



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

Devops Jenkins Assignment

Submitted to:

Dr.Uma S

Submitted by:

Advay Agarwal (18BCS002)

Perumalla Tushar(18BCS065)

Meghana Hadimani(18BCS052)

Samana B S(18BCS088)

Rahul Pryadarshini(18BCS074)

Setting up CI/CD Jenkins pipeline for kubernetes

Tools and Technologies used :

- Github
- Docker and Docker hub
- Jenkins
- Kubernetes Cluster

Prerequisites:

- NodeJS v8+
- Docker installed locally
- minikube

Step - 1

1. Create a folder and initialize a package.json file by running the `npm init` command. Then install the express module by running the `npm i express --save`
2. Then create a file called server.js
3. Running the command `node server.js` should give the output in the browser at localhost:PORT

Step - 2

1. Create a dockerfile and put the following code in it to build a docker image.

```
FROM node:15
WORKDIR /home/meghnahadimani/Desktop
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 8000
CMD [ "node", "server.js" ]
```

Where, 15 is the version of Node, WORKDIR tells the working directory of the image, RUN executes the command defined above.

EXPOSE informs the container that it needs to open port 8000. CMD stands for command.

2. Create a `.dockerignore` file and put `node_modules` in it.
3. Build a docker image. Run the following command in the project directory. `docker build -t <docker username>/nodejs-app`

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with `server.js`, `Dockerfile`, and `.dockerignore` files in a `DEVOPS` folder.
- Terminal:** Displays the command `docker build` being run, showing the build process and logs. The logs indicate the creation of intermediate containers, pulling of dependencies, and finally exposing port 8000.
- Status Bar:** Shows the date (Nov 14 16:03), terminal type (bash), and other settings like spaces and encoding.

```
Nov 14 16:03 • .dockerignore - Devops - Visual Studio Code
mehgana@mehgana-X510UNR:~/Documents/Devops$ docker build .
Sending build context to Docker daemon 69.63kB
Step 1/7 : FROM node:15
15: Pulling from library/node
bfde2ec33fbc: Pull complete
78775e2f1047: Pull complete
7b6173a10eb8: Pull complete
dc05be471d51: Pull complete
55fab5cadd3c: Pull complete
bd821d20ef8c: Pull complete
6641b69671c6: Pull complete
989c5d2d2313: Pull complete
4b57d41e8391: Pull complete
Digest: sha256:608bba799613b1ebf754034ae008849ba51e88b23271412427b76d60ae0d0627
Status: Downloaded newer image for node:15
--> 3d3f41722dfa
Step 2/7 : WORKDIR /home/mehgana/Documents/Devops/
--> Running in efcb37ca2b83
Removing intermediate container efcb37ca2b83
--> caeb42bbe067
Step 3/7 : COPY package*.json .
--> ae4958dfa58
Step 4/7 : RUN npm install
--> Running in a823bb69bf1a
added 50 packages, and audited 51 packages in 2s

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 7.7.6 -> 8.1.3
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v8.1.3>
npm notice Run `npm install -g npm@8.1.3` to update!
npm notice
Removing intermediate container a823bb69bf1a
--> e4650748f0ec
Step 5/7 : COPY .
--> f36ce9810ff69
Step 6/7 : EXPOSE 8000
--> Running in 7d472883b8be

Ln 1, Col 14 Spaces: 4 UTF-8 LF Ignore
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the same project structure as the previous screenshot.
- Terminal:** Displays the command `docker build` running, followed by the completion message "Successfully built 19dbc7ab75a".
- Status Bar:** Shows the date (Nov 14 16:03), terminal type (bash), and other settings like spaces and encoding.

```
Nov 14 16:03 • .dockerignore - Devops - Visual Studio Code
mehgana@mehgana-X510UNR:~/Documents/Devops$ docker build .
Sending build context to Docker daemon 69.63kB
Step 1/7 : FROM node:15
15: Pulling from library/node
bfde2ec33fbc: Pull complete
78775e2f1047: Pull complete
7b6173a10eb8: Pull complete
dc05be471d51: Pull complete
55fab5cadd3c: Pull complete
bd821d20ef8c: Pull complete
6641b69671c6: Pull complete
989c5d2d2313: Pull complete
4b57d41e8391: Pull complete
Digest: sha256:608bba799613b1ebf754034ae008849ba51e88b23271412427b76d60ae0d0627
Status: Downloaded newer image for node:15
--> 3d3f41722dfa
Step 2/7 : WORKDIR /home/mehgana/Documents/Devops/
--> Running in efcb37ca2b83
Removing intermediate container efcb37ca2b83
--> caeb42bbe067
Step 3/7 : COPY package*.json .
--> ae4958dfa58
Step 4/7 : RUN npm install
--> Running in a823bb69bf1a
added 50 packages, and audited 51 packages in 2s

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 7.7.6 -> 8.1.3
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v8.1.3>
npm notice Run `npm install -g npm@8.1.3` to update!
npm notice
Removing intermediate container a823bb69bf1a
--> e4650748f0ec
Step 5/7 : COPY .
--> f36ce9810ff69
Step 6/7 : EXPOSE 8000
--> Running in 7d472883b8be
Removing intermediate container 7d472883b8be
--> 3714cb57ef91
Step 7/7 : CMD [ "node", "server.js" ]
--> Running in 5cb8bb037bee
Removing intermediate container 5cb8bb037bee
--> 19dbc7ab75a
Successfully built 19dbc7ab75a
mehgana@mehgana-X510UNR:~/Documents/Devops$
```

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal output shows the build process of a Docker image:

```
meghana@meghana-X510UNR:~/Documents/Devops$ docker images
REPOSITORY          TAG      IMAGE ID   CREATED     SIZE
<none>              <none>   19cdcb7ab75a  2 minutes ago  940MB
gcr.io/k8s-minikube/kicbase   v0.0.28  e2a6c047bedd  6 weeks ago  1.08GB
hello-world          latest   feb5d9fea6a5  7 weeks ago  13.3kB
node                15      3d3f41722daf  6 months ago  936MB
meghana@meghana-X510UNR:~/Documents/Devops$ docker tag none:none nodejs-app:latest
Error response from daemon: No such image: none:none
meghana@meghana-X510UNR:~/Documents/Devops$ docker build -t meghanahadimani/nodejs-app .
Sending build context to Docker daemon 69.63B
Step 1/7 : FROM node:15
--> 3d3f41722daf
Step 2/7 : WORKDIR /home/meghana/Documents/Devops/
--> Using cache
--> caeb42bbe067
Step 3/7 : COPY package*.json .
--> Using cache
--> ae4958dfa58
Step 4/7 : RUN npm install
--> Using cache
--> e4650748f0ec
Step 5/7 : COPY .
--> Using cache
--> f36ce9910f69
Step 6/7 : EXPOSE 8000
--> Using cache
--> 3714cb57e9f1
Step 7/7 : CMD [ "node", "server.js" ]
--> Using cache
--> 19cdcb7ab75a
Successfully built 19cdcb7ab75a
Successfully tagged meghanahadimani/nodejs-app:latest
meghana@meghana-X510UNR:~/Documents/Devops$
```

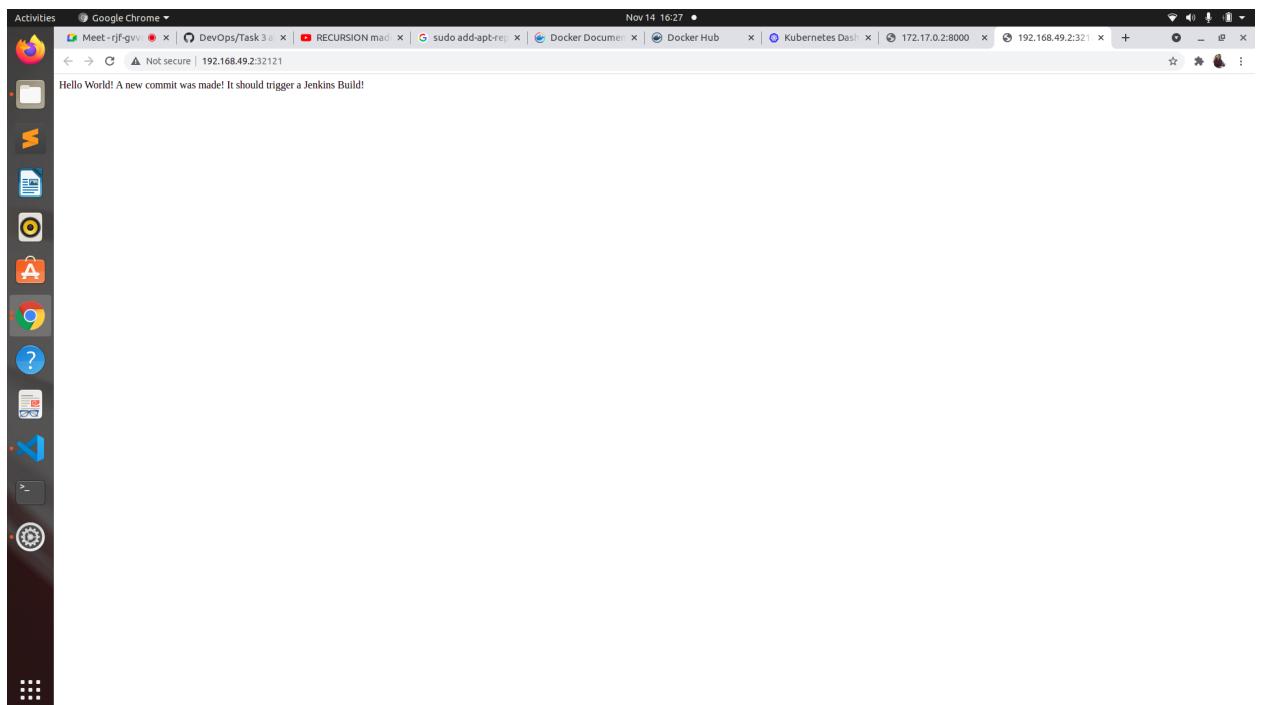
Now check the image using `docker images` command

- Run the image. Running the image with `-d` flag leaves the container running in the background and the `-p` flag redirects a public port to a private port in the container. `docker run -p 3000:8000 -d <docker username>/nodejs-app`

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal output shows the Docker image running:

```
meghana@meghana-X510UNR:~/Documents/Devops$ docker images
REPOSITORY          TAG      IMAGE ID   CREATED     SIZE
meghanahadimani/nodejs-app  latest   19cdcb7ab75a  6 minutes ago  940MB
gcr.io/k8s-minikube/kicbase   v0.0.28  e2a6c047bedd  6 weeks ago  1.08GB
hello-world          latest   feb5d9fea6a5  7 weeks ago  13.3kB
node                15      3d3f41722daf  6 months ago  936MB
meghana@meghana-X510UNR:~/Documents/Devops$ docker run -p 3000:8000 -d meghanahadimani/nodejs-app
59810362ea9648aff23fbdfe14ab95a4727f78d265825a3c9baef9832e
meghana@meghana-X510UNR:~/Documents/Devops$
```

To check if it is running, go to the browser and type URL : <docker_ip_address:8000> and you should see the output.



5. Login to docker hub through the command line and push to the docker hub registry. `docker push <username>/nodejs-app`

A screenshot of Visual Studio Code. The left sidebar shows a project structure with files like Dockerfile, package-lock.json, package.json, and server.js. The Dockerfile contains the following code:

```
FROM node:15
WORKDIR /home/meghana/Documents/Devops/
COPY package*.json .
RUN npm install
COPY .
EXPOSE 8000
CMD [ "node", "server.js" ]
```

The right side shows a terminal window with the following command and output:

```
meghana@meghana-X510UNR:~/Documents/Devops$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
59810362ea96 meghanahadimani/nodejs-app "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 0.0.0.0:3000->8000/tcp, ::1:3000->8000/tcp
4aac387107a4 gcr.io/k8s-minikube/kicbase:v0.0.28 "/usr/local/bin/entr..." 33 minutes ago Up 33 minutes 127.0.0.1:49157->22/tcp, 127.0.0.1:49156->2376/tcp, 127.0.0.1:49155->5000/tcp, 127.0.0.1:49154->8443/tcp, 127.0.0.1:49153->32443/tcp minikube
meghana@meghana-X510UNR:~/Documents/Devops$ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 59810362ea96
Template parsing error: template: :1:23: executing "" at <.NetworkSettings.Networks>: map has no entry for key "NetworkSettings"
meghana@meghana-X510UNR:~/Documents/Devops$ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 59810362ea96
Error: No such object: 59810362ea96
meghana@meghana-X510UNR:~/Documents/Devops$ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 172.17.0.2
meghana@meghana-X510UNR:~/Documents/Devops$ docker login -u meghanahadimani
Password:
WARNING! Your password will be stored unencrypted in /home/meghana/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
meghana@meghana-X510UNR:~/Documents/Devops$
```

At the bottom of the terminal, status information is shown: Ln 7, Col 28, Spaces:4, UFT-8, LF, Dockerfile.

The screenshot shows the Visual Studio Code interface. In the Explorer sidebar, there is a folder named 'DEVOPS' containing 'node_modules', '.dockerignore', 'Dockerfile', 'package-lock.json', 'package.json', and 'server.js'. The 'Dockerfile' tab is selected, displaying the following code:

```

FROM node:15
WORKDIR /home/meghana/Documents/Devops/
COPY package*.json .
RUN npm install
COPY .
EXPOSE 8000
CMD [ "node", "server.js" ]

```

The terminal tab shows the following command history and output:

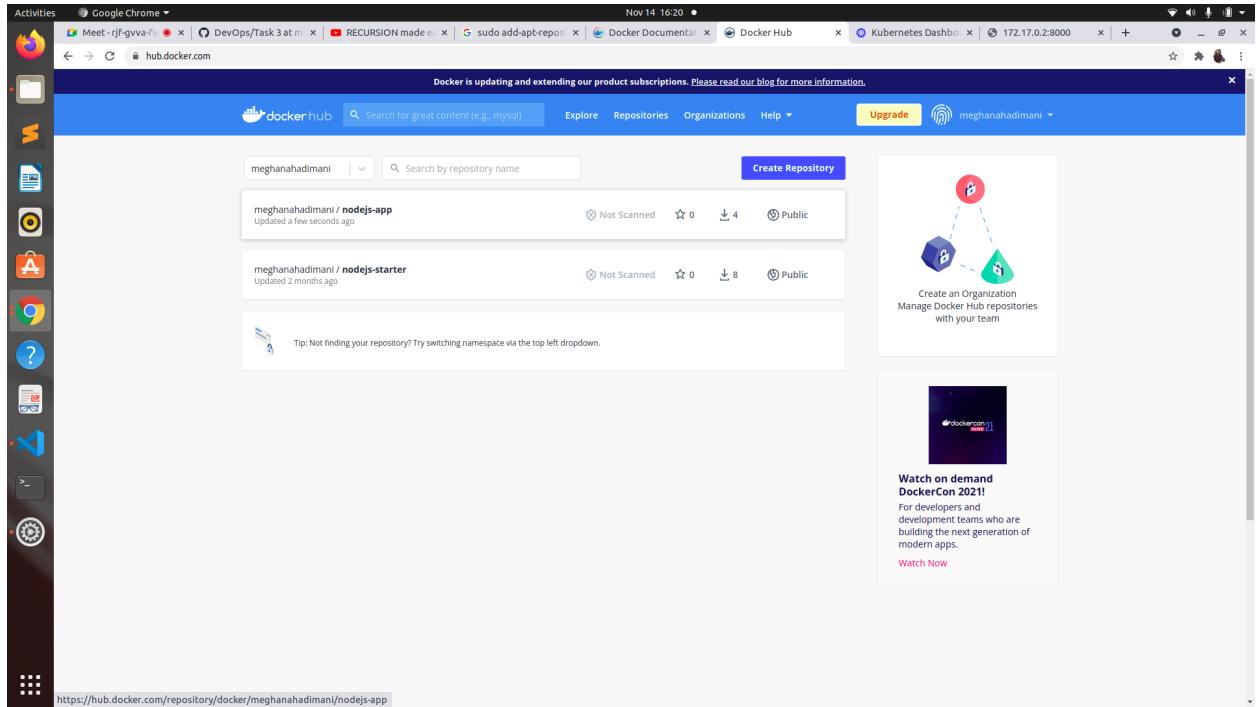
```

Template parsing error: template: :1:23: executing "" at <.NetworkSettings.Networks>: map has no entry for key "NetworkSettings"
meghana@meghana-X510UNR:~/Documents/Devops$ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 58910362ea96
Error: No such object: 58910362ea96
meghana@meghana-X510UNR:~/Documents/Devops$ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 59810362ea96
172.17.0.2
meghana@meghana-X510UNR:~/Documents/Devops$ docker login -u meghanahadimani
Password:
WARNING! Your password will be stored unencrypted in /home/meghana/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
meghana@meghana-X510UNR:~/Documents/Devops$ docker push meghanahadimani/nodejs-app
Using default tag: latest
The push refers to repository [docker.io/meghanahadimani/nodejs-app]
59810362ea96: Pushed
e659d3ef956: Pushed
81f1c2097269: Pushed
76bd9d9cbe5: Pushed
f92723791559: Mounted from library/node
f08dc1cda81: Mounted from library/node
4a9e68168053: Mounted from library/node
b257e69d416f: Mounted from library/node
1e9c28d06610: Mounted from library/node
cd8b98d77163: Mounted from library/node
ed0a3d9cbc7: Mounted from library/node
8c8e652ecd8f: Mounted from library/node
2f4eefa2e1b5: Mounted from library/node
latest: digest: sha256:52e744e9047149207466b686d554f46e0c8e3ef3d0d8ab5cacf09513012080f size: 3050
meghana@meghana-X510UNR:~/Documents/Devops$ 

```

At the bottom right of the terminal, it says 'Ln 7, Col 28 Spaces: 4 UTF-8 LF Dockerfile R Q'.



This pushes to the registry and attaches the latest tag by default.

Step - 3:

1. Create a deployment file called deployment.yaml and add the following code in it.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-deployment
  labels:
    app: helloworld
spec:
  replicas: 1
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
        - name: helloworld
          image: meghanahadimani/nodejs-app
          ports:
            - containerPort: 8000

apiVersion: v1
kind: Service
metadata:
  name: nodejs-service
  labels:
    app: helloworld
spec:
  type: NodePort
  ports:
    - port: 8000
      protocol: TCP
      targetPort: 8000
      nodePort: 32121
  selector:
    app: helloworld
```

2. Start minikube

```
deployment.yaml - Devops - Visual Studio Code
Nov 14 16:25 • deployment.yaml - Devops - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER deployment.yaml
  > node_modules
  .dockerignore
  deployment.yaml
  Dockerfile
  package-lock.json
  package.json
  server.js
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
meghana@mehgana-X510UNR:~/Documents/Devops$ minikube start --driver=docker
minikube v1.24.0 on Ubuntu 20.04
Using the docker driver based on existing profile
Starting control plane node minikube in cluster minikube
Pulling base image...
Updating container runtime Docker "minikube" to Docker 20.10.8 ...
Preparing Kubernetes v1.22.3 on Docker 20.10.8 ...
Verifying Kubernetes components...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
  Using image kubernetesui/dashboard:v2.3.1
  Using image k8s.gcr.io/ingress-nginx/controller:v1.0.4
  Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
  Using image kubernetesui/metrics-scraper:v1.0.7
  Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
Verifying Ingress addon...
Enabled addons: storage-provisioner, default-storageclass, dashboard, ingress
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

3. Then apply the yaml file using the kubectl command. `kubectl apply -f deployment.yaml` The output is: given below.

```
mehgana@mehgana-X510UNR:~/Documents/Devops$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-deployment created
service/nodejs-service created
mehgana@mehgana-X510UNR:~/Documents/Devops$
```

This creates a pod with one replica and a service file. To see the output of the pod run command `minikube service nodejs-service`

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-deployment
  labels:
    app: helloworld
spec:
  replicas: 1
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
        - name: helloworld

```

```

Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.
Verifying ingress addon...
Enabled addons: default-provisioner, default-storageclass, dashboard, ingress
Done! Kubectl is now configured to use "minikube" cluster and "default" namespace by default
mehhana@mehhana-X510UR:~/Documents/Devops$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-deployment created
service/nodejs-service created
mehhana@mehhana-X510UR:~/Documents/Devops$ minikube service nodejs-service
|-----|
| NAME | TARGET PORT | URL | |
|---|---|---|---|
| default | nodejs-service | 32121 | http://192.168.49.2:32121 |
|-----|
Opening service default/nodejs-service in default browser...
mehhana@mehhana-X510UR:~/Documents/Devops$ Opening in existing browser session.
[80254:80254:0100:000000:011473:ERROR:sandbox_linux.cc(374)] InitializeSandbox() called with multiple threads in process gpu-process.
^C
mehhana@mehhana-X510UR:~/Documents/Devops$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nodejs-deployment-6558876dc7-dlhtm   1/1     Running   0          115s
mehhana@mehhana-X510UR:~/Documents/Devops$ 

```

Ln 20, Col 15 Spaces: 2 UTF-8 LF YAML

- To see the pod has been created go the minikube dashboard using command `minikube dashboard`

Name	Namespace	Images	Labels	Pods	Created
nodejs-deployment	default	mehghanahadiman/nodejs-app	app: helloworld	1 / 1	2.minutes.ago

Name	Namespace	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
nodejs-deployment-6558876dc7-dlhtm	default	mehghanahadiman/nodejs-app	app: helloworld pod-template-hash: 6558876dc7	minikube	Running	0	-	-	2.minutes.ago

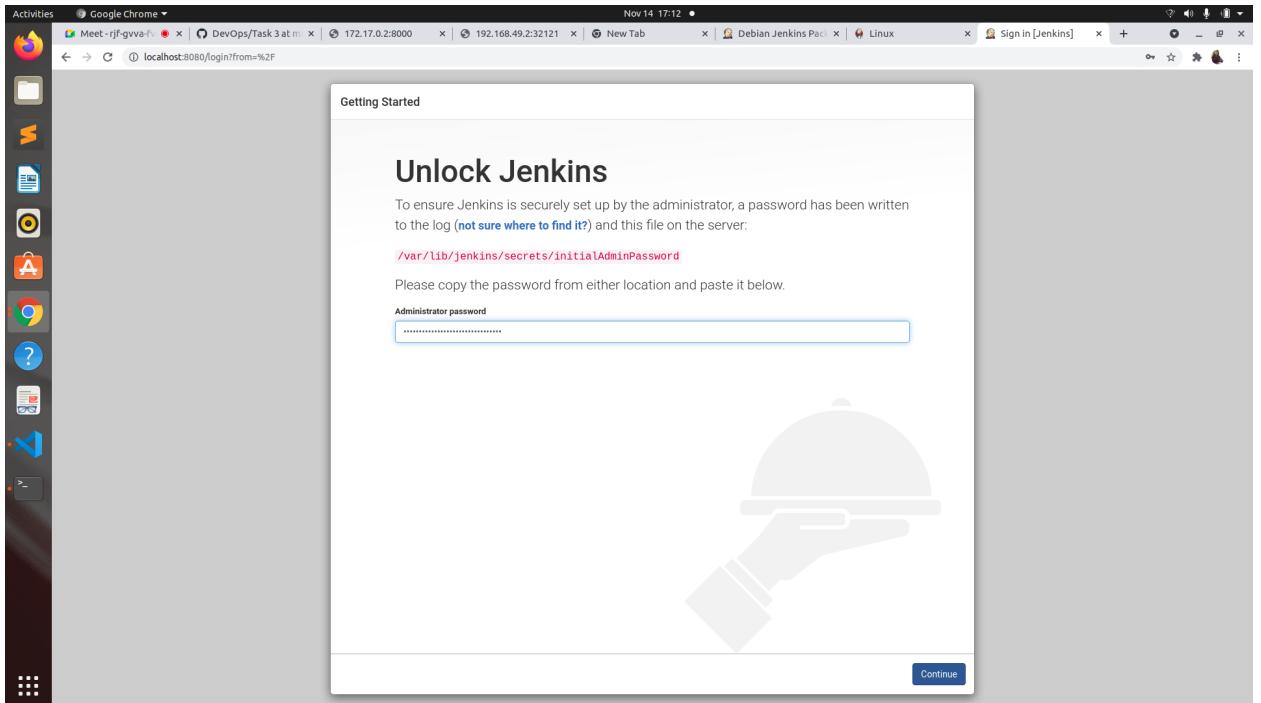
Name	Namespace	Images	Labels	Pods	Created
nodejs-deployment-6558876dc7	default	mehghanahadiman/nodejs-app	app: helloworld pod-template-hash: 6558876dc7	1 / 1	2.minutes.ago

Step - 4:

Push all the code into a git repository

Step - 5:

1. Install Jenkins Run the jenkins file, the first time it asks for an admin password. Run `cat /var/lib/jenkins/secrets/initialAdminPassword` and copy the password and paste it.



Next install the suggested plugins After installing the dashboard opens up Go to Manage Jenkins -> Manage Plugins -> Available and install plugins for Nodejs, Docker, Kubernetes, Github, Kubernetes CLI.

2. Configure Dockerhub and github credentials. Go to Manage Jenkins -> Manage credentials and add a credential for docker hub with your username and password. Create a credential id (which will be used later) and description. Similarly add github credentials, i.e. username and Personal Access Token.

The screenshot shows a Linux desktop environment with a terminal window and a Jenkins web interface. The terminal window displays Jenkins setup logs, including Java version information and the start of the Jenkins service via systemctl. The Jenkins interface shows the 'Global credentials (unrestricted)' page with two entries: 'dockerhub' and 'github', both of which are 'Username with password' type credentials.

- Run the following command to create an environment variable named KUBECONFIG and provide the .kube/config path. Jenkins goes to this path to execute kubectl commands. export KUBECONFIG=~/ .kube/config

```

Do you want to continue? [Y/n] Y
Setting up Jenkins (2.303.3) ...
meghana@meghana-X510UNR:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
meghana@meghana-X510UNR:~$ sudo systemctl start jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Sun 2021-11-14 17:00:13 IST; 1min 54s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 111015 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)
Nov 14 17:00:10 meghana-X510UNR systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: Correct java version found
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: * Starting Jenkins Automation Server Jenkins
meghana@meghana-X510UNR su[111054]: session opened for user jenkins by (uid=0)
Nov 14 17:00:11 meghana-X510UNR jenkins[111015]: Jenkins is already the newest version (2.303.3).
The following packages were automatically installed and are no longer required:
lbbmessaging-menu0 linux-headers-5.11.0-37-generic linux-hwe-5.11-headers-5.11.0-37 linux-image-5.11.0-37-generic linux-modules-5.11.0-37-generic linux-modules-extra-5.11.0-37-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
meghana@meghana-X510UNR:~$ sudo systemctl start jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Sun 2021-11-14 17:00:13 IST; 1min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 111015 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)
Nov 14 17:00:10 meghana-X510UNR systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: Correct java version found
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: * Starting Jenkins Automation Server Jenkins
meghana@meghana-X510UNR su[111054]: session opened for user jenkins by (uid=0)
Nov 14 17:00:11 meghana-X510UNR jenkins[111015]: Jenkins is already the newest version (2.303.3).
The following packages were automatically installed and are no longer required:
lbbmessaging-menu0 linux-headers-5.11.0-37-generic linux-hwe-5.11-headers-5.11.0-37 linux-image-5.11.0-37-generic linux-modules-5.11.0-37-generic linux-modules-extra-5.11.0-37-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
meghana@meghana-X510UNR:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Sun 2021-11-14 17:00:13 IST; 1min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 111015 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)
Nov 14 17:00:10 meghana-X510UNR systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: Correct java version found
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: * Starting Jenkins Automation Server Jenkins
meghana@meghana-X510UNR su[111054]: session opened for user jenkins by (uid=0)
Nov 14 17:00:11 meghana-X510UNR jenkins[111015]: Jenkins is already the newest version (2.303.3).
The following packages were automatically installed and are no longer required:
lbbmessaging-menu0 linux-headers-5.11.0-37-generic linux-hwe-5.11-headers-5.11.0-37 linux-image-5.11.0-37-generic linux-modules-5.11.0-37-generic linux-modules-extra-5.11.0-37-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
meghana@meghana-X510UNR:~$ export KUBECONFIG=~/ .kube/config
meghana@meghana-X510UNR:~$ 

```

Step - 6:

1. Create a jenkins pipeline. Add this code in a file named Jenkinsfile.

```
pipeline{  
    environment {  
        registry = "meghanahadimani/nodejs-app"  
        registryCredential = 'dockerhub'  
        dockerImage = ''  
    }  
    agent any  
    stages {  
        stage('Build') {  
            steps {  
                script {  
                    sh 'npm install'  
                }  
            }  
        }  
        stage('Building image') {  
            steps {  
                script {  
                    dockerImage = docker.build registry + ":latest"  
                }  
            }  
        }  
        stage('Push Image') {  
            steps {  
                script {  
                    docker.withRegistry( '',  
registryCredential){  
                        dockerImage.push()  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        }

    }

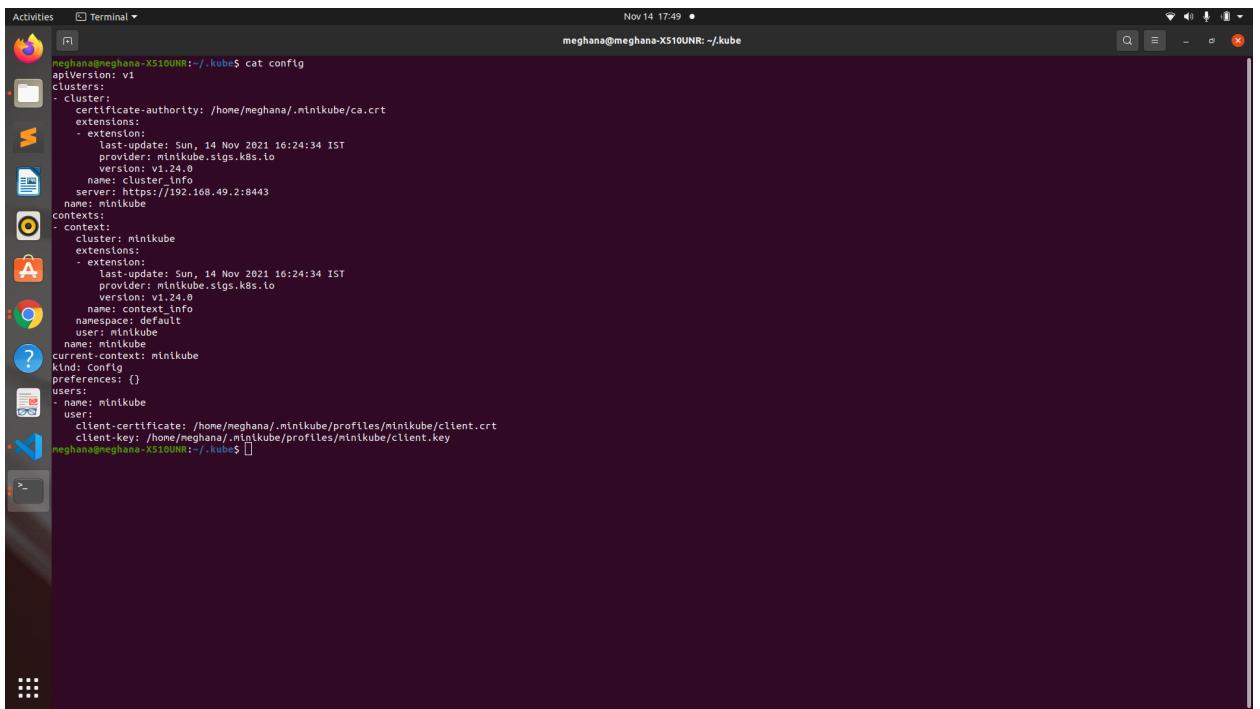
stage('Deploying into k8s'){

    steps{
        withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'jenkins-token', namespace: 'default',
serverUrl: 'https://192.168.49.2:8443') {
            sh 'kubectl apply -f deployment.yaml'
        }
    }

}

}
```

Go to the `~/.kube` directory and type the command `cat config` to get all the information about the cluster to fill inside the `withKubeConfig` method in the pipeline.



```
meghana@meghana-X510UNR:~/kube$ cat config
apiVersion: v1
clusters:
- cluster:
  certificate-authority: /home/meghana/.minikube/ca.crt
  extensions:
  - extension:
    last-update: Sun, 14 Nov 2021 16:24:34 IST
    provider: minikube.sigs.kbs.io
  version: v1
  name: cluster-info
  server: https://192.168.49.2:8443
name: minikube
contexts:
- context:
  cluster: minikube
  extensions:
  - extension:
    last-update: Sun, 14 Nov 2021 16:24:34 IST
    provider: minikube.sigs.kbs.io
  version: v1.24.0
  name: context_info
  namespace: default
  user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/meghana/.minikube/profiles/minikube/client.crt
    client-key: /home/meghana/.minikube/profiles/minikube/client.key
meghana@meghana-X510UNR:~/kube$
```

2. Push this file into github repository
3. Now create a pipeline project in jenkins. Click on New Item in the dashboard. Put a name of your choice, choose Pipeline and save it.
4. Go to the Build Triggers section and choose Github hook trigger for GITScm polling.
5. In the Pipeline phase choose Pipeline script from SCM and choose the SCM as Git and provide the URL to your github repository. Save this file and check the process by clicking on Build Now.
6. If you click on the Full Stage View option, it should look like this:

Activities Google Chrome Nov 15 20:42

localhost:8080/job/nodejs-app/ Jenkins nodejs-app [Jenkins] Meet - rjf-gvva-fvo How to Install Kubernetes Instances [EC2 Manager] 13.126.134.19 DEVOPS/server.js at master

Pipeline nodejs-app

[Recent Changes](#)

Stage View

	Git Clone	Docker build	Docker Login	Push Image to Docker Hub	SSH Into k8s Server	Put deployment.yaml onto k8smaster	Deploy spring boot
Average stage times: (Average full run time: ~39s)	1s	6s	3s	19s	175ms	3s	2s
#28 Nov 15, 2020 1 commit	2s	6s	3s	18s	160ms	6s	7s
#27 Nov 15, 2020 No changes		6s	4s	18s	133ms	1s	1s
#26 Nov 15, 1949 No changes	796ms	5s	4s	21s	136ms	1s	1s
#25 Nov 15, 1949 No changes	712ms	6s	4s	20s	136ms	1s	1s
#24 Nov 15, 1949 No changes	1s	7s	4s	20s	313ms	5s	2s

[Show all](#)

output .

Activities Google Chrome Nov 15 20:38

localhost:8080/job/nodejs-app/28 Console nodejs-app [Jenkins] Meet - rjf-gvva-fvo How to Install Kubernetes Instances [EC2 Manager] 13.126.134.19 DEVOPS/server.js at master

Console Output

Started by user Meghana Hadimani
 [Pipeline] Start of Pipeline
 [Pipeline] node
 [Pipeline] ⚡ Jenkins in /var/lib/jenkins/workspace/nodejs-app
 [Pipeline] ⚡ Jenkins stage
 [Pipeline] ⚡ Jenkins { (Git Clone)
 [Pipeline] ⚡ Jenkins git
 The recommended git tool is: git
 using credential Jenkins-token
 > git rev-parse --show-toplevel > /var/lib/jenkins/workspace/nodejs-app/.git # timeout=10
 Fetching upstream changes from https://github.com/meghanumh/DEVOPS.git
 git config remote.origin.url https://github.com/meghanumh/DEVOPS.git # timeout=10
 Fetching upstream changes from https://github.com/meghanumh/DEVOPS.git
 > git --version # timeout=10
 > git -version # git version 2.25.1'
 using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress https://github.com/meghanumh/DEVOPS.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 Checking for updates from https://github.com/meghanumh/DEVOPS.git
 git config core.sparsecheckout # timeout=10
 > git checkout -f 4e4da2c7ca7ffbe9f4a16df1c7c16563568c22eb
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 4e4da2c7ca7ffbe9f4a16df1c7c16563568c22eb # timeout=10
 Commit message: "Update server.js"
 > git rev-list --no-walk 98855efc57fd8bf38445790f6912fb9762468d6 # timeout=10
 [Pipeline] ⚡ Jenkins // stage
 [Pipeline] ⚡ Jenkins stage
 [Pipeline] ⚡ Jenkins { (Docker build)
 [Pipeline] ⚡ Jenkins sh
 + docker version
 Client: Docker Engine - Community
 Version: 20.10.10
 API version: 1.41
 Go version: go1.16.9
 Git commit: b4685636
 Built: Mon Oct 25 07:42:59 2021

me.google.com is sharing your screen. [Stop sharing](#) [Hide](#)

Activities Google Chrome ▾

Nov 15 20:42

localhost:8080/job/nodejs-app/28/console

Dashboard nodejs-app #28

```
ed0a3d9cbcc7: Waiting
4a066160805a3: Waiting
8c8e652ec0df: Waiting
76bb0808cbef: Layer already exists
76bb0808cbef: Layer already exists
00e0d433e10: Layer already exists
a579d55a21b0: Layer already exists
f0d8fcfd8a81: Layer already exists
4a066160805a3: Layer already exists
1e9c28006010: Layer already exists
b257e69d416f: Layer already exists
cddb98d77103: Layer already exists
8c8e652ec0df: Layer already exists
8c8e652ec0df: Layer already exists
2f4ea8a2e1b5: Layer already exists
c7107a5c50b0: Pushed
mehana-docker-image: digest: sha256:f30f2a30be9ee2b058b1135b36ea34a14fb27b90fa0db0de4dfcc9240def43 size: 3053
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SSH Into k8s Server)
[Pipeline] { (Put deployment.yaml onto k8smaster)
[Pipeline] sshPut
[Pipeline] Sending a file/directory to kubernetes-master[13.126.134.19]: from: /var/lib/jenkins/workspace/nodejs-app/deployment.yaml into: .
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy spring boot)
[Pipeline] { (Run command)
[Pipeline] Executing command on kubernetes-master[13.126.134.19]: kubectl apply -f deployment.yaml sudo: false
deployment.apps/nodejs: deployment unchanged
service/nodejs: service unchanged
[Pipeline] 
[Pipeline] // stage
[Pipeline] 
[Pipeline] // stage
[Pipeline] { (Run command)
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.303.3

Activities Google Chrome ▾

Nov 15 20:25

localhost:8080/job/nodejs-app/Jenkins [jenkins]

Not secure | 192.168.49.2:32121

Hello World! A new commit was made! It should trigger a Jenkins Build!

meet.google.com is sharing your screen. Stop sharing Hide Show all