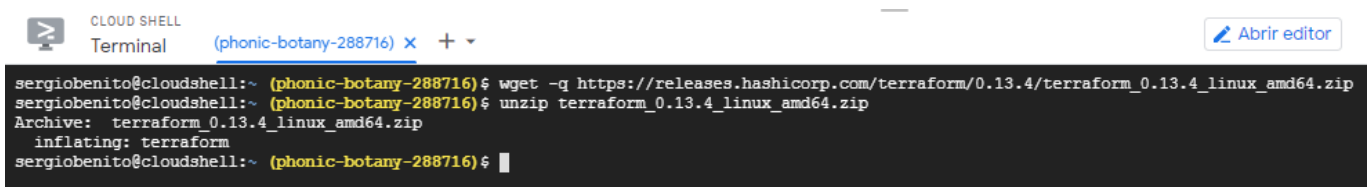


# CREATE VM ENGINE MACHINE WITH TERRAFORM

## 1. Downloading, installing and configuring Terraform

It's necessary to download the stable package of Terraform, then it's moved to `/usr/local/bin`. Introduce the following commands into the Cloud Shell Terminal.

```
wget -q
https://releases.hashicorp.com/terraform/0.13.4/terraform_0.13.4_linux_amd64.zip
unzip terraform_0.11.6_linux_amd64.zip
```

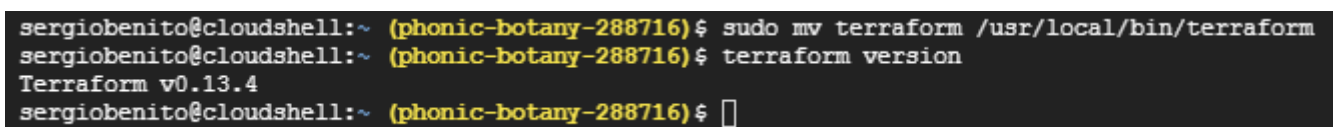


The screenshot shows a Cloud Shell terminal window with the following commands and output:

```
sergiobenito@cloudshell:~ (phonic-botany-288716) $ wget -q https://releases.hashicorp.com/terraform/0.13.4/terraform_0.13.4_linux_amd64.zip
sergiobenito@cloudshell:~ (phonic-botany-288716) $ unzip terraform_0.13.4_linux_amd64.zip
Archive: terraform_0.13.4_linux_amd64.zip
  inflating: terraform
sergiobenito@cloudshell:~ (phonic-botany-288716) $
```

When Terraform package is uncompressed, you can move it and check its version.

```
sudo mv terraform /usr/local/bin/terraform
terraform version
```

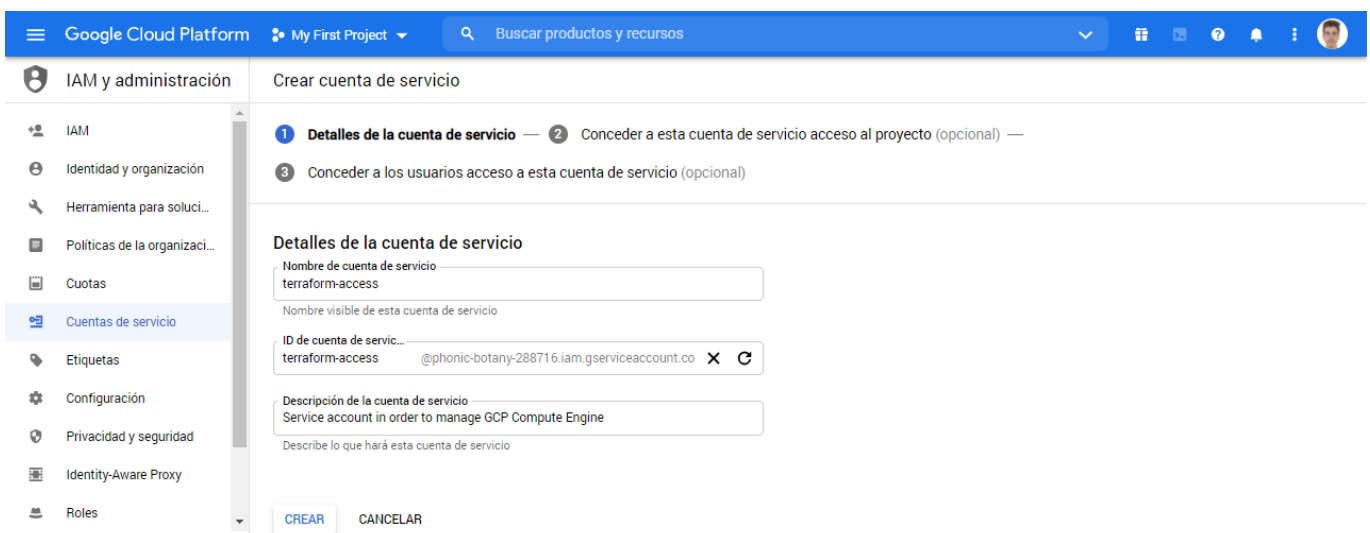


The screenshot shows a Cloud Shell terminal window with the following commands and output:

```
sergiobenito@cloudshell:~ (phonic-botany-288716) $ sudo mv terraform /usr/local/bin/terraform
sergiobenito@cloudshell:~ (phonic-botany-288716) $ terraform version
Terraform v0.13.4
sergiobenito@cloudshell:~ (phonic-botany-288716) $
```

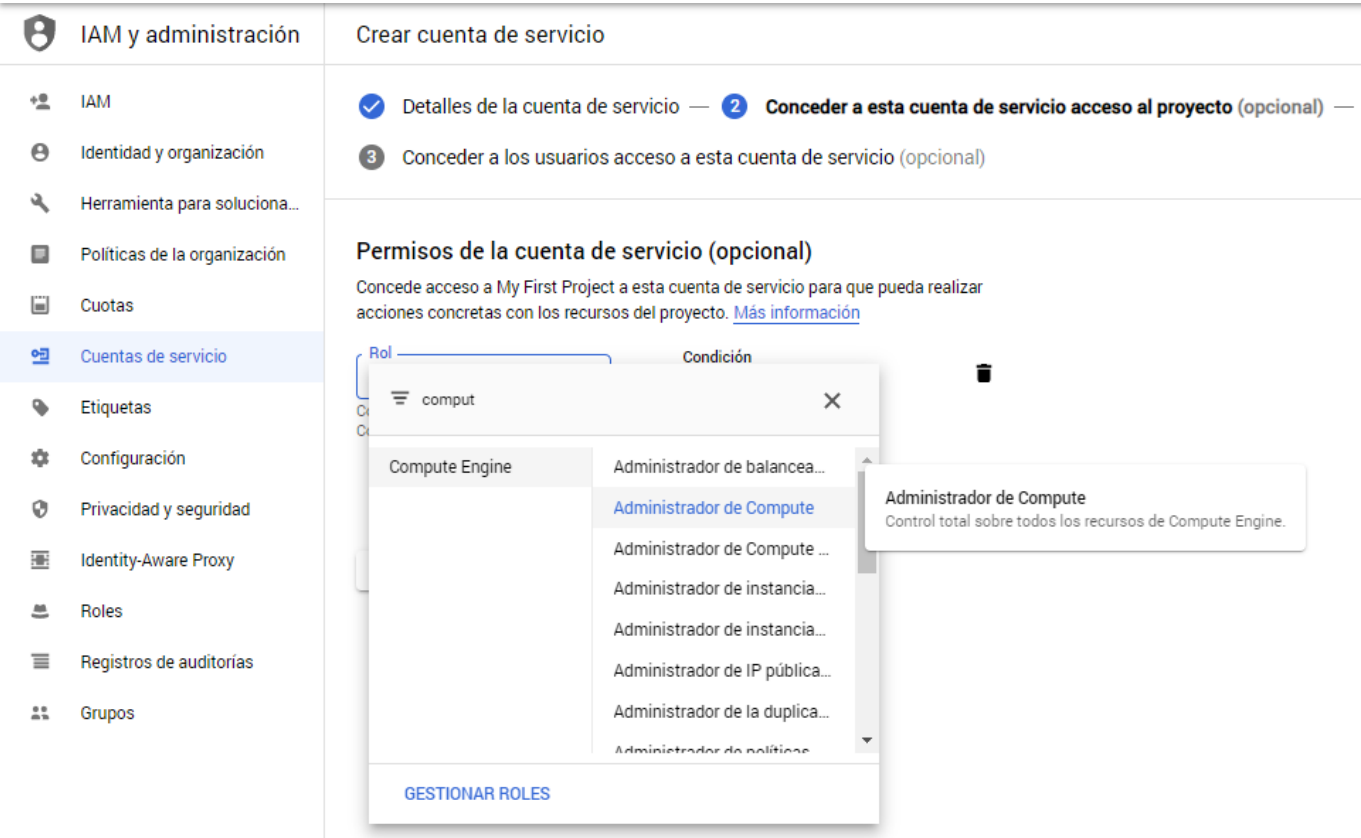
## 2. Configure the Service Account on GCP

In order to allow Terraform to create virtual machines, it has been created a service account which gives specific resources to a group or user.

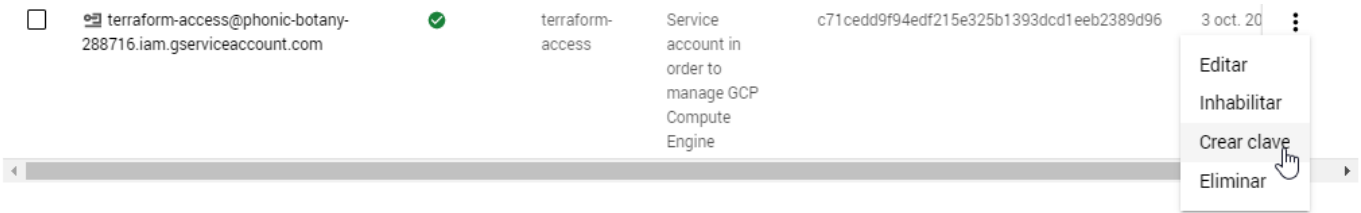


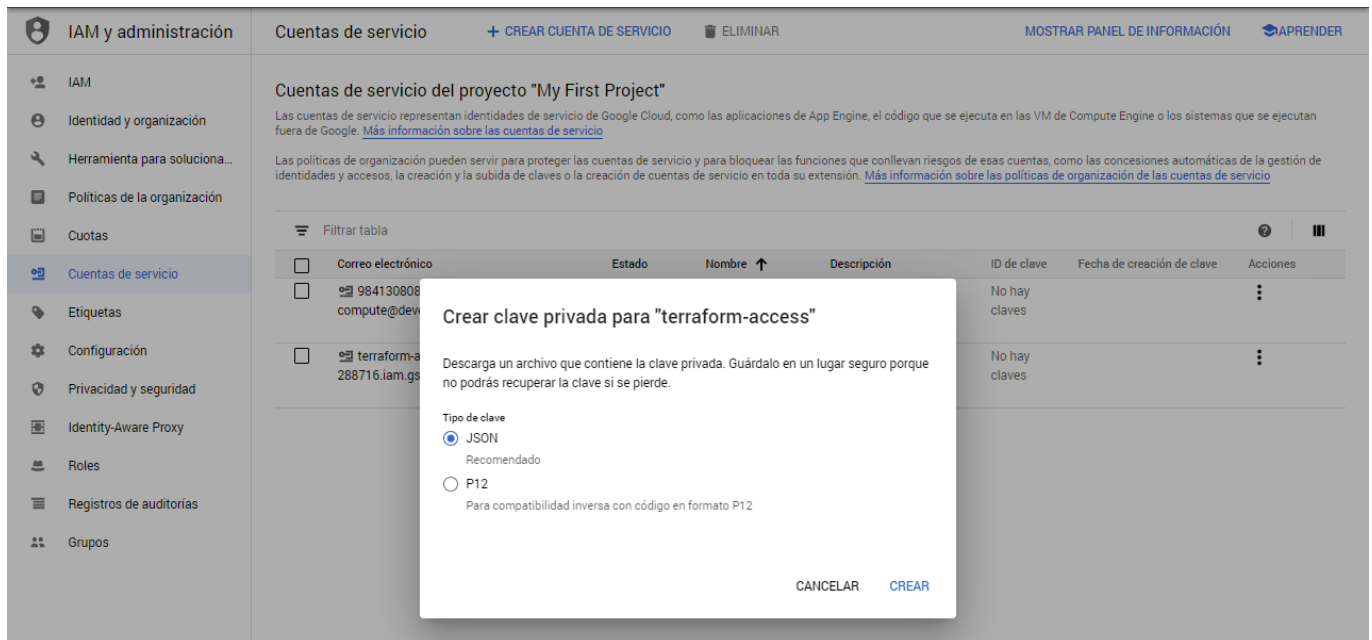
In *IAM & Admin* menu and then *Service accounts* it's possible to create and setting the specific roles for one service account.

For this case, it's needed to add the role *Compute Admin*.



Then, it's possible to create the JSON key, in this step is very important to configure `.gitignore` file to avoid to upload the `credentials.json` to the repository due to be a private key.





### 3. Configure Terraform files

Now, its the moment when it's going to create the config files of Terraform.

#### 3.1. `provider.tf`

Create a file named as `provider.tf` which will contain the configuration needed for provisioning a resource on GCP.

```
# Specify the provider (GCP, AWS, Azure)
provider "google" {
  credentials = "${file("credentials.json")}"
  project = "phonic-botany-288716"
  region = "europe-west1"
}
```

#### 3.2. `instance.tf`

This file contains the resource's configuration of each machine that user wants to create. It has several sections:

- `resource "google_compute_instance" "default"`: Describe the information about the virtual machine, such as type of image, network interface or even the initial script that will going to be executed when bootstrapping.
- `resource "google_compute_firewall" "http-server"`: This section describes the information about the firewall or the access allowed to specific port and its protocol.

For example:

```
resource "google_compute_instance" "default" {
  name          = "name-instance"
```

```
description    = "This template is used to create app server instances."
machine_type   = "e2-medium"
zone           = "europe-west1-b"

tags = ["http-server"]

boot_disk {
  initialize_params {
    size  = "15"
    type  = "pd-standard"
    image = "ubuntu-os-cloud/ubuntu-1804-lts"
  }
}

network_interface {
  network = "default"

  access_config {
    // Ephemeral IP
  }
}

metadata = {
  name = "superset"
}

metadata_startup_script = file("./start_terraform.sh")
}

resource "google_compute_firewall" "http-server" {
  name     = "default-allow-http-terraform"
  network  = "default"

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  // Allow traffic from everywhere to instances with an http-server tag
  source_ranges = ["0.0.0.0/0"]
  target_tags   = ["http-server"]
}

output "ip" {
  value =
    "${google_compute_instance.default.network_interface.0.access_config.0.nat_ip}"
}
```

### 3.3. Create startup script

In a file like `start.sh` you can define the shell commands which you want to run inside compute engine after creation. It's not mandatory but it's very interesting for example when you have to re-create instance after

being deleted.

For example, it's set the commands to install *Docker*.

```
#!/bin/bash
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
bionic stable"
sudo apt update

apt-cache policy docker-ce
sudo apt install docker-ce

sudo usermod -aG docker ${USER}
```

If it's wanted to run, for example, a *Python* command sequence instead of *Bash*, it's necessary to add to the file the following line at the start of the script:

```
#!/usr/bin/python
```

When the instance was created, you can check that the script has been run with the [following command](#):

```
sudo journalctl -u google-startup-scripts.service
```

```

sergiohenito@superset:~$ sudo journalctl -u google-startup-scripts.service
-- Logs begin at Sat 2020-10-03 15:25:52 UTC, end at Sat 2020-10-03 16:06:38 UTC. --
Oct 03 15:26:14 superset systemd[1]: Starting Google Compute Engine Startup Scripts...
Oct 03 15:26:14 superset startup-script[2113]: INFO Starting startup scripts.
Oct 03 15:26:15 superset startup-script[2113]: INFO Found startup-script in metadata.
Oct 03 15:26:15 superset sudo[2144]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt-key add -
Oct 03 15:26:15 superset sudo[2144]: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 03 15:26:15 superset startup-script[2113]: INFO startup-script: Warning: apt-key output should not be parsed (stdout is not a terminal)
Oct 03 15:26:15 superset startup-script[2113]: INFO startup-script: OK
Oct 03 15:26:15 superset sudo[2144]: pam_unix(sudo:session): session closed for user root
Oct 03 15:26:15 superset sudo[2335]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/add-apt-repository deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic InRelease
Oct 03 15:26:15 superset sudo[2335]: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Hit:1 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:2 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:4 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:5 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:6 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:7 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:8 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:9 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:10 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:11 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [168 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:12 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [360 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:13 http://security.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:14 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:15 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1354 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:16 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:17 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:18 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:19 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:20 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:21 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:22 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [168 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:23 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [269 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [151 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1067 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [239 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.3 kB]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2908 B]
Oct 03 15:26:16 superset startup-script[2113]: INFO startup-script: Get:29 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [13.0 kB]
Oct 03 15:26:17 superset startup-script[2113]: INFO startup-script: Get:30 http://archive.canonical.com/ubuntu bionic/partner amd64 Packages [2288 B]
Oct 03 15:26:17 superset startup-script[2113]: INFO startup-script: Get:31 http://archive.canonical.com/ubuntu bionic/partner Translation-en [1332 B]
Oct 03 15:26:23 superset startup-script[2113]: INFO startup-script: Fetched 21.5 MB in 4s (4923 kB/s)
Oct 03 15:26:24 superset startup-script[2113]: INFO startup-script: Reading package lists...
Oct 03 15:26:25 superset sudo[2335]: pam_unix(sudo:session): session closed for user root
Oct 03 15:26:25 superset sudo[2906]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt update
Oct 03 15:26:25 superset sudo[2906]: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:1 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:2 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:3 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:4 http://archive.canonical.com/ubuntu bionic InRelease
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:5 https://download.docker.com/linux/ubuntu bionic InRelease
Oct 03 15:26:25 superset startup-script[2113]: INFO startup-script: Hit:6 http://security.ubuntu.com/ubuntu bionic-security InRelease
Oct 03 15:26:31 superset startup-script[2113]: INFO startup-script: Reading package lists...

```

But if you don't want to use the command and see the specific file, you can read it in the following files:

- CentOS y RHEL: `/var/log/messages`
- Debian: `/var/log/daemon.log`
- Ubuntu: `/var/log/syslog`
- SLES: `/var/log/messages`

## 4. Resources creation

After the creation the previous files, it's ready to run a set of *Terraform* commands for resource creation.

1. **Terraform init:** This command initializes the terraform inside a folder and creates *.terraform* directory.

```
terraform init
```

```
sergiobenito@cloudshell:~ (phonic-botany-288716)$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "google" (hashicorp/google) 3.41.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.google: version = "~> 3.41"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
sergiobenito@cloudshell:~ (phonic-botany-288716)$
```

2. **Terraform plan (optional):** This command is used to create an execution plan. This command from

```
terraform plan
```

```
sergiobenito@cloudshell:~ (phonic-botany-288716)$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

-----

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

# google_compute_firewall.http-server will be created
+ resource "google_compute_firewall" "http-server" {
  + creation_timestamp = (known after apply)
  + destination_ranges = (known after apply)
  + direction          = (known after apply)
  + enable_logging     = (known after apply)
  + id                 = (known after apply)
  + name               = "default-allow-http-terraform"
  + network            = "default"
  + priority           = 1000
  + project            = (known after apply)
  + self_link          = (known after apply)
  + source_ranges      = [
    + "0.0.0.0/0",
  ]
  + target_tags        = [
    + "http-server",
  ]

  + allow {
    + ports = [
      + "80",
    ]
    + protocol = "tcp"
  }
}

# google_compute_instance.default will be created
+ resource "google_compute_instance" "default" {
  + can_ip_forward      = false
  + cpu_platform         = (known after apply)
  + current_status      = (known after apply)
  + deletion_protection = false
  + description          = "This template is used to create app server instances."
  + guest_accelerator    = (known after apply)
  + id                  = (known after apply)
  + instance_id          = (known after apply)
  + label_fingerprint    = (known after apply)
}
```

3. **Terraform apply:** This command is used to apply the changes required to reach the desired state of the configuration, it will apply the pre-determined set of actions generated by `terraform plan` command.

```
terraform apply
```

After the successful execution of these commands, we will see `terraform.tfstate` and `terraform.tfstate.backup` files in your folder. These files save the state of resources which will help update or destroy infrastructure in future. Don't delete these files and keep it safe.



```
sergiobenito@cloudshell:~ (phonic-botany-288716)$ terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

# google_compute_firewall.http-server will be created
+ resource "google_compute_firewall" "http-server" {
  + creation_timestamp = (known after apply)
  + destination_ranges = (known after apply)
  + direction          = (known after apply)
  + enable_logging     = (known after apply)
  + id                 = (known after apply)
  + name               = "default-allow-http-terraform"
  + network            = "default"
  + priority           = 1000
  + project            = (known after apply)
  + self_link          = (known after apply)
  + source_ranges      = [
    + "0.0.0.0/0",
  ]
  + target_tags        = [
    + "http-server",
  ]

  + allow {
    + ports = [
      + "80",
    ]
    + protocol = "tcp"
  }
}

# google_compute_instance.default will be created
+ resource "google_compute_instance" "default" {
  + can_ip_forward      = false
  + cpu_platform         = (known after apply)
  + current_status      = (known after apply)
  + deletion_protection = false
  + description         = "This template is used to create app server instances."
  + guest_accelerator   = (known after apply)
  + id                  = (known after apply)
  + instance_id         = (known after apply)
  + label_fingerprint   = (known after apply)
  + machine_type        = "e2-medium"
  + metadata             = {
    + "name" = "superset"
  }
  + metadata_fingerprint = (known after apply)
  + metadata_startup_script = "echo hi > /test.txt"
  + min_cpu_platform     = (known after apply)
}
```

You have to set yes in order to create the specific resources.

```

+ boot_disk {
  + auto_delete           = true
  + device_name           = (known after apply)
  + disk_encryption_key_sha256 = (known after apply)
  + kms_key_self_link     = (known after apply)
  + mode                 = "READ_WRITE"
  + source               = (known after apply)

  + initialize_params {
    + image = "ubuntu-os-cloud/ubuntu-1804-lts"
    + labels = (known after apply)
    + size   = (known after apply)
    + type   = (known after apply)
  }
}

+ network_interface {
  + name           = (known after apply)
  + network        = "default"
  + network_ip     = (known after apply)
  + subnetwork     = (known after apply)
  + subnetwork_project = (known after apply)

  + access_config {
    + nat_ip       = (known after apply)
    + network_tier = (known after apply)
  }
}

+ scheduling {
  + automatic_restart = (known after apply)
  + on_host_maintenance = (known after apply)
  + preemptible        = (known after apply)

  + node_affinities {
    + key       = (known after apply)
    + operator  = (known after apply)
    + values    = (known after apply)
  }
}

+ scratch_disk {
  + interface = "SCSI"
}

```

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

Then if everything was ok, the resources are created.

```

+ mode          = "READ_WRITE"
+ source        = (known after apply)

+ initialize_params {
+   image = "ubuntu-os-cloud/ubuntu-1804-lts"
+   labels = (known after apply)
+   size  = 15
+   type  = "pd-standard"
+ }

+ network_interface {
+   name          = (known after apply)
+   network       = "default"
+   network_ip    = (known after apply)
+   subnetwork    = (known after apply)
+   subnetwork_project = (known after apply)

+   access_config {
+     nat_ip    = (known after apply)
+     network_tier = (known after apply)
+   }
+ }

+ scheduling {
+   automatic_restart = (known after apply)
+   on_host_maintenance = (known after apply)
+   preemptible        = (known after apply)

+   node_affinities {
+     key      = (known after apply)
+     operator = (known after apply)
+     values   = (known after apply)
+   }
+ }
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

google_compute_firewall.http-server: Creating...
google_compute_instance.default: Creating...
google_compute_firewall.http-server: Still creating... [10s elapsed]
google_compute_instance.default: Still creating... [10s elapsed]
google_compute_firewall.http-server: Creation complete after 11s [id=projects/phonic-botany-288716/global/firewalls/default-allow-http-terraform]
google_compute_instance.default: Creation complete after 13s [id=projects/phonic-botany-288716/zones/europe-west1-b/instances/superset]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:
ip = 34.76.208.88
sergiobenito@cloudshell:~ (phonic-botany-288716) $

```

Now, you can go to your *VM Engine* console and you will see the instances which have been created by *Terraform*.

The screenshot shows the Google Cloud Platform VM Engine console. On the left is a sidebar with navigation options: Instancias de VM, Grupos de instancias, Plantillas de instancias, Nodos de único cliente, Imágenes de la máquina, Discos, Capturas, Imágenes, TPUs, Migración en Compute E..., Descuentos por uso conf..., Metadatos, Comprobaciones estado, Zonas, Grupos de puntos de con..., Operaciones, Análisis de seguridad, Gestión de parches de SO, Configuración, and Marketplace. The main panel is titled 'Detalles de la instancia de VM' and shows details for an instance named 'superset'. The instance is in the 'Running' state. It has an SSH access method, Cloud Logging enabled, and a serial console. The description states: 'This template is used to create app server instances.' The instance ID is 634974173446613258. The machine type is 'e2-medium (2 vCPUs, 4 GB de memoria)'. The reservation is 'Elegir automáticamente (predeterminado)'. The CPU platform is 'Intel Haswell'. The screen device is 'Activar dispositivo de pantalla'. The zone is 'europe-west1-b'. The tags are 'Ninguna'. The creation time is '3 oct. 2020 17:25:26'. At the bottom, there is a table for network interfaces:

Nombre	Red	Subred	IP interna principal	Intervalos de IP de alias	IP externa	Nivel de red	Reenvío de IP	Detalles de la red
nic0	default	default	10.132.0.17	—	34.76.0.97 (efímera)	Premium	Desactivado	Ver detalles

## 5. Useful links

- [Create your first Compute Engine\(VM\) in GCP using Terraform](#)
- [Ejecuta secuencias de comandos de inicio](#)
- [HashiCorp Learn - Build Infrastructure](#)
- [How to Use Terraform to Create a Virtual Machine in Google Cloud Platform](#)
- [How to Use Terraform with Google Cloud Platform?](#)
- [Terraform - google\\_compute\\_instance\\_template](#)