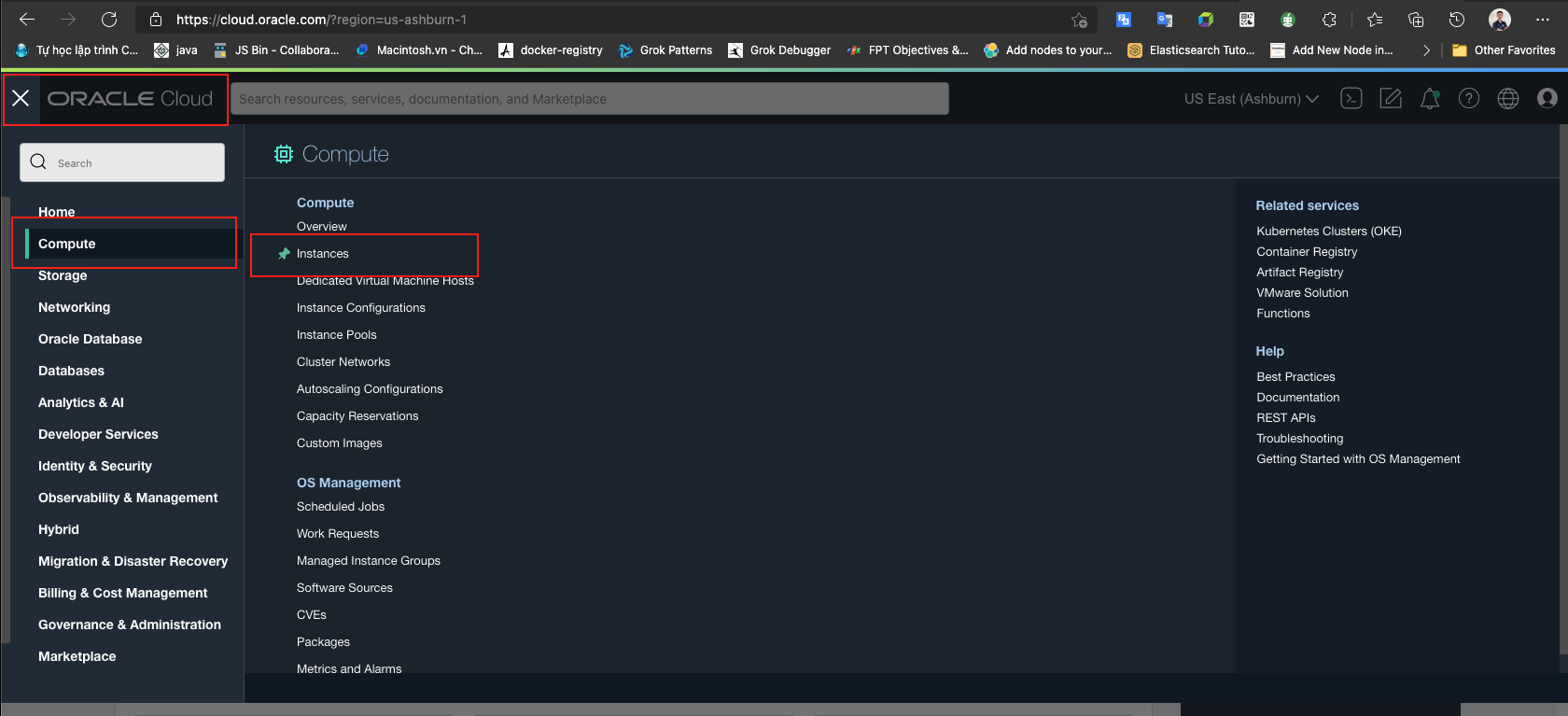
Table of Contents

[Gotoro: Guideline to Create Oracle VM 1](#_Toc118761739)

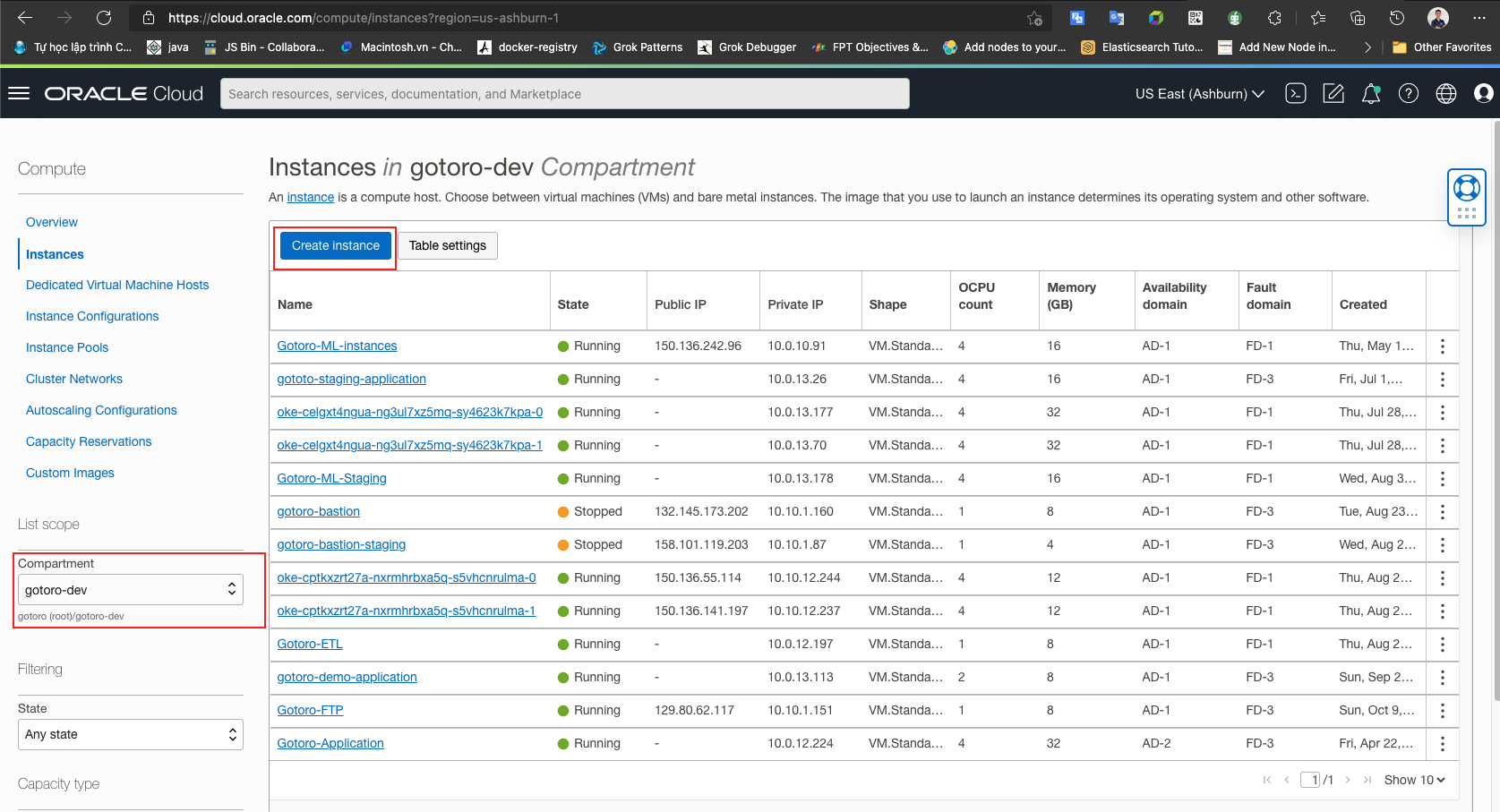
[Gotoro: Guideline to connect to Application VM server and Install Application 1](#_Toc118761740)

# Gotoro: Guideline to Create Oracle VM

Step 1: Login Oracle Cloud: [Oracle Cloud Infrastructure](https://cloud.oracle.com/?region=us-ashburn-1). Go to **Compute** > **Instances**.

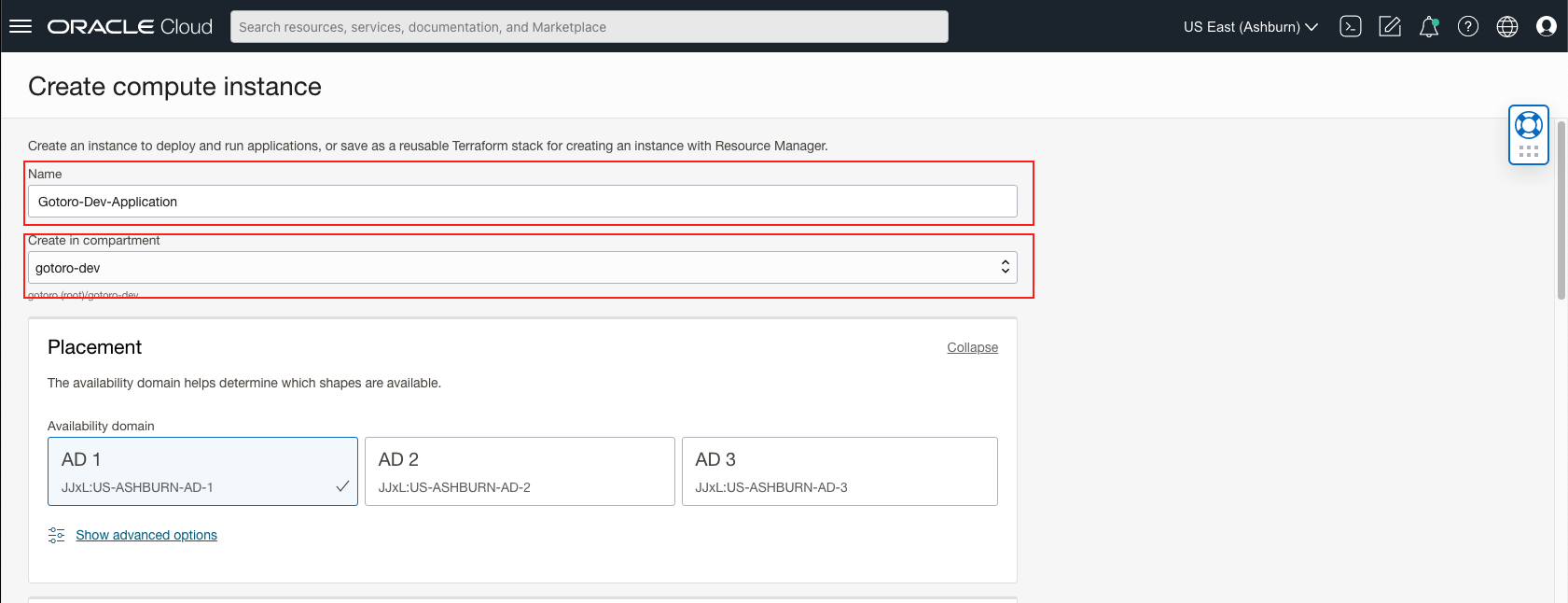


Step 2: Chose **Compartment** for your environment and **Create Instance**.

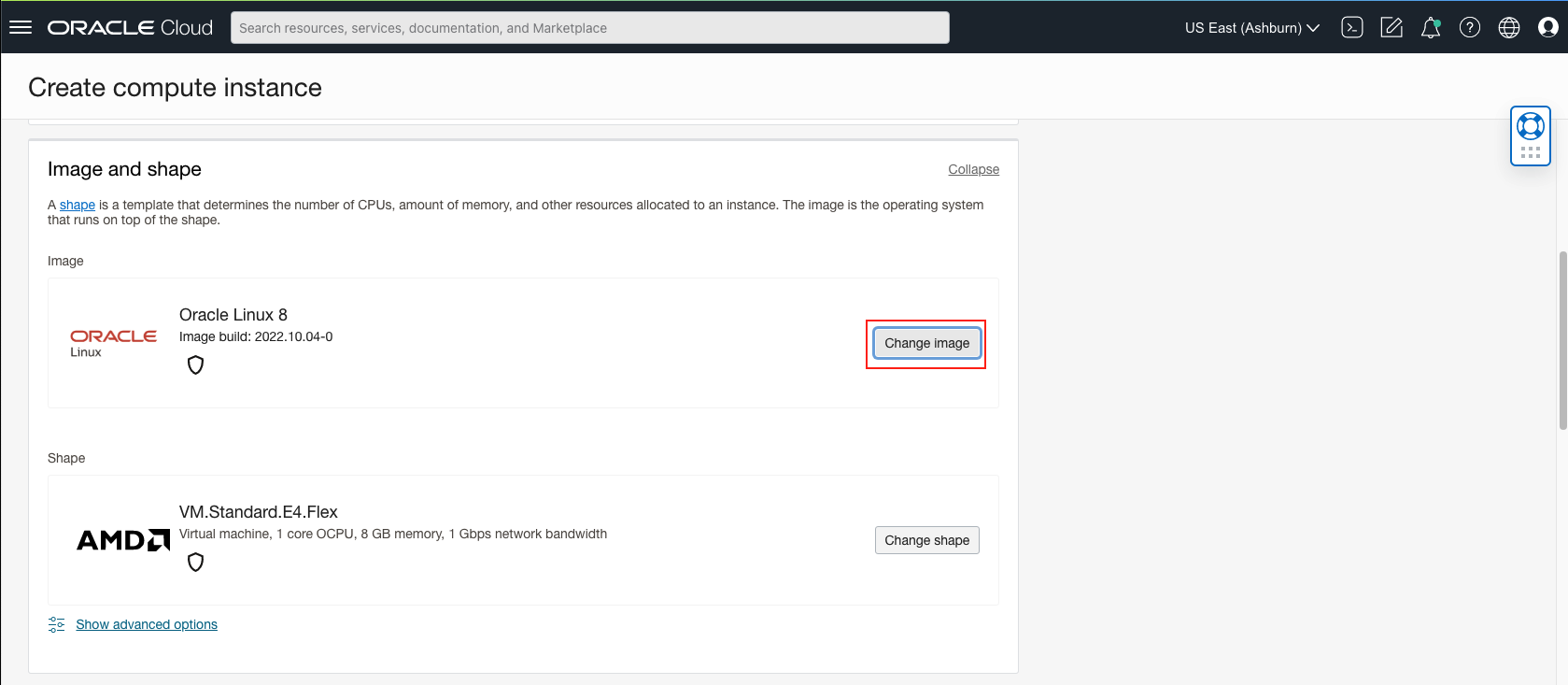


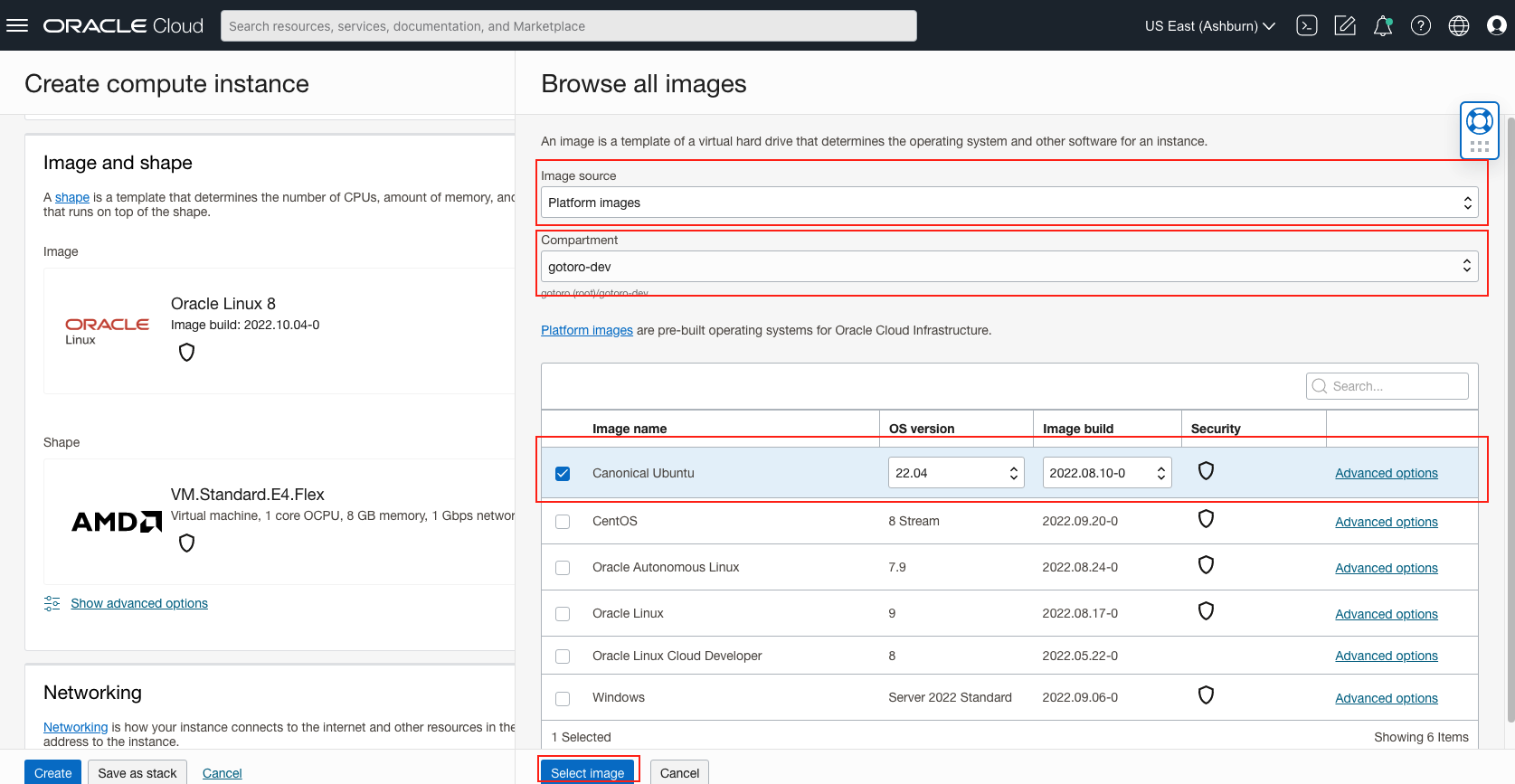
Step 3: Fill name with naming convention **Gotoro-<env>-Application**, and chose **compartment** environment.

Example for **Dev**:

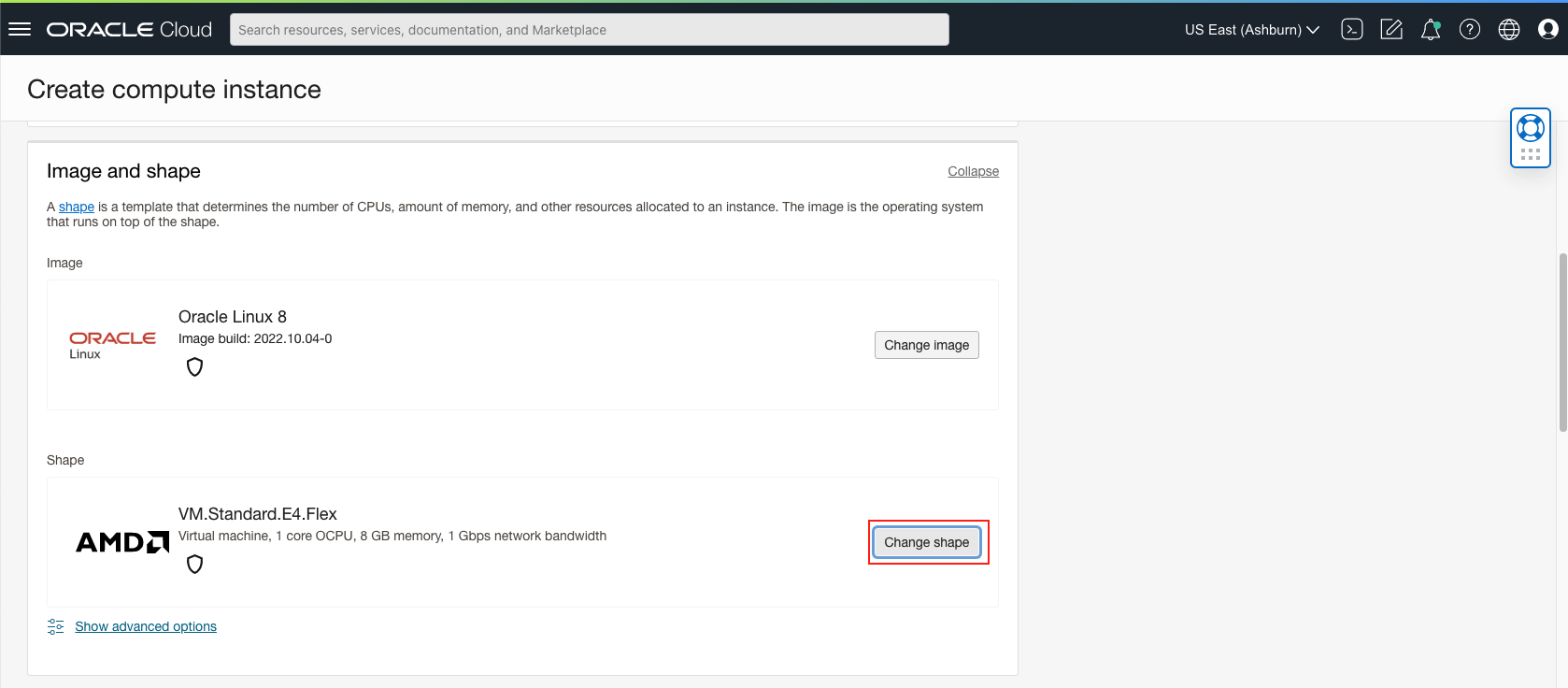


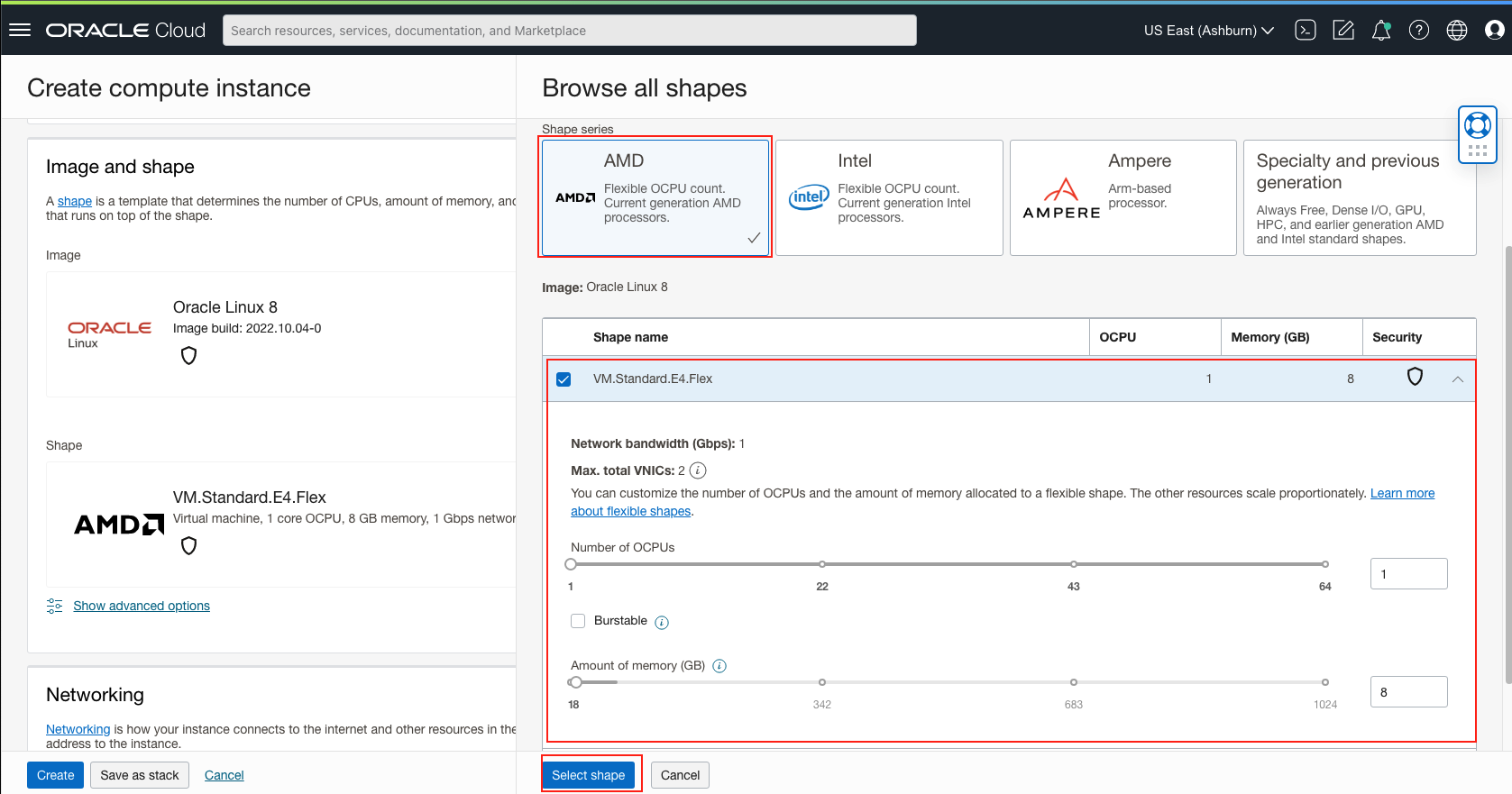
Step 4: Change Image to ubuntu:



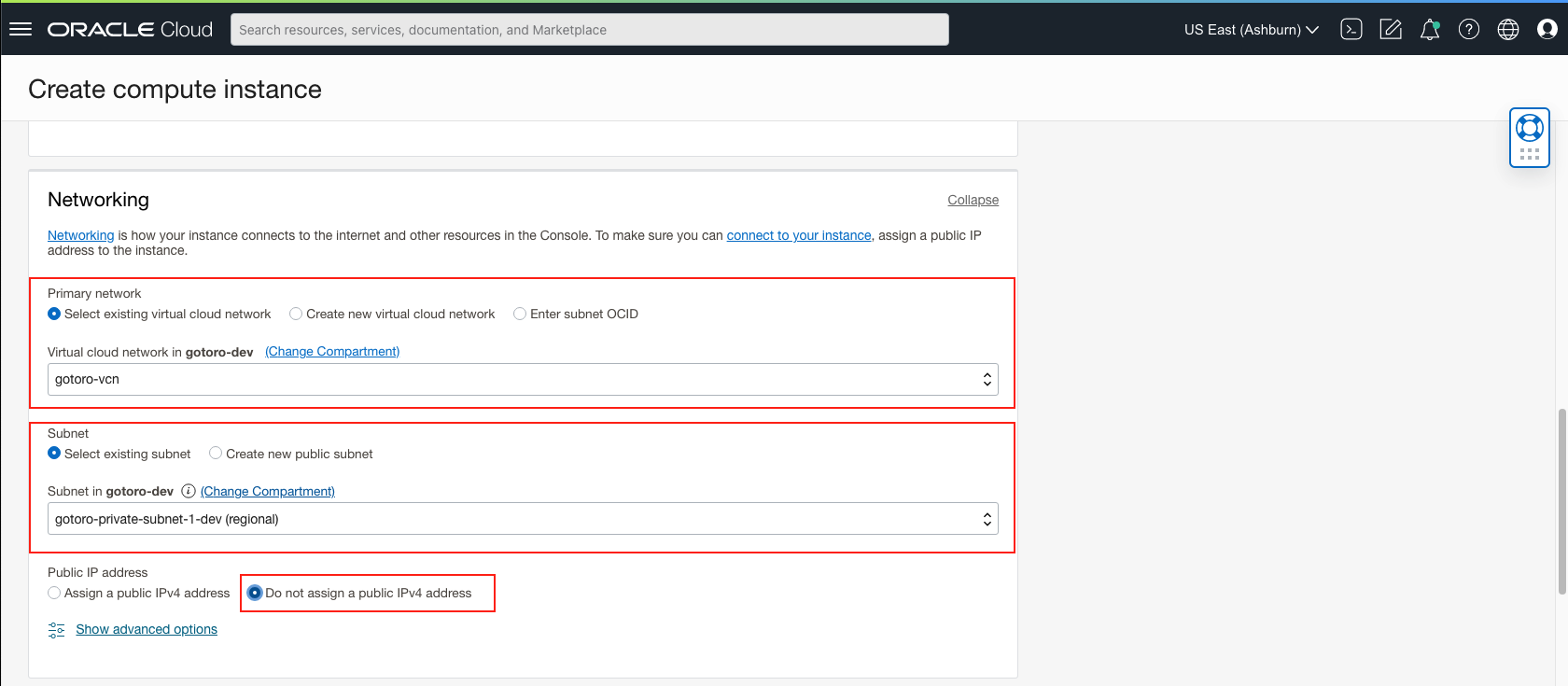


Step 5: Chose Shape for VM.





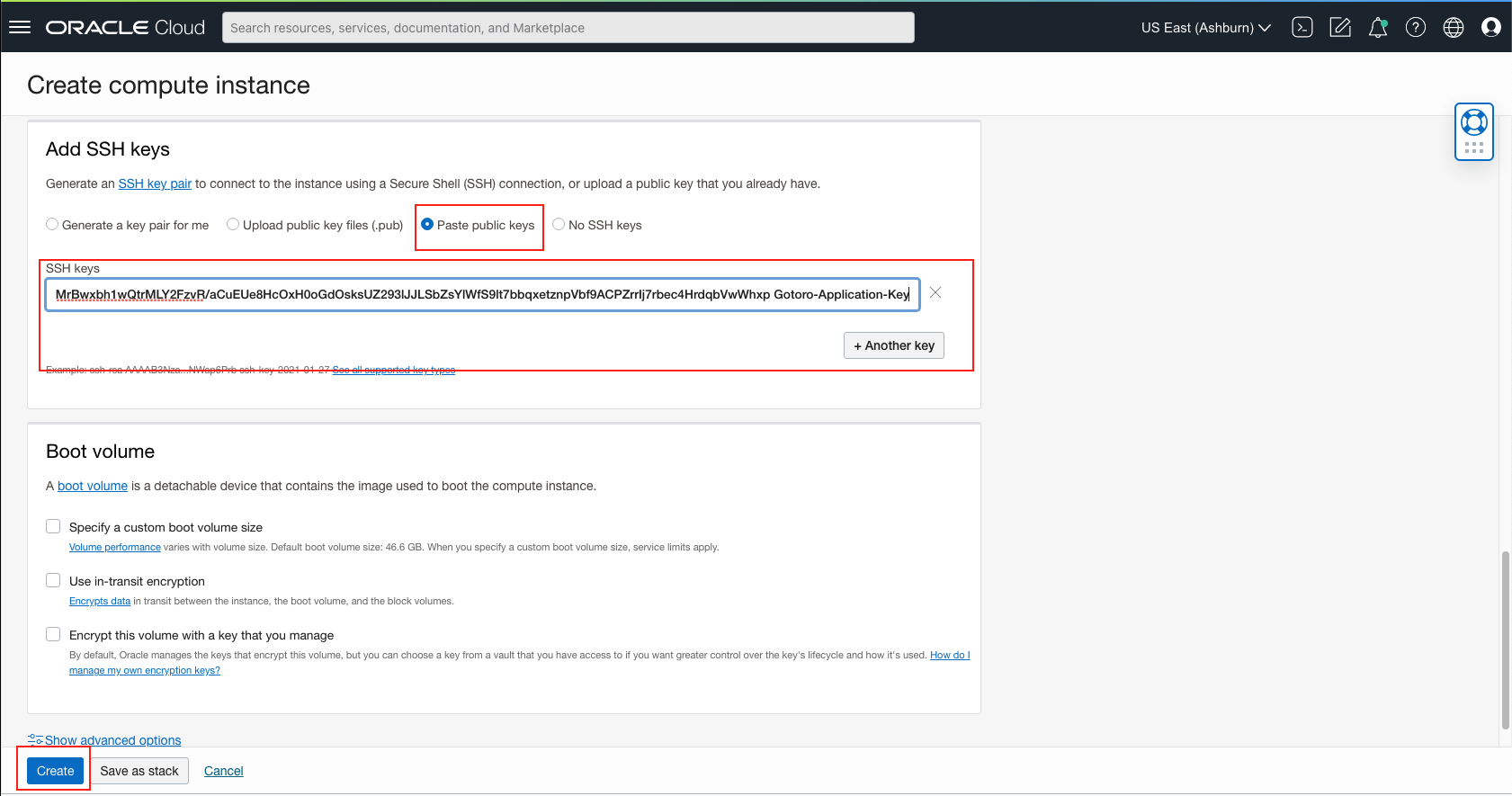
Step 6: Chose network.



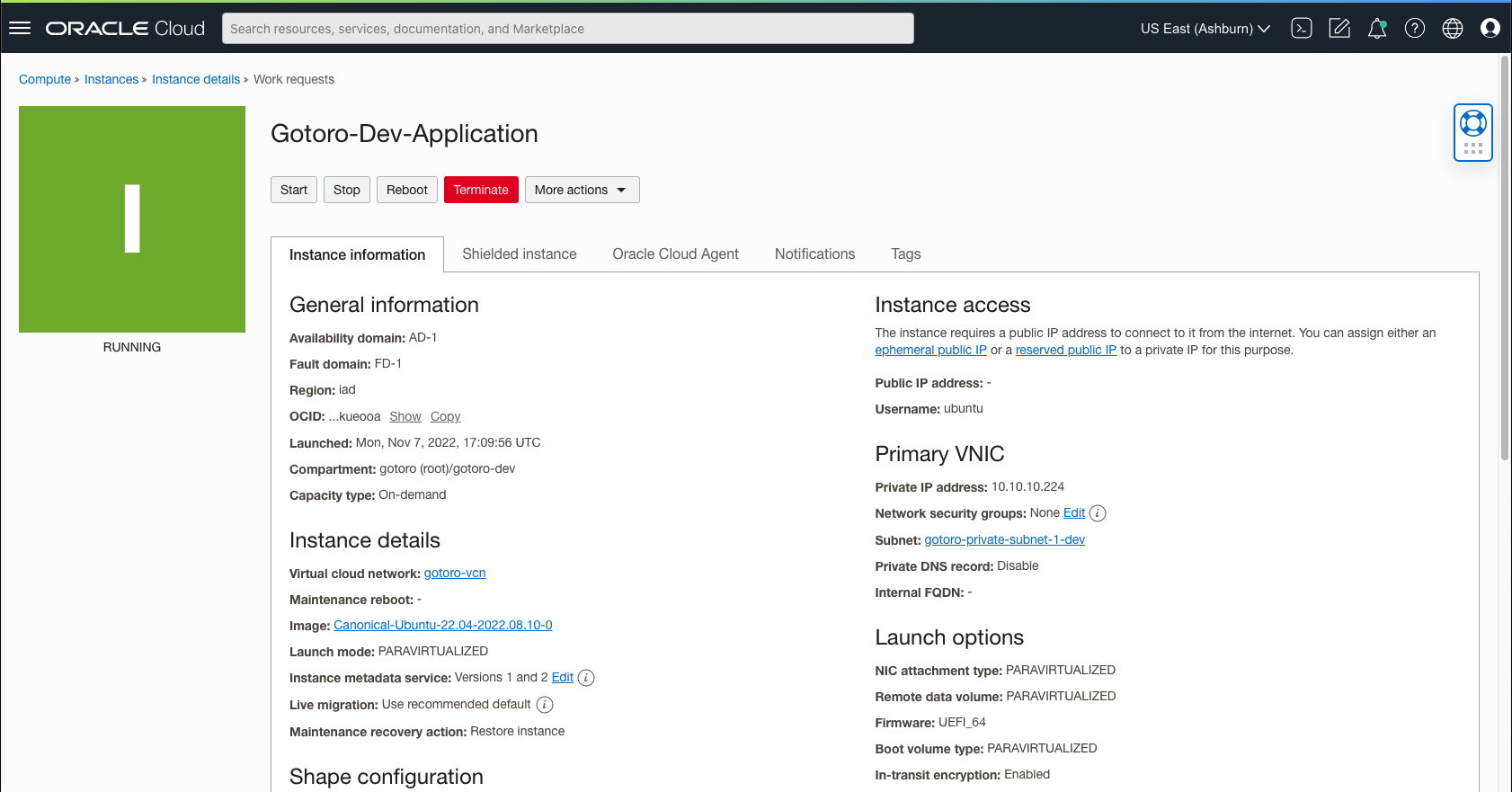
Step 7: Fill public key and Create.

Public key:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDDsJbf8vffDFVzkLtdLQlmmFEom9poo9ZJWixMcCCe0UAuIOZAquf9A4Cw6/jZhQ1WNfVU9Y1Dz5517QOAQu7qqz57ki2MkHBOJdC3Rkf2sLwh5MOtzZuBKaw/pw5ki2hycwIxhyv9AmfeMPj3+vum8+Fe19wm/d/zBCVGGXYI6vY4fLQ0CQ3USt14KCO/FdQVvfwyx8pTgGYMEFh8DT75Em2PeNFPgN98SMjalupqeejpSHDMrBwxbh1wQtrMLY2FzvR/aCuEUe8HcOxH0oGdOsksUZ293lJJLSbZsYlWfS9lt7bbqxetznpVbf9ACPZrrIj7rbec4HrdqbVwWhxp Gotoro-Application-Key



Result:



# Gotoro: Guideline to connect to Application VM server and Install Application

Step 1: Create file private key name: **Gotoro-Application-Key.pem**.

Details key:

-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAw7CW3/L33wxVc5C7XS0JZphRKJvaaKPWSVosTHAgntFALiDm

QKrn/QOAsOv42YUNVjX1VPWNQ8+ede0DgELu6qs+e5ItjJBwTiXQt0ZH9rC8IeTD

rc2bgSmsP6cOZItocnMCMYcr/QJn3jD49/r7pvPhXtfcJv3f8wQlRhl2COr2OHy0

NAkN1ErdeCgjvxXUFb38MsfKU4BmDBBYfA0++RJtj3jRT4DffEjI2pbqanno6Uhw

zKwcMW4dcELazC2Nhc70f2grhFHvB3DsR9KBnTrJLFGdvd5SSS0m2bGJVn0vZbe2

26sXrc56VW3/QAj2a6yI+623nOB63am1cFocaQIDAQABAoIBAFOy8SVCZ9CIFviB

/BGjVLUECxqCZs2A3gOrxOzAvMMRHxFb0v/v2Z/3YpDJrTtkVEr+cztud4h8gftQ

b8HpXwUST3xNDpxjnxVdyULKlbcm6qDYlvV8zrlVGK51BovssLTbCiIi/aaO49w+

OA7g08gZ4dhOHYSz65imukacV0cCcOxRQH3nxpCJj+c+oNcmEWP4TyPYFQhjU4Pj

cE9YfLvWtEudu2QvjlSkN/AVt9Sp4AE70fhJAAZmB+skMYV72bu7+qFuc88zN1jb

RT1bbuRnKv3zvHRjS7TBHekXn+rs3KQDk9zDl74y4vFUwEhuQUDX/RC5c+3WA3i/

RuSszAECgYEA4AuXneB7bWy3/DkxHxDz1aDFvzBgvy7bd7id0Z3n4eEiTAGxyy8k

lchBpfP+MbtC97P3n2hX/5/94KtlGa0RYst5sfPocitJMZXQsXeJ/8Uc4fUObIMc

ilDEMwdRMlzakCDPoK/1RrVfNaxoGAco8cMoLuZcv1ybSHC1QAhgqIECgYEA35ms

ZJMF+5VDgyTXJOu75zkNbzXvwwkrcF18E+8ifHF4sRlxu6+nXQ3AiPAru2S0Bv99

J9LAu8/qP3rYJAzlW/wLuEDNKuFQD8+5XLe/HKVPkD5ex1M3e6zIh5jKGyHioN5Y

Bx3qZeQSbruaE49bdZMbjfEN90fE0K4VM+ppP+kCgYEAgl88guo16J7zERpWszUy

LoW3jGGiap3hp/EOJQQ2AwL26jOyOgHc0CwPx3oL2PSXK3zWAzS8wEtheHNECYTk

MREmZ57/DDCUDv6Wyi49Z1BUVkg7dHl3CiWlhewkG6JcrLHrIB5EJDk2hg0lJZZF

BSoDIAd8wP4qfiTX08MMPAECgYBJokPBQyF/PiJjd6haAj9437Lq4qo4Yjap5D9O

GTwUGxIR0D1B/5Bv9m1nKyBbr8gRXeWLTb4mm0ua6Atxo8WMSy5Y2Q43hsdkzZVP

UTtgZZgT+lYf8emdksLe1JQJ51jIfTAj99gPAdKIIMOWmWkXCQDT94QP4NSRnCJj

XFiYsQKBgA5eQ0tc6EgllujdqYS9WbZOXAEZvs9an6nJkvy4kdgrC8u1JRW63evZ

cuB/qyBEPJUlpI+65a9EH5yCu1dCDm6XKAXCXrrcNnKB4NTKRcGb1JH9mwS1eTEq

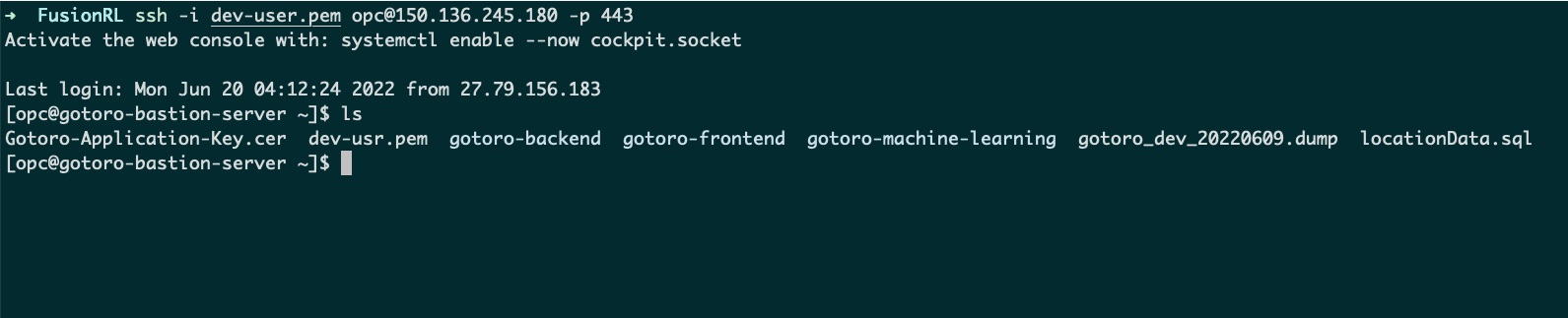
voHuEjit41SLV5F3yBzFgmy2EFyWspOED2Qjkolx6OaSxdU0U1EM

-----END RSA PRIVATE KEY-----

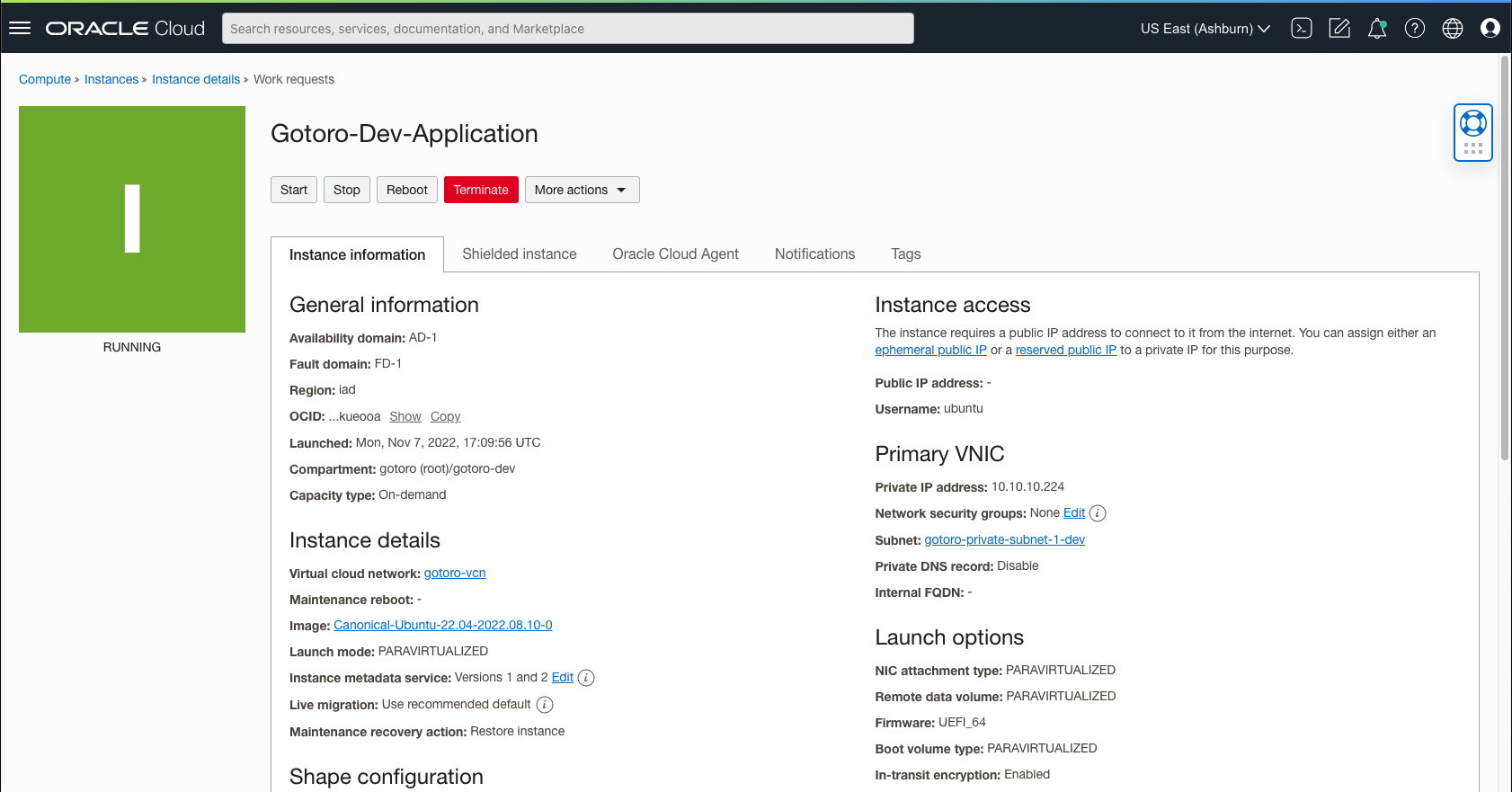
Step 2: SSH to Bastion VM by **Gotoro-Application-Key.pem**.

ssh -i dev-user.pem opc@150.136.245.180 -p 443

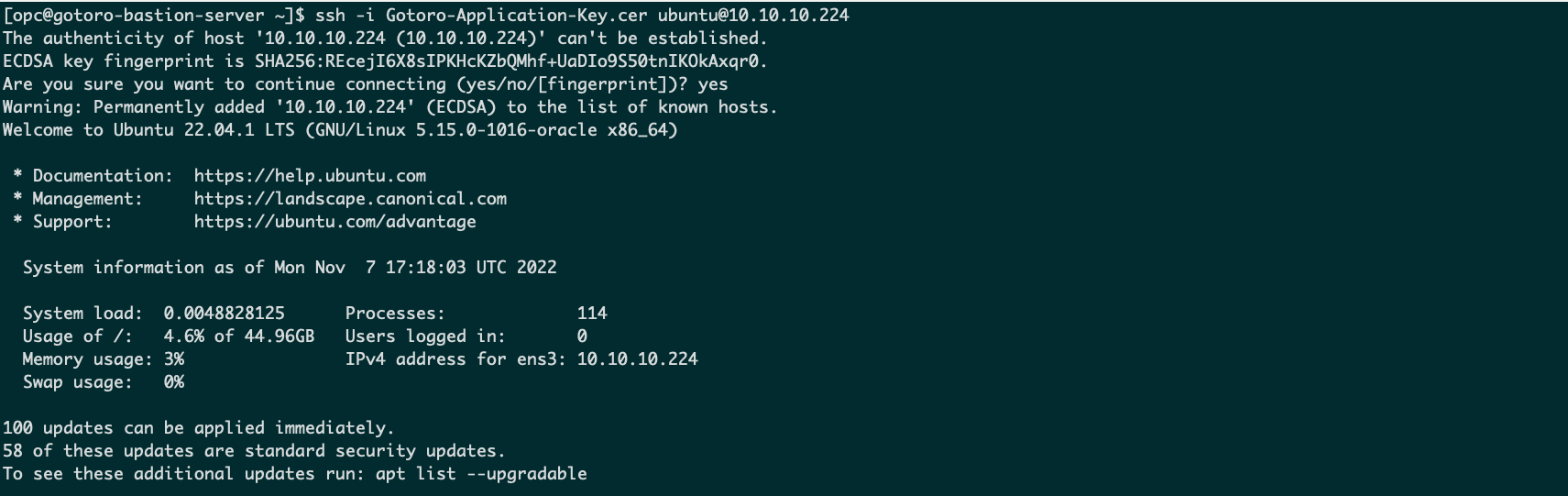
Example:



Step 3: Connect to Application VM with host is **10.10.10.224**, Username is ubuntu and key is Gotoro-Application-Key.cer



Example:



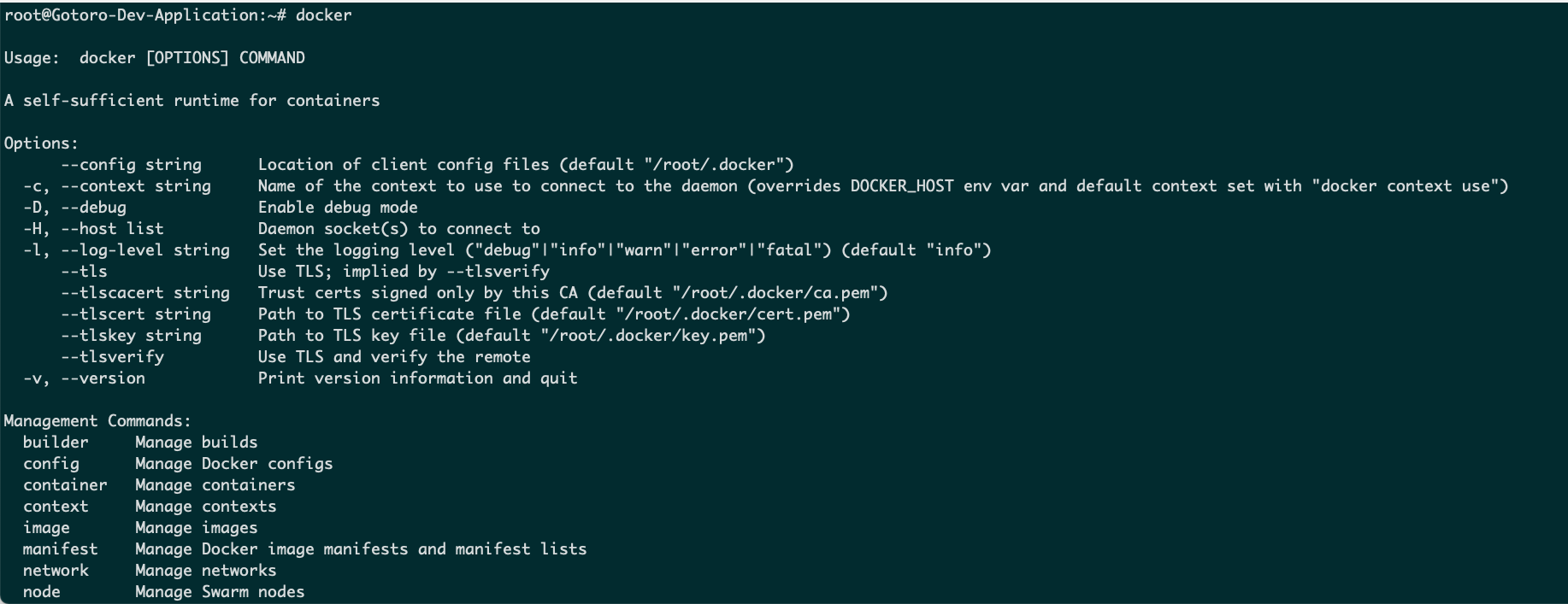
Step 4: Install Docker, Docker-compose

sudo -i

apt install docker.io -y

apt install docker-compose -y

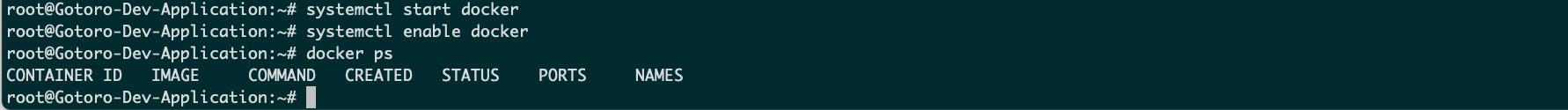
Result:



Step 5: Start, enable Docker.

systemctl start docker

systemctl enable docker



Step 6: Create folder docker-compose for application in **/home/ubuntu/<env>:**

Example for **dev:**

mkdir /home/ubuntu/gotoro-dev

Step 7: Create file docker-compose.yml for application in **/home/ubuntu/<env>:**

vi /home/ubuntu/gotoro-dev/docker-compose.yml

Value file:

version: '3'

services:

backend-<env>:

image: us-ashburn-1.ocir.io/idapel7w8ikn/gotoro-backend:<env>-1128

container\_name: gotoro-backend-<env>

ports:

- 3000:3000

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always

networks:

- gotoro-<env>-net

backend-tracking-<env>:

image: us-ashburn-1.ocir.io/idapel7w8ikn/backend-tracking:<env>-101

container\_name: backend-tracking-<env>

ports:

- 3100:3001

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always

networks:

- gotoro-<env>-net

frontend-<env>:

image: us-ashburn-1.ocir.io/idapel7w8ikn/gotoro-frontend:<env>-998

container\_name: gotoro-frontend-<env>

ports:

- 80:80

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always

networks:

- gotoro-<env>-net

mq:

image: rabbitmq:3.10-management

container\_name: gotoro-mq-<env>

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

ports:

- 5672:5672

- 15672:15672

networks:

- gotoro-<env>-net

solr:

image: solr:9

container\_name: gotoro-solr-<env>

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

ports:

- 8983:8983

restart: always

volumes:

- /data/solr/solr\_<env>:/var/solr

user: root

command: bash -c "chown -R 8983:8983 /var/solr && runuser -u solr -- solr-precreate <env>-core"

networks:

- gotoro-<env>-net

networks:

gotoro-<env>-net:

Value file example for **dev**:

version: '3'

services:

backend-dev:

image: us-ashburn-1.ocir.io/idapel7w8ikn/gotoro-backend:dev-1128

container\_name: gotoro-backend-dev

ports:

- 3000:3000

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always

networks:

- gotoro-dev-net

backend-tracking-dev:

image: us-ashburn-1.ocir.io/idapel7w8ikn/backend-tracking:dev-101

container\_name: backend-tracking-dev

ports:

- 3100:3001

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always  
 networks:

- gotoro-dev-net

frontend-dev:

image: us-ashburn-1.ocir.io/idapel7w8ikn/gotoro-frontend:dev-998

container\_name: gotoro-frontend-dev

ports:

- 80:80

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

restart: always

networks:

- gotoro-dev-net

mq:

image: rabbitmq:3.10-management

container\_name: gotoro-mq-dev

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

ports:

- 5672:5672

- 15672:15672

networks:  
 - gotoro-dev-net

solr:

image: solr:9

container\_name: gotoro-solr-dev

logging:

driver: "json-file"

options:

max-size: "1g"

max-file: "3"

ports:

- 8983:8983

restart: always

volumes:

- /data/solr/solr\_dev:/var/solr

user: root

command: bash -c "chown -R 8983:8983 /var/solr && runuser -u solr -- solr-precreate dev-core"

networks:

- gotoro-dev-net

networks:

gotoro-dev-net:

Step 8: Login oracle Registry:

docker login us-ashburn-1.ocir.io

Username: idapel7w8ikn/serviceaccount

Password: DevOps@2022@!

Step 9: Start application:

cd /home/ubuntu/gotoro-dev  
docker-compose up -d

**End Guideline.**