

Configure Jenkins Job to create Docker Image, Push this image to Docker Hub and spin up Docker Container for your Spring Boot Application

Install Docker Plugin for “Docker Build and Publish”

CloudBees Docker Build and Publish plugin

| Updates | Available | Installed | Advanced |
|-------------------------------------|-----------|--|-----------------------|
| Enabled | | | |
| | | Name ↓ | Version |
| <input checked="" type="checkbox"/> | | Authentication Tokens API Plugin This plugin provides an API for converting credentials into authentication tokens in Jenkins. | 1.3 |
| <input checked="" type="checkbox"/> | | CloudBees Docker Build and Publish plugin This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry. | 1.3.2 |


Restart Jenkins after installation


http://localhost:8080/restart

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

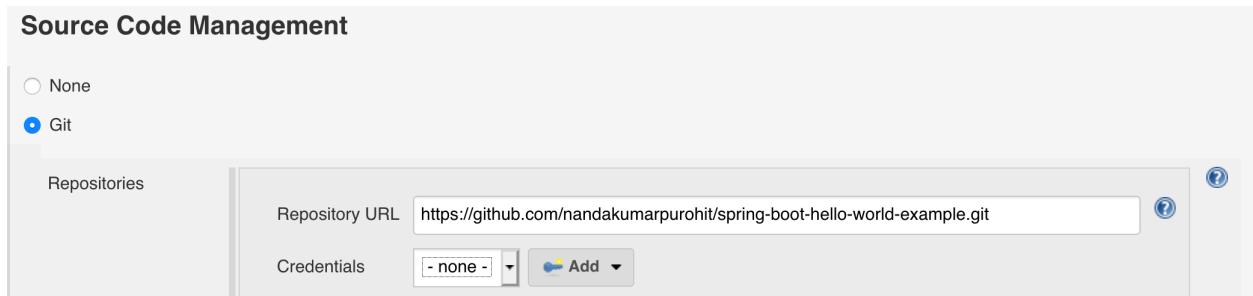
Enter an item name

» 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.

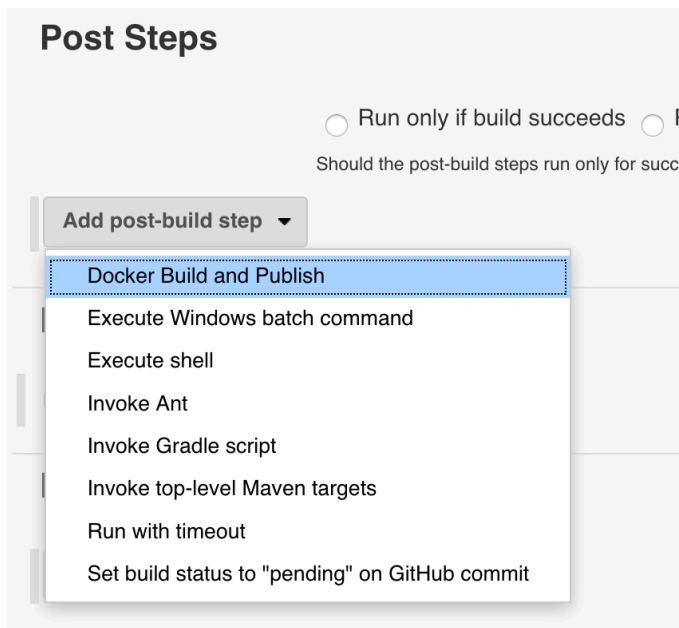
**Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

3. Specify the Git Repo URL



The screenshot shows the 'Source Code Management' configuration panel. The 'Git' radio button is selected. Under the 'Repositories' section, the 'Repository URL' is set to 'https://github.com/nandakumarpurohit/spring-boot-hello-world-example.git'. The 'Credentials' dropdown is set to '- none -' with an 'Add' button next to it.

4. Click on **Add post-build step** and select **Docker Build and Publish** Menu Item



The screenshot shows the 'Post Steps' configuration panel. At the top, there are two radio buttons: 'Run only if build succeeds' (selected) and 'Run only if build fails'. Below this is a text label: 'Should the post-build steps run only for success?'. A button labeled 'Add post-build step' is visible. A dropdown menu is open, showing a list of options. The first option, 'Docker Build and Publish', is highlighted in blue. Other options include 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

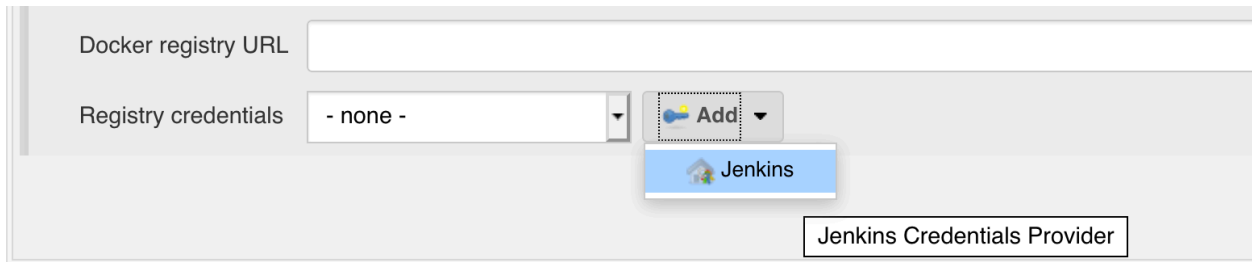
5.

Repo Name: <docker-hub-id>/<repo-name>
Tag: \${BUILD_NUMBER}


5.1 Registry Credentials


Follow these steps

Click on **Add** → **Jenkins**



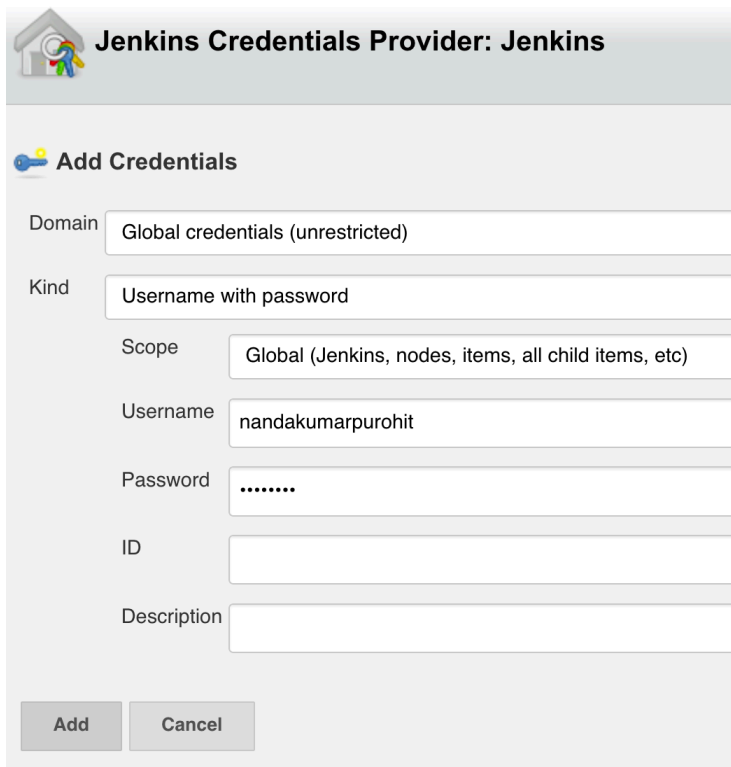
Docker registry URL


Registry credentials - none -  Add ▼


 Jenkins

Jenkins Credentials Provider

Enter **Username / Password**
Click on **Add**



 **Jenkins Credentials Provider: Jenkins**

 **Add Credentials**

Domain

Kind

Scope

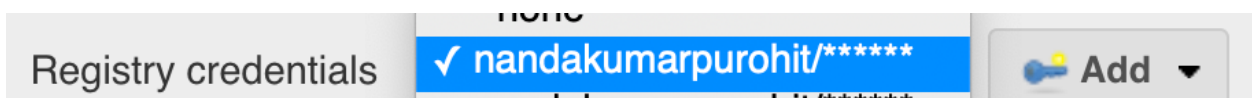
Username


Password

ID

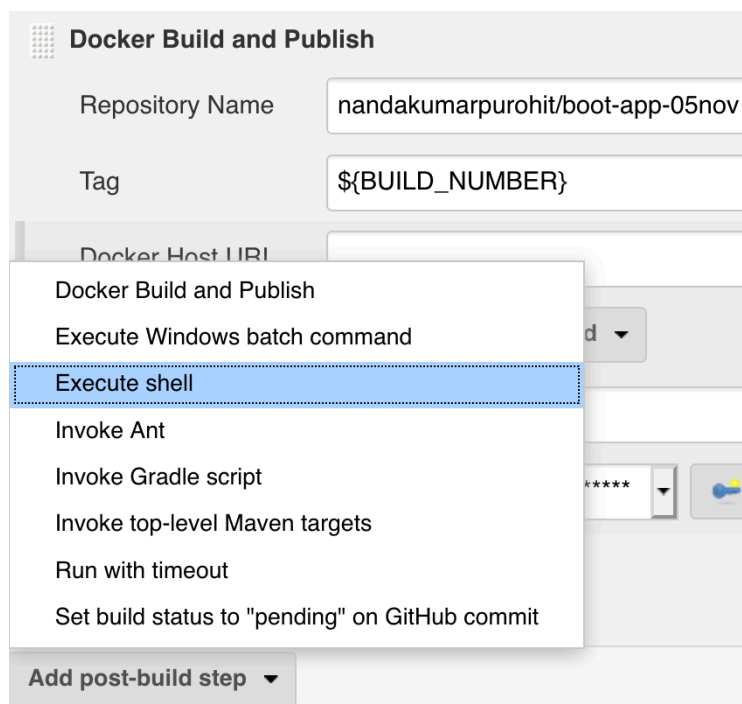
Description

Select added credential from the dropdown



Registry credentials ✓ nandakumarpurohit/*****  Add ▼



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




7. Enter the docker run command which starts a container out of an image which is pulled from docker hub

docker run -it -d -p 2222:5000 nandakumarpurohit/boot-app-05nov:\${BUILD_NUMBER}

Post-build Actions

| | | |
|----------------------|--|---|
| Repository Name | <input type="text" value="nandakumarpurohit/boot-app-05nov"/> | ? |
| Tag | <input type="text" value="\${BUILD_NUMBER}"/> | |
| Docker Host URI | <input type="text"/> | ? |
| Server credentials | <div>- none -  Add</div> | |
| Docker registry URL | <input type="text"/> | ? |
| Registry credentials | <div>nandakumarpurohit/*****  Add</div> | |

Advanced...

 **Execute shell** X ?

Command

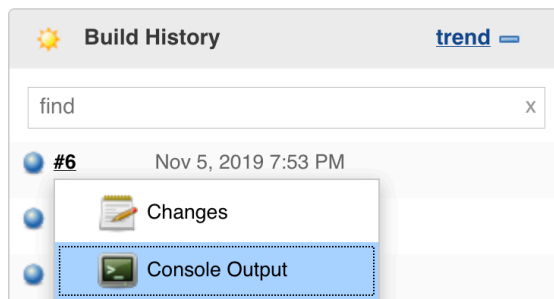
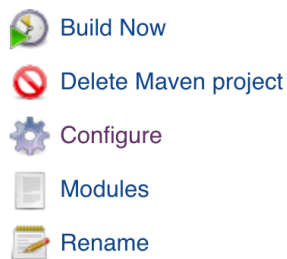
See [the list of available environment variables](#)

Advanced...

Add post-build step ▾

8. Click on **Save** button

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



10. You can observe in the console that Jenkins **Docker Build and Publish** plugin, generates the docker image, published it into our Docker Hub Repo and also creates a Docker container out of this image. See the logs below!

```
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.921 s
[INFO] Finished at: 2019-11-06T06:45:04-05:00
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /Users/nanda/.jenkins/workspace/boot-003/pom.xml to com.demosj
/spring-boot-hello-world-example-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /Users/nanda/.jenkins/workspace/boot-003/target/spring-boot-he
com.demos.springboot/spring-boot-hello-world-example/0.0.1-SNAPSHOT/spring-boot-hel
channel stopped
[boot-003] $ docker build -t nandakumarpurohit/boot-app-05nov:1 --pull=true /Users
Sending build context to Docker daemon 16.55MB
```

```
Step 1/4 : FROM openjdk:8-jdk-alpine
8-jdk-alpine: Pulling from library/openjdk
Digest: sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3
Status: Image is up to date for 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
---> 0ec8d274e3c1
Step 4/4 : ENTRYPOINT ["java","-jar","app.jar"]
---> Running in d59f17dfec13
Removing intermediate container d59f17dfec13
---> 15ed4e2b3653
Successfully built 15ed4e2b3653
Successfully tagged nandakumarpurohit/boot-app-05nov:1
[boot-003] $ docker tag 15ed4e2b3653 nandakumarpurohit/boot-app-05nov:latest
[boot-003] $ docker inspect 15ed4e2b3653
[boot-003] $ docker push nandakumarpurohit/boot-app-05nov:1
The push refers to repository [docker.io/nandakumarpurohit/boot-app-05nov]
e2efbfb4f581: Preparing
ceaf9elebef5: Preparing
9b9b7f3d56a0: Preparing
flb5933fe4b5: Preparing
9b9b7f3d56a0: Mounted from library/openjdk
ceaf9elebef5: Mounted from library/openjdk
flb5933fe4b5: Mounted from library/openjdk
```

```

e2efbfb4f581: Pushed
1: digest: sha256:22e88e6271ab0208d4de97a05ee384f39afde3850abcad8b1d46530a11918956 size: 1159
[boot-003] $ docker push nandakumarpurohit/boot-app-05nov:latest
The push refers to repository [docker.io/nandakumarpurohit/boot-app-05nov]
e2efbfb4f581: Preparing
ceaf9elebef5: Preparing
9b9b7f3d56a0: Preparing
f1b5933fe4b5: Preparing
e2efbfb4f581: Layer already exists
ceaf9elebef5: Layer already exists
f1b5933fe4b5: Layer already exists
9b9b7f3d56a0: Layer already exists
latest: digest: sha256:22e88e6271ab0208d4de97a05ee384f39afde3850abcad8b1d46530a11918956 size: 1159
[boot-003] $ /bin/sh -xe /var/folders/vv/9x81ghcj5_jd7_4zpyqny5zc0000gn/T/jenkins1642477637313047207.sh
+ docker run -it -d -p 2222:5000 nandakumarpurohit/boot-app-05nov:1
2de5697bd9c4c17ed888ff4317e3fdb224eb74157c3b4cd3ff100231adad648e
Finished: SUCCESS

```

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

decker images

```

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

```

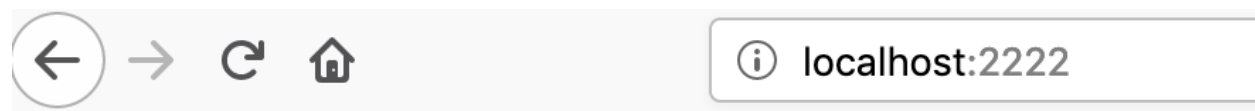
| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|----------------------------------|--------------|--------------|----------------|-------|
| nandakumarpurohit/boot-app-05nov | 1 | 15ed4e2b3653 | 56 seconds ago | 121MB |
| nandakumarpurohit/boot-app-05nov | latest | 15ed4e2b3653 | 56 seconds ago | 121MB |
| boot-app-05nov-02 | 6 | 737aa8842580 | 11 hours ago | 121MB |
| boot-app-05nov-01 | 1.8 | ed031beab71c | 16 hours ago | 121MB |
| openjdk | 8-jdk-alpine | a3562aa0b991 | 5 months ago | 105MB |

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

docker ps

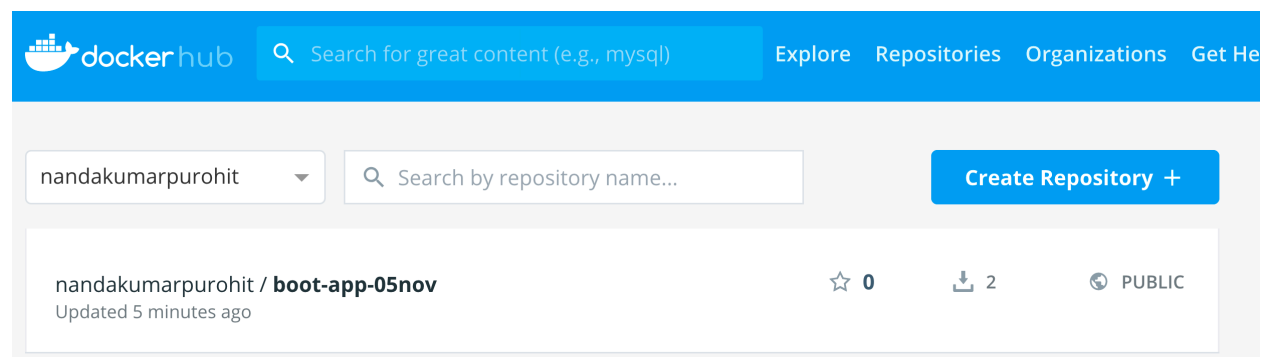
```
Nandakumars-MacBook-Pro:boot-002 nanda$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
2de5697bd9c4   nandakumarpurohit/boot-app-05nov:1 "java -jar app.jar"     6 minutes ago  Up 6 minutes  8080/tcp, 0.0.0.0:2222->5000/tcp    naughty_b
oyd            boot-app-05nov-02:6               "java -jar app.jar"     11 hours ago   Up 11 hours   8080/tcp, 0.0.0.0:3000->5000/tcp    priceless
f177593f163a   boot-app-05nov-01:1.0             "java -jar app.jar"     14 hours ago   Up 14 hours   8080/tcp, 0.0.0.0:4000->5000/tcp    crazy_sny
der
```

13. Visit **<http://localhost:2222>**





Hello Spring Boot!!

14. Go to your Docker Hub Dashboard and verify that the image is published in your repository



nandakumarpurohit / boot-app-05nov

This repository does not have a description 

 Last pushed: 5 minutes ago

Docker commands





[Public View](#)

To push a new tag to this repository,

```
docker push nandakumarpurohit/boot-app-05nov:tagname
```

Tags

This repository contains 2 tag(s).

| | | |
|--------|---|---|
| 1 |  |  5 minutes ago |
| latest |  |  5 minutes ago |

Recent builds

Link a source provider and run a build to see build results here.

