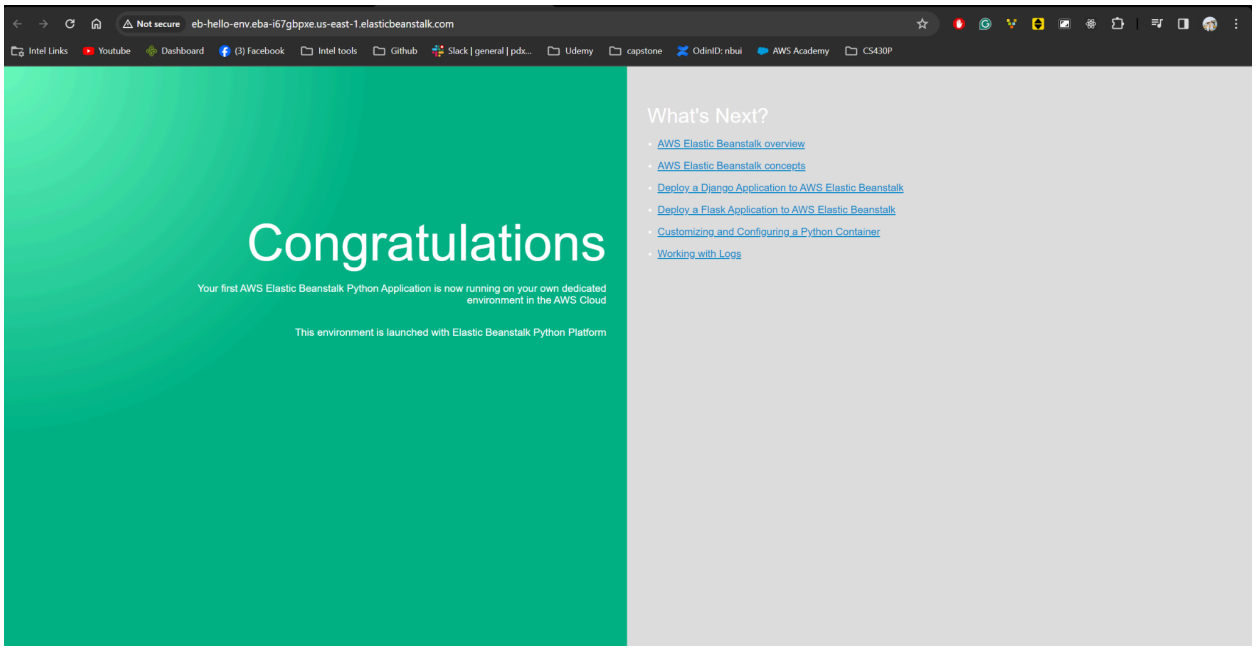
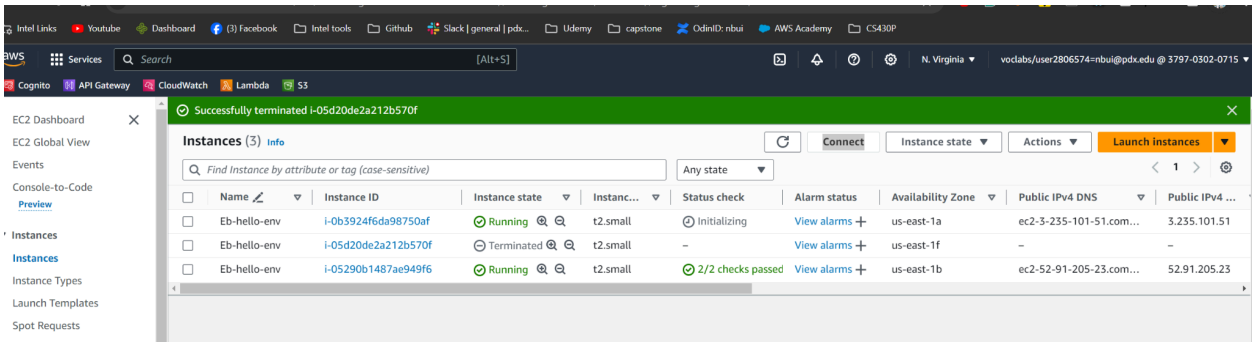
Lab 06.1a

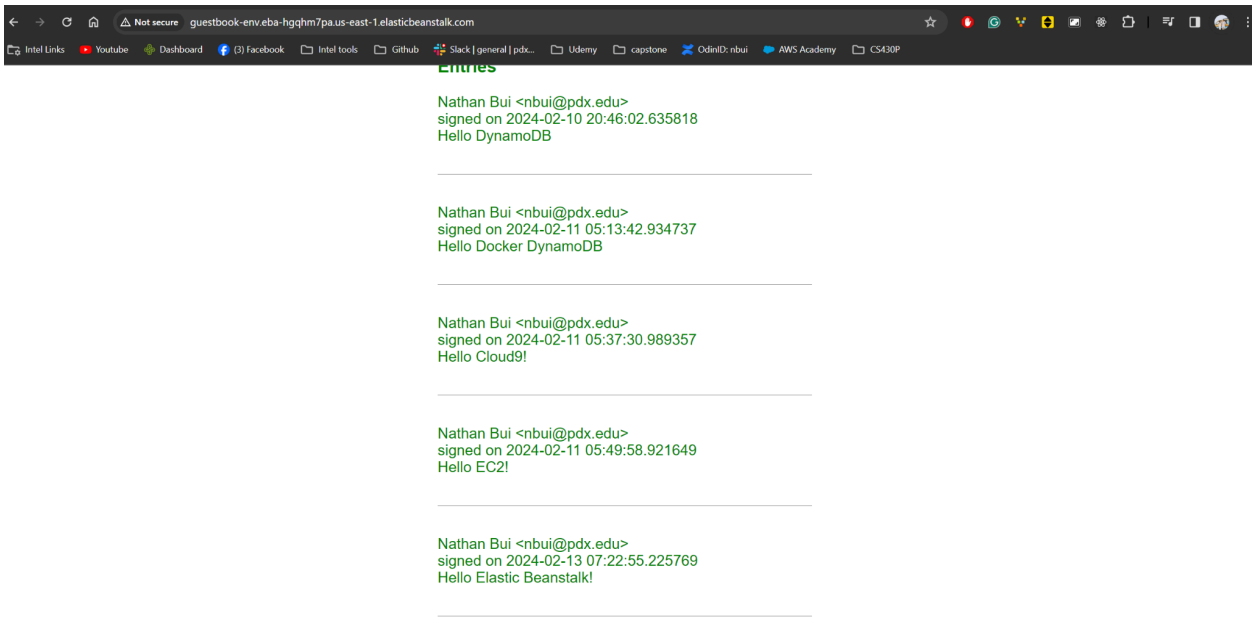
3. Running the application

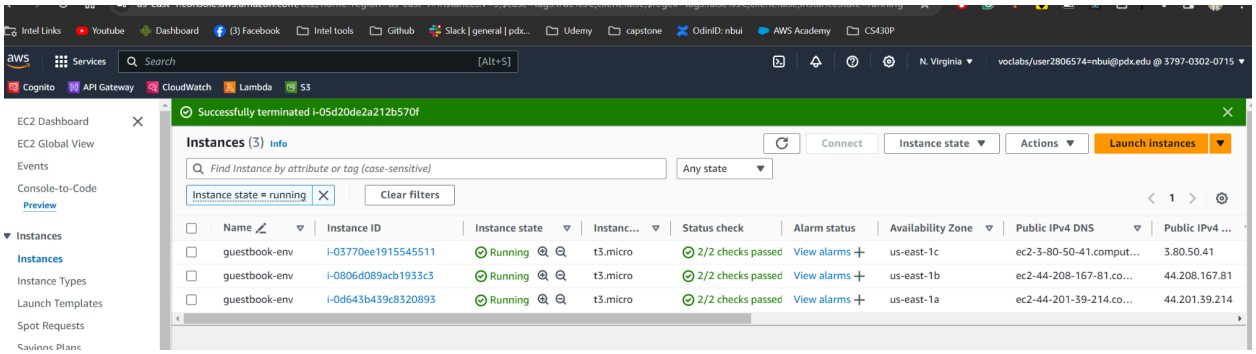


4. Handling failures seamlessly



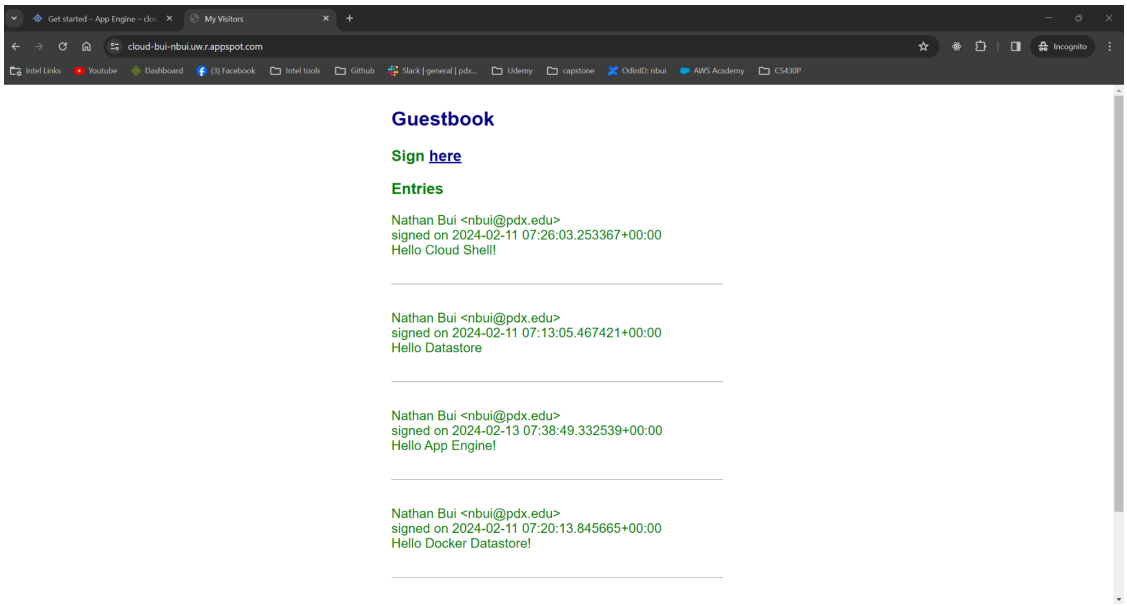
7. Deploying the Guestbook



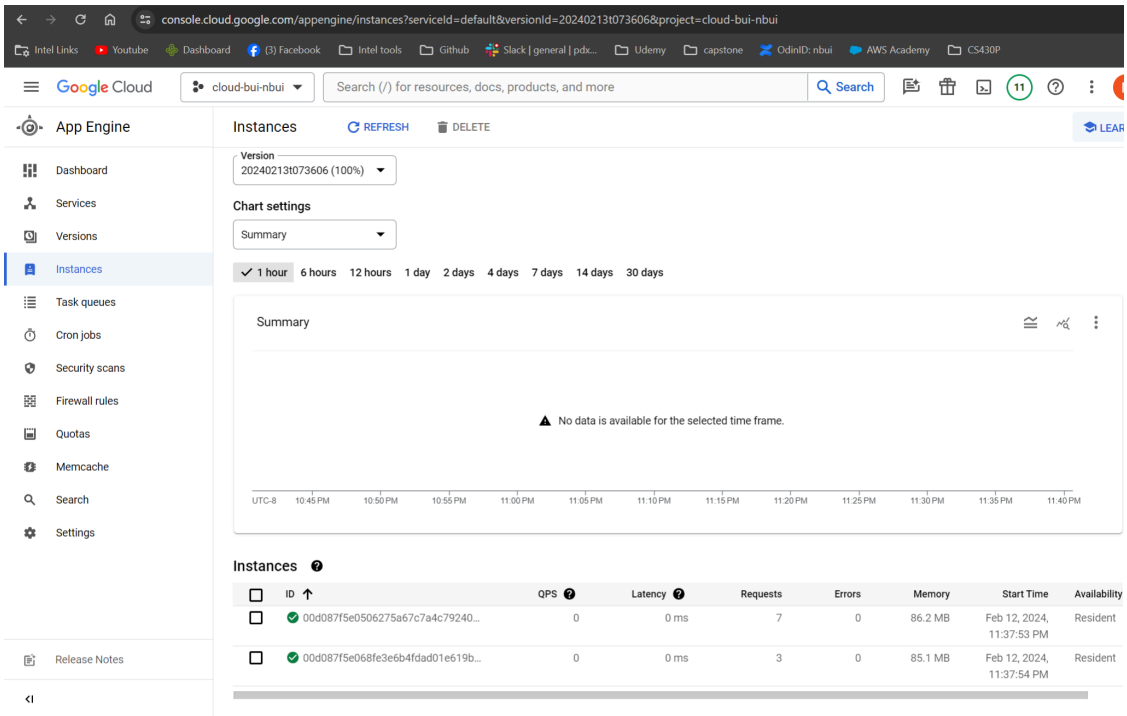


Lab 06.1g

3. Deploying the Guestbook



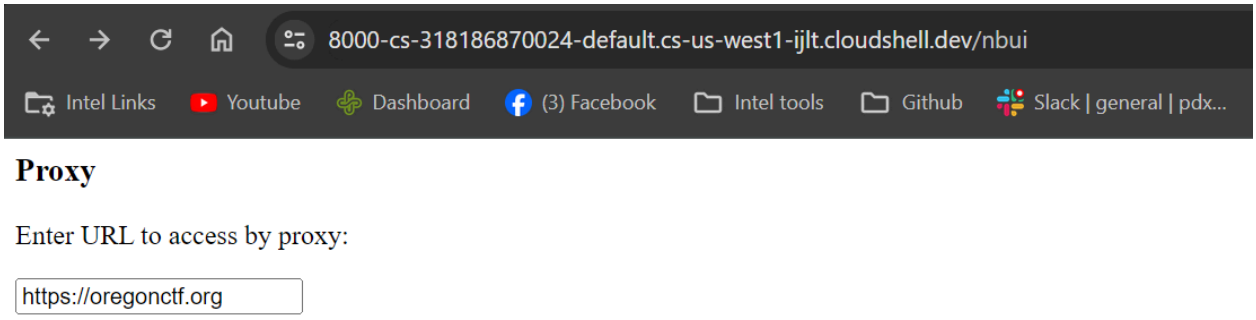
4. Handling failures seamlessly



Lab 06.2g

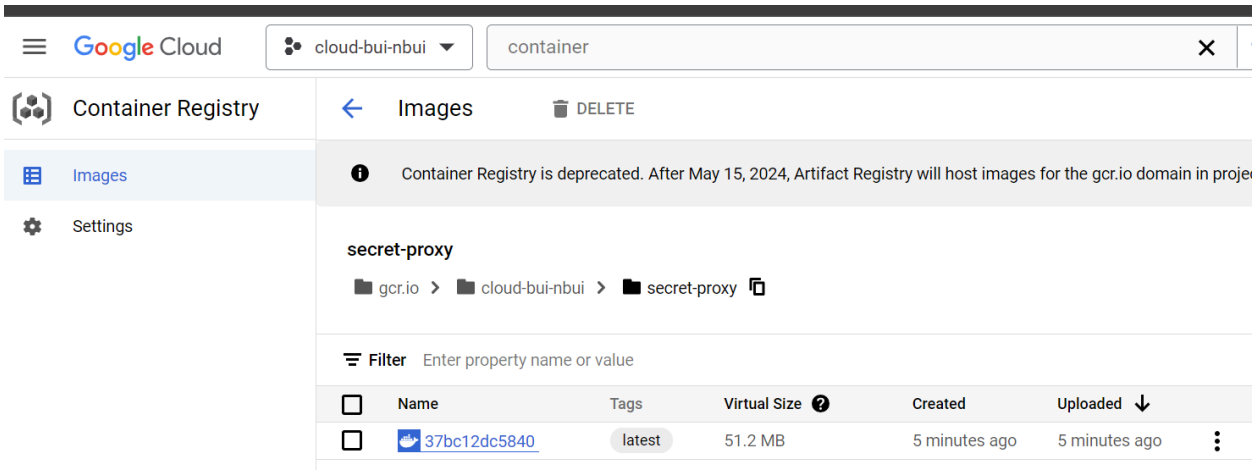
8. Setup secret proxy

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



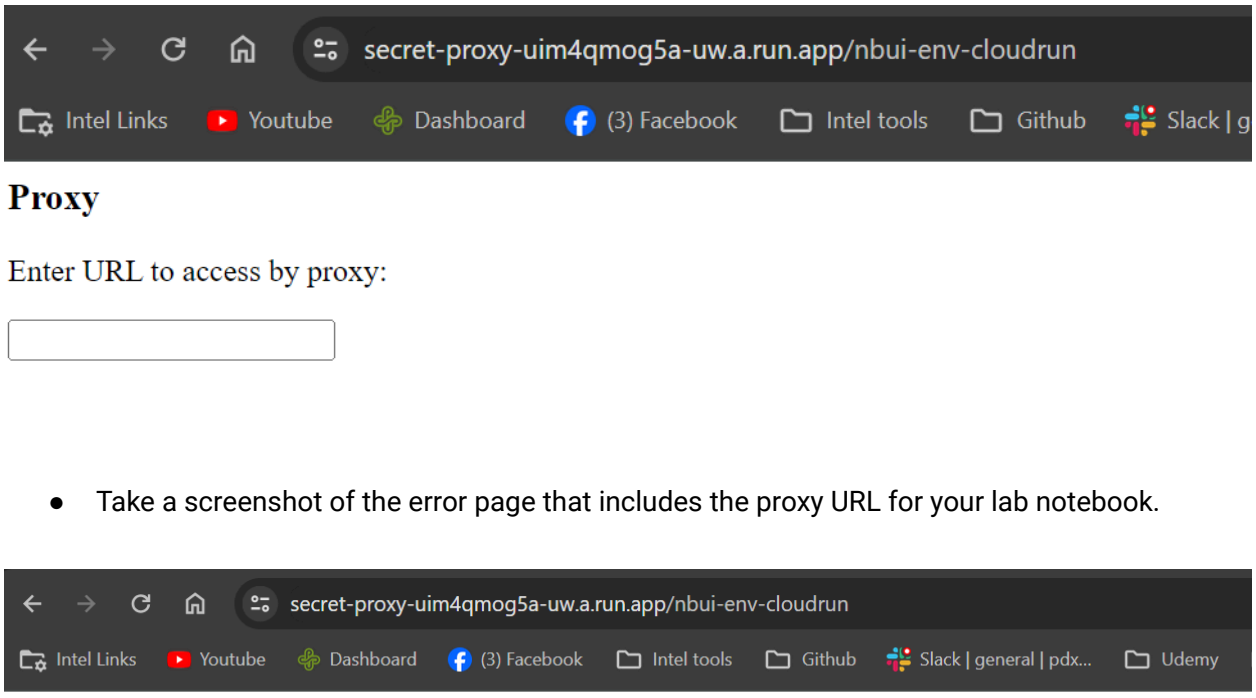
- What is the security advantage of passing in the secret proxy route as an environment variable?
    - Ensure that the secret is not hardcoded into the code so it can't be exploited. It helps to separate the secret information and the application in the same place.
    - Provide the flexibility to change/update the secret

9. Cloud Build and Container Registry

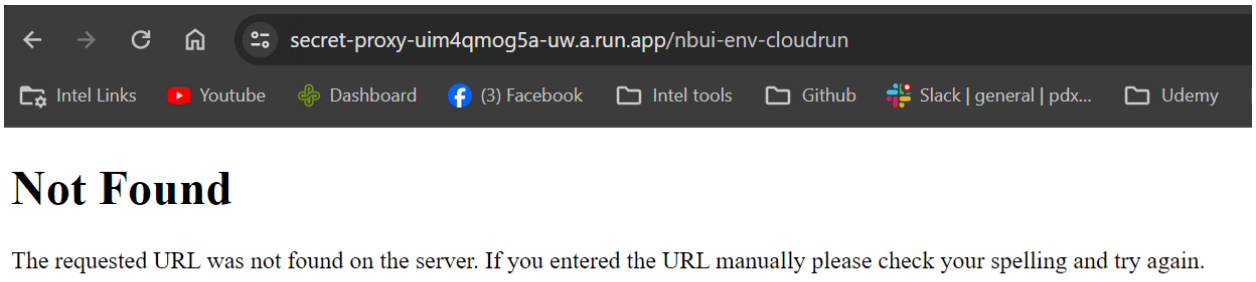


10. Deploy to Cloud Run

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

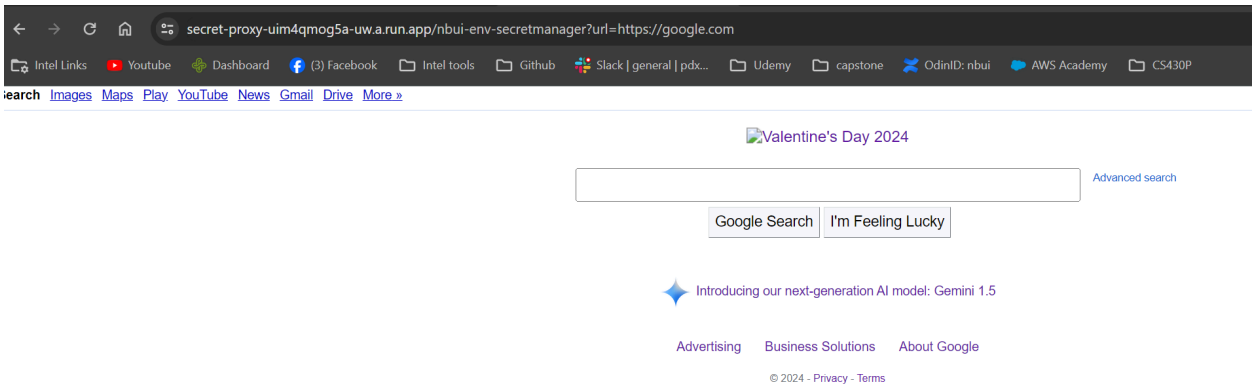


- Take a screenshot of the error page that includes the proxy URL for your lab notebook.



12. Deploy to Cloud Run with Secret Manager

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

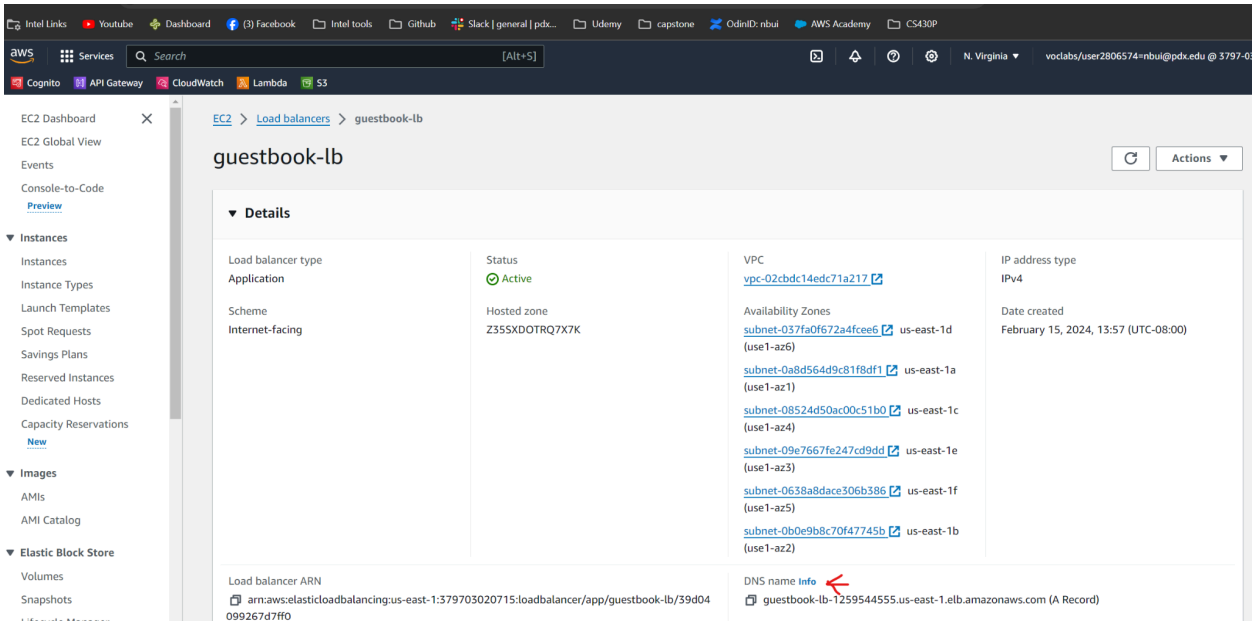


- Identify the vulnerability in your lab notebook that Google has prevented.
    - SSRF attack with Metadata endpoint bug when trying to access unauthorized resources within the platform/provider infrastructure

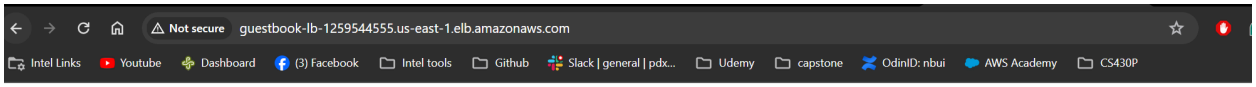
Lab 06.3a

5. Examine the service

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



6. Visit the site



Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 05:13:42.934737
Hello Docker DynamoDB

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 05:37:30.989357
Hello Cloud9!

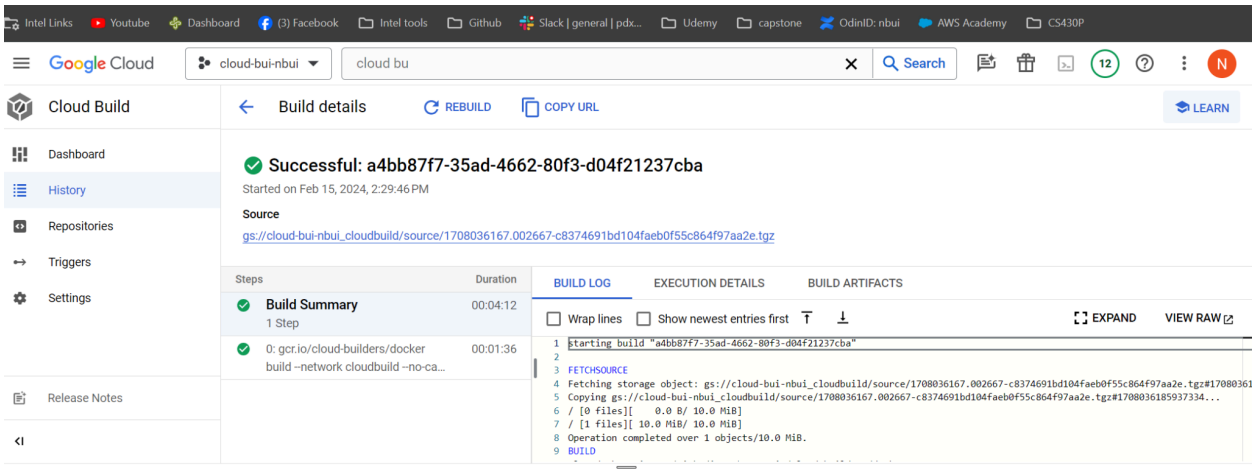
Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 05:49:58.921649
Hello EC2!

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-13 07:22:55.225769
Hello Elastic Beanstalk!

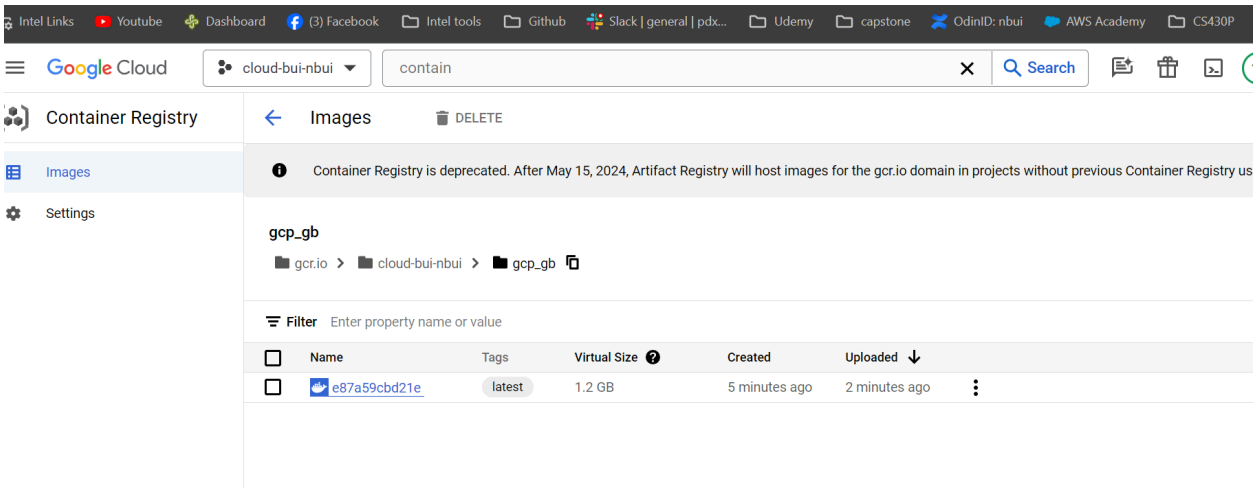
Nathan Bui <nbui@pdx.edu>
signed on 2024-02-15 22:08:24.837398
Hello ECS!

Lab 06.3g

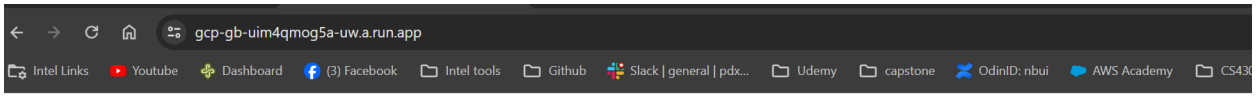
2. Prepare a container image



3. View container image



5. View the Guestbook



Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 07:13:05.467421+00:00
Hello Datastore

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-15 22:40:42.168459+00:00
Hello Cloud Run!

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-13 07:38:49.332539+00:00
Hello App Engine!

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 07:20:13.845665+00:00
Hello Docker Datastore!

Nathan Bui <nbui@pdx.edu>
signed on 2024-02-11 07:32:05.814840+00:00
Hello Compute Engine!

- 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

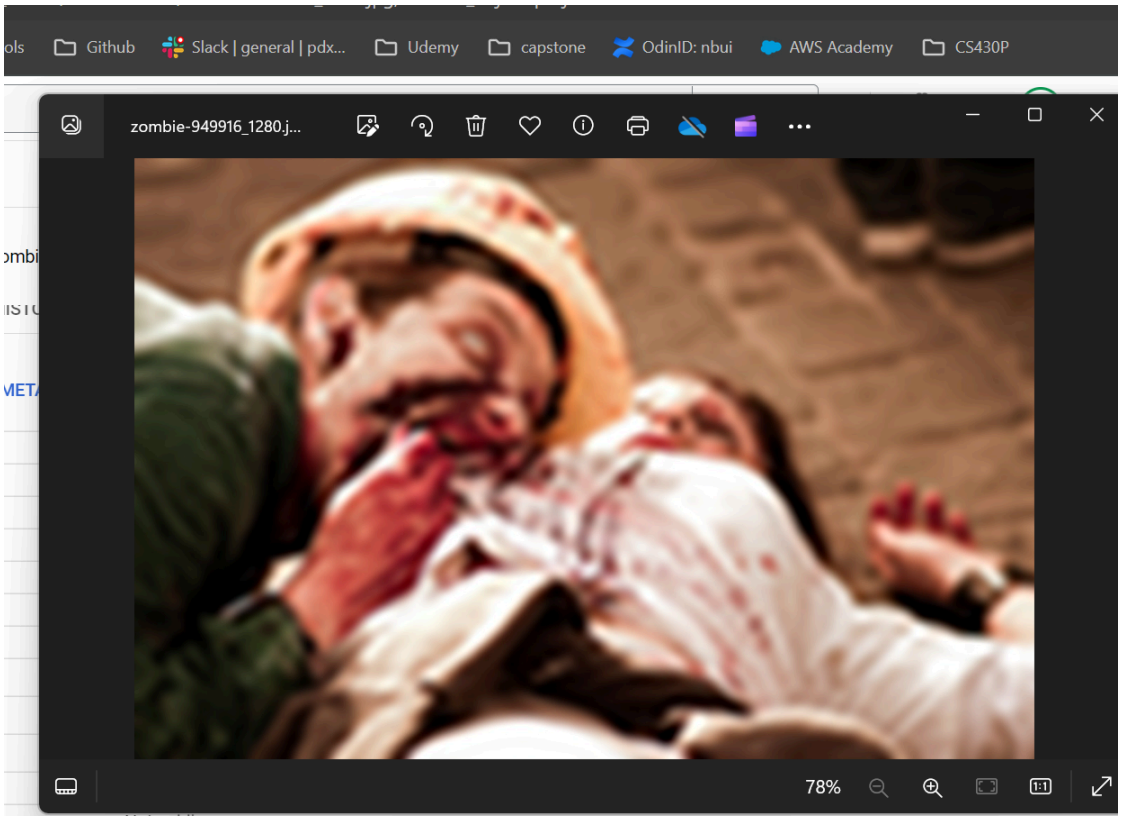
Lab 06.4g

4. -

- After downloading the file from the bucket, where is it stored?
    - The output bucket we created (its name would be saved in BLURRED_BUCKET_NAME env variable)
- What class in the ImageMagick package is used to do the blurring of the file?
    - wand.image.Image
- What lines of code perform the blurring of the image and its storage back into the filesystem?
    - Line 71-73 (resize and save the blurred image)

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.

```
Operation completed over 1 objects/353.0 KiB.
nbui@cloudshell:~/python-docs-samples/functions/imagemagick (cloud-bui-nbui)$ gcloud functions logs read
LEVEL: D
NAME: blur_offensive_images
EXECUTION_ID: x4d1tbtxkwud
TIME_UTC: 2024-02-15 23:10:17.122
LOG: Function execution took 11641 ms, finished with status: 'ok'

LEVEL: I
NAME: blur_offensive_images
EXECUTION_ID: x4d1tbtxkwud
TIME_UTC: 2024-02-15 23:10:17.119
LOG:

LEVEL: I
NAME: blur_offensive_images
EXECUTION_ID: x4d1tbtxkwud
TIME_UTC: 2024-02-15 23:10:17.119
LOG: Blurred image uploaded to: gs://cs430-nbui-2/zombie-949916_1280.jpg

LEVEL: I
NAME: blur_offensive_images
EXECUTION_ID: x4d1tbtxkwud
TIME_UTC: 2024-02-15 23:10:16.982
```

11. PubSub via CLI

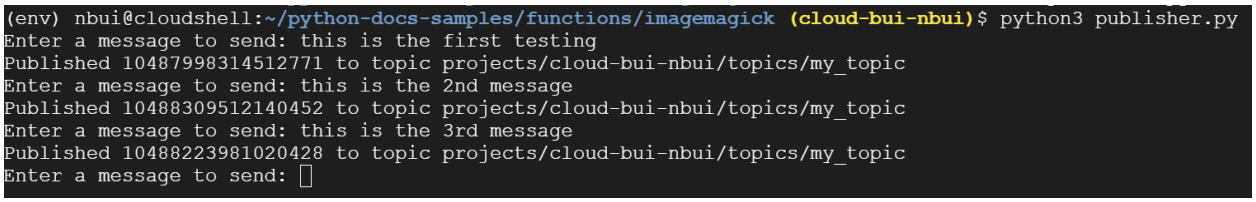
- Why are there no items returned?
  - Cause the message was published before the topic had any subscribers. Hence the message could be considered as delivered and was deleted.

12. -

```
nbui@pubsub:~$ gcloud pubsub subscriptions pull sub-${USER}
|    DATA    |      MESSAGE_ID      | ORDERING_KEY | ATTRIBUTES | DELIVERY_ATTEMPT |
                                                                   ACK_ID
                                                                           |
| Message #2 | 10488159775768003 |              |            |                  | RFAGFixdRkhRNxkIaFEOT14jPzUgKE
URBgBPAihdeTFNIEFdcGhRDRlyfWB9PgtFAwAUW3dYURsHaE5tdR-Y_ZPsS0NUa1oXCQpEVndYXB4KblRcdQR5o-OSpMKGlHgJOjqMo_TBbTuXj5
cMZiM9XhJLLD5-NTJFQV5AEkw-B0RJUytDCypYEU4EISE-MD5FUw |
nbui@pubsub:~$
```

15. Test programs and clean up

- Take a screenshot showing the messageIds and messages sent



- Take a screenshot showing the same messageIds and messages received