CI/CD Deployment for Springboot source code

CicdAppliedToSpringBootJavaAppApplication.java

```
package com.cicd.cicdappliedtospringbootjavaapp;

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


@SpringBootApplication

public class CicdAppliedToSpringBootJavaAppApplication {



	public static void main(String[] args) {
		SpringApplication.run(CicdAppliedToSpringBootJavaAppApplication.class, args);
	}

}
```

HelloController.java

```
package com.cicd.cicdappliedtospringbootjavaapp.controller;

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

@RestController
public class HelloController {

	@GetMapping("/")
	public String home() {
    return "Hello World from DZONE";
  }

}
```

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>2.1.8.RELEASE</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.cicd</groupId>
	<artifactId>cicd-applied-to-spring-boot-java-app</artifactId>
	<packaging>jar</packaging>
	<version>0.0.1-SNAPSHOT</version>
	<name>cicd-applied-to-spring-boot-java-app</name>
	<description>Implementing CI/CD on Spring Boot Java App</description>

	<properties>
		<java.version>1.8</java.version>

	<!-- solving error : Invalid or corrupt jarfile /app.jar -->	<project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
```

```
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

      <!-- property useful for  spotify's dockerfile-maven-plugin -->

      <!-- Instead of "fanjups", please add your Docker Hub username -->

      <docker.image.prefix>fanjups</docker.image.prefix>

      <!--Not adding artifacts to remote repository-->

      <maven.deploy.skip>true</maven.deploy.skip>

      <!-- GitHub OAuth token & server -->
      <github.global.server>github</github.global.server>
<github.global.oauth2Token>${env.GITHUB_OAUTH_TOKEN}</github.global.oauth2Token>

<!-- Useful for Heroku Deployment -->

<full-artifact-name>target/${project.artifactId}-${project.version}.jar</full-artifact-name>

  </properties>

  <dependencies>
      <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-web</artifactId>
      </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>
          </plugin>

          <!-- spotify's dockerfile-maven-plugin -->

          <plugin>
     <groupId>com.spotify</groupId>
     <artifactId>dockerfile-maven-plugin</artifactId>
     <version>1.4.9</version>
     <executions>
                  <execution>
                      <id>default</id>
                      <phase>install</phase>
                      <goals>
                          <goal>build</goal>
                          <goal>push</goal>
                      </goals>
                  </execution>
              </executions>
     <configuration>
      <repository>${docker.image.prefix}/${project.artifactId}</repository>

      <serverId>index.docker.io</serverId>
          <registryUrl>https://index.docker.io:443/v1/</registryUrl>
     </configuration>
  </plugin>

  <!-- maven-dependency-plugin useful for creating docker image -->

  <plugin>
```

```
                             <groupId>org.apache.maven.plugins</groupId>
                             <artifactId>maven-dependency-plugin</artifactId>
                             <executions>
                               <execution>
                                 <id>unpack</id>
                                 <phase>package</phase>
                        <goals>
                                   <goal>unpack</goal>
                                 </goals>
                                 <configuration>
                                   <artifactItems>
                                     <artifactItem>
                                       <groupId>${project.groupId}</groupId>
                                       <artifactId>${project.artifactId}</artifactId>
                                       <version>${project.version}</version>
                                     </artifactItem>
                                   </artifactItems>
                                 </configuration>
                               </execution>


                             </executions>

                  </plugin>

                  <!-- jacoco-maven-plugin useful for Codecov -->

                  <plugin>
                                    <groupId>org.jacoco</groupId>
                                    <artifactId>jacoco-maven-plugin</artifactId>
                                    <version>0.7.7.201606060606</version>
                                    <executions>
                                             <execution>
                                                      <goals>
                                                               <goal>prepare-agent</goal>
                                                      </goals>
                                             </execution>
                                             <execution>
                                                      <id>report</id>
                                                      <phase>test</phase>
                                                      <goals>
                                                               <goal>report</goal>
                                                      </goals>
                                             </execution>
                                    </executions>
                           </plugin>

                           <!-- Generating github  gh pages & maven project documents  -->

      <plugin>

         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-site-plugin</artifactId>
</plugin>

      <plugin>
                                    <groupId>com.github.github</groupId>
                                    <artifactId>site-maven-plugin</artifactId>
                                    <version>0.12</version>
                                    <configuration>
                                             <message>Building site for ${project.name} ${project.version}</message>
                                             <server>github</server>

                                    </configuration>
                                    <executions>
                                             <execution>
                                                      <goals>
                                                               <goal>site</goal>
                                                      </goals>
                                                      <phase>site</phase>
                                             </execution>
                                       </executions>
                              </plugin>

                              <!-- Useful for Heroku Deployment -->

                              <plugin>
                       <groupId>com.heroku.sdk</groupId>
                       <artifactId>heroku-maven-plugin</artifactId>
                       <version>2.0.11</version>
                       <configuration>
                        <appName>cicd-spring-boot-java-app</appName>
                        <processTypes>
                         <web>java $JAVA_OPTS -jar -Dserver.port=$PORT ${full-artifact-name}</web>
                        </processTypes>
                       </configuration>
                    </plugin>

                    </plugins>
          </build>

              <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-javadoc-plugin</artifactId>
                        <version>3.1.1</version>

       </plugin>


                    </plugins>

 </reporting>

 <licenses>
                    <license>
                              <name>MIT License</name>
                               <url>http://www.opensource.org/licenses/mit-license.php</url>
                              <distribution>repo</distribution>
                    </license>
          </licenses>

 <developers>
          <developer>
                <email>sahil.21091999@gmail.com</email>
                    <name>Sahil Agrawal</name>
                    <url>https://github.com/sahil743</url>
                    <id>sahil743</id>
          </developer>

 </developers>

 <organization>
                     <name>Sahil Agrawal</name>
                    <url>https://github.com/sahil743/</url>
          </organization>

 <issueManagement>
                    <system>GitHub Issues</system>
                     <url> https://github.com/sahil743/CI-CD_deploy_for_springboot.git /issues</url>
 </issueManagement>

 <scm>
                     <url> https://github.com/sahil743/CI-CD_deploy_for_springboot.git </url>
                    <connection>scm:git:git://github.com/sahil743/CI-CD_deploy_for_springboot.git.git</connection>
          <developerConnection>scm:git:git://github.com/sahil743/CI-
CD_deploy_for_springboot.git.git</developerConnection>
 </scm>

</project>
```

Dockerfile

```
# Start with a base image containing Java runtime
FROM openjdk:8-jdk-alpine

# Add Maintainer Info
LABEL maintainer="sahil.21091999@gmail.com"

# Add a volume pointing to /tmp
VOLUME /tmp

# Make port 8080 available to the world outside this container
EXPOSE 8080

# The application's jar file
ARG JAR_FILE

# Add the application's jar to the container
ADD ${JAR_FILE} app.jar

# Run the jar file
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]
```