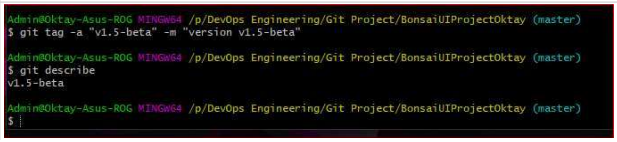
# kubectl describe services wordpress | grep -i port	

GETTING STARTED WITH DEVOPS GRADABLE ASSESMENTS

	1. **Dockerizing Jenkins Pipeline** DESCRIPTION Demonstrate the continuous integration and delivery by Dockerizing Jenkins Pipeline. **Problem Statement Scenario:** You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to Dockerize the Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for: • Availability of the application and its versions in the GitHub. ◦ Track their versions every time a code is committed to the repository. • Build the application in Docker and host it in Docker Hub. • Pull the Docker image and run it again. The company goal is to deliver the product frequently to the production with high-end quality. **You must use the following tools:** • Docker – To build the application in a Docker container and push it to Docker Hub • Docker Hub – To store the Docker image • GitHub – To store the application code and track its revisions • Git – To connect and push files from local system to GitHub • Linux (Ubuntu) – As a base operating system to start and execute the project • Jenkins – To automate the deployment process during continuous integration **Following requirements should be met:** • **Document the step-by-step process from the initial installation to the final stage.** • **Track the versions of the code in the GitHub repository** • **Availability of the application in the Docker Hub** • **Track the build status of Jenkins for every increment of the project**
• Availability of the application and its versions in the GitHub. **• Track their versions every time a code is committed to the repository.** Adding BonsaiUIProject in github # git init => to initialize the folder # git add . => to add all files and folders to staging area # git status => to see the status of the staged files/folders # git commit . -m "initial commit" => to commit changes in the staging area #git remote add origin "URL Location of the GIT" => to add your github project URL https://github.com/0ktay3min/DockerizingJenkinsPipeline.git #git push -u origin master = this will push the files to the remote repository Or #git push -f origin master = this will push the files to the remote repository. "-f stands for force commit"	
#git checkout 690c16c04813dc3532cf265575a8e863a53687ba => to check out commit ID	
# git tag -a "v1.5-beta" -m "Version v1.5-beta" => to version a code in github # git describe => to see all the versions in your github repository NOTE: do this to all commits	

git checkout master => to go back to the master branch

```
Admin@Oktay-Asus-ROG MINGW64 /p/DevOps Engineering/Git Project/BonsaiUIProjectOktay (master)
$ git log
commit 3f73c65b471bca71331ccfa1f4a1b16c79d0a1cf (HEAD -> master, tag: v1.5-beta, origin/master, main)
Author: Oktay Emin <oktay_emin@outlook.com>
Date:   Fri Oct 30 09:12:24 2020 -0500

    Extend Reports deleted

commit ca6808671ca6368e1f042eb9515b60b427f4e4ce (tag: v1.2-beta)
Author: Oktay Emin <oktay_emin@outlook.com>
Date:   Fri Oct 30 09:03:30 2020 -0500

    Test Screenshots deleted

commit 690c16c04813dc3532cf265575a8e863a53687ba (tag: v1.1-beta)
Author: Oktay Emin <oktay_emin@outlook.com>
Date:   Fri Oct 30 08:51:17 2020 -0500

    First Commit

Admin@Oktay-Asus-ROG MINGW64 /p/DevOps Engineering/Git Project/BonsaiUIProjectOktay (master)
$
```

• Build the application in Docker and host it in Docker Hub.

1- In BonsaiUIProject, create a Dockerfile
#nano Dokerfile => to create a docker file

1. Dockerizing Jenkins Pipeline

DESCRIPTION

Demonstrate the continuous integration and delivery by Dockerizing Jenkins Pipeline.

Problem Statement Scenario:
You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to Dockerize the Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for:

- Availability of the application and its versions in the GitHub.
  - Track their versions every time a code is committed to the repository.
- Build the application in Docker and host it in Docker Hub.
- Pull the Docker image and run it again.

The company goal is to deliver the product frequently to the production with high-end quality.

You must use the following tools:

- Docker – To build the application in a Docker container and push it to Docker Hub
- Docker Hub – To store the Docker image
- GitHub – To store the application code and track its revisions
- Git – To connect and push files from local system to GitHub
- Linux (Ubuntu) – As a base operating system to start and execute the project
- Jenkins – To automate the deployment process during continuous integration

Following requirements should be met:

- **Document the step-by-step process from the initial installation to the final stage.**
- **Track the versions of the code in the GitHub repository**
- **Availability of the application in the Docker Hub**
- **Track the build status of Jenkins for every increment of the project**

1. Installing Jenkins

sudo apt update
sudo apt install openjdk-8-jdk => to install Java on your local PC

```
ares@kubernetes-master:~$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment (build 1.8.0_272-8u272-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.272-b10, mixed mode)
ares@kubernetes-master:~$
```

java -version => to see what version of Java is being installed on your System.

wget –q –O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add – => This will import GPG Keys

sudo nano /etc/apt/sources.d => this will open sources.d file

Jenkins software repository
deb https://pkg.jenkins.io/debian binary => add this line inside sources.d file.

sudo apt install jenkins => this will install Jenkins in your system.

```
 GNU nano 4.8                                    /etc/apt/sources.list
# deb cdrom:[Ubuntu 20.04.1 LTS _Focal Fossa_ - Release amd64 (20200731)]/ focal main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://us.archive.ubuntu.com/ubuntu/ focal main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal universe
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://us.archive.ubuntu.com/ubuntu/ focal multiverse
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal multiverse
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates multiverse
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu focal partner
# deb-src http://archive.canonical.com/ubuntu focal partner

deb http://security.ubuntu.com/ubuntu focal-security main restricted
# deb-src http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
# deb-src http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
```

```
deb http://security.ubuntu.com/ubuntu focal-security main restricted
# deb-src http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
# deb-src http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
# deb-src http://security.ubuntu.com/ubuntu focal-security multiverse

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.


deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main

deb https://pkg.jenkins.io/debian-stable binary/
#deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable
# deb-src [arch=amd64] https://download.docker.com/linux/ubuntu focal stable
# deb-src http://apt.kubernetes.io/ kubernetes-xenial main
```

sudo service jenkins status => to see if Jenkins running
sudo service jenkins start => to start Jenkins

```
ares@kubernetes-master:~$ sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Sun 2020-11-01 08:55:07 CST; 18min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 9487)
     Memory: 0B
     CGroup: /system.slice/jenkins.service

Nov 01 08:55:03 kubernetes-master systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 01 08:55:05 kubernetes-master jenkins[1143]: Correct java version found
Nov 01 08:55:05 kubernetes-master jenkins[1143]:  * Starting Jenkins Automation Server jenkins
Nov 01 08:55:05 kubernetes-master su[1688]: (to jenkins) root on none
Nov 01 08:55:05 kubernetes-master su[1688]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Nov 01 08:55:06 kubernetes-master su[1688]: pam_unix(su-l:session): session closed for user jenkins
Nov 01 08:55:07 kubernetes-master jenkins[1143]:    ...done.
Nov 01 08:55:07 kubernetes-master systemd[1]: Started LSB: Start Jenkins at boot time.
```

sudo ufw allow 8080 => to allow port 8080 in your firewall

```
ares@kubernetes-master:~$ sudo ufw allow 8080
Rules updated
Rules updated (v6)
ares@kubernetes-master:~$
```

sudo ufw status => to verify if ufw firewall is active

```
ares@kubernetes-master:~$ sudo ufw status
Status: inactive
ares@kubernetes-master:~$
```

sudo ufw enable => to enable ufw firewall

```
ares@kubernetes-master:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

2-Installing Docker

sudo apt install apt-transport-https ca-certificates curl software-properties-common => to install few prerequisites

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - => To add GPG key

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable" => to add Docker Repository to APT sources

apt-cache policy docker-ce =>

sudo apt install docker-ce => this will install Docker in your system.

sudo service docker status => to see the status of Docker

```
ares@kubernetes-master:/etc/ufw$ sudo service docker status
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-11-01 08:55:10 CST; 38min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1479 (dockerd)
      Tasks: 58
     Memory: 224.5M
     CGroup: /system.slice/docker.service
             └─1479 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

3-Creating Docker file

sudo touch Dokerfile => this will create docker file
sudo nano Dockerfile => to open Dockerfile using nano text editor and the following lines inside the Dockerfile.

```
FROM jenkins/jenkins:lts
USER root
RUN apt-get update && \
apt-get -y install apt-transport-https \
  ca-certificates \
  curl \
  gnupg2 \
  software-properties-common && \
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg > /tmp/dkey;
apt-key add /tmp/dkey && \
add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \
  $(lsb_release -cs) \
  stable" && \
apt-get update && \
apt-get -y install docker-ce
RUN apt-get install -y docker-ce
RUN usermod -a -G docker jenkins
USER jenkins
```

```
FROM jenkins/jenkins:lts
USER root
RUN apt-get update && \
apt-get -y install apt-transport-https \
    ca-certificates \
    curl \
    gnupg2 \
    software-properties-common && \
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg > /tmp/dkey; apt-key add /tmp/dkey && \
add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \
    $(lsb_release -cs) \
    stable" && \
apt-get update && \
apt-get -y install docker-ce
RUN apt-get install -y docker-ce
RUN usermod -a -G docker jenkins
USER jenkins
```

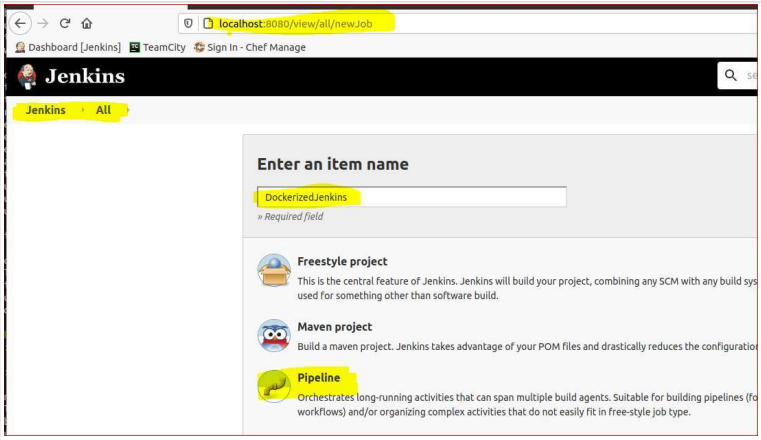
sudo docker built -t ubuntu . => to build Ubuntu image in docker

```
ares@kubernetes-master:~/Projects/DockerizingJenkinsPipeline$ sudo docker build -t ubuntu .
Sending build context to Docker daemon  2.027MB
Step 1/6 : FROM jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
3192219afd04: Pull complete
17c160265e75: Pull complete
cc4fe40d0e61: Pull complete
9d647f502a07: Pull complete
d108b8c498aa: Pull complete
1bfe918b8aa5: Pull complete
dafa1a7c0751: Pull complete
4871da813f56: Pull complete
24fa635d5c7a: Pull complete
72d7a7deec14: Pull complete
78be6e7f3e82: Pull complete
026ea665e4e8: Pull complete
f7fc83e75d1c: Pull complete
8f13ab868e19: Pull complete
af14f7667e07: Pull complete
e9a93a70ada9: Pull complete
12b275276736: Pull complete
84dbca0b43d9: Pull complete
3fada0df691f: Pull complete
fbf81cac2cc9: Pull complete
Digest: sha256:aa1ec2a6a106608a286dc06c334053fced1df87a7cb307853ca1cf8fa48ba09b
Status: Downloaded newer image for jenkins/jenkins:lts
```

```
 ---> 85d19818f04c
Successfully built 85d19818f04c
Successfully tagged ubuntu:latest
ares@kubernetes-master:~/Projects/DockerizingJenkinsPipeline$
```

4-Adding New Item on Jenkins

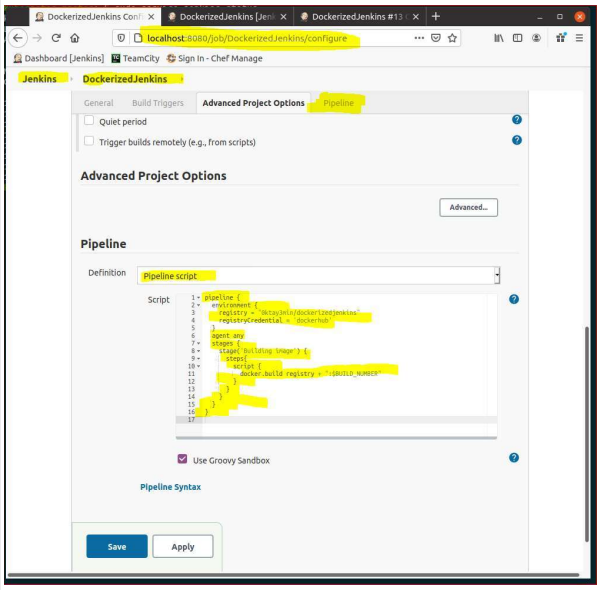
- login to your Jenkins using your browser (Ex: https://localhost:8080 . Make sure you enter previously created user credentials.
- Click New Item and enter "DockerizedJenkins", then select Pipeline and click "OK" button.



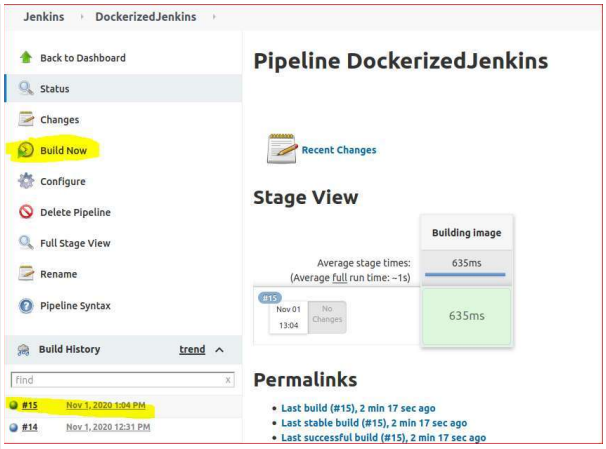
- Click Pipeline Tab and then enter the following in the Pipeline Scrip box

```
pipeline {
 environment {
  registry = "0ktay3min/dockerizedjenkins"
  registryCredential = 'dockerhub'
 }
 agent any
 stages {
  stage('Building image') {
   steps{
    script {
     docker.build registry + ":$BUILD_NUMBER"
    }
   }
  }
 }
}
```

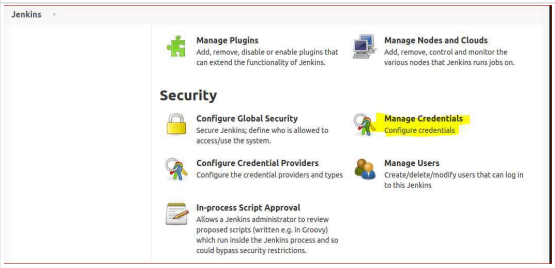
- Once done click Save button

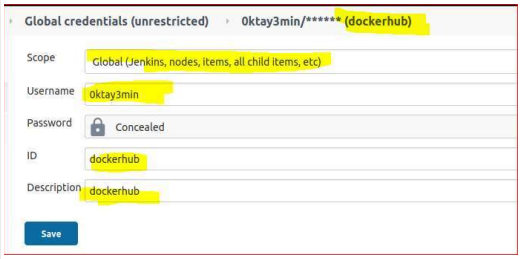


- Click Build Now button to build to project



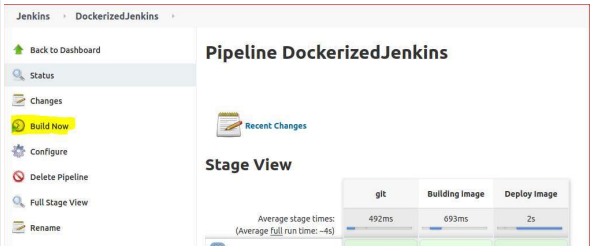
- Since we created DockerHub account previously, we need to add dockerhub login credentials to Jenkins. To do this, follow the steps below
  1. Click Manage Jenkins
  2. Click Manage Credentials





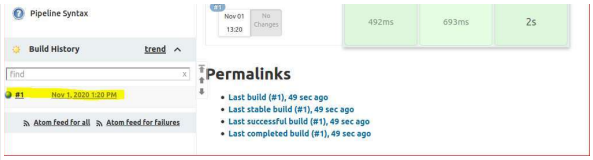
- Make sure you enter your Dockerhub account and repository name in the Jenkins file. Gitbub URL must be entered with valid user login credentials.
  1. Enter the Pipeline script below and click save

```
pipeline {
  environment {
    registry = "0ktay3min/dockerizedjenkins"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent any
  stages {
    stage("git") {
      steps {
        git credentialsId: 'github', url:
'https://github.com/0ktay3min/DockerizingJenkinsPipeline.git'
      }
    }
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
    stage('Deploy Image') {
      steps{
        script {
          docker.withRegistry( '', registryCredential ) {
            dockerImage.push()
          }
```

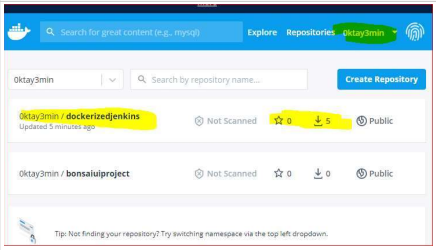


```
    }
  }
 }
}
```

- Click Build Now button

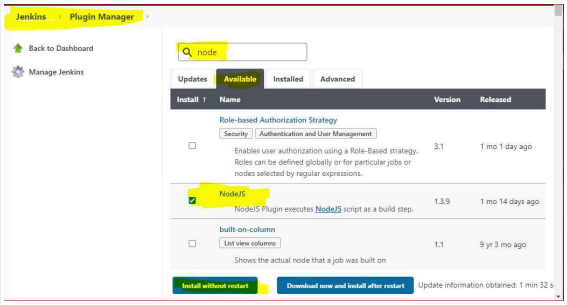


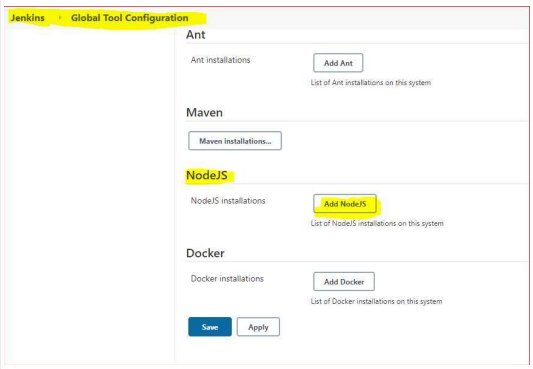
- Verify if docker image is being pushed the Docker hub



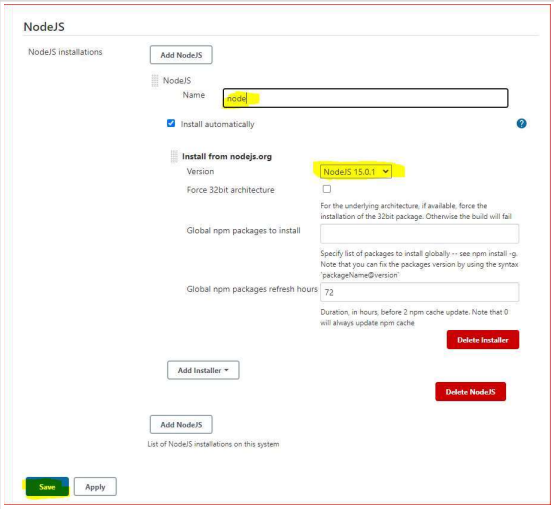
5-Adding NodeJs Plugin to Jenkins

- Click Manage Jenkins
- Click Manage Plugins
- Click available tab and search for NodeJs
- Select NodeJs and click Install without restart
- Now go back to the Manage Jenkins page and click Global Tool Configuration.
- In Global Tool Configuration locate NodeJs and click Add NodeJs



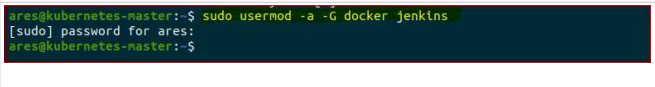


- Enter "node" as a name
- Select any version that is greater than V9
- Then click save



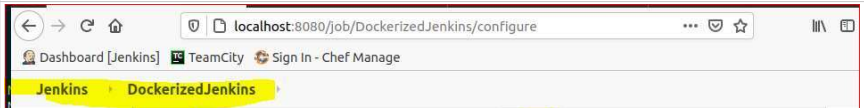
- Make sure you give Directory level access to Jenkins and Docker. To do this, enter the following command below.

sudo usermod -a -G docker jenkins => To grant directory access to docker and Jenkins.



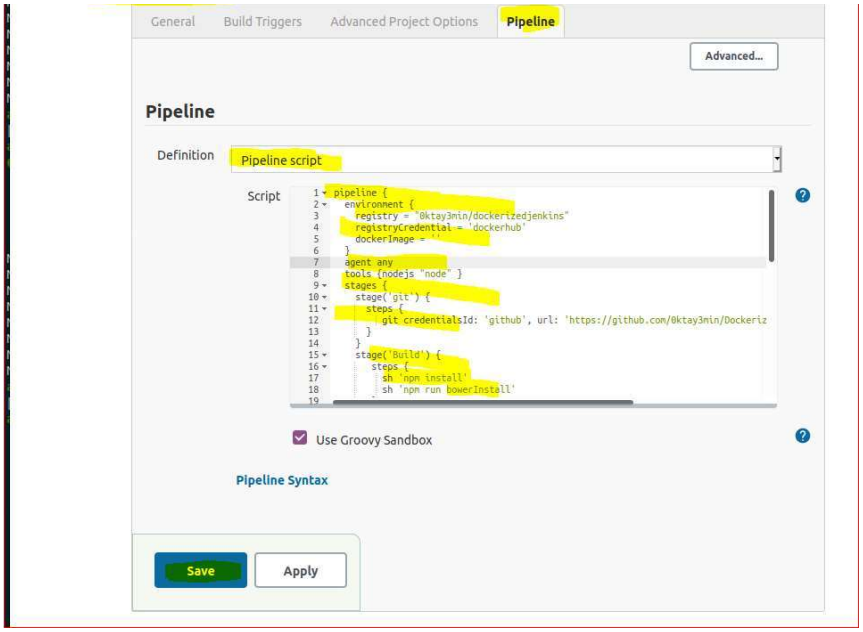
- Make sure you replace the Pipeline script with the one below

```
pipeline {
  environment {
```

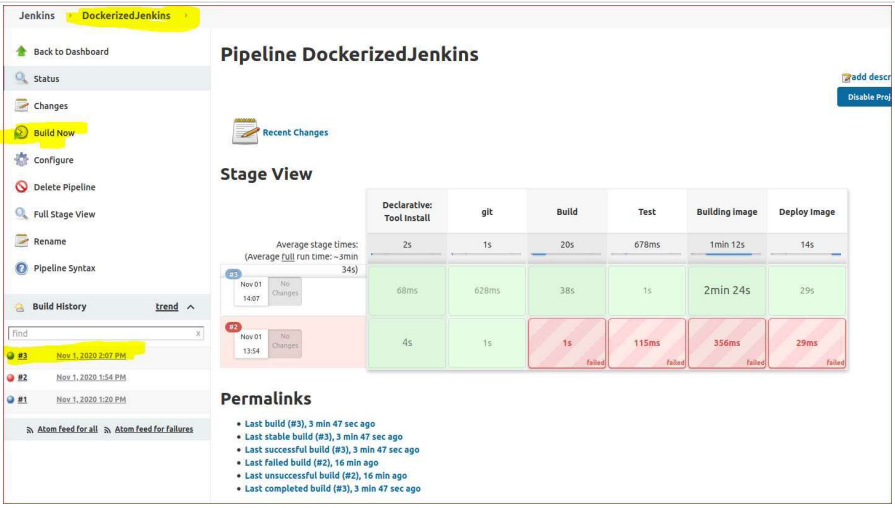


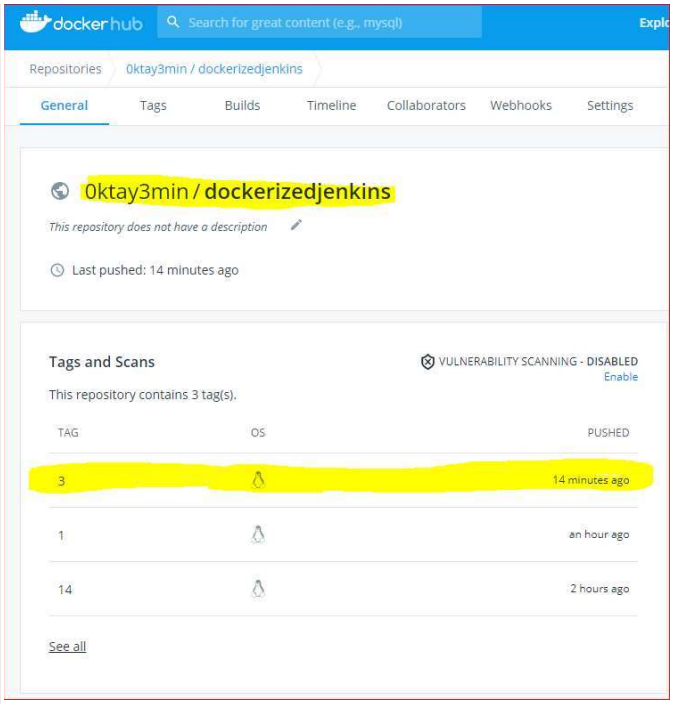
```
    registry = "0ktay3min/dockerizedjenkins"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent any
  tools {nodejs "node" }
  stages {
    stage('git') {
      steps {
        git credentialsId: 'github', url:
'https://github.com/0ktay3min/DockerizingJenkinsPipeline.git'
      }
    }
    stage('Build') {
      steps {
        sh 'npm install'
        sh 'npm run bowerInstall'
      }
    }
    stage('Test') {
      steps {
        sh 'npm test'
      }
    }
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
    stage('Deploy Image') {
      steps{
        script {
          docker.withRegistry( '', registryCredential ) {
            dockerImage.push()
          }
        }
      }
    }
  }
}
```

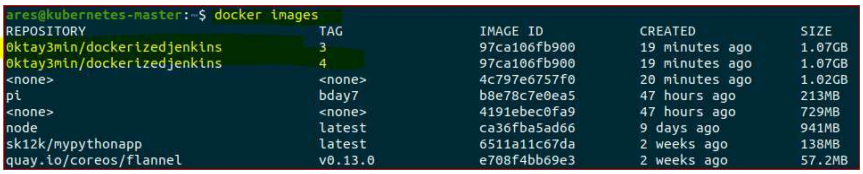
- Click Save Button



- Click Build Now button and wait for Jenkins to build the project
- Once done you can verify if image is being pushed to docker hub




docker hub
Search for great content (e.g., mysql)
Repositories
0ktay3min / dockerizedjenkins
General
Tags
Builds
Timeline
Collaborators
Webhooks
Settings
0ktay3min / dockerizedjenkins
This repository does not have a description
Last pushed: 14 minutes ago
Tags and Scans
VULNERABILITY SCANNING - DISABLED
Enable
This repository contains 3 tag(s).
TAG
OS
PUSHED
3
14 minutes ago
1
an hour ago
14
2 hours ago
See all


ares@kubernetes-master:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
0ktay3min/dockerizedjenkins 3 97ca106fb900 19 minutes ago 1.07GB
0ktay3min/dockerizedjenkins 4 97ca106fb900 19 minutes ago 1.07GB
<none> <none> 4c797e6757f0 20 minutes ago 1.02GB
pi bday7 b8e78c7e0ea5 47 hours ago 213MB
<none> <none> 4191ebec0fa9 47 hours ago 729MB
node latest ca36fba5ad66 9 days ago 941MB
sk12k/mypythonapp latest 6511a11c67da 2 weeks ago 138MB
quay.io/coreos/flannel v0.13.0 e708f4bb69e3 2 weeks ago 57.2MB