



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

DevOps(DS457)

Assignment

Developing and Deploying a Node.js app from Docker to Kubernetes

Submitted to

Dr.Uma.S

Submitted by

Team13

S.Sampath(18BCS087)

T.Umesh Anand babu(18BCS105)

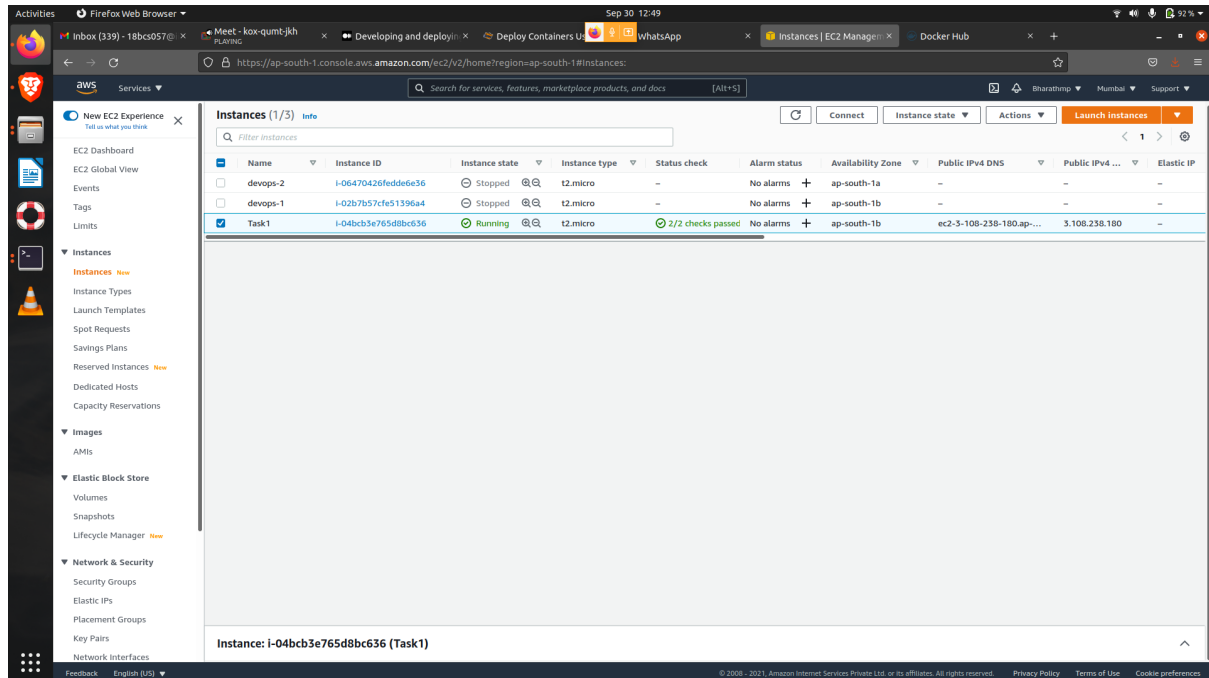
M.P. Bharath(18BCS057)

M.Venkata Kalyan Babu(18BCS049)

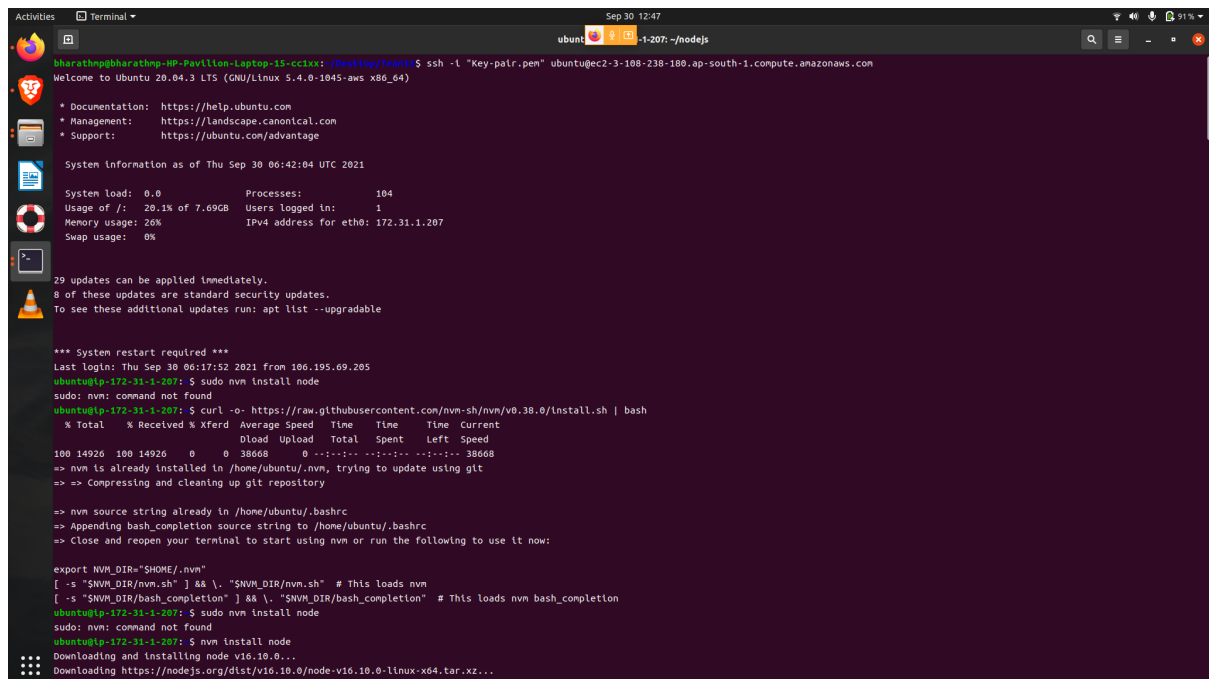
Y.Mokshith(18BCS112)

Assignment:1.2:

Creating an Ubuntu EC2 Instance (task_1)



Connecting to EC2 Instance and installing and checking node



Installing and Checking node, npm, Docker

```
Activities Terminal
Sep 30 12:47
ubuntu@ubuntu:~$ export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
ubuntu@ubuntu:~$ sudo nvm install node
Sudo: nvm: command not found
ubuntu@ubuntu:~$ sudo nvm install node
Downloading and installing node v16.10.0...
Downloading https://nodejs.org/dist/v16.10.0/node-v16.10.0-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v16.10.0 (npm v7.24.0)
Creating default alias: default -> node (-> v16.10.0)
ubuntu@ubuntu:~$ nvm use node
Now using node v16.10.0 (npm v7.24.0)
ubuntu@ubuntu:~$ nvm -v
v16.10.0
ubuntu@ubuntu:~$ npm -v
7.24.0
ubuntu@ubuntu:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libltdl1 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debotstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libltdl1 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 74.4 MB of archives.
After this operation, 360 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [39.5 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.0-rc95-0ubuntu1-20.04.2 [4087 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.2-0ubuntu1-20.04.2 [32.9 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libltdl1 amd64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu1-20.04.1 [36.9 MB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB]
Fetched 74.4 MB in 17s (4506 kB/s)
```

Initializing the node application

```
Activities Terminal
Oct 1 00:15
bharathmp@bharathmp-HP-Pavilion-Laptop-15-cd1xxc: ~/Desktop/Team13
ubuntu@ubuntu:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1-20.04.1
ubuntu@ubuntu:~$ mkdir nodejs
ubuntu@ubuntu:~$ cd nodejs
ubuntu@ubuntu:~$ nodejs$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs) bharath
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/ubuntu/nodejs/package.json:

{
  "name": "bharath",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

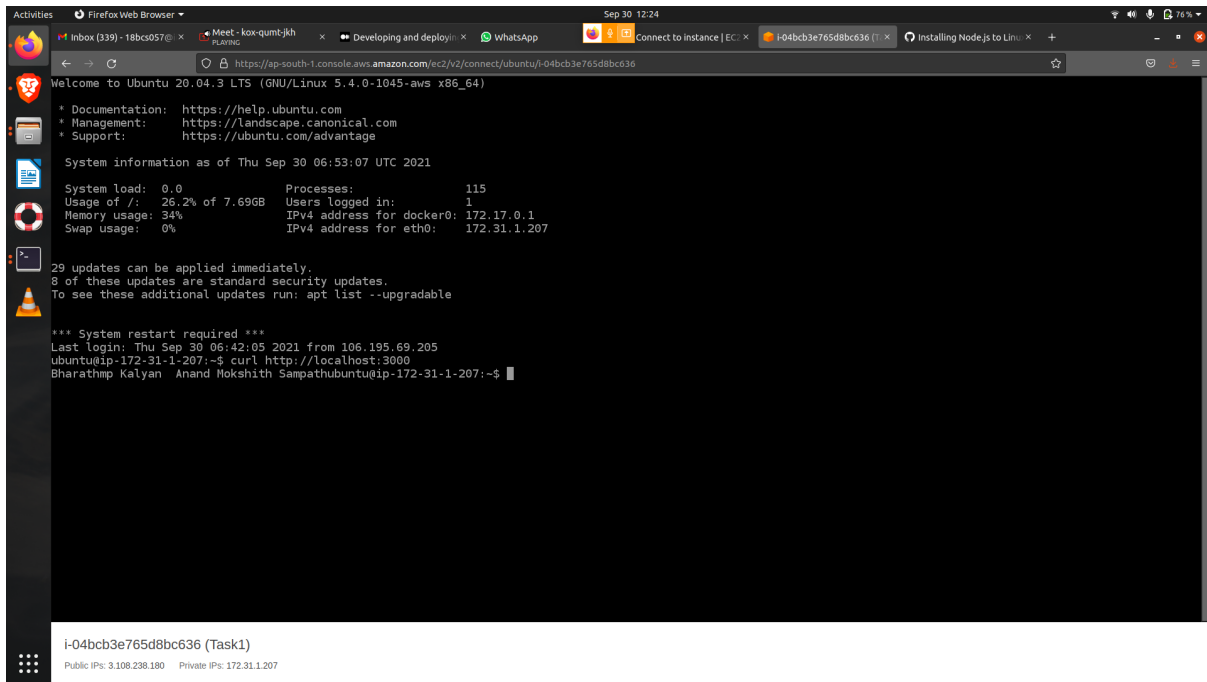
Is this OK? (yes) yes
npm notice
npm notice New patch version of npm available! 7.24.0 -> 7.24.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.24.1
npm notice Run npm install npm@7.24.1 to update!
```

```
Activities Terminal Oct 1 00:21 bharathmp@bharathmp-HP-Pavilion-Laptop-15-cdxxx: ~/Desktop/Team13

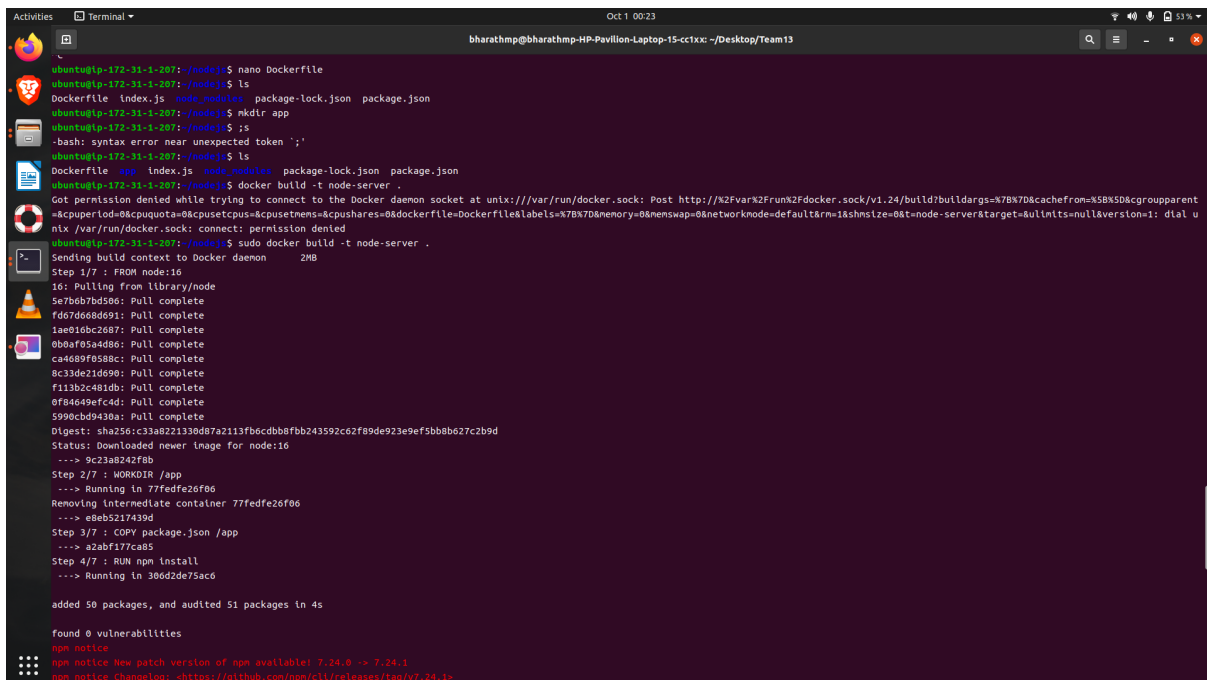
ubuntu@ip-172-31-1-207:~/nodejs$ nano index.js
ubuntu@ip-172-31-1-207:~/nodejs$ node index.js
listening on 3000
^Z
[1]+  Stopped                  node index.js
ubuntu@ip-172-31-1-207:~/nodejs$ node index.js
node:events:368
    throw er; // Unhandled 'error' event
    ^
Error: listen EADDRINUSE: address already in use :::3000
    at Server.setupListenHandle [as _listen2] (node:net:1319:16)
    at listenInCluster (node:net:1367:12)
    at Server.listen (node:net:1454:7)
    at Function.listen (/home/ubuntu/nodejs/node_modules/express/lib/application.js:618:24)
    at Object.<anonymous> (/home/ubuntu/nodejs/index.js:2:27)
    at Module._compile (node:internal/modules/cjs/loader:1101:14)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1153:10)
    at Module.load (node:internal/modules/cjs/loader:981:32)
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:79:12)
Emitted 'error' event on Server instance at:
    at emitErrorNT (node:net:1346:8)
    at processTicksAndRejections (node:internal/process/task_queues:83:21) {
  code: 'EADDRINUSE',
  errno: -98,
  syscall: 'listen',
  address: '::',
  port: 3000
}
ubuntu@ip-172-31-1-207:~/nodejs$ sudo kill -9 `sudo lsof -t -i:3000`
[1]+  Killed                  node index.js
ubuntu@ip-172-31-1-207:~/nodejs$ node index.js
listening on 3000
^C
ubuntu@ip-172-31-1-207:~/nodejs$ nano Dockerfile
ubuntu@ip-172-31-1-207:~/nodejs$ ls
Dockerfile  index.js  node_modules  package-lock.json  package.json
ubuntu@ip-172-31-1-207:~/nodejs$ mkdir app
ubuntu@ip-172-31-1-207:~/nodejs$ cd app
-bash: syntax error near unexpected token `;'
ubuntu@ip-172-31-1-207:~/nodejs$ ls
Dockerfile  app  index.js  node_modules  package-lock.json  package.json
ubuntu@ip-172-31-1-207:~/nodejs$ docker build -t node-server .
```

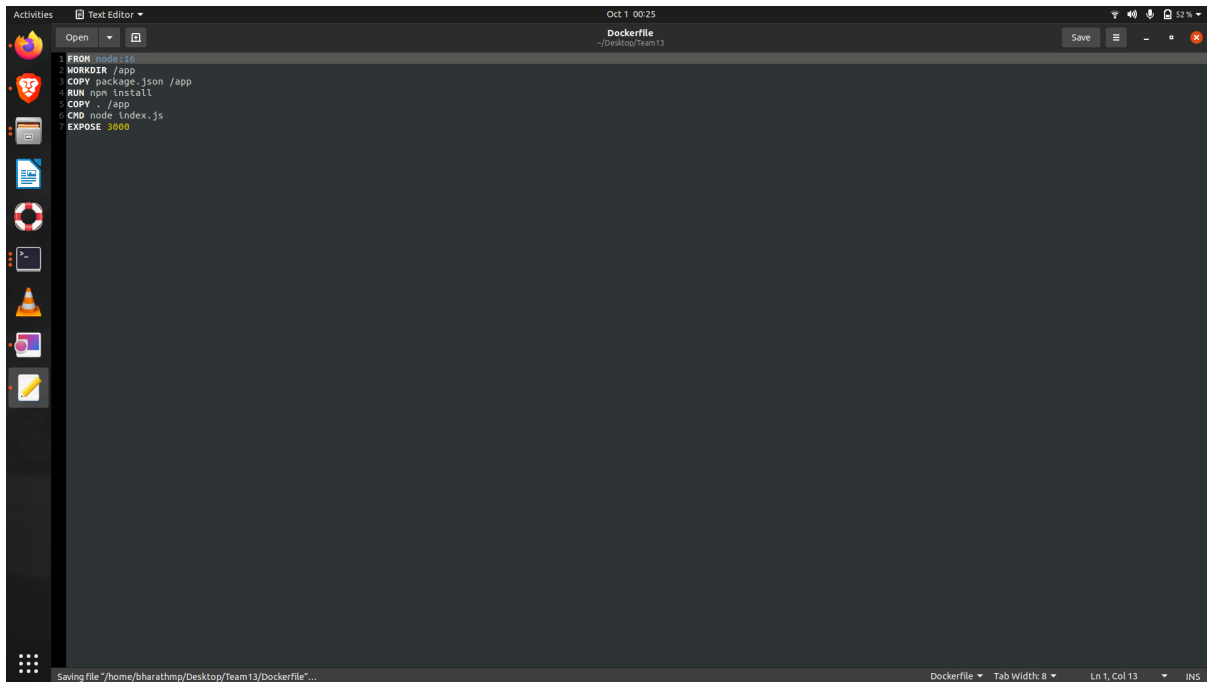
```
Activities Text Editor Oct 1 00:19 *index.js ~/Desktop/Team13 Save

1 const express = require("express");
2 const app = express(); app.listen(3000, function () {
3   console.log("listening on 3000");
4 }); app.get("/", (req, res) => {
5   res.send("bharathmp Kalyan Anand Mokshith Sampath");
6 }); app.get("/delete", (req, res) => {
7   res.send("Delete User");
8 }); app.get("/update", (req, res) => {
9   res.send("Update User");
10 }); app.get("/insert", (req, res) => {
11   res.send("Insert User");
12 });
13
```



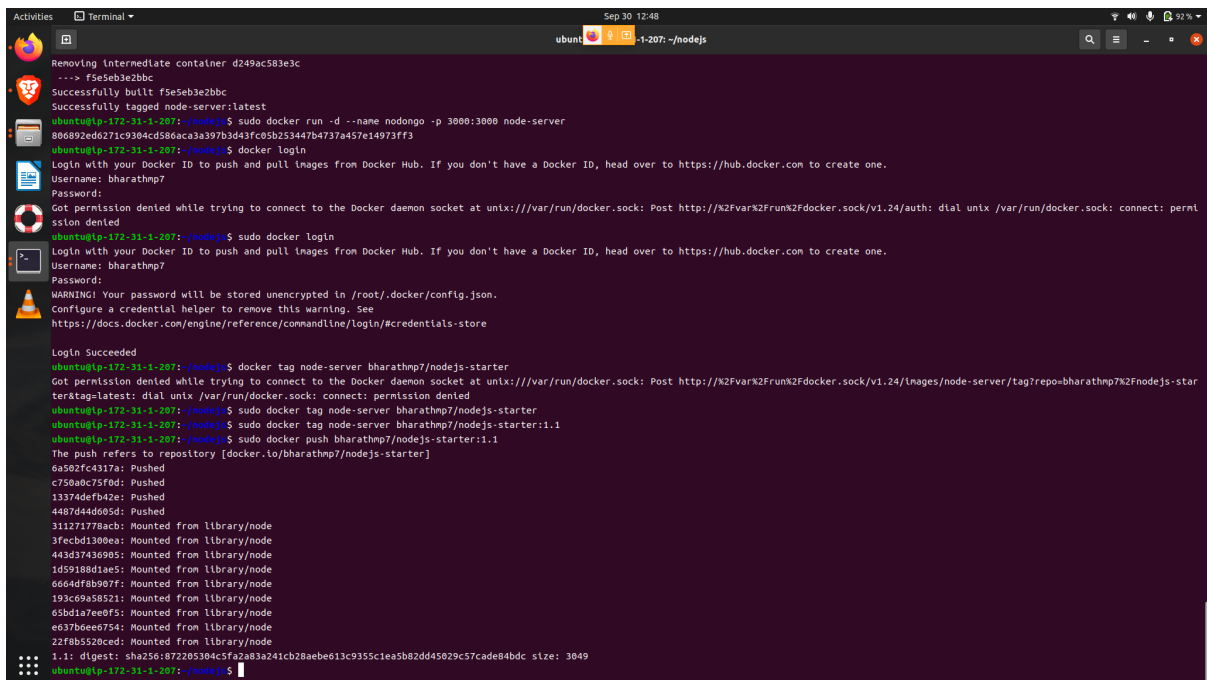
Creating a Docker file





```
1 FROM node:16
2 WORKDIR /app
3 COPY package.json /app
4 RUN npm install
5 COPY . /app
6 CMD node index.js
7 EXPOSE 3000
```

Login into Docker hub and pushing the Image.



```
Removing intermediate container d249ac583e3c
--> f5e5eb3e2bbc
Successfully built f5e5eb3e2bbc
Successfully tagged node-server:latest
ubuntu@ip-172-31-1-207:~/nodejs$ sudo docker run -d --name nodongo -p 3000:3000 node-server
806892ed6271c9304cd586aca3a397b3d43fc05b253447b4737a457e14973ff3
ubuntu@ip-172-31-1-207:~/nodejs$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: bharathmp7
Password:
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.24/auth: dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-1-207:~/nodejs$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: bharathmp7
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-1-207:~/nodejs$ docker tag node-server bharathmp7/nodejs-starter
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/node-server/tag?repo=bharathmp7%2Fnodejs-starter&tag=latest: dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-1-207:~/nodejs$ sudo docker tag node-server bharathmp7/nodejs-starter
ubuntu@ip-172-31-1-207:~/nodejs$ sudo docker tag node-server bharathmp7/nodejs-starter:1.1
ubuntu@ip-172-31-1-207:~/nodejs$ sudo docker push bharathmp7/nodejs-starter:1.1
The push refers to repository [docker.io/bharathmp7/nodejs-starter]
6a502fc4317a: Pushed
c750a0c75f0d: Pushed
13374defb42e: Pushed
4407d4d400d5: Pushed
311271778acb: Mounted from library/node
3fecbd1300ea: Mounted from library/node
443d37430985: Mounted from library/node
1d59188d1ae5: Mounted from library/node
6664dfb907f: Mounted from library/node
193c69a58521: Mounted from library/node
65bda7ee0f5: Mounted from library/node
e037b6ee6754: Mounted from library/node
22f8b5520ced: Mounted from library/node
1.1: digest: sha256:072285304c5fa2a83a241cb28aeeb613c9355c1ea5b82dd45029c57cade84bdc size: 3049
ubuntu@ip-172-31-1-207:~/nodejs$
```

Creating Deployment.yaml Using Online application called Katacode for Kubernetes.

Deploy Containers Using YAML

Copy the Service definition to the editor. The Service selects all applications with the label `webapp1`. As multiple replicas, or instances, are deployed, they will be automatically load balanced based on this common label. The Service makes the application available via a NodePort.

```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
labels:
  app: webapp1
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
  selector:
    app: webapp1
```

All Kubernetes objects are deployed in a consistent way using `kubectl`.

Deploy the Service with `kubectl create -f service.yaml`

As before, details of all the Service objects deployed with `kubectl get svc -o`. By describing the object it's possible to discover more details about the configuration

```
kubectl describe svc webapp1-svc
```

```
curl host01:30080
```

deployment.yaml

```
1 # This is your editor pane.
2 apiVersion: apps/v1 #1
3 kind: Deployment #2
4 metadata: #3
5   name: nodejs-deployment #4
6   spec: #5
7     replicas: 2 #6
8     selector: #7
9       matchLabels: #7
10        app: nodejs #7
11    template: #8
12      metadata: #9
13        labels: #10
14      spec: #11
15        containers: #12
16          - name: nodejs #13
17            image: bharat77/nodejs-starter:1.1 #15
18            ports: #16
19              - containerPort: 3000 #17
```

Terminal

```
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
$ minikube ip
172.17.0.26
$ kubectl create -f service.yaml
configmap/config created
$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
$ kubectl expose deployment nodejs-deployment --type=LoadBalancer
service/nodejs-deployment exposed
$ kubectl get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes           ClusterIP           10.96.0.1        <none>            443/TCP          6m20s
nodejs-deployment    LoadBalancer        10.102.196.108   <pending>         3000:32315/TCP   13s
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32315
Bharat77 Kalyan Anand Mokshith Sampath
```

Creating Service.yaml

Deploy Containers Using YAML

Copy the Service definition to the editor. The Service selects all applications with the label `webapp1`. As multiple replicas, or instances, are deployed, they will be automatically load balanced based on this common label. The Service makes the application available via a NodePort.

```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
labels:
  app: webapp1
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
  selector:
    app: webapp1
```

All Kubernetes objects are deployed in a consistent way using `kubectl`.

Deploy the Service with `kubectl create -f service.yaml`

As before, details of all the Service objects deployed with `kubectl get svc -o`. By describing the object it's possible to discover more details about the configuration

```
kubectl describe svc webapp1-svc
```

```
curl host01:30080
```

service.yaml

```
1 # This is your editor pane.
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   name: metallb-system
6   data:
7     config: |
8       address-pools:
9         - name: default
10         protocol: layer2
11       addresses:
12         - 172.17.15.81-172.17.15.50
```

Terminal

```
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
$ minikube ip
172.17.0.26
$ kubectl create -f service.yaml
configmap/config created
$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
$ kubectl expose deployment nodejs-deployment --type=LoadBalancer
service/nodejs-deployment exposed
$ kubectl get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes           ClusterIP           10.96.0.1        <none>            443/TCP          6m20s
nodejs-deployment    LoadBalancer        10.102.196.108   <pending>         3000:32315/TCP   13s
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32315
Bharat77 Kalyan Anand Mokshith Sampath
```

Activities Firefox Web Browser Sep 30 12:47

Inbox (339) - 18bc057@ x Meet - kox-qumt-jkh PLAYING x Developing and deploy... x Deploy Containers U x WhatsApp x Connect to instance | EC... x Docker Hub x +

https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions

TRY O'REILLY LOG IN

Deploy Containers Using YAML

Step 2 of 3

Copy the Service definition to the editor. The Service selects all applications with the label `webapp1`. As multiple replicas, or instances, are deployed, they will be automatically load balanced based on this common label. The Service makes the application available via a NodePort.

```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
labels:
  app: webapp1
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
  selector:
    app: webapp1
```

Copy to Editor

All Kubernetes objects are deployed in a consistent way using `kubectl`.

Deploy the Service with `kubectl create -f service.yaml`

As before, details of all the Service objects deployed with `kubectl get svc -f`. By describing the object it's possible to discover more details about the configuration

```
kubectl describe svc webapp1-svc
```

```
curl host01:30080
```

CONTINUE

deployment.yaml service.yaml

```
1 # This is your editor pane.
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   namespace: metallb-system
6   name: config
7 data:
8   config: |
9     address-pools:
10    - name: default
11      protocol: layer2
12    addresses:
13    - 172.17.15.41-172.17.15.51
```

Terminal

```
1 Your interactive Bash terminal: A safe place to learn and execute commands.
2
3 $
4 $ minikube start --wait=false
5 * minikube v1.8.1 on Ubuntu 18.04
6 * Using the none driver based on user configuration
7 * Running on localhost (CPUs=2, Memory=2460MB, Disk=14561MB) ...
8 * OS release is Ubuntu 18.04.4 LTS
9 * Preparing Kubernetes v1.17.3 on Docker 18.03.6 ...
10 * kubelet-resolv-conf=/run/systemd/resolve/resolve.conf
11 * Launching Kubernetes ...
12 * Enabling addons: default-storageclass, storage-provisioner
13 * Configuring local host environment ...
14 * Done! kubectl is now configured to use "minikube"
15 $
16 $ kubectl create -f deployment.yaml
17 deployment.apps/nodejs-deployment created
18 $ kubectl get deploy,po
19
20 deployment.apps/nodejs-deployment 0/2 2 0 22s
21
22 NAME READY STATUS AVAILABLE AGE
23 deployment.apps/nodejs-deployment 0/2 2 0 22s
24
25 NAME READY STATUS RESTARTS AGE
```

Activities Firefox Web Browser Sep 30 12:47

Inbox (339) - 18bc057@ x Meet - kox-qumt-jkh PLAYING x Developing and deploy... x Deploy Containers U x WhatsApp x Connect to instance | EC... x Docker Hub x +

https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions

TRY O'REILLY LOG IN

Deploy Containers Using YAML

Step 2 of 3

Copy the Service definition to the editor. The Service selects all applications with the label `webapp1`. As multiple replicas, or instances, are deployed, they will be automatically load balanced based on this common label. The Service makes the application available via a NodePort.

```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
labels:
  app: webapp1
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
  selector:
    app: webapp1
```

Copy to Editor

All Kubernetes objects are deployed in a consistent way using `kubectl`.

Deploy the Service with `kubectl create -f service.yaml`

As before, details of all the Service objects deployed with `kubectl get svc -f`. By describing the object it's possible to discover more details about the configuration

```
kubectl describe svc webapp1-svc
```

```
curl host01:30080
```

CONTINUE

deployment.yaml service.yaml

```
1 # This is your editor pane.
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   namespace: metallb-system
6   name: config
7 data:
8   config: |
9     address-pools:
10    - name: default
11      protocol: layer2
12    addresses:
13    - 172.17.15.41-172.17.15.51
```

Terminal

```
1 deployment.apps/nodejs-deployment 0/2 2 0 22s
2
3 NAME READY STATUS RESTARTS AGE
4 pod/nodejs-deployment-59b676ddb-q18pw 0/1 ContainerCreating 0 22s
5 pod/nodejs-deployment-59b676ddb-s18cm 0/1 ContainerCreating 0 22s
6 $ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
7 service/nodejs-deployment exposed
8 $ kubectl get svc
9
10 NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
11 nodejs-deployment LoadBalancer 10.97.16.51 <pending> 30080:32227/TCP 17s
12 namespace/metallb-system created
13 $ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/metallb.yaml
14 podsecuritypolicy.policy/controller created
15 podsecuritypolicy.policy/speaker created
16 serviceaccount/controller created
17 serviceaccount/speaker created
18 clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
19 clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
20 role.rbac.authorization.k8s.io/config-watcher created
21 role.rbac.authorization.k8s.io/pod-lister created
22 clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
```


The Users are shown

The screenshot displays a Katakoda lab interface for a tutorial titled "Deploy Containers Using YAML". The interface is divided into three main sections: a left sidebar with navigation icons, a central content area, and a right-hand editor/terminal area.

Central Content Area:

- Deploy Containers Using YAML** (Step 2 of 3)
- Instructions: "Copy the Service definition to the editor. The Service selects all applications with the label `webapp1`. As multiple replicas, or instances, are deployed, they will be automatically load balanced based on this common label. The Service makes the application available via a NodePort."
- Service Definition (YAML):

```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
  labels:
    app: webapp1
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
  selector:
    app: webapp1
```
- Instructions: "All Kubernetes objects are deployed in a consistent way using `kubectl`."
- Command: `kubectl create -f service.yaml`
- Instructions: "As before, details of all the Service objects deployed with `kubectl get svc -l`. By describing the object it's possible to discover more details about the configuration"
- Command: `kubectl describe svc webapp1-svc -l`
- Terminal Command: `curl host01:30080` (Status: ✓)
- CONTINUE** button

Right-Hand Editor/Terminal Area:

- deployment.yaml** (Left Pane):

```
1 # this is your editor pane.
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   namespace: metallb-system
6   name: config
7 data:
8   config: |
9     address-pools:
10     - name: default
11       protocol: layer2
12     addresses:
13       172.17.15.41-172.17.15.51
```
- service.yaml** (Right Pane):

```
1 # this is your editor pane.
2 apiVersion: v1
3 kind: Service
4 metadata:
5   namespace: metallb-system
6   name: config
7 data:
8   config: |
9     address-pools:
10     - name: default
11       protocol: layer2
12     addresses:
13       172.17.15.41-172.17.15.51
```
- Terminal** (Bottom Pane):

```
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
$ minikube ip
172.17.0.26
$ kubectl create -f service.yaml
configmap/config created
$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
$ kubectl expose deployment nodejs-deployment --type=LoadBalancer
service/nodejs-deployment exposed
$ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes                         ClusterIP           10.96.0.1        <none>            443/TCP          6m20s
nodejs-deployment                  LoadBalancer        10.102.196.108   <pending>         3000:32315/TCP   13s
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32315
Bharatshub Kalyan Anand Mokshith Sampath$
```