

Name: Yash Jayeshkumar Pandya  
Email : yashp614929@gmail.com  
Project : Course Project: Application Deployment Using Docker

---

## **Overview**

The project `Book My Consultation` is an application that will register doctors. Users will be able to register themselves and then book appointments based on their requirements and the availability status of the doctors. After consulting the available doctor, users will be able to rate their experience. In this project, I have created a microservices based backend application using spring boot framework. The deployment using docker mainly has the same project code structure and addition of deployment files. Ie. **Dockerfile** and **docker-compose** file.

The project is divided into eight different services, which are as follows:

### Doctor Service

This service defines multiple API endpoints, this service will be used to register the doctors. Once the doctor submits their registration, the BMC employee will verify their details and approve or reject the registration. If the registration is approved, the status of the doctor will change from Pending to Active. If it is rejected, the status will change to Rejected.

### User Service

This service defines multiple API endpoints, this service will be used to register the users on the application. The users can also upload documents related to their past medical history.

### Appointment Service

This service defines multiple API endpoints, this service will be used to schedule the availability of doctors and book an appointment. The users can also upload documents related to their past medical history.

### Payment Service

This service defines single API endpoints, this service will be used to make the payment, and the status of the appointment will change from PendingPayment to Confirmed.

### Rating Service

This service defines single API endpoints, this service will be used to rate the user's experience with the doctor.

## Notification Service

This service will listen to the Kafka topics to which the other services will send the messages. This service will be responsible for sending emails for the following events:

- Doctor's approval
- Doctors' rejection
- Appointment confirmation
- Prescription

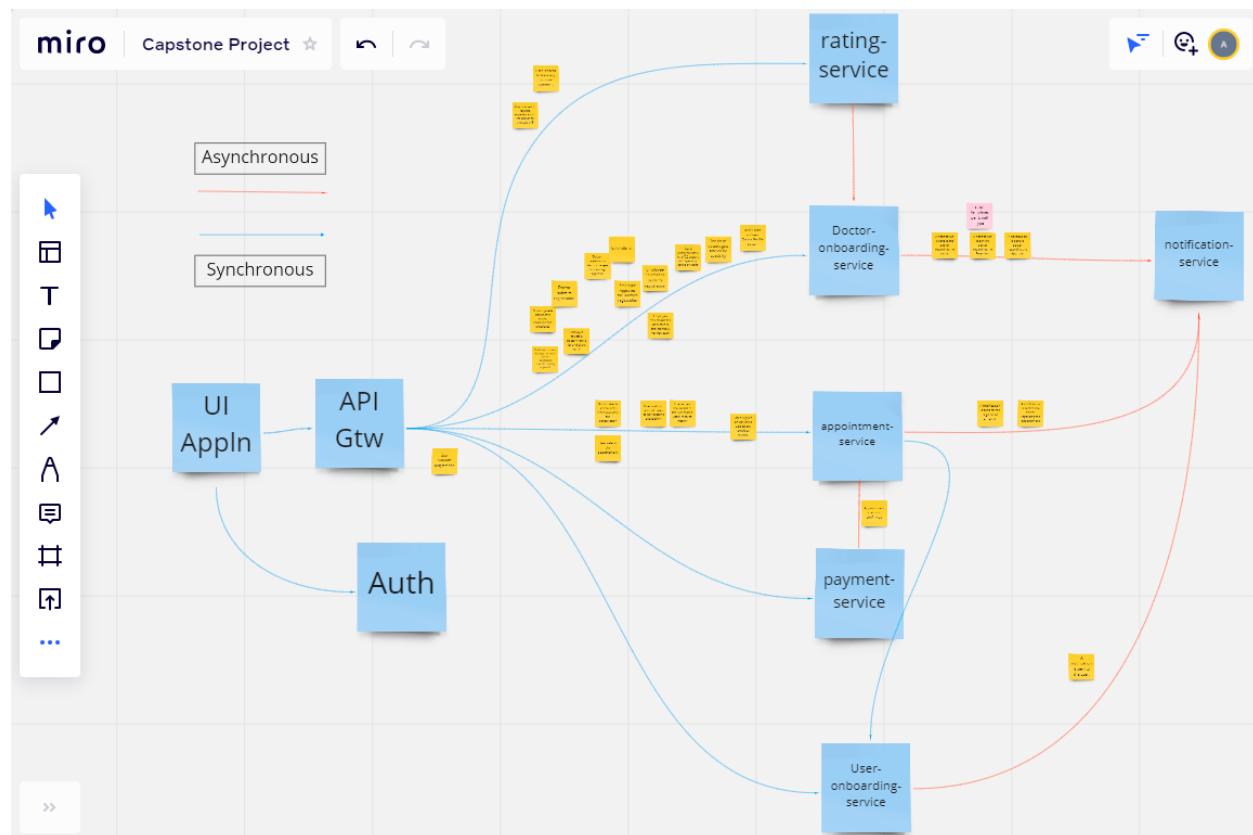
## Security Provider

This service defines an endpoint which is used to generate the token and will be used to generate the JWT token. The token validation should be part of each microservice

## BMC Gateway

This service is used to hide the actual API URLs from the UI application. The API gateway will intercept all the requests and route them to the correct service.

## Application Workflow



## Solution Architecture:

All Services are combined into a single directory called '**BookMyConsultation**'. All the defined services are a type of spring boot application.

Deployment cycle contains the following resources.

- AWS VPC
- AWS RDS
- AWS EC2
- AWS Elastic IPs
- AWS Security Groups
- Apache kafka
- AWS S3
- AWS SES
- MySQLWorkbench
- Postman

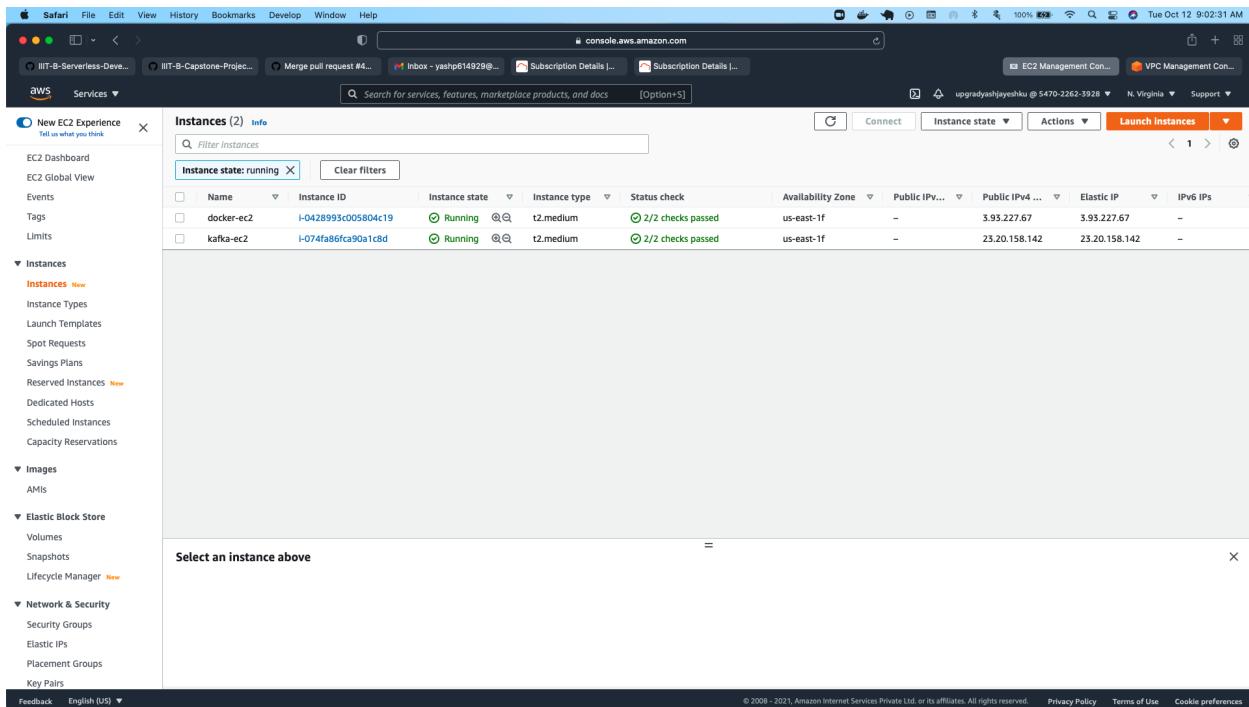
Solution approach is to **use a separate EC2 instance to install the apache kafka and do not containerize the kafka using the docker-compose**. Ie. **externalized kafka**. The connection to kafka is done using the public IP of the EC2 instance which is associated with the Elastic IP.

Firstly, a new VPC is created to launch resources that are being used while setting up the project.

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with navigation links like 'VPC Dashboard', 'Route Tables', 'Internet Gateways', 'Carrier Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Managed Prefix Lists', 'Endpoints', 'Endpoint Services', 'NAT Gateways', 'Peering Connections', 'SECURITY', 'REACHABILITY', and 'DNS FIREWALL'. The main area has a header with tabs for 'Your VPCs (1/1)', 'Actions', and 'Create VPC'. Below this, a table lists one VPC named 'docker-vpc' with details: VPC ID 'vpc-0e5bc6c0c634a8fc6', State 'Available', IPv4 CIDR '10.0.0.0/16', and DHCP options set 'dopt-67e6bd1d'. A 'Details' tab is open, showing more information: VPC ID 'vpc-0e5bc6c0c634a8fc6', State 'Available', Tenant 'Default', Default VPC 'No', Route 53 Resolver DNS Firewall rule groups 'None', and DNS hostnames 'Disabled'. It also shows the Main route table 'rtb-0d284460f7fab85bd', IPv4 CIDR '10.0.0.0/16', Owner ID '547022623928', and DNS resolution 'Enabled'. The Main network ACL 'acl-0174ea6d9171dce15' and IPv6 CIDR 'Network border group' are listed as '-'.

Two EC2 instances are created with the type t2.medium.

- **docker-ec2** instance is to deploy the application code using docker.
- **Kafka-ec2** instance is to install apache kafka for asynchronous communication for microservices.



The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences. The main content area displays a table of instances:

Instance state: running	Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4	Public IPv6	Elastic IP	IPv6 IPs
	docker-ec2	i-0428993c005804c19	Running	t2.medium	2/2 checks passed	us-east-1f	-	3.93.227.67	3.93.227.67	-
	kafka-ec2	i-074fa86fc90a1c8d	Running	t2.medium	2/2 checks passed	us-east-1f	-	23.20.158.142	23.20.158.142	-

A modal window titled "Select an instance above" is open at the bottom of the page.

New security group **docker-sg** is created and below inbound rules are added under this security group. This security group is associated with the **docker-ec2** where the application code is being deployed.

Note that all inbound rules are restricted to allow access only within developer/student's IP.

### Ports

- **22** is open to gain access to **docker-ec2** instances.
- **80** is open for all HTTP requests.
- **443** is open for all HTTPS requests.
- **27017** is open for all MongoDB.
- **8761** is open to access the Service Registry(Eureka).
- **9191** is open to access api gateway (BMC-GATEWAY).
- **8081** is open to access doctor service (DOCTOR-SERVICE).
- **8082** is open to access payment service (APPOINTMENT-SERVICE).
- **8083** is open to access user service (USER-SERVICE).

- **8084** is open to access rating service (RATING-SERVICE).
- **8085** is open to access payment service (NOTIFICATION-SERVICE).
- **8086** is open to access payment service (PAYMENT-SERVICE).
- **8088** is open to access payment service (SECURITY-PROVIDER).

Security Groups (1/1) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
<input checked="" type="checkbox"/> docker-sg	sg-00aa99502b68c324a	docker-sg	vpc-0e5bc6c0c634a8fc6	docker-sg	547022623928	12 Permission entries	1 Permission entry

Inbound rules (12)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0294df197cfbdcc	IPv4	HTTPS	TCP	443	173.177.134.208/32	–
–	sgr-07a5e76bdac6f088d	IPv4	Custom TCP	TCP	8084	173.177.134.208/32	–
–	sgr-0027b92b33aa0d93d	IPv4	SSH	TCP	22	173.177.134.208/32	–
–	sgr-0f0d347b5680386...	IPv4	Custom TCP	TCP	8086	173.177.134.208/32	–
–	sgr-07eddd39931323...	IPv4	Custom TCP	TCP	8085	173.177.134.208/32	–
–	sgr-0f73fa47d4e9459bb	IPv4	Custom TCP	TCP	8088	173.177.134.208/32	–
–	sgr-04ba225051fb7f19	IPv4	Custom TCP	TCP	8082	173.177.134.208/32	–
–	sgr-0577b9e92614f34	IPv4	Custom TCP	TCP	8083	173.177.134.208/32	–
–	sgr-009098bd2b4e2d...	IPv4	HTTP	TCP	80	173.177.134.208/32	–
–	sgr-05e80ca7fbc389	IPv4	Custom TCP	TCP	8081	173.177.134.208/32	–
–	sgr-02dec69b164f09b5	IPv4	Custom TCP	TCP	8761	173.177.134.208/32	–
–	sgr-03c9f4ca67d0293d	IPv4	Custom TCP	TCP	9191	173.177.134.208/32	–

Another security group **kafka-sg** is created and below inbound rules are added under this security group. This security group is associated with the **kafka-ec2** where the apache kafka is being deployed.

Security Groups (1/1) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
<input checked="" type="checkbox"/> kafka-sg	sg-0e4fd5f569526457	kafka-sg	vpc-0e5bc6c0c634a8fc6	kafka-sg	547022623928	9 Permission entries	1 Permission entry

sg-0e4fd5f569526457 - kafka-sg

Details Inbound rules Outbound rules Tags

Inbound rules (9)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0009f50806df59a30	IPv4	SSH	TCP	22	173.177.134.208/32	–
–	sgr-0bd31332166976...	IPv4	Custom TCP	TCP	2181	0.0.0.0/0	–
–	sgr-0d25f6a69ed6875ad	IPv4	Custom TCP	TCP	9092	0.0.0.0/0	–
–	sgr-0697a230061702...	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
–	sgr-0c16a6c5fac86990f	IPv6	Custom TCP	TCP	2181	::/0	–
–	sgr-0ef2f4096eb142b9	IPv6	Custom TCP	TCP	9092	::/0	–
–	sgr-021f2fa0087943a1d	IPv6	HTTP	TCP	80	::/0	–
–	sgr-00be86a7f6c6fed4e	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-0761c1b45550a37...	IPv6	HTTPS	TCP	443	::/0	–

Note that all inbound rules are publicly opened.

## Ports

- **22** is open to gain access to **kafka-ec2** instance.
- **80** is open for all HTTP requests.
- **443** is open for all HTTPS requests.
- **2181** is open for apache kafka.
- **9092** is open for netcat.

## EC2 instance connection

The single pem file is used to connect with both the ec2 instances. docker-ec2 and kafka-ec2.

```
ssh -i "docker-dep.pem" ubuntu@3.93.227.67  
ssh -i "docker-dep.pem" ubuntu@23.20.158.142
```

The public IPs are used to connect with EC2 instances using the ssh command. However, the IP gets changed when the EC2 instance is stopped. In-order to make a common IP until the project deployment is completed, Elastic IPs are used. Each EC2 instance is associated with the elastic IP.

Name	Allocated IPv4 address	Type	Allocation ID	Associated instance ID	Private IP address	Association ID
kafka-ec2-ip	23.20.158.142	Public IP	eipalloc-0000947ec064efd86	i-074fa86fca90a1c8d	10.0.220.157	eipassoc-0f71920380492c56f
docker-ec2-ip	3.93.227.67	Public IP	eipalloc-047f9c9c911c5adbc	i-0428993c005804c19	10.0.197.192	eipassoc-06dbfe7d5dddb85

## Docker and Docker-Compose Installation

docker is installed on the **docker-ec2** instance using the docker installation process and verifies the installation using the below command.

```
which docker  
docker -v
```

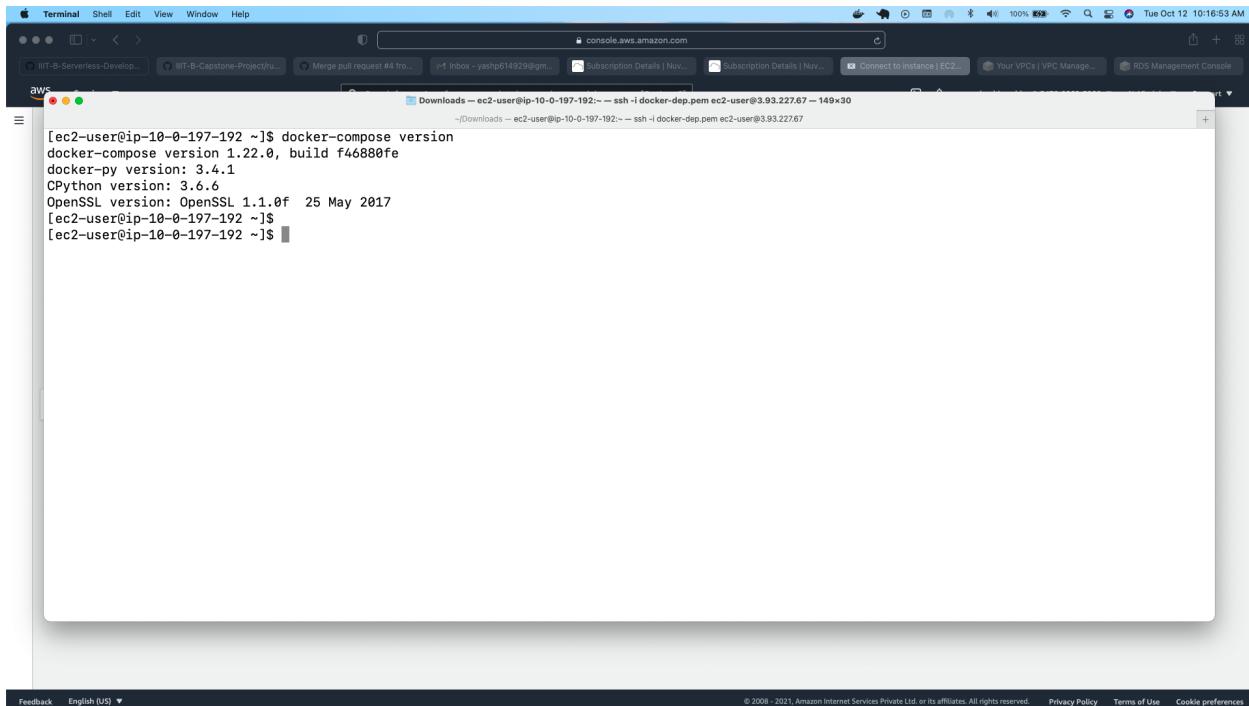


A screenshot of an AWS CloudShell terminal window titled "aws". The terminal shows the following command execution:

```
[ec2-user@ip-10-0-197-192 ~]$ which docker  
/usr/bin/docker  
[ec2-user@ip-10-0-197-192 ~]$ docker -v  
Docker version 20.10.7, build f0df350  
[ec2-user@ip-10-0-197-192 ~]$
```

docker-compose is installed on the **docker-ec2** instance using the docker-compose installation process and verifies the installation using the below command.

```
docker-compose --version
```



A screenshot of an AWS CloudShell terminal window titled "aws". The terminal shows the following command execution:

```
[ec2-user@ip-10-0-197-192 ~]$ docker-compose version  
docker-compose version 1.22.0, build f46880fe  
docker-py version: 3.4.1  
CPython version: 3.6.6  
OpenSSL version: OpenSSL 1.1.0f  25 May 2017  
[ec2-user@ip-10-0-197-192 ~]$  
[ec2-user@ip-10-0-197-192 ~]$
```

## RDS

The RDS database instance named **docker-app-rds** is created using the MySQL Engine. The RDS is created within the **docker-vpc**.

The screenshot shows the AWS RDS Management Console. On the left, a sidebar lists various RDS management options like Dashboard, Databases, Query Editor, and Performance Insights. The main area displays the 'Summary' tab for the 'docker-app-rds' database. Key details include:

DB Identifier	CPU	Status	Class
docker-app-rds	5.50%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1f

Below the summary, the 'Connectivity & security' tab is selected. It shows the endpoint and port (3306) and the associated VPC (docker-vpc). The security group 'docker-rds-sg' is listed under 'VPC security groups'.

The security group **docker-rds-sg** is created with below inbound rules and associated with **docker-app-rds** RDS instance.

## Ports

- **3306** is open to connect RDS from developer/student's IP.
- **3306** is open to connect from the docker-sg security group.

The screenshot shows the AWS EC2 Management Console. The left sidebar lists EC2 resources like Instances, Images, and Elastic Block Store. The main area shows the 'Inbound rules' tab for the 'sg-05edecab44c642637 - docker-rds-sg' security group. There are three inbound rules:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
sgr-0fbfb8f7fe15df695	IPv4	MySQL/Aurora	TCP	3306	173.177.134.208/32	-	
sgr-073fad2d596f6fdbf	-	MySQL/Aurora	TCP	3306	sg-00aa9950b68c32...	-	
sgr-0d8abe487f57f1540	-	MySQL/Aurora	TCP	3306	sg-0e4fdd5f56952645...	-	

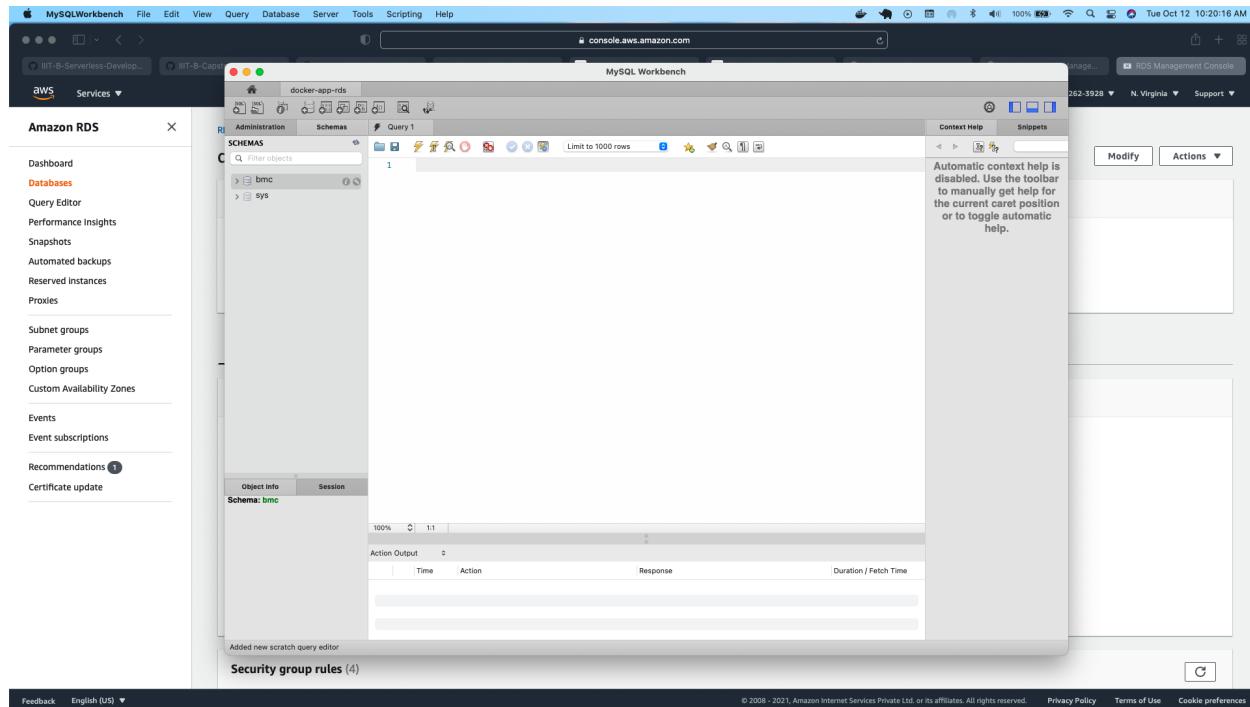
Appointment service uses the below database for availability and appointment information storage.

- bmc

All the services use the below database to validate authorization tokens (JWT).

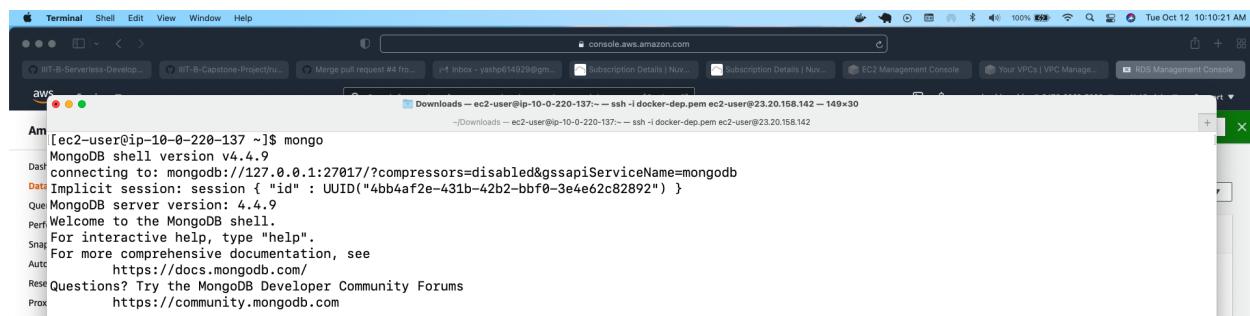
- bmc

RDS is created with username as admin and password as upgrad123



## MONGO DB

The mongo database is also installed and running on **kafka-ec2** instance.



## Kafka Installation

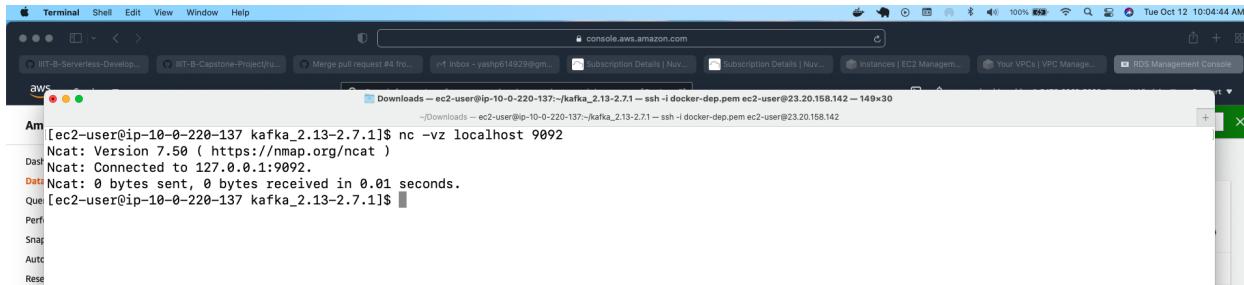
```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

As described above **kafka-ec2** instance is used to install the apache kafka and run the kafka asynchronous communication for the notification service

Kafka and Netcat are installed using the installation process. The properties file is configured with the elastic IP of **kafka-ec2** instance. Both kafka and zookeeper are running and verified using netcat as shown below.



```
[ec2-user@ip-10-0-220-137 kafka_2.13-2.7.1]$ nc -vz localhost 2181
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:2181.
Ncat: 0 bytes sent, 0 bytes received in 0.01 seconds.
[ec2-user@ip-10-0-220-137 kafka_2.13-2.7.1]$
```



```
[ec2-user@ip-10-0-220-137 kafka_2.13-2.7.1]$ nc -vz localhost 9092
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:9092.
Ncat: 0 bytes sent, 0 bytes received in 0.01 seconds.
[ec2-user@ip-10-0-220-137 kafka_2.13-2.7.1]$
```

## Application Deployment and Run instructions

The application code including the Dockerfiles and docker-compose file are grouped together and pushed to the github private repository. The codebase is pulled on to the docker-ec2 instance using the below command.

```
git clone
https://github.com/ypandy614929/IIIT-B-Capstone-Project.git
```

Please note that this is the private repository created under the developer/student's github account in order to simplify the process.

There are no docker images present initially and there are no containers running on the docker-ec2 instance. Both checks are verified using below commands.

**sudo docker images**



```
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$
```

**sudo docker ps -a**



```
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$
```

## Appointment-Service Image

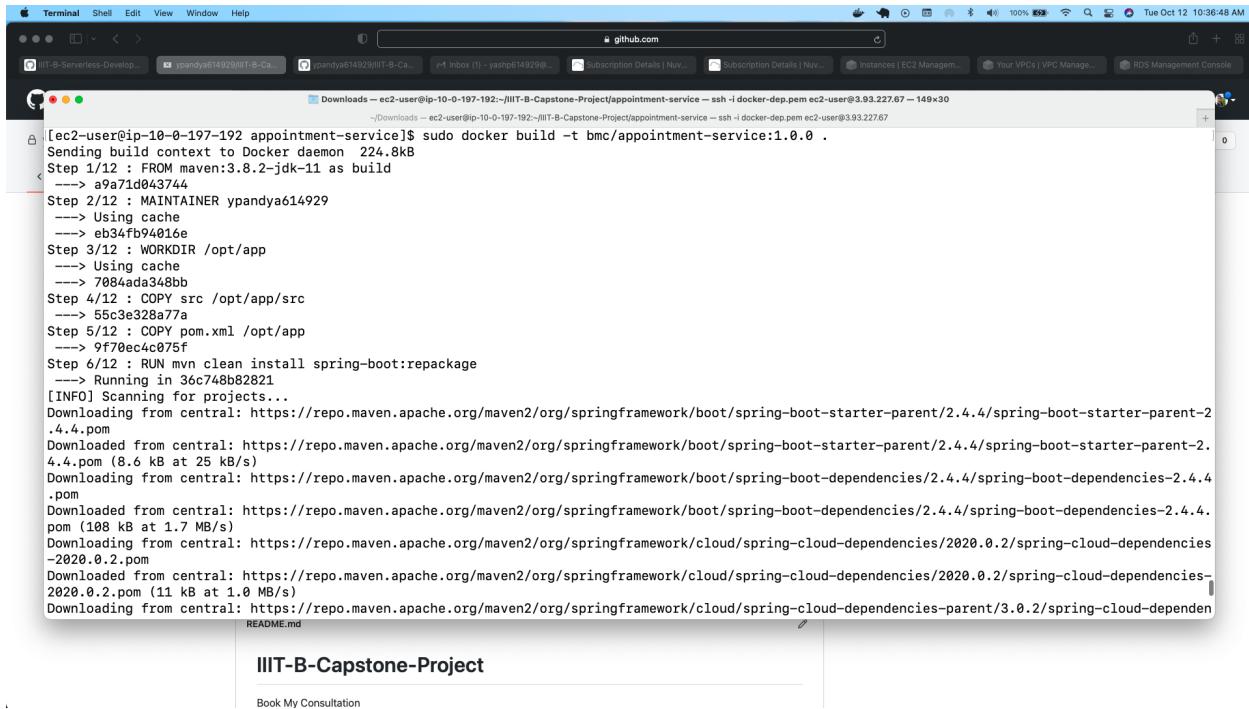
The below commands are present in the Appointment service Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/appointment-service.jar
/opt/app/appointment-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/appointment-service.jar"]
```

The docker image of the appointment service is built using the following command.

```
sudo docker build -t bmc/appointment-service:1.0.0 .
```



A screenshot of a Mac OS X desktop showing a Terminal window. The terminal is running on an EC2 instance, as indicated by the prompt [ec2-user@ip-10-0-197-192 ~] and the background AWS CloudWatch logs. The user is executing a Docker build command for the 'appointment-service' image. The build process shows various steps: sending build context to Docker daemon, using cache, setting WORKDIR to /opt/app, copying source code and pom.xml, and finally running mvn clean install spring-boot:repackage. Maven dependencies are being downloaded from central repositories. The terminal window is part of a larger desktop environment with other windows like a browser and AWS management tools visible in the background.

```
[ec2-user@ip-10-0-197-192 ~] $ sudo docker build -t bmc/appointment-service:1.0.0 .
Sending build context to Docker daemon 224.8kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d043744
Step 2/12 : MAINTAINER ypandyaa614929
--> Using cache
--> eb34fb94016e
Step 3/12 : WORKDIR /opt/app
--> Using cache
--> 70844ada348bb
Step 4/12 : COPY src /opt/app/src
--> 55c3e328a77a
Step 5/12 : COPY pom.xml /opt/app
--> 9f70ec4c875f
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in 36c748b82821
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom (8.6 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom (108 kB at 1.7 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 1.0 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependencies-parent-3.0.2.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.022 s
[INFO] Finished at: 2023-10-12T10:36:48+00:00
[INFO] Final Memory: 14M/24M
[INFO] ------------------------------------------------------------------------

```

## Bmc-Gateway Image

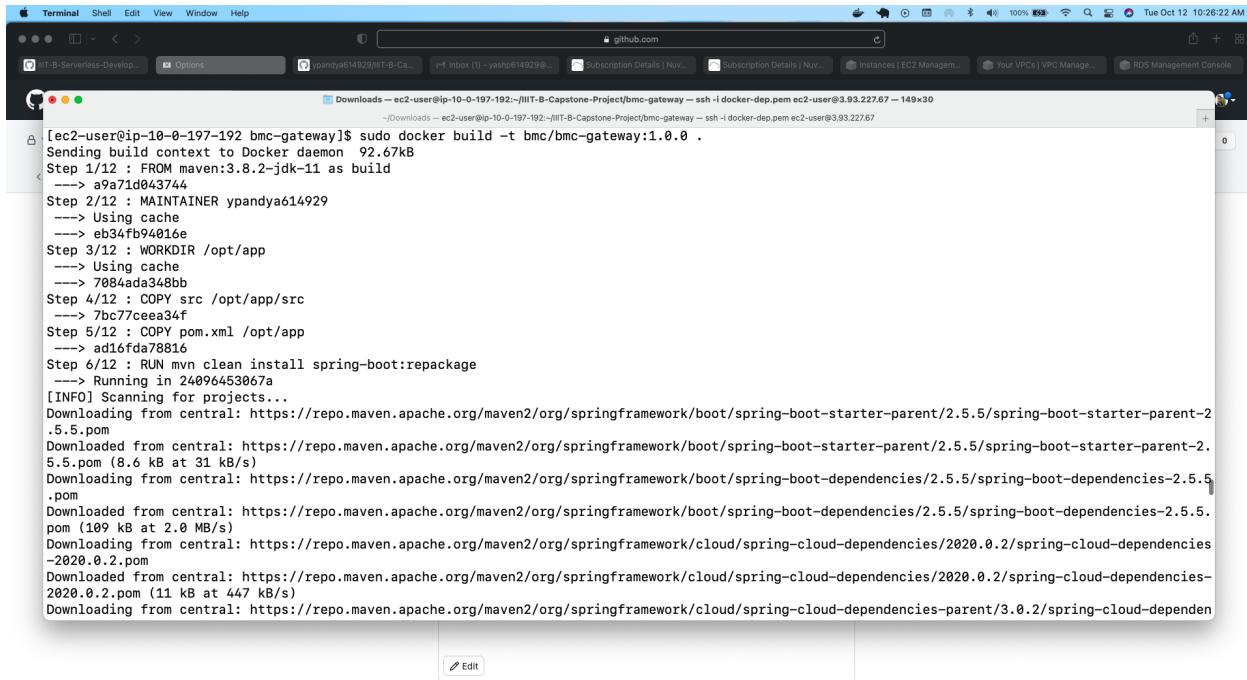
The below commands are present in the bmc-gateway Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/bmc-gateway.jar
/opt/app/bmc-gateway.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/bmc-gateway.jar" ]
```

The docker image of the bmc-gateway is built using the following command.

```
sudo docker build -t bmc/bmc-gateway:1.0.0 .
```



A screenshot of a macOS Terminal window. The title bar shows "Terminal" and the date "Tue Oct 12 10:26:22 AM". The window contains the command "sudo docker build -t bmc/bmc-gateway:1.0.0 ." followed by the Docker build logs. The logs show the process of building the image, including Maven dependencies and Spring Boot repackage steps. The terminal window has a dark theme and is running on an Apple Silicon Mac.

```
[ec2-user@ip-10-0-197-192 bmc-gateway]$ sudo docker build -t bmc/bmc-gateway:1.0.0 .
Sending build context to Docker daemon 92.67kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d043744
Step 2/12 : MAINTAINER ypandyaya614929
--> Using cache
--> eb34fb94016e
Step 3/12 : WORKDIR /opt/app
--> Using cache
--> 7084ada348bb
Step 4/12 : COPY src /opt/app/src
--> 7bc77cea34f
Step 5/12 : COPY pom.xml /opt/app
--> ad16fd78816
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in 24096453067a
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.5.5/spring-boot-starter-parent-2.5.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.5.5/spring-boot-starter-parent-2.5.5.pom (8.6 kB at 31 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.5.5/spring-boot-dependencies-2.5.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.5.5/spring-boot-dependencies-2.5.5.pom (109 kB at 2.0 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 447 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependen
```

## Doctor-Service Image

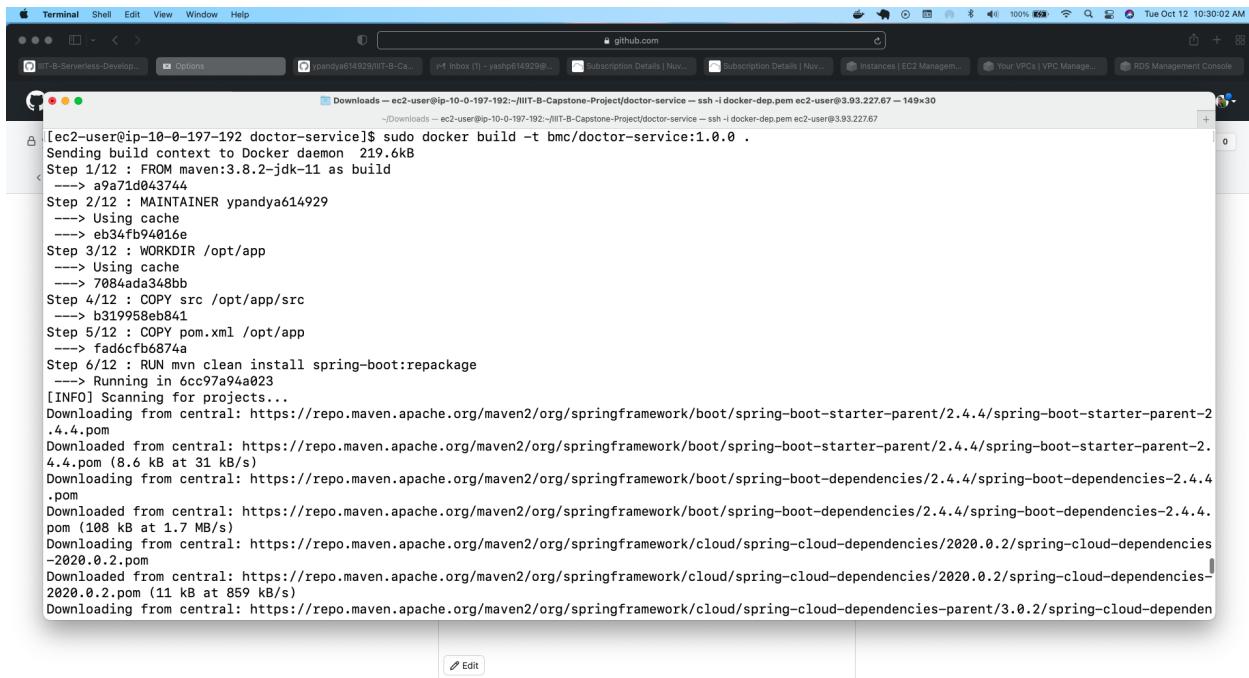
The below commands are present in the Doctor service Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaya614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaya614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/doctor-service.jar
/opt/app/doctor-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/doctor-service.jar"]
```

The docker image of the doctor service is built using the following command.

```
sudo docker build -t bmc/doctor-service:1.0.0 .
```



```
[ec2-user@ip-10-0-197-192 doctor-service]$ sudo docker build -t bmc/doctor-service:1.0.0 .
Sending build context to Docker daemon 219.6kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d0e43744
Step 2/12 : MAINTAINER ypandyaya614929
--> Using cache
--> eb34fb94916e
Step 3/12 : WORKDIR /opt/app
--> Using cache
--> 70884ada348bb
Step 4/12 : COPY src /opt/app/src
--> b319958eb841
Step 5/12 : COPY pom.xml /opt/app
--> fad6cfb6874a
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in 6cc97a94a023
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom (8.6 kB at 31 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom (108 kB at 1.7 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 859 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependen
```

## Notification-Service Image

The below commands are present in the Notification service Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaya614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaya614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/notification-service.jar
/opt/app/notification-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/notification-service.jar"]
```

The docker image of the notification service is built using the following command.

```
sudo docker build -t bmc/notification-service:1.0.0 .
```

```
[ec2-user@ip-10-0-197-192 notification-service]$ sudo docker build -t bmc/notification-service:1.0.0 .
Sending build context to Docker daemon 113.7kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d843744
Step 2/12 : MAINTAINER ypandyaa614929
--> Using cache
--> eb34fb94b16e
Step 3/12 : WORKDIR /opt/app
--> Using cache
--> 7084ada348bb
Step 4/12 : COPY src /opt/app/src
--> 01ad03b3bda2
Step 5/12 : COPY pom.xml /opt/app
--> c0309f0e1742
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in c71325b8332e
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom (8.6 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom (108 kB at 1.7 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 797 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependencies-parent-3.0.2.pom
```

## Payment-Service Image

The below commands are present in the Payment service Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/payment-service.jar
/opt/app/payment-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/payment-service.jar"]
```

The docker image of the payment service is built using the following command.

```
sudo docker build -t bmc/payment-service:1.0.0 .
```

```

[ec2-user@ip-10-0-197-192 payment-service]$ sudo docker build -t bmc/payment-service:1.0.0 .
Sending build context to Docker daemon 171.5KB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
Event: --> a9a71d043744
Tag: Step 2/12 : MAINTAINER ypandyay614929
Line: ----> Using cache
--> eb34fb94016e
Instruct: Step 3/12 : WORKDIR /opt/app
Instruct: ----> Using cache
Instruct: ----> 7084ada348bb
Last: Step 4/12 : COPY src /opt/app/src
Step: ----> 448ba0ee6563
Step: Step 5/12 : COPY pom.xml /opt/app
Save: ----> e1e108b70749
Result: Step 6/12 : RUN mvn clean install spring-boot:repackage
Detail: ----> Running in bddc8b9ef656
[INFO] Scanning for projects...
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom (8.6 kB at 24 kB/s)
AM: Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
ELAS: Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
Vol: pom (108 kB at 1.9 MB/s)
Sni: Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Lifi: Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 930 kB/s)
Net: 2020.0.2.pom (11 kB at 930 kB/s)

Security Groups
Elastic IPs
Placement Groups
Key Pairs
```

## Rating-Service Image

The below commands are present in the Rating service Dockerfile.

```

# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/rating-service.jar
/opt/app/rating-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/rating-service.jar"]
```

The docker image of the rating service is built using the following command.

```
sudo docker build -t bmc/rating-service:1.0.0 .
```

```
[ec2-user@ip-10-0-197-192 rating-service]$ sudo docker build -t bmc/rating-service:1.0.0 .
Sending build context to Docker daemon 169.5kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
3.8.2-jdk-11: Pulling from library/maven
Digest: sha256:9d2ff1b652dd5e427fc2686e11bd832d4d8f7c412632ff5631d48d58ed34d14db
Status: Downloaded newer image for maven:3.8.2-jdk-11
--> a9a71d043744
Step 2/12 : MAINTAINER ypandyay614929
--> Running in 78004d4f5a06
Removing intermediate container 78004d4f5a06
--> 4b7cac0fc57e
Step 3/12 : WORKDIR /opt/app
--> Running in 0dc7c202a5b4
Removing intermediate container 0dc7c202a5b4
--> 1b9d80b0d0103
Step 4/12 : COPY src /opt/app/src
--> 5baaa723949d
Step 5/12 : COPY pom.xml /opt/app
--> e207e780674b
Step 6/12 : RUN mvn clean install spring-boot:repackage
Security Groups
  □ - sgr-0027b92b33aad9f3d IPv4 SSH TCP 22 173.177.134.208/32 -
  □ - sgr-03e080ca7fbcb2389 IPv4 Custom TCP TCP 8081 0.0.0.0/0 -
  □ - sgr-02dec59b164f09b5 IPv4 Custom TCP TCP 8761 0.0.0.0/0 -
Feedback English (US) ▾ © 2008–2021, Amazon Web Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences
```

## Security-Provider Image

The below commands are present in the security provider server Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/security-provider.jar
/opt/app/security-provider.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/security-provider.jar"]
```

The docker image of the security provider is built using the following command.

```
sudo docker build -t bmc/security-provider:1.0.0 .
```

A screenshot of a Mac OS X terminal window titled "Terminal". The window shows a command-line session on an AWS EC2 instance. The user is building a Docker image named "bmc/security-provider:1.0.0" from the current directory. The session includes Maven dependency resolution and several Docker build steps. The terminal output is as follows:

```
[ec2-user@ip-10-0-197-192 security-provider]$ sudo docker build -t bmc/security-provider:1.0.0 .
Sending build context to Docker daemon 126.5kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d843744
Step 2/12 : MAINTAINER ypandyaa614929
--> Using cache
Step 3/12 : WORKDIR /opt/app
--> Using cache
Step 4/12 : COPY src /opt/app/src
--> 9018a436cae
Step 5/12 : COPY pom.xml /opt/app
--> 57057c19fab8
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in 204b7d7e0b9b
[INFO] Scanning for projects...
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.5.5/spring-boot-starter-parent-2.5.5.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.5.5/spring-boot-starter-parent-2.5.5.pom (8.6 kB at 32 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.5.5/spring-boot-dependencies-2.5.5.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.5.5/spring-boot-dependencies-2.5.5.pom (109 kB at 1.8 MB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 657 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependencies-parent-3.0.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependencies-parent-3.0.2.pom (11 kB at 657 kB/s)
[INFO] 
[INFO] BUILD SUCCESS
[INFO] Total time: 1.094 s
[INFO] Finished at: 2021-10-12T10:45:40+00:00
[INFO] 
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ security-provider ---
[INFO] Deleting /opt/app/target
[INFO] 
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ security-provider ---
[INFO] Installing /opt/app/target/service-provider.jar to /root/.m2/repository/bmc/security-provider/1.0.0/service-provider-1.0.0.jar
[INFO] Installing /opt/app/pom.xml to /root/.m2/repository/bmc/security-provider/1.0.0/service-provider-1.0.0.pom
[INFO] 
[INFO] --- maven-repository-plugin:1.0:repository (default) @ security-provider ---
[INFO] 
[INFO] --- maven-jar-plugin:3.1.0:jar (default-jar) @ security-provider ---
[INFO] Building jar: /opt/app/target/service-provider.jar
[INFO] 
[INFO] --- maven-repository-plugin:1.0:repository (default) @ security-provider ---
```

## Service-Registry Image

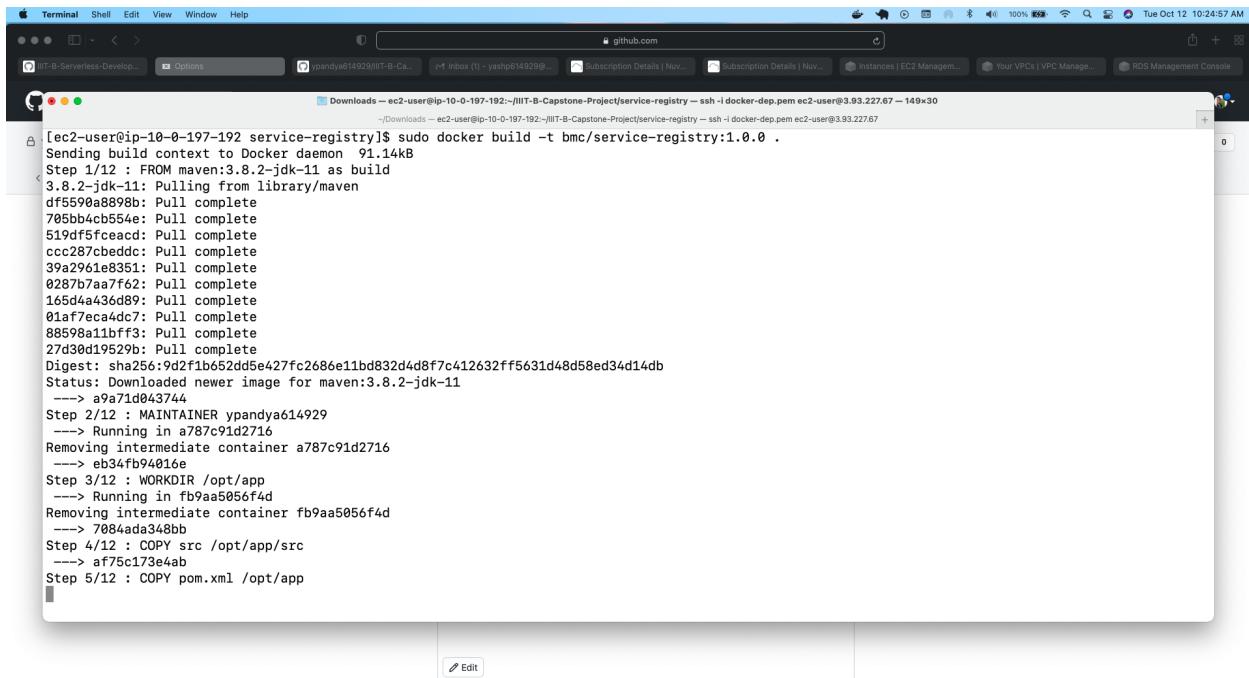
The below commands are present in the service registry server Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyaa614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/service-registry.jar
/opt/app/service-registry.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/service-registry.jar"]
```

The docker image of the service registry is built using the following command.

```
sudo docker build -t bmc/service-registry:1.0.0 .
```



A screenshot of a Mac OS X terminal window titled "Terminal". The window shows the command `sudo docker build -t bmc/service-registry:1.0.0 .` being run and its output. The output details the build process, including pulling Maven dependencies and building the Docker image. The terminal window has several tabs open at the top, including "Downloads", "Inbox", "Subscription Details", "Instances", "Your VPCs", and "ROS Management Console". The status bar at the bottom right indicates it's Tuesday, October 12, 10:24:57 AM.

```
[ec2-user@ip-10-0-197-192 service-registry]$ sudo docker build -t bmc/service-registry:1.0.0 .
Sending build context to Docker daemon 91.14kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
3.8.2-jdk-11: Pulling from library/maven
df6590a8890b: Pull complete
705bb4cb554e: Pull complete
519df5fceacd: Pull complete
ccc287cbeddc: Pull complete
39a2961e8251: Pull complete
0287b7aa7f62: Pull complete
165d4a436d89: Pull complete
01af7eca4dc7: Pull complete
88598a11bfff3: Pull complete
27d30d19529b: Pull complete
Digest: sha256:9d2f1b652dd5e427fc2686e11bd832d4d8f7c412632ff5631d48d58ed34d14db
Status: Downloaded newer image for maven:3.8.2-jdk-11
--> a9a71d043744
Step 2/12 : MAINTAINER ypandyay614929
--> Running in a787c91d2716
Removing intermediate container a787c91d2716
--> eb34fb94010e
Step 3/12 : WORKDIR /opt/app
--> Running in fb9aa5056f4d
Removing intermediate container fb9aa5056f4d
--> 7084ada348bb
Step 4/12 : COPY src /opt/app/src
--> af75c173e4ab
Step 5/12 : COPY pom.xml /opt/app
```

## User-Service Image

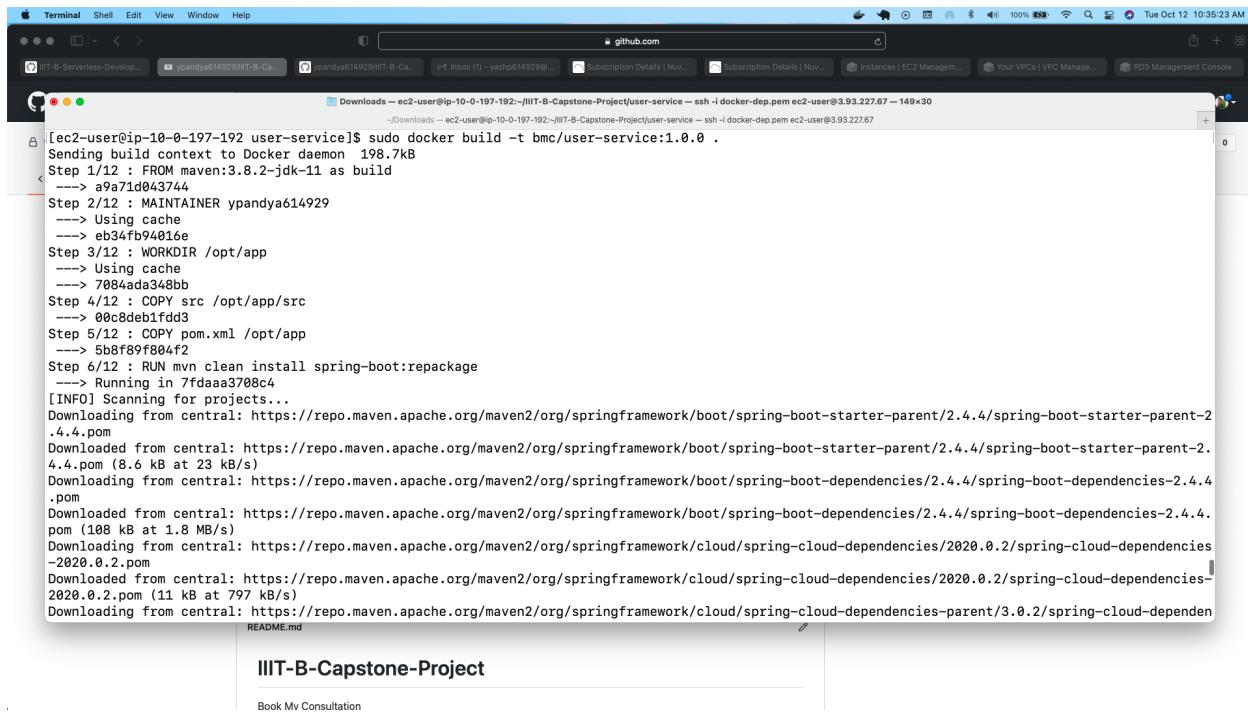
The below commands are present in the security provider server Dockerfile.

```
# 1
FROM maven:3.8.2-jdk-11 as build
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
RUN mvn clean install spring-boot:repackage

# 2
FROM openjdk:14-jdk-alpine
MAINTAINER ypandyay614929
WORKDIR /opt/app
COPY --from=build /opt/app/target/user-service.jar
/opt/app/user-service.jar
EXPOSE 8080
ENTRYPOINT [ "java", "-jar", "/opt/app/user-service.jar"]
```

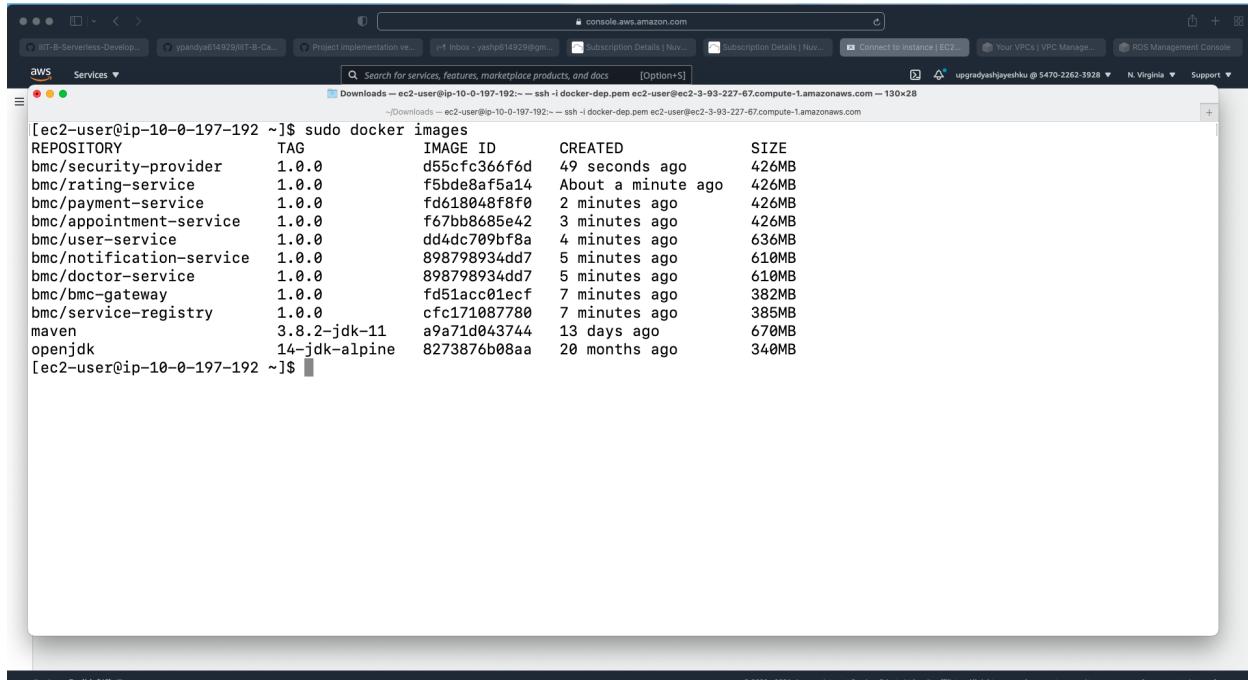
The docker image of the user service is built using the following command.

```
sudo docker build -t bmc/user-service:1.0.0 .
```



```
[ec2-user@ip-10-0-197-192 user-service]$ sudo docker build -t bmc/user-service:1.0.0 .
Sending build context to Docker daemon 198.7kB
Step 1/12 : FROM maven:3.8.2-jdk-11 as build
--> a9a71d043744
Step 2/12 : MAINTAINER ypananya614929
--> Using cache
--> eb34fb94016e
Step 3/12 : WORKDIR /opt/app
--> Using cache
--> 7084ada348bb
Step 4/12 : COPY src /opt/app/src
--> 00c8deb1fdd3
Step 5/12 : COPY pom.xml /opt/app
--> 5b8f89f804f2
Step 6/12 : RUN mvn clean install spring-boot:repackage
--> Running in 7fdaaa3708c4
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.4.4/spring-boot-starter-parent-2.4.4.pom (8.6 kB at 23 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom (108 kB at 1.8 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies/2020.0.2/spring-cloud-dependencies-2020.0.2.pom (11 kB at 797 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/cloud/spring-cloud-dependencies-parent/3.0.2/spring-cloud-dependencies-parent-3.0.2.pom
[ec2-user@ip-10-0-197-192 ~]$
```

All the necessary images are built and can be verified as below.



```
[ec2-user@ip-10-0-197-192 ~]$ sudo docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
bmc/security-provider  1.0.0     d55fcfc366f6d  49 seconds ago  426MB
bmc/rating-service    1.0.0     f5bde8af5a14  About a minute ago  426MB
bmc/payment-service   1.0.0     fd618048f8f0  2 minutes ago  426MB
bmc/appointment-service 1.0.0     f67bb8685e42  3 minutes ago  426MB
bmc/user-service       1.0.0     dd4dc709bf8a  4 minutes ago  636MB
bmc/notification-service 1.0.0     898798934dd7  5 minutes ago  610MB
bmc/doctor-service     1.0.0     898798934dd7  5 minutes ago  610MB
bmc/bmc-gateway        1.0.0     fd51acc01ecf  7 minutes ago  382MB
bmc/service-registry   1.0.0     cfc171087780  7 minutes ago  385MB
maven                 3.8.2-jdk-11  a9a71d043744  13 days ago   670MB
openjdk               14-jdk-alpine  8273876b08aa  20 months ago  340MB
[ec2-user@ip-10-0-197-192 ~]$
```

Further, after building all the images of the micro services, the deployment is carried out by running the docker-compose file.

docker-compose file contains all the specifications related to containers and which image need to use to start/run the container. All networks and volumes can also be added into the file in-order to create based on the necessity.

The sequence of container creation can be specified via depend\_on. Environment classified as a collection of variables that can be used while starting the application from the container. Port mapping is necessary in which one has to specify the port of the container and the mapping of that port with the host machine.

In our docker-compose file we have added `service-registry`, `bmc-gateway`, `notification-service`, `rating-service`, `user-service`, `doctor-service`, `appointment-service`, `payment-service`, and `security-provider`.

The **docker-compose file** is built using the below content.

```
version: '3'

services:
  service-registry:
    build: service-registry
    container_name: serviceregistry
    image: bmc/service-registry:1.0.0
    ports:
      - "8761:8761"
    networks:
      - book-my-consultation-nw
    environment:
      REGISTRY_HOST_NAME: 127.0.0.1
      REGISTRY_PORT: 8761

  bmc-gateway:
    build: bmc-gateway
    container_name: bmcgateway
    image: bmc/bmc-gateway:1.0.0
    ports:
      - "9191:9191"
    networks:
      - book-my-consultation-nw
    environment:
      REGISTRY_HOST_NAME: service-registry
      REGISTRY_PORT: 8761
    depends_on:
      - service-registry
```

```

notification-service:
  build: notification-service
  container_name: notificationservice
  image: bmc/notification-service:1.0.0
  ports:
    - "8085:8085"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    NOTIFICATION_PORT: 8085
  depends_on:
    - service-registry

rating-service:
  build: rating-service
  container_name: ratingservice
  image: bmc/rating-service:1.0.0
  ports:
    - "8084:8084"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    RATING_SVC_PORT: 8084
  depends_on:
    - service-registry
    - bmc-gateway
    - doctor-service

user-service:
  build: user-service
  container_name: userservice
  image: bmc/user-service:1.0.0
  ports:
    - "8083:8083"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    USER_SVC_PORT: 8083
  depends_on:
    - service-registry
    - bmc-gateway
    - notification-service

```

```
doctor-service:
  build: doctor-service
  container_name: doctordservice
  image: bmc/doctor-service:1.0.0
  ports:
    - "8081:8081"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    DOCTOR_SVC_PORT: 8081
  depends_on:
    - service-registry
    - bmc-gateway
    - notification-service

appointment-service:
  build: appointment-service
  container_name: appointmentservice
  image: bmc/appointment-service:1.0.0
  ports:
    - "8082:8082"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    APPOINTMENT_SVC_PORT: 8082
  depends_on:
    - service-registry
    - bmc-gateway
    - notification-service

payment-service:
  build: payment-service
  container_name: paymentservice
  image: bmc/payment-service:1.0.0
  ports:
    - "8086:8086"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    PAYMENT_SVC_PORT: 8086
  depends_on:
    - service-registry
```

```

        - bmc-gateway
        - appointment-service

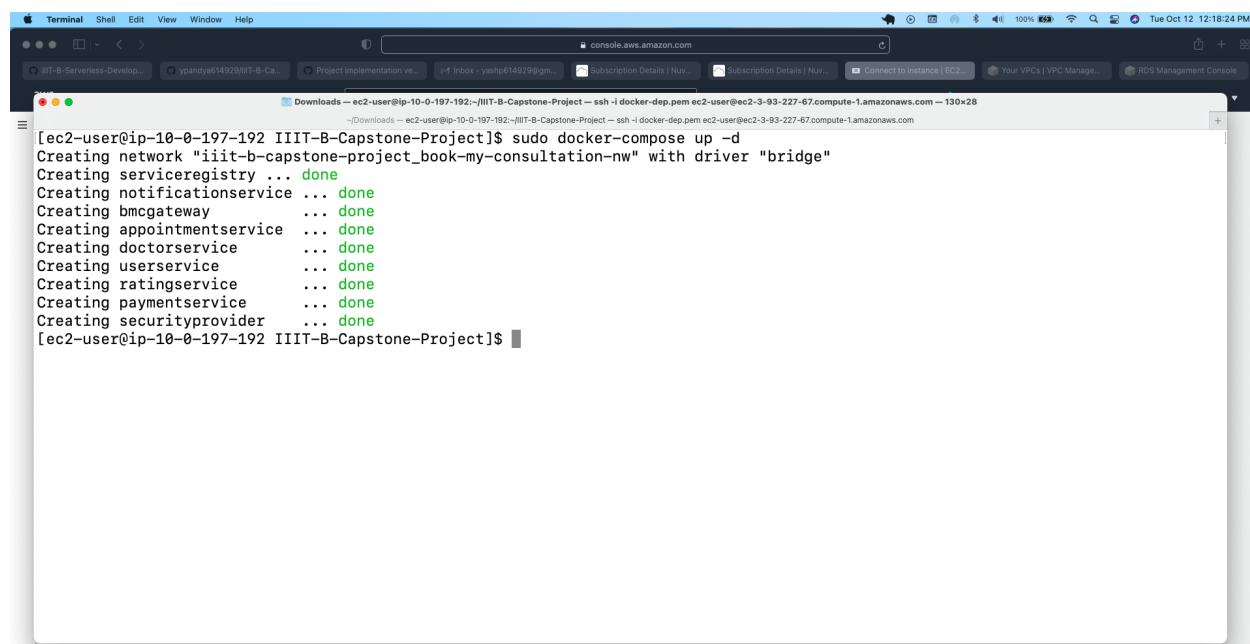
security-provider:
  build: security-provider
  container_name: securityprovider
  image: bmc/security-provider:1.0.0
  ports:
    - "8088:8088"
  networks:
    - book-my-consultation-nw
  environment:
    REGISTRY_HOST_NAME: service-registry
    REGISTRY_PORT: 8761
    SECURITY_PROVIDER_SVC_PORT: 8088
  depends_on:
    - service-registry
    - bmc-gateway
    - doctor-service
    - user-service
    - appointment-service
    - payment-service
    - rating-service

networks:
  book-my-consultation-nw:
    driver: bridge

```

To start the containers based on definition from the docker-compose can be done using the below command.

**sudo docker-compose up -d**



```

Terminal Shell Edit View Window Help
Downloads — ec2-user@ip-10-0-197-192:~/.IIIT-B-Capstone-Project — ssh -i docker-dep.pem ec2-user@ec2-3-93-227-67.compute-1.amazonaws.com — 130x28
Tue Oct 12 12:18:24 PM
Downloads — ec2-user@ip-10-0-197-192:~/.IIIT-B-Capstone-Project — ssh -i docker-dep.pem ec2-user@ec2-3-93-227-67.compute-1.amazonaws.com
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$ sudo docker-compose up -d
Creating network "iiit-b-capstone-project_book-my-consultation-nw" with driver "bridge"
Creating serviceregistry ... done
Creating notificationservice ... done
Creating bmcgateway ... done
Creating appointmentservice ... done
Creating doctorservice ... done
Creating userservice ... done
Creating ratingservice ... done
Creating paymentservice ... done
Creating securityprovider ... done
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$ 

```

Here, a bridge type of network named `book-my-consultation-nw` is created as the docker-compose start building architecture.

All the services are up and running on specified ports and can be verified using below docker command.

```
sudo docker ps -a
```

```
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d58f3e39cc66 bmc/security-provider:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8088->8088/tcp, :::8088->8088/tcp securityprovider
d6d3c5a77b61 bmc/payment-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8086->8086/tcp, :::8086->8086/tcp paymentservice
1c4f49716d9 bmc/rating-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8084->8084/tcp, :::8084->8084/tcp ratingservice
94aa9a14e6ee bmc/user-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8083->8083/tcp, :::8083->8083/tcp userservice
0e223585acd1 bmc/appointment-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8082->8082/tcp, :::8082->8082/tcp appointmentservice
102c7ac450f5 bmc/doctor-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8081->8081/tcp, :::8081->8081/tcp doctorservice
8fb167f97cba bmc/notification-service:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8085->8085/tcp, :::8085->8085/tcp notificationservice
8ca52e8fa071 bmc/bmc-gateway:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:9191->9191/tcp, :::9191->9191/tcp bmcgateway
95be29bc8998 bmc/service-registry:1.0.0 "java -jar /opt/app/—" About a minute ago Up About a minute 8080/tcp, 0.0.0.0:8761->8761/tcp, :::8761->8761/tcp serviceregistry
[ec2-user@ip-10-0-197-192 IIIT-B-Capstone-Project]$
```

Eureka can be viewed from the browser by accessing with the **docker-ec2** public IP which is Elastic IP **docker-ec2-ip** and the eureka port **8761**

Application	AMIs	Availability Zones	Status
APPOINTMENT-SERVICE	n/a (1)	(1)	UP (1) - 2142428c2296:APPOINTMENT-SERVICE:8082
BMC-GATEWAY	n/a (1)	(1)	UP (1) - a269e40084c:BMC-GATEWAY:9191
DOCTOR-SERVICE	n/a (1)	(1)	UP (1) - f05330a65b6b:DOCTOR-SERVICE:8081
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - b88244ab3178:PAYMENT-SERVICE:8086
RATING-SERVICE	n/a (1)	(1)	UP (1) - 2b0933a425d1:RATING-SERVICE:8084
SECURITY-PROVIDER	n/a (1)	(1)	UP (1) - d4b473768793a:SECURITY-PROVIDER:8088
USER-SERVICE	n/a (1)	(1)	UP (1) - 6cccd686e857c:USER-SERVICE:8083

## Storage and Approach

All the documents from the Doctor and User services are being stored at **AWS S3**.

All micro services test cases are attached as below and data validation is also performed and verified using the MySQLWorkbench and/or MongoDB shell in order to validate data consistency.

- Notification service is connected with kafka and service notification is being sent and printed on the console once the other services write the log into the kafka topic and the notification service asynchronously connects and reads the data from the kafka topic.
- Appointment service also reads data from `payment` kafka topic once payment is being made, based on async call, Appointment service reads the appointment id from the kafka topic message and updates the status to Confirmed from PendingPayment.
- Doctor service also reads data from `doctor` kafka topic once rating is being given, based on async call, Doctor service reads the doctor id and average rating from the kafka topic message and updates the average rating of the doctor

Doctor creation, User creation, Doctor approval, Doctor rejection, Appointment confirmation and Prescription issue emails are being sent to respective users by using the **AWS SES service**.

To preview notification service logs, one has to preview the logs of the notification service container.

Docker has a built-in command to analyse the container logs. The command to preview the logs from the notification container is as shown below.

```
sudo docker logs notificationservice
```

## Instructions To Run

- The RDS database URL needs to be mapped in all the services properties and username and password too (if changes).
- EC2 Elastic IP of **kafka-ec2** instance needs to be mapped in the KafkaConfig and Consumer classes.

- Kafka Consumer classes settings are mapped in application.properties as those consumers have producer and consumer both types of intrims
- Eureka/Service-Registry ie. **docker-ec2** Elastic IP needs to be mapped in the docker-compose file as **REGISTRY\_HOST\_NAME** within Environment section of **service-registry**

Once all the settings changed, including application properties for MongoDB and RDS (MySQL) and above mentioned IP settings, one need to run build images command mentioned above for each services and then docker build command to start the container and all services are running on specified ports can be verified from the eureka URL.

### Note

- Kafka is externalized using the two EC2 instances approach.

### Verification

All the micro services are running on a specified port, all micro services are verified using the postman.

- User service endpoints are defined as below and running on port 8083

```

POST /users
GET /users/{userId}
POST /users/{id}/documents
GET /users/{id}/documents/{documentName}
GET /users/{id}/documents/metadata

```

- Doctor service endpoints are defined as below and running on port 8081

```

POST /doctors
POST /doctors/{doctorId}/documents
PUT /doctors/{doctorId}/approve
PUT /doctors/{doctorId}/reject
GET /doctors
GET /doctors/{doctorId}
GET /doctors/{doctorId}/documents/metadata
GET /doctors/{doctorId}/documents/{documentName}

```

- Appointment service endpoints are defined as below and running on port 8082

```

POST /doctor/{doctorId}/availability
GET /doctor/{doctorId}/availability
POST /appointments
GET /appointments/{appointmentId}
GET /users/{userId}/appointments
POST /prescriptions

```

- Payment service endpoints are defined as below and running on port 8086

POST /payments

- Rating service endpoints are defined as below and running on port 8084

POST /ratings

- Security-Provider service endpoints are defined as below and running on port 8088

POST /generate-token

GET /user/all

- API Gateway running on port 9191 and redirects the traffic based on the matching path

## Test Results

The screenshot shows the Postman application interface. The left sidebar displays the 'BookMyConsultation' collection with several requests listed under 'user-service'. The main workspace shows a POST request to 'http://3.93.227.67:8083/users' with the following JSON body:

```

1 {
2   "firstName": "Yash",
3   "lastName": "",
4   "dob": "1997-03",
5   "emailId": "yash6149@gmail.com",
6   "mobile": "8734924"
7 }
8

```

The response status is 401 Unauthorized, and the response body is:

```

1 {
2   "timestamp": "2021-10-12T16:56:02.272+00:00",
3   "status": 401,
4   "error": "Unauthorized",
5   "message": "",
6   "path": "/users"
7 }

```

Postman interface showing a successful POST request to generate a JWT token for an admin user. The response status is 200 OK, time is 1070 ms, and size is 684 B.

```

POST /foodDelivery/security/generate-token
{
  "username": "admin-user@abc.com",
  "password": "Admin@123"
}

```

Postman interface showing a successful POST request to generate a JWT token for a user. The response status is 200 OK, time is 508 ms, and size is 684 B.

```

POST /foodDelivery/security/generate-token
{
  "username": "normal-user@abc.com",
  "password": "Test@123"
}

```

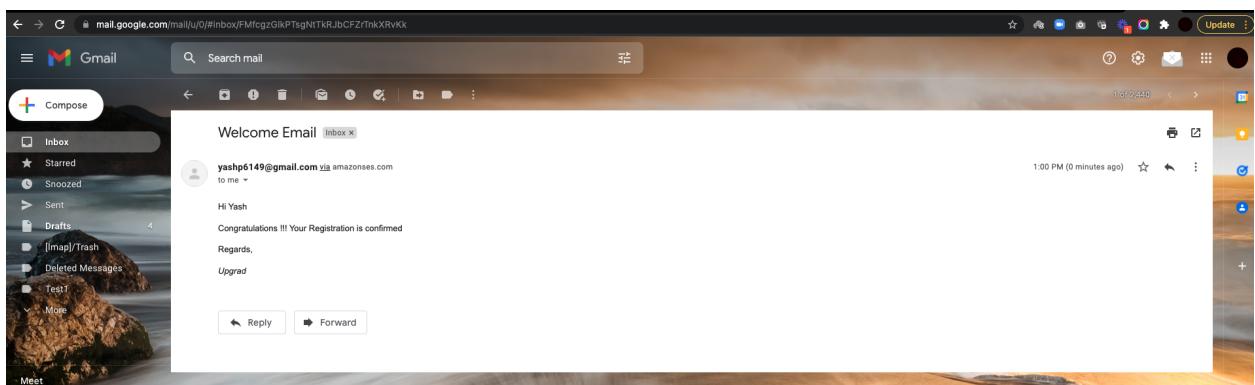
Please note that Token service only generates the token whereas all other services can validate the token separately.

Postman interface showing a successful POST request to register a new user. The response status is 201 Created, time is 872 ms, and size is 612 B.

```

POST /users
{
  "firstName": "Yash",
  "lastName": "Pandya",
  "dob": "1997-03-29",
  "emailId": "yashp6149@gmail.com",
  "mobile": "8734924982"
}

```



The screenshot shows a Postman interface with a collection named 'BookMyConsultation'. A GET request is made to 'http://3.93.227.67:8083/users/6165bf1280a5a4767c559c1d'. The response status is 201 Created, and the response body is:

```
1 {
2   "id": "6165bf1280a5a4767c559c1d",
3   "firstName": "Yash",
4   "lastName": "Pandya",
5   "email": "yashp6149@gmail.com",
6   "mobile": "8734924982",
7   "emailId": "yashp6149@gmail.com",
8   "createdDate": "2021-10-12"
9 }
```

The screenshot shows a Postman interface with the same collection. A GET request is made to 'http://3.93.227.67:8083/users/123123123'. The response status is 404 Not Found, and the response body is:

```
1 {
2   "errorCode": "ERR_RESOURCE_NOT_FOUND",
3   "errorMessage": "Requested resource is not available",
4   "errorFields": null
5 }
```

The Postman interface shows a successful file upload. The request URL is `http://3.93.227.67:8083/users/6165bf1280a5a4767c559c1d/documents`. The response status is 200 OK, time 874 ms, size 464 B. The response body contains the message "1 File(s) uploaded Successfully".

The Postman interface shows a successful file download. The request URL is `http://3.93.227.67:8083/users/6165bf1280a5a4767c559c1d/documents/wp5192934.jpeg`. The response status is 200 OK, time 269 ms, size 90.49 kB. The response body displays the binary file content.

The Postman interface shows a successful file metadata retrieval. The request URL is `http://3.93.227.67:8083/users/6165bf1280a5a4767c559c1d/documents/metadata`. The response status is 200 OK, time 242 ms, size 453 B. The response body shows the file path: "[{"path": "wp5192934.jpeg"}]".

Postman - Collections

POST Doctor Registration

Body (11) JSON

```
1 "firstName": "Yash Doctor",
2 "lastName": "Pandya",
3 "dob": "1997-03-29",
4 "emailId": "yashp6149@gmail.com",
5 "mobile": "8787878787",
6 "pan": "XAFLF4215P"
```

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2021-10-12T17:06:08.900+00:00",
3   "path": "/doctors",
4   "error": "Unauthorized",
5   "message": "",
6   "path": "/doctors"
7 }
```

Status: 401 Unauthorized Time: 445 ms Size: 632 B Save Response

Postman - Collections

POST Doctor Registration

Body (11) JSON

```
1 "firstName": "Yash Doctor",
2 "lastName": "Pandya",
3 "dob": "1997-03-29",
4 "emailId": "yashp6149@gmail.com",
5 "mobile": "8787878787",
6 "pan": "XAFLF4215P"
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0105dc12cde6447aebb61c",
3   "firstName": "Yash Doctor",
4   "lastName": "Pandya",
5   "speciality": "GENERAL_PHYSICIAN",
6   "dob": "1997-03-29",
7   "mobile": "8787878787",
8   "emailId": "yashp6149@gmail.com",
9   "status": "Pending",
10  "registrationDate": "2021-10-12"
11 }
12 }
```

Status: 201 Created Time: 1282 ms Size: 695 B Save Response

Gmail - Welcome Email

Compose

Inbox

Starred

Snoozed

Sent

Drafts

[imap]/Trash

Deleted Messages

Test1

More

Meet

Search mail

yashp6149@gmail.com via amazonsees.com to me 1:07 PM (0 minutes ago)

Hi Yash Doctor

Congratulations !!! Your Registration is confirmed

Regards,

Upgrad

Reply Forward

**Postman** | Tue Oct 12, 1:08:29 PM

My Workspace | No Environment

POST /JWT Token for admin | POST /JWT Token for user | POST Doctor Registration | GET Fetch the doctors details based on doctor ID

GET http://3.93.227.67:8081/doctors/123123123

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

DESCRIPTION Description

Cookies Code Bulk Edit

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "errorCode": "ERR_RESOURCE_NOT_FOUND",
3     "errorMessage": "Requested resource is not available",
4     "errorFields": null
5 }
```

Status: 404 Not Found Time: 191 ms Size: 553 B Save Response

Send Save

Collection: BookMyConsultation (23 requests)

- user-service
- rating-service
- appointment-service
- doctor-service
  - Fetch the doctors details based on ...
  - Fetch the list of doctors based on st...
  - PUT Approve the doctors registration
  - PUT Reject the doctors registration
  - POST Doctor Registration
  - GET Document Download
  - POST Document Upload
  - GET Metadata of the files uploaded by t...
- jwt token
- payment-service

**Postman** | Tue Oct 12, 1:08:43 PM

My Workspace | No Environment

POST /JWT Token for admin | POST /JWT Token for user | POST Doctor Registration | GET Fetch the doctors details based on doctor ID

GET http://3.93.227.67:8081/doctors/61650c122ca6e447aeabb61c

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

DESCRIPTION Description

Cookies Code Bulk Edit

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "id": "61650c122ca6e447aeabb61c",
3     "firstName": "Vash Doctor",
4     "lastName": "Pandya",
5     "speciality": "GENERAL_PHYSICIAN",
6     "dob": "1997-03-29",
7     "mobile": "8787878787",
8     "emailId": "yashp614@gmail.com",
9     "pan": "XOMLF421SP",
10    "status": "Pending",
11    "registrationDate": "2021-10-12"
12 }
```

Status: 201 Created Time: 136 ms Size: 695 B Save Response

Send Save

Collection: BookMyConsultation (23 requests)

- user-service
- rating-service
- appointment-service
- doctor-service
  - Fetch the doctors details based on ...
  - Fetch the list of doctors based on st...
  - PUT Approve the doctors registration
  - PUT Reject the doctors registration
  - POST Doctor Registration
  - GET Document Download
  - POST Document Upload
  - GET Metadata of the files uploaded by t...
- jwt token
- payment-service

**Postman** | Tue Oct 12, 1:09:41 PM

My Workspace | No Environment

POST /JWT Token for admin | POST /JWT Token for user | POST Doctor Registration | GET Fetch the doctors details based on doctor ID | PUT Approve the doctors registration

PUT http://3.93.227.67:8081/doctors/61650c122ca6e447aeabb61c/approve

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Body

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2     "approvedBy": "Yash Admin",
3     "approverComments": "Doctor is Verified"
4 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

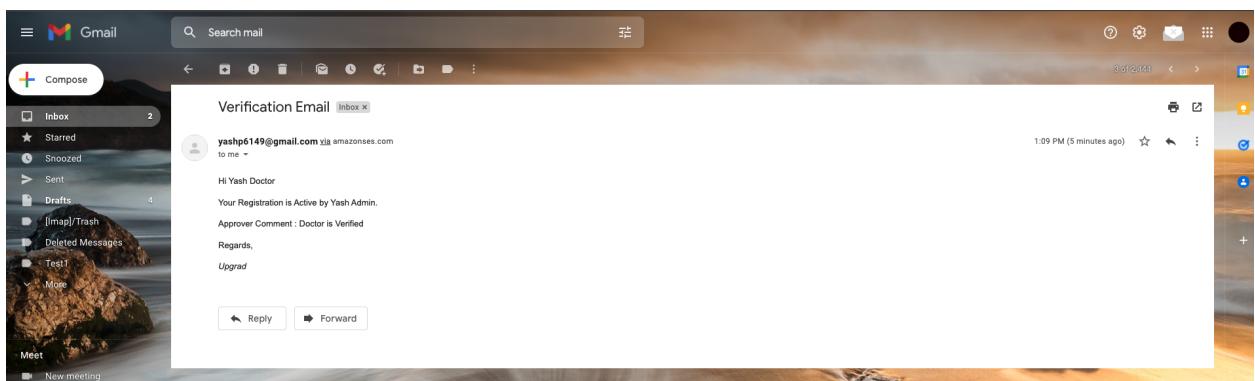
```
1 {
2     "id": "61650c122ca6e447aeabb61c",
3     "firstName": "Vash Doctor",
4     "lastName": "Pandya",
5     "speciality": "GENERAL_PHYSICIAN",
6     "dob": "1997-03-29",
7     "mobile": "8787878787",
8     "emailId": "yashp614@gmail.com",
9     "pan": "XOMLF421SP",
10    "status": "Active",
11    "approvedBy": "Yash Admin",
12    "approverComments": "Doctor is Verified",
13    "registrationDate": "2021-10-12",
14    "verificationDate": "2021-10-12"
15 }
```

Status: 200 OK Time: 74 ms Size: 787 B Save Response

Send Save

Collection: BookMyConsultation (23 requests)

- user-service
- rating-service
- appointment-service
- doctor-service
  - Fetch the doctors details based on ...
  - Fetch the list of doctors based on st...
  - PUT Approve the doctors registration
  - PUT Reject the doctors registration
  - POST Doctor Registration
  - GET Document Download
  - POST Document Upload
  - GET Metadata of the files uploaded by t...
- jwt token
- payment-service



POST JWT Token for admin

POST JWT Token for user

POST Doctor Registration

GET Fetch the doctors details based on ...

GET Fetch the list of doctors based on st...

PUT Approve the doctors registration

**PUT Reject the doctors registration**

Doctor Registration

Document Download

Document Upload

Metadata of the files uploaded by t...

Jwt token

payment-service

Reject the doctors registration

My Workspace

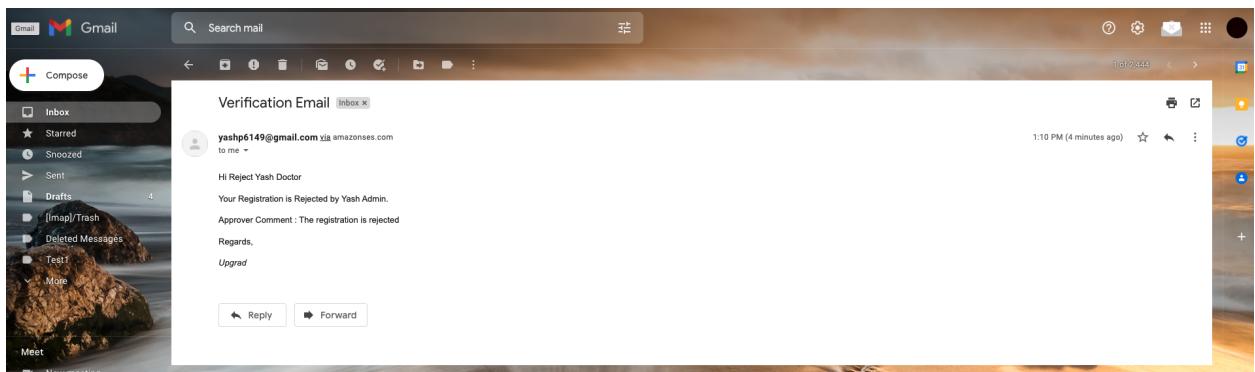
Send Save

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 "approvedBy": "Yash Admin",
2 "approverComments": "The registration is rejected"
3
4
5
```

Status: 200 OK Time: 52 ms Size: 806 B Save Response



Postman screenshot showing a successful file upload response. The request URL is `http://3.93.227.67.8081/doctors/6165c0c122ca6e447aeabb61c/documents`. The response status is 200 OK, time 2.66 s, size 464 B. The response body shows "1 File(s) uploaded Successfully".

Postman screenshot showing a successful document download response. The request URL is `http://3.93.227.67.8081/doctors/6165c0c122ca6e447aeabb61c/documents/Eureka.png`. The response status is 200 OK, time 224 ms, size 170.08 KB. The response body shows a large binary file (Eureka.png) starting with `1 PNG`.

Postman screenshot showing a successful metadata retrieval response. The request URL is `http://3.93.227.67.8081/doctors/6165c0c122ca6e447aeabb61c/documents/metadata`. The response status is 200 OK, time 197 ms, size 449 B. The response body shows the file name "Eureka.png".

Postman - Tue Oct 12, 11:14:07 PM

bookm

History Collections APIs + New Collection Trash

BookMyConsultation 23 requests

- / user-service
- / rating-service
- / appointment-service
- / doctor-service
  - GET Fetch the doctors details based on ...
  - GET Fetch the list of doctors based on st...
  - PUT Approve the doctors registration
  - PUT Reject the doctors registration
  - POST Doctor Registration
  - GET Document Download
  - POST Document Upload
  - GET Metadata of the files uploaded by t...
- / jwt token
- / payment-service

POST /JWT Token for admin ● POST /JWT Token for user ● GET Fetch the list of doctors based ... +

My Workspace

No Environment

GET http://3.93.227.67:8081/doctors?status=Active&speciality=GENERAL\_PHYSICIAN

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> status	Active	
<input checked="" type="checkbox"/> speciality	GENERAL_PHYSICIAN	
Key	Value	Description

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "6165c16c22ca6e447aeabb61c",
4     "firstName": "Reject Yash Doctor",
5     "lastName": "Pandy",
6     "speciality": "GENERAL_PHYSICIAN",
7     "dob": "1997-03-29",
8     "mobile": "8787878787",
9     "email": "yashhp149@gmail.com",
10    "pan": "XGALF4ZC987654321",
11    "status": "Active",
12    "approvedBy": "Yash Admin",
13    "approverComments": "Doctor is Verified",
14    "registrationDate": "2021-10-12",
15    "verificationDate": "2021-10-12",
16    "averageRating": 0.0
17  }
18 ]
```

Find and Replace Console

Bootcamp Build Browse

Postman - Tue Oct 12, 11:14:17 PM

bookm

History Collections APIs + New Collection Trash

BookMyConsultation 23 requests

- / user-service
- / rating-service
- / appointment-service
- / doctor-service
  - GET Fetch the doctors details based on ...
  - GET Fetch the list of doctors based on st...
  - PUT Approve the doctors registration
  - PUT Reject the doctors registration
  - POST Doctor Registration
  - GET Document Download
  - POST Document Upload
  - GET Metadata of the files uploaded by t...
- / jwt token
- / payment-service

POST /JWT Token for admin ● POST /JWT Token for user ● GET Fetch the list of doctors based ... +

My Workspace

No Environment

GET http://3.93.227.67:8081/doctors?status=Rejected&speciality=GENERAL\_PHYSICIAN

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> status	Rejected	
<input checked="" type="checkbox"/> speciality	GENERAL_PHYSICIAN	
Key	Value	Description

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "6165c16c22ca6e447aeabb61d",
4     "firstName": "Reject Yash Doctor",
5     "lastName": "Pandy",
6     "speciality": "GENERAL_PHYSICIAN",
7     "dob": "1997-03-29",
8     "mobile": "1231231231",
9     "emailId": "yashhp149@gmail.com",
10    "pan": "123YASH0321",
11    "status": "Rejected",
12    "approvedBy": "Yash Admin",
13    "approverComments": "The registration is rejected",
14    "registrationDate": "2021-10-12",
15    "verificationDate": "2021-10-12",
16    "averageRating": 0.0
17  }
18 ]
```

Find and Replace Console

Bootcamp Build Browse

**Postman** - BookMyConsultation

POST /JWT Token for admin    POST /JWT Token for user    GET Fetch the list of doctors based on status and speciality

GET http://3.93.227.67:8081/doctors?specialty=GENERAL\_PHYSICIAN

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
status		
specialty	GENERAL_PHYSICIAN	

Body Cookies (1) Headers (14) Test Results

```

1 [
2   {
3     "id": "6165c8c122ca6e447aebb61c",
4     "firstName": "Yash Doctor",
5     "lastName": "Pandya",
6     "specialty": "GENERAL_PHYSICIAN",
7     "dob": "1997-03-29",
8     "mobile": "+9878787878",
9     "emailId": "yashhp14@gmail.com",
10    "pan": "UHFA215P",
11    "status": "Approved",
12    "approvedBy": "Yash Admin",
13    "approverComments": "Doctor is Verified",
14    "registrationDate": "2021-10-12",
15    "verificationDate": "2021-10-12",
16    "averageRating": 0.0
17  },
18  {
19    "id": "6165c1622ca6e447aebb61d",
20    "firstName": "Reject Yash Doctor",
21    "lastName": "Pandya",
22    "specialty": "GENERAL_PHYSICIAN",
23    "dob": "1997-03-29",
24    "mobile": "+9232232331",
25    "emailId": "yashhp14@gmail.com",
26    "pan": "123ASH0321"
}

```

Find and Replace    Console    Bootcamp    Build    Browse    Help

**Postman** - BookMyConsultation

POST /JWT Token for admin    POST /JWT Token for user    POST Doctor updates his availability    GET Fetch the doctor's availability based on doctor ID

POST http://3.93.227.67:8082/doctor/6165c0c122ca6e447aebb61c/availability

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Body

```

1 {
2   "availabilityMap": {
3     "2021-10-12": ["10AM-11AM"]
4   }
5 }

```

Body Cookies (1) Headers (13) Test Results

```

1

```

Find and Replace    Console    Bootcamp    Build    Browse    Help

**Postman** - BookMyConsultation

POST /JWT Token for admin    POST /JWT Token for user    POST Doctor updates his availability    GET Fetch the doctor's availability based on doctor ID

GET http://3.93.227.67:8082/doctor/6165c0c122ca6e447aebb61c/availability

Headers

KEY	VALUE	DESCRIPTION
Content-Type	application/json	
Authorization	eyhbGicOjIjUzUmIj9eyJpc3MIoJlVcGdyYWQgRGVtbylslnN1Yilim5vcmlhbC	

Body Cookies (1) Headers (14) Test Results

```

1 [
2   {
3     "doctorId": "6165c8c122ca6e447aebb61c",
4     "availabilityMap": {
5       "2021-10-12": [
6         "10AM-11AM"
7       ]
8     }
}

```

Find and Replace    Console    Bootcamp    Build    Browse    Help

**Postman**

bookmycon

My Workspace

POST /JWT Token for admin

POST /JWT Token for user

POST User books an appointment

POST [User books an appointment](http://3.93.227.67.8082/appointments)

POST [User books an appointment](http://3.93.227.67.8082/appointments)

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Body

```
1 "doctor_id": "6165d1d9ea19c7d783a8851",
2 "doctor_name": "Yash Doctor",
3 "user_id": "6165d1d9ea19c7d783a8851",
4 "time_slot": "10AM-11AM",
5 "appointment_date": "2021-10-12"
6
```

Cookies Code

Send Save

Examples BUILD

No Environment

POST http://3.93.227.67.8082/appointments

Status: 200 OK Time: 458 ms Size: 534 B Save Response

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize JSON

1 2c9580857c75b99c817c75c1bbc0000

Gmail

Inbox

Compose

Search mail

Appointment Email

yashp6149@gmail.com via amazonsees.com to me 2:47 PM (0 minutes ago)

Hi Yash Pandya  
Your Appointment is registered.  
Details:  
Doctor Name : Yash Doctor  
Appointment Date : 2021-10-12  
Time : 10AM-11AM  
Status : PendingPayment  
Regards,  
Upgrad

Reply Forward

Postman

bookmycon

My Workspace

POST /JWT Token for admin

POST /JWT Token for user

POST User books an appointment

GET Fetch the details of an appointment

POST [Fetch the details of an appointment based on appointmentID](http://3.93.227.67.8082/appointments/2c9580857c75b99c817c75c1bbc0000)

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers

KEY	VALUE	DESCRIPTION
Content-Type	application/json	
Authorization	eyhbGic0JiljUzUxMjR.eyJpc3MIQjVcGdyYWQgRGVtbyslnN1YlilmSvcm1hbC	

Body

```
1 "appointment_id": "2c9580857c75b99c817c75c1bbc0000",
2 "appointment_time": "2021-10-12T10:00:00Z",
3 "doctor_id": "6165d1d9ea19c7d783a8851",
4 "doctor_name": "Yash Doctor",
5 "user_id": "6165d1d9ea19c7d783a8851",
6 "time_slot": "10AM-11AM",
7 "status": "PendingPayment",
8 "appointment_date": "2021-10-12"
9
```

Cookies Code

Send Save

Examples BUILD

No Environment

GET http://3.93.227.67.8082/appointments/2c9580857c75b99c817c75c1bbc0000

Status: 200 OK Time: 66 ms Size: 659 B Save Response

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

**Postman**

My Workspace

POST [JWT Token for admin] POST [JWT Token for user] POST User books an appointment GET Fetch all the appointments of a... + ...

Fetch all the appointments of a user based on userId

GET http://3.93.227.67:8082/users/6165d1d8eaef90c7d783a885f/appointments

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers (7 hidden)

KEY	VALUE	DESCRIPTION
Content-Type	application/json	
Authorization	eyjhbgIiJUzUmIj9eyJpc3MIQjVcGdyYWQgRGVtbylsInN1Yl6im5vcm1hbC	
Key	Value	Description

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "appointmentId": "2c9580857c75b99c017c75c1bbcb0000",
4     "doctorId": "6165d23882494729e08d27a",
5     "doctorName": null,
6     "userId": "6165d1d8eaef90c7d783a885f",
7     "timeSlot": "10:00 AM",
8     "status": "PendingPayment",
9     "appointmentDate": "2021-10-12"
10   }
11 ]

```

Status: 200 OK Time: 48 ms Size: 661 B Save Response

**Postman**

My Workspace

POST [JWT Token for admin] POST [JWT Token for user] POST User books an appointment GET Fetch all the appointments of a... POST Prescription + ...

Prescription

POST http://3.93.227.67:8082/prescriptions

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Body (11) none form-data x-www-form-urlencoded raw binary GraphQL JSON

KEY	VALUE	DESCRIPTION
appointmentId	"2c9580857c75b99c017c75c1bbcb0000"	
doctorId	"6165d23882494729e08d27a"	
doctorName	"Dr. John Doe"	
userId	"6165d1d8eaef90c7d783a885f"	
diagnosis	"Teeth Cavity"	
medicineList	[{"name": "Ibuprofen", "type": "Tablet", "dosage": "1 tablet", "duration": "1 week", "frequency": "3 times a day"}, {"name": "Paracetamol", "type": "Tablet", "dosage": "1 tablet", "duration": "1 week", "frequency": "3 times a day"}]	

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "appointmentId": "2c9580857c75b99c017c75c1bbcb0000",
4     "doctorId": "6165d23882494729e08d27a",
5     "doctorName": "Dr. John Doe",
6     "userId": "6165d1d8eaef90c7d783a885f",
7     "diagnosis": "Teeth Cavity",
8     "medicineList": [
9       {
10         "name": "Ibuprofen",
11         "type": "Tablet",
12         "dosage": "1 tablet",
13         "duration": "1 week",
14         "frequency": "3 times a day",
15       },
16       {
17         "name": "Paracetamol",
18         "type": "Tablet",
19         "dosage": "1 tablet",
20         "duration": "1 week",
21         "frequency": "3 times a day"
22       }
23     ]
24   }
25 ]

```

errorCode: "ERR\_PAYMENT\_PENDING", errorMessage: "Prescription cannot be issued since the payment status is pending"

Status: 404 Not Found Time: 109 ms Size: 560 B Save Response

**Postman**

My Workspace

POST [JWT Token for admin] POST [JWT Token for user] POST User books an appointment GET Fetch all the appointments of a... POST Prescription POST Payment endpoint + ...

Payment endpoint

POST http://3.93.227.67:8086/payments?appointmentId=2c9580857c75b99c017c75c1bbcb0000

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers (8 hidden)

KEY	VALUE	DESCRIPTION
Authorization	eyjhbgIiJUzUmIj9eyJpc3MIQjVcGdyYWQgRGVtbylsInN1Yl6im5vcm1hbC	
Key	Value	Description

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": "6165d6898728f347481ab9a",
4     "appointmentId": "2c9580857c75b99c017c75c1bbcb0000",
5     "createDate": "2021-10-12T18:46:09.233490Z"
5 ]

```

Status: 200 OK Time: 438 ms Size: 637 B Save Response

**Postman API Call - Prescription**

POST <http://3.93.227.67:8082/prescriptions>

```

1 {
  "appointmentId": "2c9590857c75b99e017c75c1bbcb0000",
  "doctorId": "6165d238b82494729e08d27a",
  "doctorName": "Yash Doctor",
  "userId": "6165d10e8ef98c70783aa885",
  "diagnosis": "Teeth Cavity",
  "medicineList": [
    {
      "name": "Calpol",
      "type": "Tablet",
      "dosage": "1 week",
      "duration": "1 week",
      "frequency": "3 times a day"
    }
  ]
}

```

Status: 200 OK Time: 200 ms Size: 469 B Save Response

**Email Confirmation - Your Prescription is Ready**

Subject: Your Prescription is Ready

Hi,  
Your Prescription is ready.  
Details:  
Doctor Id: 6165d238b82494729e08d27a  
Doctor Name: Yash Doctor  
Appointment Id : 2c9590857c75b99e017c75c1bbcb0000  
Diagnosis: Teeth Cavity  
Medicines : name=Calpol | type=Tablet | dosage=1 week | duration=1 week | frequency=3 times a day | remarks=after food | name=PainKill | type=Syrup | dosage=1 week | duration=1 week | frequency=3 times a day | remarks=after food

Regards,  
Upgrad

Reply Forward

**Postman API Call - Rate the doctor**

POST <http://3.93.227.67:8084/ratings>

```

1 {
  "doctorId": "6165d238b82494729e08d27a",
  "rating": "3",
  "comments": "Doctor is good"
}

```

Status: 200 OK Time: 960 ms Size: 469 B Save Response

**Postman API Call - Rate the doctor**

POST <http://3.93.227.67:8084/ratings>

```

1 {
  "doctorId": "6165d238b82494729e08d27a",
  "rating": "4",
  "comments": "Doctor is fab"
}

```

Status: 200 OK Time: 217 ms Size: 394 B Save Response

```

MongoDB [IIT-B-Serverless-Dev...]
> db.doctor.find({"_id" : ObjectId("6165d238b82494729e08d27a")})
{
  "_id": ObjectId("6165d238b82494729e08d27a"),
  "firstName": "Yash Doctor",
  "lastName": "Pandya",
  "speciality": "GENERAL_PHYSICIAN",
  "dob": "1997-03-29",
  "mobile": "8787878787",
  "emailId": "yashp014@gmail.com",
  "pan": "XDALF4215P",
  "status": "Active",
  "approvedBy": "Yash Admin",
  "approverComments": "Verified",
  "registrationDate": "2021-10-12",
  "verificationDate": "2021-10-12",
  "averageRating": 3.5,
  "_class": "bookmyconsultation.doctorservice.entity.DoctorEntity"
}
>

MongoDB [IIT-B-Serverless-Dev...]
> db.rating.find()
{
  "_id": ObjectId("6165d6bdbcb70209b7e6fd1d"),
  "doctorId": "6165d238b82494729e08d27a",
  "rating": 3,
  "comments": "Doctor is good",
  "_class": "bookmyconsultation.ratingservice.entity.RatingEntity"
}
{
  "_id": ObjectId("6165d6cddbc70209b7e6fd1e"),
  "doctorId": "6165d238b82494729e08d27a",
  "rating": 4,
  "comments": "Doctor is fab",
  "_class": "bookmyconsultation.ratingservice.entity.RatingEntity"
}
>

```

- Ratings are updated by doctor service via rating service using kafka to send doctor Id and rating in the topic and doctor services fetches it and updates.
- Similarly, once a doctor registration is approved or rejected, doctor service sends notification to notification service using kafka and notification service sends the email.
- When payment is being made, payment service sends a message to appointment service via kafka topic and appointment services changes status to Confirmed from PendingPayment and once Payment is completed and appointment status is Confirmed then only Prescription can be registered and sent via email.