

step 1 = create ec2 instance and install java 17 after that install Jenkins

=====

Install Jenkins =

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

sudo apt update

sudo apt install jenkins

=====

step 2 = install maven on your Jenkins server [sudo apt install maven -y]

step 3 = Add a Maven tool in Jenkins pipeline

First, make sure:

You have Maven installed on Jenkins,

or go to

Manage Jenkins → Tools → Maven installations → Add Maven,

and name it (for example) "Maven3".

Do the same for JDK if needed — e.g., name it "jdk17".

go to

Manage Jenkins → Plugins → Available Plugins → Stage View Plugin

Manage Jenkins → Plugins → Available Plugins → Maven Integration Plugin

Jenkins Pipeline

```
pipeline {  
    agent any  
  
    tools {  
        maven 'Maven3'  
        jdk 'jdk17'  
    }  
  
    stages {  
        stage('Checkout') {  
            steps {  
                git branch: 'main', url: 'https://github.com/spring-projects/spring-petclinic.git'  
            }  
        }  
  
        stage('Build') {  
            steps {  
                sh 'mvn clean package'  
            }  
        }  
    }  
}
```

=====

Install docker on your Jenkins server

1. [sudo apt install docker.io] --> docker package install command
2. [docker --version] --> check docker version
3. [sudo chmod 777 /var/run/docker.sock] --> give full permission

4. [docker pull nginx] --> Image pull from dockerhub
5. [docker images] --> list docker images
6. [docker run -itd --name myapp -p 80:80 nginx] --> run docker container from image
7. go to security group -> add rule -> 80
8. <public ip>:80
8. [docker ps] --> check running containers
9. [docker ps -a] --> check stopped containers
10. [docker exec -it myapp /bin/bash] --> login into docker container
11. [exit] --> exit from login container
12. [docker stop <container-name>/id] --> stop running container

=====

DockerFile

FROM ubuntu:latest

RUN apt-get update

RUN apt install nginx -y

COPY index.html /var/www/html/

CMD ["nginx", "-g", "daemon off;"]

1. [sudo vim Dockerfile] --> past above dockerfile content
2. [docker build -t ubuntu-nginx .] --> build image from dockerfile
3. [docker images] --> list docker images
4. [docker rmi <image name>] --> remove docker images

=====

docker integrate with Jenkins

1. Set up Docker on the EC2 instance:

```
sudo apt update -y
```

```
sudo apt install docker.io -y
```

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

```
sudo systemctl status docker
```

2. Grant Jenkins user and ubuntu user permission to docker daemon.

```
sudo su -
```

```
usermod -aG docker jenkins
```

```
usermod -aG docker ubuntu
```

```
systemctl restart docker
```

3. Install Docker Tools and Docker Plugins:

Go to "Dashboard" in your Jenkins web interface.

Navigate to "Manage Jenkins" → "Manage Plugins."

Click on the "Available" tab and search for "Docker."

Check the following Docker-related plugins:

Docker

Docker Commons

Docker Pipeline

Docker API

docker-build-step

Click on the "Install without restart" button to install these plugins.

4. Add DockerHub Credentials:

To securely handle DockerHub credentials in your Jenkins pipeline, follow these steps:

Go to "Dashboard" → "Manage Jenkins" → "Manage Credentials."

Click on "System" and then "Global credentials (unrestricted)."

Click on "Add Credentials" on the left side.

Choose "Secret text" as the kind of credentials.

Enter your DockerHub credentials (Username and Password) and give the credentials an ID (e.g., "docker").

Click "OK" to save your DockerHub credentials.

5. Once you are done with the above steps, it is better to restart Jenkins.

<http://<ec2-instance-public-ip>:8080/restart>

6. go to manage Jenkins -> plugins -> available -> nodejs -> install

7. go to manage Jenkins -> tools -> add nodejs -> node16 -> apply -> save

8. go to manage Jenkins -> tools -> add docker -> docker -> apply -> save

9. install nodejs16 on your ec2 install

```
curl -sL https://deb.nodesource.com/setup_16.x | sudo bash -
```

```
sudo apt install nodejs -y
```

10.

```
pipeline {  
    agent any  
  
    tools{  
        jdk 'jdk17'  
        nodejs 'node16'  
    }  
  
    stages {  
        stage('Checkout') {  
            steps {  
                git branch: 'main', url: 'https://github.com/linuxdocs2025/devops_batch.git'  
            }  
        }  
  
        stage('Install Dependencies') {  
            steps {  
                sh "npm install"  
            }  
        }  
  
        stage("Docker Build & Push"){  
            steps{  
                script{  
                    withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){  
                        sh "docker build -t myntra ."  
                        sh "docker tag myntra1 <your-dockerhub-name>/myntra:latest "  
                        sh "docker push <your-dockerhub-name>/myntra:latest "                    }  
                }  
            }  
        }  
    }  
}
```

}

}

}

}

}

}