Contents

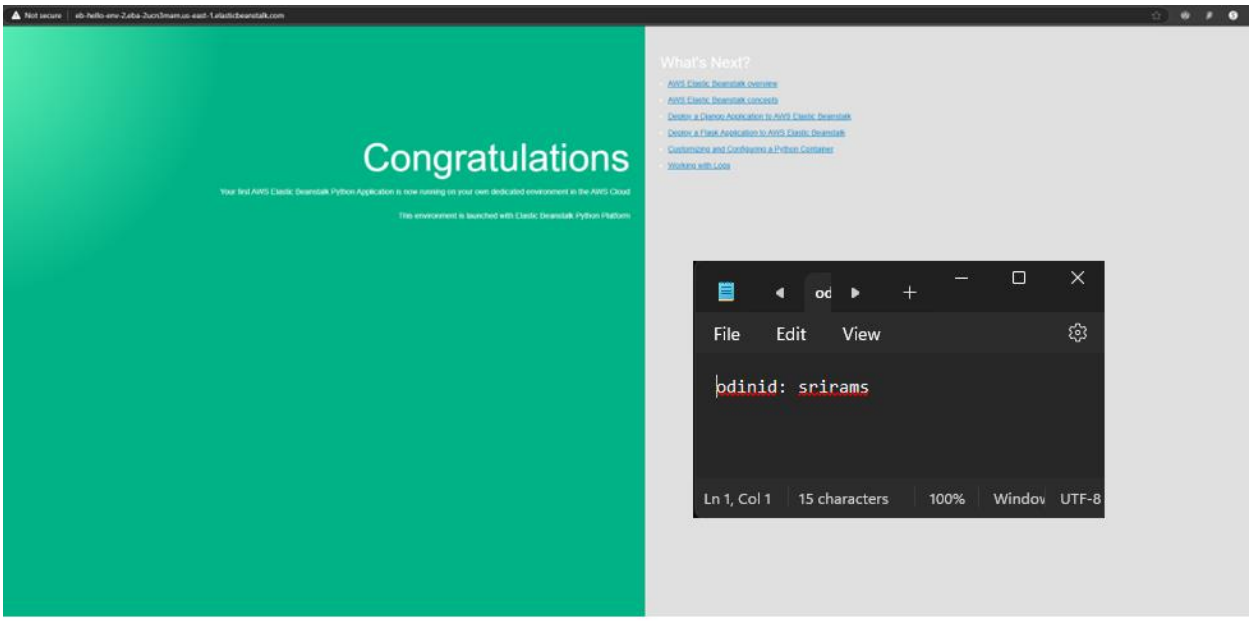
6.1a: EB Guestbook

6.1 Elastic Beanstalk

6.1.2 Elastic Beanstalk sample application

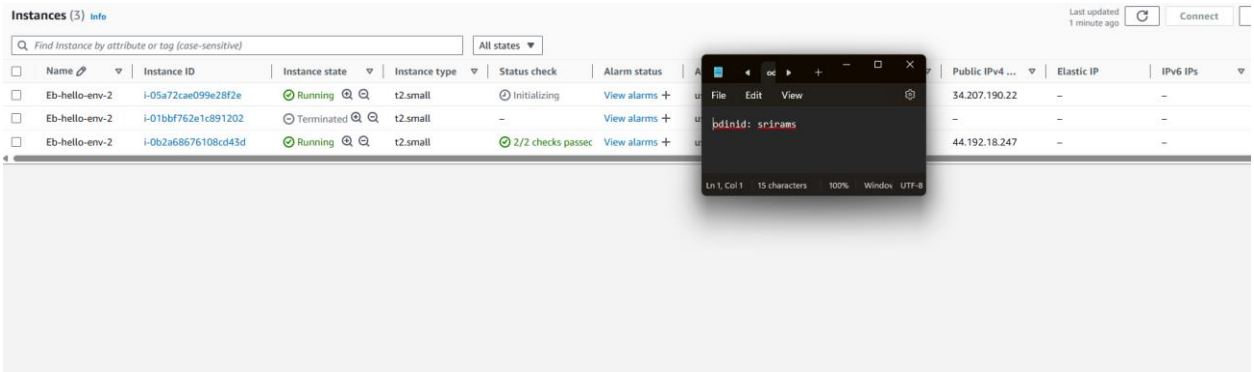
6.1.3 Running the application

- **Take a screenshot showing it has been brought up successfully**



6.1.4 Handling failures seamlessly

- **Take a screenshot of the replacement VM being started.**

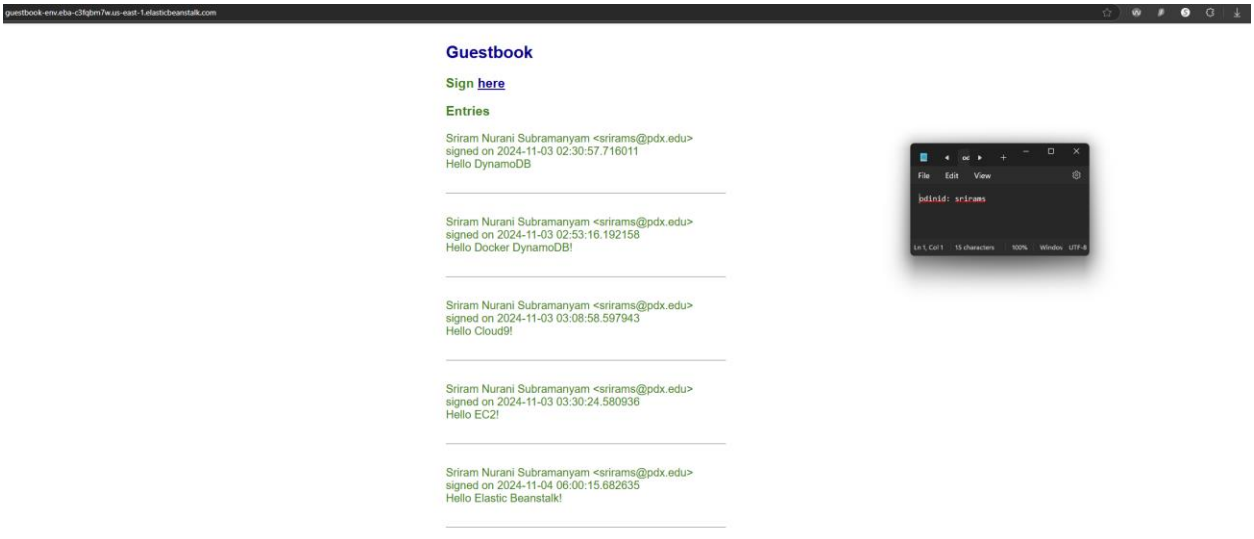


6.1.5 Clean up

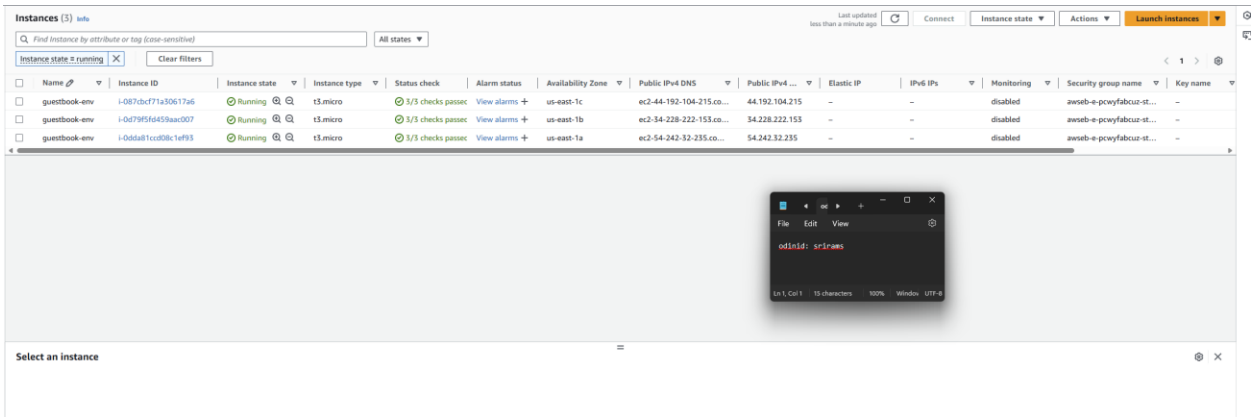
6.1.6 Elastic Beanstalk Guestbook

6.1.7 Deploying the Guestbook

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



- **Take a screenshot of them.**



6.1.8 Clean up

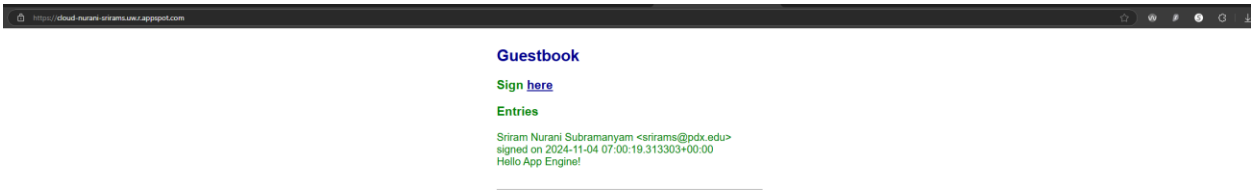
6.1g: App Engine Guestbook

6.1.1 App Engine

6.1.2 app.yaml

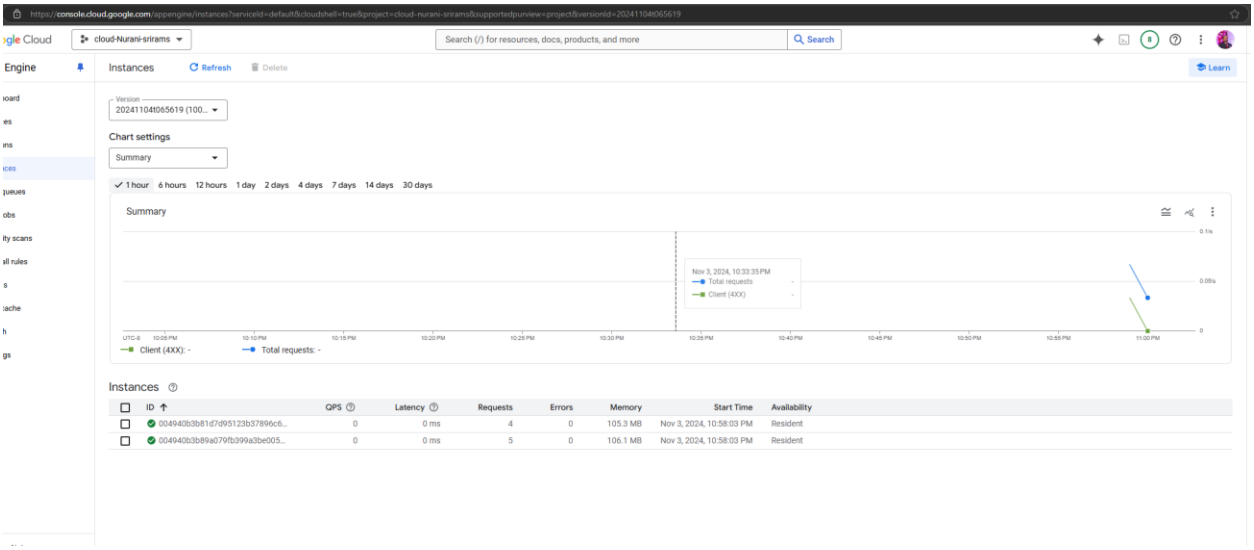
6.1.3 Deploying the Guestbook

- **Take a screenshot of the output that includes the URL in the address bar for your lab notebook.**



6.1.4 Handling failures seamlessly

- **Take a screenshot of them.**



6.1.5 Clean up

6.2g: Cloud Run, Secret Manager (Web proxy)

6.2.1 Cloud Run

6.2.2. Application

6.2.3. app.py

6.2.4. templates/proxy.html

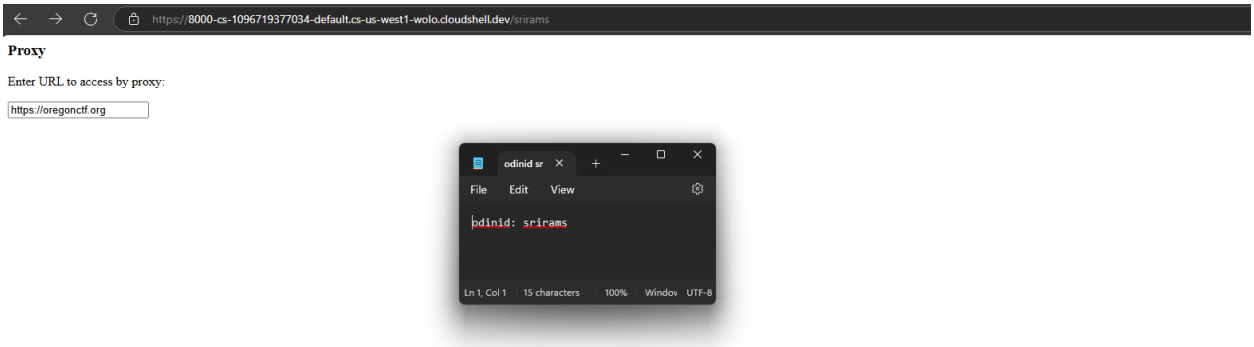
6.2.5. requirements.txt

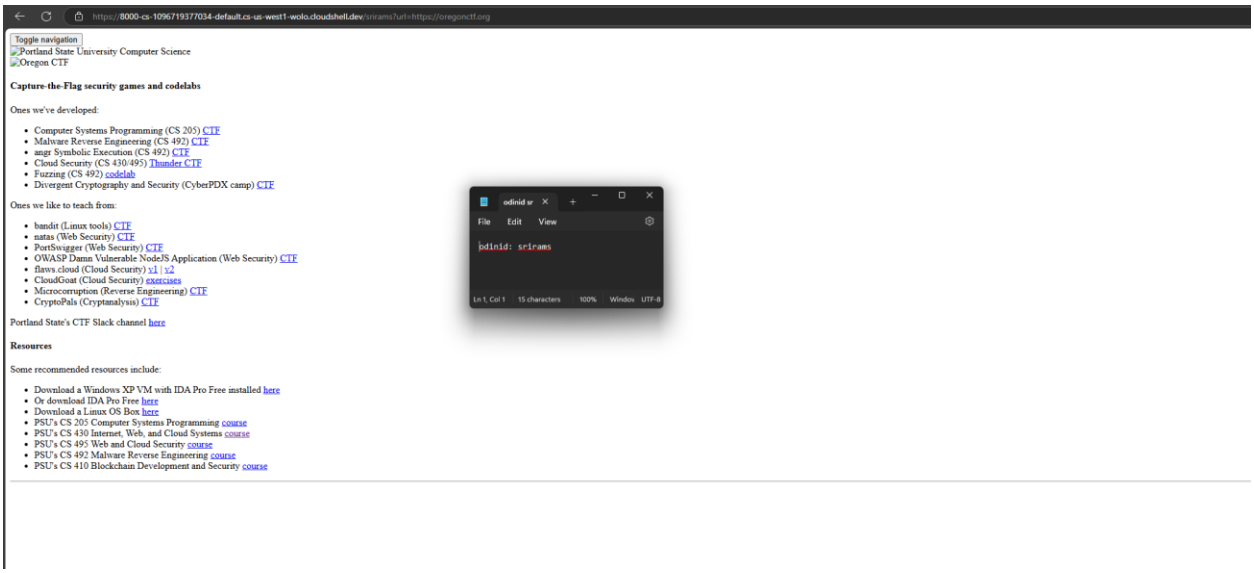
6.2.6. Dockerfile

6.2.7. Build and test in Cloud Shell

6.2.8. Setup secret proxy

- **Take a screenshot of the proxy and its results including the URL containing your OdinID**





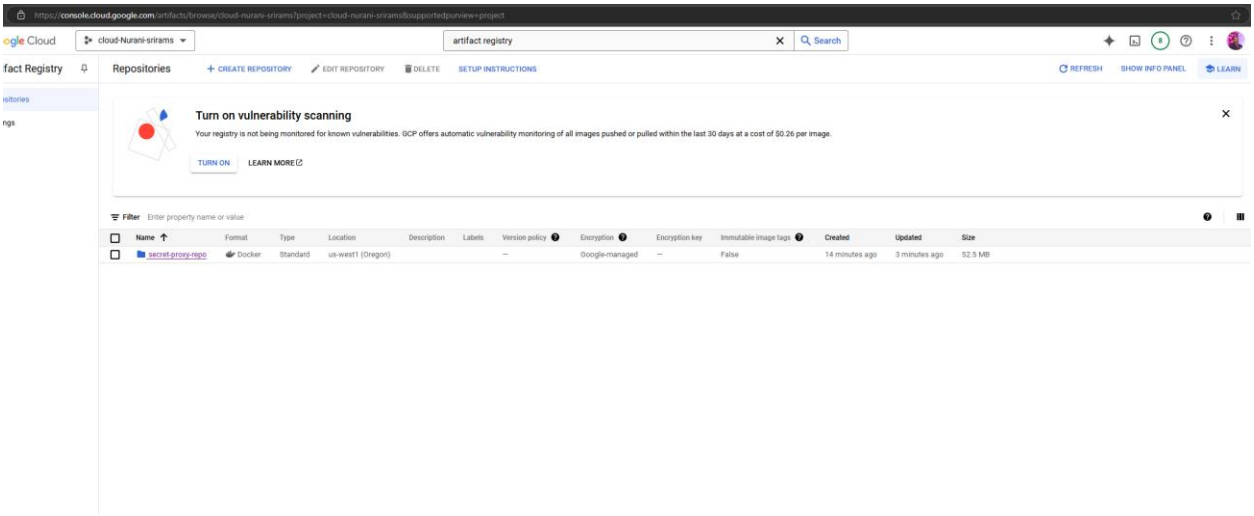
- **What is the security advantage of passing in the secret proxy route as an environment variable?**

using environment variables for passing sensitive information like secret proxy routes enhances the security posture of your application by reducing exposure, enabling dynamic configuration, and aligning with security best practices.

6.2.9. Artifact Registry

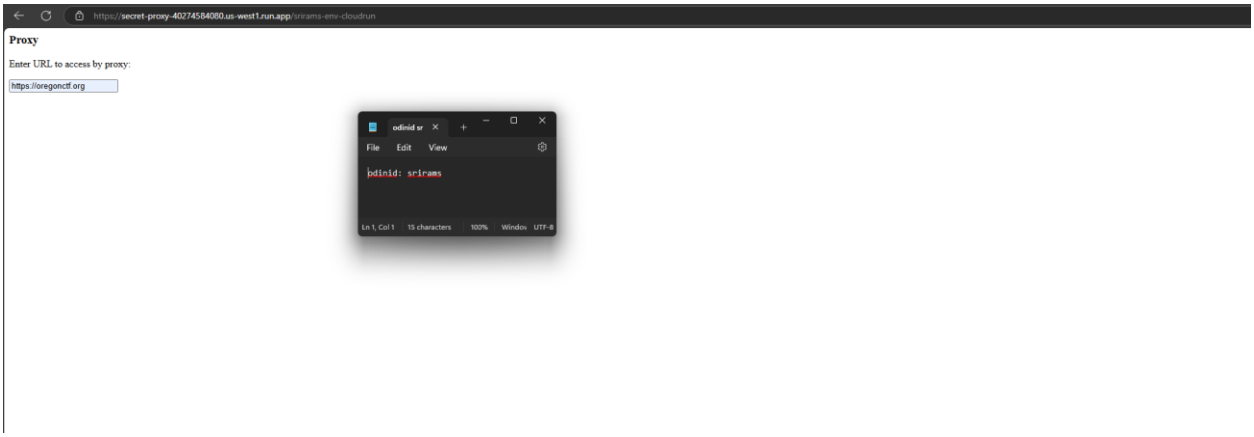
6.2.10. Cloud Build

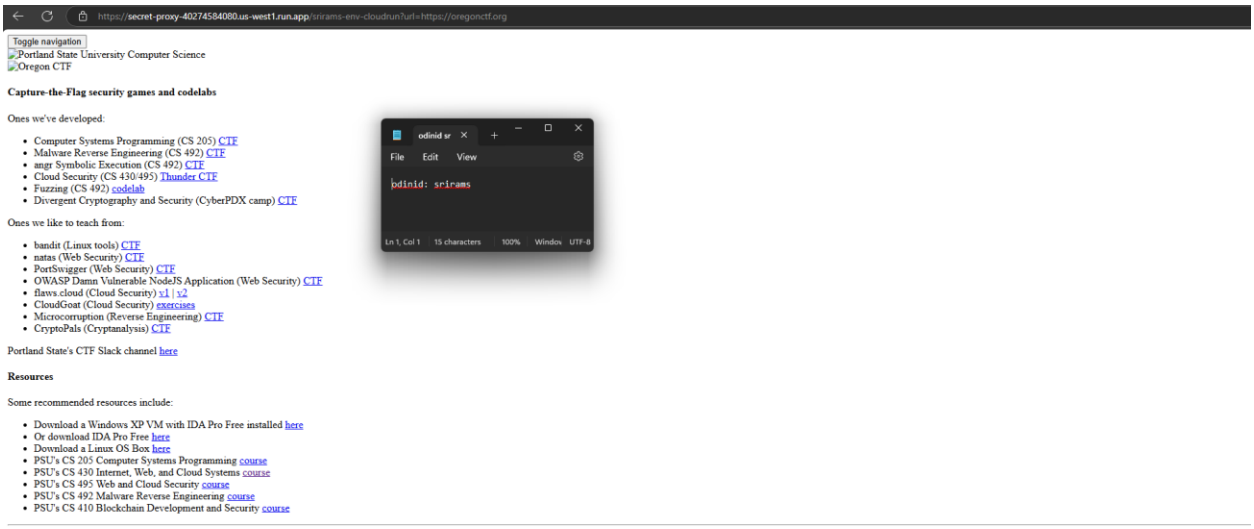
- **Take a screenshot of the image in the registry that shows the size of the container for your lab notebook.**



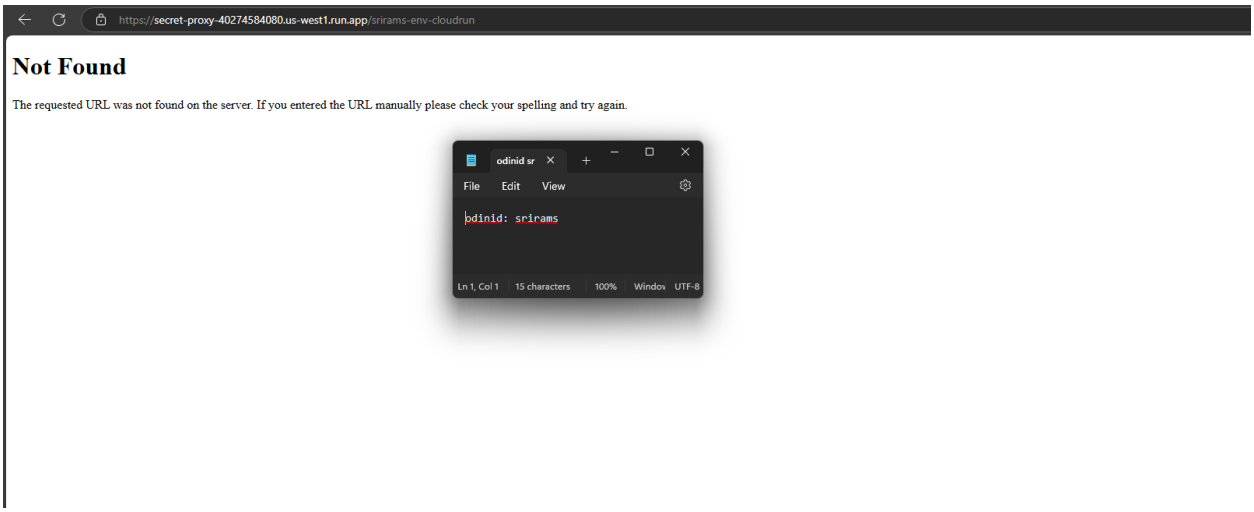
6.2.11. Deploy to Cloud Run

- **Take a screenshot of it that includes the proxy URL for your lab notebook.**





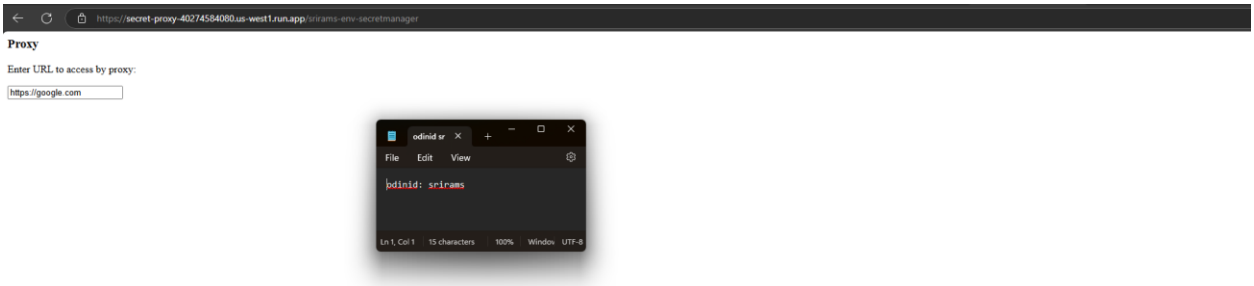
- **Take a screenshot of the error page that includes the proxy URL for your lab notebook.**

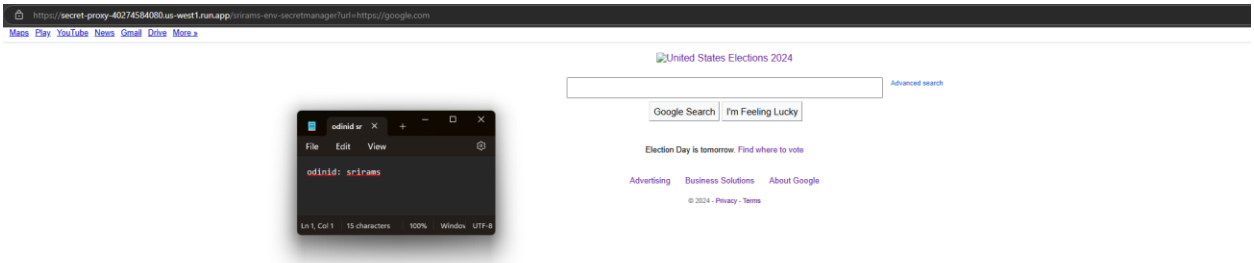


6.2.12. Secret manager

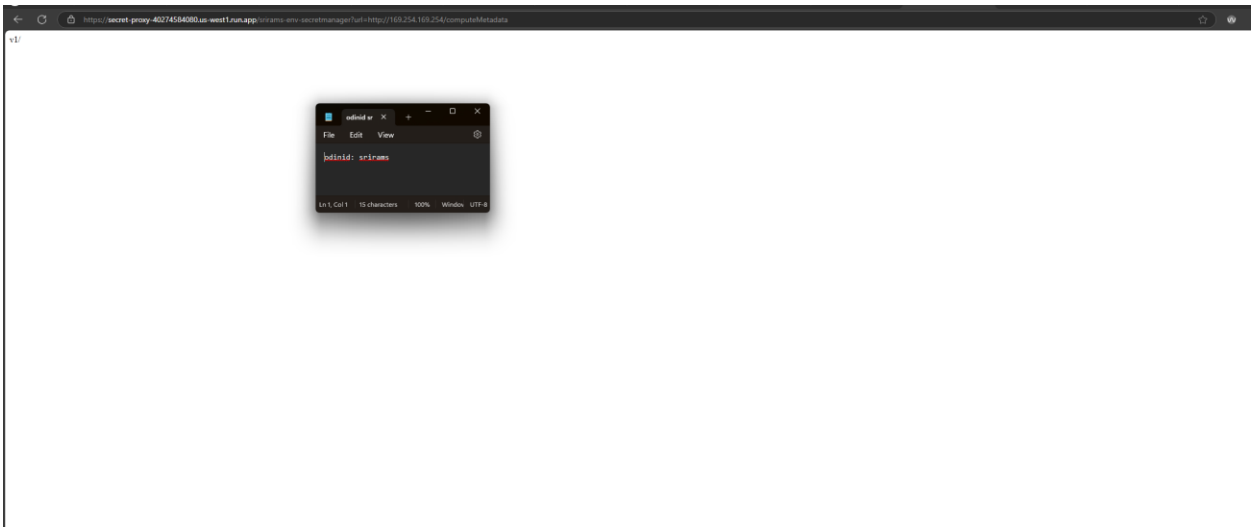
6.2.13. Deploy to Cloud Run with Secret Manager

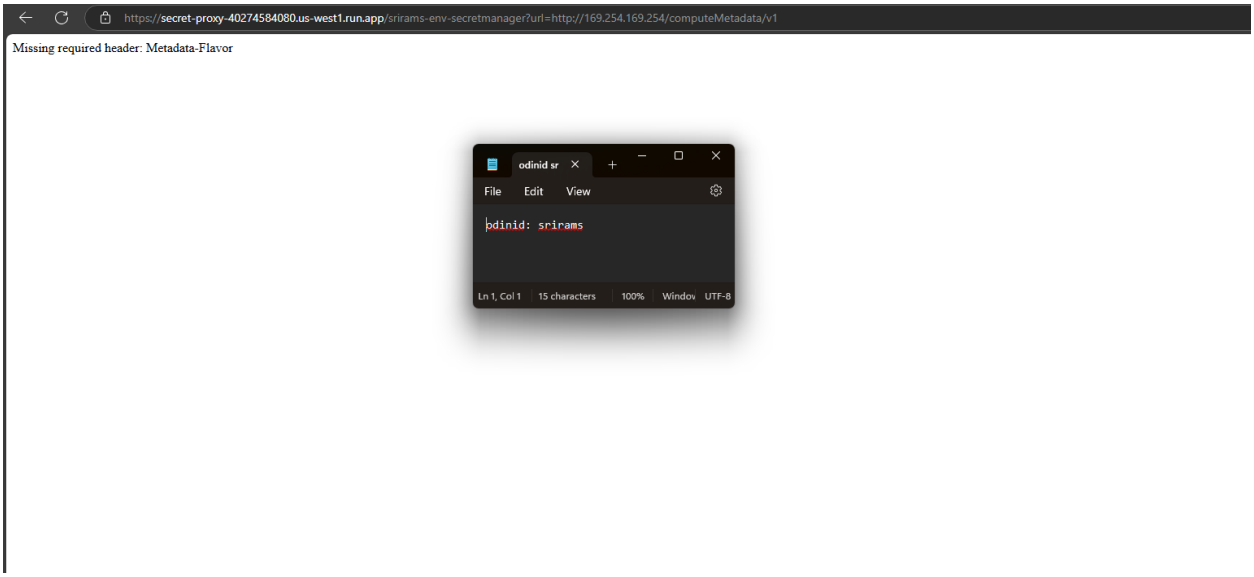
- **Take a screenshot of it that includes the proxy URL for your lab notebook.**





- **Take a screenshot of it that includes the proxy URL for your lab notebook.**





- **Identify the vulnerability in your lab notebook that Google has prevented.**

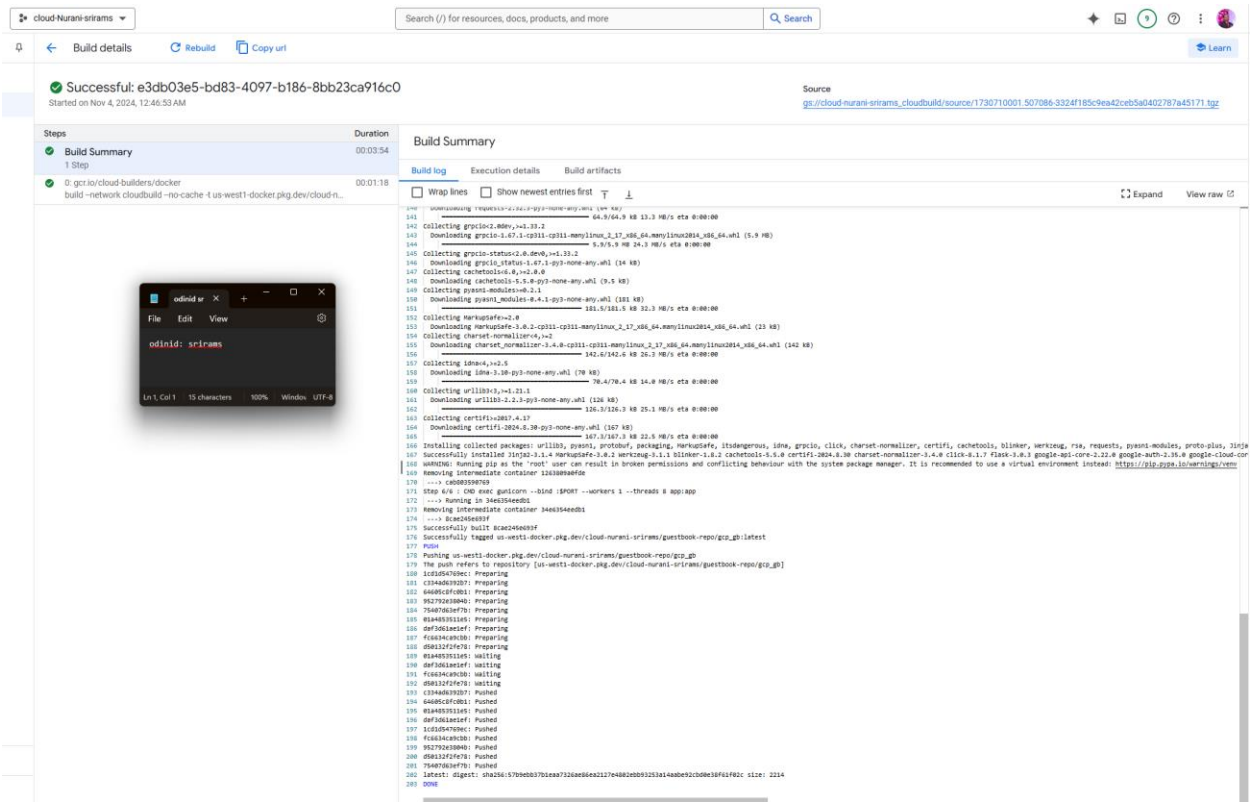
  Web security vulnerability, SSRF- Server-side request forgery

6.3g: Cloud Run Guestbook

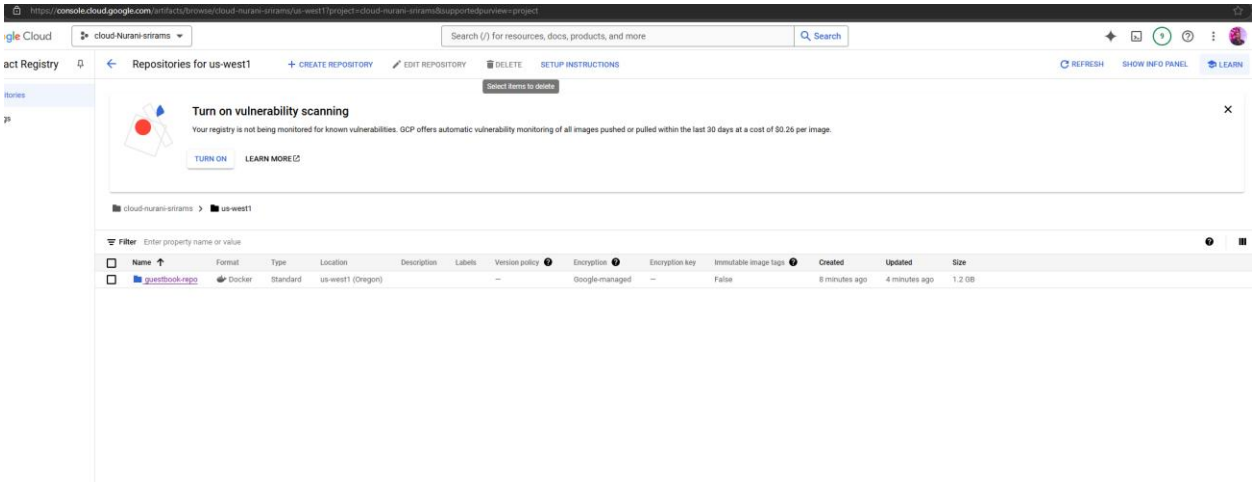
6.3.1 Cloud Run

6.3.2 Prepare a container image

- **Take a screenshot that includes the output of the command and the time it took to execute.**



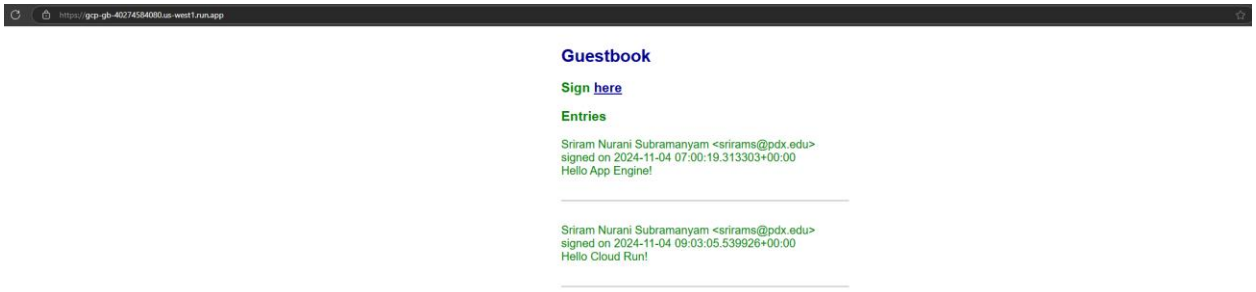
- **Take a screenshot showing the container image and its virtual size**



6.3.3Deploy container with minimal privileges

6.3.4View the Guestbook

- **Take a screenshot that includes the URL Cloud Run has created for your site.**

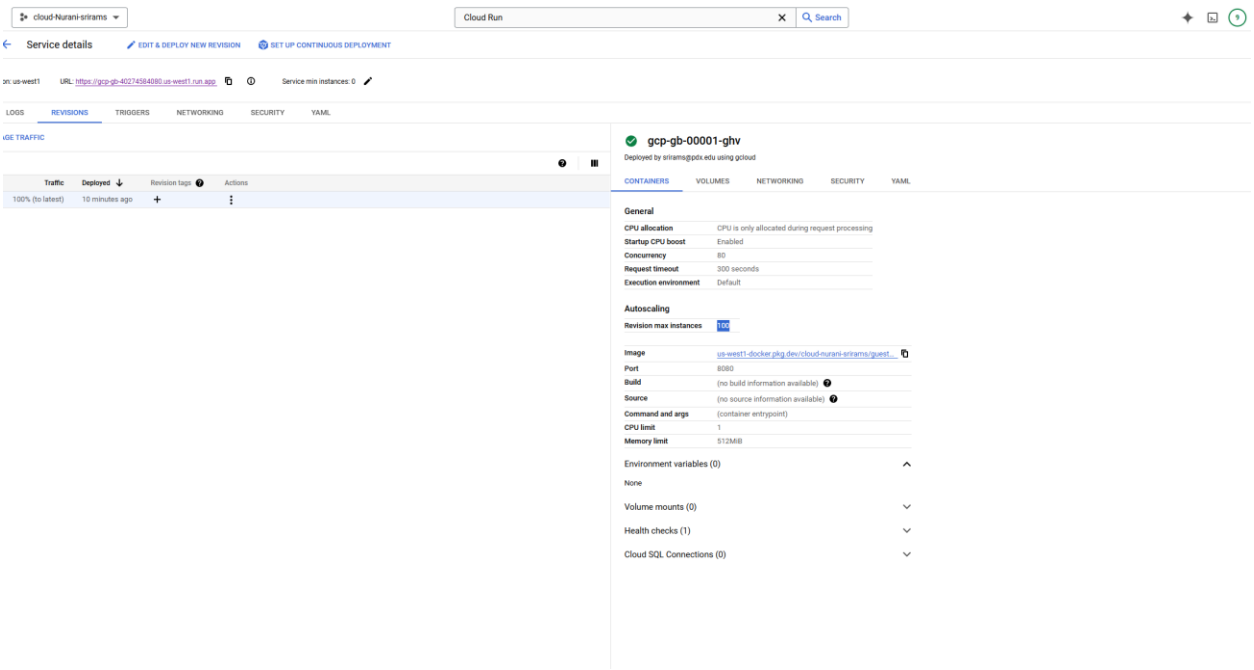


- **What port do container instances listen on?**

  8080

- **What are the maximum number of instances Cloud Run will autoscale up to for your service?**

100

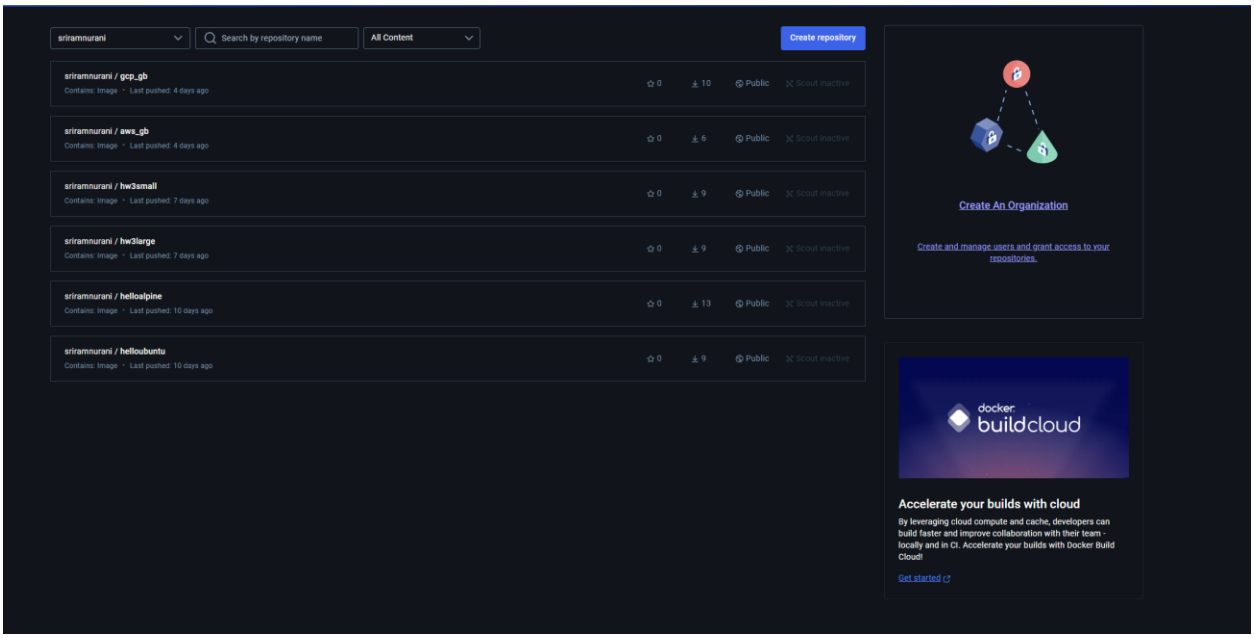


6.3.5Clean Up

6.3a: ECS Guestbook

6.3.1 Prepare a container image

- **Show that your image was uploaded to your account on Docker Hub.**



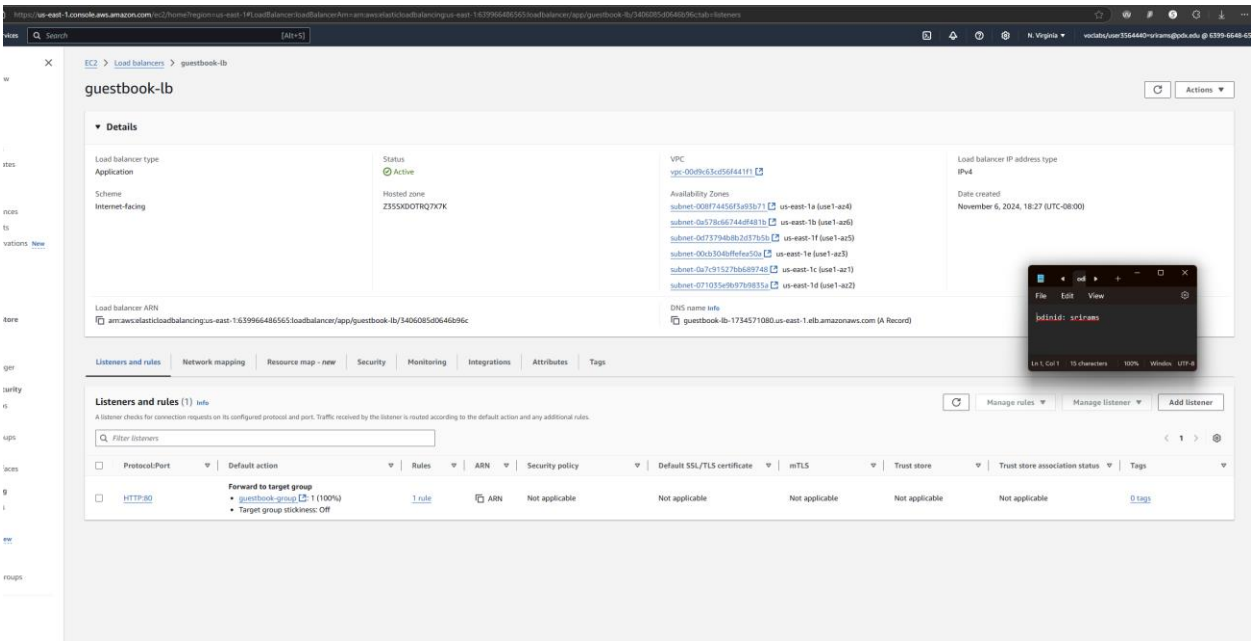
6.3.2 ECS overview

6.3.3 ECS task definition

6.3.4 Create a Cluster and a Service

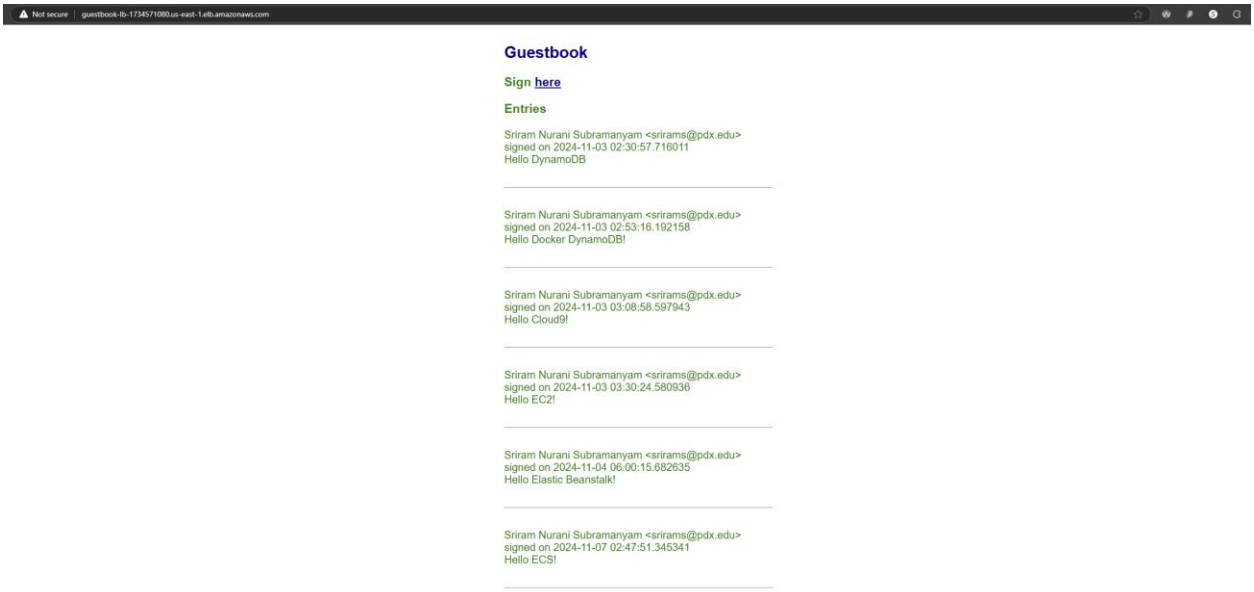
6.3.5 Examine the service

- **Take a screenshot of the DNS name of the guestbook-lb load balancer for your lab notebook**



6.3.6 Visit the site

- **Take a screenshot of the Guestbook app running in a browser that includes the DNS name of the site.**



6.3.7 Clean Up

6.4g: Cloud Functions, PubSub

6.4.1 Cloud Functions image blurring

6.4.2 Services setup

6.4.3 Code

6.4.4 -

- **After downloading the file from the bucket, where is it stored?**

  Temporary file (temp_local_filename)

- **What class in the ImageMagick package is used to do the blurring of the file?**

  Class : Image and Method : resize

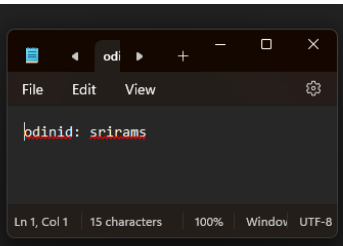
- **What lines of code perform the blurring of the image and its storage back into the filesystem?**

```
# Blur the image using ImageMagick.
with Image(filename=temp_local_filename) as image:
    image.resize(*image.size, blur=16, filter="hamming")
    image.save(filename=temp_local_filename)

print(f"Image {file_name} was blurred.")

# Upload result to a second bucket, to avoid re-triggering the function.
# You could instead re-upload it to the same bucket + tell your function
# to ignore files marked as blurred (e.g. those with a "blurred" prefix)
blur_bucket_name = os.getenv("BLURRED_BUCKET_NAME")
blur_bucket = storage_client.bucket(blur_bucket_name)
new_blob = blur_bucket.blob(file_name)
new_blob.upload_from_filename(temp_local_filename)
print(f"Blurred image uploaded to: gs://{blur_bucket_name}/{file_name}")

# Delete the temporary file.
os.remove(temp_local_filename)
```

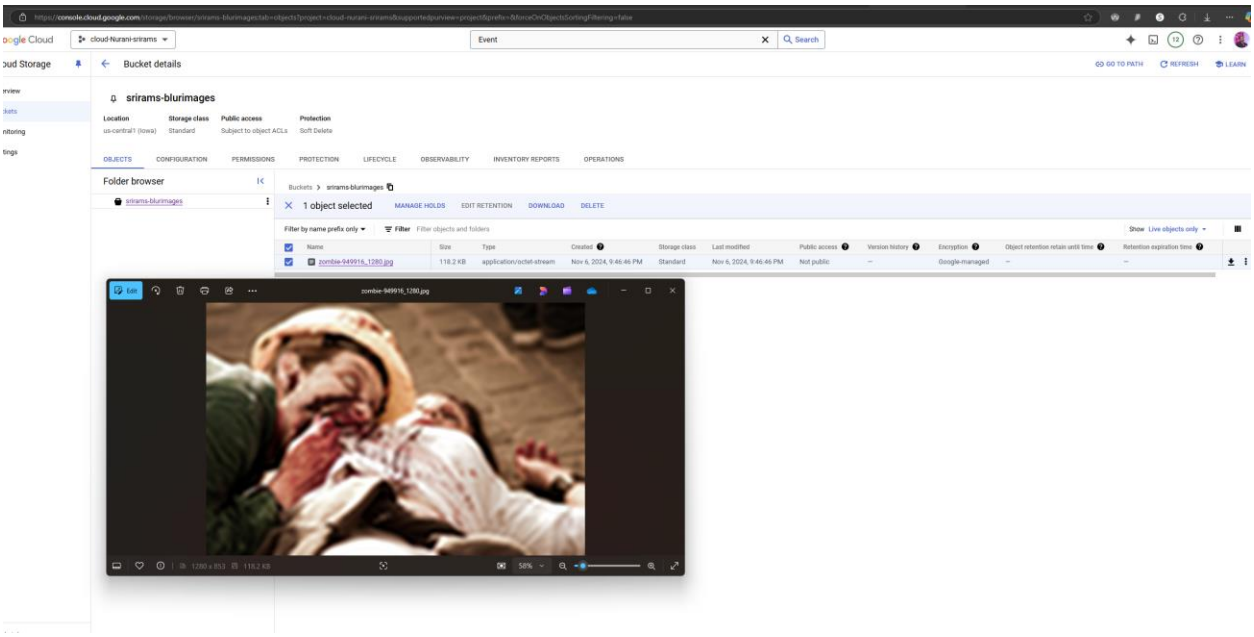


6.4.5 Set up service account

6.4.6 Deploy the function

6.4.7 Test function

- **Take a screenshot of the blurred image in the output bucket for your lab notebook**



- **Include a screenshot of the output logs that show that the above image was blurred.**

```
srirams@cloudshell:~/python-docs-samples/functions/imagemagick (cloud-nurani-srirams)$ gcloud functions logs read
LEVEL:
NAME: blur-offensive-images
EXECUTION_ID: hl3ILKBKjeLP
TIME_UTC: 2024-11-07 05:46:46.701
LOG: Blurred image uploaded to: gs://srirams-blurimages/zombie-949916_1280.jpg

LEVEL:
NAME: blur-offensive-images
EXECUTION_ID: hl3ILKBKjeLP
TIME_UTC: 2024-11-07 05:46:46.616
LOG: Image zombie-949916_1280.jpg was blurred.

LEVEL:
NAME: blur-offensive-images
EXECUTION_ID: hl3ILKBKjeLP
TIME_UTC: 2024-11-07 05:46:45.792
LOG: Image zombie-949916_1280.jpg was downloaded to /tmp/tmpywbb9zg7.

LEVEL:
NAME: blur-offensive-images
EXECUTION_ID: hl3ILKBKjeLP
TIME_UTC: 2024-11-07 05:46:45.702
LOG: The image zombie-949916_1280.jpg was detected as inappropriate.

LEVEL:
NAME: blur-offensive-images
EXECUTION_ID: hl3ILKBKjeLP
TIME_UTC: 2024-11-07 05:46:45.452
LOG: Analyzing zombie-949916_1280.jpg.

LEVEL: I
NAME: blur-offensive-images
EXECUTION_ID:
TIME_UTC: 2024-11-07 05:46:45.315
LOG: Default STARTUP TCP probe succeeded after 1 attempt for container "worker" on port 8080.

LEVEL: I
NAME: blur-offensive-images
EXECUTION_ID:
TIME_UTC: 2024-11-07 05:46:43.288
LOG:
```

6.4.8 Clean up

6.4.9 PubSub

6.4.10 Create VM for subscriber

6.4.11 PubSub via CLI

- **Why are there no items returned?**

Since there is no Subscription when message was published, the subscription would not have the message in it.

6.4.12 -

- **What is the messageId of the published message?**

12536959487920718

```
srirams@cloudshell:~ (cloud-nurani-srirams)$ gcloud pubsub topics publish topic-srirams --message="Message #2"
messageIds:
- '12536959487920718'
srirams@cloudshell:~ (cloud-nurani-srirams)$
```

- **Take a screenshot of the output of the successful pull that includes the message and its messageId.**

6.4.13 PubSub via Python

6.4.14 -

6.4.15 Test programs and clean up

- **Take a screenshot showing the messageIds and messages sent**

```
(env) srirams@cloudshell:~ (cloud-nurani-srirams)$ python3 publisher.py
Topic 'projects/cloud-nurani-srirams/topics/topic-srirams' already exists.
Enter a message to send: This is Sriram
Published 12538487302386034 to topic projects/cloud-nurani-srirams/topics/topic-srirams
Enter a message to send: I am studying Pub/Sub in IWC
Published 12538547264054017 to topic projects/cloud-nurani-srirams/topics/topic-srirams
Enter a message to send: I plan to graduate from PSU in June 2025
Published 12892348375260455 to topic projects/cloud-nurani-srirams/topics/topic-srirams
Enter a message to send:
```

- **Take a screenshot showing the same messageIds and messages received**

```
(env) srirams@pubsub:~$ nano subscriber.py
(env) srirams@pubsub:~$ python3 subscriber.py
Using existing subscription.
Listening for messages on projects/cloud-nurani-srirams/subscriptions/sub-srirams...
Received message 12536959487920718: Message #2
Received message 12538487302386034: 2024-11-07 07:47:40 (projects/cloud-nurani-srirams/topics/topic-srirams) : This is Sriram
Received message 12538547264054017: 2024-11-07 07:47:53 (projects/cloud-nurani-srirams/topics/topic-srirams) : I am studying Pub/Sub in IWC
Received message 12892348375260455: 2024-11-07 07:48:08 (projects/cloud-nurani-srirams/topics/topic-srirams) : I plan to graduate from PSU in June 2025
```