Container-Optimized OS: Qwik Start

GSP144

Overview

Container-Optimized OS is an operating system image for your Compute Engine VMs that is optimized for running Docker containers, and is Google's recommended OS for running containers on Google Cloud. In this lab you will create a Container-Optimized instance using the Cloud Console and the CLI.
Since it comes with all container-related dependencies preinstalled, Container-Optimized OS allows your cluster to quickly scale up or down in response to traffic or workload changes, optimizing your spend and improving your reliability.

Container-Optimized OS powers many Google Cloud services such as Kubernetes Engine and Cloud SQL, making it Google's go-to solution for container workloads.

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

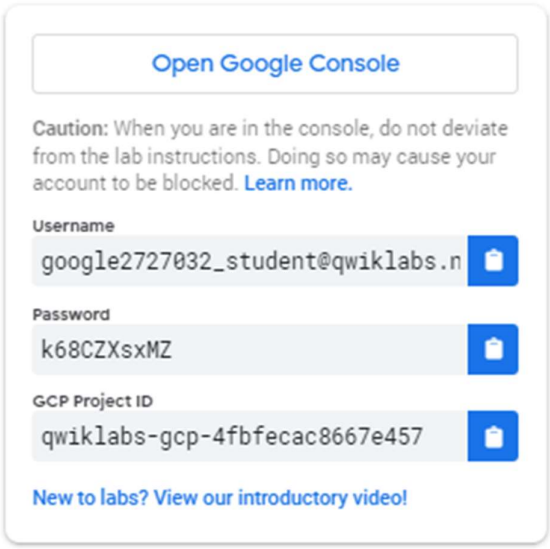
To complete this lab, you need:

- 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.

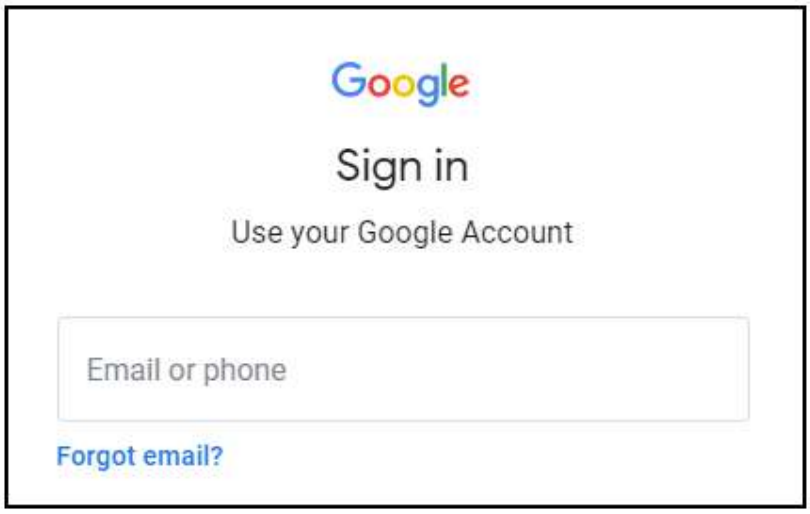
Note: If you are using a Pixelbook, open an Incognito window to run this lab.

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.

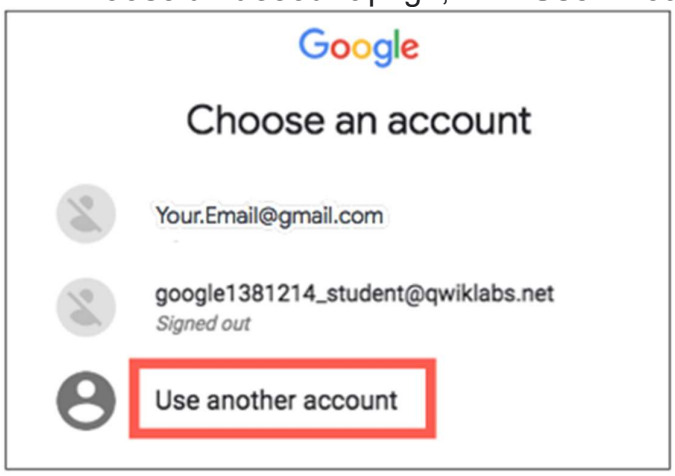


2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



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

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



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

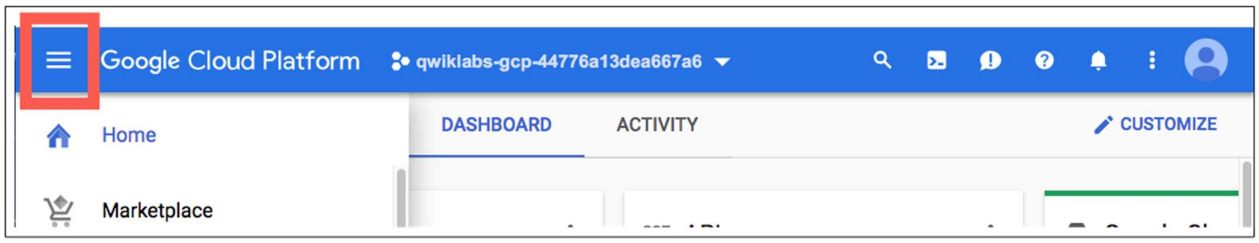
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

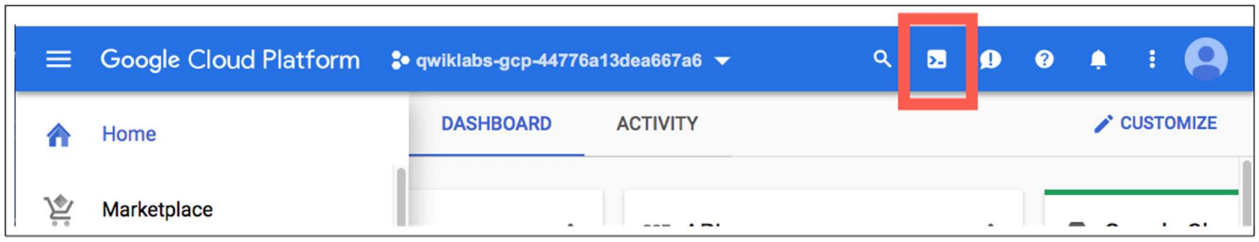
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



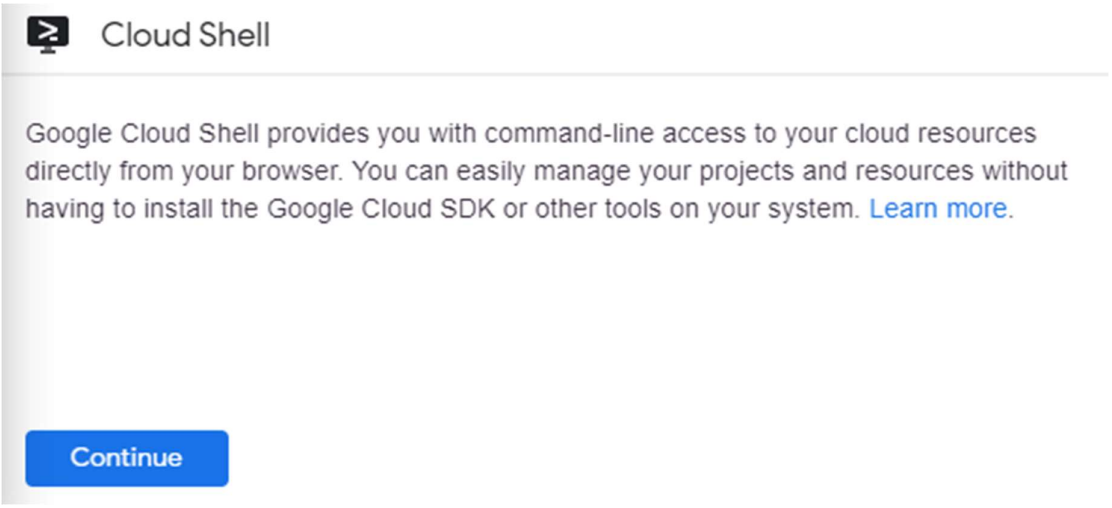
Activate Cloud Shell

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.

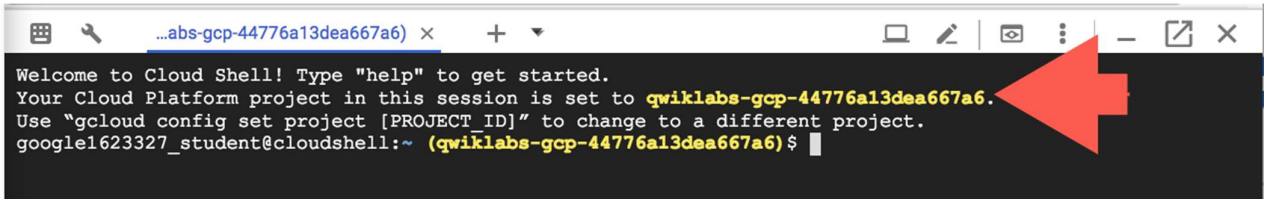
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



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. 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.

Container-Optimized OS benefits

- **Run Containers Out of the Box**: Container-Optimized OS instances come pre-installed with the Docker runtime and cloud-init. With a Container-Optimized OS instance, you can bring up your Docker container at the same time you create your VM, with no on-host setup required.
- **Smaller attack surface**: Container-Optimized OS has a smaller footprint, reducing your instance's potential attack surface.
- **Locked-down by default**: Container-Optimized OS instances include a locked-down firewall and other security settings by default.
- **Automatic Updates**: Container-Optimized OS instances are configured to automatically download weekly updates in the background; only a reboot is necessary to use the latest updates.

Use cases for Container-Optimized OS

Container-Optimized OS can be used to run most Docker containers. You should consider using Container-Optimized OS as the operating system for your Compute Engine instance if you have the following needs:

- You need support for Docker containers or Kubernetes with minimal setup.

- You need an operating system that has a small footprint and is security hardened for containers.
- You need an operating system that is tested and verified for running Kubernetes on your Compute Engine instances.

Container-Optimized OS features

Compute Engine provides several public VM images that you can use to create instances and run your container workloads. Some of these public VM images have a minimalistic container-optimized operating system that includes newer versions of Docker, rkt, or Kubernetes preinstalled. The following public image families are designed specifically to run containers:

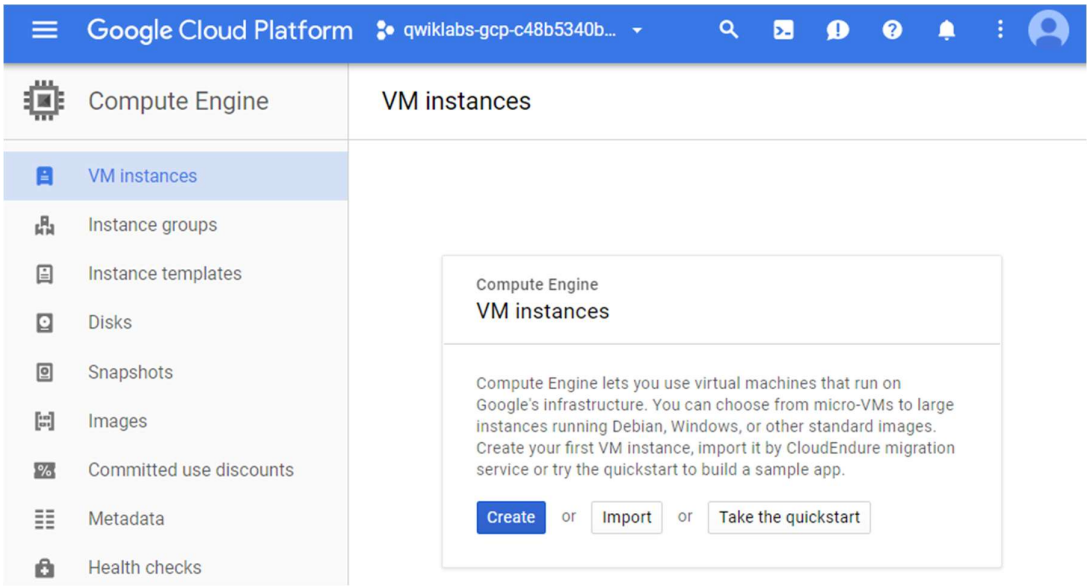
- Container-Optimized OS from Google
- Includes: Docker, Kubernetes
- Image project: cos-cloud
- Image family: cos-stable
- CoreOS
- Includes: Docker, rkt, Kubernetes
- Image project: coreos-cloud
- Image family: coreos-stable
- Ubuntu
- Includes: LXD
- Image project: ubuntu-os-cloud
- Image family: ubuntu-1604-lts
- Windows
- Includes: Docker
- Image project: windows-cloud
- Image family: windows-1709-core-for-containers

In a production environment, if you need to run specific container tools and technologies on images that do not include them by default, install those technologies manually.

Create an instance using the console

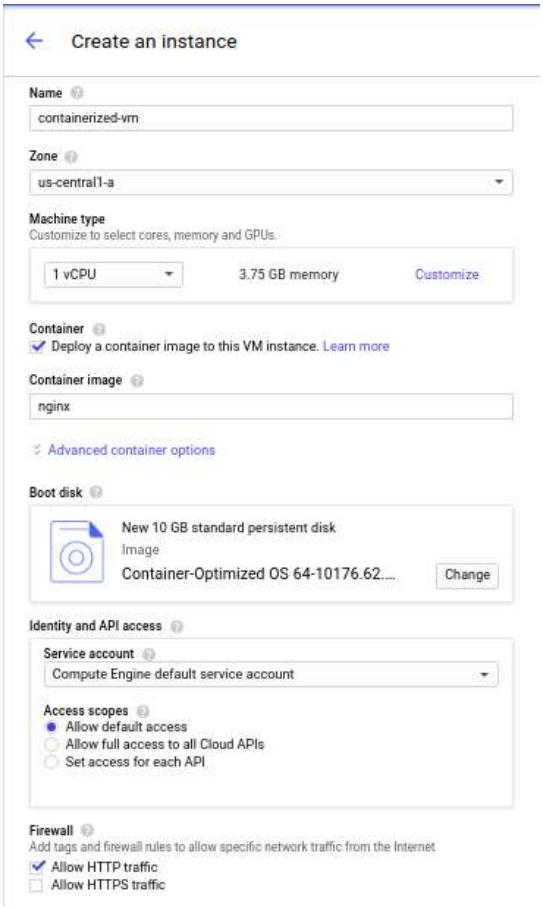
To run a Compute Engine instance with the Container-Optimized OS and a Docker container of your choice:

Click on **Compute Engine** > **VM instances**, then click on **Create**.



There are many parameters you can configure when creating a new instance. For this lab use the following:

Name	containerized-vm
Zone	us-central1-a
Machine Type	1 vCPU This is a (n1-standard-1), 3.75GB RAM instance
Container	Check the box labeled Deploy a container image to this instance
Container Image	nginx
Boot disk	Container-Optimized OS (default value)
Firewall	Check Allow HTTP traffic



Click **Create**.

Your Container-Optimized OS instance with the Docker container is up and running.

Test Completed Task

Click **Check my progress** to verify your performed task.

Verify your Docker environment

Click **SSH** on the `containerized-vm` line to SSH into the `containerized-vm` instance.

List all available Docker containers:

```
sudo docker ps
```

Example Output:

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
c62d762ed91b        nginx               "nginx -g 'daemon ..."   15 minutes ago      Up
15 minutes                          containerized-vm
```

On the VM instance console, click on the External IP for `containerized-vm` instance, which will open a new tab. You should see this default Nginx web page:

You can also see the web page by adding the External IP to `http://EXTERNAL_IP` in a new browser window or tab.

Create an instance using CLI

Now you use the Cloud Shell command line to create a Container-Optimized OS instance.

In Cloud Shell, enter the following command to see what Container-Optimized OS images are available on Google Cloud to create an instance.

```
gcloud compute images list \
    --project cos-cloud \
    --no-standard-images
```

Use the gcloud compute instances create command with --image and --image-project flags to create a cos node image instance:

```
gcloud beta compute instances create-with-container containerized-vm2 \
    --image cos-stable-72-11316-136-0 \
    --image-project cos-cloud \
    --container-image nginx \
    --container-restart-policy always \
    --zone us-central1-a \
    --machine-type n1-standard-1
```

In the above example, `cos-stable-72-11316-136-0` is one of the available cos releases. Please use the latest available image from `cos-stable` family and replace it with an image

for your VM instance. It is recommended to use `--preemptible` flag for one-off experimental instances.

Test Completed Task

Click **Check my progress** to verify your performed task.

In the Cloud Shell enter the following commands to create firewall rules to allow HTTP traffic from the internet and to enable all internal traffic within the VPC:

```
gcloud compute firewall-rules create allow-containerized-internal\
  --allow tcp:80 \
  --source-ranges 0.0.0.0/0 \
  --network default
```

Test Completed Task

Click **Check my progress** to verify your performed task.

For this step, you need the external IP address of your containerized-vm2 instance. You can look up the address in the **VM Instances** page in the Cloud Platform Console.

In a browser, enter your external IP address to verify that nginx is running:

```
http://[YOUR_EXTERNAL_IP_ADDRESS]
```

You should see **Welcome to nginx!**

Congratulations!

Finish Your Quest

Continue your Quest with Baseline: Deploy & Develop. 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.

Next Steps / Learn More

This lab is also 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 November 15, 2019
Lab Last Tested November 15, 2019