googlecoursera.qwiklabs.com

Launching a Virtual Machine on Google Cloud Using Puppet | Qwiklabs

Qwiklabs
8-10 minutes

Introduction

You're an IT Administrator for your company and you're assigned to work on a project that requires you to deploy eight virtual machines (VMs) as web servers. Each of them should have the same configuration. You'll create a VM, set up an auto-enabled service, and make it a template. Then you'll use the template to create seven more VMs.

What you'll do

- Create a VM using GCP web UI and make a template out of it
- Use a command-line interface to interact with VMs
- Learn how to configure an auto-enabled service
- Learn to use gcloud to deploy VMs with a template

Setup

What you need

To complete this lab, you need:

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

Note: If you already have your own personal GCP account or project, please don't use it for this lab.

You should install the Google Cloud SDK, initialize it, and run core gcloud commands from the command line on your local computer. To do this, follow the instructions for your operating system below:

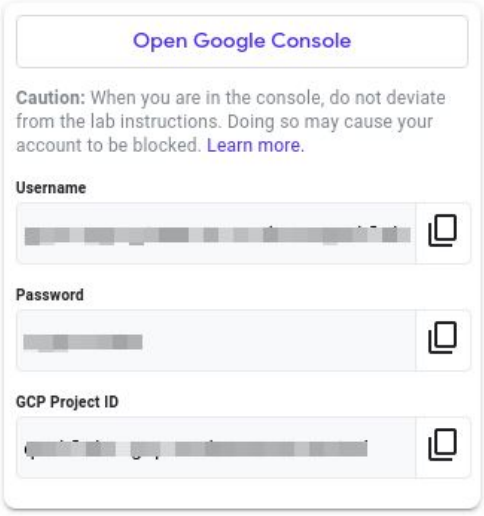
- Linux
- Debian and Ubuntu
- Red Hat and Cento
- macOS

- Windows

You'll have 90 minutes to complete this lab.

Start your lab by signing in to the Console

1. Click the **Start Lab** button. On the left is a panel populated with the temporary credentials that you'll need to use for this lab.

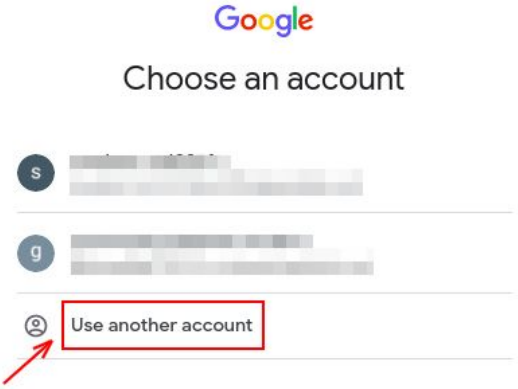


2. Copy the username, then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Choose an account** page.

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

Note: Using a new Incognito window (Chrome) or another browser for the Qwiklabs session is recommended. Alternatively, you can log out of all other Google / Gmail accounts before beginning the labs.

3. On the **Choose an account** page, click **Use another account**.



4. The **Sign in** page opens. 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. Please do **not** use your Qwiklabs credentials. If you have your own GCP account, do **not** use it for this lab in order to avoid incurring charges.

5. Click through the subsequent pages:

6. Accept the terms and conditions.

7. Do **not** add recovery options or two-factor authentication, since this is a temporary account.

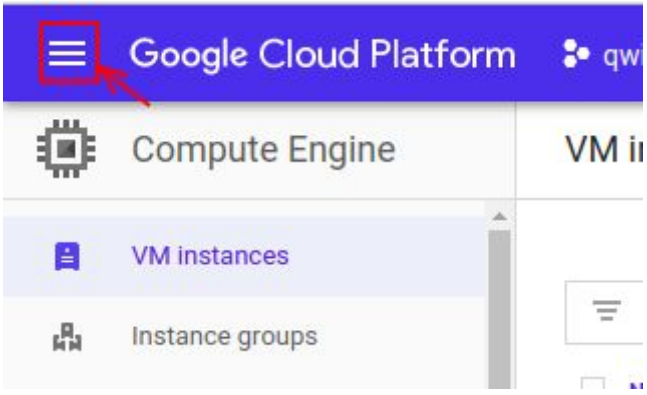
8. Do **not** sign up for free trials.

After a few moments, the GCP console opens in this tab.

Create a VM instance from the Cloud Console

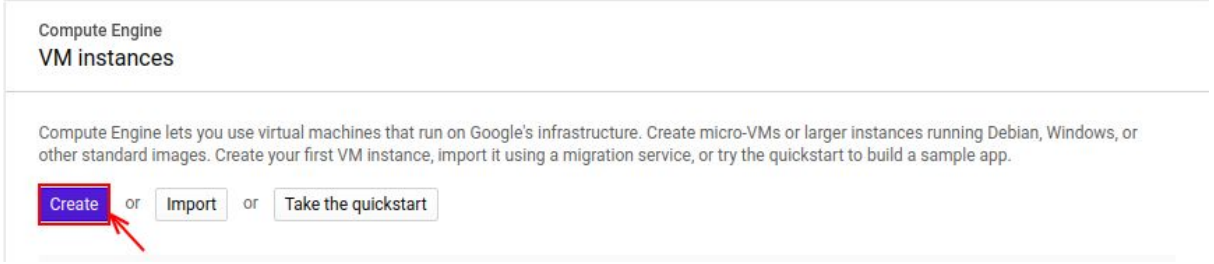
In this section, you'll learn how to create new, predefined machine types with Google Compute Engine from the Cloud Console.

In the GCP Console, on the top left of the screen, select **Navigation menu** > **Compute Engine** > **VM instances**:



This may take a moment to initialize for the first time.

To create a new instance, click **Create**.

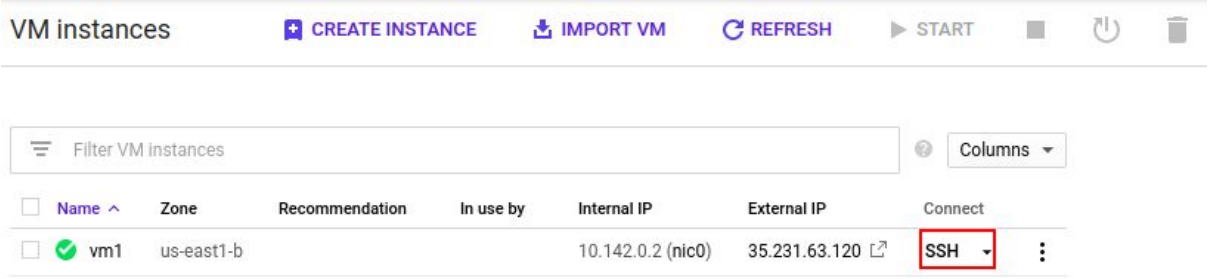


There are lots of parameters you can configure when creating a new instance. Use the following for this lab:

Field	Value	Additional Information
Name	vm1	Name for the VM instance
Region	us-east1	Learn more about regions in Regions & Zones documentation.
Zone	us-east1-b	Learn more about regions in Regions & Zones documentation.
Machine Type	n1-standard-1	**Note**: A new project has a default resource quota, which may limit the number of CPU cores. You can request more when you work on projects outside of this lab.
Boot Disk	Ubuntu 18.04 LTS	Click on the **change** button, click on the **OS images** section then select Ubuntu 18.04 LTS. Learn more about boot disk check out this link.
Boot disk type	standard persistent disk	Learn more about standard persistent disk check out this link.
Firewall	allow HTTP and HTTPS traffic	Learn more about firewall check out this link.

Leave all the other configurations set to their defaults.

After entering the above parameters, click on the **Create** button to create your VM.



SSH into vm1 by clicking on the SSH button, as shown in the image above.

Git clone

Use Git to clone the repository by using the following command:

```
git clone https://www.github.com/google/it-cert-automation-practice.git
```

Output:

```
student-03-ceebea506efb@vm1:~$ git clone https://www.github.com/google/it-cert-automation-practice.git
Cloning into 'it-cert-automation-practice'...
Username for 'https://github.com': [redacted]
Password for 'https://[redacted]@github.com':
warning: redirecting to https://github.com/google/it-cert-automation-practice.git/
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 25 (delta 4), reused 24 (delta 3), pack-reused 0
Unpacking objects: 100% (25/25), done.
```

File operation

Once you have the repository successfully cloned, navigate to the Lab3/directory.

```
cd ~/it-cert-automation-practice/Course5/Lab3
```

To list the files in the working directory Lab3/ use the **list** command.

```
ls
```

Output:

```
student-03-ebdba2b21bba@vm1:~/it-cert-automation-practice/Course5/Lab3$ ls
hello_cloud.py  hello_cloud.service
```

In order to enable hello_cloud.py to run on boot, copy the file hello_cloud.py to the /usr/local/bin/ location.

```
sudo cp hello_cloud.py /usr/local/bin/
```

Also copy hello_cloud.service to the /etc/systemd/system/ location.

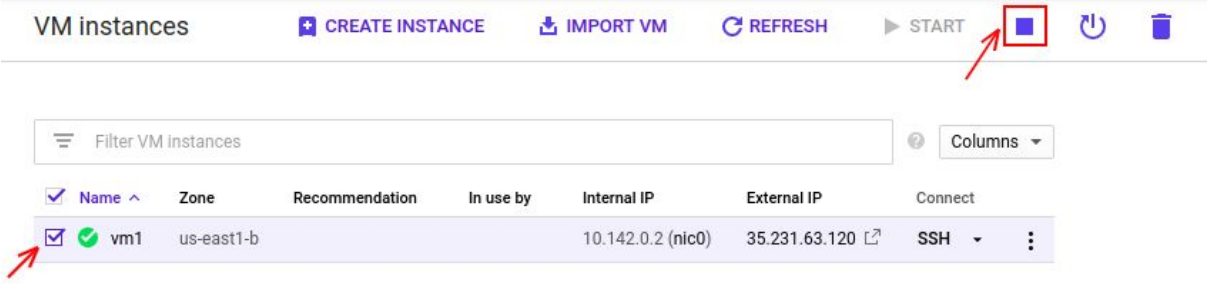
```
sudo cp hello_cloud.service /etc/systemd/system
```

Now, use the systemctl command to enable the service **hello_cloud**.

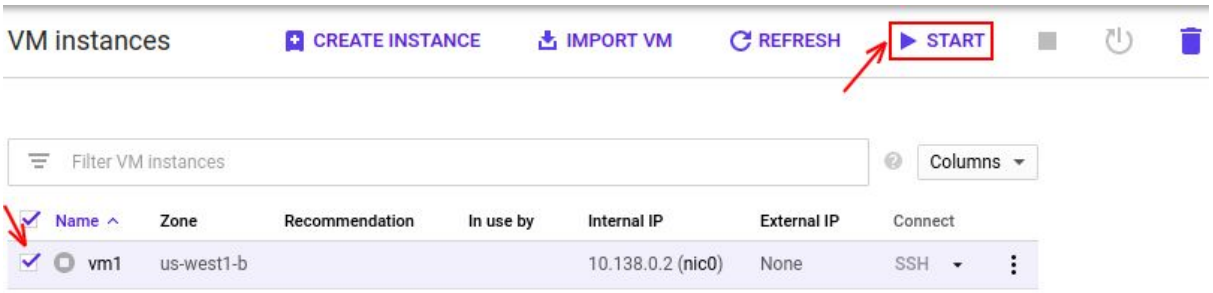
```
sudo systemctl enable hello_cloud.service
```

Restart the VM

After enabling the **hello_cloud** service, reboot the VM to ensure that the service is up. To reboot the VM instance **vm1** go to the **Compute Engine** > **VM instance** and stop the VM instance vm1 by selecting the VM instance vm1 and clicking on the **stop** button at the top.



The start method restarts an instance in a TERMINATED state. To start the VM instance **vm1**, select it first by tick marking it, then click on the **start** button at the top. You can this in the image below.



After restarting the VM instance **vm1,** visit the External IP link of the vm1 that's shown in the image below:

Output:

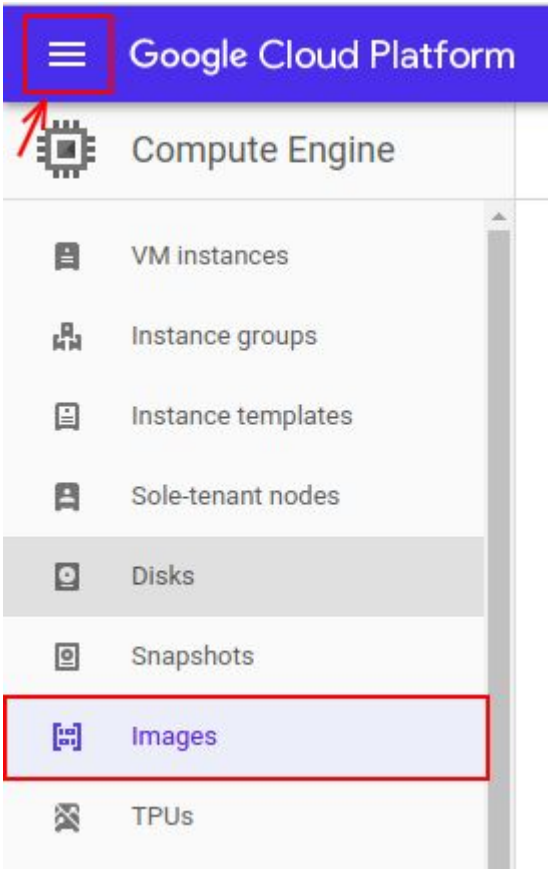
Create VMs using a template

You'll now create a template for vm1.

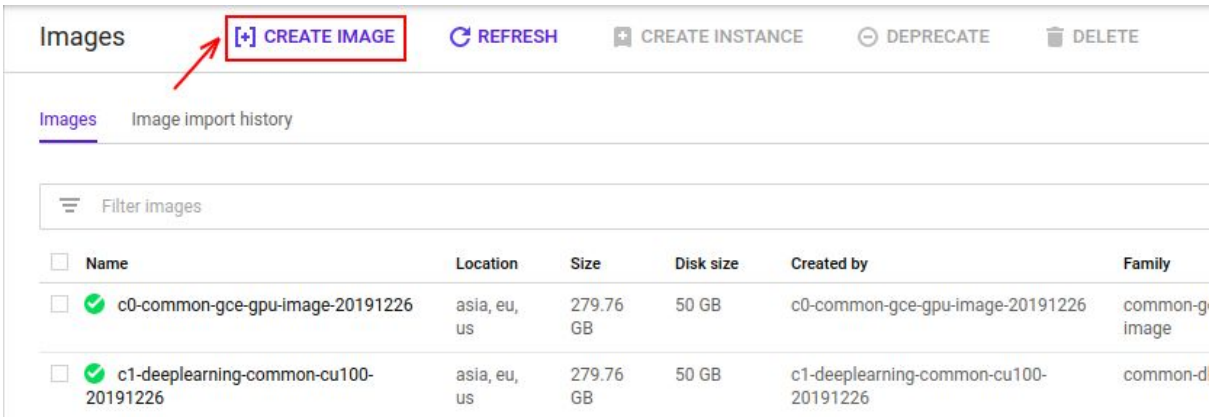
First, **shut down** the VM instance **vm1** by going to the **Compute Engine** > **VM instance**, selecting the VM instance vm1, and clicking on the **stop** button at the top.

Now, create an image named vm-image based on the **vm1** disk by following the steps below:

In the GCP Console, on the top left of the screen, select **Navigation menu > Compute Engine > Images:**



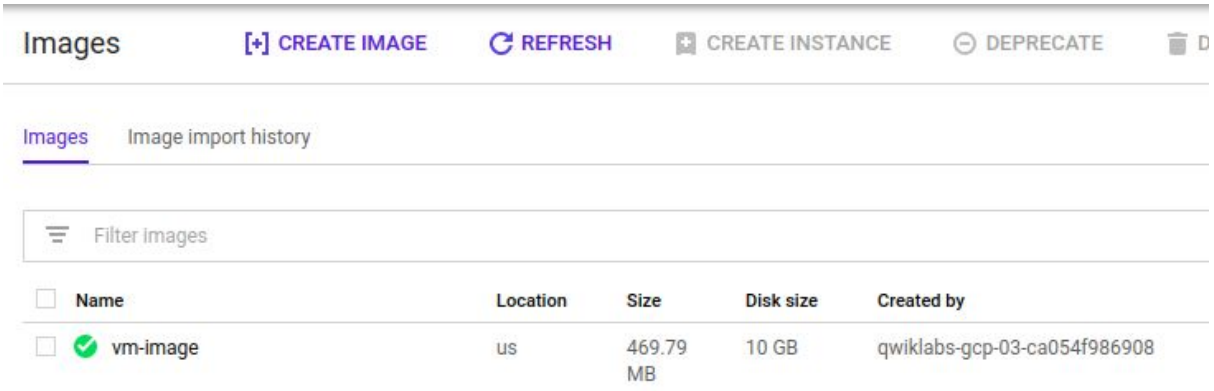
Click on the **CREATE IMAGE** button below.



Then, create an image based on the **vm1's** disk, using the following parameters:

Field	**Value**
Name	vm-image
Source	Disk
Source Disk	vm1

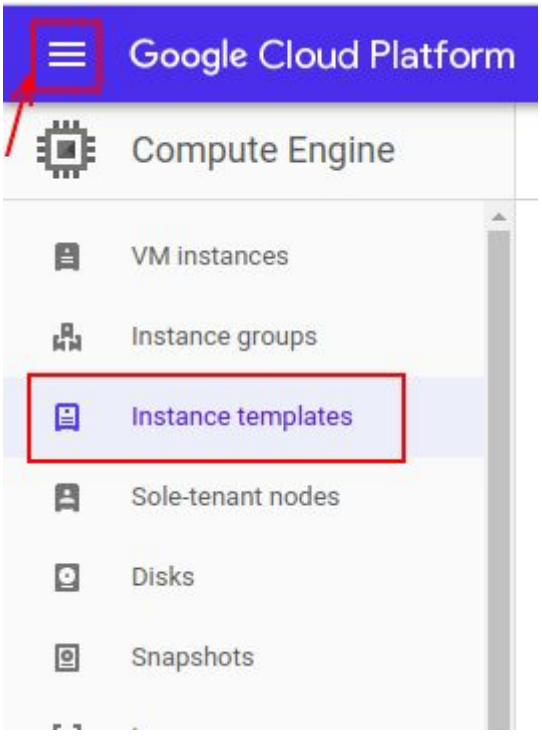
Leave all of the other values set to their default settings. Click on the **create** button to create your image.



Now, create an instance template using vm-image for the boot disk you just created.

To create a instance template, follow the instructions below:

In the GCP Console, on the top left of the screen, select **Navigation menu > Compute Engine > Instance templates:**

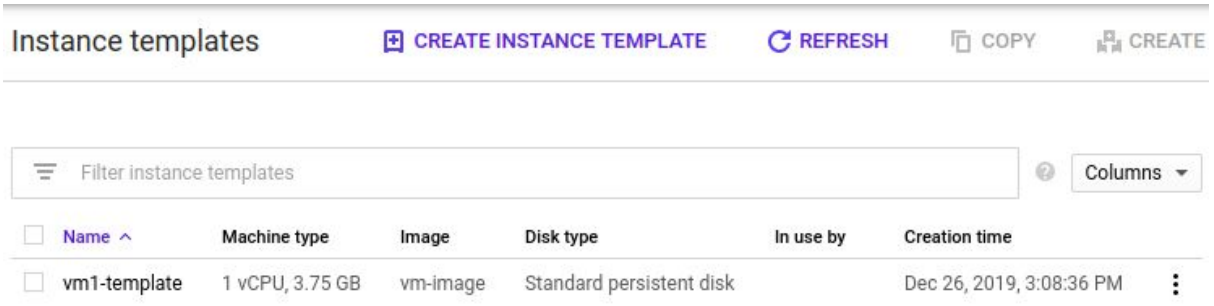


Now, click on **Create instance template** to create a new template.

There are lots of parameters that you can configure when creating a new instance. Use the following for this lab:

Field	Value	Additional information
Name	vm1-template	Name for the VM instance template
Machine Type	n1-standard-1	**Note**: A new project has a default resource quota, which may limit the number of CPU cores. You can request more when you work on projects outside of this lab.
Boot Disk	vm-image	Click on the **change** button, click on the **custom images** section. Now, select **vm-image** by selecting the project you are working on.
Boot disk type	standard persistent disk	Learn more about standard persistent disk check out this link.
Firewall	allow HTTP and HTTPS traffic	Learn more about firewall check out this link.

Leave the rest of the values set to their default settings. Click on the **create** button to create the instance template vm1-template.



Now, you'll create new VM instances using vm-image as a template on your local computer. On your local computer, type the following command:

```
gcloud compute instances create --zone us-west1-b
--source-instance-template vm1-template vm2 vm3 vm4 vm5 vm6 vm7
vm8
```

Wait for the command to finish. Once it's done, you can view the instances through the Console or by using the following gcloud command on your local terminal:

```
gcloud compute instances list
```

Now, open the external links for vm2 and vm8 to check if all the configuration set up properly as vm1.

Output:


```
Hello Cloud

Hostname: vm2
IP Address: 10.138.0.5
```


```
Hello Cloud

Hostname: vm8
IP Address: 10.138.0.6
```