CS457
DEVOPS TASK - 2

Submitted to:

Dr Uma S

Submitted by:

TEAM 8

Akshaya Reddy 18BCS017

Bhavya Tripathi 18BCS019

Harshitha M 18BCS032

Meghana N 18BCS053

S Namratha 18BCS083

Vinita Yadav 18BCS109

Deepanshu Sachdeva 18BCS114

DEVOPS ASSIGNMENT
TEAM 8

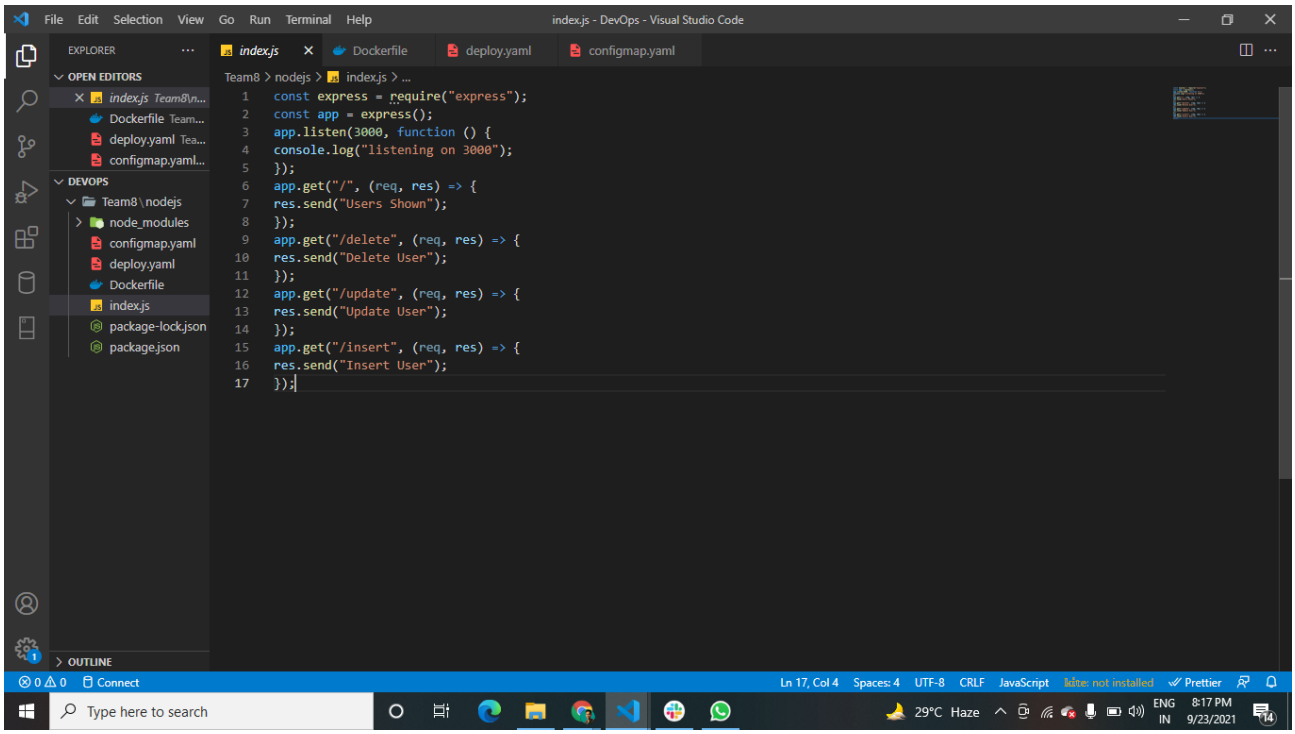
TASK1
Part B

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

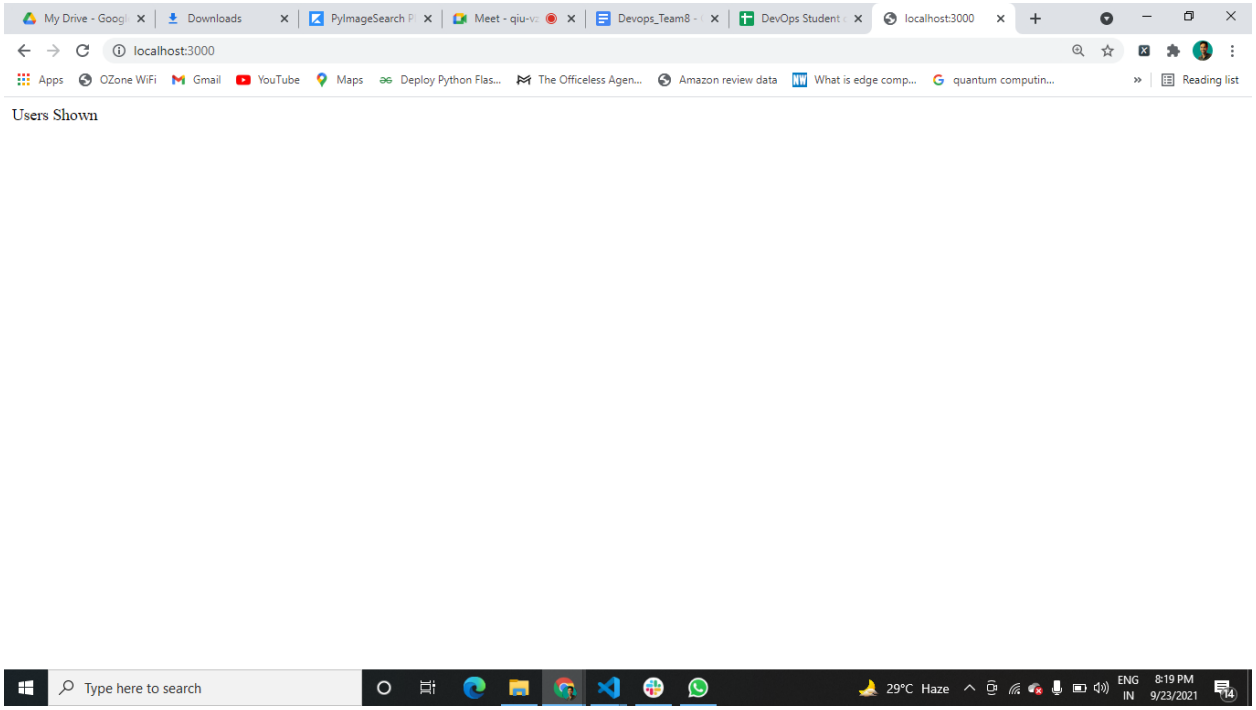
You can find the docker file here :
https://hub.docker.com/r/deepanshusachdeva5/nodejs-starter

Github link -
https://github.com/deepanshusachdeva5/CS457_DEVOPS/tree/main/Assignment2

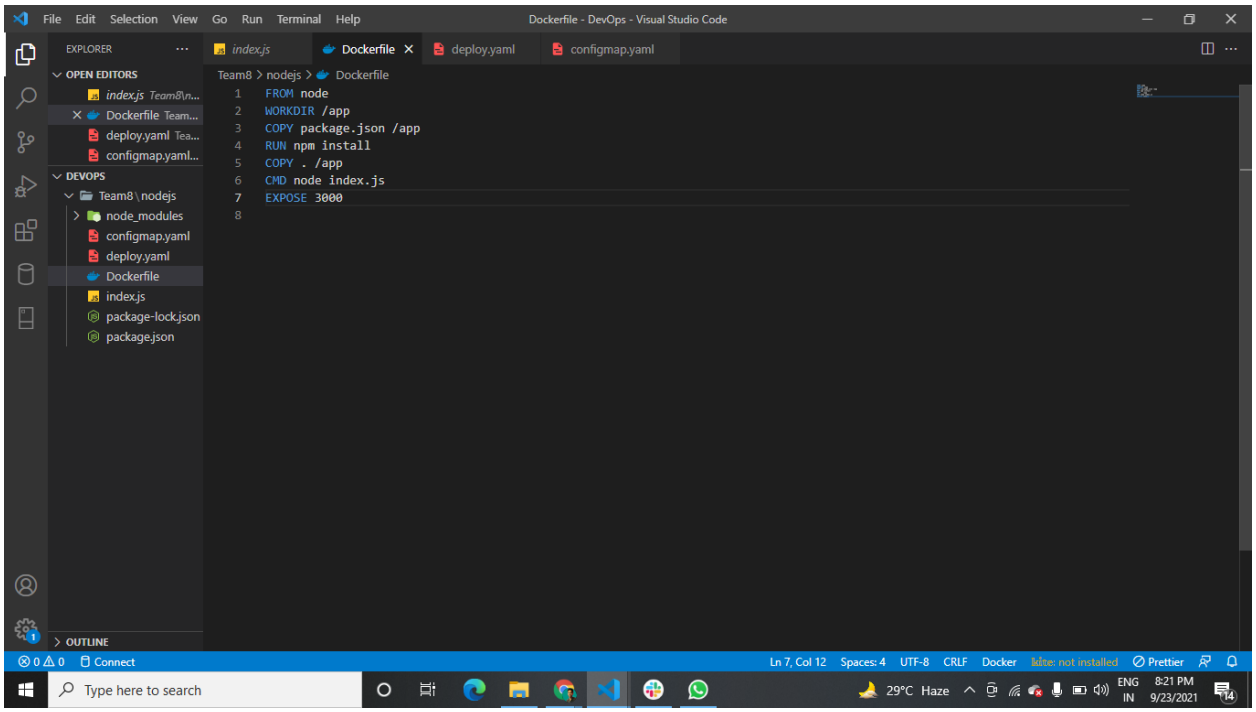
STEP1: NodeJs App



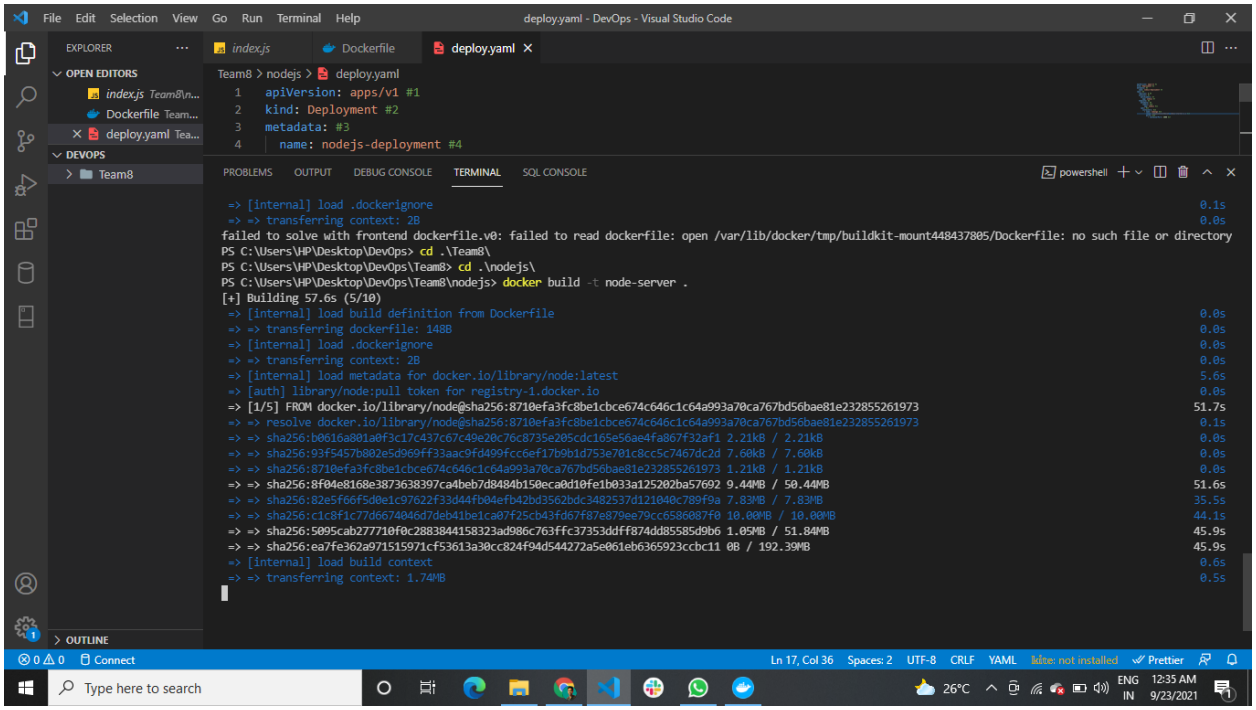
STEP2: Testing NodeJs App using npm start



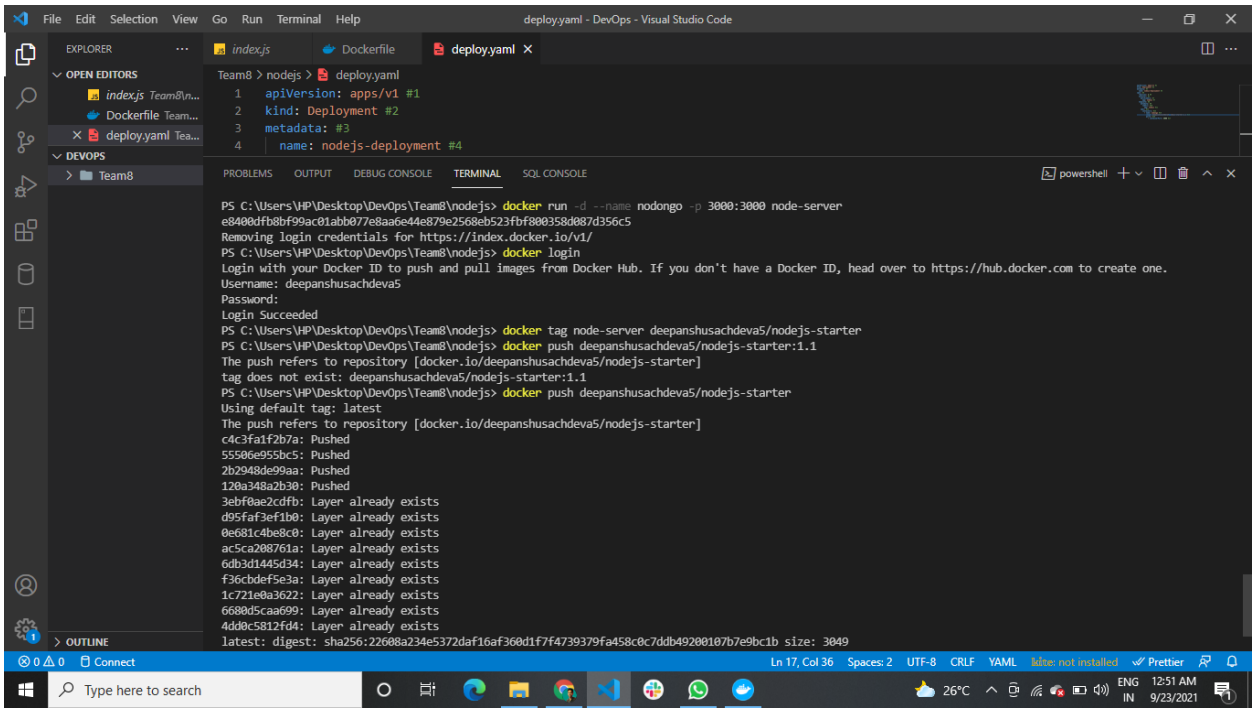
STEP3: Writing Dockerfile



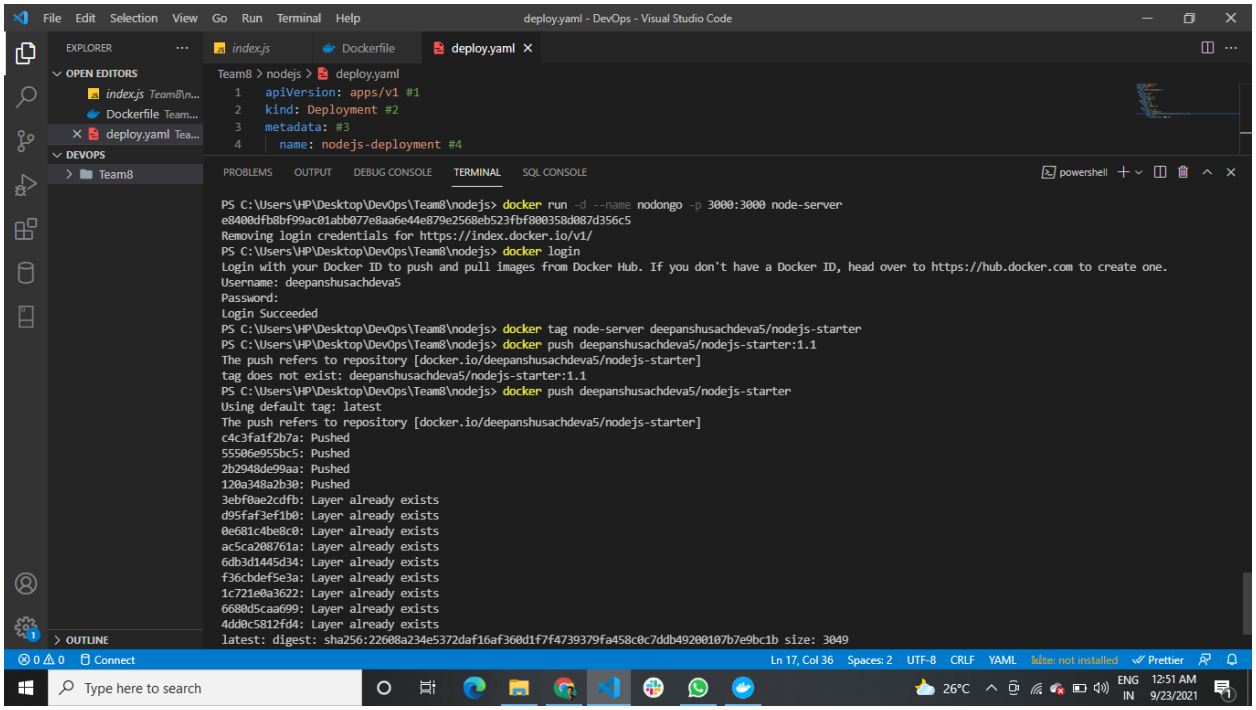
STEP4: Building Docker Image using docker build -t node-starter



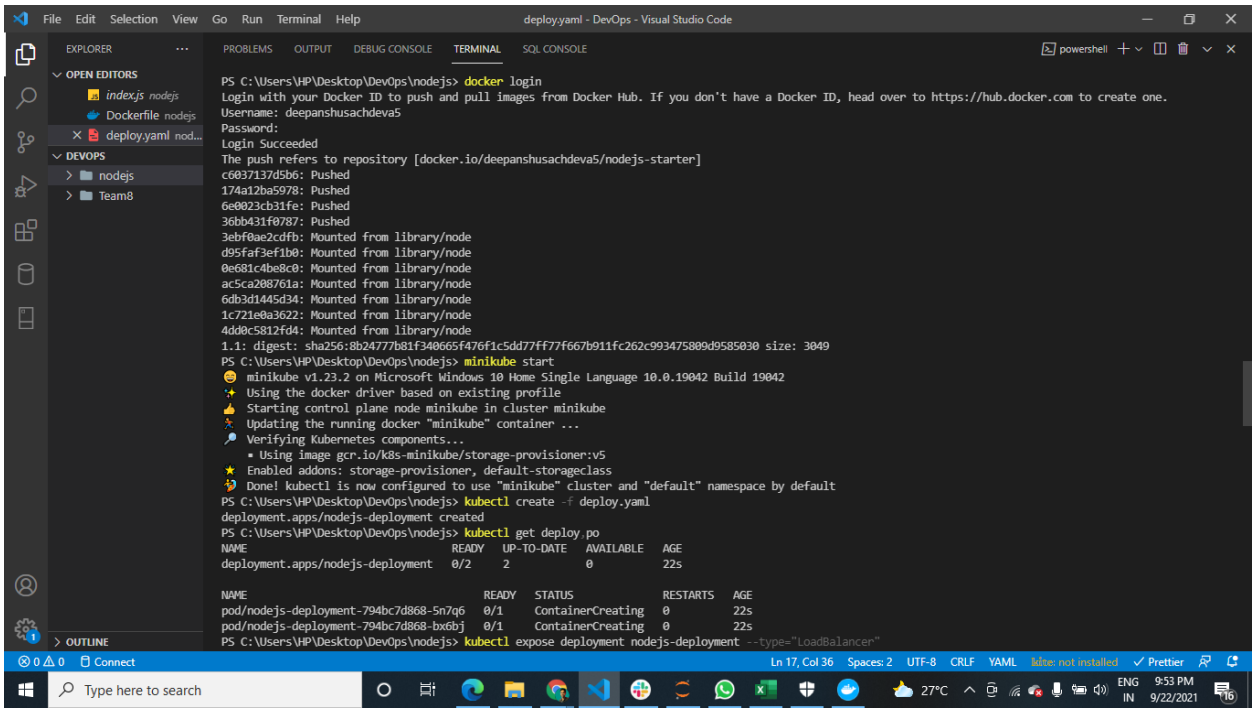
STEP5: Running Docker Image



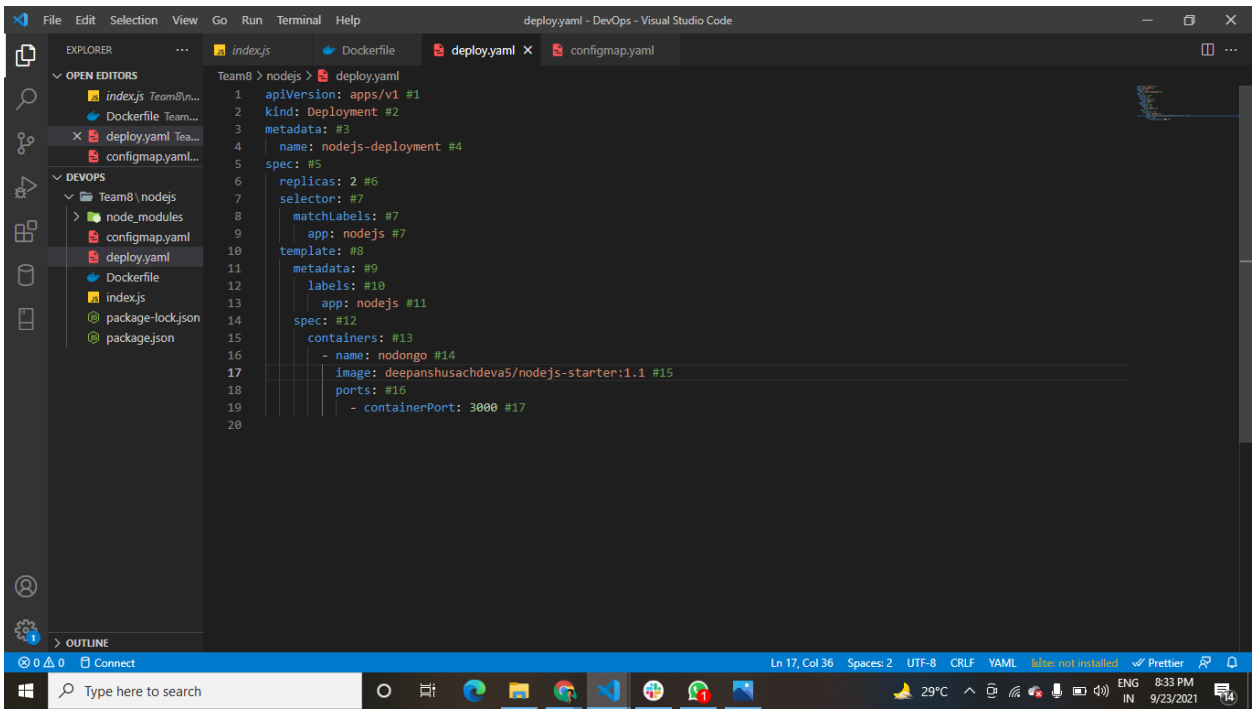
STEP6: Pushing image to docker hub after tagging



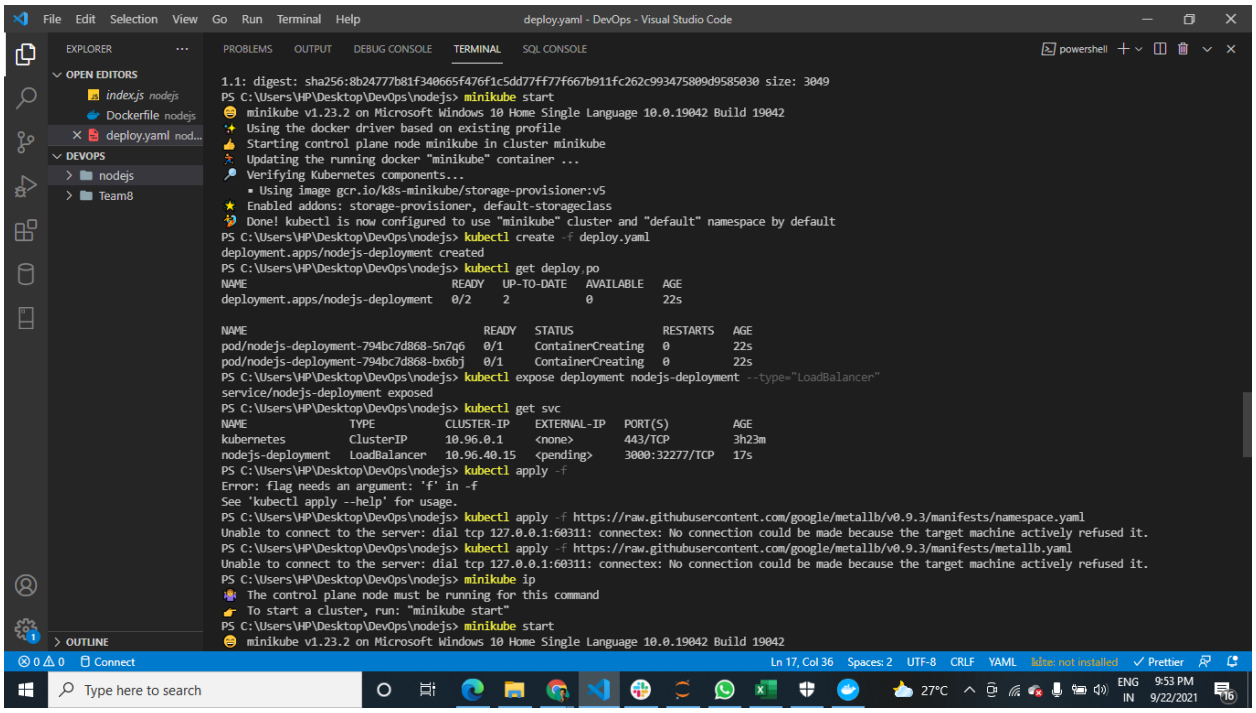
STEP7: Minikube Start



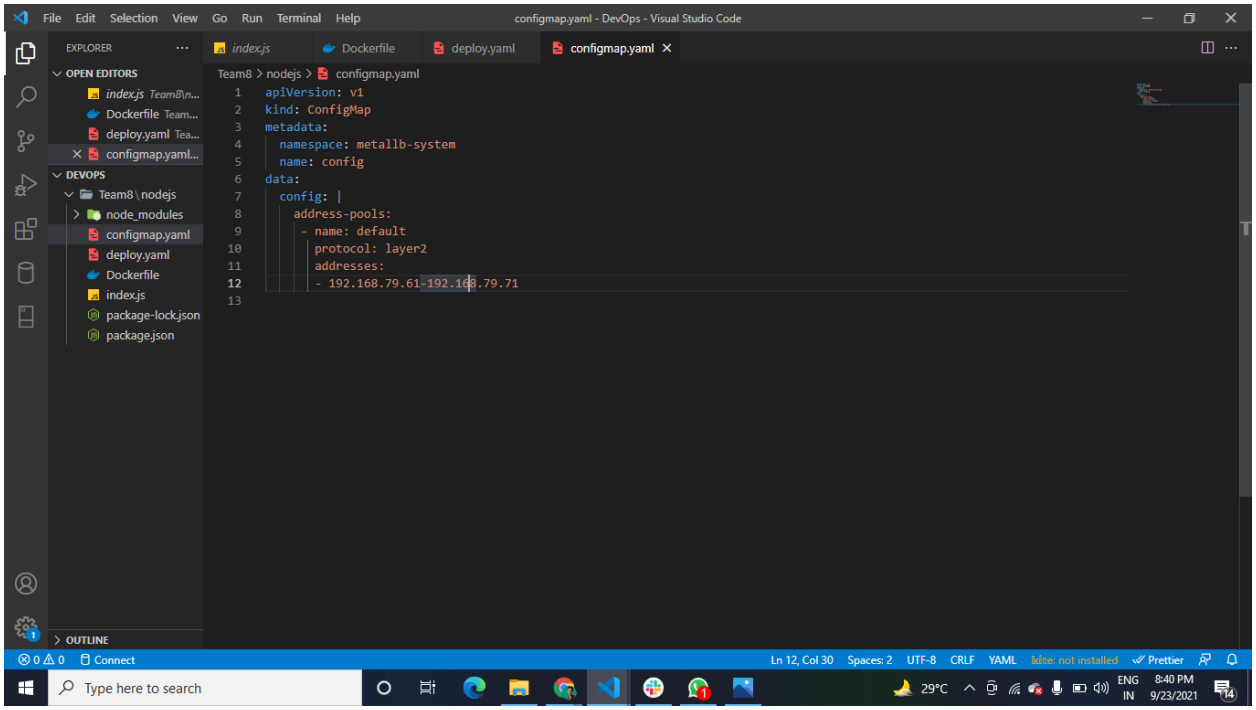
STEP8: Creating deploy.yaml file



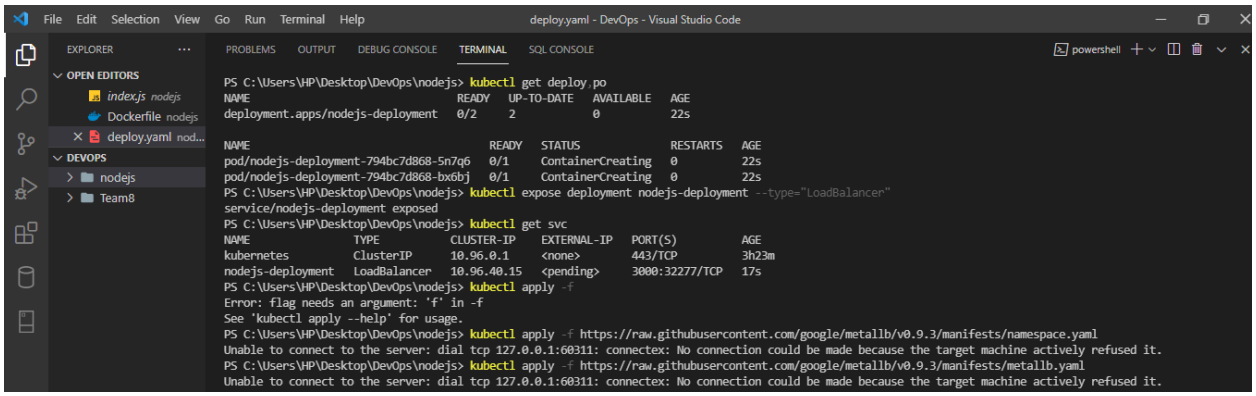
STEP9: Kubectl -f deploy.yaml



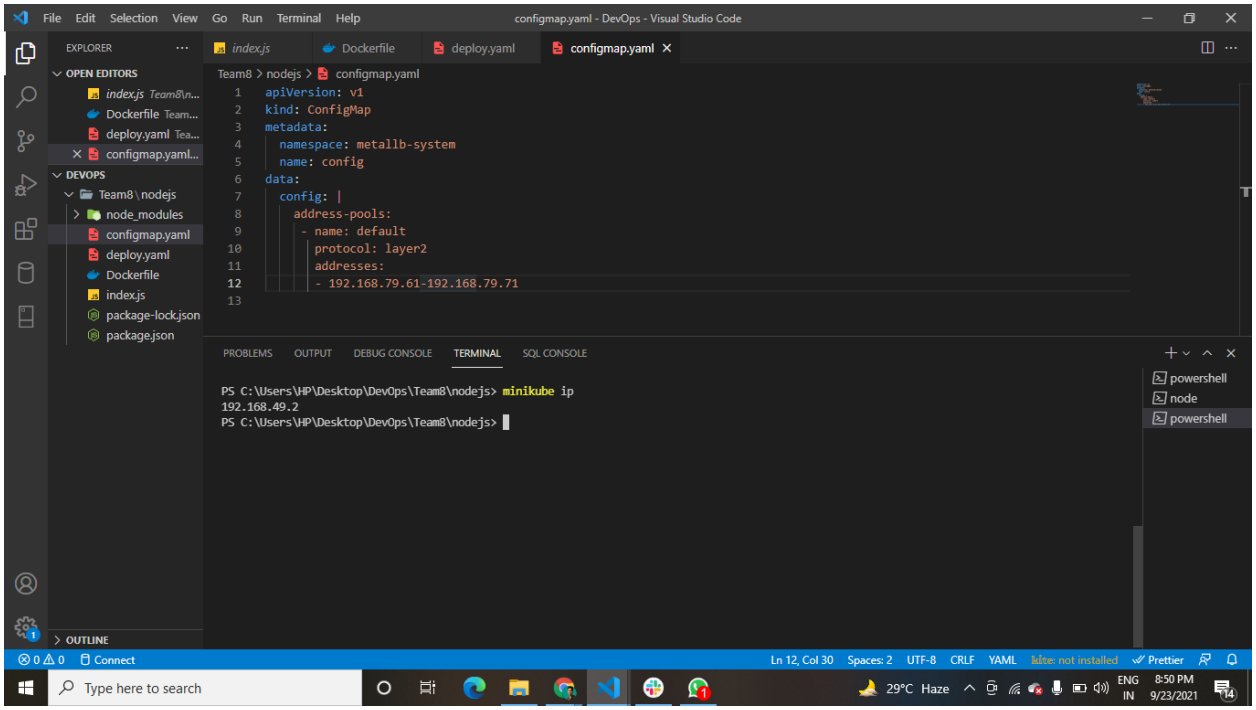
STEP10: configmap.yaml



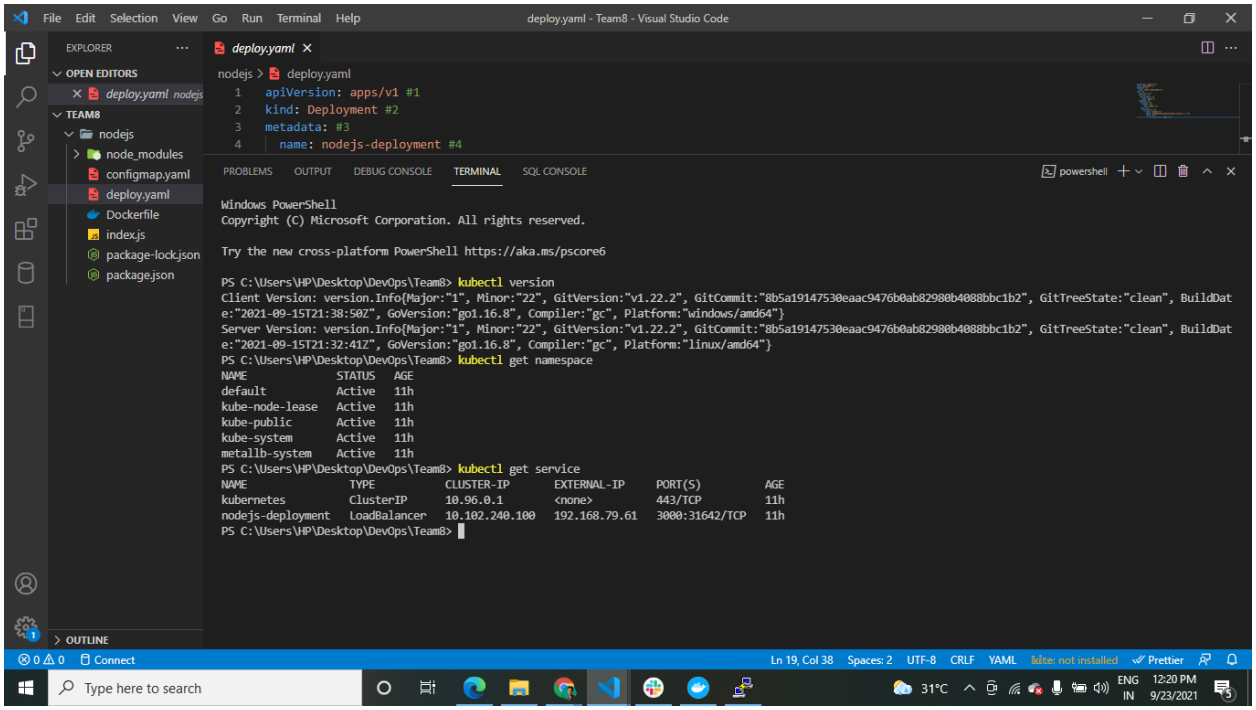
STEP11: Deploying on kubernetes using minikube



STEP12:Checking minikube ip



STEP13: Verifying external IP


File Edit Selection View Go Run Terminal Help
deploy.yaml - Team8 - Visual Studio Code
EXPLORER
OPEN EDITORS
deploy.yaml nodejs
TEAM8
nodejs
node_modules
configmap.yaml
deploy.yaml
Dockerfile
index.js
package-lock.json
package.json
OUTLINE
deploy.yaml
nodejs > deploy.yaml
1 apiVersion: apps/v1 #1
2 kind: Deployment #2
3 metadata: #3
4 name: nodejs-deployment #4
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\HP\Desktop\DevOps\Team8> kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.2", GitCommit:"8b5a19147530eaac9476b0ab82980b4088bbc1b2", GitTreeState:"clean", BuildDate:"2021-09-15T21:38:50Z", GoVersion:"go1.16.8", Compiler:"gc", Platform:"windows/amd64"}
Server Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.2", GitCommit:"8b5a19147530eaac9476b0ab82980b4088bbc1b2", GitTreeState:"clean", BuildDate:"2021-09-15T21:32:41Z", GoVersion:"go1.16.8", Compiler:"gc", Platform:"linux/amd64"}
PS C:\Users\HP\Desktop\DevOps\Team8> kubectl get namespace
NAME STATUS AGE
default Active 11h
kube-node-lease Active 11h
kube-public Active 11h
kube-system Active 11h
metallb-system Active 11h
PS C:\Users\HP\Desktop\DevOps\Team8> kubectl get service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 11h
nodejs-deployment LoadBalancer 10.102.240.100 192.168.79.61 3000:31642/TCP 11h
PS C:\Users\HP\Desktop\DevOps\Team8>
0 0 Connect
Ln 19, Col 38 Spaces: 2 UTF-8 CRLF YAML kube: not installed Prettier
Type here to search
31°C
ENG IN
12:20 PM 9/23/2021

DevOps Assignment - 2

Ansible Playbook Exercise

Set-Up:

- Installing ansible.
- Creating a remote server using AWS EC2 Instance.
  Chosen platform: Red Hat Enterprise Linux with High Availability.
- Connected to the server using ssh client.

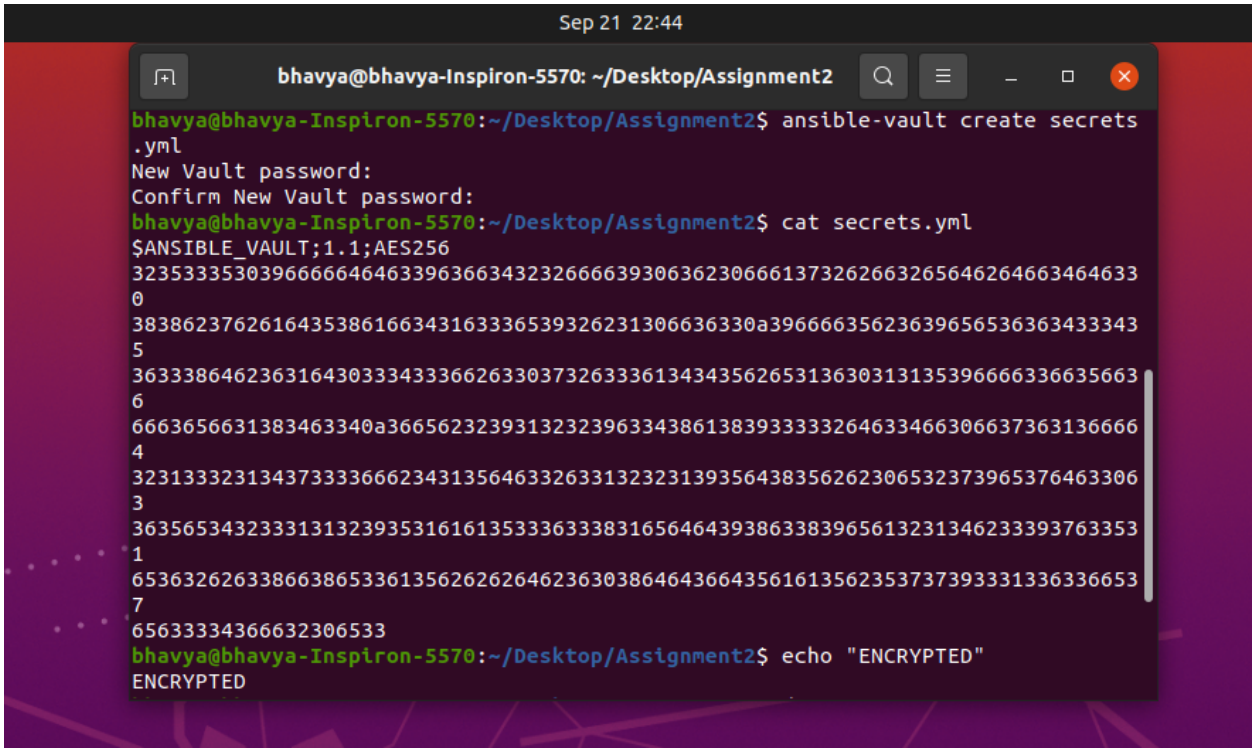
Step 1: Configuring Git login

Using username and security token.

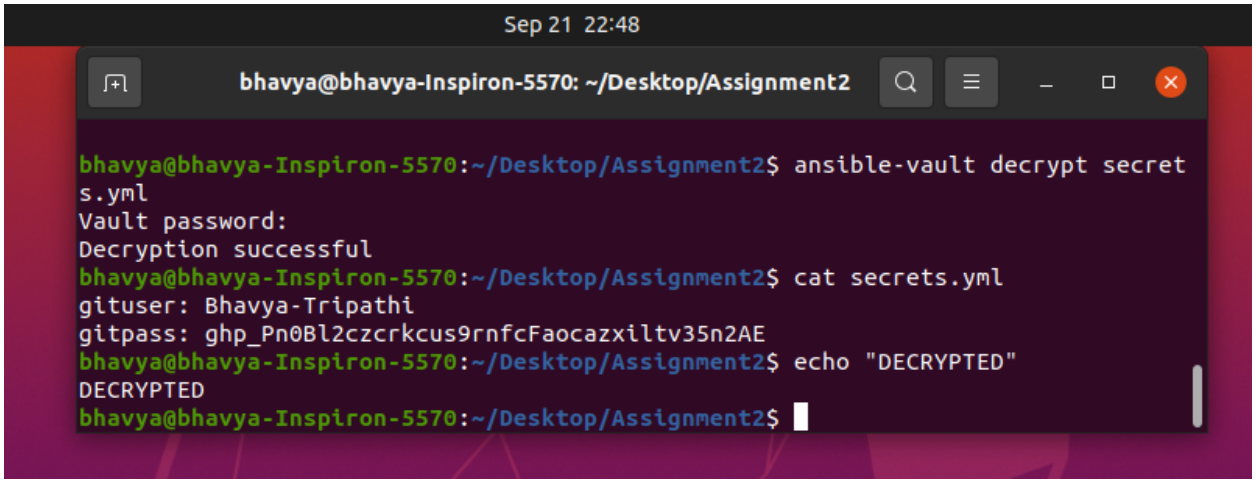
- `https://user:token@github.com/path`

Step 2: Creating Ansible vault to Store the Git username and token

Creating a vault and setting a vault password and creating a file called `secrets.yml` that stores the GitHub username and security token.
The `secrets.yml` file will be encrypted.



We can decrypt the file using the following commands:



Step 3: The Ansible Git Example Playbook

Here we have created a nodejs app and uploaded it in a private GitHub repository. Then we create the Ansible playbook. (`gitexample.yml`)

The GitHub username and token we created in the `secrets.yml` file.

```yaml
- name: Install and Launch the Simple NodeJS Application
  hosts: nodeserver
  vars_files:
     - secrets.yml
  vars:
     - destdir: /apps/SampleNodeApp
  tasks:

     - name : install Node and NPM
       become: yes
       register: ymrepo
       yum:
         name: nodejs
         state: latest

     - name : validate the nodejs installation
       debug: msg="Installation of node is Successfull"
       when: ymrepo is changed

     - name: Version of Node and NPM
       shell:
          "npm -v && node -v"
       register: versioninfo

     - name: Version Info
       debug:
         msg: "Version info {{ versioninfo.stdout_lines }}"
       when: versioninfo is changed

     - name: Download the NodeJS code from the GitRepo
       become: yes
       git:
          repo: 'https://{{gituser}}:{{gitpass}}@github.com/Bhavya-Tripathi/DevOpsAssignment2.git'
          dest: "{{ destdir }}"

     - name: Change the ownership of the directory
       become: yes
       file:
         path: "{{destdir}}"
         owner: "ec2-user"
       register: chgrpout

     - name: Install Dependencies with NPM install command
       shell:
          "npm install"
       args:
          chdir: "{{ destdir }}"
       register: npminstlout

     - name: Debug npm install command
       debug: msg='{{npminstlout.stdout_lines}}'


     - name: Start the App
       async: 10
       poll: 0
       shell:
          "(node index.js > nodesrv.log 2>&1 &)"
       args:
         chdir: "{{ destdir }}"
       register: appstart

     - name: Validating the port is open
       tags: nodevalidate
       wait_for:
         host: "localhost"
         port: 3002
         delay: 10
         timeout: 30
         state: started
         msg: "NodeJS server is not running"
```

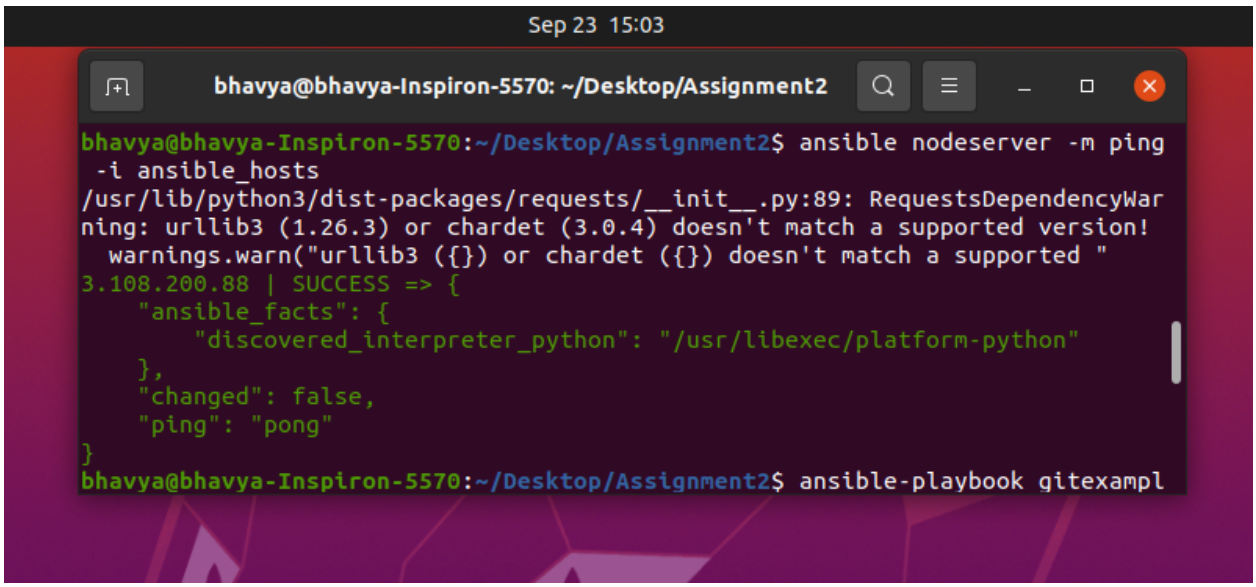
Step 4: Launch the Playbook with Ansible Git

Now we launch the playbook using the ansible-playbook command

```
ansible-playbook gitexample.yml --ask-vault-pass
```

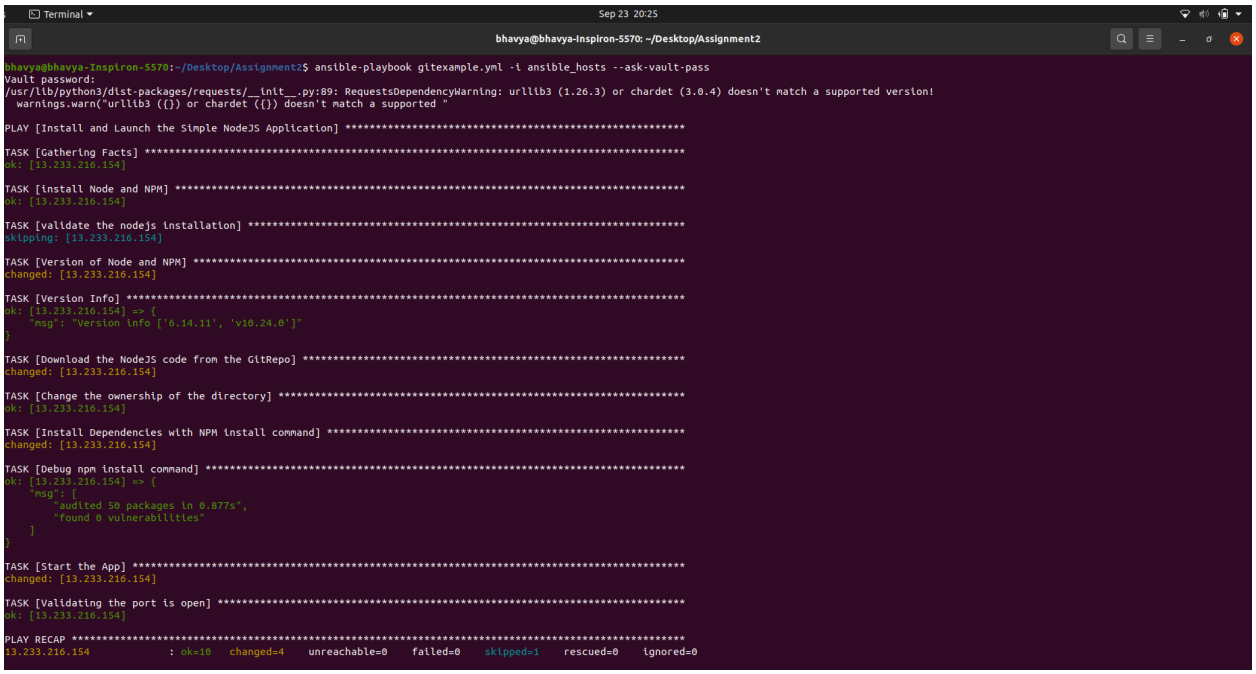
Method:

First we check if the hostgroup is reachable using the following command:
(here, nodeserver is the name of our hostgroup)

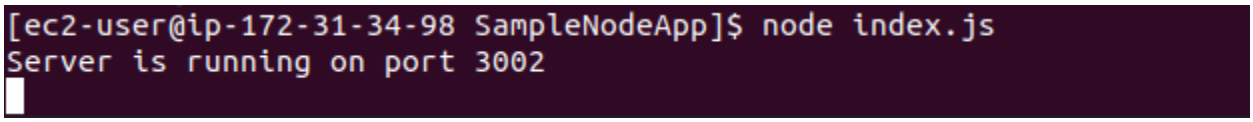


Ansible_hosts is an inventory file that contains the name of the host group, public IP address of the host server etc.

Now we launch the playbook:



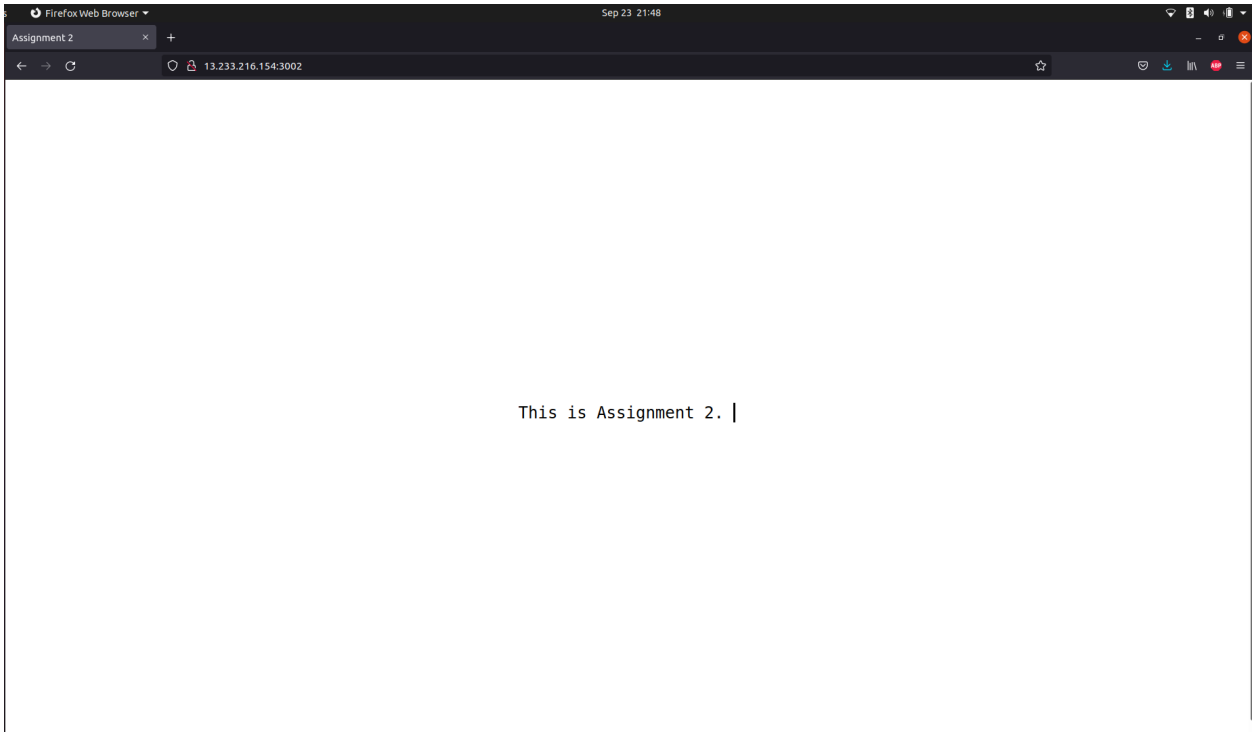
The playbook ran successfully.
This means our github repository has been cloned to our server in the `/apps/SampleNodeApp` directory.



The server is running successfully!

Step 5: Validate the Deployment

The remote server here is 13.233.216.154, now we can access the URL via http://13.233.216.154:3002.



Our Node Website has been deployed successfully!

Assignment-3 Add users to EC2 instance with SSH access using ansible

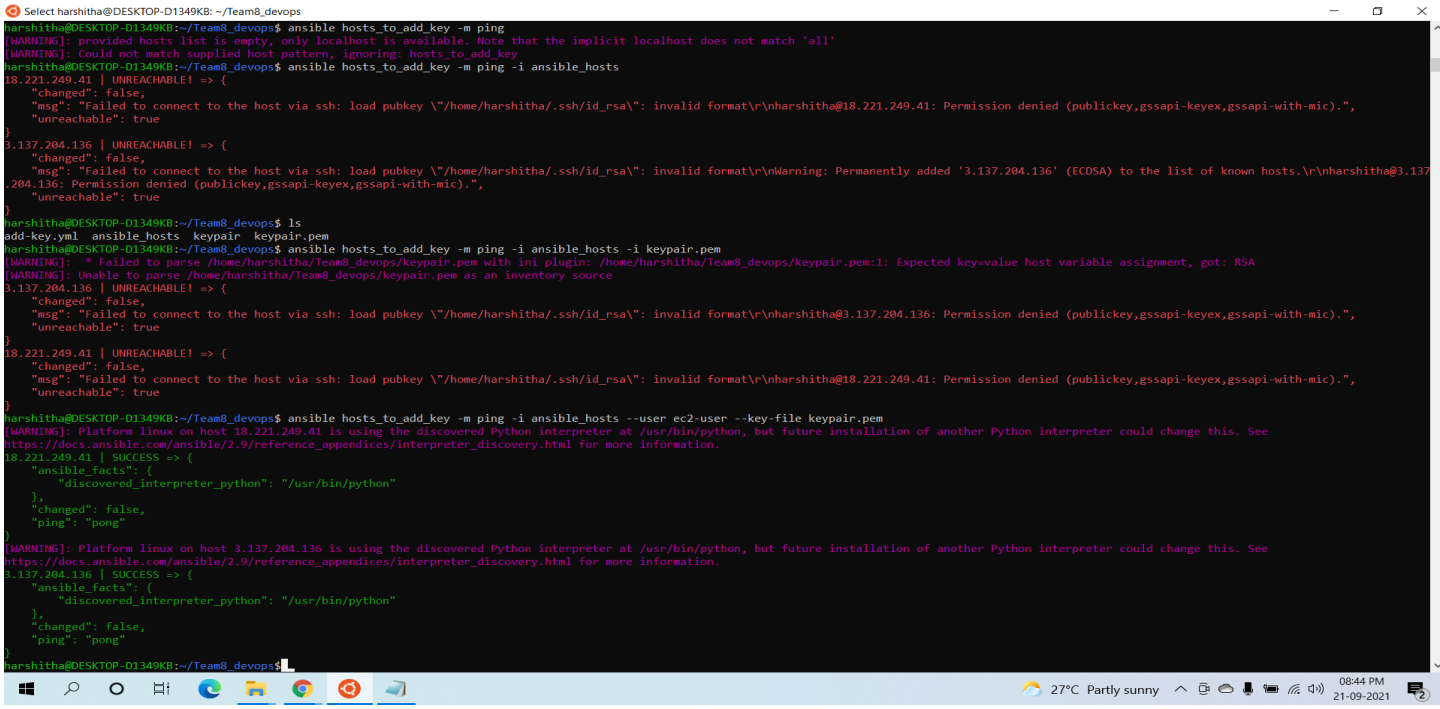
The task consists of 2 steps:
1.Create a new user account on all these EC2 instances for the user.
2.Copy the user's SSH public key into the newly created account for them to login without a password

1. Configuring the host file with multiple hosts

Creating a host group named hosts_to_add_key with host ip address,username.

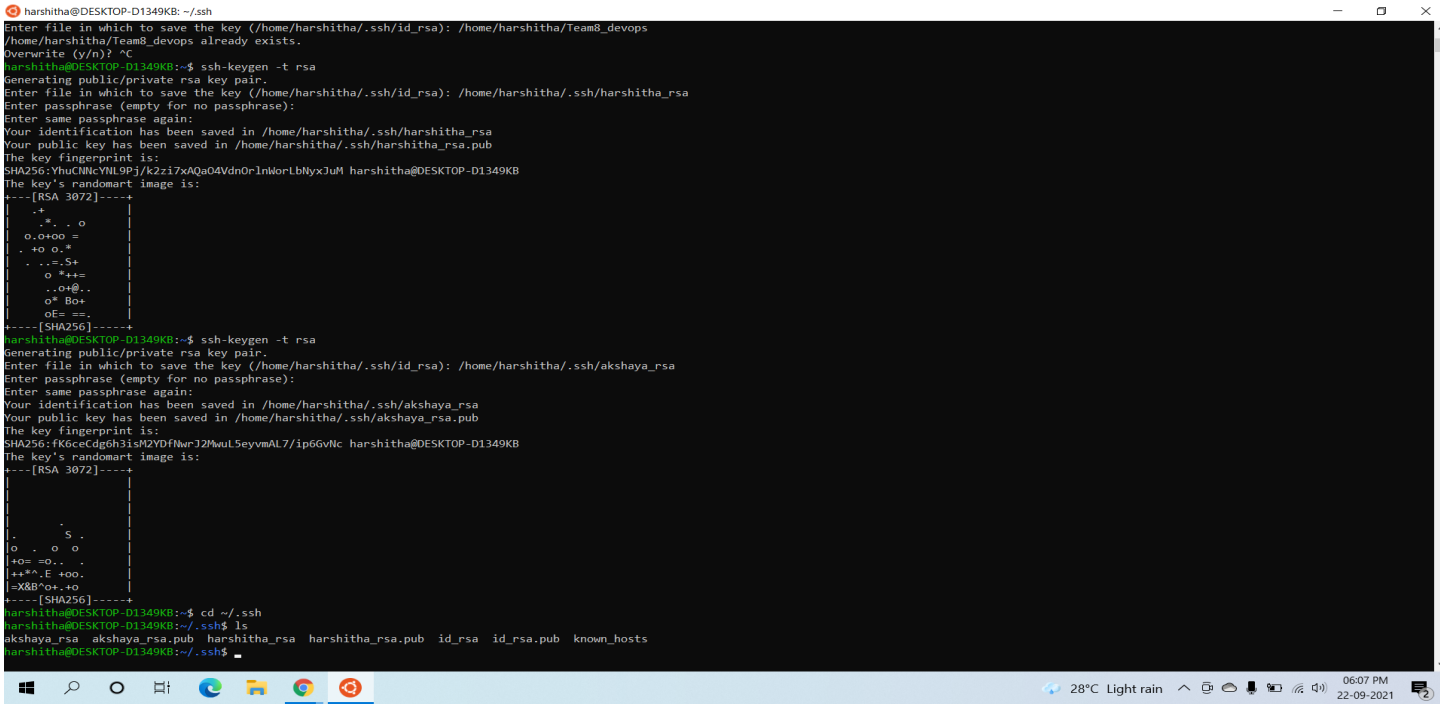
```
[hosts_to_add_key]
172.99.1.82  ansible_user=ec2-user ansible_port=22
172.99.1.56  ansible_user=ec2-user ansible_port=22

[hosts_to_add_key:vars]
ansible_ssh_common_args="-o StrictHostKeyChecking=no"
```

2. Checking if the host are available



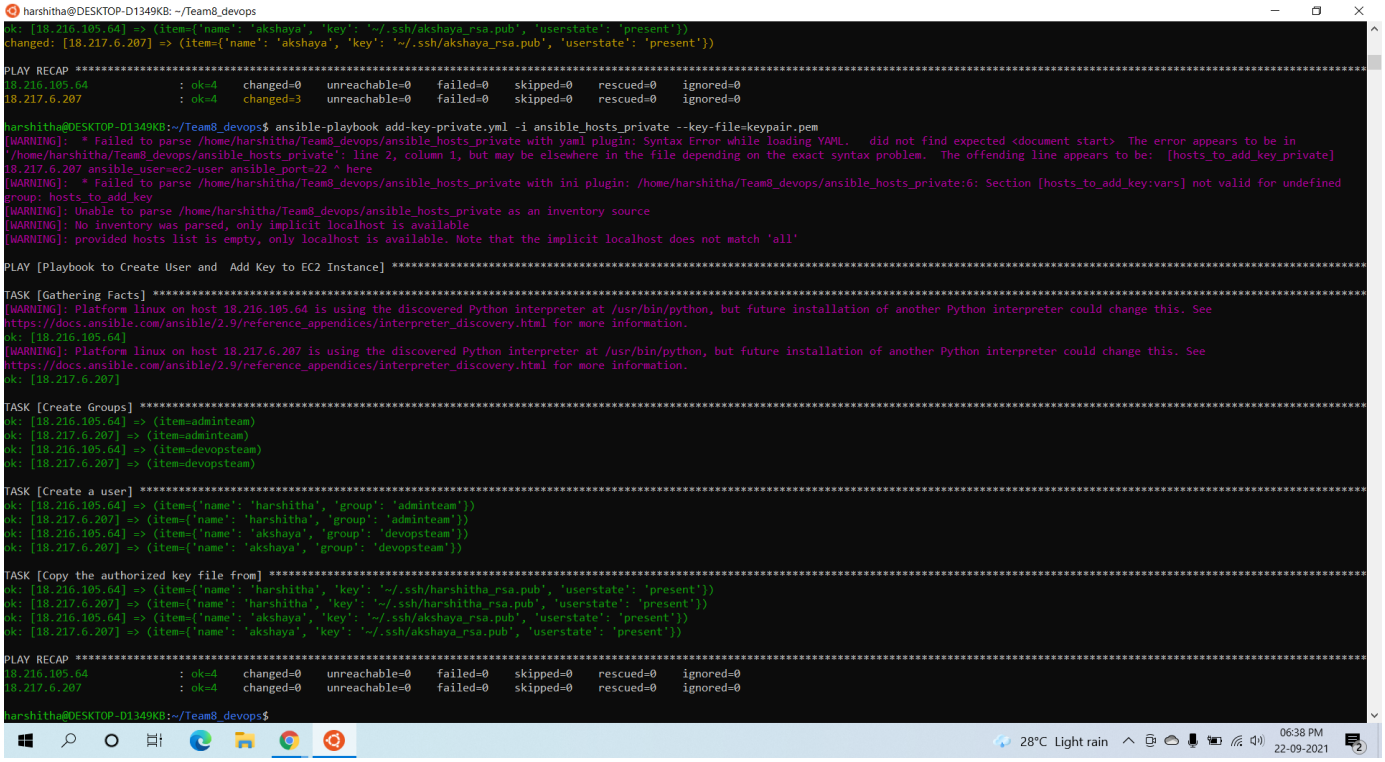
3.Playbook to add users in Ec2 instance and copy SSH key

Task-1 to create a group of users
Task-2 to create a user and map to a group created in the previous step
Create a keypair for each user

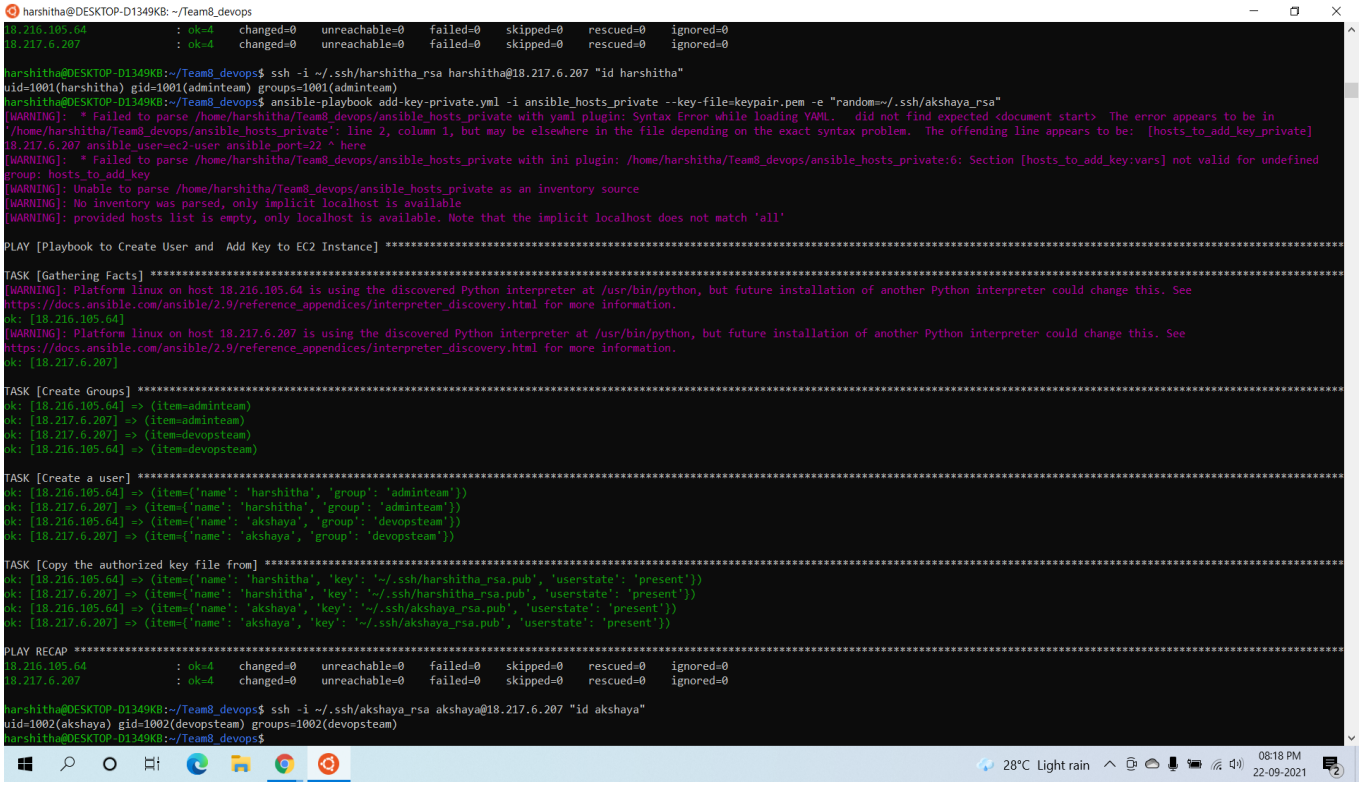


Task-3 to copy the user's SSH key to their newly created user IDs on the EC2 instance for them to able to log in.

4. Running the playbook



5. Playbook has been executed successfully and the user can log in/SSH now with his private key.I am executing `id` command with SSH connection and using my private key (akshaya_rsa) file to login without password.



Account has been created successfully.