INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DevOps & its Applications (CS457)

Assignment-1

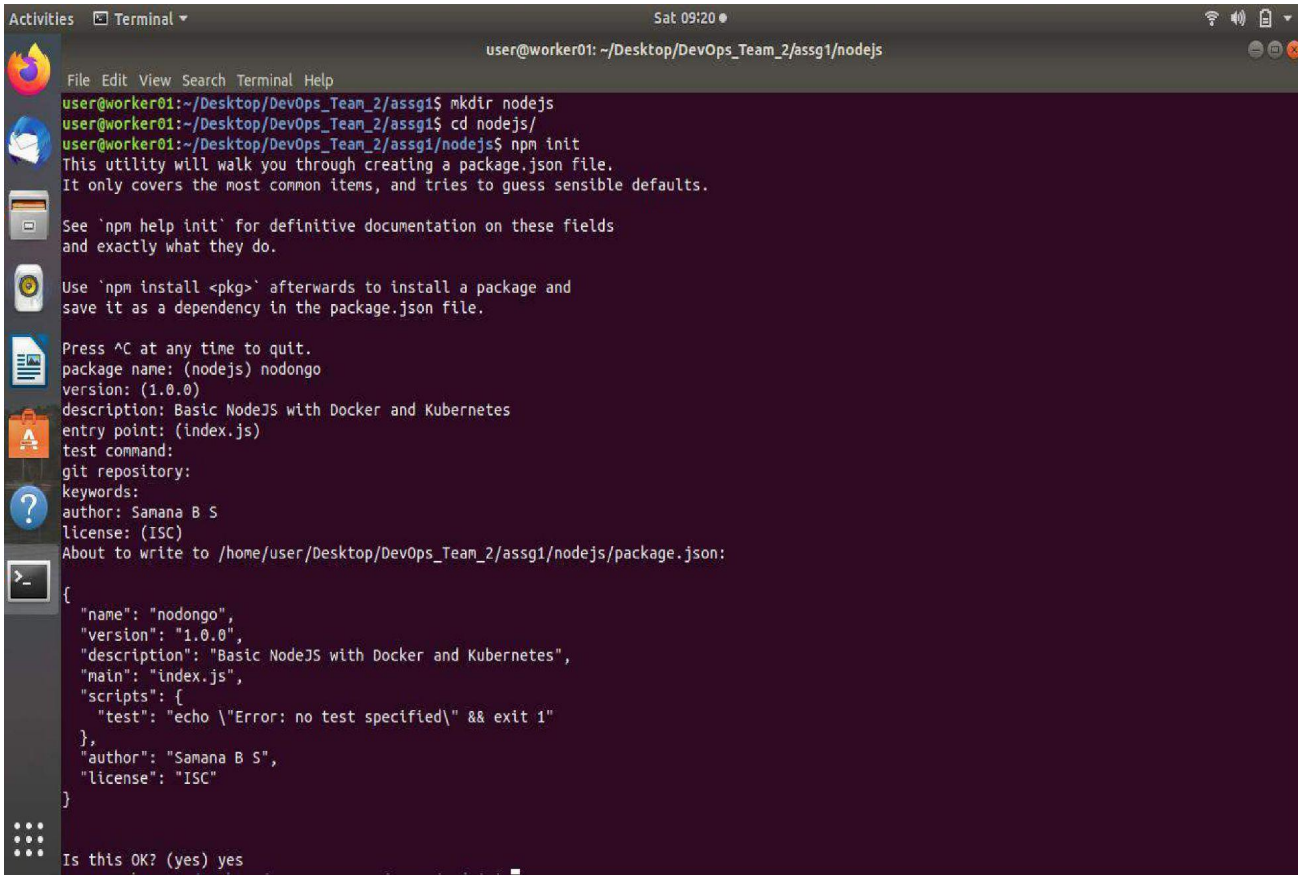
Under the guidance - **Dr. Uma S**

Submitted by:

Advay Aggarwal(18BCS002)
Meghana Hadimani(18BCS002)
Perumulla Tushar(18BCS065)
Rahul Priyadarshi(18BCS074)
Samana B S(18BCS088)

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

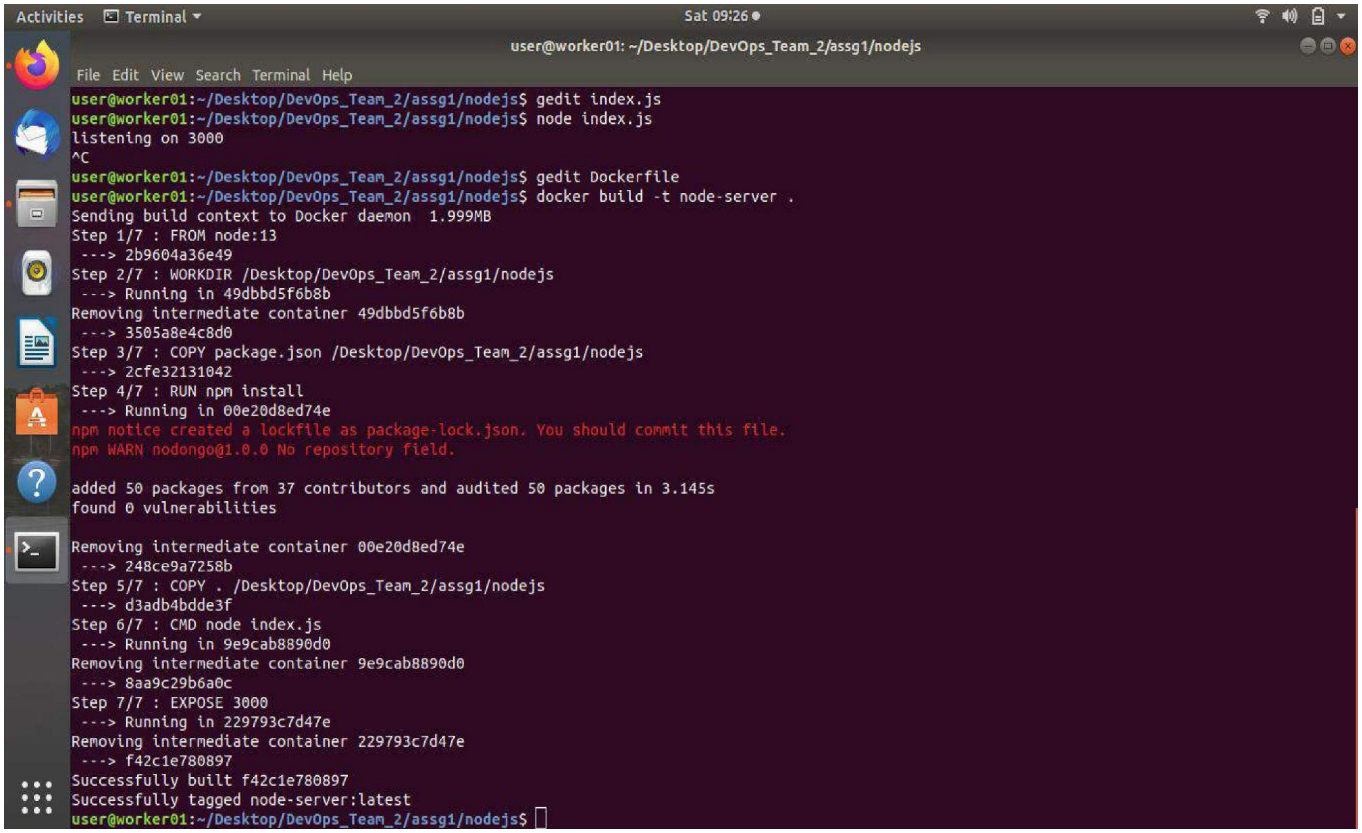
<u>Step1:</u> Making a separate directory and initializing The Node Application.



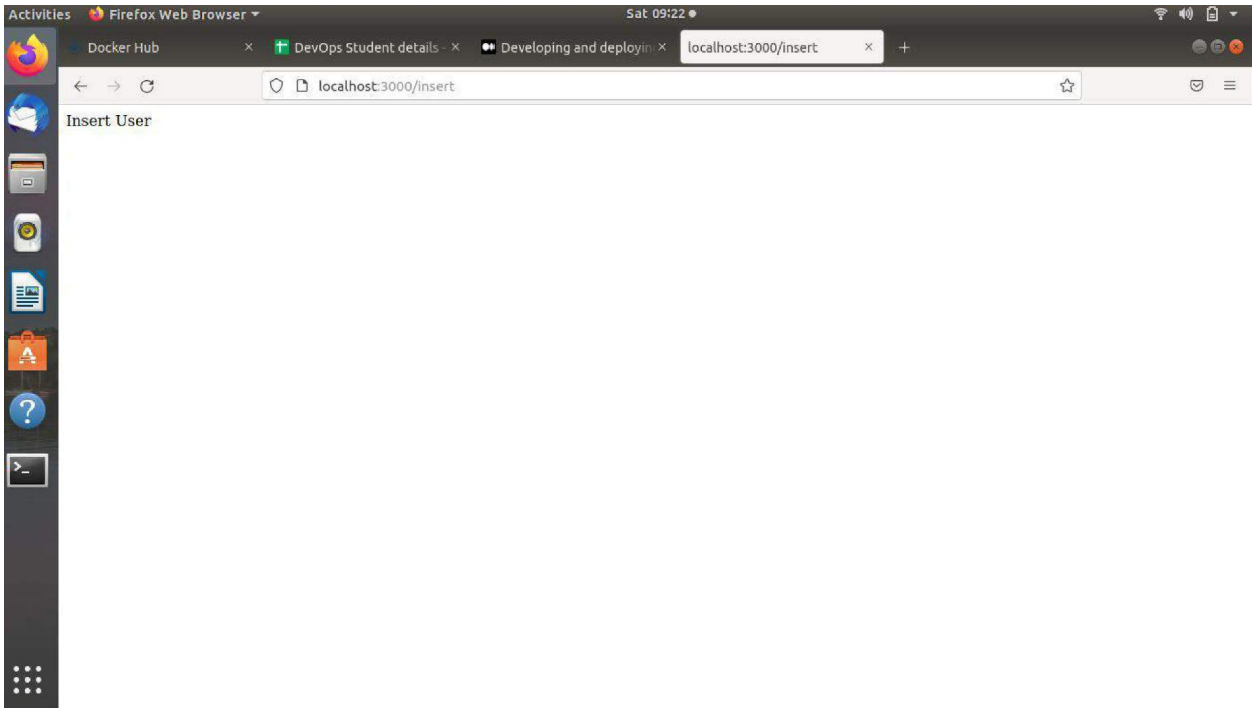
Step2: Installing Express

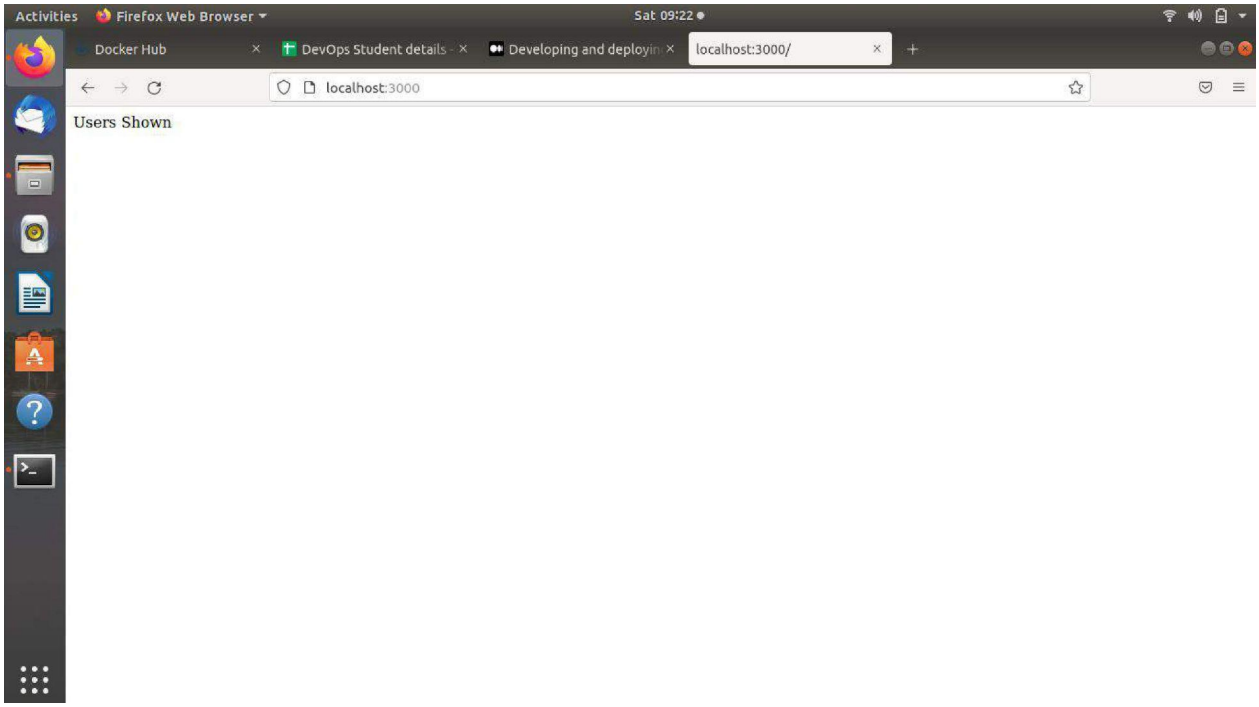
```
Activities  Terminal                                   Sat 09:22
user@worker01: ~/Desktop/DevOps_Team_2/assg1/nodejs
File  Edit  View  Search  Terminal  Help
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs) nodongo
version: (1.0.0)
description: Basic NodeJS with Docker and Kubernetes
entry point: (index.js)
test command:
git repository:
keywords:
author: Samana B S
license: (ISC)
About to write to /home/user/Desktop/DevOps_Team_2/assg1/nodejs/package.json:

{
  "name": "nodongo",
  "version": "1.0.0",
  "description": "Basic NodeJS with Docker and Kubernetes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Samana B S",
  "license": "ISC"
}


Is this OK? (yes) yes
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ npm install express --save

added 50 packages, and audited 51 packages in 2s

found 0 vulnerabilities
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ gedit index.js
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ node index.js
listening on 3000
```

Step3 and Step4: Making Index.js file and writing some code in it to test the application on the kubernetes cluster, then dockerizing the node server for which we create a Dockerfile to build the image as the server and the code is ready to deploy.



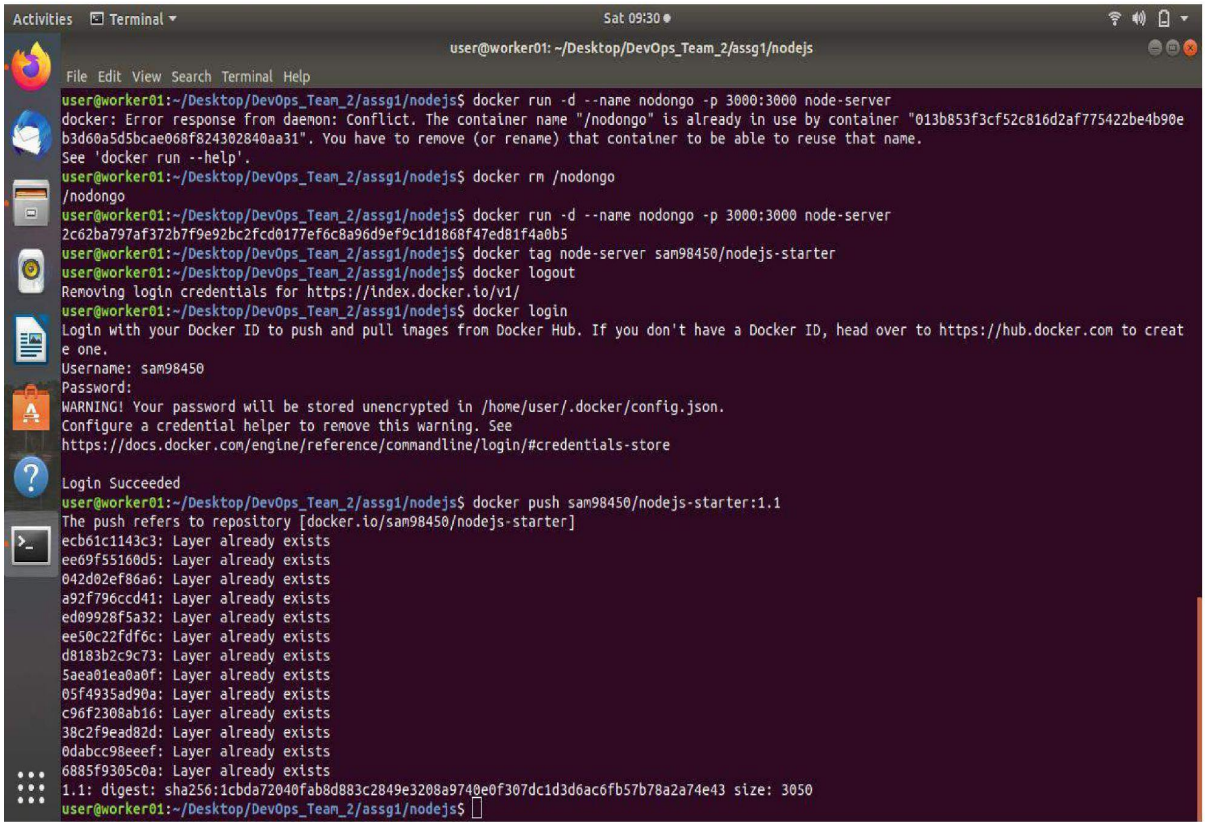
browsing localhost:3000/

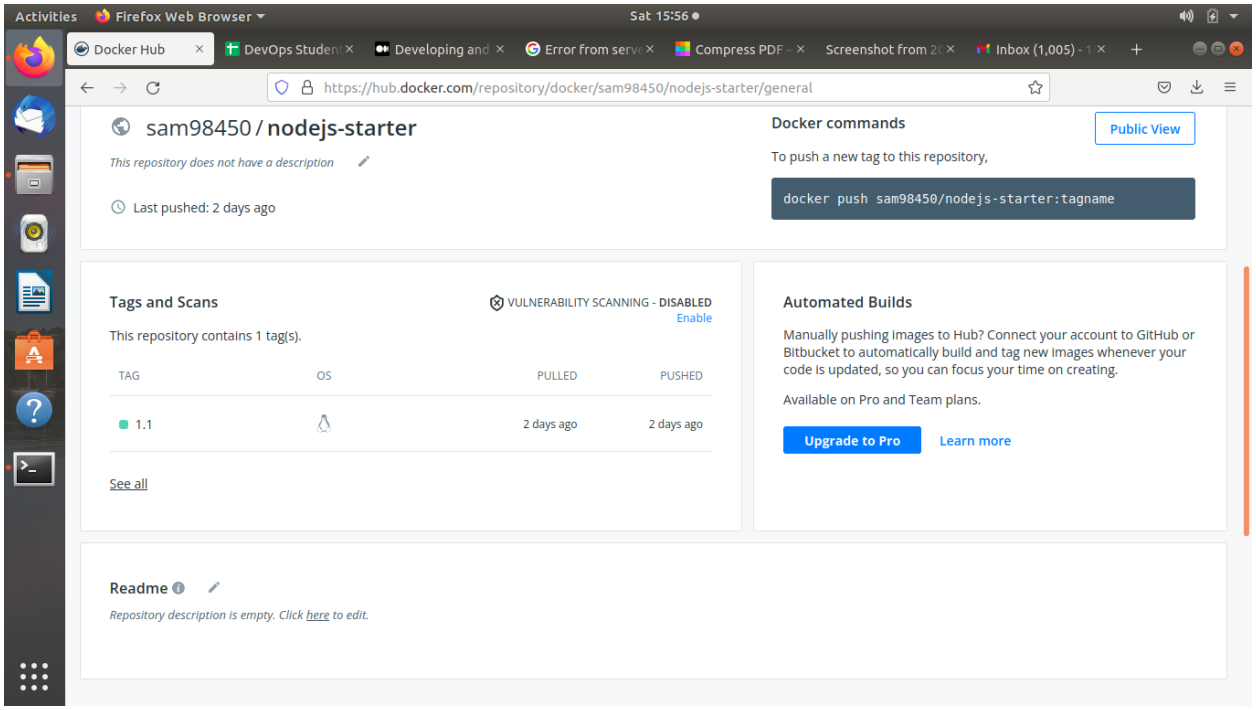




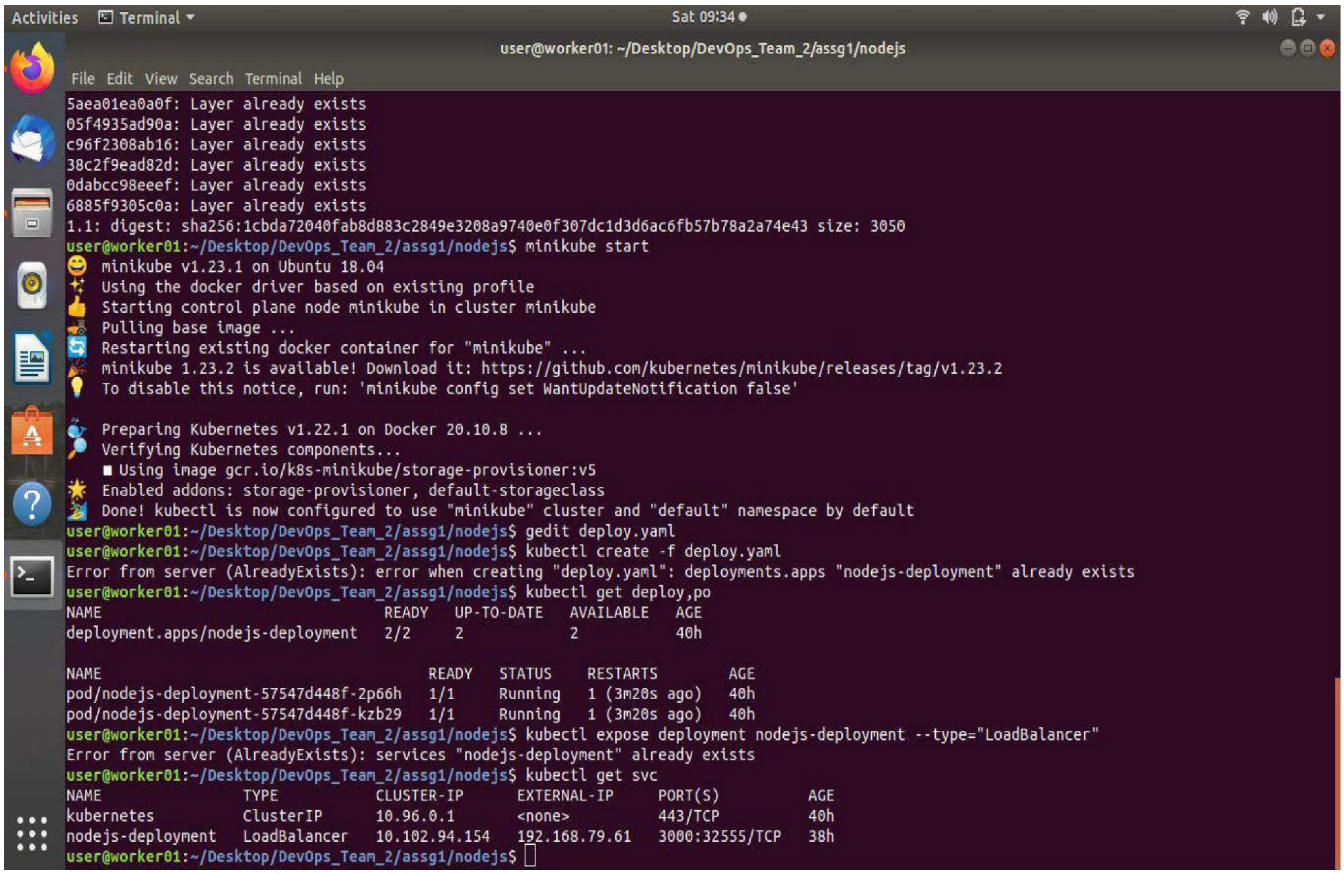
Step5 and step6: Now we create and run the container to ensure it works as intended and then we Upload the image to Docker registry Docker Hub

Creating a repo: we have named the repository as Nodejs-starter
We've tagged our existing docker image node-server to samana/nodejs-starter. After that we've pushed our docker image to the registry by using a docker push and tagged it with the 1.1 version.

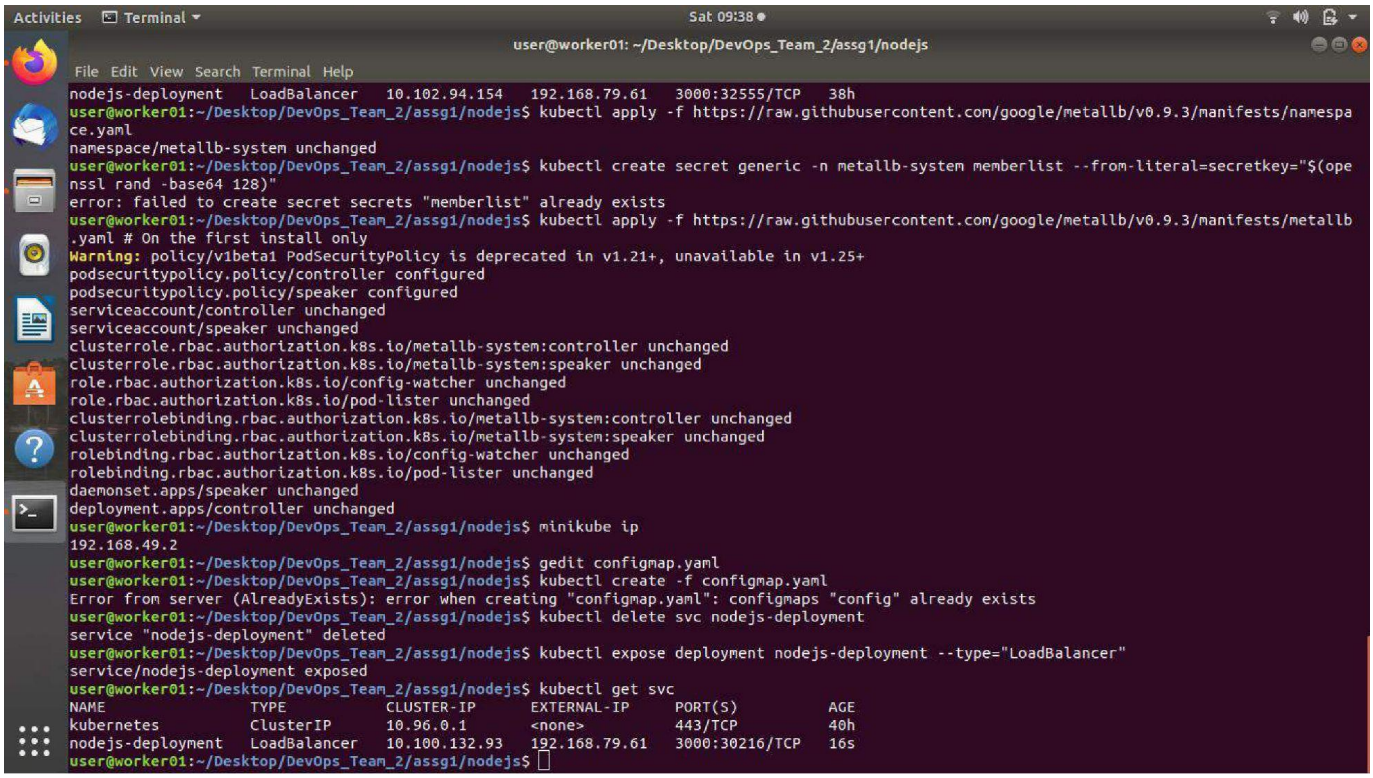



Activities
Firefox Web Browser
Sat 15:56
Docker Hub
DevOps Studen
Developing and
Error from serve
Compress PDF
Screenshot from 2
Inbox (1,005) - 1
https://hub.docker.com/repository/docker/sam98450/nodejs-starter/general
sam98450 / nodejs-starter
This repository does not have a description
Last pushed: 2 days ago
Docker commands
Public View
To push a new tag to this repository,
docker push sam98450/nodejs-starter:tagname
Tags and Scans
VULNERABILITY SCANNING - DISABLED
Enable
This repository contains 1 tag(s).
TAG
OS
PULLED
PUSHED
1.1
2 days ago
2 days ago
See all
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.
Upgrade to Pro
Learn more
Readme
Repository description is empty. Click here to edit.

Step7,Step8 and Step9: Start the kubernetes cluster, Defining a YAML file to create a Deployment in kubernetes cluster and As we've created the YAML file, we can go ahead and create a deployment from this YAML file.

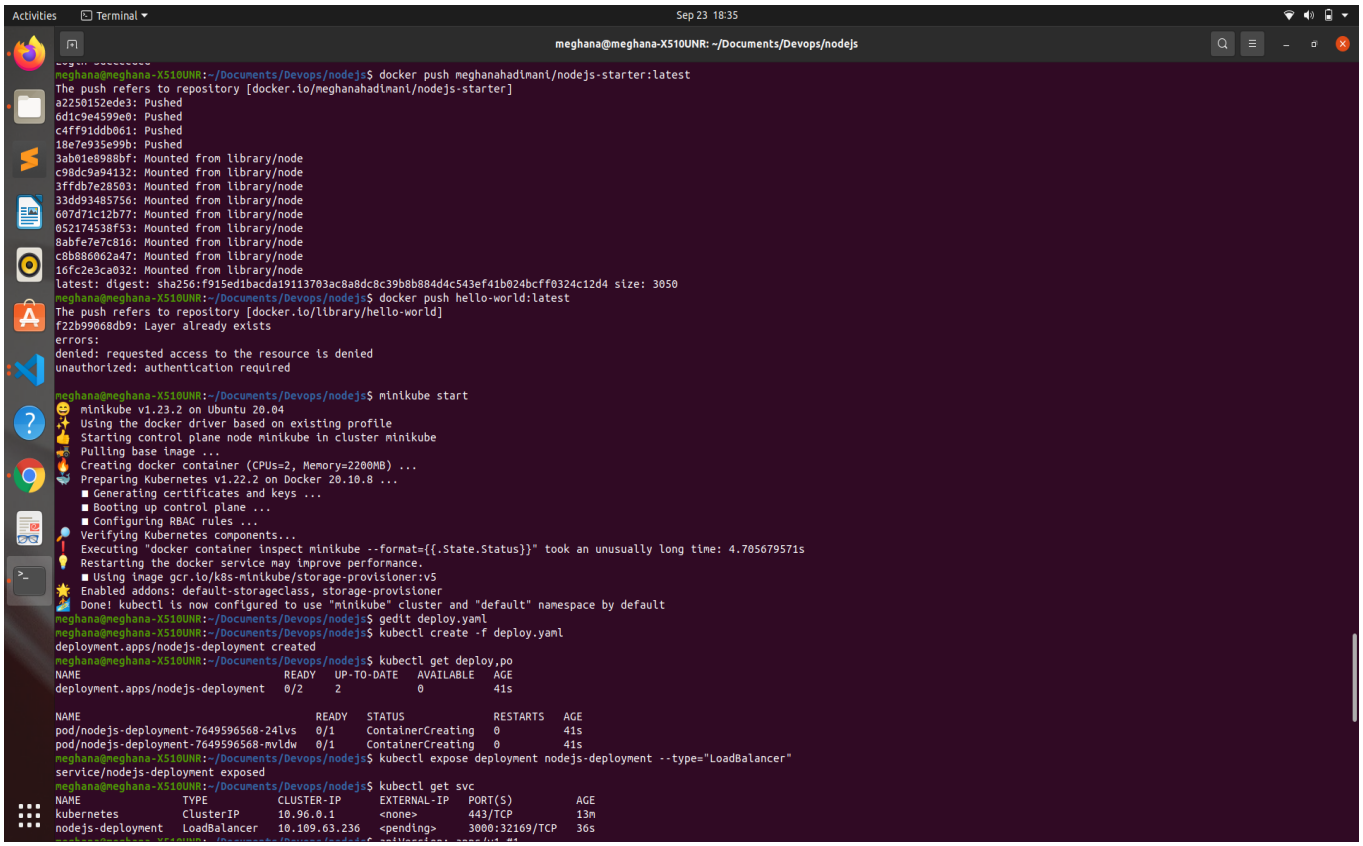


Step10: Now the next step is to expose the Deployment to the internet. **Kubectl expose** is used to expose Deployment named nodejs-deployment of the type Load Balancer.



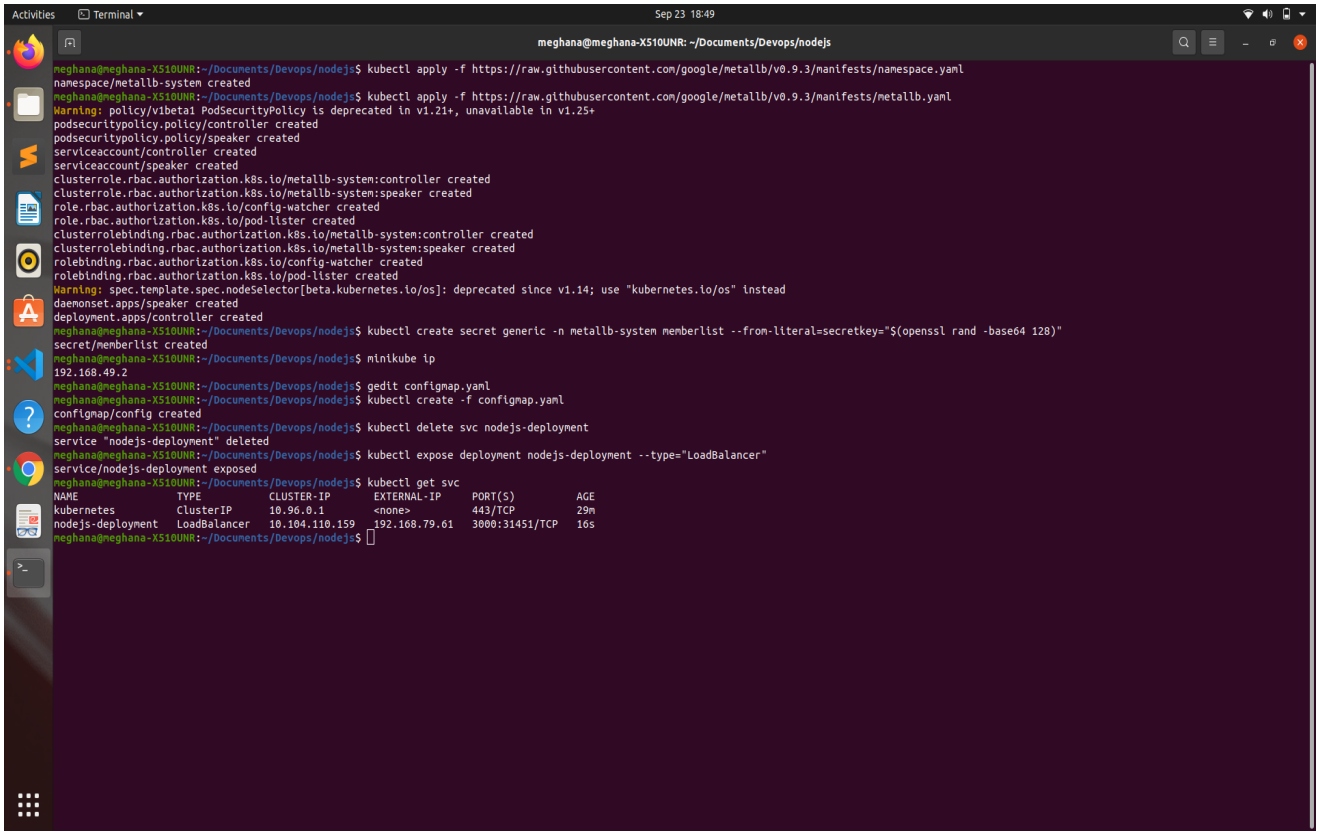
```
nodejs-deployment   LoadBalancer   10.102.94.154   192.168.79.61   3000:32555/TCP   38h
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespa
ce.yaml
namespace/metallb-system unchanged
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(ope
nssl rand -base64 128)"
error: failed to create secret secrets "memberlist" already exists
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/metallb
.yaml # On the first install only
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller configured
podsecuritypolicy.policy/speaker configured
serviceaccount/controller unchanged
serviceaccount/speaker unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker unchanged
role.rbac.authorization.k8s.io/config-watcher unchanged
role.rbac.authorization.k8s.io/pod-lister unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker unchanged
rolebinding.rbac.authorization.k8s.io/config-watcher unchanged
rolebinding.rbac.authorization.k8s.io/pod-lister unchanged
daemonset.apps/speaker unchanged
deployment.apps/controller unchanged
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ minikube ip
192.168.49.2
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ gedit configmap.yaml
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl create -f configmap.yaml
Error from server (AlreadyExists): error when creating "configmap.yaml": configmaps "config" already exists
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1       <none>          443/TCP          40h
nodejs-deployment   LoadBalancer   10.100.132.93   192.168.79.61   3000:30216/TCP   16s
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$
```

Here when we had done it for the first time we weren’t getting the External-IP



```
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ docker push meghanahadimani/nodejs-starter:latest
The push refers to repository [docker.io/meghanahadimani/nodejs-starter]
a2250152ede3: Pushed
6d1c9e4599e0: Pushed
c4ff91ddb061: Pushed
18e7e935e99b: Pushed
3ab01e8988bf: Mounted from library/node
c98dc9a94132: Mounted from library/node
3ffdb7e28503: Mounted from library/node
33dd93485756: Mounted from library/node
607d71c12b77: Mounted from library/node
052174538f53: Mounted from library/node
8abfe7e7c816: Mounted from library/node
c8b886062a47: Mounted from library/node
16fc2e3ca032: Mounted from library/node
latest: digest: sha256:f915ed1bacda19113703ac8a8dc8c39b8b884d4c543ef41b024bcff0324c12d4 size: 3050
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ docker push hello-world:latest
The push refers to repository [docker.io/library/hello-world]
f22b99068db9: Layer already exists
errors:
denied: requested access to the resource is denied
unauthorized: authentication required

meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ minikube start
😄  minikube v1.23.2 on Ubuntu 20.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.22.2 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: 4.705679571s
💡  Restarting the docker service may improve performance.
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ gedit deploy.yaml
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ kubectl create -f deploy.yaml
deployment.apps/nodejs-deployment created
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ kubectl get deploy,po
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-deployment   0/2     2            0           41s

NAME                                    READY   STATUS              RESTARTS   AGE
pod/nodejs-deployment-7649596568-24lvs   0/1     ContainerCreating   0          41s
pod/nodejs-deployment-7649596568-mvldw   0/1     ContainerCreating   0          41s
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
meghana@meghana-X510UNR:~/Documents/Devops/nodejs$ 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   10.109.63.236   <pending>     3000:32169/TCP   36s
```

Step11: As we are using minikube we'll notice that we won't get an external IP because the load balancer will not work on minikube. So we run the minikube ip and after we get it we'll create a config map for the address pool. After that we'll create a config map in the metallb-system namespace. Next, we have to delete the svc and create the service again. Now that's done, we'll be getting External IP.



```
Activities  Terminal  Sat 09:47
user@worker01: ~/Desktop/DevOps_Team_2/assg1/nodejs
File  Edit  View  Search  Terminal  Help
nodejs-deployment   LoadBalancer   10.102.94.154   192.168.79.61   3000:32555/TCP   38h
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespa
ce.yaml
namespace/metallb-system unchanged
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(ope
nssl rand -base64 128)"
error: failed to create secret secrets "memberlist" already exists
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/metallb
.yaml # On the first install only
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller configured
podsecuritypolicy.policy/speaker configured
serviceaccount/controller unchanged
serviceaccount/speaker unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker unchanged
role.rbac.authorization.k8s.io/config-watcher unchanged
role.rbac.authorization.k8s.io/pod-lister unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker unchanged
rolebinding.rbac.authorization.k8s.io/config-watcher unchanged
rolebinding.rbac.authorization.k8s.io/pod-lister unchanged
daemonset.apps/speaker unchanged
deployment.apps/controller unchanged
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ minikube ip
192.168.49.2
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ gedit configmap.yaml
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl create -f configmap.yaml
Error from server (AlreadyExists): error when creating "configmap.yaml": configmaps "config" already exists
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$ kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1       <none>          443/TCP          40h
nodejs-deployment   LoadBalancer   10.100.132.93   192.168.79.61   3000:30216/TCP   16s
user@worker01:~/Desktop/DevOps_Team_2/assg1/nodejs$
```