Course 3_Lab 4: Change firewall rules using Terraform and Cloud Shell

Scenario

Cymbal Bank's new banking application is ready to be deployed. It needs to be hosted on a Virtual Private Cloud (VPC). Your team lead, Chloe, would like to define and provision the application's network infrastructure using Terraform. Terraform can be used to securely provision cloud infrastructure in a way that's repeatable and consistent. This way, you can easily and quickly make any required changes. You have been tasked with using Terraform to create a network and firewall rules.

Here's how you'll do this task: First, you'll activate the Cloud Shell. Then, you'll clone the Terraform repo. Finally, you'll deploy the VPC network and firewall.

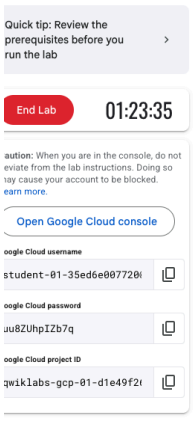
*NOTE: *Google provided all of the commands for Cloud Shell inputs.*

MY WORK:

Task 1: Clone the Terraform repo

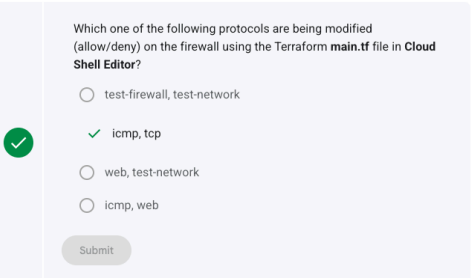
In this task a terraform repository was cloned from github. This file contains the configuration file needed to provision the firewall rules.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-01-d1e49f265c99.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_01_35ed6e007720@cloudshell:~ (qwiklabs-gcp-01-d1e49f265c99)$ cloudshell_open --repo_url "https://github.com/terraform-google-modules/docs-examples.git" --print_file "
./motd" --dir "firewall_basic" --page "editor" --tutorial "./tutorial.md" --open_in_editor "main.tf" --force_new_clone
2025/05/25 21:13:10 Cloning https://github.com/terraform-google-modules/docs-examples.git into /home/student_01_35ed6e007720/cloudshell_open/docs-examples
Cloning into '/home/student_01_35ed6e007720/cloudshell_open/docs-examples'...
remote: Enumerating objects: 7415, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 7415 (delta 0), reused 0 (delta 0), pack-reused 7413 (from 2)
Receiving objects: 100% (7415/7415), 1.95 MiB | 9.78 MiB/s, done.
Resolving deltas: 100% (5684/5684), done.
2025/05/25 21:13:11 ===

These examples use real resources that will be billed to the
Google Cloud Platform project you use - so make sure that you
run "terraform destroy" before quitting!

===
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ ls
backing_file.tf  main.tf  motd  tutorial.md
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ cat main.tf
resource "google_compute_firewall" "default" {
  name    = "test-firewall-${local.name_suffix}"
  network = google_compute_network.default.name

  allow {
    protocol = "icmp"
  }

  allow {
    protocol = "tcp"
    ports    = ["80", "8080", "1000-2000"]
  }

  source_tags = ["web"]
}

resource "google_compute_network" "default" {
  name = "test-network-${local.name_suffix}"
}
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$
```



7. Press **ENTER**.

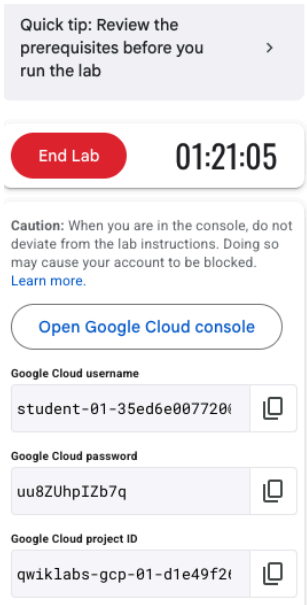
The `main.tf` file is the configuration file that defines the resources that Terraform will create. Two resources will be created: a firewall rule `google_compute_firewall` named `test-firewall-${local.name_suffix}` with rules to allow ICMP and TCP traffic from ports 80, 8080, and 1000-2000 and a VPC network `google_compute_network` named `test-network-${local.name_suffix}`. The variable `${local.name_suffix}` is a local variable that automatically generates unique names for resources.



Task 2: Deploy the VPC network and firewall

In this task, the Terraform script, main.tf, was initialized to create the VPC network and firewall rules.

```
Google Cloud Platform project you use - so make sure that you
run "terraform destroy" before quitting!

===
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ ls
backing_file.tf  main.tf  motd  tutorial.md
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ cat main.tf
resource "google_compute_firewall" "default" {
  name    = "test-firewall-${local.name_suffix}"
  network = google_compute_network.default.name

  allow {
    protocol = "icmp"
  }

  allow {
    protocol = "tcp"
    ports    = ["80", "8080", "1000-2000"]
  }

  source_tags = ["web"]
}

resource "google_compute_network" "default" {
  name = "test-network-${local.name_suffix}"
}
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ export GOOGLE_CLOUD_PROJECT=qwiklabs-gcp-01-d1e49f265c99
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/google...
- Finding latest version of hashicorp/random...
- Installing hashicorp/google v6.36.1...
- Installed hashicorp/google v6.36.1 (signed by HashiCorp)
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (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.
student_01_35ed6e007720@cloudshell:~/cloudshell_open/docs-examples/firewall_basic (qwiklabs-gcp-01-d1e49f265c99)$
```



7. The command prompt will prompt you to **Enter a value**. Type "yes", and press **ENTER**.

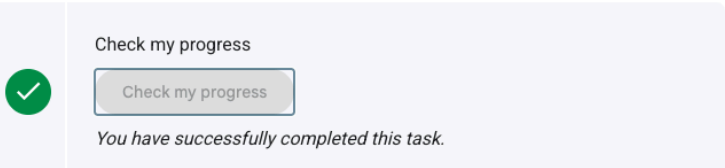
This will start creating the VPC network and firewall rules.

Once it's completed, the output should return the following message:

```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

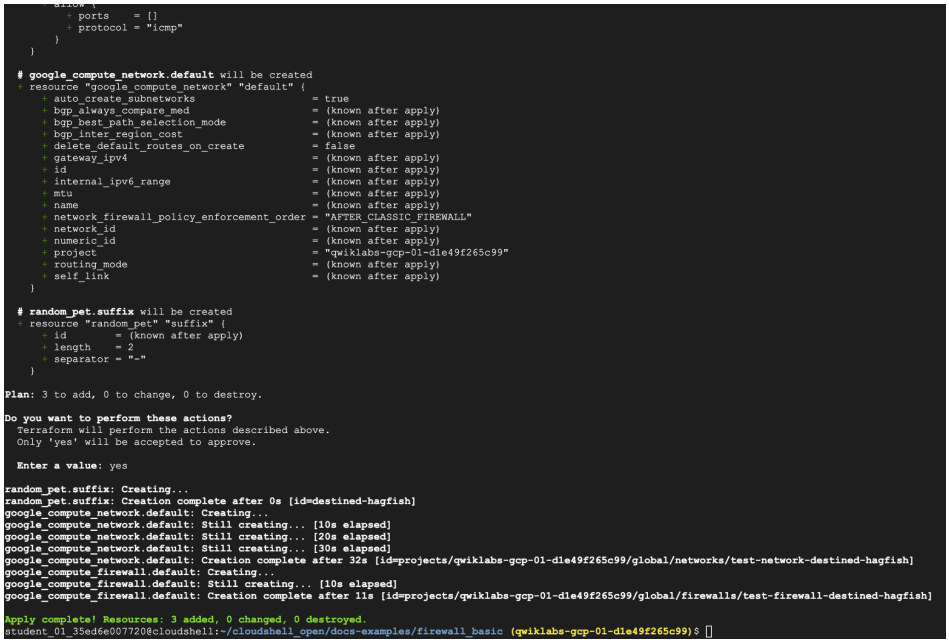
This means that the VPC and firewall have been successfully deployed.

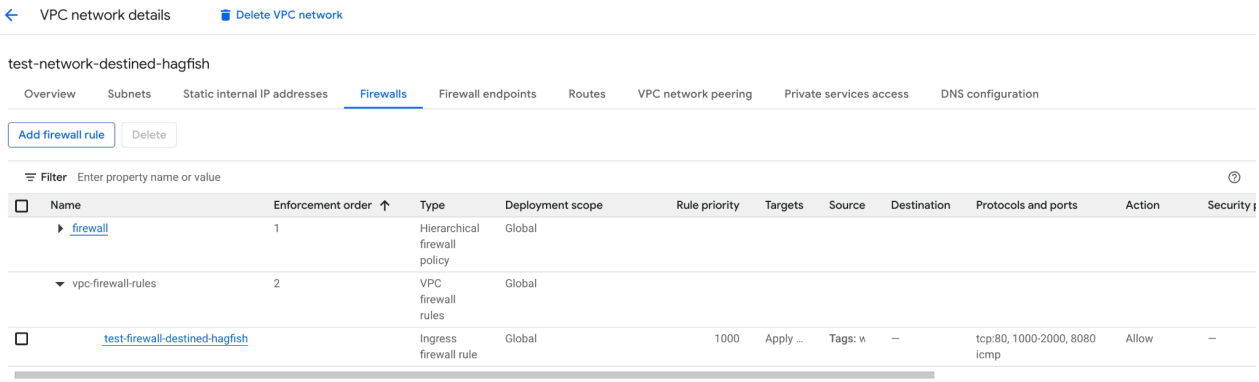
Click **Check my progress** to verify that you have completed this task correctly.



Task 3: Verify the deployment of the resources

In this task I used the VPC network service in the Google Cloud console to verify the VPC networks were created. I was also able to view that firewalls were created by navigating to the Firewalls tab in the Google console.





My Assessment

In this lab, I was able to use an IaC tool to create VPC networks and provision firewall rules to secure the networks. The purpose of provisioning a firewall is to ensure the network is protected from unauthorized access and threats. Terraform is an IaC (infrastructure as code) tool used to automate the creation, management, and deployment of infrastructure resources.

For this lab, I was able to gain some hands-on experience with Terraform as I have not worked with the tool before. This was a short introduction to an IaC tool that enhances cloud development and security and will require more learning and practice in the future to fully understand how to utilize the tool.