Google cloud platform project

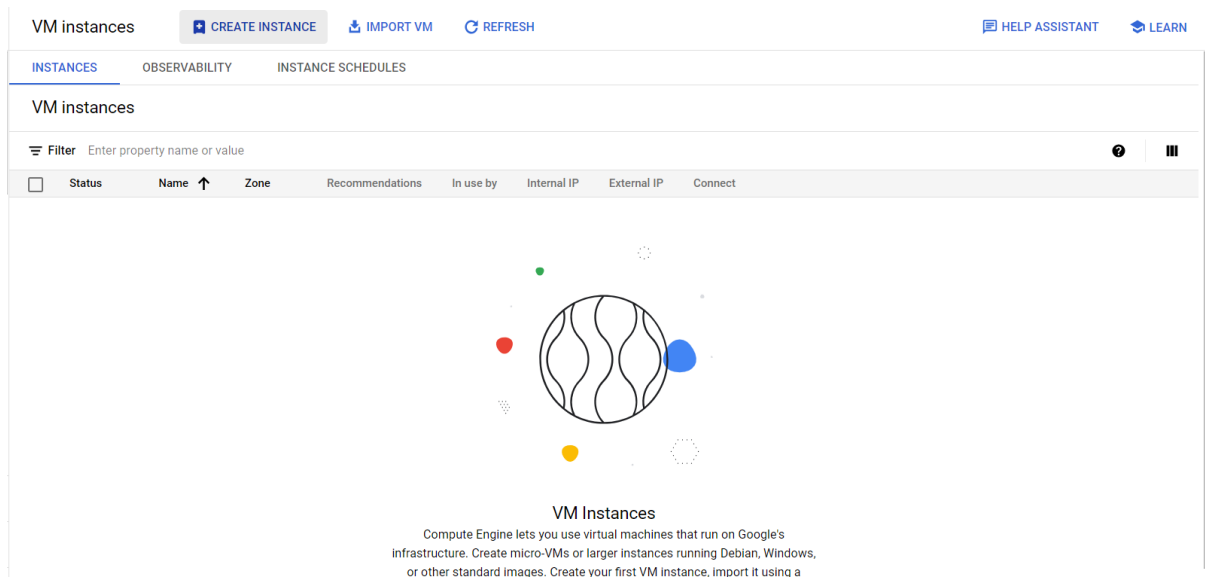## Apache Web Server on Google Cloud

by Abdulaziz alqahtani

Guidelines:

- First make your own GCP account.
- second after you make your account enable compute engine
- third create your first virtual machine

Hands-on-deck $:~

VM settings:

here you make your first VM.

VM instances CREATE INSTANCE IMPORT VM REFRESH   HELP ASSISTANT LEARN

INSTANCES OBSERVABILITY INSTANCE SCHEDULES

VM instances

Filter Enter property name or value

Status Name Zone Recommendations In use by Internal IP External IP Connect

VM Instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a

The boot disk settings.

Boot disk ❓

Name	instance-1
Type	New balanced persistent disk
Size	10 GB
License type ❓	Free
Image	🛡 Debian GNU/Linux 11 (bullseye)

Allow HTTP and HTTPS traffic.

Firewall ❓

Add tags and firewall rules to allow specific network traffic from the Internet

☑ Allow HTTP traffic

☑ Allow HTTPS traffic

VM SHH:

After we create our first VM, we'll run SHH from clicking the SHH button.

VM instances

	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP		Connect	
☐	✅	instance-1	me-central1-a			10.212.0.2 (nic0)	34.18.32.248 ↗ (nic0)		SSH ▾	⋮

**Related actions**   ∧ HIDE

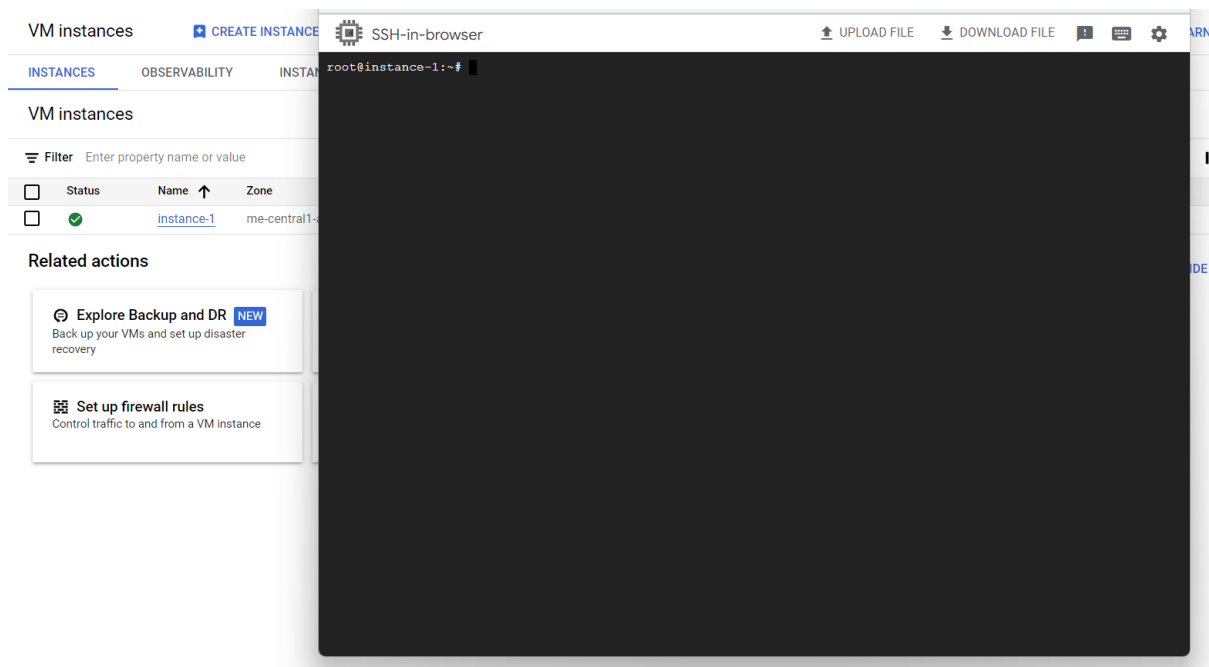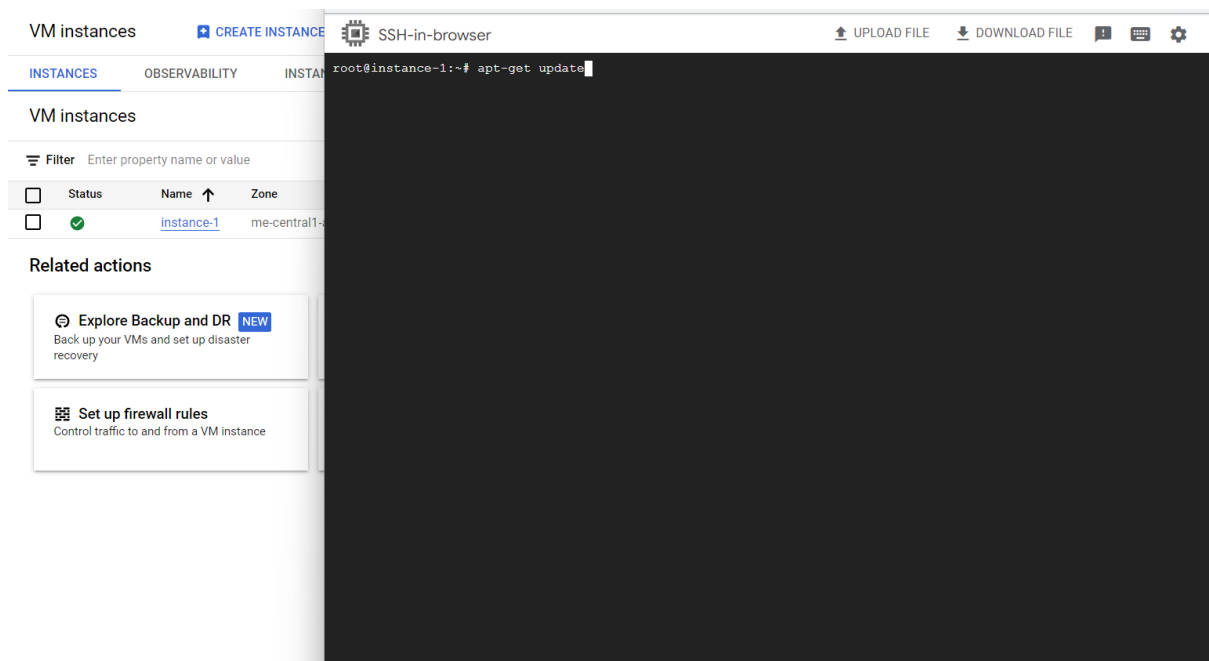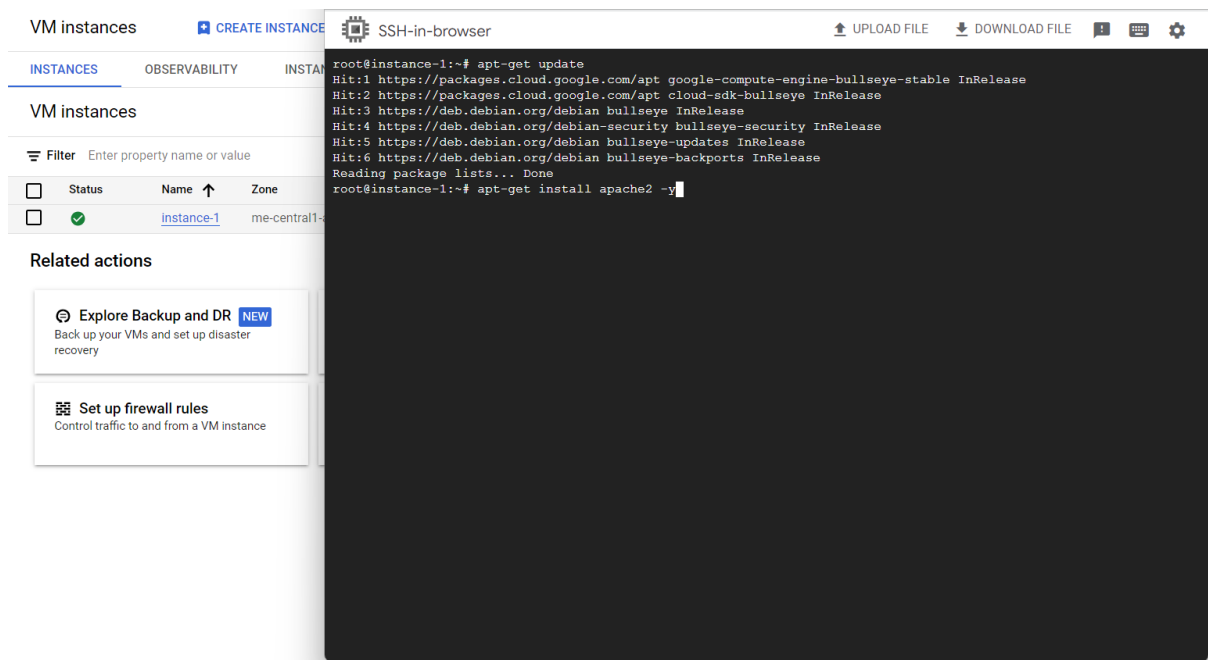| ⊖ **Explore Backup and DR** [NEW] Back up your VMs and set up disaster recovery	▭ **View billing report** View and manage your Compute Engine billing	▮ı **Monitor VMs** View outlier VMs across metrics like CPU and network	▤ **Explore VM logs** View, search, analyze, and download VM instance logs
▦ **Set up firewall rules** Control traffic to and from a VM instance	⟳ **Patch management** Schedule patch updates and view patch compliance on VM instances	⟲ **Load balance between VMs** ↗ Set up Load Balancing for your applications as your traffic and users grow	

Now we open the SHH window.

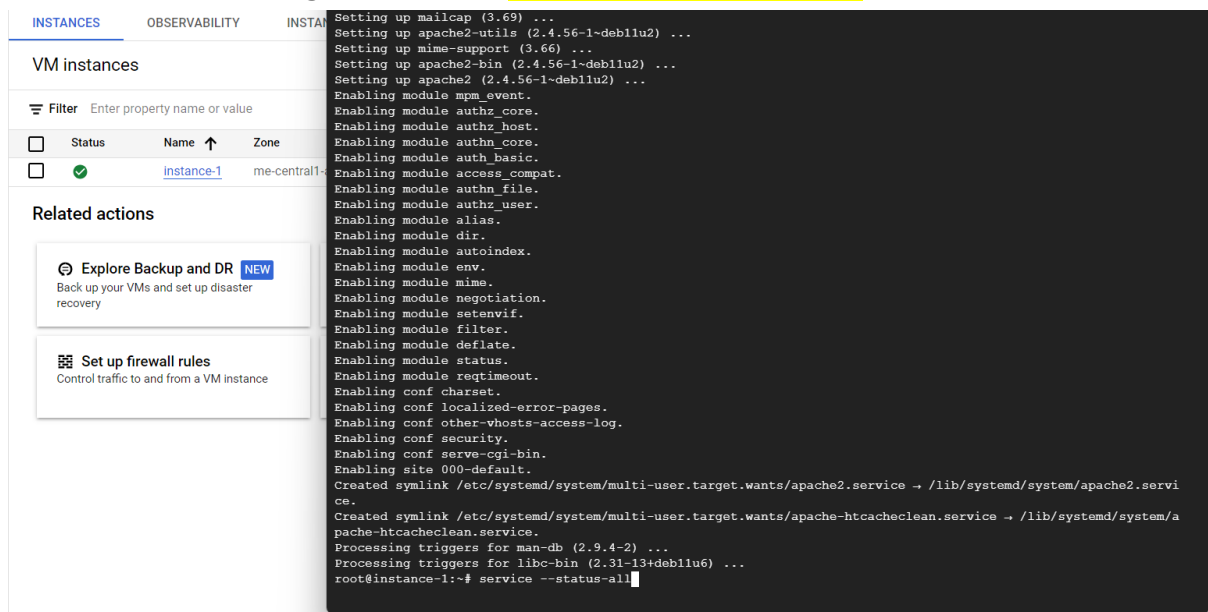First thing is to update packages by the cmd: apt-get update

After that install apache2 server the cmd: ==apt-get install apache2 -y==

we install apache2 server successfully but how to check if the server is working the cmd: ==service -status-all==

The check is completed.


```
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.servi
ce.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/a
pache-htcacheclean.service.
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+deb11u6) ...
root@instance-1:~# service --status-all
 [ + ]  apache-htcacheclean
 [ + ]  apache2
 [ + ]  apparmor
 [ + ]  chrony
 [ + ]  cron
 [ + ]  dbus
 [ + ]  exim4
 [ + ]  haveged
 [ - ]  hwclock.sh
 [ + ]  kmod
 [ + ]  networking
 [ + ]  procps
 [ + ]  rsyslog
 [ - ]  screen-cleanup
 [ + ]  ssh
 [ - ]  sudo
 [ + ]  udev
 [ + ]  unattended-upgrades
 [ - ]  uuidd
root@instance-1:~# ls
root@instance-1:~#
```
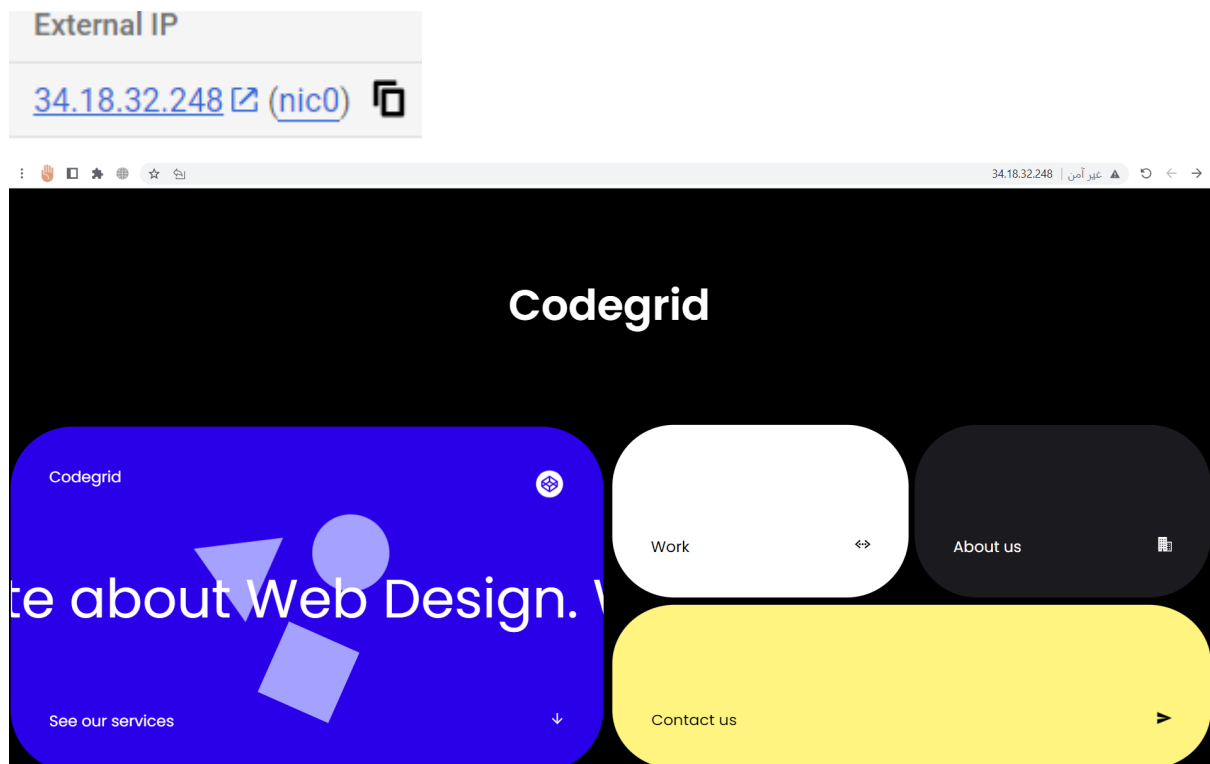
Note:
 if you have index.html upload it and move it to /var/www/html

Finally after you upload your website now go to your external IP
and click on it.

I hope that the project will help you.