Jenkins Pipeline to Deploy Docker Swarm Code:

Java class:

```
package com;
import java.util.Random;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;
@SpringBootApplication
public class DockerSwarmApplication {
public static void main(String[] args) {
SpringApplication.run(DockerSwarmApplication.class, args);
}
}
@Controller
class IndexController{
static String randomWebAppID= new String(new Random().toString());
@GetMapping("/")
@ResponseBody
public String index() {
return "[WEBAPPID "+ randomWebAppID + "] Hello World from
Spring";
}
}
```

Application properties :

```
localhost:8080
```

dockerfile :

```
FROM eclipse-temurin:17-jdk-alpine
VOLUME /tmp
```

```
COPY target/*.jar app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
Pipelinescript :
pipeline{
agent any
tools {
//Install the Maven version configured as "M3" and add it to
the path.
maven "Maven"
jdk "java1"
}
stages {
stage('Checkout from GitHub') {
steps {
git branch: 'main', url:
"https://github.com/awtraining1/sl.git"
}
}
stage('Maven Build') {
steps {
dir('PHASE5/my/spring-bootdemo') {
bat "mvn -Dmaven.test.skip=true clean package"
}
}
}
stage('Docker Image Creation') {
steps {
dir('PHASE5/my/spring-bootdemo') {
bat "docker build -t my-morning-spring-app --output
type=docker ."
}
```

```
}
}
stage('Push Docker Image') {
steps {
bat "docker tag my-morning-spring-app anithaneel/my-
morning-spring-app"
bat "docker push anithaneel/my-morning-spring-app"
}
}
}
}
```