DevOps Lab

CLOUD COMPUTE - GCP

NETWORKING

Home tasks

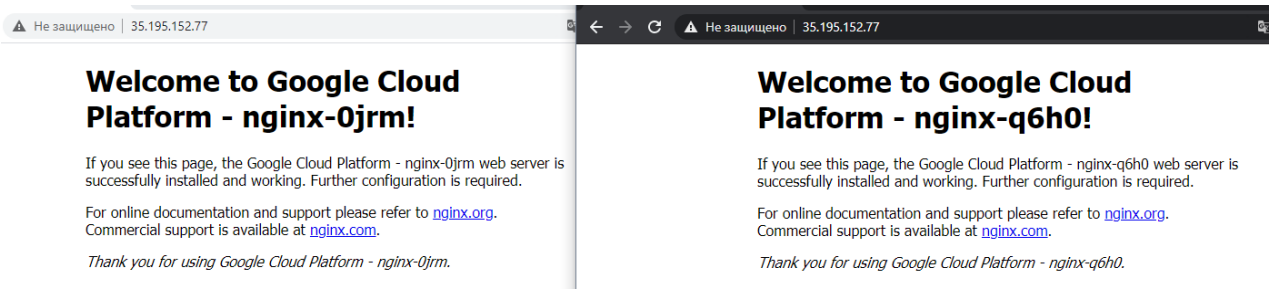
It’s aiming to gain knowledge about Networking in Google Cloud.

TASK 1

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

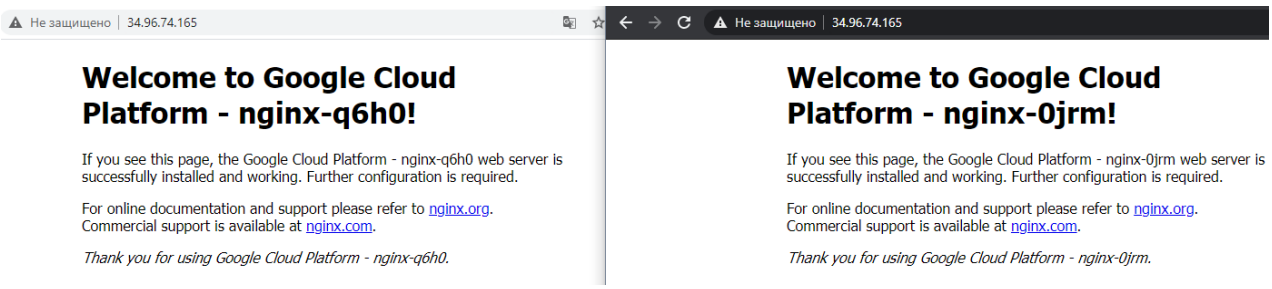
- a L3 Network Load Balancer and

```
valentinratomskiy@cloudshell:~ (amplified-coder-288007)$ gcloud compute forwarding-rules list
NAME      REGION        IP_ADDRESS     IP_PROTOCOL  TARGET
nginx-lb  europe-west1  35.195.152.77  TCP          europe-west1/targetPools/nginx-pool
valentinratomskiy@cloudshell:~ (amplified-coder-288007)$
```



- a L7 HTTP(s) Load Balancer.

```
valentinratomskiy@cloudshell:~ (amplified-coder-288007)$ gcloud compute forwarding-rules list
NAME                REGION        IP_ADDRESS     IP_PROTOCOL  TARGET
http-content-rule                 34.96.74.165   TCP          http-lb-proxy
```



Lab Link: codelabs: LoadBalancers

TASK 2

The Objectives are to learn:

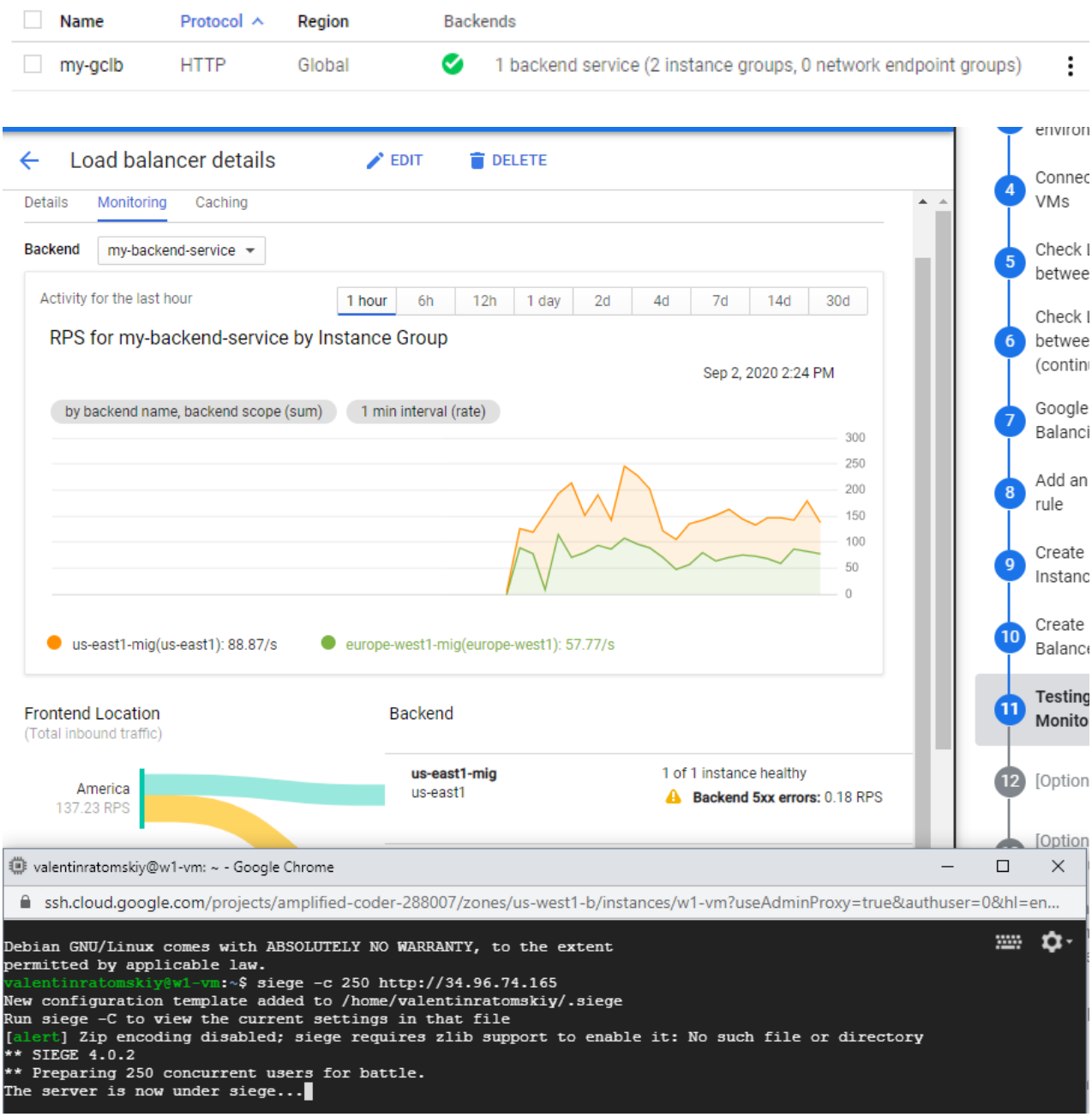
- How to measure latency between Google Compute Engine regions and zones
- How to test network connectivity and performance using open source tools
- How to set up up basic firewalling to secure your networks
- How to set up a global HTTP Load Balancer with Managed Instance Groups to automatically scale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

```
valentinratomskiy@cloudshell:~/networking101 (amplified-coder-288007)$ gcloud compute instance-groups list
NAME              LOCATION       SCOPE   NETWORK        MANAGED  INSTANCES
europe-west1-mig  europe-west1   region  networking101  Yes      3
us-east1-mig      us-east1       region  networking101  Yes      1
valentinratomskiy@cloudshell:~/networking101 (amplified-coder-288007)$
```



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

	faux-on-prem-svc	us-central1-f	10.128.0.9 (nic0)	35.223.19.30	SSH
	nat-gw-eu	europe-west1-c	192.168.20.2 (nic0)	104.155.36.226	SSH
	nat-gw-us	us-central1-f	192.168.10.2 (nic0)	35.192.71.94	SSH
	nat-node-eu	europe-west1-c	192.168.20.3 (nic0)	None	SSH
	nat-node-gcp-eu	europe-west1-c	192.168.20.5 (nic0)	35.240.92.202	SSH
	nat-node-us	us-central1-f	192.168.10.3 (nic0)	None	SSH
	nat-node-w-eu	europe-west1-c	192.168.20.4 (nic0)	None	SSH
	nat-node-w-us	us-central1-f	192.168.10.4 (nic0)	None	SSH

Result after 16th step:

```
valentinratomskiy@nat-node-us:~$ curl nat-gw-us

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
```

Step 19 (default access):

```
[valentinratomskiy@nat-node-gcp-eu ~]$ curl 35.223.19.30

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;
```

```
[valentinratomskiy@nat-node-gcp-eu ~]$ gsutil ls gs://
^CCaught CTRL-C (signal 2) - exiting
[valentinratomskiy@nat-node-gcp-eu ~]$ gcloud compute instances list
^C

Command killed by keyboard interrupt


[valentinratomskiy@nat-node-gcp-eu ~]$ curl -L www.google.com
```

Step 20:

```
[valentinratomskiy@nat-gw-eu ~]$ sudo systemctl status squid
● squid.service - Squid caching proxy
   Loaded: loaded (/usr/lib/systemd/system/squid.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-09-02 16:04:48 UTC; 12min ago
  Process: 2008 ExecStop=/usr/sbin/squid -k shutdown -f $SQUID_CONF (code=exited, status=0/SUCCESS)
  Process: 2015 ExecStart=/usr/sbin/squid $SQUID_OPTS -f $SQUID_CONF (code=exited, status=0/SUCCESS)
  Process: 2010 ExecStartPre=/usr/libexec/squid/cache_swap.sh (code=exited, status=0/SUCCESS)
 Main PID: 2017 (squid)
   CGroup: /system.slice/squid.service
           ├─2017 /usr/sbin/squid -f /etc/squid/squid.conf
           ├─2019 (squid-1) -f /etc/squid/squid.conf
           └─2020 (logfile-daemon) /var/log/squid/access.log
```

```
[valentinratomskiy@nat-node-gcp-eu ~]$ curl -I www.google.com
HTTP/1.1 403 Forbidden
Server: squid/3.5.20
Mime-Version: 1.0
Date: Wed, 02 Sep 2020 16:21:31 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 3522
X-Squid-Error: ERR_ACCESS_DENIED 0
```

```
[valentinratomskiy@nat-node-gcp-eu ~]$ curl 35.223.19.30
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html><head>
<meta type="copyright" content="Copyright (C) 1996-2016 The Squid Software Foundation and contributors">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>ERROR: The requested URL could not be retrieved</title>
<style type="text/css">/*
 * Copyright (C) 1996-2016 The Squid Software Foundation and contributors
 *
 * Squid software is distributed under GPLv2+ license and includes
 * contributions from numerous individuals and organizations.
 * Please see the COPYING and CONTRIBUTORS files for details.
 */

/*
 Stylesheet for Squid Error pages
 Adapted from design by Free CSS Templates
 http://www.freecsstemplates.org
 Released for free under a Creative Commons Attribution 2.5 License
*/
```

```
[valentinratomskiy@nat-node-gcp-eu ~]$ gsutil ls gs://
Traceback (most recent call last):
  File "/usr/lib64/google-cloud-sdk/platform/gsutil/gsutil", line 21, in <module>
    gsutil.RunMain()
  File "/usr/lib64/google-cloud-sdk/platform/gsutil/gsutil.py", line 123, in RunMain
    sys.exit(gslib.__main__.main())
  File "/usr/lib64/google-cloud-sdk/platform/gsutil/gslib/__main__.py", line 438, in main
    user_project=user_project)
  File "/usr/lib64/google-cloud-sdk/platform/gsutil/gslib/__main__.py", line 767, in _RunNamedCommandAndHandleExcep
tions
    _HandleUnknownFailure(e)
  File "/usr/lib64/google-cloud-sdk/platform/gsutil/gslib/__main__.py", line 633, in _RunNamedCommandAndHandleExcep
tions
    user_project=user_project)
```

```
[valentinratomskiy@nat-node-gcp-eu ~]$ gcloud compute instances list
ERROR: gcloud crashed (HTTPError): (403, 'Forbidden')
```

At the last stage of the lab, problems occurred, and it was not possible to make a proxy exception.

TASK 4

The Objectives are to learn:

- Secure app in custom network

Lab Link: [codelabs: custom_network]

Name ↑	Region	Subnets	Mode	IP address ranges	Gateways
▼ custom-net		2	Custom		
	us-west1	private-sub		192.168.1.0/24	192.168.1.1
	us-west1	public-sub		192.168.0.0/24	192.168.0.1

Name	Zone	Recommendation	In use by	Internal IP	External IP
private-vm	us-west1-b			192.168.1.2 (nic0)	34.82.134.235
public-vm	us-west1-a			192.168.0.2 (nic0)	35.199.162.135

Ping public VM:

```
valentinratomskiy@cloudshell:~ (amplified-coder-288007)$ ping 35.199.162.135
PING 35.199.162.135 (35.199.162.135) 56(84) bytes of data.
64 bytes from 35.199.162.135: icmp_seq=1 ttl=52 time=139 ms
64 bytes from 35.199.162.135: icmp_seq=2 ttl=52 time=139 ms
64 bytes from 35.199.162.135: icmp_seq=3 ttl=52 time=139 ms
64 bytes from 35.199.162.135: icmp_seq=4 ttl=52 time=139 ms
```

Ping from private-vm:

```
valentinratomskiy@private-vm:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.029 ms
^C
--- 192.168.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.026/0.027/0.029/0.006 ms
valentinratomskiy@private-vm:~$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=1.90 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.312 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.300 ms
^C
--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.300/0.838/1.902/0.752 ms
valentinratomskiy@private-vm:~$
```

From host to private:

```
valentinratomskiy@cloudshell:~ (amplified-coder-288007)$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
^C
--- 192.168.1.2 ping statistics ---
29 packets transmitted, 0 received, 100% packet loss, time 707ms

valentinratomskiy@cloudshell:~ (amplified-coder-288007)$ ping 34.82.134.235
PING 34.82.134.235 (34.82.134.235) 56(84) bytes of data.
^C
--- 34.82.134.235 ping statistics ---
93 packets transmitted, 0 received, 100% packet loss, time 311ms
```

From public to private:

```
valentinratomskiy@public-vm:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.56 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.295 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.300 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.284 ms
```

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

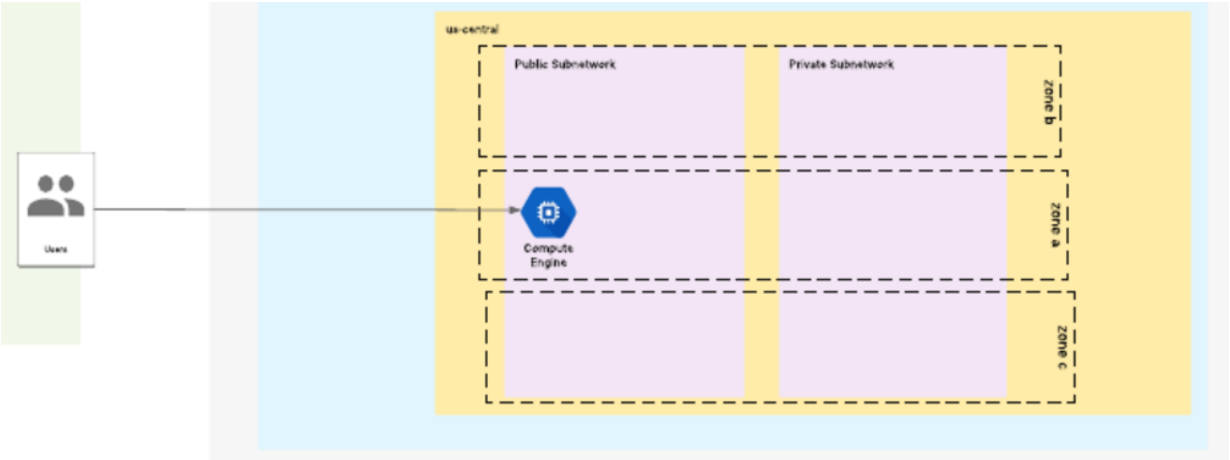
2) **google_compute_firewall**
   (to create rules for external (allow 80,22) /internal access (allow 0-65535))
   https://www.terraform.io/docs/providers/google/r/compute_firewall.html

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-10.3]-FirstName-LastName

Email should contain the link to personalized branch.