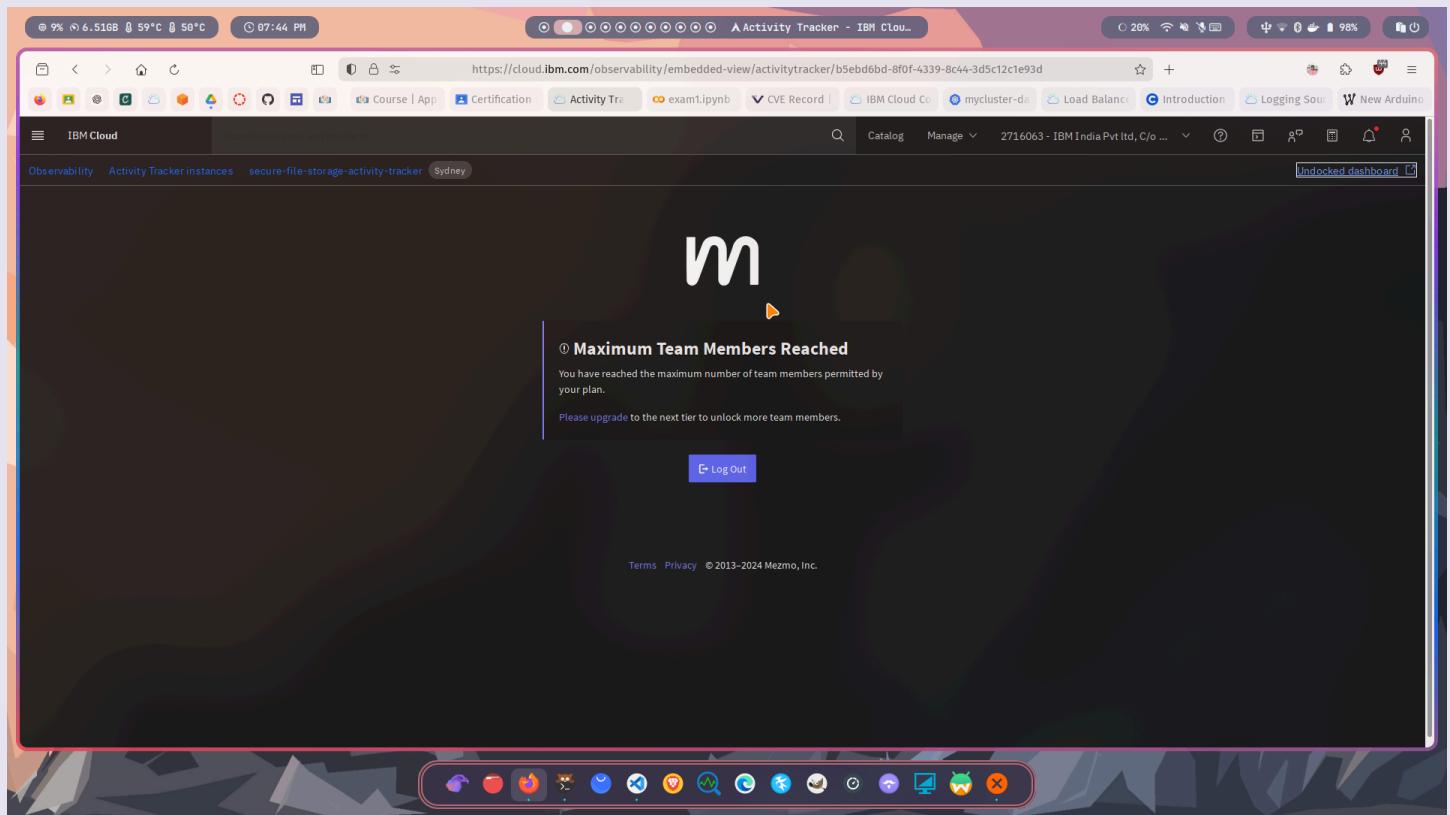


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

**Aim : To create Kubernetes cluster (if not present) on IBM Cloud and monitoring it via IBM Cloud Activity Tracker Instance, ensuring the deployment works properly on the cluster**

Steps and Screenshots :

1. As activity tracker limit is reached, try another instance for monitoring container registry and images which are used to deploy the application on k8s cluster



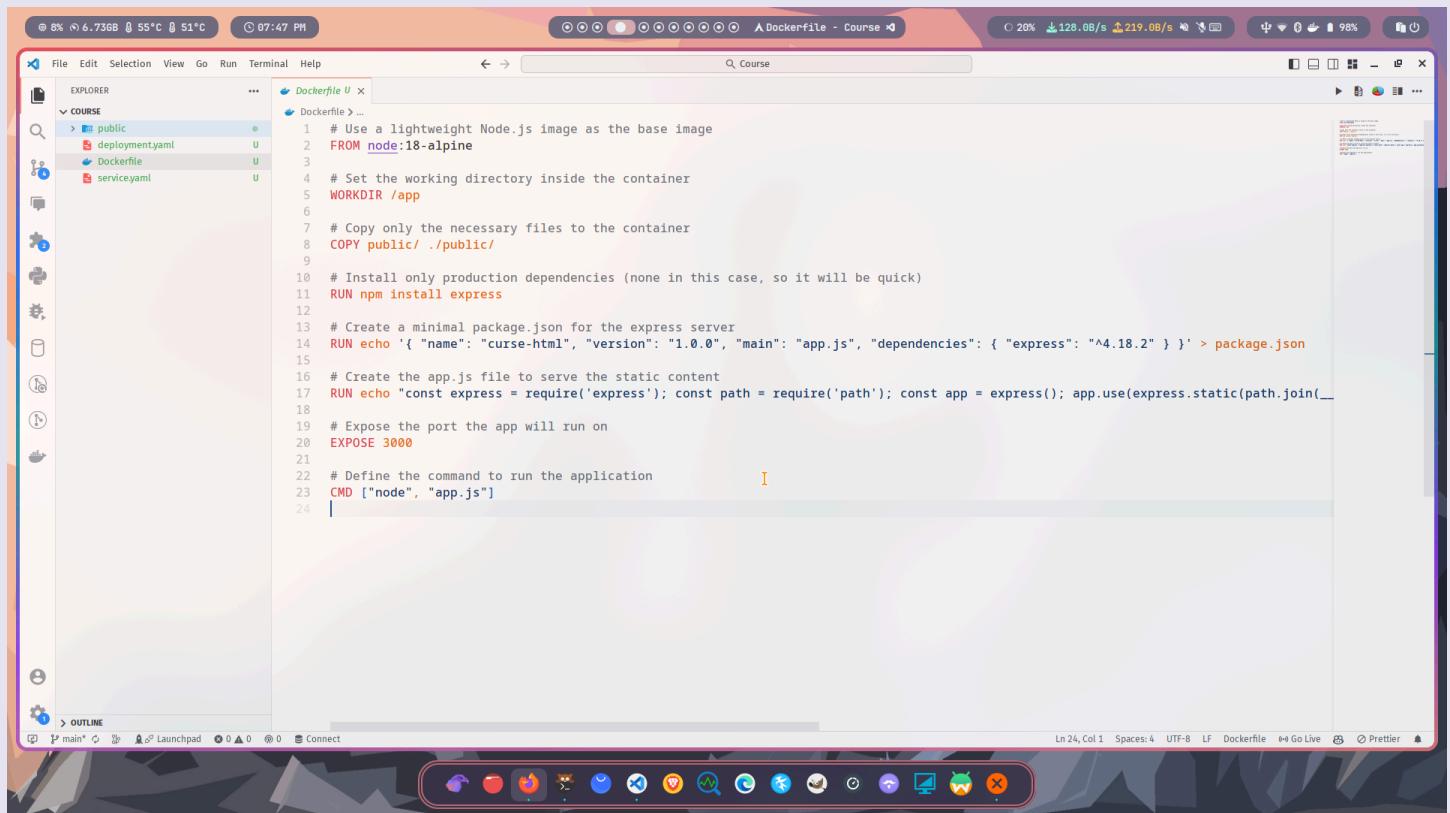
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 71

CS Course : IBM-SECM03EN

2. Create dockerfile to create js file and use setup script for optimal dockerfile, ensuring no warnings are shown using hadolint



The screenshot shows a code editor interface with a Dockerfile open in the main pane. The Dockerfile content is as follows:

```
1 # Use a lightweight Node.js image as the base image
2 FROM node:18-alpine
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy only the necessary files to the container
8 COPY public/ ./public/
9
10 # Install only production dependencies (none in this case, so it will be quick)
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, 'public', '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"]
```

The code editor has various toolbars and panels visible, including an Explorer sidebar on the left showing project files like 'public', 'deployment.yaml', 'Dockerfile', and 'service.yaml'. The bottom status bar shows file details like 'Ln 24, Col 1' and encoding 'UTF-8'.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

The screenshot shows a terminal window with the following content:

```
File Edit Selection View Go Run Terminal Help Dockerfile - Course
Course
Dockerfile > ...
1 # Use a lightweight Node.js image as the base image
2 FROM node:18-alpine
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy only the necessary files to the container
8 COPY public/ ./public/
9
10 # Install only production dependencies (none in this case, so it will be quick)
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, 'public', 'index.html'))); app.listen(3000, () => console.log('Server is running on port 3000'));" > app.js
y়l .../Course ⌘ main ? 07:55 hadolint ./Dockerfile
./Dockerfile:11 DL3016 warning: Pin versions in npm. Instead of `npm install <package>` use `npm install <package>@<version>`
./Dockerfile:14 DL3059 info: Multiple consecutive `RUN` instructions. Consider consolidation.
./Dockerfile:17 DL3059 info: Multiple consecutive `RUN` instructions. Consider consolidation.

y়l .../Course ⌘ main ? 07:55
```

The screenshot shows a terminal window with the following content:

```
File Edit Selection View Go Run Terminal Help Dockerfile - Course
Course
Dockerfile > ...
1 # Use a lightweight Node.js image as the base image
2 FROM node:18-alpine
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy only the necessary files to the container
8 COPY public/ ./public/
9
10 # Copy the setup script and execute it in a single RUN command
11 COPY setup.sh .
12 RUN chmod +x setup.sh && ./setup.sh
13
14 # Expose the port the app will run on
15 EXPOSE 3000
16
17 # Define the command to run the application
18 CMD ["node", "app.js"]

y়l .../Course ⌘ main ? 07:56 hadolint ./Dockerfile
y়l .../Course ⌘ main ? 07:56
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

Optimized Dockerfile :

```
FROM node:18-alpine

WORKDIR /app

COPY public/ ./public/

COPY setup.sh .
RUN chmod +x setup.sh && ./setup.sh

EXPOSE 3000

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

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

Setup script :

```
#!/bin/sh

# Install specific version of express (pinned version)
npm install express@4.21.1

# Create a minimal package.json for the express server
cat <<EOF >package.json
{
  "name": "curse-html",
  "version": "1.0.0",
  "main": "app.js",
  "dependencies": {
    "express": "^4.21.1"
  }
}
EOF

# Create the app.js file to serve the static content
cat <<EOF >app.js
const express = require('express');
const path = require('path');
const app = express();

```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

```
app.use(express.static(path.join(__dirname, 'public')));

const port = 3000;
app.listen(port, () => {
  console.log('Server running on http://localhost:' + port);
});

EOF
```

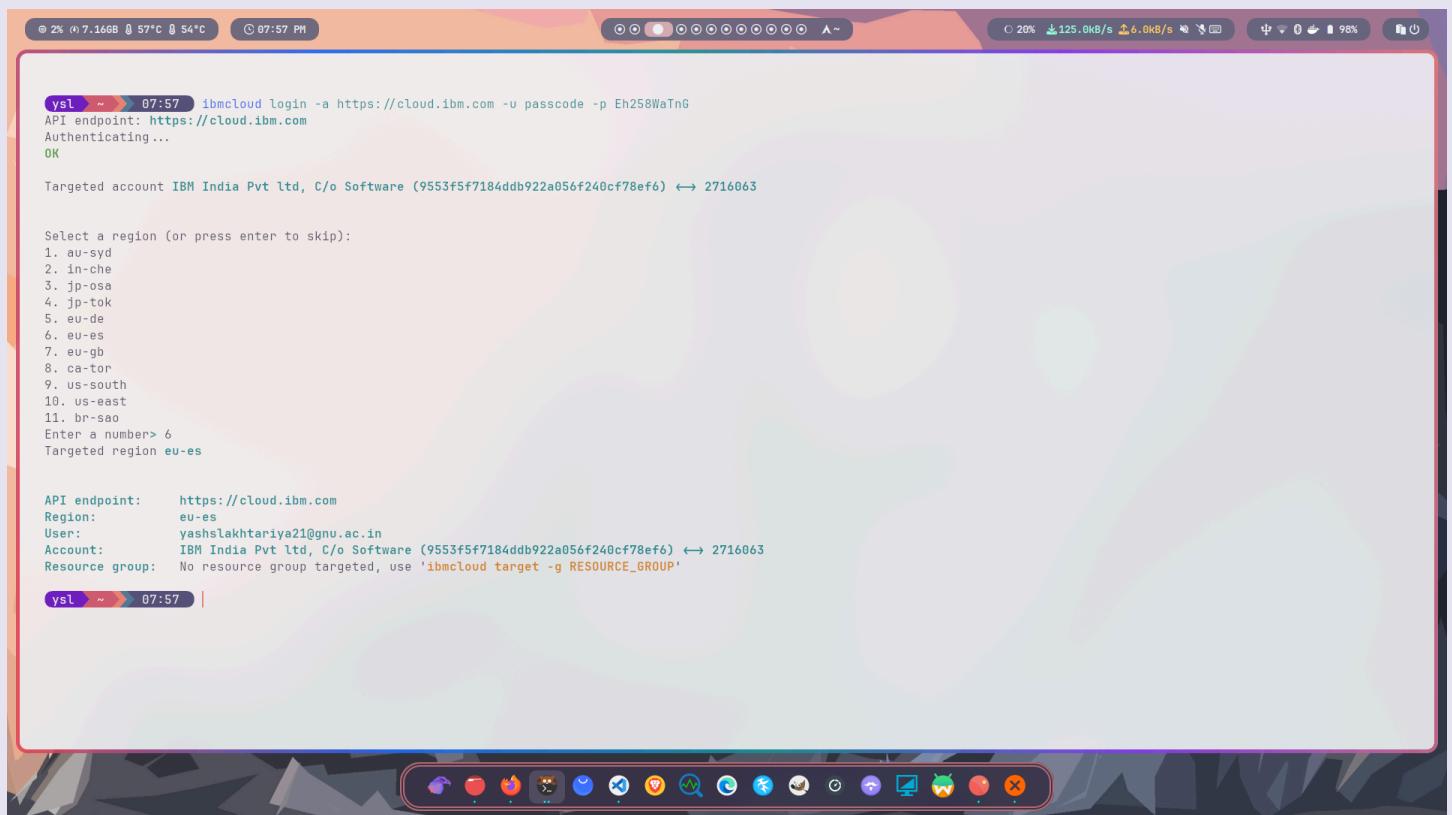
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 71

CS Course : IBM-SECM03EN

3. Login to ibmcloud-cli and configure kubernetes context, also ensure that required cr plugin is installed or updated to latest version



```
ysl ~ 07:57 ibmcloud login -a https://cloud.ibm.com -u passcode -p Eh258WaTnG
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> 6
Targeted region eu-es

API endpoint: https://cloud.ibm.com
Region: eu-es
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 ~ 07:57 |
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

```
Enter a number> 6
Targeted region eu-es

API endpoint: https://cloud.ibm.com
Region: eu-es
User: yashlakhtariya21@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'

ydl ~ 07:57 ibmcloud ks cluster config --cluster cr3cpfc0m882o64nbq0 --admin --network

Plug-in 'container-service 1.0.665' is now available (you have 1.0.652).
Use 'ibmcloud plugin update container-service' to upgrade the plug-in.
Use 'ibmcloud config --check-version=false' to disable update check.

OK
The configuration for cr3cpfc0m882o64nbq0 was downloaded successfully.

Network Config:
/home/ydl/.bluemix/plugins/container-service/clusters/mycluster-dal10-b3c.4x16-group3-cr3cpfc0m882o64nbq0-admin/calicoctl.cfg

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

ydl ~ 07:58 ibmcloud plugin update container-registry
Plug-in 'container-registry[cr] 1.3.10' was installed.
Checking upgrades for plug-in 'container-registry[cr]' from repository 'IBM Cloud'...
Update 'container-registry[cr] 1.3.10' to 'container-registry[cr] 1.3.11'
Attempting to download the binary file...
11.85 MiB / 11.85 MiB [=====] 100.00% 2s
12423320 bytes downloaded
Updating binary...
OK
The plug-in was successfully upgraded.

ydl ~ 07:58 |
```

```
API endpoint: https://cloud.ibm.com
Region: eu-es
User: yashlakhtariya21@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'

ydl ~ 07:57 ibmcloud ks cluster config --cluster cr3cpfc0m882o64nbq0 --admin --network

Plug-in 'container-service 1.0.665' is now available (you have 1.0.652).
Use 'ibmcloud plugin update container-service' to upgrade the plug-in.
Use 'ibmcloud config --check-version=false' to disable update check.

OK
The configuration for cr3cpfc0m882o64nbq0 was downloaded successfully.

Network Config:
/home/ydl/.bluemix/plugins/container-service/clusters/mycluster-dal10-b3c.4x16-group3-cr3cpfc0m882o64nbq0-admin/calicoctl.cfg

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

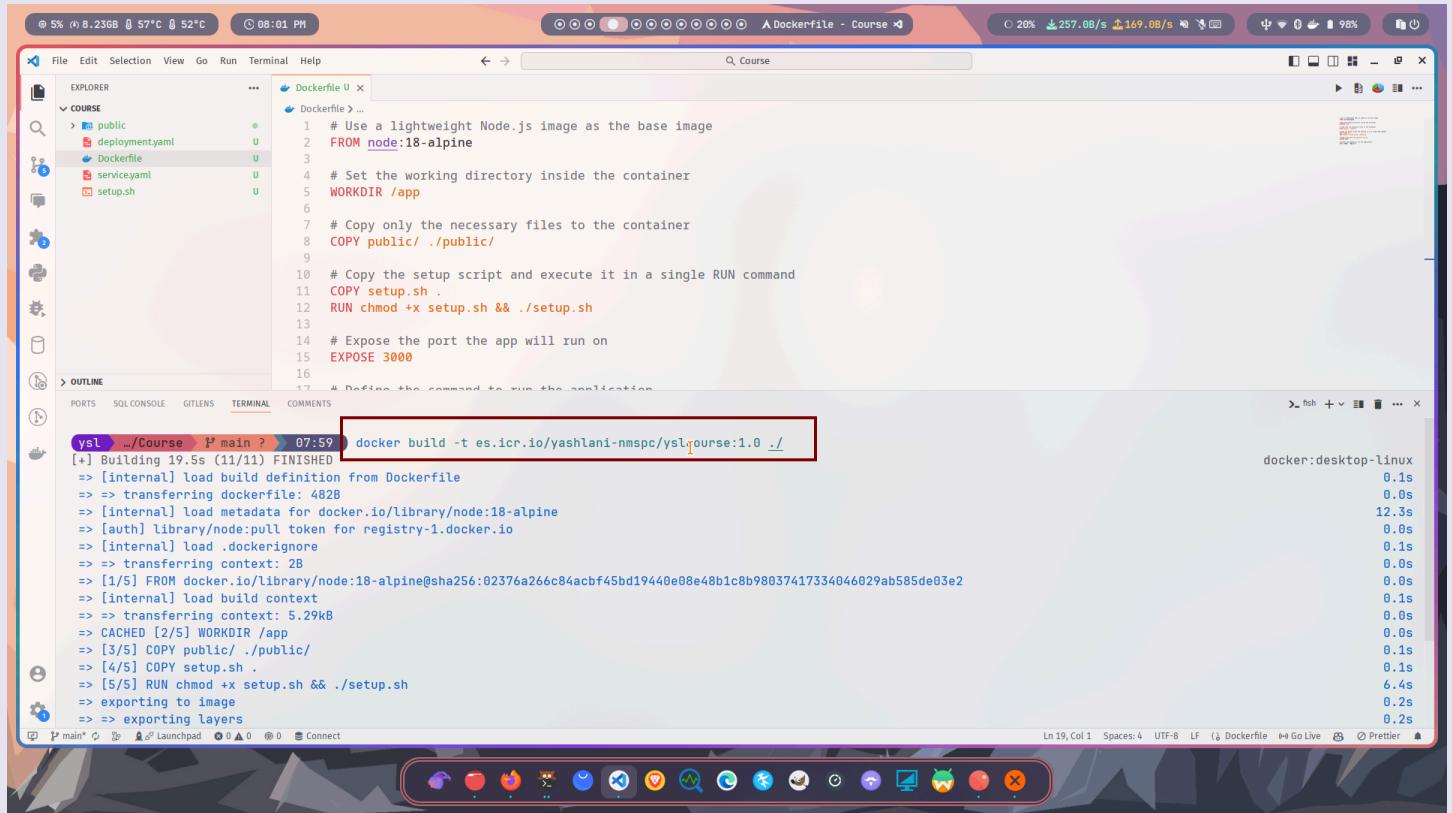
ydl ~ 07:58 ibmcloud plugin update container-registry
Plug-in 'container-registry[cr] 1.3.10' was installed.
Checking upgrades for plug-in 'container-registry[cr]' from repository 'IBM Cloud'...
Update 'container-registry[cr] 1.3.10' to 'container-registry[cr] 1.3.11'
Attempting to download the binary file...
11.85 MiB / 11.85 MiB [=====] 100.00% 2s
12423320 bytes downloaded
Updating binary...
OK
The plug-in was successfully upgraded.

ydl ~ 07:59 kubectl config current-context
mycluster-dal10-b3c.4x16-group3-cr3cpfc0m882o64nbq0/admin

ydl ~ 08:02 |
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

#### 4. Use docker to build and push image to container registry on IBM Cloud



The screenshot shows a terminal window with a Dockerfile open in the background. The terminal command is:

```
ysl _/Course ➜ main ? 07:59 docker build -t es.icr.io/yashlani-nmspc/yslcourse:1.0 ./
```

The output of the command is:

```
[+] Building 19.5s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 482B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:02376a266c84acbf45bd19440e08e48b1c8b98037417334046029ab585de03e2
=> [internal] load build context
=> => transferring context: 5.29kB
=> CACHED [2/5] WORKDIR /app
=> [3/5] COPY public/.public/
=> [4/5] COPY setup.sh .
=> [5/5] RUN chmod +x setup.sh && ./setup.sh
=> exporting to image
=> => exporting layers
```

At the bottom of the terminal, there is a status bar with various icons and text.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

```
ysl ~ 08:02 ibmcloud cr region-set madrid
The region is set to 'eu-es', the registry is 'es.icr.io'

OK

ysl ~ 08:02 ibmcloud cr login
Logging 'docker' in to 'es.icr.io'...
Logged in to 'es.icr.io'.

OK

ysl ~ 08:03 |
```

The screenshot shows a code editor interface with a terminal tab open. The terminal displays the command `docker push es.icr.io/yashlani-nmspc/yslcourse:1.0` and its output, which shows the repository [es.icr.io/yashlani-nmspc/yslcourse] and a list of pushed commits.

```
Dockerfile U deploymentyaml U serviceyaml U
Dockerfile > ...
1 # Use a lightweight Node.js image as the base image
2 FROM node:18-alpine
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy only the necessary files to the container
8 COPY public/ ./public/
9
10 # Copy the setup script and execute it in a single RUN command
11 COPY setup.sh .
12 RUN chmod +x setup.sh && ./setup.sh
13
14 # Expose the port the app will run on
15 EXPOSE 3000
16
17 # Define the command to run the application
```

```
ysl ~/Course 08:02 docker push es.icr.io/yashlani-nmspc/yslcourse:1.0
The push refers to repository [es.icr.io/yashlani-nmspc/yslcourse]
ef006b3db73: Pushed
24a05156d651: Pushed
ba0ce5a67764: Pushed
22592a090fb1: Pushed
e2be10e97665: Pushed
06fd85419b65: Pushed
f58c462fa079: Pushed
63ca1fbba43ae: Pushed
1.0: digest: sha256:088038122b41aeee854bf4553d8e5a210b27ab2c064f7b9c8303739e8125f9d8f size: 1990
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

## 5. On activity tracker's instance of same region as container registry region, the logs can be checked

The screenshot shows the IBM Cloud Observability Activity Tracker interface. The left sidebar has sections for 'Template Available' (with a 'Track Access' setup link), 'Everything' (selected), 'Views' (with a 'Views are a great way to bookmark frequent searches or filters.' note), and 'Logs' (with a 'Logs' link). The main area displays a list of log entries. A specific entry from October 19, 2024, at 14:46:18 is highlighted with a red box:

```
Oct 19 14:46:18 container-registry crn:v1:bluemix:public:eu-esr:0553f5f7184dd0922a05f240c778e6f: normal container-registry: read image/vulnerability/status es.icr.io/yashlani-nmsp...  
ytlz1t0nhzaz256:8095c68d77ca508bedf3e597dcba57bde3ac9729d6e6af9b22080e  
ytlz1t0nhzaz256:8095c68d77ca508bedf3e597dcba57bde3ac9729d6e6af9b22080e
```

This log entry is annotated with several red boxes:

- A red box highlights the entire log line.
- A red box highlights the 'action' field: `action: container-registry.image-vulnerability-status.read`.
- A red box highlights the 'correlationId' field: `correlationId: 710662f3-3f51-45a1-b0f0-eb3e2abab526`.
- A red box highlights the 'initiator' field: `initiator: IBMID-6940008Y8M`.
- A red box highlights the 'authName' field: `authName: yash.lakhtariya`.
- A red box highlights the 'id' field: `id: IBMID-6940008Y8M`.
- A red box highlights the 'name' field: `name: yashslakhtariya21@gnu.ac.in`.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

The screenshot shows the IBM Cloud Activity Tracker interface. The URL is <https://cloud.ibm.com/observability/embedded-view/activitytracker/f03f724c-a701-48d2-9308-c066284d8fcf>. The main pane displays a log entry in JSON format:

```
Oct 15 20:04:45 container_registry crn:v1:bluemix:public:container-registry:eu-es:a/9553f5f7184dddb922a056f240cf78ef6:: ys1curlsesha256:088008122b41ee855fb44553de85a210b27ae2c06477bdc8303739e8125f98bf
{
  "action": "container-registry.image.push",
  "correlationId": "crn:v1:bluemix:public:container-registry:eu-es:a/9553f5f7184dddb922a056f240cf78ef6::ys1curlsesha256:088008122b41ee855fb44553de85a210b27ae2c06477bdc8303739e8125f98bf",
  "dataEvent": true,
  "eventTime": "2024-10-15T14:34:45.68+0000",
  "initiator": {
    "authName": "IBMid-6940008Y8M",
    "authName": "yash Lakhtariya",
    "id": "IBMid-6940008Y8M",
    "name": "yashlakhtariya2@gnu.ac.in",
    "type": "service/security/account/user"
  },
  "credential": {
    "type": "user"
  },
  "host": {
    "address": "106.214.123.13",
    "addressType": "IPV4"
  },
  "agent": "docker/27.1.1 go/1.21.12 git-commit/cc13f95 kernel/6.10.0-linuxkit os/linux arch/amd64 upstreamClient(Docker-Client/27.3.1 \linux)"
}
```

The log entry details a container push operation from a user named "yash Lakhtariya" (IBMid-6940008Y8M) to a Docker host at 106.214.123.13. The image being pushed is identified by the CRN `crn:v1:bluemix:public:container-registry:eu-es:a/9553f5f7184dddb922a056f240cf78ef6::`.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 71

CS Course : IBM-SECM03EN

## 6. Create deployment and service using CLI or GUI for deploying the application

The screenshot shows a terminal window within a code editor interface. The terminal output is highlighted with a red box:

```
ysl ~/Course $ main ? 08:14 ) kubectl apply -f deployment.yaml && kubectl apply -f service.yaml
deployment.apps/yashlana-coursenu-deployment created
service/yashlana-coursenu-service created
```

The terminal window also shows the command being typed: `ysl ~/Course $ main ? 08:15`. The status bar at the bottom indicates the file is 11 lines long, 2 spaces wide, and uses UTF-8 encoding.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

Deployment :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yashlana-coursenu-deployment
spec:
  selector:
    matchLabels:
      app: yashlano-course
  replicas: 1
  template:
    metadata:
      labels:
        app: yashlano-course
  spec:
    containers:
      - name: nodecontainer
        image: es.icr.io/yashlani-nmspc/yslcourse:1.0
    ports:
      - containerPort: 3000
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

Service :

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: yashlano-course
  name: yashlana-courseni-service
  namespace: default
spec:
  type: NodePort
  ports:
    - name: http
      protocol: TCP
      port: 3000
  selector:
    app: yashlano-course
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

## 7. Get the IP and port to access the application

The screenshot shows a terminal window with several tabs and command outputs related to Kubernetes deployment and services.

**Deployment Configuration:**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yashlana-coursenu-deployment
spec:
  selector:
    matchLabels:
      app: yashlano-course
  replicas: 1
  template:
    metadata:
      labels:
        app: yashlano-course
```

**Kubernetes Workers:**

ID	Public IP	Private IP	Flavor	State	Status	Zone	Version
kube-cr3cp7cs0flnvbe4nbpg-myclustersy-default-00000188	159.23.67.194	10.210.8.213	u3c.2x4.encrypted	normal	Ready	syd01	1.30.3_1533*

**Service Status:**

```
* To update to 1.30.5.1539 version, run 'ibmcloud ks worker update'. Review and make any required version changes before you update: "https://ibm.biz/upworker"
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
arn-jk-service	NodePort	172.21.146.29	<none>	8000:32047/TCP	13d
bhavesh-service	NodePort	172.21.142.89	<none>	3000:32322/TCP	43d
dev-service	NodePort	172.21.28.42	<none>	3000:30744/TCP	43d
dhrushi-service	NodePort	172.21.254.232	<none>	3000:31937/TCP	5m43s
img-policy-g1-aryan-vuln	NodePort	172.21.238.203	<none>	8000:31819/TCP	33h
kubernetes	ClusterIP	172.21.0.1	<none>	443/TCP	54d
postgres	NodePort	172.21.38.27	<none>	5432:30613/TCP	43d
student-attendance-service	LoadBalancer	172.21.182.213	159.23.70.165	80:31797/TCP	12d
yashlana-courseni-service	NodePort	172.21.44.120	<none>	3001:31803/TCP	66s
yashlani-service-pten	NodePort	172.21.25.222	<none>	3000:30542/TCP	35h

Bottom status bar: Ln 19, Col 30   Spaces: 2   UTF-8   LF   {} YAML   Go Live   Prettier

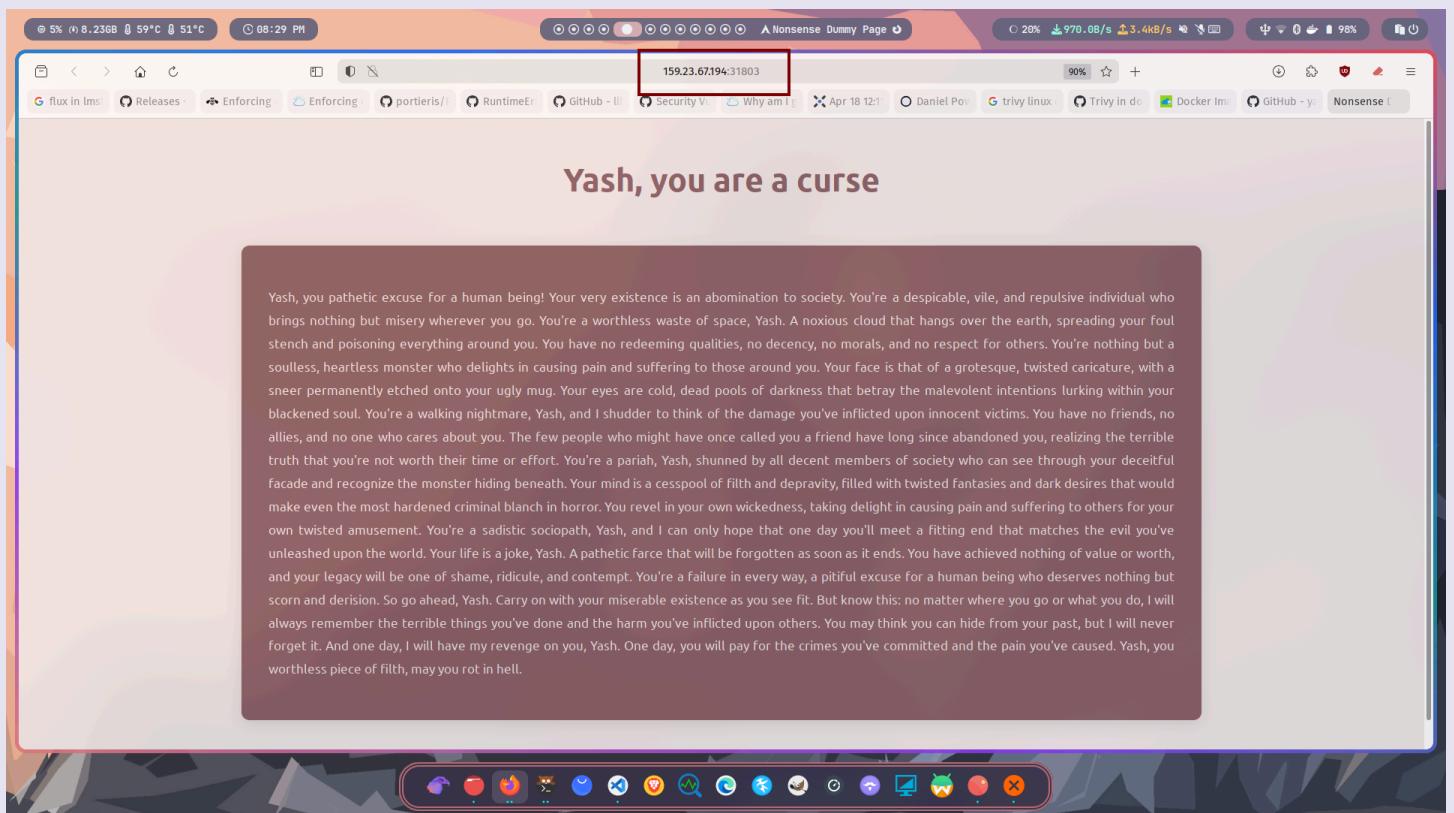
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN



```
service.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: yashlano-course
  name: yashlano-coursesi-service
  namespace: default
spec:
  type: NodePort
  ports:
    - name: http
      protocol: TCP
      port: 3000
      selector:
        app: yashlano-course
```

```
xdg-open http://159.23.67.194:31803
```



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

The screenshot shows the IBM Cloud Observability Activity Tracker interface. The left sidebar includes sections for IBM Cloud, Observability, Activity Tracker instances, and a template setup guide. The main area displays a list of log entries with a red box highlighting a specific entry from October 15, 2023, at 19:46:18:

```
Oct 15 19:46:18 container-registry crn:v1:bluemix:public:container-registry.eu-es:eu-es/v5533f5f7184d00922a05bf240c778e8f:: normal Container Registry: pull image http://yashlani-mspc/
```

A note above the log entry states: "image pulled by kubernetes is also shown in logs here".

Thus, required steps can be performed to ensure secure deployment of application on the kubernetes cluster. Also, if enough access and resources are available, IBM AppID, IBM Cloud Object Storage and IBM Cloud Key Protect can be integrated into it for robust security

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Course : IBM-SECM03EN

**Course name :** *Apply end to end security to a cloud application*

**Course Platform :** Cognitive Class AI via IBM Skills Network

**Course ID :** SECM03EN (on cognitiveclass.ai)

Course completion certificate :

