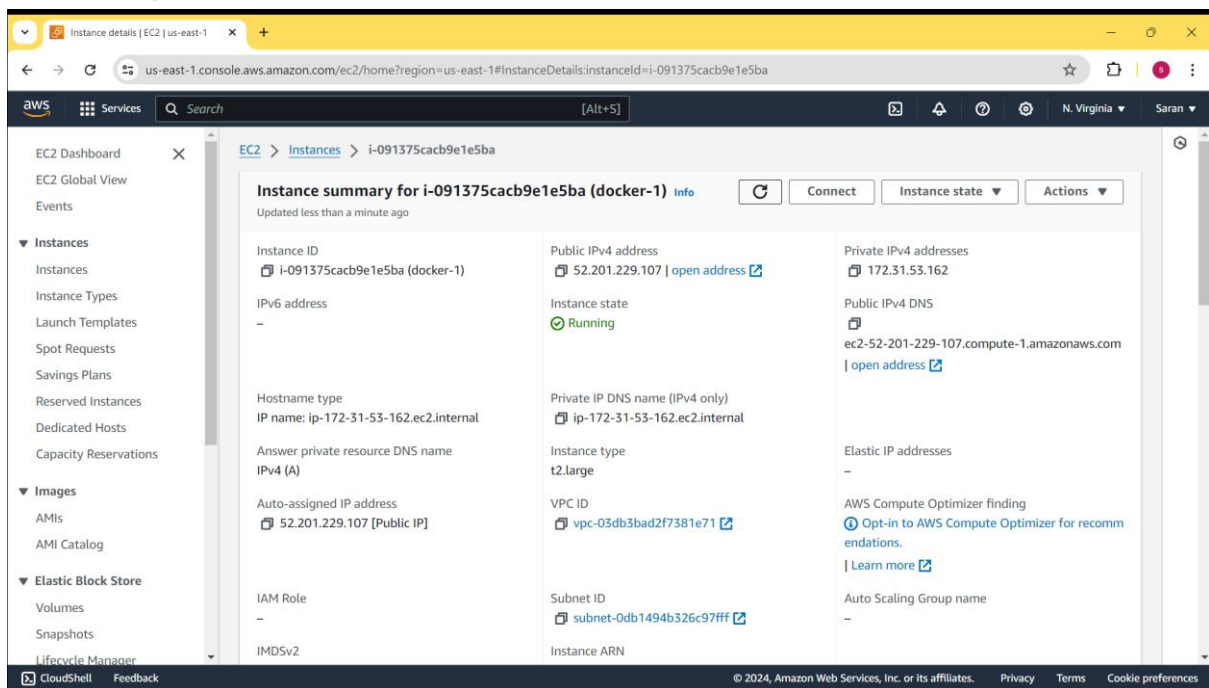
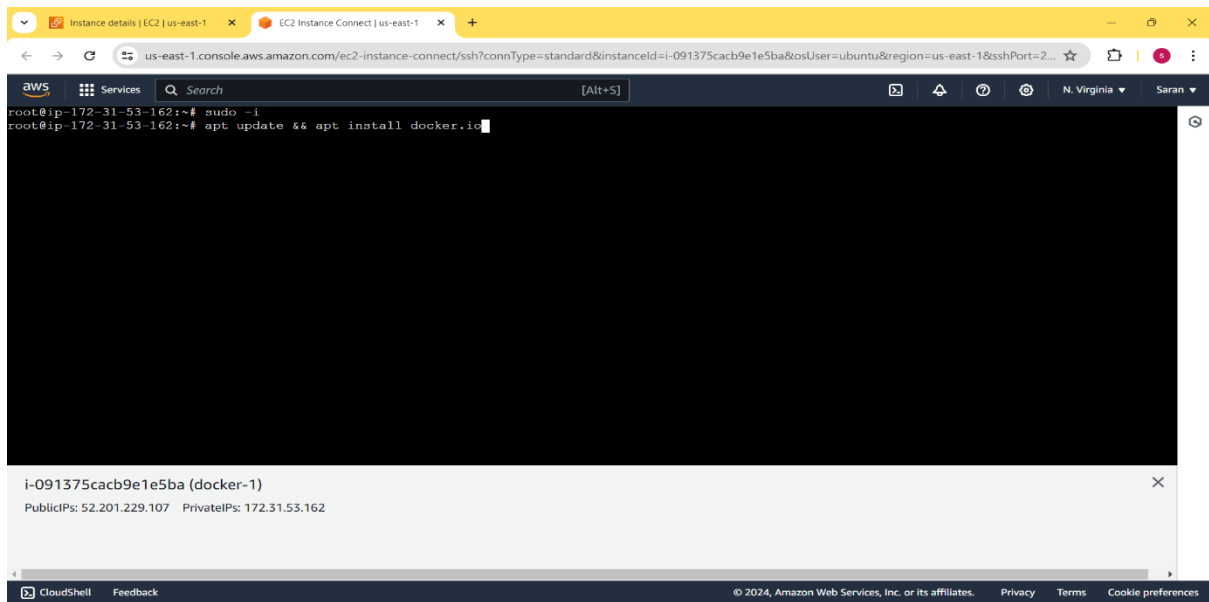


# EXPOSING A SIMPLE PYTHON SCRIPT IN DOCKER CONTAINERS THROUGH PORTS

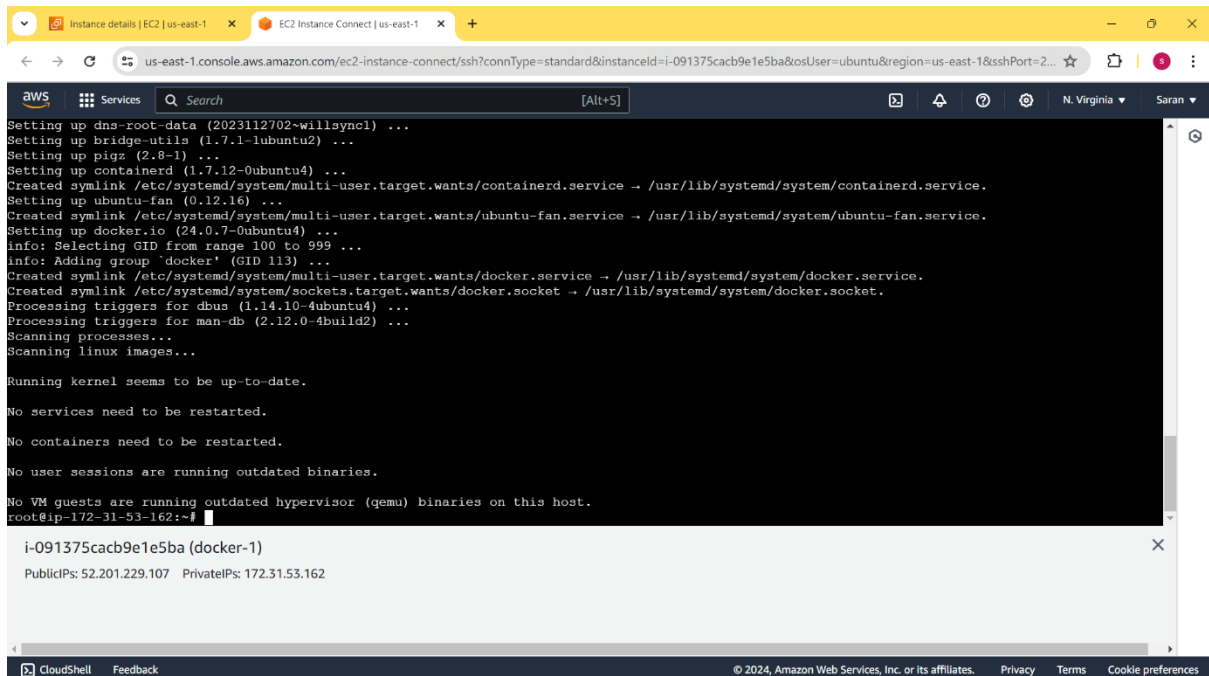
- First launch an EC2 instance of instance-type “t2.large”



- Next connect to the instance and run these commands `Sudo -i apt update && apt install docker.io`



- After installing docker write “Dockerfile” and python application script using “.py” extension.



Instance details | EC2 | us-east-1 x EC2 Instance Connect | us-east-1 x +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-091375cacb9e1e5ba&osUser=ubuntu&region=us-east-1&sshPort=2... ☆ | 5 | :

aws Services Search [Alt+S] N. Virginia Saran

```
root@ip-172-31-53-162:~# vi Dockerfile
```

i-091375cacb9e1e5ba (docker-1)  
PublicIPs: 52.201.229.107 PrivateIPs: 172.31.53.162

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Instance details | EC2 | us-east-1 x EC2 Instance Connect | us-east-1 x what is flask in docker - Google S... x +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-091375cacb9e1e5ba&osUser=ubuntu&region=us-east-1&sshPort=2... ☆ | 5 | :

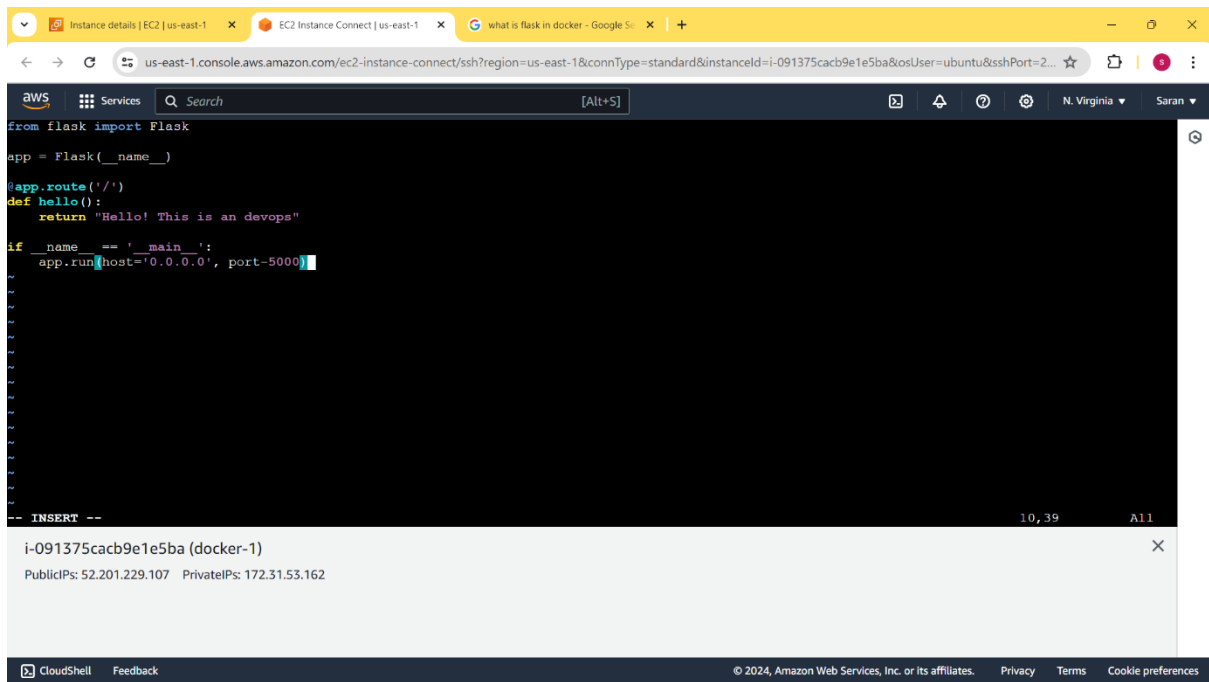
aws Services Search [Alt+S] N. Virginia Saran

```
FROM python:3.9-slim
COPY ..
RUN pip install flask
EXPOSE 5000
ENV FLASK_APP=app.py
CMD ["python", "app.py"]
```

"Dockerfile" 11L, 114B 11, 24 A1

i-091375cacb9e1e5ba (docker-1)  
PublicIPs: 52.201.229.107 PrivateIPs: 172.31.53.162

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



The screenshot shows the AWS CloudShell interface. The top navigation bar includes the AWS logo, a search bar, and the region 'N. Virginia'. The main area displays a code editor with a Python Flask application snippet. Below the code editor, a metadata box shows the instance ID 'i-091375cacb9e1e5ba (docker-1)' and its public and private IP addresses. The footer contains the 'CloudShell' logo, a 'Feedback' link, and copyright information for Amazon Web Services, Inc. (© 2024).

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello! This is an devops"

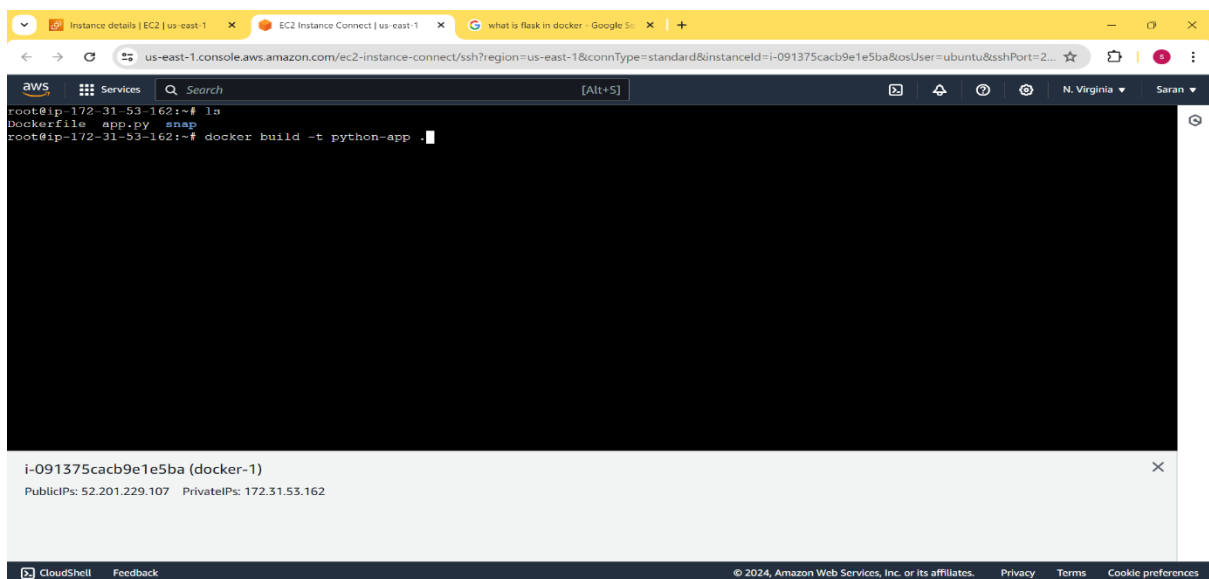
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

-- INSERT --

i-091375cacb9e1e5ba (docker-1)  
PublicIPs: 52.201.229.107 PrivateIPs: 172.31.53.162

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Now it is time to build the image from our Dockerfile.



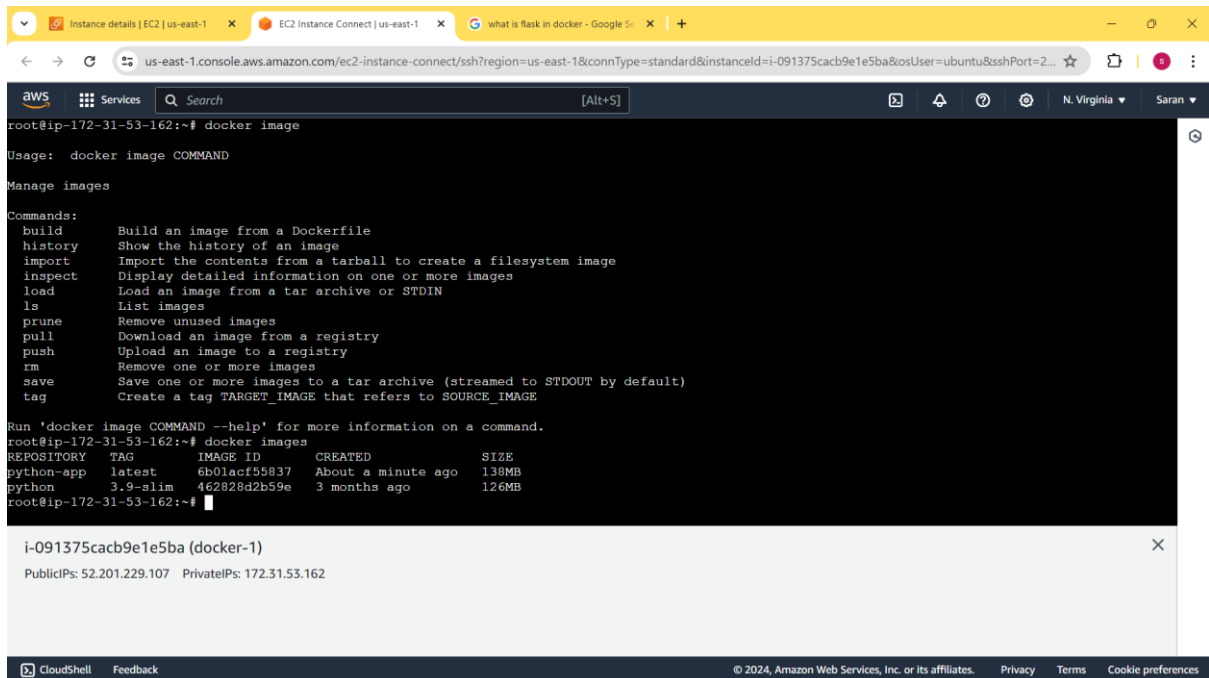
The screenshot shows the AWS CloudShell interface with the same navigation bar as the previous image. The main area displays a terminal window where the user has run the following commands: 'ls', 'cat Dockerfile', and 'docker build -t python-app .'. The metadata box below the terminal shows the same instance ID and IP addresses. The footer is identical to the previous screenshot.

```
root@ip-172-31-53-162:~# ls
Dockerfile app.py snap
root@ip-172-31-53-162:~# docker build -t python-app .
```

i-091375cacb9e1e5ba (docker-1)  
PublicIPs: 52.201.229.107 PrivateIPs: 172.31.53.162

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Now create a container using this image.



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal displays the following commands and output:

```
root@ip-172-31-53-162:~# docker image
Usage: docker image COMMAND

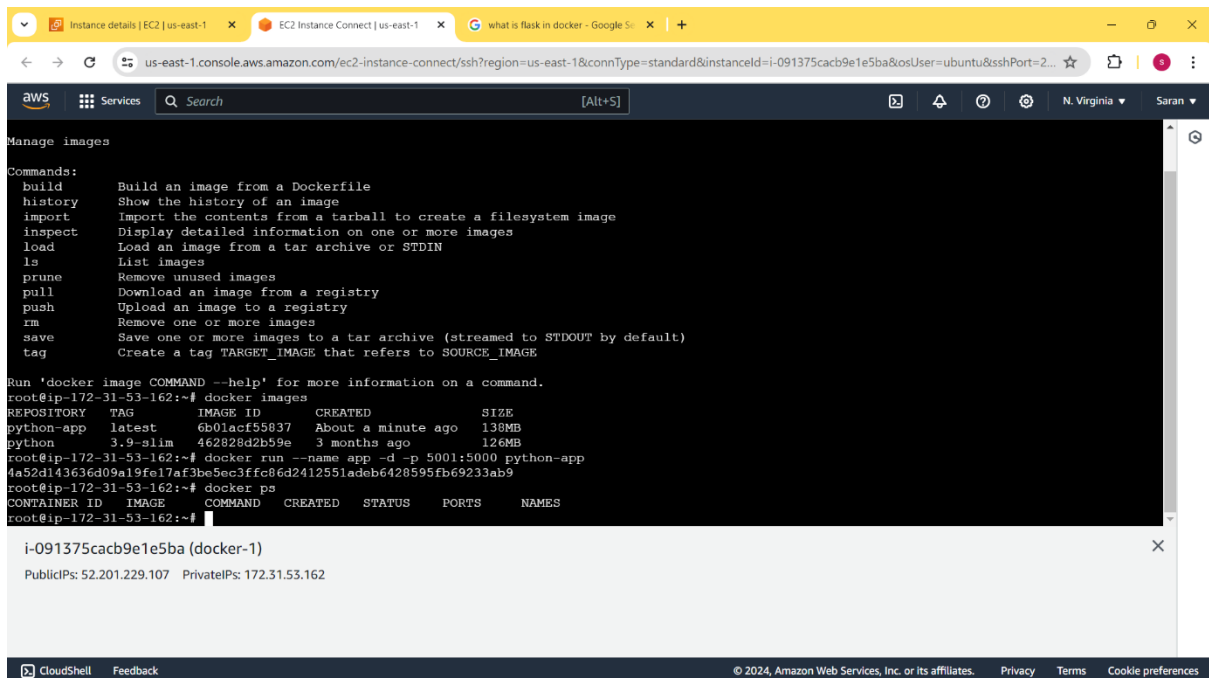
Manage images

Commands:
  build      Build an image from a Dockerfile
  history    Show the history of an image
  import     Import the contents from a tarball to create a filesystem image
  inspect    Display detailed information on one or more images
  load       Load an image from a tar archive or STDIN
  ls         List images
  prune      Remove unused images
  pull       Download an image from a registry
  push       Upload an image to a registry
  rm         Remove one or more images
  save       Save one or more images to a tar archive (streamed to STDOUT by default)
  tag        Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
root@ip-172-31-53-162:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
python-app    latest     6b01acf55837  About a minute ago  138MB
python        3.9-slim   462828d2b59e  3 months ago   126MB
root@ip-172-31-53-162:~#
```

Below the terminal, a summary box for instance `i-091375cacb9e1e5ba (docker-1)` is visible, showing PublicIPs: 52.201.229.107 and PrivateIPs: 172.31.53.162.

- Check whether the container is running.



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal displays the following commands and output:

```
root@ip-172-31-53-162:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
python-app    latest     6b01acf55837  About a minute ago  138MB
python        3.9-slim   462828d2b59e  3 months ago   126MB
root@ip-172-31-53-162:~# docker run --name app -d -p 5001:5000 python-app
4a52d143636d09e19fe17af3be5ec3ffc86d2412551adeb6428595fb69233ab9
root@ip-172-31-53-162:~# docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS              PORTS              NAMES
4a52d143636d09e19fe17af3be5ec3ffc86d2412551adeb6428595fb69233ab9 python-app  "python app"             Up 2 minutes ago    0.0.0.0:5001->0.0.0.0:5000   app
root@ip-172-31-53-162:~#
```

Below the terminal, a summary box for instance `i-091375cacb9e1e5ba (docker-1)` is visible, showing PublicIPs: 52.201.229.107 and PrivateIPs: 172.31.53.162.

- Now access the application using the public-ip address of the instance and the port number.

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows the navigation menu with categories like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area shows the 'Instance summary for i-091375cacb9e1e5ba (docker-1)'. The instance is in a 'Running' state. The public IPv4 address is 52.201.229.107. The instance type is t2.large. The console also shows the private IP address, hostname type, and various DNS names.

Property	Value
Instance ID	i-091375cacb9e1e5ba (docker-1)
Public IPv4 address	52.201.229.107   <a href="#">open address</a>
Private IPv4 addresses	172.31.53.162
Instance state	Running
Public IPv4 DNS	ec2-52-201-229-107.compute-1.amazonaws.com   <a href="#">open address</a>
Private IP DNS name (IPv4 only)	ip-172-31-53-162.ec2.internal
Instance type	t2.large
Elastic IP addresses	-
AWS Compute Optimizer finding	<a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   <a href="#">Learn more</a>
Auto Scaling Group name	-
Hostnames type	IP name: ip-172-31-53-162.ec2.internal
Answer private resource DNS name IPv4 (A)	-
Auto-assigned IP address	52.201.229.107 [Public IP]
VPC ID	vpc-03db3bad2f7381e71
Subnet ID	subnet-0db1494b326c97fff
Instance ARN	-
IAM Role	-
IMDSv2	-

- We successfully run our python script in docker container.

