07.1a: Terraform AWS Guestbook

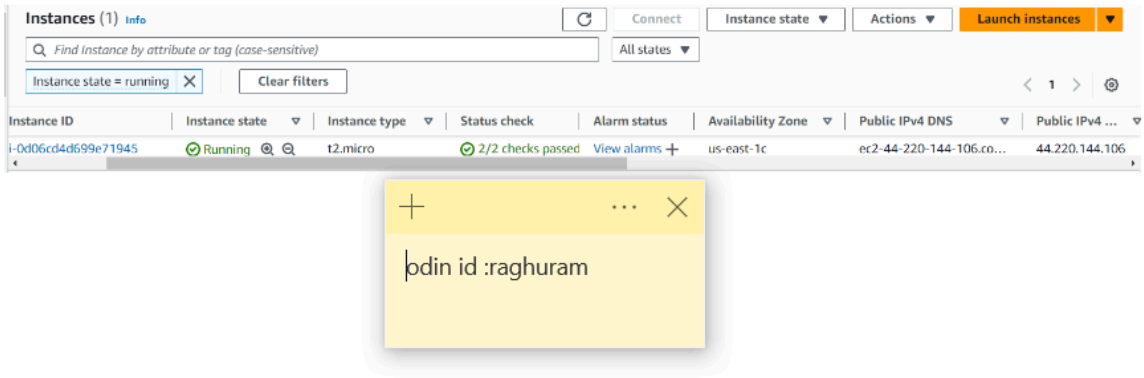
4. Launching configuration

- Take a screenshot showing the completion of the command including its output

```
      + subnet_id                            = (known after apply)
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + ec2instance = (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

aws_instance.guestbook: Creating...
aws_instance.guestbook: Still creating... [10s elapsed]
aws_instance.guestbook: Still creating... [20s elapsed]
aws_instance.guestbook: Still creating... [30s elapsed]
aws_instance.guestbook: Creation complete after 32s [id=i-0d06cd4d699e71945]

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

Outputs:

ec2instance = "44.220.144.106"
```

pdin id :raghuram

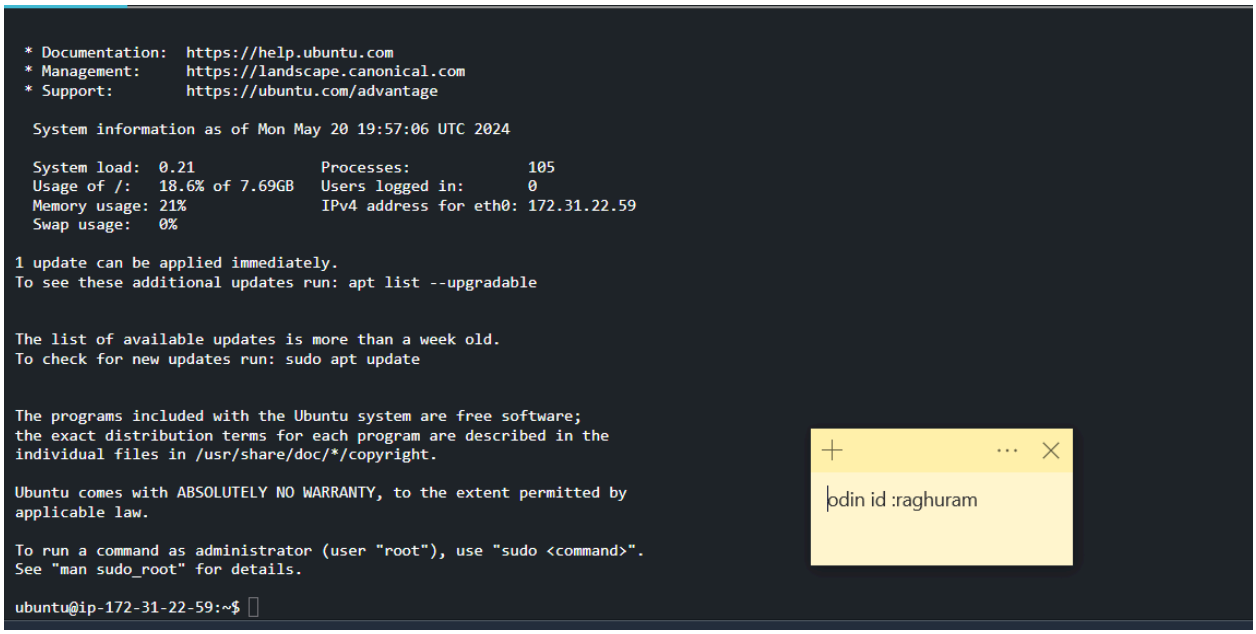
- Take a screenshot that includes the VM's IP addresses



pdin id :raghuram

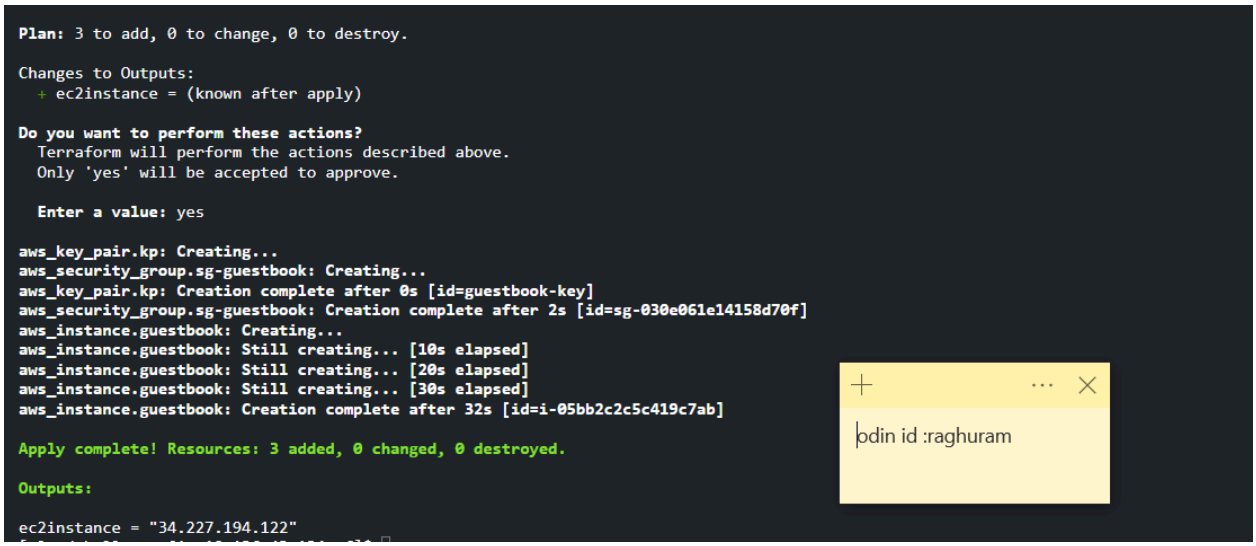
6. Adding ssh access

- Take a screenshot of the successful ssh login from Cloud Shell.



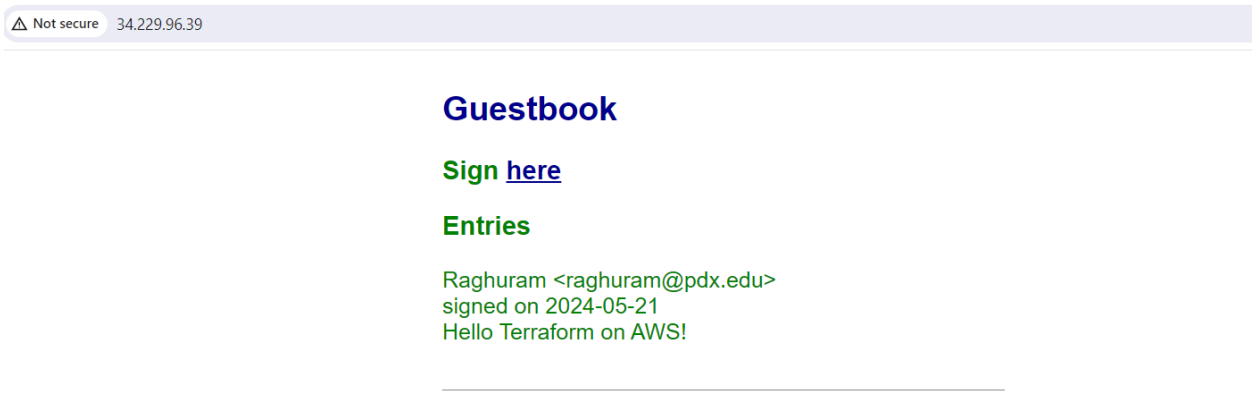
7. Adding the Guestbook application

- Take a screenshot of the output of the command that includes the IP address of the instance



8. View the Guestbook

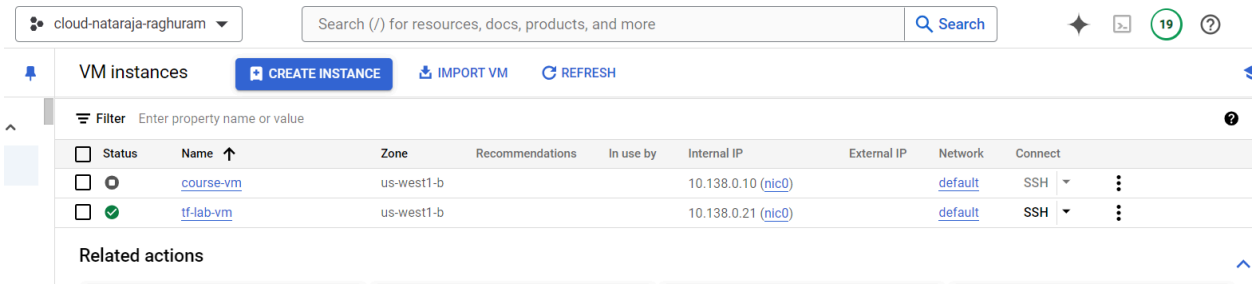
- Take a screenshot of the Guestbook including the URL with the entry in it.



07.1g: Terraform GCP Guestbook

4. Launching configuration

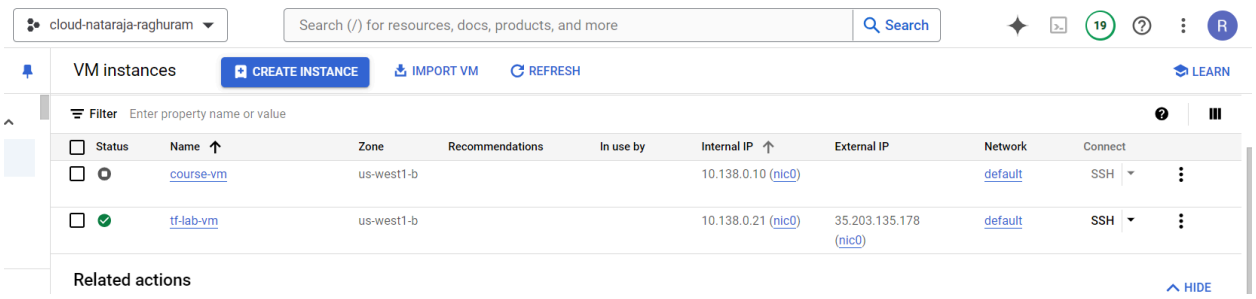
- Take a screenshot that includes the VM's IP addresses



- Take a screenshot showing the completion of the command including its output

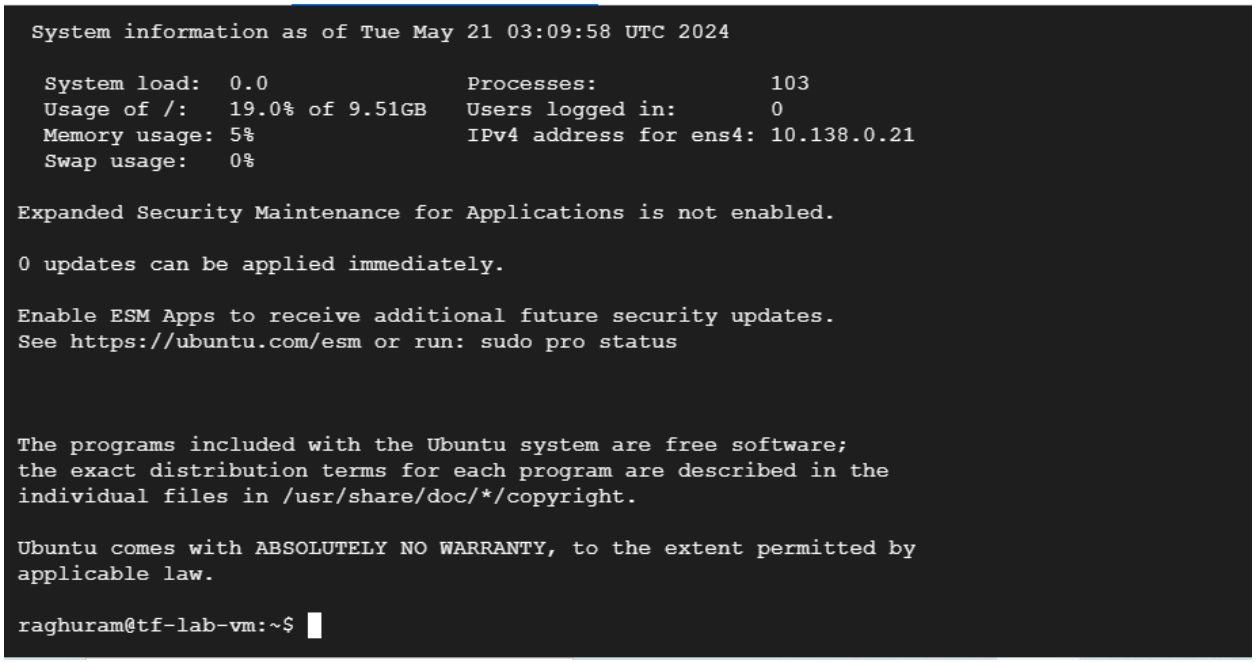
```
google_compute_address.static: Creating...
google_compute_address.static: Still creating... [10s elapsed]
google_compute_address.static: Creation complete after 11s [id=projects/cloud-nataraja-raghuram/regions/us-west1/addresses/ipv4-address]
google_compute_instance.default: Modifying... [id=projects/cloud-nataraja-raghuram/zones/us-west1-b/instances/tf-lab-vm]
google_compute_instance.default: Still modifying... [id=projects/cloud-nataraja-raghuram/zones/us-west1-b/instances/tf-lab-vm, 10s elapsed]
google_compute_instance.default: Modifications complete after 12s [id=projects/cloud-nataraja-raghuram/zones/us-west1-b/instances/tf-lab-vm]

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

Outputs:

ip = "35.203.135.178"
raghuram@cloudshell:~/tf (cloud-nataraja-raghuram)$
```

- Take a screenshot that includes the VM's IP addresses



6. Adding ssh access

- Take a screenshot of the successful ssh login from Cloud Shell.



- What resources are being added, changed, or destroyed?

Addition: Google Compute Engine has a new instance called google_compute_instance.default.

Adjustments are being made to the current (google_compute_instance.default) by modifying attributes such as metadata_startup_script, tags, and other configurations.

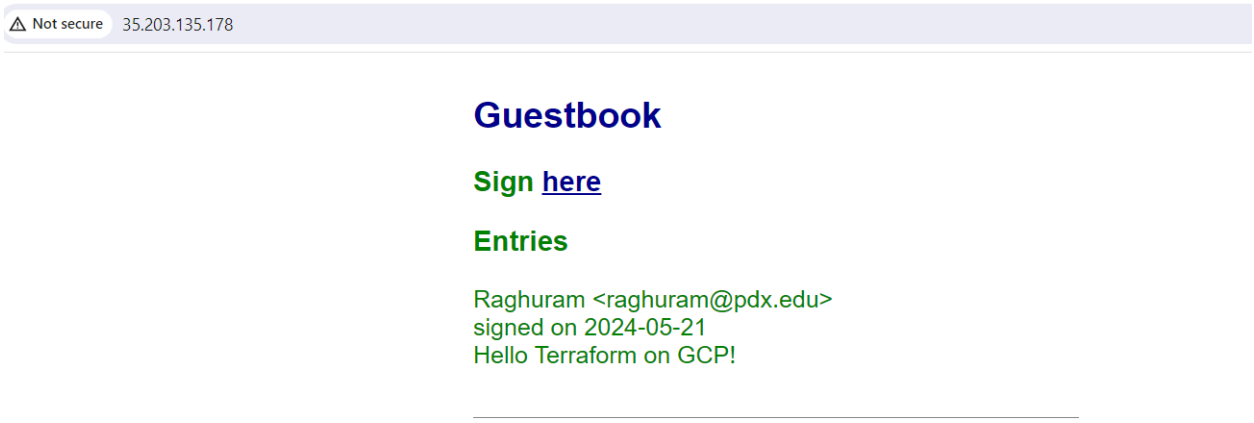
Destruction: A destruction alert has been sent for (google_compute_instance.default).

- What part of the configuration forces a replacement to occur?

The metadata_startup_script field forces a replacement

8. View the Guestbook

- Take a screenshot of the Guestbook including the URL with the entry in it.



07.2g: Kubernetes Guestbook

4. Create Kubernetes cluster

- What is the name of the Instance Template dynamically generated to create the two nodes (VMs)?

  gke-guestbook-default-pool-7b383a81

- What is the name of the Instance Group dynamically generated that the two nodes belong to?

  gke-guestbook-default-pool-7b383a81-grp

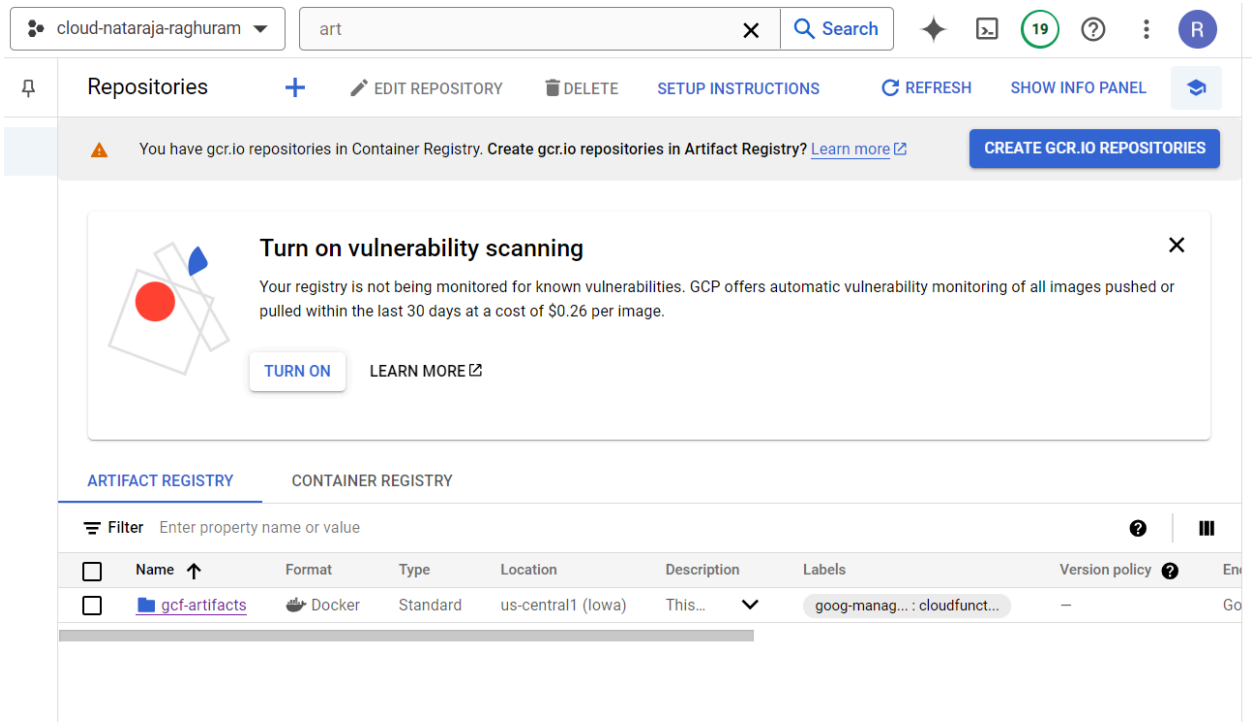
- What are the names of the two nodes?

  gke-guestbook-default-pool-7b383a81-clvf

  Gke-guestbook-default-pool-7b383a81-v0s1

5. Prepare a container image

- Take a screenshot of the container image created



7. Deploy the configuration

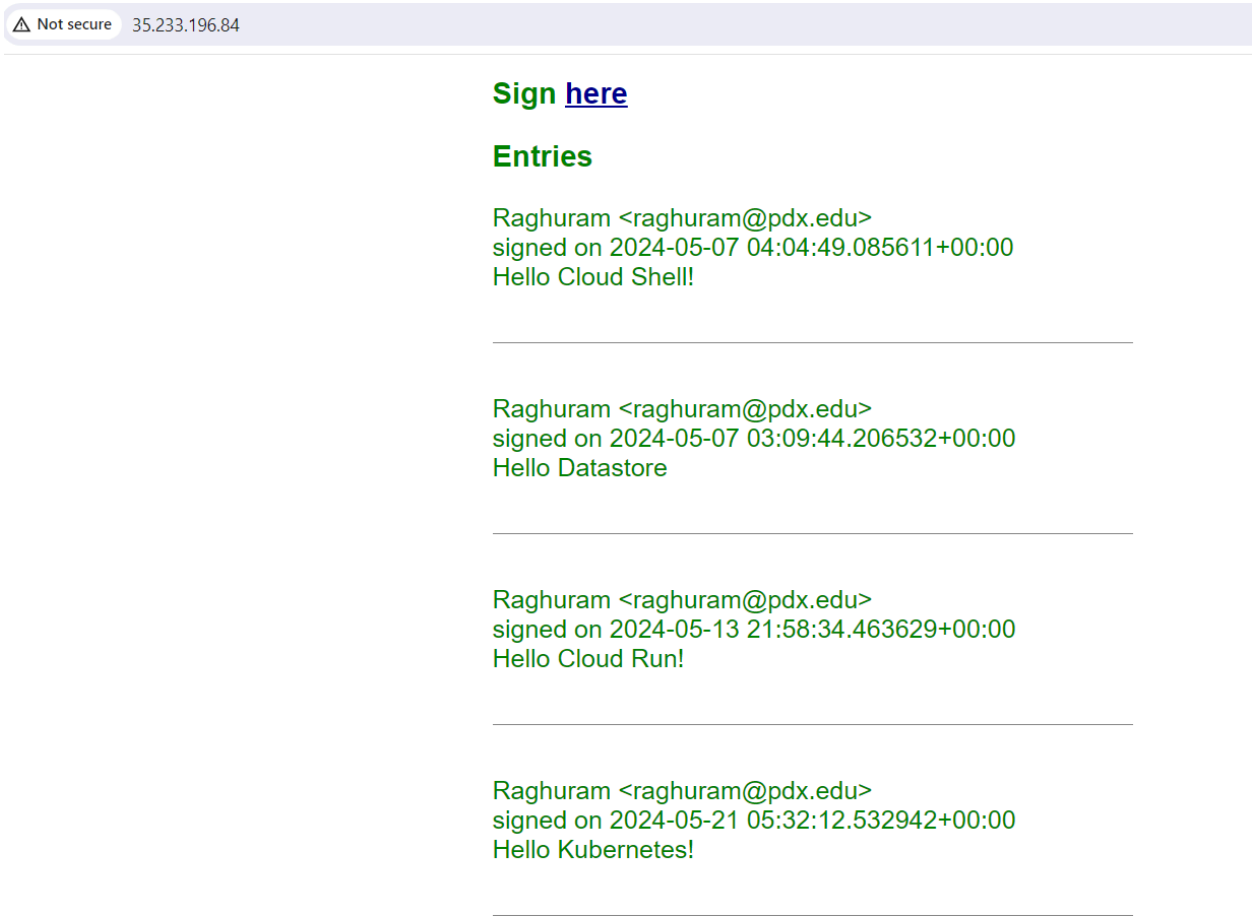
- Take a screenshot of the output of the following command when all 3 replicas reach a "Running" state.

```
raghuram@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nataraja-raghuram)$ kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
guestbook-replicas-h2xsk      0/1     ContainerCreating   0          26s
guestbook-replicas-v2njv      0/1     ContainerCreating   0          26s
guestbook-replicas-xxfcg      0/1     ContainerCreating   0          26s
raghuram@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nataraja-raghuram)$
```

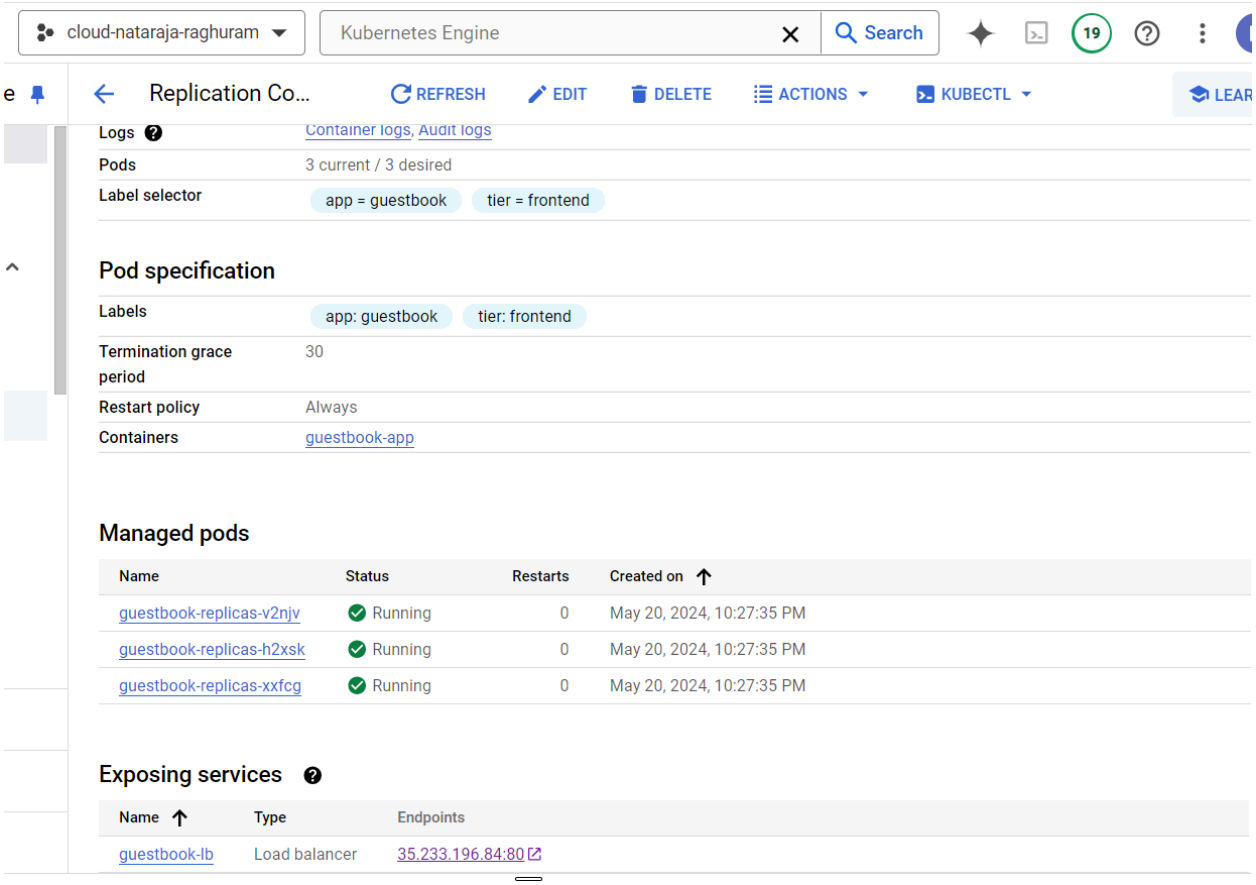
- Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.

```
raghuram@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nataraja-raghuram)$ kubectl get services
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
guestbook-lb   LoadBalancer   10.4.220.34    35.233.196.84   80:31212/TCP   110s
kubernetes     ClusterIP      10.4.208.1     <none>          443/TCP        70m
raghuram@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nataraja-raghuram)$
```

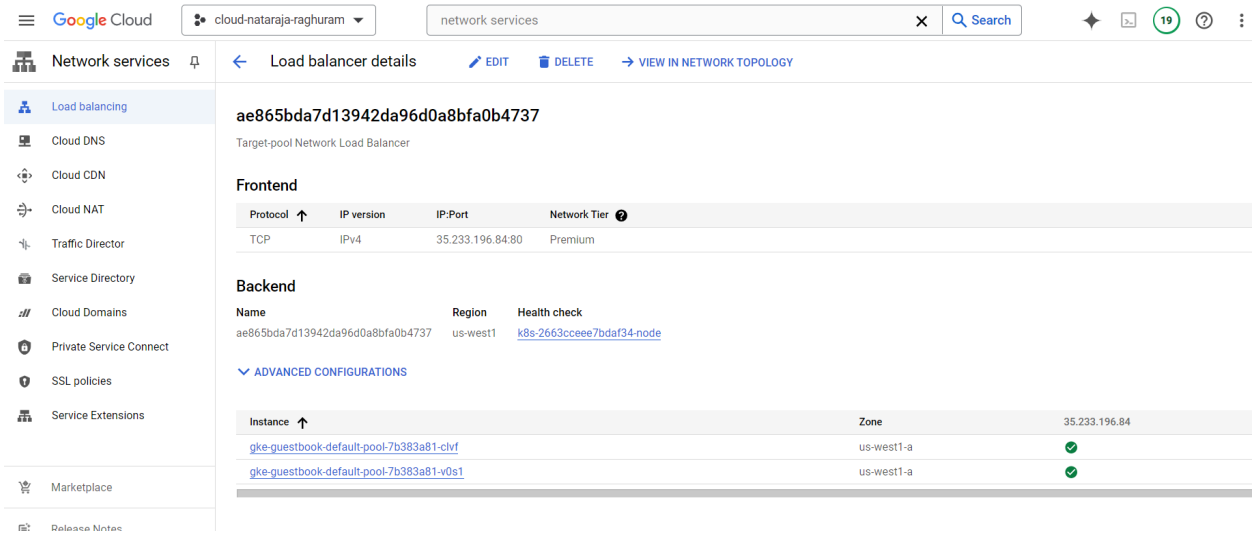
- Take a screenshot of the Guestbook including the URL with the entry in it.



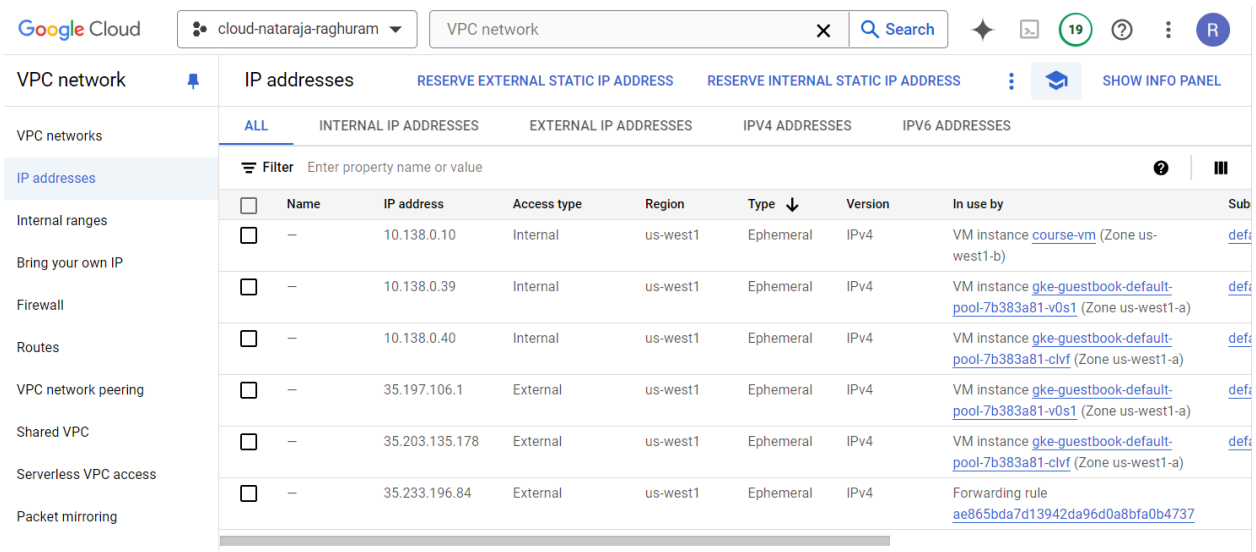
- Take a screenshot of the managed guestbook pods and the service being exposed.



- Take a screenshot of the load balancer and its details

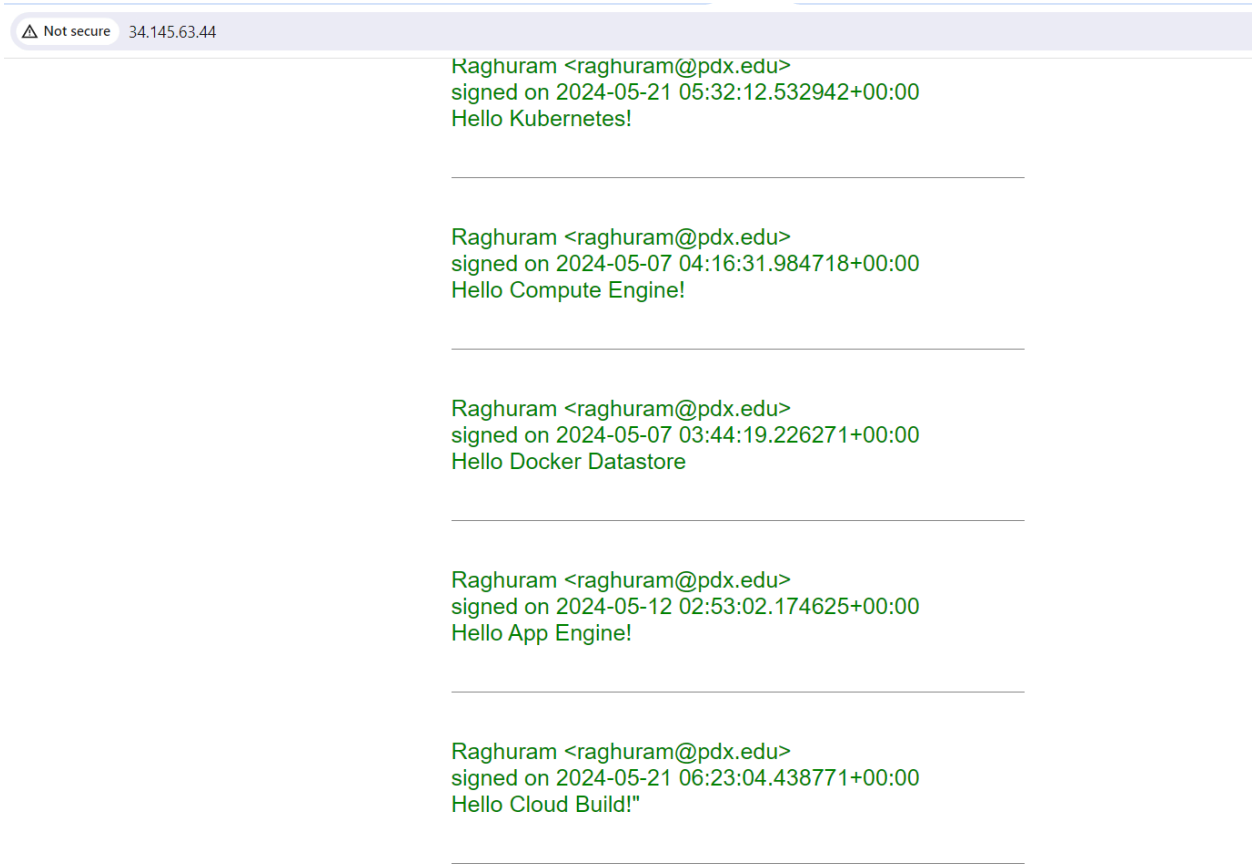


- Take a screenshot of the addresses allocated and indicate the ones associated with nodes versus the one associated with the load balancer.



12. Deploy and view application

- Take a screenshot of the Guestbook including the URL with the entry in it.



07.3g: APIs (Slack, Knowledge Graph)

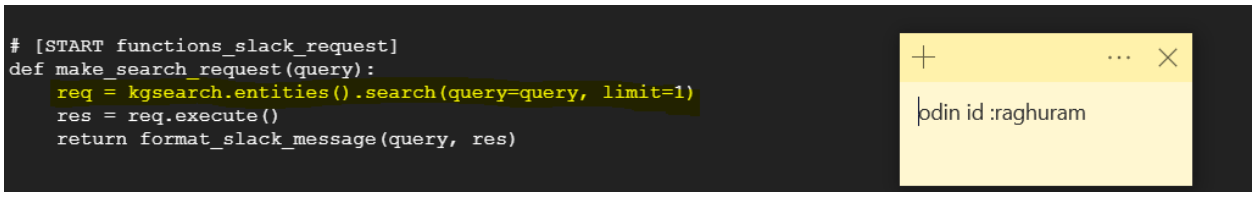
2. Code

- Does Google provide a Python package specifically for accessing the Knowledge Graph API?

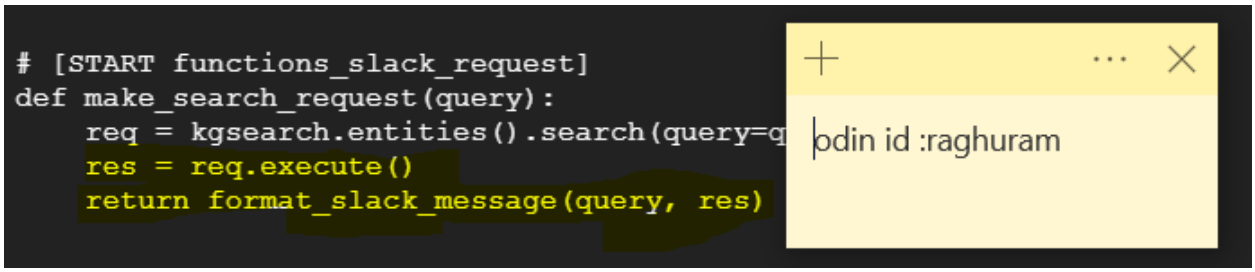
NO

3. Code

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



- 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?



‘ execute()’ is the method

- What is the Python data type that is used to represent the formatted message?

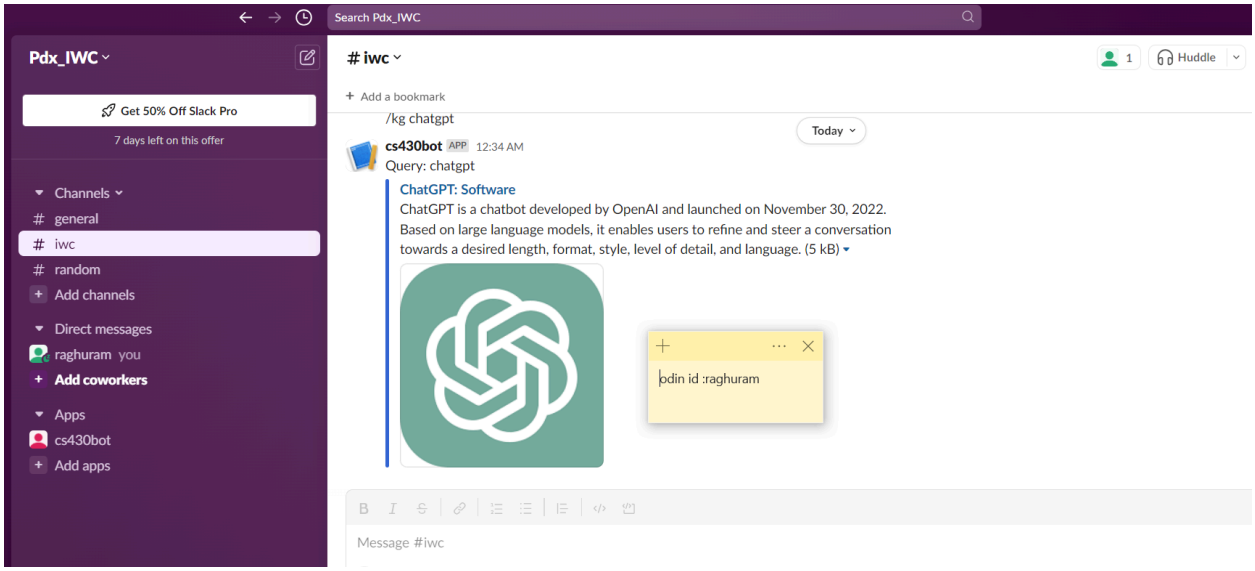
  formatted message is a dictionary.

- What are the three main attributes of the formatted message passed back to Slack?

  "response_type", "text", "attachments"

8. Test the command

- Take a screenshot of its response for your lab notebook.



07.4g: ML APIs

3. Vision

- Show the output for your lab notebook

- What is the name of the function?

  detect_labesl_uri

- What type of Vision client is instantiated in it?

  vision.ImageAnnotatorClient is instantiated in this function.

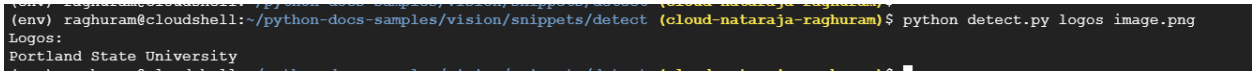
- What method is invoked in the Vision client to perform the detection?

  label_detection

- What is the name of the attribute in the response object that contains the results we seek?

  Response.label_annotations

- Take a screenshot of the output for the above commands

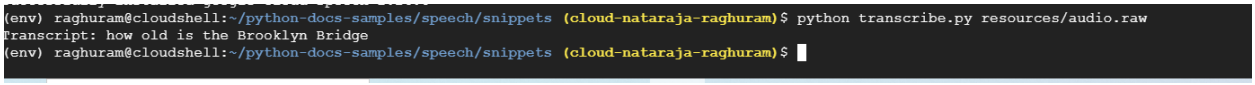


- What method is invoked in the Vision client to perform the detection?

  logo_detection

4. Speech

- Show the output for your lab notebook



- What is the name of the function?

  transcribe_file

- What method is invoked in the Speech client to perform the detection?

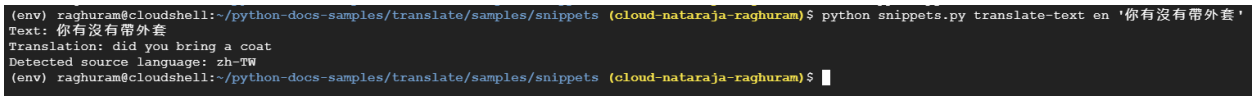
  recognize

- What is the name of the attribute in the response object that contains the results we seek?

  Transcript

5. Translate

- Show the output for your lab notebook



- What is the name of the function?

  translate_text

- What method is invoked in the Translate client to perform the detection?

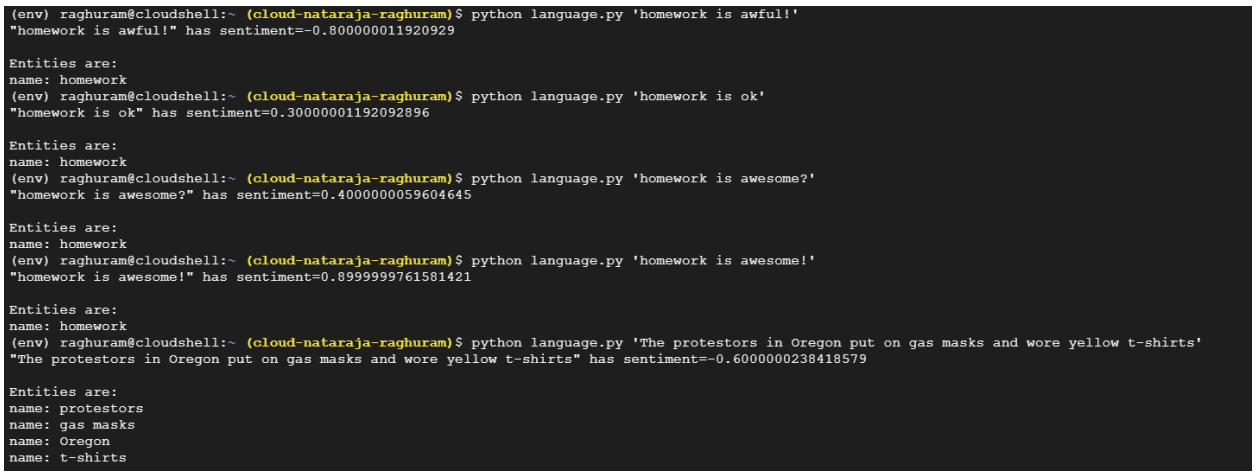
  translate_client.translate

- What is the name of the attribute in the response object that contains the results we seek?

  result["translatedText"]

6. Natural Language

- Show the output for your lab notebook



8. Code

- What is the name of the function that performs the transcription?

  transcribe_gcs

- What is the name of the function that performs the translation?

  translate_text

- What is the name of the function that performs the entity analysis on the translation?

  entities_text

- What is the name of the function that performs the entity analysis on the image?

  detect_labels_uri

- If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?

to fix the program's problem of classifying the image and audio as unrelated even though they contain related content, take into consideration the following changes:
Synonym Recognition:
Include a feature or make use of a library or service that identifies synonyms or related terms.
Adjust Matching Logic:
Make changes to the compare_audio_to_image function's comparison logic.

- If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?

Image labels and entities must match

Consider labels as synonyms or closely related terms of entities instead of matching strings directly.

Make sure that related terms and synonyms are recognized correctly in several languages.

- If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?

It is necessary to improve the comparison logic in order to resolve the present problem where the program is unable to identify the connection between audio and image content (for example, "ostriches" in audio and "Ostrich" in image labels).

- What are the 3 labels with the highest confidence that the Video Intelligence API associates with the video and what are the confidences for each?

sports, basketball, player

- What is the name of the client class in the package that is used?

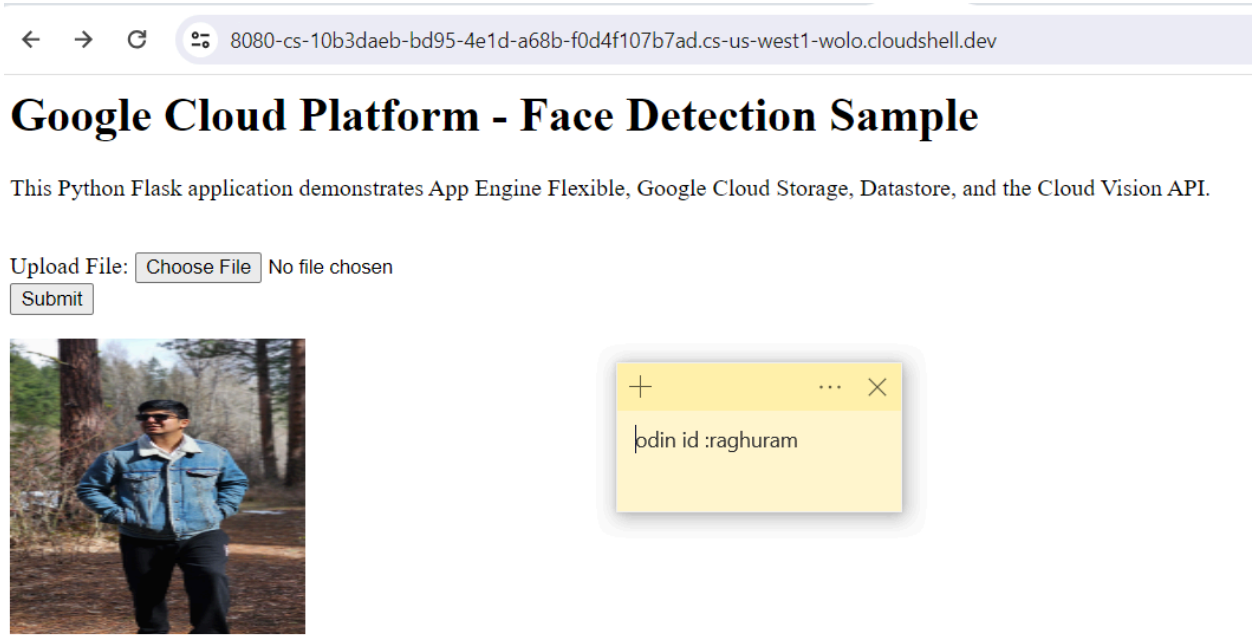
VideoIntelligenceServiceClient

- What method is used in that class to perform the annotation?

Annotate_video

16. Application

- Take a screenshot for your lab notebook that includes the URL.



- What line of code creates the query for previous detections?

  query = datastore_client.query(kind="Faces")

- What line of code sends the query to Cloud Datastore?

  image_entities = list(query.fetch())

- **Show the line that retrieves the name of the storage bucket to use.**

  bucket = storage_client.get_bucket(CLOUD_STORAGE_BUCKET)

- **What form field is used to specify the uploaded photo?**

  photo = request.files["file"]

- **Show the line that copies the photo's contents to the storage bucket.**

  blob.upload_from_string(photo.read(), content_type=photo.content_type)

- **What method in Vision's annotation client is used to perform the analysis?**

  faces = vision_client.face_detection(image=image).face_annotations

- **What fields are stored in Cloud Datastore for each image?**

  entity["blob_name"] = blob.name

  entity["image_public_url"] = blob.public_url

  entity["timestamp"] = current_datetime

  entity["joy"] = face_joy

- **What happens at the end of the** `upload_photo` route?

  return redirect("/")