Spring Boot Deployment Guide

Deploying a Spring Boot Application on AWS EC2 with MySQL and HTTPS

Abdul-Aziz Mohammed

01-04-2025

Contents

1. Launch EC2 Instance (Ubuntu Server)

```
# Connect to EC2 Instance
ssh -i crud_app_key.pem ubuntu@<EC2_PUBLIC_IP>

# Switch User to Root
sudo -i

# Update packages
sudo apt update && sudo apt upgrade -y

# Install Java (for Spring Boot)
sudo apt install openjdk-21-jdk -y

# Check Java version
java -version
```

---

2. Install and Configure MySQL on EC2

```
# Install MySQL server
sudo apt install mysql-server -y

# Run secure installation
sudo mysql_secure_installation
```

Login to MySQL

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

Create Database and User

```
CREATE DATABASE crud_app_db;
CREATE USER 'spring_user'@'%' IDENTIFIED BY 'Aws!54321';
GRANT ALL PRIVILEGES ON crud_app_db.* TO 'spring_user'@'%';
FLUSH PRIVILEGES;
EXIT;
```

---

3. Import Local Database into EC2 MySQL

On your **local machine**:

```
# Export database to a file
mysqldump -u root -p my_local_db > my_local_db.sql

# Copy dump file to EC2
scp -i aws-key.pem my_local_db.sql ubuntu@<EC2_PUBLIC_IP>:/tmp/
```

On your **EC2 instance**:

```
# Import into MySQL
mysql -u spring_user -p crud_app_db < /tmp/my_local_db.sql

# Log in as root
sudo mysql -u root -p

# Show Databases
show databases;

# Access a Database
use enter_database_name;

# List Tables
show tables;
```

---

4. Transfer Spring Boot JAR to EC2

On local:

```
scp -i crud_app_key.pem /path/to/CrudApp-0.0.1-SNAPSHOT.jar ubuntu@<EC2_PUBLIC_IP>:/home/ubuntu/
```

On EC2:

```
sudo mkdir -p /opt/crudservice
sudo mv /home/ubuntu/CrudApp-0.0.1-SNAPSHOT.jar /opt/crudservice/
```

---

5. Create Application Properties File

```
sudo nano /opt/microservice/application_prod.properties
```

Paste:

```
spring.application.name=CrudApp

server.servlet.context-path=/crud-app
```

```
spring.thymeleaf.cache=false
spring.main.allow-circular-references=true

spring.datasource.url=jdbc:mysql://localhost:3306/crud_app_db?serverTimezone=UTC
spring.datasource.username=spring_user
spring.datasource.password=Aws!54321

spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.show-sql=true

spring.mvc.format.date=yyyy-MM-dd
spring.mvc.format.date-time=yyyy-MM-dd'T'HH:mm:ss
spring.mvc.format.time=HH:mm:ss
```

---

6. Create Systemd Service

```
sudo nano /etc/systemd/system/crudapp.service
```

Paste:

```
[Unit]
Description=Spring Boot CRUD App
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/opt/crudservice
ExecStart=/usr/bin/java -jar /opt/crudservice/CrudApp-0.0.1-SNAPSHOT.jar --spring.config.location=
                  /opt/crudservice/application_prod.properties
SuccessExitStatus=143
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Enable and start:

```
sudo systemctl daemon-reload
sudo systemctl enable crudapp.service
sudo systemctl start crudapp.service
sudo systemctl status crudapp.service
```

---

7. Install and Configure Nginx

```
sudo apt install nginx -y
```

Configure reverse proxy:

```
sudo nano /etc/nginx/sites-available/default
```

Paste:

```
server {
    listen 80;
    server_name your-domain.com;

    location / {
```

```
        proxy_pass http://localhost:8080/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Enable site:

```
sudo nginx -t
sudo systemctl restart nginx
```

---

8. Setup HTTPS with Let's Encrypt

```
sudo apt install certbot python3-certbot-nginx -y
```

Request certificate:

```
sudo certbot --nginx -d your-domain.com
```

Auto-renew test:

```
sudo certbot renew --dry-run
```

---

9. Final Check

- App should be available at: `https://your-domain.com`
- Service check:

```
sudo systemctl status employee
```