

Deploy Application on Cloud Code:

Create RDS in AWS:

* creating new databases

->Standard Class

->Mysql

->Free tier

->Dbname

->admin

->Set password

->Allow Public access

*copy the ending point and port

->Endpoint :database-1.cvlw14xgzwhp.us-east-1.rds.amazonaws.com

->Port:3306

Create new Database connection[MysqlWorkbench]:

*CANNOT CONNECT TO DATABASE SERVER

While creating database new connection,we will face port issue we can't connect it..!

1. open security group[inbound rule -> add rule -> MYSQL/Aurora,anywhere ->save rule]
2. issue will be solved ,open root user first after that add new connection database
3. open database[hostname:rds url,3306,admin,root987654321]

*create database

->create database awsd;

->use awsd;

->show tables;

->desc employee;

->select * from employee;

Create Spring Initializer project :

1. create spring project with spring initializer from web
java,maven,3.1.8,name,17,dependencies
2. unzip the folder
3. import project and run

Application.properties:

```
server.port=9091
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://database-1.cvlw14xgzwhp.us-east-1.rds.amazonaws.com:3306/awsd
spring.datasource.username=admin
spring.datasource.password=root987654321
```

spring.jpa.hibernate.ddl-auto=update
pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.8</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>RestAPI_With_Aws_Db</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <configuration>
          <image>
            <builder>paketobuildpacks/builder-
jammy-base:latest</builder>
```

```

        </image>
    </configuration>
</plugin>
</plugins>
</build>

</project>

```

Class & Packages:

Com.main(RestApiWithAwsDbApplication):
package com;

```

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

```

```

@SpringBootApplication(scanBasePackages = "com")
@EntityScan(basePackages = "com.entity")
@EnableJpaRepositories(basePackages = "com.repository")
public class RestApiWithAwsDbApplication {

    public static void main(String[] args) {
        SpringApplication.run(RestApiWithAwsDbApplication.class, args);
        System.err.println("Spring Boot is up..");
    }

}

```

Com.Controller(EmployeeController):
package com.controller;

```

import java.util.List;

```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

```

```

import com.entity.Employee;
import com.service.EmployeeService;

```

```

@RestController
@RequestMapping("employee")
public class EmployeeController {

```

```

    @Autowired
    EmployeeService employeeService;

```

```

    // http://localhost:9091/employee/findAll
    @GetMapping(value = "findAll", produces = MediaType.APPLICATION_JSON_VALUE)
    public List<Employee> findAllEmployee() {
        return employeeService.findAll();
    }

```

```

    }

    // http://localhost:9091/employee/store
    // {"name":"Ravi","salary":34000}

    @PostMapping(value = "store", consumes = MediaType.APPLICATION_JSON_VALUE)
    public String storeEmployee(@RequestBody Employee emp) {
        return employeeService.storeEmployee(emp);
    }
}

```

Com.entity(Employee):
package com.entity;

```

import jakarta.annotation.Generated;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

```

@Entity

```

public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int eid;
    private String name;
    private float salary;
    public int getEid() {
        return eid;
    }
    public void setEid(int eid) {
        this.eid = eid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public float getSalary() {
        return salary;
    }
    public void setSalary(float salary) {
        this.salary = salary;
    }
    @Override
    public String toString() {
        return "Employee [eid=" + eid + ", name=" + name + ", salary=" + salary + "]";
    }
    public Employee(int eid, String name, float salary) {
        super();
        this.eid = eid;
        this.name = name;
        this.salary = salary;
    }
}

```

```

        public Employee() {
            super();
            // TODO Auto-generated constructor stub
        }

    }

Com.repository(EmployeeRepository):
package com.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.entity.Employee;
@Repository
public interface EmployeeRepository extends JpaRepository<Employee, Integer>{

}

Com.service(EmployeeService):
package com.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.entity.Employee;
import com.repository.EmployeeRepository;

@Service
public class EmployeeService {

    @Autowired
    EmployeeRepository employeeRepository;

    public String storeEmployee(Employee emp) {
        employeeRepository.save(emp);
        return "Employee record stored";
    }

    public List<Employee> findAll(){
        return employeeRepository.findAll();
    }

}

```

Now,Run the main class

Inserted data from Postman tool:

*while using postman the eclipse project should be run[Spring Boot is up..]

Post -> url[http://localhost:9091/employee/store]

Header -> content-type,application-json

Body -> raw,json

[{"name":"Ravi","salary":34000}]

Get -> url[http://localhost:9091/employee/findAll]

And stop the both eclipse project and postman

Create EC2 Instance:

- 1.create instance
- 2.create key pair
- 3.edit inbound rule -> add rule,custom tcp, our port number[9091],anywhere,save rule
4. connect EC2 instance

Conect EC2 with localGitBash:

open gitbash the aws key pair file is present

1. paste the url [ssh -i "RESTAPI.pem" ec2-user@ec2-54-83-253-65.compute1.amazonaws.com]
2. sudo yum install java
java --version
wget {url of s3 bucket instances}
ls
java -jar[demo-0.0.1-SNAPSHOT.jar]

3. with public ip address[http://54.83.253.65:9091/employee/findAll] the output will be shown.