

[Status](#)[Changes](#)[Console Output](#)[View as plain text](#)[Edit Build Information](#)[Parameters](#)[Git Build Data](#)[Thread Dump](#)[Pause/resume](#)[Replay](#)[Pipeline Steps](#)[Workspaces](#)

Console Output

Started by user [admin_jenkins](#)

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Ansible-slave](#) in /opt/build/workspace/zenda

[Pipeline] {

[Pipeline] stage

[Pipeline] { (fetch latest code from git)

```
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Avoid second fetch
Checking out Revision 7e537bdf70e73c82ddb3c964afc2576924b14465
(refs/remotes/origin/main)
Commit message: "update in jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build and push docker image)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
Cloning repository https://github.com/pranavmurali1994/zenda.git
  > git init /opt/build/workspace/zenda # timeout=10
Fetching upstream changes from https://github.com/pranavmurali1994/zenda.git
  > git --version # timeout=10
  > git --version # 'git version 2.40.1'
  > git fetch --tags --force --progress --
https://github.com/pranavmurali1994/zenda.git +refs/heads/*:refs/remotes/origin/*
# timeout=10
  > git config remote.origin.url https://github.com/pranavmurali1994/zenda.git #
timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* #
timeout=10
  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
  > git config core.sparsecheckout # timeout=10
  > git checkout -f 7e537bdf70e73c82ddb3c964afc2576924b14465 # timeout=10
  > git branch -a -v --no-abbrev # timeout=10
  > git checkout -b main 7e537bdf70e73c82ddb3c964afc2576924b14465 # timeout=10
+ docker build -t pranavmurali1994/nodejs:v1.0 -f
/opt/build/workspace/zenda/Dockerfile/Dockerfile
/opt/build/workspace/zenda/Dockerfile/
Sending build context to Docker daemon 4.096kB
```

Step 1/7 : FROM node:14

---> 1d12470fa662

Step 2/7 : WORKDIR /app

---> Using cache

---> 4513055904b2

Step 3/7 : COPY package*.json ./

```
---> Using cache
---> 43d07c8983e4
Step 4/7 : RUN npm install
---> Using cache
---> cf4ed686e0d0
Step 5/7 : COPY . .
---> Using cache
---> d806fe8c0a3b
Step 6/7 : EXPOSE 3000
---> Using cache
---> e84fe1d4f71c
Step 7/7 : CMD [ "npm", "start" ]
---> Using cache
---> 2d7f280fb766
Successfully built 2d7f280fb766
```

Dashboard zenda #1

```
+ docker push pranavmurali1994/nodejs:v1.0
The push refers to repository [docker.io/pranavmurali1994/nodejs]
ab19a7993906: Preparing
acbf302a7464: Preparing
d4202549b5b2: Preparing
1927f460233c: Preparing
0d5f5a015e5d: Preparing
3c777d951de2: Preparing
f8a91dd5fc84: Preparing
cb81227abde5: Preparing
e01a454893a9: Preparing
c45660adde37: Preparing
fe0fb3ab4a0f: Preparing
f1186e5061f2: Preparing
b2dba7477754: Preparing
3c777d951de2: Waiting
f8a91dd5fc84: Waiting
cb81227abde5: Waiting
e01a454893a9: Waiting
c45660adde37: Waiting
fe0fb3ab4a0f: Waiting
f1186e5061f2: Waiting
b2dba7477754: Waiting
d4202549b5b2: Layer already exists
acbf302a7464: Layer already exists
ab19a7993906: Layer already exists
1927f460233c: Layer already exists
```

```

0d5f5a015e5d: Layer already exists
f8a91dd5fc84: Layer already exists
3c777d951de2: Layer already exists
cb81227abde5: Layer already exists
e01a454893a9: Layer already exists
c45660adde37: Layer already exists
fe0fb3ab4a0f: Layer already exists
f1186e5061f2: Layer already exists
b2dba7477754: Layer already exists
v1.0: digest:
sha256:e9431039a84f4986d83ac0e8b6fb5dbcb846e81d2423a6d928577b6b7118b9c0 size:
3043
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploying application using Ansible)
[Pipeline] ansiblePlaybook
[zenda] $ ansible-playbook /opt/build/workspace/zenda/ansible-k8s/tasks/main.yml
-i /opt/build/workspace/zenda/ansible-k8s/host.yml -e image_id=v1.0

PLAY [eks] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 10.20.1.27 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [10.20.1.27]

TASK [copy kubernetes latest definition file] *****
ok: [10.20.1.27]

TASK [deploy application] *****
changed: [10.20.1.27]

PLAY RECAP *****
10.20.1.27          : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }

```

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS