# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**

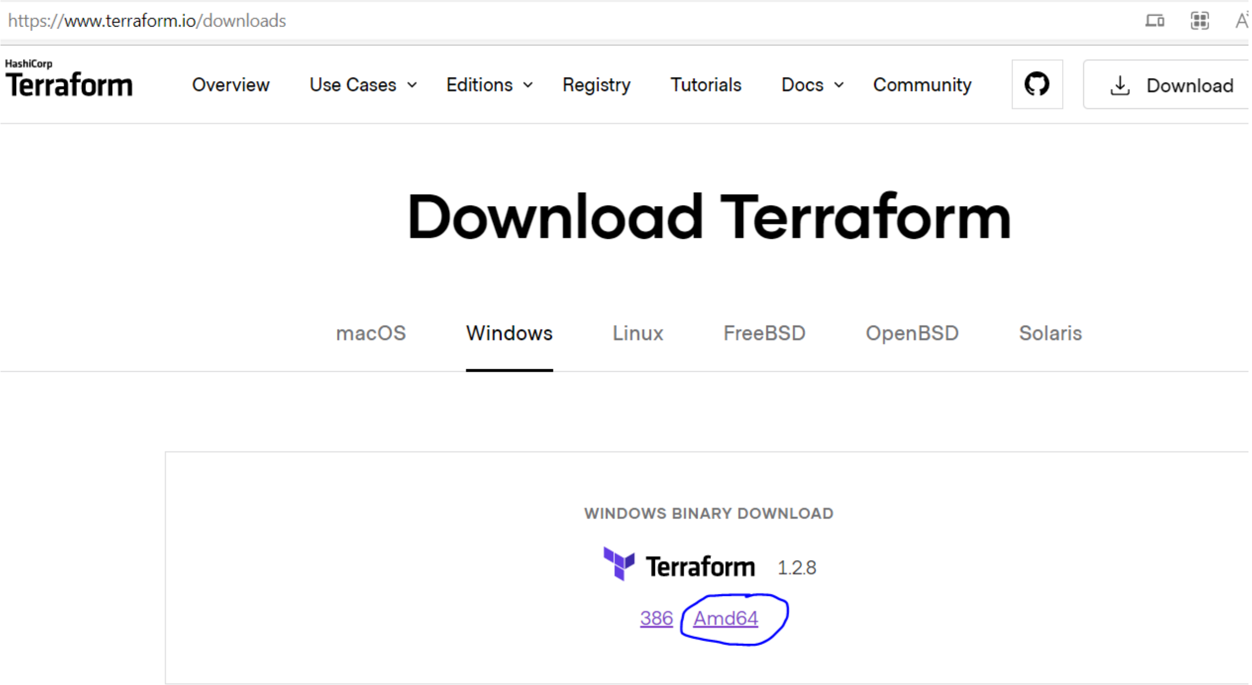
## Terraform GCP pre-requisites:

- Install Terraform and define in Windows environment path if it is in Windows OS
- Generate Access Key

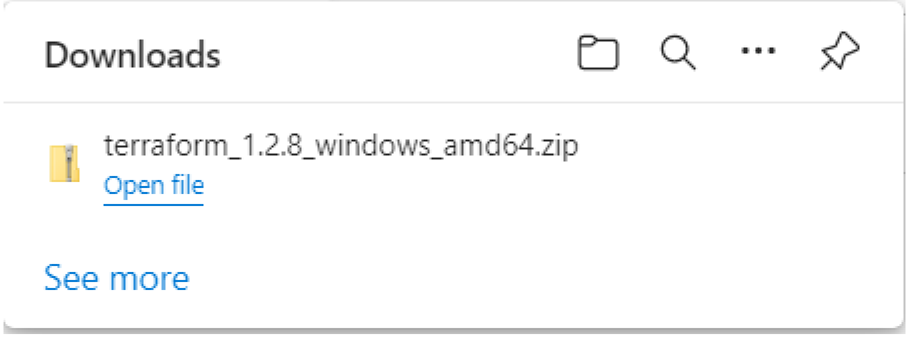
## Pre-requisite: Install Terraform

**Download terraform exe file in terraform site:**
https://www.terraform.io/downloads

**Since I am using windows, go to windows tab and click Amd64**



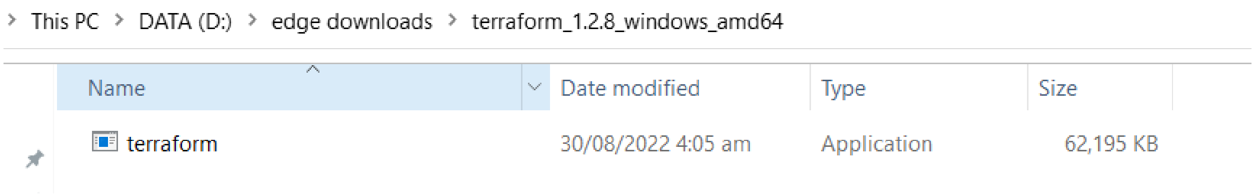
**Then it will download**



**Then extract file you can see terraform executable file**

# Terraform GCP Sample: VM Instances and VPC

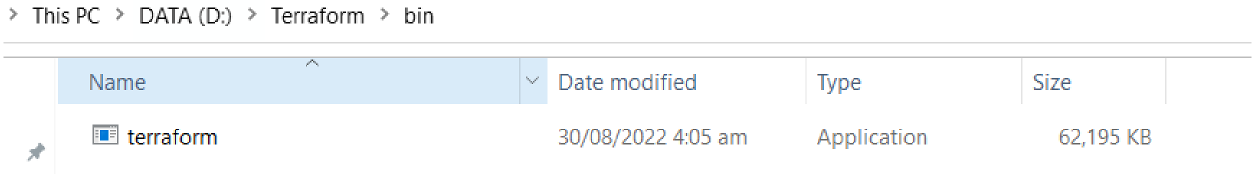
**By Ronald Stewart Lim**



› This PC › DATA (D:) › edge downloads › terraform_1.2.8_windows_amd64

| Name | Date modified | Type | Size |
|---|---|---|---|
| terraform | 30/08/2022 4:05 am | Application | 62,195 KB |

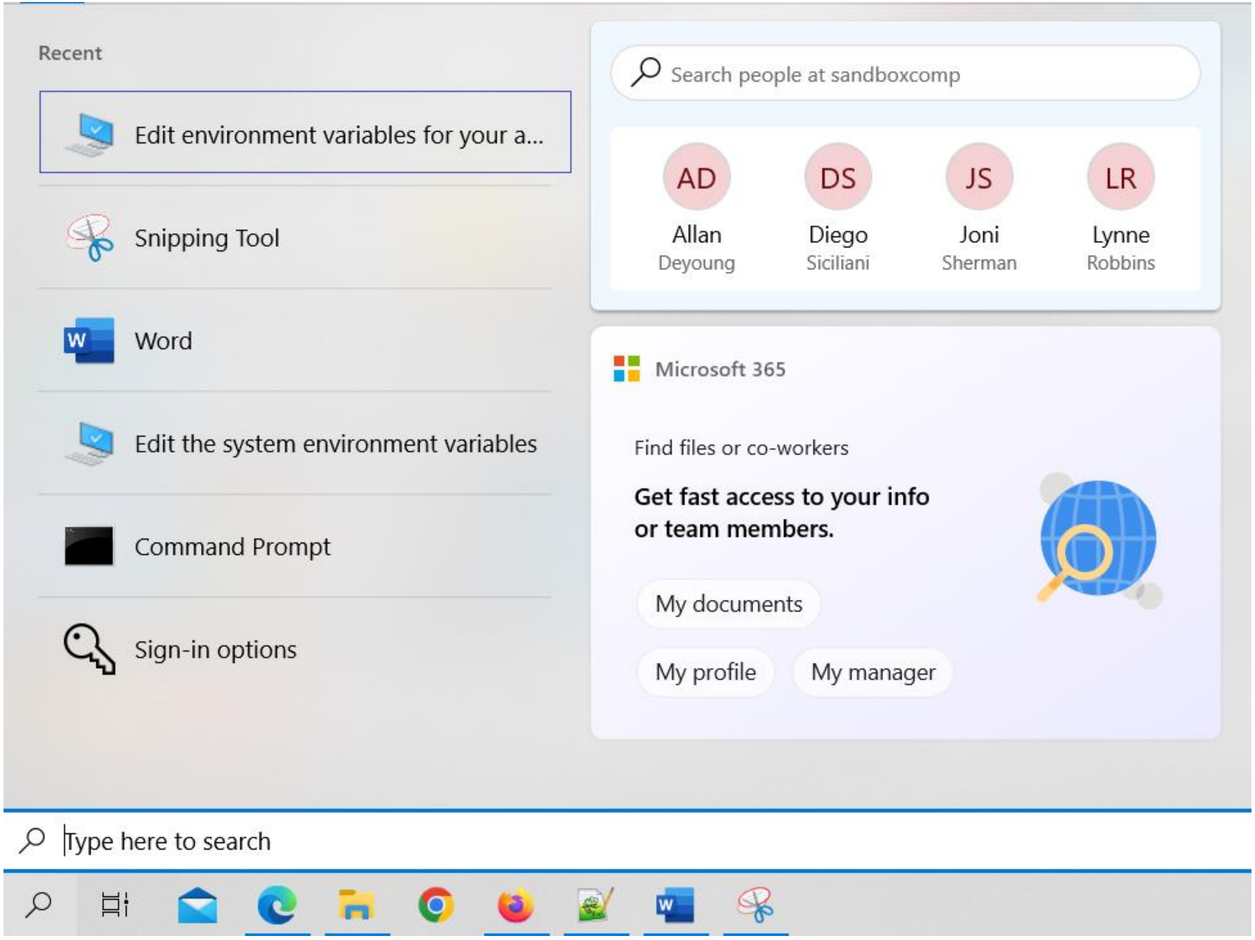
**Then create a directory and copy the file**



› This PC › DATA (D:) › Terraform › bin

| Name | Date modified | Type | Size |
|---|---|---|---|
| terraform | 30/08/2022 4:05 am | Application | 62,195 KB |

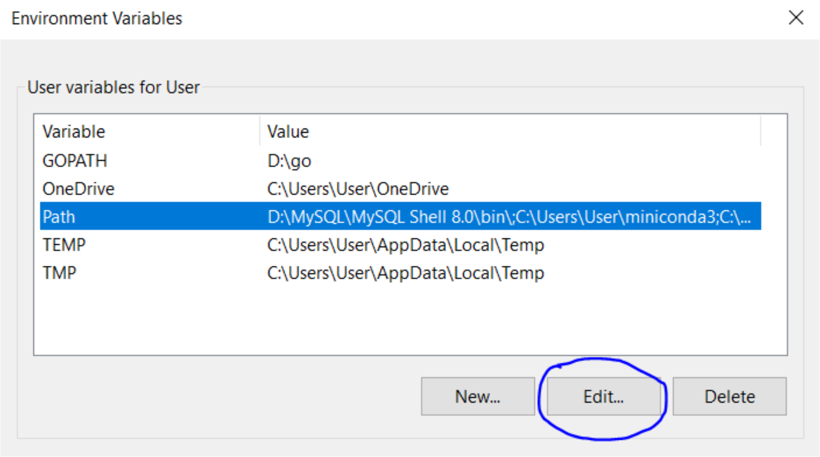
**Then edit windows user path variable**



**Go to Path variable and click Edit… button**

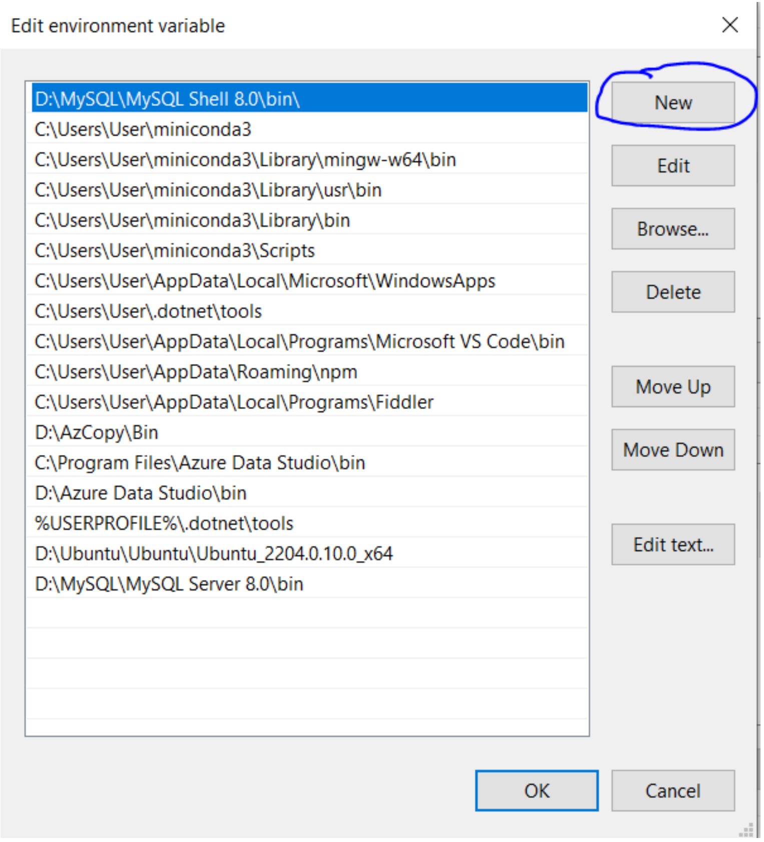
# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**

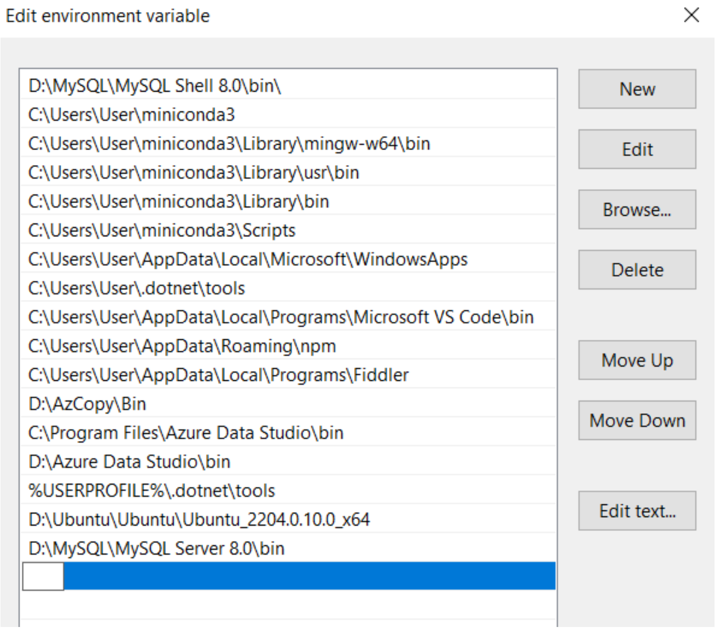


| Variable | Value |
|---|---|
| GOPATH | D:\go |
| OneDrive | C:\Users\User\OneDrive |
| Path | D:\MySQL\MySQL Shell 8.0\bin\;C:\Users\User\miniconda3;C:\... |
| TEMP | C:\Users\User\AppData\Local\Temp |
| TMP | C:\Users\User\AppData\Local\Temp |

### Click New button to Add Terraform path



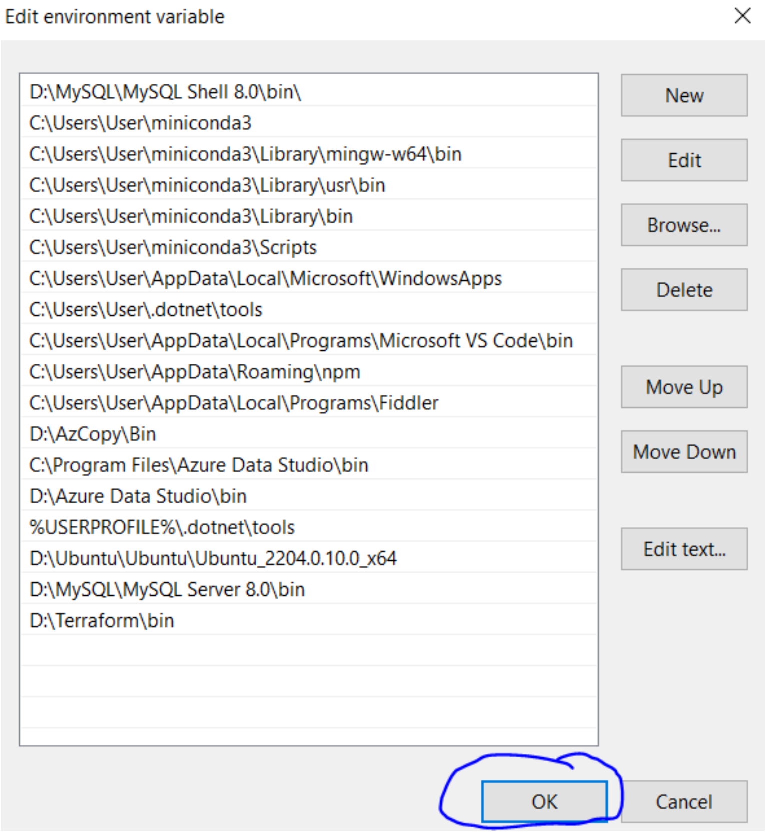
D:\MySQL\MySQL Shell 8.0\bin\
C:\Users\User\miniconda3
C:\Users\User\miniconda3\Library\mingw-w64\bin
C:\Users\User\miniconda3\Library\usr\bin
C:\Users\User\miniconda3\Library\bin
C:\Users\User\miniconda3\Scripts
C:\Users\User\AppData\Local\Microsoft\WindowsApps
C:\Users\User\.dotnet\tools
C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\User\AppData\Roaming\npm
C:\Users\User\AppData\Local\Programs\Fiddler
D:\AzCopy\Bin
C:\Program Files\Azure Data Studio\bin
D:\Azure Data Studio\bin
%USERPROFILE%\.dotnet\tools
D:\Ubuntu\Ubuntu\Ubuntu_2204.0.10.0_x64
D:\MySQL\MySQL Server 8.0\bin



Edit environment variable

D:\MySQL\MySQL Shell 8.0\bin\
C:\Users\User\miniconda3
C:\Users\User\miniconda3\Library\mingw-w64\bin
C:\Users\User\miniconda3\Library\usr\bin
C:\Users\User\miniconda3\Library\bin
C:\Users\User\miniconda3\Scripts
C:\Users\User\AppData\Local\Microsoft\WindowsApps
C:\Users\User\.dotnet\tools
C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\User\AppData\Roaming\npm
C:\Users\User\AppData\Local\Programs\Fiddler
D:\AzCopy\Bin
C:\Program Files\Azure Data Studio\bin
D:\Azure Data Studio\bin
%USERPROFILE%\.dotnet\tools
D:\Ubuntu\Ubuntu\Ubuntu_2204.0.10.0_x64
D:\MySQL\MySQL Server 8.0\bin

**Click OK Button after specifying the path**



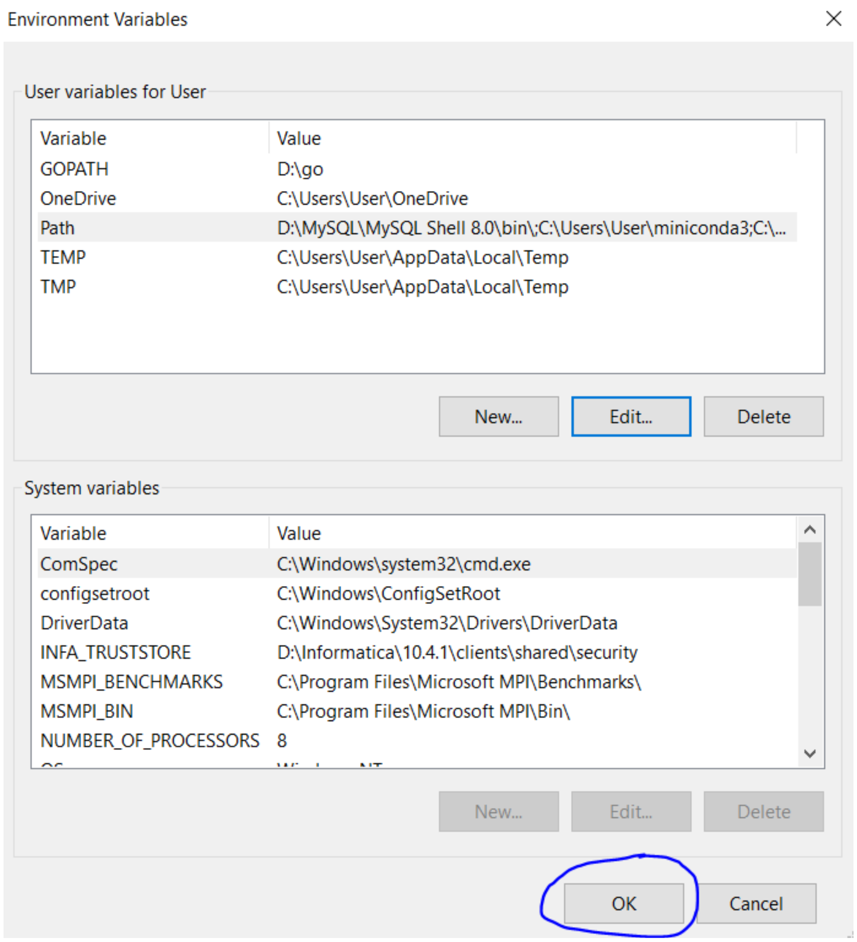
Edit environment variable

D:\MySQL\MySQL Shell 8.0\bin\
C:\Users\User\miniconda3
C:\Users\User\miniconda3\Library\mingw-w64\bin
C:\Users\User\miniconda3\Library\usr\bin
C:\Users\User\miniconda3\Library\bin
C:\Users\User\miniconda3\Scripts
C:\Users\User\AppData\Local\Microsoft\WindowsApps
C:\Users\User\.dotnet\tools
C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\User\AppData\Roaming\npm
C:\Users\User\AppData\Local\Programs\Fiddler
D:\AzCopy\Bin
C:\Program Files\Azure Data Studio\bin
D:\Azure Data Studio\bin
%USERPROFILE%\.dotnet\tools
D:\Ubuntu\Ubuntu\Ubuntu_2204.0.10.0_x64
D:\MySQL\MySQL Server 8.0\bin
D:\Terraform\bin

**Click OK Button again**

# Terraform GCP Sample: VM Instances and VPC

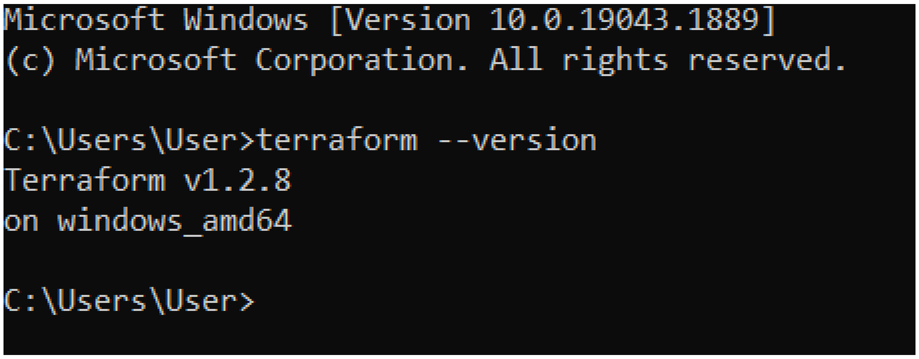
**By Ronald Stewart Lim**



| Variable | Value |
|---|---|
| GOPATH | D:\go |
| OneDrive | C:\Users\User\OneDrive |
| Path | D:\MySQL\MySQL Shell 8.0\bin\;C:\Users\User\miniconda3;C:\... |
| TEMP | C:\Users\User\AppData\Local\Temp |
| TMP | C:\Users\User\AppData\Local\Temp |

| Variable | Value |
|---|---|
| ComSpec | C:\Windows\system32\cmd.exe |
| configsetroot | C:\Windows\ConfigSetRoot |
| DriverData | C:\Windows\System32\Drivers\DriverData |
| INFA_TRUSTSTORE | D:\Informatica\10.4.1\clients\shared\security |
| MSMPI_BENCHMARKS | C:\Program Files\Microsoft MPI\Benchmarks\ |
| MSMPI_BIN | C:\Program Files\Microsoft MPI\Bin\ |
| NUMBER_OF_PROCESSORS | 8 |

**Open cmd and try to check if terraform is already accessible in any directory by typing:** ***terraform --version***



```
Microsoft Windows [Version 10.0.19043.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>terraform --version
Terraform v1.2.8
on windows_amd64

C:\Users\User>
```

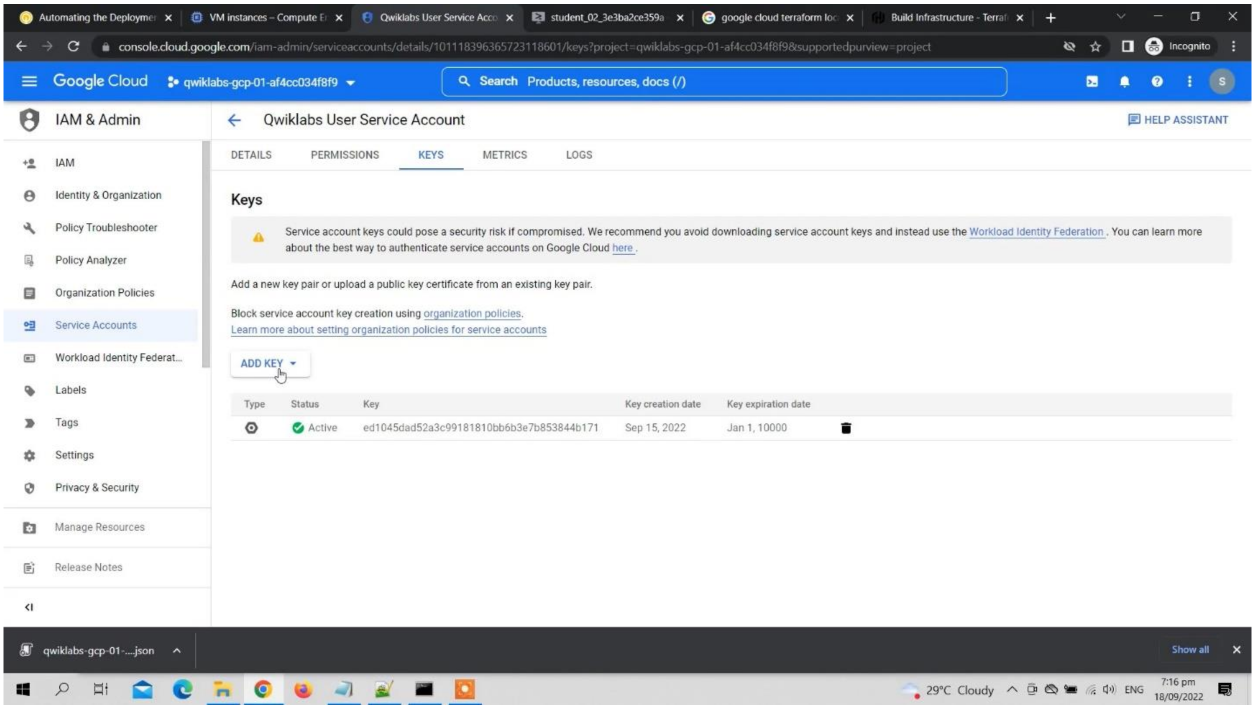
## Pre-requisite: Generate Access Key

**Now Terraform environment are ready. We will prepare the GCP access key required for executing Terraform scripts**

# Terraform GCP Sample: VM Instances and VPC

## By Ronald Stewart Lim

### Go to Google Cloud Console IAM Service Accounts and click a Service Account



Keys

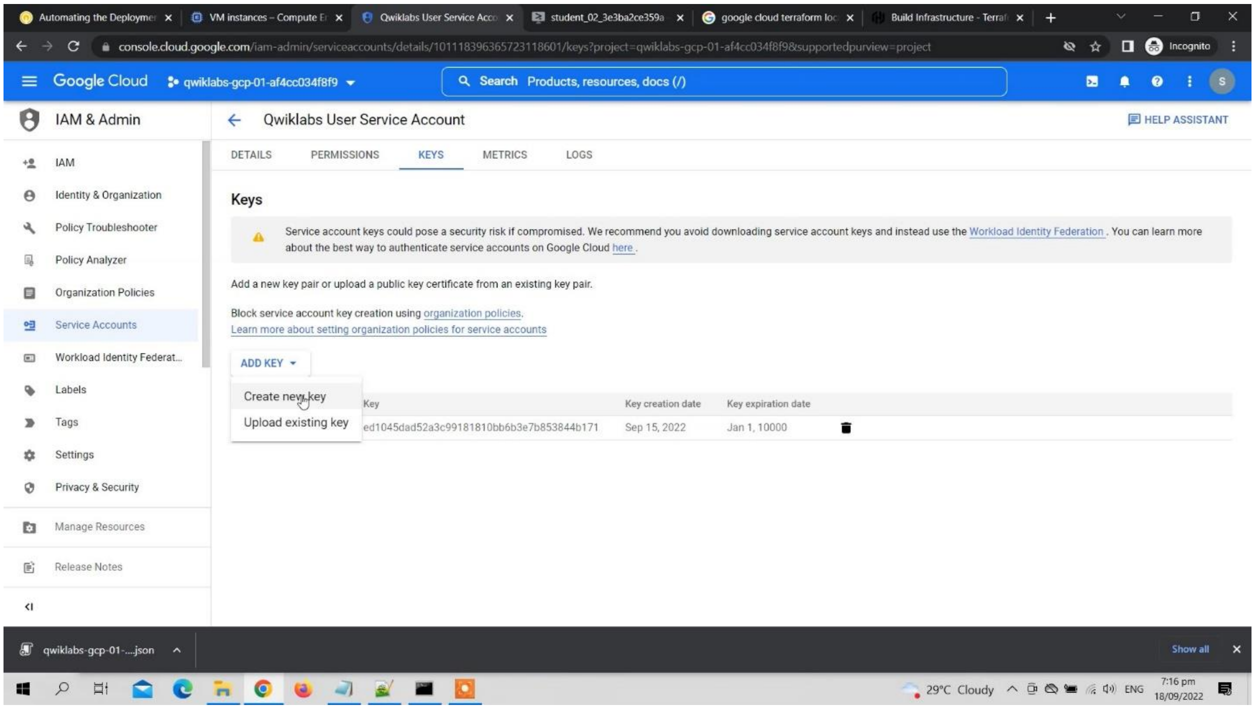
Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

| Type | Status | Key | Key creation date | Key expiration date |
|---|---|---|---|---|
| | Active | ed1045dad52a3c99181810bb6b3e7b853844b171 | Sep 15, 2022 | Jan 1, 10000 |

### Click Add Key and click Create new key and choose JSON


console.cloud.google.com/iam-admin/serviceaccounts/details/101118396365723118601/keys?project=qwiklabs-gcp-01-af4cc034f8f9&supportedpurview=project
Google Cloud
qwiklabs-gcp-01-af4cc034f8f9
IAM & Admin
Qwiklabs User Service Account
DETAILS
PERMISSIONS
KEYS
METRICS
LOGS
ADD KEY
Create new key
Upload existing key
qwiklabs-gcp-01-....json
Show all

Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .

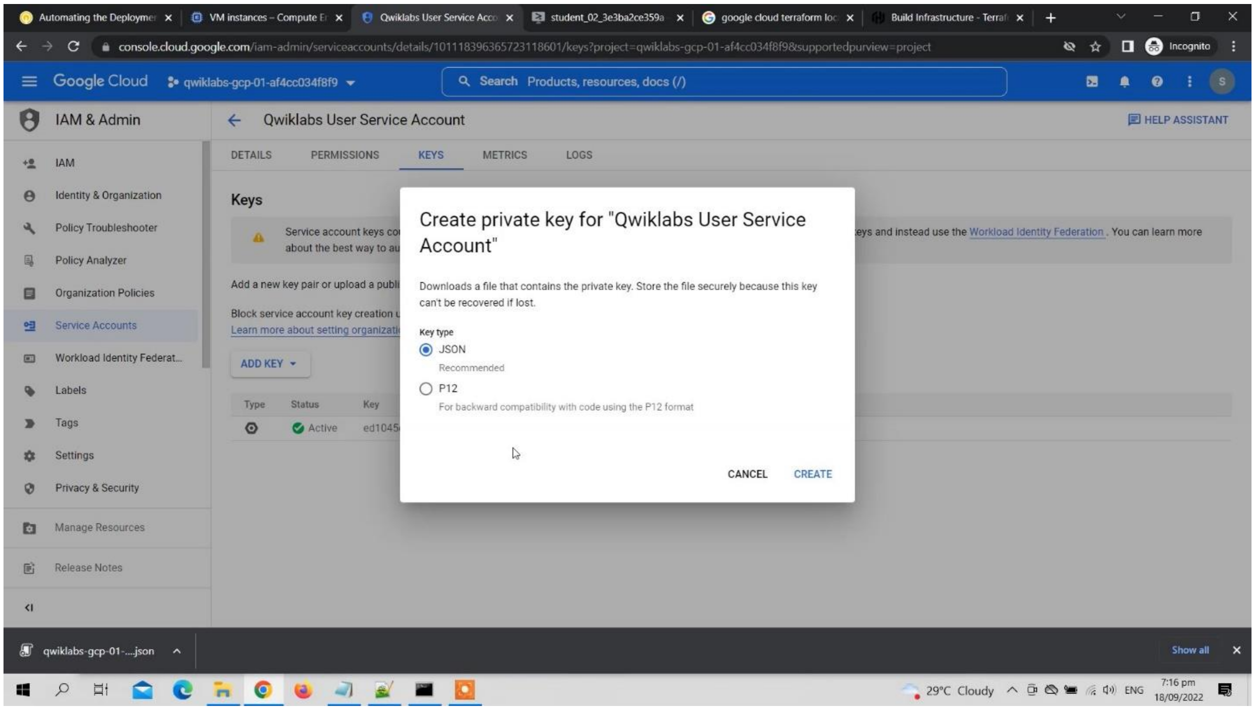
Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

| Key | Key creation date | Key expiration date |
|---|---|---|
| ed1045dad52a3c99181810bb6b3e7b853844b171 | Sep 15, 2022 | Jan 1, 10000 |

# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**



Create private key for "Qwiklabs User Service Account"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

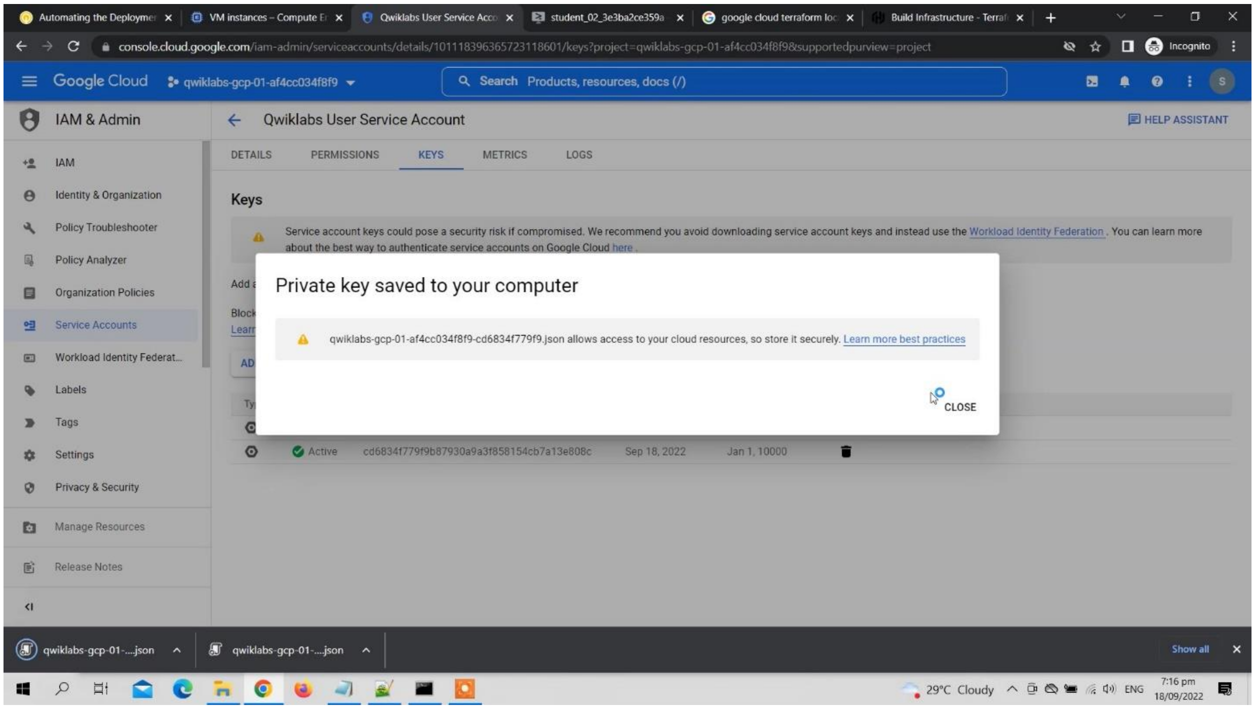
Key type

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format

CANCEL CREATE

**Then click CREATE button. Notice that the JSON Private Key file is downloaded.**



Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .

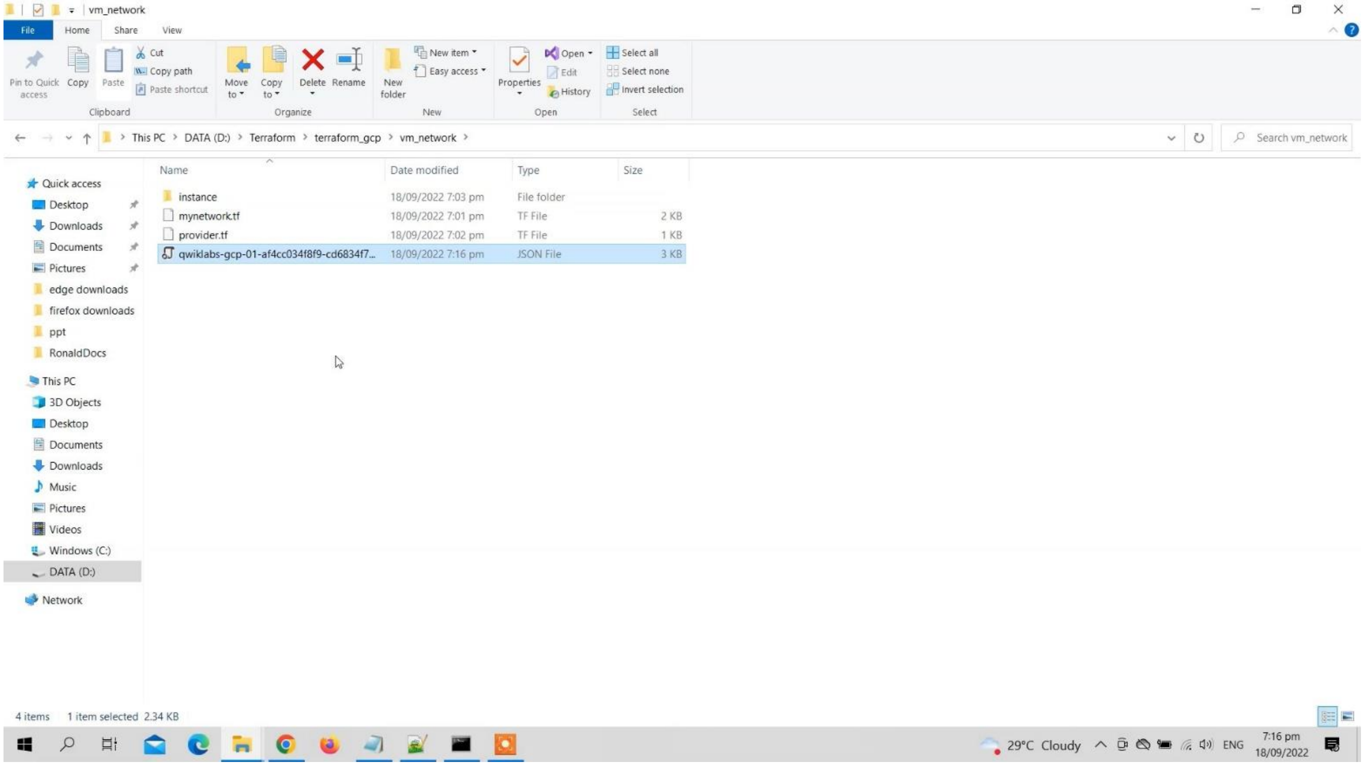
Private key saved to your computer

qwiklabs-gcp-01-af4cc034f8f9-cd6834f779f9.json allows access to your cloud resources, so store it securely. Learn more best practices

CLOSE

Active cd6834f779f9b87930a9a3f858154cb7a13e808c Sep 18, 2022 Jan 1, 10000

**Then copy the file to the desired directory**



| Name | Date modified | Type | Size |
|---|---|---|---|
| instance | 18/09/2022 7:03 pm | File folder | |
| mynetwork.tf | 18/09/2022 7:01 pm | TF File | 2 KB |
| provider.tf | 18/09/2022 7:02 pm | TF File | 1 KB |
| qwiklabs-gcp-01-af4cc034f8f9-cd6834f7... | 18/09/2022 7:16 pm | JSON File | 3 KB |

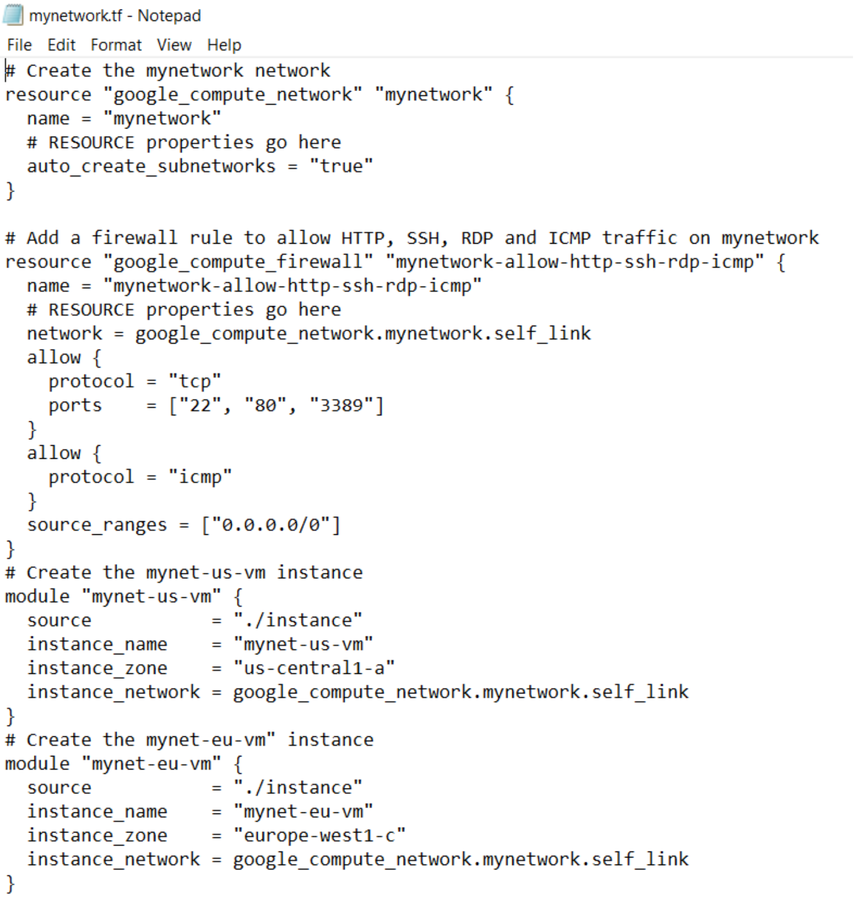
4 items 1 item selected 2.34 KB

Please see sample files to be used

**mynetwork.tf**

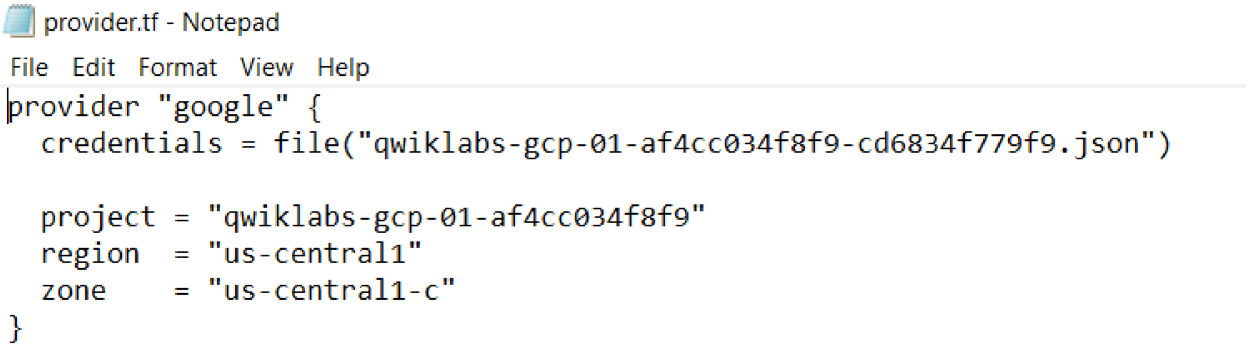
# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**



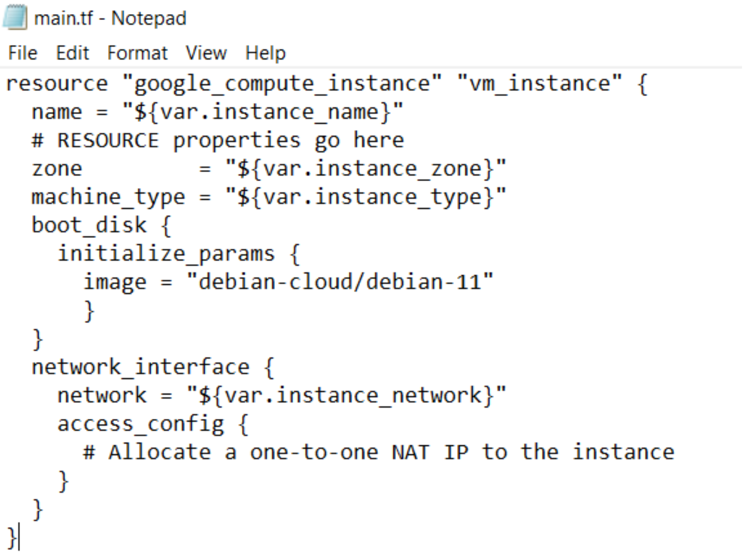
```
# Create the mynetwork network
resource "google_compute_network" "mynetwork" {
  name = "mynetwork"
  # RESOURCE properties go here
  auto_create_subnetworks = "true"
}

# Add a firewall rule to allow HTTP, SSH, RDP and ICMP traffic on mynetwork
resource "google_compute_firewall" "mynetwork-allow-http-ssh-rdp-icmp" {
  name = "mynetwork-allow-http-ssh-rdp-icmp"
  # RESOURCE properties go here
  network = google_compute_network.mynetwork.self_link
  allow {
    protocol = "tcp"
    ports    = ["22", "80", "3389"]
  }
  allow {
    protocol = "icmp"
  }
  source_ranges = ["0.0.0.0/0"]
}
# Create the mynet-us-vm instance
module "mynet-us-vm" {
  source           = "./instance"
  instance_name    = "mynet-us-vm"
  instance_zone    = "us-central1-a"
  instance_network = google_compute_network.mynetwork.self_link
}
# Create the mynet-eu-vm" instance
module "mynet-eu-vm" {
  source           = "./instance"
  instance_name    = "mynet-eu-vm"
  instance_zone    = "europe-west1-c"
  instance_network = google_compute_network.mynetwork.self_link
}
```

**Provider.tf (Note, you need to specify the project ID and the location of Private Key JSON file and region and zone**



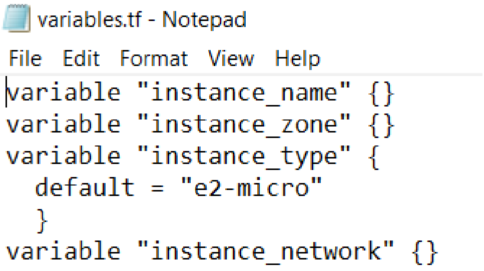
```
provider "google" {
  credentials = file("qwiklabs-gcp-01-af4cc034f8f9-cd6834f779f9.json")

  project = "qwiklabs-gcp-01-af4cc034f8f9"
  region  = "us-central1"
  zone    = "us-central1-c"
}
```

**main.tf**



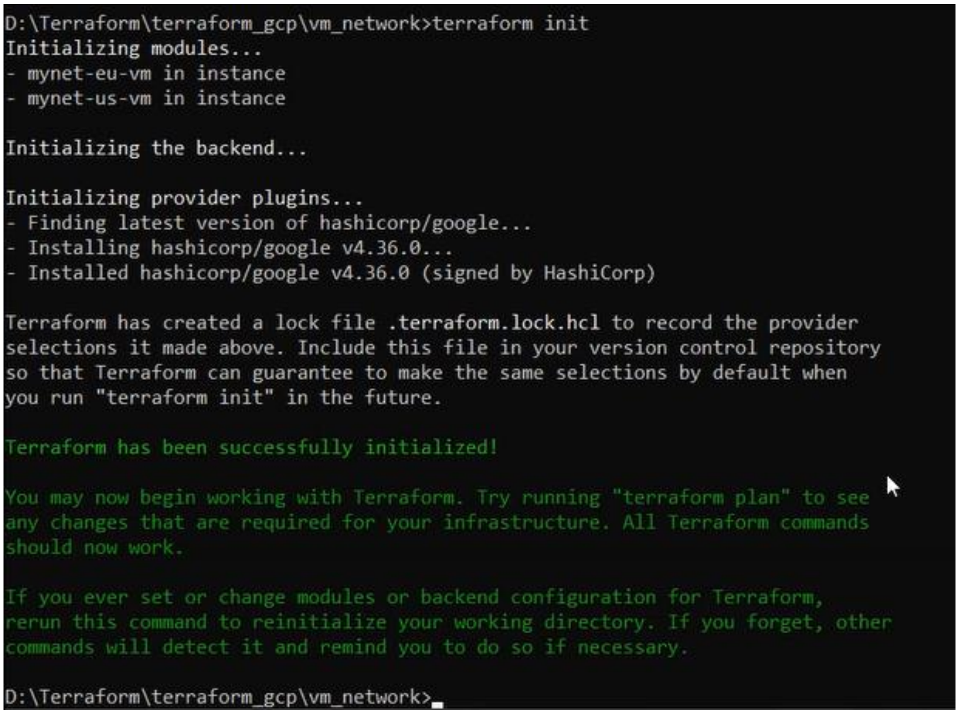
```
resource "google_compute_instance" "vm_instance" {
  name = "${var.instance_name}"
  # RESOURCE properties go here
  zone         = "${var.instance_zone}"
  machine_type = "${var.instance_type}"
  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
      }
  }
  network_interface {
    network = "${var.instance_network}"
    access_config {
      # Allocate a one-to-one NAT IP to the instance
    }
  }
}
```

**variables.tf**



```
variable "instance_name" {}
variable "instance_zone" {}
variable "instance_type" {
  default = "e2-micro"
  }
variable "instance_network" {}
```

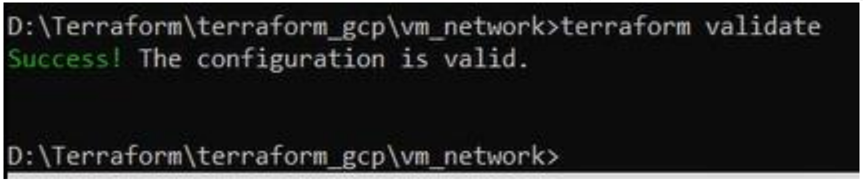
**Initialize the Terraform using command below:**
***terraform init***



```
D:\Terraform\terraform_gcp\vm_network>terraform init
Initializing modules...
- mynet-eu-vm in instance
- mynet-us-vm in instance

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/google...
- Installing hashicorp/google v4.36.0...
- Installed hashicorp/google v4.36.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.

D:\Terraform\terraform_gcp\vm_network>
```

**To check and validate the terraform-based codes, execute the command below:**
***terraform validate***

**If the validation found no errors, then it will output the success message**

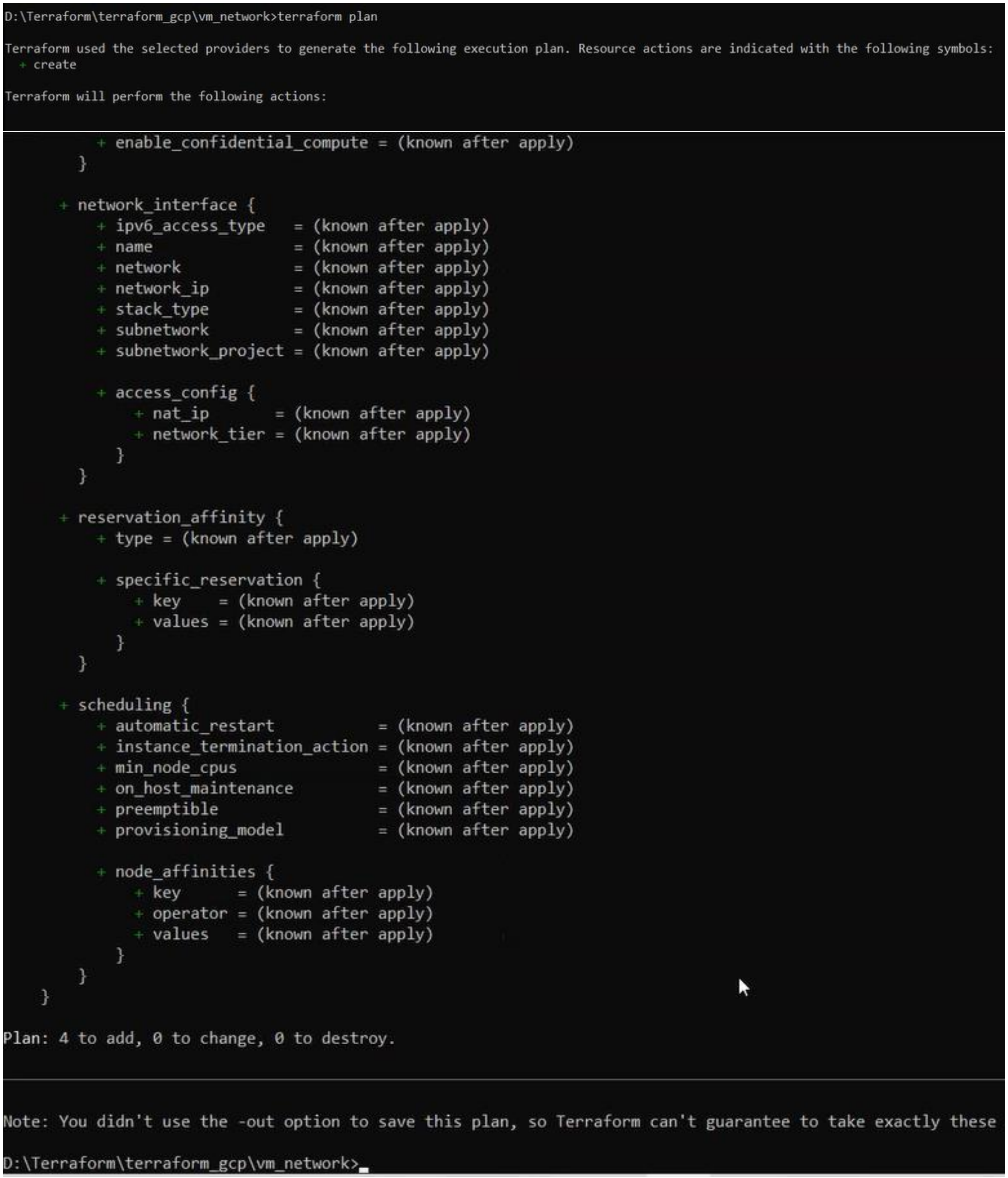


```
D:\Terraform\terraform_gcp\vm_network>terraform validate
Success! The configuration is valid.


D:\Terraform\terraform_gcp\vm_network>
```

**(Optional) Check the Terraform execution plan by using the command below:**
***terraform plan***
Sample output below:

# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**

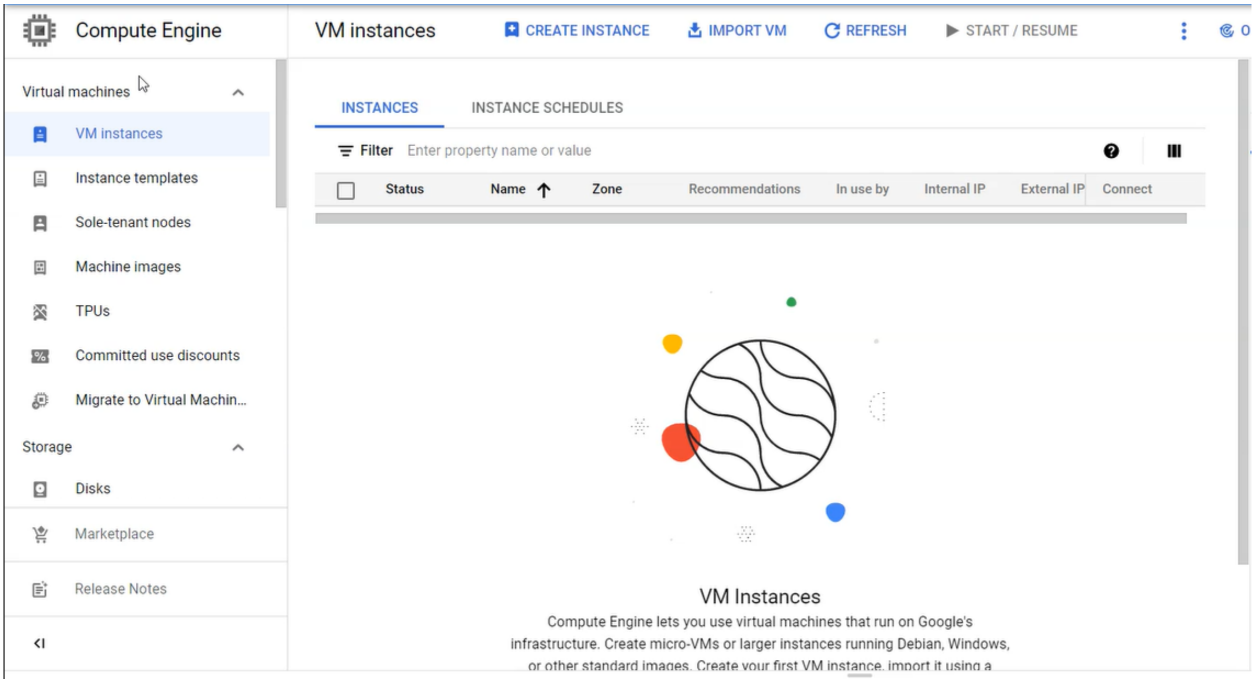


```
D:\Terraform\terraform_gcp\vm_network>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

          + enable_confidential_compute = (known after apply)
        }

      + network_interface {
          + ipv6_access_type   = (known after apply)
          + name               = (known after apply)
          + network            = (known after apply)
          + network_ip         = (known after apply)
          + stack_type         = (known after apply)
          + subnetwork         = (known after apply)
          + subnetwork_project = (known after apply)

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

      + reservation_affinity {
          + type = (known after apply)

          + specific_reservation {
              + key    = (known after apply)
              + values = (known after apply)
            }
        }

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

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

Plan: 4 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these

D:\Terraform\terraform_gcp\vm_network>_
```

**Before applying changes, check the before state of the VM instances by going to Compute Engine -> VM instances. Notice that the instances are not yet created.**

# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**



VM Instances

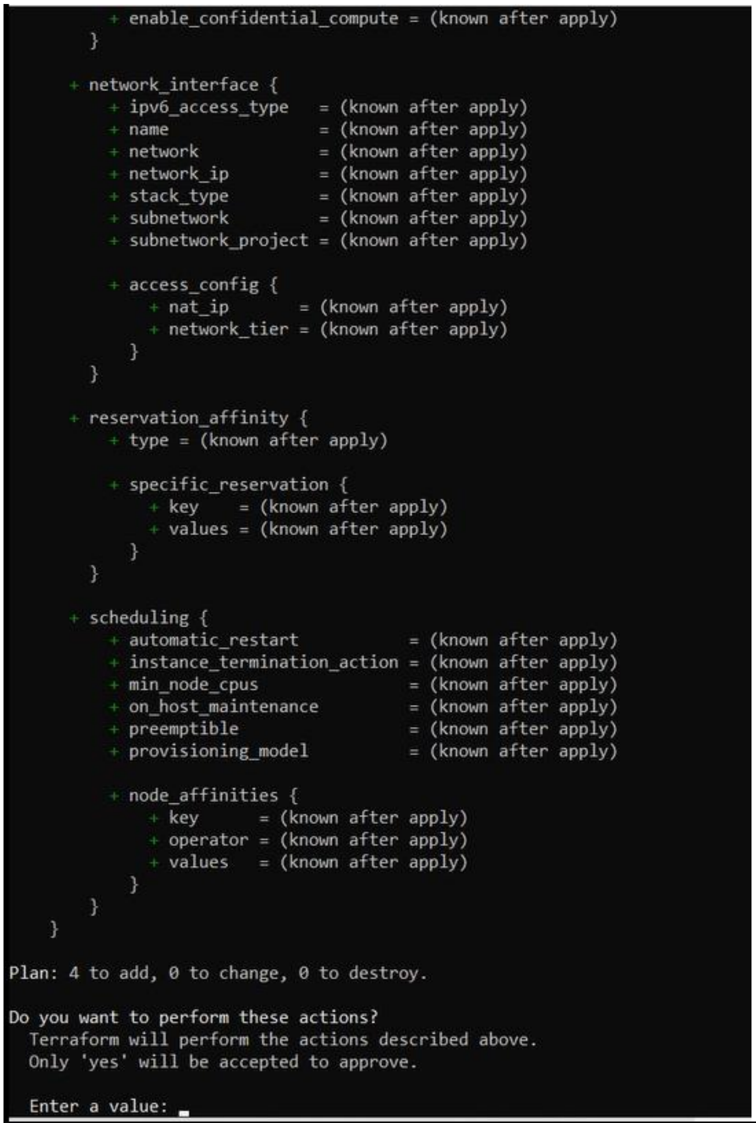
Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a

**Execute the command below and confirm changes to create VM instances in GCP:**
***terraform apply***

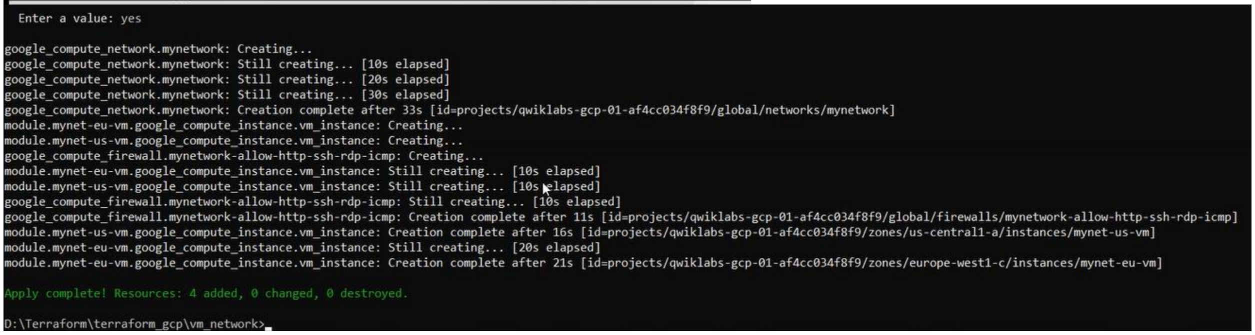
**Note: terraform apply command already contains terraform plan so no need to execute terraform plan**
**terraform apply = terraform plan + confirmation to apply changes**

sample output:

```
D:\Terraform\terraform_gcp\vm_network>terraform apply
```

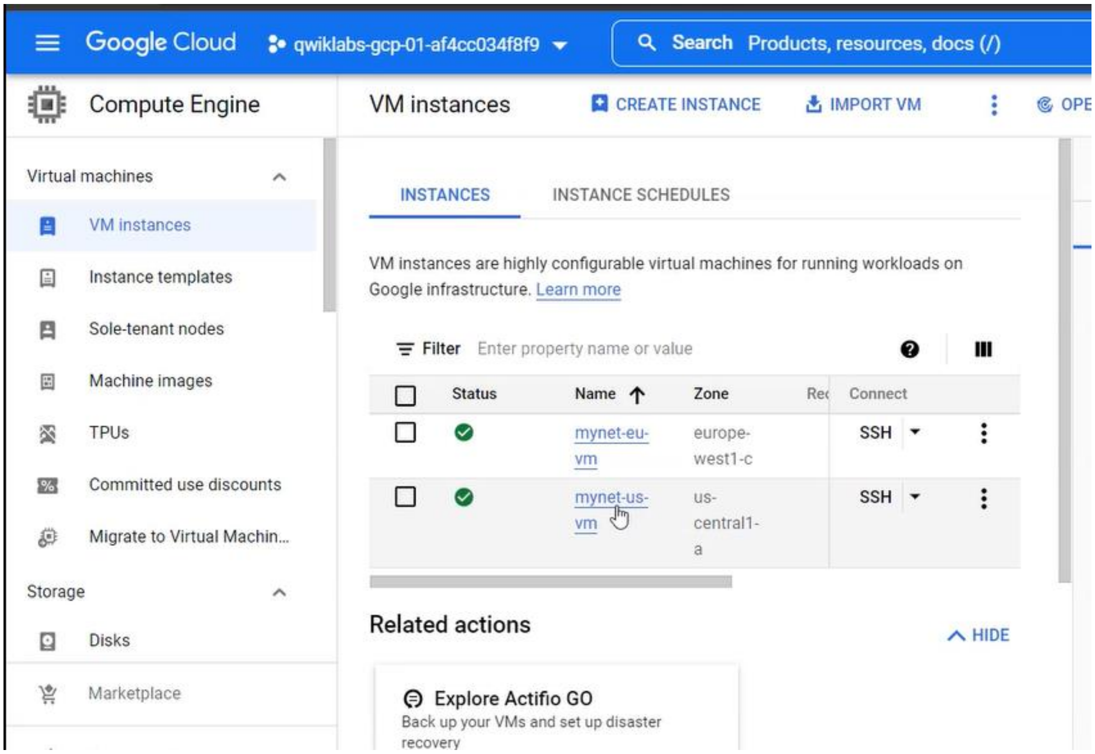


```
          + enable_confidential_compute = (known after apply)
        }

      + network_interface {
          + ipv6_access_type   = (known after apply)
          + name               = (known after apply)
          + network            = (known after apply)
          + network_ip         = (known after apply)
          + stack_type         = (known after apply)
          + subnetwork         = (known after apply)
          + subnetwork_project = (known after apply)

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

      + reservation_affinity {
          + type = (known after apply)

          + specific_reservation {
              + key    = (known after apply)
              + values = (known after apply)
            }
        }

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

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

Plan: 4 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: 
```



```
  Enter a value: yes

google_compute_network.mynetwork: Creating...
google_compute_network.mynetwork: Still creating... [10s elapsed]
google_compute_network.mynetwork: Still creating... [20s elapsed]
google_compute_network.mynetwork: Still creating... [30s elapsed]
google_compute_network.mynetwork: Creation complete after 33s [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork]
module.mynet-eu-vm.google_compute_instance.vm_instance: Creating...
module.mynet-us-vm.google_compute_instance.vm_instance: Creating...
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Creating...
module.mynet-eu-vm.google_compute_instance.vm_instance: Still creating... [10s elapsed]
module.mynet-us-vm.google_compute_instance.vm_instance: Still creating... [10s elapsed]
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Still creating... [10s elapsed]
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Creation complete after 11s [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/firewalls/mynetwork-allow-http-ssh-rdp-icmp]
module.mynet-us-vm.google_compute_instance.vm_instance: Creation complete after 16s [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/instances/mynet-us-vm]
module.mynet-eu-vm.google_compute_instance.vm_instance: Still creating... [20s elapsed]
module.mynet-eu-vm.google_compute_instance.vm_instance: Creation complete after 21s [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/europe-west1-c/instances/mynet-eu-vm]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

D:\Terraform\terraform_gcp\vm_network>
```

**Check the after state of the VM instances by going to Compute Engine -> VM instances. Notice that the instances are now created successfully.**



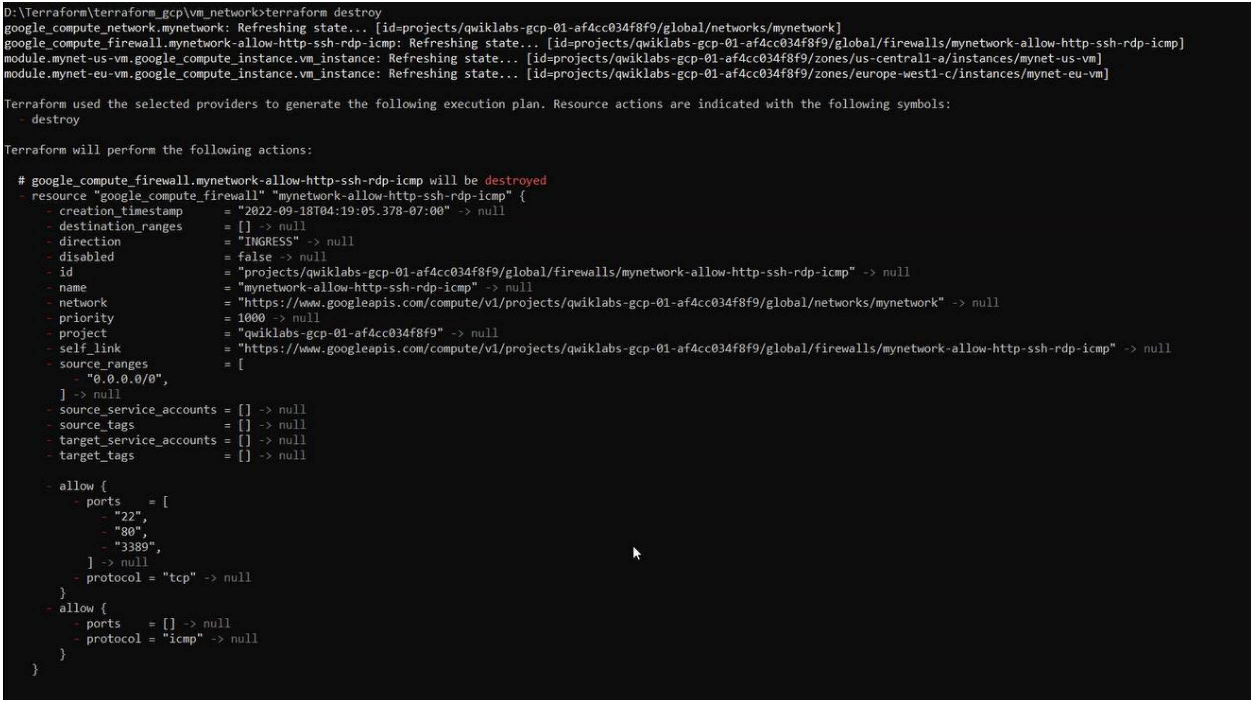
VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

| □ | Status | Name | Zone | Rec | Connect |
|---|---|---|---|---|---|
| □ | ✓ | mynet-eu-vm | europe-west1-c | | SSH |
| □ | ✓ | mynet-us-vm | us-central1-a | | SSH |

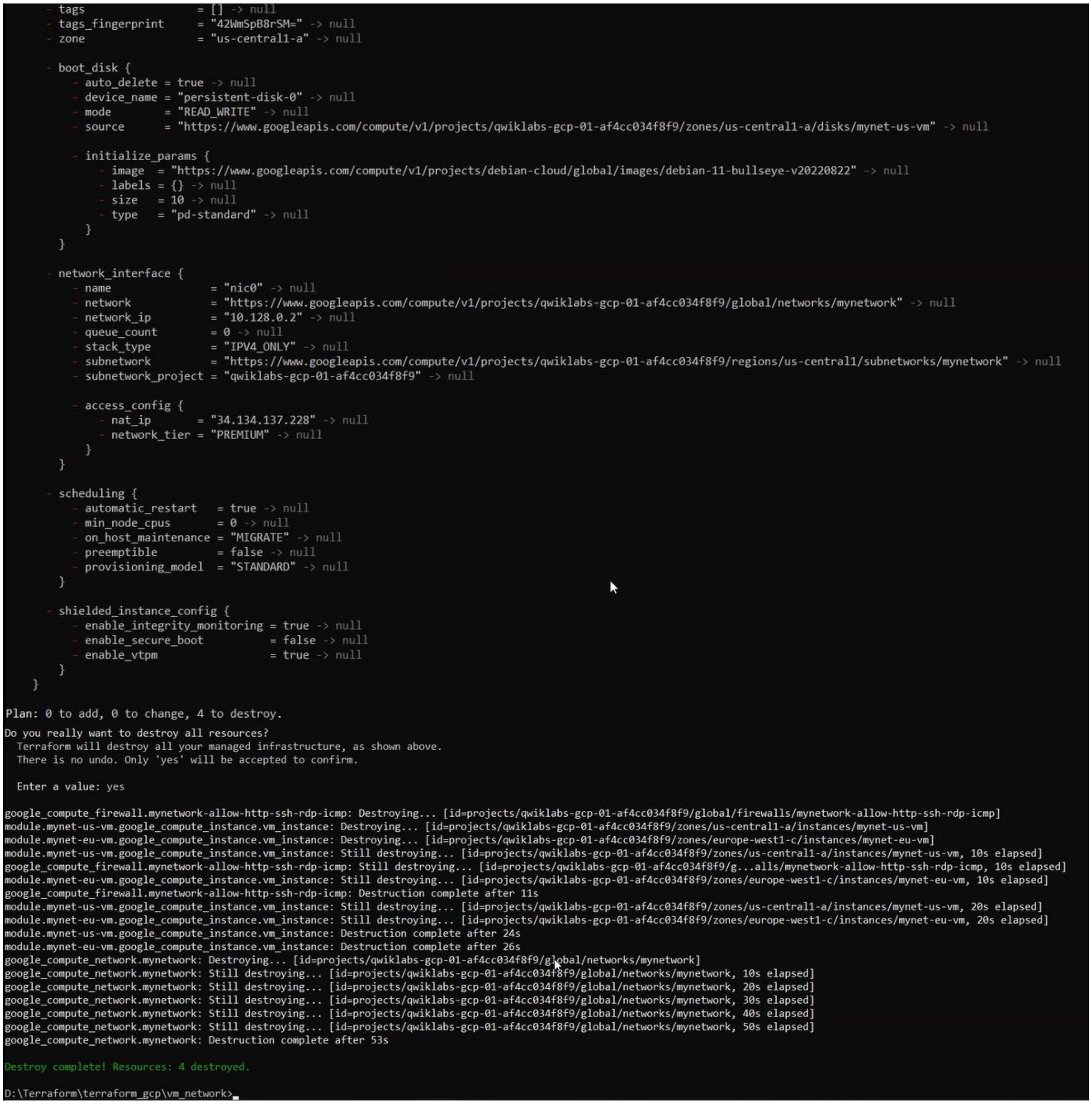
Back up your VMs and set up disaster recovery

**Then let us try to destroy what we created for learning purposes using command below and confirm:**
***terraform destroy***

**sample output:**



```
D:\Terraform\terraform_gcp\vm_network>terraform destroy
google_compute_network.mynetwork: Refreshing state... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork]
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Refreshing state... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/firewalls/mynetwork-allow-http-ssh-rdp-icmp]
module.mynet-us-vm.google_compute_instance.vm_instance: Refreshing state... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/instances/mynet-us-vm]
module.mynet-eu-vm.google_compute_instance.vm_instance: Refreshing state... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/europe-west1-c/instances/mynet-eu-vm]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp will be destroyed
  - resource "google_compute_firewall" "mynetwork-allow-http-ssh-rdp-icmp" {
      - creation_timestamp      = "2022-09-18T04:19:05.378-07:00" -> null
      - destination_ranges      = [] -> null
      - direction               = "INGRESS" -> null
      - disabled                = false -> null
      - id                      = "projects/qwiklabs-gcp-01-af4cc034f8f9/global/firewalls/mynetwork-allow-http-ssh-rdp-icmp" -> null
      - name                    = "mynetwork-allow-http-ssh-rdp-icmp" -> null
      - network                 = "https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork" -> null
      - priority                = 1000 -> null
      - project                 = "qwiklabs-gcp-01-af4cc034f8f9" -> null
      - self_link               = "https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-af4cc034f8f9/global/firewalls/mynetwork-allow-http-ssh-rdp-icmp" -> null
      - source_ranges           = [
          - "0.0.0.0/0",
        ] -> null
      - source_service_accounts = [] -> null
      - source_tags             = [] -> null
      - target_service_accounts = [] -> null
      - target_tags             = [] -> null

      - allow {
          - ports    = [
              - "22",
              - "80",
              - "3389",
            ] -> null
          - protocol = "tcp" -> null
        }
      - allow {
          - ports    = [] -> null
          - protocol = "icmp" -> null
        }
    }
```

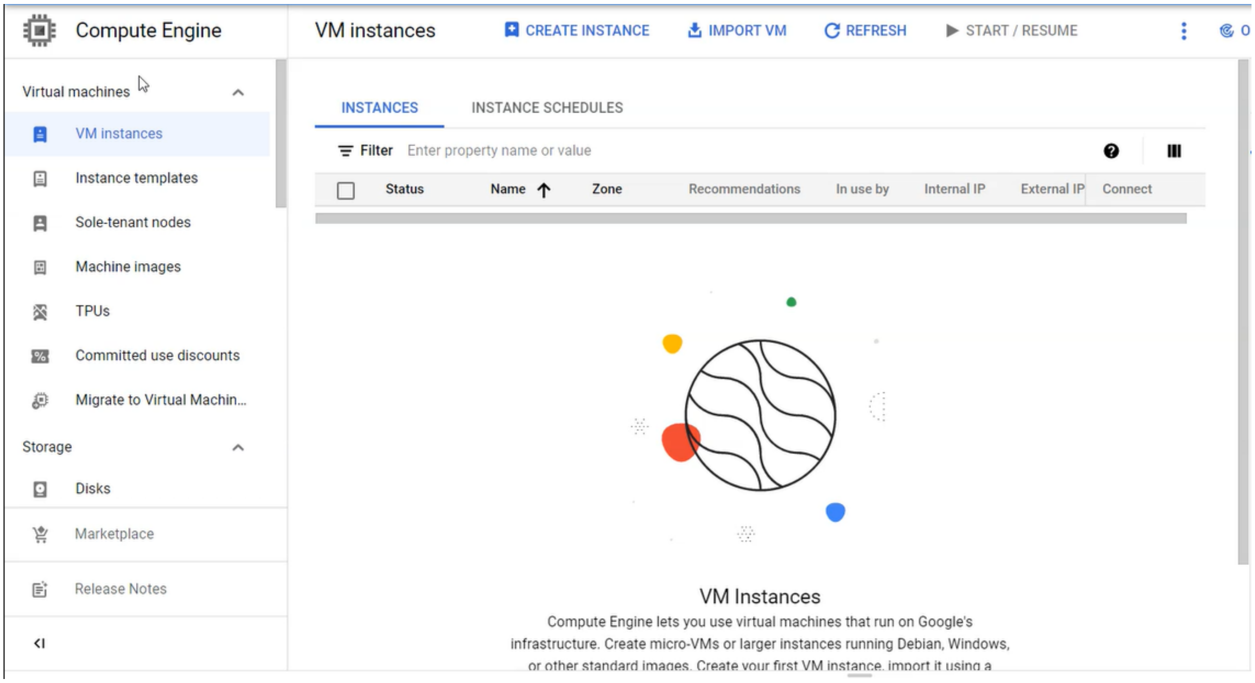


```
      - tags                 = [] -> null
      - tags_fingerprint     = "42WmSpB8rSM=" -> null
      - zone                 = "us-central1-a" -> null

      - boot_disk {
          - auto_delete = true -> null
          - device_name = "persistent-disk-0" -> null
          - mode        = "READ_WRITE" -> null
          - source      = "https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/disks/mynet-us-vm" -> null

          - initialize_params {
              - image  = "https://www.googleapis.com/compute/v1/projects/debian-cloud/global/images/debian-11-bullseye-v20220822" -> null
              - labels = {} -> null
              - size   = 10 -> null
              - type   = "pd-standard" -> null
            }
        }

      - network_interface {
          - name               = "nic0" -> null
          - network            = "https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork" -> null
          - network_ip         = "10.128.0.2" -> null
          - queue_count        = 0 -> null
          - stack_type         = "IPV4_ONLY" -> null
          - subnetwork         = "https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-af4cc034f8f9/regions/us-central1/subnetworks/mynetwork" -> null
          - subnetwork_project = "qwiklabs-gcp-01-af4cc034f8f9" -> null

          - access_config {
              - nat_ip       = "34.134.137.228" -> null
              - network_tier = "PREMIUM" -> null
            }
        }

      - scheduling {
          - automatic_restart   = true -> null
          - min_node_cpus       = 0 -> null
          - on_host_maintenance = "MIGRATE" -> null
          - preemptible         = false -> null
          - provisioning_model  = "STANDARD" -> null
        }

      - shielded_instance_config {
          - enable_integrity_monitoring = true -> null
          - enable_secure_boot          = false -> null
          - enable_vtpm                 = true -> null
        }
    }

Plan: 0 to add, 0 to change, 4 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/firewalls/mynetwork-allow-http-ssh-rdp-icmp]
module.mynet-us-vm.google_compute_instance.vm_instance: Destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/instances/mynet-us-vm]
module.mynet-eu-vm.google_compute_instance.vm_instance: Destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/europe-west1-c/instances/mynet-eu-vm]
module.mynet-us-vm.google_compute_instance.vm_instance: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/instances/mynet-us-vm, 10s elapsed]
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/g...alls/mynetwork-allow-http-ssh-rdp-icmp, 10s elapsed]
module.mynet-eu-vm.google_compute_instance.vm_instance: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/europe-west1-c/instances/mynet-eu-vm, 10s elapsed]
google_compute_firewall.mynetwork-allow-http-ssh-rdp-icmp: Destruction complete after 11s
module.mynet-us-vm.google_compute_instance.vm_instance: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/us-central1-a/instances/mynet-us-vm, 20s elapsed]
module.mynet-eu-vm.google_compute_instance.vm_instance: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/zones/europe-west1-c/instances/mynet-eu-vm, 20s elapsed]
module.mynet-us-vm.google_compute_instance.vm_instance: Destruction complete after 24s
module.mynet-eu-vm.google_compute_instance.vm_instance: Destruction complete after 26s
google_compute_network.mynetwork: Destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork]
google_compute_network.mynetwork: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork, 10s elapsed]
google_compute_network.mynetwork: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork, 20s elapsed]
google_compute_network.mynetwork: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork, 30s elapsed]
google_compute_network.mynetwork: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork, 40s elapsed]
google_compute_network.mynetwork: Still destroying... [id=projects/qwiklabs-gcp-01-af4cc034f8f9/global/networks/mynetwork, 50s elapsed]
google_compute_network.mynetwork: Destruction complete after 53s

Destroy complete! Resources: 4 destroyed.

D:\Terraform\terraform_gcp\vm_network>
```

**Check the destroy state of the VM instances by going to Compute Engine -> VM instances. Notice that the instances are now deleted.**

# Terraform GCP Sample: VM Instances and VPC

**By Ronald Stewart Lim**



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