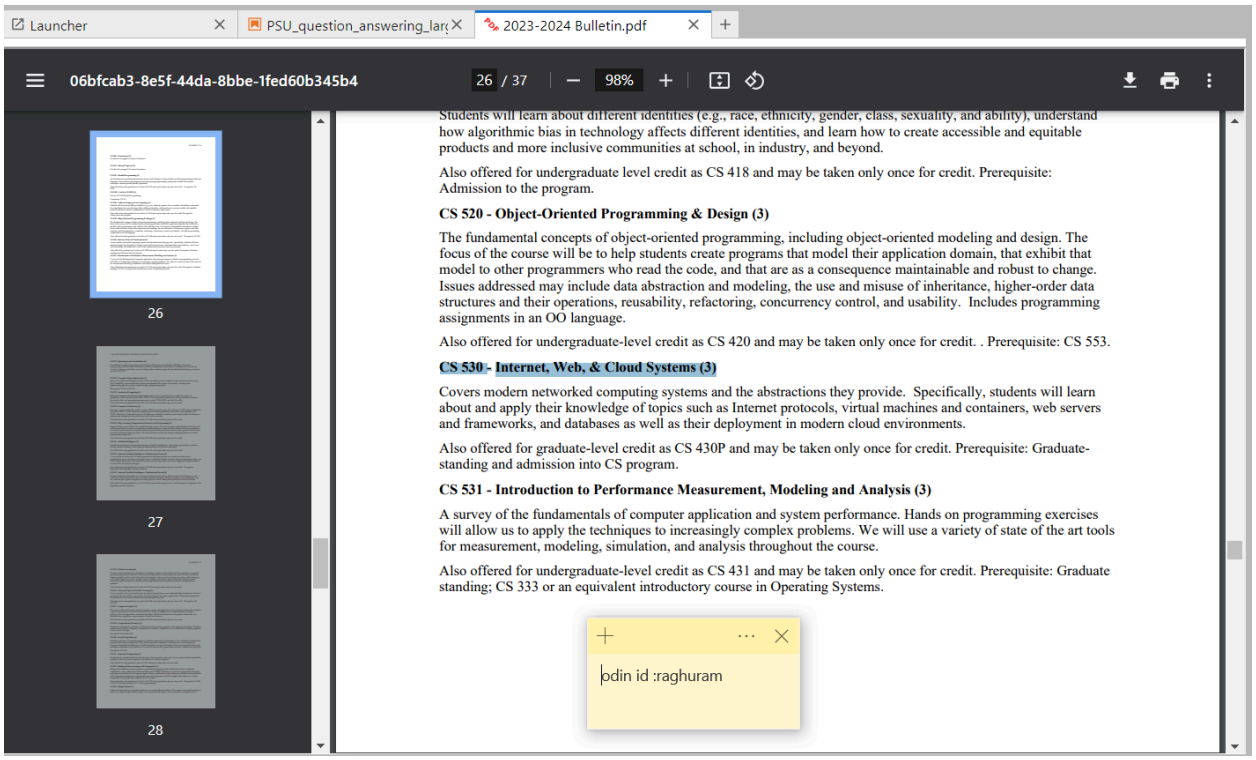
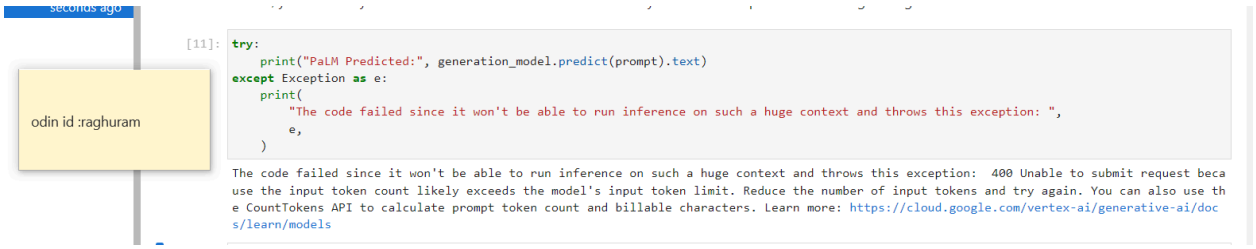
10.1g: LLMs

4. Walk through notebook

- **Take a screenshot that includes your OdinID showing the page number and the description of the class for your lab notebook**



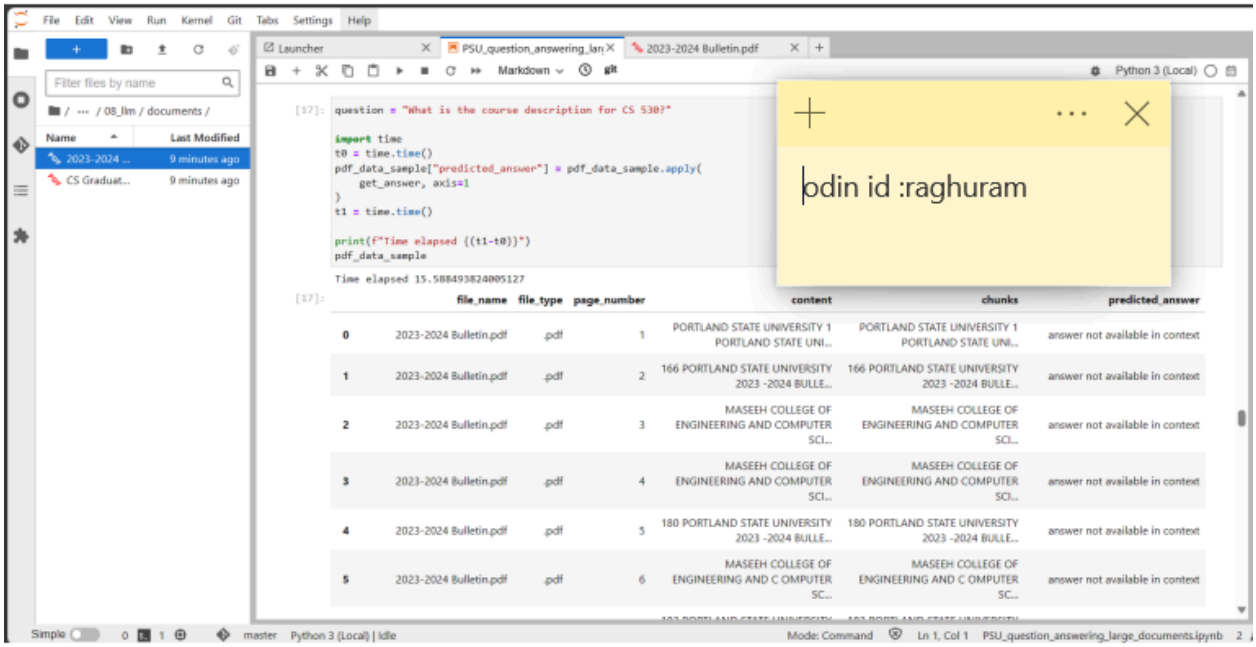
- **Take a screenshot that includes your OdinID showing the error that is returned for your lab notebook**



- **Provide an explanation as to why the description is not returned for your lab notebook**

  The lab notebook description cannot be provided because the necessary context, referred to as context[:5000], is unavailable in the provided code. Without this information, it's not possible to generate a meaningful response

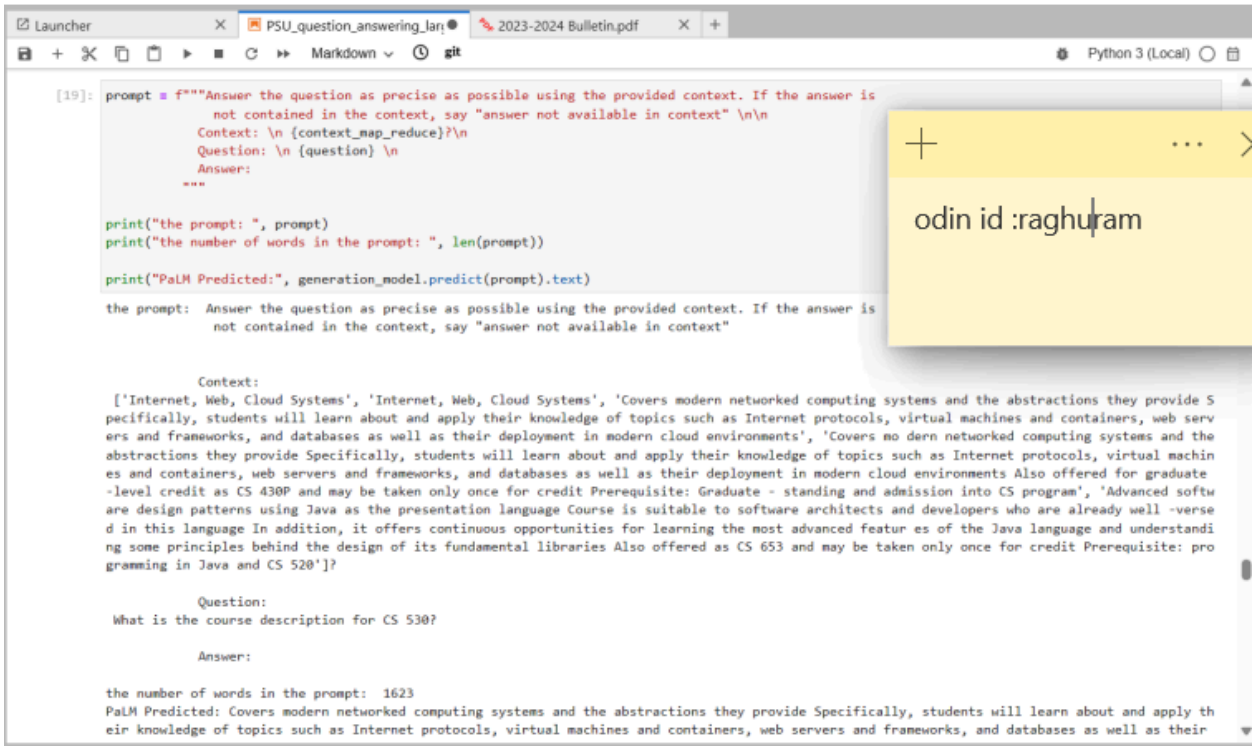
- **Take a screenshot including your OdinID that shows how long it took to perform the prediction across every chunk**



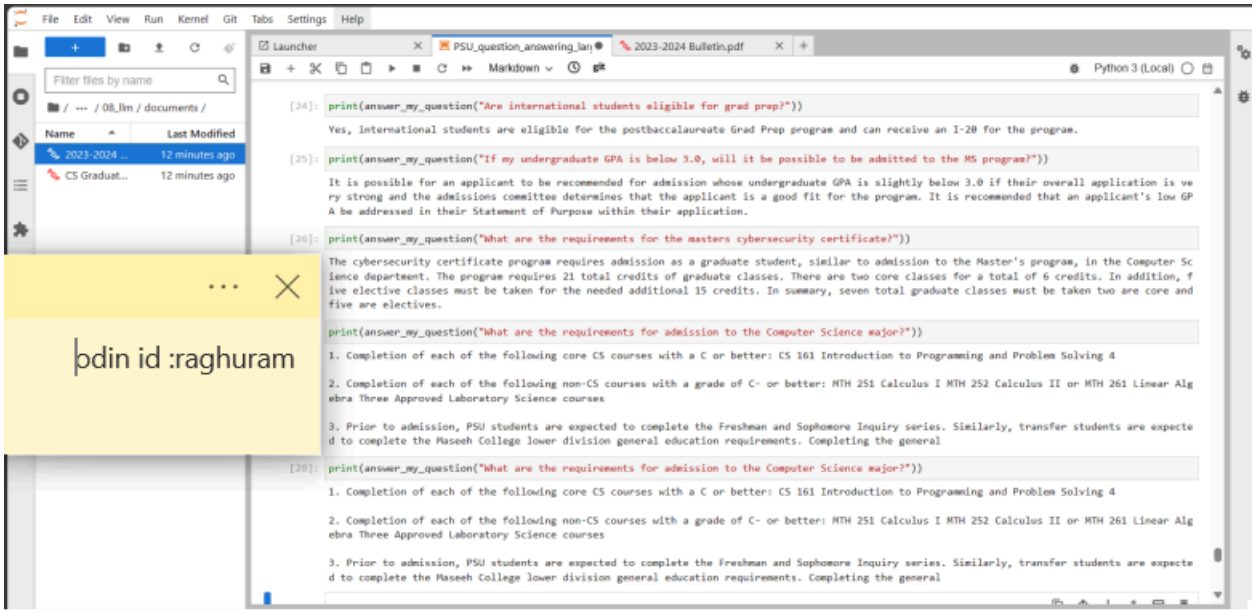
- **How many chunks returned predictions?**

  3 chunks returned

- **Take a screenshot that includes your OdinID showing the result that is returned for your lab notebook**



- **Take a screenshot including your OdinID that shows the results of the queries**



5. Final questions and clean-up

- Which of the approaches described would have issues with token limits on LLMs?

  Token limits on LLMs can cause issues with the "Stuffing" technique, which involves consolidating all data into one prompt. This might exceed the token limit, especially with substantial data.

- Which of the approaches would result in the most queries for the LLM to handle? How many LLM requests are performed from a single user query in this approach?

  Multiple requests are made

- Which of the approaches requires one to search a vector database for an appropriate context that is then sent to the LLM?

  The Map Reduce with embeddings

10.2g: CDN

6. Deployment

- Take a screenshot of the output to include in your lab notebook. How many networks, subnetworks, and VM instances have been created?

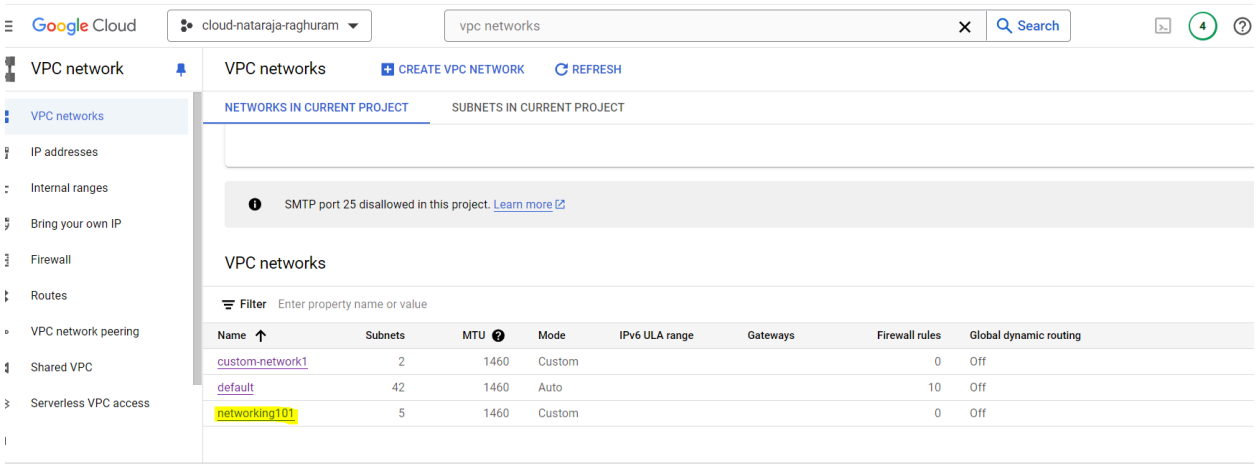
```
raghuram@cloudshell:~/networking101 (cl[illegible] raghuram)$ gcloud deployment-manager deployments create networking101 --config networking-lab.yaml
The fingerprint of the deployment is b'77lgfjJSBMZLDQzNE9MR-g=='
Waiting for create [operation-1718066612628-61a928e256b5a-0dfebe44-bd62ee95]...done.
Create operation operation-1718066612628-61a928e256b5a-0dfebe44-bd62ee95 completed successfully.
NAME: asia-east1
TYPE: compute.v1.subnetwork
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: asia1-vm
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: e1-vm
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: eu1-vm
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: europe-west1
TYPE: compute.v1.subnetwork
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: networking101
TYPE: compute.v1.network
STATE: COMPLETED
ERRORS: []
INTENT:
```

```
NAME: networking101
TYPE: compute.v1.network
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: us-east5
TYPE: compute.v1.subnetwork
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: us-west-s1
TYPE: compute.v1.subnetwork
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: us-west-s2
TYPE: compute.v1.subnetwork
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: w1-vm
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: w2-vm
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:
raghuram@cloudshell:~/networking101 (cloud-nataraja-raghuram)$
```

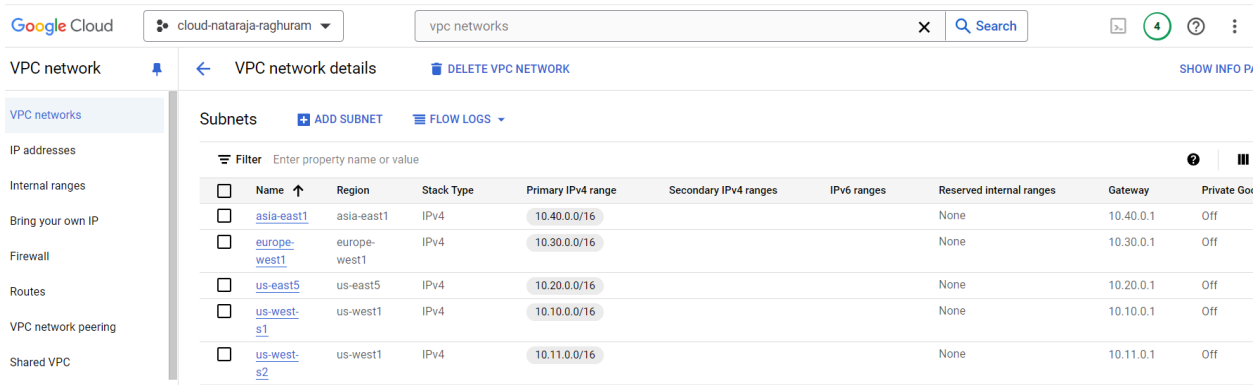
5-subnetworks
1-network
5-instances

- Visit the web console for VPC network and show the network and the subnetworks that have been created. Validate that it has created the infrastructure in the initial figure. Note the lack of firewall rules that have been created

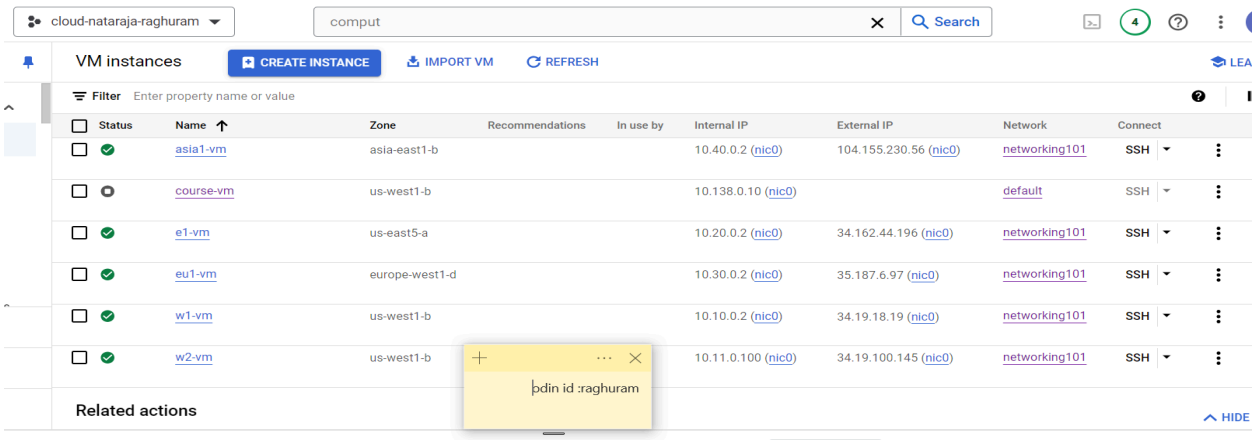
network



subnetwork



- Visit the web console for Compute Engine and show all VMs that have been created, their internal IP addresses and the subnetworks they have been instantiated on. Validate that it has created the infrastructure shown in the initial figure.

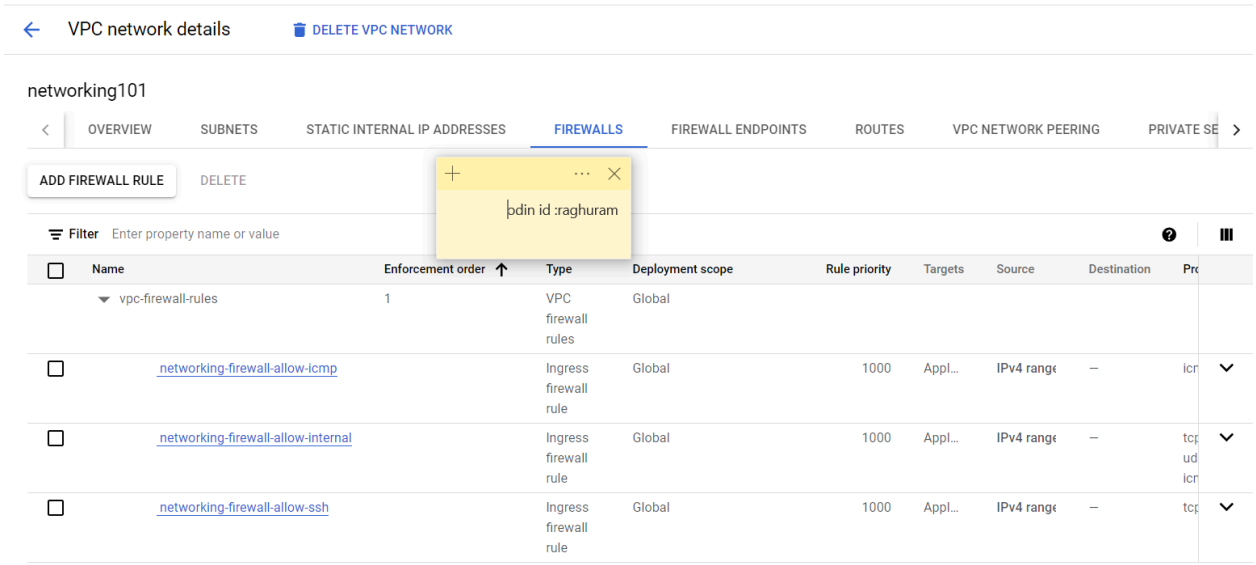


- Click on the ssh button for one of the VMs and attempt to connect. Did it succeed?

  No it did not connect.
  ERROR: (gcloud.compute.ssh) [/usr/bin/ssh] exited with return code [255].

8. Update deployment

- Take a screenshot that indicates the new rules have been deployed



- Given this, fill in the table with the measured latencies between the 6 pairs and include it in your lab notebook. Use the shortest latency measured for each pair.

Location pair	Ideal latency	Measured latency
us-west1 us-east5	~45 ms	49.7 ms
us-west1 europe-west1	~93 ms	134ms
us-west1 asia-east1	~114 ms	118.6 ms
us-east5 europe-west1	~76 ms	87.4ms
us-east5 asia-east1	~141 ms	166.9 ms
europe-west1 asia-east1	~110 ms	250.5 ms

16. Test groups

- Are the instances in the same availability zone or in different ones?

  They are all in different zones

- List all availability zones that your servers show up in for your lab notebook.

					(nic0)			
e1-vm	us-east5-a				10.20.0.2 (nic0)	34.162.44.196 (nic0)	networking101	S
eu1-vm	europe-west1-d				10.30.0.2 (nic0)	35.187.6.97 (nic0)	networking101	S
europe-west1-mig-2fgw	europe-west1-c		europe-west1-mig		10.30.0.3 (nic0)	34.38.145.242 (nic0)	networking101	S
europe-west1-mig-c1tw	europe-west1-b		europe-west1-mig		10.30.0.5 (nic0)	35.195.51.136 (nic0)	networking101	S
europe-west1-mig-w8b1	europe-west1-d		europe-west1-mig		10.30.0.4 (nic0)	34.38.137.163 (nic0)	networking101	S

pdin id :raghuram

Not secure 34.38.137.163

Networking 101 Lab

Client IP

Your IP address : 71.59.145.43

pdin id :raghuram

Hostname

Server Hostname: europe-west1-mig-w8b1

Server Location

Region and Zone: europe-west1-d

Not secure 35.195.51.136

Networking 101 Lab

Client IP

Your IP address : 71.59.145.43

pdin id :raghuram

Hostname

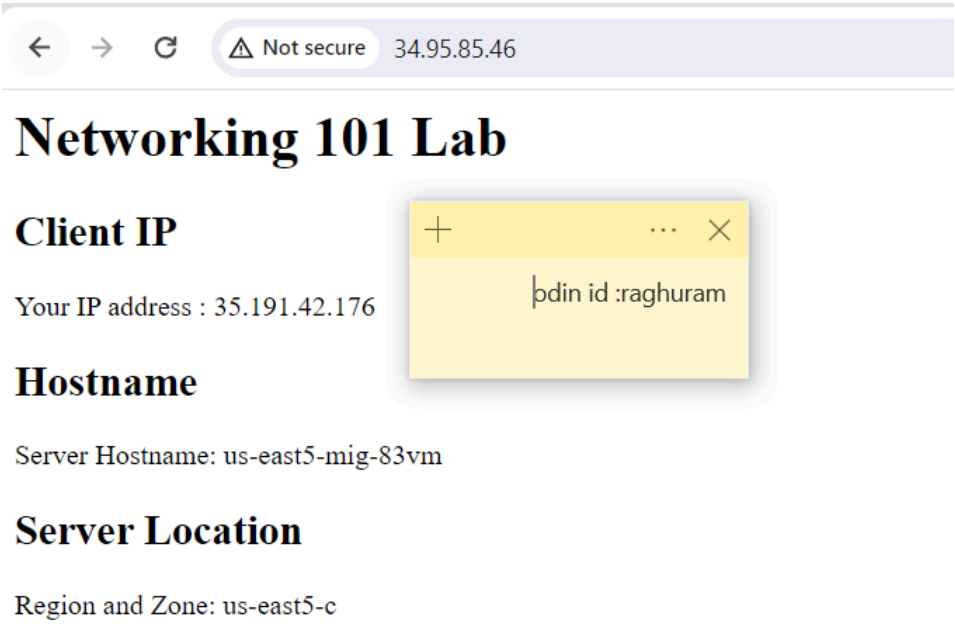
Server Hostname: europe-west1-mig-c1tw

Server Location

Region and Zone: europe-west1-b

19. Test load balancer

- Show a screenshot of the page that is returned.

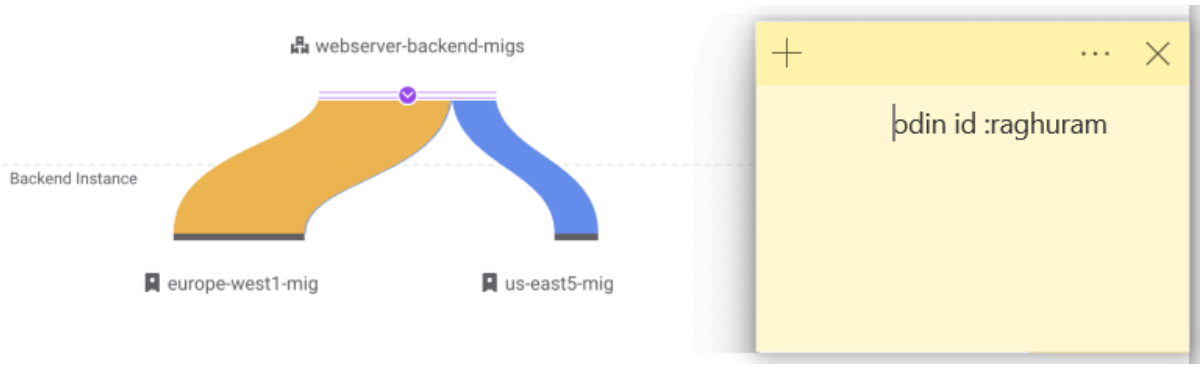


- Which availability zone does the server handling your request reside in?

  Region and Zone: us-east5-c

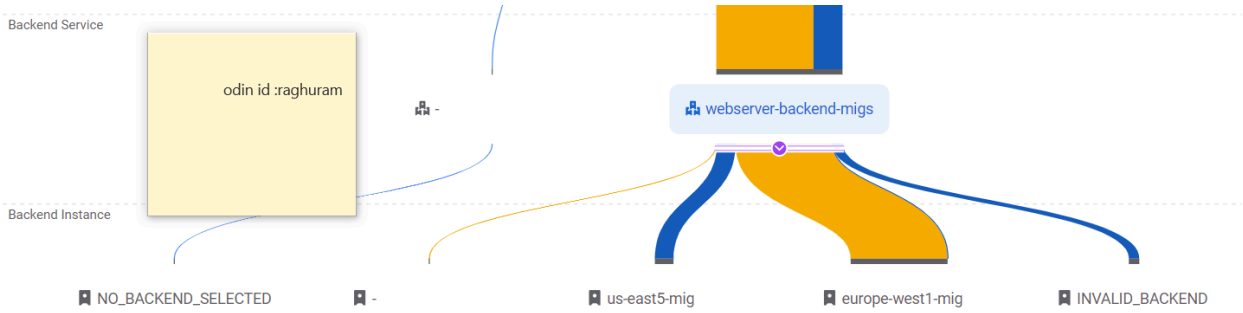
20. Siege! (Part 1)

- Take a screenshot of the initial traffic distribution



.

- Take a screenshot of the UI as additional instances are brought up and show that the traffic distribution shifts



21. Siege! (Part 2)

- Show a screenshot of the final traffic distribution.