

# Build and Deploy a Docker Image to a Kubernetes Cluster

**GSP304**



# 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 only recommended for students who have Compute Engine skills. Are you up for the challenge?

## Topics tested

- Build and tag a Docker Image of a sample application
- Push the tagged image to Container Registry
- Create a Kubernetes cluster
- Deploy the application to the Kubernetes cluster

## 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:

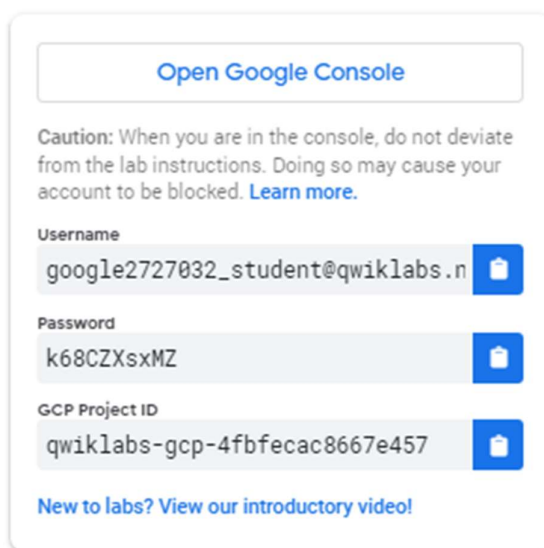
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

**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.

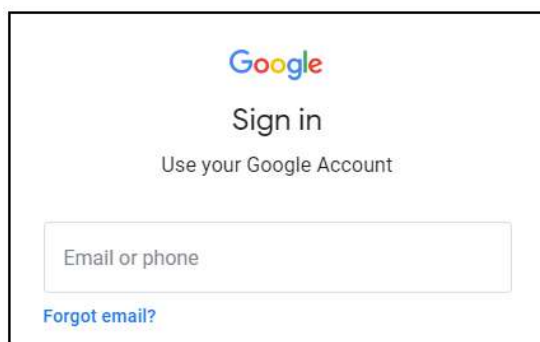
### How to start your lab and sign in to the Google Cloud Console

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.



A screenshot of a web interface for starting a lab. At the top is a button labeled "Open Google Console". Below it is a caution message: "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.](#)". Underneath are three input fields, each with a copy icon to its right. The first field is labeled "Username" and contains "google2727032\_student@qwiklabs.n". The second field is labeled "Password" and contains "k68CZXsxMZ". The third field is labeled "GCP Project ID" and contains "qwiklabs-gcp-4fbfecac8667e457". At the bottom is a link that says "New to labs? View our introductory video!"

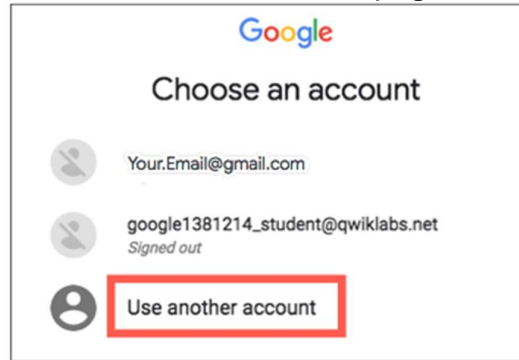
2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



A screenshot of the Google sign-in page. It features the Google logo at the top, followed by the text "Sign in" and "Use your Google Account". Below this is a text input field labeled "Email or phone". At the bottom left is a link that says "Forgot email?"

**Tip:** Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **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.

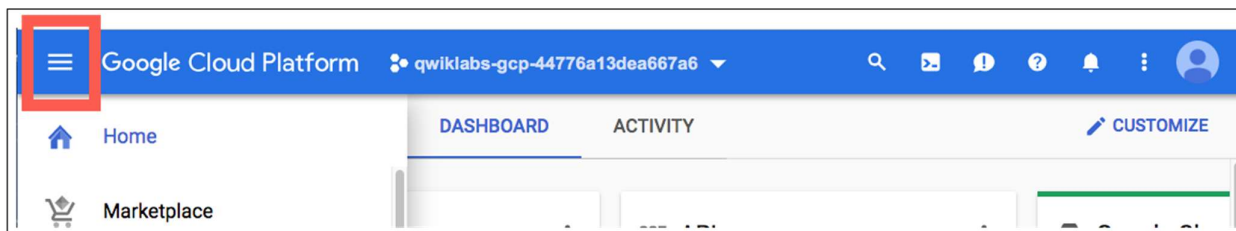
**Important:** You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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

Your development team is interested in adopting a containerized microservices approach to application architecture. You need to test a sample application they have provided for you to make sure that it can be deployed to a Google Kubernetes container. The development group provided a simple Go application called `echo-web` with a Dockerfile and the associated context that allows you to build a Docker image immediately.

## Your challenge

To test the deployment, you need to download the sample application, then build the Docker container image using a tag that allows it to be stored on the Container Registry. Once the image has been built, you'll push it out to the Container Registry before you can deploy it.

With the image prepared you can then create a Kubernetes cluster, then deploy the sample application to the cluster.

**Note:** In order to ensure accurate lab activity tracking you must use `echo-app` as the container repository image name, call your Kubernetes cluster `echo-cluster`, create the Kubernetes cluster in `us-central1-a` zone and use `echo-web` for the deployment name.

## Create a Kubernetes Cluster

Your test environment is limited in capacity, so you should limit the test Kubernetes cluster you are creating to just two `n1-standard-2` instances. You must call your cluster `echo-cluster`.

Click *Check my progress* to verify the objective.

## Build a tagged Docker Image

The sample application, including the Dockerfile and the application context files, are contained in an archive called `echo-web.tar.gz`. The archive has been copied to a Cloud

Storage bucket belonging to your lab project called `gs://[PROJECT_ID]`. You must deploy this with a tag called `v1`.

## Push the image to the Google Container Registry

Your organization has decided that it will always use the `gcr.io` Container Registry hostname for all projects. The sample application is a simple web application that reports some data describing the configuration of the system where the application is running. It is configured to use TCP port 8000 by default.

Click *Check my progress* to verify the objective.

## Deploy the application to the Kubernetes Cluster

Even though the application is configured to respond to HTTP requests on port 8000, you must configure the service to respond to normal web requests on port 80. When configuring the cluster for your sample application, call your deployment `echo-web`.

Click *Check my progress* to verify the objective.

Click *Check my progress* to verify the objective.

## Troubleshooting

**Receiving a 504, Gateway timeout error:** This might just indicate that the application hasn't quite initialized yet, but it could also be caused by a mismatch between the default port that is set in the Dockerfile (TCP port 8000) and the choice of application port you configured when deploying the application image, or when you configured external access.

**Not receiving assessment score for the last three objectives:** This might just indicate that you have created your Kubernetes cluster in the different zone rather than `us-central1-a` zone which is expected in the lab.

# Congratulations!



Google Cloud

## Cloud Architecture: Design, Implement, and Manage

INFRASTRUCTURE MODERNIZATION

### Finish Your Quest

This self-paced lab is part of the Qwiklabs [Cloud Architecture: Design, Implement, and Manage](#) Quest. 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 this Quest](#) and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

### Take Your Next Lab

Continue your Quest with [Scale Out and Update a Containerized Application on a Kubernetes Cluster](#), or check out these suggestions:

- [Migrate a MySQL Database to Google Cloud SQL](#)
- [Deploy a Compute Instance with a Remote Startup Script](#)

## Next Steps / Learn More

Have you checked out the [Data Science on the Google Cloud Platform](#) Quest? Students are given the opportunity to practice all aspects of ingestion, preparation, processing, querying, exploring and visualizing data sets using Google Cloud tools and services. The exercises in the quest are taken from book **Data Science on the Google Cloud Platform** by Valliappa Lakshmanan, published by O'Reilly Media, Inc.

## 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 11, 2021

Copyright 2021 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.

LINK : [https://www.youtube.com/watch?v=pMvMF-YG\\_w8](https://www.youtube.com/watch?v=pMvMF-YG_w8)

```
gsutil cp -r gs://[PROJECT_ID] /echo-web.tar.gz .
tar -xzf echo-web.tar.gz
rm echo-web.tar.gz
cd echo-web/
docker build -t echo-app:v1 .
docker tag echo-app:v1 gcr.io/$DEVSHHELL_PROJECT_ID/echo-app:v1
docker push gcr.io/$DEVSHHELL_PROJECT_ID/echo-app:v1
gcloud config set compute/zone us-central1-a
gcloud container clusters create echo-cluster --num-nodes=2 --machine-type=n1-standard-2
kubectl create deployment echo-web --image=gcr.io/$DEVSHHELL_PROJECT_ID/echo-app:v1
kubectl expose deployment echo-web --type=LoadBalancer --port 80 --target-port 8000
```