<https://www.jenkins.io/download/>

java -jar jenkins.war

java -jar jenkins.war –httpPort=9090(default 80800)

Jenkins initial setup is required. An admin user has been created and a password generated.

Please use the following password to proceed to installation:

fbcc7047ad0e49159cc1faefbc57a1b4

This may also be found at: C:\Users\Manimozhi\.jenkins\secrets\initialAdminPassword

ManimozhiB-UN

Manimozhimca!15

For java

Javac filename.java

Javac filename

Use windows to execute commands for compilation

Build Triggers:

Poll scm: \* \* \* \* \*

To build spring using maven:

<https://github.com/Kaleakash/springbootwithjwar>

repo url: https://github.com/Kaleakash/springbootwithjwar.git

Build Steps: invoke toplevel build targets

Maven commands:

clean

compile

test

package

For angular apps add angular by plugin

Add nodejs plugin from manage jenkin->plugins->available->nodejs

From tools install nodeja version

url: <https://github.com/Kaleakash/aws-docker-jenkin-angular>

gir url: <https://github.com/Kaleakash/aws-docker-jenkin-angular.git>

Build environment:

Provide node&npm/folder path

Using Execute shell

node --version

npm --version

npm install -g @angular/cli@16.0.1

npm install

ng build

Git Manimozhi!15

Token(ghp\_BbKBj8eE5EdaEwQcRLbuhS7Z2ggfc73mXBJu)

>git remote add https:// ghp\_BbKBj8eE5EdaEwQcRLbuhS7Z2ggfc73mXBJu

[I0eYxDF@github.com/Manimozhi91/JenkinsSpring.git](mailto:I0eYxDF@github.com/Manimozhi91/JenkinsSpring.git)

https://ghp\_BbKBj8eE5EdaEwQcRLbuhS7Z2ggfc73mXBJu

https://<TOKEN>@github.com/<username>/<repository>.git

.

git push <https://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@github.com/Manimozhi91/JenkinsSpring.git>

git remote add origin https://github.com/jglick/simple-maven-project-with-tests.git

rm -rf .git

https://ghp\_pbIsBW8q6Wsjt607es740e8lfepbeU2EKmOi@github.com/Manimozhi91/simplilearn.git

https://ghp\_pbIsBW8q6Wsjt607es740e8lfepbeU2EKmOi@https://github.com/Manimozhi91/PetClinic.git

Pipeline:

Script

pipeline {

agent any

stages {

stage('Check the software version') {

steps {

echo 'Checking Software version'

sh "java --version"

}

}

stage('Compile the program') {

steps {

echo 'Compile the program'

sh "javac"

}

}

stage('Run the program') {

steps {

echo 'Run the program '

sh "java"

}

}

stage('Test the the program') {

steps {

echo 'Test the program'

}

}

}

}

[https://github.com/Kaleakash/Jenkins\_post\_pipe\_line\_job-- simple](https://github.com/Kaleakash/Jenkins_post_pipe_line_job--%20simple) java

<https://github.com/Kaleakash/spring-boot-with-pipeline-jenkin-aws/blob/master/Jenkinsfile---maven>

<https://notepad.pw/share/w8XkjGRnvb9DPkzovREq>

pipeline {

agent any

tools {

// Install the Maven version configured as "M3" and add it to the path.

maven "MAVEN"

}

stages {

stage('Build') {

steps {

// Get some code from a GitHub repository

git 'https://github.com/Kaleakash/Jenkins\_post\_pipe\_line\_job.git'

// Run Maven on a Unix agent.

sh "javac Demo.java"

sh "java Demo"

// To run Maven on a Windows agent, use

// bat "mvn -Dmaven.test.failure.ignore=true clean package"

}

post {

// If Maven was able to run the tests, even if some of the test

// failed, record the test results and archive the jar file.

success {

echo "success post message"

}

}

}

}

}

TO GENERATE WAR FILE :

Jar -cf filename.war index.jsp

Pushing code to github:

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git init

Initialized empty Git repository in C:/Users/Manimozhi/course4/spring-boot-with-docker-compose/.git/

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git add .

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: .gitignore

new file: .mvn/wrapper/maven-wrapper.jar

new file: .mvn/wrapper/maven-wrapper.properties

new file: Dockerfile

new file: docker-compose.yml

new file: mvnw

new file: mvnw.cmd

new file: pom.xml

new file: src/main/java/com/SpringBootWithDockerComposeApplication.java

new file: src/main/java/com/bean/Product.java

new file: src/main/java/com/controller/ProductController.java

new file: src/main/java/com/repository/ProductRepository.java

new file: src/main/java/com/service/ProductService.java

new file: src/main/resources/application.properties

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git commit -m "my commit"

[master (root-commit) 7a34e16] my commit

14 files changed, 773 insertions(+)

create mode 100644 .gitignore

create mode 100644 .mvn/wrapper/maven-wrapper.jar

create mode 100644 .mvn/wrapper/maven-wrapper.properties

create mode 100644 Dockerfile

create mode 100644 docker-compose.yml

create mode 100644 mvnw

create mode 100644 mvnw.cmd

create mode 100644 pom.xml

create mode 100644 src/main/java/com/SpringBootWithDockerComposeApplication.java

create mode 100644 src/main/java/com/bean/Product.java

create mode 100644 src/main/java/com/controller/ProductController.java

create mode 100644 src/main/java/com/repository/ProductRepository.java

create mode 100644 src/main/java/com/service/ProductService.java

create mode 100644 src/main/resources/application.properties

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git remote add origin [https:// ghp\_YEeWFDmwWnSP7X10NcwpUPCHDF0koh2B6dex@github.com/Manimozhi91/AWSJenkins.git](https://%20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@github.com/Manimozhi91/AWSJenkins.git)

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git remote add origin https://ghp\_YEeWFDmwWnSP7X10NcwpUPCHDF0koh2B6dex@github.com/Manimozhi91/AWSJenkins.git

C:\Users\Manimozhi\course4\spring-boot-with-docker-compose>git push -u origin HEAD

For AWS <https://github.com/Kaleakash/spring_aws_docker_jenkin> --jenkin file

http://100.26.186.41:8080/

pipeline {

tools{

MAVEN

}

agent any

stages {

stage("version of software"){

steps{

sh "java --version"

sh “mvn --version”

//bat "java --version"

}

}

stage("Build program"){

steps{

sh “mvn clean”

sh “mvn package”

}

}

stage("run the java program"){

steps{

sh “docker-compose down”

sh “docker-compose up –build-d”

}

}

// }

}

post {

always {

echo 'This will always run'

}

success {

echo 'This will run only if successful'

}

failure {

echo 'This will run only if failed'

}

changed {

echo 'This will run only if the state of the Pipeline has changed'

echo 'Example : if the Pipeline was previously failing but is now successful'

}

}

}

<http://54.86.20.11:9090/product/findAll>

http://3.95.60.215:9090/product/findAll