

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

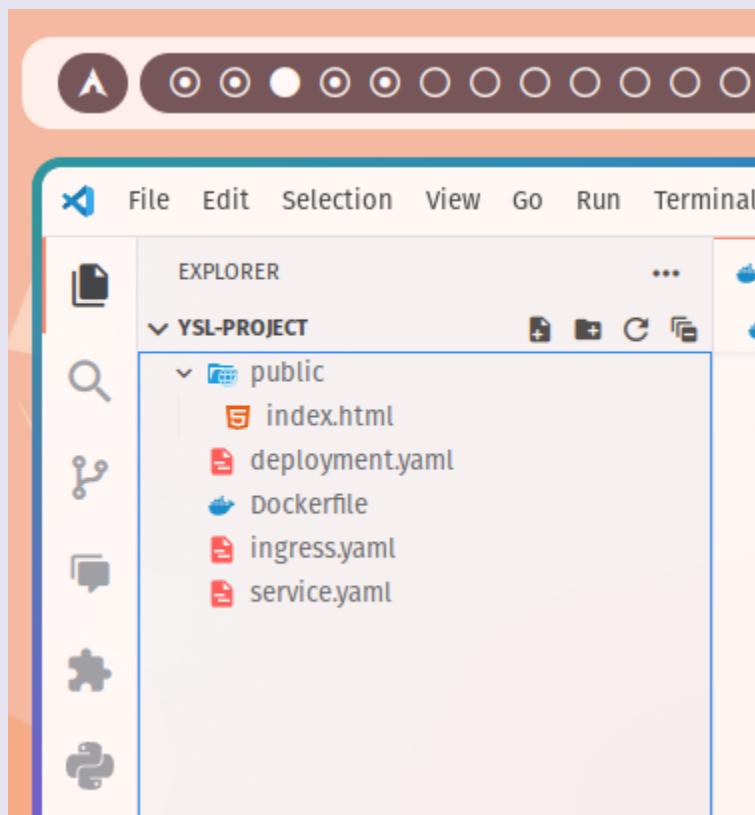
CS Practical 4

Scenario : Create docker container for deploying on containerized platform. Here you need to build Node js Web Application on Docker Containers and will run the it using Kubernetes Service on IBM Cloud

Steps and Screenshots :

1. Create Dockerfile for node project

Here, project contains public folder with html file to be hosted using express module, whose structure looks like :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

Dockerfile :

```
# Use a lightweight Node.js image as the base image
FROM node:18-alpine

# Set the working directory inside the container
WORKDIR /app

# Copy only the necessary files to the container
COPY public/ ./public/

# Install only production dependencies (so it will be quick)
RUN npm install express

# Create a minimal package.json for the express server
RUN echo '{ "name": "curse-html", "version": "1.0.0", "main": "app.js",
"dependencies": { "express": "^4.18.2" } }' > package.json

# Create the app.js file to serve the static content
RUN echo "const express = require('express'); const path =
require('path'); const app = express();
app.use(express.static(path.join(__dirname, 'public'))); const port =
3000; app.listen(port, () => { console.log('Server running on
http://localhost:' + port); });" > app.js

EXPOSE 3000

CMD ["node", "app.js"]
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CS Practical 4

2. Build docker image in local system

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

- File Explorer:** Shows a project named "YSL-PROJECT" containing files "index.html" and "Dockerfile".
- Terminal:** Displays the command "docker build -t yslprjkt:1.0 .". The output of the build process is shown, indicating the creation of a Docker image named "y়slprjkt:1.0".
- Dockerfile Content:**

```
11 RUN npm install express
12
13 # Create a minimal package.json for the express server
14 RUN echo '{ "name": "curse-html", "version": "1.0.0", "main": "app.js", "dependencies": { "express": "^4.18.2" } }' > package.json
15
16 # Create the app.js file to serve the static content
17 RUN echo "const express = require('express'); const path = require('path'); const app = express(); app.use(express.static(path.join(__dirname, 'public'))); app.get('/', (req, res) => res.sendFile(path.join(__dirname, 'index.html'))); app.listen(3000, () => console.log('Server is running on port 3000'))" > app.js
18
19 # Expose the port the app will run on
20 EXPOSE 3000
21
22 # Define the command to run the application
23 CMD ["node", "app.js"]
```
- Bottom Status Bar:** Shows various icons and status information, including "Ln 24, Col 1", "Spaces: 4", "UTF-8", "LF", "Dockerfile", and "Go Live".

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CS Practical 4

3. For secure login using password through local system CLI, some tools like ibmcloud-cli, docker, etc. requires gpg keys and pass initiation

The screenshot shows a desktop environment with a pink and orange gradient background. A terminal window titled "Kitty Terminal" is open, displaying system information and a GPG key generation process.

System Information (top of terminal):

- OS : Garuda Linux x86_64
- Kernel : Linux 6.10.8-zen1-zen
- Packages : 2156 (pacman)[stable]
- Display : 1920x1080 @ 60 Hz in 17" [Built-in]
- WM : Hyprland (Wayland)
- Terminal : kitty 0.36.2

GPG Key Generation (bottom of terminal):

```
ysl ~ 07:59 gpg --generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Yash Lakhtariya
Email address: yashslakhtariya21@gnu.ac.in
You selected this USER-ID:
  "Yash Lakhtariya <yashslakhtariya21@gnu.ac.in>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? 
```

The desktop interface includes a dock at the bottom with icons for various applications like a terminal, file manager, browser, and code editor.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

```
ysl ~ 08:00 gpg --list-keys
[keyboxd]
-----
pub ed25519 2018-11-30 [SC]
248970928458509C5080AC0350585C4E9518F26
uid [ unknown] proton@srp.modulus
sub cv25519 2018-11-30 [E]

pub ed25519 2024-09-05 [SC] [expires: 2027-09-05]
620C719EC27ABAFA0D2438B2C095A2586B3B5B5
uid [ultimate] Yash Lakhtariya <yashslakhtariya21@gnu.ac.in>
sub cv25519 2024-09-05 [E] [expires: 2027-09-05]

pub ed25519 2019-11-20 [SC] [expires: 2024-11-18]
76C0969C076A23EC4849F462D3E3E8ACF99DE8EA
uid [ unknown] Ashish Kulkarni (Open Source) <ashish@kulkarni.dev>
sub cv25519 2019-11-20 [E] [expires: 2024-11-18]
```

```
ysl ~ 08:00 gpg --list-keys
[keyboxd]
-----
pub ed25519 2018-11-30 [SC]
248970928458509C5080AC0350585C4E9518F26
uid [ unknown] proton@srp.modulus
sub cv25519 2018-11-30 [E]

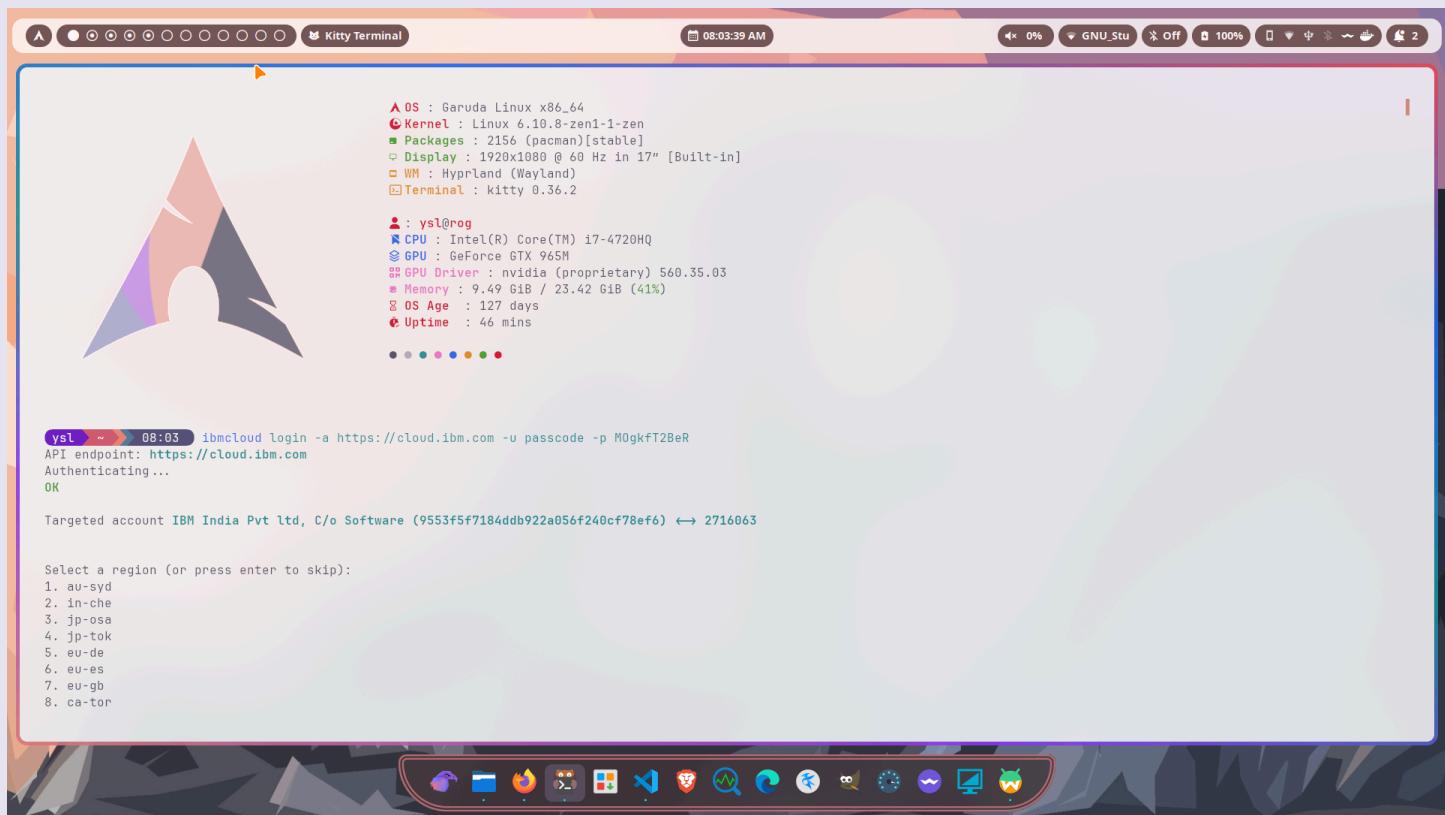
pub ed25519 2024-09-05 [SC] [expires: 2027-09-05]
620C719EC27ABAFA0D2438B2C095A2586B3B5B5
uid [ultimate] Yash Lakhtariya <yashslakhtariya21@gnu.ac.in>
sub cv25519 2024-09-05 [E] [expires: 2027-09-05]

pub ed25519 2019-11-20 [SC] [expires: 2024-11-18]
76C0969C076A23EC4849F462D3E3E8ACF99DE8EA
uid [ unknown] Ashish Kulkarni (Open Source) <ashish@kulkarni.dev>
sub cv25519 2019-11-20 [E] [expires: 2024-11-18]
```

```
ysl ~ 08:01 pass init 620C719EC27ABAFA0D2438B2C095A2586B3B5B5|
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

4. Copy CLI login command from IBM Cloud homepage



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CS Practical 4

ibmcloud login -a https://cloud.ibm.com -u passcode -p M0gkFT2BeR
API endpoint: https://cloud.ibm.com
Authenticating...
OK

Targeted account IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) ↔ 2716063

Select a region (or press enter to skip):
1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. eu-de
6. eu-es
7. eu-gb
8. ca-tor
9. us-south
10. us-east
11. br-sao
Enter a number> 1
Targeted region au-syd

API endpoint: https://cloud.ibm.com
Region: au-syd
User: yashslakhtariya21@gnu.ac.in
Account: IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) ↔ 2716063
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'

```
ysl ~ 08:03 |
```

Select a region (or press enter to skip):
1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. eu-de
6. eu-es
7. eu-gb
8. ca-tor
9. us-south
10. us-east
11. br-sao
Enter a number> 1
Targeted region au-syd

API endpoint: https://cloud.ibm.com
Region: au-syd
User: yashslakhtariya21@gnu.ac.in
Account: IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) ↔ 2716063
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'

```
yol ~ 08:03 ibmcloud plugin install container-service
```

Looking up 'container-service' from repository 'IBM Cloud' ...
Plug-in 'container-service[kubernetes-service/ks] 1.0.652' found in repository 'IBM Cloud'
Plug-in 'container-service[kubernetes-service/ks] 1.0.652' was already installed. Do you want to re-install it or not? [y/N] > n
Plugin installation was canceled.

```
yol ~ 08:04 ibmcloud plugin install container-registry
```

Looking up 'container-registry' from repository 'IBM Cloud' ...
Plug-in 'container-registry[cr] 1.3.10' found in repository 'IBM Cloud'
Plug-in 'container-registry[cr] 1.3.10' was already installed. Do you want to re-install it or not? [y/N] > n
Plugin installation was canceled.

```
yol ~ 08:04 |
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

The screenshot shows the IBM Cloud Kubernetes cluster overview for 'mycluster-dal10-b3c.4x16-group3'. The 'Overview' tab is selected. Key status indicators include:

- Worker node status:** 1 of 1 (Healthy)
- Add-on status:** 0 of 0 (Healthy)
- Master status:** Normal
- Ingress status:** Warning (Yellow)

Cluster details:

- Cluster ID:** cr3cpfcs0m882o64nbq0 (highlighted with a red box)
- Version:** 1.30.4_1534
- Infrastructure:** Classic
- Master location:** Sydney
- Image security enforcement:** Enable

Worker node health: 1 total worker nodes (100% Normal)

Networking: Service endpoint URL (Public enabled, Copy link), Ingress subdomain (mycluster-dal10-b3c-4x16-3e4769f510db5aaf1089354e49621b41-0000.au-syd.containers.appdomain.cloud)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4



```
▲ OS : Garuda Linux x86_64
● Kernel : Linux 6.10.8-zen1-1-zen
■ Packages : 2156 (pacman)[stable]
□ Display : 1920x1080 @ 60 Hz in 17" [Built-in]
□ WM : Hyprland (Wayland)
▣ Terminal : kitty 0.36.2

● : ysl@rog
CPU : Intel(R) Core(TM) i7-4720HQ
GPU : GeForce GTX 965M
GPU Driver : nvidia (proprietary) 560.35.03
Memory : 9.50 GiB / 23.42 GiB (41%)
OS Age : 127 days
Uptime : 48 mins

● ● ● ● ● ● ●
```

```
ysl ~ 08:05 ibmcloud ks cluster config --cluster cr3cpfc0m882o64nbq0
OK
The configuration for cr3cpfc0m882o64nbq0 was downloaded successfully.

Added context for cr3cpfc0m882o64nbq0 to the current kubeconfig file.
You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.

ysl ~ 08:05 kubectl config current-context
mycluster-dal10-b3c.4x16-group3/cr3cpfc0m882o64nbq0

ysl ~ 08:06 kubectl get nodes
NAME      STATUS   ROLES   AGE   VERSION
10.210.8.252   Ready    <none>  17d   v1.30.3+IKS

ysl ~ 08:10 ibmcloud cr namespace-add yashlani-nmspc
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

```
ibmcloud cr namespace-list
Listing namespaces for account 'IBM India Pvt ltd, C/o Software' in registry 'au.icr.io' ...

Namespace
aniketnamespace
aryannamespace
cb-namespace-one
devnm
dhairyam
dhruvinamespace
harshnamespace
kirtannamespace
krupalnamespace
kshitijname
nazirnamespace
prarthinspace
raj-profile-project
rajsphere
ronak
shivamnamespace
testapikey
vishwanamespace
vivek
yashlani-nmspc
OK
```

```
ibmcloud cr namespace-list
Listing namespaces for account 'IBM India Pvt ltd, C/o Software' in registry 'au.icr.io' ...

Namespace
aniketnamespace
aryannamespace
cb-namespace-one
devnm
dhairyam
dhruvinamespace
harshnamespace
kirtannamespace
krupalnamespace
kshitijname
nazirnamespace
prarthinspace
raj-profile-project
rajsphere
ronak
shivamnamespace
testapikey
vishwanamespace
vivek
yashlani-nmspc
```

here it will ask for password set in gpg key we created

Passphrase:

Please enter the passphrase to unlock the OpenPGP secret key:
"Yash Lakhtariya <yash.lakhtariya2@gnu.ac.in>"
255-bit ECDH key, ID DB7223c221735D5,
created 2024-09-05 (main key ID 2C095A2586B3B5B5).

Password:

Save in password manager

Cancel **OK**

```
ibmcloud cr login
Logging 'docker' in to 'au.icr.io'...
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

```
ysl ~ 08:10 ibmcloud cr namespace-list
Listing namespaces for account 'IBM India Pvt ltd, C/o Software' in registry 'au.icr.io'...

Namespace
aniketnamespace
aryannamespace
cb-namespace-one
devnm
dhairyanm
dhruvinamespace
harshnamespace
kirtannamespace
krupanamespace
kshitijname
nazirnamespace
prarthinspace
raj-profile-project
rajspace
ronak
shivamnamespace
testapikey
vishwanamespace
vivek
yashlani-nmspc

OK

ysl ~ 08:11 ibmcloud cr login
Logging 'docker' in to 'au.icr.io'...
Logged in to 'au.icr.io'.

OK

ysl ~ 08:12 |
```

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

- Terminal:** The terminal shows the command `ibmcloud cr namespace-list` being run, listing various namespaces.
- Dockerfile:** The Dockerfile contains the following code:

```
1 # Install only production dependencies (none in this case, so it will be quick)
2 RUN npm install express
3 # Create a minimal package.json for the express server
4 RUN echo '{ "name": "curse-html", "version": "1.0.0", "main": "app.js", "dependencies": { "express": "^4.18.2" } }' > package.json
5
6 # Create the app.js file to serve the static content
7 RUN echo "const express = require('express'); const path = require('path'); const app = express(); app.use(express.static(path.join(__dirname, 'public'))); app.get('/', (req, res) => res.sendFile(path.join(__dirname, 'index.html'))); const port = 3000; app.listen(port, () => console.log(`Server is running on port ${port}`));" > app.js
8
9 # Expose the port the app will run on
10 EXPOSE 3000
11
12 # Define the command to run the application
13 CMD ["node", "app.js"]
```
- Output Panel:** The output panel shows the build process of the Docker image, including transferring context, copying files, installing dependencies, and creating layers. It also shows the final image sha256: and its naming to docker.io/library/yslprjkt:1.0.
- Terminal:** The terminal shows the command `docker tag yslprjkt:1.0 au.icr.io/yashlani-nmspc/yslprjkt:1.0` being run.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

kirtannamespace
krupanamespace
kshitijname
nazirnamespace
prarthinspace
raj-profile-project
rajspace
ronak
shivammamespace
testapikey
vishwanamespace
vivek
yashlani-nmspc

OK

```
ysl ~ 08:11 ibmcloud cr login
Logging 'docker' in to 'au.icr.io'...
Logged in to 'au.icr.io'.
```

OK

```
ysl ~ 08:12 docker push au.icr.io/yashlani-nmspc/yslprjkt:1.0
The push refers to repository [au.icr.io/yashlani-nmspc/yslprjkt]
d0af2ed7c3fc: Pushed
f7569e66d1a8: Pushed
d23cf1a847ac: Pushed
7eef98fe54c: Pushed
22592a090fb1: Pushed
e2be10e97665: Pushed
06fd85419b65: Pushed
f58c462fa079: Pushed
63calfbb43ae: Pushed
1.0: digest: sha256:f81d0e4606bbc7f02b4cbc2ed6d8e84c8c3d663e1babcc9cdc1e2cfcc35d354820 size: 2197
```

ysl ~ 08:16 |

```
ysl ~ 08:17 ibmcloud cr image-list | grep 'yslprjkt'
au.icr.io/yashlani-nmspc/yslprjkt          1.0
MB -
f81d0e4606bb    yashlani-nmspc   24 minutes ago  47
```

ysl ~ 08:17 |

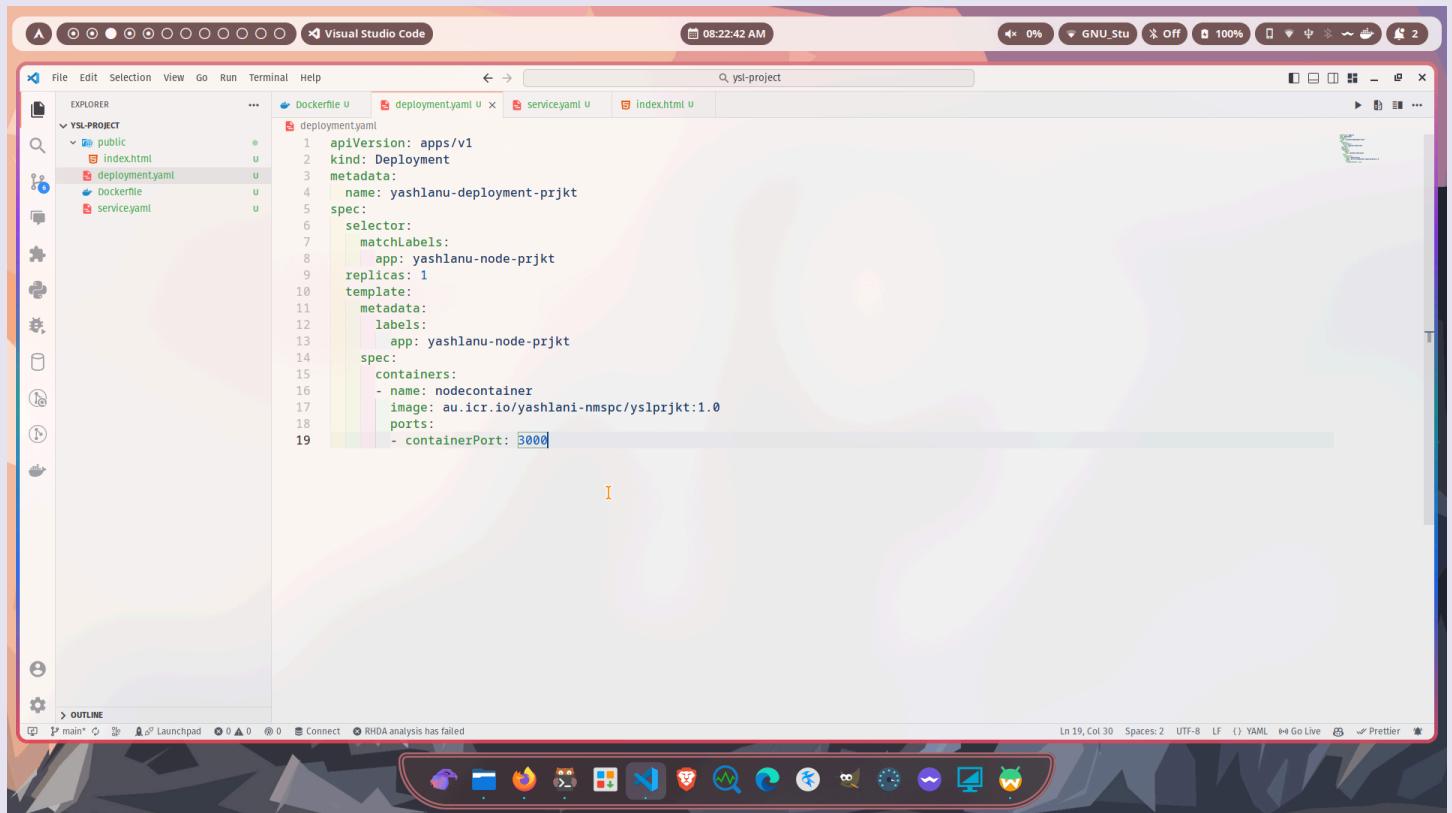
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CS Practical 4

5. Create deployment and service using YAML files



The screenshot shows a Visual Studio Code interface with a project named 'YSL-PROJECT'. The 'EXPLORER' sidebar on the left lists files: Dockerfile, deployment.yaml (selected), index.html, and service.yaml. The main editor area displays the 'deployment.yaml' file content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yashlanu-deployment-prjkt
spec:
  selector:
    matchLabels:
      app: yashlanu-node-prjkt
  replicas: 1
  template:
    metadata:
      labels:
        app: yashlanu-node-prjkt
    spec:
      containers:
        - name: nodecontainer
          image: au.icr.io/yashlani-nmspc/yslprjkt:1.0
          ports:
            - containerPort: 3000
```

The status bar at the bottom indicates the file is 19 lines long, 30 columns wide, and uses UTF-8 encoding.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

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

- Top Bar:** Shows icons for file operations, a search bar with "ysi-project", and system status indicators like battery level (0%), volume (80%), and screen brightness (100%).
- File Explorer:** On the left, under "YSL-PROJECT", there are files: Dockerfile, deployment.yaml, index.html, service.yaml, and Dockerfile. The service.yaml file is currently open.
- Code Editor:** The content of the service.yaml file is displayed:

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: yashlanu-node-prjkt
    name: yashlani-prjkt-service
    namespace: default
spec:
  type: NodePort
  ports:
    - name: http
      protocol: TCP
      port: 3000
  selector:
    app: yashlanu-node-prjkt
```
- Terminal:** At the bottom, the terminal shows the command "ytl .../ysl-project" and the output "main ? 08:21".
- Bottom Status Bar:** Shows "Ln 15, Col 29" and other status indicators.
- Bottom Icons:** A row of small icons representing various extensions or tools available in the VS Code ecosystem.

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

- File Explorer:** Shows a project named "yasl-project" with files: Dockerfile, deployment.yaml, service.yaml, index.html, and Dockerfile.
- Terminal:** Displays the following command history:

```
yasl > ./yasl-project $ main ? 08:22 kubectl apply -f deployment.yaml
deployment.apps/yashlanu-deployment-prjkt created

yasl > ./yasl-project $ main ? 08:22 kubectl apply -f service.yaml
service/yashlanu-prjkt-service created

yasl > ./yasl-project $ main ? 08:22
```
- Status Bar:** Shows the date and time as 08:22:59 AM, battery level at 0%, and system status as Off.
- Bottom Taskbar:** Includes icons for various applications like Terminal, File Manager, and a browser.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

The screenshot shows a Visual Studio Code window with the following details:

- File Explorer:** Shows files like Dockerfile, deployment.yaml, service.yaml, and index.html.
- Terminal:** Displays command-line output from `kubectl`:

 - Deployment:** `kubectl get deployments.apps` output:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
ronak-deploy	0/1	1	0	6d21h
yashlanu-deployment	1/1	1	1	3d22h
yashlanu-deployment-prjkt	1/1	1	1	25s
 - Pods:** `kubectl get pods` output:

NAME	READY	STATUS	RESTARTS	AGE
ronak-deploy-59d4d6fdbd9-t9nz9	0/1	ImagePullBackOff	0	6d21h
yashlanu-deployment-7cc5f79b5f-g4sqm	1/1	Running	0	3d22h
yashlanu-deployment-prjkt-689fdb48b6-bnqzz	1/1	Running	0	33s

The screenshot shows the Kubernetes dashboard interface with the following sections:

- Workloads:** A sidebar menu with options like Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Services, Config and Storage, Config Maps, Persistent Volume Claims, Secrets, Storage Classes, and Cluster.
- Deployments:** A table listing Deployments with columns: Name, Images, Labels, Pods, and Created. The row for "yashlanu-deployment-prjkt" is highlighted with a red box.
- Pods:** A table listing Pods with columns: Name, Images, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created. The row for "yashlanu-deployment-prjkt-689fdb48b6-bnqzz" is highlighted with a red box.
- Replica Sets:** A table listing Replica Sets with columns: Name, Images, Labels, Pods, and Created.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

6. Try scaling the deployment

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal shows a sequence of commands being run in a directory named 'ysl-project'.

```
deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: yashlanu-deployment-prjkt
5 spec:
6   selector:
7     matchLabels:
8       app: yashlanu-node-prjkt
9   replicas: 1
10  template:
11    metadata:
12      labels:
13        app: yashlanu-node-prjkt
14    spec:
15      containers:
16        - name: nodecontainer

PORTS SQL CONSOLE GITLENS TERMINAL COMMENTS
```

```
08:23 kubectl get pods
NAME READY STATUS RESTARTS AGE
ronak-deploy-59d4d6fb9-t9nz9 0/1 ImagePullBackOff 0 6d21h
yashlanu-deployment-7cc5f79b5f-g4sqm 1/1 Running 0 3d22h
yashlanu-deployment-prjkt-689fdb48b6-bnqzz 1/1 Running 0 33s
```

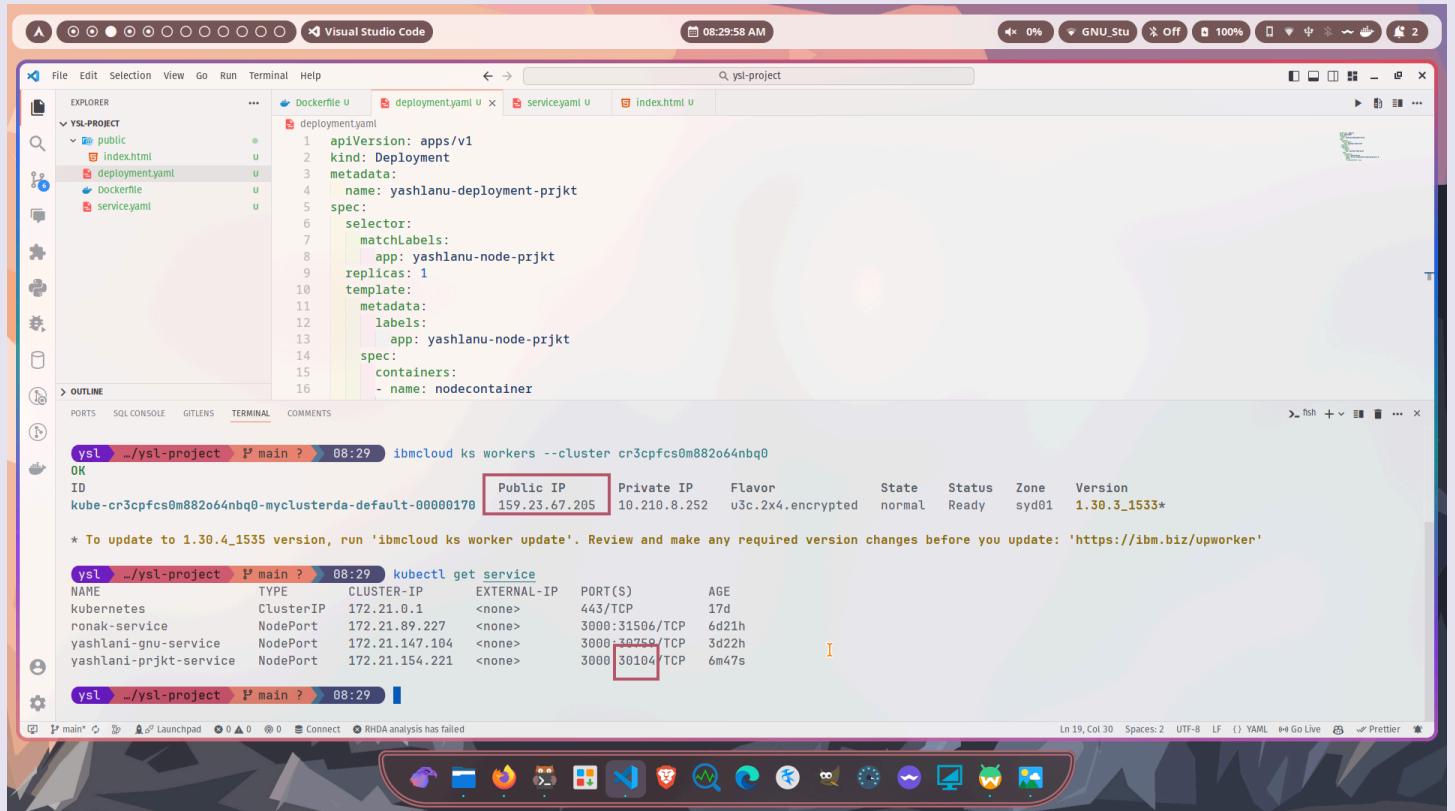
```
08:23 kubectl scale deployment yashlanu-deployment-prjkt --replicas=2
deployment.apps/yashlanu-deployment-prjkt scaled
```

```
08:26 kubectl get pods
NAME READY STATUS RESTARTS AGE
ronak-deploy-59d4d6fb9-t9nz9 0/1 ImagePullBackOff 0 6d21h
yashlanu-deployment-7cc5f79b5f-g4sqm 1/1 Running 0 3d22h
yashlanu-deployment-prjkt-689fdb48b6-bnqzz 1/1 Running 0 4m4s
yashlanu-deployment-prjkt-689fdb48b6-qhv6r 1/1 Running 0 13s
```

The terminal output shows the deployment being scaled from 1 to 2 replicas. The final state lists two pods: 'yashlanu-deployment-prjkt-689fdb48b6-bnqzz' and 'yashlanu-deployment-prjkt-689fdb48b6-qhv6r', both running.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4

7. Get exposed public IP and port using given commands and open in browser



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal shows the following commands and their outputs:

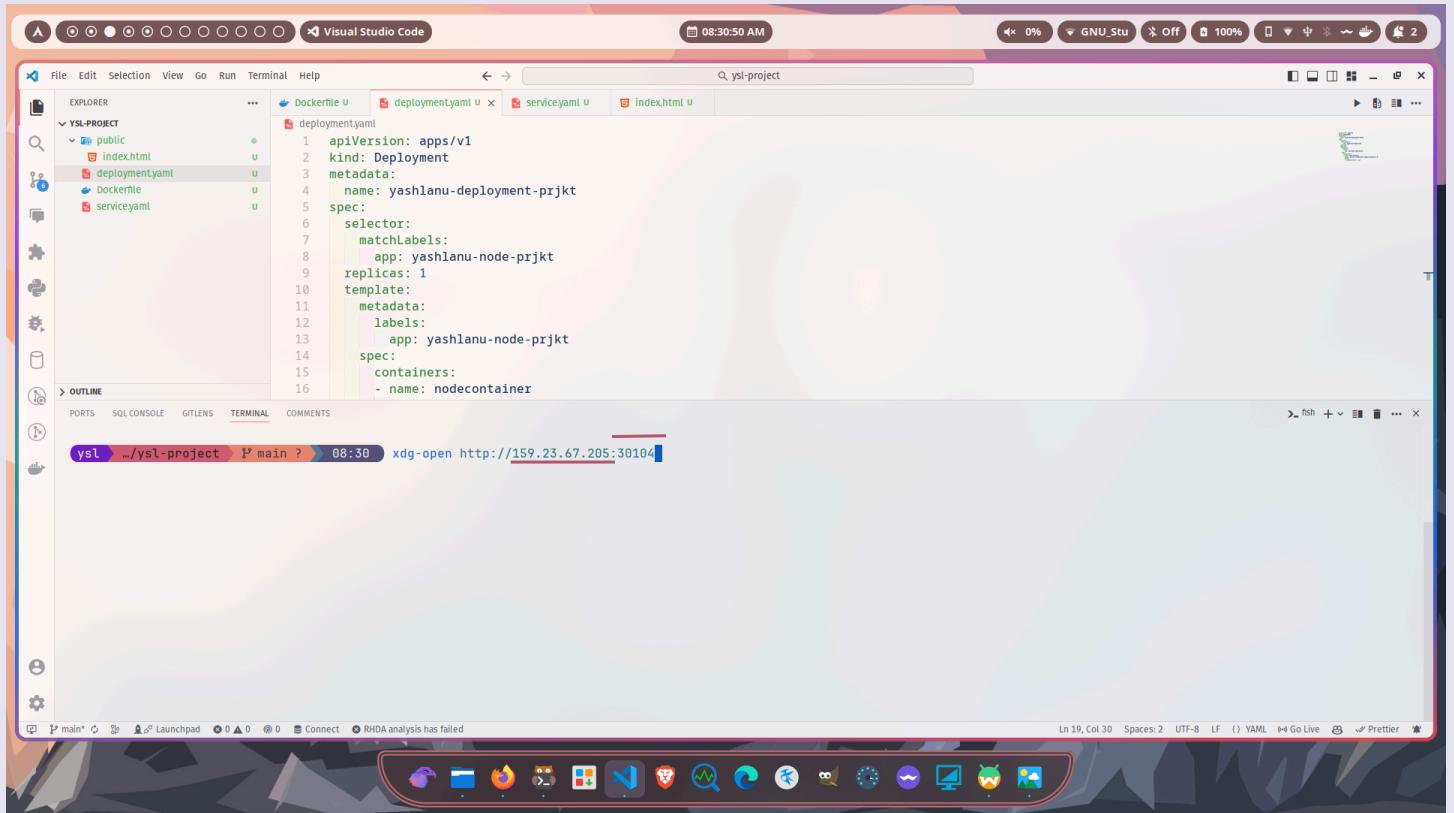
```
ysl .../ysl-project $ ibmcloud ks workers --cluster cr3cpfc80m882o64nbq0
OK
ID          Public IP      Private IP    Flavor   State  Status  Zone  Version
kube-cr3cpfc80m882o64nbq0-myclusterda-default-00000170  159.23.67.205  10.210.8.252  u3c.2x4.encrypted  normal  Ready   syd01  1.30.3_1533*
* To update to 1.30.4_1535 version, run 'ibmcloud ks worker update'. Review and make any required version changes before you update: 'https://ibm.biz/upworker'

ysl .../ysl-project $ kubectl get service
NAME        TYPE      CLUSTER-IP     EXTERNAL-IP   PORT(S)      AGE
kubernetes  ClusterIP  172.21.0.1    <none>       443/TCP     17d
ronak-service  NodePort  172.21.89.227  <none>       3000:31506/TCP  6d21h
yashlanu-gnu-service  NodePort  172.21.147.104  <none>       3000:30750/TCP  3d22h
yashlanu-prjkt-service  NodePort  172.21.154.221  <none>       3000:30104/TCP  6m47s

ysl .../ysl-project $
```

The terminal window has a red box highlighting the "Public IP" column in the first command's output. The status bar at the bottom of the terminal shows "Ln 19, Col 30".

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CS Practical 4



The screenshot shows the Visual Studio Code interface with the title bar "Visual Studio Code" and the status bar showing "08:30:50 AM". The project "YSL-PROJECT" is open, and the "deployment.yaml" file is selected in the Explorer sidebar. The code editor displays the following YAML configuration:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yashlanu-deployment-prjkt
spec:
  selector:
    matchLabels:
      app: yashlanu-node-prjkt
  replicas: 1
  template:
    metadata:
      labels:
        app: yashlanu-node-prjkt
    spec:
      containers:
        - name: nodecontainer
```

The terminal below shows the command "xdg-open http://159.23.67.205:30104" being run.

