

NWEN243 Project 3 Lab Report Part B

Raashna Chand

This lab report documents how I followed the steps laid out in part B of the instructions of the project. Note that the answers to the last two questions are on page 11.

1.



New instance from part 3A image created.

```
[ec2-user@ip-172-31-94-67 ~]$ sudo crontab -r  
[ec2-user@ip-172-31-94-67 ~]$
```

Crontab removed. Instance rebooted.

2.

a.

```
[ec2-user@ip-172-31-94-67 ~]$ sudo yum update  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
No packages marked for update
```

sudo yum update ran. No updates.

b.

```
[ec2-user@ip-172-31-94-67 ~]$ sudo amazon-linux-extras install docker  
Installing docker
```

Docker installed.

c.

```
[ec2-user@ip-172-31-94-67 ~]$ sudo service docker start  
Redirecting to /bin/systemctl start docker.service
```

Docker started.

d.

```
[ec2-user@ip-172-31-94-67 ~]$ sudo usermod -a -G docker ec2-user
```

Default user added.

e.

```
[ec2-user@ip-172-31-94-67 ~]$ exit  
logout  
Connection to 44.203.47.204 closed.  
raashna@raashna-IdeaPad-Flex-5-14ARE05:~/Desktop/uni/NWEN243/project 3$
```

Logged out of instance.

f.

```
raashna@raashna-IdeaPad-Flex-5-14ARE05:~/Desktop/uni/NWEN243/project 3$ ssh -i n  
wenproj3b.pem ec2-user@44.203.47.204  
Last login: Tue Oct 3 01:35:42 2023 from 151.210.163.252  
#  
~\##### Amazon Linux 2  
~~\#####  
~~\##### AL2 End of Life is 2025-06-30.  
~~\#/V~'.'->  
~~  
~~ A newer version of Amazon Linux is available!  
~~. ./ /  
~~ / /  
~~ /m/' /  
[ec2-user@ip-172-31-94-67 ~]$
```

SSH-ed back in.

g.

```
[ec2-user@ip-172-31-94-67 ~]$ docker info
Client:
 Context:      default
 Debug Mode:  false
 Plugins:
  buildx: Docker Buildx (Docker Inc., 0.0.0+unknown)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
```

No containers.

3.

```
[ec2-user@ip-172-31-94-67 ~]$ ls
Dockerfile      MusicGuruHealthCheck.class  run.sh
helloworld.class MusicGuruHealthCheck.java   SongEntry.class
musicdata.txt   MusicGuruServer.class
MusicGuruClient.class MusicGuruServer.java
```

Source files added.

4.



File created on local machine.

5.

```
FROM openjdk:8
COPY *.java /usr/src/MGS/
COPY musicdata.txt /usr/src/MGS/
WORKDIR /usr/src/MGS
RUN javac MusicGuruServer.java
EXPOSE 5000
CMD ["java", "MusicGuruServer", "5000"]
```

Contents of Dockerfile.

6-7.

```
[ec2-user@ip-172-31-80-74 ~]$ docker build mgs -t .
invalid argument "." for "-t, --tag" flag: invalid reference format
See 'docker build --help'.
[ec2-user@ip-172-31-80-74 ~]$ docker build -t mgs .
Sending build context to Docker daemon 10.41MB
Step 1/7 : FROM openjdk:8
8: Pulling from library/openjdk
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
52a8c426d30b: Pull complete
8754a66e0050: Pull complete
Digest: sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8
Status: Downloaded newer image for openjdk:8
---> b273004037cc
Step 2/7 : COPY *.java /usr/src/MGS/
---> e34c2b2fa49c
Step 3/7 : COPY musicdata.txt /usr/src/MGS/
---> 8e0e375258d2
Step 4/7 : WORKDIR /usr/src/MGS
---> Running in eb6129352c99
Removing intermediate container eb6129352c99
---> e351c59f22b4
Step 5/7 : RUN javac MusicGuruServer.java
---> Running in 8e935fe689a7
Removing intermediate container 8e935fe689a7
---> f69df5b4f954
Step 6/7 : EXPOSE 5000
---> Running in b8a3cb5dcf33
Removing intermediate container b8a3cb5dcf33
---> e4705b17ab86
Step 7/7 : CMD ["java", "MusicGuruServer", "5000"]
---> Running in 194eddb5a737
Removing intermediate container 194eddb5a737
---> b69f7f98bceb
Successfully built b69f7f98bceb
Successfully tagged mgs:latest
```

Image built.

8.

```
[ec2-user@ip-172-31-80-74 ~]$ docker images --filter reference=mgs
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
mgs              latest      b69f7f98bceb  30 seconds ago 526MB
```

Seems like image was built properly.

9.

```
[ec2-user@ip-172-31-80-74 ~]$ docker run -t -i -p 5000:5000 mgs &
[1] 6660
```

Mapped exposed port on container to port on machine. Process ID is 6660.

10.

```
[ec2-user@ip-172-31-80-74 ~]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
119dd2d0f855   mgs        "java MusicGuruServe..." 36 seconds ago Up 35 seconds 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  modest_goldstine
```

Everything is running fine.

13.

```
[ec2-user@ip-172-31-80-74 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
119dd2d0f855	mgs	"java MusicGuruServe..."	10 minutes ago	Up 10 minutes	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp	modest_goldstine

Prior docker executions.

15-17.

Create your account

Signing up for Docker is fast and free.

Email

Username

Password

Account signed up for. Email also validated.

18-19

```
[ec2-user@ip-172-31-80-74 ~]$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: raashnachand
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/conf
ig.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
Logged in.
```

20.

```
[ec2-user@ip-172-31-80-74 ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
119dd2d0f855	mgs	"java MusicGuruServe..."	23 minutes ago	Up 23 minutes

```
0.0.0.0:5000->5000/tcp, :::5000->5000/tcp modest_goldstine
[ec2-user@ip-172-31-80-74 ~]$
```

Server still running.

21.

```
[ec2-user@ip-172-31-80-74 ~]$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mgs	latest	b69f7f98bceb	28 minutes ago	526MB
openjdk	8	b273004037cc	14 months ago	526MB

Checking the name of my docker image's repository.

```
[ec2-user@ip-172-31-80-74 ~]$ docker tag mgs raashnachand/mgs
```

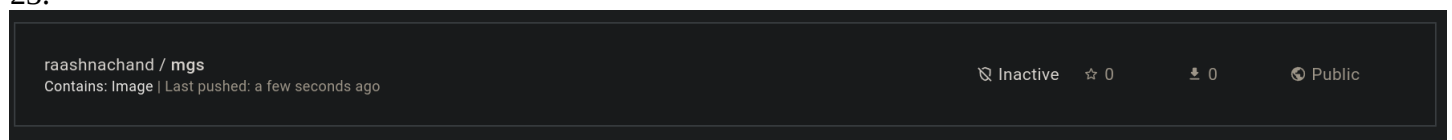
Added image tag that references the repository.

22.

```
[ec2-user@ip-172-31-80-74 ~]$ docker push raashnachand/mgs:latest
The push refers to repository [docker.io/raashnachand/mgs]
c07481fa8159: Pushed
1fe5284d65b3: Pushed
246df53f8f27: Pushed
6b5aaff44254: Pushing 131.4MB/209.2MB
53a0b163e995: Pushed
b626401ef603: Pushed
9b55156abf26: Pushing 73.96MB/152MB
293d5db30c9f: Pushed
03127cdb479b: Pushed
9c742cd6c7a5: Pushing 29.28MB/124MB
```

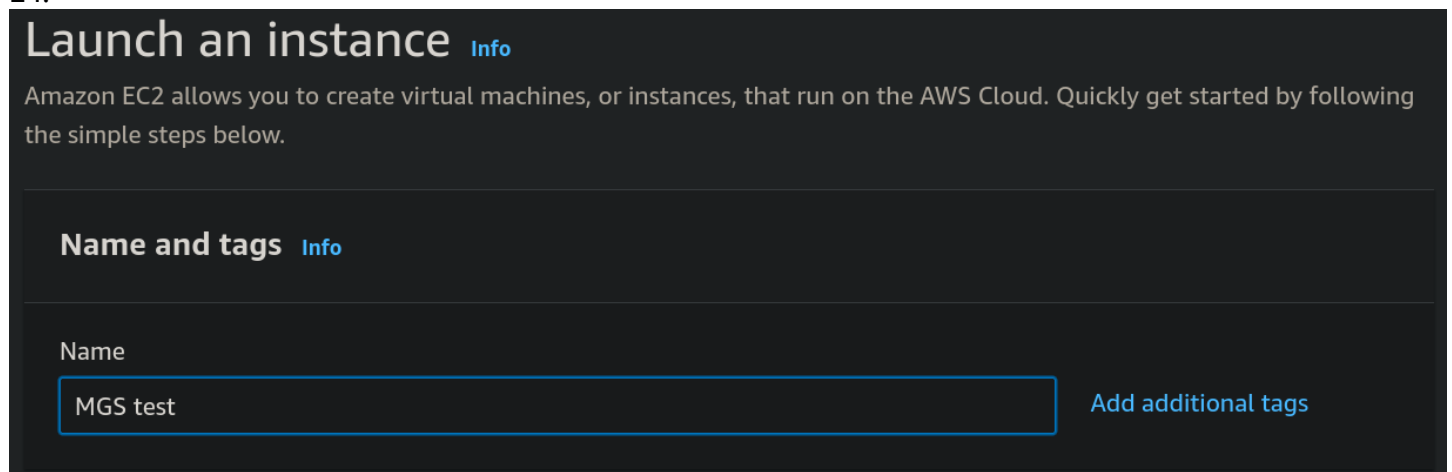
In the process of being pushed.

23.



Repository made.

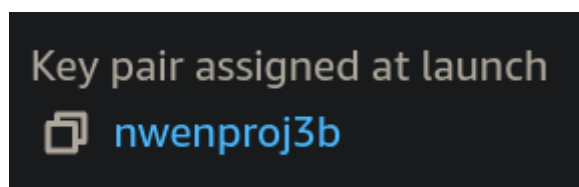
24.



Creating new vanilla instance with Amazon Linux 2 AMI HVM.

25.

Same keys used.



26.

```
[ec2-user@ip-172-31-95-145 ~]$ sudo yum update
Last metadata expiration check: 0:04:10 ago on Wed Oct 4 08:07:56 2023.
Dependencies resolved.
Nothing to do.
Complete!
```

SSH-ing into new instance, doing all the updates and installations which won't be documented.

27.


```
[ec2-user@ip-172-31-85-225 ~]$ docker pull docker.io/raashnachand/mgs:latest
latest: Pulling from raashnachand/mgs
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
52a8c426d30b: Pull complete
8754a66e0050: Pull complete
9d574abf8b28: Pull complete
ab0c5a2e2736: Pull complete
6e35ef1c934f: Pull complete
Digest: sha256:85685fe0d056978fd6b712d409fb8cbce16c12e925a8b8610f83f55762450019
Status: Downloaded newer image for raashnachand/mgs:latest
docker.io/raashnachand/mgs:latest
[ec2-user@ip-172-31-85-225 ~]$
```

Docker image pulled from my repository.

28.

```
[ec2-user@ip-172-31-85-225 ~]$ docker run -t -i -p 5000:5000 raashnachand/mgs:la
test &
[1] 3775
```

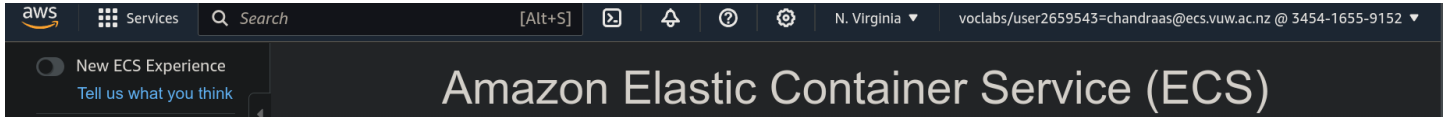
Running the container. PID of 3775.

29.

```
raashna@raashna-IdeaPad-Flex-5-14ARE05:~/Desktop/uni/NWEN243/project 3/proj 3b$
java MusicGuruClient 52.87.159.216 5000 1976
Range: 1950-2009
Sending year 1976
In 1976 the number 3 song was More Than a Feeling by Boston
(172.17.0.2)
```

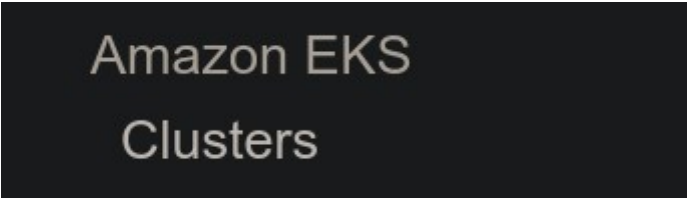
Runs perfectly.

30-31.



ECS dashboard. New ECS experience turned off.

32.



Clusters.

Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.

[Get started](#)

Get started.

33-34.

Edit container

Container name*

MGS-container

Image*

docker.io/raashnachand/mgs:latest

Private repository authentication*

Memory Limits (MiB)

Soft limit

▼

128

+

 Add Hard limit

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port

5000

Protocol

tcp ▼

Configuration settings.

35.

Configure task definition: first-run-task-definition

Task definition details

Task definition name*

first-run-task-definition



Network mode*

awsvpc



Task execution role

LabRole



Compatibilities*

FARGATE



[Learn more](#) about compatibilities

Task definition configured.

36.

Define your service

Edit

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name MGS-container-service

Number of desired tasks 1

Security group Automatically create new

A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

Load balancer type ☒ None

☐ Application Load Balancer

Service definition unchanged.

37.

Configure your cluster

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the [Amazon ECS documentation](#).

Cluster name mgs-cluster

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.

VPC ID Automatically create new



Subnets Automatically create new



Name changed, defaults left as is.

38-40.

Launch Status

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

BackView service

Additional features that you can add to your service after creation

Scale based on metrics

You can configure scaling rules based on CloudWatch metrics

Preparing service : 9 of 9 complete

ECS resource creation

Cluster `mgs-cluster`

Task definition `first-run-task-definition:1`

Service `MGS-container-service`

Additional AWS service integrations

Log group `/ecs/first-run-task-definition`

CloudFormation stack `EC2ContainerService-mgs-cluster`

VPC `vpc-026455630a9ae74ce`

Subnet 1 `subnet-09c6df4d76b384730`

Subnet 2 `subnet-0ad20fad16ca2ed8d`

Security group `sg-03dda87d073ea070a`

complete

complete

complete

complete

complete

complete

complete

complete

complete

complete

complete

complete

complete

All complete.

41-42.

Service : MGS-container-service

UpdateDelete

Cluster

`mgs-cluster`

Desired count

1

Status

ACTIVE

Pending count

0

Task definition

`first-run-task-definition:1`

Running count

1

Service type

REPLICA

Launch type

FARGATE

Service role

`AWSServiceRoleForECS`

Created By

`arn:aws:iam::345416559152:role/voclabs`

Task is being provisioned.

43.

DetailsTasksEventsAuto ScalingDeploymentsMetricsTagsLogs

Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

Network Access

Allowed VPC	<code>vpc-026455630a9ae74ce</code>
Allowed subnets	<code>subnet-09c6df4d76b384730,subnet-0ad20fad16ca2ed8d</code>
Security groups*	<code>sg-03dda87d073ea070a</code>
Auto-assign public IP	ENABLED

All tabs showing.

46.

Details	Tasks	Events	Auto Scaling	Deployments	Metrics	Tags	Logs
Last updated on October 5, 2023 10:59:14 AM (0m ago)							
Task status: Running Stopped							
Filter in this page							< 1-1 > Page size 50
Task	Task Definition...	Last status	Desired status...	Group	Launch type	Platform versi...	
d0b3f11b902a4...	first-run-task-def...	RUNNING	RUNNING	service:MGS-co...	FARGATE	1.4.0	

Selecting “tasks”.

47.

Network

Network mode	awsvpc
ENI Id	eni-015316daae91a41bd
Subnet Id	subnet-09c6df4d76b384730
Private IP	10.0.0.6
Public IP	3.88.142.89
Mac address	0a:37:1b:8f:4d:61

Public IP address there.

48.

```

raashna@raashna-IdeaPad-Flex-5-14ARE05:~/Desktop/uni/NWEN243/project 3/proj 3b$
java MusicGuruClient 3.88.142.89 5000 1976
Range: 1950-2009
Sending year 1976
In 1976 the number 10 song was Play That Funky Music by Wild Cherry
(10.0.0.6)

```

Works!!!

50.

Everything is set up the same, apart from the names and the number of desired tasks, seen here:

Set up service: MGS2-service

Service name*

MGS2-service

Number of desired tasks*

3

51.

Q1: The IP addresses for each replica are unique.

Private IP	10.0.0.169	Private IP	10.0.1.192	Private IP	10.0.1.143
Public IP	52.204.192.118	Public IP	3.87.111.228	Public IP	54.227.170.47

Along with their MAC addresses too.

52.

Q2: We might implement a load balancer so that the client can use one domain name instead of having to remember three IP addresses.