

**—/100**

**Checkpoints**

Creating a Kubernetes Engine cluster

Check my progress

/ 25

Create a new Deployment - hello-server

Check my progress

/ 25

Create a Kubernetes Service

Check my progress

/ 25

Clean up: Delete the cluster

Check my progress

/ 25

**Google Kubernetes Engine: Qwik Start [ACE]**

30 minutes Free

Rate Lab

**Overview**

[Google Kubernetes Engine](https://cloud.google.com/kubernetes-engine/) (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The GKE environment consists of multiple machines (specifically [Google Compute Engine](https://cloud.google.com/compute) instances) grouped together to form a [container cluster](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture). In this lab, you will get hands on practice with container creation and application deployment with GKE.

**Cluster orchestration with GKE**

GKE clusters are powered by the [Kubernetes](https://kubernetes.io/) 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 GKE cluster, you also gain the benefit of advanced cluster management features that Google Cloud provides. These include:

* [Load-balancing](https://cloud.google.com/compute/docs/load-balancing-and-autoscaling) for Compute Engine instances.
* [Node Pools](https://cloud.google.com/kubernetes-engine/docs/node-pools) to designate subsets of nodes within a cluster for additional flexibility.
* [Automatic scaling](https://cloud.google.com/kubernetes-engine/docs/cluster-autoscaler) of your cluster's node instance count.
* [Automatic upgrades](https://cloud.google.com/kubernetes-engine/docs/node-auto-upgrade) for your cluster's node software.
* [Node auto-repair](https://cloud.google.com/kubernetes-engine/docs/node-auto-repair) to maintain node health and availability.
* [Cloud Logging and Monitoring](https://cloud.google.com/kubernetes-engine/docs/how-to/logging) for visibility into your cluster.

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

**Setup and Requirements**

**Qwiklabs setup**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.pngand make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

**Activate Google Cloud Shell**

Google 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. Google Cloud Shell provides command-line access to your GCP resources.

1. In GCP console, on the top right toolbar, click the Open Cloud Shell button.



1. 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:



**gcloud** is the command-line tool for Google Cloud Platform. 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

Full documentation of **gcloud** is available on [Google Cloud gcloud Overview](https://cloud.google.com/sdk/gcloud) .

**Setting a default compute zone**

Your [compute zone](https://cloud.google.com/compute/docs/regions-zones/#available) 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 will receive the following output:

Updated property [compute/zone].

**Creating a GKE cluster**

A [cluster](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture) consists of at least one *cluster master* machine and multiple worker machines called *nodes*. Nodes are [Compute Engine virtual machine (VM) instances](https://cloud.google.com/compute/docs/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 ... 1.14.10-gke.27 3 RUNNING

Click **Check my progress** to verify the objective. Create a Google Kubernetes Engine cluster

**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](https://cloud.google.com/kubernetes-engine/docs/concepts/kubernetes-engine-overview) to it. For this lab you'll run hello-app in your cluster.

GKE uses Kubernetes objects to create and manage your cluster's resources. Kubernetes provides the [Deployment](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/) object for deploying stateless applications like web servers. [Service](https://kubernetes.io/docs/concepts/services-networking/service/) objects define rules and load balancing for accessing your application from the Internet.

Run the following [kubectl run](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#run) 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

This Kubernetes command creates a Deployment object that represents hello-app. In this command:

* --image specifies a container image to deploy. In this case, the command pulls the example image from a [Google Container Registry](https://cloud.google.com/container-registry/docs) 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.
* --port specifies the port that the container exposes.

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](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#expose) command:

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

You should receive the following output:

service/hello-server exposed

Click **Check my progress** to verify the objective. Create a new Deployment - hello-server

Passing in type="LoadBalancer" creates a Compute Engine load balancer for your container.

Inspect the hello-server Service by running [kubectl get](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get):

kubectl get service hello-server

You should receive a similar output:

NAME TYPE ... EXTERNAL-IP PORT(S) AGE

hello-server LoadBalancer ... 35.184.112.169 8080:32069/TCP 2m

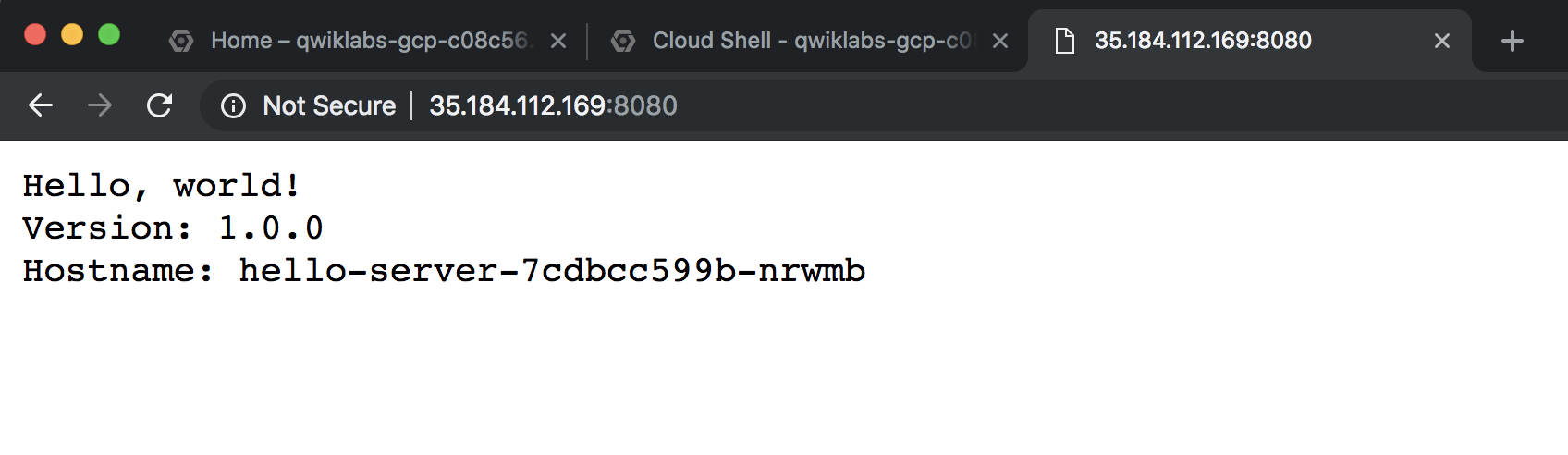
**Note:** It might take a minute for an external IP address to be generated—run the above command again if the EXTERNAL-IP column is still pending.

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:



Click **Check my progress** to verify the objective. Create a Kubernetes Service

**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 GKE clusters, view the [documentation](https://cloud.google.com/kubernetes-engine/docs/how-to/deleting-a-cluster).

Click **Check my progress** to verify the objective. Clean up: Delete the cluster

**End your lab**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 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.