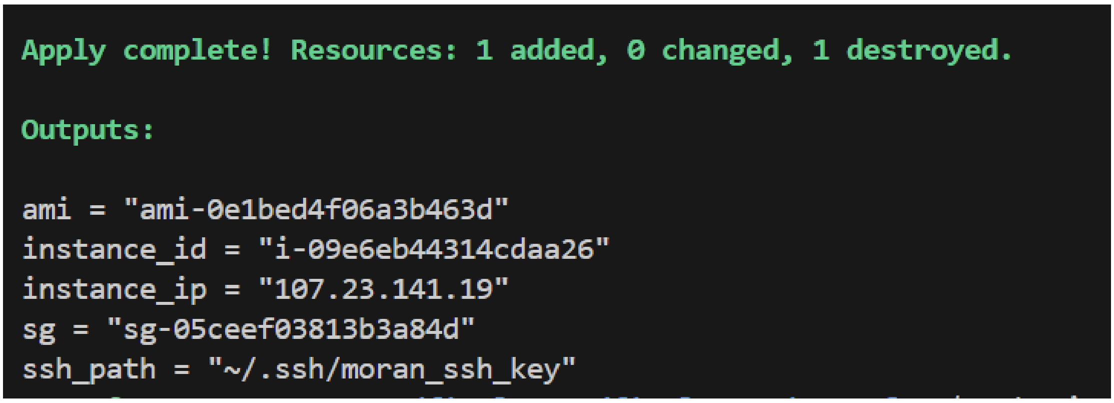
Moran Danino- Final Exam:

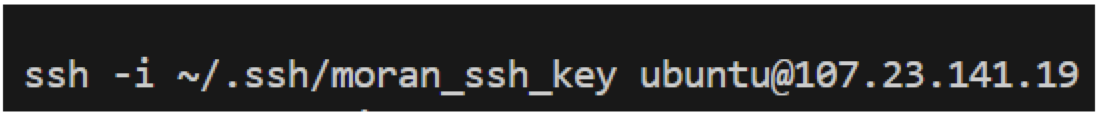
Terraform:

outputs:



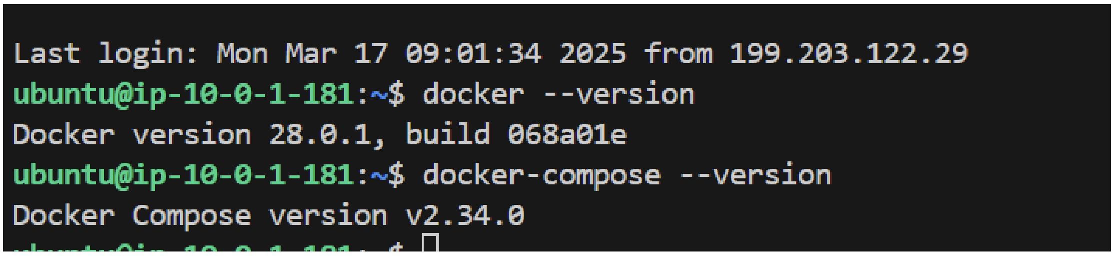
```
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

Outputs:

ami = "ami-0e1bed4f06a3b463d"
instance_id = "i-09e6eb44314cdaa26"
instance_ip = "107.23.141.19"
sg = "sg-05ceef03813b3a84d"
ssh_path = "~/.ssh/moran_ssh_key"
```

Get into the machine with ssh:



```
ssh -i ~/.ssh/moran_ssh_key ubuntu@107.23.141.19
```

Inside the machine: docker and docker-compose installs:



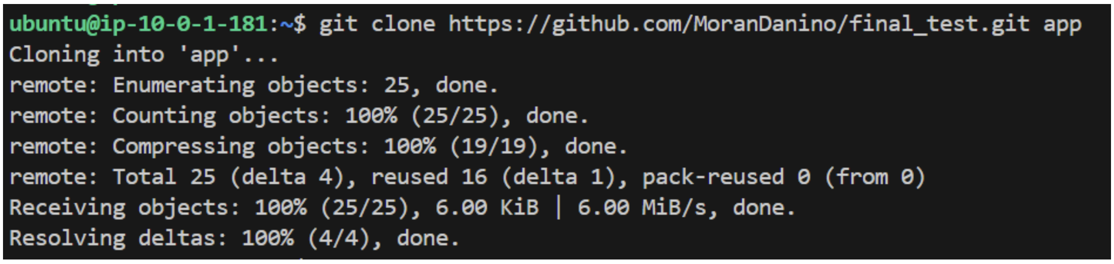
```
Last login: Mon Mar 17 09:01:34 2025 from 199.203.122.29
ubuntu@ip-10-0-1-181:~$ docker --version
Docker version 28.0.1, build 068a01e
ubuntu@ip-10-0-1-181:~$ docker-compose --version
Docker Compose version v2.34.0
```

notice: I got errors about vpc not exist, so for the test I used a new one.

Docker:

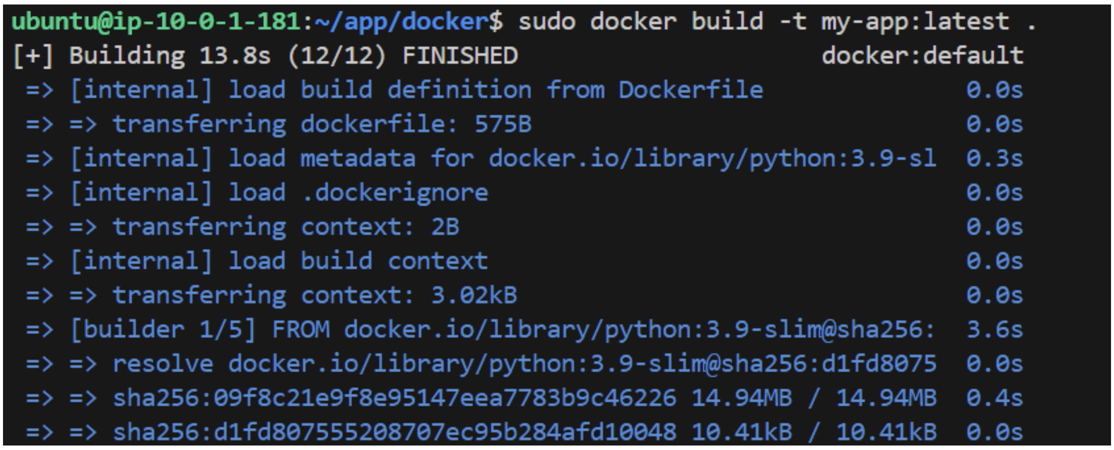
Inside the machine:

Clone to my repository and change branch to docker branch



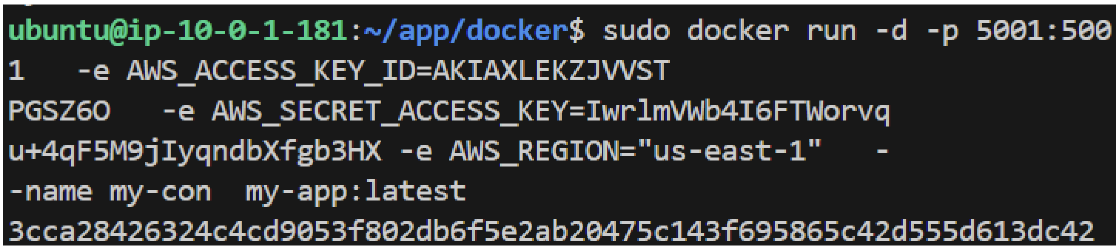
```
ubuntu@ip-10-0-1-181:~$ git clone https://github.com/MoranDanino/final_test.git app
Cloning into 'app'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 25 (delta 4), reused 16 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (25/25), 6.00 KiB | 6.00 MiB/s, done.
Resolving deltas: 100% (4/4), done.
```

build the image:



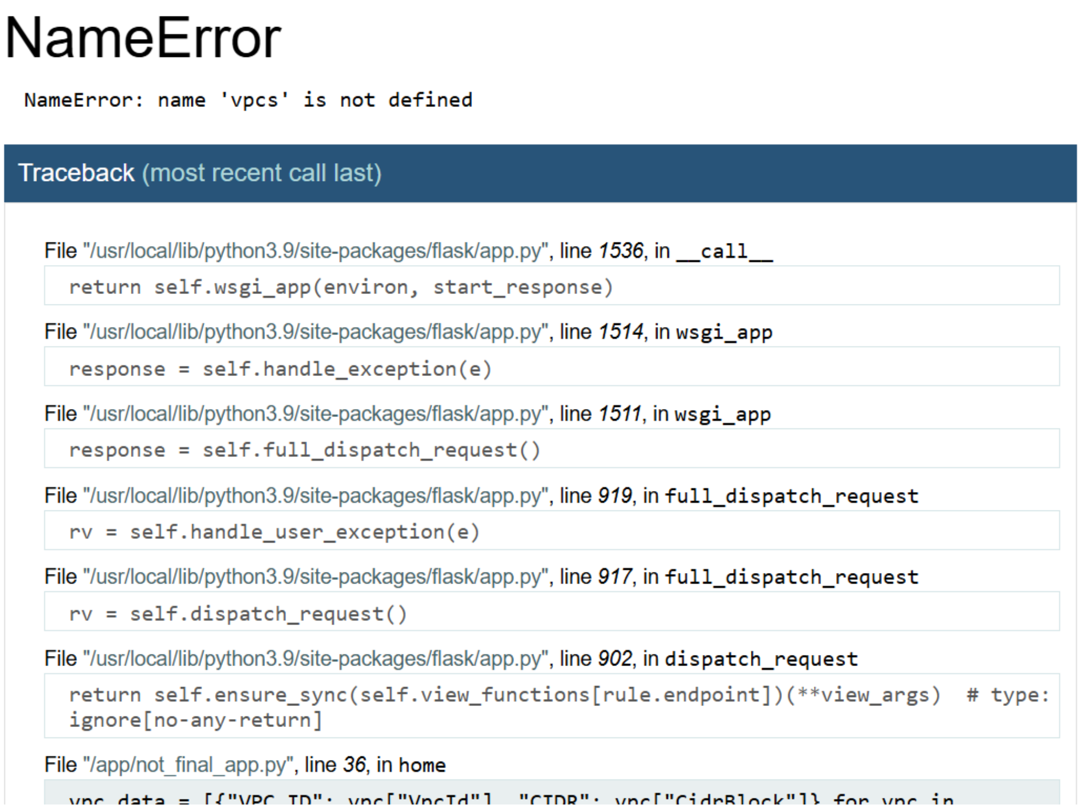
```
ubuntu@ip-10-0-1-181:~/app/docker$ sudo docker build -t my-app:latest .
[+] Building 13.8s (12/12) FINISHED                          docker:default
 => [internal] load build definition from Dockerfile                   0.0s
 => => transferring dockerfile: 575B                                   0.0s
 => [internal] load metadata for docker.io/library/python:3.9-sl  0.3s
 => [internal] load .dockerignore                                      0.0s
 => => transferring context: 2B                                        0.0s
 => [internal] load build context                                      0.0s
 => => transferring context: 3.02kB                                    0.0s
 => [builder 1/5] FROM docker.io/library/python:3.9-slim@sha256:  3.6s
 => => resolve docker.io/library/python:3.9-slim@sha256:d1fd8075  0.0s
 => => sha256:09f8c21e9f8e95147eea7783b9c46226 14.94MB / 14.94MB  0.4s
 => => sha256:d1fd807555208707ec95b284afd10048 10.41kB / 10.41kB  0.0s
```

run the container:



```
ubuntu@ip-10-0-1-181:~/app/docker$ sudo docker run -d -p 5001:500
1   -e AWS_ACCESS_KEY_ID=AKIAXLEKZJVVST
PGSZ6O   -e AWS_SECRET_ACCESS_KEY=IwrlmVWb4I6FTWorvq
u+4qF5M9jIyqndbXfgb3HX -e AWS_REGION="us-east-1"   -
-name my-con  my-app:latest
3cca28426324c4cd9053f802db6f5e2ab20475c143f695865c42d555d613dc42
```

open the browser in http://107.23.141.19:5001
and got what needed:



NameError

```
NameError: name 'vpcs' is not defined
```

Traceback (most recent call last)

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *1536*, in `__call__`

```
return self.wsgi_app(environ, start_response)
```

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *1514*, in `wsgi_app`

```
response = self.handle_exception(e)
```

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *1511*, in `wsgi_app`

```
response = self.full_dispatch_request()
```

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *919*, in `full_dispatch_request`

```
rv = self.handle_user_exception(e)
```

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *917*, in `full_dispatch_request`

```
rv = self.dispatch_request()
```

File "/usr/local/lib/python3.9/site-packages/flask/app.py", line *902*, in `dispatch_request`

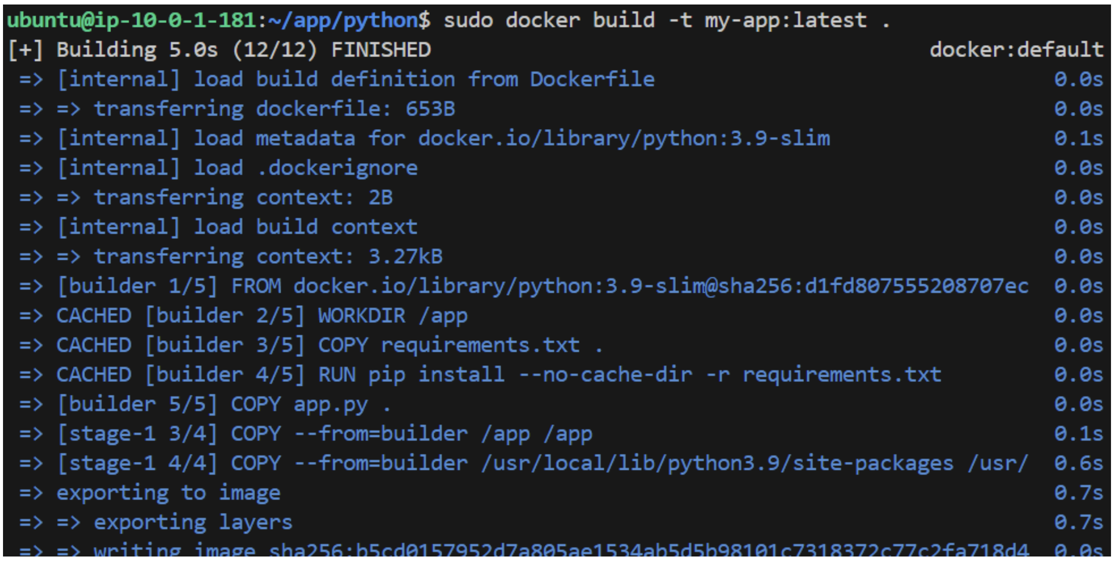
```
return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)  # type: ignore[no-any-return]
```

File "/app/not_final_app.py", line *36*, in `home`

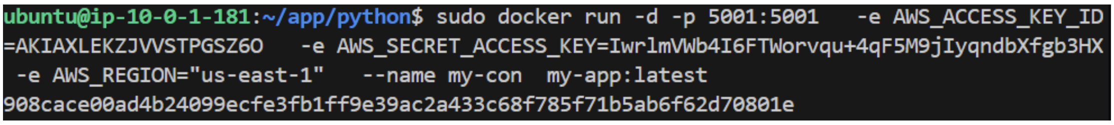
```
vpc_data = [{"VPC_ID": vpc["VpcId"], "CIDR": vpc["CidrBlock"]} for vpc in
```

python:

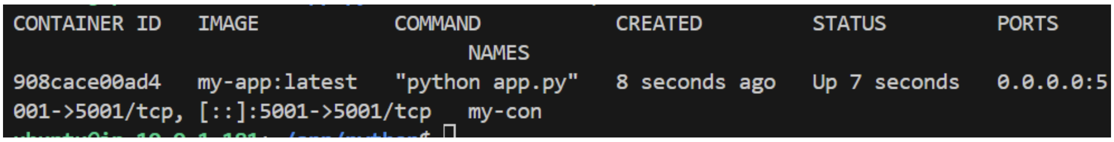
fixing the script and then run the dockerfile again with the updated app:



```
ubuntu@ip-10-0-1-181:~/app/python$ sudo docker build -t my-app:latest .
[+] Building 5.0s (12/12) FINISHED                                     docker:default
 => [internal] load build definition from Dockerfile                            0.0s
 => => transferring dockerfile: 653B                                            0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim              0.1s
 => [internal] load .dockerignore                                               0.0s
 => => transferring context: 2B                                                 0.0s
 => [internal] load build context                                               0.0s
 => => transferring context: 3.27kB                                             0.0s
 => [builder 1/5] FROM docker.io/library/python:3.9-slim@sha256:d1fd807555208707ec  0.0s
 => CACHED [builder 2/5] WORKDIR /app                                           0.0s
 => CACHED [builder 3/5] COPY requirements.txt .                                0.0s
 => CACHED [builder 4/5] RUN pip install --no-cache-dir -r requirements.txt     0.0s
 => [builder 5/5] COPY app.py .                                                 0.0s
 => [stage-1 3/4] COPY --from=builder /app /app                                 0.1s
 => [stage-1 4/4] COPY --from=builder /usr/local/lib/python3.9/site-packages /usr/  0.6s
 => exporting to image                                                          0.7s
 => => exporting layers                                                         0.7s
 => => writing image sha256:b5cd0157952d7a805ae1534ab5d5b98101c7318372c77c2fa718d4  0.0s
```

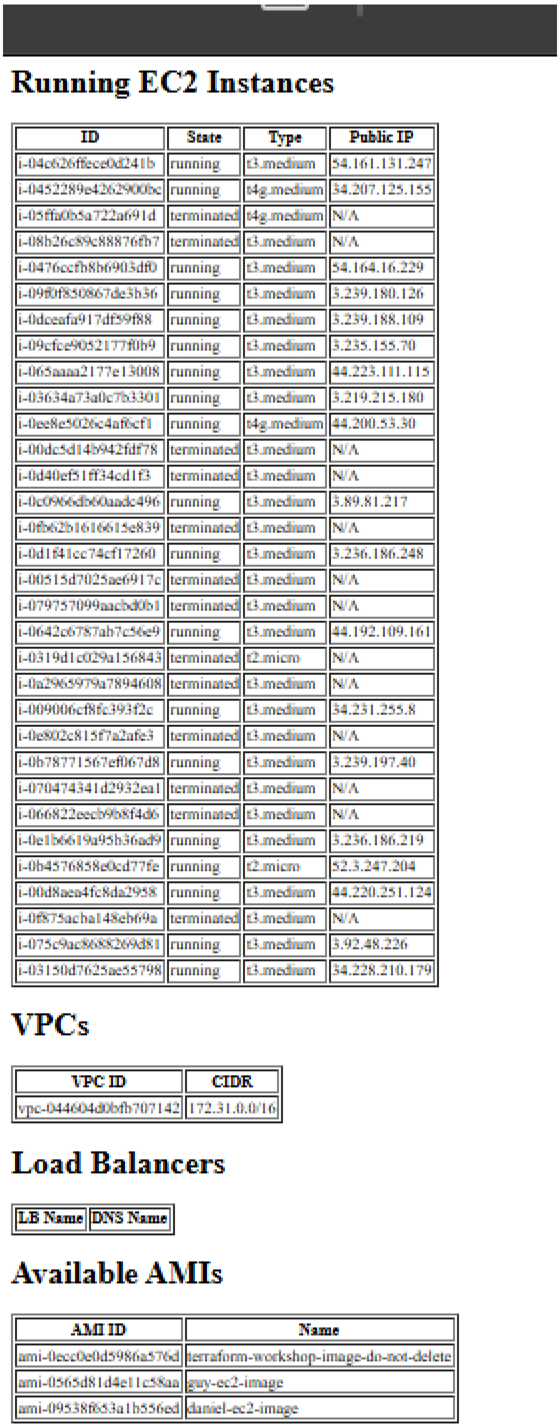


```
ubuntu@ip-10-0-1-181:~/app/python$ sudo docker run -d -p 5001:5001   -e AWS_ACCESS_KEY_ID
=AKIAXLEKZJVVSTPGSZ6O   -e AWS_SECRET_ACCESS_KEY=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
 -e AWS_REGION="us-east-1"   --name my-con  my-app:latest
908cace00ad4b24099ecfe3fb1ff9e39ac2a433c68f785f71b5ab6f62d70801e
```



```
CONTAINER ID   IMAGE           COMMAND           CREATED         STATUS         PORTS
                                   NAMES
908cace00ad4   my-app:latest   "python app.py"   8 seconds ago   Up 7 seconds   0.0.0.0:5
001->5001/tcp, [::]:5001->5001/tcp   my-con
ubuntu@ip-10-0-1-181:~/app/python$ ▯
```

output as needed:



Running EC2 Instances

ID	State	Type	Public IP
i-04c626ffece0d241b	running	t3.medium	54.161.131.247
i-0452289e4262900bc	running	t4g.medium	34.207.125.155
i-05ffa0b5a722a691d	terminated	t4g.medium	N/A
i-08b26c89c88876fb7	terminated	t3.medium	N/A
i-0476ccfb8b6903df0	running	t3.medium	54.164.16.229
i-09f0f850867de3b36	running	t3.medium	3.239.180.126
i-0dceafa917df59f88	running	t3.medium	3.239.188.109
i-09cfce9052177f0b9	running	t3.medium	3.235.155.70
i-065aaaa2177e13008	running	t3.medium	44.223.111.115
i-03634a73a0c7b3301	running	t3.medium	3.219.215.180
i-0ee8e5026c4af6cf1	running	t4g.medium	44.200.53.30
i-00dc5d14b942fdf78	terminated	t3.medium	N/A
i-0d40ef51ff34cd1f3	terminated	t3.medium	N/A
i-0c0966db60aadc496	running	t3.medium	3.89.81.217
i-0fb62b1616615e839	terminated	t3.medium	N/A
i-0d1f41cc74cf17260	running	t3.medium	3.236.186.248
i-00515d7025ae6917c	terminated	t3.medium	N/A
i-079757099aacbd0b1	terminated	t3.medium	N/A
i-0642c6787ab7c56e9	running	t3.medium	44.192.109.161
i-0319d1c029a156843	terminated	t2.micro	N/A
i-0a2965979a7894608	terminated	t3.medium	N/A
i-009006cf8fc393f2c	running	t3.medium	34.231.255.8
i-0e802c815f7a2afe3	terminated	t3.medium	N/A
i-0b78771567ef067d8	running	t3.medium	3.239.197.40
i-070474341d2932ea1	terminated	t3.medium	N/A
i-066822eecb9b8f4d6	terminated	t3.medium	N/A
i-0e1b6619a95b36ad9	running	t3.medium	3.236.186.219
i-0b4576858e0cd77fe	running	t2.micro	52.3.247.204
i-00d8aea4fc8da2958	running	t3.medium	44.220.251.124
i-0f875acba148eb69a	terminated	t3.medium	N/A
i-075c9ac8688269d81	running	t3.medium	3.92.48.226
i-03150d7625ae55798	running	t3.medium	34.228.210.179

VPCs

VPC ID	CIDR
vpc-044604d0bfb707142	172.31.0.0/16

Load Balancers

LB Name	DNS Name

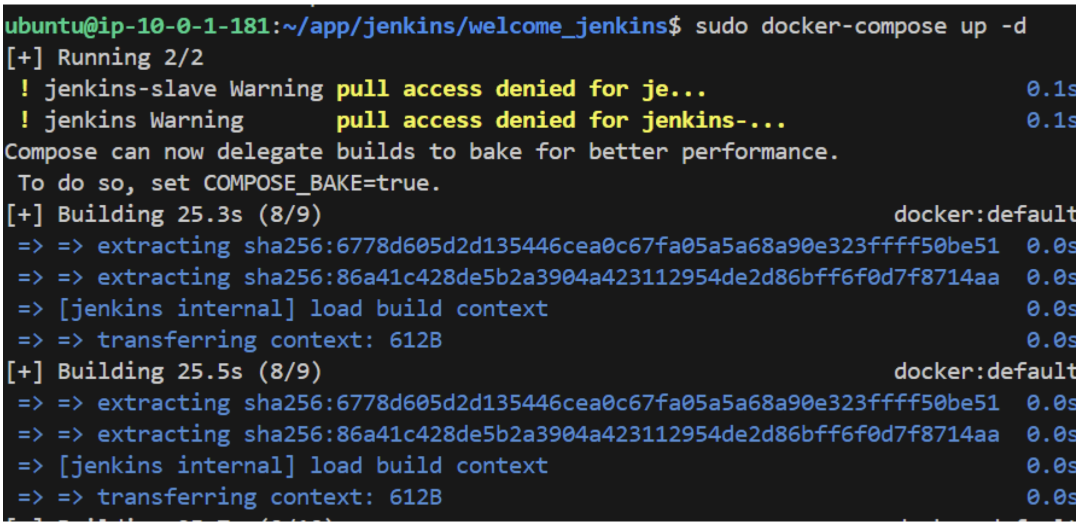
Available AMIs

AMI ID	Name
ami-0ecc0e0d5986a576d	terraform-workshop-image-do-not-delete
ami-0565d81d4e11c58aa	guy-ec2-image
ami-09538f653a1b556ed	daniel-ec2-image

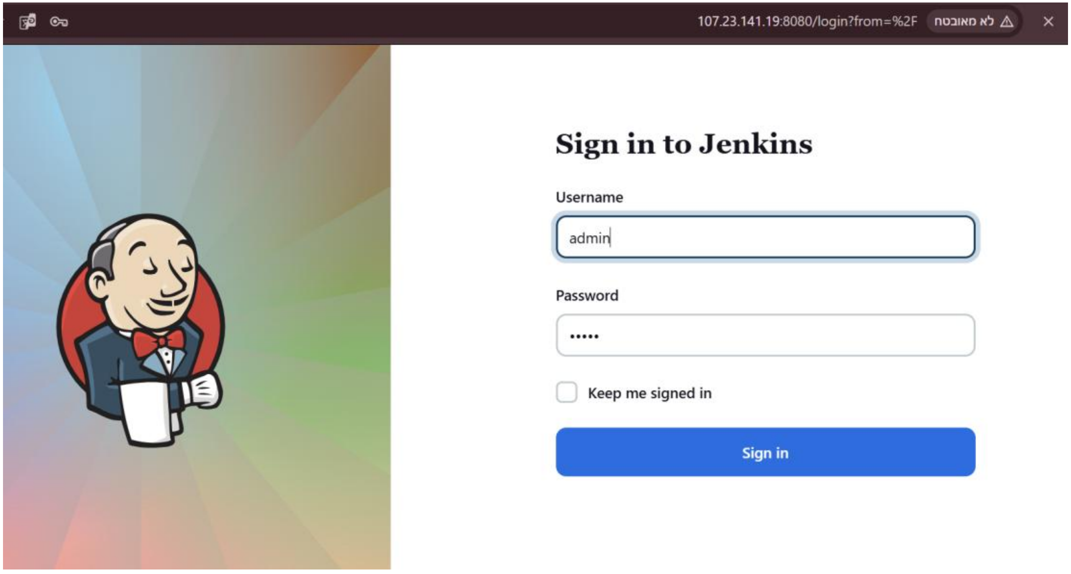
Jenkins:

I wrote the Jenkins file according to the mission.

Download and connect Jenkins from the machine:



```
ubuntu@ip-10-0-1-181:~/app/jenkins/welcome_jenkins$ sudo docker-compose up -d
[+] Running 2/2
 ! jenkins-slave Warning pull access denied for je...                    0.1s
 ! jenkins Warning       pull access denied for jenkins-....             0.1s
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 25.3s (8/9)                                         docker:default
 => => extracting sha256:6778d605d2d135446cea0c67fa05a5a68a90e323ffff50be51  0.0s
 => => extracting sha256:86a41c428de5b2a3904a423112954de2d86bff6f0d7f8714aa  0.0s
 => [jenkins internal] load build context                                    0.0s
 => => transferring context: 612B                                            0.0s
[+] Building 25.5s (8/9)                                         docker:default
 => => extracting sha256:6778d605d2d135446cea0c67fa05a5a68a90e323ffff50be51  0.0s
 => => extracting sha256:86a41c428de5b2a3904a423112954de2d86bff6f0d7f8714aa  0.0s
 => [jenkins internal] load build context                                    0.0s
 => => transferring context: 612B                                            0.0s
```



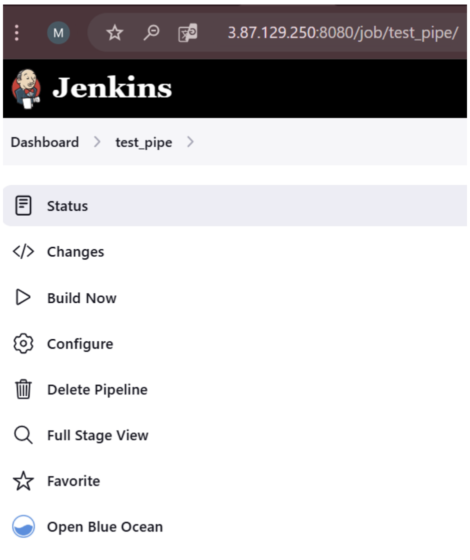
107.23.141.19:8080/login?from=%2F

Sign in to Jenkins

Username: admin

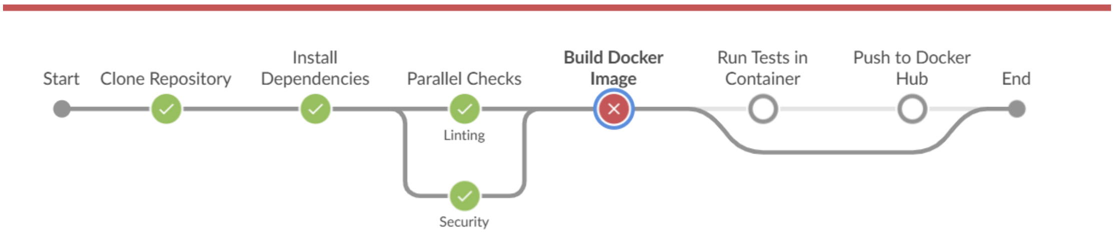
Password: •••••

☐ Keep me signed in



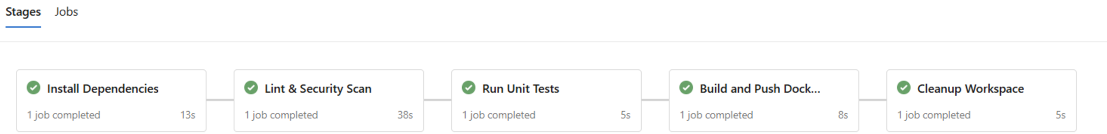
3.87.129.250:8080/job/test_pipe/

pipeline success:



tried to upload the image to docker, I did it many times and this is not working. Same in azure. I couldn't find the problem

azure:
azure pipeline:



✓ Install Dependencies
1 job completed 13s

✓ Lint & Security Scan
1 job completed 38s

✓ Run Unit Tests
1 job completed 5s

✓ Build and Push Dock...
1 job completed 8s

✓ Cleanup Workspace
1 job completed 5s

I did mock In the build because that was not working

k8s:

I couldn't run.
this is what I would do:
I created service and deployment.
in the cli:

Kubectl apply -f .

Kubectl get service

Check http://<load balancer dns>:port

helm:
couldn't make.
what I would do:

first install helm:
curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
helm version

helm create final-app #create the helm chart

-update Chart, values, deployment, service with the vars

helm template . > all.yaml

then:

```
helm upgrade --install my-app . -n my-app --create-namespace
```

do some change and then:

```
helm rollback
```