

Assignment 8 – Load Balancers

DS 561 – Cloud Computing

Vishwas Bhaktavatsala

U74206902

vishwasb@bu.edu

GitHub Url: <https://github.com/vishwas21/DS561-vishwas-assignments>

Command used to run the `http_client.py`:

(Can be run by installing all the dependencies in the `requirements.txt`)

```
python3 http_client.py -d 34.30.67.192 -p 8085 -b none -i 10000 -n 20 -v -w none
```

Load Balancer:

IP: 34.30.67.192

Port: 8085

Load balancing – Network ser

console.cloud.google.com/net-services/loadbalancing/list/loadBalancers?referrer=search&authuser=2&project=ds...

Personal Boston University Web 3.0 Apps Taxes Income Sources

Google Cloud

DS561 Vishwas Assignment

load

Search

1

VPN

Network services

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

SSL policies

Marketplace

Release Notes

Load balancing

+ CREATE LOAD BALANCER

REFRESH

DELETE

LEARN

LOAD BALANCERS

BACKENDS

FRONTENDS

Filter

Enter property name or value

Name	Load balancer type	Access type	Protocols	Region	Backends
assignment8-lb	Network (Passthrough target-pool)	External	TCP	us-central1	1 target pool (2 instances)

To view or delete load balancing resources like forwarding rules and target proxies, go to the [load balancing components view](#).

Get started with Cloud Load Balancing

Cloud Load Balancing overview

Help document

Overview of the different types of load balancers available in Google Cloud.

Choose a load balancer

Help document

Determine which load balancer best serves your needs.

Set up a global external Application Load Balancer

Tutorial

Set up a global external Application Load Balancer with VM instance group backends.

Set up an external Application Load Balancer by using Ingress

Tutorial

20 min

Use an Ingress resource in Kubernetes to create an external Application Load Balancer that routes traffic to your application.

Serve static content from Cloud Storage buckets

Tutorial

15 min

Serve static content from multiple Cloud Storage buckets with an external Application Load Balancer.

Set up an internal passthrough Network Load Balancer

https://console.cloud.google.com/net-services/loadbalancing/details/network/us-central1/assignment8-lb?project=ds561-vishb-assignment&authuser=2&supportedpurview=project

assignment8-lb

Target-pool Network Load Balancer

Frontend

Protocol ↑	IP version	IP:Port	Network Tier ?
TCP	IPv4	34.30.67.192:8085	Premium

Backend

Name	Region	Health check
assignment8-lb	us-central1	hc-http-8085

ADVANCED CONFIGURATIONS

Instance ↑	Zone
assignment4-http-startup-vm-1	us-central1-b
assignment4-http-startup-vm-2	us-central1-a

VM Instance:

VM instances									
<div><div><div></div><div>Filter</div></div><div>Enter property name or value</div><div><div></div><div></div></div></div>									
<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect	
<input type="checkbox"/>	✓	assignment4-http-client	us-central1-a	<div><div></div><div>Save \$6 / mo</div></div>		10.128.0.6 (nic0)	34.41.239.203 ↗ (nic0)	SSH <div></div>	<div></div>
<input type="checkbox"/>	✓	assignment4-http-startup-vm-1	us-central1-b		assignme... <div></div>	10.128.0.13 (nic0)	35.193.11.248 ↗ (nic0)	SSH <div></div>	<div></div>
<input type="checkbox"/>	✓	assignment4-http-startup-vm-2	us-central1-a		assignme... <div></div>	10.128.0.14 (nic0)	34.27.2.159 ↗ (nic0)	SSH <div></div>	<div></div>

Client Run for 100 requests:

```
200 OK
us-central1-b
Requesting /7909.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /1957.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /6543.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /165.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /9538.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /7283.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /237.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /7690.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /9899.html from 34.30.67.192 8085
400 BAD REQUEST
us-central1-a
Requesting /5569.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /63.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /2720.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /7309.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /3290.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /7264.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /756.html from 34.30.67.192 8085
200 OK
us-central1-a
Requesting /7094.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /6968.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /2464.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /7026.html from 34.30.67.192 8085
200 OK
us-central1-b
Requesting /6442.html from 34.30.67.192 8085
200 OK
us-central1-b
Response Statistics: {'us-central1-a': 54, 'us-central1-b': 46}
vishwas@assignment4-http-client: $
```

Inference:

1. Load balancer was able to figure out that one of the VMs have stopped working and rerouted the traffic within a matter of 1 to 2 seconds.
2. Load balancer was a little slow in figuring out that the VM is back up and running, can be attributed to the VM startup time and the startup script execution time. This time it took around 15 sec.
3. The Ratio of requests served by each VM is almost the same. When tested with a small number of requests, it was around 50% and when tested with a higher number like 100 requests according to the screen shot 54 requests were handled by one VM while the rest 46 were handled by the other.