DevOps Lab

CLOUD COMPUTE - GCP

NETWORKING

Home tasks

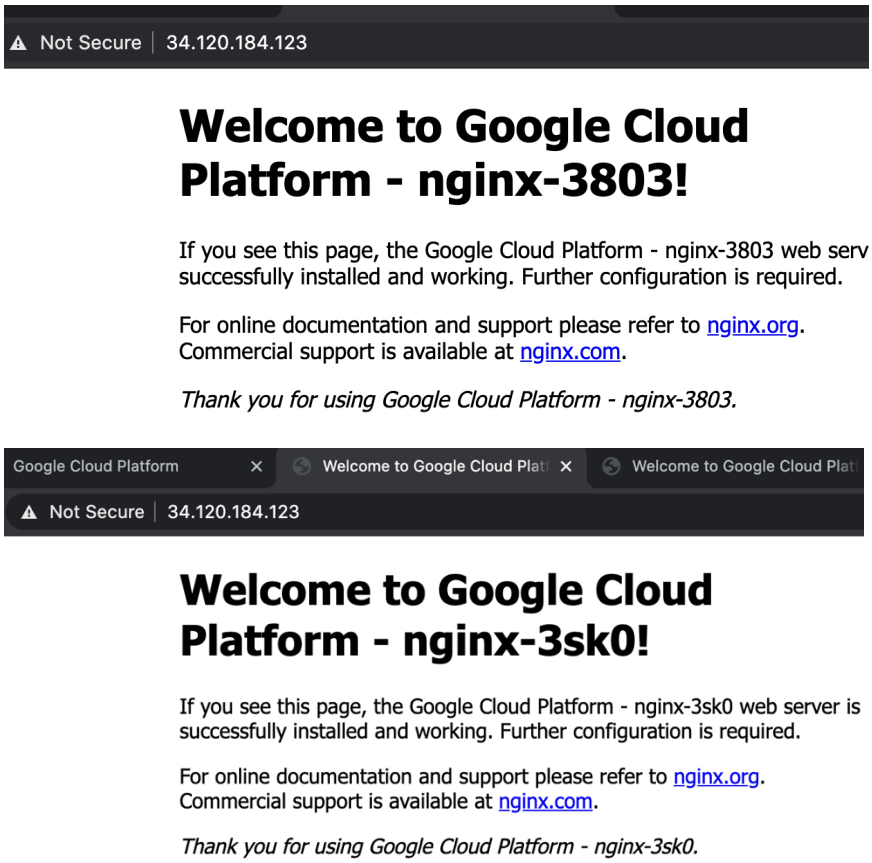
It’s aiming to gain knowledge about Networking in Google Cloud.

TASK 1

Learn about two types of load balancers in Google Cloud Platform:

- a L3 Network Load Balancer and
- a L7 HTTP(s) Load Balancer.

Lab Link: codelabs: LoadBalancers




```
prostovovan@cloudshell:~ (keen-bebop-288318)$ gcloud compute forwarding-rules list
NAME                 REGION  IP_ADDRESS      IP_PROTOCOL  TARGET
http-content-rule            34.120.184.123  TCP          http-lb-proxy
prostovovan@cloudshell:~ (keen-bebop-288318)$ curl http://34.120.184.123 -LI
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 02 Sep 2020 19:27:25 GMT
Content-Type: text/html
Content-Length: 786
Last-Modified: Wed, 02 Sep 2020 18:51:13 GMT
ETag: "5f4fe9a1-312"
Accept-Ranges: bytes
Via: 1.1 google

prostovovan@cloudshell:~ (keen-bebop-288318)$ curl http://34.120.184.123 -LI
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 02 Sep 2020 19:27:34 GMT
Content-Type: text/html
Content-Length: 786
Last-Modified: Wed, 02 Sep 2020 18:51:14 GMT
ETag: "5f4fe9a2-312"
Accept-Ranges: bytes
Via: 1.1 google

System Preferences
```

TASK 3

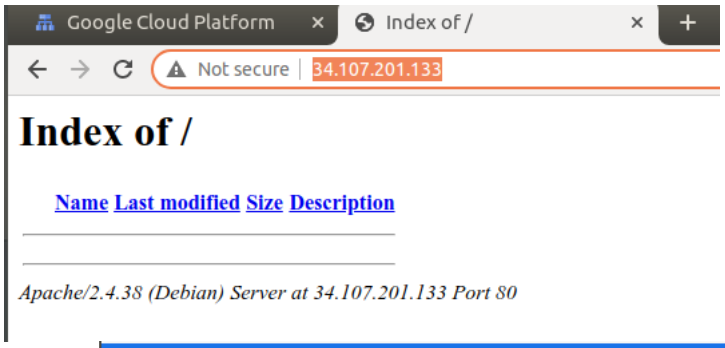
The Objectives are to learn:

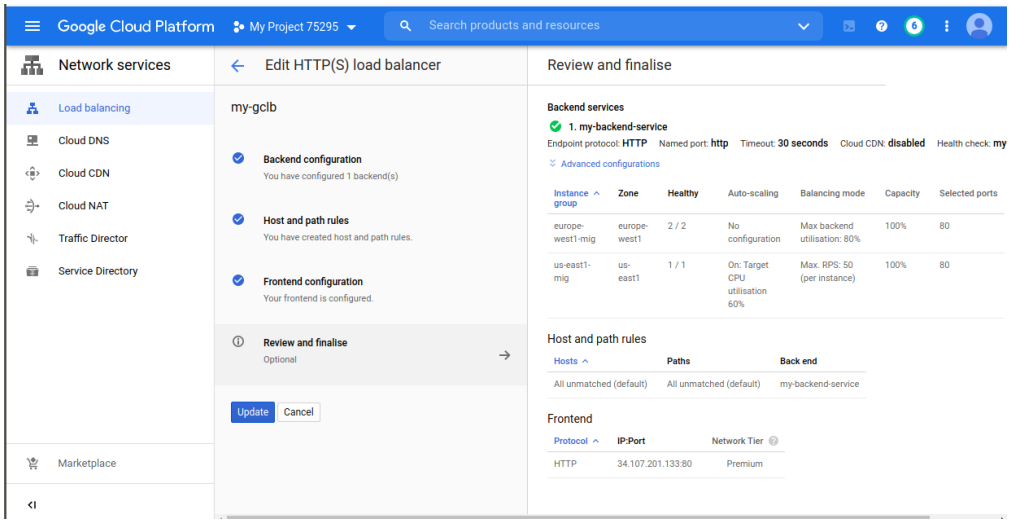
- How to measure latency between Google Compute Engine regions and zones
- How to test network connectivity and performance using open source tools
- How to set up up basic firewalling to secure your networks
- How to set up a global HTTP Load Balancer with Managed Instance Groups to automatically scale your resources up and down based on request load
- How to test and monitor your HTTP Load Balancer setup

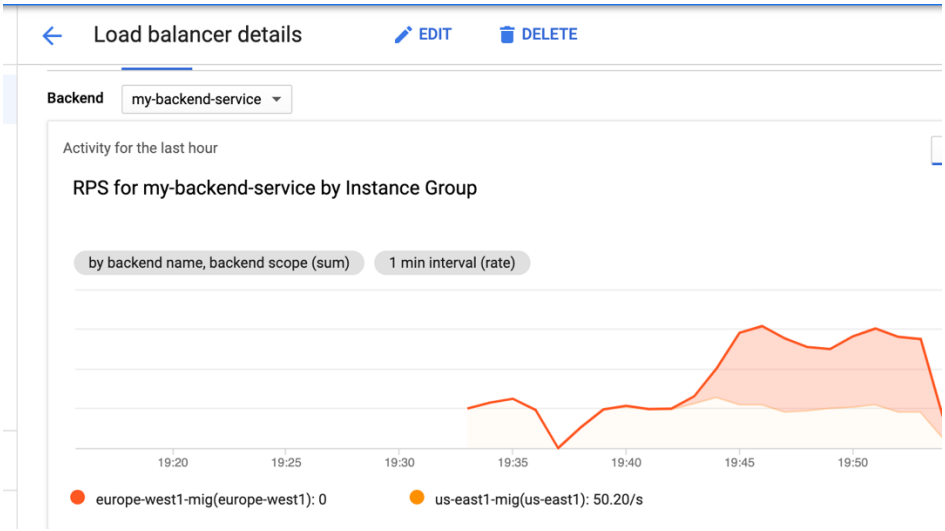
These exercises are ordered to reflect a common cloud developer experience as follows:

1. Set up your lab environment and learn how to work with your GCP environment.
2. Use of common open source tools to explore your network around the world.
3. Deploy a common use case: use of HTTP Load Balancing and Managed Instance Groups to host a scalable, multi-region web server.
4. Testing and monitoring your network and instances.
5. Cleanup.

Lab Link: codelabs: Neworking 101







TASK 3

The Objectives are to learn:

- Setting up NAT gateways
- How to restrict network traffic that certain tiers of an app cannot talk to each other
- Setting up alternate connectivity options to instances
- Map an external service to look like an internal service
- How to setup an Egress proxy limiting access to specific resources

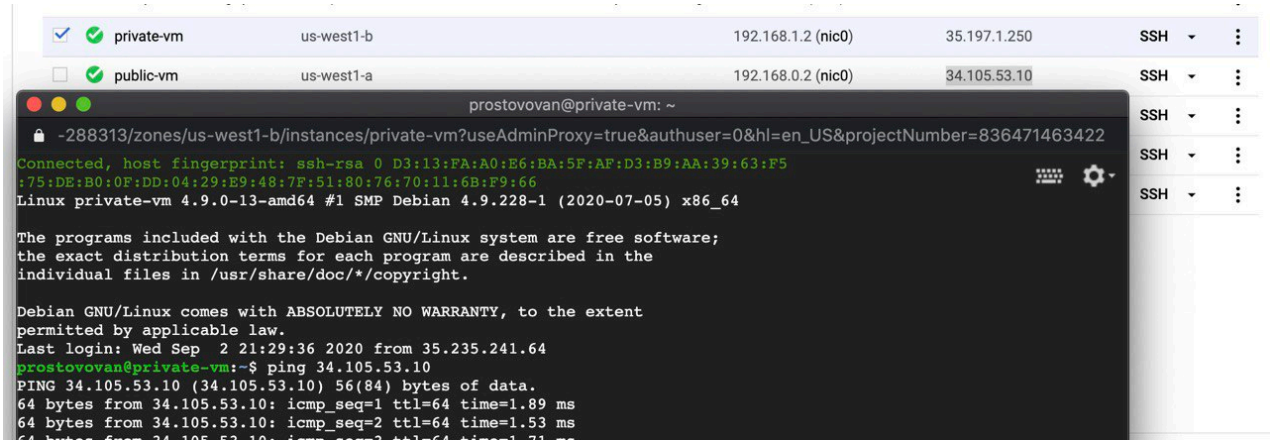
Lab Link: codelabs: Neworking 102

TASK 4

The Objectives are to learn:

- Secure app in custom network

Lab Link: codelabs: custom_network



TASK 5

Create network configuration via terraform.
Resources should be used:

1) **google_compute_network** (to create network)
   https://www.terraform.io/docs/providers/google/r/compute_network.html

   Network name: ${student_name}-vpc

2) **google_compute_firewall**
   (to create rules for external (allow 80,22) /internal access (allow 0-65535))
   https://www.terraform.io/docs/providers/google/r/compute_firewall.html

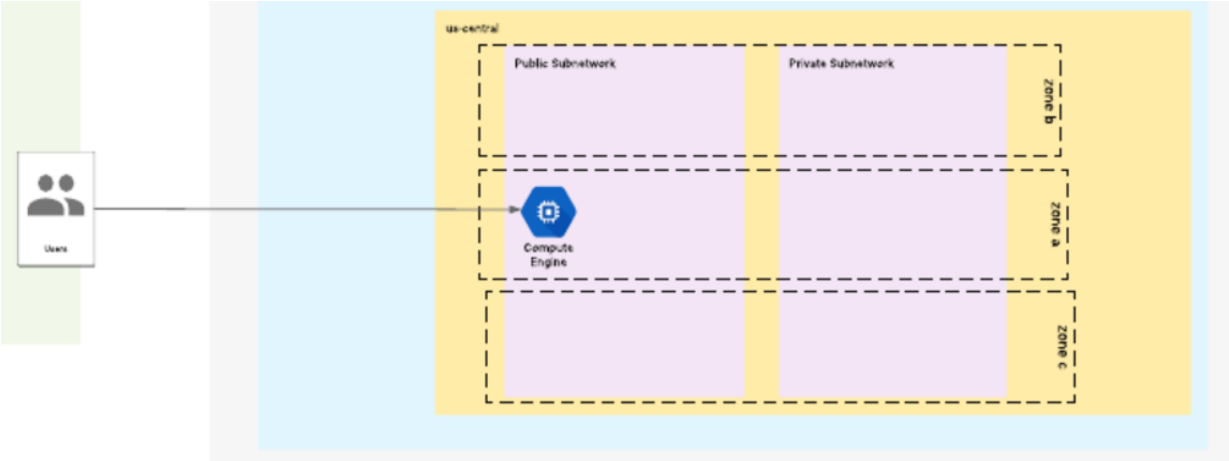
3) **google_compute_subnetwork**
   https://www.terraform.io/docs/providers/google/r/compute_subnetwork.html
   ranges:
   - Public range: 10."${student_IDnum}".1.0/24
   - Private range: 10."${student_IDnum}".2.0/24

4) **google_compute_instance**
   https://www.terraform.io/docs/providers/google/r/compute_instance.html

   1. nginx with default page "Hello from ${student_name}"

All resources should contain description (where it's possible)

Network topology.



All **reports**/code please place into repository:
https://github.com/MNT-Lab/google-cloud-module into appropriate branches: *first char of name + surname*.

For example:
Student: Siarhei Ivanou
Branch Name: **sivanou**

Format depends on case: README.md/scripts/terraform files

Email pattern: [MNT-CD-8.3]-FirstName-LastName

Email should contain the link to personalized branch.