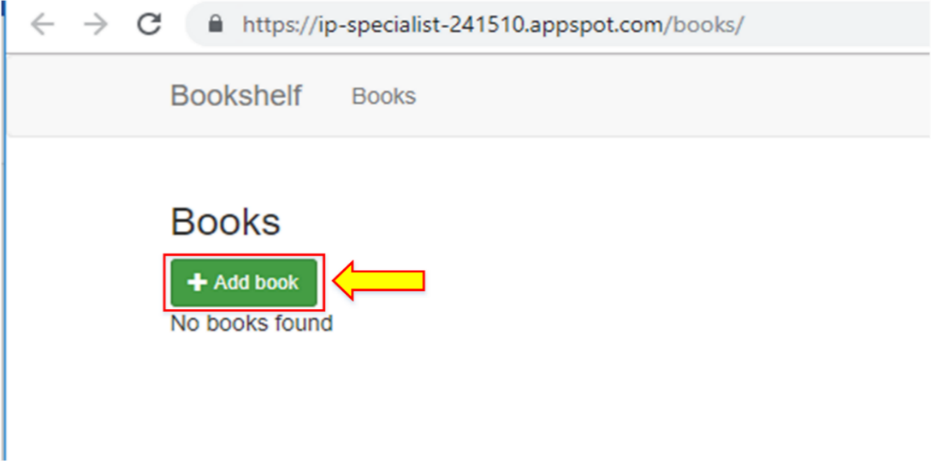
1. Define the size, then click “OK”.



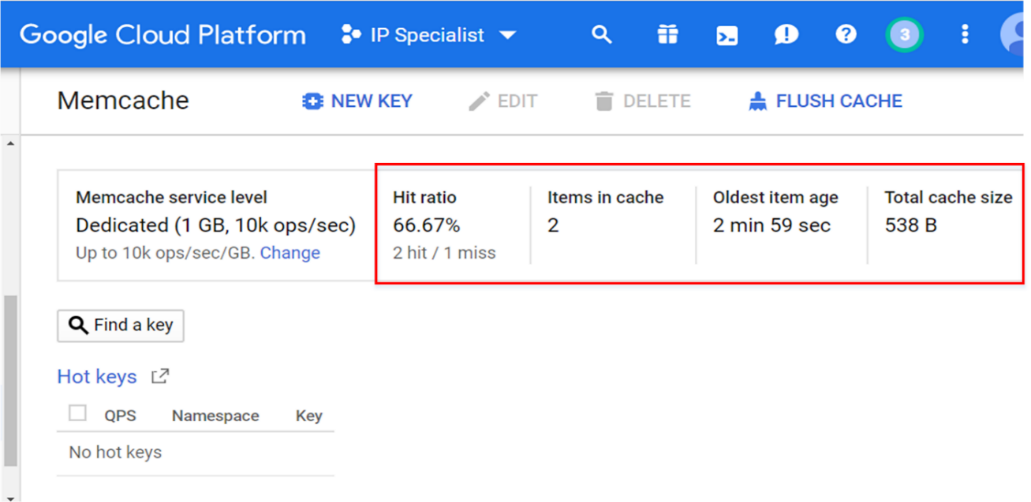
The Memcache service level has been changed to “Dedicated”.



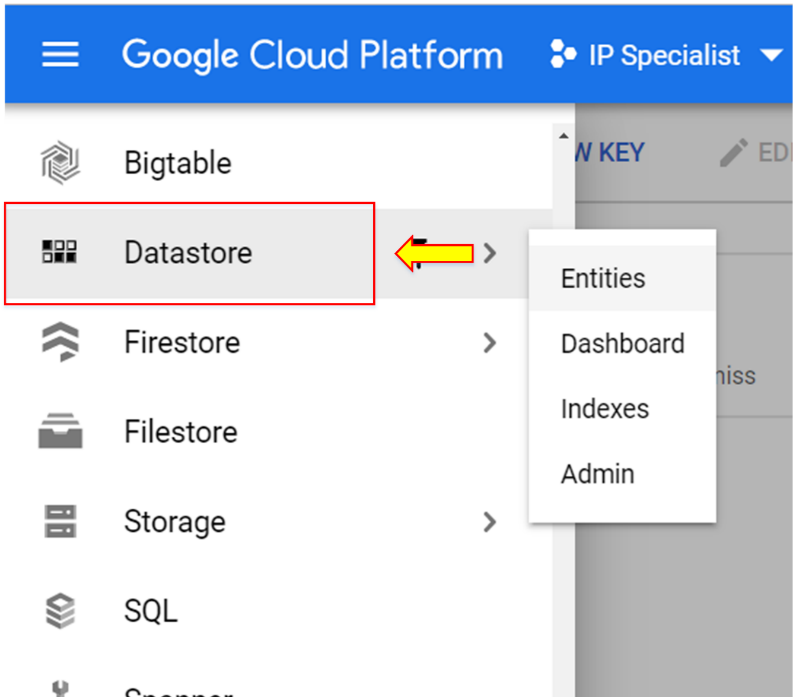
1. Go to the App and click “Add book” and add two books.



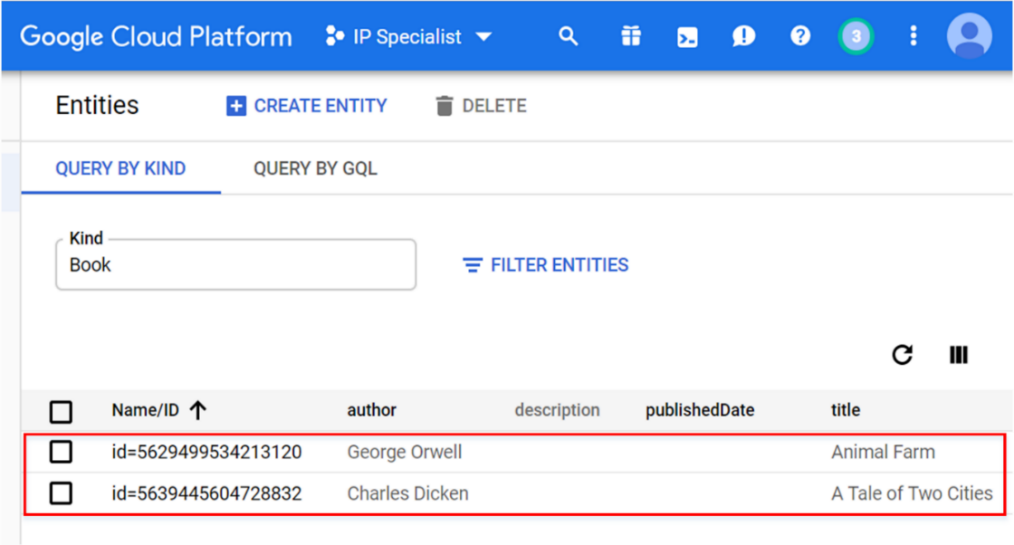
1. Go back to “Memcache”, where the information will be added. There are 2 items in the cache.



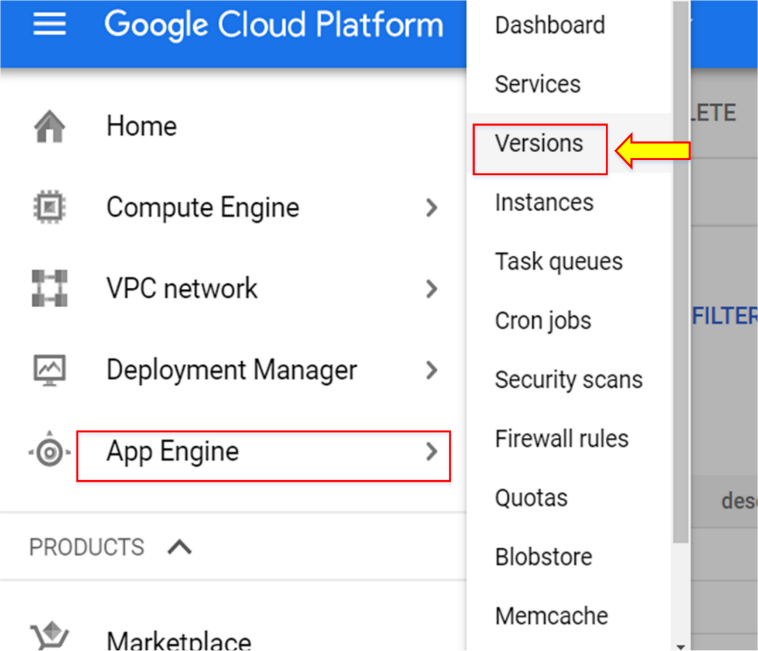
1. Go to “Datastore”.



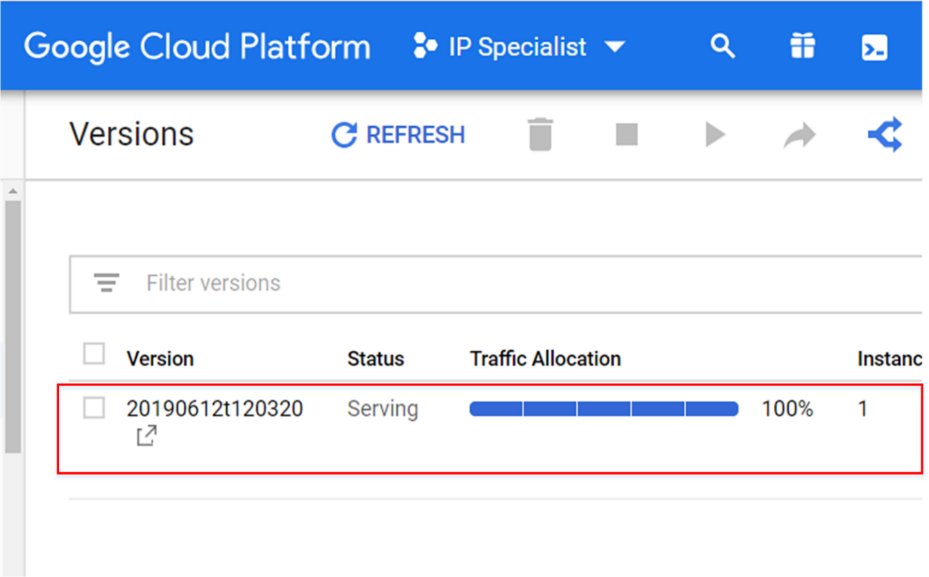
Two books entries are shown here.



1. Go to “Versions” in App Engine service.

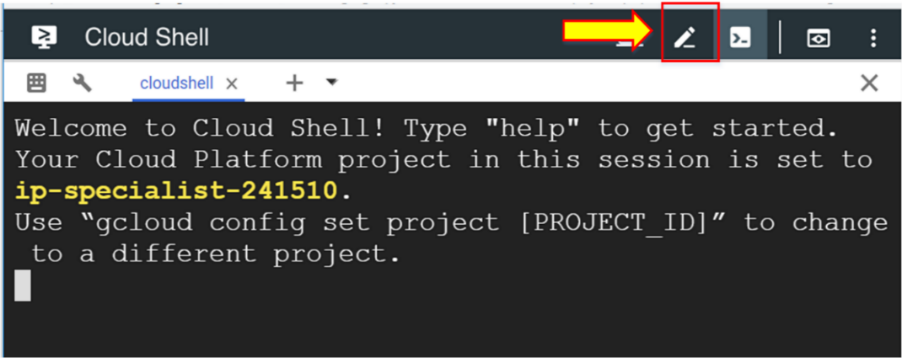


Only one version is available.

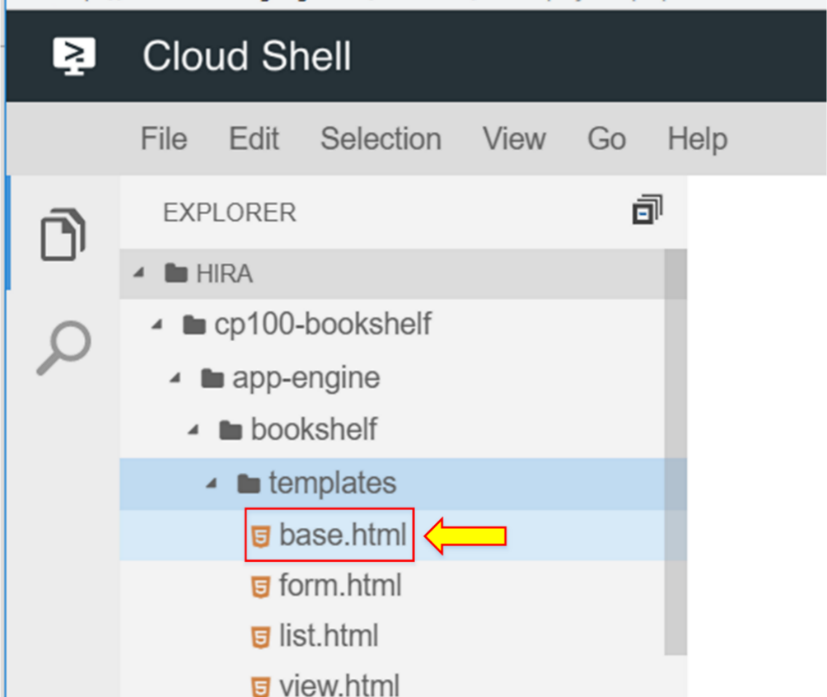


Creating a new version with risky updates

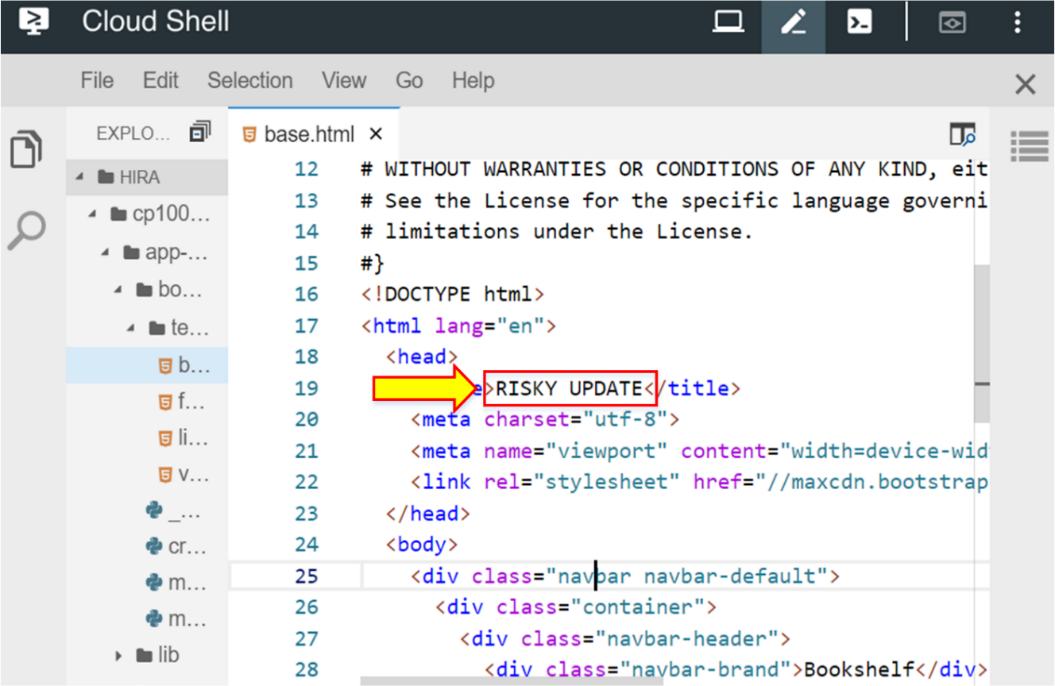
1. Go to Editor.



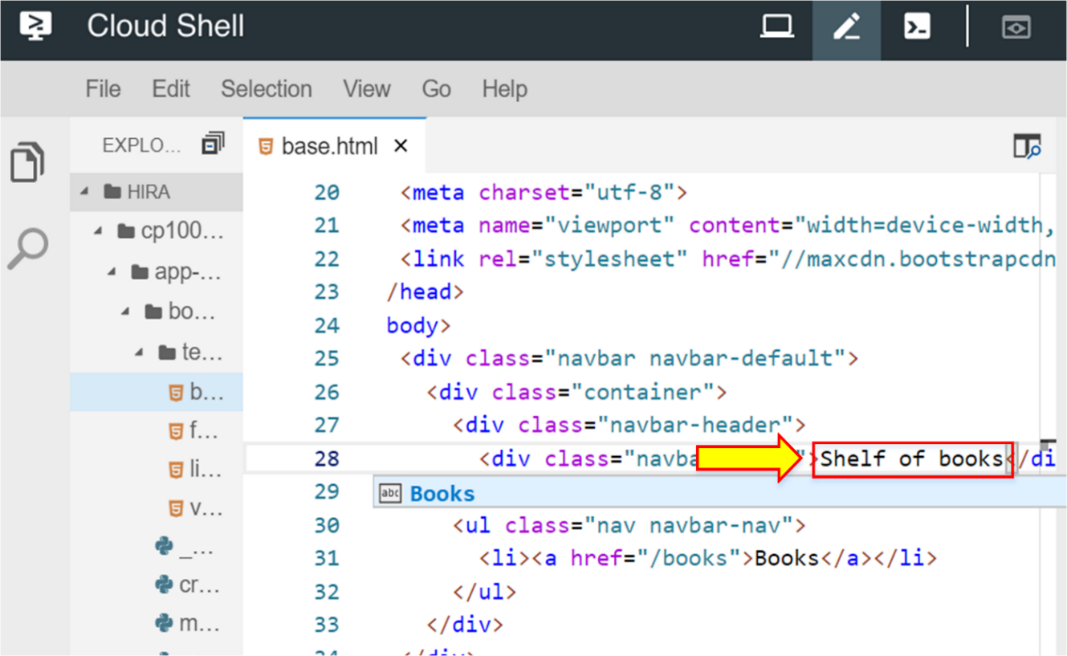
1. Select the “base.html” file.



1. Change the tile to “RISKY UPDATE”.

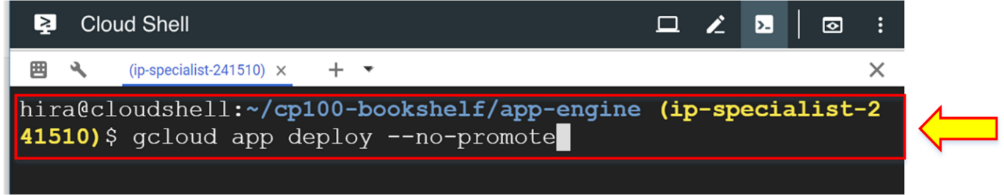


1. Replace “Bookshelf” with “Shelf of books”.

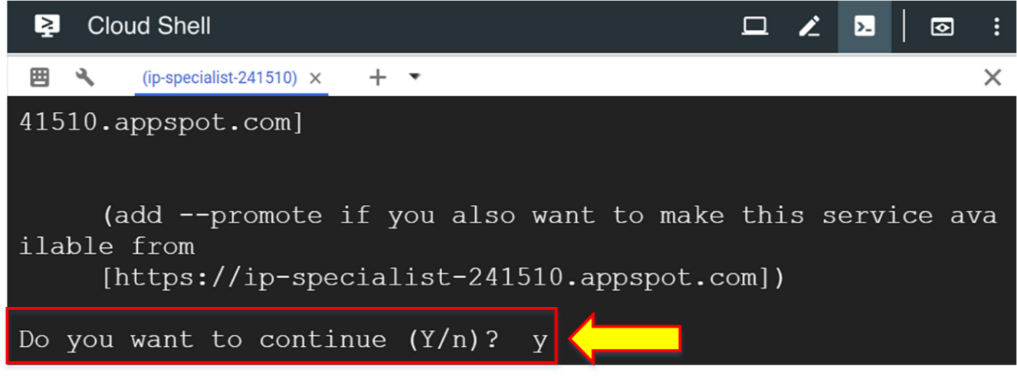


1. Go to Cloud Shell to deploy the second version of the application and enter the command “gcloud app deploy --no-promote”.

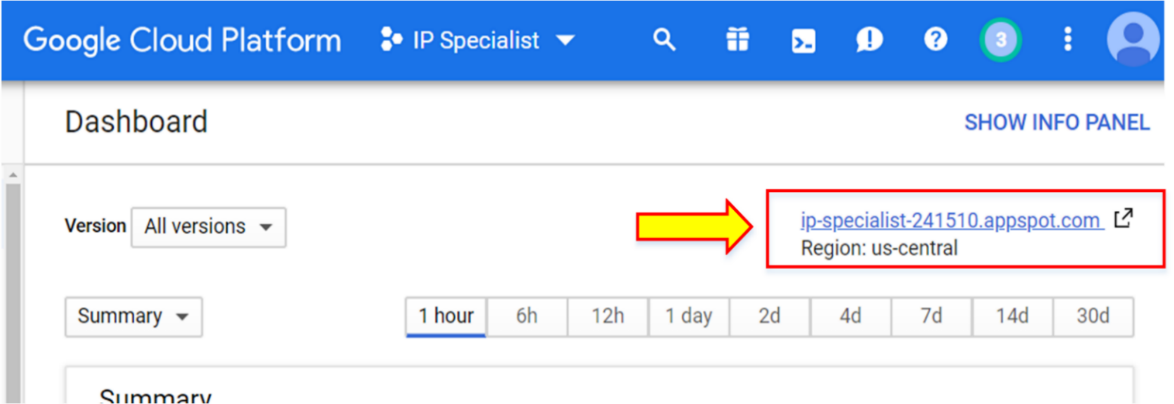
This command will deploy the new version and prevent directing all the traffic to the new versions with risky updates.



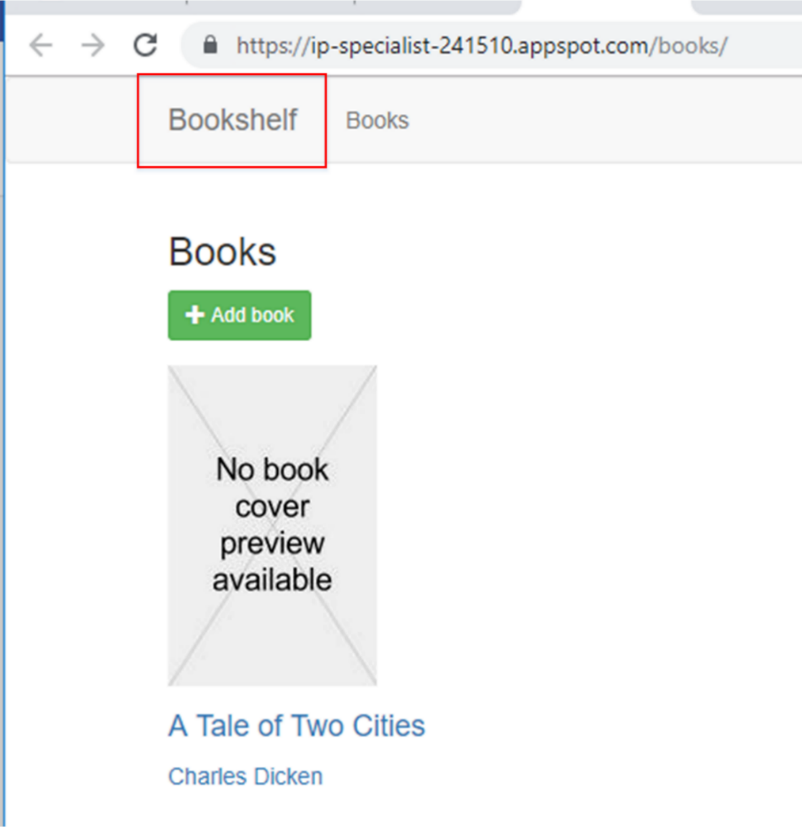
1. Enter “Y”.



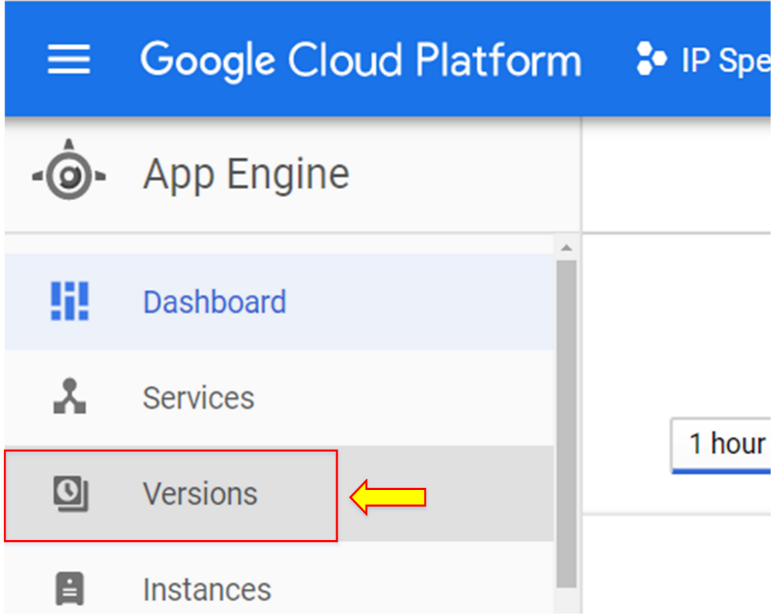
1. Go to the “App Engine” dashboard and click the link.



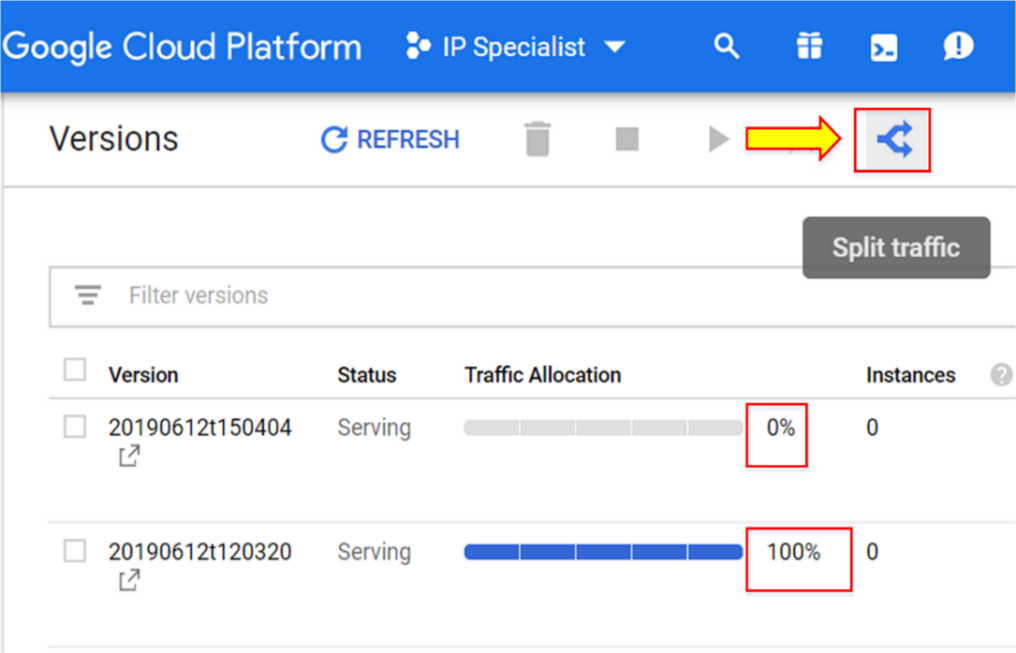
Traffic is still directed to the old version.



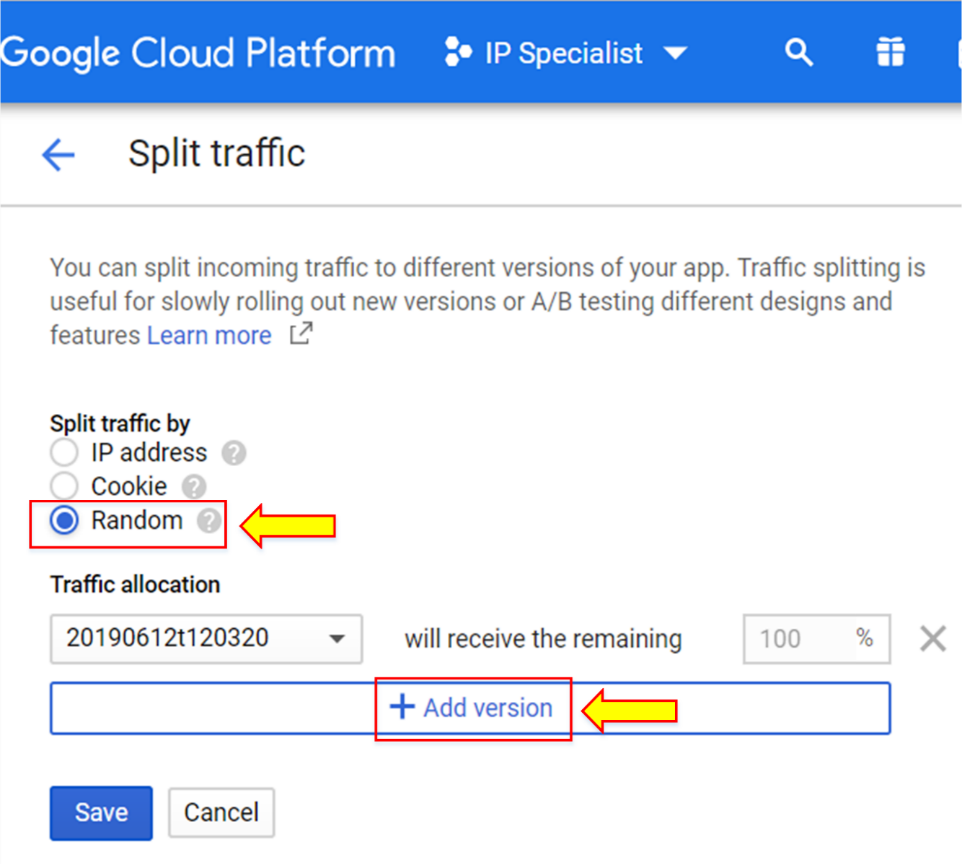
1. Go to “Versions”.



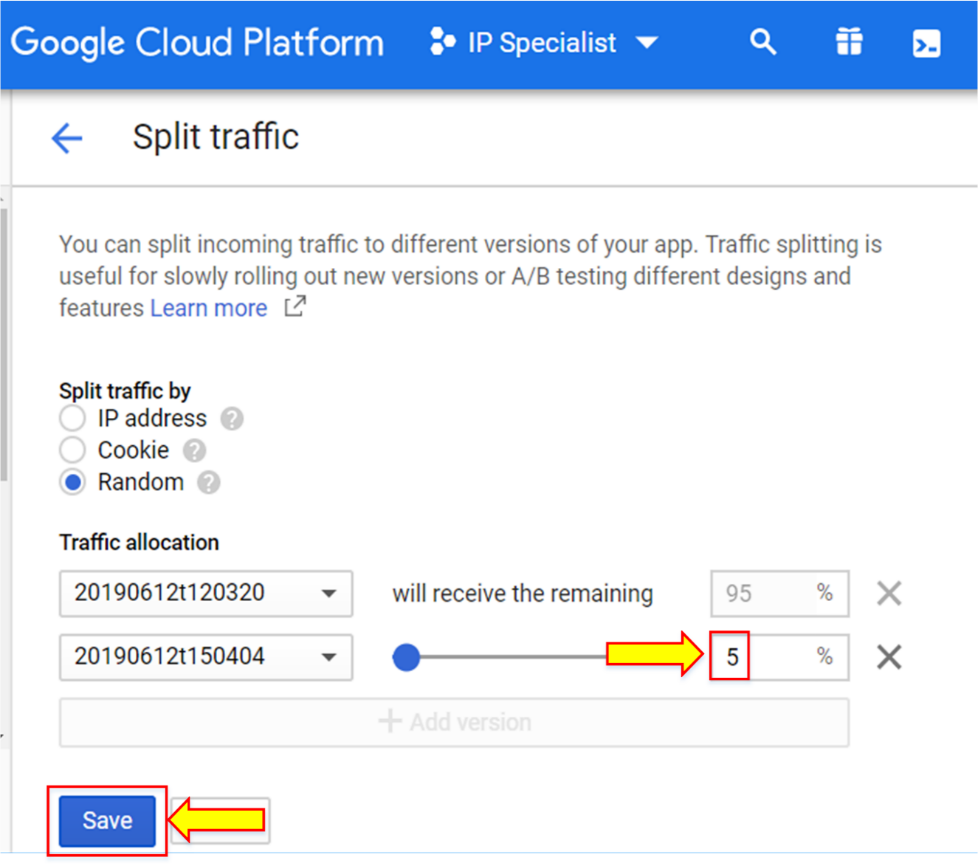
1. The old version is getting 100% of the traffic, while the risky update version has 0% of traffic. Click “Split traffic” to direct some traffic to the new version of the application before making it live for all traffic.



1. Select “Random” and click “Add version”.



1. Enter the amount of traffic as a percentage. Here, we have entered “5”.
2. Click “Save”.



1. Refresh your app window multiple times; it will direct you to the new version of the application.

