Week#4 Labs

Neha Agrawal

Table of Contents

04.1a: nginx EC2 Guestbook

1. EC2 VM

No screenshots required

2. Register a DNS name for VM

3. Checkout code

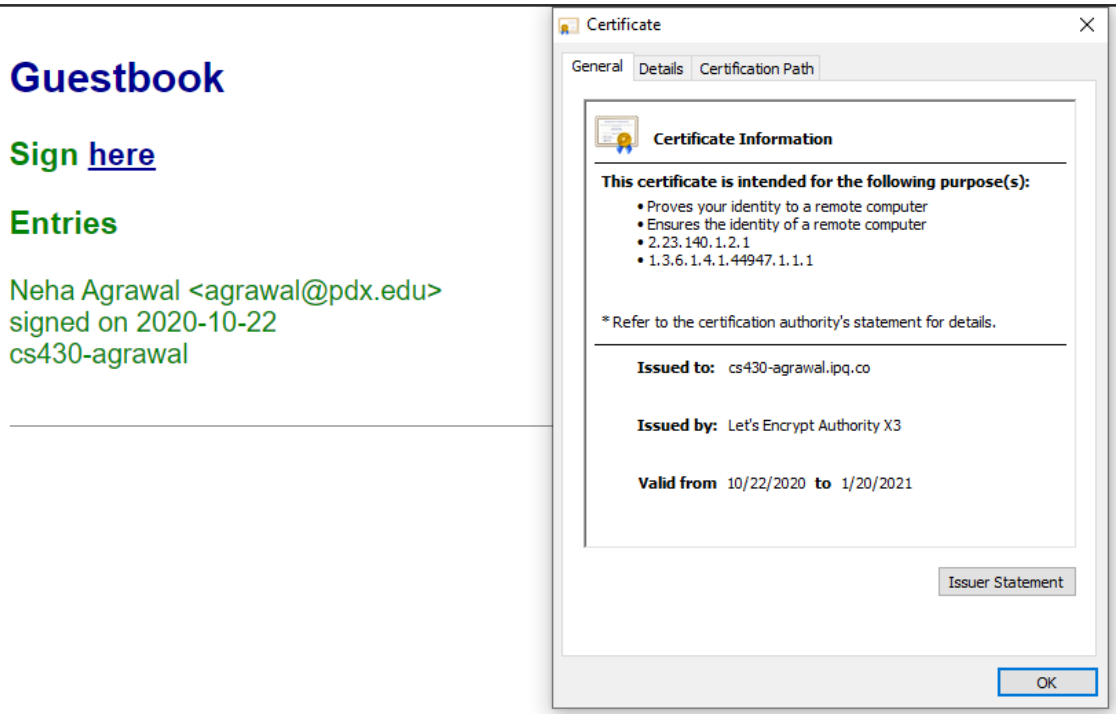
No screenshots required

4. Examine code

No screenshots required

5. Install the application

Bring the site up in a browser and note it's valid certificate. Add an entry to the guestbook, then show a screenshot of the site along with its Let's Encrypt certificate.



6. Clean up

No screenshots required

04.1g: nginx Compute Engine Guestbook

1. Compute Engine VM

No screenshots required

2. Register a DNS name for VM

Create a dns record for an IP address instantly

The record *cs430-agrawal-psu.ipq.co*

resolves to IP address *34.83.34.162*

3. Checkout code

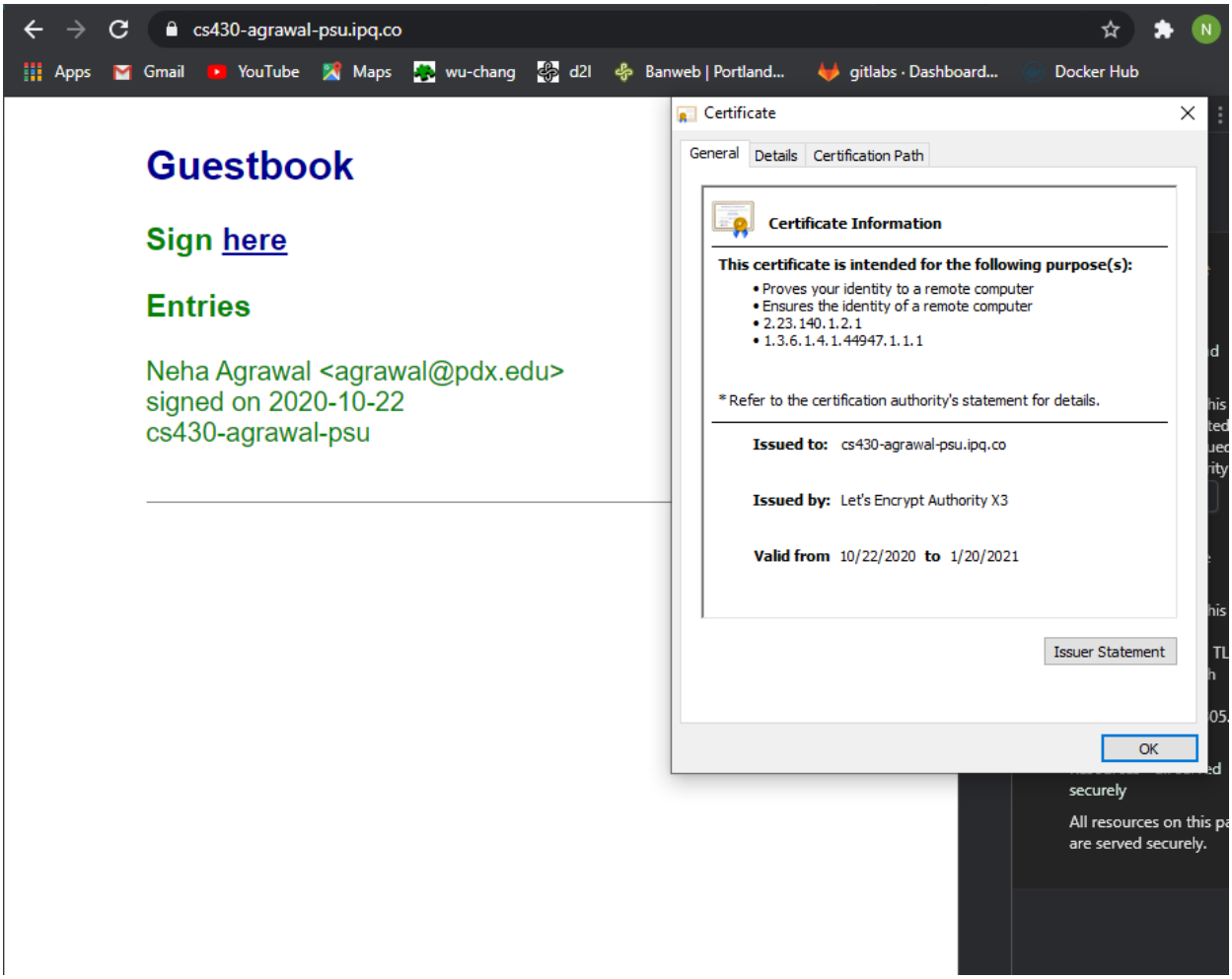
No screenshots required

4. Examine code

No screenshots required

5. Install the application

Bring the site up in a browser and note its valid certificate. Add an entry to the guestbook, then show a screenshot of the site along with its Let's Encrypt certificate



6. Clean up

No screenshots required

04.2g: Docker Guestbook

1. Containers

No screenshots required

2. Version 1: Ubuntu

No screenshots required

3. Build and run the Ubuntu-based container

Show the image generated and its size in a screenshot for your lab notebook using the command:

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
helloubuntu         latest              a0f304f2e5bd        28 seconds ago      446MB
ubuntu              18.04               56def654ec22        3 weeks ago         63.2MB
```

4. Docker commands

Within the container, show the contents of the current directory via `ls`, the contents of the file specifying the Linux standard base being used (`/etc/lsb-release`), and the output of the process listing command (`ps -ef`). Exit out of the shell and container.

1. ls
2. cat /etc/lsb-release
3. ps -ef

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker exec -it hellou /bin/bash
root@552f04263cb2:/app# ls
Dockerfile.alpine  app.py   index.py   requirements.txt  sign.pyc  templates
Dockerfile.ubuntu  gbmodel  index.pyc  sign.py           static
root@552f04263cb2:/app# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.5 LTS"
root@552f04263cb2:/app# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 23:04 ?        00:00:00 python app.py
root           7       1  0 23:04 ?        00:00:03 /usr/bin/python /app/app.py
root          23       0  0 23:11 pts/0    00:00:00 /bin/bash
root          34      23  0 23:11 pts/0    00:00:00 ps -ef
```

5. Docker Hub Ubuntu

No screenshots required

6. Running from Docker Hub

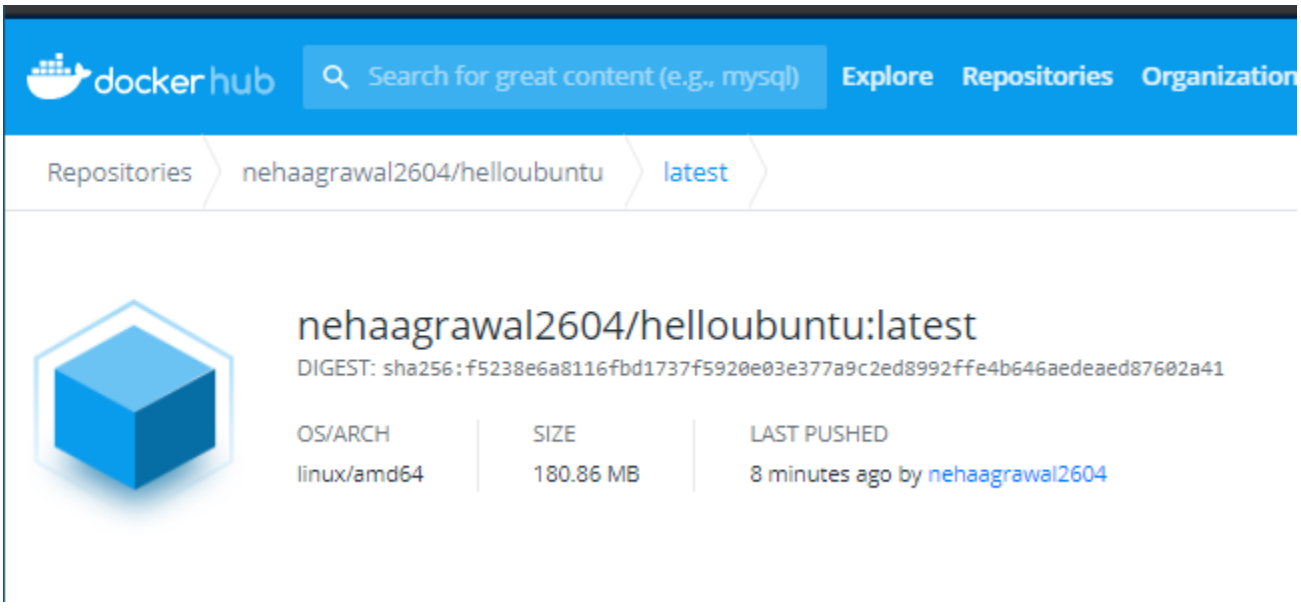
- **Run the image directly from Docker Hub and show a screenshot of the output of the command in your lab notebook.**

```
docker run -di -p 8000:5000 --name hellou <dockerhub_id>/helloubuntu
```

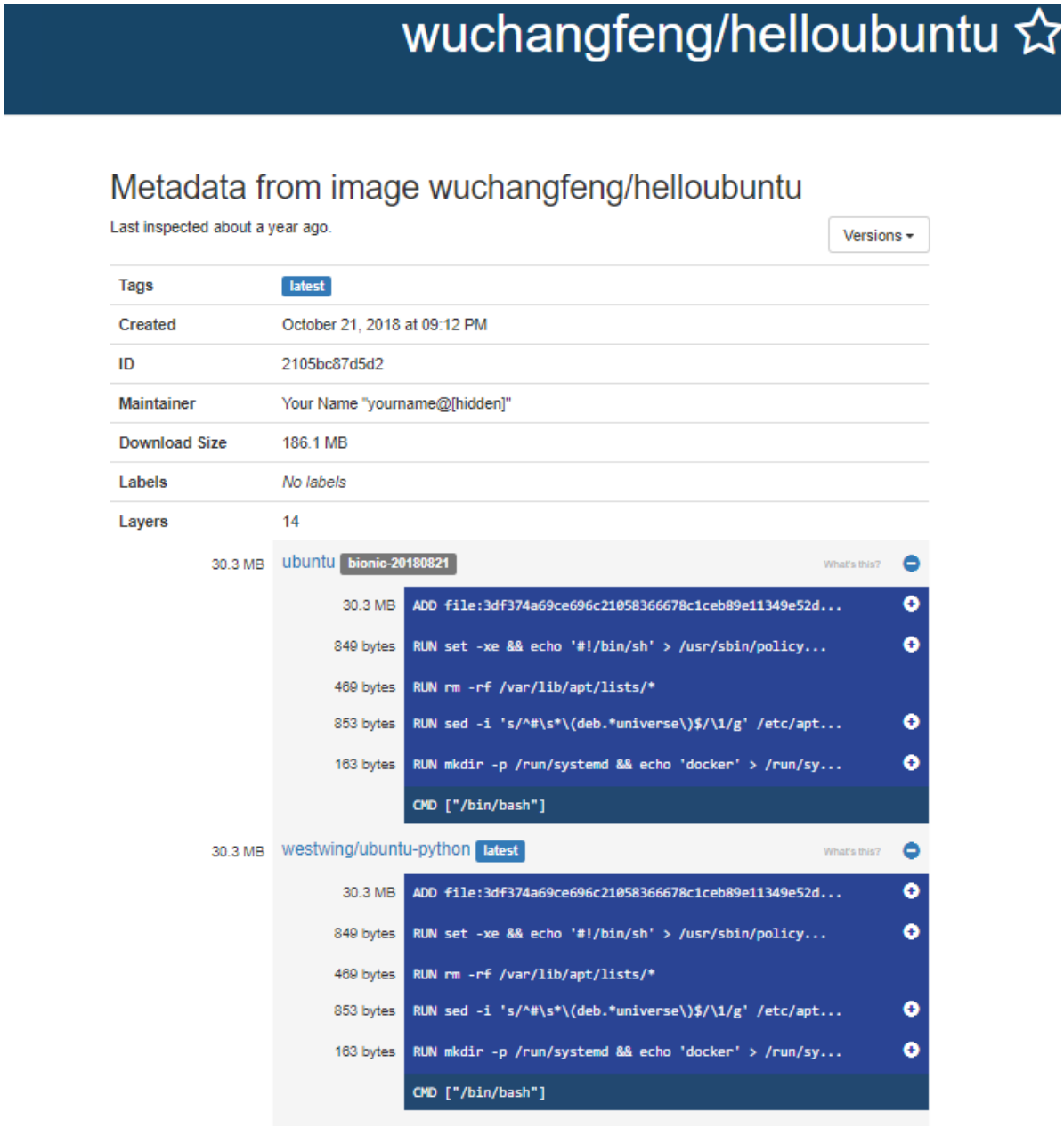
Output:

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker run -di -p 8000:5000 --name hellou
 nehaagrawal2604/helloubuntu
Unable to find image 'nehaagrawal2604/helloubuntu:latest' locally
latest: Pulling from nehaagrawal2604/helloubuntu
171857c49d0f: Already exists
419640447d26: Already exists
61e52f862619: Already exists
05f3f4883fef: Pull complete
8d960613ebf8: Pull complete
676f0d0a82cb: Pull complete
33260fa4dcdd: Pull complete
Digest: sha256:f5238e6a8116fbd1737f5920e03e377a9c2ed8992ffe4b646aedeaed87602a41
Status: Downloaded newer image for nehaagrawal2604/helloubuntu:latest
25855b8511decccdd30e9d3697a56dee070a244c03d23a07087e7fd9a2b5cd85
```

- **Then, log into Docker Hub with a web browser, navigate to the container image, and take a screenshot of the container image and its size.**



- **Finally, visit https://microbadger.com/ and show the container image metadata using MicroBadger that describes the individual layers of the container. Note that if this site takes too long to return a result, you may visit the same container image name under the wuchangfeng account.**



7. Version 2: Alpine

No screenshots required

8. Build and run the Alpine-based container

- **Show the image generated and its size in a screenshot for your lab notebook. How much smaller is the image?**

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
helloalpine         latest              2f9586a036ca        33 seconds ago      54.4MB
python              alpine              dc68588b1801        19 hours ago        44.3MB
ubuntu              18.04               56def654ec22        3 weeks ago         63.2MB
```

The size of alpine-based docker image has reduced to 54.4 MB as compared to our previous ubuntu based docker image whose size was 446 MB

- **Show the output of this command in a screenshot for your lab notebook. What might have happened?**

```
docker exec -it helloa /bin/bash
```

Output:

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker exec -it helloa /bin/bash
OCI runtime exec failed: exec failed: container_linux.go:349: starting container process caused "
exec: \"/bin/bash\": stat /bin/bash: no such file or directory": unknown
```

The reason for above output is: Our alpine-based docker image does not have the binary /bin/bash installed to reduce the image size. This facilitates quick startup and efficient storage.

- **Then, replace /bin/bash with /bin/sh and repeat the command. Within the container, show the contents of the file specifying the Alpine release being used (/etc/alpine-release) and the output of the process listing command (ps -ef)**

1.cat /etc/alpine-release
2. ps -ef

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker exec -it helloa /bin/sh
/app # cat /etc/alpine-release
3.12.1
/app # ps -ef
PID   USER     TIME  COMMAND
    1 root      0:00 python app.py
    6 root      0:05 /usr/local/bin/python /app/app.py
   21 root      0:00 /bin/sh
   27 root      0:00 ps -ef
```

9. Docker Hub Alpine

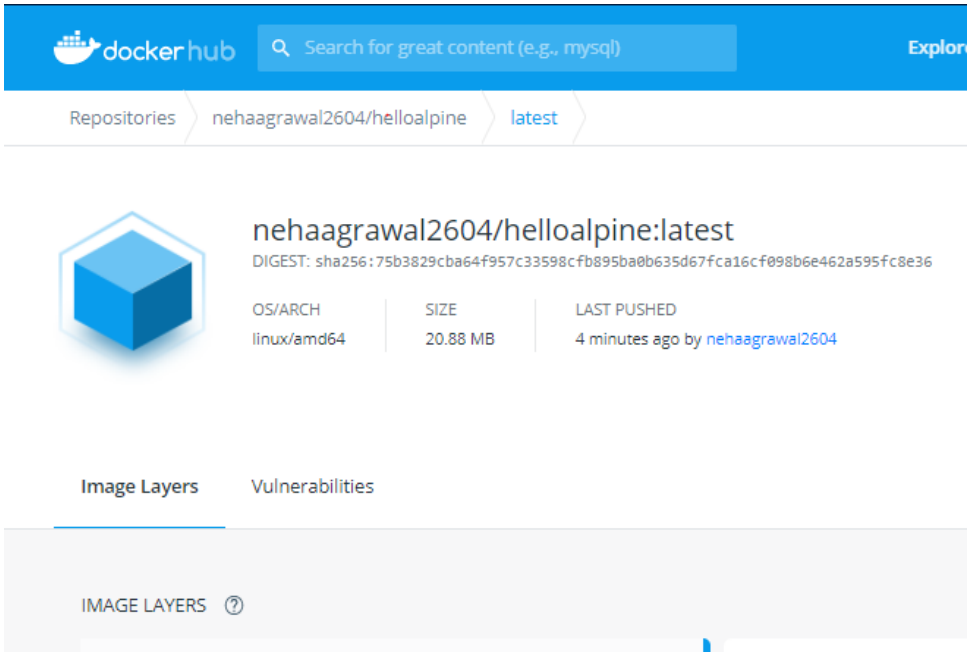
- **Run the image directly from Docker Hub and show a screenshot of the output of the command in your lab notebook.**

```
docker run -di -p 8000:5000 --name helloa <dockerhub_id>/helloalpine
```

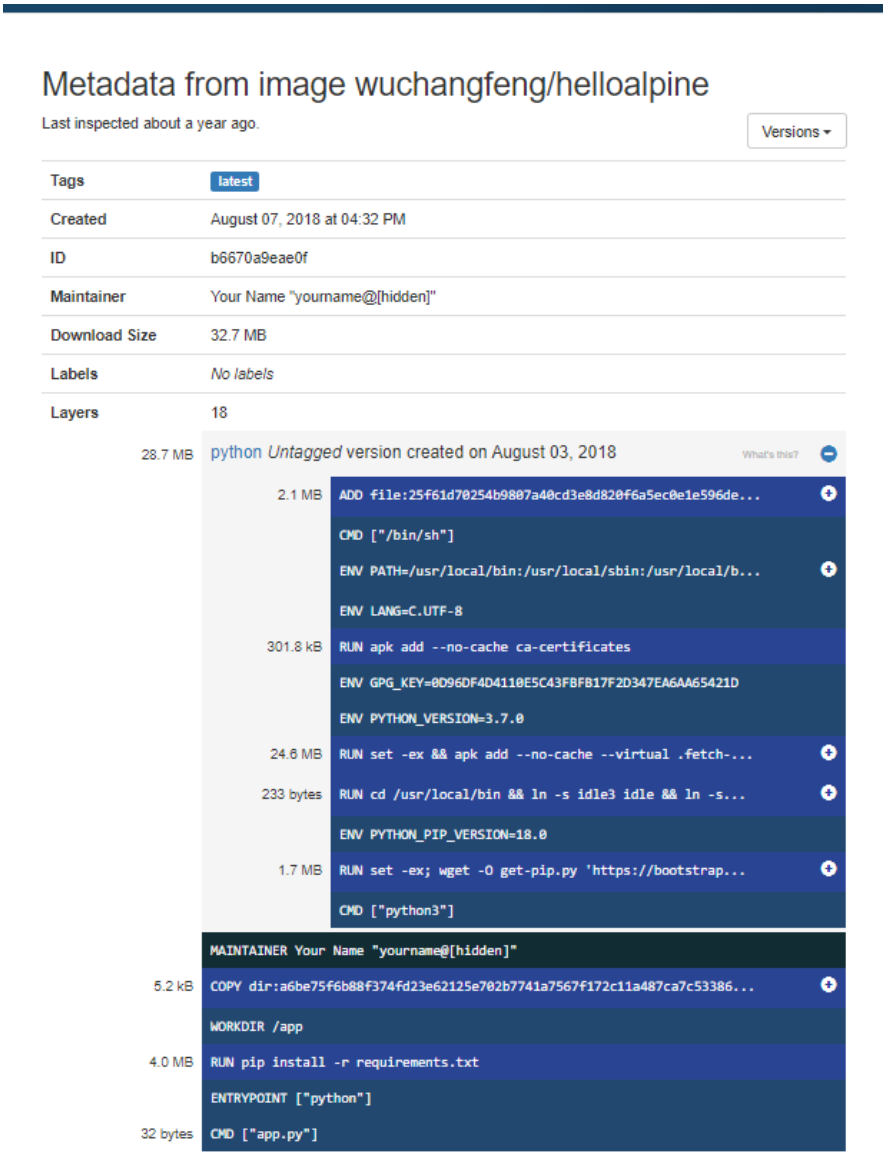
Output:

```
agrawal@agrawal-VirtualBox:~/cs430-src/04_container_dockerhub$ docker run -di -p 8000:5000 --name
 helloa nehaagrawal2604/helloalpine
Unable to find image 'nehaagrawal2604/helloalpine:latest' locally
latest: Pulling from nehaagrawal2604/helloalpine
188c0c94c7c5: Already exists
55578f60cda7: Already exists
692da2fcb614: Already exists
599e2857d4f0: Already exists
4b3bf1abad55: Already exists
c3298040f4c3: Pull complete
5ce60546cddf: Pull complete
Digest: sha256:75b3829cba64f957c33598cfb895ba0b635d67fca16cf098b6e462a595fc8e36
Status: Downloaded newer image for nehaagrawal2604/helloalpine:latest
542bbae783f56c8106f64bec512ff2da6cc52b92f88135cddf61160619e75347
```

- **Then, log into Docker Hub with a web browser, navigate to the container image, and take a screenshot of the container image and its size.**

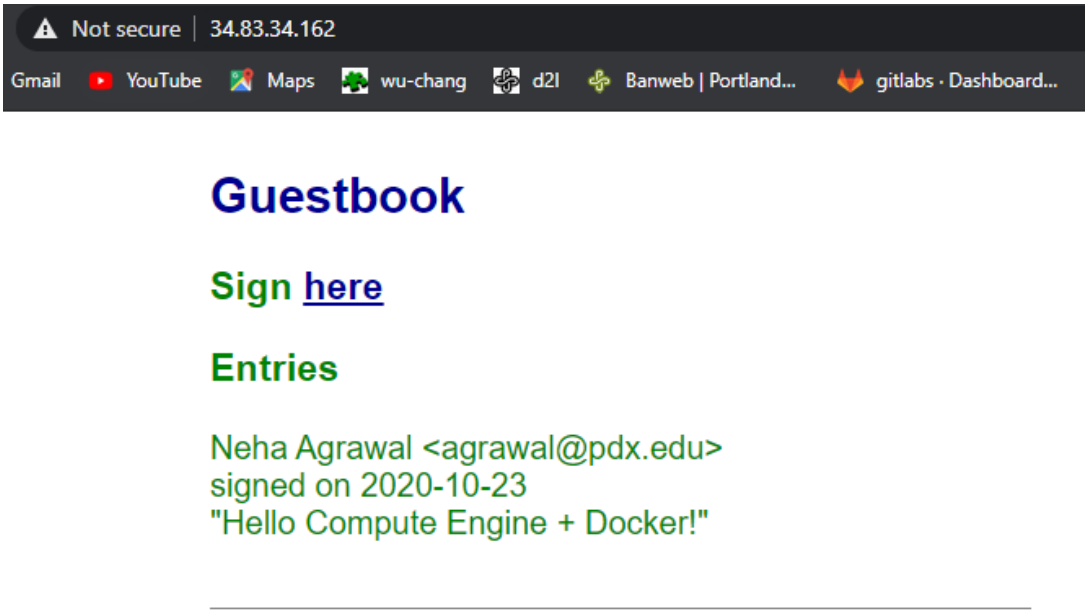


- **Finally, visit https://microbadger.com/ and show the container image metadata using MicroBadger that describes the individual layers of the container. Note that if this site takes too long to return a result, you may visit the same container image name under the wuchangfeng account.**



10. Compute Engine Ubuntu VM deployment

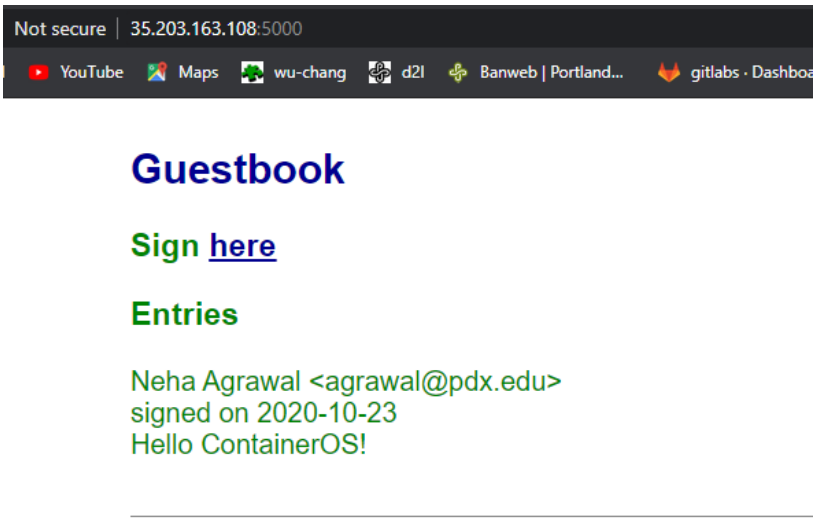
Show in a screenshot that the site is running via the VM's external IP address with a guestbook entry with the message "Hello Compute Engine + Docker!"



11. Compute Engine ContainerOS VM deployment (1)

No screenshots required

12. Compute Engine ContainerOS VM deployment (2)



13. Clean up

No screenshots required