# Compute Instance in Google's Cloud Platform

# Demo





Instructor: In this demonstration we're going to set up a Compute instance in Google's Cloud platform. Let's get started. Here we see a quick introduction, including a Starter Checklist of different things that we can do. For now, we're going to look at this Compute section and get started by creating a new virtual machine instance.

In addition to the name, we must specify the region and zone this will be deployed in. Then we can look at our machine configuration, specifying the CPU size, memory allocation, et cetera. There's a cost associated with higher-capacity configurations because you are reserving more of the compute power for your instance.

There are additional configuration options, including the ability to deploy a container image to this VM instance.

Exploring the boot disk options, we will select a publicly available Debian image to get started. There are settings to select the underlying hardware that will be used, along with the size of our boot disk. If we did have our own image, virtual machine snapshots or existing disks, we could indeed choose to use those instead of our current configuration.

The Identity and API Access section allows us to select the default service account that will get enabled within our instance and allows us to programmatically access the environment. There are a number of other potential API settings and permissions that we could assign within the instance, but for now we're just going to accept the default permissions and default access.

We can take a look at what this would look like in code if we wanted to programmatically script a similar setup, and we see here the gcloud command that we would use if we wanted to deploy an instance using Cloud Shell. Once the instance is set up, we can manage additional access and assign users and/or access policies to find who can work with our new instance. Once we drill into the instance, we can see our remote connection options. We'll use the embedded SSH client within the GCP

console to access our newly created LINUX machine.

First thing we want to do is update our root password, because we are, in fact, using a base image created by somebody else. We'll take a quick look at etc/password to see other accounts that come with our VM and make a note to come back for a more detailed review.

If we take a look at the firewall rules you can see the default rules that were set up allow management protocols, RDP and SSH inbound to our instance, as well as internet control message protocol, which is used to verify things are running using commands like ping. There's also another rule in here that allows traffic flow within our VPC on a private 10.128 subnet. You'll need to tune these rules to provide only the needed access inbound, outbound and within your VPC.

Let's set up a couple of connectivity tests to verify things are working as expected. The first one we'll evaluate if TCP Port 80 is accessible on our internet IP. It should not be, because we do not have an inbound firewall rule allowing this, and of course, if our instance is not listening on that port it will also fail. Next, we will configure a check on TCP Port 22 and SSH. This check should work, and it's a good test to ensure that the virtual machine is up and accessible from a specific IP address. Given our current ACL, this is anywhere on the internet.

We can indeed change that firewall rule and restrict the source IPs that can access our environment. Can enable logging for packets that match those rules or set additional options.

We're going to jump over to our OS patch management configuration for this instance. As with all computers, we need to ensure it is up-to-date with the latest operating system patches and software versions. Let's see how we can leverage GCP to help with this. We'll create a new patch deployment, and we must first define the zone which we'll be targeting. We can specify which instances we will attempt to apply patches to using a prefix filter or a label, if we have one defined. We must give our deployment a name and specify additional patch parameters based on operating systems within those zones, and the default tools for applying those patches, like yum or apt. Keep in mind, if you specify a particular patch configuration and your instance doesn't match that criteria, it will fail.

Also note, Google Cloud Platform is leveraging those API service accounts to reach into your environment and execute these commands, and there are just a few more settings to identify when to initiate the patching task, along with the rollout and reboot options, and finally we will deploy our patch job to complete our initial setup.