Here we have used Ubuntu 20.04 AMI to Create EC2s. If you'd like to go with any other OS, commands would be changed.

AWS Deployment

Access to EC2 via Putty

To access an AWS EC2 instance using PuTTY on Windows, you need to convert the private key file from the .pem format to PuTTY's .ppk format and then use PuTTY to establish an SSH connection. Here are the steps:

Step 1: Convert. pem Key to .ppk Format

1. Download and install PuTTY if you haven't already. You can get it from the PuTTY website.
2. Launch PuTTYgen, which comes with PuTTY, and click the "Load" button.
3. In the file dialog, select your .pem private key file.
4. Click "Open."
5. PuTTYgen will prompt you to convert the key into PuTTY's own format. Click the "Save private key" button.
6. Save the key in .ppk format.

Step 2: Configure PuTTY

1. Launch PuTTY.
2. In the "Session" category, enter your EC2 instance's public IP address or DNS hostname in the "Host Name (or IP address)" field.
3. In the "Connection" > "Data" category, enter the username used for your EC2 instance. For Ubuntu 20.4, the username is typically ubuntu. For other distributions, check the official documentation for the correct username.
4. In the "Connection" > "SSH" > "Auth" category, click the "Browse" button and select the .ppk private key file you created earlier.
5. Optionally, you can save this configuration for future use by entering a name in the "Saved Sessions" field and clicking the "Save" button.
6. Click "Open" to initiate the SSH connection.

Step 3: Connect to EC2 Instance

PuTTY will use the private key to authenticate, and if everything is configured correctly, it should establish a connection to your EC2 instance.

Once connected, you can use the terminal provided by PuTTY to interact with your EC2 instance as if you were using SSH from a Linux terminal.

1. SETTING UP THE Database

1.1. Configure MySQL

To Install as root user, type below command

```
sudo su
```

Then update the local package with below command.

```
sudo apt update
```

Install the mysql server package with below command.

```
sudo apt install mysql-server
```

Check the MySQL server status

```
sudo systemctl status mysql
```

1.2. Start MySQL database server.

```
sudo systemctl start mysql
```

1.3. Login into MySQL DB

```
mysql -u root -p
```

No password to enter.

1.4. Create the data base with below query.

```
CREATE DATABASE IF NOT EXISTS `car_rental_db`;
show databases;
```

1.5. Create tables in database.

```
USE `car_rental_db`;
```

1.6. Create Car Table

```
CREATE TABLE IF NOT EXISTS `t_cars` (
  `id` int NOT NULL AUTO_INCREMENT,
  `car_code` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,

  `make` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,

  `model` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,

  `year` int NOT NULL,

  `license_plate` varchar(20) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,

  `availability` tinyint(1) NOT NULL,

  `image_url` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,

  `location_uuid` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,

  `per_hour_rate` decimal(10,2) NOT NULL,

  `per_day_rate` decimal(10,2) NOT NULL,

  `leasing_rate` decimal(10,2) NOT NULL,

  `car_description` varchar(255) DEFAULT NULL,

  `mileage` bigint DEFAULT NULL,

  `transmission` varchar(255) DEFAULT NULL,

  `seats` int DEFAULT NULL,

  `luggage` int DEFAULT NULL,

  `fuel` varchar(50) DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

1.7. Enter Data into table (‘t_cars’)

```
INSERT INTO `t_cars` (`id`, `car_code`, `make`, `model`, `year`,
`license_plate`, `availability`, `image_url`, `location_uuid`,
`per_hour_rate`, `per_day_rate`, `leasing_rate`,
`car_description`, `mileage`, `transmission`, `seats`, `luggage`,
`fuel`) VALUES

  (1, '45f6f72c-6f6a-4b36-9d33-954ccb62be0a', 'Lexus', 'Lexus
2022', 2022, 'BGH 9072', 1, 'assets/img/lexus1.jpg', '71d45ae7-
92f5-4621-9d3d-1e407967a79f', 12.99, 111.12, 4243.98, 'A small
river named Duden flows by their place and supplies it with the
necessary regelialia.', 12120, 'Manul', 5, 5, 'Petrol');
```

1.8. Create location table.

```
CREATE TABLE IF NOT EXISTS `t_locations` (

  `id` int NOT NULL AUTO_INCREMENT,

  `location_name` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,

  `location_address` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,

  `location_uuid` varchar(36) NOT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

1.9. Enter Data into Location table.

```
INSERT INTO `t_locations` (`id`, `location_name`,
`location_address`, `location_uuid`) VALUES

      (1, 'Wellawaya', 'Wellwaya Road Wellwaya', '71d45ae7-92f5-
4621-9d3d-1e407967a79f'),

      (2, 'Colombo', 'Colombo Road Colombo', '5fbf593f-cf44-4562-
825a-6b932b117587'),

      (3, 'Kandy', 'Kandy Road Kandy', 'a62414fd-c56e-4de0-9409-
f7b7f5a7c71d'),

      (4, 'Jaffna', 'Jaffna Road Jaffna', '5e1a43cc-b67b-4e98-
b0bb-6ae33535180e'),

      (5, 'Gampaha', 'Gampaha Road Gampaha', '5056fb9b-bbbc-404d-
964b-a50bc26235f2'),

      (6, 'Galle', 'Galle Road Galle', 'cde00356-1321-4bbc-b648-
afcbd9f3dfcf'),

      (7, 'Matara', 'Matara Road Matara', '34043148-d701-4f11-
9525-5fb965be2f2d');
```

1.10. Create a reservation table.

```
CREATE TABLE IF NOT EXISTS `t_reservations` (
  `id` int NOT NULL AUTO_INCREMENT,
  `reservation_number` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,
  `start_date` date NOT NULL,
  `end_date` date NOT NULL,
  `reservation_status` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,
  `reservation_car_code` varchar(255) CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci NOT NULL,
  `car_quantity` bigint NOT NULL,
  `total_cost` decimal(10,2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

1.11. Insert data into the Reservation table.

```
INSERT INTO `t_reservations` (`id`, `reservation_number`,
`start_date`, `end_date`, `reservation_status`,
`reservation_car_code`, `car_quantity`, `total_cost`) VALUES
      (1, 'd2b42c10-6f31-4604-be5e-8cd146cb2411', '2023-09-13',
'2023-09-21', '2', '45f6f72c-6f6a-4b36-9d33-954ccb62be0a', 2,
888.96),
      (2, '1ca01598-c2bd-4885-a5ab-69c8dc3aab62', '2023-09-06',
'2023-09-21', '0', '45f6f72c-6f6a-4b36-9d33-954ccb62be0a', 1,
1666.80);
```

1.12. Change localhost to public.

```
SELECT user, host FROM mysql.user;
UPDATE mysql.user SET Host = '%' WHERE User = 'root';
exit;
```

and make sure to restart the database.

```
sudo systemctl restart mysql
```

1.13. Again, Login to the MySQL DB to change the root password.

```
mysql -u root -p
```

```
(no need to type password just enter)
```

```
ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY
'demoadmin';
```

```
exit;
```

and make sure to restart the database.

Run the following command.

```
sudo systemctl restart mysql
```

1.14. Change my SQL configuration's exiting bind address.

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

and change the bind-address as follow,

```
bind-address  -------->    0.0.0.0
```

and make sure to restart the database.

Run the following command.

```
sudo systemctl restart mysql
```

2. SETTING UP THE MICROSERVICES

2.1. Configure Microservices (sprint boot)

2.2. Install Java package with below command.

```
sudo apt update

sudo apt install openjdk-8-jdk

java -version
```

2.3. Download the car reservation application jar files.

Create Common Directory to keep microservices.

```
mkdir DevOpsLabApp
```

Change the directory.

```
cd DevOpsLabApp
```

Clone DevOps_Lab_App into DevOpsLabApp Directory

```
git clone https://github.com/Milinda96/DevOps_Lab_App.git
```

2.4 Create each micro service under DevOpsLabApp directory.

2.4.1 Discovery Service

```
mkdir discovery-service

cd discovery-service
```

Copy the application.properties file into the discovery service directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/discovery-service/src/main/resources/application.properties /home/ubuntu/DevOpsLabApp/discovery-service/
```

Open the application properties with following command.

```
nano application.properties
```

Inside that edit the below line and save it

```
eureka.instance.hostname=<your-ec2-instance-public ip>
```

copy the *discovery service jar* file into the *discovery service* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/discovery-service/target/discovery-service-0.0.1-SNAPSHOT.jar /home/ubuntu/DevOpsLabApp/discovery-service/
```

Run the discovery service.

```
nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java_pid.hprof -jar /home/ubuntu/DevOpsLabApp/discovery-service/discovery-service-0.0.1-SNAPSHOT.jar > discovery-service.log 2>&1 &
```

2.4.2 API Gateway

Create api-gateway directory.

```
mkdir api-gateway
```

Change the directory.

```
cd api-gateway
```

Copy the *application.properties* file into the *api-gateway* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/api-gateway/src/main/resources/application.properties /home/ubuntu/DevOpsLabApp/api-gateway/
```

Open the application properties with following command.

```
nano application.properties
```

Inside that edit the below line and save it

```
eureka.instance.client.serviceUrl.defaultZone=http://<your-ec2-instance-ip>:8761/eureka/

spring.cloud.gateway.routes[0].uri=http://<your-ec2-instance-ip>:9001

spring.cloud.gateway.routes[1].uri=http://<your-ec2-instance-ip>:9003

spring.cloud.gateway.routes[2].uri=http://<your-ec2-instance-ip>:9002

spring.cloud.gateway.routes[3].uri=http://<your-ec2-instance-ip>:8761

spring.cloud.gateway.routes[4].uri=http://<your-ec2-instance-ip>:8761
```

copy the *api-gateway* jar file into the *api gateway* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/api-gateway/target/api-gateway-0.0.1-SNAPSHOT.jar /home/ubuntu/DevOpsLabApp/api-gateway/
```

Run the api gateway service.

```
nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java_pid.hprof -jar /home/ubuntu/DevOpsLabApp/api-gateway/api-gateway-0.0.1-SNAPSHOT.jar > api-gateway.log 2>&1 &
```

2.4.3 Location Service

Create Location service directory.

```
mkdir location-service
```

Change Location service directory.

```
cd location-service
```

copy the *application.properties* file into the *Locations service* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/location-service/src/main/resources/application.properties /home/ubuntu/DevOpsLabApp/location-service/
```

Open the application properties with following command.

```
nano application.properties
```

after that edit, the file

```
spring.datasource.url=jdbc:mysql://<your-db-ec2-public-ip>:3306/car_rental_db
```

```
eureka.instance.client.serviceUrl.defaultZone=http://<your-ec2-public-ip>:8761/eureka/
```

and save it.

Copy the *location service* jar file into the *location service* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/location-service/target/location-service-0.0.1-SNAPSHOT.jar /home/ubuntu/DevOpsLabApp/location-service/
```

Run the Location microservice.

```
nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java_pid.hprof -jar /home/ubuntu/DevOpsLabApp/location-service/location-service-0.0.1-SNAPSHOT.jar > location.log 2>&1 &
```

2.4.4 Car service

Create Car Service directory.

```
mkdir car-service
```

Change Car service directory.

```
cd car-service
```

Copy the *application.properties* file into the *car service* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/car-service/src/main/resources/application.properties /home/ubuntu/DevOpsLabApp/car-service/
```

Open the application properties with following command.

```
nano application.properties
```

after that edit, the file

```
spring.datasource.url=jdbc:mysql://<your-db-ec2-public-ip>:3306/car_rental_db

eureka.instance.client.serviceUrl.defaultZone=http://<your-ec2-public-ip>:8761/eureka/

location.api.url=_http://<your-ec2-public-ip>:9003
```

save it.

copy the *car-service* jar file into the *car-service* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-services/car-service/target/car-service-0.0.1-SNAPSHOT.jar /home/ubuntu/DevOpsLabApp/car-service/
```

Run car-service microservice.

```
nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java_pid.hprof -jar /home/ubuntu/DevOpsLabApp/car-service/car-service-0.0.1-SNAPSHOT.jar > car-service.log 2>&1 &
```

2.4.5 Reservation

Create the reservation-service directory.

```
mkdir reservation-service
```

Change directory.

```
cd reservation-service
```

Copy the *application.properties* file into the *reservation* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-
services/reservation-
service/src/main/resources/application.properties
/home/ubuntu/DevOpsLabApp/reservation-service/
```

Open the application properties with following command.

```
nano application.properties
```

after that edit, the file

```
spring.datasource.url=jdbc:mysql://<your-db-ec2-public -
ip>:3306/car_rental_db
```

```
eureka.instance.client.serviceUrl.defaultZone=http://<your-ec2-
public -ip>:8761/eureka/
```

Copy the *reservation* jar file into the *reservation* directory.

```
cp /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-
services/reservation-service/target/reservation-service-0.0.1-
SNAPSHOT.jar /home/ubuntu/DevOpsLabApp/reservation-service/
```

Save it.

Run the reservation microservice.

```
nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=./java_pid.hprof -jar
/home/ubuntu/DevOpsLabApp/reservation-service/reservation-
service-0.0.1-SNAPSHOT.jar > reservation.log 2>&1 &
```

3. SETTING UP THE UI

3.1. Create Common Directory to keep Lab App.

```
mkdir /home/ubuntu/DevOpsLabApp
```

Change the directory.

```
cd AIOpsLab
```

3.2. Clone DevOps_Lab_App into DevOpsLabApp Directory.

```
git clone https://github.com/Milinda96/DevOps_Lab_App.git
```

3.3. Install Apache package with below command.

```
sudo apt update

sudo apt install apache2

sudo systemctl status apache2
```

3.4. Create directory in Apache server.

```
cd /var/www/html/

mkdir carrental
```

3. 5. Copy UI build files into Apache server

```
cd /home/ubuntu/DevOpsLabApp/DevOps_Lab_App/devops-lab-app-
client/dist/car-booking-app/


cp -r * /var/www/html/carrental
```

If you want to change the IP address belonging to the API-gateway service where defined in the environment.prod.ts file after the deployment, please follow the instructions as described.

Open the .js file name starting with **main.**

Ex: main.839c7f0432b827dca771.js

Open the vim editor.

```
vim main.839c7f0432b827dca771.js
```

Search for a http://<your-earlier-api-gateway-ip-address>:9004

Note:

1. To search for an IP address 1st, type / and your IP address, and then it will highlight.
2. Next, do Enter.
3. Then enter the insert button and you will be able to edit the IP address with the new IP address as you want.
4. After that enter the ESC button and type :wq! and do Enter.