Dharwad
ज्ञानेन विकासः

INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

DevOps (CS457)

ASSIGNMENT 1: Task 2

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

Submitted to:
Dr. Uma S

Submitted by:

Team 1
Sumith Sai Budde (18BCS101)

Syed Sufyan Ahmed (18BCS103)

Shaik Fharook (18BCS091)

Parvati Jayakumar (18BEC036)

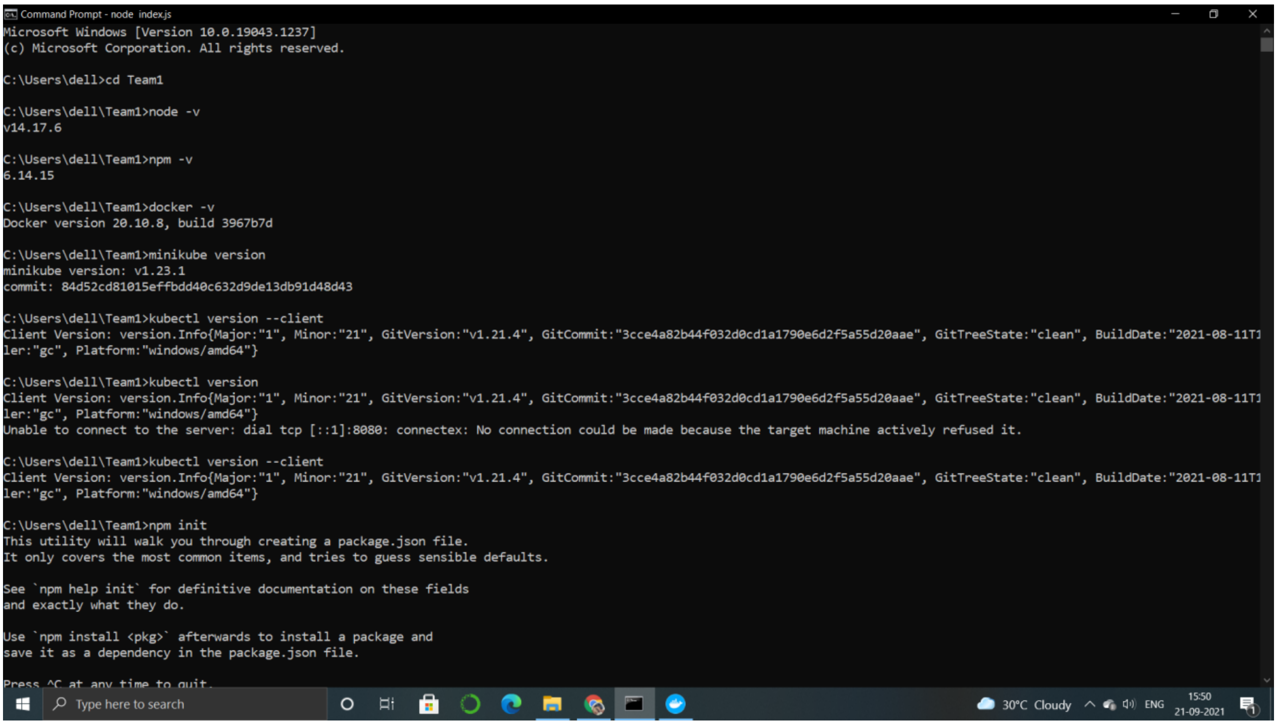
P Chethan Krishna (18BEC040)

G Rithika (18BCS031)

Pokala Dattatreya (18BCS067)

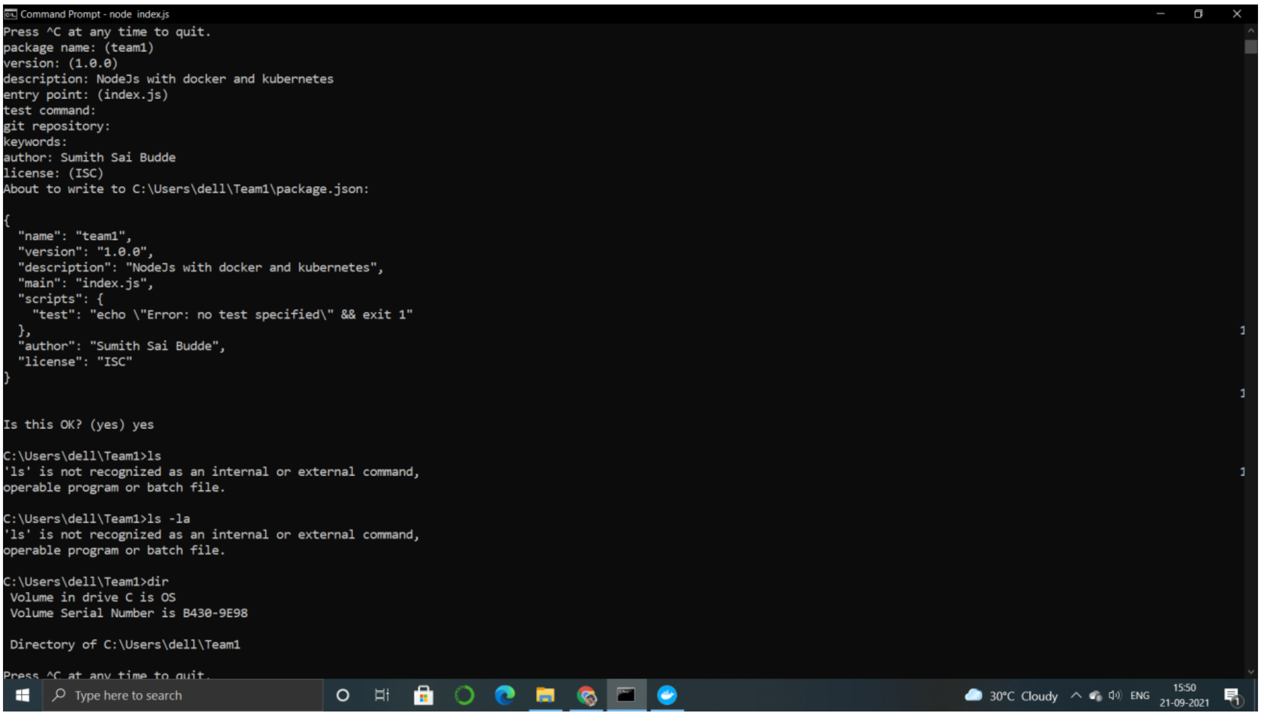
Rama Dundi Saketh (18BCS076)

Installing and checking node , npm ,docker ,minikube ,kubectl .



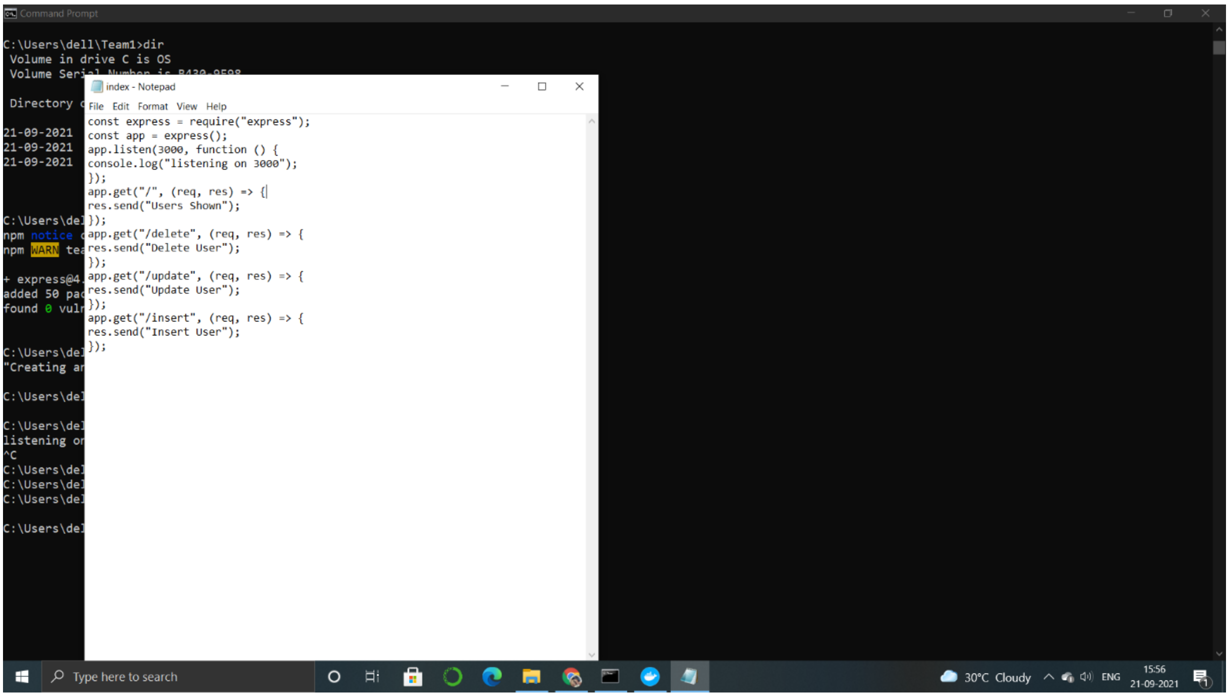
```
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dell>cd Team1

C:\Users\dell\Team1>node -v
v14.17.6

C:\Users\dell\Team1>npm -v
6.14.15

C:\Users\dell\Team1>docker -v
Docker version 20.10.8, build 3967b7d

C:\Users\dell\Team1>minikube version
minikube version: v1.23.1
commit: 84d52cd81015effbdd40c632d9de13db91d48d43

C:\Users\dell\Team1>kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.4", GitCommit:"3cce4a82b44f032d0cd1a1790e6d2f5a55d20aae", GitTreeState:"clean", BuildDate:"2021-08-11T1
ler:"gc", Platform:"windows/amd64"}

C:\Users\dell\Team1>kubectl version
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.4", GitCommit:"3cce4a82b44f032d0cd1a1790e6d2f5a55d20aae", GitTreeState:"clean", BuildDate:"2021-08-11T1
ler:"gc", Platform:"windows/amd64"}
Unable to connect to the server: dial tcp [::1]:8080: connectex: No connection could be made because the target machine actively refused it.

C:\Users\dell\Team1>kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.4", GitCommit:"3cce4a82b44f032d0cd1a1790e6d2f5a55d20aae", GitTreeState:"clean", BuildDate:"2021-08-11T1
ler:"gc", Platform:"windows/amd64"}

C:\Users\dell\Team1>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.
```

Initializing the node application and installing express server :

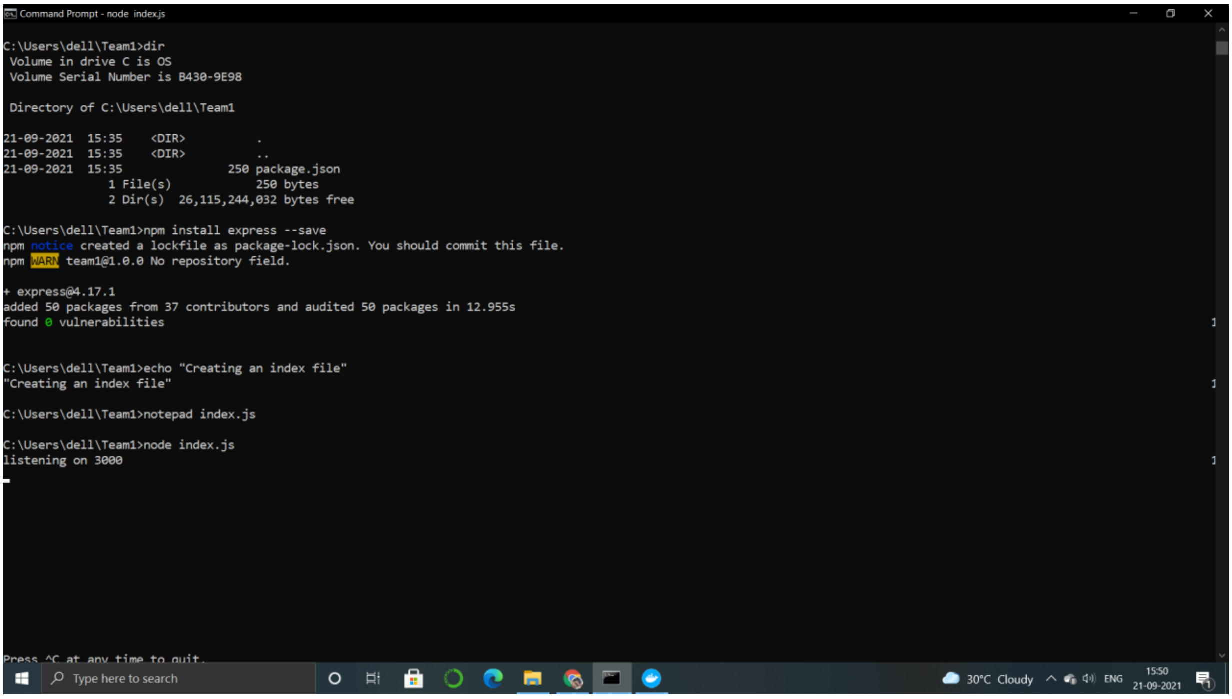


```
Press ^C at any time to quit.
package name: (team1)
version: (1.0.0)
description: NodeJs with docker and kubernetes
entry point: (index.js)
test command:
git repository:
keywords:
author: Sumith Sai Budde
license: (ISC)
About to write to C:\Users\dell\Team1\package.json:

{
  "name": "team1",
  "version": "1.0.0",
  "description": "NodeJs with docker and kubernetes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Sumith Sai Budde",
  "license": "ISC"
}


Is this OK? (yes) yes

C:\Users\dell\Team1>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\dell\Team1>ls -la
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\dell\Team1>dir
 Volume in drive C is OS
 Volume Serial Number is B430-9E98

 Directory of C:\Users\dell\Team1

Press ^C at any time to quit.
```

<u>Configuring the application :</u>

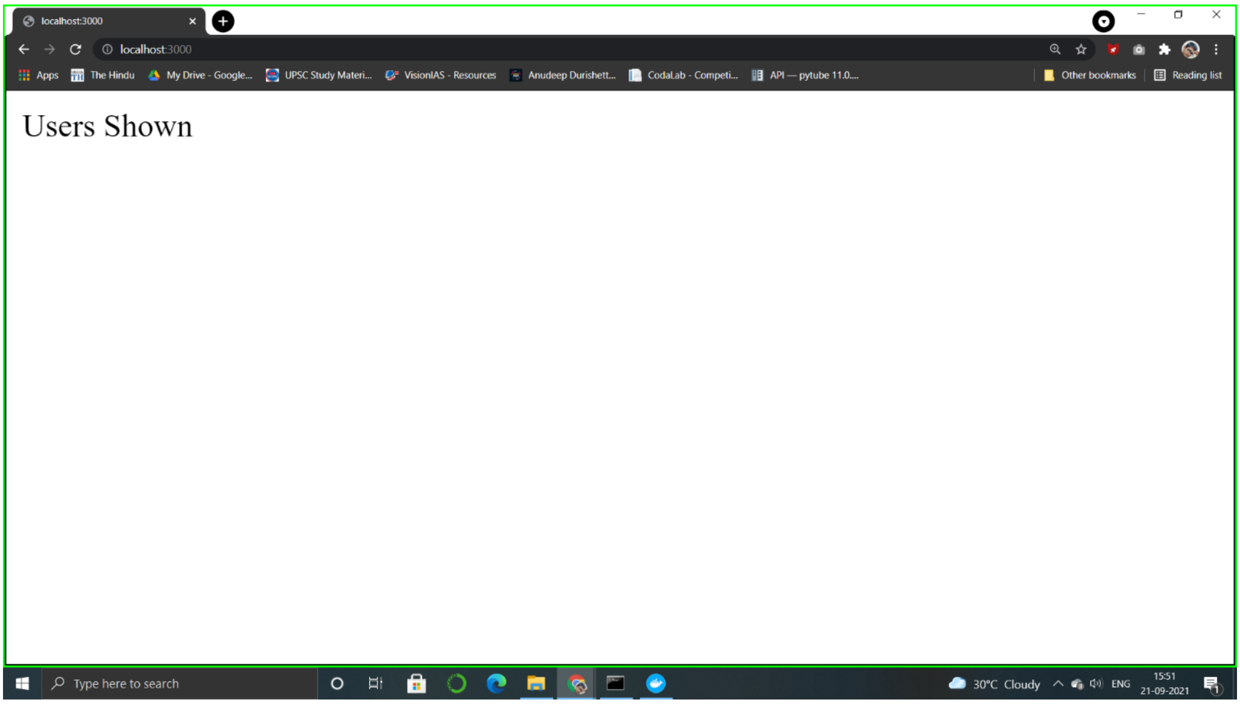


```
const express = require("express");
const app = express();
app.listen(3000, function () {
console.log("listening on 3000");
});
app.get("/", (req, res) => {
res.send("Users Shown");
});
app.get("/delete", (req, res) => {
res.send("Delete User");
});
app.get("/update", (req, res) => {
res.send("Update User");
});
app.get("/insert", (req, res) => {
res.send("Insert User");
});
```

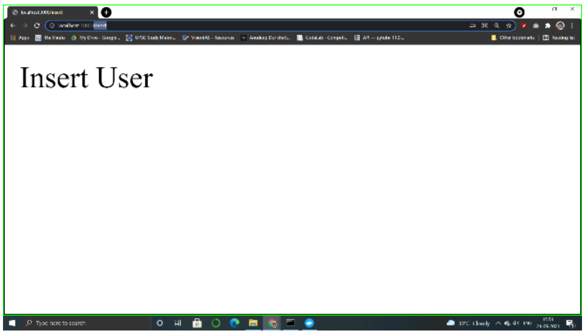


```
C:\Users\dell\Team1>dir
 Volume in drive C is OS
 Volume Serial Number is B430-9E98

 Directory of C:\Users\dell\Team1

21-09-2021  15:35    <DIR>          .
21-09-2021  15:35    <DIR>          ..
21-09-2021  15:35               250 package.json
               1 File(s)            250 bytes
               2 Dir(s)  26,115,244,032 bytes free

C:\Users\dell\Team1>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN team1@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 12.955s
found 0 vulnerabilities


C:\Users\dell\Team1>echo "Creating an index file"
"Creating an index file"

C:\Users\dell\Team1>notepad index.js

C:\Users\dell\Team1>node index.js
listening on 3000

Press ^C at any time to quit.
```

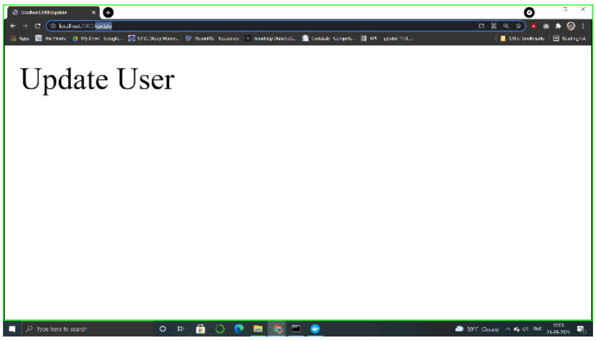
Nodejs WebApp: checking the server by browsing localhost:3000/



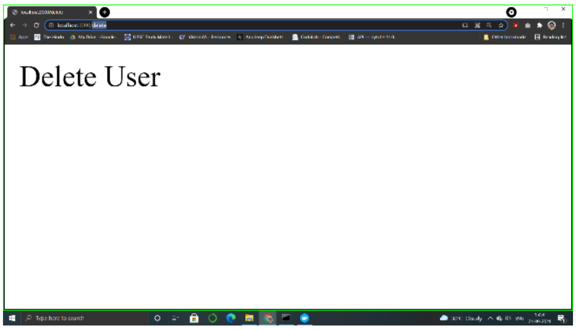
Users Shown



Insert User

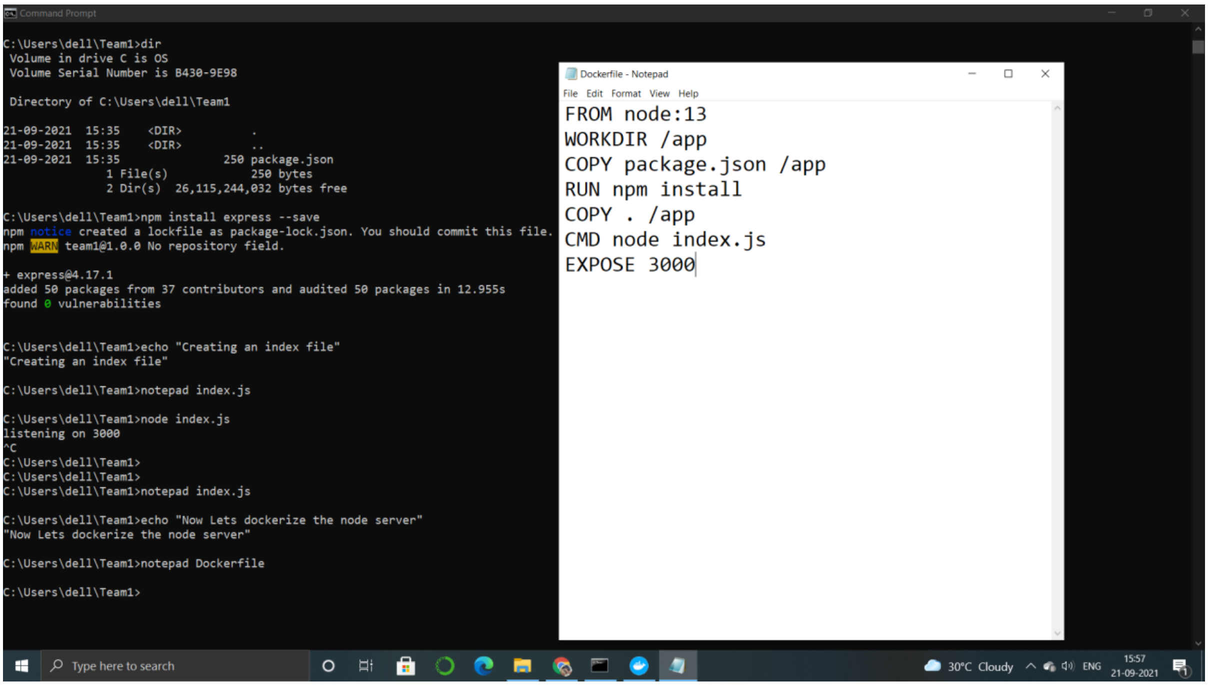


Update User



Delete User

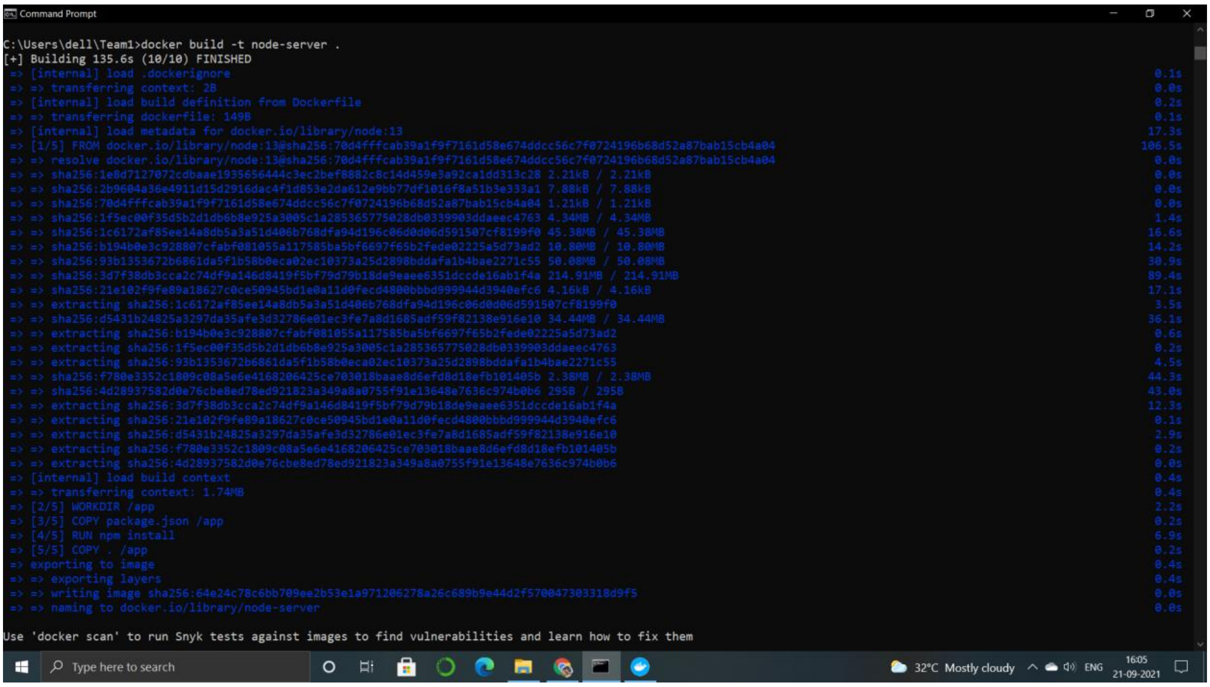
Dockerizing the node server :



```
C:\Users\dell\Team1>dir
 Volume in drive C is OS
 Volume Serial Number is B430-9E98

 Directory of C:\Users\dell\Team1

21-09-2021  15:35    <DIR>          .
21-09-2021  15:35    <DIR>          ..
21-09-2021  15:35               250 package.json
               1 File(s)            250 bytes
               2 Dir(s)  26,115,244,032 bytes free

C:\Users\dell\Team1>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN team1@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 12.955s
found 0 vulnerabilities


C:\Users\dell\Team1>echo "Creating an index file"
"Creating an index file"

C:\Users\dell\Team1>notepad index.js

C:\Users\dell\Team1>node index.js
listening on 3000
^C
C:\Users\dell\Team1>
C:\Users\dell\Team1>
C:\Users\dell\Team1>notepad index.js

C:\Users\dell\Team1>echo "Now Lets dockerize the node server"
"Now Lets dockerize the node server"

C:\Users\dell\Team1>notepad Dockerfile

C:\Users\dell\Team1>
```

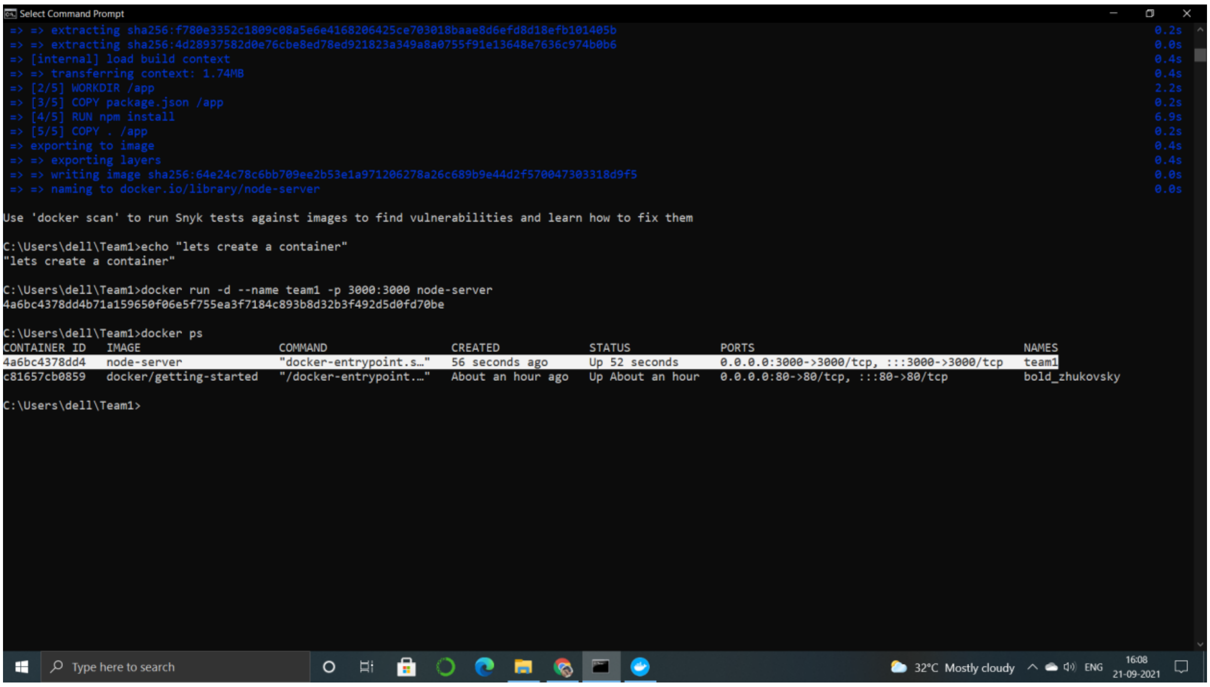
```
FROM node:13
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
CMD node index.js
EXPOSE 3000
```

Building the image :



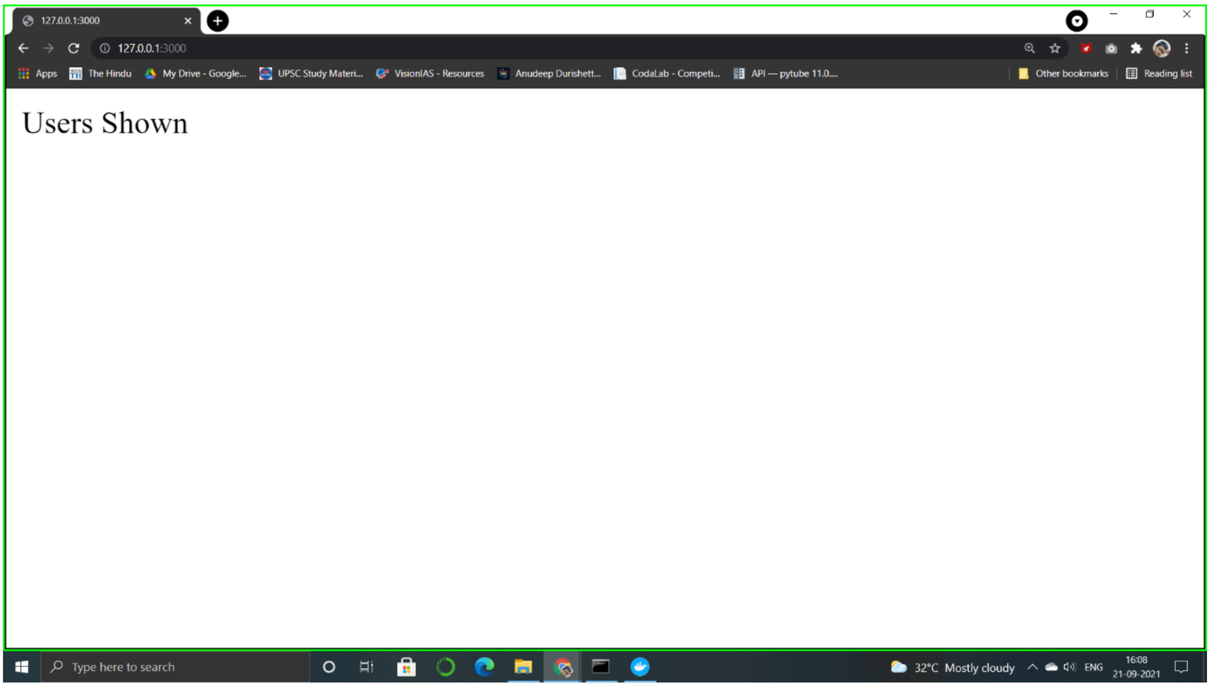
```
C:\Users\dell\Team1>docker build -t node-server .
[+] Building 135.6s (10/10) FINISHED
 => [internal] load .dockerignore                                                                                        0.1s
 => => transferring context: 2B                                                                                          0.0s
 => [internal] load build definition from Dockerfile                                                                     0.2s
 => => transferring dockerfile: 149B                                                                                     0.1s
 => [internal] load metadata for docker.io/library/node:13                                                              17.3s
 => [1/5] FROM docker.io/library/node:13@sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04       106.5s
 => => resolve docker.io/library/node:13@sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04         0.0s
 => => sha256:1e8d7127072cdbaae1935656444c3ec2bef8882c8c14d459e3a92ca1dd313c28 2.21kB / 2.21kB                          0.0s
 => => sha256:2b9604a36e4911d15d2916dac4f1d853e2da612e9bb77df1016f8a51b3e333a1 7.88kB / 7.88kB                          0.0s
 => => sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04 1.21kB / 1.21kB                          0.0s
 => => sha256:1f5ec00f35d5b2d1db6b8e925a3005c1a285365775028db0339903ddaeec4763 4.34MB / 4.34MB                          1.4s
 => => sha256:1c6172af85ee14a8db5a3a51d406b768dfa94d196c06d0d06d591507cf8199f0 45.38MB / 45.38MB                       16.6s
 => => sha256:b194b0e3c928807cfabf081055a117585ba5bf6697f65b2fede02225a5d73ad2 10.80MB / 10.80MB                       14.2s
 => => sha256:93b1353672b6861da5f1b58b0eca02ec10373a25d2898bddafa1b4bae2271c55 50.08MB / 50.08MB                       30.9s
 => => sha256:3d7f38db3cca2c74df9a146d8419f5bf79d79b18de9eaee6351dccde16ab1f4a 214.91MB / 214.91MB                     89.4s
 => => sha256:21e102f9fe89a18627c0ce50945bd1e0a11d0fecd4800bbbd999944d3940efc6 4.16kB / 4.16kB                         17.1s
 => => extracting sha256:1c6172af85ee14a8db5a3a51d406b768dfa94d196c06d0d06d591507cf8199f0                               3.5s
 => => sha256:d5431b24825a3297da35afe3d32786e01ec3fe7a8d1685adf59f82138e916e10 34.44MB / 34.44MB                       36.1s
 => => extracting sha256:b194b0e3c928807cfabf081055a117585ba5bf6697f65b2fede02225a5d73ad2                               0.6s
 => => extracting sha256:1f5ec00f35d5b2d1db6b8e925a3005c1a285365775028db0339903ddaeec4763                               0.2s
 => => extracting sha256:93b1353672b6861da5f1b58b0eca02ec10373a25d2898bddafa1b4bae2271c55                               4.5s
 => => sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b 2.38MB / 2.38MB                         44.3s
 => => sha256:4d28937582d0e76cbe8ed78ed921823a349a8a0755f91e13648e7636c974b0b6 295B / 295B                             43.0s
 => => extracting sha256:3d7f38db3cca2c74df9a146d8419f5bf79d79b18de9eaee6351dccde16ab1f4a                              12.3s
 => => extracting sha256:21e102f9fe89a18627c0ce50945bd1e0a11d0fecd4800bbbd999944d3940efc6                               0.1s
 => => extracting sha256:d5431b24825a3297da35afe3d32786e01ec3fe7a8d1685adf59f82138e916e10                               2.9s
 => => extracting sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b                               0.2s
 => => extracting sha256:4d28937582d0e76cbe8ed78ed921823a349a8a0755f91e13648e7636c974b0b6                               0.0s
 => [internal] load build context                                                                                        0.4s
 => => transferring context: 1.74MB                                                                                      0.4s
 => [2/5] WORKDIR /app                                                                                                   2.2s
 => [3/5] COPY package.json /app                                                                                         0.2s
 => [4/5] RUN npm install                                                                                                6.9s
 => [5/5] COPY . /app                                                                                                    0.2s
 => exporting to image                                                                                                   0.4s
 => => exporting layers                                                                                                  0.4s
 => => writing image sha256:64e24c78c6bb709ee2b53e1a971206278a26c689b9e44d2f570047303318d9f5                             0.0s
 => => naming to docker.io/library/node-server                                                                           0.0s

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

Creating a container and running it :



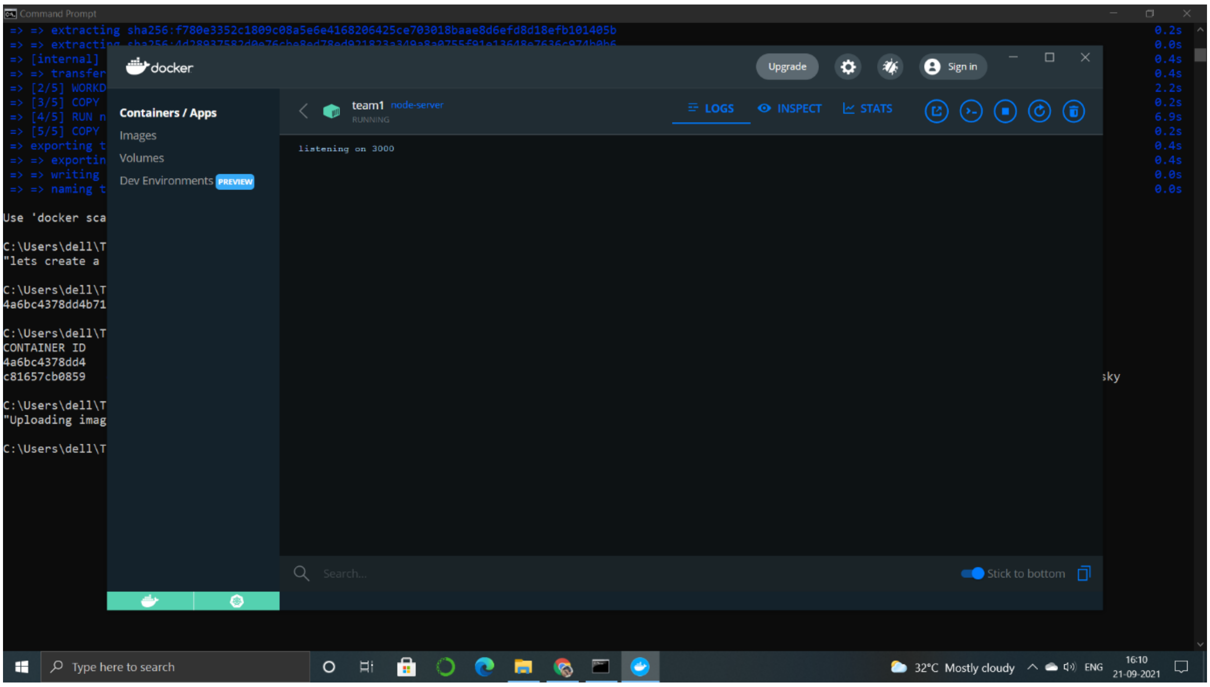
```
 => => extracting sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b      0.2s
 => => extracting sha256:4d28937582d0e76cbe8ed78ed921823a349a8a0755f91e13648e7636c974b0b6      0.0s
 => [internal] load build context                                                            0.4s
 => => transferring context: 1.74MB                                                          0.4s
 => [2/5] WORKDIR /app                                                                       2.2s
 => [3/5] COPY package.json /app                                                             0.2s
 => [4/5] RUN npm install                                                                    6.9s
 => [5/5] COPY . /app                                                                        0.2s
 => exporting to image                                                                       0.4s
 => => exporting layers                                                                      0.4s
 => => writing image sha256:64e24c78c6bb709ee2b53e1a971206278a26c689b9e44d2f570047303318d9f5  0.0s
 => => naming to docker.io/library/node-server                                               0.0s

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

C:\Users\dell\Team1>echo "lets create a container"
"lets create a container"

C:\Users\dell\Team1>docker run -d --name team1 -p 3000:3000 node-server
4a6bc4378dd4b71a159650f06e5f755ea3f7184c893b8d32b3f492d5d0fd70be

C:\Users\dell\Team1>docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED             STATUS              PORTS                                       NAMES
4a6bc4378dd4   node-server              "docker-entrypoint.s…"   56 seconds ago      Up 52 seconds       0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   team1
c81657cb0859   docker/getting-started   "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:80->80/tcp, :::80->80/tcp           bold_zhukovsky

C:\Users\dell\Team1>
```

Testing if the container is running by browsing 127.0.0.1:3000/ :



Users Shown

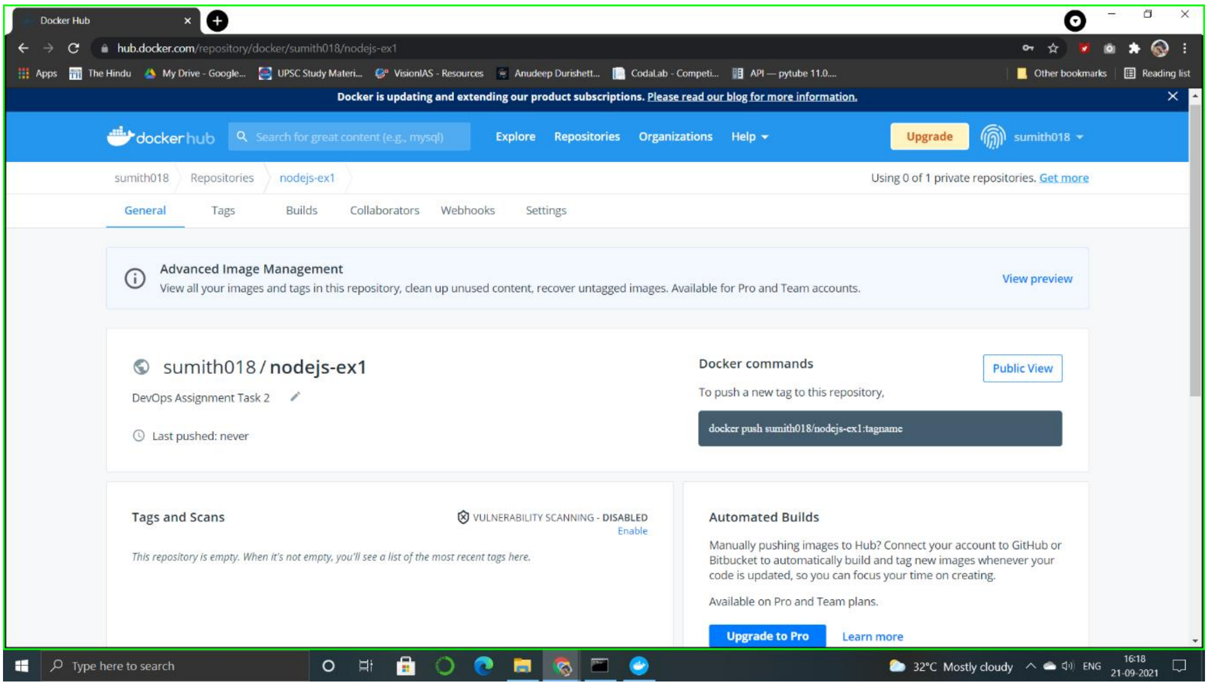
Testing if the container is running using docker desktop :



```
=> => extracting sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b    0.2s
=> => extracting                                                                              0.0s
=> [internal]                                                                                 0.4s
=> => transfer                                                                                0.4s
=> [2/5] WORKD                                                                                2.2s
=> [3/5] COPY                                                                                 0.2s
=> [4/5] RUN n                                                                                6.9s
=> [5/5] COPY                                                                                 0.2s
=> exporting t                                                                                0.4s
=> => exportin                                                                                0.4s
=> => writing                                                                                 0.0s
=> => naming t                                                                                0.0s

Use 'docker sca

C:\Users\dell\T
"lets create a

C:\Users\dell\T
4a6bc4378dd4b71

C:\Users\dell\T
CONTAINER ID
4a6bc4378dd4
c81657cb0859                                                                         sky

C:\Users\dell\T
"Uploading imag

C:\Users\dell\T
```

```
listening on 3000
```

Uploading the image to Docker Hub into a repository nodejs-ex1 :



Docker is updating and extending our product subscriptions. Please read our blog for more information.

Using 0 of 1 private repositories. Get more

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.

sumith018 / nodejs-ex1

DevOps Assignment Task 2

Last pushed: never

Docker commands

To push a new tag to this repository,

```
docker push sumith018/nodejs-ex1:tagname
```

Tags and Scans

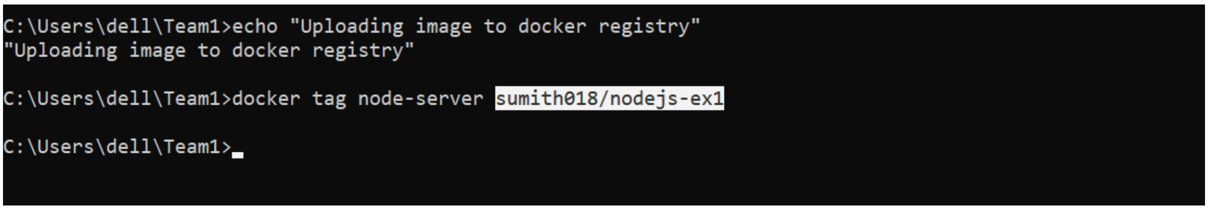
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

Automated Builds

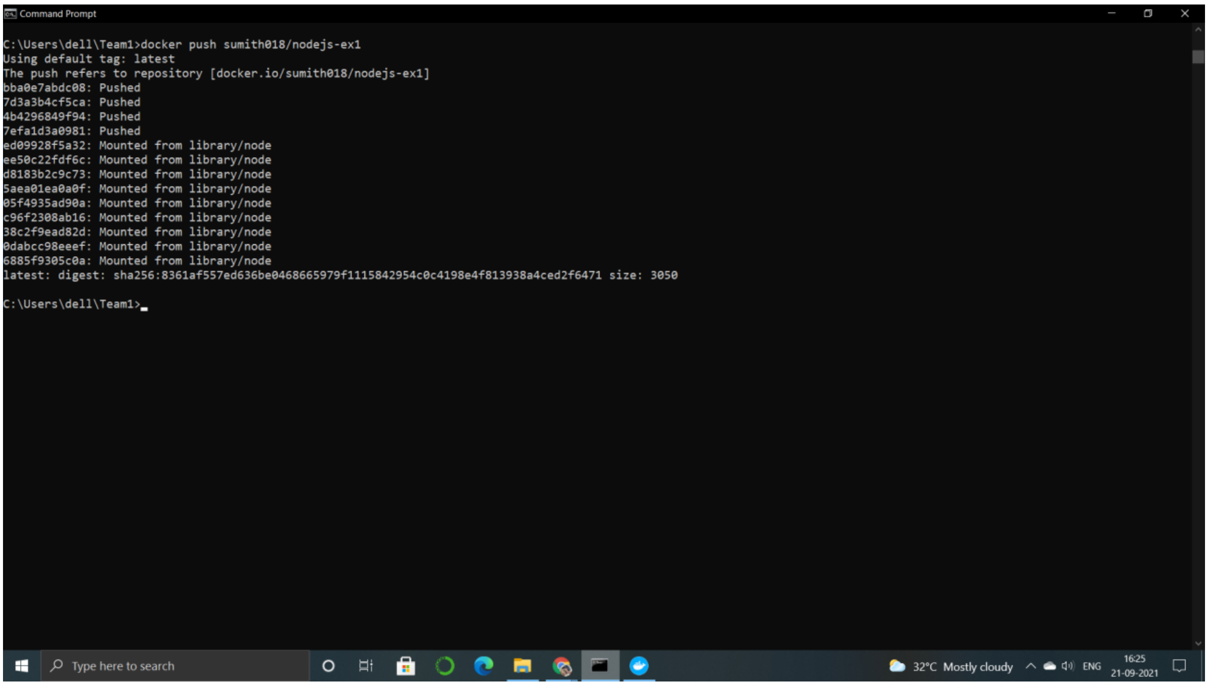
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available on Pro and Team plans.

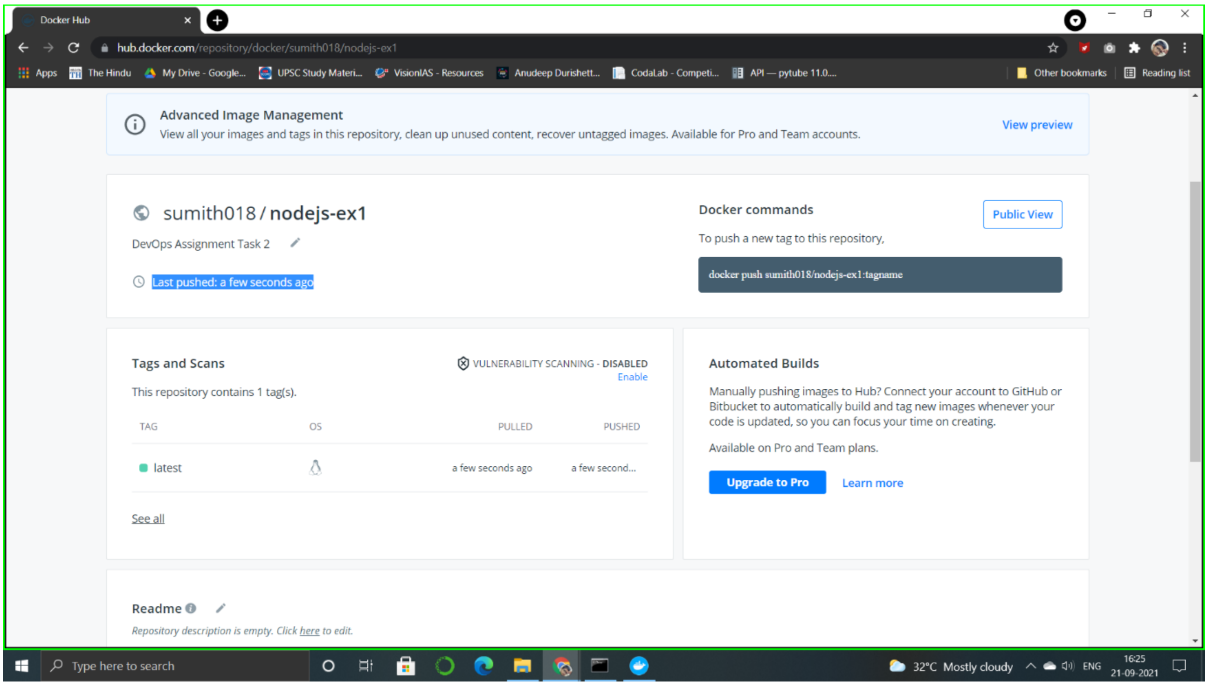
Tagging and pushing the docker image to Docker Hub :



```
C:\Users\dell\Team1>echo "Uploading image to docker registry"
"Uploading image to docker registry"

C:\Users\dell\Team1>docker tag node-server sumith018/nodejs-ex1

C:\Users\dell\Team1>_
```



```
C:\Users\dell\Team1>docker push sumith018/nodejs-ex1
Using default tag: latest
The push refers to repository [docker.io/sumith018/nodejs-ex1]
bba0e7abdc08: Pushed
7d3a3b4cf5ca: Pushed
4b4296849f94: Pushed
7efa1d3a0981: Pushed
ed09928f5a32: Mounted from library/node
ee50c22fdf6c: Mounted from library/node
d8183b2c9c73: Mounted from library/node
5aea01ea0a0f: Mounted from library/node
05f4935ad90a: Mounted from library/node
c96f2308ab16: Mounted from library/node
38c2f9ead82d: Mounted from library/node
0dabcc98eeef: Mounted from library/node
6885f9305c0a: Mounted from library/node
latest: digest: sha256:8361af557ed636be0468665979f1115842954c0c4198e4f813938a4ced2f6471 size: 3050

C:\Users\dell\Team1>_
```



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.

sumith018 / **nodejs-ex1**

DevOps Assignment Task 2

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push sumith018/nodejs-ex1:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		a few seconds ago	a few second...

Automated Builds

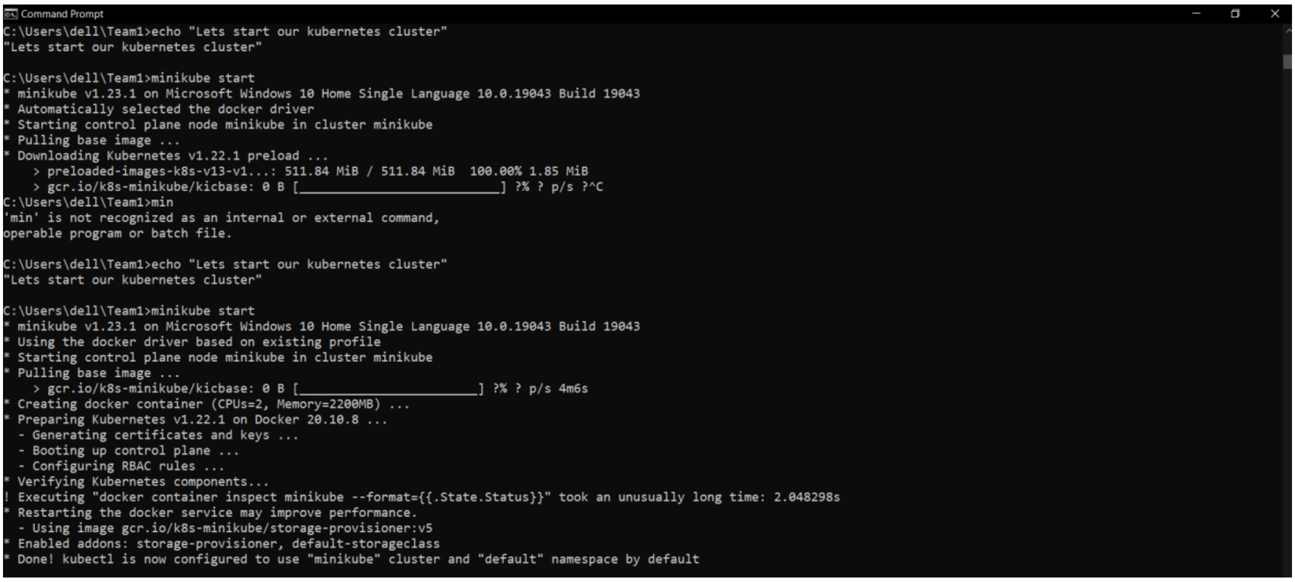
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available on Pro and Team plans.

Readme

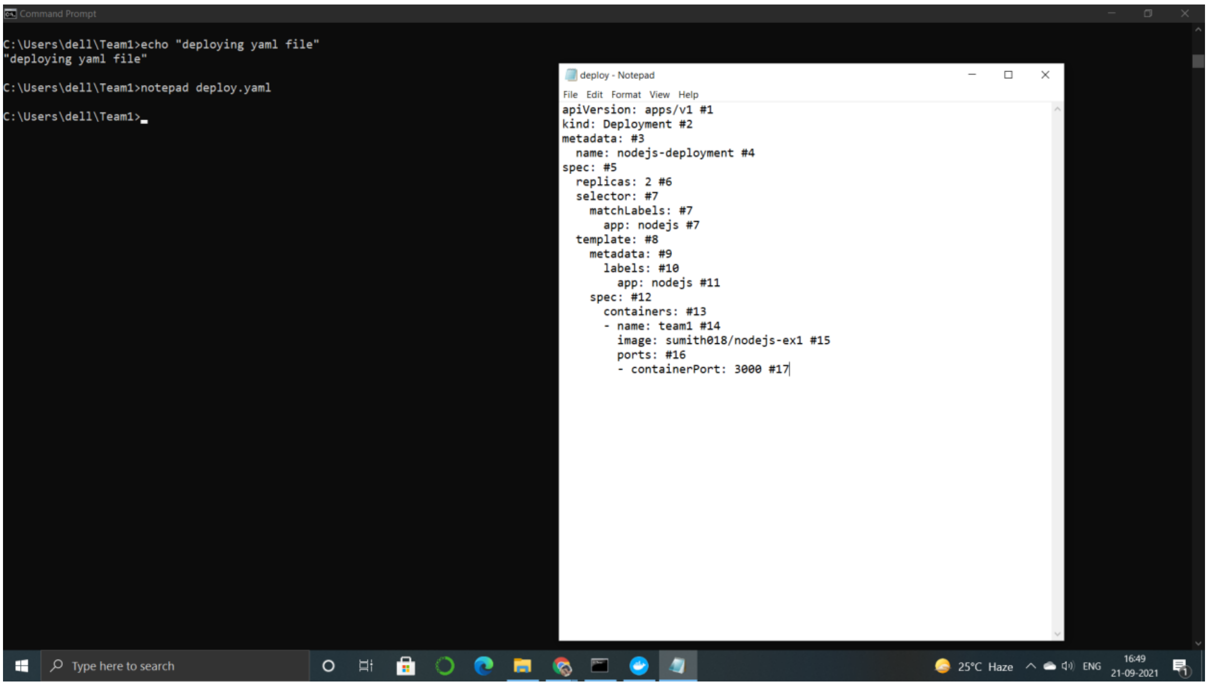
Repository description is empty. Click here to edit.

Starting the Kubernetes cluster using minikube :



```
C:\Users\dell\Team1>echo "Lets start our kubernetes cluster"
"Lets start our kubernetes cluster"

C:\Users\dell\Team1>minikube start
* minikube v1.23.1 on Microsoft Windows 10 Home Single Language 10.0.19043 Build 19043
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.22.1 preload ...
    > preloaded-images-k8s-v13-v1...: 511.84 MiB / 511.84 MiB  100.00% 1.85 MiB
    > gcr.io/k8s-minikube/kicbase: 0 B [___________________________] ?% ? p/s ?^C
C:\Users\dell\Team1>min
'min' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\dell\Team1>echo "Lets start our kubernetes cluster"
"Lets start our kubernetes cluster"

C:\Users\dell\Team1>minikube start
* minikube v1.23.1 on Microsoft Windows 10 Home Single Language 10.0.19043 Build 19043
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
    > gcr.io/k8s-minikube/kicbase: 0 B [________________________] ?% ? p/s 4m6s
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.22.1 on Docker 20.10.8 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 2.048298s
* Restarting the docker service may improve performance.
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

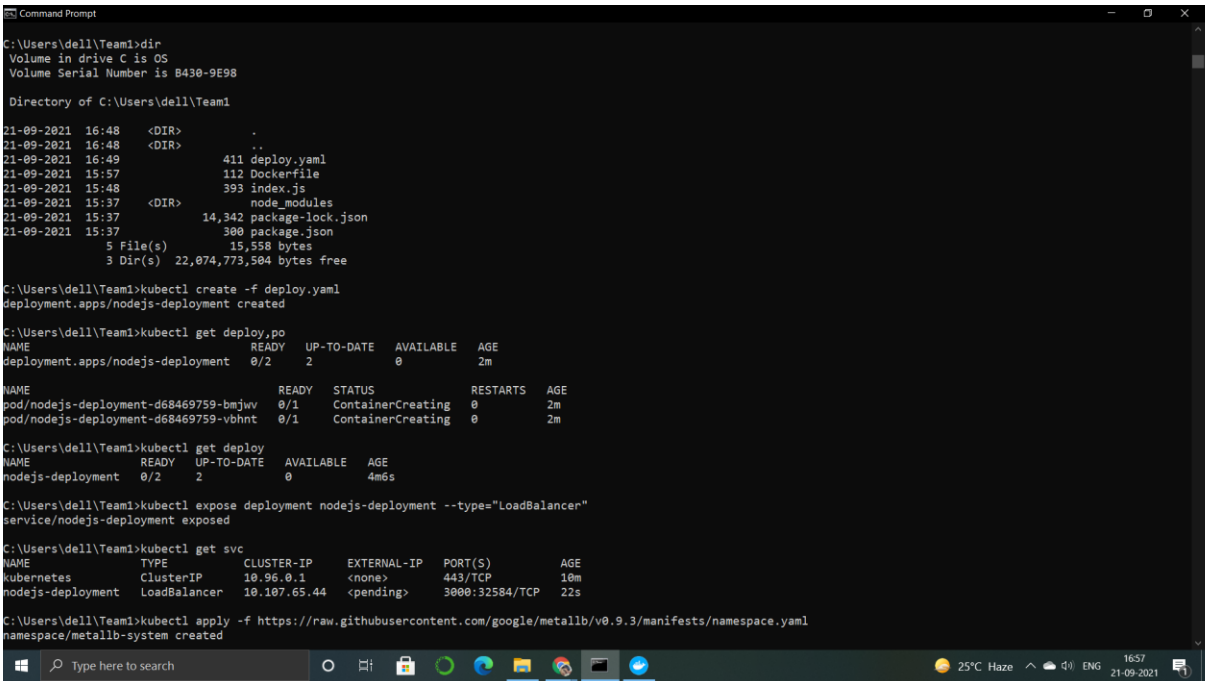
Defining a yaml file to create a deployment in our cluster :



```
C:\Users\dell\Team1>echo "deploying yaml file"
"deploying yaml file"

C:\Users\dell\Team1>notepad deploy.yaml

C:\Users\dell\Team1>_
```

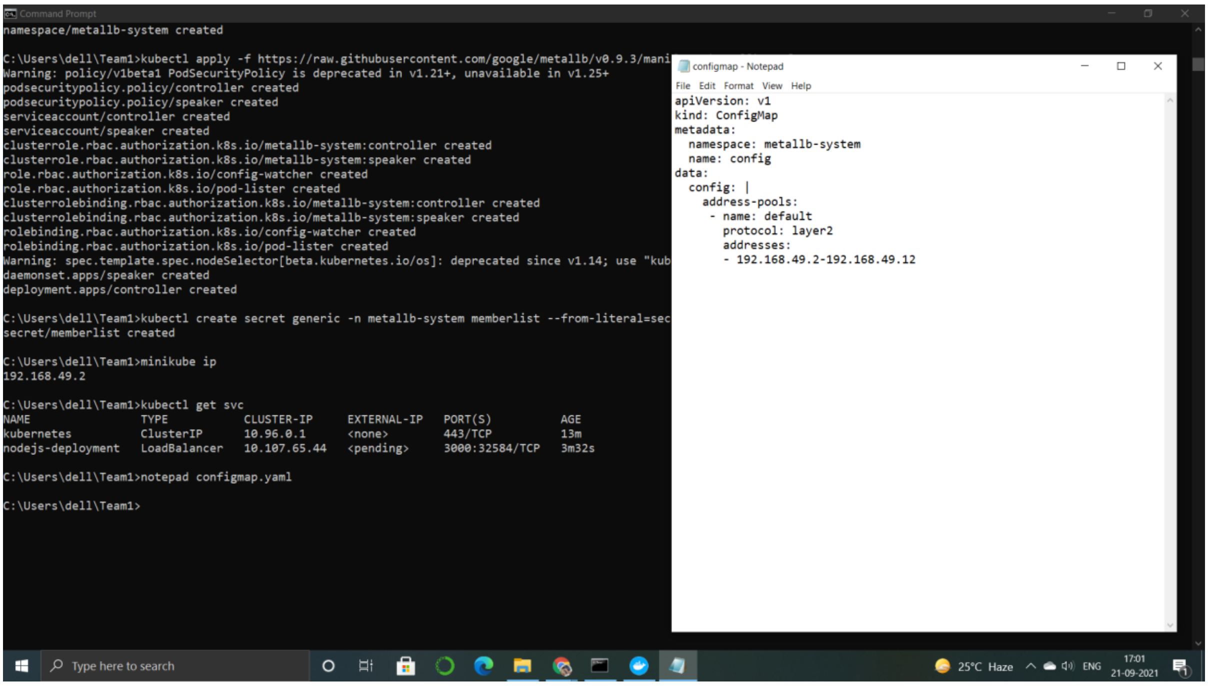
```
apiVersion: apps/v1 #1
kind: Deployment #2
metadata: #3
  name: nodejs-deployment #4
spec: #5
  replicas: 2 #6
  selector: #7
    matchLabels: #7
      app: nodejs #7
  template: #8
    metadata: #9
      labels: #10
        app: nodejs #11
    spec: #12
      containers: #13
      - name: team1 #14
        image: sumith018/nodejs-ex1 #15
        ports: #16
        - containerPort: 3000 #17
```

Creating a deployment in the cluster using yaml file defined above and exposing it to internet :



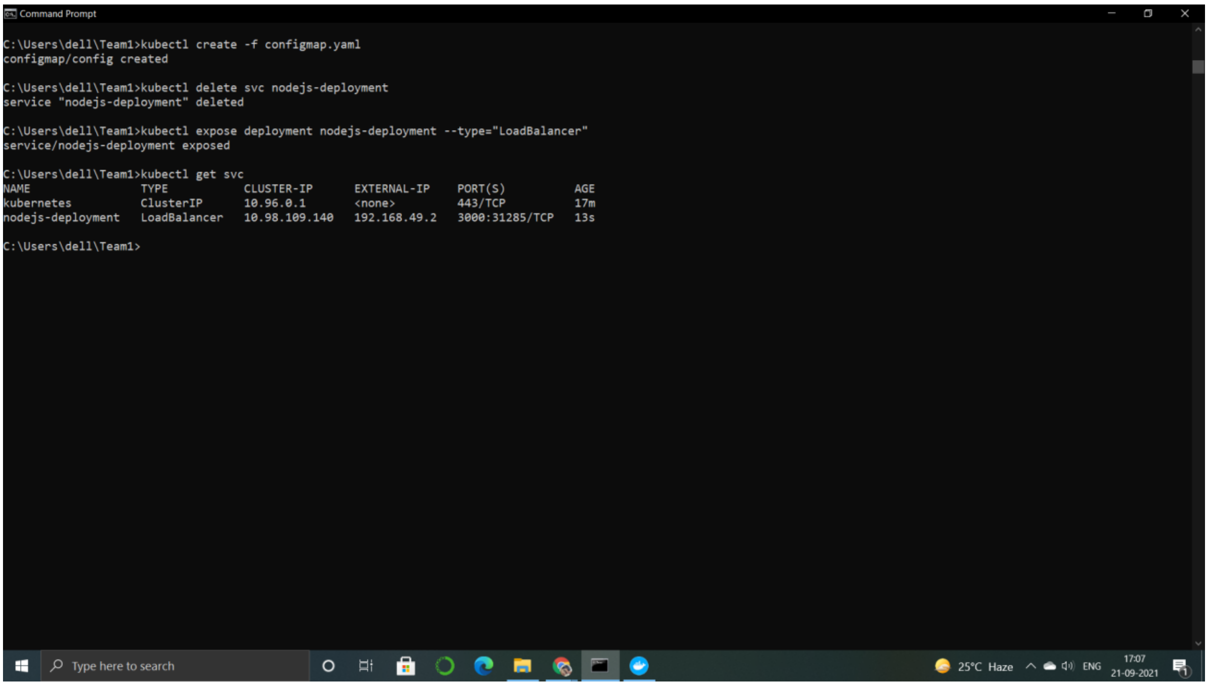
```
C:\Users\dell\Team1>dir
 Volume in drive C is OS
 Volume Serial Number is B430-9E98

 Directory of C:\Users\dell\Team1

21-09-2021  16:48    <DIR>          .
21-09-2021  16:48    <DIR>          ..
21-09-2021  16:49               411 deploy.yaml
21-09-2021  15:57               112 Dockerfile
21-09-2021  15:48               393 index.js
21-09-2021  15:37    <DIR>          node_modules
21-09-2021  15:37            14,342 package-lock.json
21-09-2021  15:37               300 package.json
               5 File(s)         15,558 bytes
               3 Dir(s)  22,074,773,504 bytes free

C:\Users\dell\Team1>kubectl create -f deploy.yaml
deployment.apps/nodejs-deployment created

C:\Users\dell\Team1>kubectl get deploy,po
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-deployment   0/2     2            0           2m

NAME                                    READY   STATUS              RESTARTS   AGE
pod/nodejs-deployment-d68469759-bmjwv   0/1     ContainerCreating   0          2m
pod/nodejs-deployment-d68469759-vbhnt   0/1     ContainerCreating   0          2m

C:\Users\dell\Team1>kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nodejs-deployment   0/2     2            0           4m6s

C:\Users\dell\Team1>kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed

C:\Users\dell\Team1>kubectl get svc
NAME                TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1      <none>        443/TCP          10m
nodejs-deployment   LoadBalancer   10.107.65.44   <pending>     3000:32584/TCP   22s

C:\Users\dell\Team1>kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespace.yaml
namespace/metallb-system created
```

Using MetalLB in our minikube environment to get an external ip :

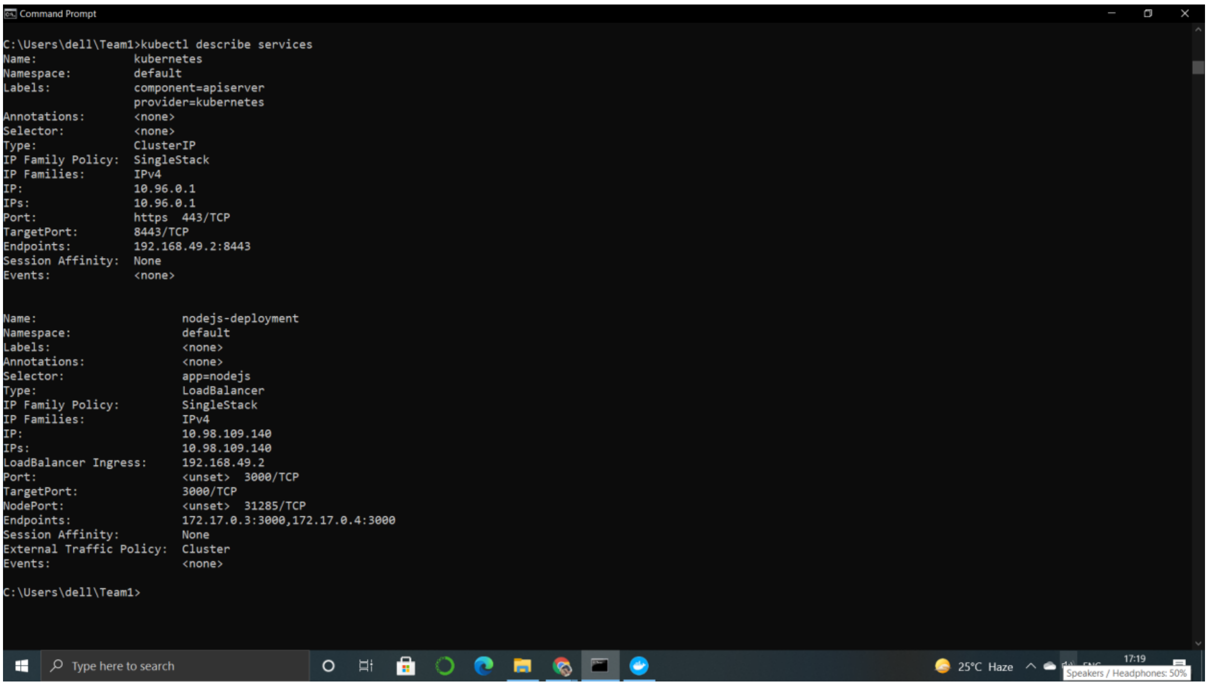


```
namespace/metallb-system created

C:\Users\dell\Team1>kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/mani
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14; use "kub
daemonset.apps/speaker created
deployment.apps/controller created

C:\Users\dell\Team1>kubectl create secret generic -n metallb-system memberlist --from-literal=sec
secret/memberlist created

C:\Users\dell\Team1>minikube ip
192.168.49.2

C:\Users\dell\Team1>kubectl get svc
NAME                TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1      <none>        443/TCP          13m
nodejs-deployment   LoadBalancer   10.107.65.44   <pending>     3000:32584/TCP   3m32s

C:\Users\dell\Team1>notepad configmap.yaml

C:\Users\dell\Team1>
```

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
     - name: default
       protocol: layer2
       addresses:
       - 192.168.49.2-192.168.49.12
```

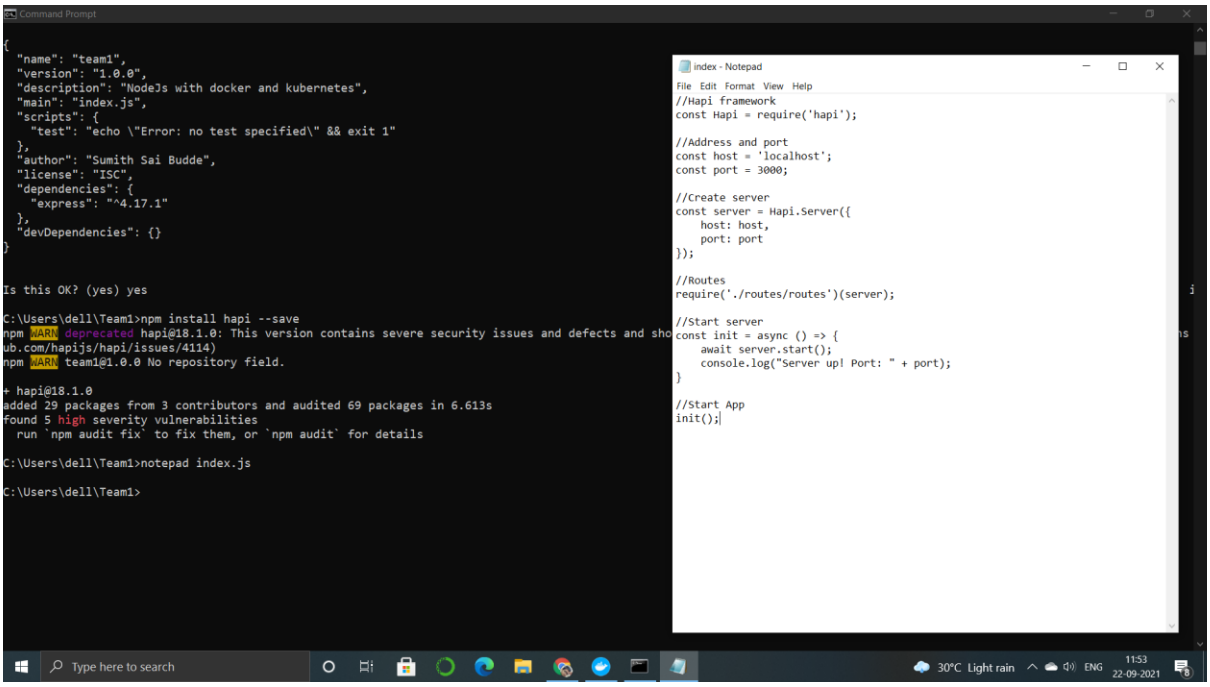


```
C:\Users\dell\Team1>kubectl create -f configmap.yaml
configmap/config created

C:\Users\dell\Team1>kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted

C:\Users\dell\Team1>kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed

C:\Users\dell\Team1>kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1       <none>         443/TCP          17m
nodejs-deployment   LoadBalancer   10.98.109.140   192.168.49.2   3000:31285/TCP   13s

C:\Users\dell\Team1>
```



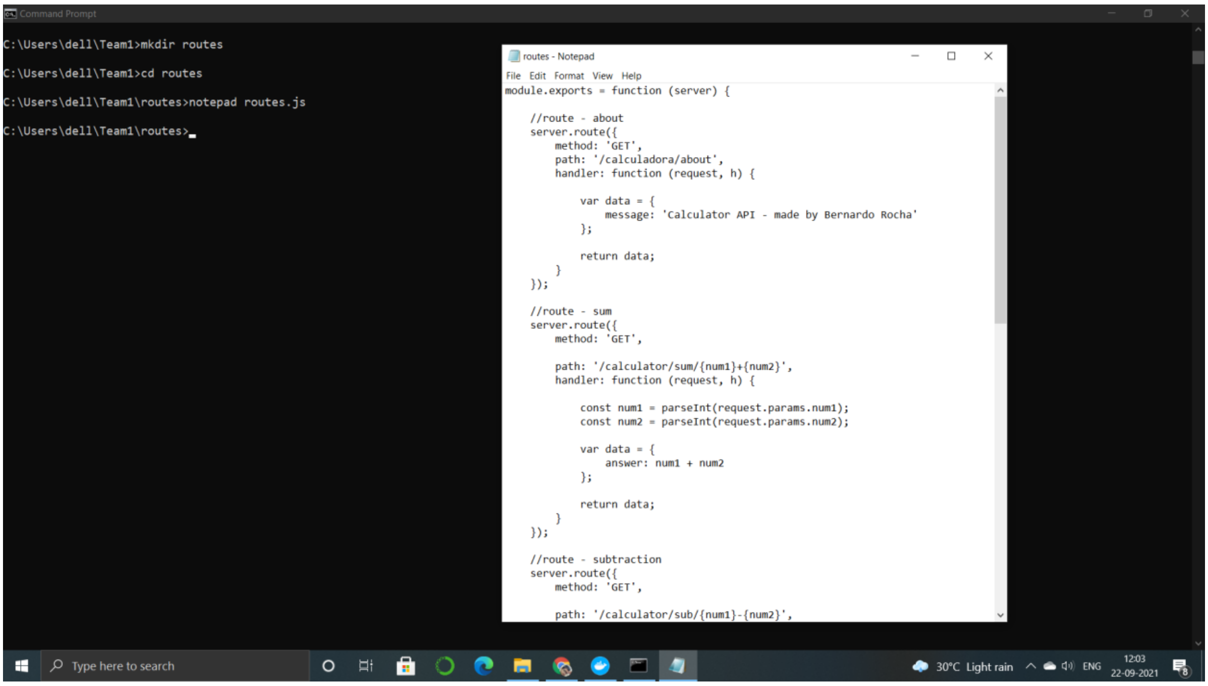
```
C:\Users\dell\Team1>kubectl describe services
Name:              kubernetes
Namespace:         default
Labels:            component=apiserver
                   provider=kubernetes
Annotations:       <none>
Selector:          <none>
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.0.1
IPs:               10.96.0.1
Port:              https  443/TCP
TargetPort:        8443/TCP
Endpoints:         192.168.49.2:8443
Session Affinity:  None
Events:            <none>


Name:                     nodejs-deployment
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=nodejs
Type:                     LoadBalancer
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.98.109.140
IPs:                      10.98.109.140
LoadBalancer Ingress:     192.168.49.2
Port:                     <unset>  3000/TCP
TargetPort:               3000/TCP
NodePort:                 <unset>  31285/TCP
Endpoints:                172.17.0.3:3000,172.17.0.4:3000
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>

C:\Users\dell\Team1>
```

Using the same methodology to deploy a calculator app using a Hapi server :



```
{
  "name": "team1",
  "version": "1.0.0",
  "description": "NodeJs with docker and kubernetes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Sumith Sai Budde",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1"
  },
  "devDependencies": {}
}


Is this OK? (yes) yes

C:\Users\dell\Team1>npm install hapi --save
npm WARN deprecated hapi@18.1.0: This version contains severe security issues and defects and sho
ub.com/hapijs/hapi/issues/4114)
npm WARN team1@1.0.0 No repository field.

+ hapi@18.1.0
added 29 packages from 3 contributors and audited 69 packages in 6.613s
found 5 high severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\dell\Team1>notepad index.js

C:\Users\dell\Team1>
```

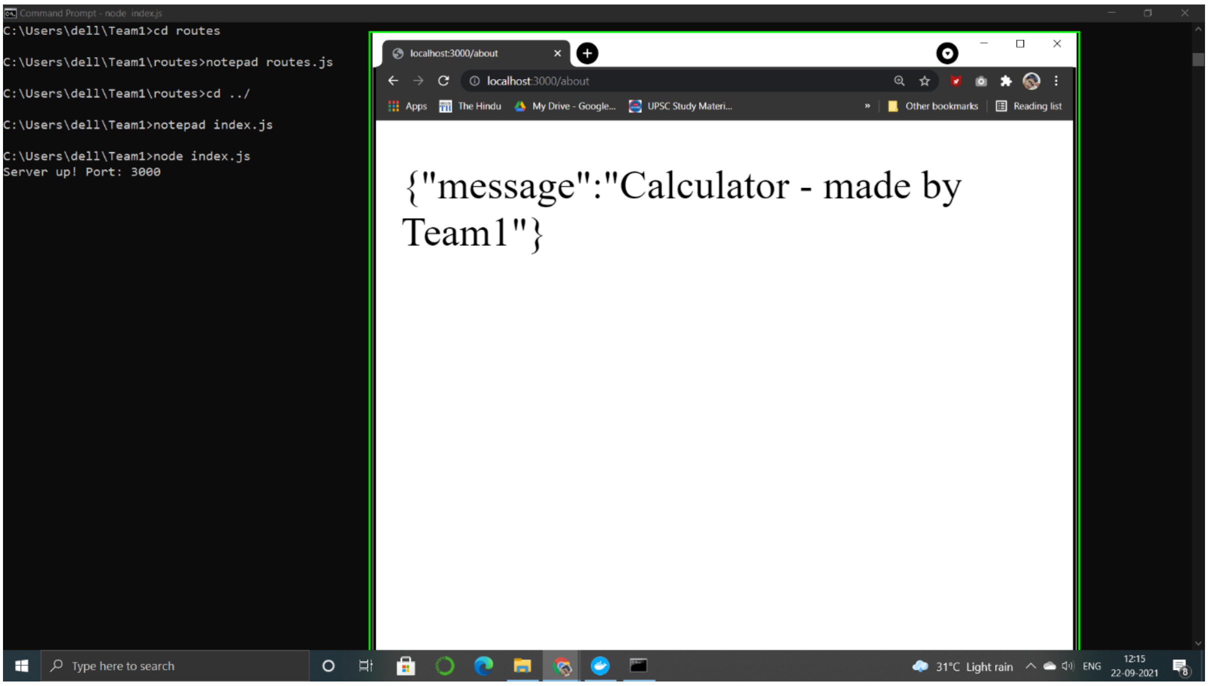
```
//Hapi framework
const Hapi = require('hapi');

//Address and port
const host = 'localhost';
const port = 3000;

//Create server
const server = Hapi.Server({
    host: host,
    port: port
});

//Routes
require('./routes/routes')(server);

//Start server
const init = async () => {
    await server.start();
    console.log("Server up! Port: " + port);
}

//Start App
init();
```



```
C:\Users\dell\Team1>mkdir routes

C:\Users\dell\Team1>cd routes

C:\Users\dell\Team1\routes>notepad routes.js

C:\Users\dell\Team1\routes>_
```

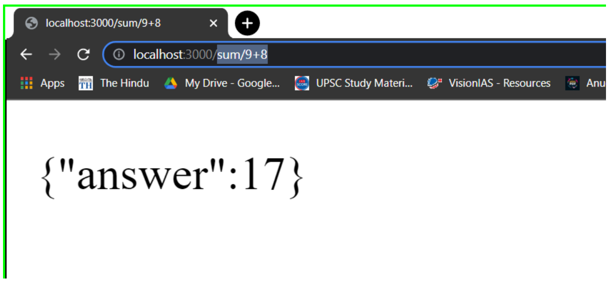
```
module.exports = function (server) {

    //route - about
    server.route({
        method: 'GET',
        path: '/calculadora/about',
        handler: function (request, h) {

            var data = {
                message: 'Calculator API - made by Bernardo Rocha'
            };

            return data;
        }
    });

    //route - sum
    server.route({
        method: 'GET',

        path: '/calculator/sum/{num1}+{num2}',
        handler: function (request, h) {

            const num1 = parseInt(request.params.num1);
            const num2 = parseInt(request.params.num2);

            var data = {
                answer: num1 + num2
            };

            return data;
        }
    });

    //route - subtraction
    server.route({
        method: 'GET',

        path: '/calculator/sub/{num1}-{num2}',
```

Calculator App :

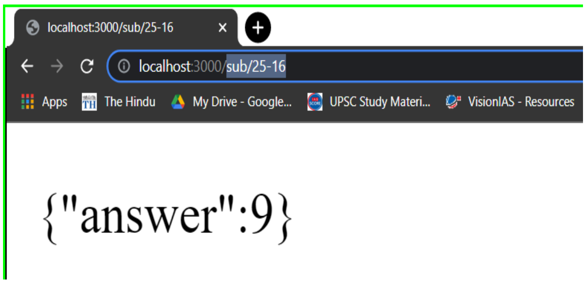


```
C:\Users\dell\Team1>cd routes

C:\Users\dell\Team1\routes>notepad routes.js

C:\Users\dell\Team1\routes>cd ../

C:\Users\dell\Team1>notepad index.js

C:\Users\dell\Team1>node index.js
Server up! Port: 3000
```

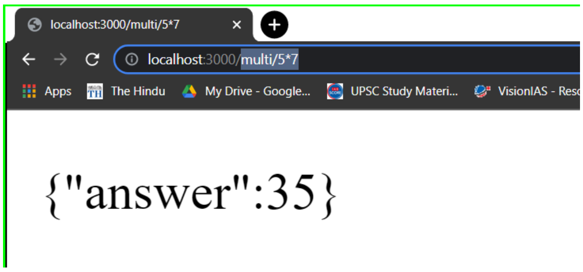
```
{"message":"Calculator - made by Team1"}
```



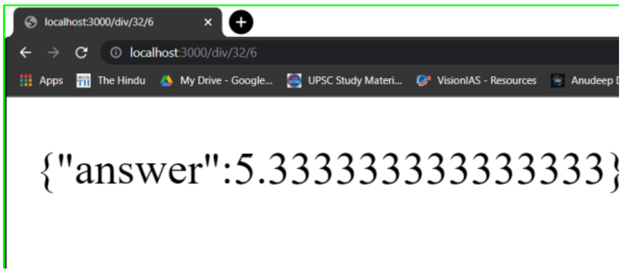
```
{"answer":17}
```



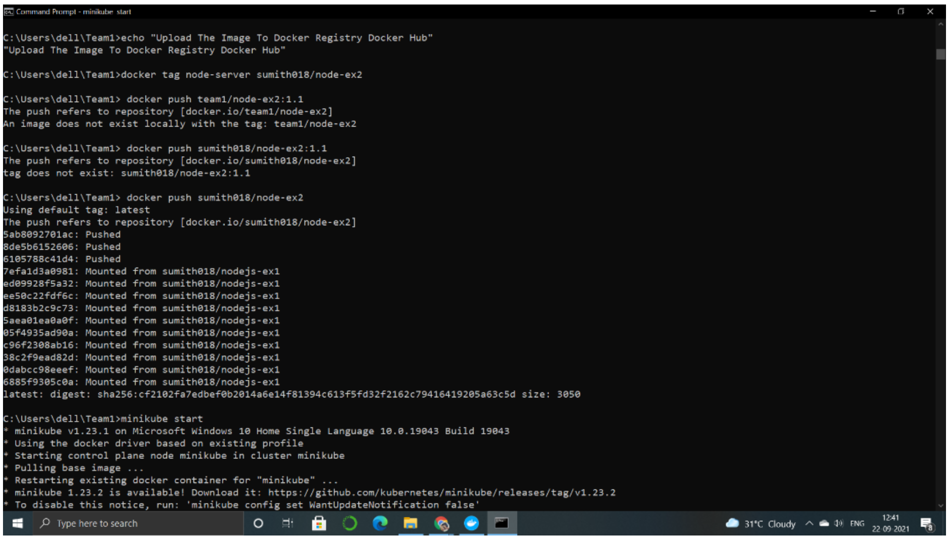
```
{"answer":9}
```



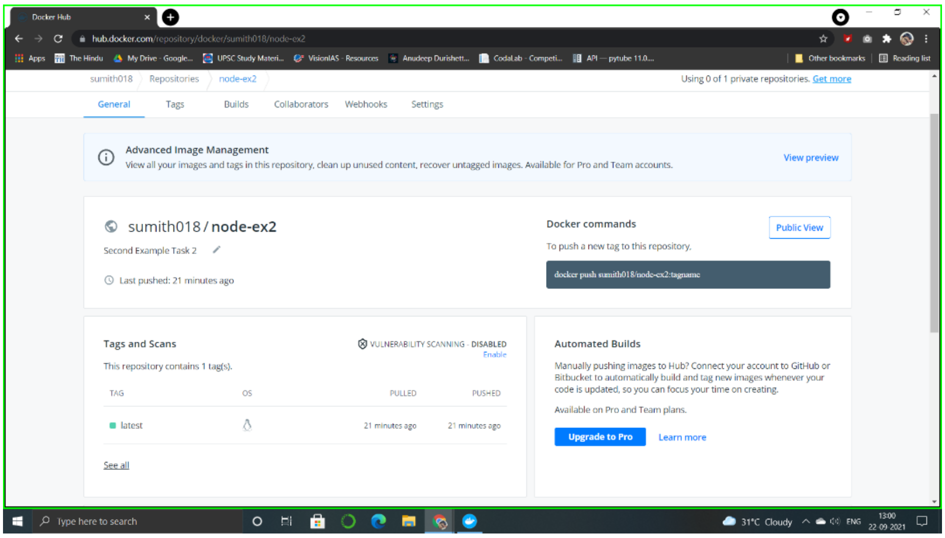
```
{"answer":35}
```



```
{"answer":5.333333333333333}
```



```
C:\Users\dell\Team1>echo "Upload The Image To Docker Registry Docker Hub"
"Upload The Image To Docker Registry Docker Hub"

C:\Users\dell\Team1>docker tag node-server sumith018/node-ex2

C:\Users\dell\Team1> docker push team1/node-ex2:1.1
The push refers to repository [docker.io/team1/node-ex2]
An image does not exist locally with the tag: team1/node-ex2

C:\Users\dell\Team1> docker push sumith018/node-ex2:1.1
The push refers to repository [docker.io/sumith018/node-ex2]
tag does not exist: sumith018/node-ex2:1.1

C:\Users\dell\Team1> docker push sumith018/node-ex2
Using default tag: latest
The push refers to repository [docker.io/sumith018/node-ex2]
5ab8092701ac: Pushed
8de5b6152606: Pushed
6105788c41d4: Pushed
7efa1d3a0981: Mounted from sumith018/nodejs-ex1
ed09928f5a32: Mounted from sumith018/nodejs-ex1
ee50c22fdf6c: Mounted from sumith018/nodejs-ex1
d8183b2c9c73: Mounted from sumith018/nodejs-ex1
5aea01ea0a0f: Mounted from sumith018/nodejs-ex1
05f4935ad90a: Mounted from sumith018/nodejs-ex1
c96f2308ab16: Mounted from sumith018/nodejs-ex1
38c2f9ead82d: Mounted from sumith018/nodejs-ex1
0dabcc98eeef: Mounted from sumith018/nodejs-ex1
6885f9305c0a: Mounted from sumith018/nodejs-ex1
latest: digest: sha256:cf2102fa7edbef0b2014a6e14f81394c613f5fd32f2162c79416419205a63c5d size: 3050

C:\Users\dell\Team1>minikube start
* minikube v1.23.1 on Microsoft Windows 10 Home Single Language 10.0.19043 Build 19043
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Restarting existing docker container for "minikube" ...
* minikube 1.23.2 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.23.2
* To disable this notice, run: 'minikube config set WantUpdateNotification false'
```



Using 0 of 1 private repositories. Get more

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.

sumith018 / node-ex2

Second Example Task 2

Last pushed: 21 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push sumith018/node-ex2:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

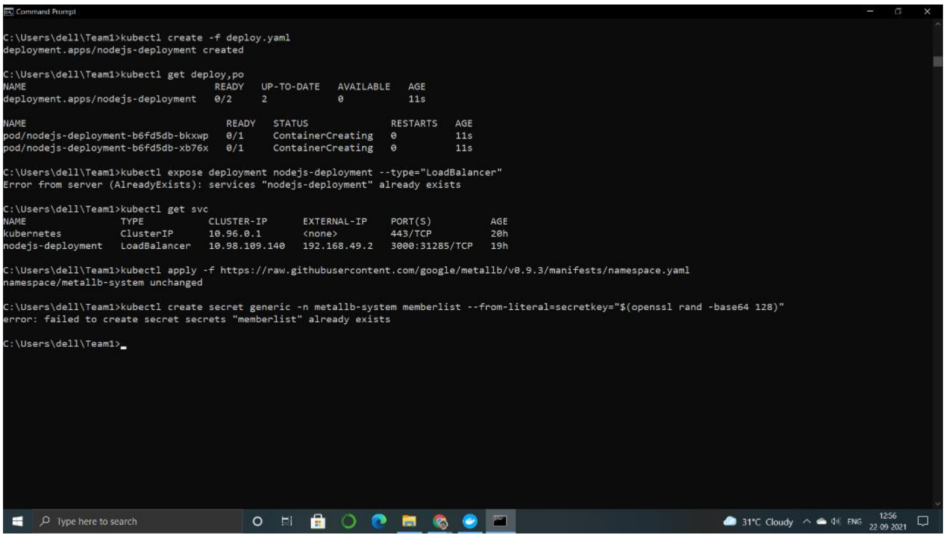
This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest	🐧	21 minutes ago	21 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available on Pro and Team plans.



```
C:\Users\dell\Team1>kubectl create -f deploy.yaml
deployment.apps/nodejs-deployment created

C:\Users\dell\Team1>kubectl get deploy,po
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-deployment   0/2     2            0           11s

NAME                                    READY   STATUS              RESTARTS   AGE
pod/nodejs-deployment-b6fd5db-bkxwp    0/1     ContainerCreating   0          11s
pod/nodejs-deployment-b6fd5db-xb76x    0/1     ContainerCreating   0          11s

C:\Users\dell\Team1>kubectl expose deployment nodejs-deployment --type="LoadBalancer"
Error from server (AlreadyExists): services "nodejs-deployment" already exists

C:\Users\dell\Team1>kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1       <none>         443/TCP          20h
nodejs-deployment   LoadBalancer   10.98.109.140   192.168.49.2   3000:31285/TCP   19h

C:\Users\dell\Team1>kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespace.yaml
namespace/metallb-system unchanged

C:\Users\dell\Team1>kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
error: failed to create secret secrets "memberlist" already exists

C:\Users\dell\Team1>_
```