TASK1:THREE TIER WEB APPLICATION USING DOCKER

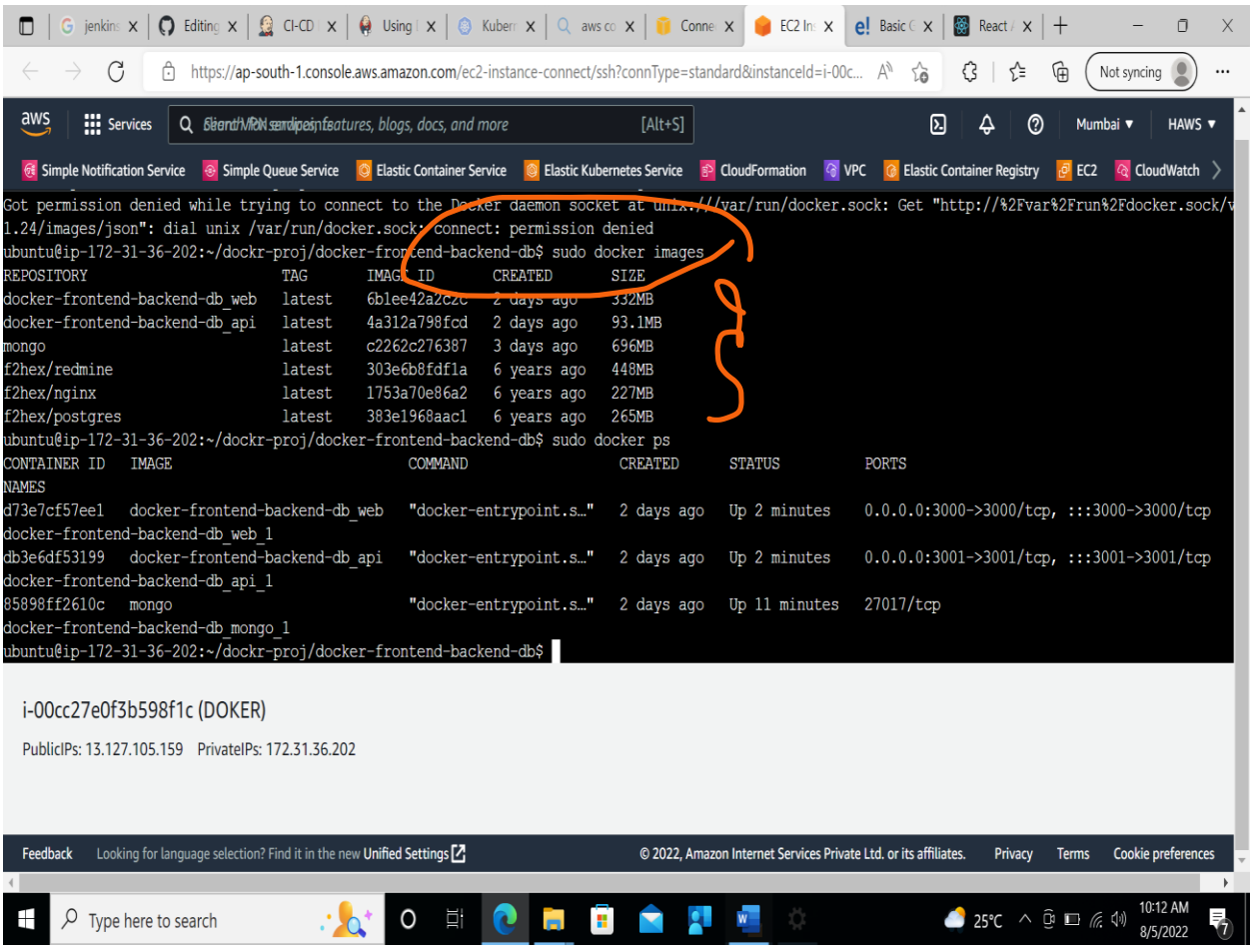
Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template. Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments.

docker search to search for public images on the Docker hub

docker images to search for images in our local

pull that from the Docker hub using the command docker pull.



A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.

We can create container from an image using the command

`docker run  - -name container-name -it image`

We can list all the running containers by using the following command.

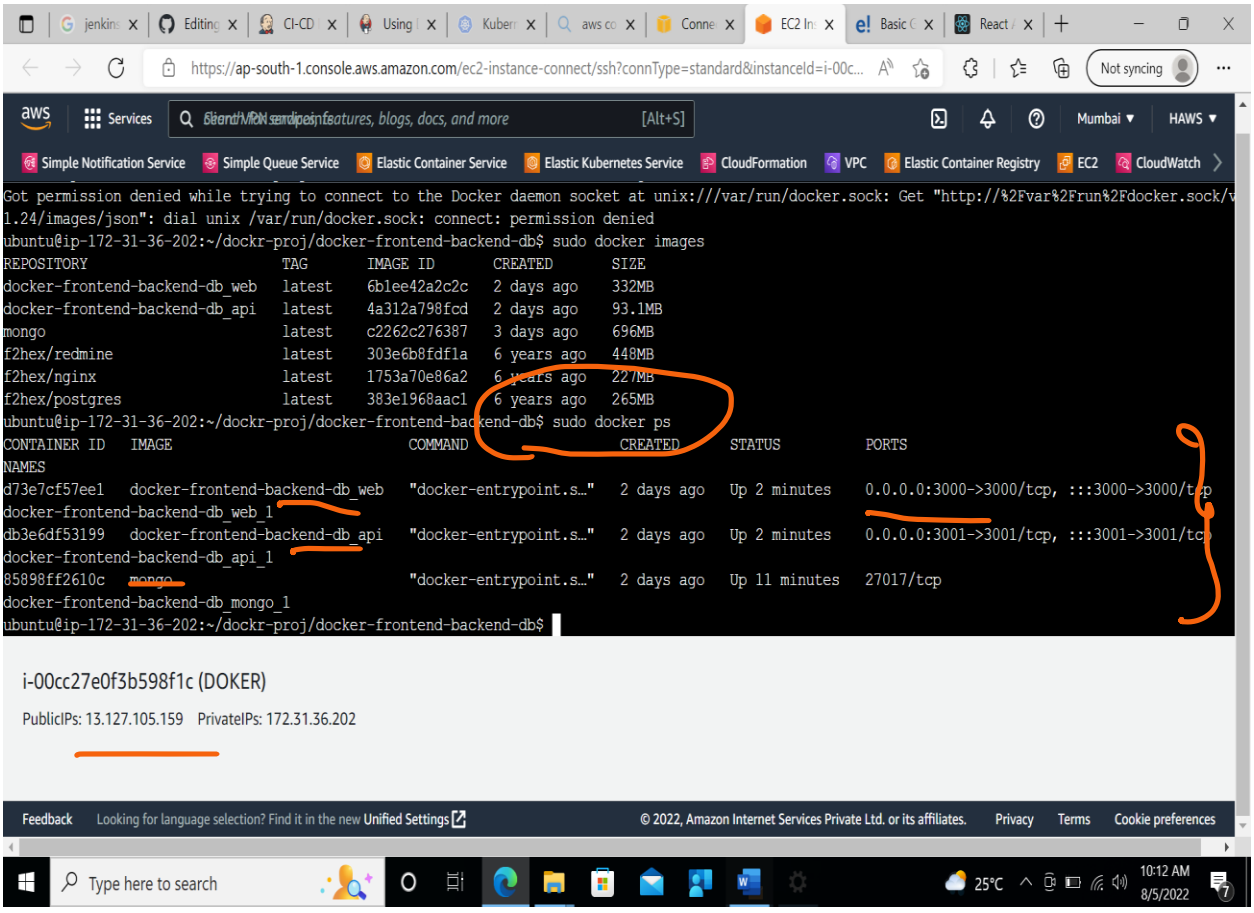
`docker ps`

We can list all the running & non running containers by using the following command.

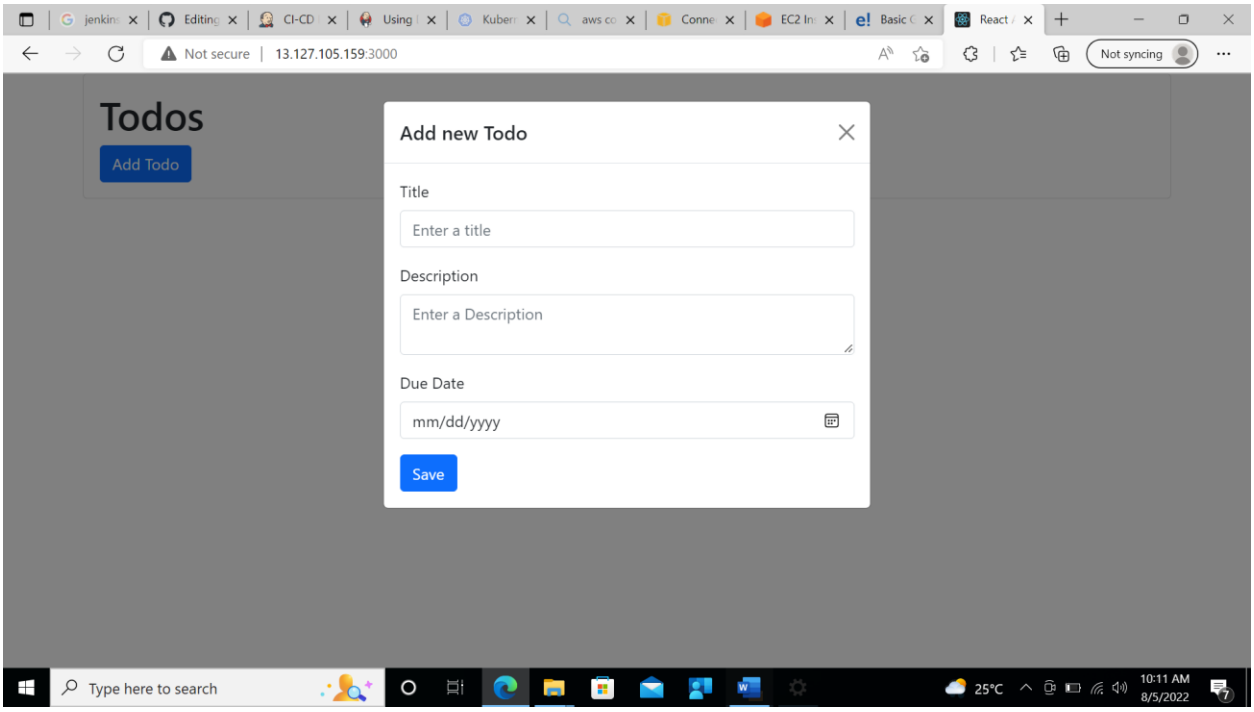
`docker ps -a`

Docker Compose is **a tool that was developed to help define and share multi-container applications**. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down.

In this task docker compose is used to spinup a todo application.



To verify appl running status http://public-ip:container running port will give the page displayed below



Git url: https://github.com/mhanumanth/DOCKER-TASK.git

TASK1.1:THRE TIER WEBAPPLICATION USING KUBERNETES

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

This task deploy the guestbook application deploying the application using configuration files. The configuration file mechanism allows you to have more fine-grained control over all of resources being created within the Kubernetes cluster.

deployment configuration for guestbook application file create a deployment object named 'guestbook' with a pod containing a single container running the image ibmcom/ guestbook:v1. Also the configuration specifies replicas set to 3 and Kubernetes tries to make sure that at least three active pods are running at all times.

To create a Deployment using this configuration file the command is:

```
kubectl create -f guestbook-deployment.yaml
```

We can change the number of replicas in the configuration, Kubernetes will try to add, or remove, pods from the system to match your request. To make these modifications by using the following command:

```
kubectl edit deployment guestbook-v1
```

to make the change effective when you edit the deployment locally we use command

```
kubectl apply -f guestbook-deployment.yaml
```

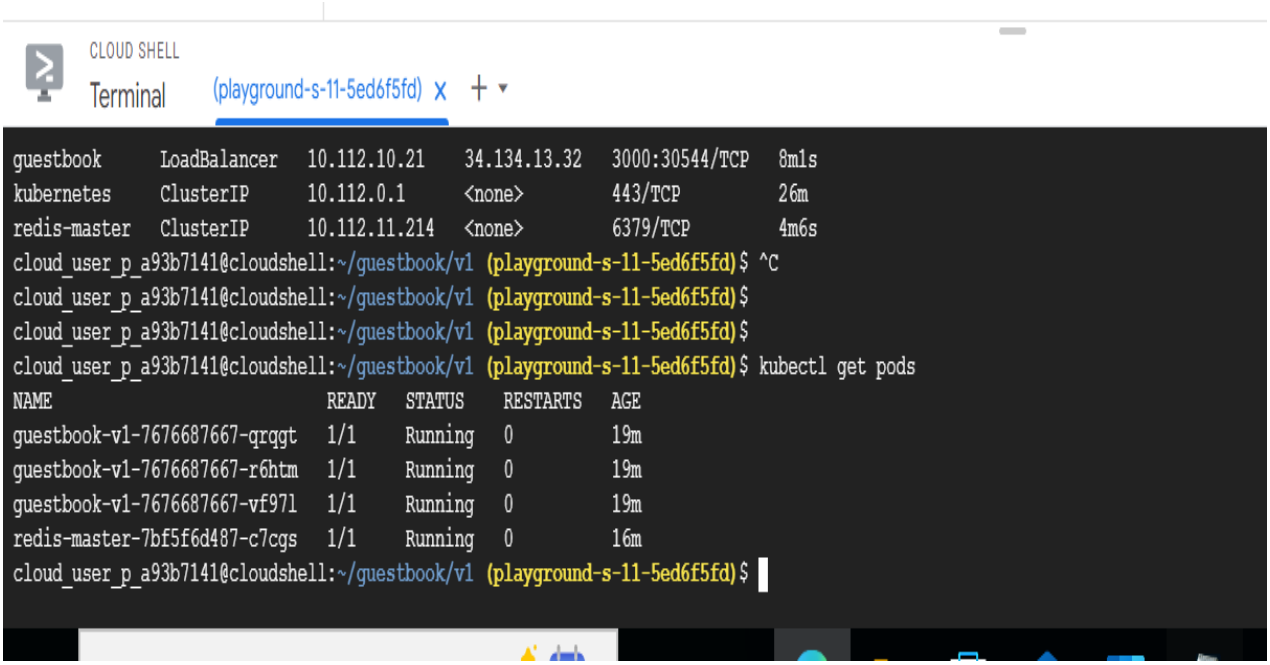
We define a Service object to expose the deployment to external clients.

```
kubectl create -f guestbook-service.yaml
```

The above configuration creates a Service resource named guestbook. A Service can be used to create a network path for incoming traffic to your running application. In this case, we are setting up a route from port 3000 on the cluster to the "http-server" port on our app, which is port 3000 per the Deployment container spec.

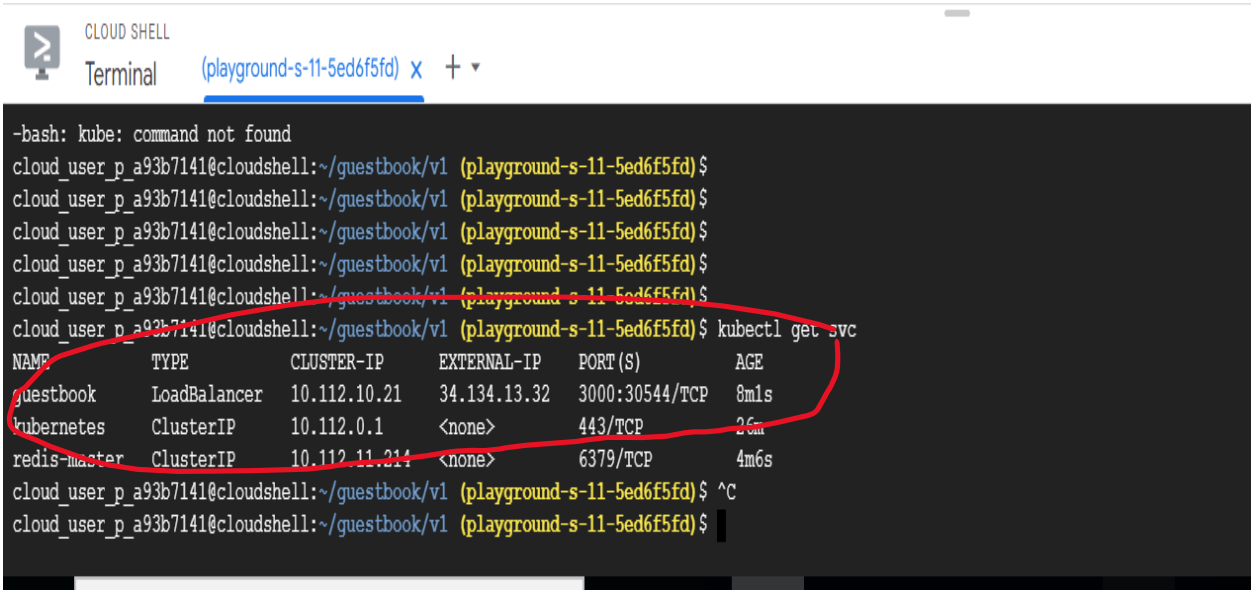
To oberve the pods running, we can use the command

kubectl get pods

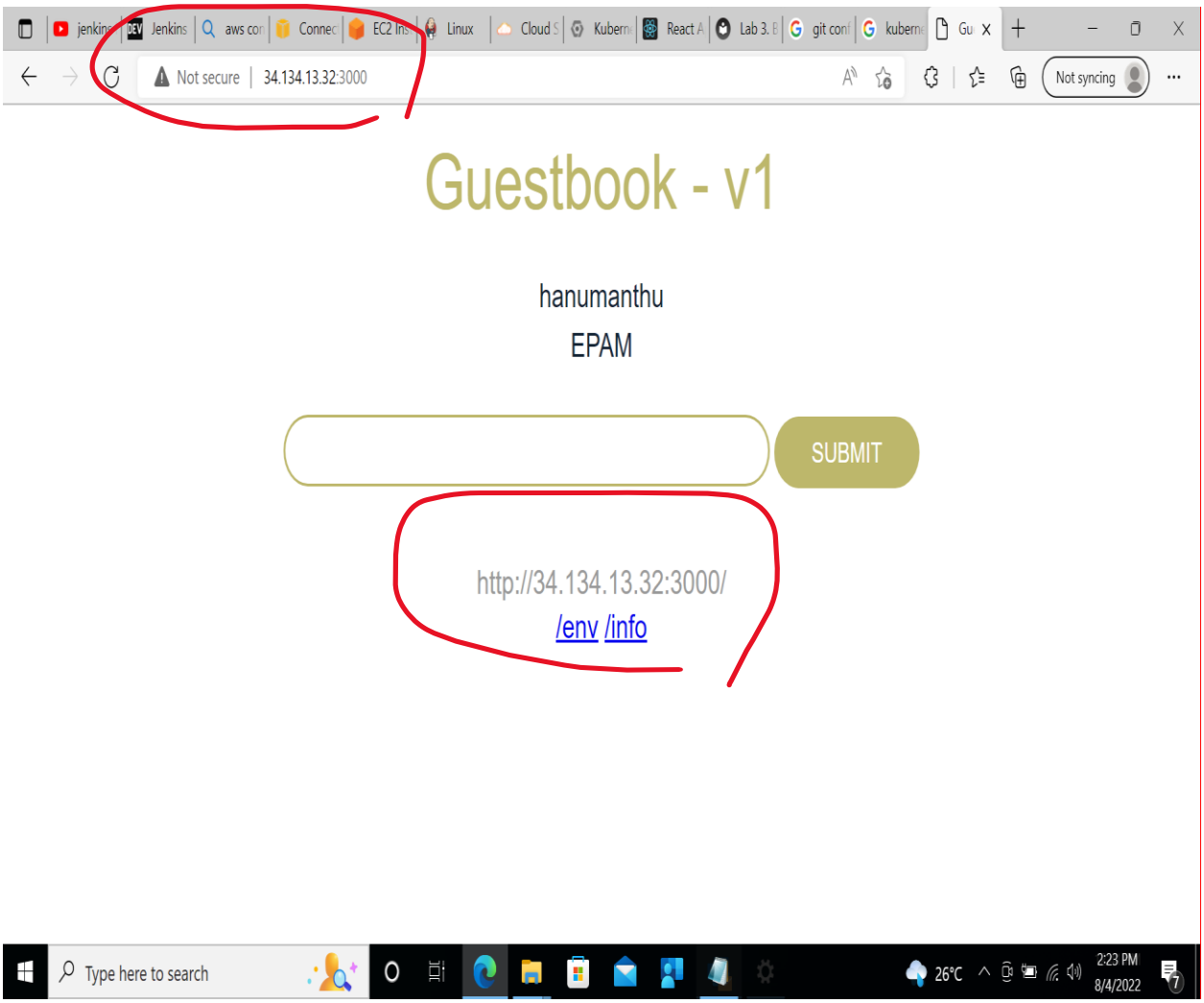


To get the sevices running we can use the command

kubectl get service or kubectl get svc



Test guestbook app using a browser of your choice using the url <your-cluster-ip>:<node-port>



Git url: https://github.com/hanumanthsaisiva/KUBERNETES-TASK.git

TASK2: INFRASTRUCTURE AS CODE USING TERRAFORM (MODULES)

Terraform creates and manages resources on cloud platforms and other services through their application programming interfaces (APIs). Providers enable Terraform to work with virtually any platform or service with an accessible API.

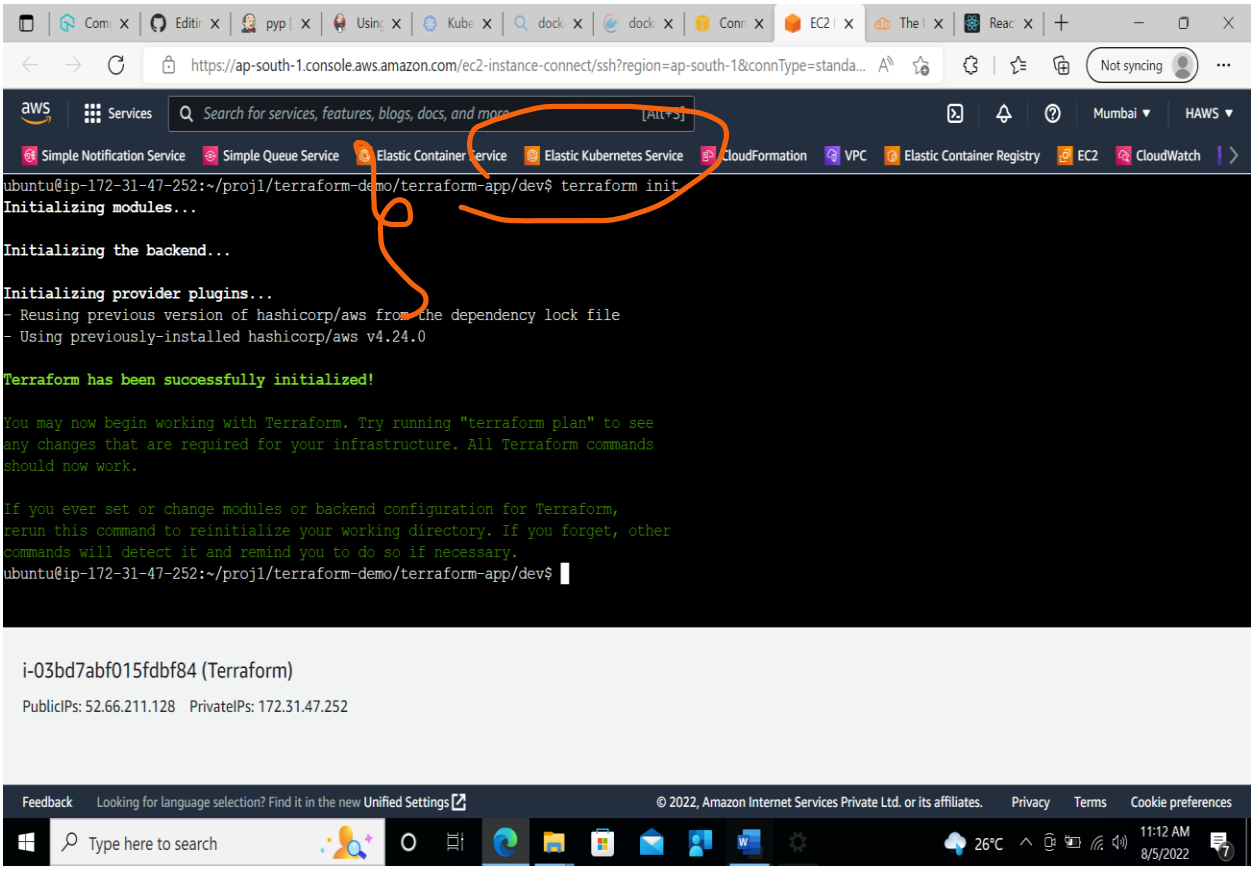
In this task iam going to crate, destroy three resources to aws account

1.ec2-instance

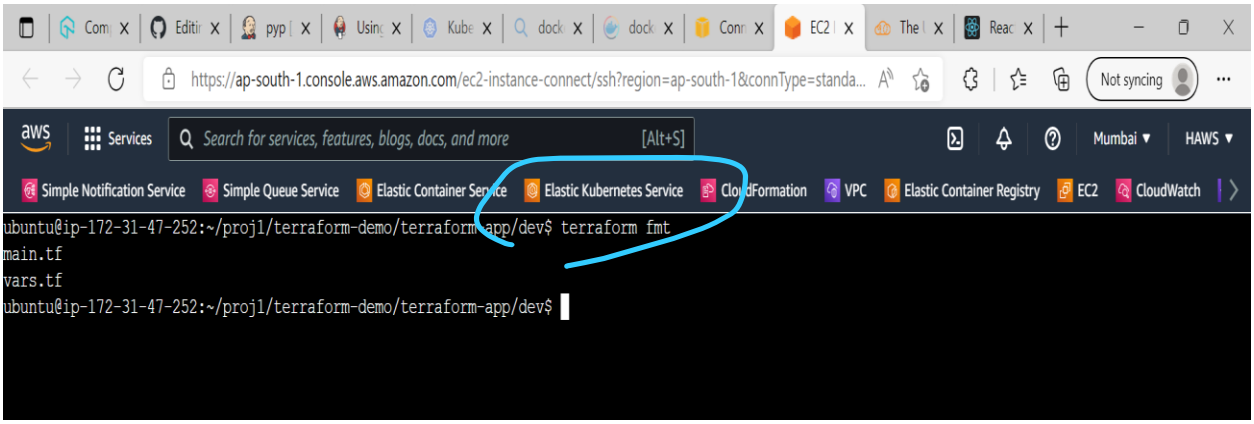
2.vpc

3.subnet

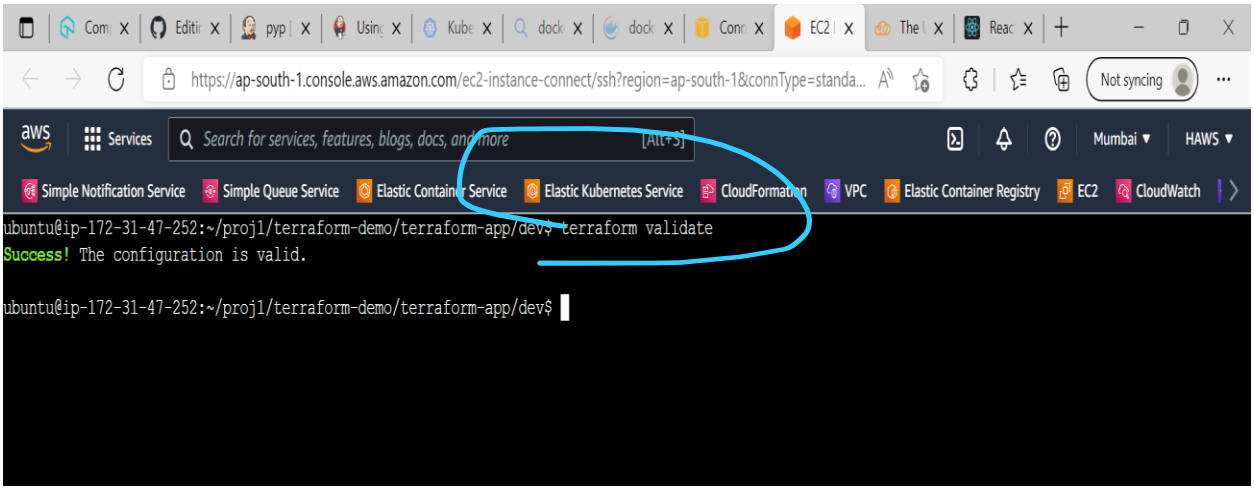
- **terraform init** command initialize directory, pull down providers



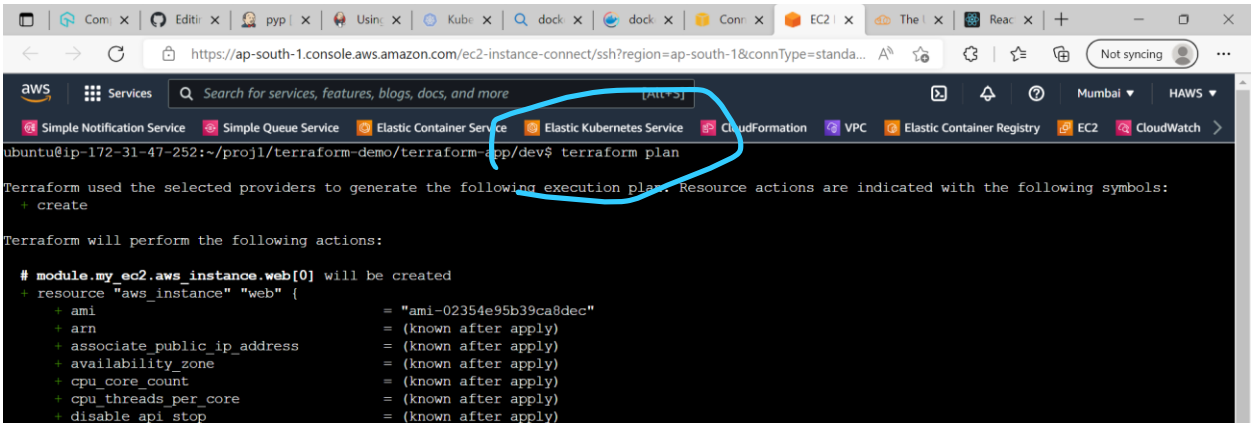
terraform fmt #format code per HCL canonical standard

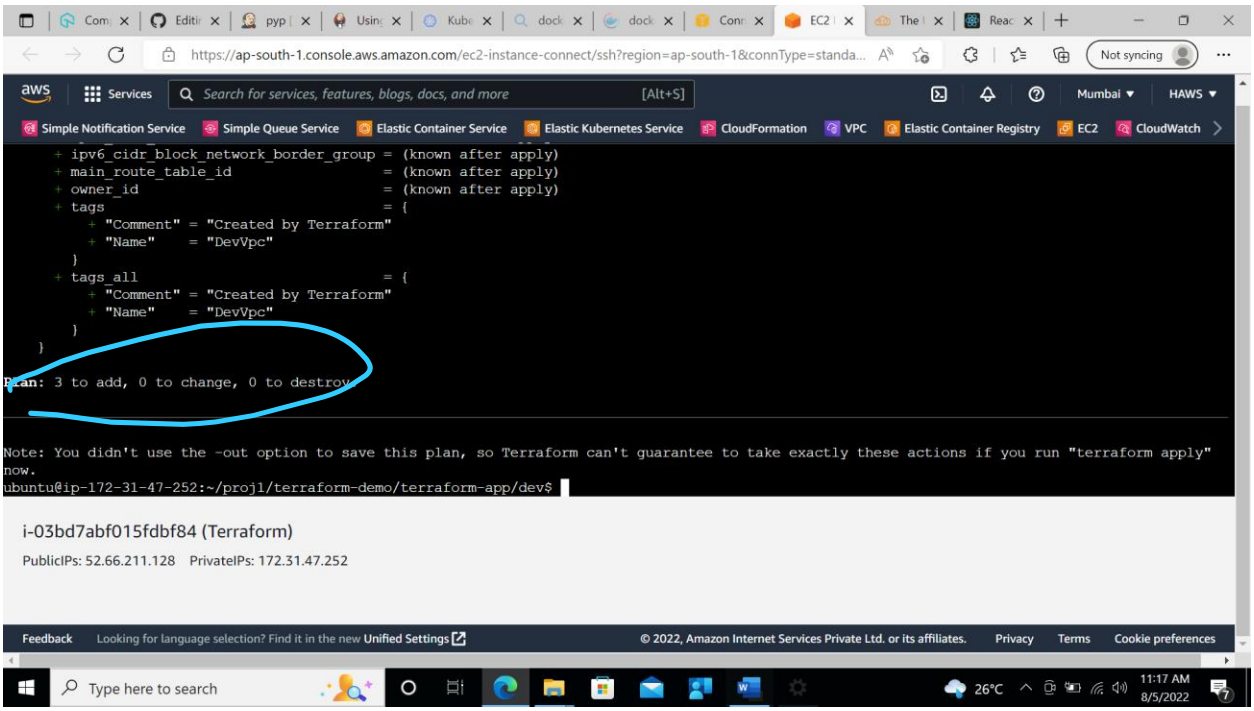


terraform validate #validate code for syntax

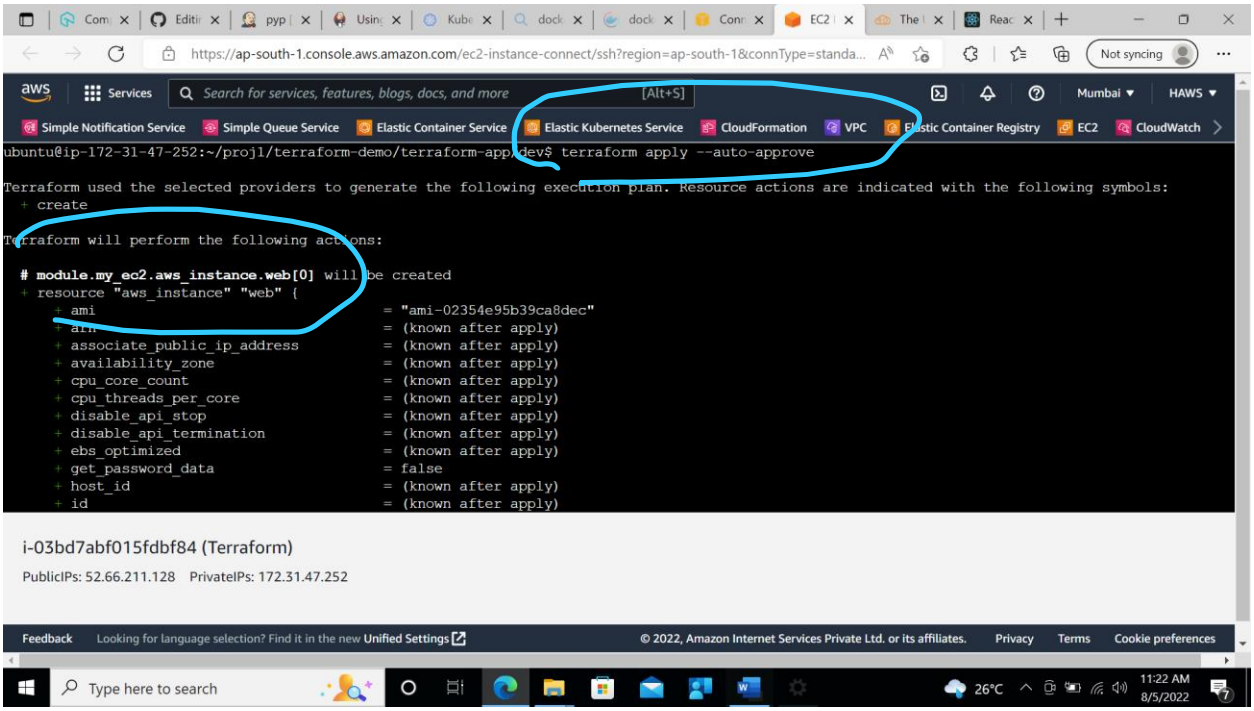


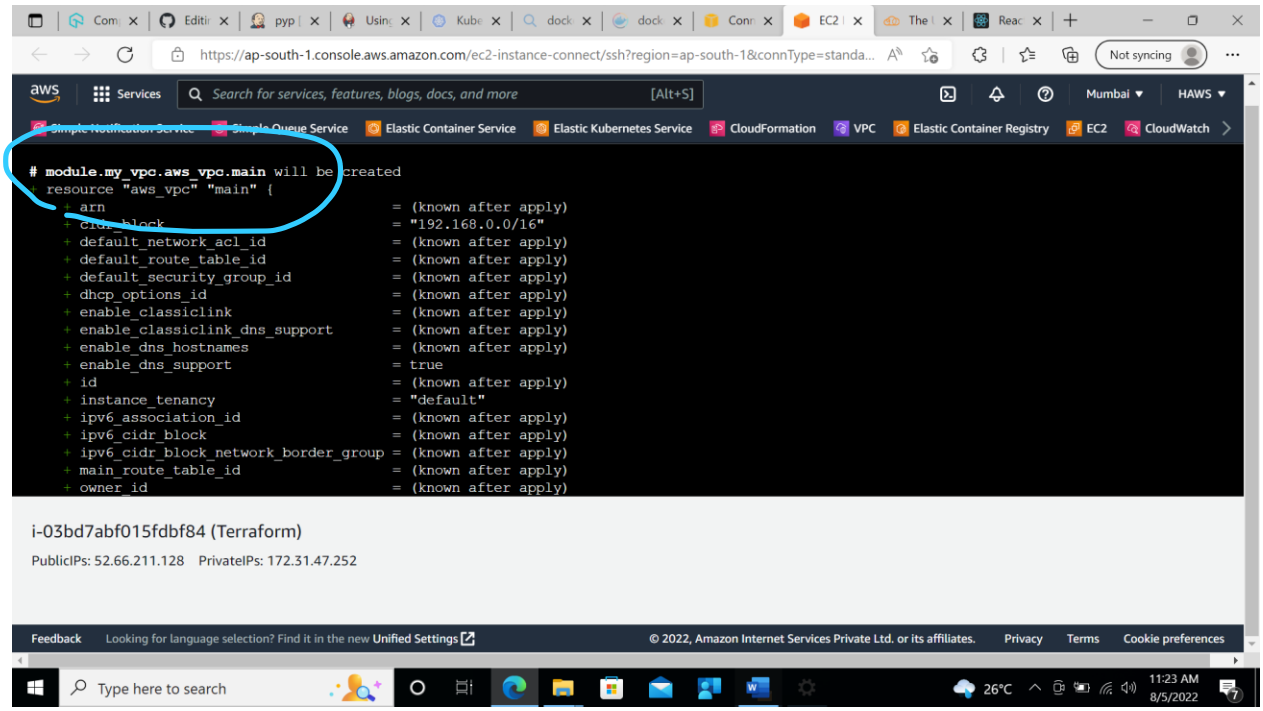
terraform plan -out plan.out #output the deployment plan

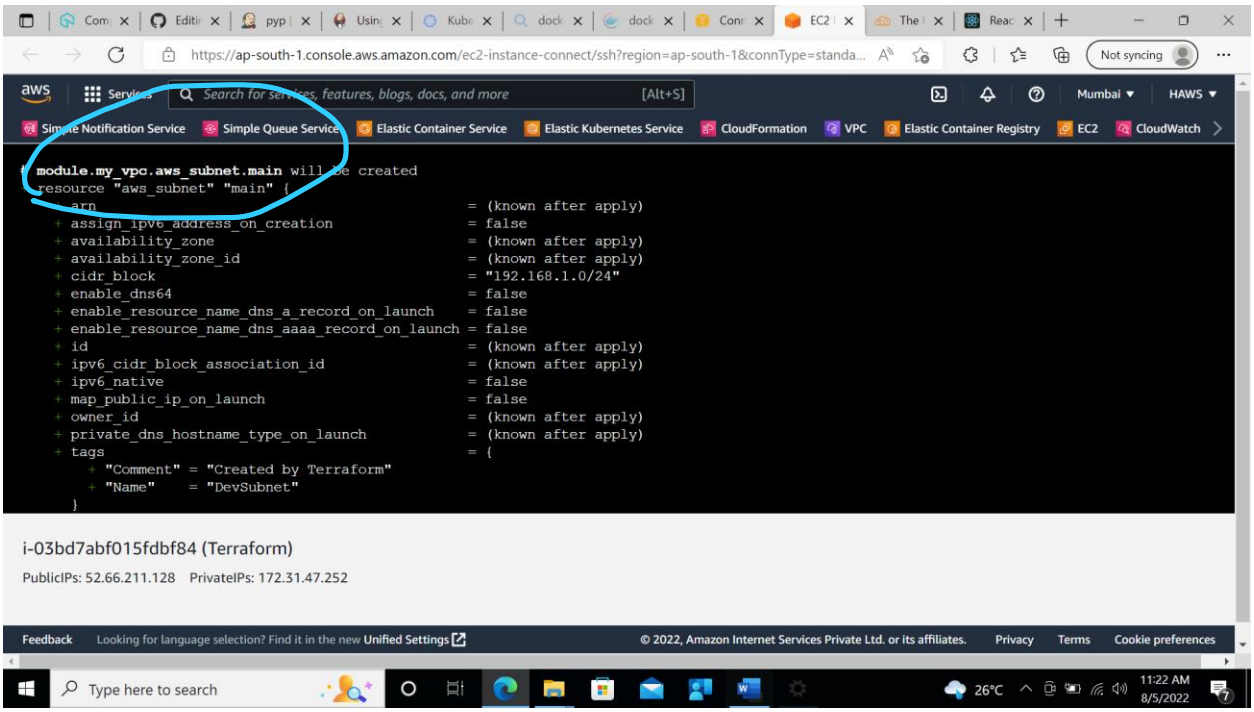


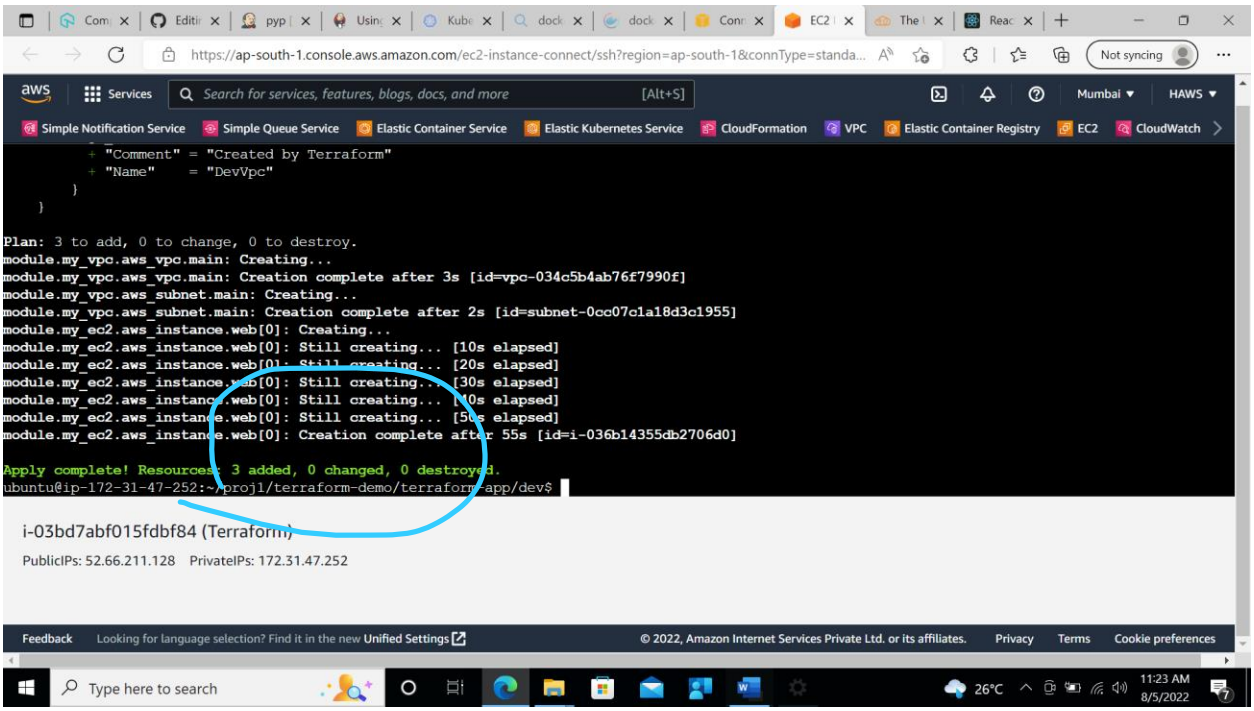


terraform apply --auto-approve #apply changes without being prompted for “yes”

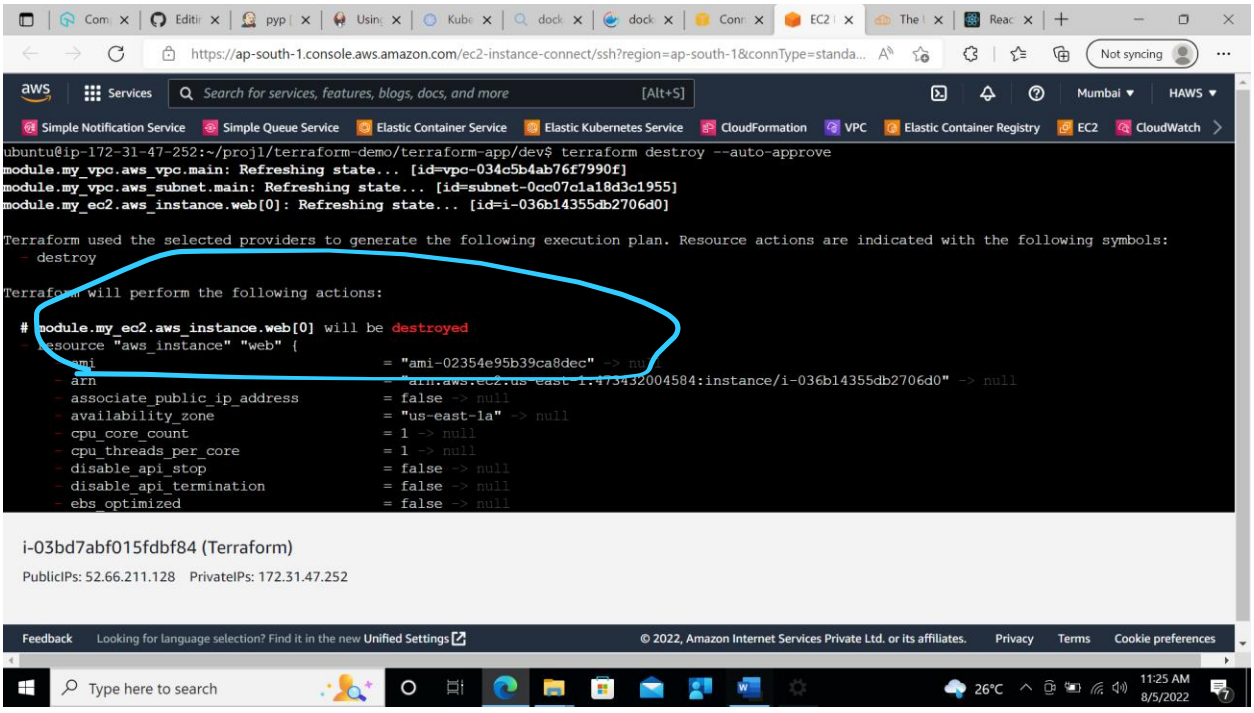


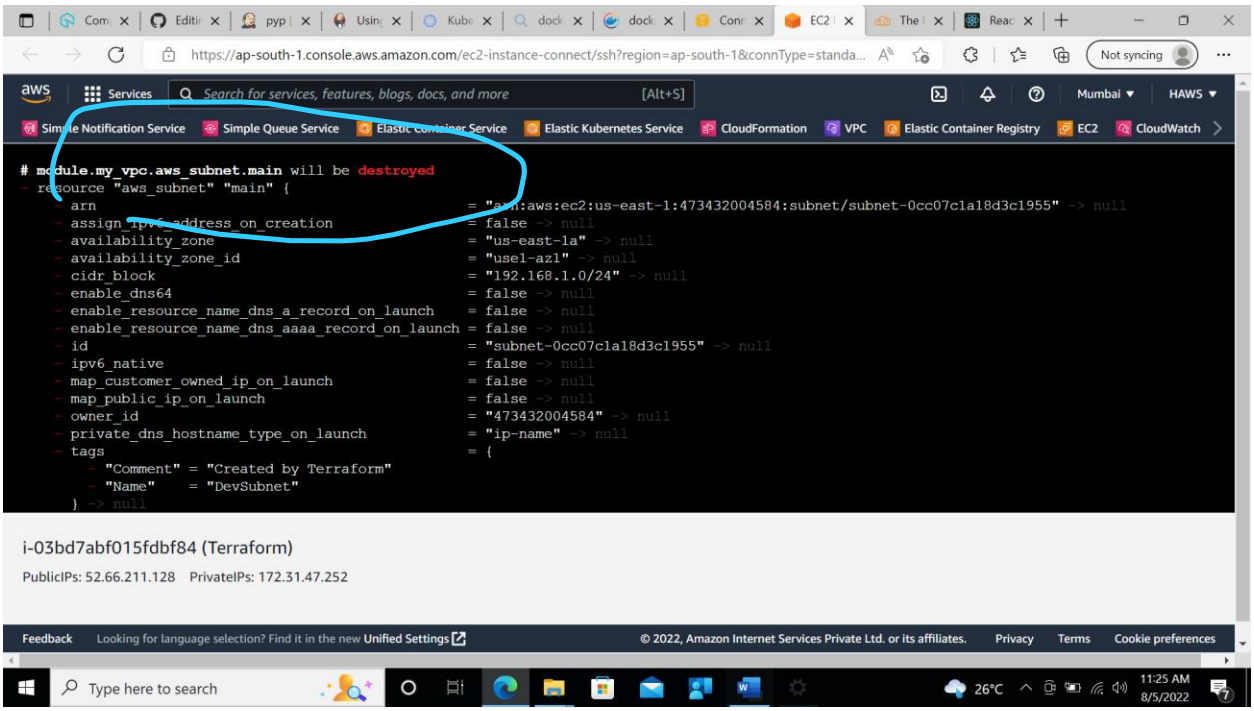

https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standa...
Not syncing
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
HAWS
Elastic Container Service
Elastic Kubernetes Service
CloudFormation
VPC
Elastic Container Registry
EC2
CloudWatch
# module.my_vpc.aws_vpc.main will be created
+ resource "aws_vpc" "main" {
+ arn = (known after apply)
+ cidr_block = "192.168.0.0/16"
+ default_network_acl_id = (known after apply)
+ default_route_table_id = (known after apply)
+ default_security_group_id = (known after apply)
+ dhcp_options_id = (known after apply)
+ enable_classiclink = (known after apply)
+ enable_classiclink_dns_support = (known after apply)
+ enable_dns_hostnames = (known after apply)
+ enable_dns_support = true
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
i-03bd7abf015fdbf84 (Terraform)
PublicIPs: 52.66.211.128 PrivateIPs: 172.31.47.252
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
26°C
11:23 AM
8/5/2022

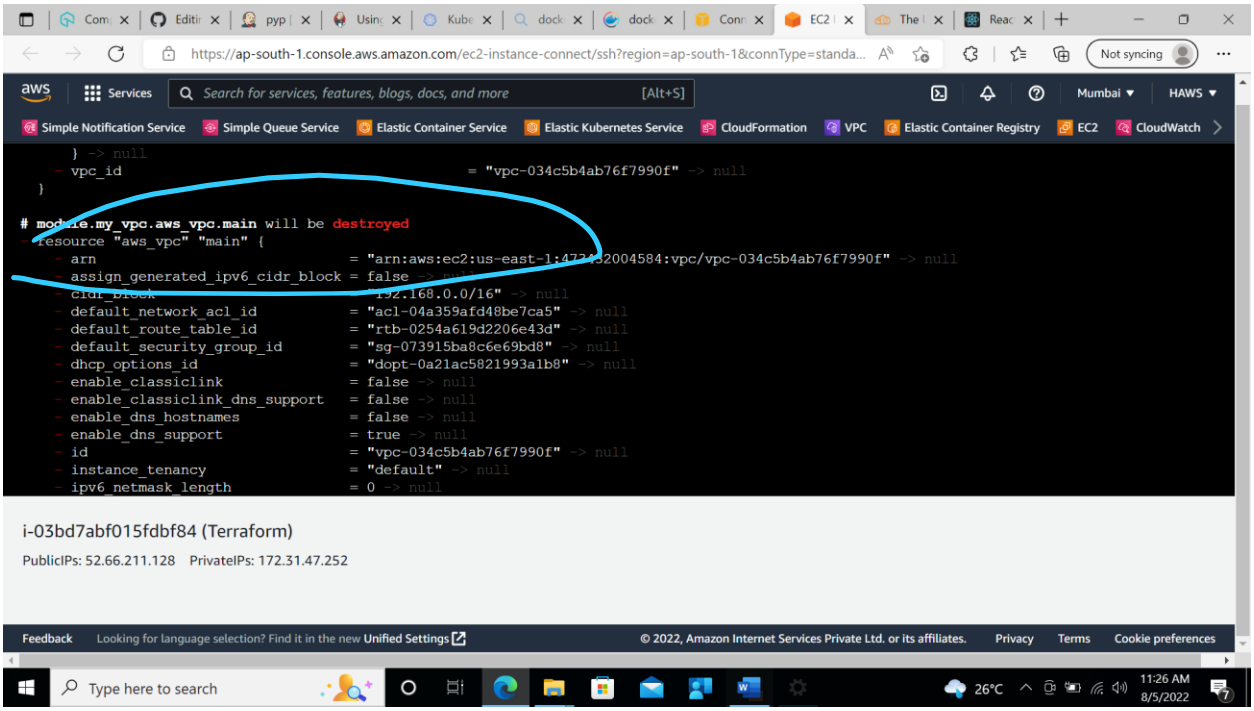

https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standa...
Not syncing
aws
Mumbai
HAWS
Simple Notification Service
Simple Queue Service
Elastic Container Service
Elastic Kubernetes Service
CloudFormation
VPC
Elastic Container Registry
EC2
CloudWatch
# module.my_vpc.aws_subnet.main will be created
+ resource "aws_subnet" "main" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = (known after apply)
+ availability_zone_id = (known after apply)
+ cidr_block = "192.168.1.0/24"
+ enable_dns64 = false
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id = (known after apply)
+ ipv6_cidr_block_association_id = (known after apply)
+ ipv6_native = false
+ map_public_ip_on_launch = false
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags = {
+ "Comment" = "Created by Terraform"
+ "Name" = "DevSubnet"
}
i-03bd7abf015fdbf84 (Terraform)
PublicIPs: 52.66.211.128 PrivateIPs: 172.31.47.252
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
26°C
11:22 AM
8/5/2022

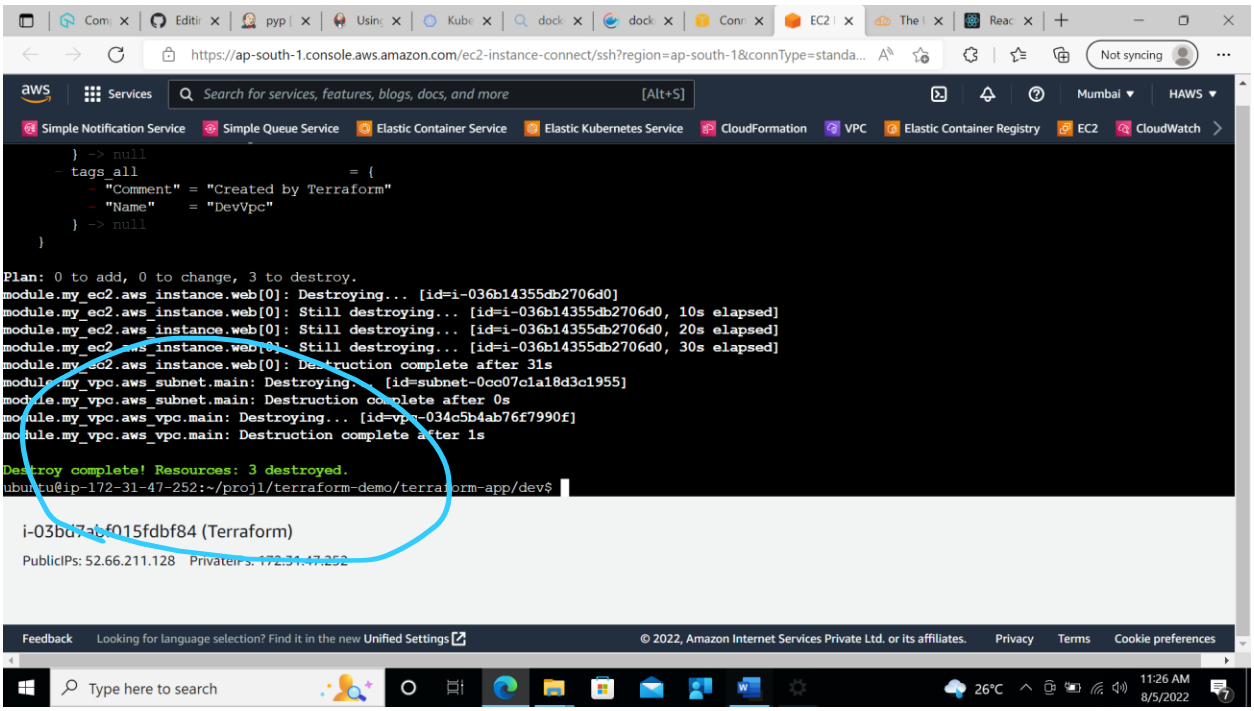


terraform destroy --auto-approve #destroy/cleanup deployment without being prompted for “yes”




https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standa...
Not syncing
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
HAWS
Simple Notification Service
Simple Queue Service
Elastic Container Service
Elastic Kubernetes Service
CloudFormation
VPC
Elastic Container Registry
EC2
CloudWatch
# module.my_vpc.aws_subnet.main will be destroyed
- resource "aws_subnet" "main" {
- arn = "arn:aws:ec2:us-east-1:473432004584:subnet/subnet-0cc07c1a18d3c1955" -> null
- assign_ipv6_address_on_creation = false -> null
- availability_zone = "us-east-1a" -> null
- availability_zone_id = "use1-az1" -> null
- cidr_block = "192.168.1.0/24" -> null
- enable_dns64 = false -> null
- enable_resource_name_dns_a_record_on_launch = false -> null
- enable_resource_name_dns_aaaa_record_on_launch = false -> null
- id = "subnet-0cc07c1a18d3c1955" -> null
- ipv6_native = false -> null
- map_customer_owned_ip_on_launch = false -> null
- map_public_ip_on_launch = false -> null
- owner_id = "473432004584" -> null
- private_dns_hostname_type_on_launch = "ip-name" -> null
- tags = {
- "Comment" = "Created by Terraform"
- "Name" = "DevSubnet"
} -> null
i-03bd7abf015fdbf84 (Terraform)
PublicIPs: 52.66.211.128 PrivateIPs: 172.31.47.252
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
26°C
11:25 AM
8/5/2022


https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standa...
Not syncing
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
HAWS
Simple Notification Service
Simple Queue Service
Elastic Container Service
Elastic Kubernetes Service
CloudFormation
VPC
Elastic Container Registry
EC2
CloudWatch
} -> null
- vpc_id = "vpc-034c5b4ab76f7990f" -> null
}
# module.my_vpc.aws_vpc.main will be destroyed
- resource "aws_vpc" "main" {
- arn = "arn:aws:ec2:us-east-1:473432004584:vpc/vpc-034c5b4ab76f7990f" -> null
- assign_generated_ipv6_cidr_block = false -> null
- cidr_block = "192.168.0.0/16" -> null
- default_network_acl_id = "acl-04a359afd48be7ca5" -> null
- default_route_table_id = "rtb-0254a619d2206e43d" -> null
- default_security_group_id = "sg-073915ba8c6e69bd8" -> null
- dhcp_options_id = "dopt-0a21ac5821993a1b8" -> null
- enable_classiclink = false -> null
- enable_classiclink_dns_support = false -> null
- enable_dns_hostnames = false -> null
- enable_dns_support = true -> null
- id = "vpc-034c5b4ab76f7990f" -> null
- instance_tenancy = "default" -> null
- ipv6_netmask_length = 0 -> null
i-03bd7abf015fdbf84 (Terraform)
PublicIPs: 52.66.211.128 PrivateIPs: 172.31.47.252
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
26°C
11:26 AM
8/5/2022



Github url: https://github.com/mhanumanth/TERRAFORM-TASK.git

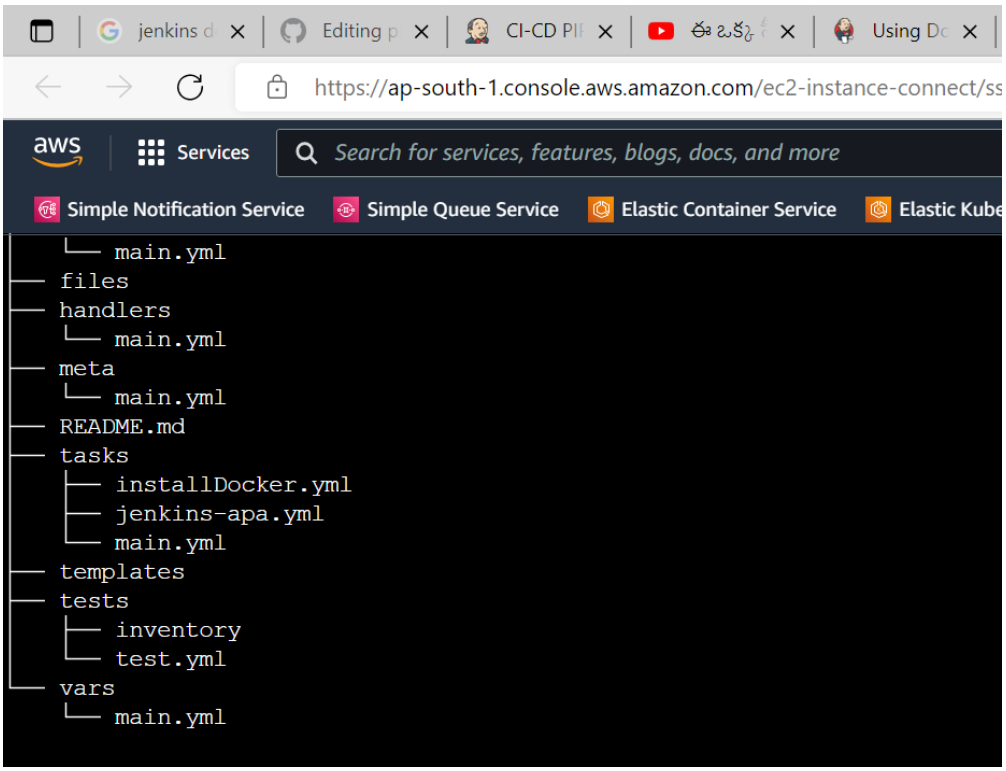
TASK3:CONFIGURATION MANAGEMENTUSING ANSIBLE

Ansible roles used to develop reusable automation components by grouping and encapsulating related automation artifacts, like configuration files, templates, tasks, and handlers.

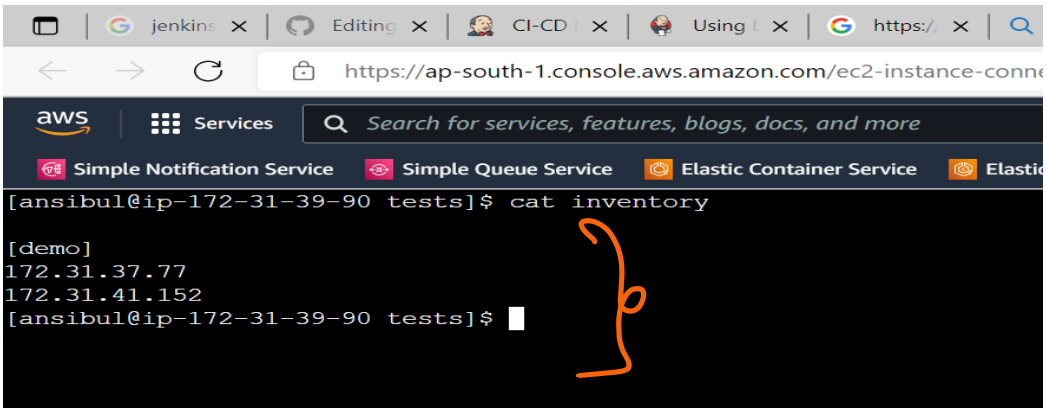
We use the command ansible-galaxy to initialize the role in the ansible master:

ansible-galaxy init CMAR

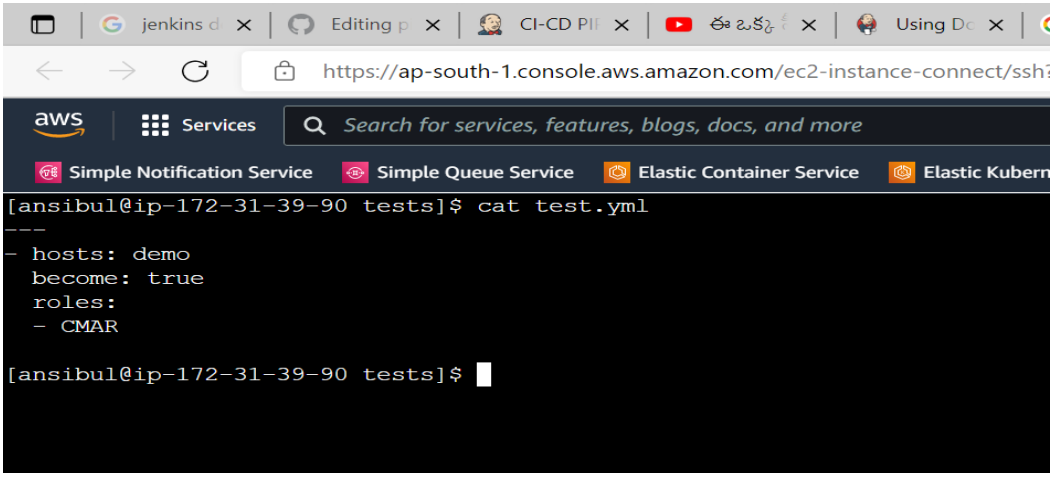
tree . will give the tree structure created by ansible-galaxy



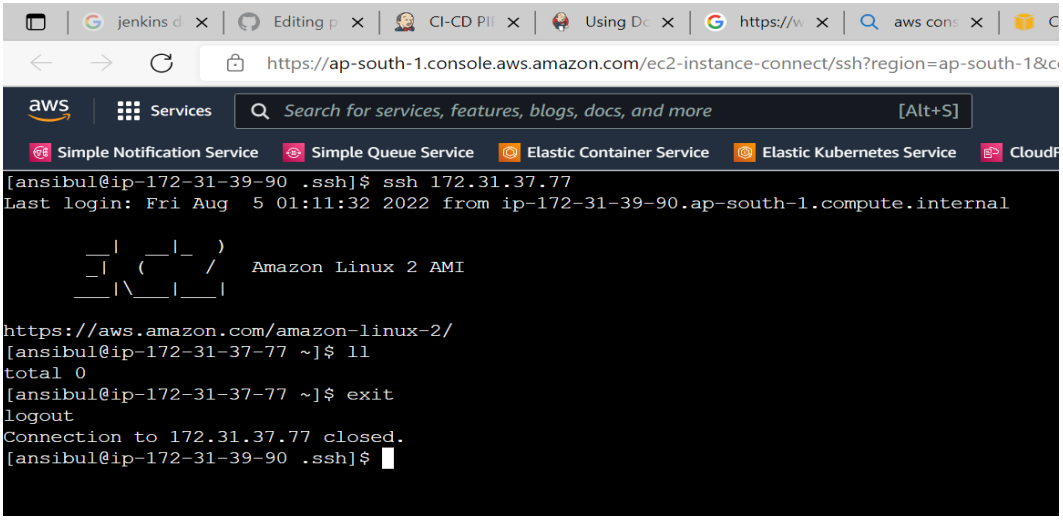
In this Task iam going to install Python,Ansible,Docker and Jenkins on 2 client Machines Whose ip's are defined in tests/inventory:



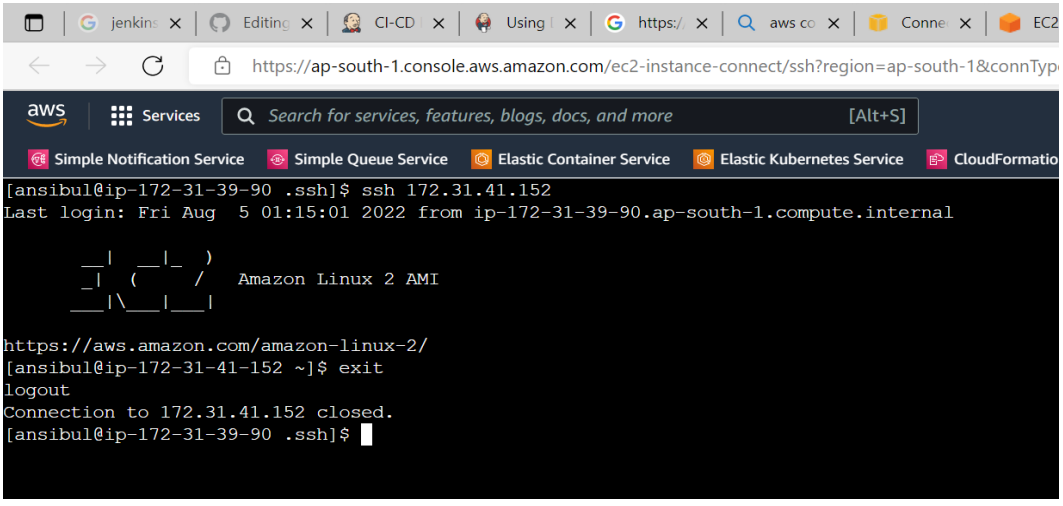
The playbook which we execute to install softwares on client machines is



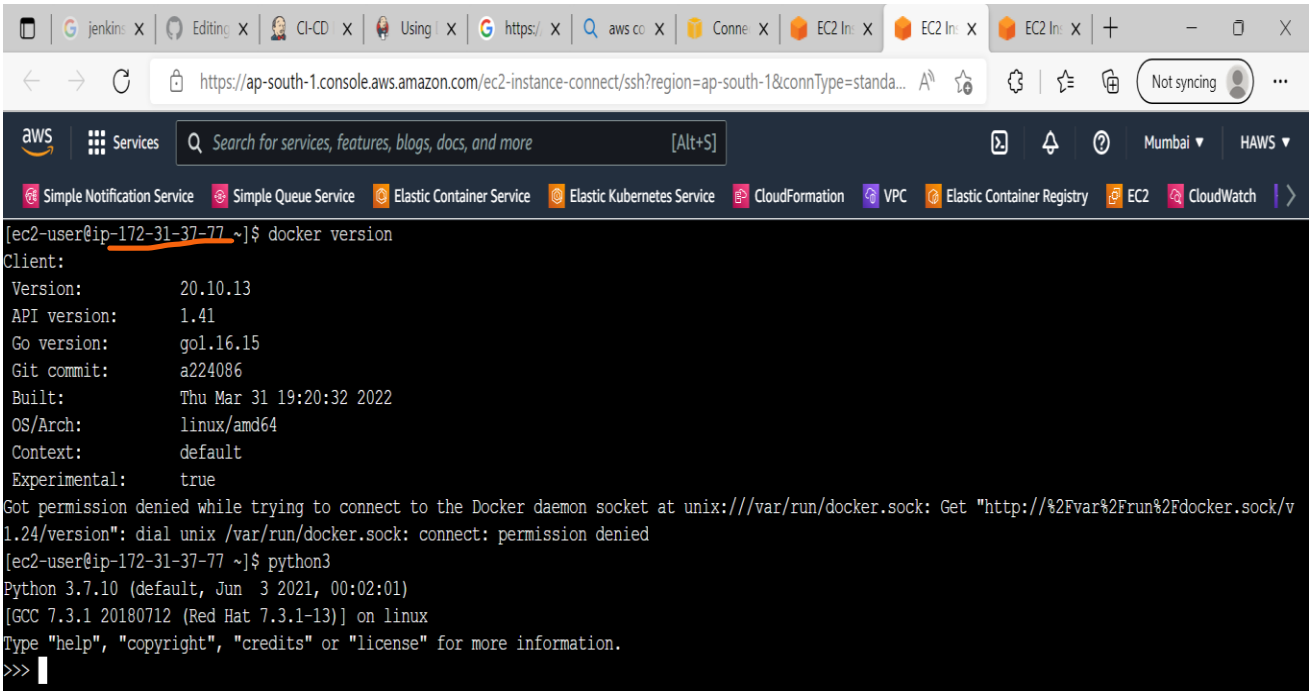
ssh connectivity to the client machines can be observed:@172.31.37.77

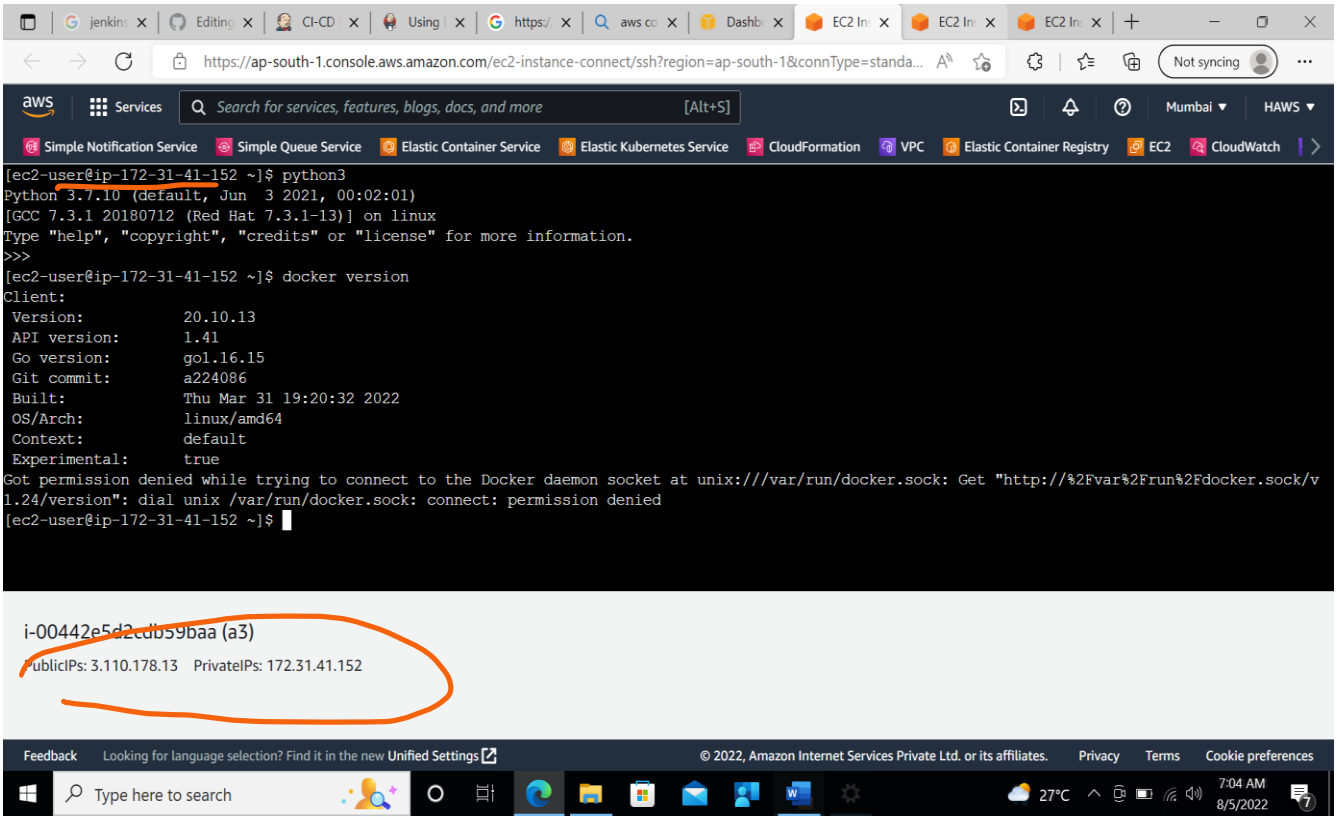


@172.31.41.152



After running the Ansible-Playbook test.yml ,we can observe the client node on which docker python installations.





Git-hub Repository URL: https://github.com/mhanumanth/CMAR.git

TASK4: APPLICATION CODE MANAGEMENT USING GIT

GIT stands for **Global Information Tracker**, **Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems)

To initialize the working directory

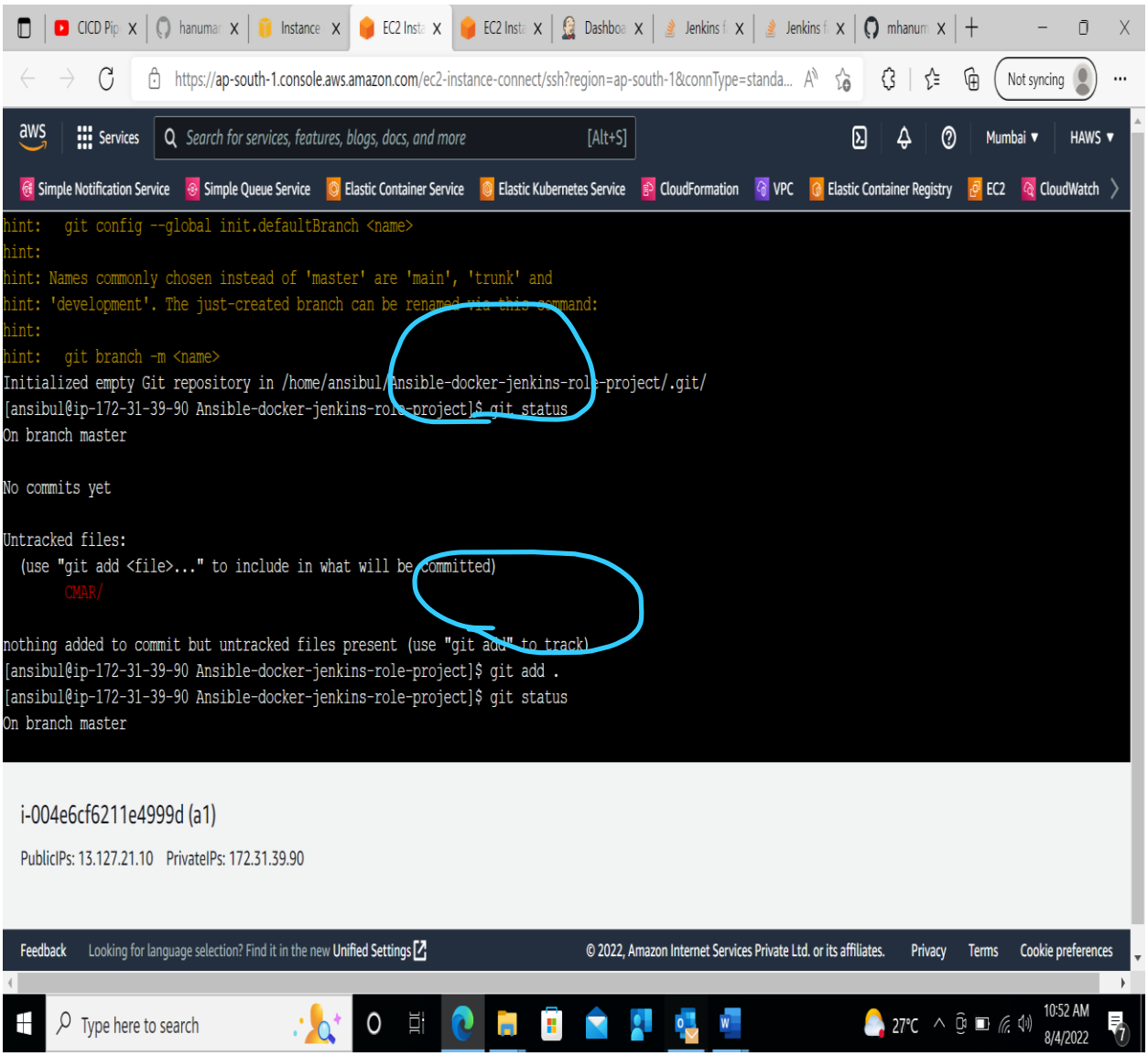
git init

The command that lists all the files that have to be committed.

git status

The command that adds a file to the staging area

git add



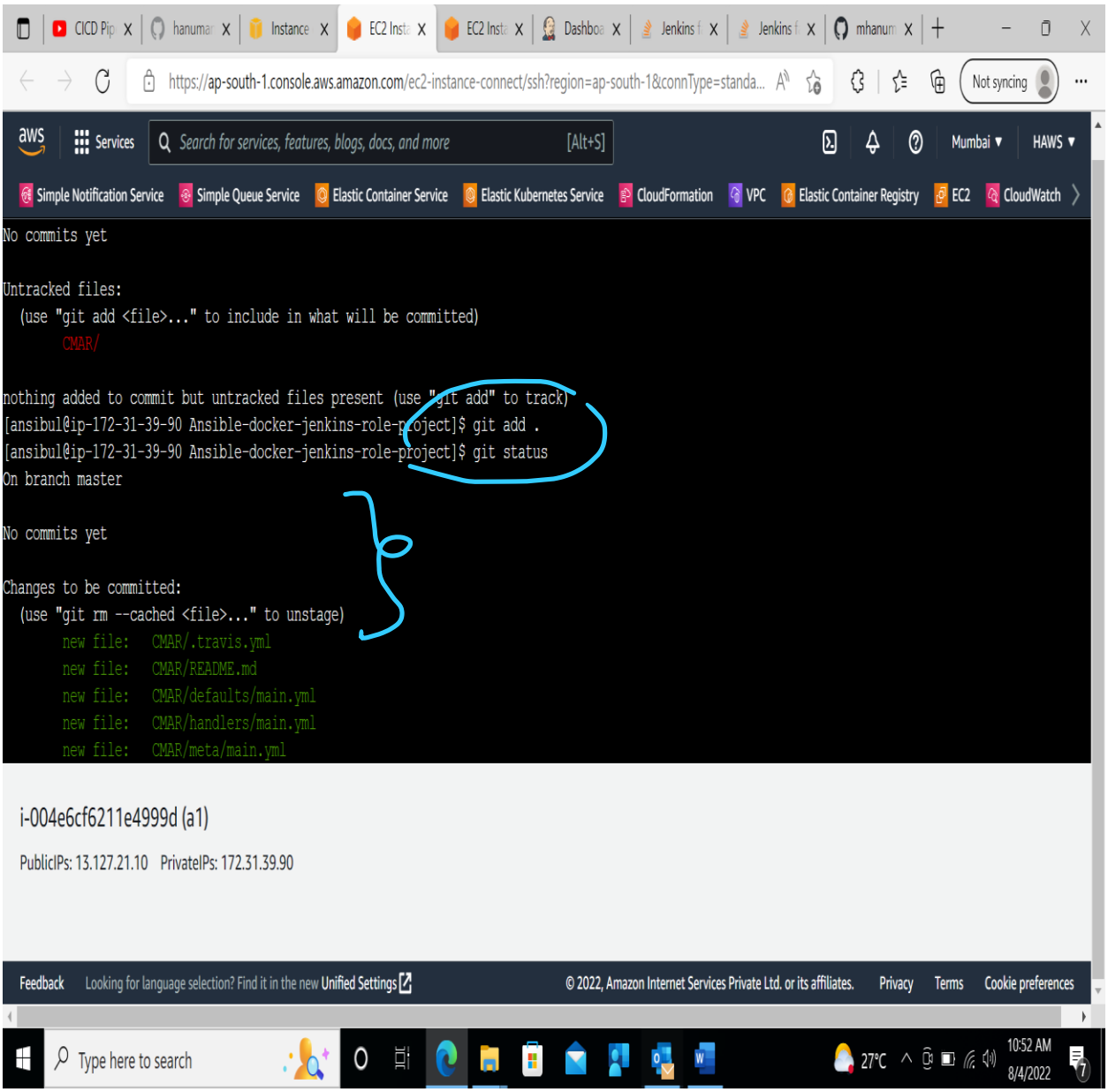
After git add the status of the files and the color changes from red to green

After git add command we can check the status using git status command to see this translation

Red represents unstaged files

Green represents staged files

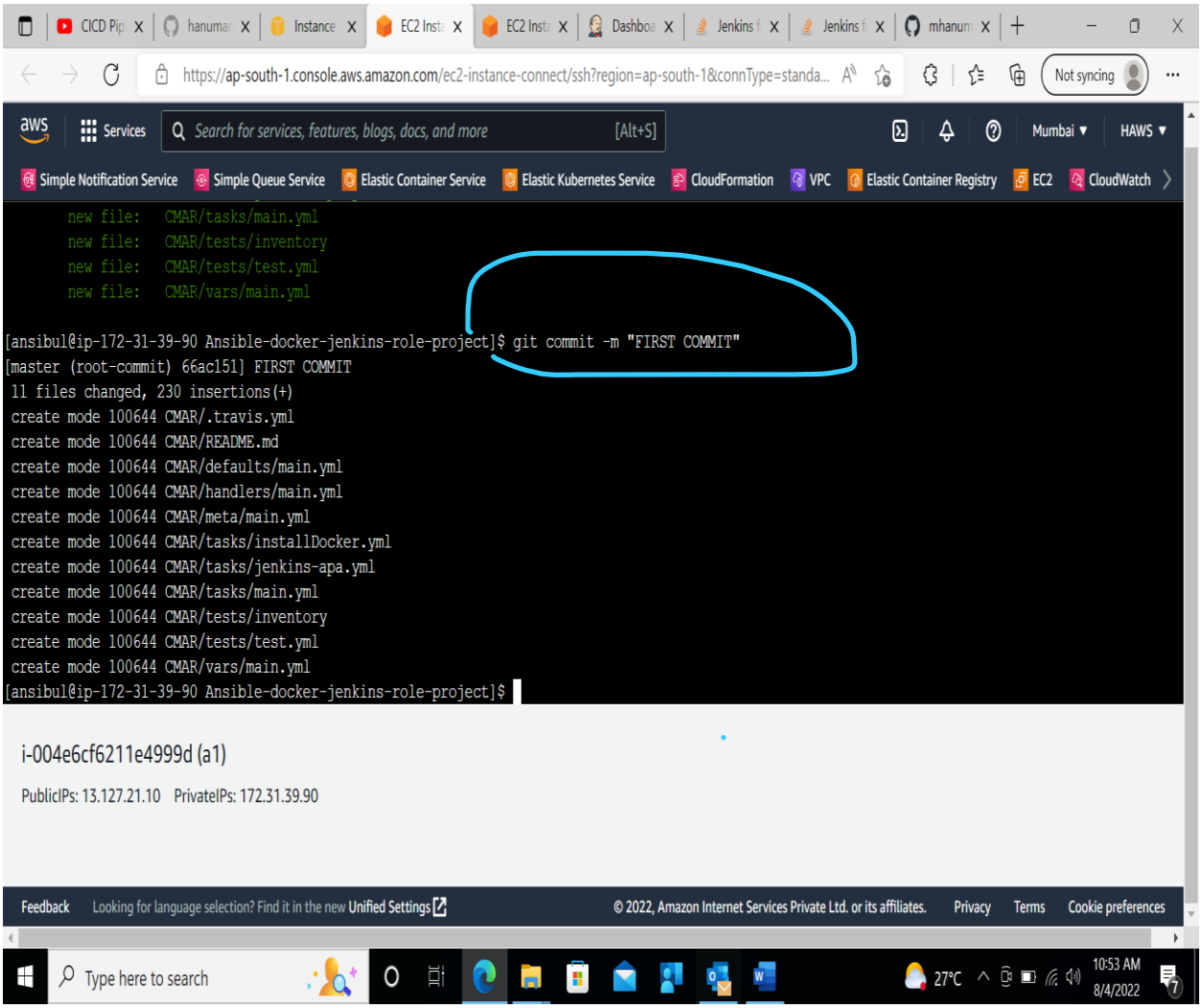
Staging files are ready to commit into the local repository



The command that records or snapshots the file permanently in the version history

git commit

git commit -m “[commit message]”



To push the contents committed to our local repository to the git hub repository we should use the following commands in red color
Before that we need to check whether we have configured out git hub repo, if not we need to use the commands in yellow to configure to our github repository

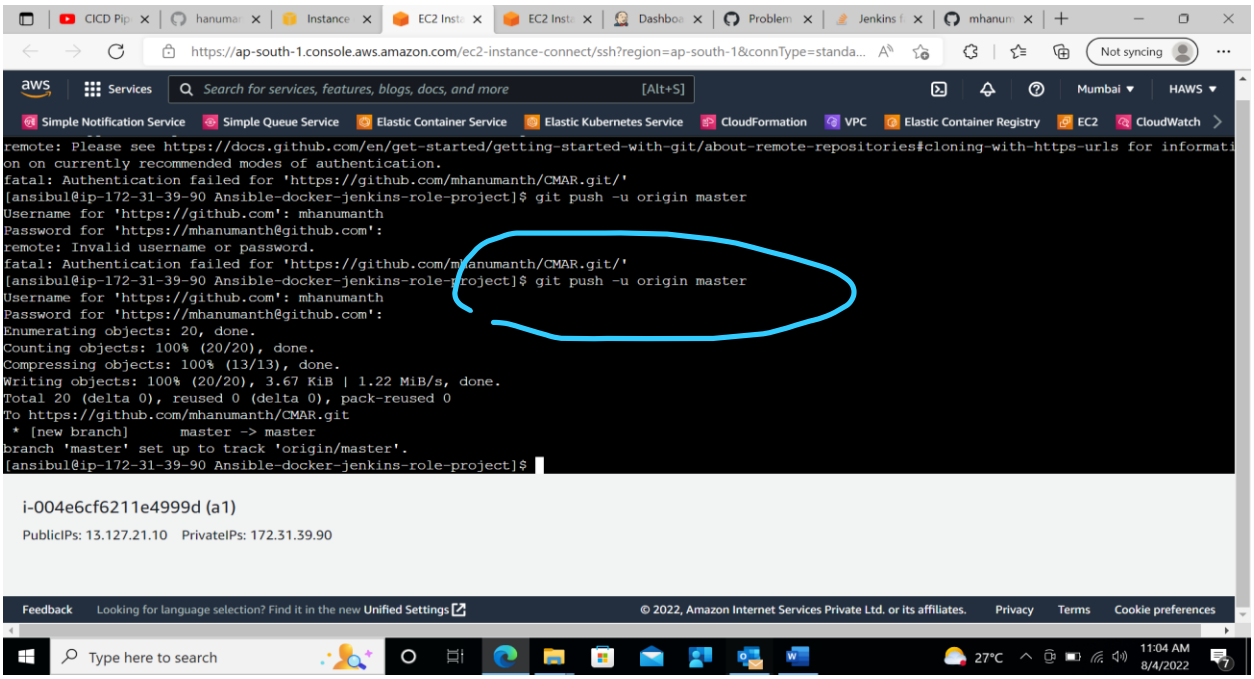
git config

git config –global user.name “[name]”

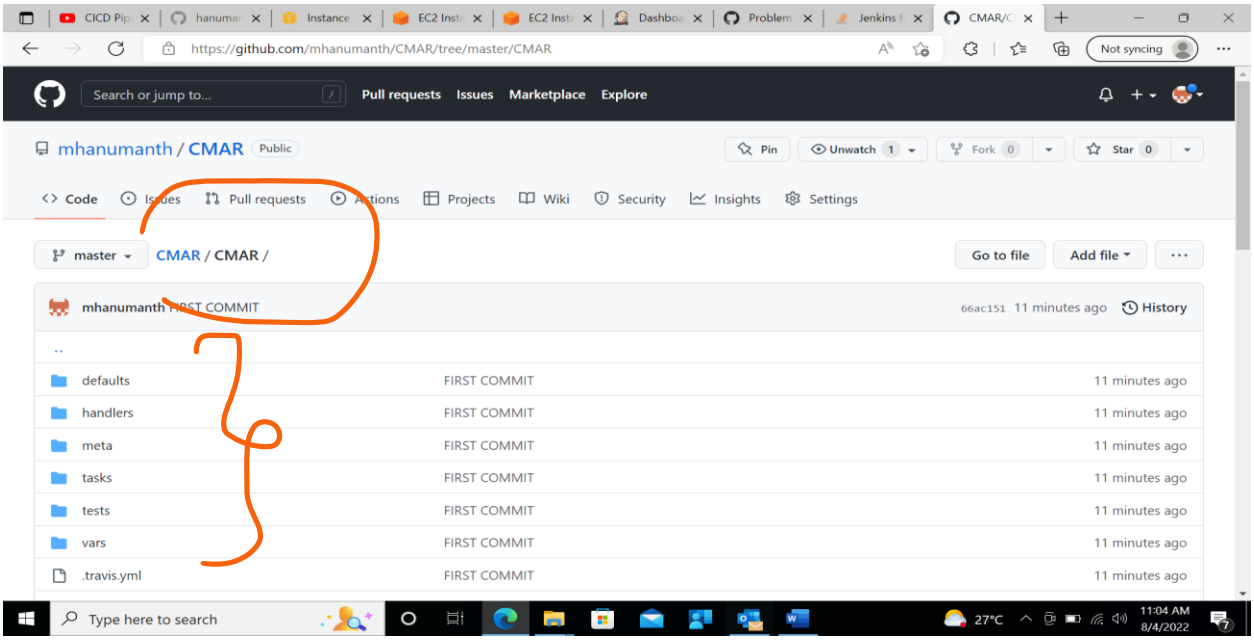
git config –global user.email “[email address]”

git remote add origin https://github.com/mhanumanth/git-repo-name.git

git push -u origin main



One git push is successful,we can view the committed files in repository



TASK5: BUILDING CICD PIPELINE USING JENKINS

Jenkins Pipeline is a stack of Jenkins plugins and other tools which helps implementing and continuous integration and delivery pipelines.

In this pipeline it includes 3 stages

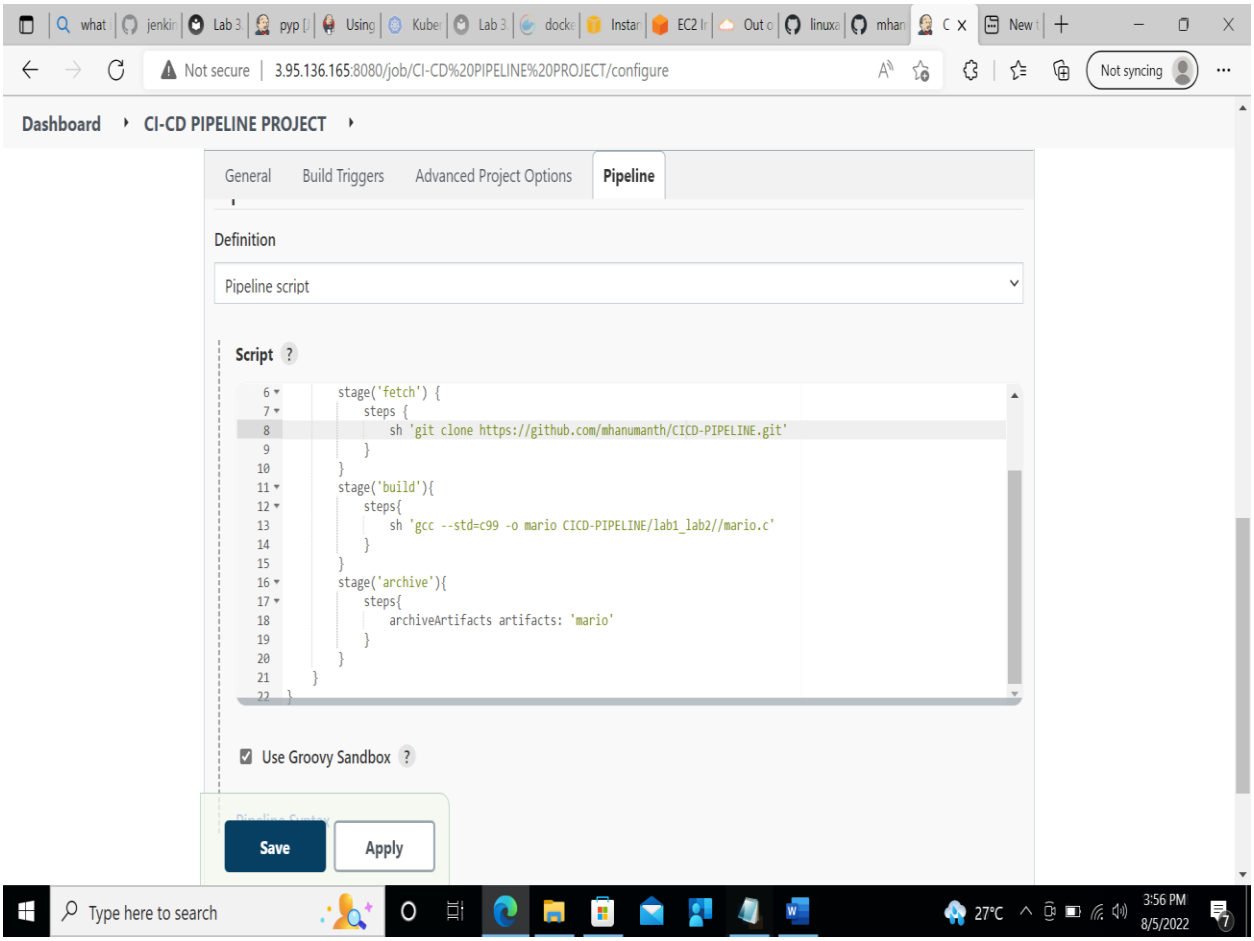
1.A Preparation stage where we clone code from git repo

https://github.com/mhanumanth/CICD-PIPELINE.git

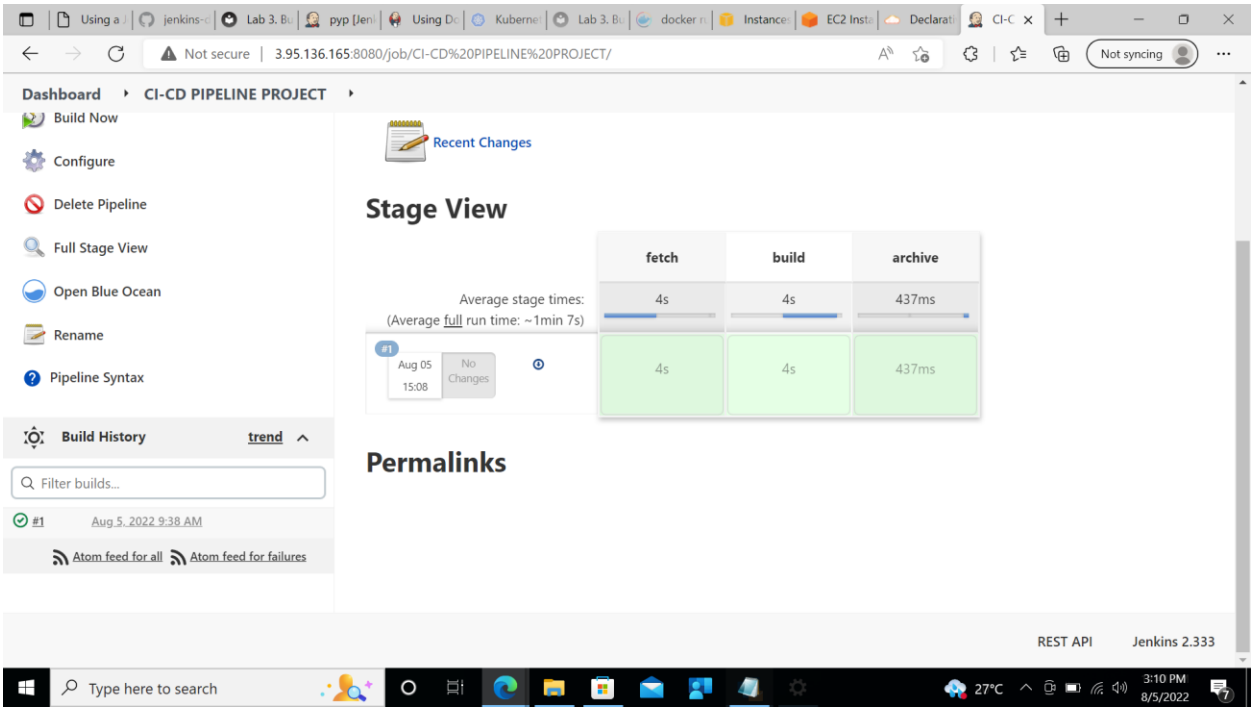
2.A Build stage where actual code is assembled and built providing an executable binary.

3. An archiving stage where we preserve the output of our Build stage,we use archiveArtifacts

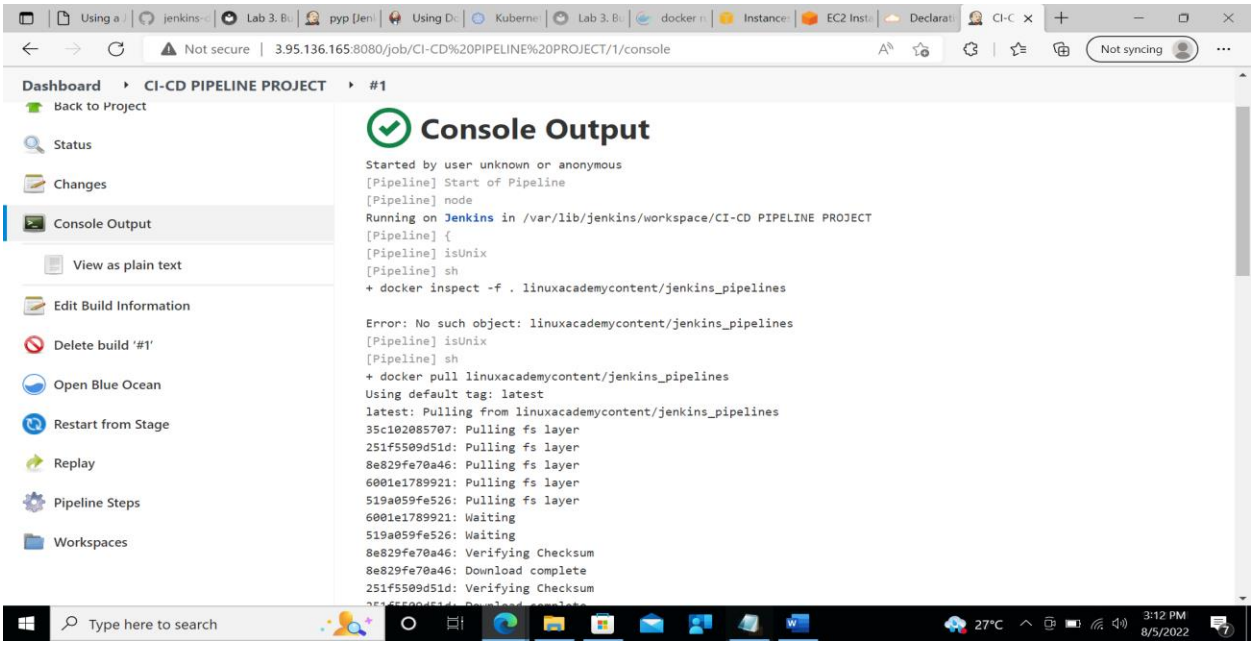
The pipeline script -



Buid stage view



Console Output-



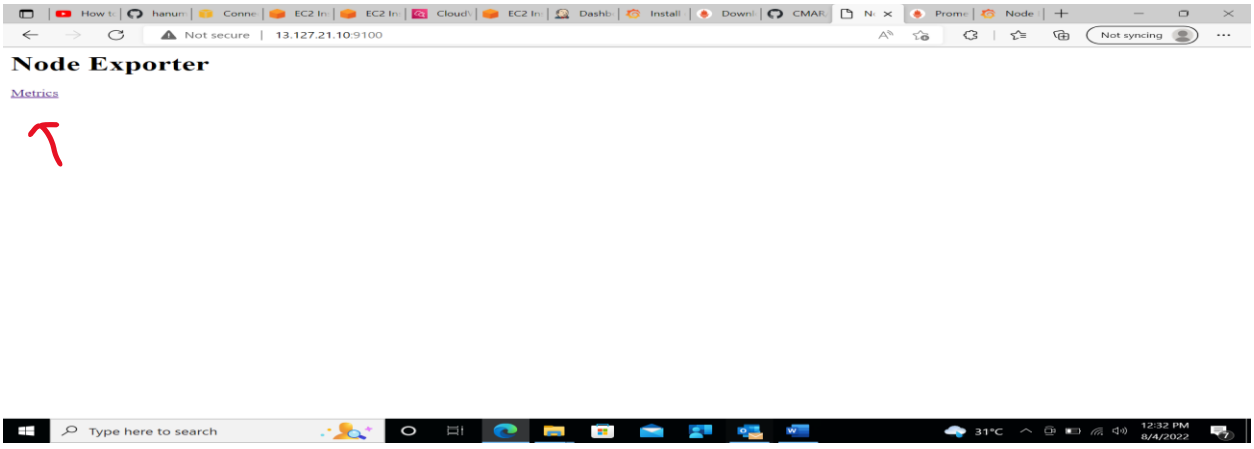
Git URL: https://github.com/mhanumanth/CICD-PIPELINE.git

TASK6: BUILDING MONITORING TASK

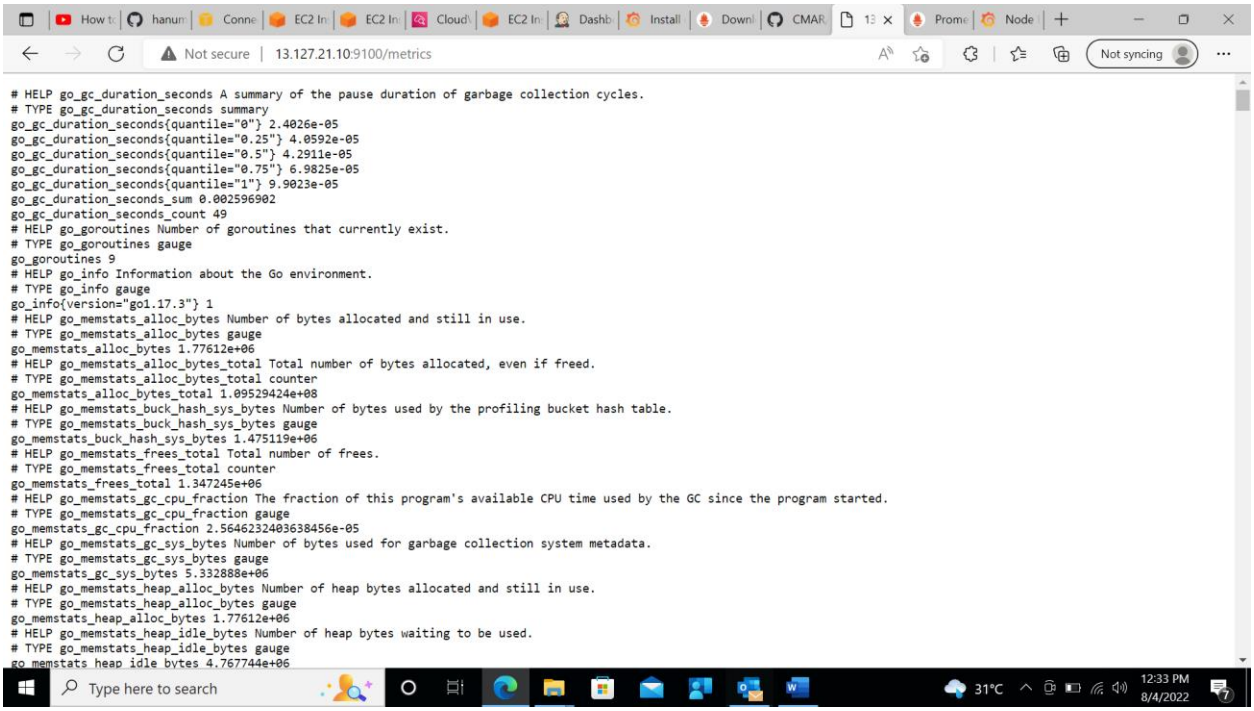
Monitoring tools like Prometheus and Grafana are both built for time-series data. **Prometheus excels in metric data collection, whereas Grafana champions metric visualizations**. Both tools are open source, free and have vibrant communities

With Monitoring assess the performance of servers and applications. By monitoring we can detect the issues, prevent the failures and helps in capacity planning etc.

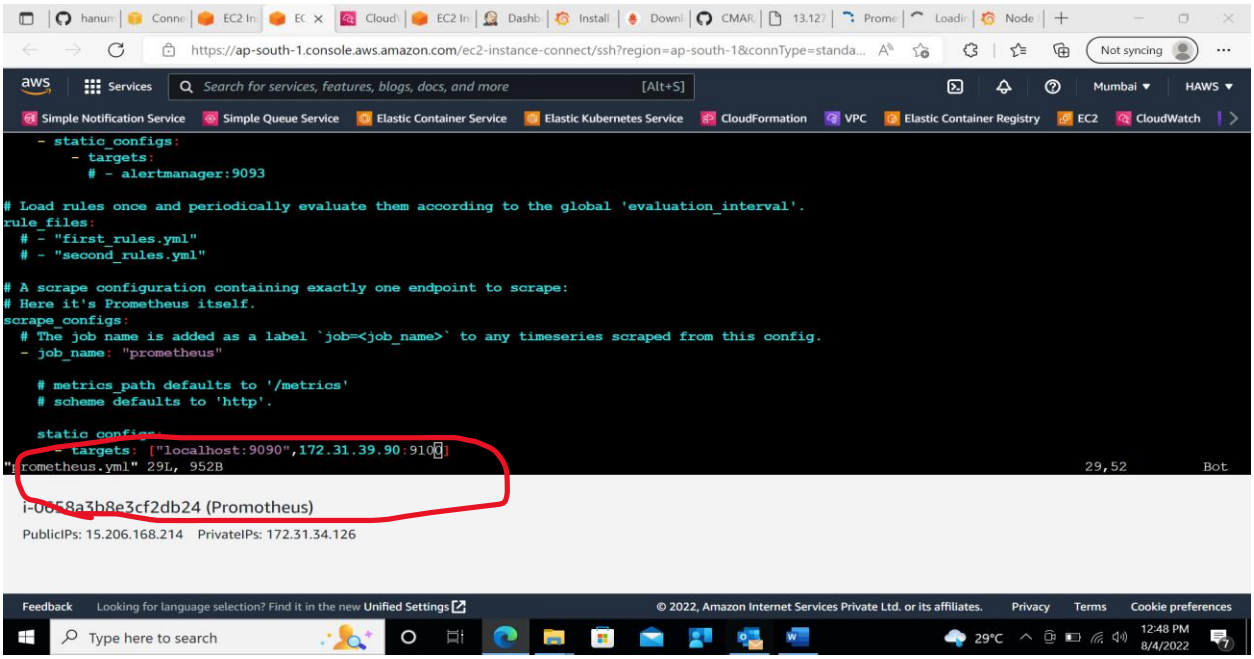
I had installed node_port on the VM to be monitored, which will export all the metrics of this VM



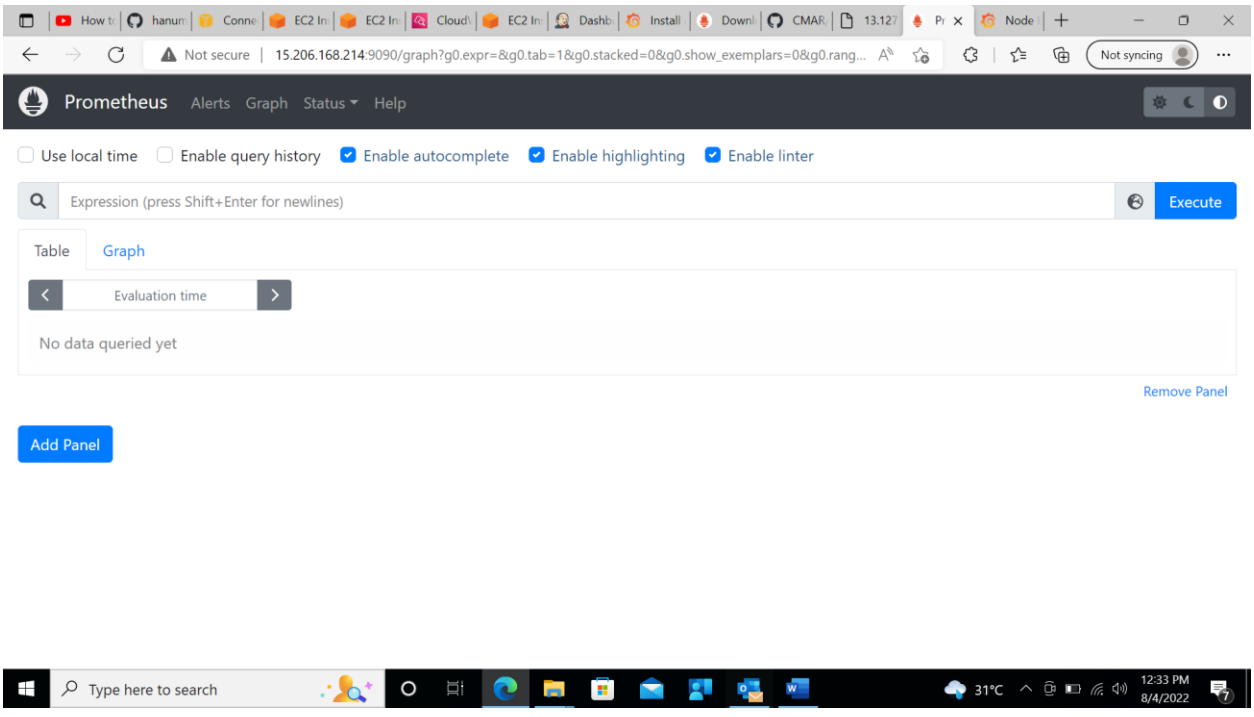
After clicking on the metrics



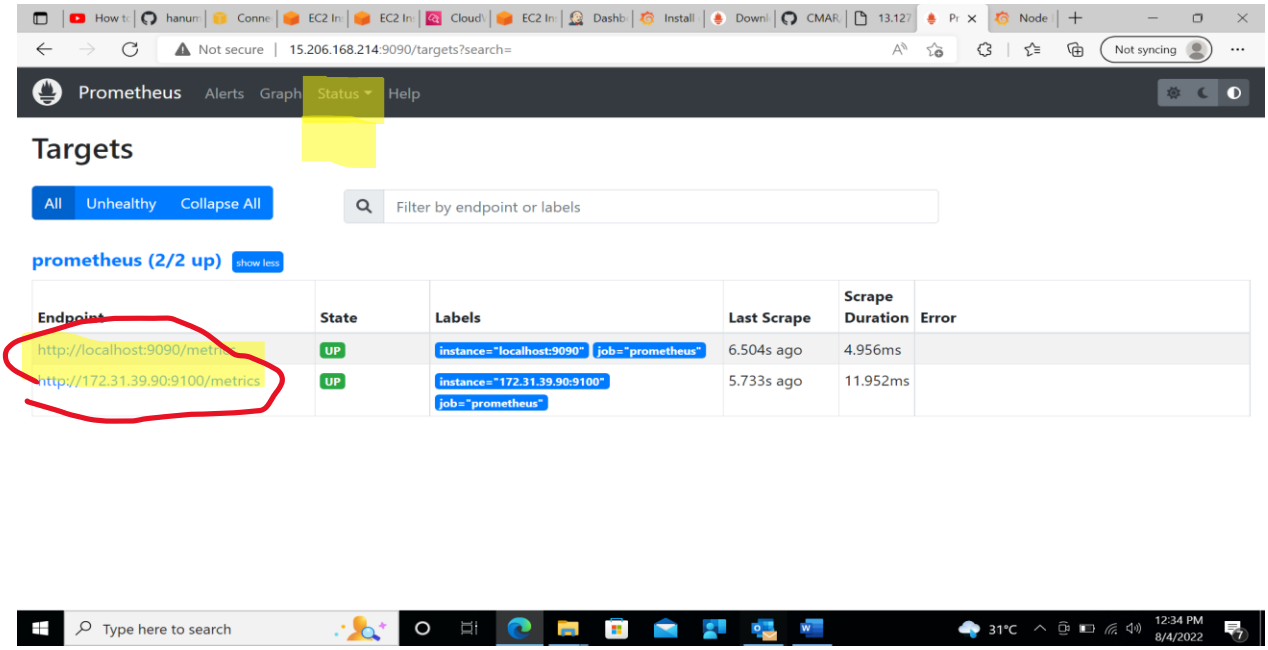
In Prometheus.yml file we need to add the ip address of the system to be monitored



Prometheus runs on the port 9090



After selecting the targets in the status dropdown we can observe the monitoring machines



In Graphana we can visualize the different metrics as shown below