

# Dockerizing the Jenkins Pipeline - Project 1

## Prerequisites:

- Docker – To build the application in a Docker container and push it to Docker Hub
- Docker Hub – You need the account and one repository to push the docker image
- GitHub – to store application code
- Git – to pull and push the app code
- Linux (Ubuntu) – As a base operating system to start and execute the project
- Jenkins – To automate the deployment process during continuous integration

Before going to step 1 make sure you have Jenkins installed and configured with required plug-ins

*\*2 Plug-ins required for this project, NodeJS and Docker Pipeline, steps are listed below.*

## References:

Github Project repository:

<https://github.com/anisha-thakkar2020/SimpliLearnDevOpsProj1>

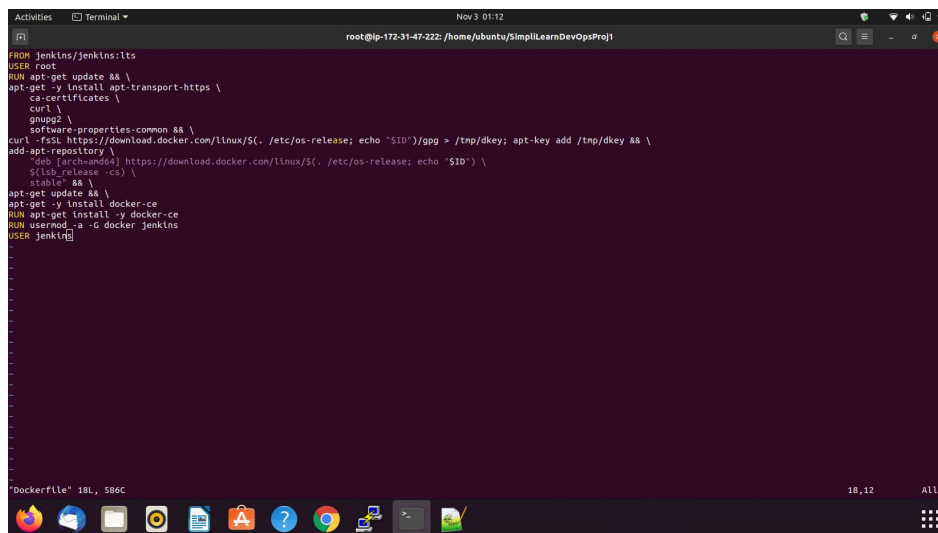
Dockerhub Repository:

[https://hub.docker.com/repository/docker/anishamthakkar/dockerizing\\_jenkins1](https://hub.docker.com/repository/docker/anishamthakkar/dockerizing_jenkins1)

## Instructions:

### Step 1 - Create the Dockerfile

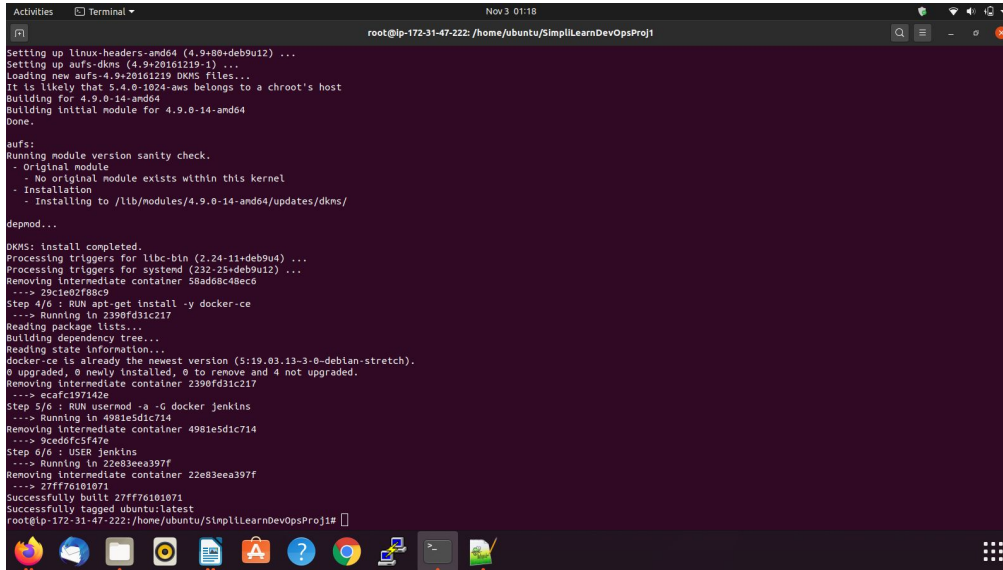
<https://github.com/anisha-thakkar2020/SimpliLearnDevOpsProj1/blob/main/Dockerfile>

A screenshot of a terminal window titled "Activities" and "Terminal". The terminal shows the content of a Dockerfile. The Dockerfile starts with "FROM jenkins/jenkins:its", sets the user to "root", and runs several commands: "apt-get update", "apt-get install apt-transport-https", "curl -s 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 install docker-ce", and "usermod -s /bin/bash docker\_jenkins". The terminal output shows the progress of these commands, including the installation of docker-ce and the modification of the user "docker\_jenkins". The terminal window is running on a system with IP 172.31.47.222, and the current directory is /home/ubuntu/SimpliLearnDevOpsProj1. The terminal window is titled "Dockerfile" 1BL, 586C. The bottom of the terminal window shows the system tray with various icons and the date/time "18,12 All".

```
FROM jenkins/jenkins:its
USER root
RUN apt-get update && \
  apt-get -y install apt-transport-https \
  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 -s /bin/bash docker_jenkins
USER jenkins
```

## Step 2 Build Docker image with below command

docker build -t ubuntu .



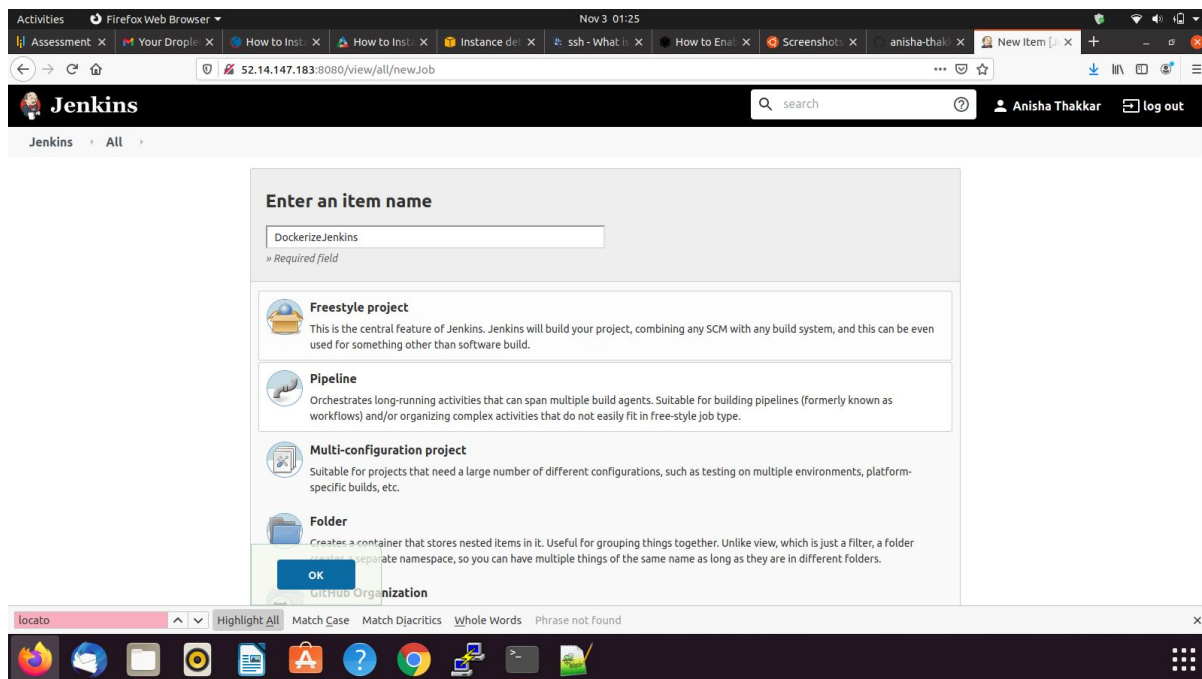
```
Setting up linux-headers-amd64 (4.9.0-14-amd64) ...
Setting up aufs-dkms (4.9+20161219-1) ...
Loading new aufs-4.9+20161219 DKMS files...
It is likely that 5.4.0-1024-aws belongs to a chroot's host
Building for 4.9.0-14-amd64
Building initial module for 4.9.0-14-amd64
Done.
aufs:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/4.9.0-14-amd64/updates/dkms/
depmod...
DKMS: install completed.
Processing triggers for lsb-bin (2.24-11+deb9u4) ...
Processing triggers for systemd (232-25+deb9u1) ...
Removing intermediate container 58ad08c48ec0
--> 29c1e02f8ac9
Step 4/6 : RUN apt-get install -y docker-ce
--> Running in 2390fd31c217
Reading package lists...
Building dependency tree...
Reading state information...
docker-ce is already the newest version (5:19.03.13-3-0-debian-stretch).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
Removing intermediate container 2390fd31c217
--> ecaf197142e
Step 5/6 : RUN usermod -o -G docker jenkins
--> Running in 4981e5d1c714
Removing intermediate container 4981e5d1c714
--> 9ced0f3f47e
Step 6/6 : USER jenkins
--> Running in 22e83eea397f
Removing intermediate container 22e83eea397f
--> 27ff76101071
Successfully built 27ff76101071
Successfully tagged ubuntu:latest
root@ip-172-31-47-222:/home/ubuntu/SimpliLearnDevOpsProj1#
```

## Step 3 - Create the Pipeline in Jenkins

### 3.1 Jenkins->New Item->

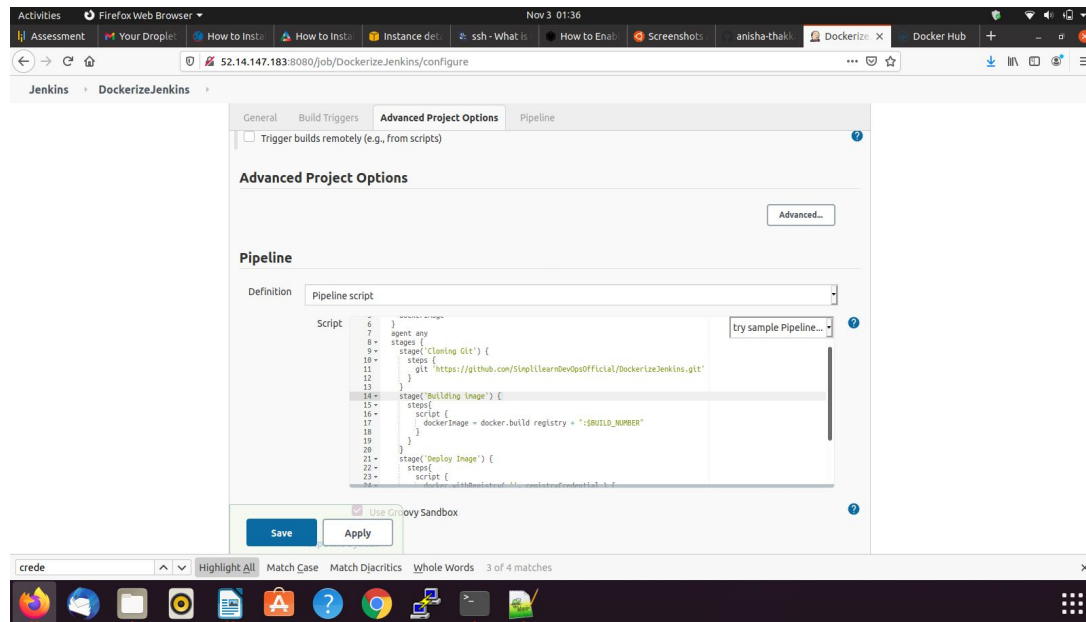
Give the project name DockerizingJenkins

Select the Pipeline as a project type



3.2 Copy the script from below link in the Pipeline script in Jenkins and click Save

[https://github.com/anisha-thakkar2020/SimpliLearnDevOpsProj1/blob/main/PipelineScript\\_DockerImage](https://github.com/anisha-thakkar2020/SimpliLearnDevOpsProj1/blob/main/PipelineScript_DockerImage)



#### Step 4. Add Dockerhub credential

4.1 Go to Manage Jenkins->Manage Credentials->Global Credentials

4.2 Add your Docker hub credentials, give “dockerhb” for ID and description field and save it.

#### Step 5. Download NodeJs and Docker Pipeline plug-ins

5.1 Go to Manage Jenkins->Manage Plug-ins->Available

5.2 Search for “**NodeJS**”, and install the plug-in

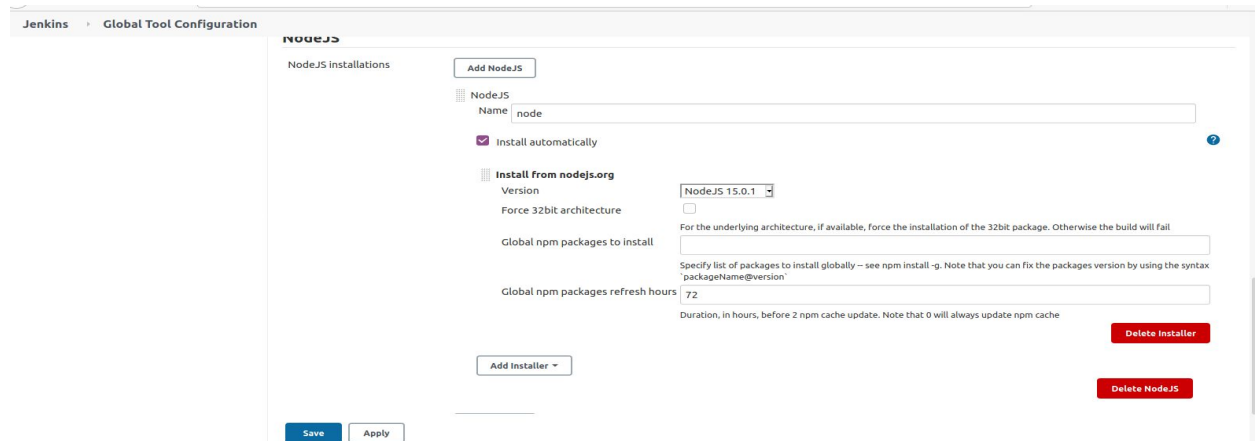
5.3 Search for “**Docker Pipeline**” and install it

5.4 Restart Jenkins once plug-ins are installed.

## Step 6 Configure Node.js

6.1 Go to Manage Jenkins->Global Tool Configuration

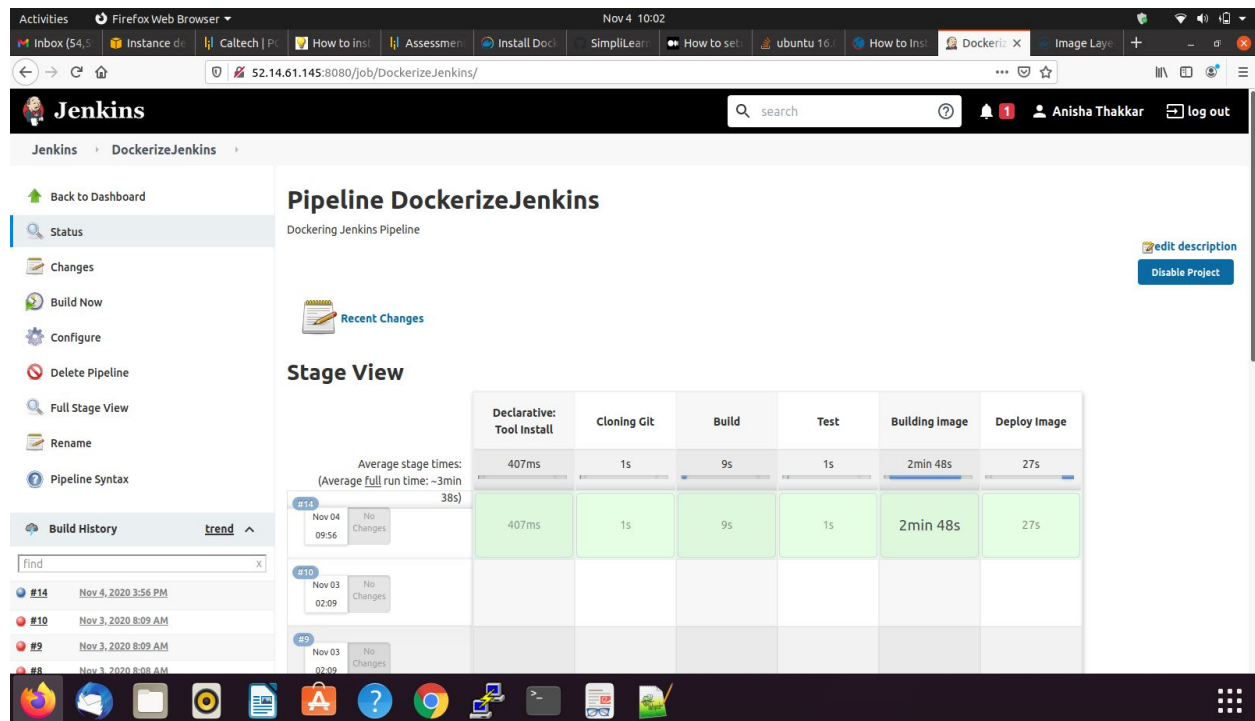
6.2 Under NodeJs, give the name as “node” and select nodejs 9+ version to install



The screenshot shows the Jenkins 'Global Tool Configuration' page for 'NodeJS'. The 'Name' field is set to 'node'. The 'Install automatically' checkbox is checked. Under 'Install from nodejs.org', the 'Version' is set to 'NodeJS 15.0.1'. The 'Force 32bit architecture' checkbox is unchecked. The 'Global npm packages to install' field is empty. The 'Global npm packages refresh hours' is set to '72'. There are buttons for 'Add NodeJS', 'Add Installer', 'Delete Installer', and 'Delete NodeJS'. At the bottom, there are 'Save' and 'Apply' buttons.

## Step 7. Build the Pipeline Project

7.1 Go to Jenkins->DockerizeJenkins->Build Now



The screenshot shows the Jenkins 'Pipeline DockerizeJenkins' page. The left sidebar contains links for 'Back to Dashboard', 'Status', 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. The main area displays the 'Pipeline DockerizeJenkins' with a 'Dockering Jenkins Pipeline' description. Below this is a 'Recent Changes' section. The 'Stage View' table shows the following stages and their durations:

	Declarative: Tool Install	Cloning Git	Build	Test	Building image	Deploy image
Average stage times: (Average full run time: ~3min 38s)	407ms	1s	9s	1s	2min 48s	27s
#14 Nov 04 09:56	407ms	1s	9s	1s	2min 48s	27s
#10 Nov 03 02:09						
#9 Nov 03 02:09						

The bottom of the page shows a 'Build History' section with a search bar and a list of builds. The bottom of the browser window shows the Ubuntu 16.04 desktop environment with various application icons.