Creating a Persistent Disk

Overview

Google Compute Engine lets you create and run virtual machines on Google infrastructure. You can create virtual machines running different operating systems, including multiple flavors of Linux (Debian, Ubuntu, Suse, Red Hat, CoreOS) and Windows Server!

Google Compute Engine provides persistent disks for use as the primary storage for your virtual machine instances. Like physical hard drives, persistent disks exist independently of the rest of your machine – if a virtual machine instance is deleted, the attached persistent disk continues to retain its data and can be attached to another instance.

There are 2 types of persistent disks:

- Standard persistent disk
- SSD Persistent disk

Learn more about the differences in Storage Options. Each type of persistent disks will have different capacity limits. Read more in the Persistent Disk documentation

In this hands-on lab you'll learn how to a create persistent disk and attach it to a virtual machine.

What you'll do

- Create a new VM instance and attach a persistent disk

- Format and mount a persistent disk

Prerequisites

- Familiarity with standard Linux text editors such as `vim`, `emacs` or `nano` will be helpful

Setup

What you need

To complete this lab, you need:

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

How to start your lab and sign in to the Console

- Open https://console.cloud.google.com/
- Enter login credentials

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

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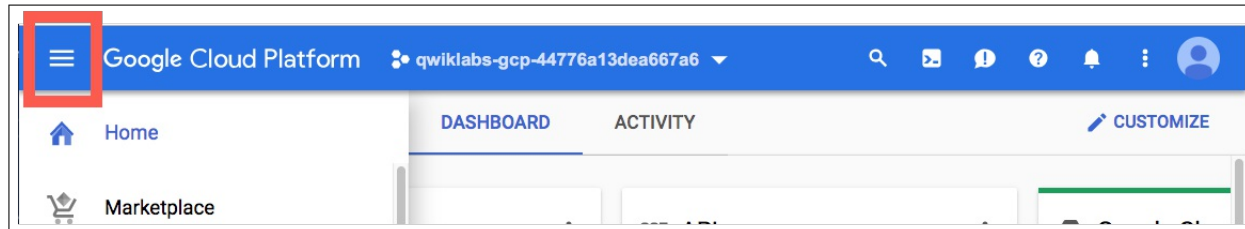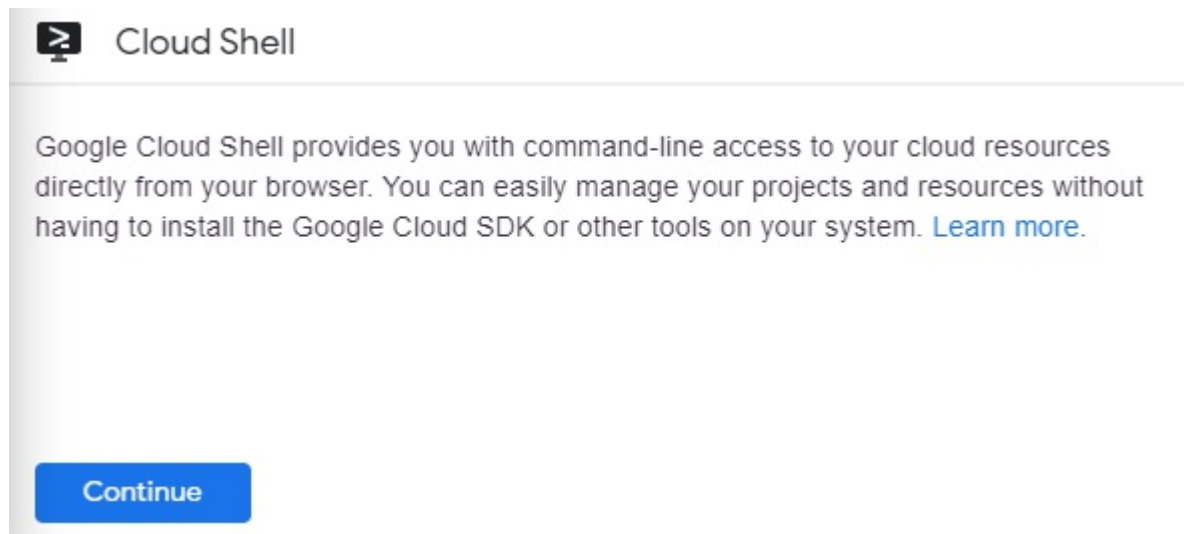

Activate Google Cloud Shell

Google 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. Google Cloud Shell provides command-line access to your GCP resources.

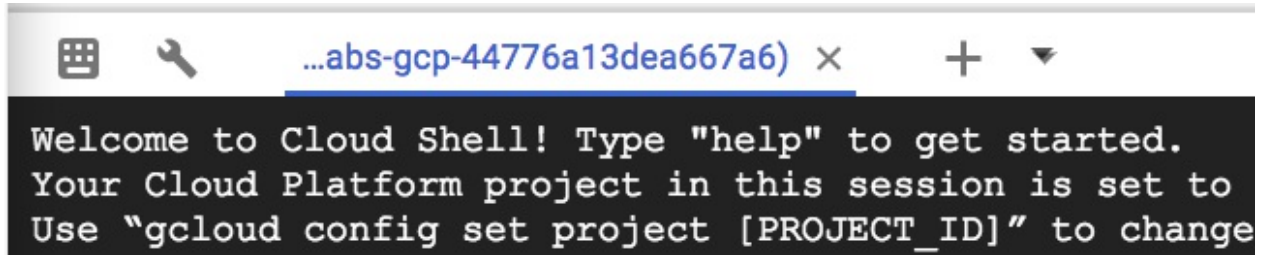1. In GCP console, on the top right toolbar, click the Open Cloud Shell button.
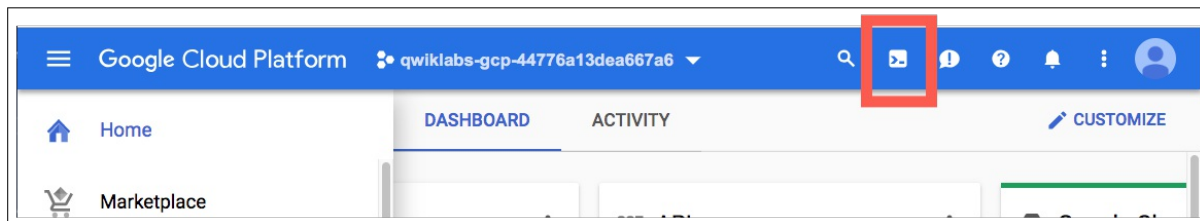
2. Click **Continue**.

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 Platform. 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@testlabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

Output:

```
[core]
project = <project_ID>
```

Example output:

```
[core]
project = testlabs-gcp-44776a13dea667a6
```

Full documentation of **gcloud** is available on Google Cloud gcloud Overview .

Create a new instance

First, let's create a Compute Engine virtual machine instance that has only a boot disk.

You can learn more creating a virtual machine instance in a different lab, or refer to the Google Compute Engine documentation.

In Cloud Shell command line, use the `gcloud` command to create a new virtual machine instance named `gcelab`:

```
gcloud compute instances create gcelab --zone us-central1-c
```

(Output)

```
Created [...].
NAME      ZONE           MACHINE_TYPE  PREEMPTIBLE INTERNAL_IP EXTERNAL_IP    STATUS
gcelab us-central1-c n1-standard-1             10.240.X.X  X.X.X.X        RUNNING
```

The newly created virtual machine instance will have a default 10 GB persistent disk as the boot disk.

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

Create a new instance in the specified zone.

Create a new persistent disk

Because we want to attach this disk to the virtual machine instance we created in the previous step, the zone must be the same.

Still in the Cloud Shell command line, use the following command to create a new disk named `mydisk`:

```
gcloud compute disks create mydisk --size=200GB \
--zone us-central1-c
```

(Output)

```
NAME    ZONE           SIZE_GB TYPE          STATUS
mydisk us-central1-c 200       pd-standard READY
```

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

Create a new persistent disk in the specified zone

# Attaching a disk

Attaching the persistent disk

You can attach a disk to a running virtual machine. Let's attach the new disk (`mydisk`) to the virtual machine instance you just created (`gcelab`).

Use the following command to attach the disk:

```
gcloud compute instances attach-disk gcelab --disk mydisk --zone us-central1-c
```

(Output)

```
Updated [https://www.googleapis.com/compute/v1/projects/testlabs-gcp-d12e3215bb368ac5/zones/us-central1-c/instances/gcelab
```
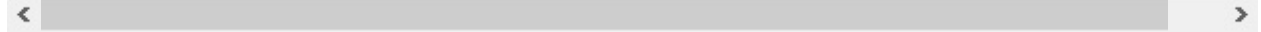
That's it!

Finding the persistent disk in the virtual machine

The persistent disk is now available as a block device in the virtual
machine instance. Let's take a look.

1. SSH into the virtual machine:

```
gcloud compute ssh gcelab --zone us-central1-c
```

(Output)

```
WARNING: The public SSH key file for gcloud does not exist.
WARNING: The private SSH key file for gcloud does not exist.
WARNING: You do not have an SSH key for gcloud.
WARNING: SSH keygen will be executed to generate a key.
This tool needs to create the directory
[/home/gcpstaging8246_student/.ssh] before being able to generate SSH
keys.
Do you want to continue (Y/n)?  y
```

2. At the prompt, enter `y` to continue.
3. When prompted for an RSA key pair passphrase, press __enter
   __for no passphrase, and then press __enter __again to confirm
   no passphrase.

(Output)

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gcpstaging8246_student/.ssh/google_compute_en
gine.
Your public key has been saved in /home/gcpstaging8246_student/.ssh/google_compute_engine
.pub.
The key fingerprint is:
6c:04:bf:29:95:0d:93:bc:fe:00:2c:85:86:f8:7a:53 gcpstaging8246_student@cs-6000-devshell-v
m-dbb9559d-4412-4801-ad8c-bdaf885541a9
The key's randomart image is:
+---[RSA 2048]----+
| . . ...o.       |
|. . o .oo=       |
| . . o  =..      |
|  . E o+.o       |
| . . ..oS        |
|. o   oo          |
| . .    o         |
|          .      |
```

```
    |                 |
    +-----------------+
Updating project ssh metadata...\Updated [https://www.googleapis.com/compute/v1/projects/
testlabs-gcp-d12e3215bb368ac5].
Updating project ssh metadata...done.
Waiting for SSH key to propagate.
Warning: Permanently added 'compute.7714273689800906026' (ECDSA) to the list of known hosts.
Linux gcelab 4.9.0-4-amd64 #1 SMP Debian 4.9.51-1 (2017-09-28) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

4. Now find the disk device by listing the disk devices in
   `/dev/disk/by-id/.`

```
ls -l /dev/disk/by-id/
```

(Output)

```
lrwxrwxrwx 1 root root  9 Feb 27 02:24 google-persistent-disk-0 -> ../../sda
lrwxrwxrwx 1 root root 10 Feb 27 02:24 google-persistent-disk-0-part1 -> ../../sda1
lrwxrwxrwx 1 root root  9 Feb 27 02:25 google-persistent-disk-1 -> ../../sdb
lrwxrwxrwx 1 root root  9 Feb 27 02:24 scsi-0Google_PersistentDisk_persistent-disk-0 -> ../../sda
lrwxrwxrwx 1 root root 10 Feb 27 02:24 scsi-0Google_PersistentDisk_persistent-disk-0-part1 -> ../../sda1
lrwxrwxrwx 1 root root  9 Feb 27 02:25 scsi-0Google_PersistentDisk_persistent-disk-1 -> ../../sdb
```

You found the file, the default name is:

`scsi-0Google_PersistentDisk_persistent-disk-1.`

If you want a different device name, when you attach the disk, you would
specify the `device-name` parameter. For example, to specify a device
name, when you attach the disk you would use the command:

```
gcloud compute instances attach-disk gcelab --disk mydisk --device-name <YOUR_DEVICE_NAME> --zone us-central1-c
```

Formatting and mounting the persistent disk

Once you find the block device, you can partition the disk, format it,
and then mount it using the following Linux utilities:

- `mkfs:` creates a filesystem
- `mount` : attaches to a filesystem

Make a mount point:

```
sudo mkdir /mnt/mydisk
```

Next, format the disk with a single `ext4` filesystem using the
mkfs
tool. This command deletes all data from the specified disk:

```
sudo mkfs.ext4 -F -E lazy_itable_init=0,lazy_journal_init=0,discard /dev/disk/by-id/scsi-0Google_PersistentDisk_persistent
```

Last lines of the output.

```
Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

Now use the mount
tool to mount the disk to the instance with the `discard` option enabled:

```
sudo mount -o discard,defaults /dev/disk/by-id/scsi-0Google_PersistentDisk_persistent-disk-1 /mnt/mydisk
```

That's it!

Automatically mount the disk on restart

By default the disk will not be remounted if your virtual machine
restarts. To make sure the disk is remounted on restart, you need to add
an entry into `/etc/fstab` .

Open `/etc/fstab` in nano to edit.

```
sudo nano /etc/fstab
```

Add the following below the line that starts with "UUID=…"

```
/dev/disk/by-id/scsi-0Google_PersistentDisk_persistent-disk-1 /mnt/mydisk ext4 defaults 1 1
```

`/etc/fstab` content should look like this:

```
UUID=e084c728-36b5-4806-bb9f-1dfb6a34b396 / ext4 defaults 1 1
/dev/disk/by-id/scsi-0Google_PersistentDisk_persistent-disk-1 /mnt/mydisk ext4 defaults 1 1
```

Save and exit nano by pressing `Ctrl+o` , `Enter` , `Ctrl+x` , in that
order.

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

Attaching and Mounting the persistent disk.

Test your knowledge

Test your knowledge about Google cloud Platform by taking our quiz.

For migrating data from a persistent disk to another region, reorder the
following steps in which they should be performed:

1. Attach disk
2. Create disk
3. Create snapshot
4. Create instance
5. Unmount file system(s)

Local SSDs

Google Compute Engine can also attach local SSDs. Local SSDs are physically attached to the server hosting the virtual machine instance to which they are mounted. This tight coupling offers superior performance, with very high input/output operations per second (IOPS) and very low latency compared to persistent disks.

Local SSD performance offers:

- Less than 1 ms of latency
- Up to 680,000 read IOPs and 360,000 write IOPs

These performance gains require certain trade-offs in availability, durability, and flexibility. Because of these trade-offs, local SSD storage is not automatically replicated and all data can be lost in the event of a host error or a user configuration error that makes the disk unreachable. Users must take extra precautions to backup their data.

This lab does not cover local SSDs. To maximize the local SSD performance, you'll need to use a special Linux image that supports NVMe. You can learn more about local SSDs in the Local SSD documentation.

Congratulations!

You've learned how to create, find, and attach persistent disks to a virtual machine instance and the key difference between persistent disks and local SSDs. You can use persistent disks to setup and configure your database servers.

Next Steps / Learn More

- Persistent Disk Documentation
- `gcloud` Documentation and tutorial video.