Phase 5 Project

CI/CD Deployment for Spring Boot Application - Codes

M. Ramakrishna

GitHub Link:

https://github.com/Ramakrishana1/Phase5_Project

EmployeeController.java

```java
package com.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

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

        @GetMapping(value = "sayHello")
        public String greeting() {
                return "Welcome to Spring Boot using Docker";
        }

        @GetMapping(value = "welcome/{name}")
        public String sayWelcome(@PathVariable("name")String name) {
                return "Welcome to Spring Boot using Docker User"+name;
        }
}
```

<div align="center">

SpringBootRestApplication.java

</div>

```java
package com;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com")
public class SpringBootRestApplication {

        public static void main(String[] args) {
                SpringApplication.run(SpringBootRestApplication.class, args);
                System.err.println("Application running on port 9191");
        }

}
```

<div align="center">

Dockerfile:

</div>

```dockerfile
FROM openjdk:11
COPY ./target/MySpringboot.jar .
CMD ["java","-jar","MySpringboot.jar"]
```

<div align="center">

application.properties:

</div>

```properties
server.port=9191
```

<div align="center">

Jenkin Pipeline Code:

</div>

```groovy
pipeline {
  agent any
  tools {
    // Install the Maven version configured as "M3" and add it to the path.
    maven "MyMaven"
    git "MyGit"
  }
  stages {
    stage('Build') {
      steps {
        // Get some code from a GitHub repository
        git branch: 'main', url: 'https://github.com/Ramakrishana1/Phase5_Project.git'

        // Run Maven on a Unix agent.
        sh "mvn -Dmaven.test.failure.ignore=true clean package"
      }
    }
    stage('Install') {
      steps {
        // Get some code from a GitHub repository
        git branch: 'main', url: 'https://github.com/Ramakrishana1/Phase5_Project.git'
```

```
            // Run Maven on a Unix agent.
            sh "mvn -Dmaven.test.failure.ignore=true install package"
          }
        }
       stage('Compile') {
          steps {
            // Get some code from a GitHub repository
            git branch: 'main', url: 'https://github.com/Ramakrishana1/Phase5_Project.git'

            // Run Maven on a Unix agent.
            sh "mvn -Dmaven.test.failure.ignore=true compile package"
          }
        }
      }
    }
}
```