Change firewall rules using Terraform and Cloud Shell

Objective:

To deploy and manage firewall rules and a Virtual Private Cloud (VPC) network using **Terraform** in **Google Cloud Platform (GCP)** through **Cloud Shell**, demonstrating Infrastructure as Code (IaC) practices.

Tools & Technologies Used:

- **Google Cloud Platform (GCP)**
- **Cloud Shell**
- **Terraform**
- **GitHub**
- **Google Cloud Console**

Task 1. Clone the Terraform repo

In this task, you'll clone the Terraform example repository using the Cloud Shell terminal. The Terraform example contains the configuration file, which you'll use to provision the firewall rules.

1. In the Google Cloud console, click the **Activate Cloud Shell**
2. Click **Continue**.

It should only take a few moments to provision and connect to the Cloud Shell environment.

3. Copy the following command into the Cloud Shell terminal:

```
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
```

Copied!

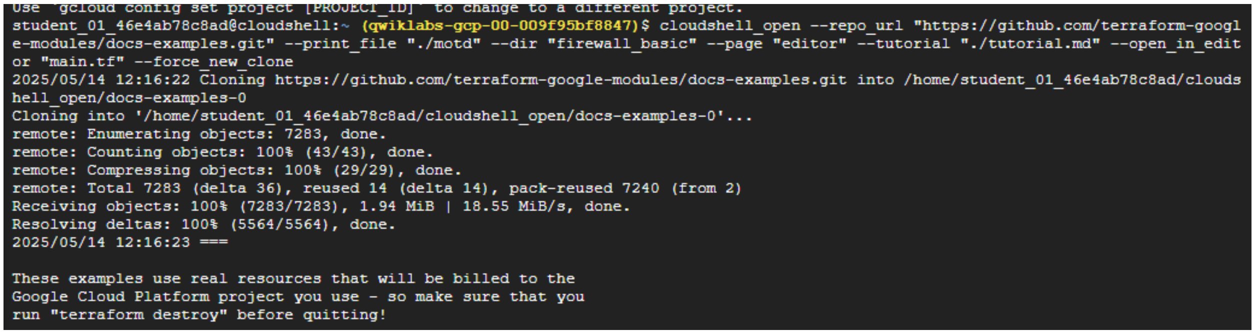
content_copy

This command clones the Terraform example directory.

4. Press **ENTER**.

This command performs the following actions:

- Clones the terraform-google-modules.
- Prints the motd file name.
- Switches to the firewall_basic directory.
- Checks the cloned files, for example tutorial.md.
- Opens main.tf in Cloud Shell Editor.

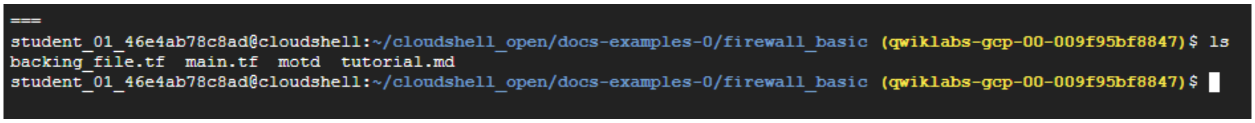


```
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_01_46e4ab78c8ad@cloudshell:~ (qwiklabs-gcp-00-009f95bf8847)$ cloudshell_open --repo_url "https://github.com/terraform-googl
e-modules/docs-examples.git" --print_file "./motd" --dir "firewall_basic" --page "editor" --tutorial "./tutorial.md" --open_in_edit
or "main.tf" --force_new_clone
2025/05/14 12:16:22 Cloning https://github.com/terraform-google-modules/docs-examples.git into /home/student_01_46e4ab78c8ad/clouds
hell_open/docs-examples-0
Cloning into '/home/student_01_46e4ab78c8ad/cloudshell_open/docs-examples-0'...
remote: Enumerating objects: 7283, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 7283 (delta 36), reused 14 (delta 14), pack-reused 7240 (from 2)
Receiving objects: 100% (7283/7283), 1.94 MiB | 18.55 MiB/s, done.
Resolving deltas: 100% (5564/5564), done.
2025/05/14 12:16:23 ===

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!
```

5. Copy the following command into the Cloud Shell terminal to list the contents of the directory:

ls

You should notice that several files in the directory have been downloaded: backing_file.tf, main.tf, motd, and tutorial.md.



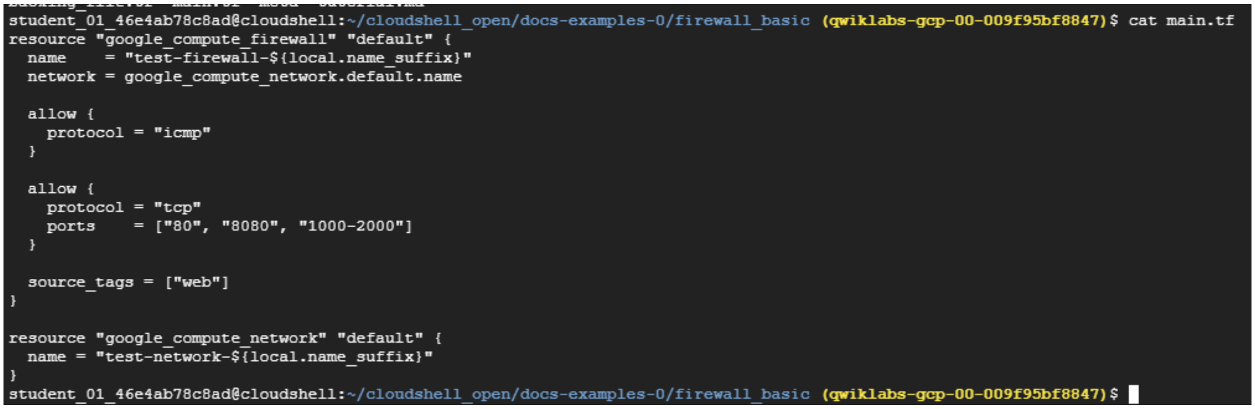
```
===
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ ls
backing_file.tf  main.tf  motd  tutorial.md
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$
```

6. Copy the following command into the Cloud Shell terminal to analyze the configuration of the firewall rule:

cat main.tf

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.



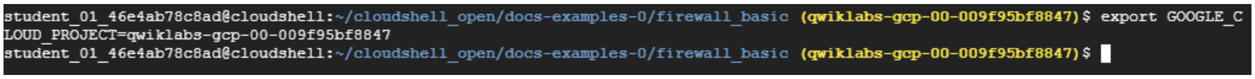
```
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ cat main.tf
resource "google_compute_firewall" "default" {
  name    = "test-firewall-${local.name_suffix}"
  network = google_compute_network.default.name

  allow {
    protocol = "icmp"
  }

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

  source_tags = ["web"]
}

resource "google_compute_network" "default" {
  name = "test-network-${local.name_suffix}"
}
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ █
```

Task 2. Deploy the VPC network and firewall

In this task, you'll deploy a new VPC network and a new firewall rule. This task provides hands-on experience with building a VPC network and subnets.

1. Copy the following command into the Cloud Shell terminal.

export GOOGLE_CLOUD_PROJECT=Project ID



```
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ export GOOGLE_C
LOUD_PROJECT=qwiklabs-gcp-00-009f95bf8847
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ █
```

This command sets the project ID.

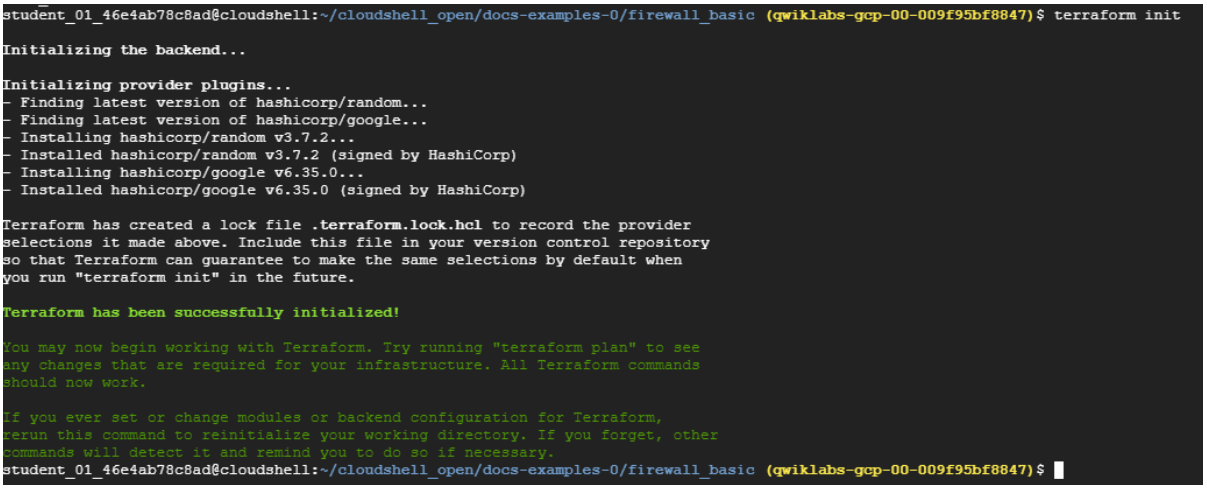
2. Press **ENTER**.

3. Copy the following command into the Cloud Shell terminal:

terraform init

This command initializes the Terraform script.

4. Press **ENTER**.

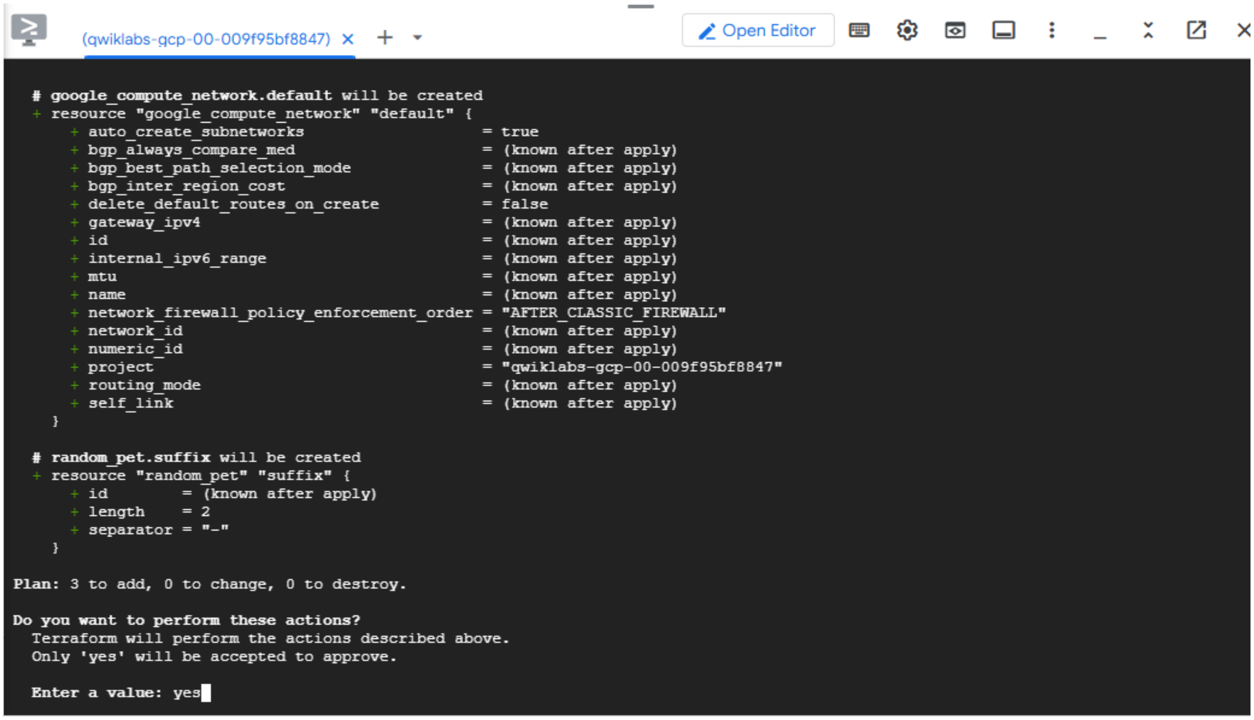


```
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/google...
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (signed by HashiCorp)
- Installing hashicorp/google v6.35.0...
- Installed hashicorp/google v6.35.0 (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.
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$
```

The output should return a message stating that the Terraform has been successfully initialized. Take a moment to examine the output. You'll notice that Terraform will create a new firewall and VPC network

5. Once the initialization is complete, copy the following command into the Cloud Shell terminal:

terraform apply



```
  # google_compute_network.default will be created
  + resource "google_compute_network" "default" {
      + auto_create_subnetworks                   = true
      + bgp_always_compare_med                    = (known after apply)
      + bgp_best_path_selection_mode              = (known after apply)
      + bgp_inter_region_cost                     = (known after apply)
      + delete_default_routes_on_create           = false
      + gateway_ipv4                              = (known after apply)
      + id                                        = (known after apply)
      + internal_ipv6_range                       = (known after apply)
      + mtu                                       = (known after apply)
      + name                                      = (known after apply)
      + network_firewall_policy_enforcement_order = "AFTER_CLASSIC_FIREWALL"
      + network_id                                = (known after apply)
      + numeric_id                                = (known after apply)
      + project                                   = "qwiklabs-gcp-00-009f95bf8847"
      + routing_mode                              = (known after apply)
      + self_link                                 = (known after apply)
    }

  # random_pet.suffix will be created
  + resource "random_pet" "suffix" {
      + id        = (known after apply)
      + length    = 2
      + separator = "-"
    }

Plan: 3 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
```

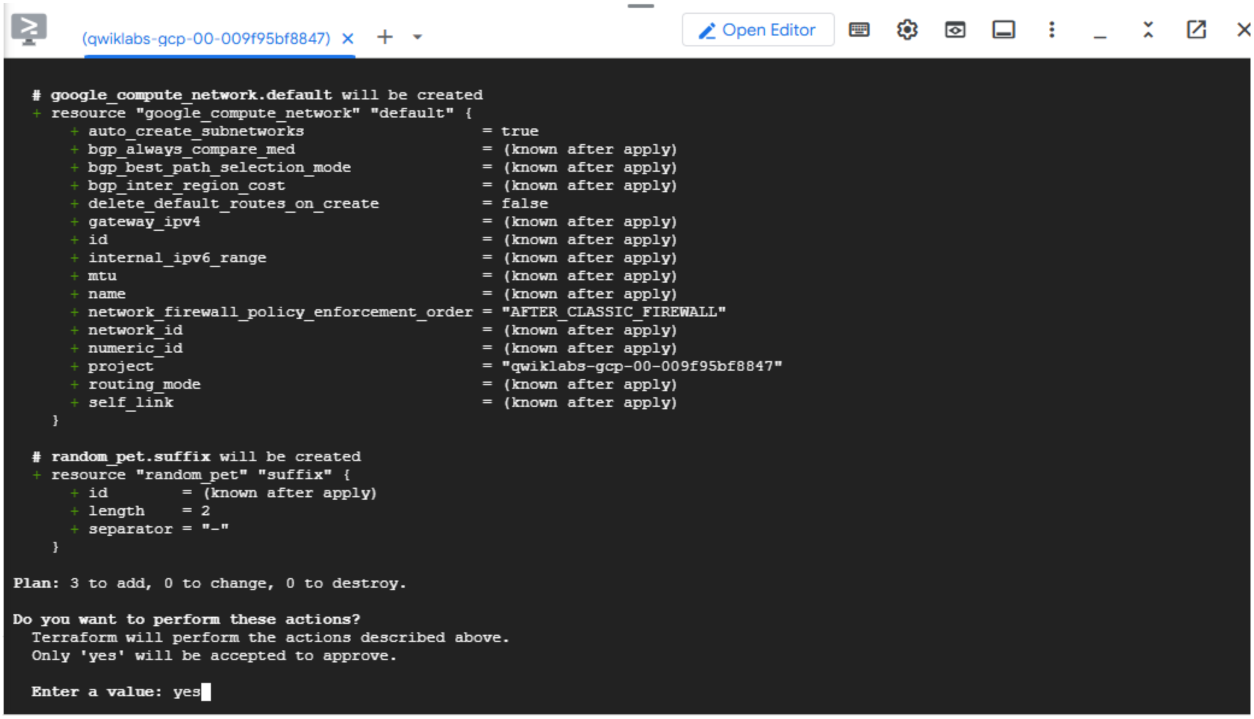
This command applies the changes and deploys the Terraform script.

6. Press **ENTER**.

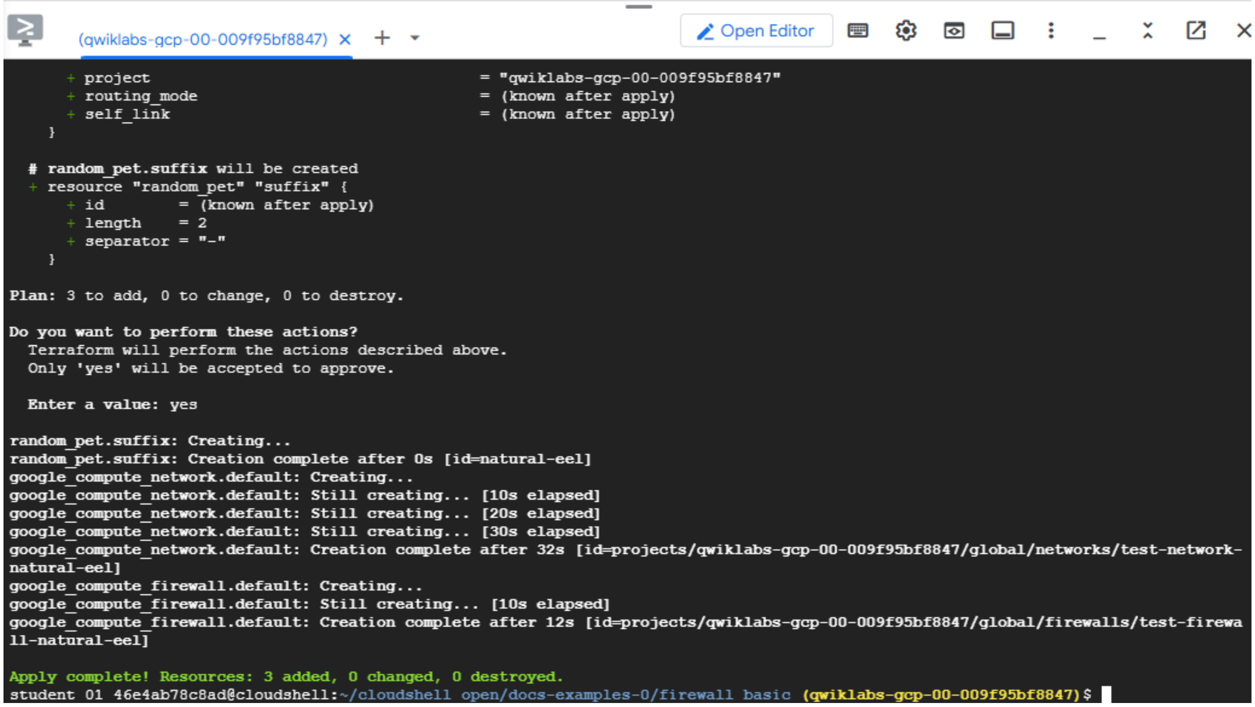
7. The command prompt will prompt you to **Enter a value**. Type "yes", and press **ENTER**.

   This will start creating the VPC network and firewall rules.

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



```
  # google_compute_network.default will be created
  + resource "google_compute_network" "default" {
      + auto_create_subnetworks                   = true
      + bgp_always_compare_med                    = (known after apply)
      + bgp_best_path_selection_mode              = (known after apply)
      + bgp_inter_region_cost                     = (known after apply)
      + delete_default_routes_on_create           = false
      + gateway_ipv4                              = (known after apply)
      + id                                        = (known after apply)
      + internal_ipv6_range                       = (known after apply)
      + mtu                                       = (known after apply)
      + name                                      = (known after apply)
      + network_firewall_policy_enforcement_order = "AFTER_CLASSIC_FIREWALL"
      + network_id                                = (known after apply)
      + numeric_id                                = (known after apply)
      + project                                   = "qwiklabs-gcp-00-009f95bf8847"
      + routing_mode                              = (known after apply)
      + self_link                                 = (known after apply)
    }

  # random_pet.suffix will be created
  + resource "random_pet" "suffix" {
      + id        = (known after apply)
      + length    = 2
      + separator = "-"
    }

Plan: 3 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
```



```
      + project                                 = "qwiklabs-gcp-00-009f95bf8847"
      + routing_mode                            = (known after apply)
      + self_link                               = (known after apply)
    }

  # random_pet.suffix will be created
  + resource "random_pet" "suffix" {
      + id        = (known after apply)
      + length    = 2
      + separator = "-"
    }

Plan: 3 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

random_pet.suffix: Creating...
random_pet.suffix: Creation complete after 0s [id=natural-eel]
google_compute_network.default: Creating...
google_compute_network.default: Still creating... [10s elapsed]
google_compute_network.default: Still creating... [20s elapsed]
google_compute_network.default: Still creating... [30s elapsed]
google_compute_network.default: Creation complete after 32s [id=projects/qwiklabs-gcp-00-009f95bf8847/global/networks/test-network-
natural-eel]
google_compute_firewall.default: Creating...
google_compute_firewall.default: Still creating... [10s elapsed]
google_compute_firewall.default: Creation complete after 12s [id=projects/qwiklabs-gcp-00-009f95bf8847/global/firewalls/test-firewa
ll-natural-eel]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
student_01_46e4ab78c8ad@cloudshell:~/cloudshell_open/docs-examples-0/firewall_basic (qwiklabs-gcp-00-009f95bf8847)$
```

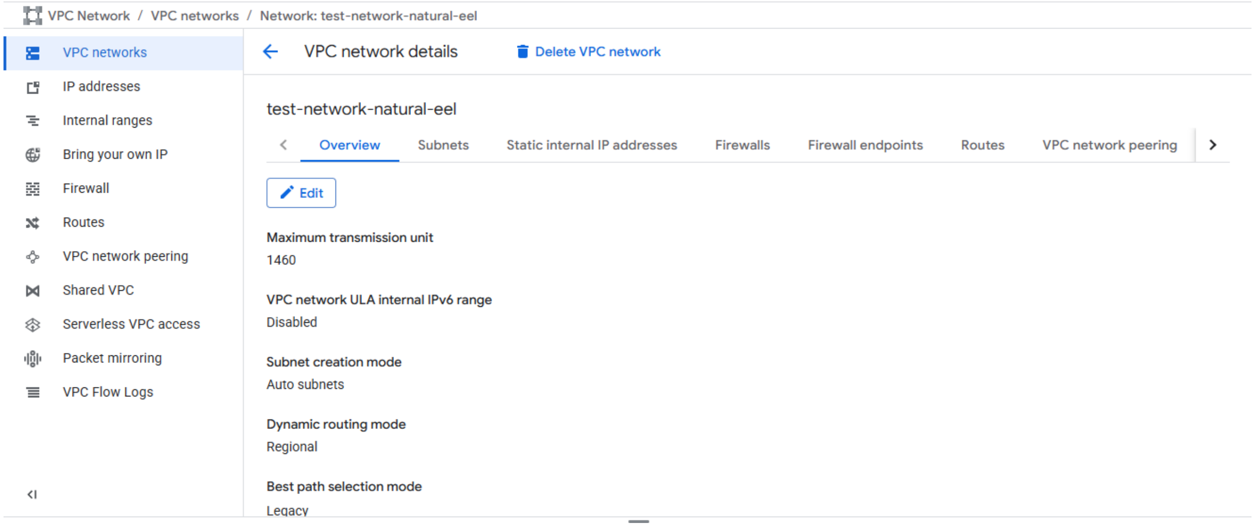
Task 3. Verify the deployment of the resources

In this task, you'll verify that the newly created VPC and firewall rules have been successfully deployed.

1. In the Google Cloud console, from the Navigation menu (), select **VPC network > VPC networks**. The VPC networks page opens.

2. You should notice two VPC networks, **default** and the newest one you just created, **test-network**. Click **test-network** to access the VPC network details.

3. Click **Firewalls**. Use the expand arrow to expand **vpc-firewall-rules**. Under **Protocols and ports** and **Action** you should notice the firewall rules are the same rules as defined in the configuration file: **Allow** and **tcp:80**, **1000-2000**, **8080 icmp**.



test-network-natural-eel

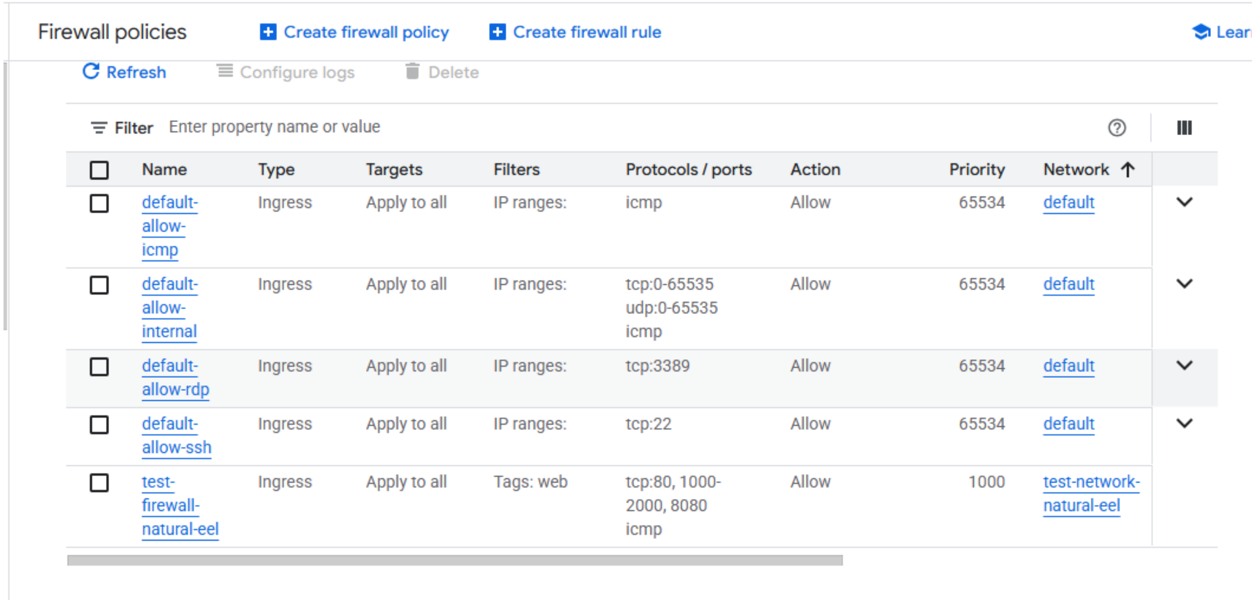
Maximum transmission unit
1460

VPC network ULA internal IPv6 range
Disabled

Subnet creation mode
Auto subnets

Dynamic routing mode
Regional

Best path selection mode
Legacy



□	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑
□	default-allow-icmp	Ingress	Apply to all	IP ranges:	icmp	Allow	65534	default
□	default-allow-internal	Ingress	Apply to all	IP ranges:	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
□	default-allow-rdp	Ingress	Apply to all	IP ranges:	tcp:3389	Allow	65534	default
□	default-allow-ssh	Ingress	Apply to all	IP ranges:	tcp:22	Allow	65534	default
□	test-firewall-natural-eel	Ingress	Apply to all	Tags: web	tcp:80, 1000-2000, 8080 icmp	Allow	1000	test-network-natural-eel