CS530 F23 Week 10 Lab
Prof. Feng

Amelia Miner
11/24/23

✅ indicates that a section was completed but did not request any screenshots or written answers.

TABLE OF CONTENTS:

Amelia Miner
11/24/23

I. Lab 10.1g: LLMs

1. Large Language Models

- In this lab we'll run a Jupyter notebook on Vertex AI to experiment with different ways of implementing a custom document Q&A.
  ✅

2. Deploy Notebook

- From Cloud Shell, deploy a user notebook on Vertex AI:

```
gcloud notebooks instances create llm-jupyter-instance \
  --vm-image-project=deeplearning-platform-release \
  --vm-image-family=common-cpu-notebooks \
  --machine-type=e2-standard-4 \
  --location=us-central1-a
```

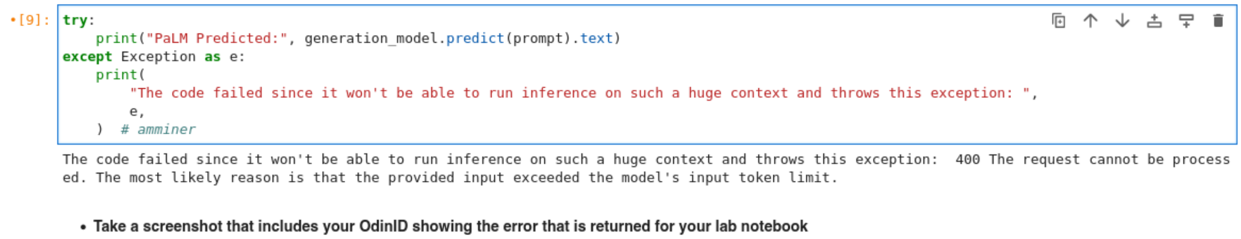
✅

3. launch Document Q&A notebook

- Bring the notebook up. Clone the course repository (https://github.com/wu4f/cs430-src) using the GUI. Use the file explorer to navigate to the Python notebook located at cs430-src/08_llm/PSU_question_answering_large_documents.ipynb.
  ✅

4. Walk through the notebook

- Follow the instructions in the notebook, which will ask you to take some screenshots at various points.
  - Too many tokens when attempting to stuff a large context:

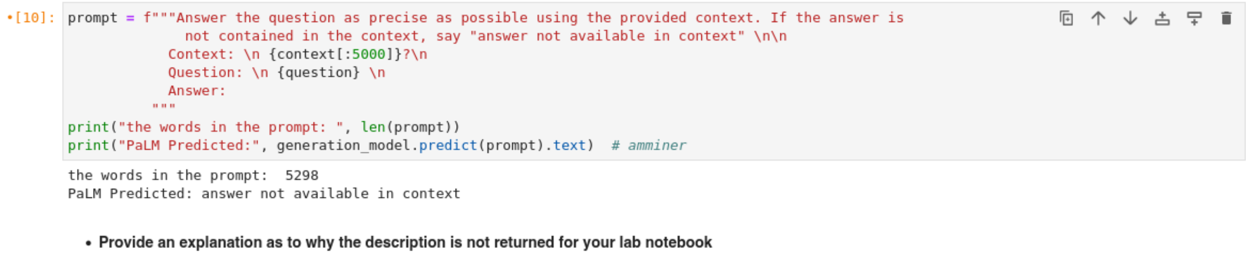


```
•[9]: try:
          print("PaLM Predicted:", generation_model.predict(prompt).text)
      except Exception as e:
          print(
              "The code failed since it won't be able to run inference on such a huge context and throws this exception: ",
              e,
          )  # amminer
```

```
The code failed since it won't be able to run inference on such a huge context and throws this exception:  400 The request cannot be process
ed. The most likely reason is that the provided input exceeded the model's input token limit.
```

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

  - Limiting the context to the first 5000 words that were parsed from the documents:



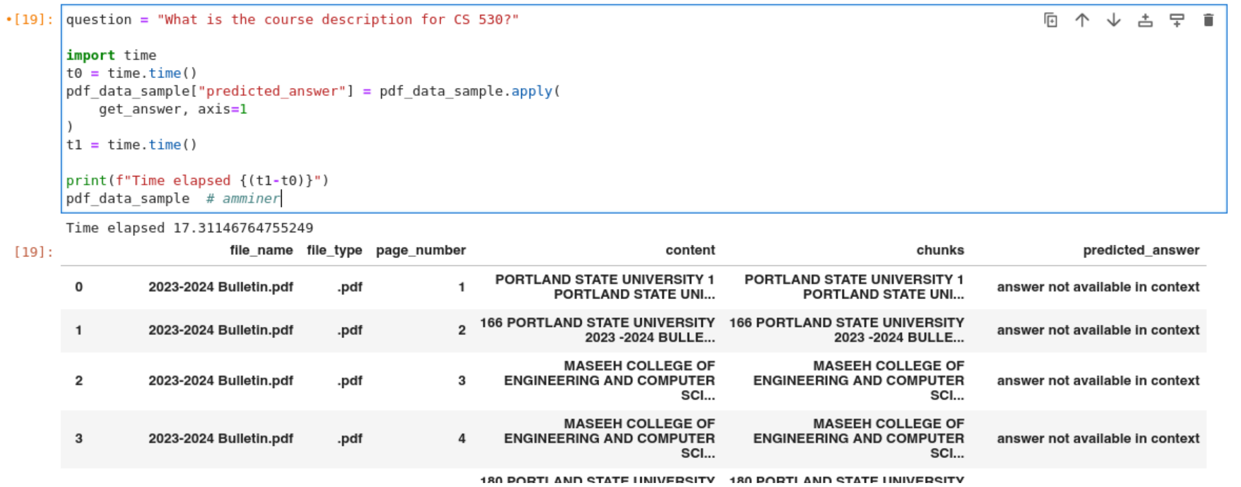
```
•[10]: prompt = f"""Answer the question as precise as possible using the provided context. If the answer is
                   not contained in the context, say "answer not available in context" \n\n
                 Context: \n {context[:5000]}?\n
                 Question: \n {question} \n
                 Answer:
               """
       print("the words in the prompt: ", len(prompt))
       print("PaLM Predicted:", generation_model.predict(prompt).text)  # amminer
```

```
the words in the prompt:  5298
PaLM Predicted: answer not available in context
```

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

The description is not returned because the requested information is not contained in the provided context.

  - map-reduce: Show how long it took to process all of the chunks



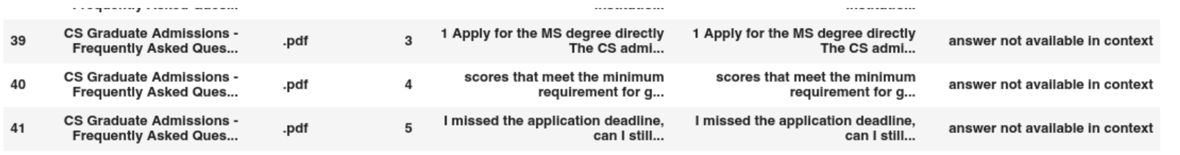
```
•[19]: question = "What is the course description for CS 530?"

       import time
       t0 = time.time()
       pdf_data_sample["predicted_answer"] = pdf_data_sample.apply(
           get_answer, axis=1
       )
       t1 = time.time()

       print(f"Time elapsed {(t1-t0)}")
       pdf_data_sample  # amminer
```

```
Time elapsed 17.31146764755249
```

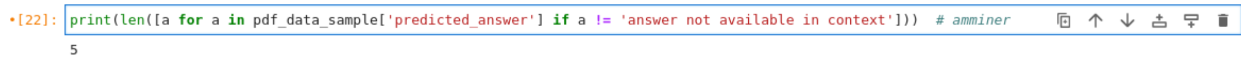
[19]:

	file_name	file_type	page_number	content	chunks	predicted_answer
0	2023-2024 Bulletin.pdf	.pdf	1	PORTLAND STATE UNIVERSITY 1 PORTLAND STATE UNI...	PORTLAND STATE UNIVERSITY 1 PORTLAND STATE UNI...	answer not available in context
1	2023-2024 Bulletin.pdf	.pdf	2	166 PORTLAND STATE UNIVERSITY 2023 -2024 BULLE...	166 PORTLAND STATE UNIVERSITY 2023 -2024 BULLE...	answer not available in context
2	2023-2024 Bulletin.pdf	.pdf	3	MASEEH COLLEGE OF ENGINEERING AND COMPUTER SCI...	MASEEH COLLEGE OF ENGINEERING AND COMPUTER SCI...	answer not available in context
3	2023-2024 Bulletin.pdf	.pdf	4	MASEEH COLLEGE OF ENGINEERING AND COMPUTER SCI...	MASEEH COLLEGE OF ENGINEERING AND COMPUTER SCI...	answer not available in context
				180 PORTLAND STATE UNIVERSITY	180 PORTLAND STATE UNIVERSITY	

  - How many chunks returned predictions?



39	CS Graduate Admissions - Frequently Asked Ques...	.pdf	3	1 Apply for the MS degree directly The CS admi...	1 Apply for the MS degree directly The CS admi...	answer not available in context
40	CS Graduate Admissions - Frequently Asked Ques...	.pdf	4	scores that meet the minimum requirement for g...	scores that meet the minimum requirement for g...	answer not available in context
41	CS Graduate Admissions - Frequently Asked Ques...	.pdf	5	I missed the application deadline, can I still...	I missed the application deadline, can I still...	answer not available in context

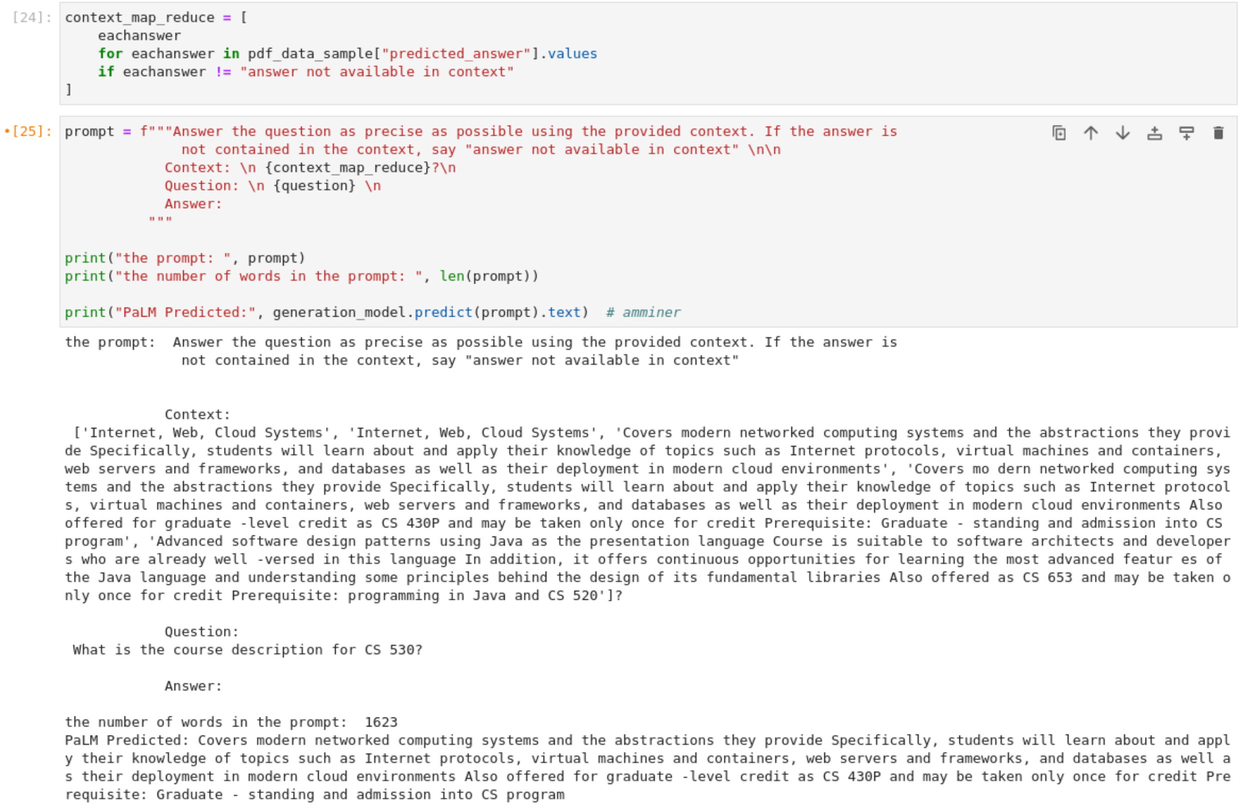


```
•[22]: print(len([a for a in pdf_data_sample['predicted_answer'] if a != 'answer not available in context']))  # amminer
```

```
5
```

5 chunks.

- Show the result of combining these answers into a new prompt:



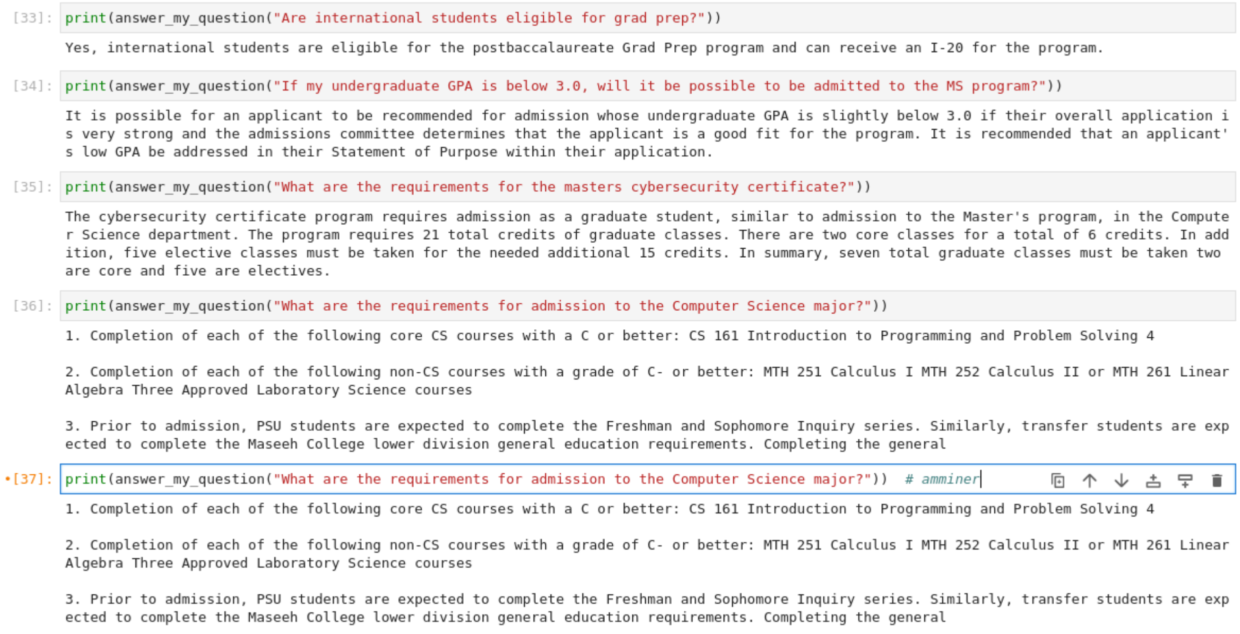
```
[24]: context_map_reduce = [
          eachanswer
          for eachanswer in pdf_data_sample["predicted_answer"].values
          if eachanswer != "answer not available in context"
      ]
```

```
•[25]: prompt = f"""Answer the question as precise as possible using the provided context. If the answer is
                     not contained in the context, say "answer not available in context" \n\n
                   Context: \n {context_map_reduce}?\n
                   Question: \n {question} \n
                   Answer:
                 """

       print("the prompt: ", prompt)
       print("the number of words in the prompt: ", len(prompt))

       print("PaLM Predicted:", generation_model.predict(prompt).text)  # amminer
```

```
the prompt:  Answer the question as precise as possible using the provided context. If the answer is
              not contained in the context, say "answer not available in context"


            Context:
 ['Internet, Web, Cloud Systems', 'Internet, Web, Cloud Systems', 'Covers modern networked computing systems and the abstractions they provi
de Specifically, students will learn about and apply their knowledge of topics such as Internet protocols, virtual machines and containers,
web servers and frameworks, and databases as well as their deployment in modern cloud environments', 'Covers mo dern networked computing sys
tems and the abstractions they provide Specifically, students will learn about and apply their knowledge of topics such as Internet protocol
s, virtual machines and containers, web servers and frameworks, and databases as well as their deployment in modern cloud environments Also
offered for graduate -level credit as CS 430P and may be taken only once for credit Prerequisite: Graduate - standing and admission into CS
program', 'Advanced software design patterns using Java as the presentation language Course is suitable to software architects and developer
s who are already well -versed in this language In addition, it offers continuous opportunities for learning the most advanced featur es of
the Java language and understanding some principles behind the design of its fundamental libraries Also offered as CS 653 and may be taken o
nly once for credit Prerequisite: programming in Java and CS 520']?

            Question:
 What is the course description for CS 530?

            Answer:

the number of words in the prompt:  1623
PaLM Predicted: Covers modern networked computing systems and the abstractions they provide Specifically, students will learn about and appl
y their knowledge of topics such as Internet protocols, virtual machines and containers, web servers and frameworks, and databases as well a
s their deployment in modern cloud environments Also offered for graduate -level credit as CS 430P and may be taken only once for credit Pre
requisite: Graduate - standing and admission into CS program
```

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

- Ask a handful of questions using the map-reduce with embeddings method.



```
[33]: print(answer_my_question("Are international students eligible for grad prep?"))

      Yes, international students are eligible for the postbaccalaureate Grad Prep program and can receive an I-20 for the program.

[34]: print(answer_my_question("If my undergraduate GPA is below 3.0, will it be possible to be admitted to the MS program?"))

      It is possible for an applicant to be recommended for admission whose undergraduate GPA is slightly below 3.0 if their overall application i
      s very strong and the admissions committee determines that the applicant is a good fit for the program. It is recommended that an applicant'
      s low GPA be addressed in their Statement of Purpose within their application.

[35]: print(answer_my_question("What are the requirements for the masters cybersecurity certificate?"))

      The cybersecurity certificate program requires admission as a graduate student, similar to admission to the Master's program, in the Compute
      r Science department. The program requires 21 total credits of graduate classes. There are two core classes for a total of 6 credits. In add
      ition, five elective classes must be taken for the needed additional 15 credits. In summary, seven total graduate classes must be taken two
      are core and five are electives.

[36]: print(answer_my_question("What are the requirements for admission to the Computer Science major?"))

      1. Completion of each of the following core CS courses with a C or better: CS 161 Introduction to Programming and Problem Solving 4

      2. Completion of each of the following non-CS courses with a grade of C- or better: MTH 251 Calculus I MTH 252 Calculus II or MTH 261 Linear
      Algebra Three Approved Laboratory Science courses

      3. Prior to admission, PSU students are expected to complete the Freshman and Sophomore Inquiry series. Similarly, transfer students are exp
      ected to complete the Maseeh College lower division general education requirements. Completing the general

•[37]: print(answer_my_question("What are the requirements for admission to the Computer Science major?"))  # amminer

      1. Completion of each of the following core CS courses with a C or better: CS 161 Introduction to Programming and Problem Solving 4

      2. Completion of each of the following non-CS courses with a grade of C- or better: MTH 251 Calculus I MTH 252 Calculus II or MTH 261 Linear
      Algebra Three Approved Laboratory Science courses

      3. Prior to admission, PSU students are expected to complete the Freshman and Sophomore Inquiry series. Similarly, transfer students are exp
      ected to complete the Maseeh College lower division general education requirements. Completing the general
```

- **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?
  Stuffing.

- 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?
  Map-reduce. The LLM is called once for each chunk, plus at least one more call once useful chunks have been combined based on the initial responses. Exact number depends on the input and the limitations of the model.

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

- Delete the notebook.
  ✅

II. Lab 10.2g: CDN

1. Part 1: Networks and VMs

- In this lab we'll explore an infrastructure as code solution for deploying a flexible, scalable website with a CDN in a repeatable manner.
  The network is named networking101 with instances in 3 regions, us-east1 10.20.0.0/16 and europe-west1 10.30.0.0/16 plus us-west1 10.11.0.0/16 for stress testing.
  The figure in this step shows an additional region, asia-east1, which is not in line with the description provided, just a heads up for the grader/instructor.
  ✅

2. Deployment Specification

- In cloud shell, copy the files for this lab from a bucket like so:
  `gsutil cp -r gs://cs430/networking101`
  cd in. gcloud's Deployment Manager works from a main YAML file (networking-lab.yaml) which references several jinja template files.
  ✅

3. Network Deployment Specification

- Each jinja file declares a component of the infrastructure we wish to deploy. Examine compute-engine-template.jinja and network-template.jinja. The former manages the resources defined in the rest of the jinja files.
  ✅

4. Subnetwork Deployment Specification

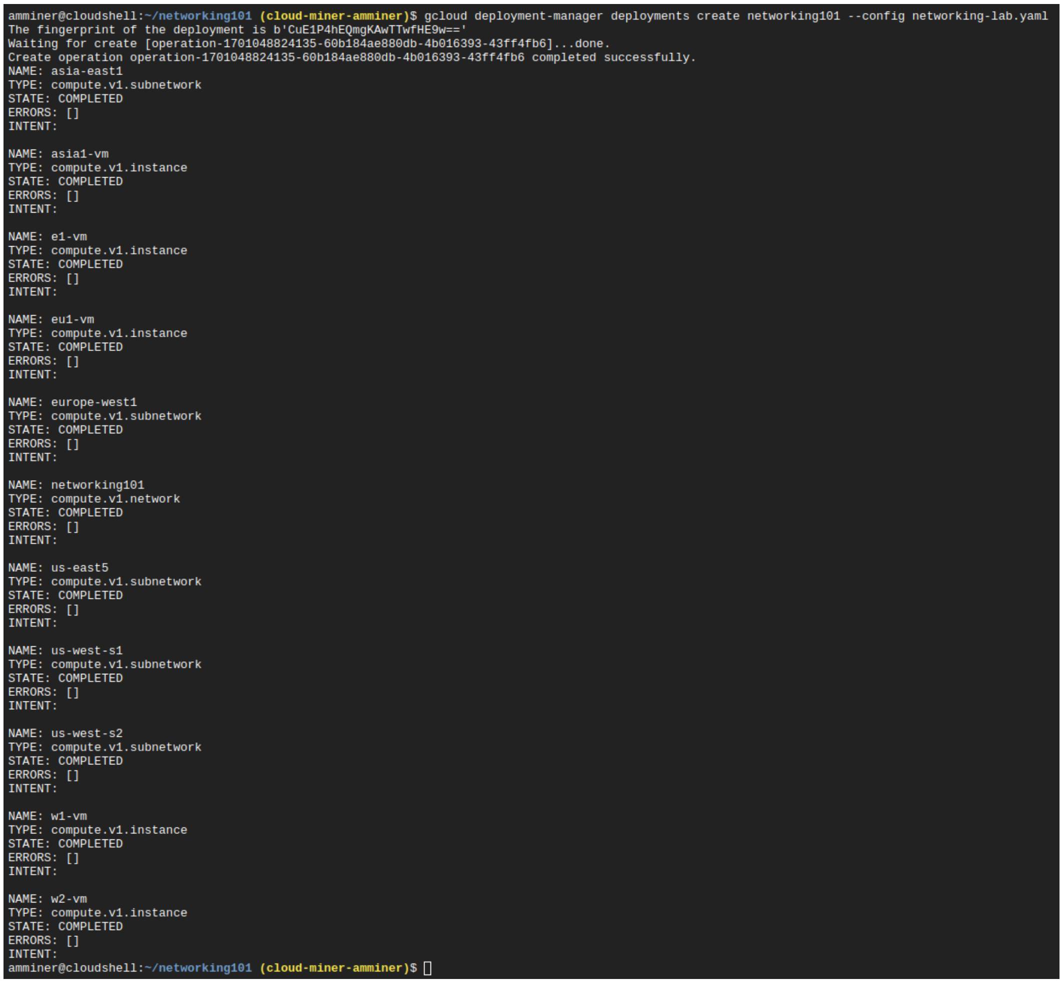
- examine the portions of compute-engine-template.jinja which define the subnetworks using subnetwork-template.jinja.
  ✅

5. Virtual Machine Deployment Specification

- Examine the portion of compute-engine-template.jinja that defines the VMs we'll bring up. Note that we will be deploying 5 VMs - this will be expensive if you're not timely about working through the lab and deleting the resources!
  Look at the startup script portion of the metadata block in vm-template.jinja.
  ✅

6. Deployment

- In cloud shell, deploy:
  `gcloud deployment-manager deployments create networking101 --config networking-lab.yaml`
  Show the output. How many networks, subnets, and VMs were created?
  1, 5, and 5, respectively.

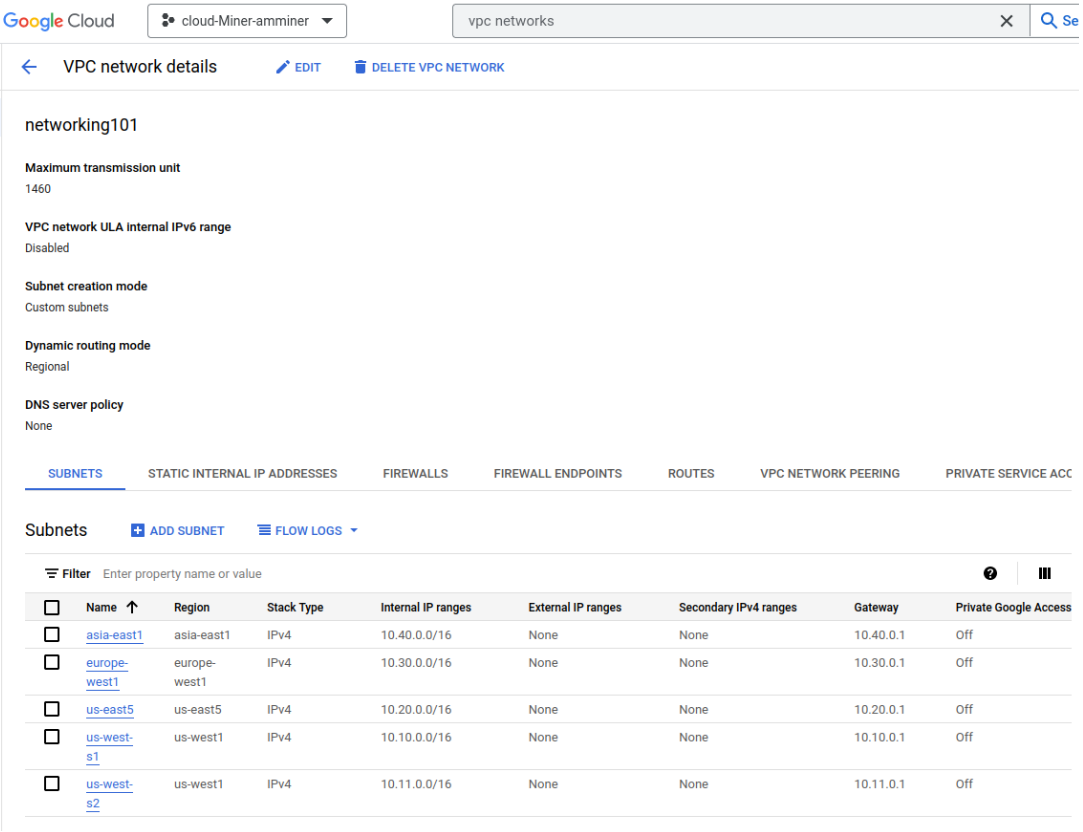


```
amminer@cloudshell:~/networking101 (cloud-miner-amminer)$ gcloud deployment-manager deployments create networking101 --config networking-lab.yaml
The fingerprint of the deployment is b'CuE1P4hEQmgKAwTTwfHE9w=='
Waiting for create [operation-1701048824135-60b184ae880db-4b016393-43ff4fb6]...done.
Create operation operation-1701048824135-60b184ae880db-4b016393-43ff4fb6 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: 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:
amminer@cloudshell:~/networking101 (cloud-miner-amminer)$
```

CS530 F23 Week 10 Lab
Prof. Feng

Amelia Miner
11/24/23

- 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.



networking101

Maximum transmission unit
1460

VPC network ULA internal IPv6 range
Disabled

Subnet creation mode
Custom subnets

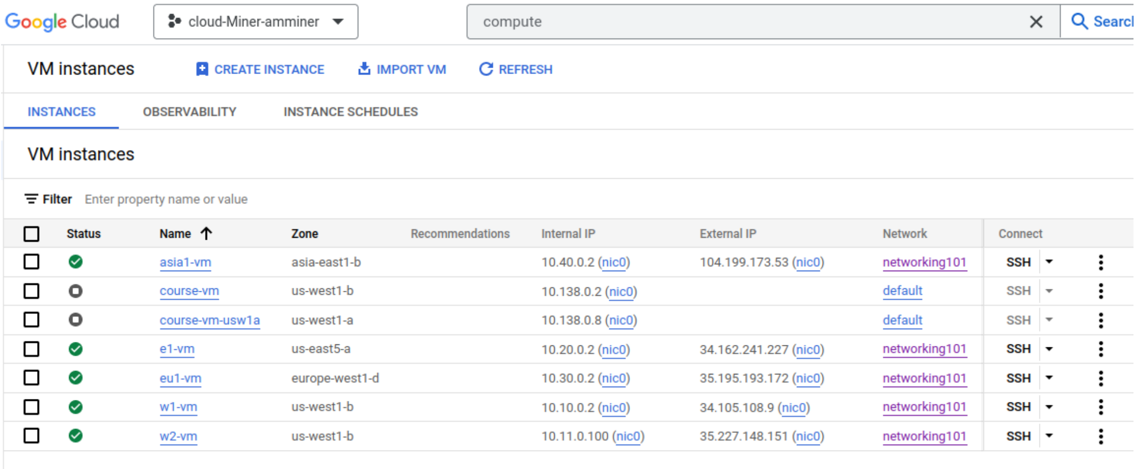
Dynamic routing mode
Regional

DNS server policy
None

Subnets

□	Name ↑	Region	Stack Type	Internal IP ranges	External IP ranges	Secondary IPv4 ranges	Gateway	Private Google Access
□	asia-east1	asia-east1	IPv4	10.40.0.0/16	None	None	10.40.0.1	Off
□	europe-west1	europe-west1	IPv4	10.30.0.0/16	None	None	10.30.0.1	Off
□	us-east5	us-east5	IPv4	10.20.0.0/16	None	None	10.20.0.1	Off
□	us-west-s1	us-west1	IPv4	10.10.0.0/16	None	None	10.10.0.1	Off
□	us-west-s2	us-west1	IPv4	10.11.0.0/16	None	None	10.11.0.1	Off

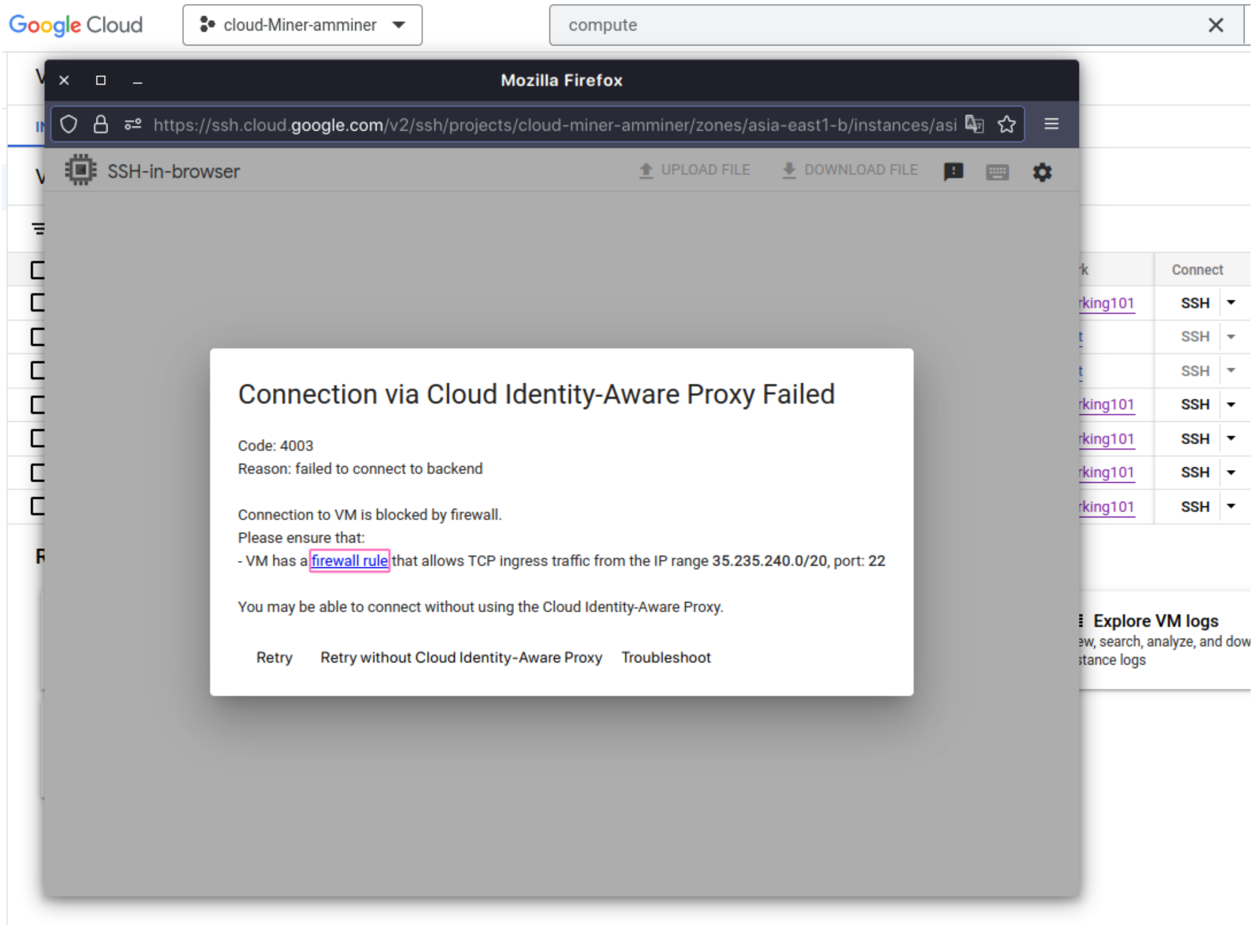
- Visit the web console for Compute Engine and show all VMs that have been created, their internal IP addresses and the subnetworks they have been instantiated on. Validate that it has created the infrastructure shown in the initial figure.



VM instances

□	Status	Name ↑	Zone	Recommendations	Internal IP	External IP	Network	Connect
□	✓	asia1-vm	asia-east1-b		10.40.0.2 (nic0)	104.199.173.53 (nic0)	networking101	SSH
□	○	course-vm	us-west1-b		10.138.0.2 (nic0)		default	SSH
□	○	course-vm-usw1a	us-west1-a		10.138.0.8 (nic0)		default	SSH
□	✓	e1-vm	us-east5-a		10.20.0.2 (nic0)	34.162.241.227 (nic0)	networking101	SSH
□	✓	eu1-vm	europe-west1-d		10.30.0.2 (nic0)	35.195.193.172 (nic0)	networking101	SSH
□	✓	w1-vm	us-west1-b		10.10.0.2 (nic0)	34.105.108.9 (nic0)	networking101	SSH
□	✓	w2-vm	us-west1-b		10.11.0.100 (nic0)	35.227.148.151 (nic0)	networking101	SSH

- Click on the ssh button for one of the VMs and attempt to connect. Did it succeed?
  No.



Connection via Cloud Identity-Aware Proxy Failed

Code: 4003
Reason: failed to connect to backend

Connection to VM is blocked by firewall.
Please ensure that:
- VM has a firewall rule that allows TCP ingress traffic from the IP range **35.235.240.0/20**, port: **22**

You may be able to connect without using the Cloud Identity-Aware Proxy.

Retry   Retry without Cloud Identity-Aware Proxy   Troubleshoot

7. Firewall deployment specification

- As it should be, our network is configured to block by default, an instance of the concept "secure by default". Add the following to the imports in networking-lab.yaml:

```
- path: firewall-template.jinja
```

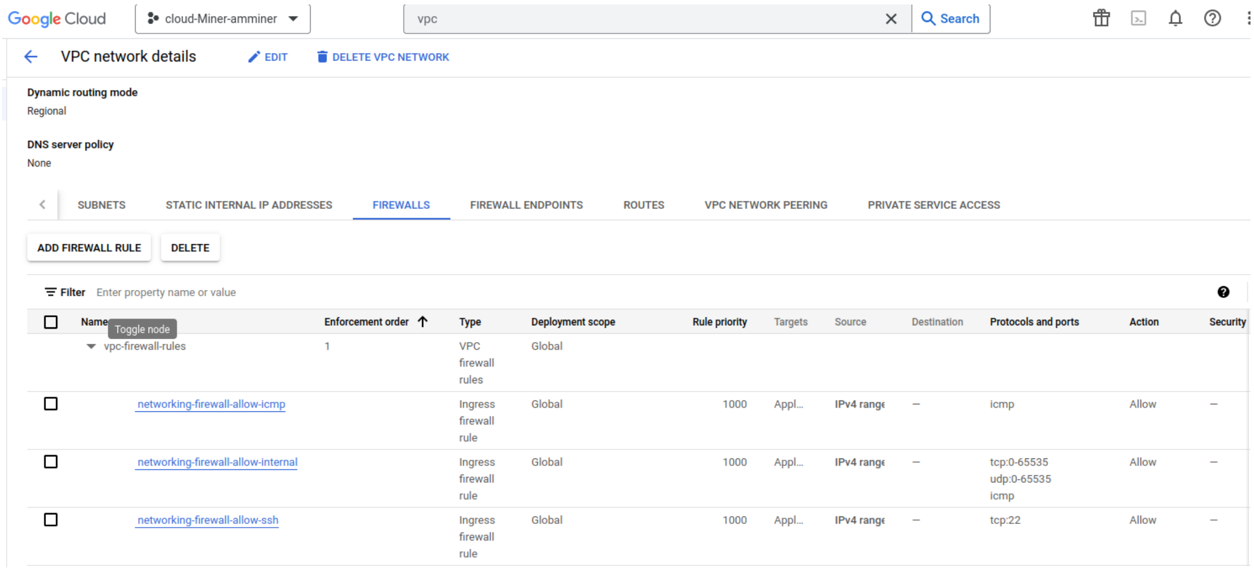
  Then under resources:

```
- name: networking-firewall
  type: firewall-template.jinja
  properties:
    network: networking101
```

  Create firewall-template.jinja and copy the contents in from the lab materials.
  ✅

8. Update deployment

- We can update the deployment in-place with `gcloud deployment-manager deployments update networking101 --config networking-lab.yaml` View the network in the VPC console. Take a screenshot indicating that the new rules have been deployed.

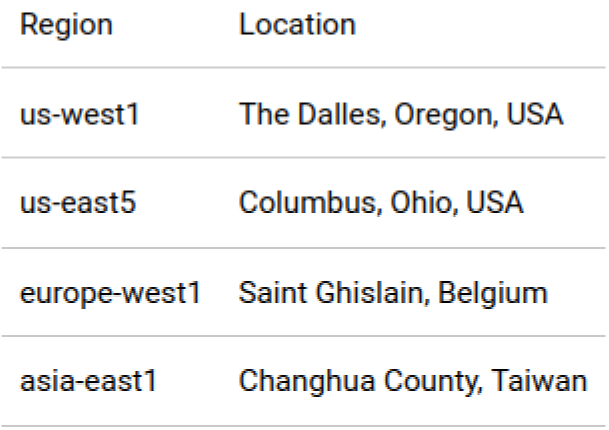


Dynamic routing mode
Regional

DNS server policy
None

□	Name	Enforcement order	Type	Deployment scope	Rule priority	Targets	Source	Destination	Protocols and ports	Action	Security
	vpc-firewall-rules	1	VPC firewall rules	Global							
□	networking-firewall-allow-icmp		Ingress firewall rule	Global	1000	Appl...	IPv4 range	–	icmp	Allow	–
□	networking-firewall-allow-internal		Ingress firewall rule	Global	1000	Appl...	IPv4 range	–	tcp:0-65535 udp:0-65535 icmp	Allow	–
□	networking-firewall-allow-ssh		Ingress firewall rule	Global	1000	Appl...	IPv4 range	–	tcp:22	Allow	–

- ssh into each VM. We'll use pings to measure latency between regions. For reference this is where our servers are located:



Region	Location
us-west1	The Dalles, Oregon, USA
us-east5	Columbus, Ohio, USA
europe-west1	Saint Ghislain, Belgium
asia-east1	Changhua County, Taiwan

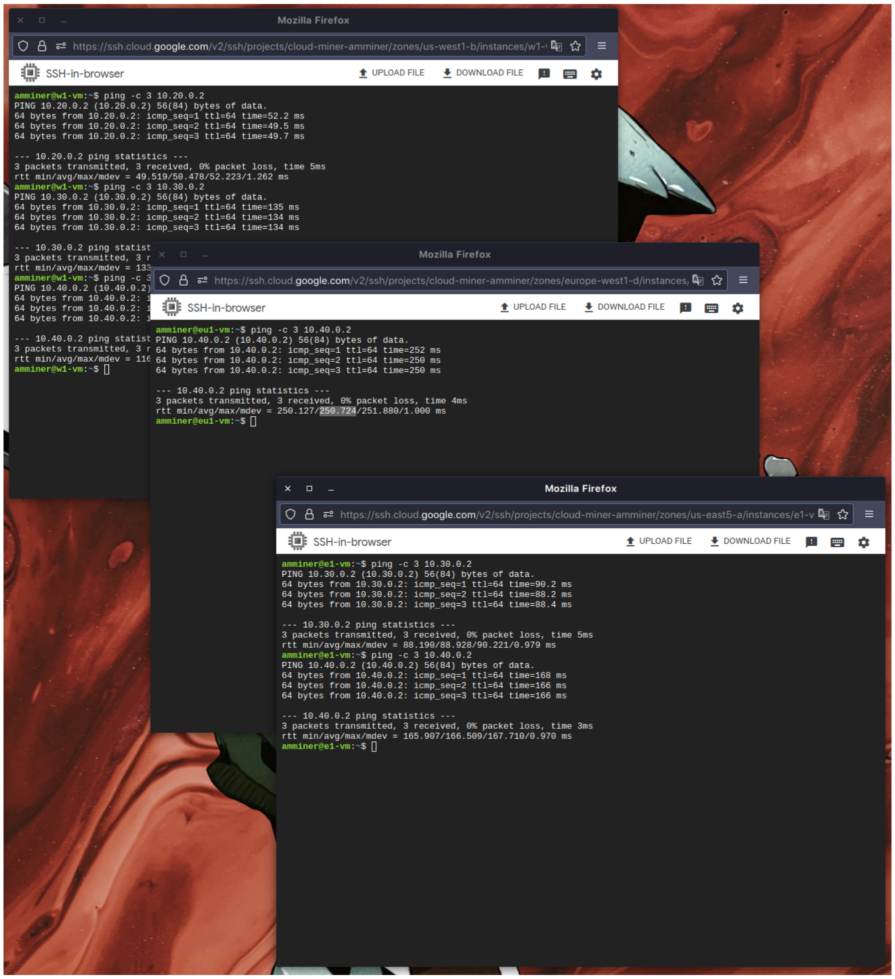
✅

9. Latency measurements

- We can determine the “ideal” latency using the speed of light and the geographic distance between the regions. Then we can measure the actual latency with a pairwise ping. Fill in the measured latency for these 6 pairs:

Location pair	ideal latency	measured latency
us-west1 us-east5	~45 ms	**50.478**
us-west1 europe-west1	~93 ms	**134.953**
us-west1 asia-east1	~114 ms	**118.328**
us-east5 europe-west1	~76 ms	**88.928**
us-east5 asia-east1	~141 ms	**166.509**
europe-west1 asia-east1	~110 ms	**250.724**

Heads up, it looks like using the target VM’s name only works when it’s in the same region as the VM pinging it… Maybe something is misconfigured?


Mozilla Firefox
https://ssh.cloud.google.com/v2/ssh/projects/cloud-miner-amminer/zones/us-west1-b/instances/w1-
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
amminer@w1-vm:~$ ping -c 3 10.20.0.2
PING 10.20.0.2 (10.20.0.2) 56(84) bytes of data.
64 bytes from 10.20.0.2: icmp_seq=1 ttl=64 time=52.2 ms
64 bytes from 10.20.0.2: icmp_seq=2 ttl=64 time=49.5 ms
64 bytes from 10.20.0.2: icmp_seq=3 ttl=64 time=49.7 ms

--- 10.20.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 49.519/50.478/52.223/1.262 ms
amminer@w1-vm:~$ ping -c 3 10.30.0.2
PING 10.30.0.2 (10.30.0.2) 56(84) bytes of data.
64 bytes from 10.30.0.2: icmp_seq=1 ttl=64 time=135 ms
64 bytes from 10.30.0.2: icmp_seq=2 ttl=64 time=134 ms
64 bytes from 10.30.0.2: icmp_seq=3 ttl=64 time=134 ms

--- 10.30.0.2 ping statist
3 packets transmitted, 3 r
rtt min/avg/max/mdev = 133
amminer@w1-vm:~$ ping -c 3
PING 10.40.0.2 (10.40.0.2)
64 bytes from 10.40.0.2: i
64 bytes from 10.40.0.2: i
64 bytes from 10.40.0.2: i

--- 10.40.0.2 ping statist
3 packets transmitted, 3 r
rtt min/avg/max/mdev = 116
amminer@w1-vm:~$ 
```

Mozilla Firefox
https://ssh.cloud.google.com/v2/ssh/projects/cloud-miner-amminer/zones/europe-west1-d/instances,
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
amminer@eu1-vm:~$ ping -c 3 10.40.0.2
PING 10.40.0.2 (10.40.0.2) 56(84) bytes of data.
64 bytes from 10.40.0.2: icmp_seq=1 ttl=64 time=252 ms
64 bytes from 10.40.0.2: icmp_seq=2 ttl=64 time=250 ms
64 bytes from 10.40.0.2: icmp_seq=3 ttl=64 time=250 ms

--- 10.40.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 4ms
rtt min/avg/max/mdev = 250.127/250.724/251.880/1.000 ms
amminer@eu1-vm:~$ 
```

Mozilla Firefox
https://ssh.cloud.google.com/v2/ssh/projects/cloud-miner-amminer/zones/us-east5-a/instances/e1-v
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
amminer@e1-vm:~$ ping -c 3 10.30.0.2
PING 10.30.0.2 (10.30.0.2) 56(84) bytes of data.
64 bytes from 10.30.0.2: icmp_seq=1 ttl=64 time=90.2 ms
64 bytes from 10.30.0.2: icmp_seq=2 ttl=64 time=88.2 ms
64 bytes from 10.30.0.2: icmp_seq=3 ttl=64 time=88.4 ms

--- 10.30.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 88.190/88.928/90.221/0.979 ms
amminer@e1-vm:~$ ping -c 3 10.40.0.2
PING 10.40.0.2 (10.40.0.2) 56(84) bytes of data.
64 bytes from 10.40.0.2: icmp_seq=1 ttl=64 time=168 ms
64 bytes from 10.40.0.2: icmp_seq=2 ttl=64 time=166 ms
64 bytes from 10.40.0.2: icmp_seq=3 ttl=64 time=166 ms

--- 10.40.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 165.907/166.509/167.710/0.970 ms
amminer@e1-vm:~$ 
```

10. Part 2: Scaling via Instance Groups and Load Balancing

- Compute Engine can use “managed instance groups” to auto-scale. A single VM template is used per group. This also requires a load balancer instance.
  ✅

11. Firewall rule for HTTP

- Instead of doing this via a deployment update, we’ll demonstrate how it can be done with the usual gcloud CLI:

```
gcloud compute firewall-rules create networking-firewall-allow-http \
  --allow tcp:80 --network networking101 --source-ranges 0.0.0.0/0 \
  --target-tags http-server
```

  ✅

12. Instance Templates

- We’ll use one template per region. Note how many layers there are to this - the template gets a startup script from a bucket which performs some installation and configuration steps including downloading index.php and substituting region information in. Create templates for us-east5 and europe-west1 like so, with the correct regions/subnets:

```
gcloud compute instance-templates create "us-east5-template" \
  --image-project debian-cloud \
  --image debian-10-buster-v20221102 \
  --machine-type e2-micro \
  --subnet "us-east5" \
  --metadata "startup-script-url=gs://cs430/networking101/website/startup.sh" \
  --region "us-east5" \
  --tags "http-server"
```

  Visit the CE console to ensure that they were created.
  ✅

13. Health check

- We’ve relied on the deployment manager and gcloud CLI tools so far. We will continue to do so, but for the remaining steps, follow along in the web console; keep in mind that anything you can do graphically can be done programmatically and vice versa, generally. We must specify a health check for managing the instance groups. This is easy since our type of application is common:

```
gcloud compute health-checks create http instance-health-check \
  --check-interval=10s \
  --port=80 \
  --timeout=5s \
  --unhealthy-threshold=3
```

  ✅

14. Managed instance group (europe-west1-mig)

- Create and configure the European group:

```
gcloud compute instance-groups managed create europe-west1-mig \
  --size 3 \
  --region europe-west1 \
  --template europe-west1-template
gcloud compute instance-groups managed update europe-west1-mig \
  --health-check instance-health-check \
  --initial-delay 120 \
  --region europe-west1
gcloud compute instance-groups set-named-ports europe-west1-mig \
  --named-ports=http:80 --region europe-west1
```

Note that this group does not auto scale.

✅

15. Managed instance group (us-east5-mig)

- Create and configure the American group:

```
gcloud compute instance-groups managed create us-east5-mig \
  --size 1 \
  --region us-east5 \
  --template us-east5-template
gcloud compute instance-groups managed update us-east5-mig \
  --health-check instance-health-check \
  --initial-delay 120 \
  --region us-east5
gcloud compute instance-groups managed set-autoscaling
us-east5-mig \
  --mode on \
  --region us-east5 \
  --min-num-replicas=1 --max-num-replicas=5 \
  --cool-down-period=45 \
  --scale-based-on-load-balancing \
  --target-load-balancing-utilization=0.8
gcloud compute instance-groups set-named-ports us-east5-mig \
  --named-ports=http:80 --region us-east5
```

Note that this group auto scales.

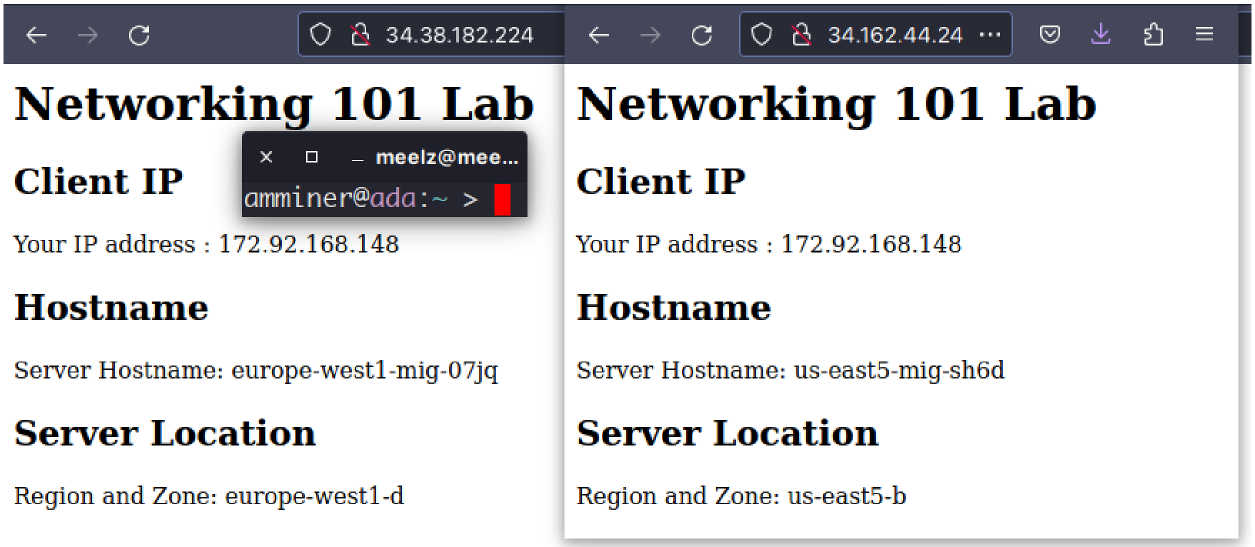
The wording at the top of this step is a little confusing since we've already created the first group.

✅

16. test groups

- Ensure that everything is as it should be in the output of `gcloud compute instance-groups list`. Wait a minute to let the dust settle then go to the CE console and visit the server in the US group. Visit the us-east instance. Repeat with an instance from europe-west1-mig. Are the instances in the same availability zone or in different ones?

  Different zones… Is this a trick question?



34.38.182.224

Networking 101 Lab

Client IP

Your IP address : 172.92.168.148

Hostname

Server Hostname: europe-west1-mig-07jq

Server Location

Region and Zone: europe-west1-d

```
amminer@ada:~ >
```

34.162.44.24

Networking 101 Lab

Client IP

Your IP address : 172.92.168.148

Hostname

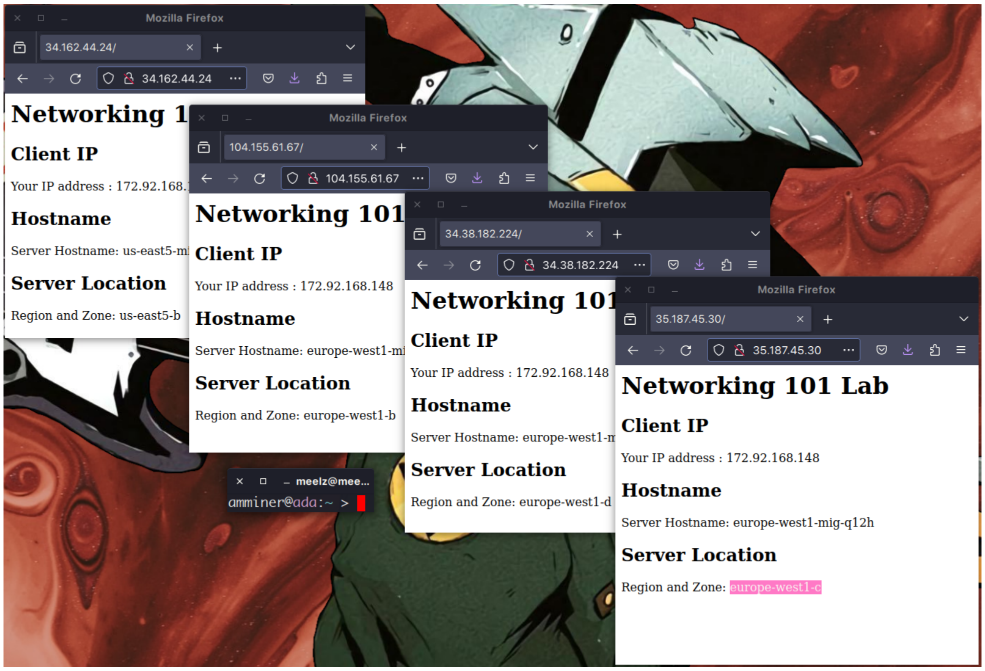
Server Hostname: us-east5-mig-sh6d

Server Location

Region and Zone: us-east5-b

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

  us-east5-b, europe-west1-b, europe-west1-d, europe-west1-c



34.162.44.24/

34.162.44.24

Networking 1

Client IP

Your IP address : 172.92.168.

Hostname

Server Hostname: us-east5-mi

Server Location

Region and Zone: us-east5-b

104.155.61.67/

104.155.61.67

Networking 101

Client IP

Your IP address : 172.92.168.148

Hostname

Server Hostname: europe-west1-mi

Server Location

Region and Zone: europe-west1-b

```
amminer@ada:~ >
```

34.38.182.224/

34.38.182.224

Networking 101

Client IP

Your IP address : 172.92.168.148

Hostname

Server Hostname: europe-west1-m

Server Location

Region and Zone: europe-west1-d

35.187.45.30/

35.187.45.30

Networking 101 Lab

Client IP

Your IP address : 172.92.168.148

Hostname

Server Hostname: europe-west1-mig-q12h

Server Location

Region and Zone: europe-west1-c

17. HTTP load balancer

- Create a backend service on the HTTP port (webserver-backend-migs) and then add the two instance groups to the service:

```
gcloud compute backend-services create webserver-backend-migs \
  --protocol=http --port-name=http --timeout=30s \
  --health-checks=instance-health-check \
  --global
gcloud compute backend-services add-backend webserver-backend-migs \
  --instance-group=europe-west1-mig --instance-group-region=europe-west1 \
  --balancing-mode=utilization --max-utilization=0.8 \
  --global
gcloud compute backend-services add-backend webserver-backend-migs \
  --instance-group=us-east5-mig --instance-group-region=us-east5 \
  --balancing-mode=rate --max-rate-per-instance=50 \
  --global
```

  then create the load balancer and point it to the backend, and create an HTTP proxy that will forward HTTP requests from the load balancer to the backend:

```
gcloud compute url-maps create webserver-frontend-lb \
  --default-service webserver-backend-migs
gcloud compute target-http-proxies create webserver-proxy \
  --url-map webserver-frontend-lb
```

  finally we allocate an IPv4 address to associate a forwarding rule with in order to take incoming HTTP requests and send them to the proxy:

```
gcloud compute addresses create webserver-frontend-ip \
  --ip-version=ipv4 --global
gcloud compute forwarding-rules create webserver-frontend-fwrule \
  --ip-protocol=tcp --ports=80 --address=webserver-frontend-ip \
  --target-http-proxy webserver-proxy \
  --global
```

  ✅

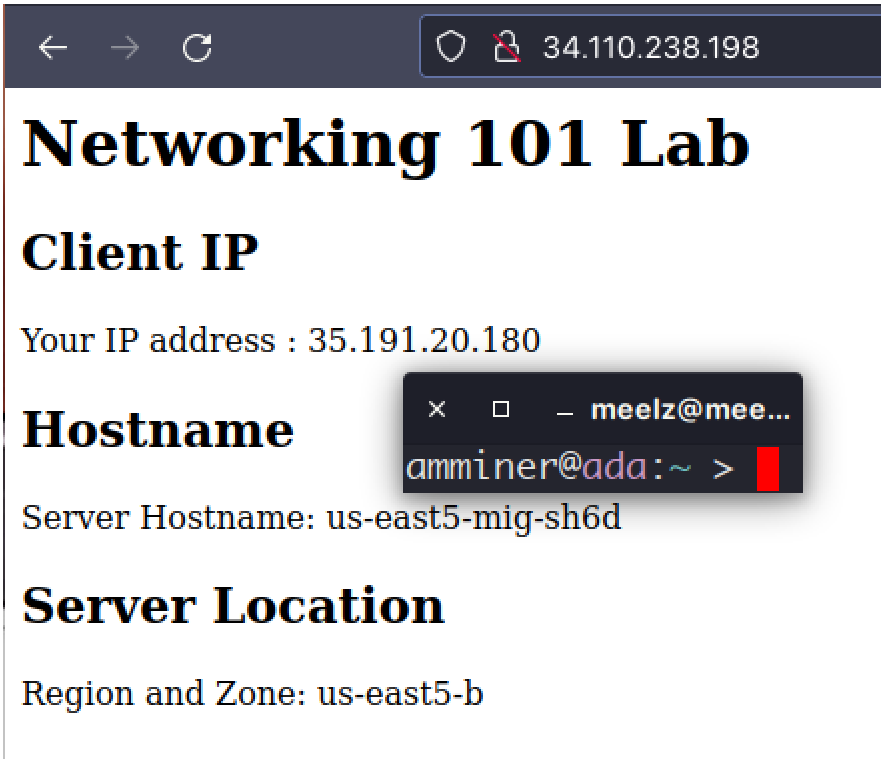
18. HTTP load balancer [sic - pt 2?]

- “The configuration via the web console is fairly involved” - **I found the commands above to be fairly involved as well, to be fair. Maybe this is just up to my inexperience in this area, but I found the web console method to be easier to follow.**
  ✅

19. Test load balancer

- Visit "Network Services"=>"Load Balancing", then click on the load balancer you instantiated. Visit the IP address that it handles requests on. Show a screenshot of the page that is returned. Which availability zone does the server handling your requests reside in?

  us-east5-b



Networking 101 Lab

Client IP

Your IP address : 35.191.20.180

Hostname

```
amminer@ada:~ >
```

Server Hostname: us-east5-mig-sh6d

Server Location

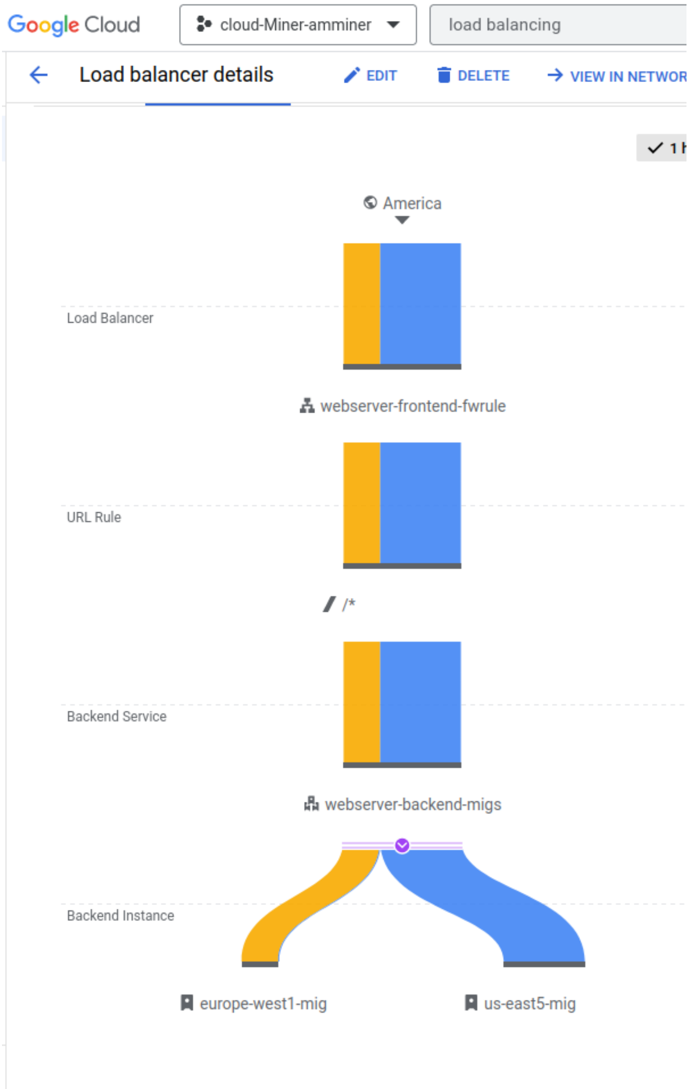
Region and Zone: us-east5-b

20. Siege! (part 1)

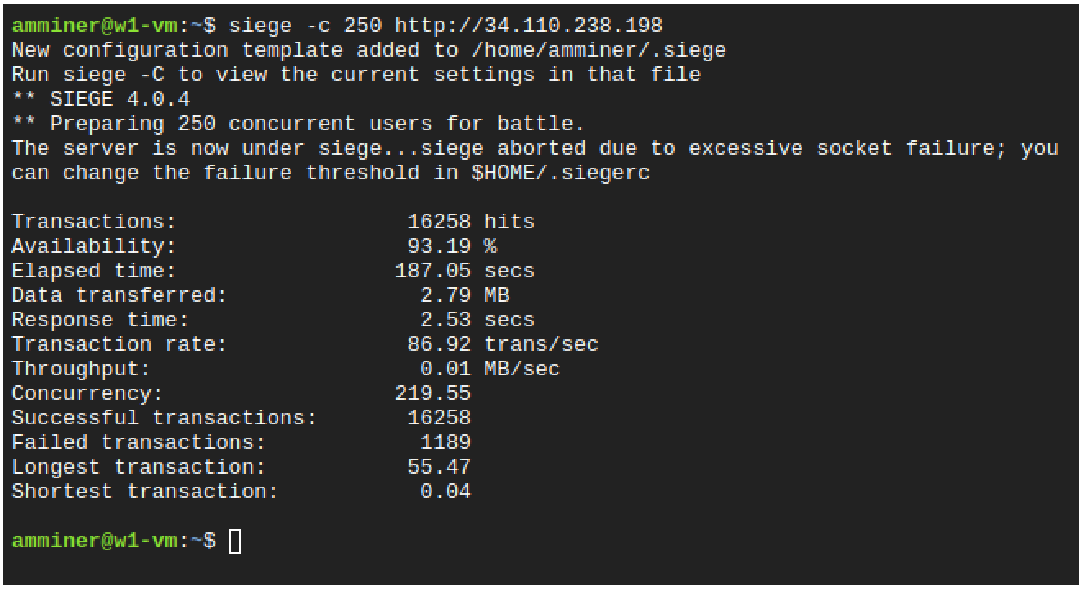
- From the original deployment manager deployment, ssh into w1-vm and eu1-vm separately. On w1-vm, launch a siege on the load balancer's IP address. 250 concurrent requests should be enough to impact auto scaling, but bump the number up if it isn't:

  ```
  siege -c 250 http://<LoadBalancerIP>
  ```

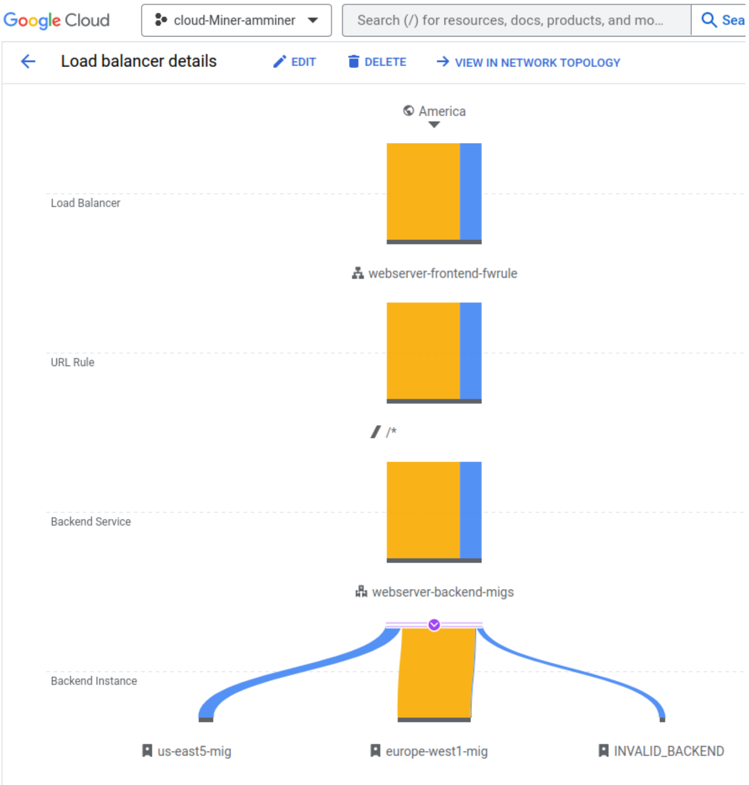
  Go to "Network Services" -> "Load Balancing" and click on your load balancer, then go to the monitoring tab. Select the backend and expand its details.
  In the console you should see the initial state of the system with eu-west1 receiving significant traffic since it already has a few idle instances and the single instance in us-east can't handle all the traffic we're generating.
  This part of the lab really, really did not behave as expected for me. The load balancer monitoring GUI took a long time to update. By the time it did, when I was expecting the initial state described above, I got something that looks more like the final state that's supposed to come into being after waiting for the system to adapt:



- Keep things spinning for 5-10 minutes until the system adapts. Show the UI in this state where more us-based instances have been brought up. Note that eventually most of the traffic is handled by the 5 vms in us-east5-mig (the maximum we configured).
  By the time I had waited this long, the siege had ended (siege aborted due to excessive socket failure), and I had something that looked more like what I was hoping for in the initial state:

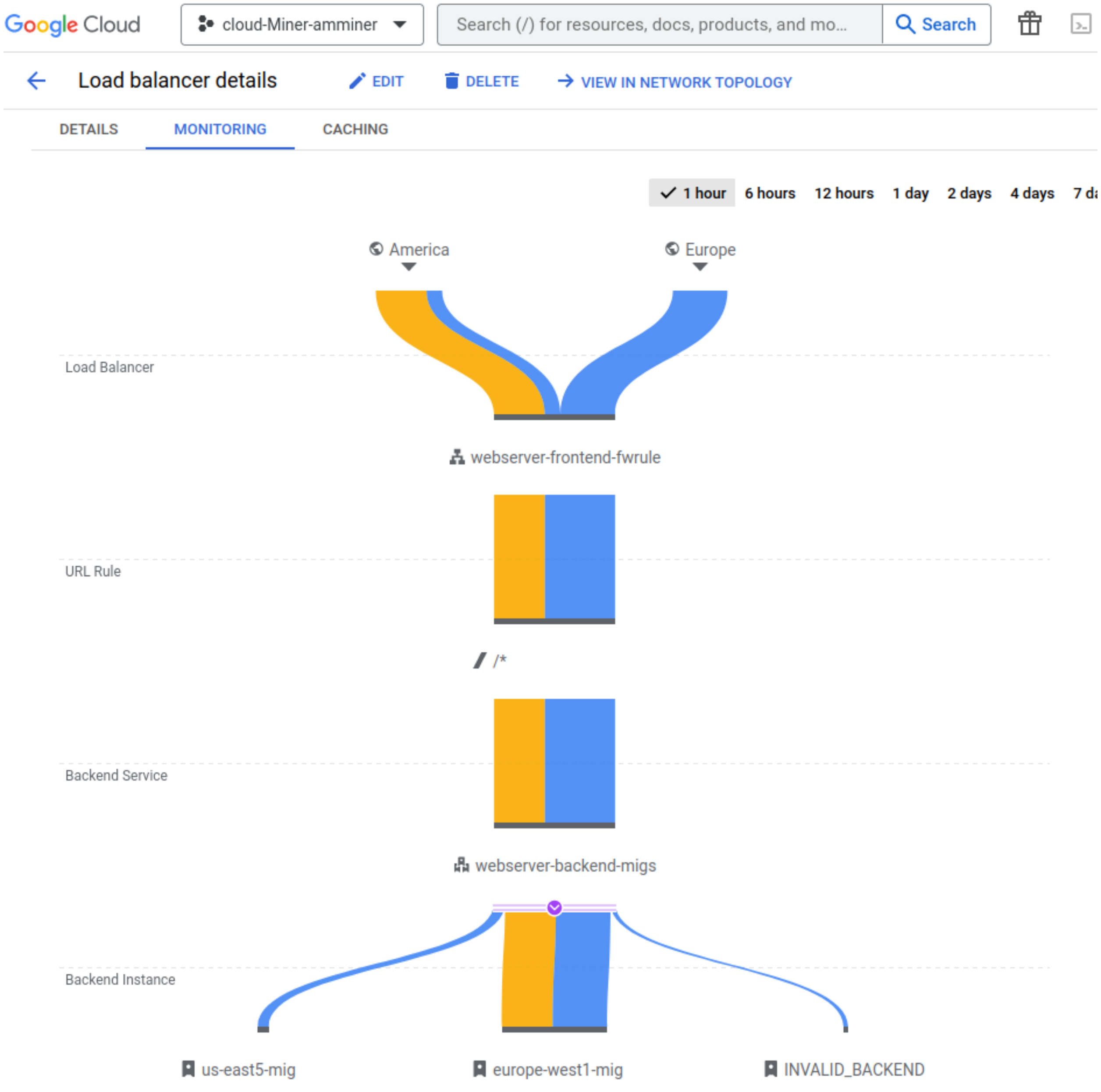


```
amminer@w1-vm:~$ siege -c 250 http://34.110.238.198
New configuration template added to /home/amminer/.siege
Run siege -C to view the current settings in that file
** SIEGE 4.0.4
** Preparing 250 concurrent users for battle.
The server is now under siege...siege aborted due to excessive socket failure; you
can change the failure threshold in $HOME/.siegerc

Transactions:                  16258 hits
Availability:                  93.19 %
Elapsed time:                 187.05 secs
Data transferred:               2.79 MB
Response time:                  2.53 secs
Transaction rate:              86.92 trans/sec
Throughput:                     0.01 MB/sec
Concurrency:                  219.55
Successful transactions:       16258
Failed transactions:            1189
Longest transaction:           55.47
Shortest transaction:           0.04

amminer@w1-vm:~$ 
```



21. Siege! (part 2)

- Stop the siege running on w1-vm. Then, go to eu1-vm and launch an identical siege on your load balancer IP address…
  Heads up, the command here uses a different -c parameter, so calling it identical is a little confusing. The first sentence in the paragraph following the code box also appears to have been mangled in the writing process in a way that I'm all too familiar with from my own technical writing.
  Show a screenshot of the final traffic distribution.



🤷‍♀️

22. clean-up

- Delete the load balancing setup:

  ```
  gcloud compute forwarding-rules delete webserver-frontend-fwrule --global
  gcloud compute target-http-proxies delete webserver-proxy
  gcloud compute addresses delete webserver-frontend-ip --global
  gcloud compute url-maps delete webserver-frontend-lb
  gcloud compute backend-services delete webserver-backend-migs --global
  ```

  The managed instance groups:

  ```
  gcloud compute instance-groups managed delete us-east5-mig --region us-east5
  gcloud compute instance-groups managed delete europe-west1-mig --region europe-west1
  ```

  The health checks, instance templates, and the HTTP firewall rule:

  ```
  gcloud compute health-checks delete instance-health-check
  gcloud compute instance-templates delete us-east5-template
  gcloud compute instance-templates delete europe-west1-template
  gcloud compute firewall-rules delete networking-firewall-allow-http
  ```

  and finally the initial deployment:

  ```
  gcloud deployment-manager deployments delete networking101
  ```

  ✅