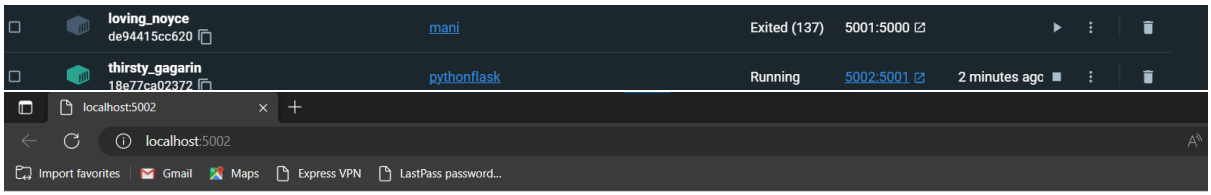
Python Flask Application

- We need login to docker by using docker login
- Next need to build the image by using docker build -t “image_name”

```
anikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonf
ask (master)
 ls
Dockerfile  'Instllation guidlines'   __pycache__/   app.py   requirements.txt   venv/

anikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonflask (master)
 docker build -t pythonflask .
1 [internal] load build definition from Dockerfile
1 sha256:f30a3334740e83ac9b8bf54fea945bd55c692b289aadbcf1100f715a8810138c
1 transferring dockerfile: 274B 0.0s done
1 DONE 0.0s

2 [internal] load .dockerignore
2 sha256:090b5891cd526aa05d84379a7edcd865a6e06f7b4c9217cc1db45cb44f5309f7
2 transferring context: 2B done
2 DONE 0.0s

3 resolve image config for docker.io/docker/dockerfile:1
3 sha256:ac072d521901222eeef550f52282877f196e16b0247844be9ceb1ccc1eac391d
3 ...

4 [auth] docker/dockerfile:pull token for registry-1.docker.io
4 sha256:920a632b02678c2aaf8df490d45c484ef4e1c2eb6319000878c8b1cc383fb0d1
4 DONE 0.0s

3 resolve image config for docker.io/docker/dockerfile:1
3 sha256:ac072d521901222eeef550f52282877f196e16b0247844be9ceb1ccc1eac391d
3 DONE 4.6s

5 docker-image://docker.io/docker/dockerfile:1@sha256:d2d74ff22a0e47b21f4bbde337e2ac4cd0a02a2226ef79264878db3dc7e87df8
5 sha256:500ce23963cd44fe15ba5c94c58b59fff1ce10656da471e9840cfa49b024a111
5 resolve docker.io/docker/dockerfile:1@sha256:d2d74ff22a0e47b21f4bbde337e2ac4cd0a02a2226ef79264878db3dc7e87df8 0.0s done
5 sha256:d2d74ff22a0e47b21f4bbde337e2ac4cd0a02a2226ef79264878db3dc7e87df8 7.65kB / 7.65kB done
5 sha256:6107562e90a8e27b67c589b63fce32010ddf37f48b701600cfd405e296651a5d 482B / 482B done
5 sha256:89543fde88d7f1fb9104eb8832cfe0e33bbacf255e98e916338940bbd657754f 2.91kB / 2.91kB done
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 0B / 11.55MB 0.1s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 1.05MB / 11.55MB 0.8s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 2.10MB / 11.55MB 1.1s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 3.15MB / 11.55MB 1.3s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 4.19MB / 11.55MB 1.5s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 5.24MB / 11.55MB 1.6s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 6.29MB / 11.55MB 1.8s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 7.34MB / 11.55MB 1.9s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 8.39MB / 11.55MB 2.1s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 9.44MB / 11.55MB 2.2s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 10.49MB / 11.55MB 2.4s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 11.55MB / 11.55MB 2.5s
5 sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 11.55MB / 11.55MB 2.5s done
5 extracting sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 0.1s
5 extracting sha256:dd092abd7f3683f4e8e7a66e770a1cc279b2132ac7f66d3c11b7d4a0cb529b7d 0.3s done
5 DONE 2.9s

6 [internal] load build definition from Dockerfile
6 sha256:6c081b3f30486fb16e9c5a5d7ec06641bb5dc94d7de63625d64c1a2d40d79c3a
6 DONE 0.0s
```

- Next we need to create a repository in docker hub and we need tag that in our git bash and we need to push our image into that repository.
- After that we need to run our image using docker run -d -p port number image name
- Need to check in our docker desktop if it running.
- Next we need to open that port number in our localhost for checking required output.

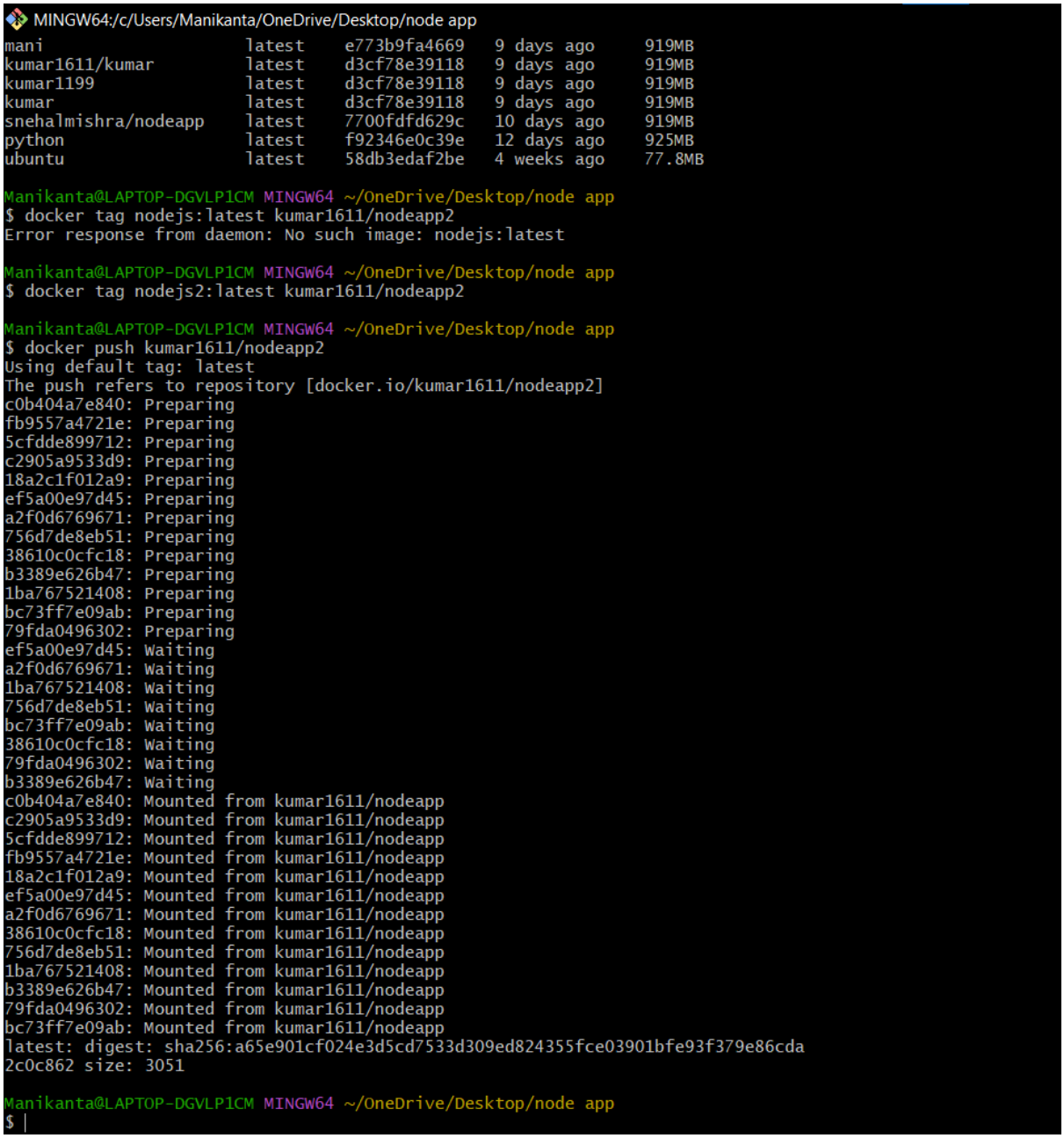
```
Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonflask (master)
$ docker tag pythonflask:latest kumar1611/pythonflask

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonflask (master)
$ docker push kumar1611/pythonflask
Using default tag: latest
The push refers to repository [docker.io/kumar1611/pythonflask]
94b20717f620: Preparing
6987edb939fd: Preparing
c5d991a92fc5: Preparing
5c275dc46323: Preparing
af7d57f541a7: Preparing
0fc22002fc74: Preparing
0a68b1d9e7e0: Preparing
8d60832b730a: Preparing
63b3cf45ece8: Preparing
0a68b1d9e7e0: Waiting
8d60832b730a: Waiting
63b3cf45ece8: Waiting
0fc22002fc74: Waiting
af7d57f541a7: Mounted from library/python
c5d991a92fc5: Pushed
5c275dc46323: Pushed
0fc22002fc74: Mounted from library/python
6987edb939fd: Pushed
94b20717f620: Pushed
0a68b1d9e7e0: Mounted from library/python
8d60832b730a: Mounted from library/python
63b3cf45ece8: Mounted from library/python
latest: digest: sha256:8d1c208c393843d552c85f23fd75fa617242013538bec697c68b62d17c2f9c5e size: 2206

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonflask (master)
$ docker run -d -p 5002:5001 pythonflask
18e77ca023723fc8e8bb355078262a2ad26a68d06990ea76aef71bc8ecffa1a4

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/clone 2/LearningGIt/pythonflask (master)
$
```

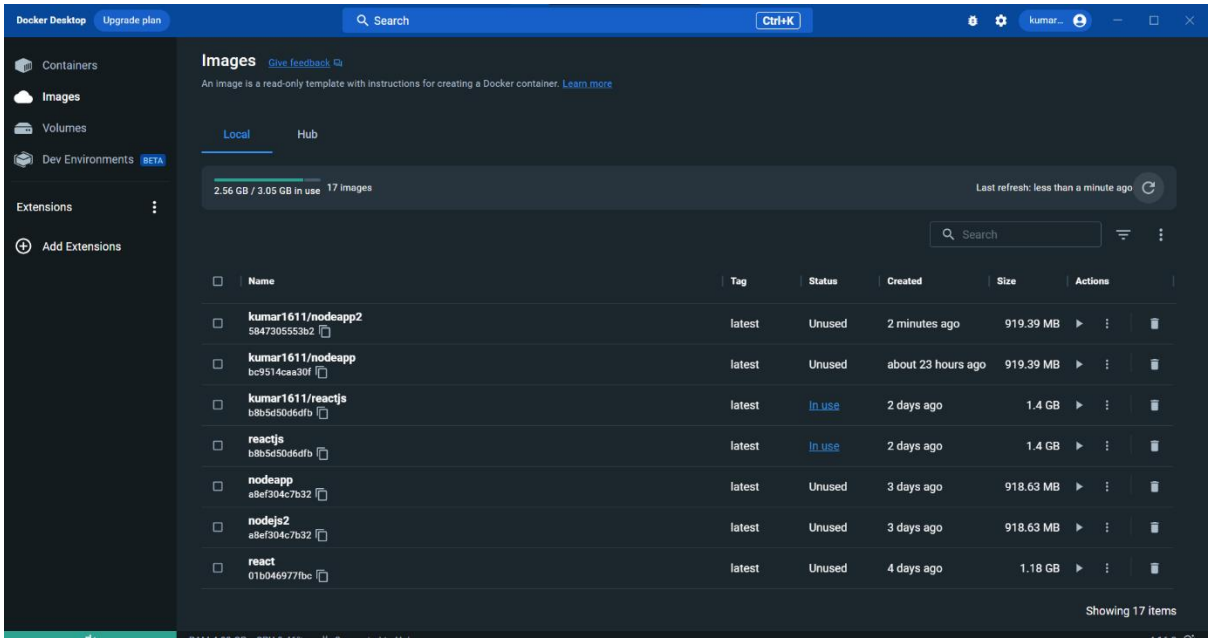


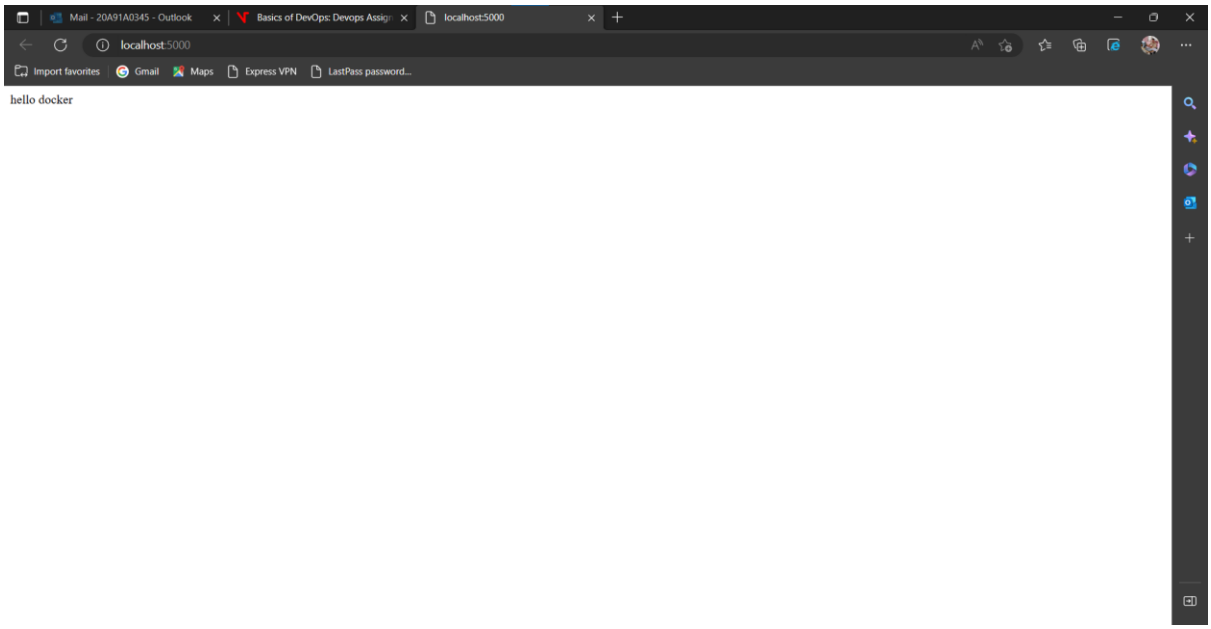
welcome to the flask tutorials

NODEJS APPLICATION

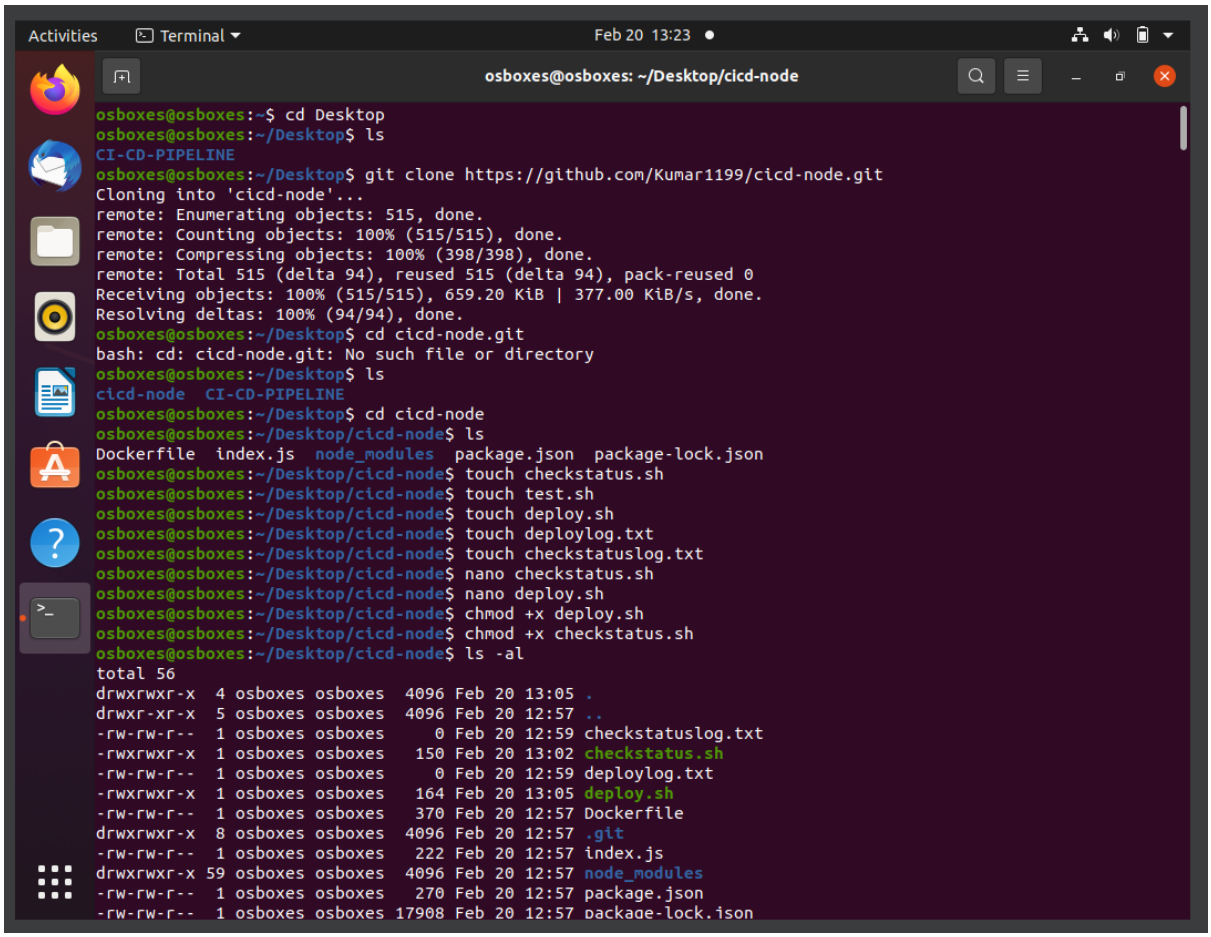
```
MINGW64:/c/Users/Manikanta/OneDrive/Desktop/node app

#9 [4/5] RUN npm install
#9 sha256:e2916dcb21dee466d8737e0bc9e8e4853fadcc0797e5e2da082b8fb453971a86
#9 CACHED

#6 [2/5] WORKDIR /usr/src/app
#6 sha256:762be2d1f5cc6d6ea66e0aff00c82de2644aaeea22ac24677dfc0d15fbe9dfea
#6 CACHED

#8 [3/5] COPY package*.json ./
#8 sha256:52f25d2208c0e659b8c8d77d2ba820dde9c9475ed6599812115feae425e194aa
#8 CACHED

#10 [5/5] COPY . .
#10 sha256:a84abb6268aad3c0604d2bb1c708f3f0e0a0b2485e42b3e12dd50836a18ba00b
#10 CACHED

#11 exporting to image
#11 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#11 exporting layers done
#11 writing image sha256:a8ef304c7b325cc34df9a9bc767a465c7d33ed1effc266e2de897a2
f30f40aad done
#11 naming to docker.io/library/nodejs2
#11 naming to docker.io/library/nodejs2 done
#11 DONE 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and l
earn how to fix them

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
kumar1611/nodeapp       latest    bc9514caa30f   23 hours ago    919MB
kumar1611/reactjs       latest    b8b5d50d6dfb   47 hours ago    1.41GB
reactjs                 latest    b8b5d50d6dfb   47 hours ago    1.41GB
nodeapp                 latest    a8ef304c7b32   2 days ago      919MB
nodejs2                 latest    a8ef304c7b32   2 days ago      919MB
kumar1611/nodeapp       <none>    a8ef304c7b32   2 days ago      919MB
react                   latest    01b046977fbc   3 days ago      1.19GB
kumar1611/pythonflask   latest    c03d98cf6d50   9 days ago      156MB
kk                      latest    c03d98cf6d50   9 days ago      156MB
pythonflask             latest    c03d98cf6d50   9 days ago      156MB
mani                    latest    e773b9fa4669   9 days ago      919MB
kumar1611/kumar         latest    d3cf78e39118   9 days ago      919MB
kumar1199               latest    d3cf78e39118   9 days ago      919MB
kumar                   latest    d3cf78e39118   9 days ago      919MB
snehalmishra/nodeapp    latest    7700fdfd629c   10 days ago     919MB
python                  latest    f92346e0c39e   12 days ago     925MB
ubuntu                  latest    58db3edaf2be   4 weeks ago     77.8MB

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker tag nodejs:latest kumar1611/nodeapp2
Error response from daemon: No such image: nodejs:latest

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker tag nodejs2:latest kumar1611/nodeapp2

Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
```

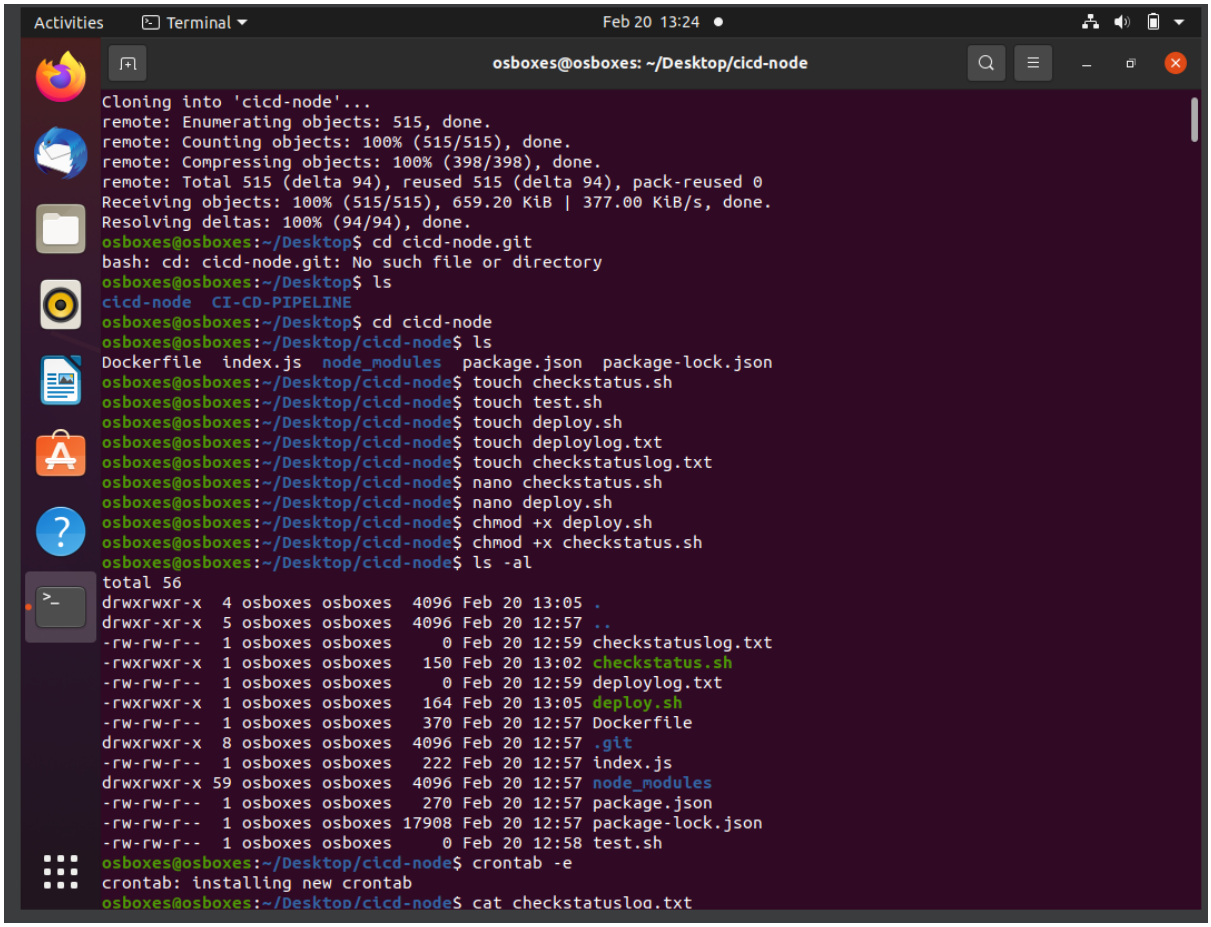

MINGW64:/c/Users/Manikanta/OneDrive/Desktop/node app
mani latest e773b9fa4669 9 days ago 919MB
kumar1611/kumar latest d3cf78e39118 9 days ago 919MB
kumar1199 latest d3cf78e39118 9 days ago 919MB
kumar latest d3cf78e39118 9 days ago 919MB
snehalmishra/nodeapp latest 7700fdfd629c 10 days ago 919MB
python latest f92346e0c39e 12 days ago 925MB
ubuntu latest 58db3edaf2be 4 weeks ago 77.8MB
Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker tag nodejs:latest kumar1611/nodeapp2
Error response from daemon: No such image: nodejs:latest
Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker tag nodejs2:latest kumar1611/nodeapp2
Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$ docker push kumar1611/nodeapp2
Using default tag: latest
The push refers to repository [docker.io/kumar1611/nodeapp2]
c0b404a7e840: Preparing
fb9557a4721e: Preparing
5cfdde899712: Preparing
c2905a9533d9: Preparing
18a2c1f012a9: Preparing
ef5a00e97d45: Preparing
a2f0d6769671: Preparing
756d7de8eb51: Preparing
38610c0cfc18: Preparing
b3389e626b47: Preparing
1ba767521408: Preparing
bc73ff7e09ab: Preparing
79fda0496302: Preparing
ef5a00e97d45: Waiting
a2f0d6769671: Waiting
1ba767521408: Waiting
756d7de8eb51: Waiting
bc73ff7e09ab: Waiting
38610c0cfc18: Waiting
79fda0496302: Waiting
b3389e626b47: Waiting
c0b404a7e840: Mounted from kumar1611/nodeapp
c2905a9533d9: Mounted from kumar1611/nodeapp
5cfdde899712: Mounted from kumar1611/nodeapp
fb9557a4721e: Mounted from kumar1611/nodeapp
18a2c1f012a9: Mounted from kumar1611/nodeapp
ef5a00e97d45: Mounted from kumar1611/nodeapp
a2f0d6769671: Mounted from kumar1611/nodeapp
38610c0cfc18: Mounted from kumar1611/nodeapp
756d7de8eb51: Mounted from kumar1611/nodeapp
1ba767521408: Mounted from kumar1611/nodeapp
b3389e626b47: Mounted from kumar1611/nodeapp
79fda0496302: Mounted from kumar1611/nodeapp
bc73ff7e09ab: Mounted from kumar1611/nodeapp
latest: digest: sha256:a65e901cf024e3d5cd7533d309ed824355fce03901bfe93f379e86cda
2c0c862 size: 3051
Manikanta@LAPTOP-DGVLP1CM MINGW64 ~/OneDrive/Desktop/node app
$

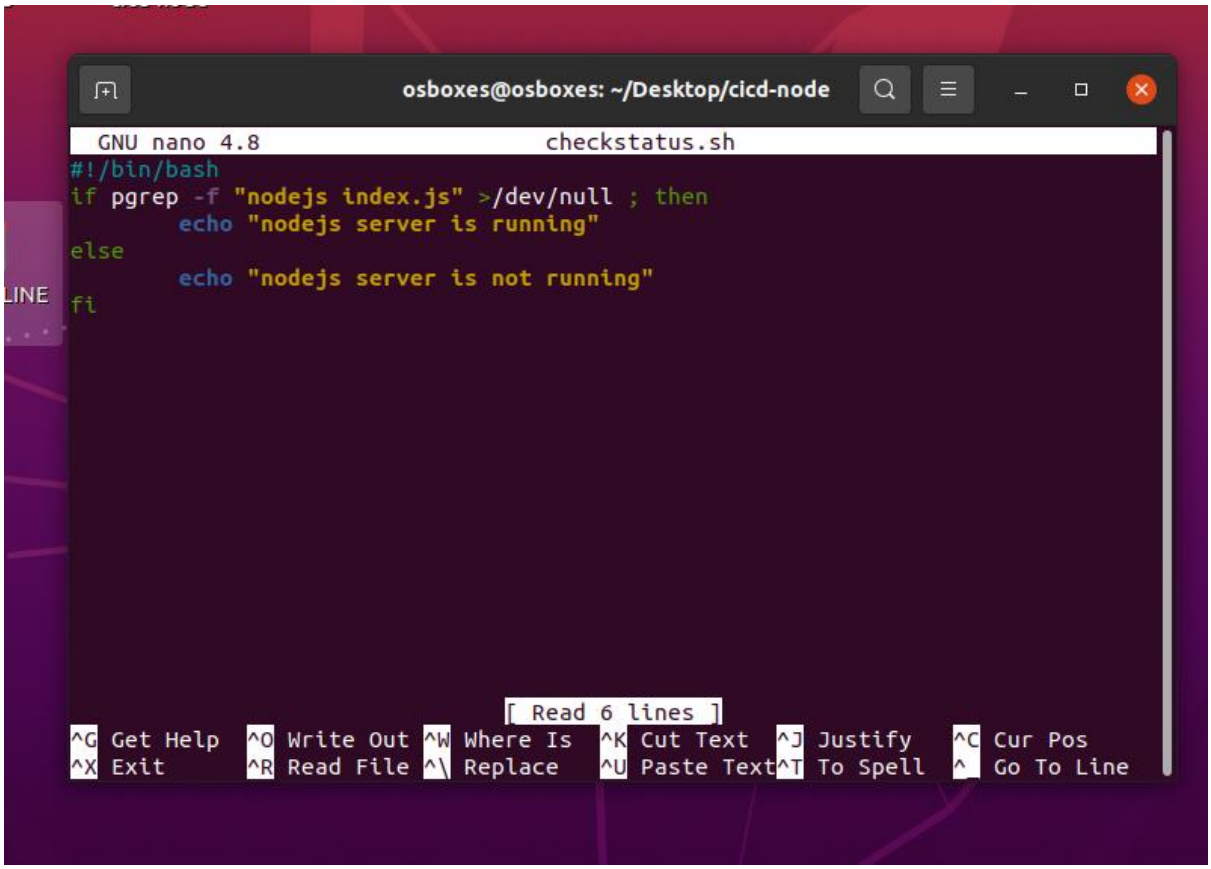

Docker Desktop
Upgrade plan
Search
Ctrl+K
kumar...
Containers
Images
Volumes
Dev Environments
BETA
Extensions
Add Extensions
Images
Give feedback
An image is a read-only template with instructions for creating a Docker container. Learn more
Local
Hub
2.56 GB / 3.05 GB in use
17 images
Last refresh: less than a minute ago
Search
Name
Tag
Status
Created
Size
Actions
kumar1611/nodeapp2
5847305553b2
latest
Unused
2 minutes ago
919.39 MB
kumar1611/nodeapp
bc9514caa30f
latest
Unused
about 23 hours ago
919.39 MB
kumar1611/reactjs
b8b5d50d6dfb
latest
In use
2 days ago
1.4 GB
reactjs
b8b5d50d6dfb
latest
In use
2 days ago
1.4 GB
nodeapp
a8ef304c7b32
latest
Unused
3 days ago
918.63 MB
nodejs2
a8ef304c7b32
latest
Unused
3 days ago
918.63 MB
react
01b046977fbc
latest
Unused
4 days ago
1.18 GB
Showing 17 items

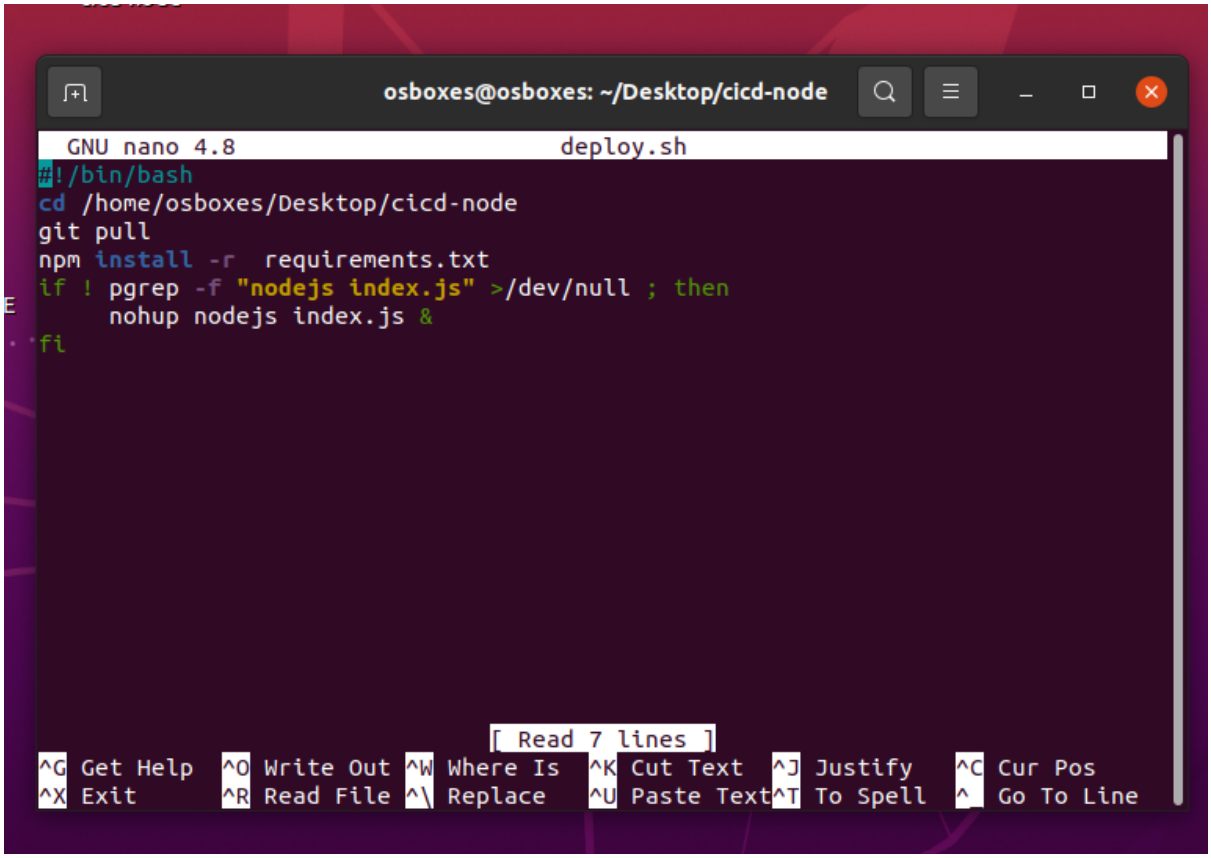

Mail - 20A91A0345 - Outlook
Basics of DevOps: Devops Assign
localhost:5000
localhost:5000
Import favorites
Gmail
Maps
Express VPN
LastPass password...
hello docker

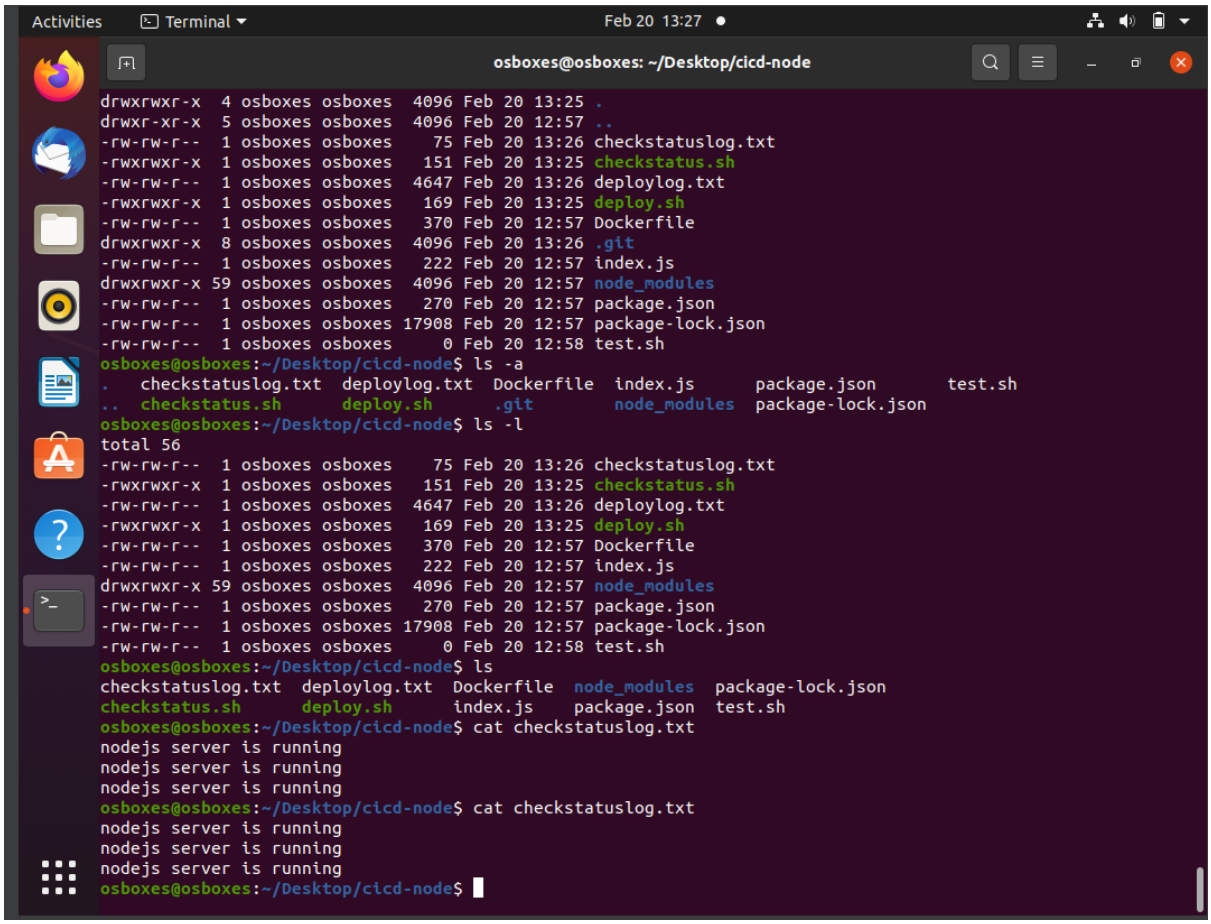
Nodejs CI-CD pipeline

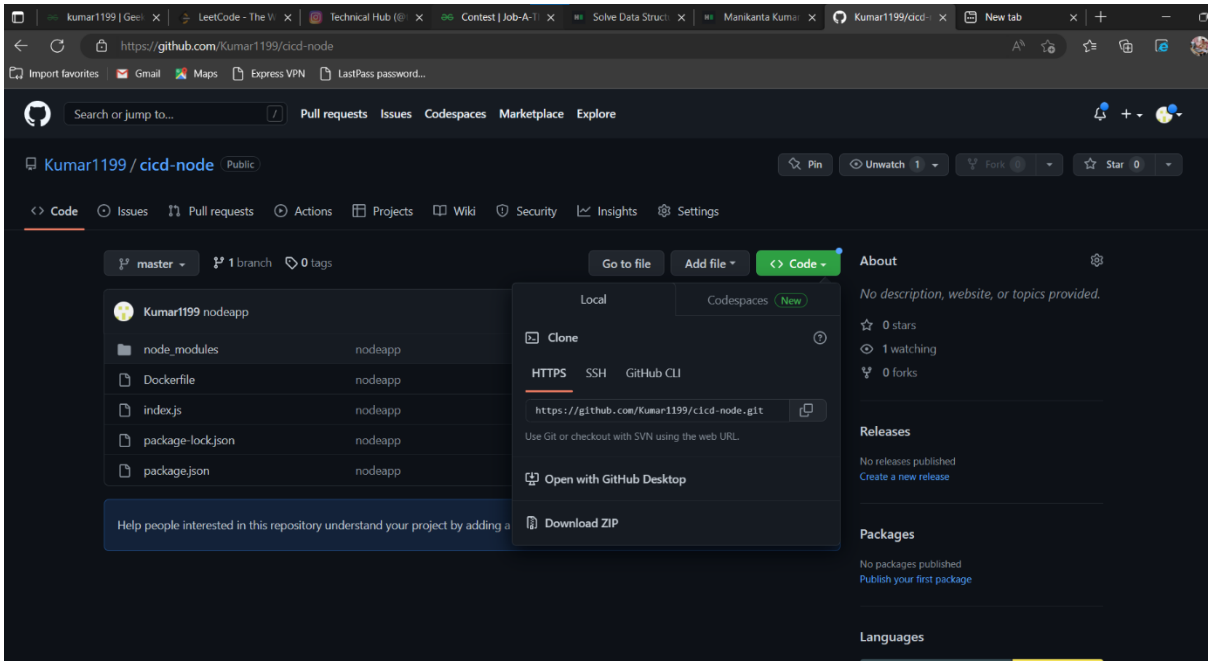



Activities
Terminal
Feb 20 13:24
osboxes@osboxes: ~/Desktop/cicd-node
Cloning into 'cicd-node'...
remote: Enumerating objects: 515, done.
remote: Counting objects: 100% (515/515), done.
remote: Compressing objects: 100% (398/398), done.
remote: Total 515 (delta 94), reused 515 (delta 94), pack-reused 0
Receiving objects: 100% (515/515), 659.20 KiB | 377.00 KiB/s, done.
Resolving deltas: 100% (94/94), done.
osboxes@osboxes:~/Desktop$ cd cicd-node.git
bash: cd: cicd-node.git: No such file or directory
osboxes@osboxes:~/Desktop$ ls
cicd-node CI-CD-PIPELINE
osboxes@osboxes:~/Desktop$ cd cicd-node
osboxes@osboxes:~/Desktop/cicd-node$ ls
Dockerfile index.js node_modules package.json package-lock.json
osboxes@osboxes:~/Desktop/cicd-node$ touch checkstatus.sh
osboxes@osboxes:~/Desktop/cicd-node$ touch test.sh
osboxes@osboxes:~/Desktop/cicd-node$ touch deploy.sh
osboxes@osboxes:~/Desktop/cicd-node$ touch deploylog.txt
osboxes@osboxes:~/Desktop/cicd-node$ touch checkstatuslog.txt
osboxes@osboxes:~/Desktop/cicd-node$ nano checkstatus.sh
osboxes@osboxes:~/Desktop/cicd-node$ nano deploy.sh
osboxes@osboxes:~/Desktop/cicd-node$ chmod +x deploy.sh
osboxes@osboxes:~/Desktop/cicd-node$ chmod +x checkstatus.sh
osboxes@osboxes:~/Desktop/cicd-node$ ls -al
total 56
drwxrwxr-x 4 osboxes osboxes 4096 Feb 20 13:05 .
drwxr-xr-x 5 osboxes osboxes 4096 Feb 20 12:57 ..
-rw-rw-r-- 1 osboxes osboxes 0 Feb 20 12:59 checkstatuslog.txt
-rwxrwxr-x 1 osboxes osboxes 150 Feb 20 13:02 checkstatus.sh
-rw-rw-r-- 1 osboxes osboxes 0 Feb 20 12:59 deploylog.txt
-rwxrwxr-x 1 osboxes osboxes 164 Feb 20 13:05 deploy.sh
-rw-rw-r-- 1 osboxes osboxes 370 Feb 20 12:57 Dockerfile
drwxrwxr-x 8 osboxes osboxes 4096 Feb 20 12:57 .git
-rw-rw-r-- 1 osboxes osboxes 222 Feb 20 12:57 index.js
drwxrwxr-x 59 osboxes osboxes 4096 Feb 20 12:57 node_modules
-rw-rw-r-- 1 osboxes osboxes 270 Feb 20 12:57 package.json
-rw-rw-r-- 1 osboxes osboxes 17908 Feb 20 12:57 package-lock.json
-rw-rw-r-- 1 osboxes osboxes 0 Feb 20 12:58 test.sh
osboxes@osboxes:~/Desktop/cicd-node$ crontab -e
crontab: installing new crontab
osboxes@osboxes:~/Desktop/cicd-node$ cat checkstatuslog.txt


osboxes@osboxes: ~/Desktop/cicd-node
GNU nano 4.8 checkstatus.sh
#!/bin/bash
if pgrep -f "nodejs index.js" >/dev/null ; then
echo "nodejs server is running"
else
echo "nodejs server is not running"
fi
[Read 6 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line


osboxes@osboxes: ~/Desktop/cicd-node
GNU nano 4.8 deploy.sh
#!/bin/bash
cd /home/osboxes/Desktop/cicd-node
git pull
npm install -r requirements.txt
if ! pgrep -f "nodejs index.js" >/dev/null ; then
nohup nodejs index.js &
fi
[Read 7 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line


Activities
Terminal
Feb 20 13:27
osboxes@osboxes: ~/Desktop/cicd-node
drwxrwxr-x 4 osboxes osboxes 4096 Feb 20 13:25 .
drwxr-xr-x 5 osboxes osboxes 4096 Feb 20 12:57 ..
-rw-rw-r-- 1 osboxes osboxes 75 Feb 20 13:26 checkstatuslog.txt
-rwxrwxr-x 1 osboxes osboxes 151 Feb 20 13:25 checkstatus.sh
-rw-rw-r-- 1 osboxes osboxes 4647 Feb 20 13:26 deploylog.txt
-rwxrwxr-x 1 osboxes osboxes 169 Feb 20 13:25 deploy.sh
-rw-r--r-- 1 osboxes osboxes 370 Feb 20 12:57 Dockerfile
drwxrwxr-x 8 osboxes osboxes 4096 Feb 20 13:26 .git
-rw-r--r-- 1 osboxes osboxes 222 Feb 20 12:57 index.js
drwxrwxr-x 59 osboxes osboxes 4096 Feb 20 12:57 node_modules
-rw-r--r-- 1 osboxes osboxes 270 Feb 20 12:57 package.json
-rw-rw-r-- 1 osboxes osboxes 17908 Feb 20 12:57 package-lock.json
-rw-r--r-- 1 osboxes osboxes 0 Feb 20 12:58 test.sh
osboxes@osboxes:~/Desktop/cicd-node$ ls -a
. checkstatuslog.txt deploylog.txt Dockerfile index.js package.json test.sh
.. checkstatus.sh deploy.sh .git node_modules package-lock.json
osboxes@osboxes:~/Desktop/cicd-node$ ls -l
total 56
-rw-rw-r-- 1 osboxes osboxes 75 Feb 20 13:26 checkstatuslog.txt
-rwxrwxr-x 1 osboxes osboxes 151 Feb 20 13:25 checkstatus.sh
-rw-rw-r-- 1 osboxes osboxes 4647 Feb 20 13:26 deploylog.txt
-rwxrwxr-x 1 osboxes osboxes 169 Feb 20 13:25 deploy.sh
-rw-r--r-- 1 osboxes osboxes 370 Feb 20 12:57 Dockerfile
-rw-r--r-- 1 osboxes osboxes 222 Feb 20 12:57 index.js
drwxrwxr-x 59 osboxes osboxes 4096 Feb 20 12:57 node_modules
-rw-r--r-- 1 osboxes osboxes 270 Feb 20 12:57 package.json
-rw-rw-r-- 1 osboxes osboxes 17908 Feb 20 12:57 package-lock.json
-rw-r--r-- 1 osboxes osboxes 0 Feb 20 12:58 test.sh
osboxes@osboxes:~/Desktop/cicd-node$ ls
checkstatuslog.txt deploylog.txt Dockerfile node_modules package-lock.json
checkstatus.sh deploy.sh index.js package.json test.sh
osboxes@osboxes:~/Desktop/cicd-node$ cat checkstatuslog.txt
nodejs server is running
nodejs server is running
nodejs server is running
osboxes@osboxes:~/Desktop/cicd-node$ cat checkstatuslog.txt
nodejs server is running
nodejs server is running
nodejs server is running
osboxes@osboxes:~/Desktop/cicd-node$



Steps for CI-CD Pipeline for python applications using Jenkins

1. First we need setup the Jenkins like download and installation of jdk file and after that installation of Jenkins in the path where jdk is present.
2. Next we need to setup the Jenkins by running that in our localhost.
3. Next we need to select our required plugins such as for running python applications we need python plugin and git hub plugin.
4. Next to that we need setup the global tool configuration this is for to configure Jenkins with our local git for that we need give our local git path to it.
5. Next we need select new items->free style project->need to provide description->next source code management means the code which we wanted to run cicd pipeline we need push it in the github and we need to give the url of that repository if in case if it is a private repo we need to provide our credentials->check whether which branch is given->next build triggers we

need to select poll scm in that we need to give * * * * * to check for every minute->next we need to add build steps in that we need to select docker build and publish->in that we need give docker hub repository name and we need provide registry credentials->after that apply and save.

6. Finally we need build the application.