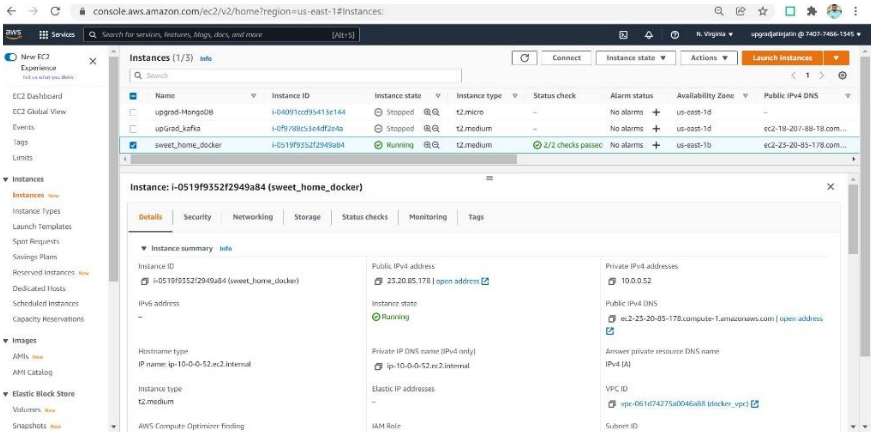
Dockerized Hotel Booking Application – Sweet Home

Deployment Work Done by - Jatin

Coding Logic -

Steps Followed to solve the given project problem:

1. Create an EC2 Instance with Ubuntu having own VPC.



console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:

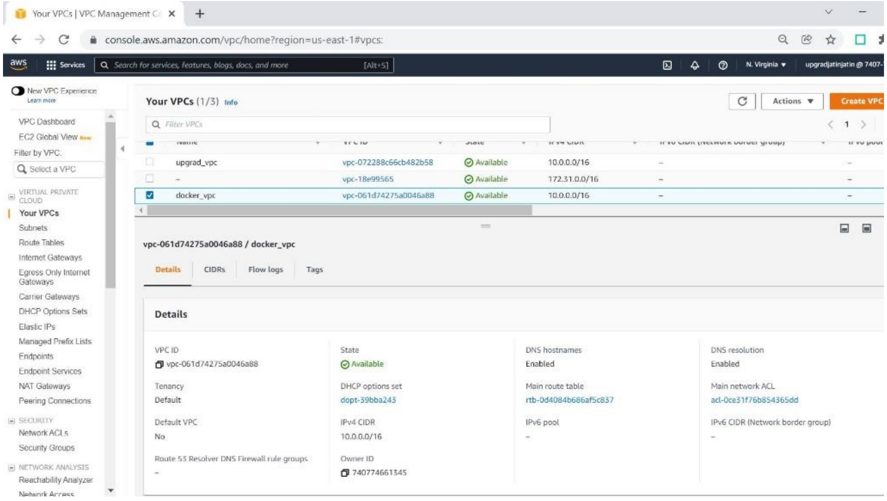
Instances (1/3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
upgrad-MongoDB	i-04091ccd95413e144	Stopped	t2.micro	–	No alarms	us-east-1d	–
upGrad_kafka	i-0f9788c53e4df2e4a	Stopped	t2.medium	–	No alarms	us-east-1d	ec2-18-207-88-18.com...
sweet_home_docker	i-0519f9352f2949a84	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	ec2-23-20-85-178.com...

Instance: i-0519f9352f2949a84 (sweet_home_docker)

Instance summary

Instance ID: i-0519f9352f2949a84 (sweet_home_docker)
Public IPv4 address: 23.20.85.178 | open address
Private IPv4 addresses: 10.0.0.52
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-23-20-85-178.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-10-0-0-52.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-0-52.ec2.internal
Answer private resource DNS name: IPv4 (A)
Instance type: t2.medium
Elastic IP addresses: –
VPC ID: vpc-061d74275a0046a88 (docker_vpc)



console.aws.amazon.com/vpc/home?region=us-east-1#vpcs:

Your VPCs (1/3)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool
upgrad_vpc	vpc-072288c66cb482b58	Available	10.0.0.0/16	–	–
–	vpc-18e99565	Available	172.31.0.0/16	–	–
docker_vpc	vpc-061d74275a0046a88	Available	10.0.0.0/16	–	–

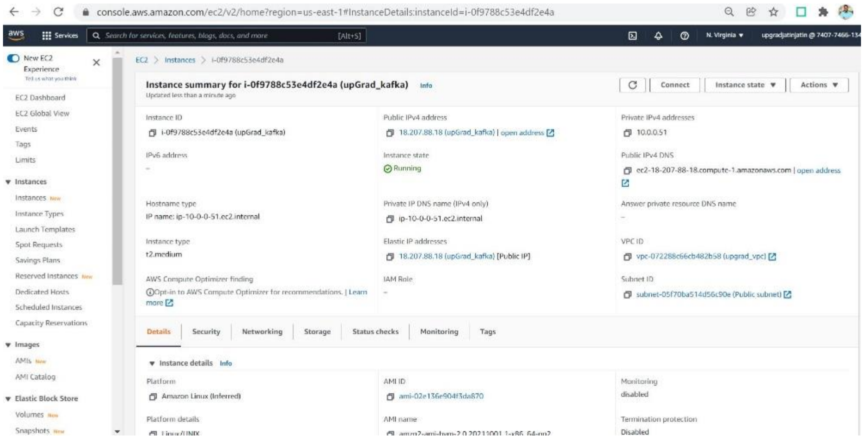
vpc-061d74275a0046a88 / docker_vpc

Details

VPC ID: vpc-061d74275a0046a88
State: Available
DNS hostnames: Enabled
DNS resolution: Enabled
Tenancy: Default
DHCP options set: dopt-39bba243
Main route table: rtb-0d4084b686af5c837
Main network ACL: acl-0ce31f76b854365dd
Default VPC: No
IPv4 CIDR: 10.0.0.0/16
IPv6 pool: –
IPv6 CIDR (Network border group): –
Route 53 Resolver DNS Firewall rule groups: –
Owner ID: 740774661345

2. Configured the Security Group for appropriate inbound rules.

3. Create and EC2 instance to host Externalized Kafka.



console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-0f9788c53e4df2e4a

Instance summary for i-0f9788c53e4df2e4a (upGrad_kafka)

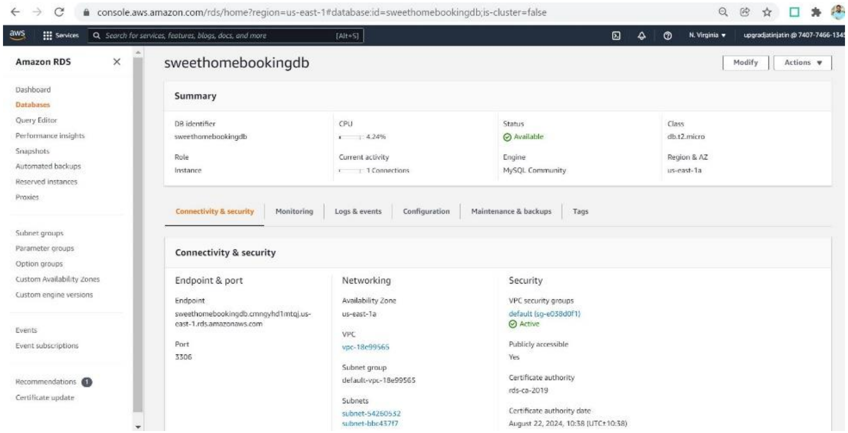
Updated less than a minute ago

Instance ID: i-0f9788c53e4df2e4a (upGrad_kafka)
Public IPv4 address: 18.207.88.18 (upGrad_kafka) | open address
Private IPv4 addresses: 10.0.0.51
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-18-207-88-18.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-10-0-0-51.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-0-51.ec2.internal
Answer private resource DNS name: –
Instance type: t2.medium
Elastic IP addresses: 18.207.88.18 (upGrad_kafka) [Public IP]
VPC ID: vpc-072288c66cb482b58 (upgrad_vpc)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-05f70ba514d56c90e (Public subnet)

Instance details

Platform: Amazon Linux (Inferred)
AMI ID: ami-02e136e904f3da870
Monitoring: disabled
Platform details: Linux/UNIX
AMI name: amzn2-ami-hvm-2.0.20211001.1-x86_64-gp2
Termination protection: Disabled

4. RDS Instance is created with MySQL to host the Databases for the application.



sweethomebookingdb

Summary

DB identifier: sweethomebookingdb
CPU: 4.24%
Status: Available
Class: db.t2.micro
Role: Instance
Current activity: 1 Connections
Engine: MySQL Community
Region & AZ: us-east-1a

Connectivity & security

Endpoint & port

Endpoint: sweethomebookingdb.cmngyhd1mtqj.us-east-1.rds.amazonaws.com
Port: 3306

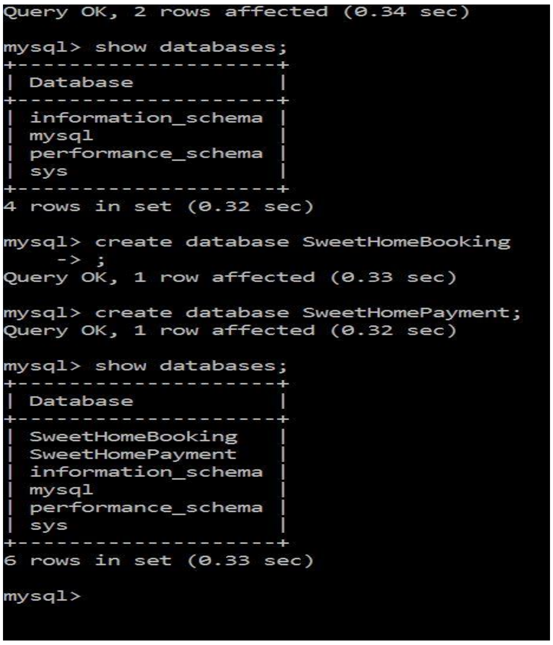
Networking

Availability Zone: us-east-1a
VPC: vpc-18e99565
Subnet group: default-vpc-18e99565
Subnets: subnet-54260532, subnet-bbc437f7

Security

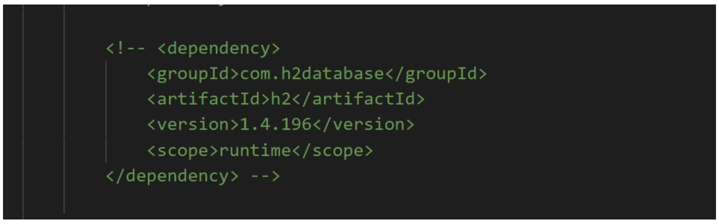
VPC security groups: default (sg-e038d0f1) Active
Publicly accessible: Yes
Certificate authority: rds-ca-2019
Certificate authority date: August 22, 2024, 10:38 (UTC+10:38)

5. Databases **SweetHomeBooking** and **SweetHomePayment** created.



```
Query OK, 2 rows affected (0.34 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.32 sec)

mysql> create database SweetHomeBooking
    -> ;
Query OK, 1 row affected (0.33 sec)

mysql> create database SweetHomePayment;
Query OK, 1 row affected (0.32 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| SweetHomeBooking   |
| SweetHomePayment   |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.33 sec)

mysql>
```

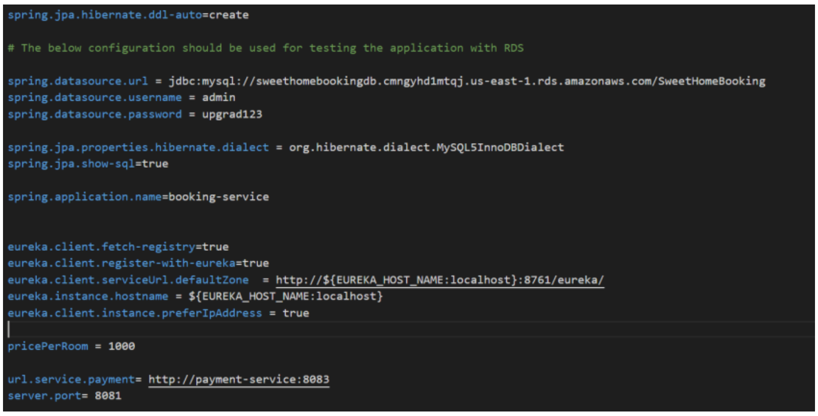
6. Commented the h2 database dependency for both Payment and Booking Service since it is not required.



```
<!-- <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.4.196</version>
    <scope>runtime</scope>
</dependency> -->
```

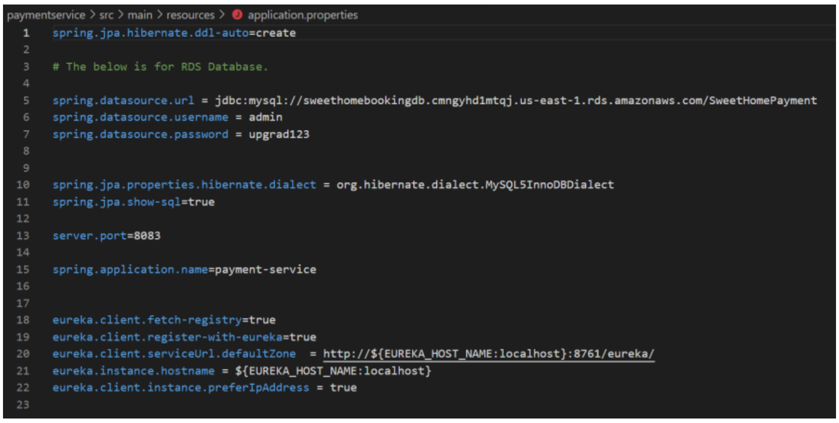
7. Update the application.properties for booking and payment services to add RDS and Eureka Server configurations with others

Booking Service



```
spring.jpa.hibernate.ddl-auto=create

# The below configuration should be used for testing the application with RDS

spring.datasource.url = jdbc:mysql://sweethomebookingdb.cmngyhd1mtqj.us-east-1.rds.amazonaws.com/SweetHomeBooking
spring.datasource.username = admin
spring.datasource.password = upgrad123

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.show-sql=true

spring.application.name=booking-service


eureka.client.fetch-registry=true
eureka.client.register-with-eureka=true
eureka.client.serviceUrl.defaultZone  = http://${EUREKA_HOST_NAME:localhost}:8761/eureka/
eureka.instance.hostname = ${EUREKA_HOST_NAME:localhost}
eureka.client.instance.preferIpAddress = true

pricePerRoom = 1000

url.service.payment= http://payment-service:8083
server.port= 8081
```

Payment Service

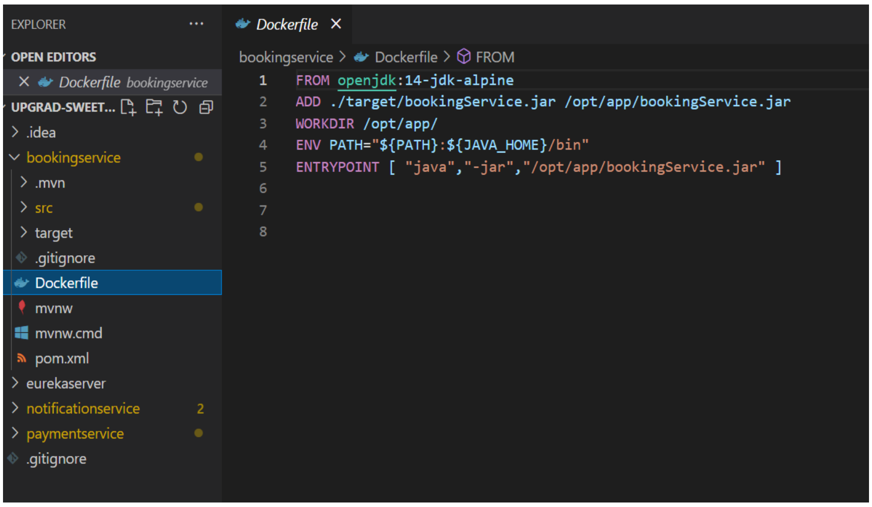


paymentservice > src > main > resources > application.properties

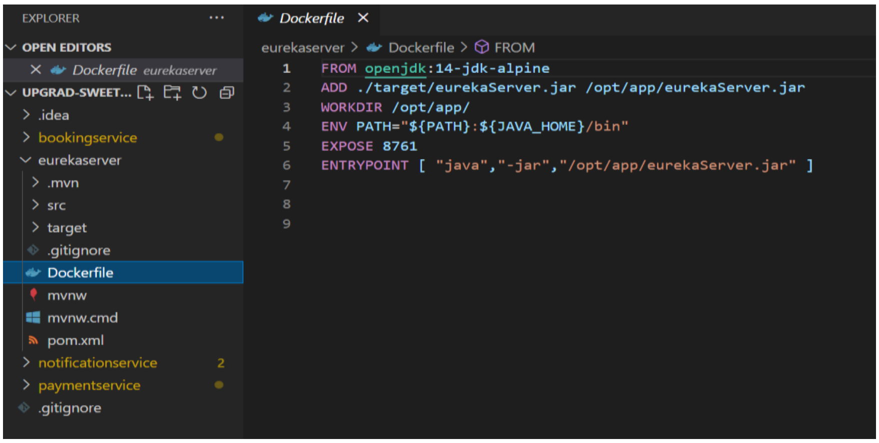
```
spring.jpa.hibernate.ddl-auto=create

# The below is for RDS Database.

spring.datasource.url = jdbc:mysql://sweethomebookingdb.cmngyhd1mtqj.us-east-1.rds.amazonaws.com/SweetHomePayment
spring.datasource.username = admin
spring.datasource.password = upgrad123


spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.show-sql=true

server.port=8083

spring.application.name=payment-service


eureka.client.fetch-registry=true
eureka.client.register-with-eureka=true
eureka.client.serviceUrl.defaultZone  = http://${EUREKA_HOST_NAME:localhost}:8761/eureka/
eureka.instance.hostname = ${EUREKA_HOST_NAME:localhost}
eureka.client.instance.preferIpAddress = true

```

8. Made the required code changes in codestub to add the ip addresses for Kafka server. We are using externalized kafka server here.

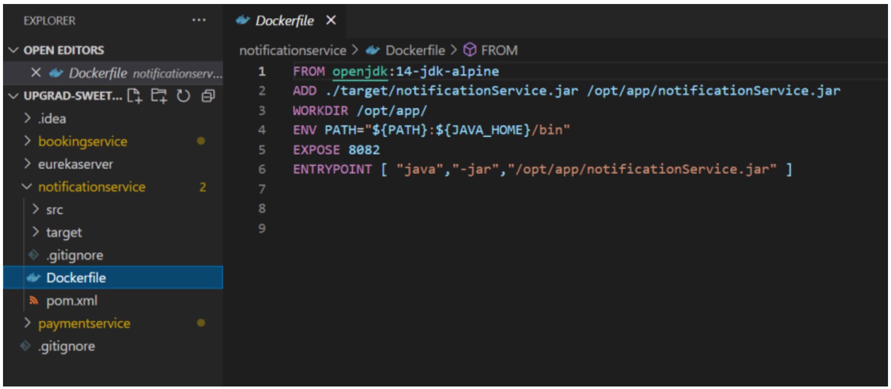
9. Built the Docker file and Docker-Compose file in appropriate directories.



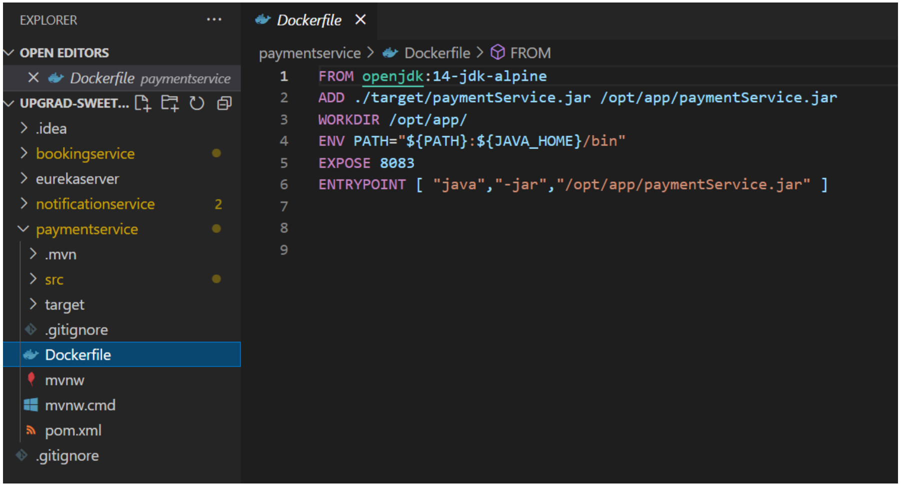
```
FROM openjdk:14-jdk-alpine
ADD ./target/bookingService.jar /opt/app/bookingService.jar
WORKDIR /opt/app/
ENV PATH="${PATH}:${JAVA_HOME}/bin"
ENTRYPOINT [ "java","-jar","/opt/app/bookingService.jar" ]



```



```
FROM openjdk:14-jdk-alpine
ADD ./target/eurekaServer.jar /opt/app/eurekaServer.jar
WORKDIR /opt/app/
ENV PATH="${PATH}:${JAVA_HOME}/bin"
EXPOSE 8761
ENTRYPOINT [ "java","-jar","/opt/app/eurekaServer.jar" ]
```

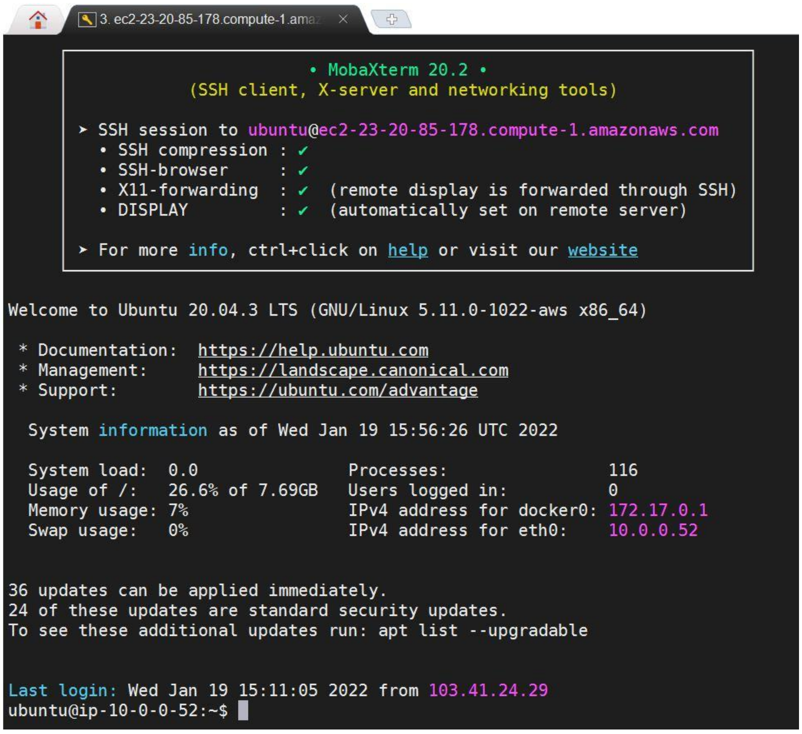


```
FROM openjdk:14-jdk-alpine
ADD ./target/notificationService.jar /opt/app/notificationService.jar
WORKDIR /opt/app/
ENV PATH="${PATH}:${JAVA_HOME}/bin"
EXPOSE 8082
ENTRYPOINT [ "java","-jar","/opt/app/notificationService.jar" ]
```



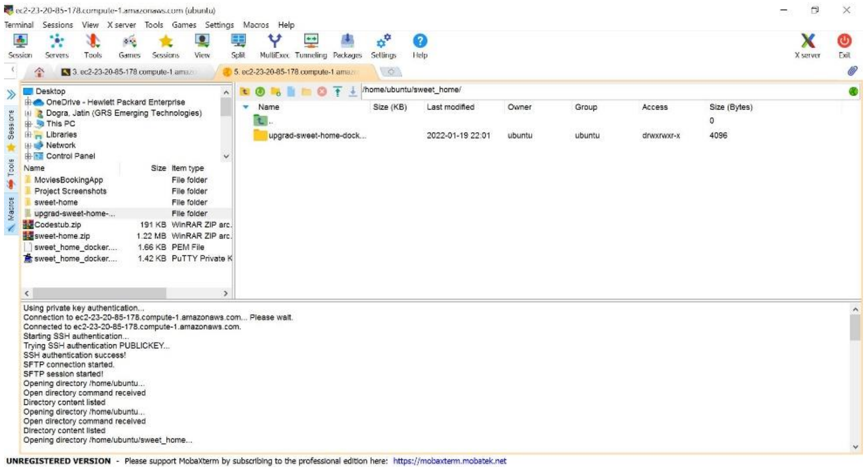
```
FROM openjdk:14-jdk-alpine
ADD ./target/paymentService.jar /opt/app/paymentService.jar
WORKDIR /opt/app/
ENV PATH="${PATH}:${JAVA_HOME}/bin"
EXPOSE 8083
ENTRYPOINT [ "java","-jar","/opt/app/paymentService.jar" ]
```

10. Logged into the EC2 instance



```
              • MobaXterm 20.2 •
       (SSH client, X-server and networking tools)

 ➤ SSH session to ubuntu@ec2-23-20-85-178.compute-1.amazonaws.com
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan 19 15:56:26 UTC 2022

  System load:  0.0                Processes:                116
  Usage of /:   26.6% of 7.69GB    Users logged in:          0
  Memory usage: 7%                 IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                 IPv4 address for eth0:    10.0.0.52


36 updates can be applied immediately.
24 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Wed Jan 19 15:11:05 2022 from 103.41.24.29
ubuntu@ip-10-0-0-52:~$
```

11. Uploaded the code the files to the EC2 instance



```
Name                        Size     Item type
MoviesBookingApp                     File folder
Project Screenshots                  File folder
sweet-home                           File folder
upgrad-sweet-home-...                File folder
Codestub.zip               191 KB    WinRAR ZIP arc.
sweet-home.zip            1.22 MB    WinRAR ZIP arc.
sweet_home_docker....     1.66 KB    PEM File
sweet_home_docker....     1.42 KB    PuTTY Private K
```

```
/home/ubuntu/sweet_home/
Name                      Size (KB)  Last modified     Owner    Group    Access      Size (Bytes)
..                                                                                   0
upgrad-sweet-home-dock...            2022-01-19 22:01  ubuntu   ubuntu   drwxrwxr-x  4096
```

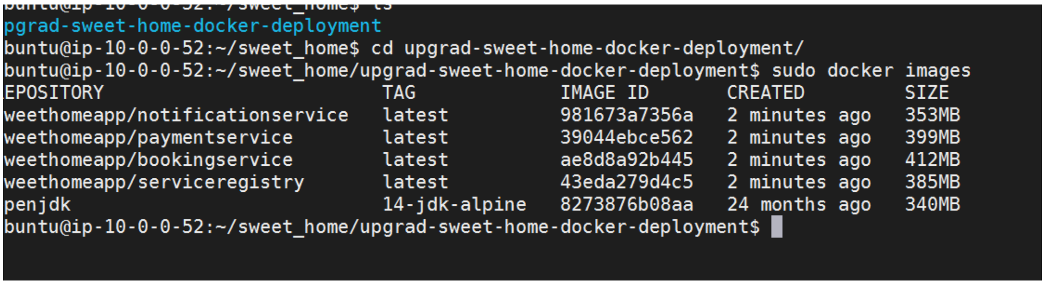
```
Using private key authentication...
Connection to ec2-23-20-85-178.compute-1.amazonaws.com... Please wait.
Connected to ec2-23-20-85-178.compute-1.amazonaws.com.
Starting SSH authentication...
Trying SSH authentication PUBLICKEY...
SSH authentication success!
SFTP connection started.
SFTP session started!
Opening directory /home/ubuntu...
Open directory command received
Directory content listed
Opening directory /home/ubuntu...
Open directory command received
Directory content listed
Opening directory /home/ubuntu/sweet_home...
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

12. Built the Jar file for Every service using command “mvn clean install -DskipTests”

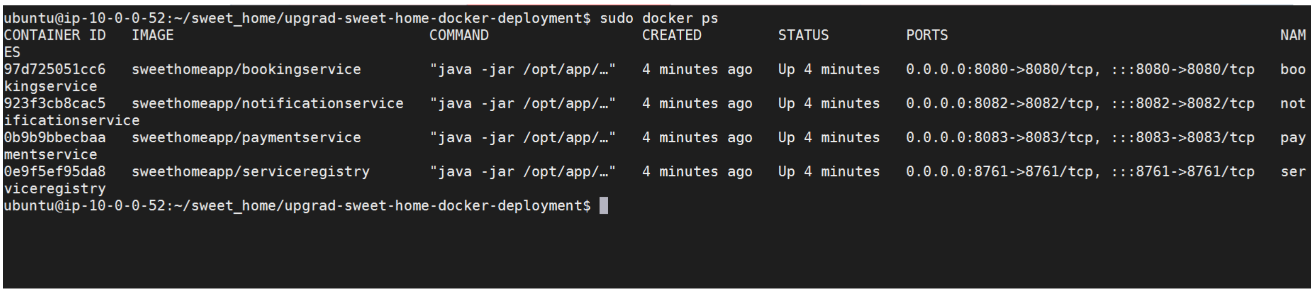
13. The Docker compose file built the docker images and run them for all the services.

Docker Images Created



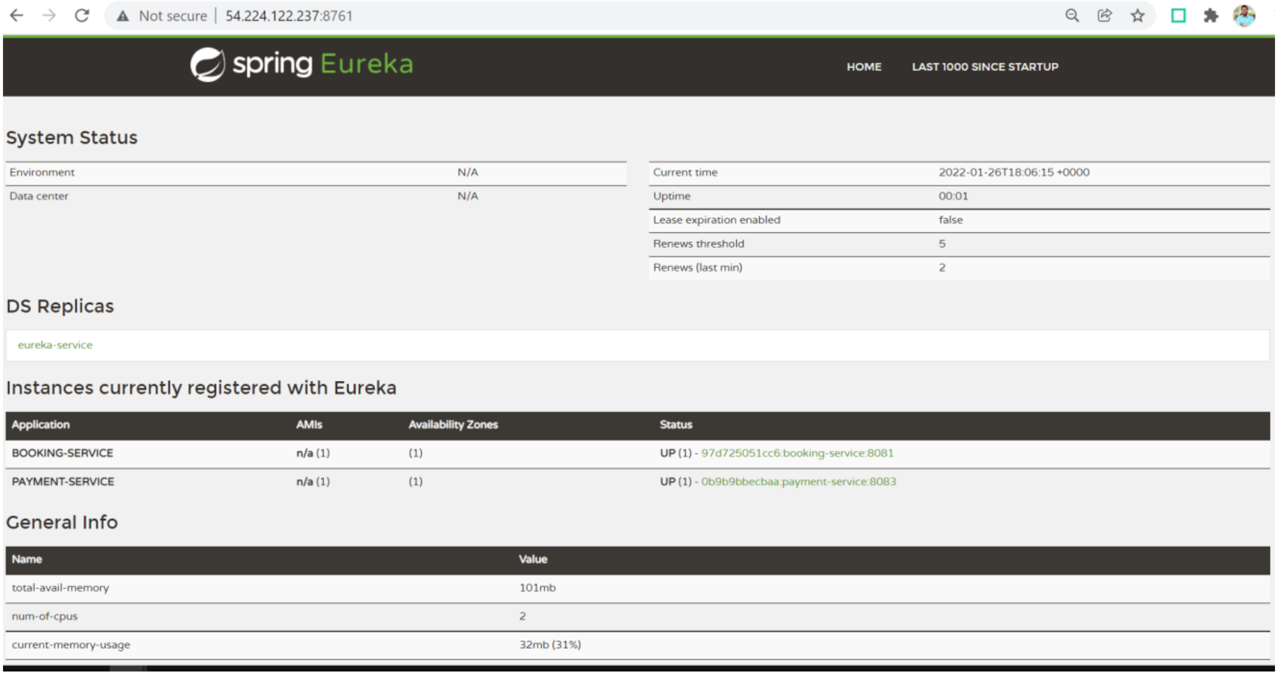
```
buntu@ip-10-0-0-52:~/sweet_home$ ls
pgrad-sweet-home-docker-deployment
buntu@ip-10-0-0-52:~/sweet_home$ cd upgrad-sweet-home-docker-deployment/
buntu@ip-10-0-0-52:~/sweet_home/upgrad-sweet-home-docker-deployment$ sudo docker images
EPOSITORY                          TAG             IMAGE ID       CREATED          SIZE
weethomeapp/notificationservice    latest          981673a7356a   2 minutes ago    353MB
weethomeapp/paymentservice         latest          39044ebce562   2 minutes ago    399MB
weethomeapp/bookingservice         latest          ae8d8a92b445   2 minutes ago    412MB
weethomeapp/serviceregistry        latest          43eda279d4c5   2 minutes ago    385MB
penjdk                             14-jdk-alpine   8273876b08aa   24 months ago    340MB
buntu@ip-10-0-0-52:~/sweet_home/upgrad-sweet-home-docker-deployment$
```

Run the docker containers using with docker-compose which used the created images and verified the running containers using *sudo docker ps*



```
ubuntu@ip-10-0-0-52:~/sweet_home/upgrad-sweet-home-docker-deployment$ sudo docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED         STATUS         PORTS                                       NAM
ES
97d725051cc6   sweethomeapp/bookingservice      "java -jar /opt/app/…"   4 minutes ago   Up 4 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   boo
kingservice
923f3cb8cac5   sweethomeapp/notificationservice "java -jar /opt/app/…"   4 minutes ago   Up 4 minutes   0.0.0.0:8082->8082/tcp, :::8082->8082/tcp   not
ificationservice
0b9b9bbecbaa   sweethomeapp/paymentservice      "java -jar /opt/app/…"   4 minutes ago   Up 4 minutes   0.0.0.0:8083->8083/tcp, :::8083->8083/tcp   pay
mentservice
0e9f5ef95da8   sweethomeapp/serviceregistry     "java -jar /opt/app/…"   4 minutes ago   Up 4 minutes   0.0.0.0:8761->8761/tcp, :::8761->8761/tcp   ser
viceregistry
ubuntu@ip-10-0-0-52:~/sweet_home/upgrad-sweet-home-docker-deployment$ 
```

14. Verifying the Service register.



Not secure | 54.224.122.237:8761

spring Eureka

HOME   LAST 1000 SINCE STARTUP

System Status

Environment	N/A
Data center	N/A

Current time	2022-01-26T18:06:15 +0000
Uptime	00:01
Lease expiration enabled	false
Renews threshold	5
Renews (last min)	2

DS Replicas

eureka-service

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
BOOKING-SERVICE	n/a (1)	(1)	UP (1) - 97d725051cc6:booking-service:8081
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - 0b9b9bbecbaa:payment-service:8083

General Info

Name	Value
total-avail-memory	101mb
num-of-cpus	2
current-memory-usage	32mb (31%)