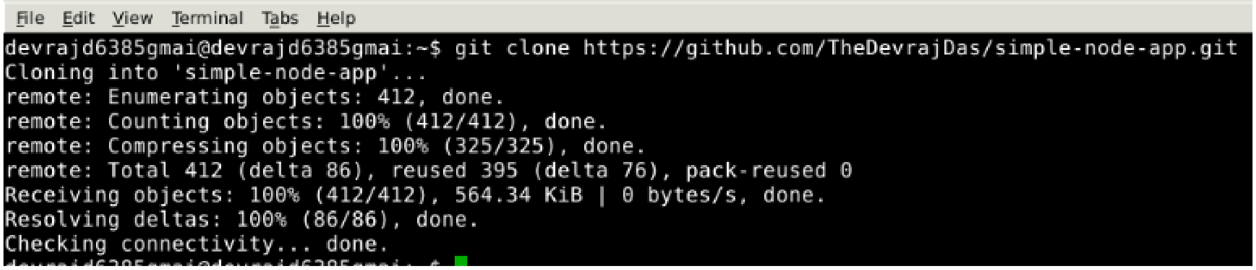
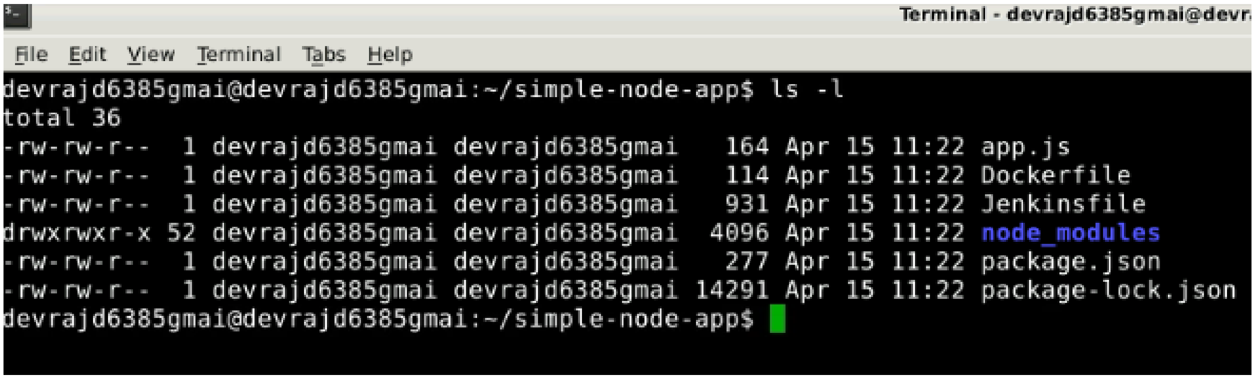
Project title: Build a Docker Jenkins Pipeline to Implement CI/CD Workflow

1. git clone https://github.com/TheDevrajDas/simple-node-app.git



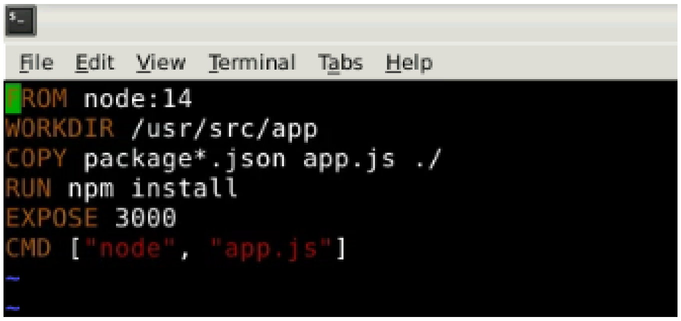
```
devrajd6385gmai@devrajd6385gmai:~$ git clone https://github.com/TheDevrajDas/simple-node-app.git
Cloning into 'simple-node-app'...
remote: Enumerating objects: 412, done.
remote: Counting objects: 100% (412/412), done.
remote: Compressing objects: 100% (325/325), done.
remote: Total 412 (delta 86), reused 395 (delta 76), pack-reused 0
Receiving objects: 100% (412/412), 564.34 KiB | 0 bytes/s, done.
Resolving deltas: 100% (86/86), done.
Checking connectivity... done.
```

2. cd simple-node-app
3. ls -l (check for Dockerfile)


Terminal - devrajd6385gmai@devr
File Edit View Terminal Tabs Help

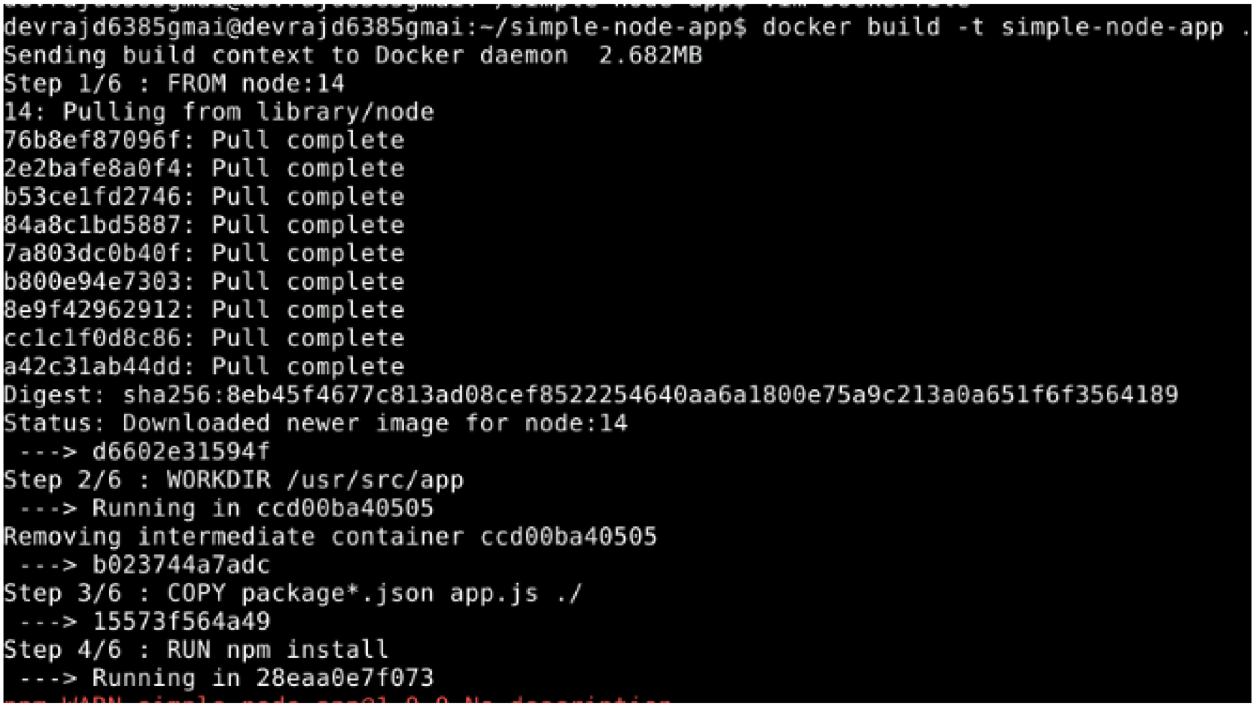
```
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ ls -l
total 36
-rw-rw-r--  1 devrajd6385gmai devrajd6385gmai   164 Apr 15 11:22 app.js
-rw-rw-r--  1 devrajd6385gmai devrajd6385gmai   114 Apr 15 11:22 Dockerfile
-rw-rw-r--  1 devrajd6385gmai devrajd6385gmai   931 Apr 15 11:22 Jenkinsfile
drwxrwxr-x 52 devrajd6385gmai devrajd6385gmai  4096 Apr 15 11:22 node_modules
-rw-rw-r--  1 devrajd6385gmai devrajd6385gmai   277 Apr 15 11:22 package.json
-rw-rw-r--  1 devrajd6385gmai devrajd6385gmai 14291 Apr 15 11:22 package-lock.json
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$
```

4. Vim Dockerfile (to see its contents.)

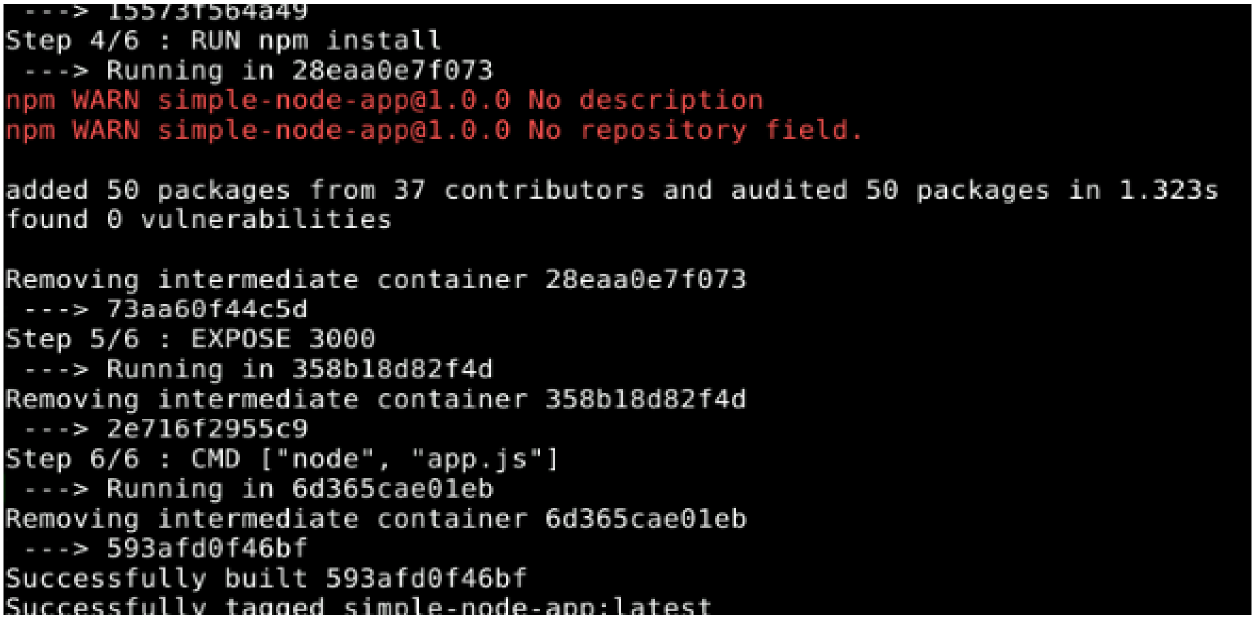


```
FROM node:14
WORKDIR /usr/src/app
COPY package*.json app.js ./
RUN npm install
EXPOSE 3000
CMD ["node", "app.js"]
~
~
```

5. docker build -t simple-node-app . (to build the docker file)

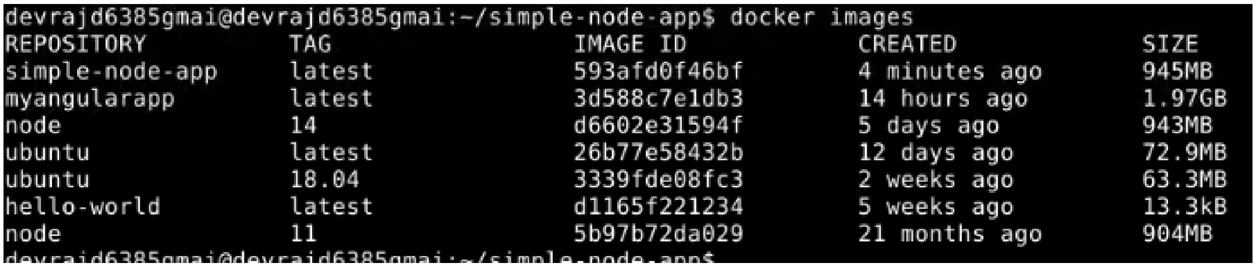


```
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ vim Dockerfile
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ docker build -t simple-node-app .
Sending build context to Docker daemon  2.682MB
Step 1/6 : FROM node:14
14: Pulling from library/node
76b8ef87096f: Pull complete
2e2bafe8a0f4: Pull complete
b53ce1fd2746: Pull complete
84a8c1bd5887: Pull complete
7a803dc0b40f: Pull complete
b800e94e7303: Pull complete
8e9f42962912: Pull complete
cc1c1f0d8c86: Pull complete
a42c31ab44dd: Pull complete
Digest: sha256:8eb45f4677c813ad08cef8522254640aa6a1800e75a9c213a0a651f6f3564189
Status: Downloaded newer image for node:14
 ---> d6602e31594f
Step 2/6 : WORKDIR /usr/src/app
 ---> Running in ccd00ba40505
Removing intermediate container ccd00ba40505
 ---> b023744a7adc
Step 3/6 : COPY package*.json app.js ./
 ---> 15573f564a49
Step 4/6 : RUN npm install
 ---> Running in 28eaa0e7f073
npm WARN simple-node-app@1.0.0 No description
```



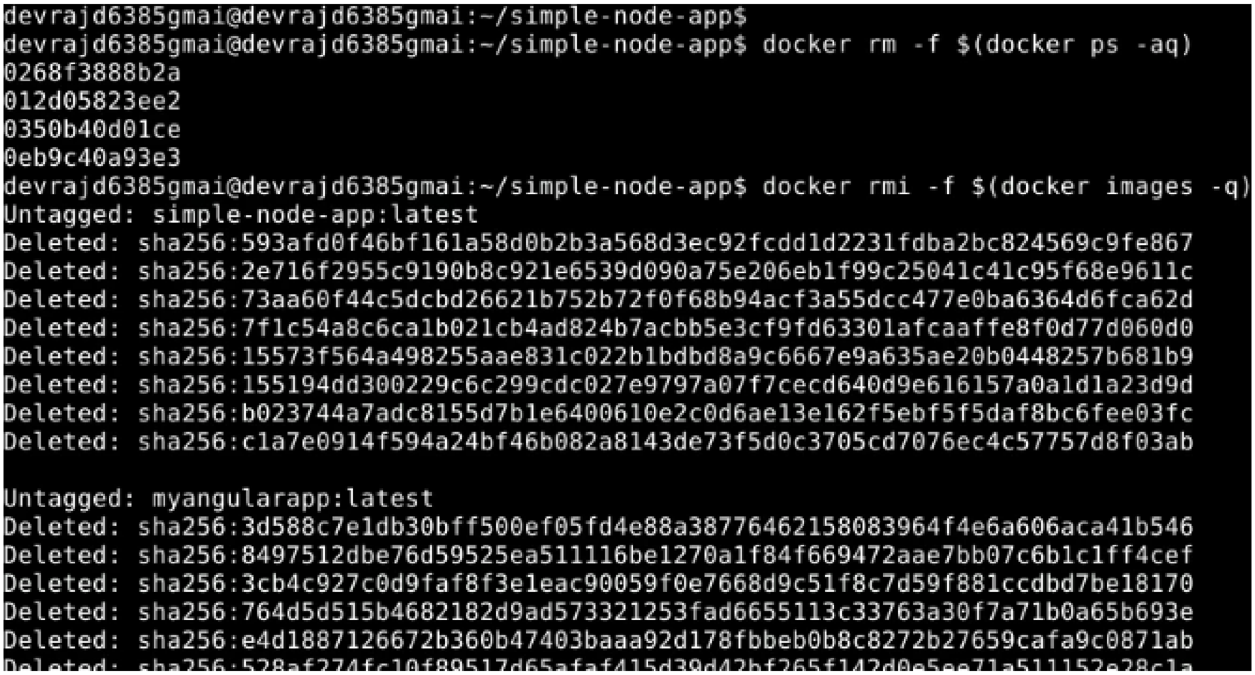
```
 ---> 15573f564a49
Step 4/6 : RUN npm install
 ---> Running in 28eaa0e7f073
npm WARN simple-node-app@1.0.0 No description
npm WARN simple-node-app@1.0.0 No repository field.

added 50 packages from 37 contributors and audited 50 packages in 1.323s
found 0 vulnerabilities

Removing intermediate container 28eaa0e7f073
 ---> 73aa60f44c5d
Step 5/6 : EXPOSE 3000
 ---> Running in 358b18d82f4d
Removing intermediate container 358b18d82f4d
 ---> 2e716f2955c9
Step 6/6 : CMD ["node", "app.js"]
 ---> Running in 6d365cae01eb
Removing intermediate container 6d365cae01eb
 ---> 593afd0f46bf
Successfully built 593afd0f46bf
Successfully tagged simple-node-app:latest
```

6. docker images (Check for simple-node-app image)



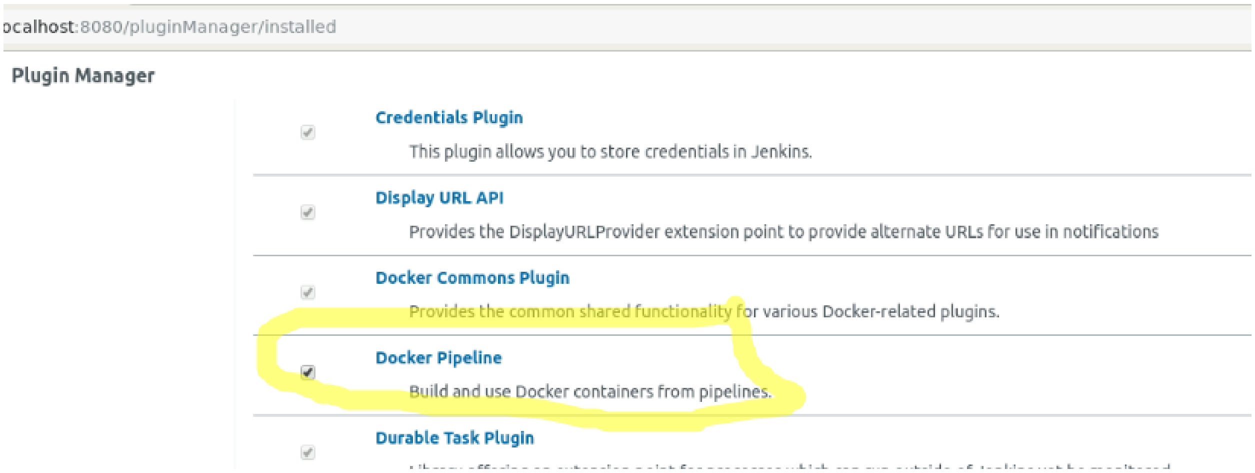
```
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
simple-node-app     latest              593afd0f46bf        4 minutes ago       945MB
myangularapp        latest              3d588c7e1db3        14 hours ago        1.97GB
node                14                  d6602e31594f        5 days ago          943MB
ubuntu              latest              26b77e58432b        12 days ago         72.9MB
ubuntu              18.04               3339fde08fc3        2 weeks ago         63.3MB
hello-world         latest              d1165f221234        5 weeks ago         13.3kB
node                11                  5b97b72da029        21 months ago       904MB
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$
```

7. Clean up running containers and images Run following commands :
   docker rm -f $(docker ps -aq) ---> Removing all running containers
   docker rmi -f $(docker images -q) ---> Removing all the images



```
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ docker rm -f $(docker ps -aq)
0268f3888b2a
012d05823ee2
0350b40d01ce
0eb9c40a93e3
devrajd6385gmai@devrajd6385gmai:~/simple-node-app$ docker rmi -f $(docker images -q)
Untagged: simple-node-app:latest
Deleted: sha256:593afd0f46bf161a58d0b2b3a568d3ec92fcdd1d2231fdba2bc824569c9fe867
Deleted: sha256:2e716f2955c9190b8c921e6539d090a75e206eb1f99c25041c41c95f68e9611c
Deleted: sha256:73aa60f44c5dcbd26621b752b72f0f68b94acf3a55dcc477e0ba6364d6fca62d
Deleted: sha256:7f1c54a8c6ca1b021cb4ad824b7acbb5e3cf9fd63301afcaaffe8f0d77d060d0
Deleted: sha256:15573f564a498255aae831c022b1bdbd8a9c6667e9a635ae20b0448257b681b9
Deleted: sha256:155194dd300229c6c299cdc027e9797a07f7cecd640d9e616157a0a1d1a23d9d
Deleted: sha256:b023744a7adc8155d7b1e6400610e2c0d6ae13e162f5ebf5f5daf8bc6fee03fc
Deleted: sha256:c1a7e0914f594a24bf46b082a8143de73f5d0c3705cd7076ec4c57757d8f03ab

Untagged: myangularapp:latest
Deleted: sha256:3d588c7e1db30bff500ef05fd4e88a38776462158083964f4e6a606aca41b546
Deleted: sha256:8497512dbe76d59525ea511116be1270a1f84f669472aae7bb07c6b1c1ff4cef
Deleted: sha256:3cb4c927c0d9faf8f3e1eac90059f0e7668d9c51f8c7d59f881ccdbd7be18170
Deleted: sha256:764d5d515b4682182d9ad573321253fad6655113c33763a30f7a71b0a65b693e
Deleted: sha256:e4d1887126672b360b47403baaa92d178fbbeb0b8c8272b27659cafa9c0871ab
Deleted: sha256:528af274fc10f89517d65afaf415d39d42bf265f142d0e5ee71a511152e28c1a
```

8. Install Jenkins and login
9. Manage Jenkins ---> Manage Plugins ---> Search for “docker pipeline” --> select it and install without restart.



ocalhost:8080/pluginManager/installed

Plugin Manager

☑ **Credentials Plugin**
This plugin allows you to store credentials in Jenkins.

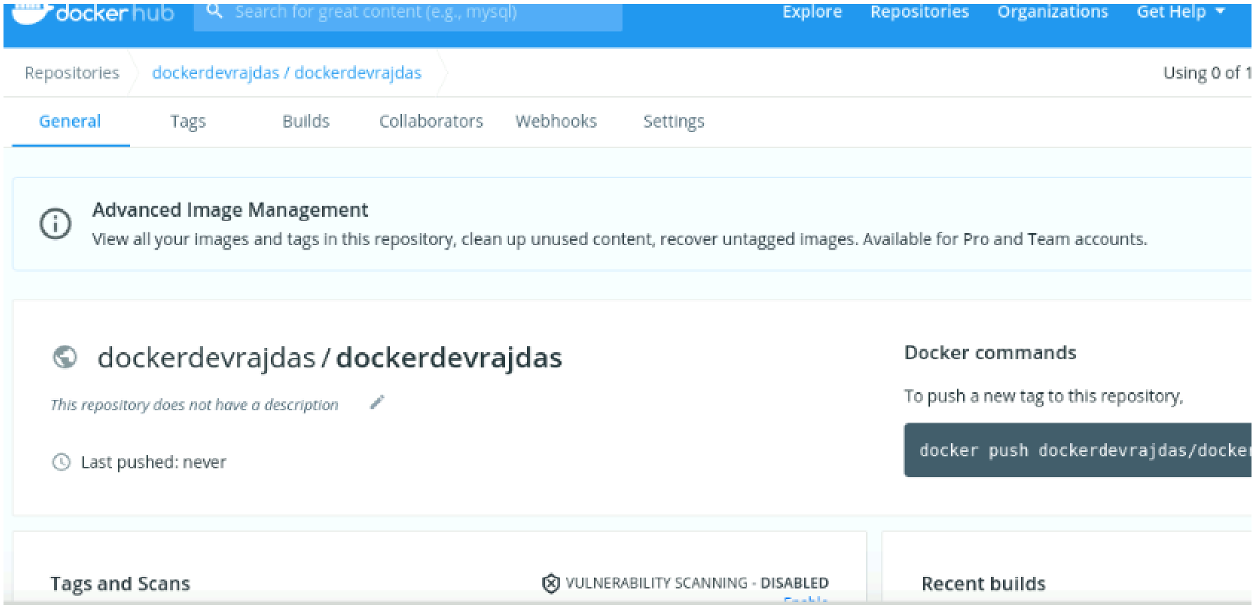
☑ **Display URL API**
Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications

☑ **Docker Commons Plugin**
Provides the common shared functionality for various Docker-related plugins.

☑ **Docker Pipeline**
Build and use Docker containers from pipelines.

☑ **Durable Task Plugin**

10. Visit https://hub.docker.com/ and create a docker hub account
Create Repository to push our docker image → Click on create repository, add name and click create.



Repositories › dockerdevrajdas / dockerdevrajdas

Using 0 of 1

Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

dockerdevrajdas / dockerdevrajdas

This repository does not have a description

Last pushed: never

Docker commands

To push a new tag to this repository,

```
docker push dockerdevrajdas/docke
```

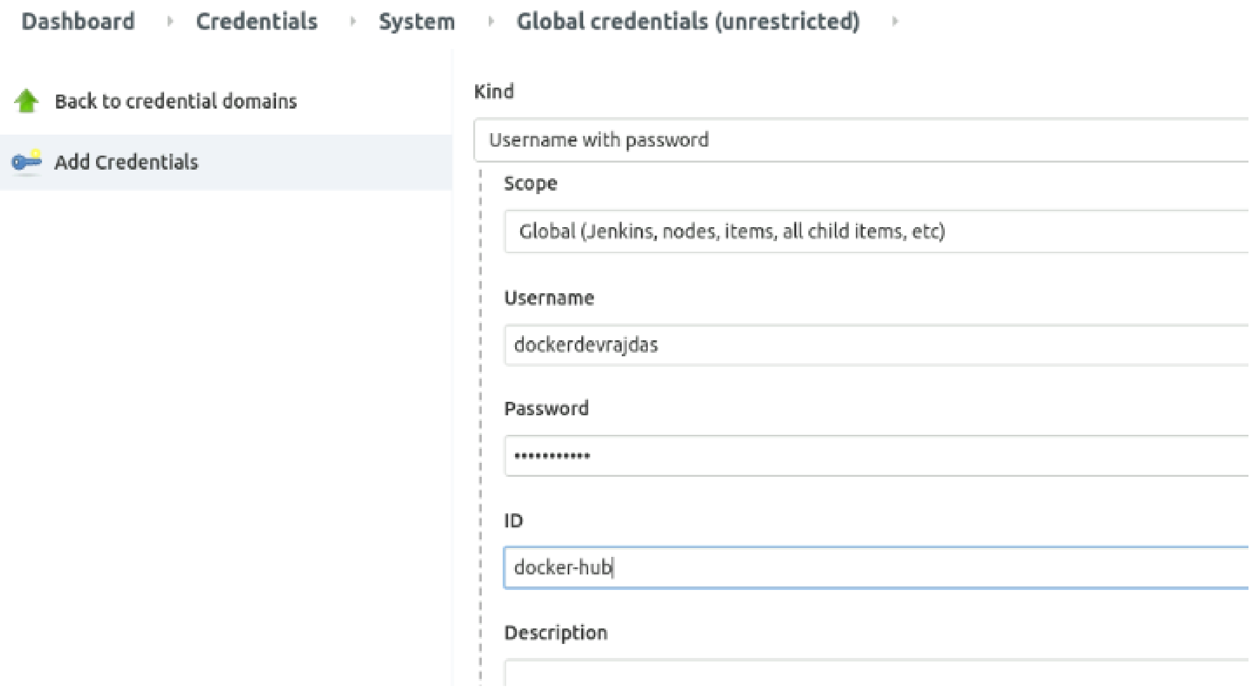
Tags and Scans

VULNERABILITY SCANNING - DISABLED

Recent builds

11. Go to home page of jenkins ---> Click Manage Jenkins ---> Click Manage Credentials ----> Click Add Credentials ---> Select Kind as username with password ---> Keep Scope as it is ----> Add your docker hub username and

password in the respective fields ---> Enter “docker-hub” value in Id field and Save.



Dashboard › Credentials › System › Global credentials (unrestricted) ›

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

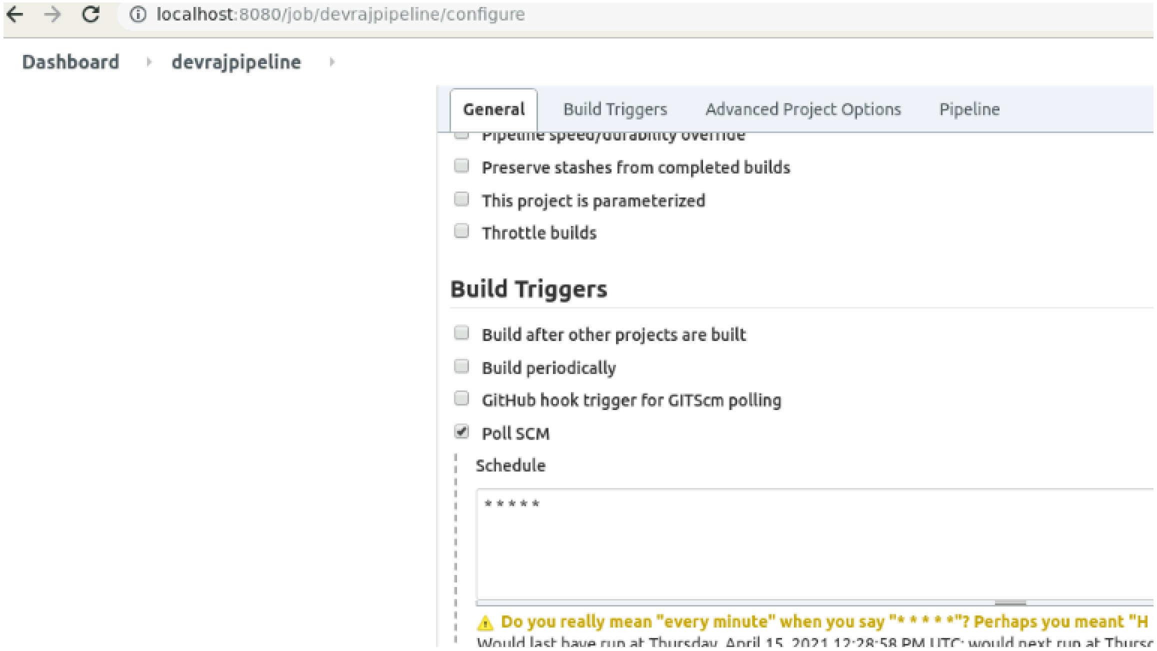
Username: dockerdevrajdas

Password: •••••••••••

ID: docker-hub

Description:

12. Go to jenkins home page ---> click new item → Enter your pipeline name ---> Select type as “Pipeline” and click ok
You will see pipeline configure window, Go to Triggers Section and Set Poll SCM to “* * * * *”



localhost:8080/job/devrajpipeline/configure

Dashboard › devrajpipeline ›

□ Pipeline speed/durability override
□ Preserve stashes from completed builds
□ This project is parameterized
□ Throttle builds

Build Triggers

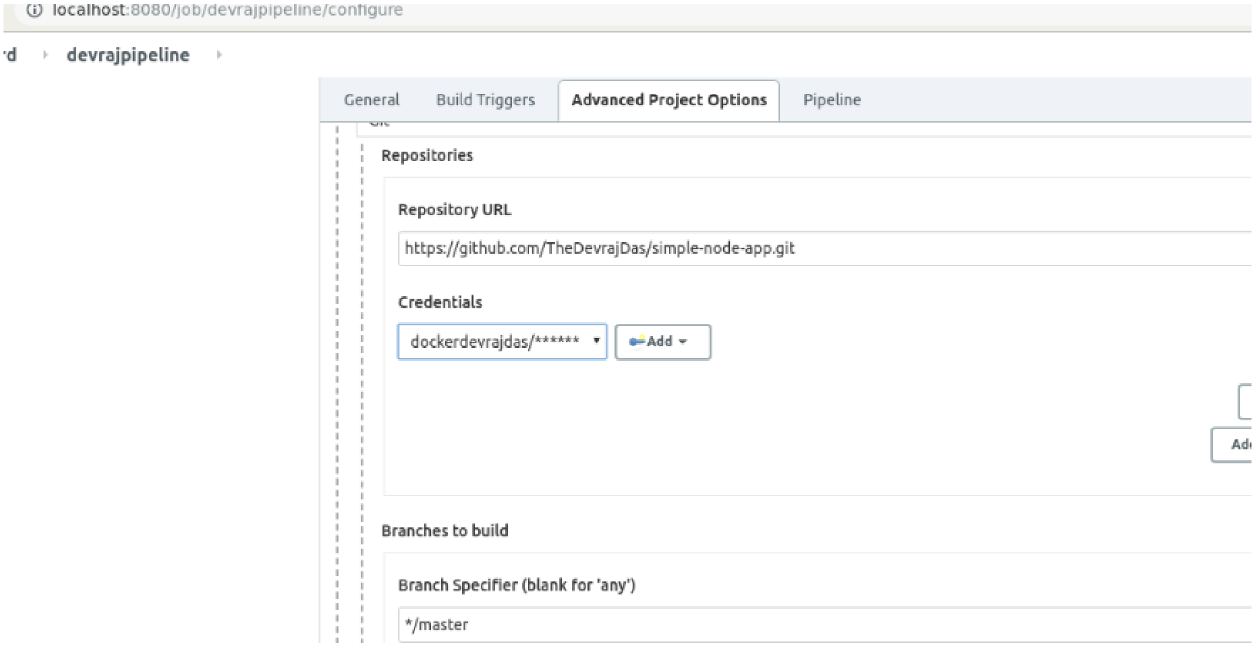
□ Build after other projects are built
□ Build periodically
□ GitHub hook trigger for GITScm polling
☑ Poll SCM

Schedule

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H
Would last have run at Thursday, April 15, 2021 12:28:58 PM UTC; would next run at Thursc

13. Go below to Pipeline Section and select option as “Pipeline from SCM” ---> Select Git ---> Enter our project github repository url ---> Keep credentials as “None” as it is public repository and click save



localhost:8080/job/devrajpipeline/configure

'd › devrajpipeline ›

General | Build Triggers | Advanced Project Options | Pipeline

Repositories

Repository URL

https://github.com/TheDevrajDas/simple-node-app.git

Credentials

dockerdevrajdas/****** ▾ Add ▾

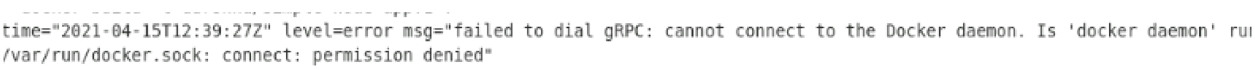
Ad

Branches to build

Branch Specifier (blank for 'any')

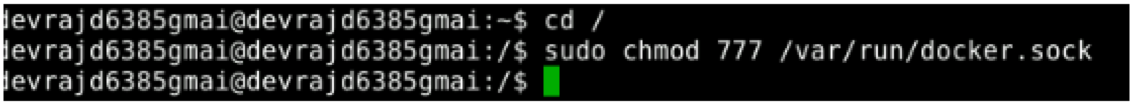
*/master

14. Click on Build Now.
15. First build will fail as docker daemon does not have elevated privileges



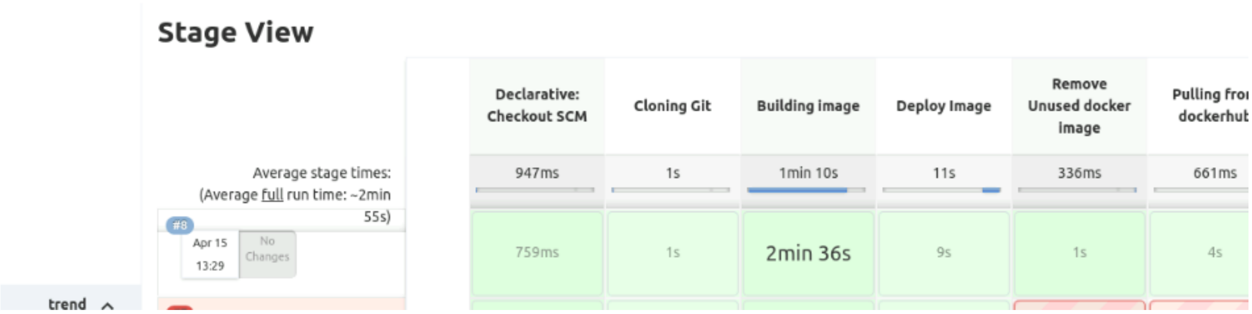
```
time="2021-04-15T12:39:27Z" level=error msg="failed to dial gRPC: cannot connect to the Docker daemon. Is 'docker daemon' rui
/var/run/docker.sock: connect: permission denied"
```

16. Go to terminal.
    Cd /
    Sudo chmod 777 /var/run/docker.sock



```
levrajd6385gmai@devrajd6385gmai:~$ cd /
levrajd6385gmai@devrajd6385gmai:/$ sudo chmod 777 /var/run/docker.sock
levrajd6385gmai@devrajd6385gmai:/$ 
```

17. build now



Stage View

	Declarative: Checkout SCM	Cloning Git	Building image	Deploy Image	Remove Unused docker image	Pulling fro dockerhub
Average stage times: (Average full run time: ~2min 55s)	947ms	1s	1min 10s	11s	336ms	661ms
#8 Apr 15 13:29 No Changes	759ms	1s	2min 36s	9s	1s	4s