

UB Social

API REST Spring Boot

9. Deploy na AWS





Instância EC2

- EC2 / Instâncias / Executar instâncias
 - Nome: prjCursoSpringBoot1
 - AMI: Ubuntu Server (nível gratuito)
 - Tipo: t3.micro (verificar se região é t2 ou t3 nível gratuito)
 - Criar par de chaves:
 - Nome: keySpringBoot1
 - Tipo: RSA
 - Formato: conforme SO de sua máquina
 - Criar par de chaves (download .pem ou .ppk)
 - Rede:
 - Atribuir IP público automaticamente: habilitar
 - Grupo de segurança criado automaticamente, com regras:
 - Permitir tráfego HTTP qualquer lugar
 - Permitir tráfego HTTPS qualquer lugar
 - Permitir tráfego SSH qualquer lugar
 - Armazenamento: 8GiB gp3 (nível gratuito)
 - Criar (executar instância)



Habilitar porta Spring Boot

- EC2 / Instâncias / Clicar na instância criada
- Aba Segurança / Grupo de segurança
 - Editar regras de entrada
 - Adicionar regra:
 - Tipo: TCP personalizado, Protocolo: TCP, Intervalo de portas: 8080, Origem: Qualquer-local-IPv4 (0.0.0.0/0), Descrição: API REST Spring Boot
- Salvar regras



Conectar instância

- EC2 / Instâncias / Clicar na instância criada / Conectar:
 - Habilitar Conecte-se usando um IP público
- Conectar



Instalar API REST Spring Boot

- Na instância CLI, executar comandos:
 - `sudo apt update && sudo apt install git curl openjdk-21-jdk mysql-server -y && sudo systemctl enable mysql --now`
 - `sudo mysql`
 - `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '123';`
 - `FLUSH PRIVILEGES;`
 - Ctrl+d (sair do MySQL)
 - `mysql -u root -p` (informar senha 123)
 - `CREATE DATABASE bookstore;`
 - `SHOW DATABASES;`
 - Ctrl+d (sair do MySQL)
 - `git clone https://github.com/mateusschwede/workover_spring_boot_api_rest.git && cd workover_spring_boot_api_rest && cd bookstore && sudo chmod +x gradlew`
 - `./gradlew clean build --refresh-dependencies`
 - `./gradlew bootRun &`



Consumir API REST Spring Boot - cURL

- Listar books: curl http://localhost:8080/books
- Adicionar book:

```
curl -X POST http://localhost:8080/books \
```

```
-H "Content-Type: application/json" \
```

```
-d '{
```

```
  "title": "O Senhor dos Anéis",
```

```
  "author": "J.R.R. Tolkien",
```

```
  "published_date": "1954-07-29",
```

```
  "isbn": "9780261102385",
```

```
  "pages": 1178,
```

```
  "cover": "Ficção",
```

```
  "language": "Português"
```

```
}'
```



Consumir API REST Spring Boot - cURL

- Visualizar book específico: curl http://localhost:8080/books/1
- Editar book:

```
curl -X PUT http://localhost:8080/books/1 \
```

```
-H "Content-Type: application/json" \
```

```
-d '{
```

```
  "title": "O Senhor dos Anéis",
```

```
  "author": "J.R.R. Tolkien",
```

```
  "published_date": "1954-10-29",
```

```
  "isbn": "9780261102385",
```

```
  "pages": 1178,
```

```
  "cover": "Ficção",
```

```
  "language": "Português - ATUALIZADO"
```

```
}'
```



Consumir API REST Spring Boot - cURL

- Deletar book: `curl -X DELETE http://localhost:8080/books/1`



Consumir API REST Spring Boot - Postman

- EC2 / Instâncias / Clicar na instância criada / Descrição / IPv4 público: copiar IP
- Pode-se visualizar IPv4 via CLI: `ip addr show eth0`
- Listar books: GET `http://IPv4-PUBLICO-DA-EC2:8080/books`
- Adicionar book: POST `http://IPv4-PUBLICO-DA-EC2:8080/books`

Body raw (JSON):

```
{  
  "title": "O Senhor dos Anéis",  
  "author": "J.R.R. Tolkien",  
  "published_date": "1954-07-29",  
  "isbn": "9780261102385",  
  "pages": 1178,  
  "cover": "Ficção",  
  "language": "Português"  
}
```



Consumir API REST Spring Boot - Postman

- Visualizar book específico: GET <http://IPv4-PUBLICO-DA-EC2:8080/books/1>
- Editar book: PUT <http://IPv4-PUBLICO-DA-EC2:8080/books/1>

Body raw (JSON):

```
{  
  "title": "O Senhor dos Anéis",  
  "author": "J.R.R. Tolkien",  
  "published_date": "1954-10-29",  
  "isbn": "9780261102385",  
  "pages": 1178,  
  "cover": "Ficção",  
  "language": "Português - ATUALIZADO"  
}
```

- Deletar book: DELETE <http://IPv4-PUBLICO-DA-EC2:8080/books/1>



Parar instância AWS

- Na instância, executar comandos:
 - Parar servidor Gradle: `pkill -f gradle`
 - Parar MySQL: `sudo systemctl stop mysql`
- EC2 / Instâncias / Selecionar instância / Ações / Estado da instância / Encerrar instância
- EC2 / Grupos de segurança / Selecionar grupo de segurança da instância / Excluir
- EC2 / Chaves de acesso / Selecionar keySpringBoot1.pem / Excluir



UB Social

ubsocial.github.io