**Deploy a NodeJS Application in a Docker Container**

**Task 1: Launching and Connecting to your EC2 instance**

Launch an EC2 instance in us-east-1 region

-Ubuntu AMI

-Key pair: vockey

**Step 2: Connect to the instance from the console using EC2 Instance Connect**

Click on Launch instance

Click on connect to your instance and then click on connect

**Task 2: Install and set up Dependencies**

Run the following commands to install dependencies

```
sudo apt update
sudo apt install -y nodejs npm
sudo apt install -y docker.io
sudo systemctl enable docker
sudo systemctl start docker
```

**Task 3: Create the Project Files**

Run the following commands to create and change to Project Repository

```
mkdir node-docker-app
cd node-docker-app
```

Create app.js File

```
nano app.js
```

Put the below code in app.js

```
const http = require("http");
const server = http.createServer((req, res) => {
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello from Node.js!");
```

```
});
server.listen(3000);
```

Initialize NodeJS Project

```
npm init -y
```

Create a .dockerignore File

```
nano .dockerignore
```

Write the below lines in .dockerignore file

```
node_modules
npm-debug.log
```

Create Dockerfile

```
nano Dockerfile
```

Write the below commands in Dockerfile

```
FROM node:18
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "app.js"]
```

**Task 4: Build and Run Docker Container and Access the Application**

Build Docker Image using following command (Admin privilege is important)

```
sudo docker build -t hello-world-node .
```

Run Docker Container using following command (Admin privilege is important)

```
sudo docker run -d -p 3000:3000 hello-world-node
```

Edit Security Group of EC2 Instance to add Inbound Rules allowing TCP 3000 Port Requests from anywhere-IPv4

Select Type : All

TCPSource: Anywhere

Access the Application by opening the following URL (Copy and paste the EC2 Public IP)

Go to EC2

Click on instances , select your instance and copy your public id

EC2-Public-IP:3000

---

9) Implement a distributed application on Hadoop framework to count word frequency with MapReduce

**Task 1: Create S3 Bucket for I/O**

Create an S3 Bucket - Give a name to it and click on create bucket Please remember the bucket name

upload the input File and WordCount Runnable JAR File

Download the file from link

https://myawsbucket-hadoop-1602-21-733-102.s3.us-east-1.amazonaws.com/WordCount.jar

Create a text file with file name as **input** and add the following into it

File name : **input**

**Content in file :**

Hello

World

Hello

Hi

Upload these above input file and file downloaded

**Task 2: Create an EMR Cluster**

Application bundle - Core Hadoop

-EC2 Key Pair: vockey

-EMR Service Role: EMR_DefaultRole

-EC2 Instance Profile: EMR_EC2_DefaultRole

Click on create cluster

If there is Validation Error, stop and start the Lab and try again

Cluster creation can take a long time (6 minutes approx)

**Task 3: Add Step in Cluster**

Add a Step in created Cluster

-Select JAR File (WordCount.jar)uploaded to Bucket

JAR Location Select the jar file from browse s3

-Give Arguments as follows

s3://BUCKET_NAME/input.txt

s3://BUCKET_NAME/output

Replace BUCKET_NAME with your bucket name

Click on Add step

Step completion can take a long time

[Wait until the  all the log files get displayed , refresh it every 2min to check]

**Task 4: Open Output**

**Open s3 and click on  bucket created by you**

**You should see output folder created and inside that there are 4 files . download each file to see the results**