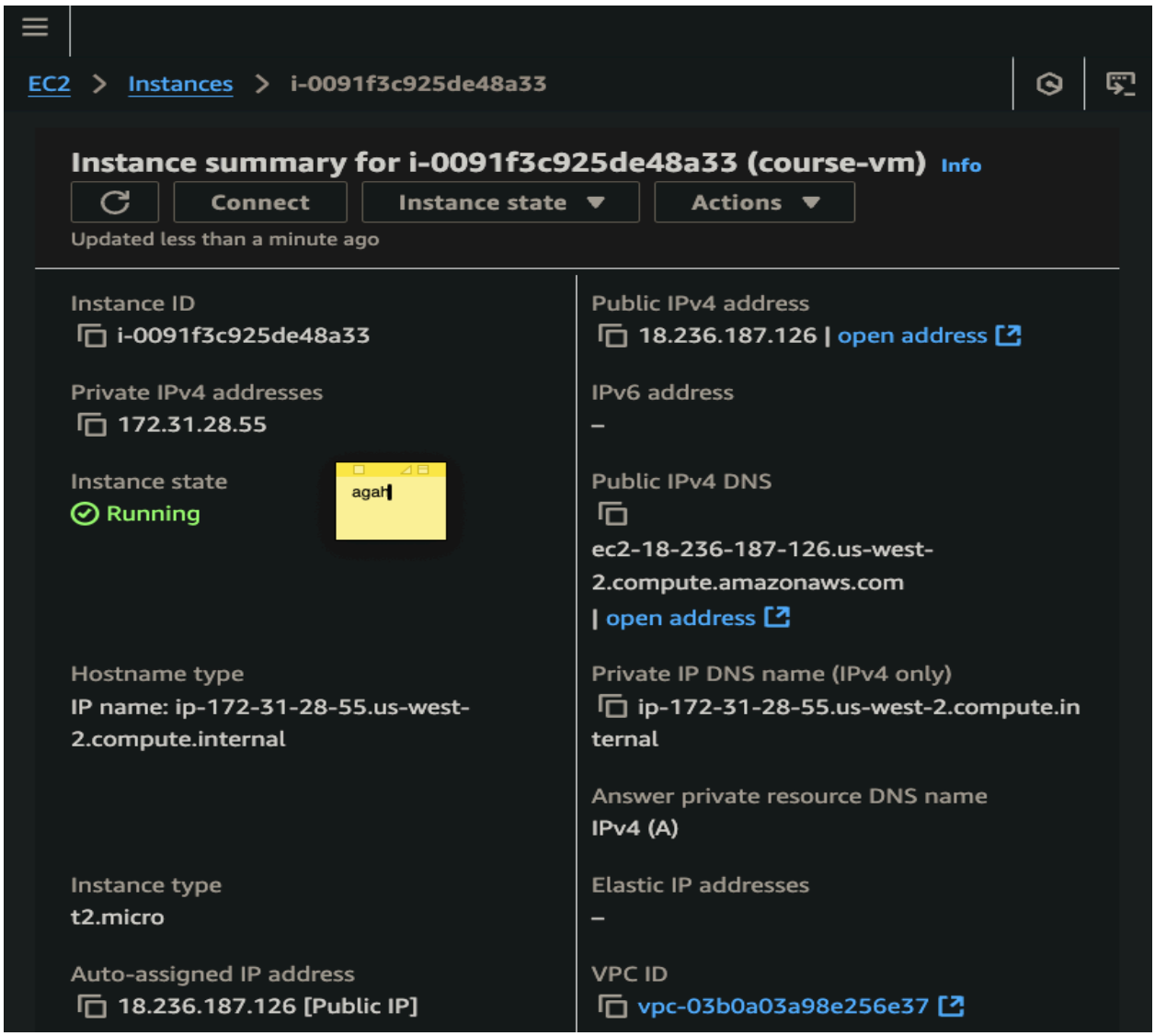
AHMAD CS530 Week7

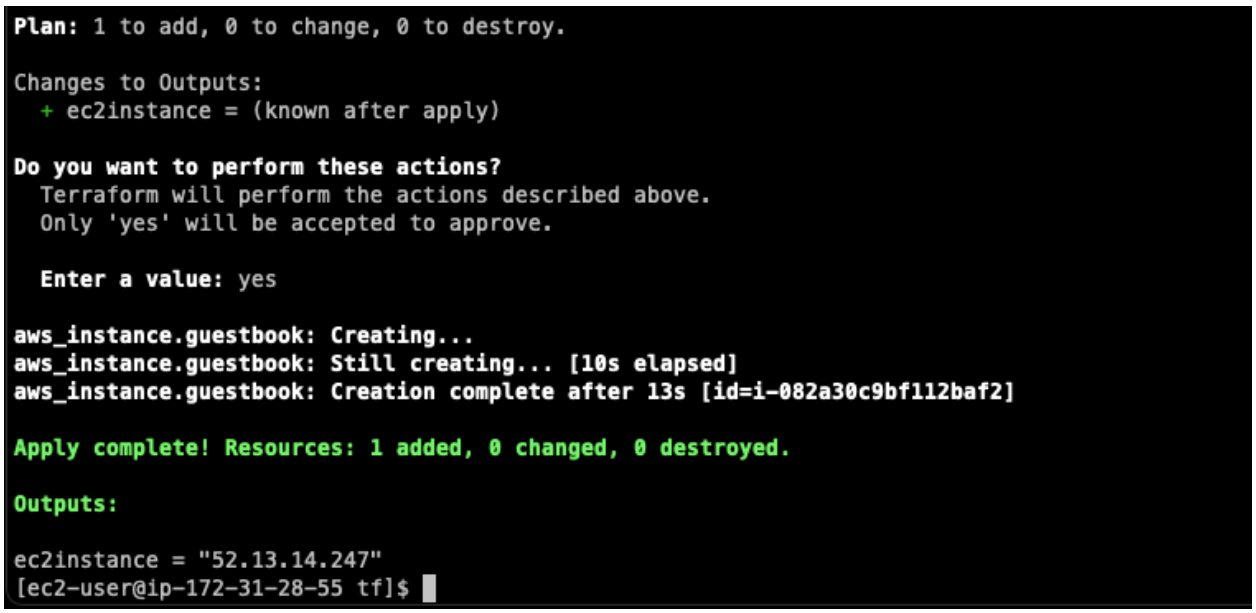
07.1a: Terraform AWS Guestbook

7.1a.4. Launching configuration

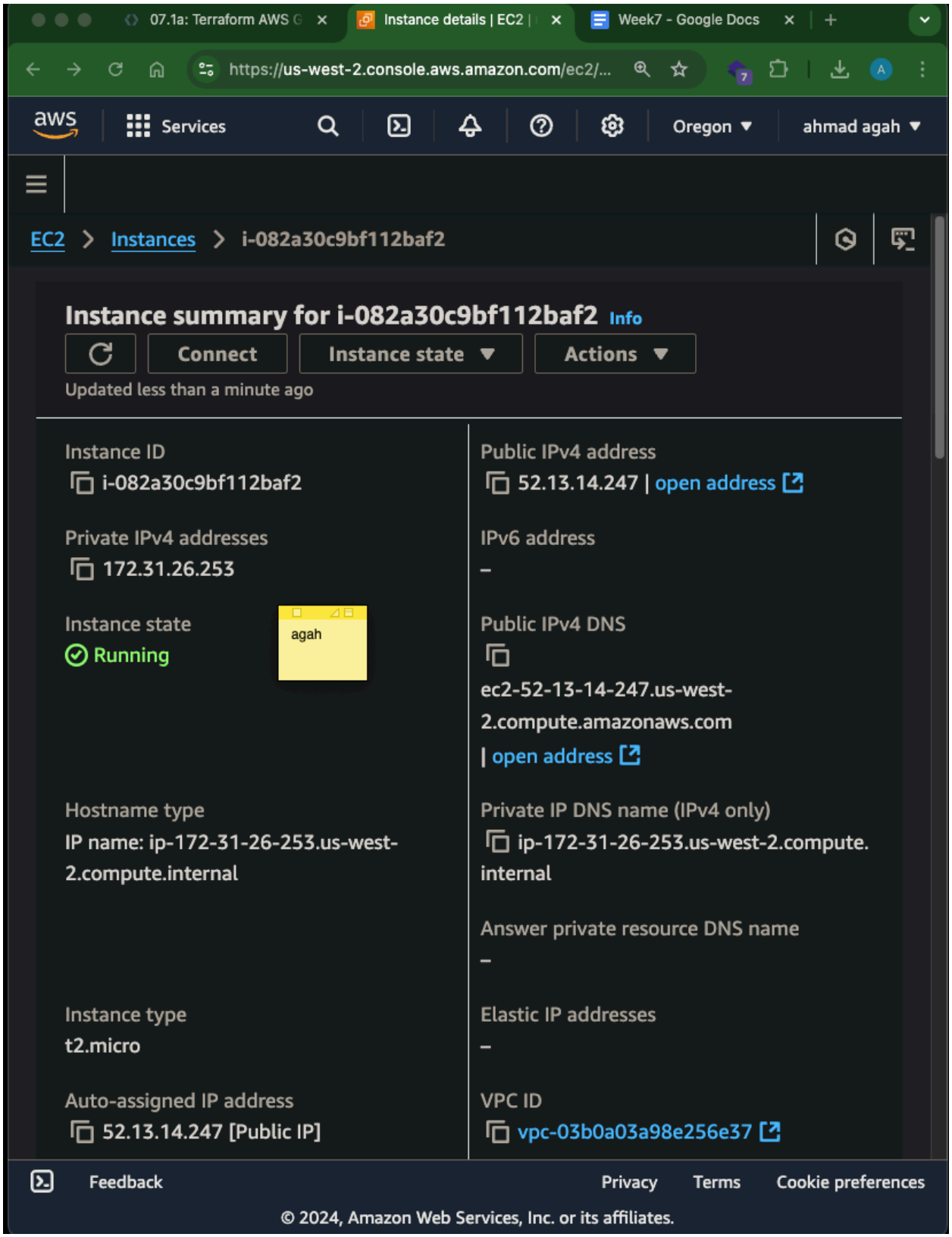
the IP address of the EC2 instance is shown below, so I don't need to include my Odin username in every screenshot.



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



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

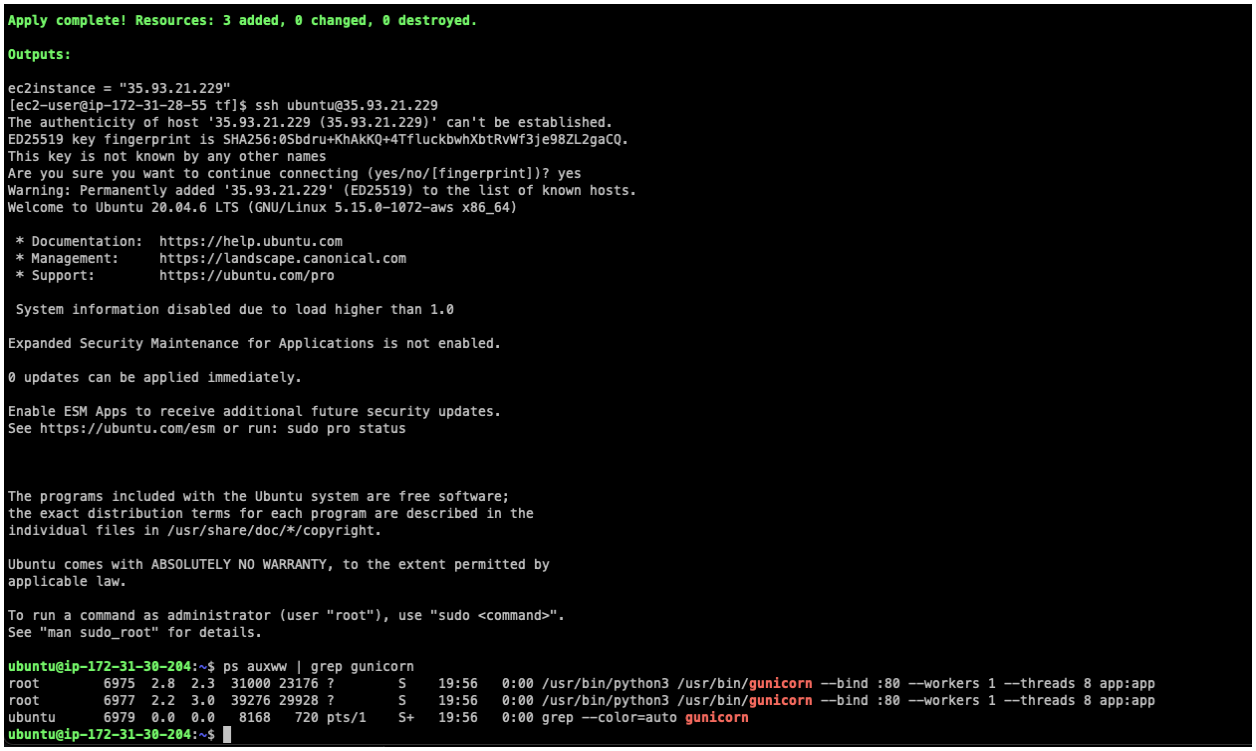


7.1a.6. Adding ssh access

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

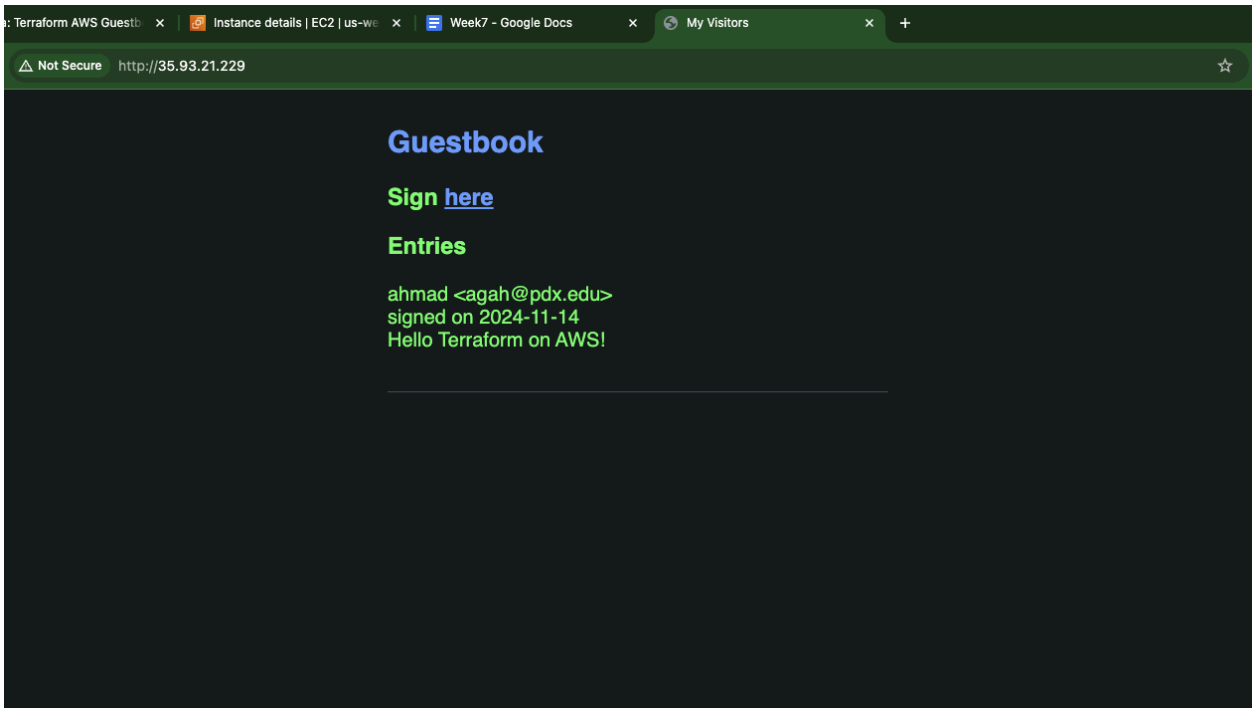
7.1a.7. Adding the Guestbook application

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



7.1a.8. View the Guestbook

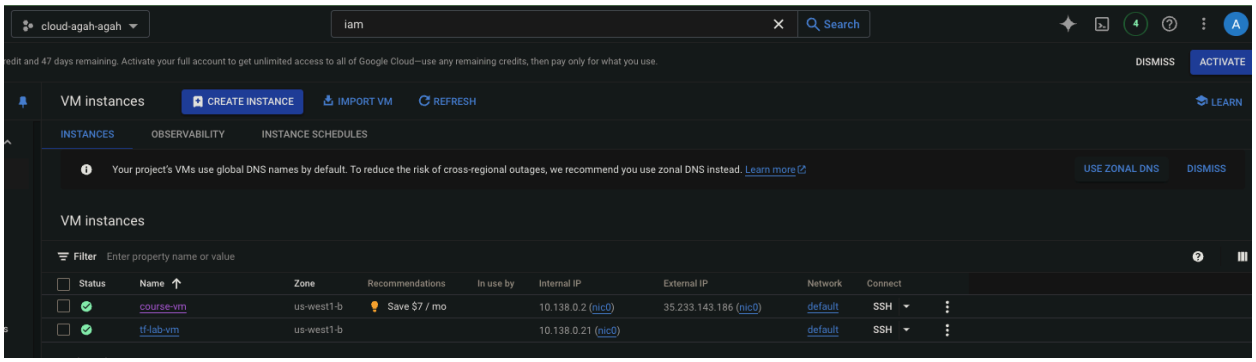
- Take a screenshot of the Guestbook including the URL with the entry in it.



07.1g: Terraform GCP Guestbook

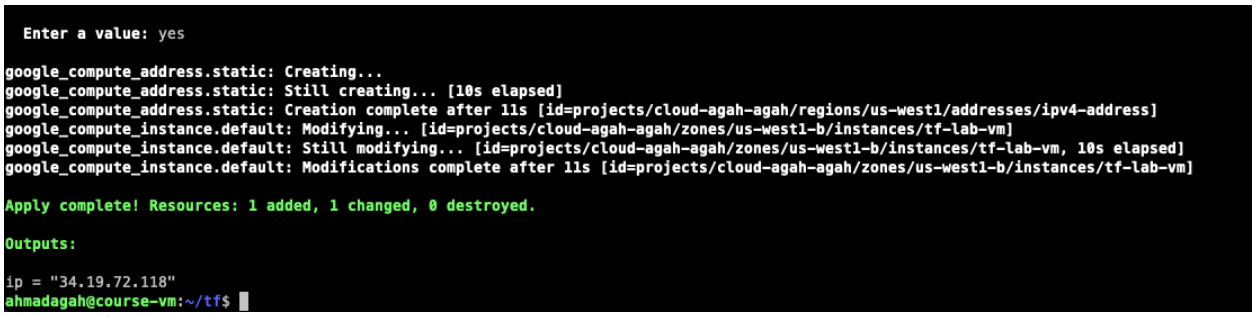
7.1g.4. Launching configuration

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

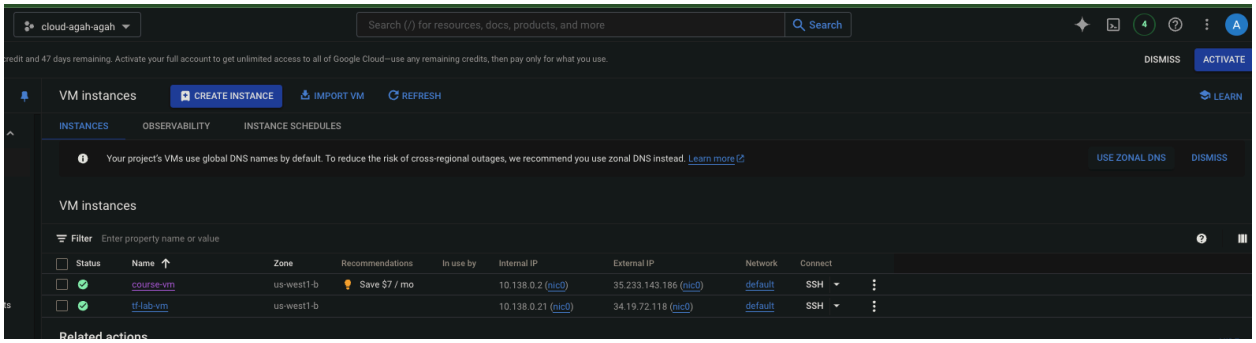


7.1g.5. Adding an external IP address

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

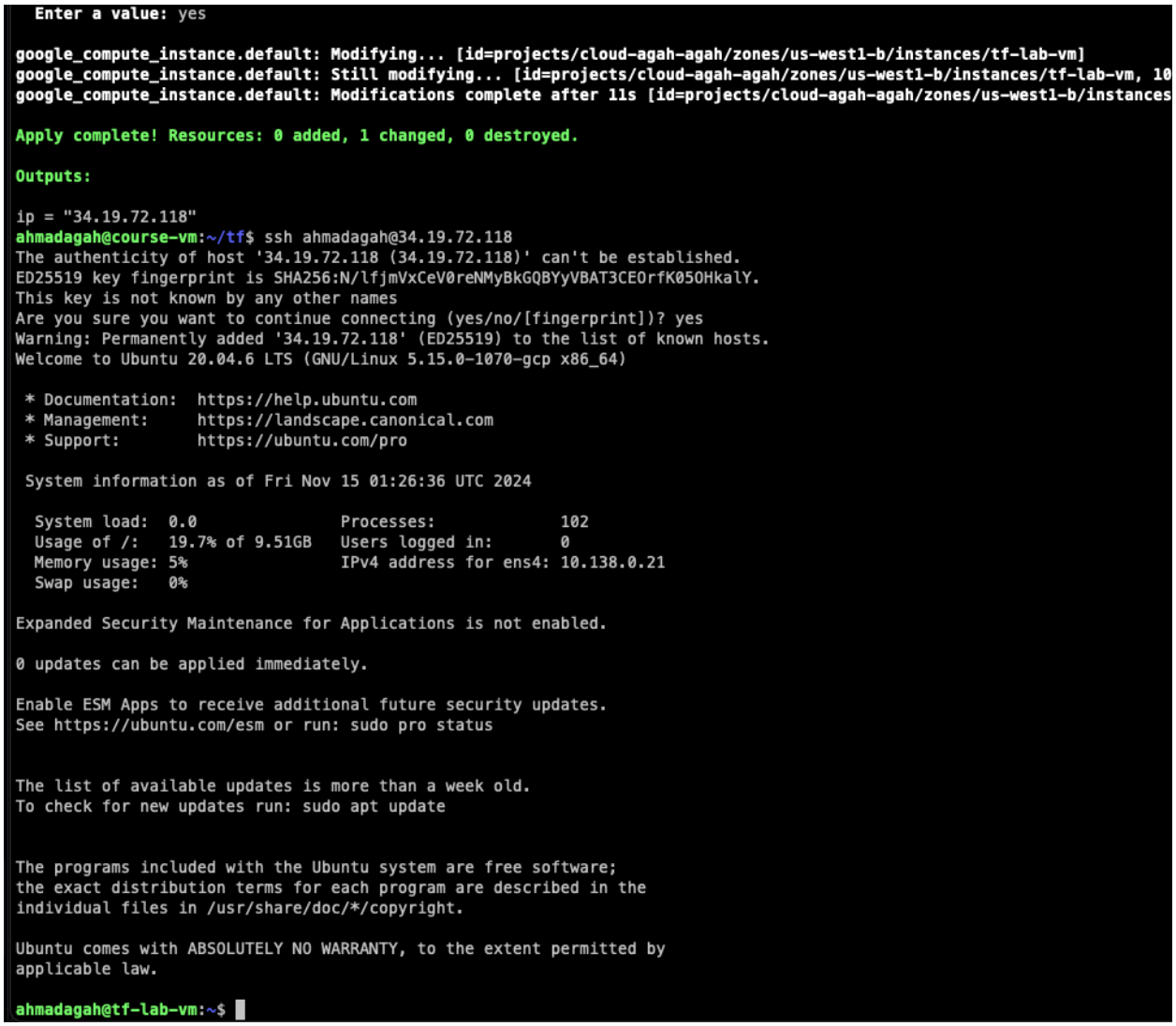


- Take a screenshot that includes the VM's IP addresses



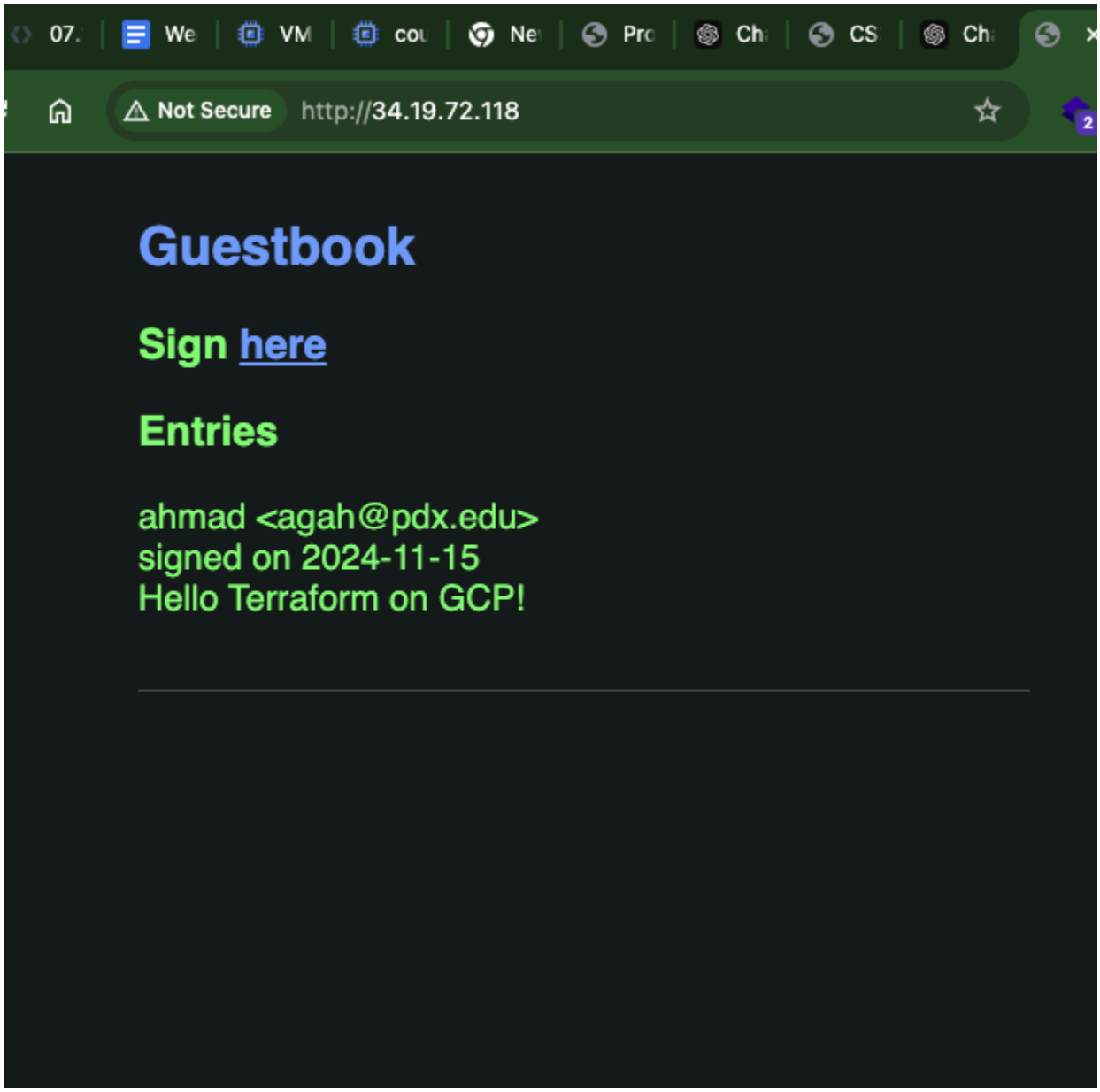
7.1g.6. Adding ssh access

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



7.1g.8. View the Guestbook

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



07.2g: Kubernetes Guestbook

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

gke-guestbook-default-pool-11fe4be0

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

gke-guestbook-default-pool-11fe4be0-grp

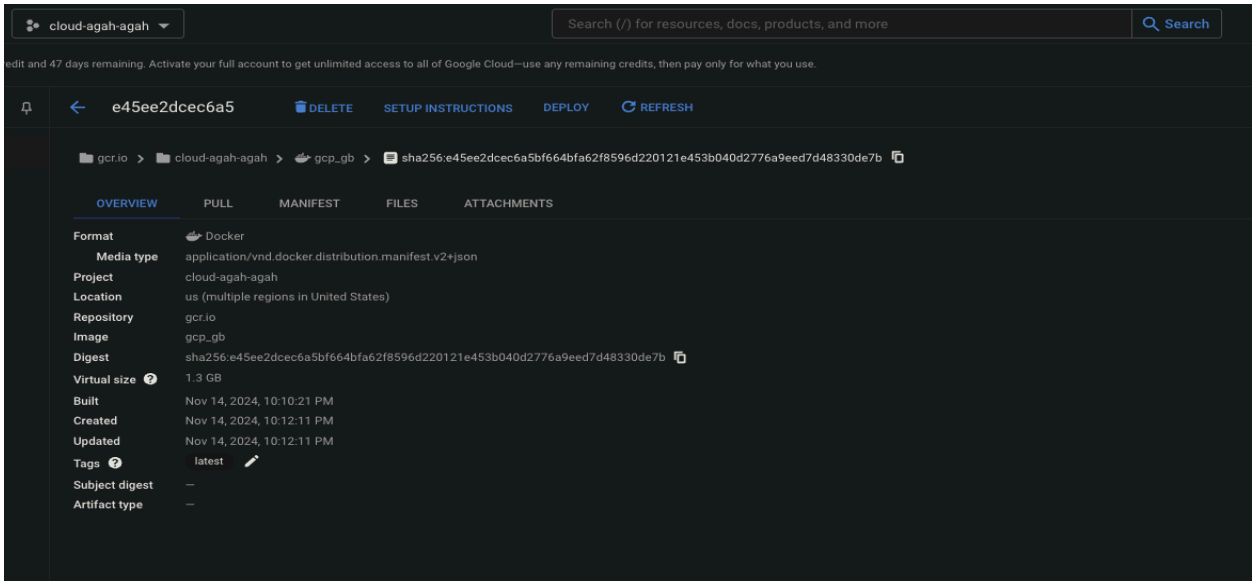
- What are the names of the two nodes?

gke-guestbook-default-pool-11fe4be0-rkn4
gke-guestbook-default-pool-11fe4be0-t008

7.2g.5. Prepare a container image

I'm not displaying my Odin ID when you see my project name, which includes my Odin ID (cloud-agah-agah). You previously deducted points for not displaying my Odin ID, even if the screenshot contained my Odin ID elsewhere.

- Take a screenshot of the container image created



7.2g.7. Deploy the configuration

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

```
agah@cloudshell:~/cs430-src/05_gcp_datastore (cloud-agah-agah)$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
guestbook-replicas-6xfkv      1/1     Running   0          7m44s
guestbook-replicas-vdwt5      1/1     Running   0          7m44s
guestbook-replicas-vk9bx      1/1     Running   0          7m44s
agah@cloudshell:~/cs430-src/05_gcp_datastore (cloud-agah-agah)$
```

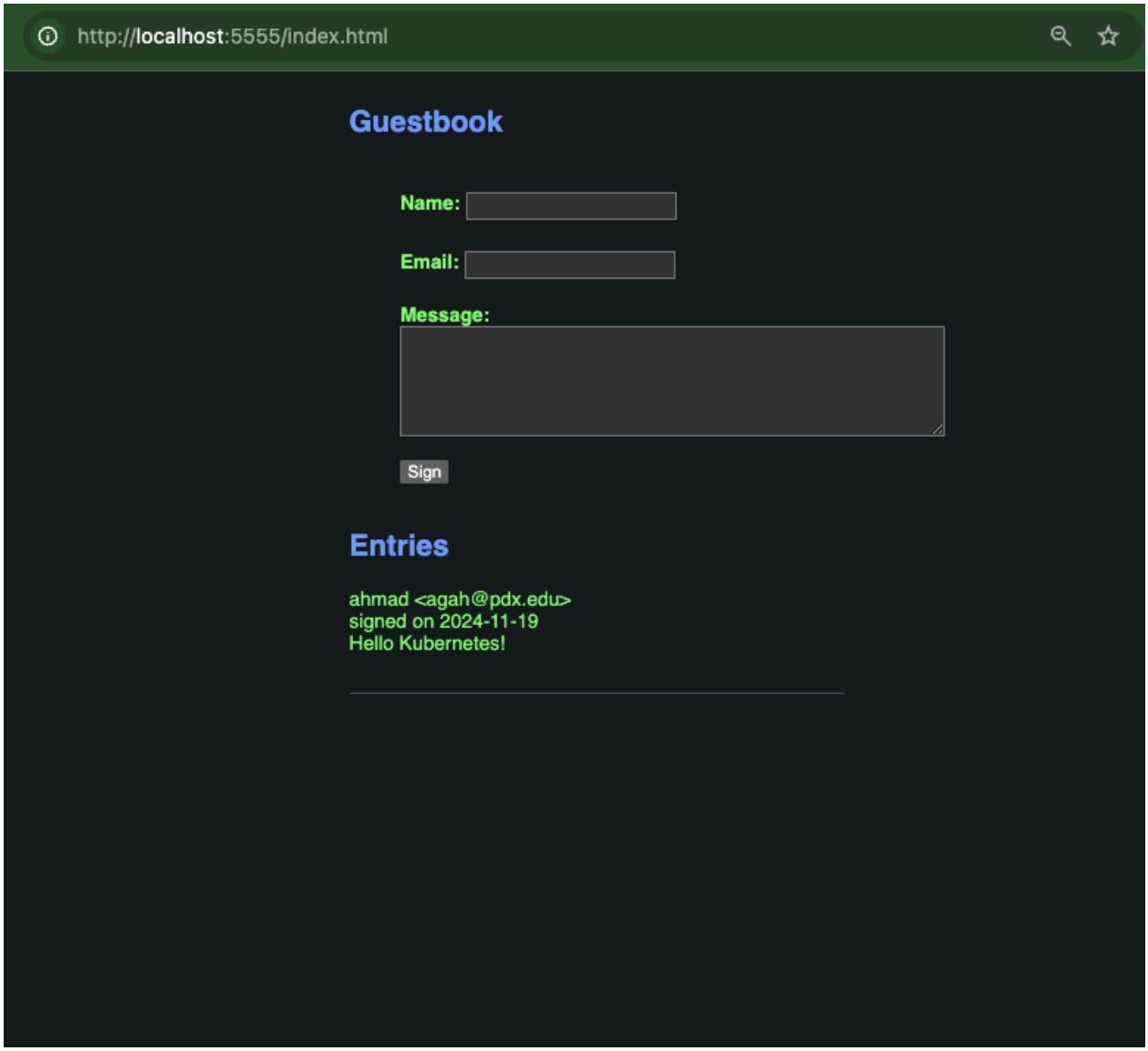
- Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.

```
agah@cloudshell:~/cs430-src/05_gcp_datastore (cloud-agah-agah)$ kubectl get services
NAME           TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
guestbook-lb   LoadBalancer   34.118.237.137   34.169.83.44   80:31999/TCP   8m44s
kubernetes     ClusterIP      34.118.224.1     <none>         443/TCP        68m
agah@cloudshell:~/cs430-src/05_gcp_datastore (cloud-agah-agah)$
```

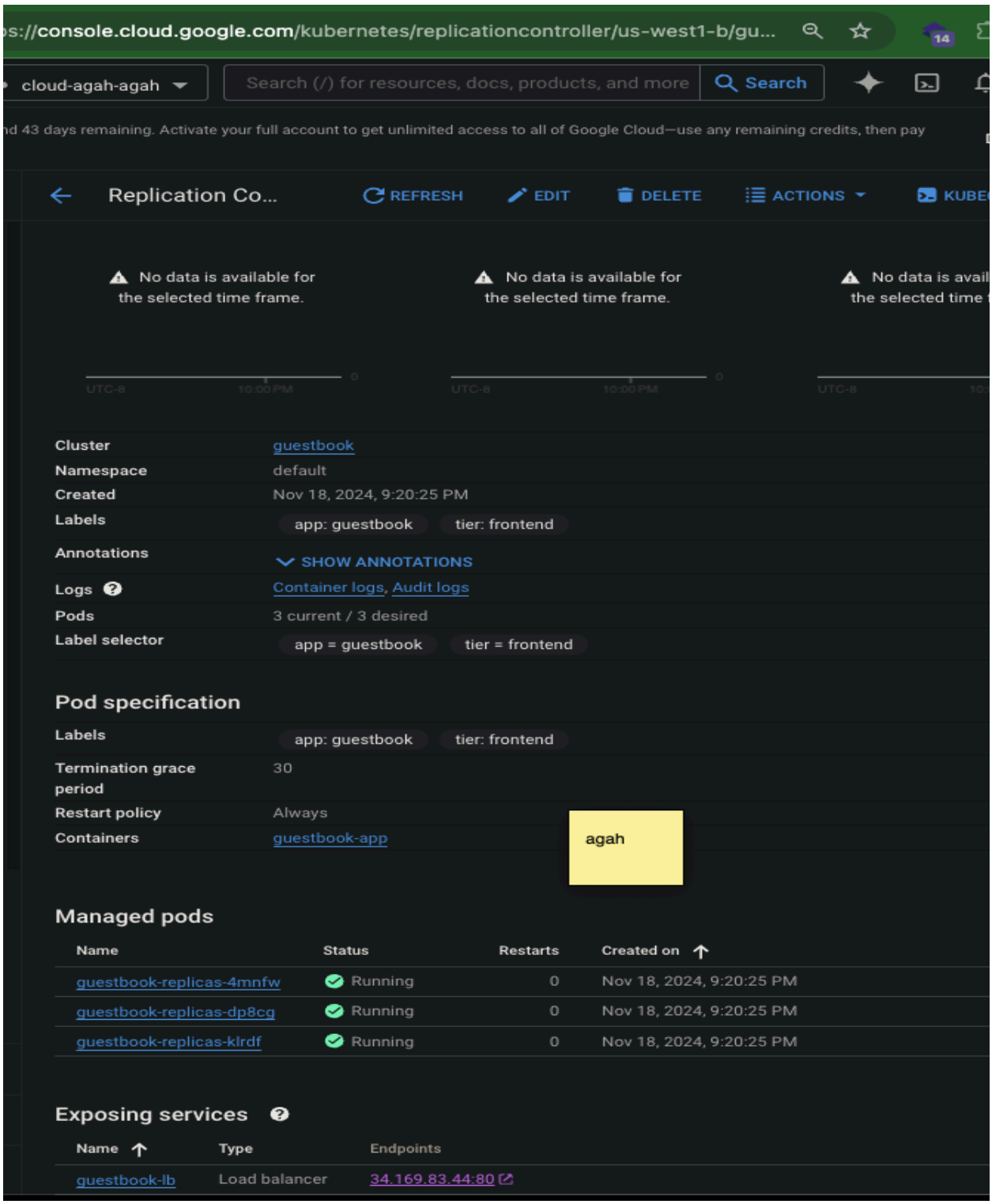
7.2g.8. View the Guestbook

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

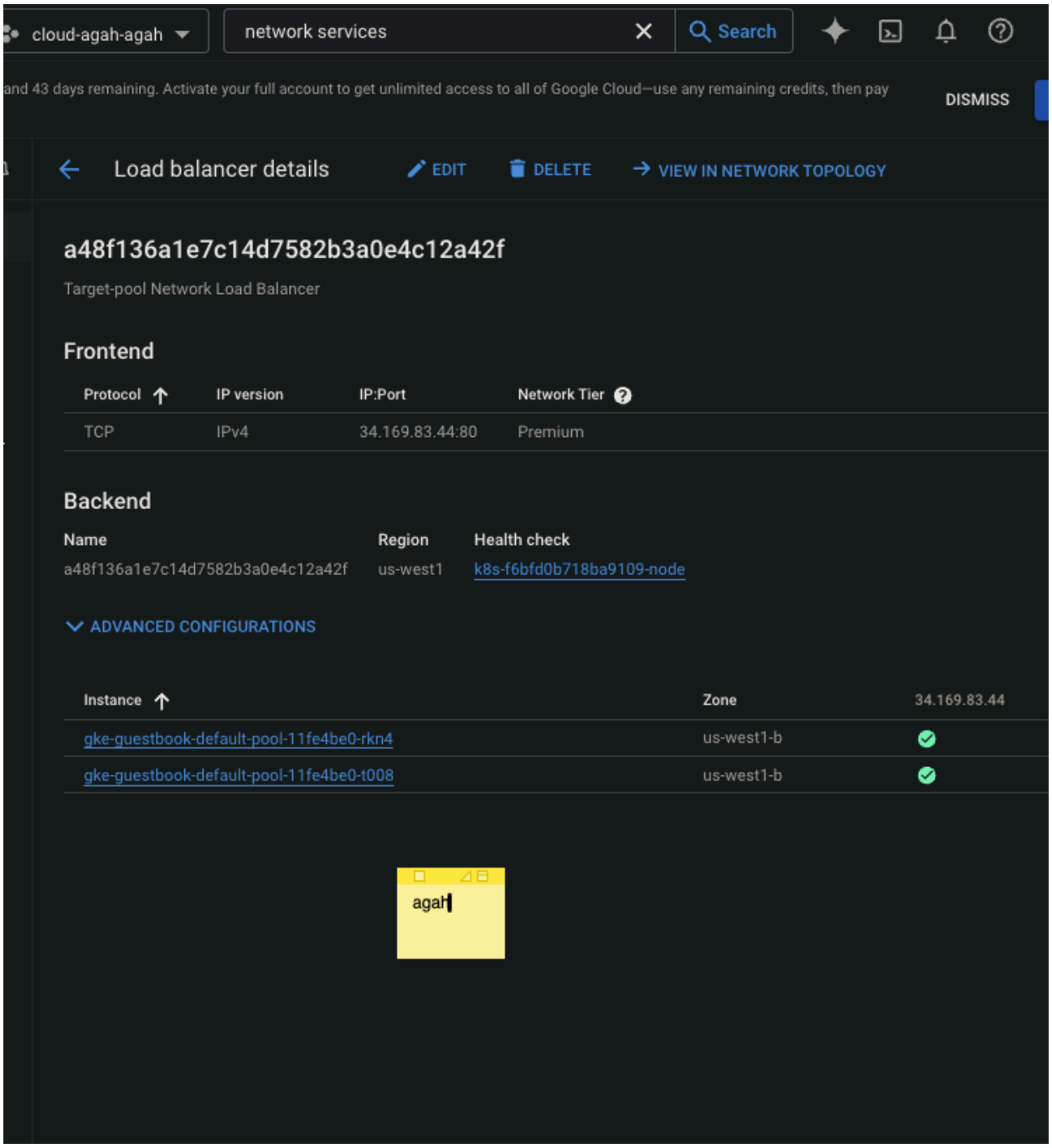
’m using a Compute Engine instance instead of Cloud Shell because Cloud Shell doesn’t support features like highlighting in Vim, and I can’t locate the cursor when entering commands. I’m forwarding the port to 5555, so you’ll see `localhost:5555` in the browser.


http://localhost:5555/index.html
Guestbook
Name:
Email:
Message:
Sign
Entries
ahmad <agah@pdx.edu>
signed on 2024-11-19
Hello Kubernetes!

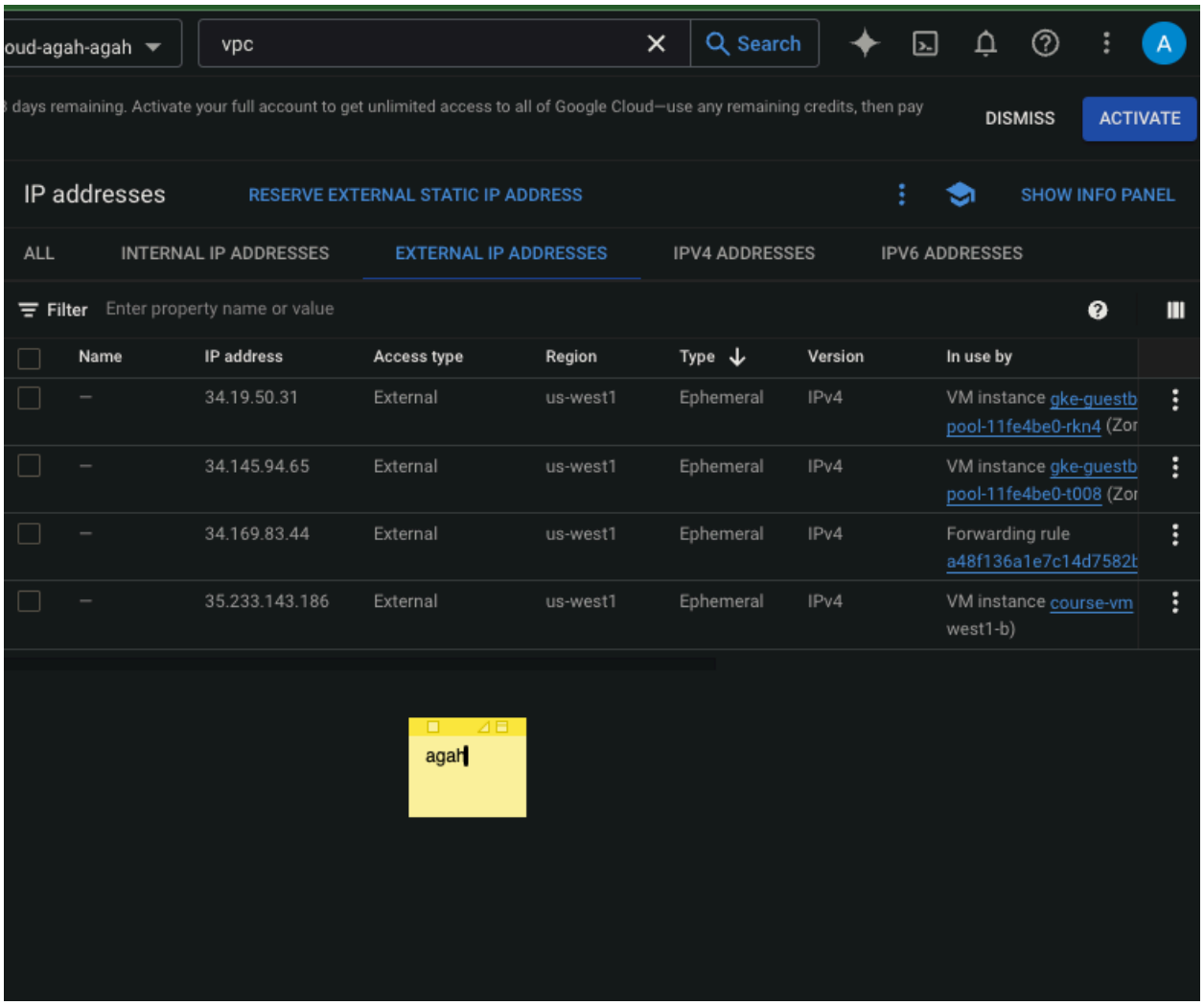
- Take a screenshot of the managed guestbook pods and the service being exposed.



- 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.

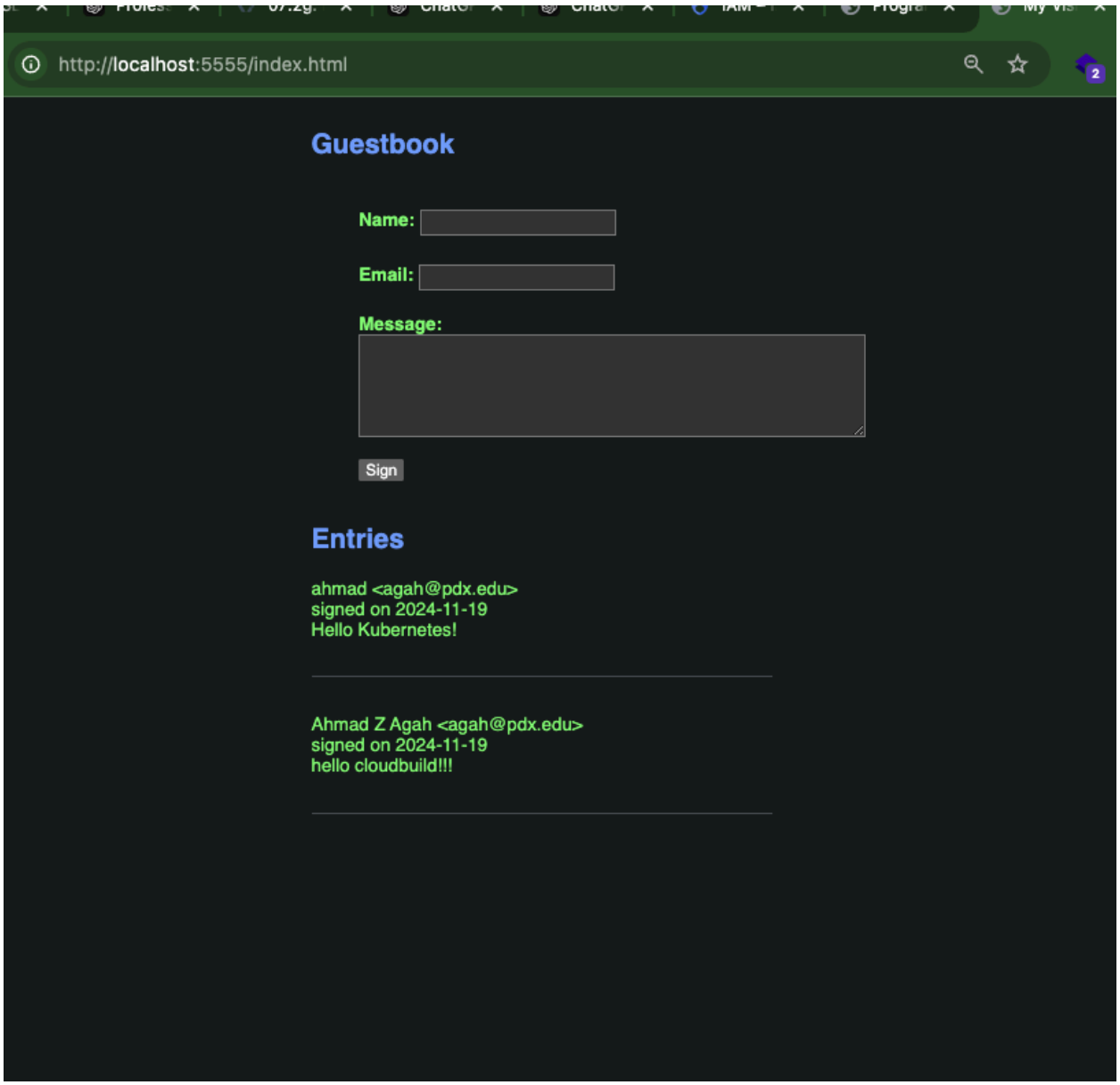


34.19.50.31 and 34.145.94.65 are associated with the VM instances in the gke-guestbook-default-pool, indicating these are IP addresses assigned to the nodes

34.169.83.44 is used by a forwarding rule, which suggests it is the IP address associated with the load balancer

7.2g.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

07.3g.2. Code

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

No, Google does not provide a dedicated Python package specifically for the Knowledge Graph API. Instead, developers use the Google API Client Library for Python, such as googleapiclient.discovery, to access various Google APIs, including the Knowledge Graph API. The googleapiclient.discovery.build function is used to create an interface for interacting with the Knowledge Graph API

07.3g.3. Code

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

req = kgsearch.entities().search(query=query, limit=1)

- 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?

res = req.execute()

The method that sends the query to the Knowledge Graph API is `execute()`.
This method is called on the `req` object (the request object created previously) to send the query to the API and retrieve the response. The response is then stored in the `res` variable.

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

The Python data type used to represent the formatted message is a **dictionary**.

```
message = {
    "response_type": "in_channel",
    "text": f"Query: {query}",
```

```
    "attachments": [],
}
```

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

`response_type`:

- Specifies how the response should be displayed in Slack.
- Example value: `"in_channel"`, which means the response will be visible to everyone in the channel.

`text`:

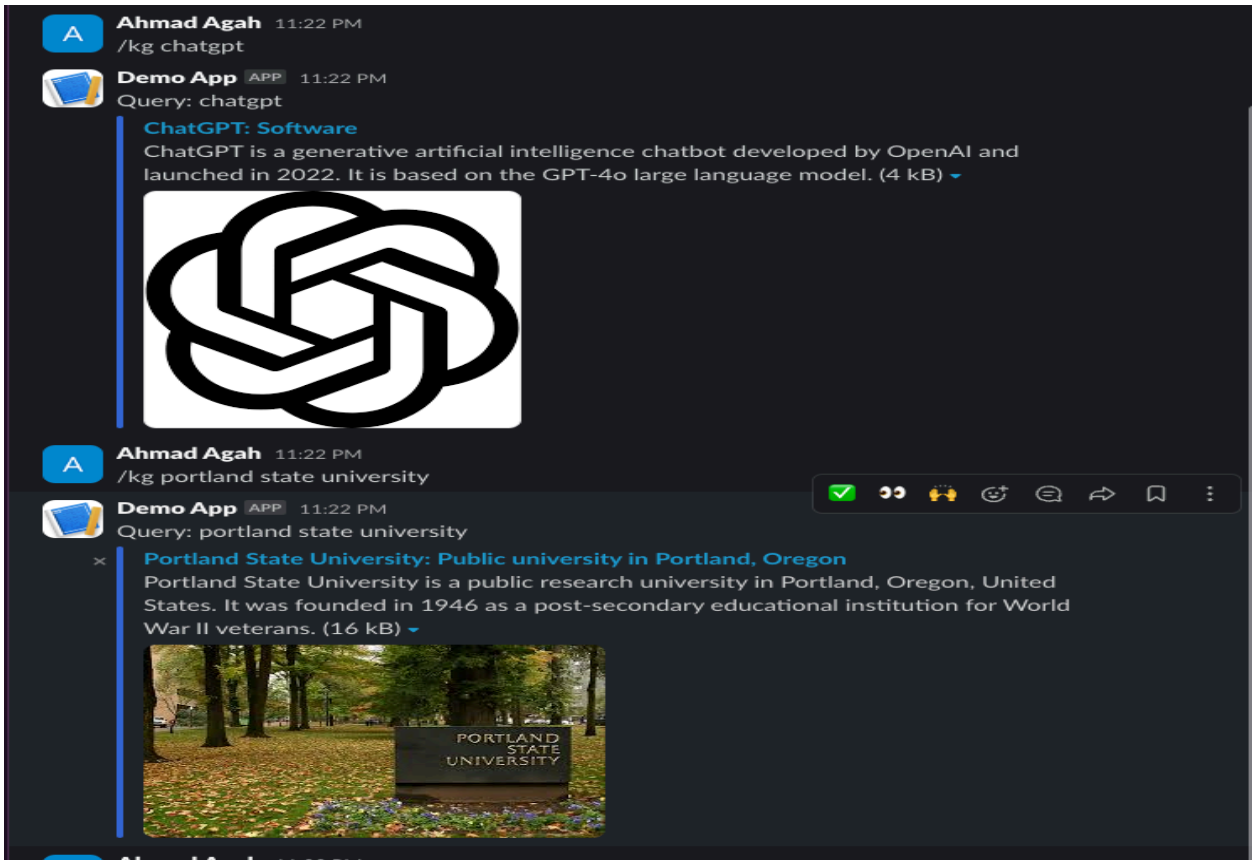
- The main text content of the message, which typically includes the original query or a summary.
- Example value: `f"Query: {query}"`, showing the search term the user submitted.

`attachments`:

- An array that contains additional rich content for the message, such as detailed descriptions, links, images, and formatted text.
- Each attachment is a dictionary that can include attributes like `title`, `title_link`, `text`, `image_url`, and `color`.

07.3g.8. Test the command

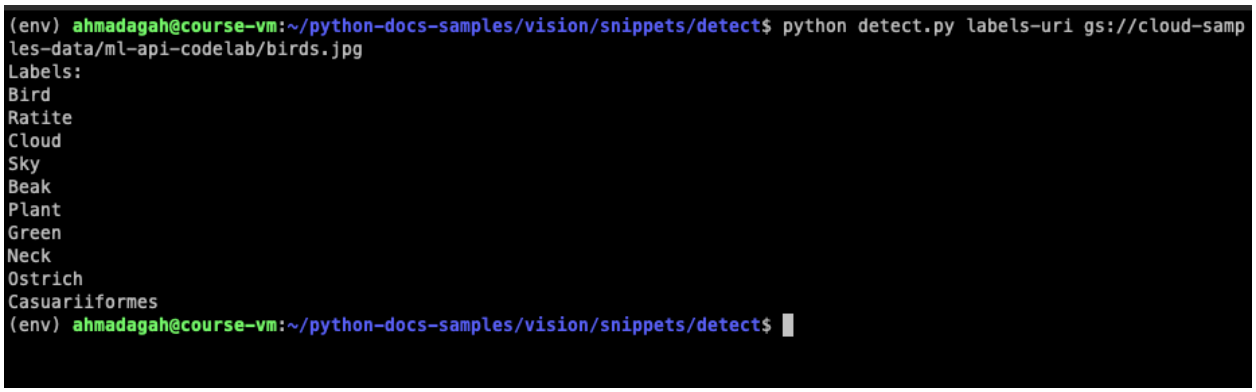
- Take a screenshot of its response for your lab notebook.



07.4g: ML APIs

07.4g.3. Vision

- Show the output for your lab notebook



- What is the name of the function?

The function is detect_labels_uri.

- What type of Vision client is instantiated in it?

The type of Vision client instantiated is vision.ImageAnnotatorClient().

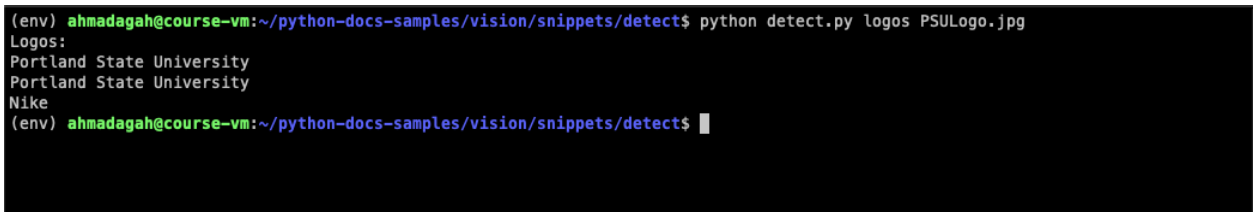
- What method is invoked in the Vision client to perform the detection?

The method invoked is label_detection.

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

The attribute is label_annotations.

- Take a screenshot of the output for the above commands



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

  response = client.logo_detection(image=image)

07.4g.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?

The method used in the Speech client is recognize().

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

The attribute in the response object containing the results is `results`

07.4g.5. Translate

- Show the output for your lab notebook

- What is the name of the function?

The function name is translate_text

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

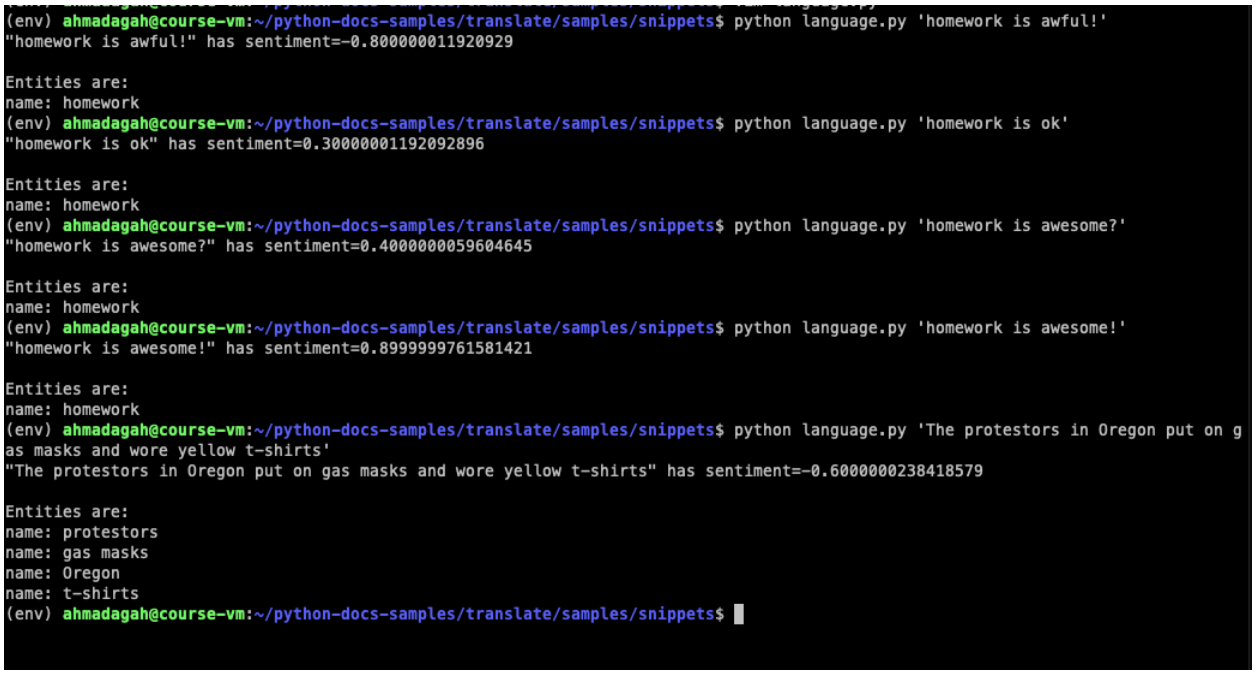
The method invoked is translate from the translate_Client object.

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

The attributes in the response object are "translatedText" for the translated output, "input" for the input text, and "detectedSourceLanguage" for the detected language of the source text.

07.4g.6. Natural Language

- Show the output for your lab notebook



07.4g.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 (Note: This function performs label detection, which identifies objects and labels in an image, rather than analyzing textual entities.)

07.4g.9. Test integration

- If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?
- **Use of Partial Matching**:

- The updated program now uses difflib.SequenceMatcher to compare entities and labels, allowing for partial matches based on a similarity ratio.
- This change ensures that even if the terms are not identical but are sufficiently similar (e.g., "soccer ball" vs. "football"), the program can recognize them as related.

Threshold-Based Matching:

- A threshold value (e.g., > 0.7) was introduced to determine the minimum similarity ratio for a match. This helps the program identify terms that share a high degree of similarity, capturing more relevant relationships between audio and images.

```
 # naive check for whether entities intersect with labels
    has_match = False
    for entity in entities:
        for label in labels:
            # Use SequenceMatcher to check for partial matches with a
threshold
            if difflib.SequenceMatcher(None, entity, label).ratio() > 0.7:
                print('The audio and image both contain:
{}'.format(entity))
                has_match = True
                break
         if has_match:
            break
```

```
Transcription: willst du mit uns Fußball spielen
Translation: do you want to play football with us
Entities: ['football']
Image labels: ['Sports equipment', 'Soccer', 'Football', 'Plant', 'Ball', 'Player', 'Playing sports', 'Socc
er ball', 'Ball game', 'Team sport']
Partial match found: football - Football
The audio and image both contain: football
(env) ahmadagah@course-vm:~/python-docs-samples/translate/samples/snippets$
```

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

  Synonym Matching: A function called `are_synonyms()` was added using WordNet to check if two words are synonyms.
  Semantic Similarity: A function called `are_semantically_similar()` was added using `spaCy` to compare the semantic similarity between words.

```python
def are_synonyms(word1, word2):
    """Check if two words are synonyms using WordNet."""
    synonyms = wordnet.synsets(word1)
    for syn in synonyms:
        if word2 in syn.lemma_names():
            return True
    return False

def are_semantically_similar(word1, word2):
    """Check if two words are semantically similar using spaCy."""
    doc1 = nlp(word1)
    doc2 = nlp(word2)
    return doc1.similarity(doc2) > 0.7  # Adjust the threshold as needed
```

The following piece of code is added to the function compare_audio_to_image()

```python
# naive check for whether entities intersect with labels
    has_match = False
    # Check for partial matches, synonym matches, and semantic similarity
    for entity in entities:
        for label in labels:
            if difflib.SequenceMatcher(None, entity, label).ratio() > 0.7:
                print(f'Partial match found: {entity} - {label}')
```

```
                has_match = True
            elif are_synonyms(entity, label):
                print(f'Synonym match found: {entity} - {label}')
                has_match = True
            elif are_semantically_similar(entity, label):
                print(f'Semantic match found: {entity} - {label}')
                has_match = True
            if has_match:
                print('The audio and image both contain:
{}'.format(entity))
                break
        if has_match:
            break
```

```
Transcription: bisikletimi sokağa bırak
Translation: leave my bike on the street
Entities: ['bike', 'street']
Image labels: ['Bicycle', 'Clothing', 'Footwear', 'Tire', 'Wheel', 'Bicycles--Equipment and supplies', 'Lan
d vehicle', 'Shoe', 'Bicycle frame', 'Bicycle wheel']
Semantic match found: bike - Bicycle wheel
The audio and image both contain: bike
(env) ahmadagah@course-vm:~/python-docs-samples/translate/samples/snippets$
```

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

The function returns that there was a match because changes were made in the previous step.

```
Transcription: çok fazla deve kuşu var
Translation: There are too many ostriches
Entities: ['ostriches']
Image labels: ['Bird', 'Ratite', 'Cloud', 'Sky', 'Beak', 'Plant', 'Green', 'Neck', 'Ostrich', 'Casuariiform
es']
/home/ahmadagah/python-docs-samples/translate/samples/snippets/solution.py:46: UserWarning: [W008] Evaluati
ng Doc.similarity based on empty vectors.
  return doc1.similarity(doc2) > 0.7  # Adjust the threshold as needed
Partial match found: ostriches - Ostrich
The audio and image both contain: ostriches
(env) ahmadagah@course-vm:~/python-docs-samples/translate/samples/snippets$
```

07.4g.13. Video Intelligence

- 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?

The three labels with the highest confidence that the Video Intelligence API associates with the video and their respective confidences are:

1. **Basketball**: Confidence of **0.913787069364929**
2. **Stadium**: Confidence of **0.5119114518165588**
3. **Games**: Confidence of **0.7823163270950317**

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

The client class used in the package is videointelligence.VideoIntelligenceServiceClient.

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

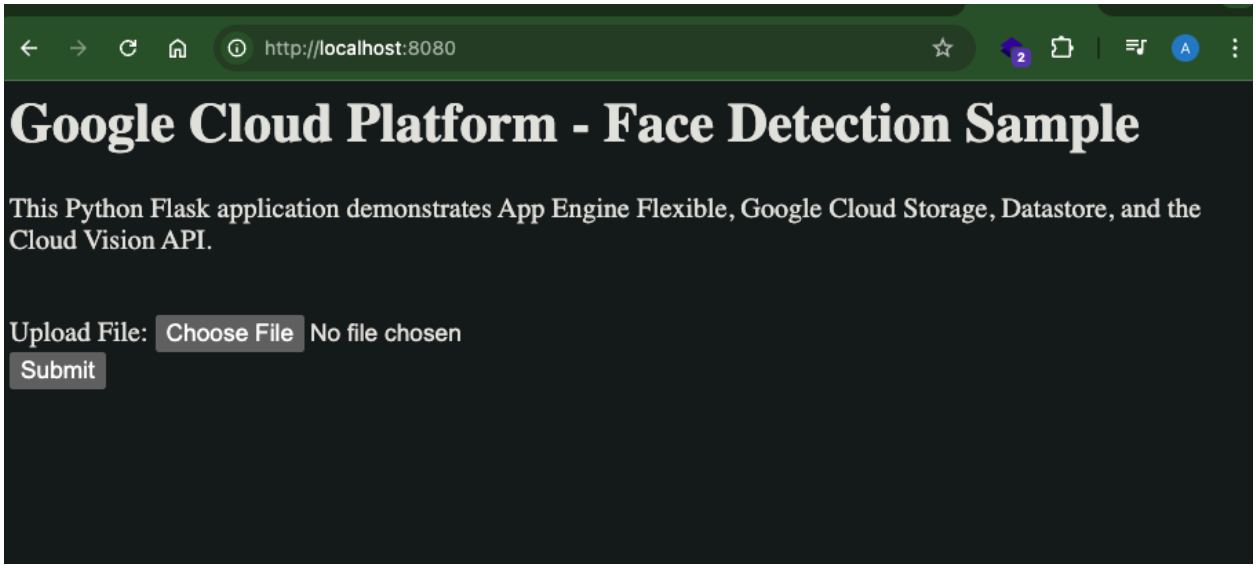
The method used to perform the annotation is annotate_video().

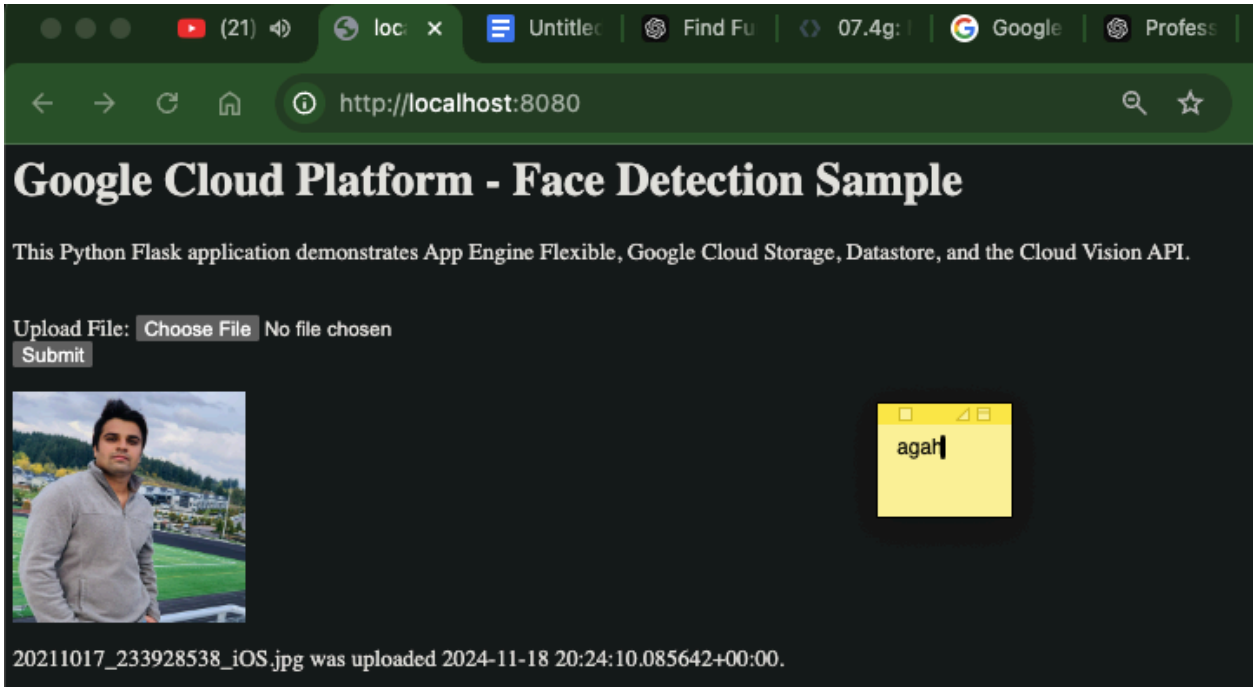
07.4g.16. Application

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

I am using local port forwarding to a Compute Engine instance with the following command. I'm just letting you know in case any URLs don't match what you expect. I don't think Cloud Shell is sufficient for understanding these concepts, so I use a Compute Engine ARM instance instead.

```
ssh -L [local_port]:[remote_server]:[remote_port] user@remote-server
```





07.4g.17. Code

- 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?

blob_name
image_public_url
timestamp
joy

- What happens at the end of the upload_photo route?

The user is redirected to the homepage:
return redirect("/")