***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.***

**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 in to 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 local host 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*](https://github.com/Milinda96/DevOps_Lab_App.git)

**2.4 Create each micro services 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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/discovery-service/src/main/resources/application.properties /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/discovery-service/target/discovery-service-0.0.1-SNAPSHOT.jar /home/ubuntu/AIOpsLab/discovery-service/

cp discovery-service-0.0.1-SNAPSHOT\_BK.jar /home/ubuntu/AIOpsLab/discovery-service/

**Enable the discovery service**

nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java\_pid.hprof -jar /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/api-gateway/src/main/resources/application.properties /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/api-gateway/target/ api-gateway-0.0.1-SNAPSHOT.jar /home/ubuntu/AIOpsLab/api-gateway/

**Enable the api gateway service**

*nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java\_pid.hprof -jar /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/location-service/src/main/resources/application.properties /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/location-service/target/location-service-0.0.1-SNAPSHOT.jar /home/ubuntu/AIOpsLab/location-service/*

**Enable the Location\_Service microservice**

*nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java\_pid.hprof -jar /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/car-service/src/main/resources/application.properties /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/car-service/target/car-service-0.0.1-SNAPSHOT.jar /home/ubuntu/AIOpsLab/car-service/

**Enable car-service microservices**

*nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java\_pid.hprof -jar /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/reservation-service/src/main/resources/application.properties /home/ubuntu/AIOpsLab/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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-services/reservation-service/target/reservation-service-0.0.1-SNAPSHOT.jar /home/ubuntu/AIOpsLab/reservation-service/

**Save it**

**Enable the reservation microservices**

*nohup java -Xms256m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./java\_pid.hprof -jar /home/ubuntu/AIOpsLab/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/AIOpsLab Change the directory

cd AIOpsLab

**3.2 Clone AIOPS Hackathon\_Lab\_Application into AIOpsLab Directory**

*git clone* [*https://github.com/Milinda96/AIOPs\_Hackathon\_Lab\_Application.git*](https://github.com/Milinda96/AIOPs_Hackathon_Lab_Application.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/AIOpsLab/AIOPs\_Hackathon\_Lab\_Application/aiops-hackathon-client/

Based on your team please use the relevant command to copy build-files to Apache server

cp -r AWS\_LAB\_NewRelic\_Front/\* /var/www/html/carrental

cp -r AWS\_LAB\_Dyanatrace\_Frontend/\* /var/www/html/carrental

cp -r AWS\_LAB\_Datadog\_frontend/\* /var/www/html/carrental

cp -r AIOps\_LAB\_Splunk\_Frontend/\* /var/www/html/carrental