Kubernetes Engine: Qwik Start

30 minutes1 Credit
Rate Lab

GSP100

Google Cloud Self-Paced Labs

Overview

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Kubernetes Engine environment consists of multiple machines (specifically Compute Engine instances) grouped together to form a container cluster. In this lab, you get hands on practice with container creation and application deployment with GKE.

Cluster orchestration with Kubernetes Engine

Kubernetes Engine clusters are powered by the Kubernetes open source cluster management system. Kubernetes provides the mechanisms through which you interact with your container cluster. You use Kubernetes commands and resources to deploy and manage your applications, perform administration tasks and set policies, and monitor the health of your deployed workloads.

Kubernetes draws on the same design principles that run popular Google services and provides the same benefits: automatic management, monitoring and liveness probes for application containers, automatic scaling, rolling updates, and more. When you run your applications on a container cluster, you're using technology based on Google's 10+ years of experience running production workloads in containers.

Kubernetes on Google Cloud

When you run a Kubernetes Engine cluster, you also gain the benefit of advanced cluster management features that Google Cloud provides. These include:

- Load-balancing for Compute Engine instances.
- Node Pools to designate subsets of nodes within a cluster for additional flexibility.
- Automatic scaling of your cluster's node instance count.
- Automatic upgrades for your cluster's node software.
- Node auto-repair to maintain node health and availability.
- Logging and Monitoring with Cloud Monitoring for visibility into your cluster.

Now that you have a basic understanding of Kubernetes, you learn how to deploy a containerized application with Kubernetes Engine in less than 30 minutes. Scroll down and follow the steps below to get your lab environment set up.

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.

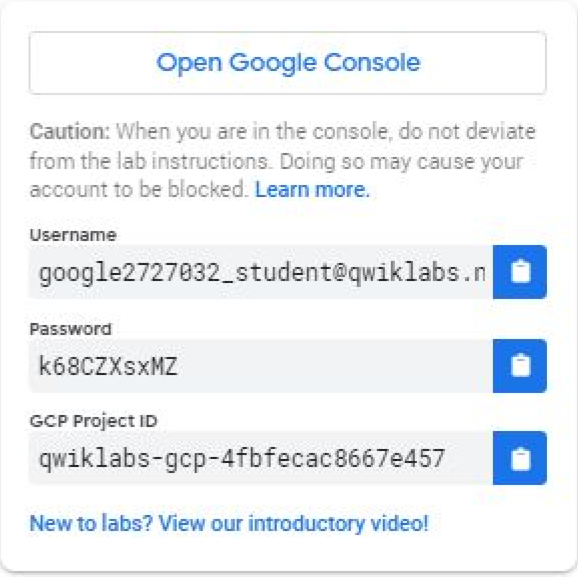
Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

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.



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.

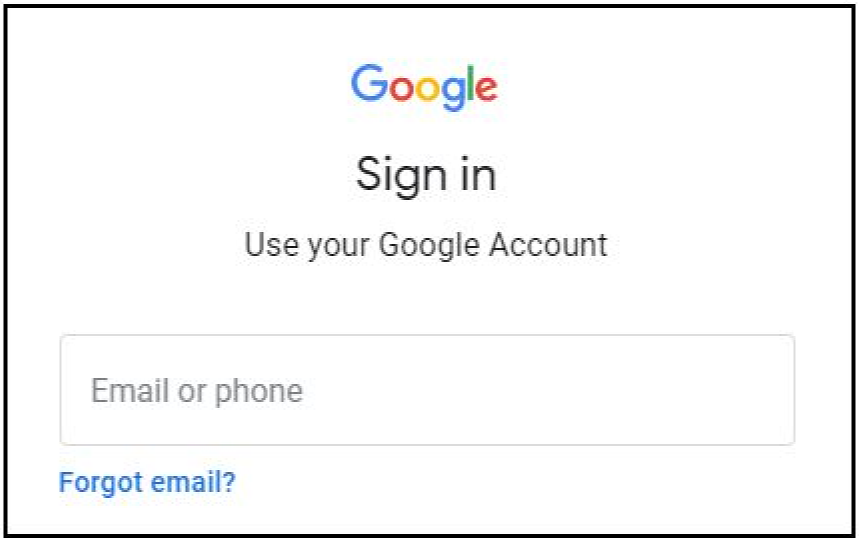
Username
google2727032_student@qwiklabs.n

Password
k68CZXsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

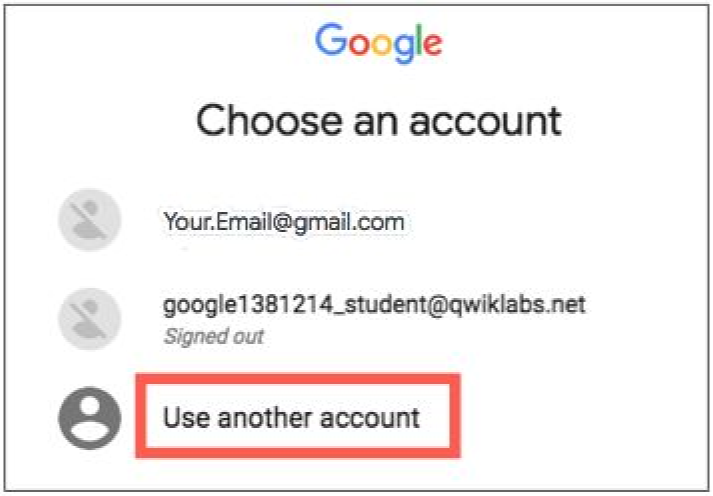
New to labs? View our introductory video!

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**.



Google

Choose an account

Your.Email@gmail.com

google1381214_student@qwiklabs.net
Signed out

Use another account

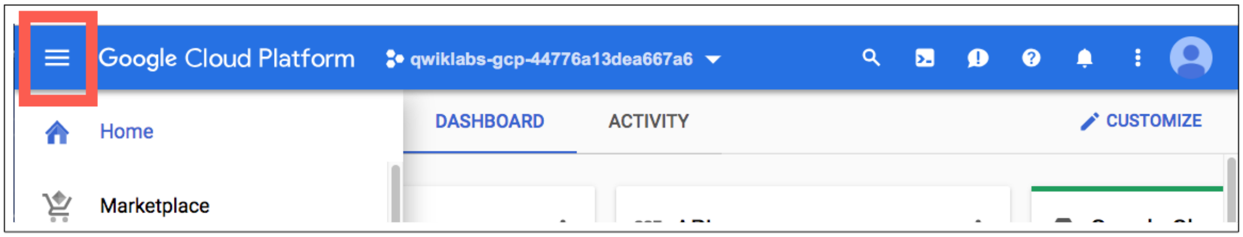
3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

   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.

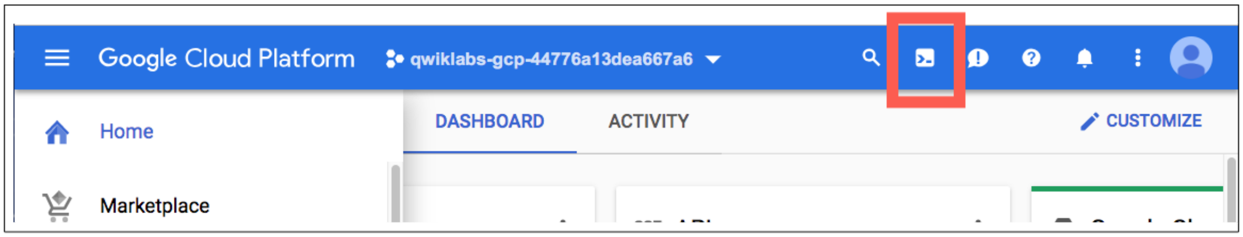
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



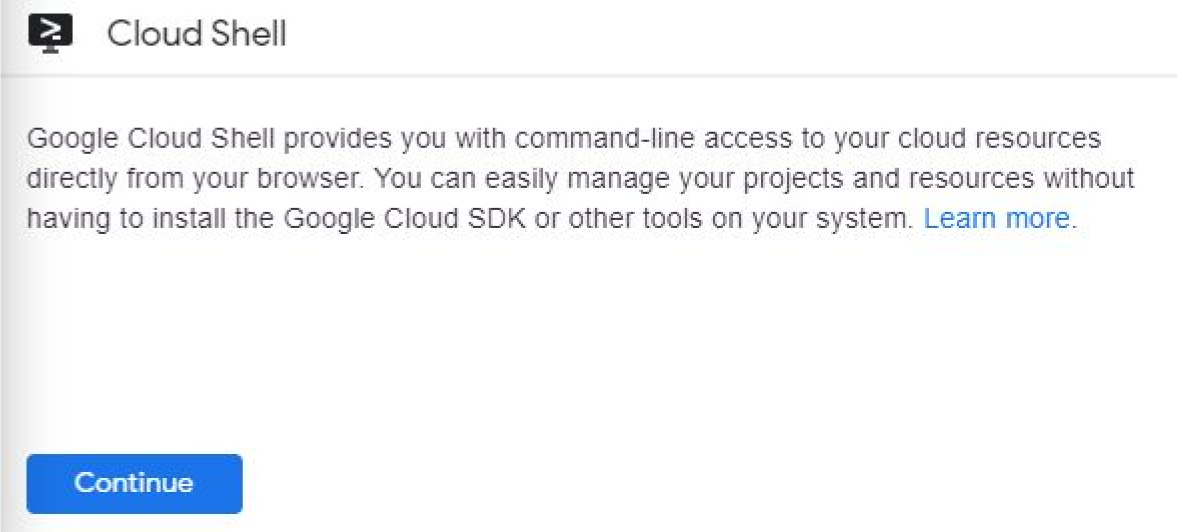
Activate Cloud Shell

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.



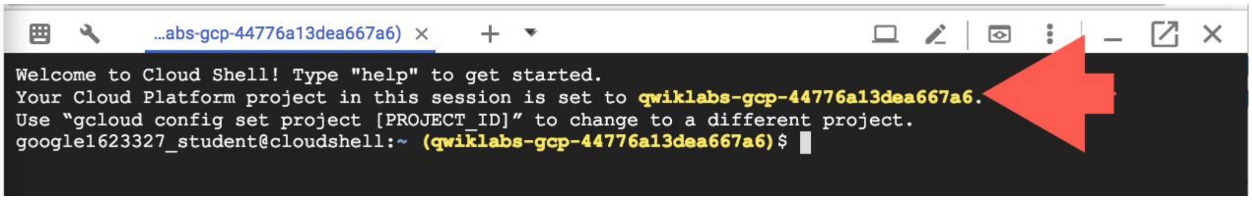
Click **Continue**.



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.

Setting a default compute zone

Your compute zone is an approximate regional location in which your clusters and their resources live. For example, `us-central1-a` is a zone in the `us-central1` region.

Start a new session in Cloud Shell and run the following command to set your default compute zone to `us-central1-a`:

```
gcloud config set compute/zone us-central1-a
```

You receive the following output:

```
Updated property [compute/zone].
```

Creating a Kubernetes Engine cluster

A cluster consists of at least one *cluster master* machine and multiple worker machines called *nodes*. Nodes are Compute Engine virtual machine (VM) instances that run the Kubernetes processes necessary to make them part of the cluster.

To create a cluster, run the following command, replacing `[CLUSTER-NAME]` with the name you choose for the cluster (for example `my-cluster`). Cluster names must start with a letter, end with an alphanumeric, and cannot be longer than 40 characters.

```
gcloud container clusters create [CLUSTER-NAME]
```

You can ignore any warnings in the output. It might take several minutes to finish creating the cluster. Soon after you should receive a similar output:

```
NAME        LOCATION       ...   NODE_VERSION  NUM_NODES  STATUS
my-cluster  us-central1-a  ...   1.15.12-gke.2  3          RUNNING
```

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

Create a Kubernetes Engine cluster

Check my progress

Get authentication credentials for the cluster

After creating your cluster, you need to get authentication credentials to interact with the cluster.

To authenticate the cluster run the following command, replacing `[CLUSTER-NAME]` with the name of your cluster:

```
gcloud container clusters get-credentials [CLUSTER-NAME]
```

You should receive a similar output:

```
Fetching cluster endpoint and auth data.
kubeconfig entry generated for my-cluster.
```

Deploying an application to the cluster

Now that you have created a cluster, you can deploy a containerized application to it. For this lab you'll run `hello-app` in your cluster.

Kubernetes Engine uses Kubernetes objects to create and manage your cluster's resources. Kubernetes provides the Deployment object for deploying stateless applications like web servers. Service objects define rules and load balancing for accessing your application from the Internet.

Run the following kubectl create command in Cloud Shell to create a new Deployment `hello-server` from the `hello-app` container image:

```
kubectl create deployment hello-server
--image=gcr.io/google-samples/hello-app:1.0
```

You should receive the following output:

```
deployment.apps/hello-server created
```

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

Create a new Deployment - hello-server

Check my progress

This Kubernetes command creates a Deployment object that represents `hello-server`. In this case, `--image` specifies a container image to deploy. The command pulls the example image from a Google Container Registry bucket. `gcr.io/google-samples/hello-app:1.0` indicates the specific image version to pull. If a version is not specified, the latest version is used.

Now create a Kubernetes Service, which is a Kubernetes resource that lets you expose your application to external traffic, by running the following kubectl expose command:

```
kubectl expose deployment hello-server --type=LoadBalancer --port 8080
```

In this command:

- `--port` specifies the port that the container exposes.
- `type="LoadBalancer"` creates a Compute Engine load balancer for your container.

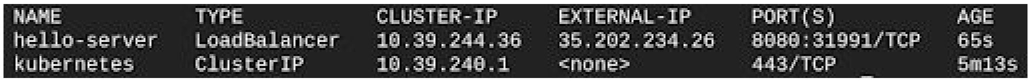
You should receive the following output:

```
service/hello-server exposed
```

Inspect the `hello-server` Service by running kubectl get:

```
kubectl get service
```

You should receive a similar output:



```
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)          AGE
hello-server   LoadBalancer   10.39.244.36   35.202.234.26   8080:31991/TCP   65s
kubernetes     ClusterIP      10.39.240.1    <none>          443/TCP          5m13s
```

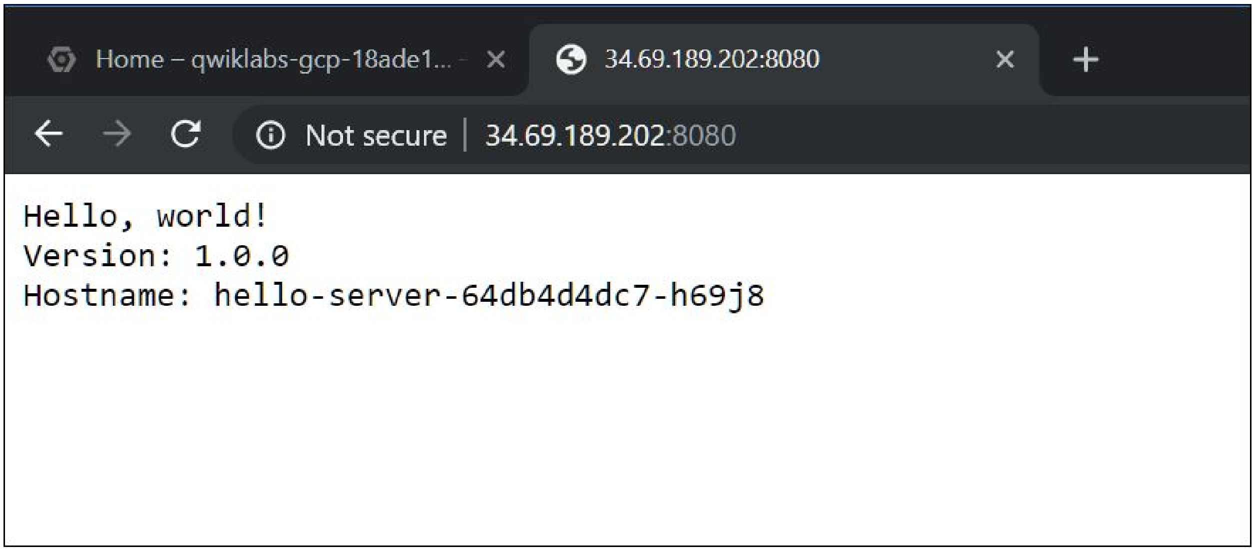
Note: It might take a minute for an external IP address to be generated. Run the above command again if the `EXTERNAL-IP` column is in "pending" status.

From this command's output, copy the Service's external IP address from the `EXTERNAL IP` column.

View the application from your web browser using the external IP address with the exposed port:

```
http://[EXTERNAL-IP]:8080
```

Your page should resemble the following:



```
Hello, world!
Version: 1.0.0
Hostname: hello-server-64db4d4dc7-h69j8
```

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

Create a Kubernetes Service

Check my progress

Clean Up

Run the following to delete the cluster:

```
gcloud container clusters delete [CLUSTER-NAME]
```

When prompted, type **Y** to confirm. Deleting the cluster can take a few minutes. For more information on deleted Google Kubernetes Engine clusters, view the documentation.

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

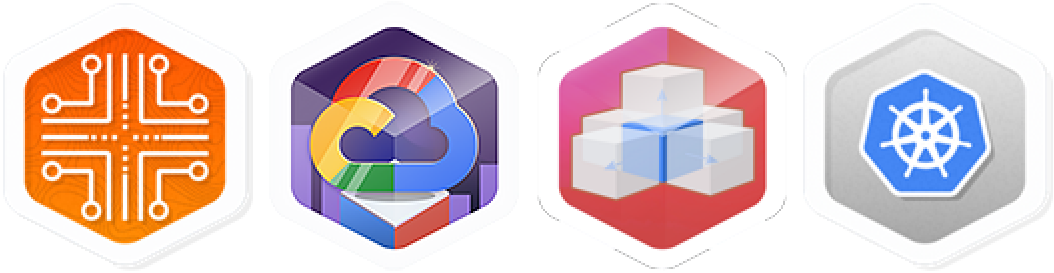
Clean up: Delete the cluster

Check my progress

Congratulations!

You have just deployed a containerized application to Kubernetes Engine!

Finish Your Quest



Continue your Baseline: Infrastructure, Google Cloud Essentials, Kubernetes in Google Cloud, and VM Migration 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 a Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Next Steps / 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 September 9, 2020

Lab Last Tested September 9, 2020