

DS561-HW8

U43188754

Github Link:

<https://github.com/tigeryi1998/ds561-tigeryi/tree/main/hw8>

```
git clone git@github.com:tigeryi1998/ds561-tigeryi.git
```

Create 2 VMs

← Create an instance EQUIVALENT CODE

To create a VM instance, select one of the options:

- New VM instance**
Create a single VM instance from scratch
- New VM instance from template
Create a single VM instance from an existing template
- New VM instance from machine image
Create a single VM instance from an existing machine image
- Marketplace
Deploy a ready-to-go solution onto a VM instance

Name *
hw8-vm1

MANAGE TAGS AND LABELS

Region *
us-east1 (South Carolina)
Region is permanent

Zone *
us-east1-c
Zone is permanent

Monthly estimate
\$7.11
That's about \$0.01 hourly
Pay for what you use: no upfront costs and per second billing

Item Monthly estimate

2 vCPU + 1 GB memory	\$6.11
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Total	\$7.11

Machine configuration

☒ General purpose ☐ Compute optimized ☐ Memory optimized ☐ GPUs

Machine types for common workloads, optimized for cost and flexibility

Series	Description	vCPUs	Memory	Platf
<input type="radio"/> C3	Consistently high performance	4 - 176	8 - 1,408 GB	Intel
<input type="radio"/> C3D	Consistently high performance	4 - 360	8 - 2,880 GB	AMD
<input checked="" type="radio"/> E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Base
<input type="radio"/> N2	Balanced price & performance	2 - 128	2 - 864 GB	Intel
<input type="radio"/> N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD
<input type="radio"/> T2A	Scale-out workloads	1 - 48	4 - 192 GB	Amp
<input type="radio"/> T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD
<input type="radio"/> N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel

Machine type

Create VM hw8-vm1 in the zone us-east1-c

Network Tag:
network-hw8

Startup script:

```
#!/bin/bash
sudo apt-get update
sudo apt-get install apache2 -y
sudo service apache2 restart
```

```
echo '<!doctype html><html><body><h1>www1</h1></body></html>' | tee  
/var/www/html/index.html
```

Create VM hw8-vm2 in the zone us-east1-d

Network Tag:
network-hw8

Startup script:

```
#!/bin/bash  
sudo apt-get update  
sudo apt-get install apache2 -y  
sudo service apache2 restart  
echo '<!doctype html><html><body><h1>ww2</h1></body></html>' | tee  
/var/www/html/index.html
```

Add Firewall Rule

firewall-hw8, target tag: network-hw8

[←](#) Create a firewall rule

Action on match ?

☒ Allow

☐ Deny

Targets

Specified target tags

▼ ?

Target tags *

network-hw8 ✕

Source filter

IPv4 ranges

▼ ?

Source IPv4 ranges *

0.0.0.0/0 ✕

?

Second source filter

None

▼ ?

Destination filter

None

▼ ?

Protocols and ports ?

☐ Allow all

☒ Specified protocols and ports

☒ TCP

Ports

80

E.g. 20, 50-60

Network Load Balancer

Create the Load Balancer for VM1 and VM2

Health Check is healthy for both VM1 and VM2

The screenshot shows the 'Load balancer details' page for a 'Target-pool Network Load Balancer' named 'hw8-lb'. The left sidebar lists network services, with 'Load balancing' selected. The main content area shows the frontend configuration (TCP, IPv4, IP:Port 104.196.34.137:80, Premium tier) and the backend configuration. The backend table lists two instances, 'hw8-vm1' and 'hw8-vm2', both in the 'us-east1' region with a 'basic-check' health check, and both showing a green checkmark for health status.

Protocol	IP version	IP:Port	Network Tier
TCP	IPv4	104.196.34.137:80	Premium

Name	Region	Health check
hw8-lb	us-east1	basic-check

Instance	Zone	104.196.34.137
hw8-vm1	us-east1-c	✓
hw8-vm2	us-east1-d	✓

Response Header

On VM, do a curl command to get the zone of the current VM:

```
curl -s "http://metadata.google.internal/computeMetadata/v1/instance/zone" -H "Metadata-Flavor: Google"
projects/946005535036/zones/us-east1-c
```

The screenshot shows a Cloud Shell terminal window with the command `curl -s "http://metadata.google.internal/computeMetadata/v1/instance/zone" -H "Metadata-Flavor: Google"` being executed. The output of the command is `projects/946005535036/zones/us-east1-c`.

```
tigeryi@hw8-vm1:~$ curl -s "http://metadata.google.internal/computeMetadata/v1/instance/zone" -H "Metadata-Flavor: Google"
projects/946005535036/zones/us-east1-c
tigeryi@hw8-vm1:~$
```

In the HTTP Web Server VM to add the following line to get the Zone of VM:

VM1 in zone: us-east1-c

VM2 in zone: us-east1-d

```
import requests

GCP_ZONE = requests.get(
    "http://metadata/computeMetadata/v1/instance/zone",
    headers={'Metadata-Flavor': 'Google'}).text

# hw8-vm1
GCP_ZONE = "projects/946005535036/zones/us-east1-c"

# hw8-vm2
GCP_ZONE = "projects/946005535036/zones/us-east1-d"

# response header
self.zone = GCP_ZONE
self.wfile.write(bytes("Response header ", "utf-8"))
self.wfile.write(bytes('zone: {}\n'.format(self.zone), "utf-8"))
```

In the terminal, run the following code to start the web server on VM1

The HTTP GET method will generate a 200 OK response.

-z tag from arg parser is to distinguish between zones of VM1 and VM2

```
$python3 http-server+sql.py -l -z projects/946005535036/zones/us-east1-c
"GET /files/5010.html HTTP/1.1" 200 -
```

On the VM2, also start the web server by running the following code in the terminal.

```
$python3 http-server+sql.py -l -z projects/946005535036/zones/us-east1-d
"GET /files/5010.html HTTP/1.1" 200 -
```

Client Extract Response Header

```
conn.request("GET", filename, headers=headers)

res = conn.getresponse()
data = res.read()

if verbose:
    print(res.status, res.reason)
    print(res.msg)
    print(data)
```

Response header

projects/946005535036/zones/us-east1-c

In the terminal, run the following code to start the client requesting html file from VM1
So I've shown that the client can extract and print response header from the web server.
I call the "-v" in the arg parser to explicitly show the verbose response from the web server.

```
$python3 http-client.py -d localhost -b 'none' -w files -i 10000 -n 1 -p 8080 -r 0 -v
```

```
Requesting /files/5010.html
```

```
200 OK
```

```
Server: BaseHTTP/0.6 Python/3.8.8
```

```
Date: Tue, 21 Nov 2023 03:01:25 GMT
```

```
Content-type: text/html
```

```
...
```

```
Response header zone: projects/946005535036/zones/us-east1-c
```

```
...
```

Kill the web server on VM1


VM instances									
CREATE INSTANCE IMPORT VM REFRESH LEARN									
INSTANCES OBSERVABILITY INSTANCE SCHEDULES									
VM instances									
Filter Enter property name or value									
<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect	
<input type="checkbox"/>	⦿	hw8-vm1	us-east1-c		hw8-lb	10.142.0.6 (nic0)		SSH	⋮
<input type="checkbox"/>	✔	hw8-vm2	us-east1-d		hw8-lb	10.142.0.7 (nic0)	34.139.251.162 ↗ (nic0)	SSH	⋮

It takes about 15-20s for the load balancer to reroute the client request from web server VM1 all to VM2 to serve the GET requests. The only verbose response from the client is in the zone d

...

Response header zone: projects/946005535036/zones/us-east1-d

Restart the web server on VM1



Compute Engine

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

LEARN

INSTANCES

OBSERVABILITY

INSTANCE SCHEDULES

VM instances

Filter

Enter property name or value

?

⌵

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✔	hw8-vm1	us-east1-c		hw8-lb	10.142.0.6 (nic0)	34.139.251.162 ↗ (nic0)	SSH ▾ ⋮
<input type="checkbox"/>	✔	hw8-vm2	us-east1-d		hw8-lb	10.142.0.7 (nic0)	34.139.119.178 ↗ (nic0)	SSH ▾ ⋮

Restart VM1, load balancer takes 15-20s to notice VM1 is back and some of the requests are routed back to VM1, so the response header from VM1 is again shown.

...

Response header zone: projects/946005535036/zones/us-east1-c