Create and Manage Cloud Resources: Challenge Lab

GSP313

Google Cloud Self-Paced Labs

Overview

In a challenge lab you're given a scenario and a set of tasks. Instead of following step-by-step instructions, you will use the skills learned from the labs in the quest to figure out how to complete the tasks on your own! An automated scoring system (shown on this page) will provide feedback on whether you have completed your tasks correctly.

When you take a challenge lab, you will not be taught new Google Cloud concepts. You are expected to extend your learned skills, like changing default values and reading and researching error messages to fix your own mistakes.

To score 100% you must successfully complete all tasks within the time period!

This lab is recommended for students who have enrolled in the labs in the Create and Manage Cloud Resources quest. Be sure to review those labs before starting this lab. Are you ready for the challenge?
Topics tested:

- Create an instance
- Create a 3-node Kubernetes cluster and run a simple service
- Create an HTTP(s) load balancer in front of two web servers

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

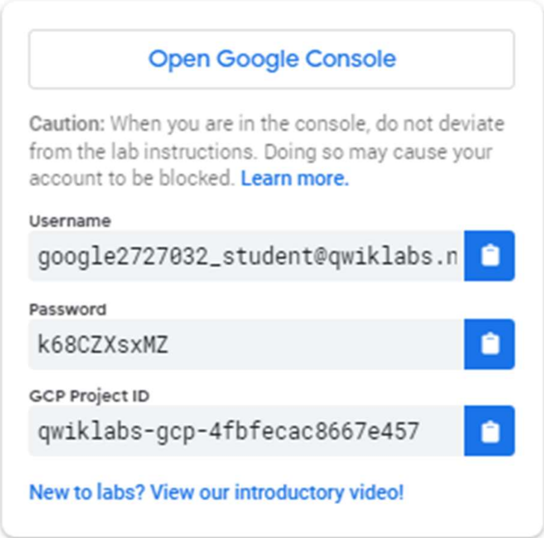
To complete this lab, you need:

- 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.

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.

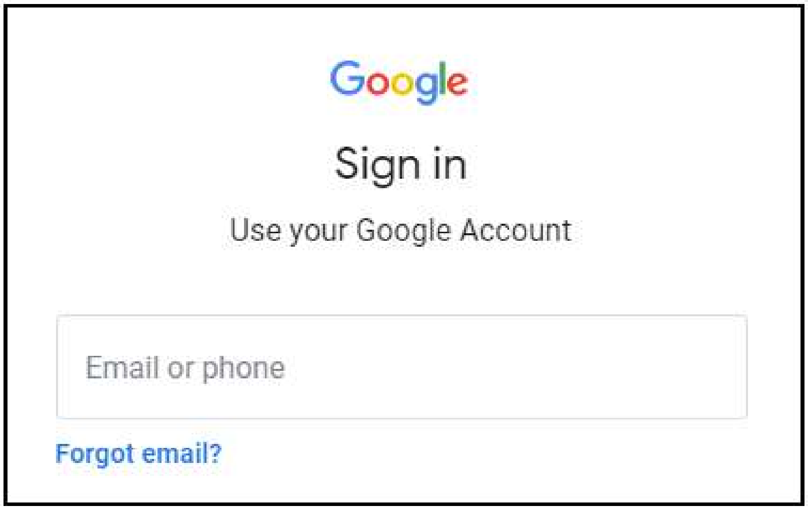
Username
google2727032_student@qwiklabs.n

Password
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.



Google

Sign in

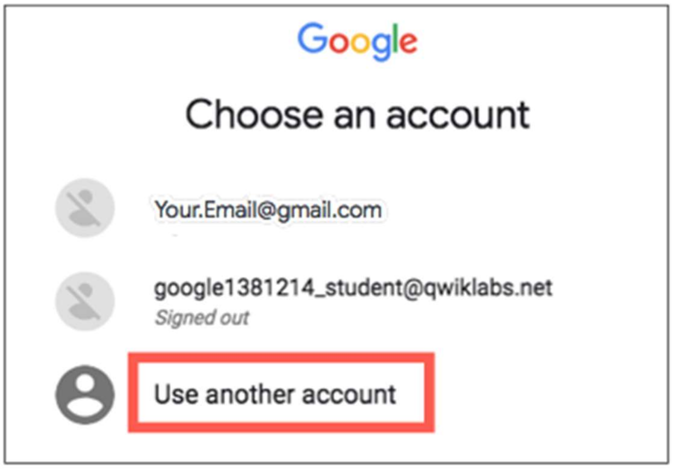
Use your Google Account

Email or phone

Forgot email?

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.

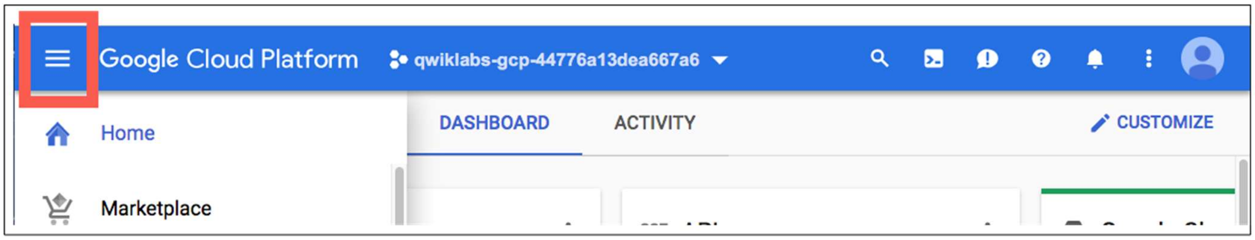
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.



Challenge scenario

You have started a new role as a Junior Cloud Engineer for Jooli, Inc. You are expected to help manage the infrastructure at Jooli. Common tasks include provisioning resources for projects.

You are expected to have the skills and knowledge for these tasks, so step-by-step guides are not provided.

Some Jooli, Inc. standards you should follow:

1. Create all resources in the default region or zone, unless otherwise directed.

2. Naming normally uses the format *team-resource*; for example, an instance could be named **nucleus-webserver1**.

3. Allocate cost-effective resource sizes. Projects are monitored, and excessive resource use will result in the containing project's termination (and possibly yours), so plan carefully. This is the guidance the monitoring team is willing to share: unless directed, use **f1-micro** for small Linux VMs, and use **n1-standard-1** for Windows or other applications, such as Kubernetes nodes.

Your challenge

As soon as you sit down at your desk and open your new laptop, you receive several requests from the Nucleus team. Read through each description, and then create the resources.

Task 1: Create a project jumphost instance

You will use this instance to perform maintenance for the project.

Requirements:

- Name the instance **nucleus-jumphost**.
- Use an *f1-micro* machine type.
- Use the default image type (Debian Linux).

Click *Check my progress* to verify the objective.

Create a project jumphost instance

Check my progress

If a green check mark isn't displayed, click the score fly-out on the upper-right, and then click **Run Step** on the relevant step. A pop-up box will give you advice.

Task 2: Create a Kubernetes service cluster

There is a limit to the resources you are allowed to create in your project. If you don't get the result you expected, delete the cluster before you create another cluster. If you don't, the lab might end and you might be blocked. In order to get your account unblocked, you will have to reach out to Qwiklabs Support.

The team is building an application that will use a service running on Kubernetes. You need to:

- Create a cluster (in the us-east1-b zone) to host the service.
- Use the Docker container hello-app (`gcr.io/google-samples/hello-app:2.0`) as a place holder; the team will replace the container with their own work later.
- Expose the app on port 8080.

Click *Check my progress* to verify the objective.

Create a Kubernetes cluster

Check my progress

If a green check mark isn't displayed, click the score fly-out on the upper-right, and then click **Run Step** on the relevant step. A pop-up box will give you advice.

Task 3: Set up an HTTP load balancer

You will serve the site via nginx web servers, but you want to ensure that the environment is fault-tolerant. Create an HTTP load balancer with a managed instance group of **2 nginx web servers**. Use the following code to configure the web servers; the team will replace this with their own configuration later.

There is a limit to the resources you are allowed to create in your project, so do not create more than 2 instances in your managed instance group. If you do, the lab might end and you might be banned.

```
cat << EOF > startup.sh
#! /bin/bash
apt-get update
apt-get install -y nginx
service nginx start
sed -i -- 's/nginx/Google Cloud Platform - '"\$HOSTNAME"'/' /var/www/html/index.nginx-debian.html
EOF
content_copy
```

You need to:

- Create an instance template.
- Create a target pool.
- Create a managed instance group.
- Create a firewall rule to allow traffic (80/tcp).
- Create a health check.
- Create a backend service, and attach the managed instance group.
- Create a URL map, and target the HTTP proxy to route requests to your URL map.
- Create a forwarding rule.

Click *Check my progress* to verify the objective.

Create the website behind the HTTP load balancer

Check my progress

If a green check mark isn't displayed, click the score fly-out on the upper-right, and then click **Run Step** on the relevant step. A pop-up box will give you advice.

Congratulations!

Google Cloud
Create and Manage Cloud Resources
INFRASTRUCTURE MODERNIZATION SKILL

Earn Your Next Skill Badge

This self-paced lab is part of the Create and Manage Cloud Resources Quest. Completing this skill badge quest earns you the badge above, to recognize your achievement. Share your badge on your resume and social platforms, and announce your accomplishment using #GoogleCloudBadge.
This skill badge quest is part of Google's Associate Cloud Engineer and Professional Cloud Architect learning paths. Continue your learning journey by enrolling in the Perform Foundational Infrastructure Tasks in Google Cloud quest. See other available Qwiklabs Quests available in the catalog.

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 February 12, 2021
Lab Last Tested November 4, 2020

Solution:

Task 1: Create a project jumphost instance

Navigation menu > Compute engine > VM Instance

create **nucleus-jumphost**

series = **N1**

machine type = **f1-micro**

Task 2: Create a Kubernetes service cluster

```
gcloud config set compute/zone us-east1-b
gcloud container clusters create nucleus-jumphost
gcloud container clusters get-credentials nucleus-jumphost
kubectl create deployment hello-app --image=gcr.io/google-samples/hello-app:2.0
kubectl expose deployment hello-app --type=LoadBalancer --port 8080
kubectl get service
```

wait 5 mins to get external IP address

Task 3: Setup an HTTP load balancer

```
cat << EOF > startup.sh
#! /bin/bash
apt-get update
apt-get install -y nginx
service nginx start
sed -i -- 's/nginx/Google Cloud Platform - '"\$HOSTNAME"'/' /var/www/html/index.nginx-debian.html
EOF
```

1 .Create an instance template :

```
gcloud compute instance-templates create nginx-template \
--metadata-from-file startup-script=startup.sh
```

2 .Create a target pool :

```
gcloud compute target-pools create nginx-pool
```

then n

then 20

3 .Create a managed instance group :

```
gcloud compute instance-groups managed create nginx-group \
--base-instance-name nginx \
--size 2 \
--template nginx-template \
--target-pool nginx-pool
```

```
gcloud compute instances list
```

4 .Create a firewall rule to allow traffic (80/tcp) :

```
gcloud compute firewall-rules create www-firewall --allow tcp:80
```

```
gcloud compute forwarding-rules create nginx-lb \
--region us-east1 \
--ports=80 \
--target-pool nginx-pool
```

```
gcloud compute forwarding-rules list
```

5 .Create a health check :

```
gcloud compute http-health-checks create http-basic-check
```

```
gcloud compute instance-groups managed \
set-named-ports nginx-group \
--named-ports http:80
```

6 .Create a backend service and attach the manged instance group :

```
gcloud compute backend-services create nginx-backend \
--protocol HTTP --http-health-checks http-basic-check --global
```

```
gcloud compute backend-services add-backend nginx-backend \
--instance-group nginx-group \
--instance-group-zone us-east1-b \
--global
```

7 .Create a URL map and target HTTP proxy to route requests to your URL map :

```
gcloud compute url-maps create web-map \
--default-service nginx-backend
```

```
gcloud compute target-http-proxies create http-lb-proxy \
--url-map web-map
```

8 .Create a forwarding rule :

```
gcloud compute forwarding-rules create http-content-rule \
--global \
--target-http-proxy http-lb-proxy \
--ports 80
```

```
gcloud compute forwarding-rules list
```

wait 4-5 mins