I created a network and application load balancers and distributed traffic among my vm instances. I started by

- Creating multiple **web server VM instances**
- Setting up a **Network Load Balancer**
- Setting up an **Application Load Balancer**
- And Tested how traffic is distributed among the 3 VM instances.

Step 1: I set the default Region and Zone using the commands

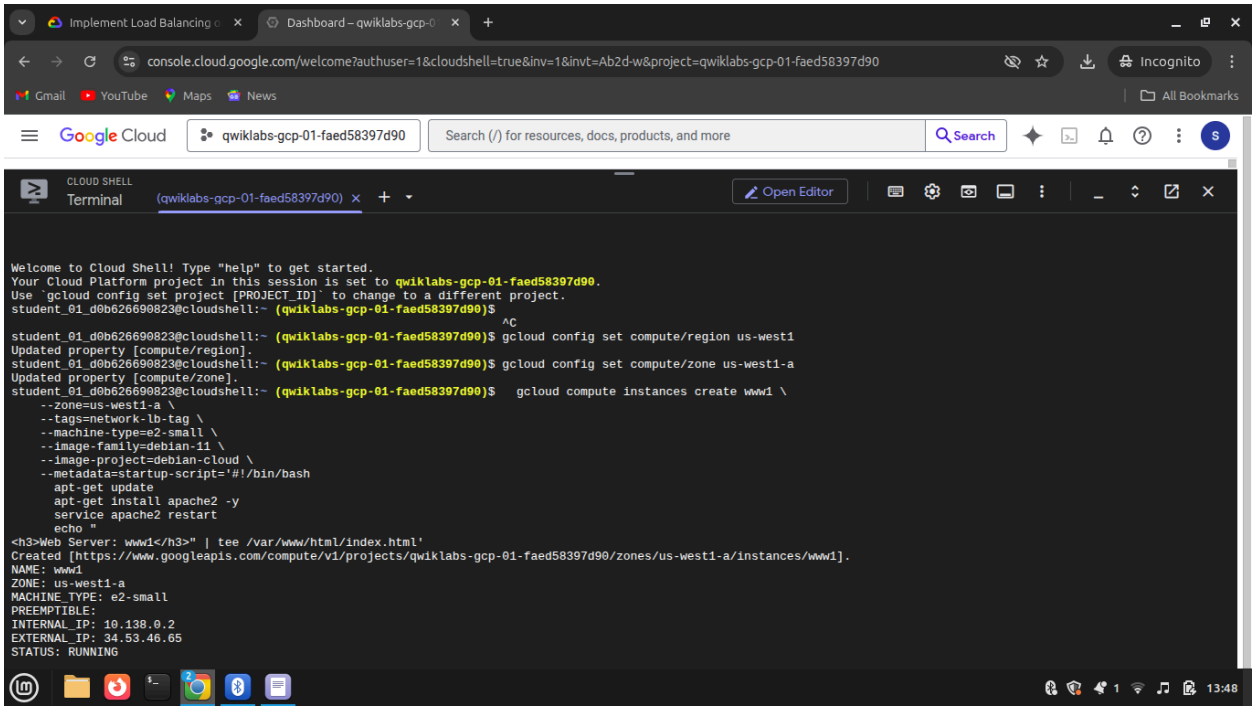
gcloud config set compute/region us-west1
gcloud config set compute/zone us-west1-a

Step 2: I created Multiple Web Server VMs

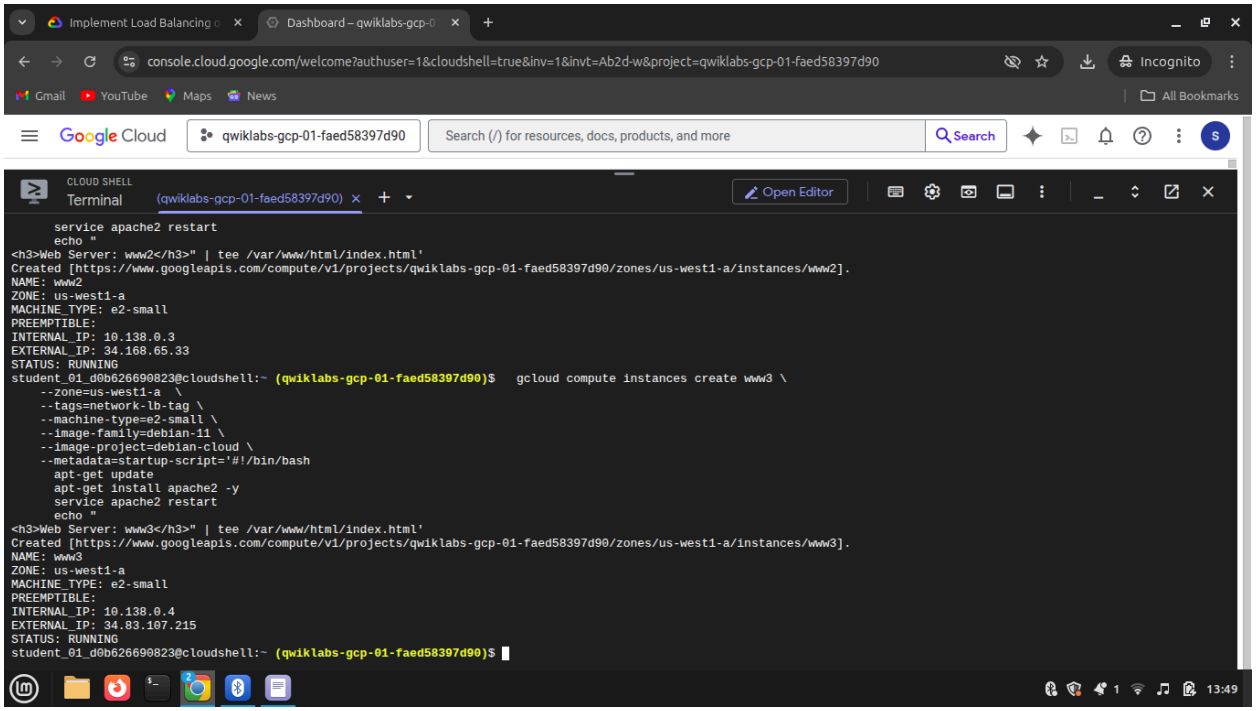
I created 3 VMs in my cloudshell named `www1`, `www2`, `www3`. I used the command below to create the 3 vms with apache installed. The `startup-script` in the command installs Apache and sets the web page text to "Web Server: www1".
The `--tags=network-lb-tag` would be used later to apply firewall rules to all 3 VMs.

```
gcloud compute instances create www1 \

  --zone=us-west1-a \
  --tags=network-lb-tag \
  --machine-type=e2-small \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    service apache2 restart
    echo "<h3>Web Server: www1</h3>" > /var/www/html/index.html'
```

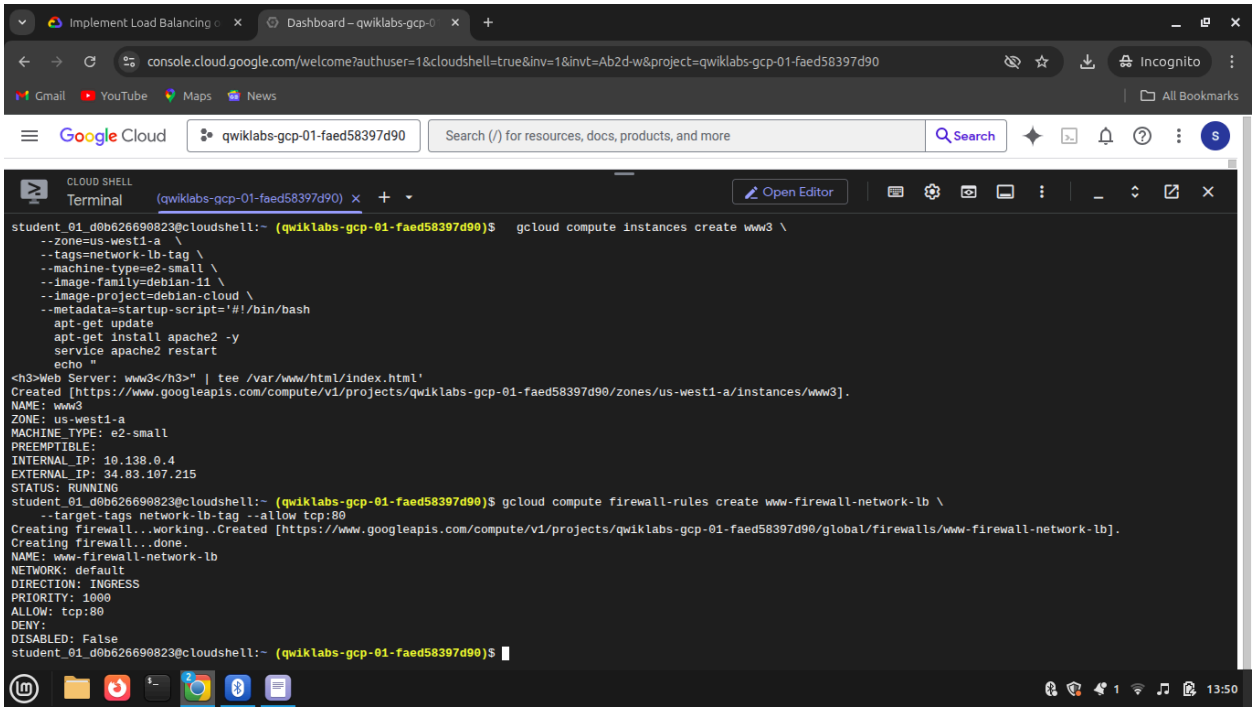


I repeated this same command for the other 2 vms that i created www2 and www3, changing the echo line accordingly.



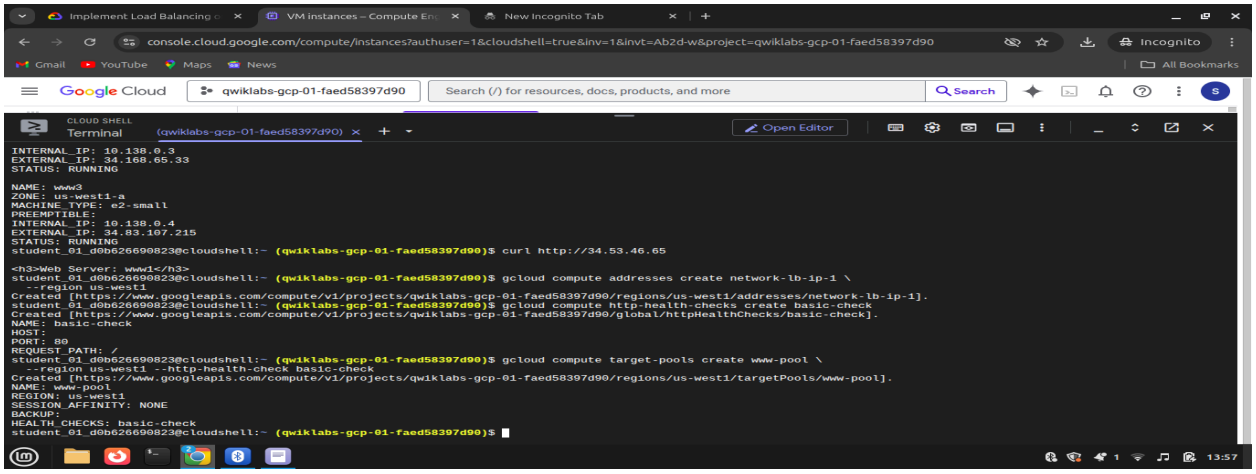
Next, i Created a Firewall Rule for HTTP Access. This allows users on the internet to connect to port 80 (HTTP) of the VMs. I used the command

**gcloud compute firewall-rules create www-firewall-network-lb **
** --target-tags network-lb-tag --allow tcp:80**

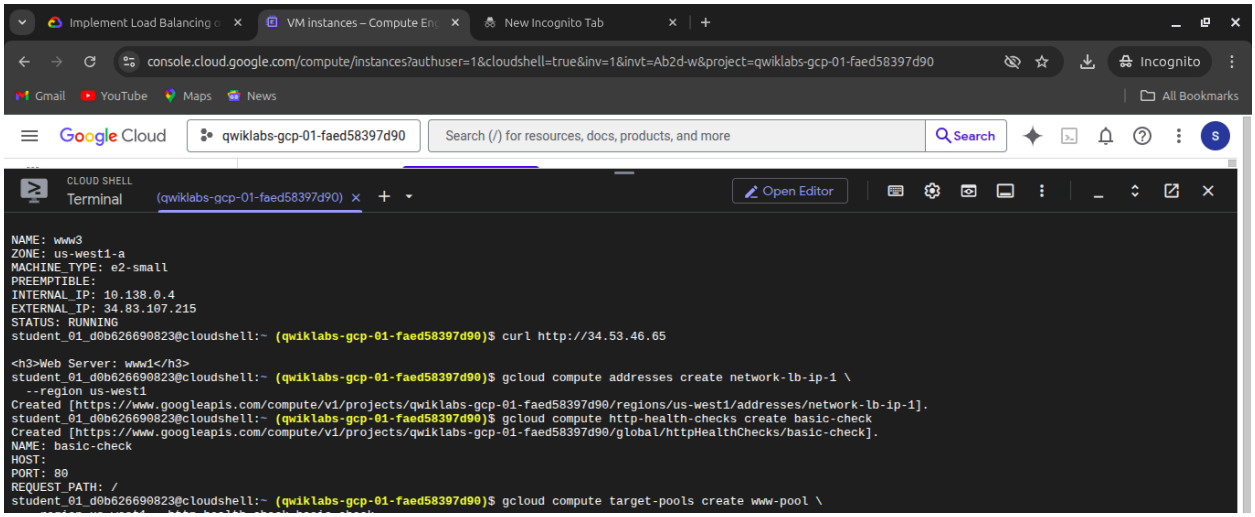


Step 3: I Set Up a Network Load Balancer. I started by creating a Reserve a static IP:

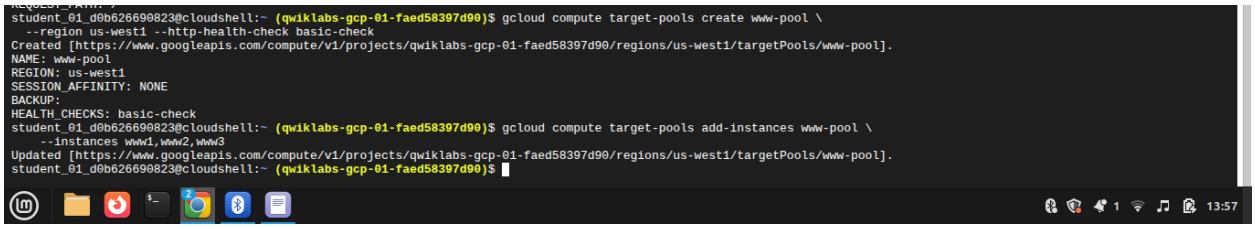
gcloud compute addresses create network-lb-ip-1 --region us-west1



Also, I created a Health Check using
gcloud compute http-health-checks create basic-check creates an HTTP Health Check named `basic-check` in Google Cloud. This command tell Google Cloud to send an HTTP request to port 80 and expect a 200 OK response (meaning the web server is running)

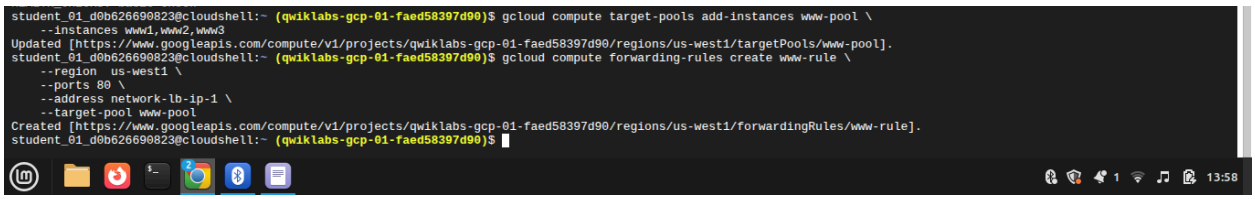


Next, I created a target pool in the same region as my instances with the command gcloud compute target-pools create www-pool --region us-west1 --http-health-check basic-check



Next, I added the 3 VMs to the Pool with the command

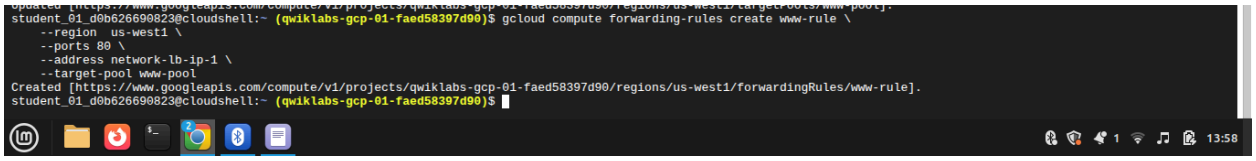
gcloud compute target-pools add-instances www-pool --instances www1,www2,www3



After that, I created a Forwarding Rule with the command

```
gcloud compute forwarding-rules create www-rule \
  --region us-west1 \
  --ports 80 \
  --address network-lb-ip-1 \
  --target-pool www-pool
```

This rule forwards traffic from the static IP to the target pool which distributes it to my 3 VMs.



Step 4: I sent traffic to my instances

I started by testing the Load Balancer using the command

```
IPADDRESS=$(gcloud compute forwarding-rules describe www-rule --region us-west1 --format="json" | jq -r .IPAddress)

echo $IPADDRESS
while true; do curl -m1 $IPADDRESS; done
```

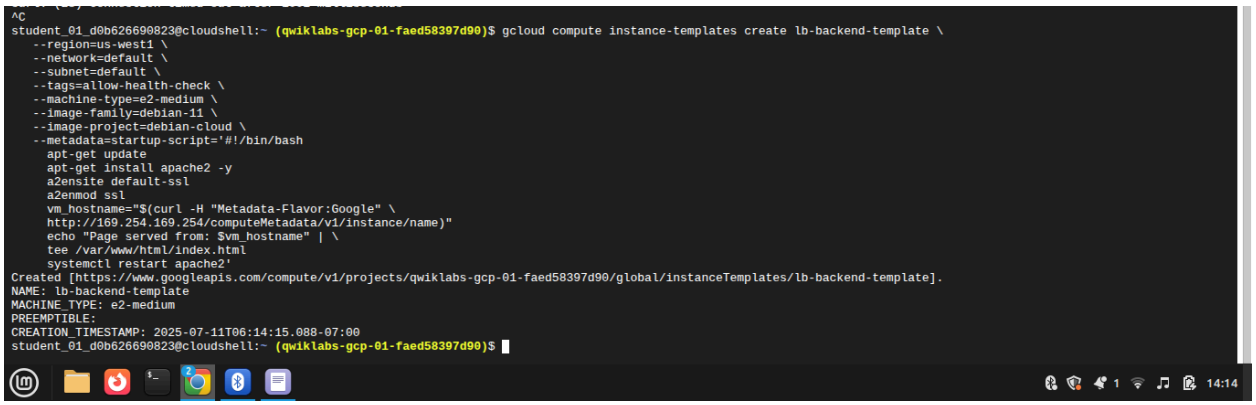
This process Loops and sends HTTP requests to the Load Balancer's IP. The response alternates between www1, www2, and www3 proving that traffic is being balanced.

Step 5: I set up an Application Load Balancer (Layer 7)

I created an Instance Template:

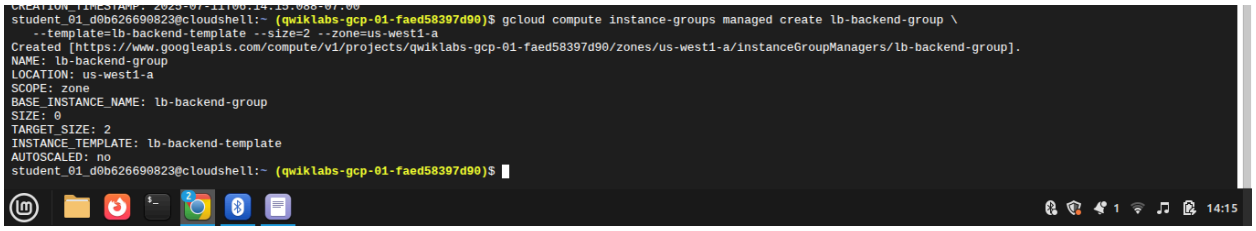
```
gcloud compute instance-templates create lb-backend-template \
   --region=Region \
   --network=default \
   --subnet=default \
   --tags=allow-health-check \
```

```
  --machine-type=e2-medium \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    a2ensite default-ssl
    a2enmod ssl
    vm_hostname="$(curl -H "Metadata-Flavor:Google" \
    http://169.254.169.254/computeMetadata/v1/instance/name)"
    echo "Page served from: $vm_hostname" | \
    tee /var/www/html/index.html
    systemctl restart apache2'
```



Next, I created a Managed Instance Group:

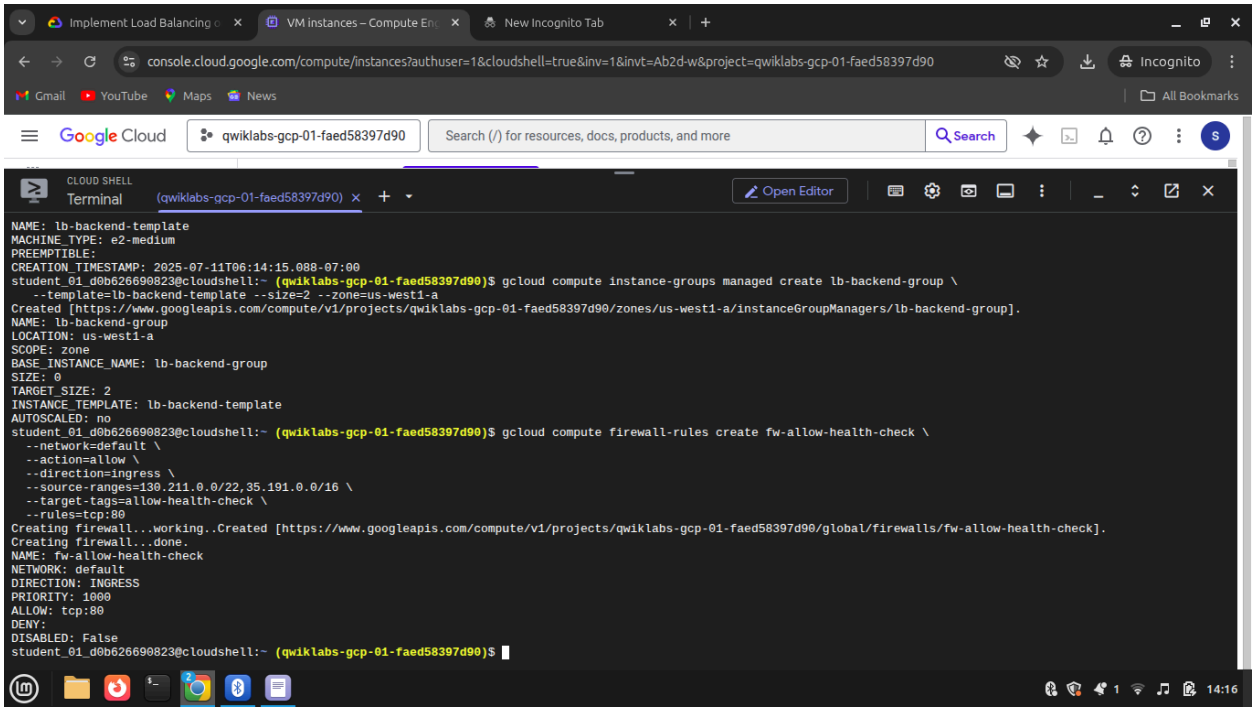
```
gcloud compute instance-groups managed create lb-backend-group \
  --template=lb-backend-template --size=2 --zone=us-west1-a
```



Next, I added and allowed Health Checks:

```
gcloud compute firewall-rules create fw-allow-health-check \
  --network=default \
  --action=allow \
```

--direction=ingress \
--source-ranges=130.211.0.0/22,35.191.0.0/16 \
--target-tags=allow-health-check \
--rules=tcp:80

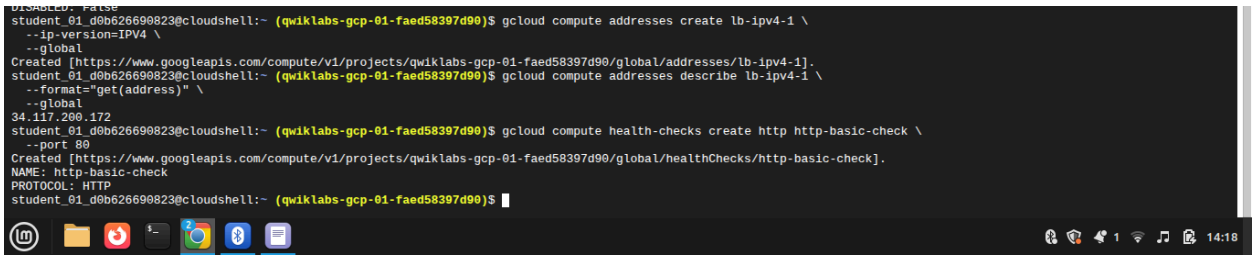


Now that the instances are up and running, I set up a global static external IP address that people can use to reach my load balance using the command gcloud compute addresses create lb-ipv4-1 --ip-version=IPV4 --global

I also created a health check gcloud compute health-checks create http http-basic-check --port 80

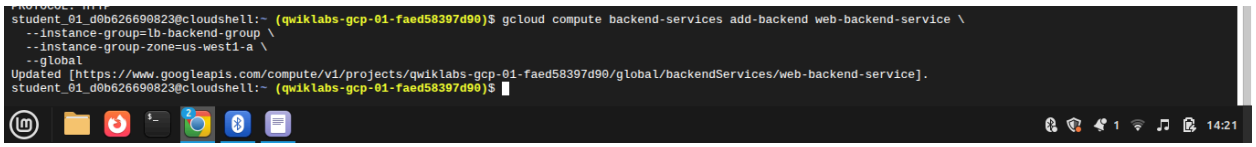
And a backend service with the command gcloud compute backend-services create web-backend-service \

--protocol=HTTP --port-name=http --health-checks=http-basic-check --global



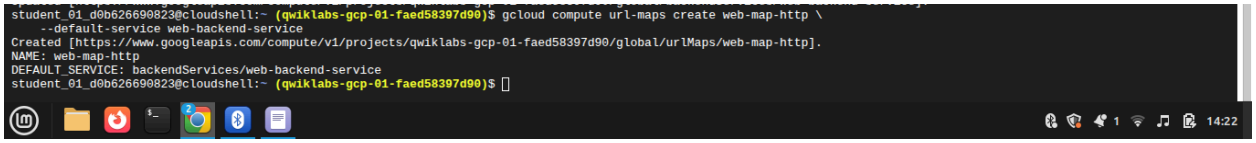
Next, I added my instance group as the backend to the backend service

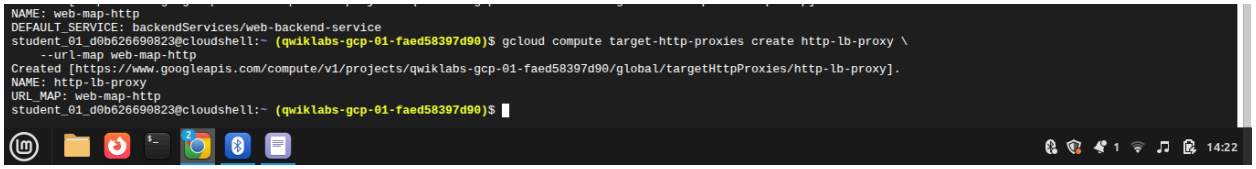
```
gcloud compute backend-services add-backend web-backend-service \
  --instance-group=lb-backend-group --instance-group-zone=us-west1-a --global
```



After that, I created a url map to route the incoming requests to the default backend service and also, I created a target HTTP proxy to route requests to my url map

```
gcloud compute url-maps create web-map-http --default-service web-backend-service
gcloud compute target-http-proxies create http-lb-proxy --url-map web-map-http
```





Next, I created a global forwarding rule to route incoming requests to the proxy:

```
gcloud compute forwarding-rules create http-content-rule \
  --address=lb-ipv4-1 --global --target-http-proxy=http-lb-proxy --ports=80
```