

---

## Task: Deploy a Java + MySQL Web Application on Tomcat using AWS RDS

### Objective

You'll deploy a Java-based login web application connected to a MySQL database hosted on **AWS RDS**.

This task will test your ability to integrate backend code, configure databases, and deploy a complete app to a web server.

---

### Step-by-Step Task Instructions

#### **1** Clone the project

Clone the Java Web App repository from GitHub:

```
git clone https://github.com/mrtechreddy/aws-rds-java.git
cd LoginWebApp
```

#### **2** Review the code structure

You'll see:

```
LoginWebApp/
├── pom.xml
├── src/
│   ├── main/
│   │   └── webapp/
│   │       ├── WEB-INF/web.xml
│   │       ├── index.jsp
│   │       └── login.jsp
```

```
|— register.jsp
|— userRegistration.jsp
|— success.jsp
|— welcome.jsp
```

---

### 3 Create a MySQL database on AWS RDS

a. Log in to your AWS Management Console

b. Navigate to: **RDS** → **Databases** → **Create database**

c. Choose:

- Engine: **MySQL 8.x**
- Template: **Free tier** (if applicable)
- DB identifier: **jwt**
- Username: **admin**
- Password: **admin123**
- Public access: ☒ **Enable**
- VPC security group: Allow inbound **3306** from your system or EC2 instance IP.

d. After creation, note your RDS endpoint — it looks like:

ai-db.crno3nsbxzen.us-west-1.rds.amazonaws.com

---

### 4 Create the required table

Connect to RDS MySQL via CLI:

```
mysql -h <your-rds-endpoint> -u admin -p
```

Then run:

```
CREATE DATABASE jwt;
```

```
USE jwt;
```

```
CREATE TABLE `USER` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `first_name` VARCHAR(100) NOT NULL,  
  `last_name` VARCHAR(100) NOT NULL,  
  `email` VARCHAR(150) NOT NULL,  
  `username` VARCHAR(100) NOT NULL,  
  `password` VARCHAR(255) NOT NULL,  
  `regdate` DATE NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `unique_username` (`username`),  
  UNIQUE KEY `unique_email` (`email`)  
) ENGINE=InnoDB  
  DEFAULT CHARSET=utf8mb4  
  COLLATE=utf8mb4_unicode_ci;
```

---

## 5 Update database credentials in the code

Edit both **login.jsp** and **userRegistration.jsp** to replace:

```
String jdbcURL =  
"jdbc:mysql://<your-rds-endpoint>:3306/jwt?useSSL=false&allowPublicKeyRetrieval=t  
rue&serverTimezone=UTC";  
String dbUser = "admin";  
String dbPass = "admin123";
```

with your **actual RDS endpoint**.

---

## **6 Build the WAR file using Maven**

From the project root, run:

```
mvn clean package
```

After a successful build, you'll get:

```
target/LoginWebApp.war
```

---

## **7 Deploy to Apache Tomcat**

**a.** Copy the WAR file to your Tomcat webapps directory:

```
sudo cp target/LoginWebApp.war /opt/tomcat/webapps/
```

**b.** Restart Tomcat:

```
sudo systemctl restart tomcat
```

or if running manually:

```
/opt/tomcat/bin/shutdown.sh
```

```
/opt/tomcat/bin/startup.sh
```

**c.** Access the app in your browser:

```
http://<EC2-public-ip>:8080/LoginWebApp/
```

---

## 8 Test your deployment

- Open the app → Register a new user
- Verify the data in your RDS table
- Then log in using the same credentials
- Confirm successful redirection to [success.jsp](#) or [welcome.jsp](#)

---



## Bonus Challenges



1. Secure your DB credentials using environment variables instead of hardcoding.
2. Deploy Tomcat on an EC2 instance inside the same VPC as RDS for secure connectivity.
3. Add SSL to your Tomcat app using Nginx reverse proxy on port 443.
4. Automate the deployment using Jenkins or GitHub Actions.

---

## Deliverables

When done, submit:

-  Screenshot of your app's login page running on Tomcat
-  Screenshot of your user data in AWS RDS

-  Screenshot of successful registration or login
-  Your GitHub repo link with the working code and DB integration