CS530 F23 Week 7 Lab
Prof. Feng

Amelia Miner
11/12/23

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

TABLE OF CONTENTS:

**I. Lab 7.1a: Terraform AWS Guestbook ..... 3**
1. Terraform ..... 3
2. Setup ..... 3
3. Initial Configuration ..... 3
4. Launching Configuration ..... 3
5. Adding Network Access ..... 4
6. Adding ssh access ..... 4
7. Adding the guestbook application ..... 4
8. View the guestbook ..... 4
9. Clean up ..... 4
**II. Lab 7.1g: Terraform GCP Guestbook ..... 4**
1. Terraform ..... 4
2. Setup ..... 4
3. Initial Configuration ..... 4
4. Launching Configuration ..... 5
5. Adding an external IP Address ..... 6
6. Adding ssh access ..... 6
7. Adding the guestbook application ..... 7
8. View the Guestbook ..... 8
9. Clean up ..... 8
**III. Lab 7.2g: Kubernetes Guestbook ..... 9**
1. Kubernetes ..... 9
2. Setup ..... 9
3. Assigning Privileges ..... 9
4. Create Kubernetes Cluster ..... 9
5. Prepare a container image ..... 11
6. kubernetes.yaml ..... 11
7. Deploy the configuration ..... 12
8. view the guestbook ..... 13
9. delete workload and service ..... 15
10. CI/CD build automation ..... 16
11. Configure build automation ..... 16
12. Deploy and view application ..... 17
13. Clean up ..... 18
**IV. Lab 7.3g: APIs (Slack, Knowledge Graph) ..... 18**
1. Slack and Knowledge Graph Integration ..... 18
2. Code ..... 18
3. Code ..... 18

4. Knowledge Graph Setup....20
5. Create a Slack Workspace....20
6. Configure and Deploy....20
7. Create Slack Command....20
8. Test the command....21
**V. Lab 7.4g: ML APIs....21**

Amelia Miner
11/12/23

# I. Lab 7.1a: Terraform AWS Guestbook

## 1. Terraform

✅

## 2. Setup

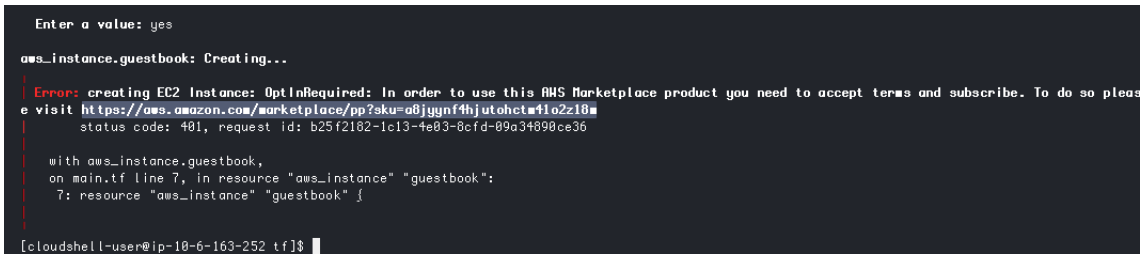
- In AWS cloud shell, install yum-utils, add the hashicorp repo, and install terraform. Make a directory for your terraform deployment and cd in. ✅

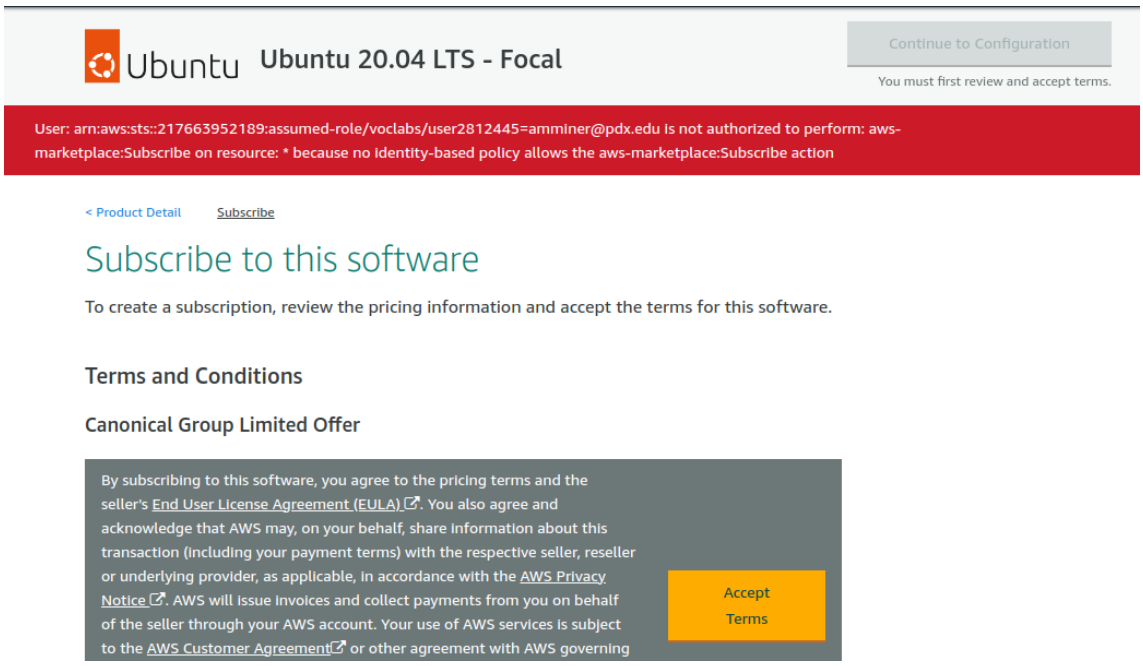
## 3. Initial Configuration

- Examine and copy over the main.tf file contents from the lab. ✅

## 4. Launching Configuration

- init, plan, and apply the terraform configuration. Show the output of the appliy command. **I can't complete this task due to what looks like a permissions misconfiguration for my aws account. I do not have control over this permission as far as I can tell. I'm moving on to the next lab for now.**



Amelia Miner


5. Adding Network Access
- todo

6. Adding ssh access
- todo

7. Adding the guestbook application
- todo

8. View the guestbook
- todo

9. Clean up
- todo

# II. Lab 7.1g: Terraform GCP Guestbook

1. Terraform

   ✅

2. Setup

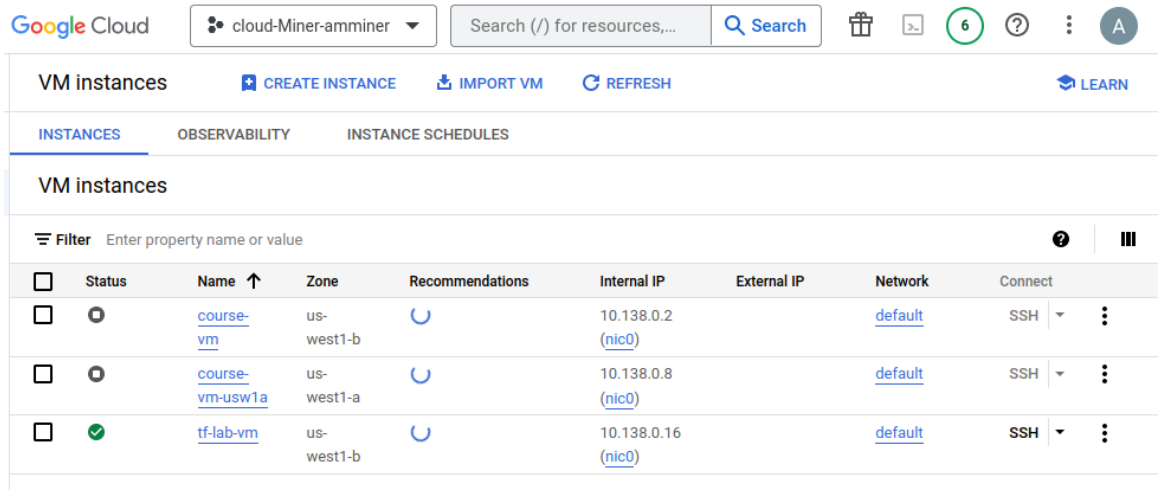
   ✅

3. Initial Configuration
- Copy the main.tf code. Substitute in the fmi and your project name. ✅

Amelia Miner
11/12/23

# 4. Launching Configuration

- Init, plan and apply. Show the VM's IP addresses in the CE web console.



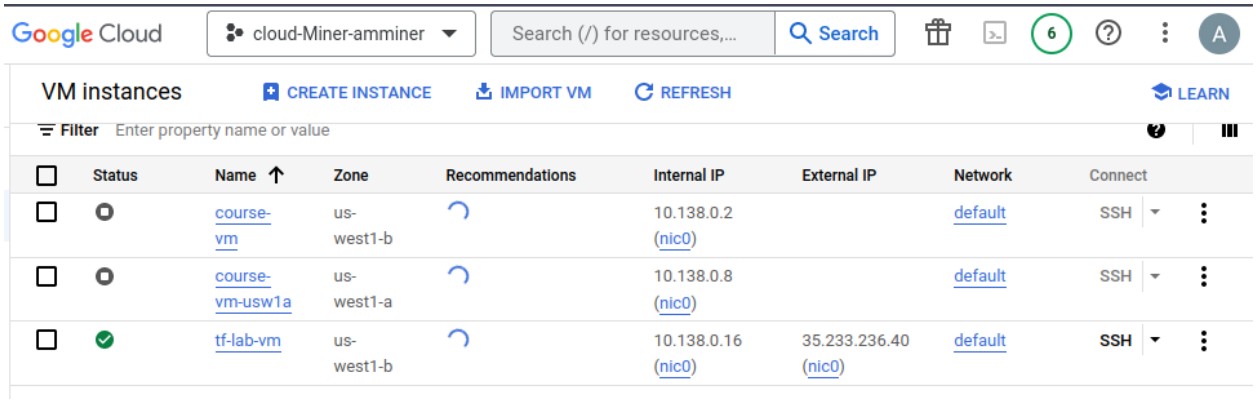
Amelia Miner
11/12/23

# 5. Adding an external IP Address

- Copy the code in from the lab to add an external IP to the machine. Plan. Apply. Show the output.

```
Changes to Outputs:
  + ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

google_compute_address.static: Creating...
google_compute_address.static: Still creating... [10s elapsed]
google_compute_address.static: Creation complete after 11s [id=projects/cloud-miner-amminer/regions/us-west1/addresses/ipv4-addr
ss]
google_compute_instance.default: Modifying... [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf-lab-vm]
google_compute_instance.default: Still modifying... [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf-lab-vm, 10s e
apsed]
google_compute_instance.default: Modifications complete after 11s [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf
lab-vm]

Apply complete! Resources: 1 added, 1 changed, 0 destroyed.

Outputs:

ip = "35.233.236.40"
amminer@cloudshell:~/tf (cloud-miner-amminer)$
```

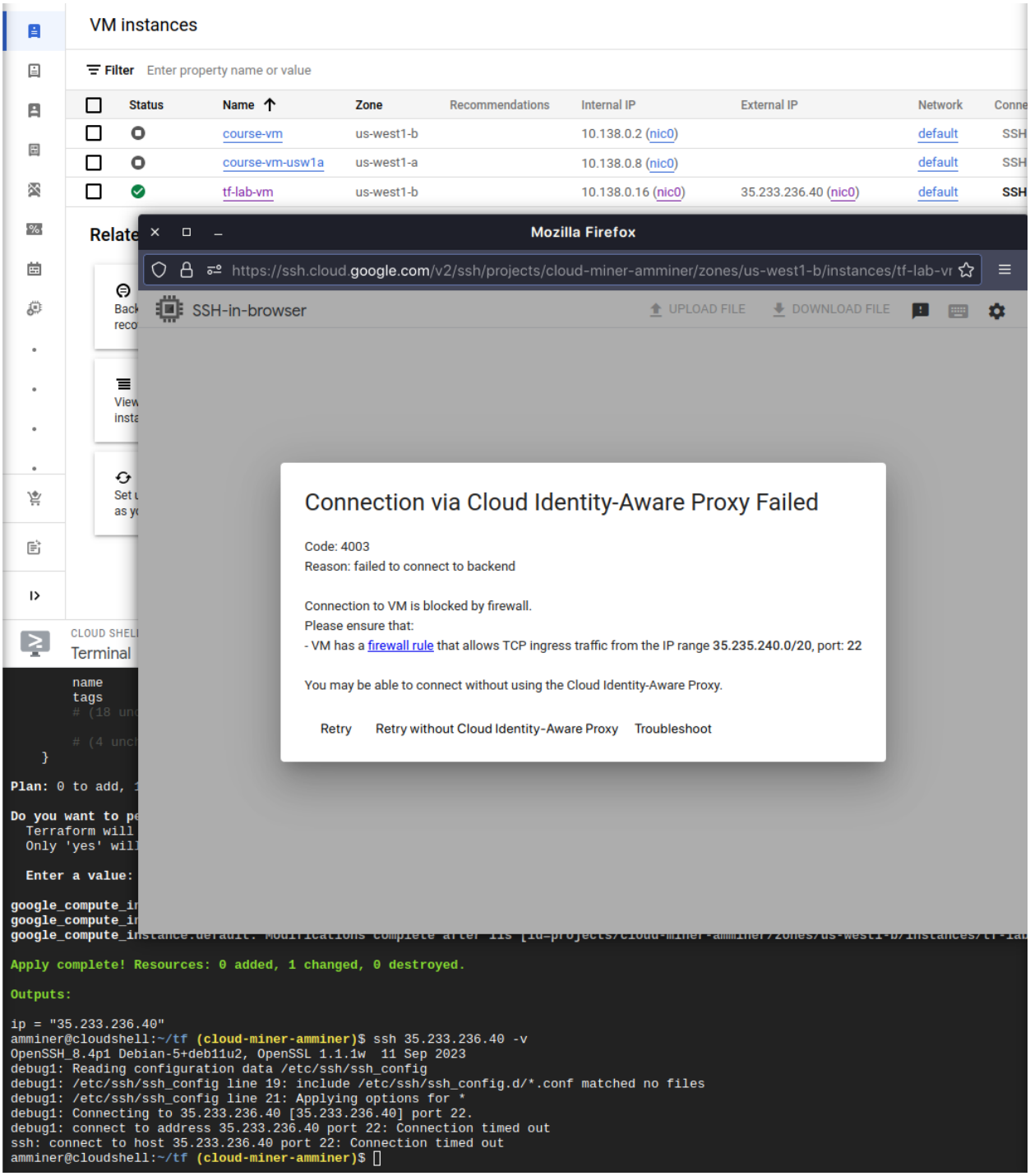
- Visit the Compute Engine web console and refresh it to see that the IP address has been bound to the VM. Take a screenshot that includes the VM's IP addresses.




# 6. Adding ssh access

- Create ssh keys for the cloud console and add the key file to the terraform config. ssh in from cloud shell and show the successful log in.
  **I can' access the machine. Any attempt I make to ssh in eventually times out. Google cloud's GUI says I'm being blocked by the machine's firewall but there's a rule in place to allow ssh. When I look at the VM details in CE I see that the ssh keys were copied over successfully but I still can't log in. It appears to be some kind of backend issue with google cloud. Moving on to the next lab for now.**

Amelia Miner
11/12/23




## 7. Adding the guestbook application

todo

Amelia Miner


## 8. View the Guestbook

todo

## 9. Clean up

todo

Amelia Miner


# III. Lab 7.2g: Kubernetes Guestbook

## 1. Kubernetes

✅

## 2. Setup

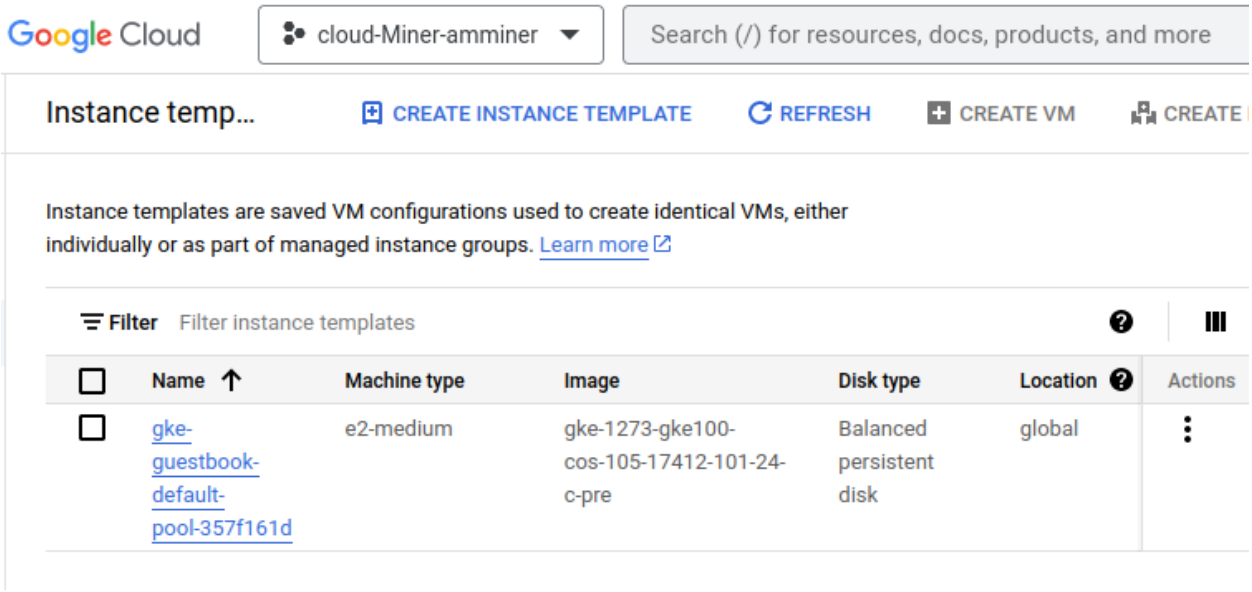
✅

## 3. Assigning Privileges

✅

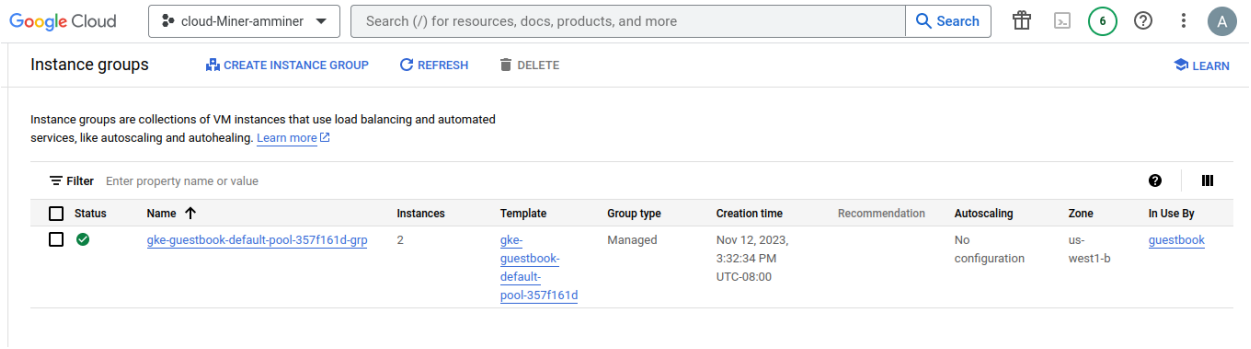
## 4. Create Kubernetes Cluster

- What is the name of the Instance Template dynamically generated to create the two nodes (VMs)?
  **gke-guestbook-default-pool-357f161d**

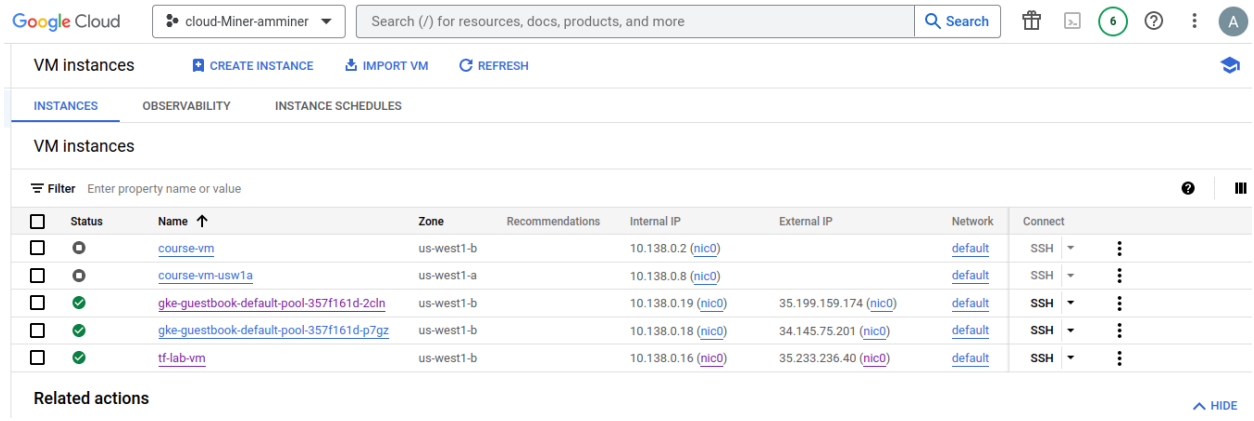



- What is the name of the Instance Group dynamically generated that the two nodes belong to?
  **gke-guestbook-default-pool-357f161d-grp**

Amelia Miner
11/12/23

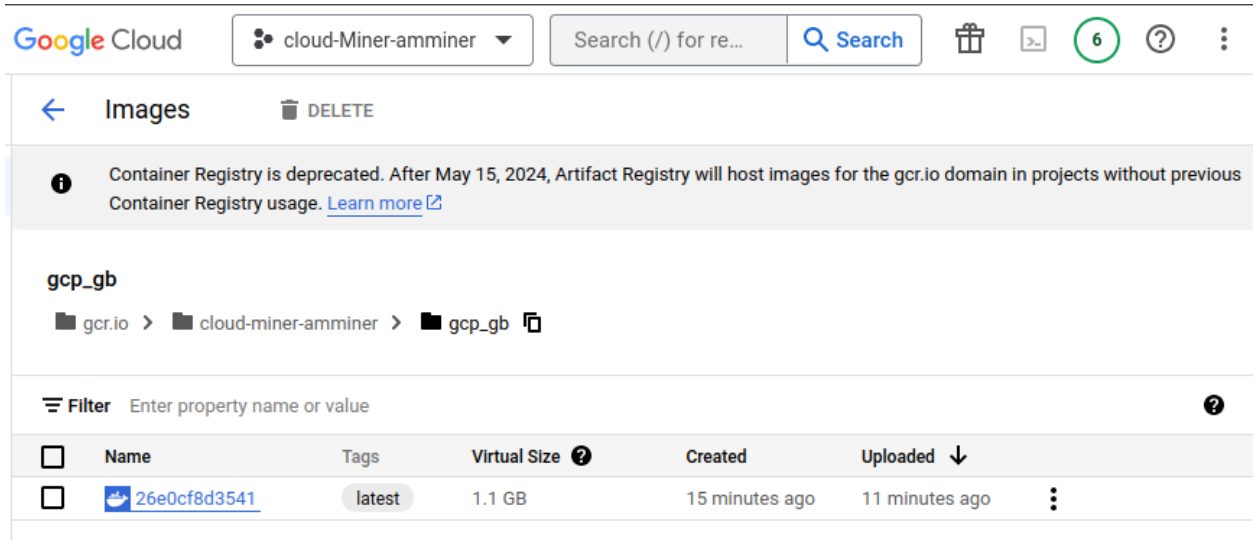



- What are the names of the two nodes?
  **gke-guestbook-default-pool-357f161d-2cln and gke-guestbook-default-pool-357f161d-p7gz**




# 5. Prepare a container image

- Visit the Artifact Registry UI and navigate to its container images section. Show the container image created



Amelia Miner


## 6. kubernetes.yaml

- Copy over the kubernetes configuration yaml, editing in your project ID. ✅

## 7. Deploy the configuration

- Obtain credentials from the cluster:
  `gcloud container clusters get-credentials guestbook --zone us-west1-b`
  Then deploy the configuration on the cluster.
  `kubectl create -f kubernetes.yaml`
  Note that the configuration file is portable and can run on any other cloud provider, locally, etc. Get the status of the pods running in the cluster.
  `kubectl get pods`
  Take a screenshot of the output of the following command when all 3 replicas reach a "Running" state.
  **When I did this in the order the lab instructs, it appeared to run successfully but failed to bring up my pods. When I did it manually they failed at the container image pull stage. I inverted the order of the declarations so that the Service is specified at the top of the file and the ReplicationController at the bottom. This worked.**

```
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl create -f kubernetes.yaml
replicationcontroller/guestbook-replicas created
amminer@cloudshell:~ (cloud-miner-amminer)$ kubernets get pods
bash: kubernets: command not found
amminer@cloudshell:~ (cloud-miner-amminer)$ ^Cbectl create -f kubernetes.yaml
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get pods
NAME                       READY   STATUS              RESTARTS   AGE
guestbook-replicas-khdjf   0/1     ContainerCreating   0          35s
guestbook-replicas-pz2ps   0/1     ContainerCreating   0          35s
guestbook-replicas-vrws9   0/1     ContainerCreating   0          35s
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
guestbook-replicas-khdjf   1/1     Running   0          2m20s
guestbook-replicas-pz2ps   1/1     Running   0          2m20s
guestbook-replicas-vrws9   1/1     Running   0          2m20s
amminer@cloudshell:~ (cloud-miner-amminer)$
```

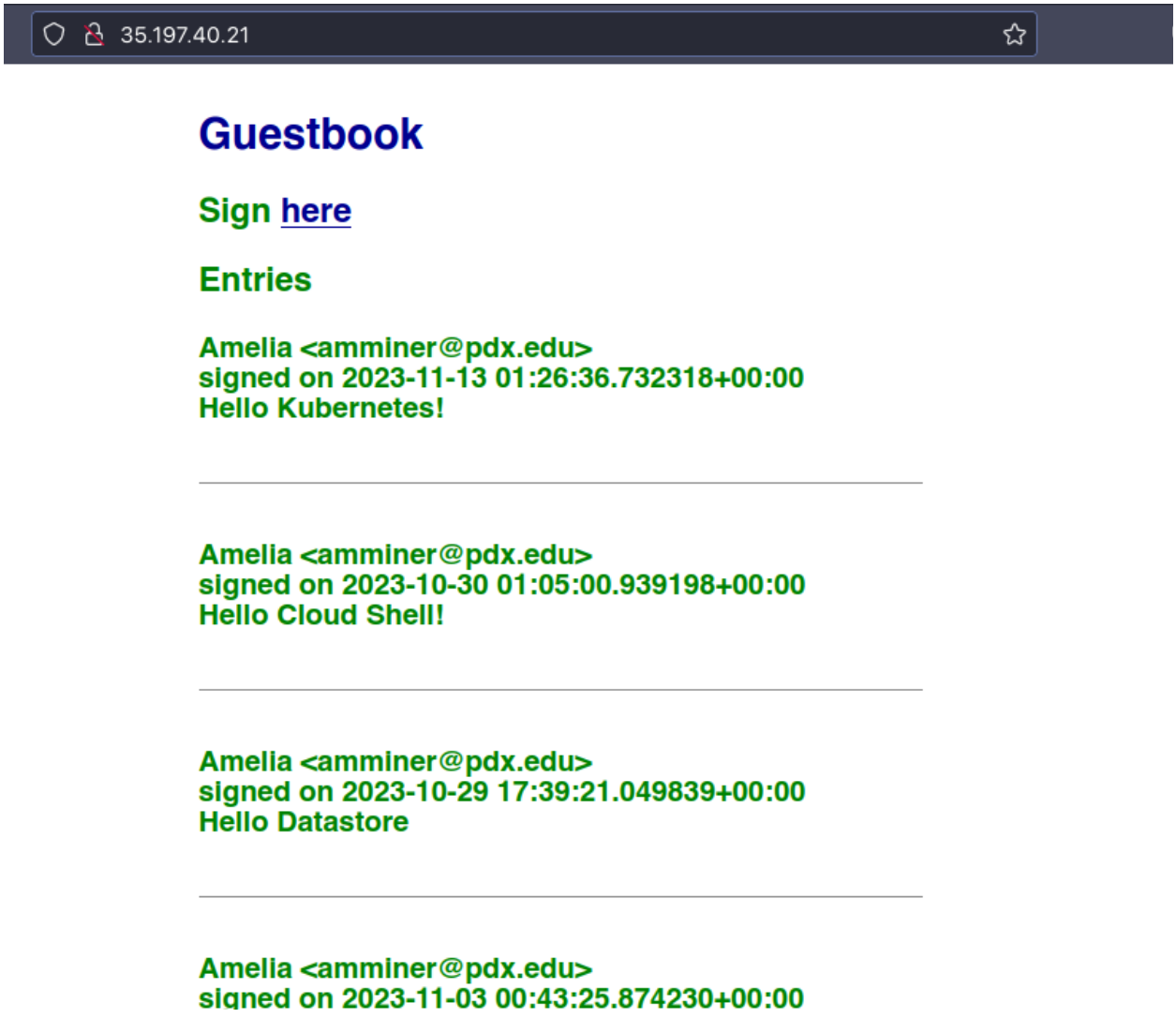
- Then, find the service that is exported from the deployment.
  `kubectl get services`
  Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.

```
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get services
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
guestbook-lb   LoadBalancer   10.20.12.144   35.197.40.21   80:31133/TCP   17m
kubernetes     ClusterIP      10.20.0.1      <none>         443/TCP        106m
amminer@cloudshell:~ (cloud-miner-amminer)$
```

Amelia Miner

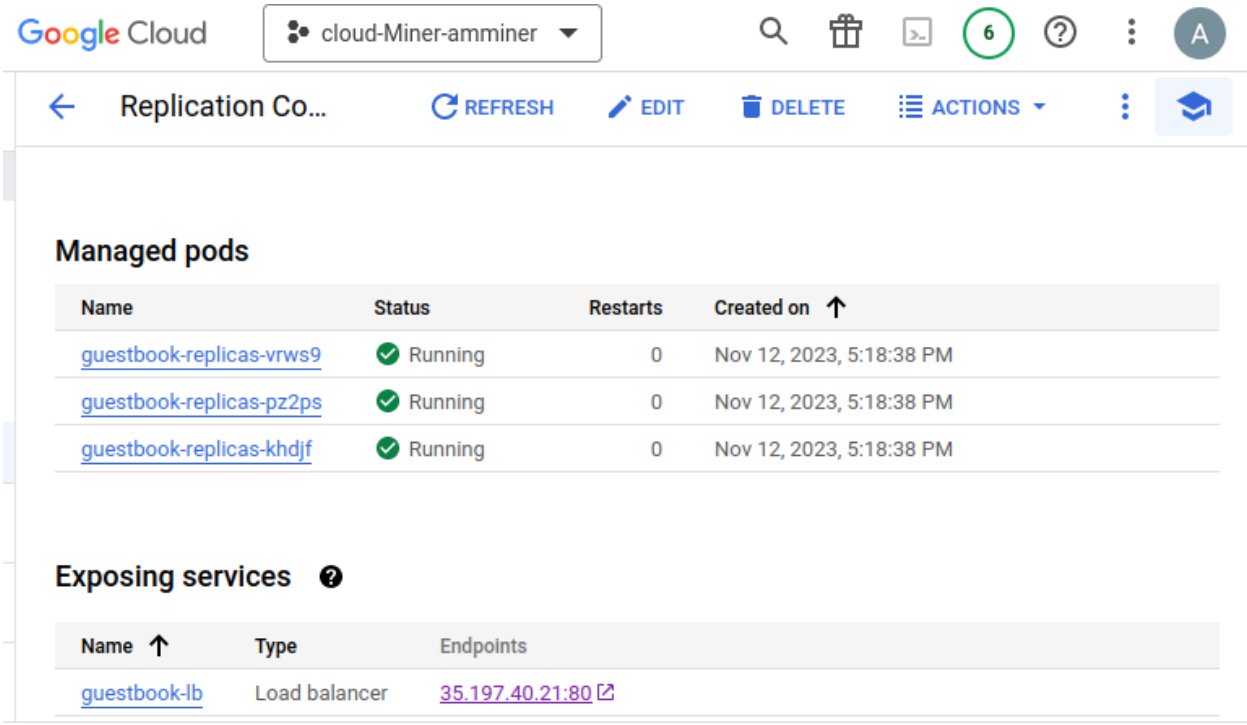

## 8. view the guestbook

- Bring the guestbook up at its external IP and sign it "Hello Kubernetes!".

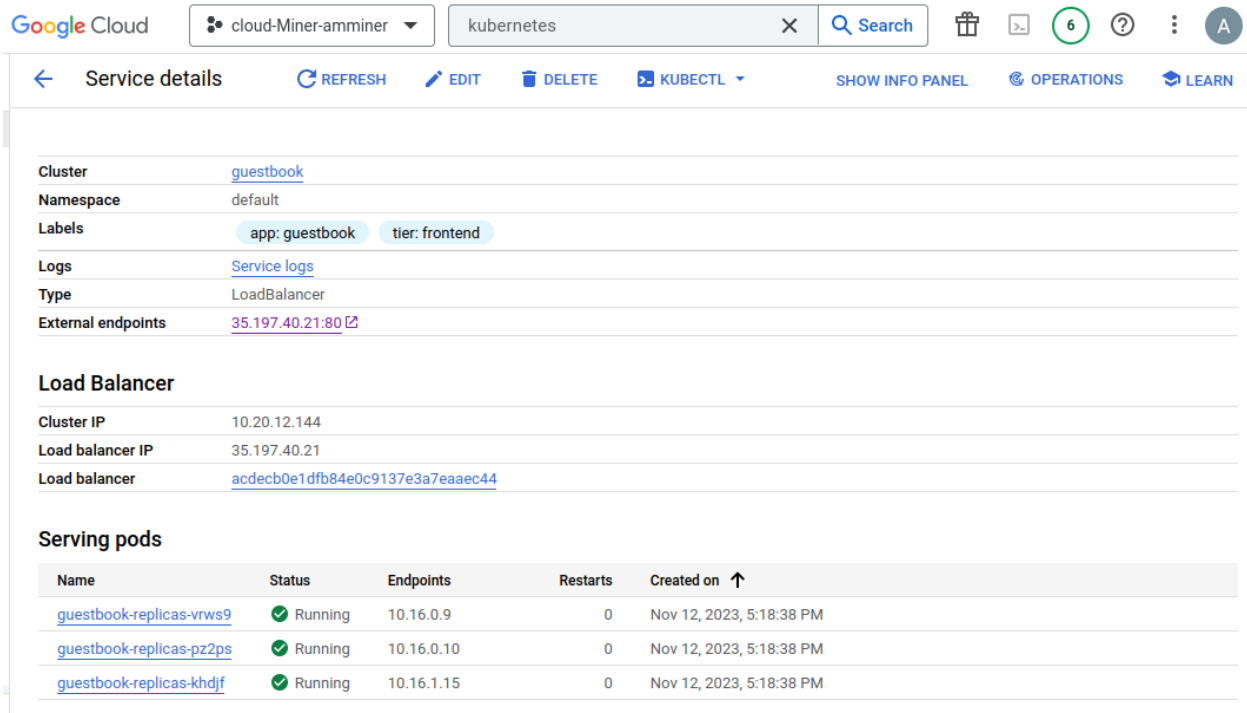


Amelia Miner
11/12/23

- View the resources kubernetes has deployed across your cloud project:
  - Take a screenshot of the managed guestbook pods and the service being exposed:

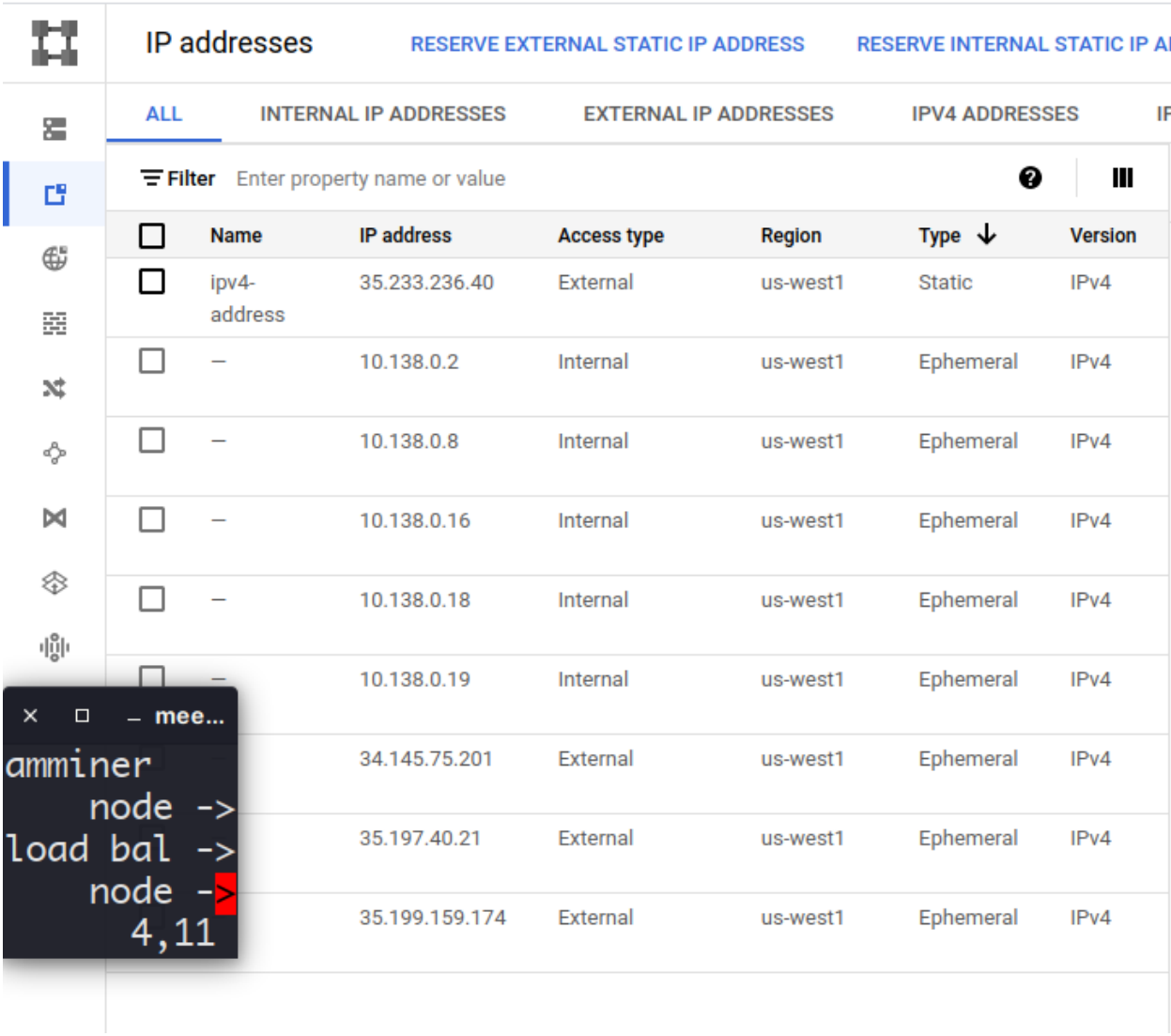



  - Take a screenshot of the load balancer and its details:



Amelia Miner
11/12/23

- Take a screenshot of the addresses allocated and indicate the ones associated with nodes versus the load balancer.




## 9. delete workload and service

```
kubectl delete -f kubernetes.yaml
gcloud container images delete
gcr.io/${GOOGLE_CLOUD_PROJECT}/gcp_gb
```

✅

Amelia Miner


## 10. CI/CD build automation

- Run:
  ```
  GOOGLE_CLOUD_PROJECT_NUMBER=$(gcloud projects describe
  $GOOGLE_CLOUD_PROJECT --format="value(projectNumber)")

  echo $GOOGLE_CLOUD_PROJECT_NUMBER

  gcloud projects add-iam-policy-binding ${GOOGLE_CLOUD_PROJECT}
  --member
  serviceAccount:${GOOGLE_CLOUD_PROJECT_NUMBER}@cloudbuild.gservice
  account.com --role=roles/container.developer
  ```

  ✅

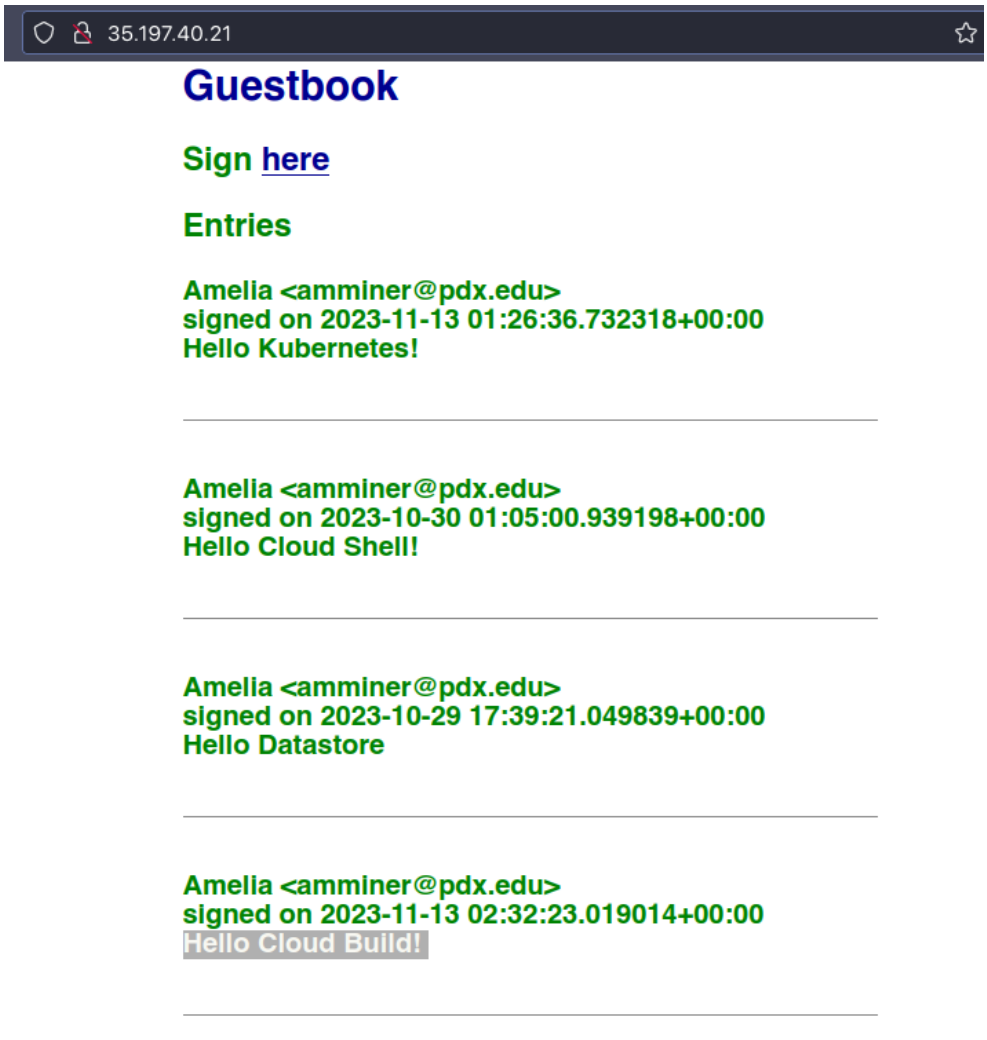
## 11. Configure build automation

- In cloud shell create cloudbuild.yaml from the code in the lab.
  ✅

Amelia Miner
11/12/23

## 12. Deploy and view application

- Build and deploy the app:
  `gcloud builds submit --config=cloudbuild.yaml`
  Visit the load balancer and sign the guest book “Hello Cloud Build!”.




## 13. Clean up

- `kubectl delete -f kubernetes.yaml`
- `gcloud container images delete gcr.io/${GOOGLE_CLOUD_PROJECT}/gcp_gb`
  ✅

Amelia Miner
11/12/23

# IV. Lab 7.3g: APIs (Slack, Knowledge Graph)

## 1. Slack and Knowledge Graph Integration

We're setting up a slack command /kg to query google's knowledge graph API on demaind via a cloud function. The slack client sends a payload + token to the cloud function's trigger endpoint. The function verifies the token and sends a request to the knowledge graph API + an API key.

- `git clone https://github.com/GoogleCloudPlatform/python-docs-samples.git`
- `cd python-docs-samples/functions/slack/`

✅

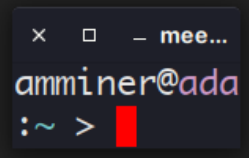
## 2. Code

- Does Google provide a Python package specifically for accessing the Knowledge Graph API?
  **No. Google provides the googleapiclient.discovery module as a generic interface to all of their backend APIs. It uses parameters to its build function to determine which API to instantiate.**

## 3. Code

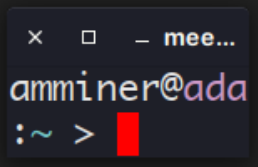
- Show the source line that constructs the query we wish to send to the Knowledge Graph API.

```
81
82      return message
83
84
85  # [END functions_slack_format]
86
87
88  # [START functions_slack_request]
89  def make_search_request(query):
90      req = kgsearch.entities().search(query=query, limit=1)
91      res = req.execute()
92      return format_slack_message(query, res)
93
94
95  # [END functions_slack_request]
96
97
98  # [START functions_slack_search]
99  @functions_framework.http
100 def kg_search(request):
```



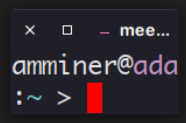

- Show the source line that then executes the query and saves the response. What is the name of the method that sends the query to the Knowledge Graph API?
  **the execute method of the knowledge graph search query object, whose name is req in the program, sends the query to the API.**

```
83
84
85 # [END functions_slack_format]
86
87
88 # [START functions_slack_request]
89 def make_search_request(query):
90     req = kgsearch.entities().search(query=query, limit=1)
91     res = req.execute()
92     return format_slack_message(query, res)
93
94
95 # [END functions_slack_request]
96
97
98 # [START functions_slack_search]
99 @functions_framework.http
```




- What is the Python data type that is used to represent the formatted message?
  **A dictionary.**

```
42 # [START functions_slack_format]
43 def format_slack_message(query, response):
44     entity = None
45     if (
46         response
47         and response.get("itemListElement") is not None
48         and len(response["itemListElement"]) > 0
49     ):
50         entity = response["itemListElement"][0]["result"]
51
52     message = {
53         "response_type": "in_channel",
54         "text": f"Query: {query}",
55         "attachments": [],
56     }
57
58     attachment = {}
59     if entity:
60         name = entity.get("name", "")
61         description = entity.get("description", "")
62         detailed_desc = entity.get("detailedDescription", {})
63         url = detailed_desc.get("url")
64         article = detailed_desc.get("articleBody")
65         image_url = entity.get("image", {}).get("contentUrl")
66
67         attachment["color"] = "#3367d6"
68         if name and description:
69             attachment["title"] = "{}: {}".format(entity["name"], entity["description"])
70         elif name:
71             attachment["title"] = name
72         if url:
73             attachment["title_link"] = url
74         if article:
75             attachment["text"] = article
76         if image_url:
77             attachment["image_url"] = image_url
78     else:
79         attachment["text"] = "No results match your query."
80     message["attachments"].append(attachment)
81
82     return message
83
84
85 # [END functions_slack_format]
86
87
88 # [START functions_slack_request]
89 def make_search_request(query):
90     req = kgsearch.entities().search(query=query, limit=1)
91     res = req.execute()
92     return format_slack_message(query, res)
93
94
95 # [END functions_slack_request]
96
```




- What are the three main attributes of the formatted message passed back to Slack?
  **response_type, text, and attachments… see above.**

Amelia Miner


## 4. Knowledge Graph Setup

- Enable the KG API and issue an API key.

```
gcloud services enable kgsearch.googleapis.com
gcloud alpha services api-keys create \
  --display-name="KG API Key" \
  --api-target=service=kgsearch.googleapis.com
```

✅

## 5. Create a Slack Workspace

- Also create a slack app and associate it with your workspace. Obtain the slack app's signing secret: basic info -> app credentials -> show. Keep this page up.

✅

## 6. Configure and Deploy

- in Cloud Shell, deploy the code substituting your kg key and slack secret. Note the URL of the function endpoint.

✅

**https://us-central1-cloud-miner-amminer.cloudfunctions.net/kg_search**

## 7. Create Slack Command

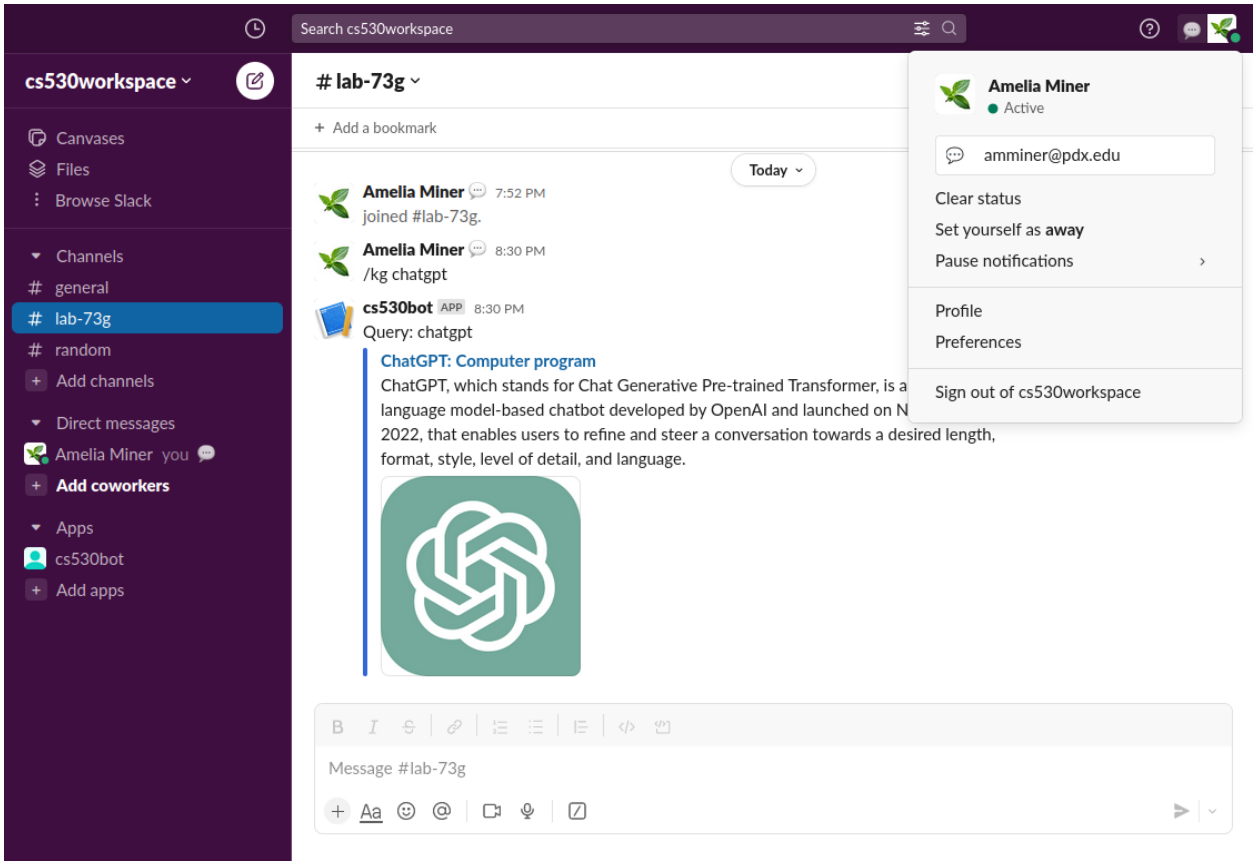
- Create the /kg command and connect it to the above endpoint. Install the app on your workspace.

✅

Amelia Miner
11/12/23

## 8. Test the command

- Run the command /kg chatgpt in your workspace.




- Visit Cloud Shell and examine the logs for the function. Delete the function.
  ✅

# V. Lab 7.4g: ML APIs