1. Create your projects / services such as account-service or customer-service or any other services.
2. At the root level of the project create Docker file and Jenkins file with name Dockerfile and Jenkinsfile.
3. Under Dockerfile include the script below.

FROM openjdk

MAINTAINER swapnil.kumar54@gmail.com

ADD target/customer-service.jar customer-service.jar

ENTRYPOINT ["java", "-jar", "/customer-service.jar"]

EXPOSE 3333

1. Under Jenkinsfile include the pipeline details.

node {

withMaven(maven:'maven') {

stage('Checkout') {

git url: 'https://github.com/SwapnilRawath/microservice1.git', credentialsId: 'githubnew', branch: 'master'

}

stage('Build') {

sh 'mvn clean install'

def pom = readMavenPom file:'pom.xml'

print pom.version

env.version = pom.version

}

stage('Image') {

dir ('account-service') {

def app = docker.build "rawath0408/account-service:${env.version}"

docker.withRegistry('https://index.docker.io/v1/', 'dockerhub') {

app.push()

}

}

}

stage ('Run') {

//docker.withRegistry('https://index.docker.io/v1/', 'dockerhub') {

docker.image("rawath0408/account-service:${env.version}").run('-p 2222:2222 -h account --name account')

//} //--link discovery'

}

stage ('Final') {

//build job: 'customer-service-pipeline', wait: false

}

}

}

Note : “rawath0408/account-service” is a DockerHub repository.

“https://github.com/SwapnilRawath/microservice1.git' is a Git repository URL.

“githubnew” is a credential Id created on Jenkins.

“dockerhub” is a docker hub credential Id created on Jenkins.

“’maven’” is name of Maven configured on Jenkins.

1. Move the code to Git repository.
2. Now create an EC2 instance thorugh your AWS Console and connect it.
3. On EC2 instance run below commands for required softwares.

$ sudo yum update" to apply all updates.

$ sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins.io/redhat/jenkins.repo

$ sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key

$ sudo yum install jenkins –y

$ sudo wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.rpm>

$ sudo yum install -y jdk-8u141-linux-x64.rpm

$ sudo yum install docker

$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

rawath0408/customer-service 1.0-SNAPSHOT 39ccb9148cfe 14 hours ago 533MB

rawath0408/account-service 1.0-SNAPSHOT f15392aee321 15 hours ago 533MB

If any permission denied issue with docker, run the below command.

$ sudo chmod 666 /var/run/docker.sock

For running Jenkins :

$ sudo service jenkins start

Jenkins will run on 8080 default port. Use the instance URL and port to open Jenkins tool.

Ex: <http://ec2-15-206-157-247.ap-south-1.compute.amazonaws.com:8080/>

Create credentials and start using Jenkins.

1. Install below plugins on Jenkins and configure which are needed to build your application.

Manage Jenkins 🡪 Manage Plugins

Ant Plugin

Build Timeout

Command Agent Launcher Plugin

Docker Compose Build Step Plugin

Docker Pipeline

Email Extension Plugin

GitHub Branch Source Plugin

GitHub Plugin

Gradle Plugin

LDAP Plugin

Matrix Authorization Strategy Plugin

Oracle Java SE Development Kit Installer Plugin

OWASP Markup Formatter Plugin

PAM Authentication Plugin

Pipeline

Pipeline Maven Integration Plugin

Pipeline Utility Steps

Pipeline: GitHib Groovy Libraries

Timestamper

Workspace Cleanup Plugin

Note : Few of the above plugins are pre-installed.

1. Now Configure tools.

Manage Jenkins 🡪 Global Tool Configuration

JDK 🡪Add JDK

Name : Java <<anyname>>

JAVA\_HOME: /usr/java/jdk1.8.0\_141/ <<Path of Java\_Home on EC2 instance>>

Git 🡪 Add Git

Name:Git <<anyname>>

Path to Git Executable: /usr/bin/git <<Path of Git on EC2 instance>>

Maven 🡪 Add Maven

Name:maven<<anyname>> (this will be used in Jenkinsfile)

MAVEN\_HOME: / usr/share/apache-maven <<Path of Maven on EC2 instance>>

Docker 🡪 Add Docker

Name:MyDocker<<anyname>>

Installation root: /usr/bin/docker <<Path of Docker on EC2 instance>>

1. Create your Job on Jenkins. Select New Item, Give a name for your Job and select Pipeline or Freestyle Project (for regular build without pipeline).
2. Configure the Freestyle Project job for regular build.

Source Code Management :

Select Git option

Provide Repository details and branch details.

Build:

Select Execute Shell from Add Build Steps.

Add scripts to run build and run the application.

Ex: mvn clean install

Java –jar /xxxxx/xxx.jar

Save it

1. Configure the Pipeline job for build with pipeline.

Pipeline:

Under ‘Definition’ select ‘Pipeline script from SCM’.

Select Git under SCM and configure the Git repository details.

Unser ‘Script Path’ provide ‘./<<servicename>>/Jenkinsfile’

Ex (./account-service/Jenkinsfile)

Save it

1. Configure your credentials for Git, Docker repositories etc. under Global Credentials.

Manage Jenkins 🡪 Manage Credentials.

Select Global from ‘Stores scoped to Jenkins’.

Add Credentials