


JAVA FSD PHASE 4 PROJECT- Integration and Deployment related Screenshots.

Name: Syed Arfaat

Infrastructure Overview

Setting up AWS: Hosting platform that provides scalable infrastructure, leveraging services like EC2 for computation, S3 for storage, and RDS for database management to ensure reliability and performance.



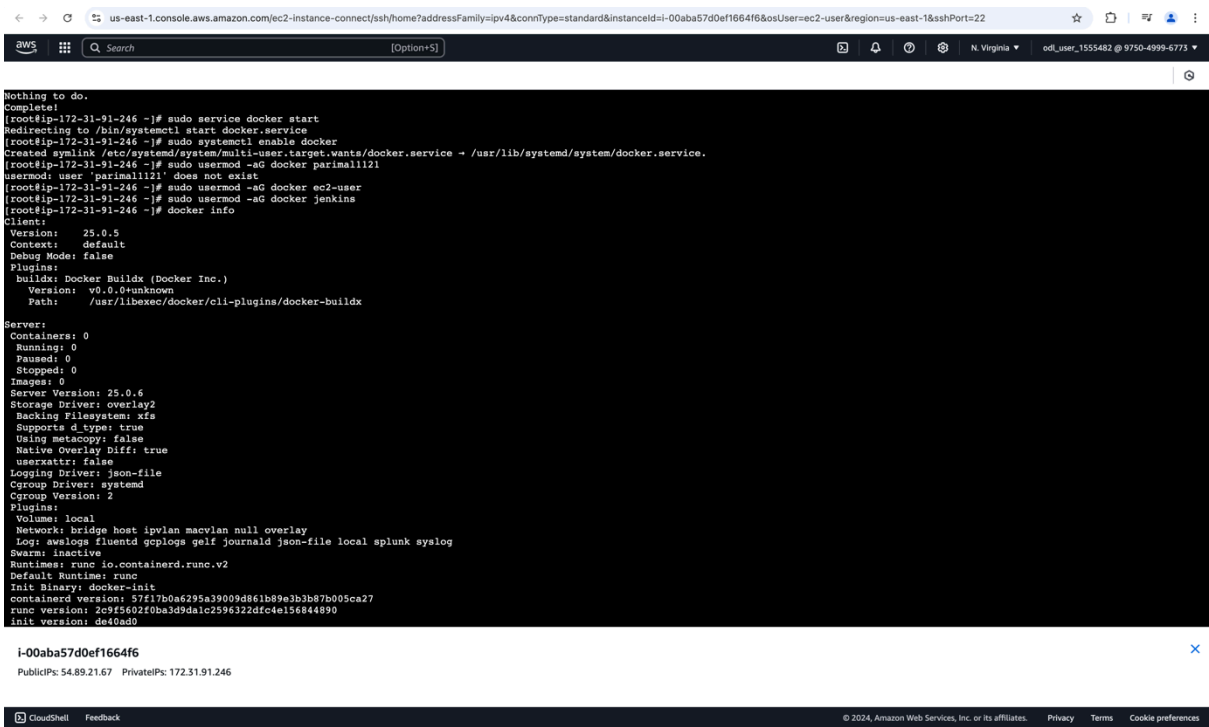
```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Dec 30 08:38:12 2024 from 18.206.107.27
[ec2-user@ip-172-31-91-246 ~]$ sudo -i
[root@ip-172-31-91-246 ~]# sudo systemctl start jenkins
[root@ip-172-31-91-246 ~]# sudo systemctl enable --now jenkins
[root@ip-172-31-91-246 ~]# pwd
/root
[root@ip-172-31-91-246 ~]# etc/
-bash: etc/: No such file or directory
[root@ip-172-31-91-246 ~]# /etc
-bash: /etc: is a directory
[root@ip-172-31-91-246 ~]# /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
[root@ip-172-31-91-246 ~]# chmod 440 /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
[root@ip-172-31-91-246 ~]# vi /var/lib/jenkins/secrets/initialAdminPassword
[root@ip-172-31-91-246 ~]#
```

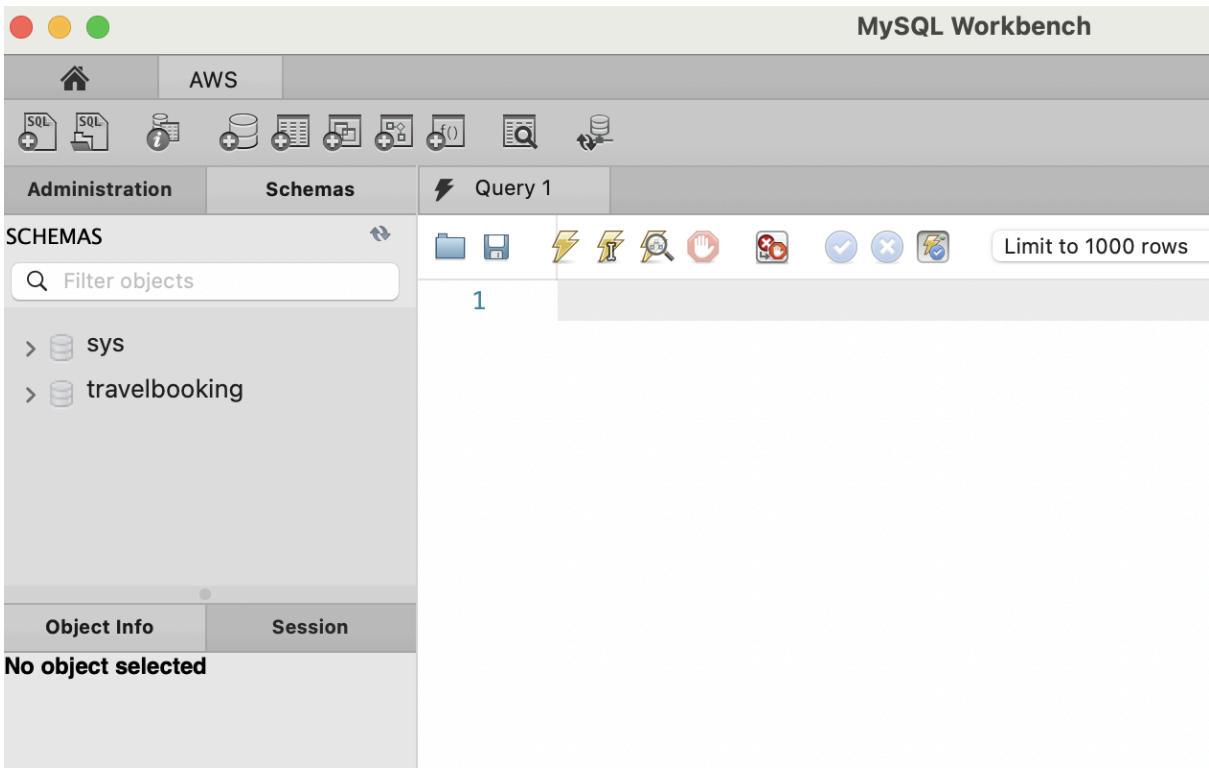
i-00aba57d0ef1664f6
PublicIPs: 54.89.21.67 PrivateIPs: 172.31.91.246

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Setting up Docker containers and images in AWS: Enables microservice architecture by running application components in isolated containers, ensuring environment consistency across development and production.



Creating MySQL database on AWS: Manages user, cab, and booking data with a highly available and scalable RDS-hosted MySQL database.



Automation and Deployment

Jenkins in AWS: Automates CI/CD pipelines, ensuring that code changes are built, tested, and deployed seamlessly. Jenkins fetches code from repositories and deploys it to AWS environments, facilitating faster delivery.

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `ec2-52-54-239-250.compute-1.amazonaws.com:8080/job/TravelBook/configure`. The page title is "TravelBook Config [Jenkins]". The breadcrumb navigation shows "Dashboard > TravelBook > Configuration".

The main section is titled "Configure". On the left, there are three tabs: "General" (selected), "Advanced Project Options", and "Pipeline".

Under the "General" tab, there are two checkboxes: "Quiet period" and "Trigger builds remotely (e.g., from scripts)".

The "Advanced Project Options" section is expanded, showing a dropdown menu set to "Advanced".

The "Pipeline" section is visible, with a "Definition" dropdown set to "Pipeline script".

The "Script" section contains a text area with the following Groovy code:

```
1 pipeline {
2   agent any
3
4   environment {
5     SPRING_DATASOURCE_URL = "jdbc:mysql://mydb.czysui4e2ib9.us-east-1.rds.amazonaws.com:3306/mydb"
6   }
7
8   stages {
9     stage('Clone Repository') {
10      steps {
11        git 'https://github.com/Parimal11/TravelBook.git'
12      }
13    }
14
15    stage('Build with Maven') {
```

Below the script area, there is a checkbox labeled "Use Groovy Sandbox" which is checked. A link "Pipeline Syntax" is also present.

At the bottom, there are two buttons: "Save" and "Apply".

The footer of the page shows "REST API" and "Jenkins 2.479.2".

API Endpoints

User Management

- **PostMapping: Users: Registers new users by accepting and validating user details like name, email, and contact information.**
- **GetMapping: Users: Retrieves registered user details, including role-based filtering for administrative tasks.**

Cab Management

PostMapping: Cabs: Adds new cabs, allowing admins to register details like model, license number, and driver information.

The screenshot displays a REST client interface with a list of requests at the top. The selected request is a POST to `http://localhost:8080/api/users/bookings`. Below the request bar, the 'Body' tab is active, showing a JSON payload: `{ "username": "user1", "password": "user1234", "role": "USER" }`. The response section at the bottom shows a 201 status code with a response time of 214 ms and a body size of 337 B. The response body is displayed in 'Pretty' JSON format: `{ "id": 4, "username": "user1", "password": "user1234", "role": "USER", "bookings": null }`.

```
POST http://localhost:8080/api/users/bookings
```

Save

POST http://localhost:8080/api/users/users Send

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

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

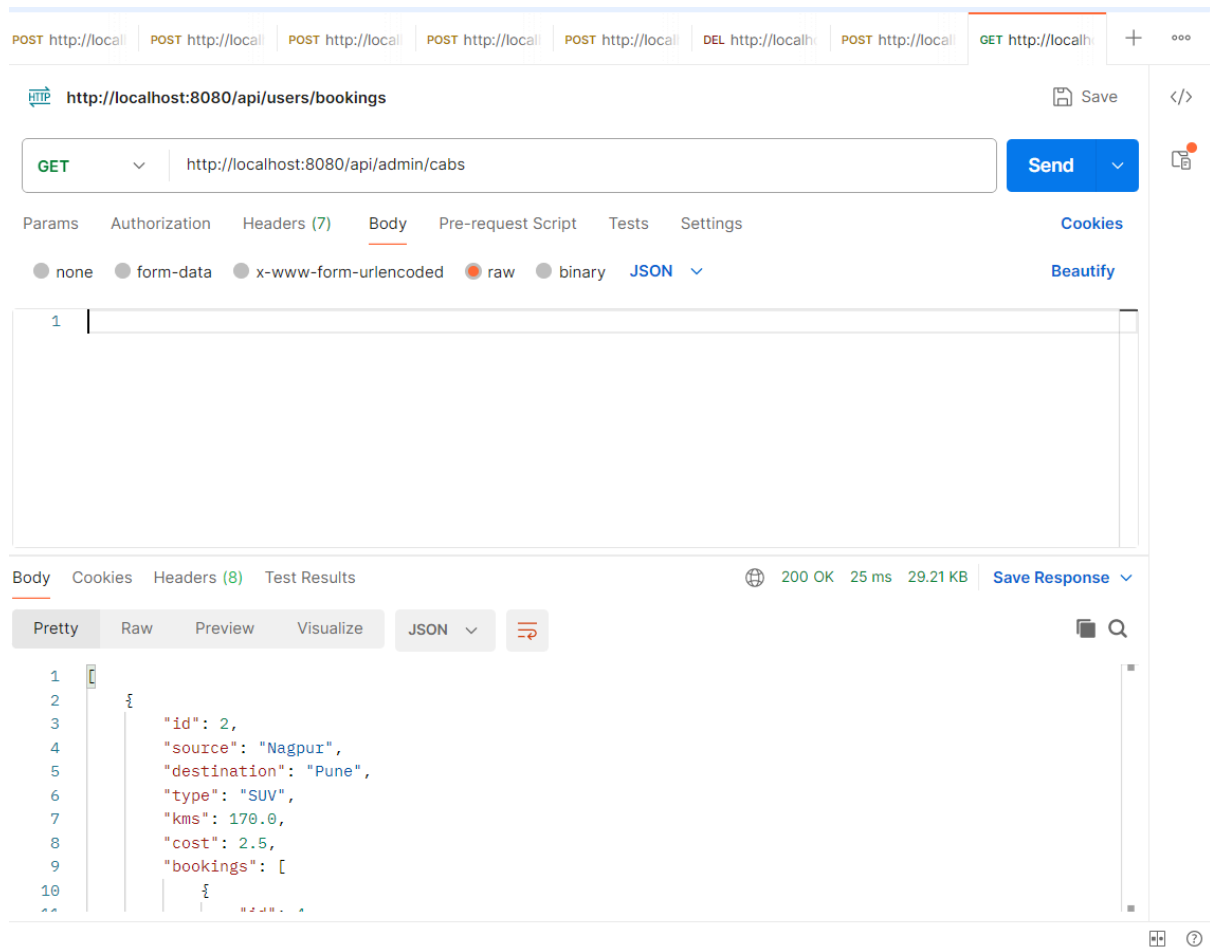
```
1 {
2   "username": "user1",
3   "password": "user1234",
4   "role": "USER"
5 }
```

Body Cookies Headers (8) Test Results 201 Created 214 ms 337 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 4,
3   "username": "user1",
4   "password": "user1234",
5   "role": "USER",
6   "bookings": null
7 }
```

GetMapping: Cabs: Fetches available cab details with optional filters for type, availability, and location.



DeleteMapping: Cabs: Removes decommissioned or unavailable cabs from the system.

DELETE http://localhost:8080/api/admin/cabs/3

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

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

```
1 {
2   "id": 2,
3   "source": "Nagpur",
4   "destination": "Akola",
5   "type": "SUV",
6   "kms": 170.0,
7   "cost": 2.5
8 }
```

Body Cookies Headers (6) Test Results 204 No Content 23 ms 201 B Save Response

Pretty Raw Preview Visualize Text

1

Booking Management

- **GetMapping: Booking:** Retrieves booking details based on criteria like user ID or booking status.

GET http://localhost:8080/api/admin/bookings

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

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (8) Test Results 200 OK 1346 ms 29.19 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 4,
3   "user": {
4     "id": 3,
5     "username": "abc",
6     "password": "abc123",
7     "role": "USER",
8     "bookings": [
9     ]
10  }
```

PutMapping: Bookings: Updates booking details such as trip schedules or status modifications.

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/api/admin/bookings/7`. The request body is a JSON object representing a booking with the following details:

```
{  "id": 3,  "cab": {    "id": 2,    "status": "BOOKED",    "pickup_location": "Nagpur",    "drop_location": "Akola",    "fare": 425  }}
```

The response status is 200 OK, and the response body is a JSON object representing the updated booking:

```
{  "id": 7,  "user": {    "id": 3,    "username": null,    "password": null,    "role": null,    "bookings": null  },  "cab": {    "id": 2,    "status": "BOOKED",    "pickup_location": "Nagpur",    "drop_location": "Akola",    "fare": 425  }}
```

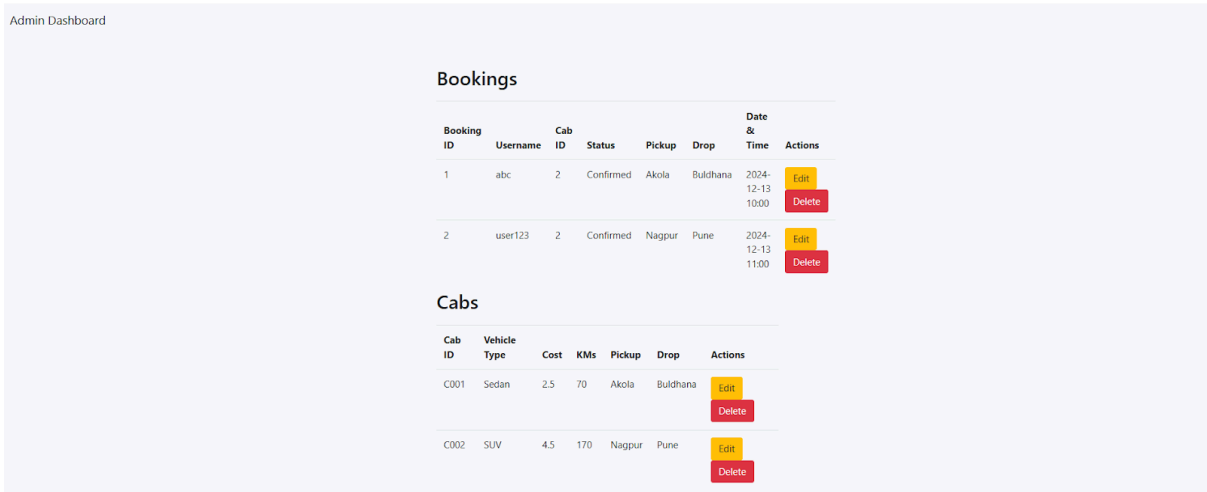
- **DeleteMapping: Bookings:** Deletes outdated or invalid bookings to maintain system accuracy.

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8080/api/admin/bookings/4`. The response status is 204 No Content, indicating that the booking has been successfully deleted.

Frontend Features

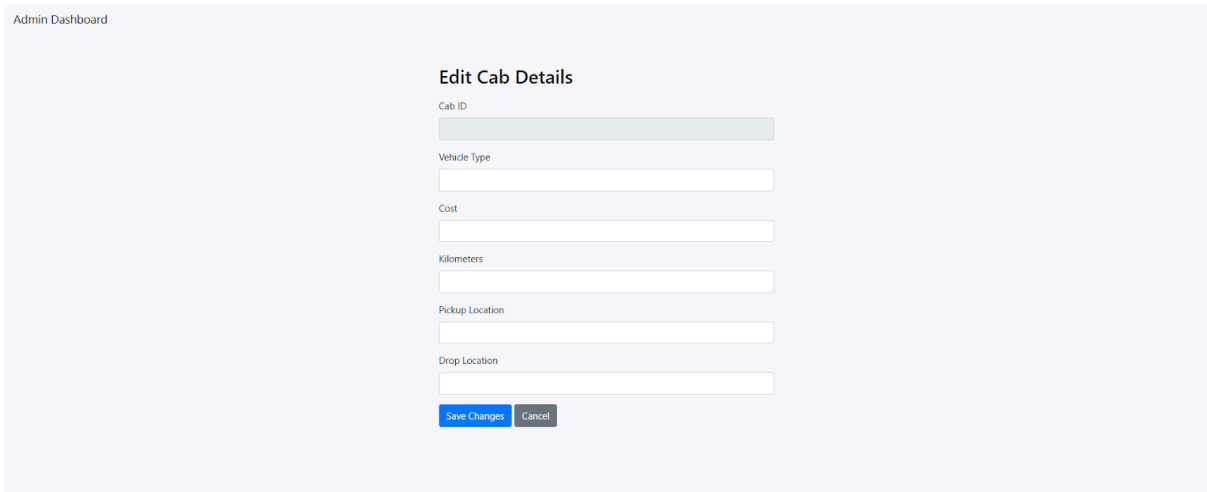
Admin Dashboard

- **A centralized control panel for managing users, cabs, and bookings. It includes options to add, edit, or delete resources.**



Cab Operations

- **Edit Cab: Updates cab information, including driver assignments and availability.**



- **Delete Cabs:** Simplifies the removal of cabs directly from the admin interface.

Cabs

Cab ID	Vehicle Type	Cost	KMs	Pickup	Drop	Actions
C001	Sedan	2.5	70	Akola	Buldhana	<div>EditDelete</div>

Booking Operations

- **Edit Booking:** Allows changes to trip schedules, drivers, or booking statuses.

Admin Dashboard

Edit Booking Details

Booking ID

Username

Cab ID

Booking Status

Confirmed

Pickup Location

Drop Location

Date & Time

dd-mm-yyyy --:--

Save ChangesCancel

- **Delete Bookings:** Streamlines the process of removing invalid bookings via the admin interface.

Admin Dashboard

Bookings

Booking ID	Username	Cab ID	Status	Pickup	Drop	Date & Time	Actions
1	abc	2	Confirmed	Akola	Buldhana	2024-12-13 10:00	<div>EditDelete</div>

User Dashboard

- **Provides customers with tools to search for available cabs and book rides effortlessly.**

User Dashboard

Search for a Cab

Pickup Location

Drop Location

[Search](#)

Authentication

- **Login Page: Ensures secure access for users and admins, with role-based permissions and authentication.**

Cab Booking Home User Login

Welcome to Cab Booking System

Please login as an admin or user to continue.

[Admin Login](#)

Username:

Password:

[User Login](#)

Cab Booking

- **Simplifies the cab selection and booking process with user-friendly features.**

127.0.0.1:5500/user.html

Your Orders Arrays Introduction... is responsible Project File

All Bookmarks

User Dashboard

127.0.0.1:5500 says
Cab with ID 1 has been booked!

[OK](#)

Pickup Location

Drop Location

[Search](#)

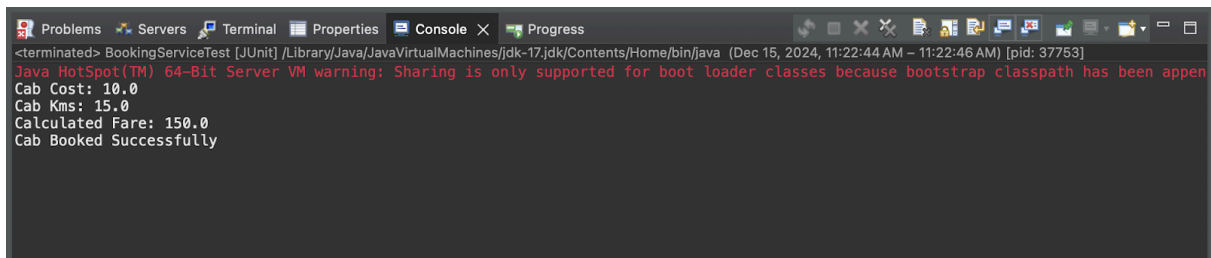
Sedan

Fare: Rs475

[Book Cab](#)

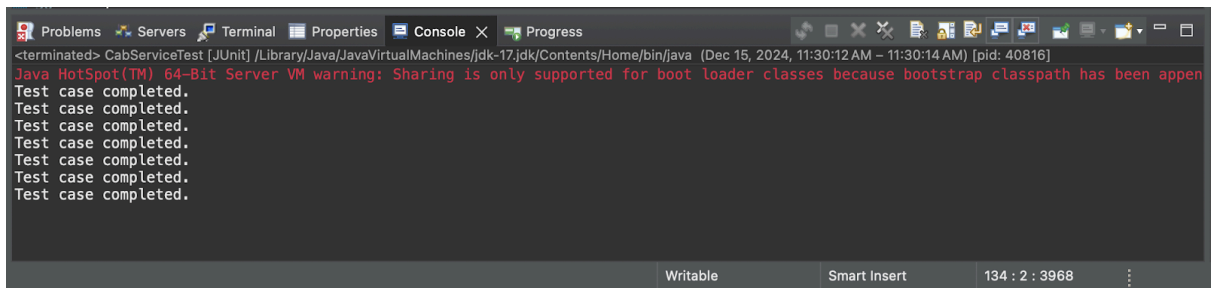
JUnit Testing

- **BookingService JUnit:** Ensures reliability by testing booking creation, updates, and deletions.



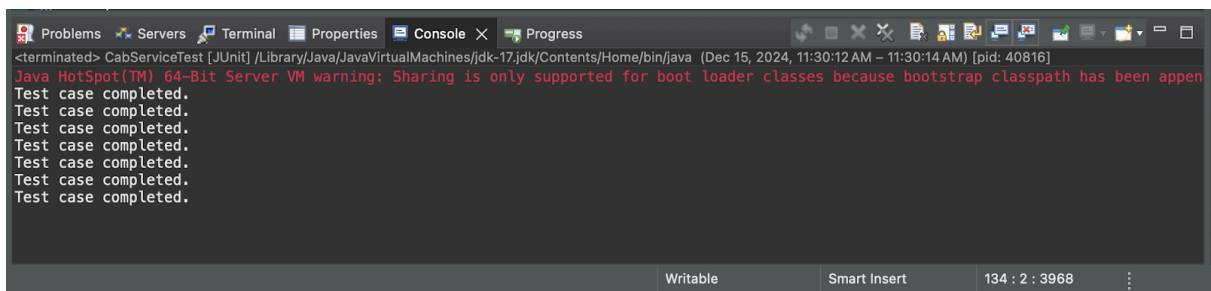
```
<terminated> BookingServiceTest [JUnit] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 15, 2024, 11:22:44 AM ~ 11:22:46 AM) [pid: 37753]
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Cab Cost: 10.0
Cab Kms: 15.0
Calculated Fare: 150.0
Cab Booked Successfully
```

- **CabService JUnit:** Validates functionality for cab-related operations, including registration and availability checks.



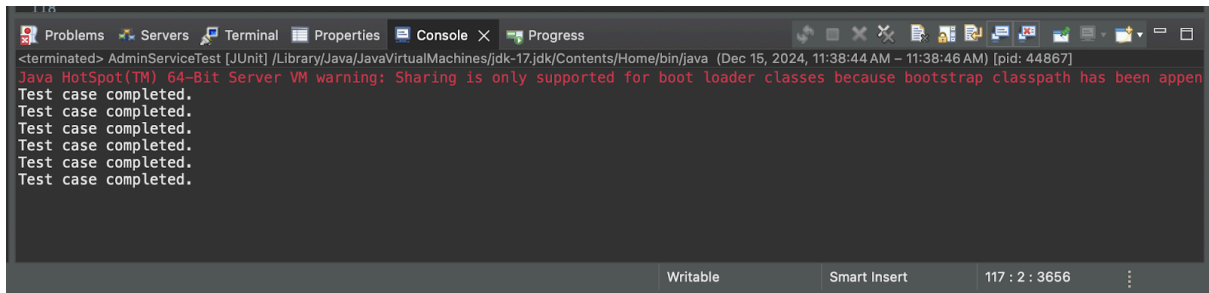
```
<terminated> CabServiceTest [JUnit] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 15, 2024, 11:30:12 AM ~ 11:30:14 AM) [pid: 40816]
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
```

- **UserService JUnit:** Tests for user registration, authentication, and profile updates.



```
<terminated> CabServiceTest [JUnit] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 15, 2024, 11:30:12 AM ~ 11:30:14 AM) [pid: 40816]
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
Test case completed.
```

- **AdminService JUnit:** Covers admin-specific tasks like managing users, cabs, and bookings.



Deployment Workflow

- **Jenkins Deployment:** Automates end-to-end deployment from code fetch to live updates in AWS environments, reducing manual intervention and improving efficiency.

TravelBook Config [Jenkins]

Not Secure ec2-52-54-239-250.compute-1.amazonaws.com:8080/job/TravelBook/configure

Incognito

Dashboard > TravelBook > Configuration

Configure

General

Advanced Project Options

Pipeline

☐ Quiet period ?

☐ Trigger builds remotely (e.g., from scripts) ?

Advanced Project Options

Advanced

Pipeline

Definition

Pipeline script

Script ?

try sample Pipeline...

```
1 pipeline {
2   agent any
3
4   environment {
5     SPRING_DATASOURCE_URL = "jdbc:mysql://mydb.czysui4e2ib9.us-east-1.rds.amazonaws.com:3306/mydb"
6   }
7
8   stages {
9     stage('Clone Repository') {
10      steps {
11        git 'https://github.com/Parimal11/TravelBook.git'
12      }
13    }
14
15    stage('Build with Maven') {
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply

REST API

Jenkins 2.479.2