



Configure Jenkins Job to create Docker Image and spin up Docker Container for your Spring Boot Application


1. Click on **New Item** link
2. Enter the **job name**, select **Maven Project** and click on **OK**


Enter an item name


» Required field


**Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


**Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

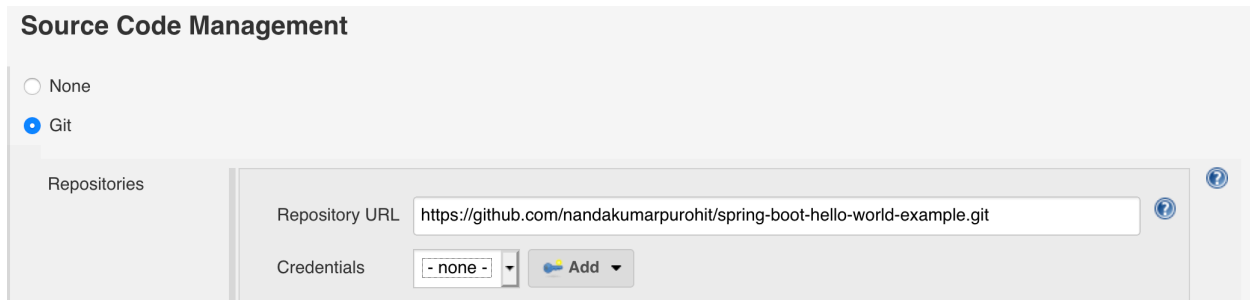
**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

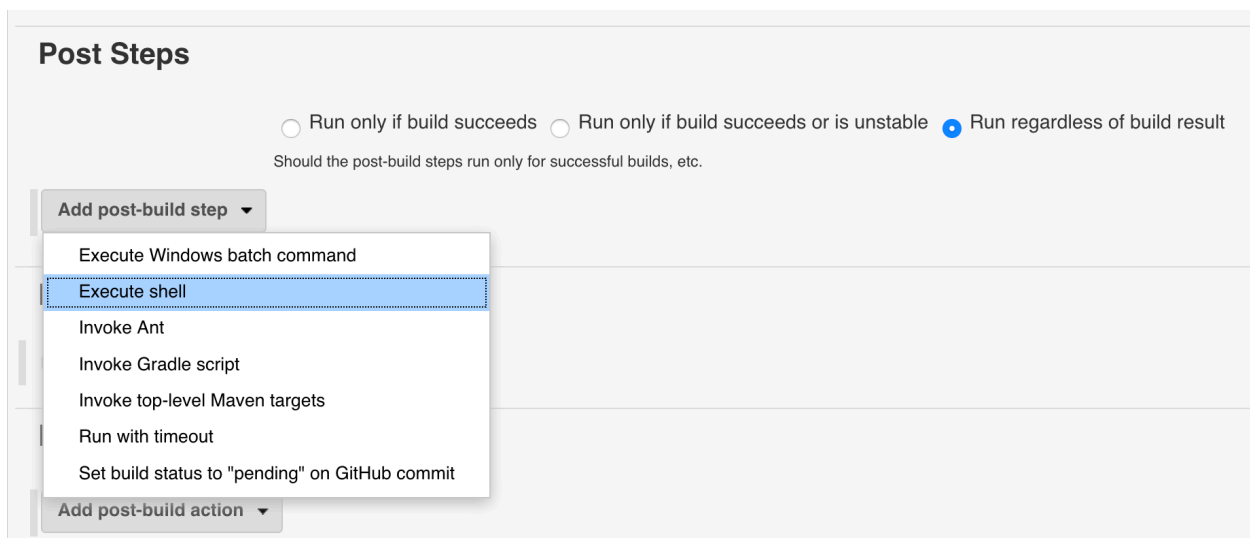
OK

3. Specify the Git Repo URL



The screenshot shows the 'Source Code Management' configuration panel. At the top, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Below this, there is a 'Repositories' section. In this section, the 'Repository URL' is set to 'https://github.com/nandakumarpurohit/spring-boot-hello-world-example.git'. The 'Credentials' dropdown is set to '- none -', and there is an 'Add' button next to it. A help icon (?) is visible on the right side of the panel.

4. Click on **Add post-build step** and select **Execute shell**



The screenshot shows the 'Post Steps' configuration panel. At the top, there are three radio buttons: 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result'. The third option is selected. Below this, there is a text label: 'Should the post-build steps run only for successful builds, etc.'. A button labeled 'Add post-build step' is visible, and a dropdown menu is open below it. The menu contains the following options: 'Execute Windows batch command', 'Execute shell' (which is highlighted), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. At the bottom of the panel, there is another button labeled 'Add post-build action'.

5. Enter the docker build & docker run commands which you tested on command prompt in earlier exercise.

```
docker build -t boot-app-05nov-02:${BUILD_NUMBER} ${WORKSPACE}  
docker run -it -d -p 3000:5000 boot-app-05nov-02:${BUILD_NUMBER}
```

Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Execute shell

Command

```
docker build -t boot-app-05nov-02:${BUILD_NUMBER} ${WORKSPACE}
docker run -it -d -p 3000:5000 boot-app-05nov-02:${BUILD_NUMBER}
```






[See the list of available environment variables](#)


Advanced...

Add post-build step ▼

6. Click on **Save** button

7. Click on **Build Now** link and go to the **Console Output** of the current build in progress

-  Build Now
-  Delete Maven project
-  Configure
-  Modules
-  Rename


 Build History


[trend](#)

find X

#6

Nov 5, 2019 7:53 PM

 Changes

 Console Output

8. You can observe in the console that Jenkins build process picks up docker build command, generates the docker image and also creates a Docker container

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.071 s
[INFO] Finished at: 2019-11-05T19:53:22-05:00
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /Users/nanda/.jenkins/workspace/boot-002/pom.xml to com.demo.springboot/spring-boot-hello-world-example/0.0.1-SNAPSHOT
/spring-boot-hello-world-example-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /Users/nanda/.jenkins/workspace/boot-002/target/spring-boot-hello-world-example-0.0.1-SNAPSHOT.jar to
com.demo.springboot/spring-boot-hello-world-example/0.0.1-SNAPSHOT/spring-boot-hello-world-example-0.0.1-SNAPSHOT.jar
[boot-002] $ /bin/sh -xe /var/folders/vv/9x81ghcj5_jd7_4zpyqny5zc0000gn/T/jenkins9150309080332977211.sh
channel stopped
+ docker build -t boot-app-05nov-02:6 /Users/nanda/.jenkins/workspace/boot-002
Sending build context to Docker daemon 16.55MB

Step 1/4 : FROM openjdk:8-jdk-alpine
--> a3562aa0b991
Step 2/4 : EXPOSE 8080
--> Using cache
--> c084233bb3ce
Step 3/4 : ADD target/spring-boot-hello-world-example-0.0.1-SNAPSHOT.jar app.jar
--> 124c0dd8b9c2
Step 4/4 : ENTRYPOINT ["java","-jar","app.jar"]
--> Running in 5c9a80cf5752
Removing intermediate container 5c9a80cf5752
--> 737aa8842580
Successfully built 737aa8842580
Successfully tagged boot-app-05nov-02:6
+ docker run -it -d -p 3000:5000 boot-app-05nov-02:6
f177593f163a1de2feaca72ade8d94e68e10e22c7bda1c19137217f9ef2cedc2
Finished: SUCCESS
```

9. Go to CMD prompt and verify that the new decker image is created

decker images

```
Nandakumars-MacBook-Pro:boot-002 nanda$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
boot-app-05nov-02	6	737aa8842580	15 seconds ago	121MB
boot-app-05nov-01	1.0	ed031beab71c	5 hours ago	121MB
openjdk	8-jdk-alpine	a3562aa0b991	5 months ago	105MB

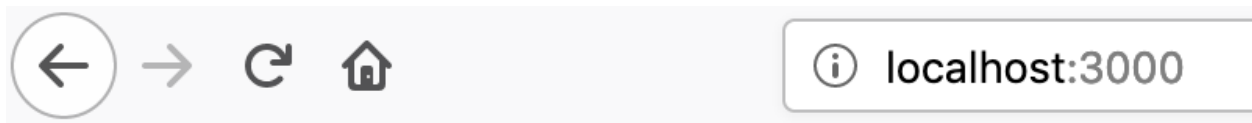
10. Also verify that the Docker container is also started for the image as part of this Jenkins Job

docker ps

```
Nandakumars-MacBook-Pro:boot-002 nanda$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
f177593f163a	boot-app-05nov-02:6	"java -jar app.jar"	25 seconds ago	Up 23 seconds	8080/tcp, 0.0.0.0:3000->5000/tcp
priceless_tereshkova					
fa76d0fb68e5	boot-app-05nov-01:1.0	"java -jar app.jar"	3 hours ago	Up 3 hours	8080/tcp, 0.0.0.0:4000->5000/tcp
crazy_snyder					

11. Visit **<http://localhost:3000>**



Hello Spring Boot!!

