**Task 1: Set 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.

1. To **set your default compute zone** to us-central1-a, start a new session in Cloud Shell, and run the following command:

gcloud config set compute/zone us-central1-a

Copied!

content\_copy

**Expected output** (Do not copy):

Updated property [compute/zone].

**Task 2: Create 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.

**Note:**Cluster names must start with a letter and end with an alphanumeric, and cannot be longer than 40 characters.

1. To **create a cluster**, run the following command, replacing [CLUSTER-NAME] with the name you choose for the cluster (**for example**:my-cluster).

gcloud container clusters create [CLUSTER-NAME]

Copied!

content\_copy

You can ignore any warnings in the output. It might take several minutes to finish creating the cluster.

**Expected output** (Do not copy):

NAME: my-cluster

LOCATION: us-central1-a

MASTER\_VERSION: 1.21.5-gke.1302

MASTER\_IP: 34.69.232.119

MACHINE\_TYPE: e2-medium

NODE\_VERSION: 1.21.5-gke.1302

NUM\_NODES: 3

STATUS: RUNNING

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

Create a GKE cluster

Check my progress

**Task 3: Get authentication credentials for the cluster**

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

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

gcloud container clusters get-credentials [CLUSTER-NAME]

Copied!

content\_copy

**Expected output** (Do not copy):

Fetching cluster endpoint and auth data.

kubeconfig entry generated for my-cluster.

**Task 4: Deploy an application to the cluster**

You can now deploy a [containerized application](https://cloud.google.com/kubernetes-engine/docs/concepts/kubernetes-engine-overview) to the cluster. 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.

1. To **create a new Deployment** hello-server from the hello-app container image, run the following [kubectl create](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands" \l "create" \t "_blank) command:

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

Copied!

content\_copy

**Expected output** (Do not copy):

deployment.apps/hello-server created

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 [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.

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

Create a new Deployment: hello-server

Check my progress

1. To **create a Kubernetes Service**, which is a Kubernetes resource that lets you expose your application to external traffic, run the following [kubectl expose](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands" \l "expose" \t "_blank) command:

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

Copied!

content\_copy

In this command:

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

**Expected output** (Do not copy):

service/hello-server exposed

1. To **inspect** the hello-server Service, run [kubectl get](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands" \l "get" \t "_blank):

kubectl get service

Copied!

content\_copy

**Expected output** (Do not copy):

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> 433/TCP 5m13s

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

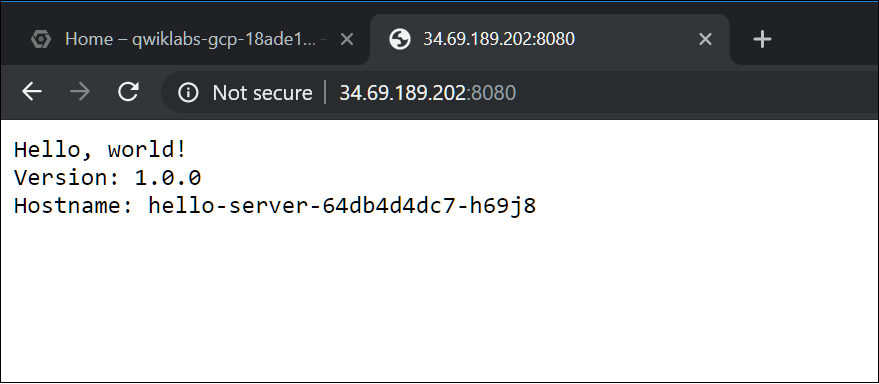
1. To view the application from your web browser, open a new tab and enter the following address, replacing [EXTERNAL IP] with the EXTERNAL-IP for hello-server.

http://[EXTERNAL-IP]:8080

Copied!

content\_copy

**Expected output**:



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

Create a Kubernetes Service

Check my progress

**Task 5: Deleting the cluster**

1. To **delete** the cluster, run the following command:

gcloud container clusters delete [CLUSTER-NAME]

Copied!

content\_copy

1. 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.

Delete the cluster

Check my progress

**Congratulations!**