

## 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**.

```
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://$var/run/docker.sock/v1.24/images/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-36-202:~/dockr-proj/docker-frontend-backend-db$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
docker-frontend-backend-db_web    latest   6b1ee42a2c2c  2 days ago   332MB
docker-frontend-backend-db_api    latest   4a312a798fc0  2 days ago   93.1MB
mongo                latest   c2262c276387  3 days ago   696MB
f2hex/redmine        latest   303e6b8fdf1a  6 years ago  448MB
f2hex/nginx          latest   1753a70e86a2  6 years ago  227MB
f2hex/postgres       latest   383e1968aac1  6 years ago  265MB
ubuntu@ip-172-31-36-202:~/dockr-proj/docker-frontend-backend-db$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED          STATUS           PORTS
NAMES
d73e7cf57ee1        docker-frontend-backend-db_web    "docker-entrypoint.s..."  2 days ago     Up 2 minutes   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
docker-frontend-backend-db_web_1
db3e6df53199        docker-frontend-backend-db_api    "docker-entrypoint.s..."  2 days ago     Up 2 minutes   0.0.0.0:3001->3001/tcp, :::3001->3001/tcp
docker-frontend-backend-db_api_1
85898ff2610c        mongo                "docker-entrypoint.s..."  2 days ago     Up 11 minutes   27017/tcp
docker-frontend-backend-db_mongo_1
ubuntu@ip-172-31-36-202:~/dockr-proj/docker-frontend-backend-db$
```

i-00cc27e0f3b598f1c (DOKER)

PublicIPs: 13.127.105.159 PrivateIPs: 172.31.36.202



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
```

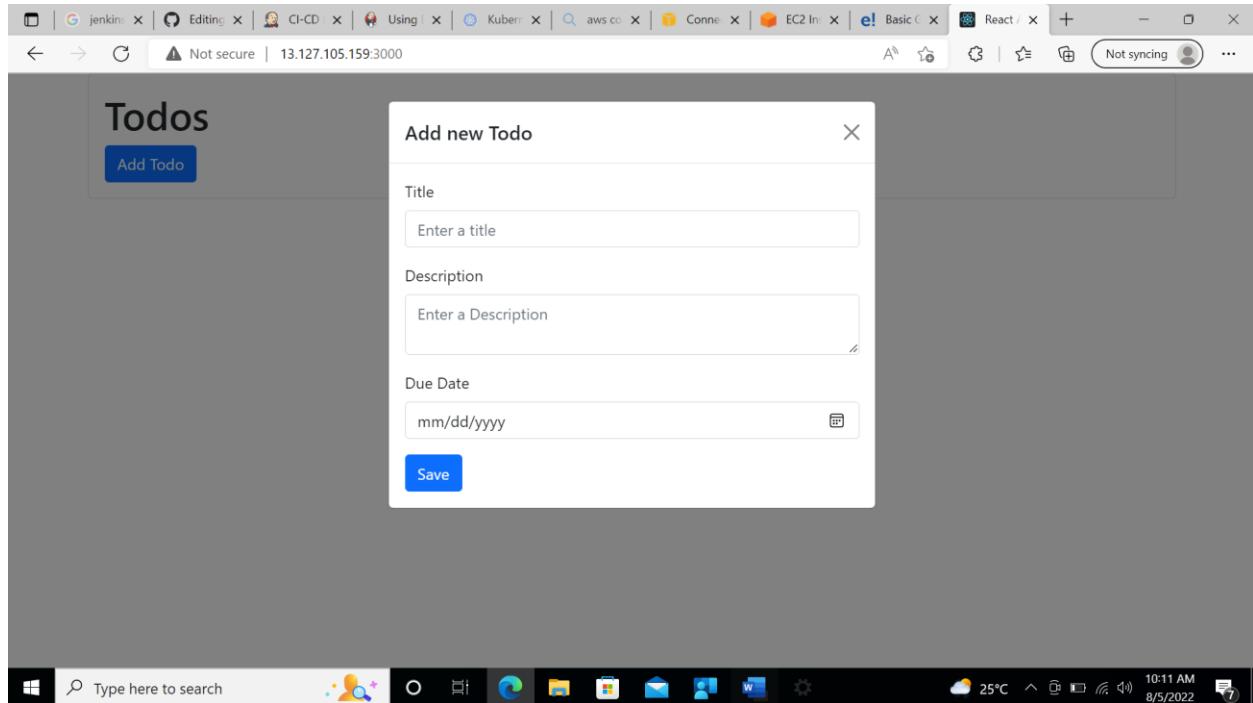
Docker Compose is **a tool that was developed to help define and share multi-container applications**. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down.

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

The screenshot shows a terminal window on an AWS CloudWatch interface. At the top, there's a navigation bar with various AWS services like Jenkins, CI-CD, Using, Kuber, AWS CloudWatch, EC2 Instances, Basic, React, etc. Below the bar, the URL is https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-00cc27e0f3b598f1c. The main area has a search bar and a filter for 'Simple Notification Service'. The terminal output shows a command to check Docker daemon socket permissions followed by a list of Docker images. A red circle highlights the 'COMMAND' column in a table of running containers. Another red circle highlights the 'IMAGE' column. A large red curly brace groups the 'COMMAND' and 'IMAGE' columns together, indicating they are part of the same table structure. The table includes columns for CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, and PORTS. The status for most containers is 'Up 2 minutes' except for one which is 'Up 11 minutes'. The ports listed are 0.0.0.0:3000->3000/tcp, 0.0.0.0:3001->3001/tcp, and 27017/tcp. At the bottom, it shows the instance ID i-00cc27e0f3b598f1c (DOKER) and the public/private IP addresses 13.127.105.159 and 172.31.36.202 respectively. The bottom of the screen shows the Windows taskbar with the Start button, a search bar, and various pinned icons like File Explorer, Edge, Mail, and Settings. The system tray shows the date (8/5/2022), time (10:12 AM), battery level (25%), and network status.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d73e7cf5ee1	docker-frontend-backend-db_web	"docker-entrypoint.s..."	2 days ago	Up 2 minutes	0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
docker-frontend-backend-db_1	docker-frontend-backend-db_api	"docker-entrypoint.s..."	2 days ago	Up 2 minutes	0.0.0.0:3001->3001/tcp, :::3001->3001/tcp
db3e6df53199	mongo	"docker-entrypoint.s..."	2 days ago	Up 11 minutes	27017/tcp
8589ff2610c	docker-frontend-backend-db_mongo_1	"docker-entrypoint.s..."	2 days ago	Up 11 minutes	

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
```

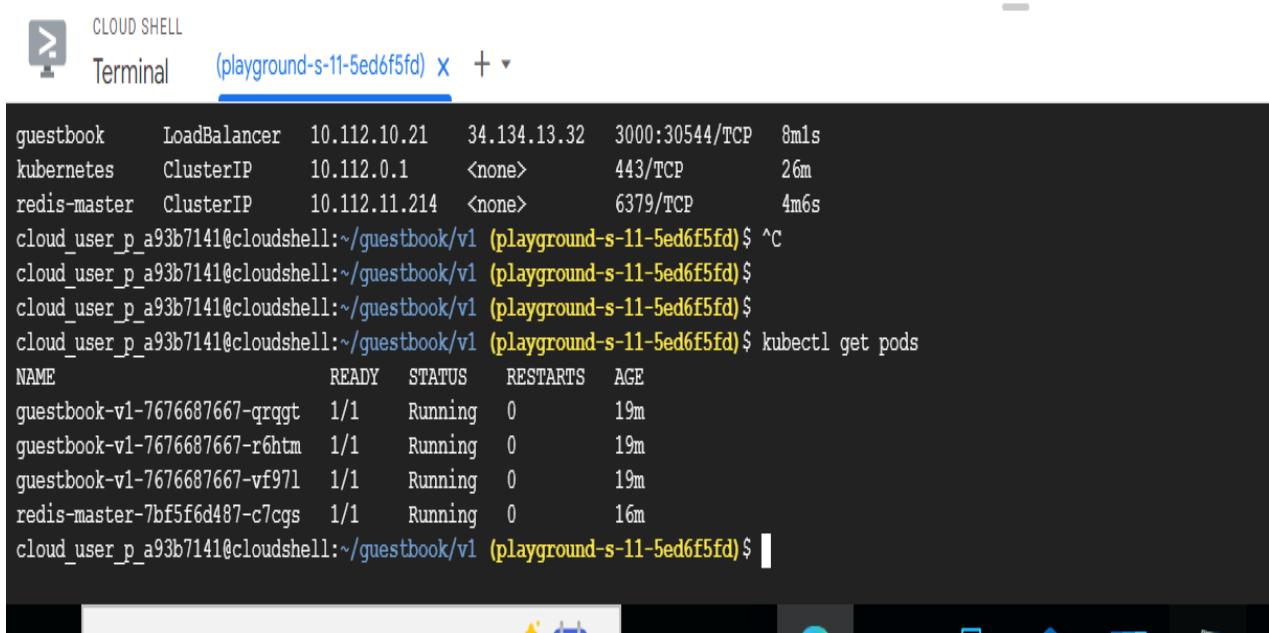
We define a Service object to expose the deployment to external clients.

```
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 observe the pods running, we can use the command

kubectl get pods

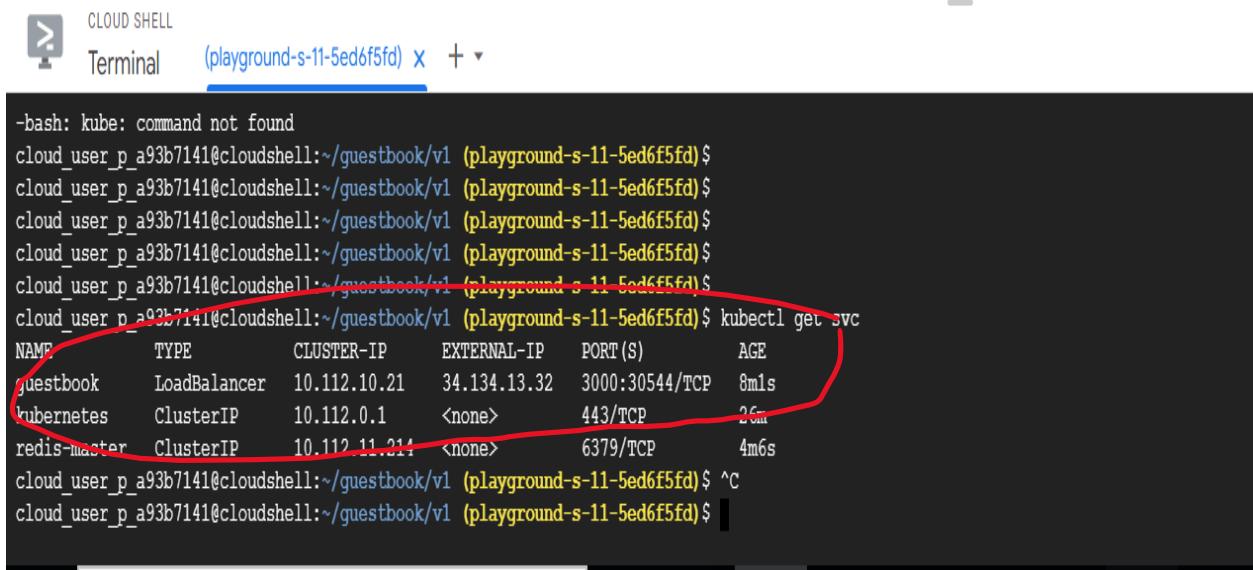


```
CLOUD SHELL
Terminal (playground-s-11-5ed6f5fd) + ▾

guestbook   LoadBalancer  10.112.10.21  34.134.13.32  3000:30544/TCP  8m1s
kubernetes  ClusterIP    10.112.0.1    <none>        443/TCP      26m
redis-master ClusterIP   10.112.11.214 <none>        6379/TCP     4m6s
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$ ^C
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
guestbook-v1-7676687667-qrqgt  1/1    Running   0          19m
guestbook-v1-7676687667-r6htm  1/1    Running   0          19m
guestbook-v1-7676687667-vf971  1/1    Running   0          19m
redis-master-7bf5f6d487-c7cgs  1/1    Running   0          16m
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
```

To get the services running we can use the command

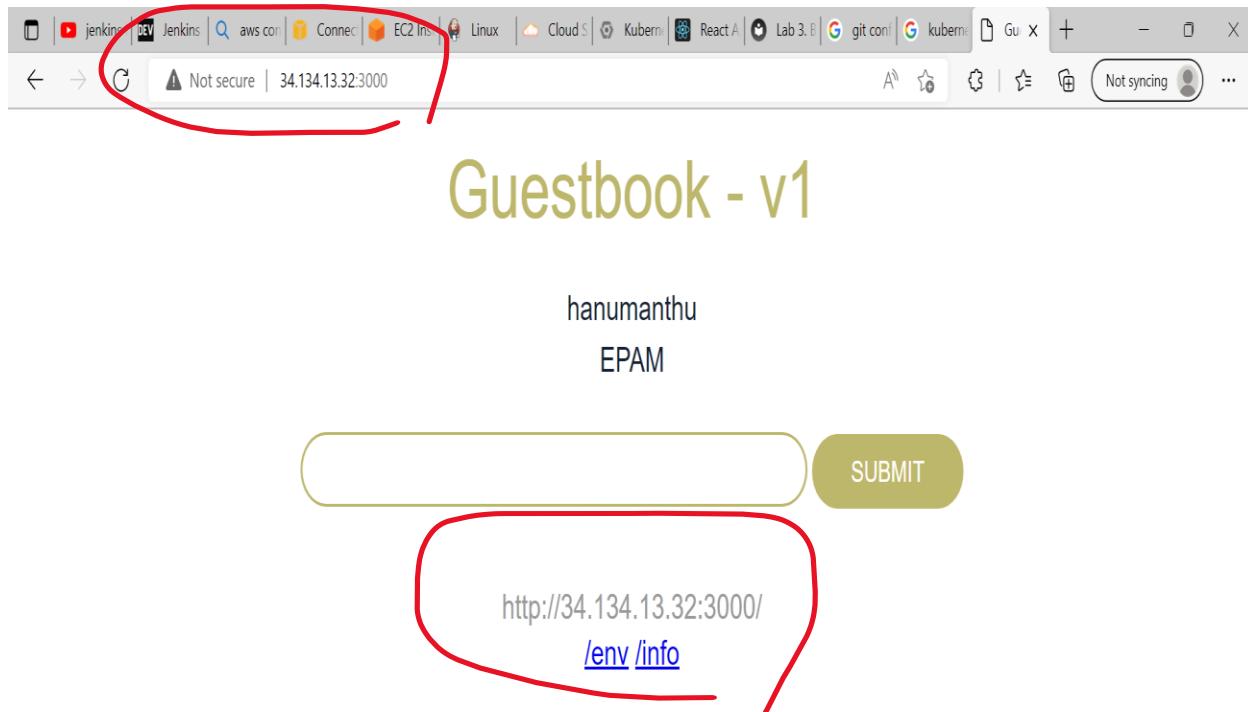
kubectl get service or kubectl get svc



```
CLOUD SHELL
Terminal (playground-s-11-5ed6f5fd) + ▾

-bash: kube: command not found
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$ kubectl get svc
NAME        TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
guestbook   LoadBalancer  10.112.10.21  34.134.13.32  3000:30544/TCP  8m1s
kubernetes  ClusterIP   10.112.0.1    <none>        443/TCP      26m
redis-master ClusterIP   10.112.11.214 <none>        6379/TCP     4m6s
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$ ^C
cloud_user_p_a93b7141@cloudshell:~/guestbook/v1 (playground-s-11-5ed6f5fd)$
```

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.

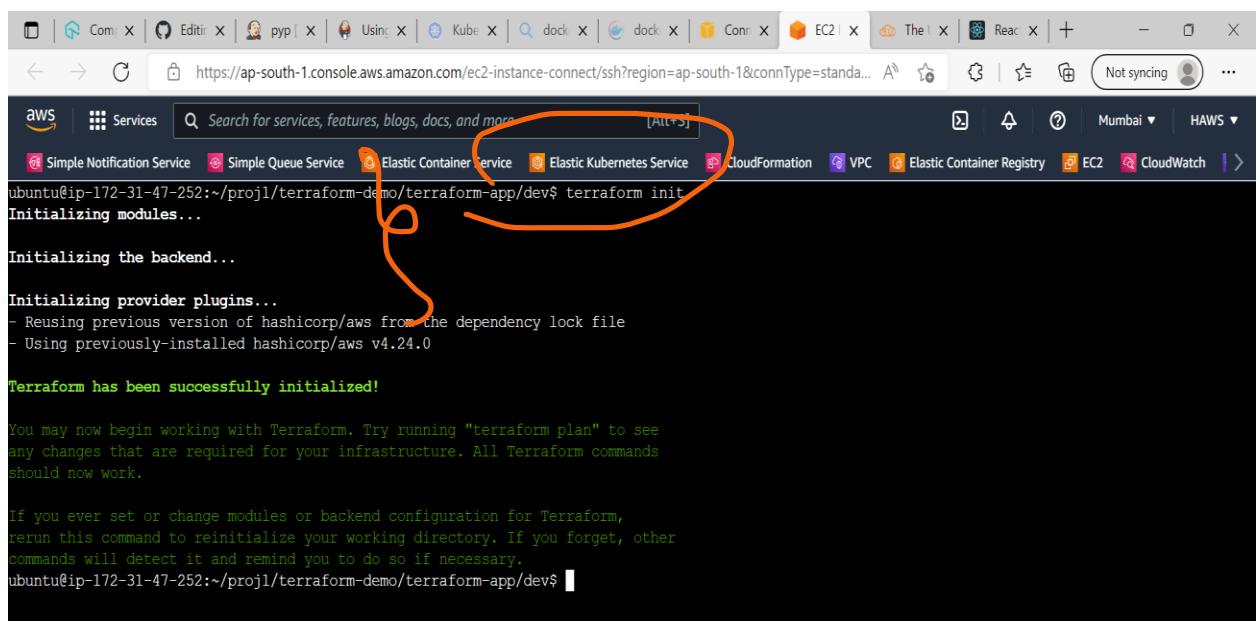
In this task iam going to crate, destroy three resources to aws account

1.ec2-instance

2.vpc

3.subnet

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



```
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform init
Initializing modules...
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.24.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$
```

i-03bd7abf015fdbf84 (Terraform)  
PublicIPs: 52.66.211.128 PrivateIPs: 172.31.47.252



## terraform fmt #format code per HCL canonical standard

A screenshot of a terminal window titled "Ubuntu" from the AWS CloudShell interface. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard>. The terminal shows the command `terraform fmt` being run in a directory containing `main.tf` and `vars.tf`. A blue oval highlights the AWS navigation bar at the top.

```
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform fmt
```

```
main.tf
vars.tf
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$
```

## terraform validate #validate code for syntax

A screenshot of a terminal window titled "Ubuntu" from the AWS CloudShell interface. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard>. The terminal shows the command `terraform validate` being run, resulting in the message "Success! The configuration is valid." A blue oval highlights the AWS navigation bar at the top.

```
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform validate
```

```
Success! The configuration is valid.

ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$
```

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

A screenshot of a terminal window titled "Ubuntu" from the AWS CloudShell interface. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard>. The terminal shows the command `terraform plan` being run. It displays the execution plan, indicating actions like "create" for resources defined in the `aws_instance` module. A blue oval highlights the AWS navigation bar at the top.

```
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform plan
```

```
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# module.my_ec2.aws_instance.web[0] will be created
+ resource "aws_instance" "web" {
    + ami                                = "ami-02354e95b39ca8dec"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
```

```

+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id                = (known after apply)
+ owner_id                           = (known after apply)
+ tags                               = {
    + "Comment" = "Created by Terraform"
    + "Name"    = "DevVpc"
}
+ tags_all                           = {
    + "Comment" = "Created by Terraform"
    + "Name"    = "DevVpc"
}

Plan: 3 to add, 0 to change, 0 to destroy

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ 

```

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

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

```

ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform apply --auto-approve

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  
 + create

Terraform will perform the following actions:

```

# module.my_ec2.aws_instance.web[0] will be created
+ resource "aws_instance" "web" {
    + ami
    + arn
    + associate_public_ip_address
    + availability_zone
    + cpu_core_count
    + cpu_threads_per_core
    + disable_api_stop
    + disable_api_termination
    + ebs_optimized
    + get_password_data
    + host_id
    + id
    = "ami-02354e95b39ca8dec"
    = (known after apply)
    = false
    = (known after apply)
    = (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

AWS CloudWatch terminal window showing Terraform output. A blue oval highlights the first few lines of the code.

```
# module.my_vpc.aws_vpc.main will be created
resource "aws_vpc" "main" {
```

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

26°C 11:23 AM 8/5/2022

AWS CloudWatch terminal window showing Terraform output. A blue oval highlights the first few lines of the code.

```
# module.my_vpc.aws_subnet.main will be created
resource "aws_subnet" "main" {
```

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

26°C 11:22 AM 8/5/2022

```

aws | Services | Search for services, features, blogs, docs, and more [Alt+S]
Simple Notification Service Simple Queue Service Elastic Container Service Elastic Kubernetes Service CloudFormation VPC Elastic Container Registry EC2 CloudWatch >
@ Simple Notification Service Simple Queue Service Elastic Container Service Elastic Kubernetes Service CloudFormation VPC Elastic Container Registry EC2 CloudWatch >
  + "Comment" = "Created by Terraform"
  + "Name"    = "DevVpc"
}

Plan: 3 to add, 0 to change, 0 to destroy.

module.my_vpc.aws_vpc.main: Creating...
module.my_vpc.aws_vpc.main: Creation complete after 3s [id=vpc-034c5b4ab76f7990f]
module.my_vpc.aws_subnet.main: Creating...
module.my_vpc.aws_subnet.main: Creation complete after 2s [id=subnet-0cc07c1a18d3c1955]
module.my_ec2.aws_instance.web[0]: Creating...
module.my_ec2.aws_instance.web[0]: Still creating... [10s elapsed]
module.my_ec2.aws_instance.web[0]: Still creating... [20s elapsed]
module.my_ec2.aws_instance.web[0]: Still creating... [30s elapsed]
module.my_ec2.aws_instance.web[0]: Still creating... [40s elapsed]
module.my_ec2.aws_instance.web[0]: Still creating... [50s elapsed]
module.my_ec2.aws_instance.web[0]: Creation complete after 55s [id=i-036b14355db2706d0]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ 

```

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

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

```

aws | Services | Search for services, features, blogs, docs, and more [Alt+S]
Simple Notification Service Simple Queue Service Elastic Container Service Elastic Kubernetes Service CloudFormation VPC Elastic Container Registry EC2 CloudWatch >
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ terraform destroy --auto-approve
module.my_vpc.aws_vpc.main: Refreshing state... [id=vpc-034c5b4ab76f7990f]
module.my_vpc.aws_subnet.main: Refreshing state... [id=subnet-0cc07c1a18d3c1955]
module.my_ec2.aws_instance.web[0]: Refreshing state... [id=i-036b14355db2706d0]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# module.my_ec2.aws_instance.web[0] will be destroyed
- resource "aws_instance" "web" {
  ami                                = "ami-02354e95b39ca8dec" -> null
  arn                                = "arn:aws:ec2:us-east-1:47532004584:instance/i-036b14355db2706d0" -> null
  associate_public_ip_address          = false -> null
  availability_zone                   = "us-east-1a" -> null
  cpu_core_count                      = 1 -> null
  cpu_threads_per_core                = 1 -> null
  disable_api_stop                    = false -> null
  disable_api_termination             = false -> null
  ebs_optimized                       = false -> 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

A screenshot of a web browser displaying the AWS CloudWatch interface. The URL is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/shell?region=ap-south-1&connType=standard>. The page shows a Terraform configuration for an AWS Subnet. A blue oval highlights the first few lines of the code:

```
# module.my_vpc.aws_subnet.main will be destroyed
resource "aws_subnet" "main" {
```

The configuration includes various parameters such as `arn`, `assign_ip_address_on_creation`, `availability_zone`, `availability_zone_id`, `cidr_block`, `enable_dns64`, `enable_resource_name_dns_a_record_on_launch`, `enable_resource_name_dns_aaaa_record_on_launch`, `id`, `ipv6_native`, `map_customer_owned_ip_on_launch`, `map_public_ip_on_launch`, `owner_id`, `private_dns_hostname_type_on_launch`, and `tags`.

Below the code, it shows the instance details:

i-03bd7abf015fdbf84 (Terraform)  
Public IPs: 52.66.211.128 Private IPs: 172.31.47.252

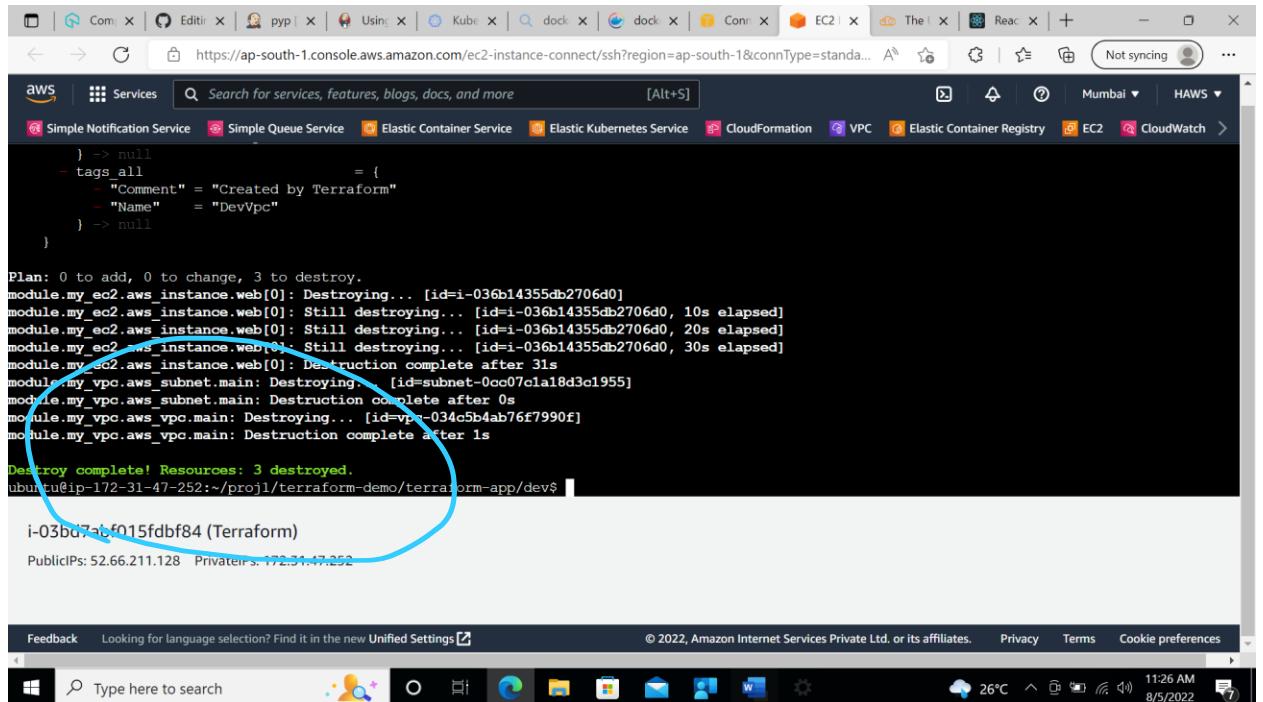
A screenshot of a web browser displaying the AWS CloudWatch interface. The URL is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/shell?region=ap-south-1&connType=standard>. The page shows a Terraform configuration for an AWS VPC. A blue oval highlights the first few lines of the code:

```
# module.my_vpc.aws_vpc.main will be destroyed
resource "aws_vpc" "main" {
```

The configuration includes parameters like `arn`, `assign_generated_ipv6_cidr_block`, `cidr_block`, `default_network_acl_id`, `default_route_table_id`, `default_security_group_id`, `dhcp_options_id`, `enable_classiclink`, `enable_classiclink_dns_support`, `enable_dns_hostnames`, `enable_dns_support`, `id`, `instance_tenancy`, and `ipv6_netmask_length`.

Below the code, it shows the instance details:

i-03bd7abf015fdbf84 (Terraform)  
Public IPs: 52.66.211.128 Private IPs: 172.31.47.252



A screenshot of a Windows desktop environment. The main window is a browser displaying the AWS CloudWatch logs for a Terraform destroy operation. The URL is <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/shell?region=ap-south-1&connType=standard>. The logs show the destruction of three resources: an EC2 instance, a VPC subnet, and a VPC. A blue oval highlights the log output. The taskbar at the bottom shows several open applications, including a terminal window with the command `terraform destroy` running.

```
aws | Com | Edit | pyp | Using | Kube | dock | dock | Conn | EC2 | The | React | + | - | □ | X
← → ⌂ https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/shell?region=ap-south-1&connType=standard... A Not syncing ... Mumbai HAWS
aws Services Search for services, features, blogs, docs, and more [Alt+S]
Simple Notification Service Simple Queue Service Elastic Container Service Elastic Kubernetes Service CloudFormation VPC Elastic Container Registry EC2 CloudWatch
tags all = {
  "Comment" = "Created by Terraform"
  "Name" = "DevVpc"
} -> null
}
Plan: 0 to add, 0 to change, 3 to destroy.
module.my_ec2.aws_instance.web[0]: Destroying... [id=i-036b14355db2706d0]
module.my_ec2.aws_instance.web[0]: Still destroying... [id=i-036b14355db2706d0, 10s elapsed]
module.my_ec2.aws_instance.web[0]: Still destroying... [id=i-036b14355db2706d0, 20s elapsed]
module.my_ec2.aws_instance.web[0]: Still destroying... [id=i-036b14355db2706d0, 30s elapsed]
module.my_ec2.aws_instance.web[0]: Destruction complete after 31s
module.my_vpc.aws_subnet.main: Destroying... [id=subnet-0cc07c1a18d3c1955]
module.my_vpc.aws_subnet.main: Destruction complete after 0s
module.my_vpc.aws_vpc.main: Destroying... [id=vpc-034c5b4ab76f7990f]
module.my_vpc.aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-47-252:~/proj1/terraform-demo/terraform-app/dev$ i-036b14355db2706d0 (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
Windows Type here to search 26°C 11:26 AM 8/5/2022
```

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

## TASK3:CONFIGURATION MANAGEMENT USING 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

The screenshot shows a browser window with multiple tabs open. The active tab is titled "https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ss". The content area displays the directory structure of a Lambda function named "CMAR". The structure is as follows:

```
└── main.yml
   ├── files
   ├── handlers
   │   └── main.yml
   ├── meta
   │   └── main.yml
   ├── README.md
   ├── tasks
   │   ├── installDocker.yml
   │   ├── jenkins-apache.yml
   │   └── main.yml
   ├── templates
   ├── tests
   │   └── inventory
   │       └── test.yml
   └── vars
       └── main.yml
```

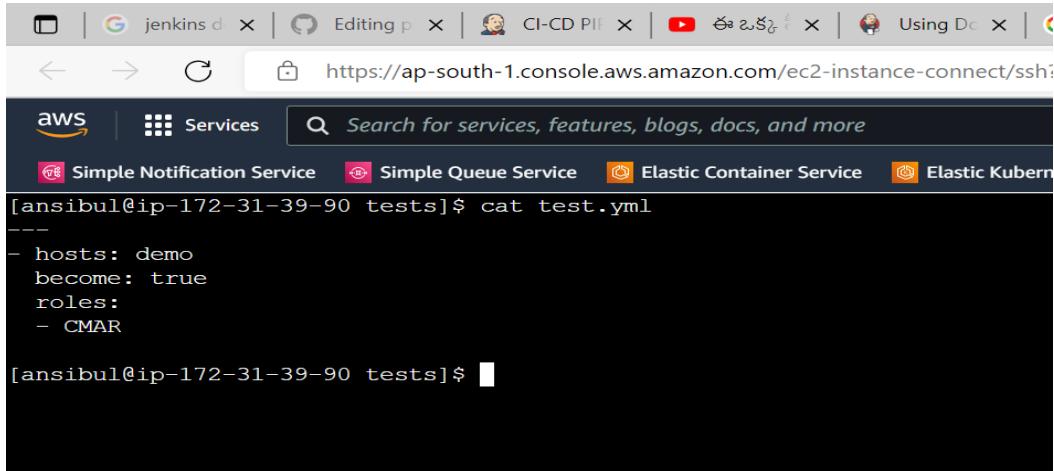
In this Task iam going to install Python,Ansible,Docker and Jenkins on 2 client Machines  
Whose ip's are defined in tests/inventory:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ss". The content area displays the contents of the "inventory" file located in the "tests" directory of the Lambda function "CMAR". The file contains the following content:

```
[demo]
172.31.37.77
172.31.41.152
```

An orange curly brace is drawn over the two IP addresses in the inventory file.

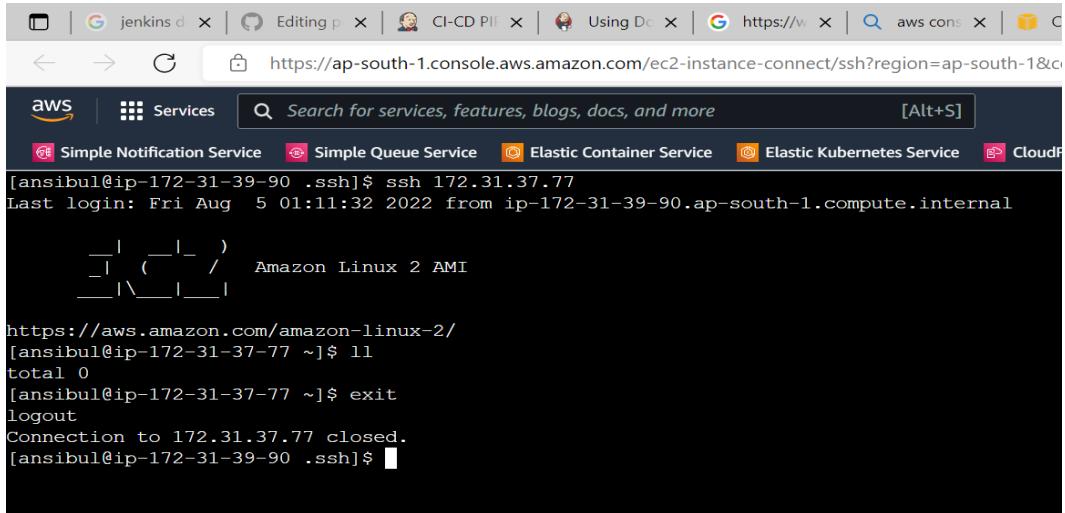
The playbook which we execute to install softwares on client machines is



```
[ansibul@ip-172-31-39-90 tests]$ cat test.yml
---
- hosts: demo
  become: true
  roles:
    - CMAR

[ansibul@ip-172-31-39-90 tests]$
```

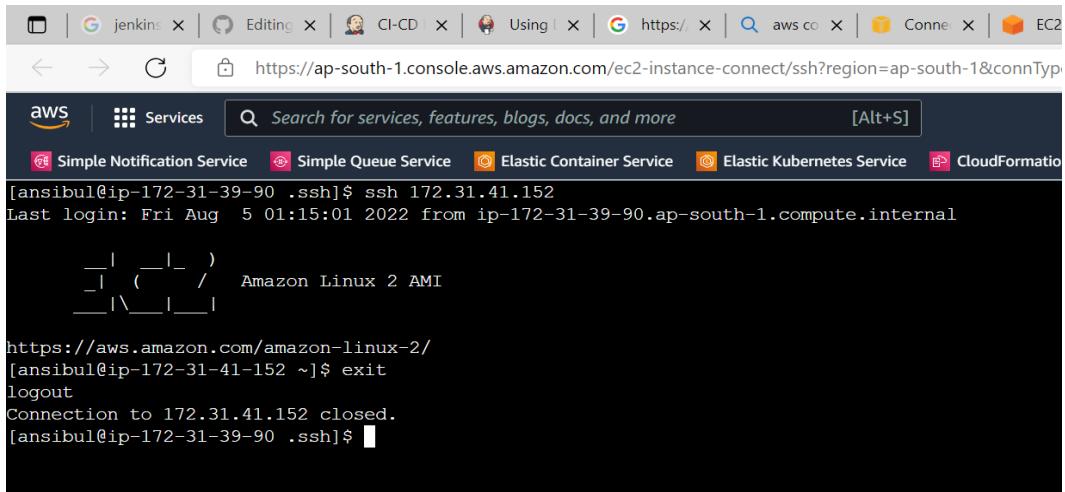
ssh connectivity to the client machines can be observed:@ 172.31.37.77



```
[ansibul@ip-172-31-39-90 .ssh]$ ssh 172.31.37.77
Last login: Fri Aug  5 01:11:32 2022 from ip-172-31-39-90.ap-south-1.compute.internal
 _ _ | _ _ )
 _ | ( _ /   Amazon Linux 2 AMI
 __| \__|__|_ |

https://aws.amazon.com/amazon-linux-2/
[ansibul@ip-172-31-37-77 ~]$ ll
total 0
[ansibul@ip-172-31-37-77 ~]$ exit
logout
Connection to 172.31.37.77 closed.
[ansibul@ip-172-31-39-90 .ssh]$
```

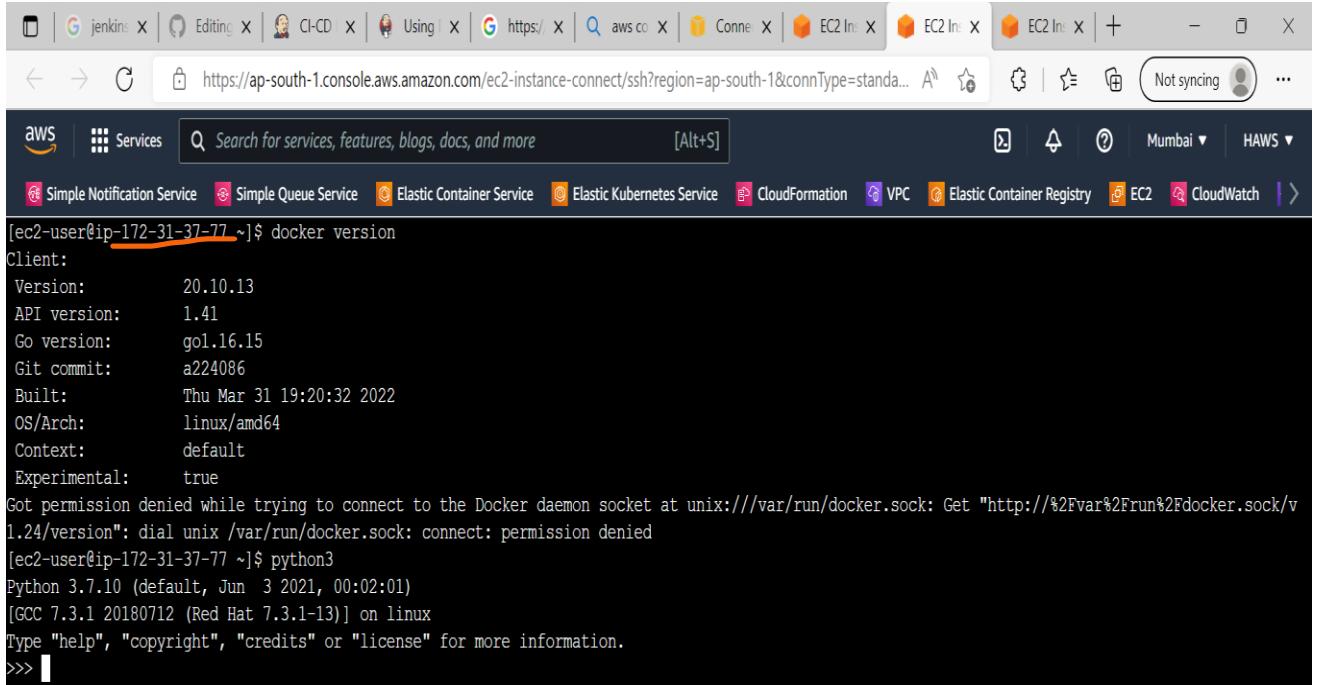
@ 172.31.41.152



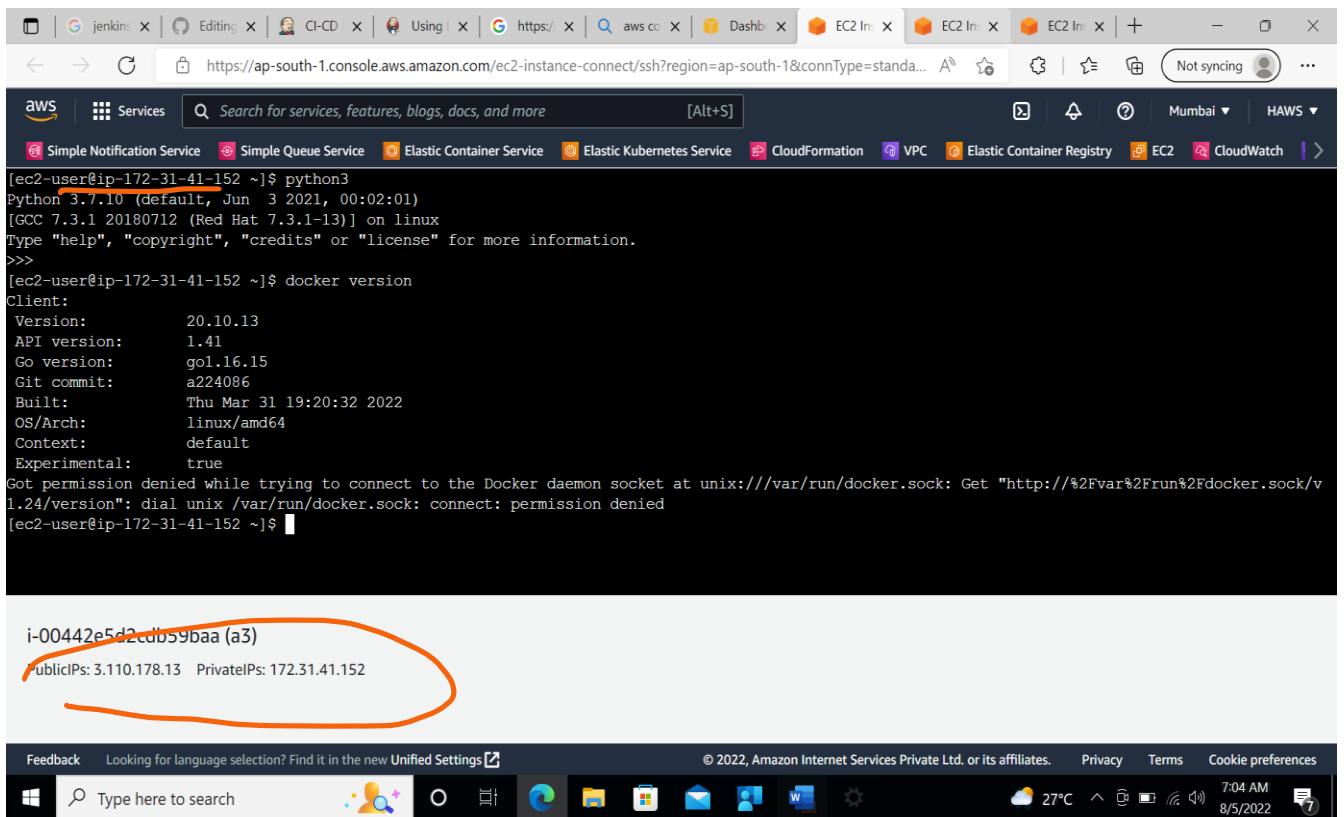
```
[ansibul@ip-172-31-39-90 .ssh]$ ssh 172.31.41.152
Last login: Fri Aug  5 01:15:01 2022 from ip-172-31-39-90.ap-south-1.compute.internal
 _ _ | _ _ )
 _ | ( _ /   Amazon Linux 2 AMI
 __| \__|__|_ |

https://aws.amazon.com/amazon-linux-2/
[ansibul@ip-172-31-41-152 ~]$ exit
logout
Connection to 172.31.41.152 closed.
[ansibul@ip-172-31-39-90 .ssh]$
```

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



```
[ec2-user@ip-172-31-37-77 ~]$ docker version
Client:
Version: 20.10.13
API version: 1.41
Go version: go1.16.15
Git commit: a224086
Built: Thu Mar 31 19:20:32 2022
OS/Arch: linux/amd64
Context: default
Experimental: true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://<ip>/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-37-77 ~]$ python3
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



```
[ec2-user@ip-172-31-41-152 ~]$ python3
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
[ec2-user@ip-172-31-41-152 ~]$ docker version
Client:
Version: 20.10.13
API version: 1.41
Go version: go1.16.15
Git commit: a224086
Built: Thu Mar 31 19:20:32 2022
OS/Arch: linux/amd64
Context: default
Experimental: true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://<ip>/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-41-152 ~]$
```

i-00442e5d2cd59baa (a3)  
PublicIPs: 3.110.178.13 PrivateIPs: 172.31.41.152

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

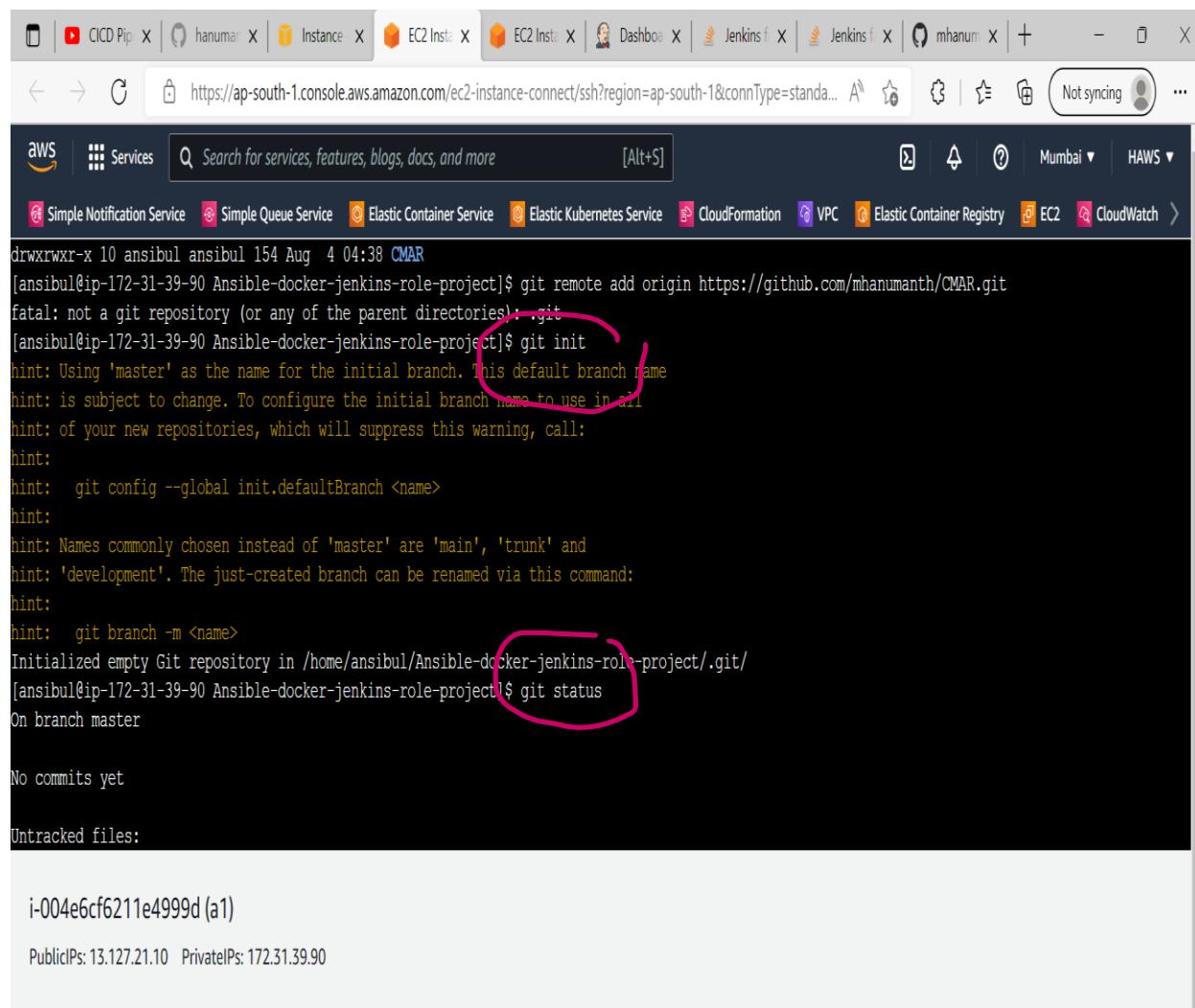
## TASK4: APPLICATION CODE MANAGEMENT USING GIT

**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 screenshot shows a terminal window with the AWS Lambda interface at the top. The terminal output is as follows:

```
drwxrwxr-x 10 ansibul ansibul 154 Aug  4 04:38 CMAR
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git remote add origin https://github.com/mhanumanth/CMAR.git
fatal: not a git repository (or any of the parent directories): .git
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ansibul/Ansible-docker-jenkins-role-project/.git/
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git status
On branch master

No commits yet

Untracked files:

  i-004e6cf6211e4999d (a1)

Public IPs: 13.127.21.10 Private IPs: 172.31.39.90
```

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

A screenshot of a terminal window on a Windows desktop. The terminal shows the output of a 'git status' command. Three specific lines of output are circled with blue ovals:

- '[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]\$ git status'
- '(use "git add <file>..." to include in what will be committed)
- 'nothing added to commit but untracked files present (use "git add" to track)'

Below the terminal, the Windows taskbar is visible, showing icons for File Explorer, Edge browser, Mail, and other system icons. The system tray shows the date (8/4/2022), time (10:52 AM), battery level (27%), and network status.

```
hint: git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /home/ansibul/Ansible-docker-jenkins-role-project/.git/
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .OMAR/

nothing added to commit but untracked files present (use "git add" to track)
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git add .
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git status
On branch master
```

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 screenshot shows the AWS CloudWatch Metrics console for a Jenkins pipeline build. The top navigation bar includes tabs for CICD Pipelines, hanuman, Instance, EC2 Instances, EC2 Instances, Dashboard, Jenkins, Jenkins, hanuman, and Not syncing. The URL in the address bar is https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standalone. The main content area displays the Jenkins pipeline status:

```
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git add .
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git status
On branch master
No commits yet
```

A blue curly brace is drawn around the "No commits yet" message and the "Changes to be committed" section.

Untracked files:

```
(use "git add <file>..." to include in what will be committed)
CMAR/
```

nothing added to commit but untracked files present (use "git add" to track)

```
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git add .
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git status
On branch master
No commits yet
```

Changes to be committed:

```
(use "git rm --cached <file>..." to unstage)
 new file: CMAR/.travis.yml
 new file: CMAR/README.md
 new file: CMAR/defaults/main.yml
 new file: CMAR/handlers/main.yml
 new file: CMAR/meta/main.yml
```

i-004e6cf6211e4999d (a1)

Public IPs: 13.127.21.10 Private IPs: 172.31.39.90

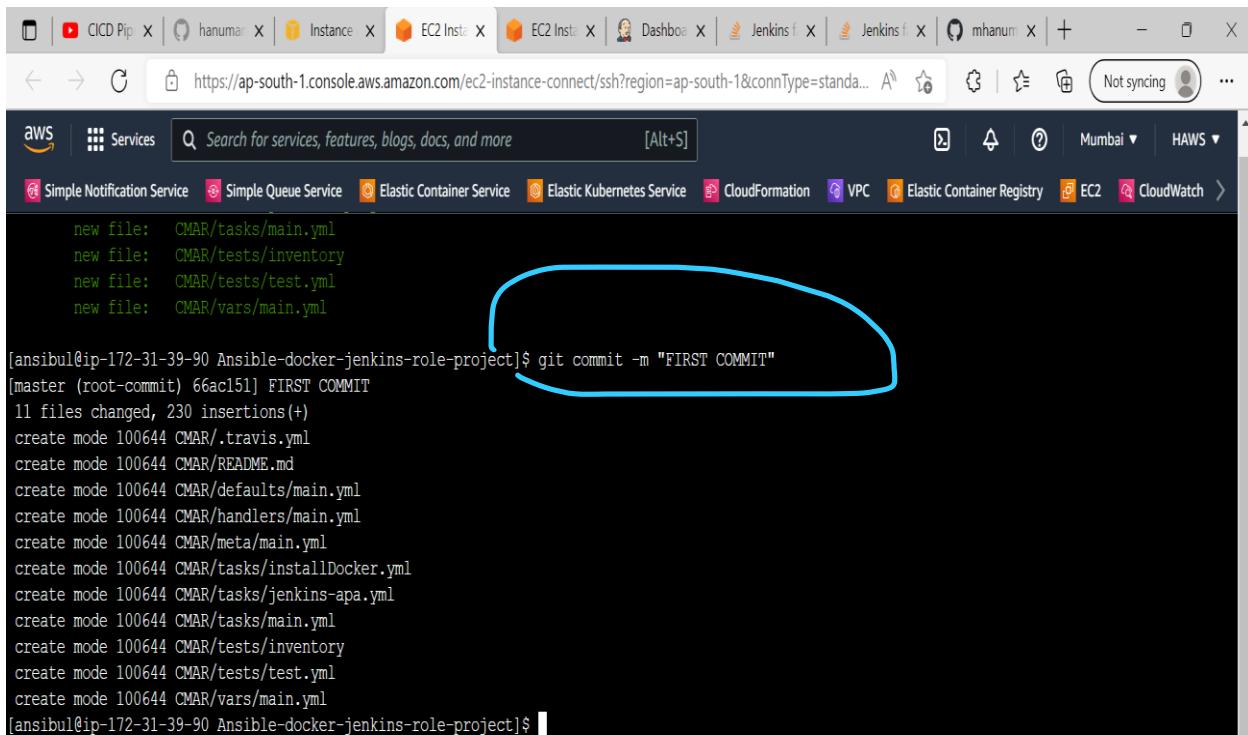
Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

27°C 10:52 AM 8/4/2022

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

**git commit**

**git commit -m “[ commit message]”**



A screenshot of a terminal window titled "git commit -m \"[ commit message]\"". The terminal shows the command being typed and its execution. A blue oval highlights the command line.

```
new file: CMAR/tasks/main.yml
new file: CMAR/tests/inventory
new file: CMAR/tests/test.yml
new file: CMAR/vars/main.yml

[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git commit -m "FIRST COMMIT"
[master (root-commit) 66ac151] FIRST COMMIT
11 files changed, 230 insertions(+)
create mode 100644 CMAR/.travis.yml
create mode 100644 CMAR/README.md
create mode 100644 CMAR/defaults/main.yml
create mode 100644 CMAR/handlers/main.yml
create mode 100644 CMAR/meta/main.yml
create mode 100644 CMAR/tasks/installDocker.yml
create mode 100644 CMAR/tasks/jenkins-apa.yml
create mode 100644 CMAR/tasks/main.yml
create mode 100644 CMAR/tests/inventory
create mode 100644 CMAR/tests/test.yml
create mode 100644 CMAR/vars/main.yml
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$
```

i-004e6cf6211e4999d (a1)

PublicIPs: 13.127.21.10 PrivateIPs: 172.31.39.90



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

27°C 10:53 AM 8/4/2022

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

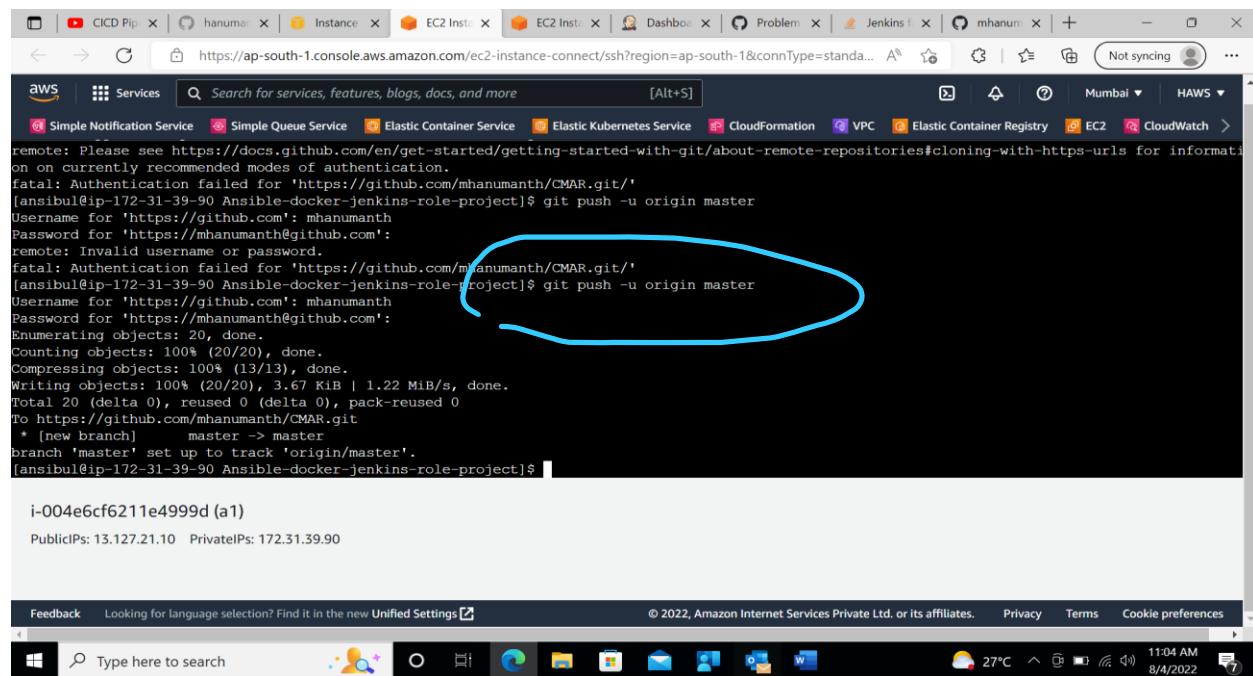
**git config**

**git config --global user.name “[name]”**

**git config --global user.email “[email address]”**

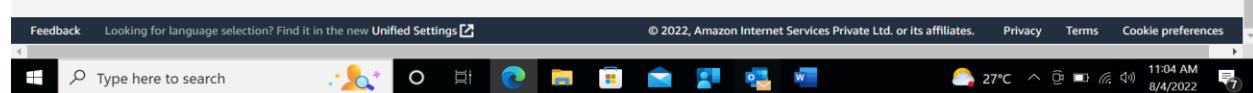
```
git remote add origin https://github.com/mhanumanth/git-repo-name.git
```

```
git push -u origin main
```

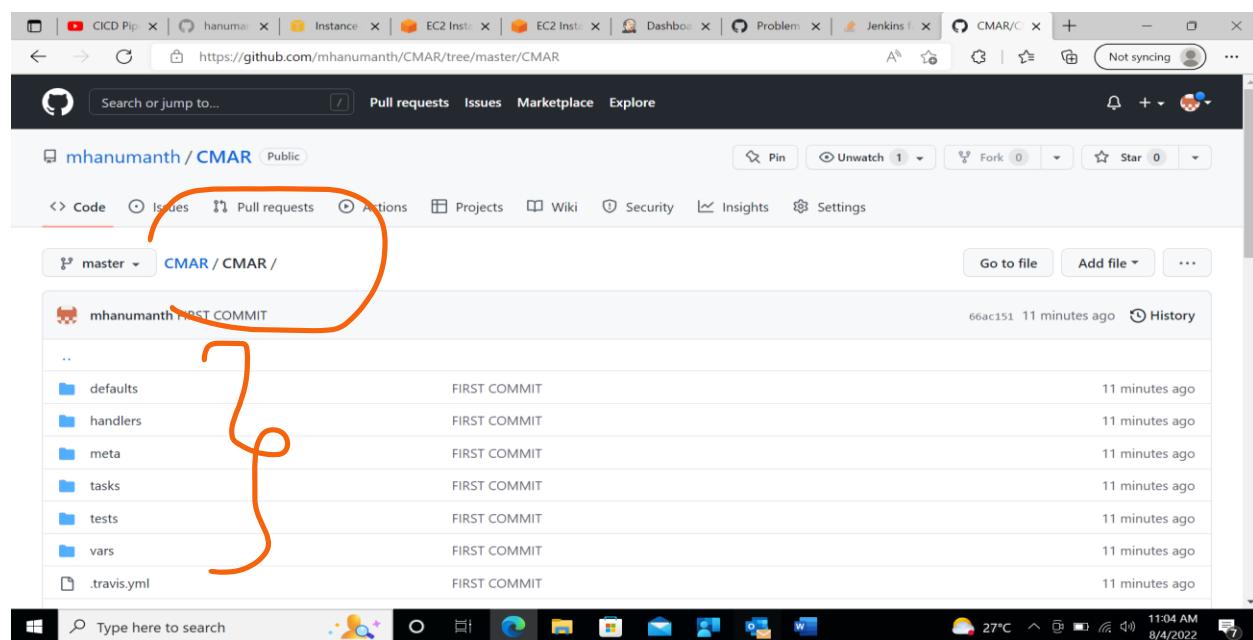


```
aws | Services | Search for services, features, blogs, docs, and more [Alt+S]
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/mhanumanth/CMAR.git'
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git push -u origin master
Username for 'https://github.com': mhanumanth
Password for 'https://mhanumanth@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/mhanumanth/CMAR.git'
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$ git push -u origin master
Username for 'https://github.com': mhanumanth
Password for 'https://mhanumanth@github.com':
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (20/20), 3.67 KIB | 1.22 MiB/s, done.
Total 20 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mhanumanth/CMAR.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[ansibul@ip-172-31-39-90 Ansible-docker-jenkins-role-project]$
```

i-004e6cf6211e4999d (a1)  
PublicIPs: 13.127.21.10 PrivateIPs: 172.31.39.90



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



mhanumanth / CMAR (Public)

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master / CMAR /

mhanumanth FIRST COMMIT 66ac151 11 minutes ago History

..

File	Commit	Time
defaults	FIRST COMMIT	11 minutes ago
handlers	FIRST COMMIT	11 minutes ago
meta	FIRST COMMIT	11 minutes ago
tasks	FIRST COMMIT	11 minutes ago
tests	FIRST COMMIT	11 minutes ago
vars	FIRST COMMIT	11 minutes ago
.travis.yml	FIRST COMMIT	11 minutes ago



## 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

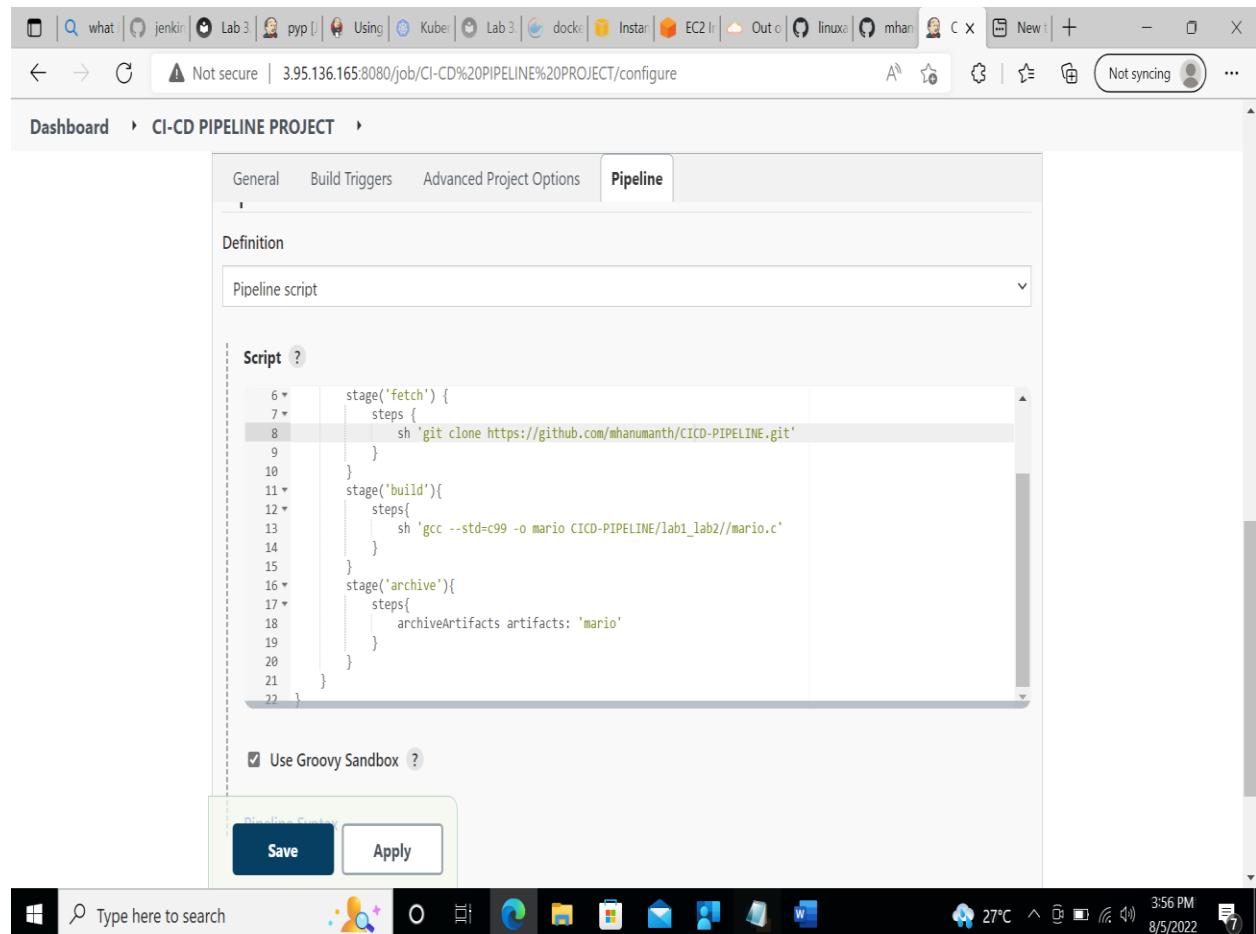
1.A Preparation stage where we clone code from git repo

<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 -



```
6 v
7 v
8 stage('fetch') {
9     steps {
10         sh 'git clone https://github.com/mhanumanth/CICD-PIPELINE.git'
11     }
12 }
13 stage('build'){
14     steps{
15         sh 'gcc --std=c99 -o mario CICD-PIPELINE/lab1_lab2//mario.c'
16     }
17 }
18 stage('archive'){
19     steps{
20         archiveArtifacts artifacts: 'mario'
21     }
22 }
```

Use Groovy Sandbox ?

**Pipeline Configuration** **Save** **Apply**

## Buid stage view

The screenshot shows the Jenkins Stage View page for the 'CI-CD PIPELINE PROJECT'. On the left sidebar, there are several options: Build Now, Configure, Delete Pipeline, Full Stage View, Open Blue Ocean, Rename, and Pipeline Syntax. The 'Stage View' section displays a timeline with three stages: 'fetch', 'build', and 'archive'. Below the timeline, a table provides detailed information for each stage: 'fetch' took 4s, 'build' took 4s, and 'archive' took 437ms. A note indicates 'No Changes' for the build step. The 'Permalinks' section at the bottom provides links for the build history and atom feeds.

## Console Output-

The screenshot shows the Jenkins Console Output page for build #1 of the 'CI-CD PIPELINE PROJECT'. The left sidebar lists various build actions: Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#1', Open Blue Ocean, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The main content area displays the console output log, which starts with 'Started by user unknown or anonymous' and continues with a detailed log of Docker pull and inspect commands for the Jenkins pipeline container.

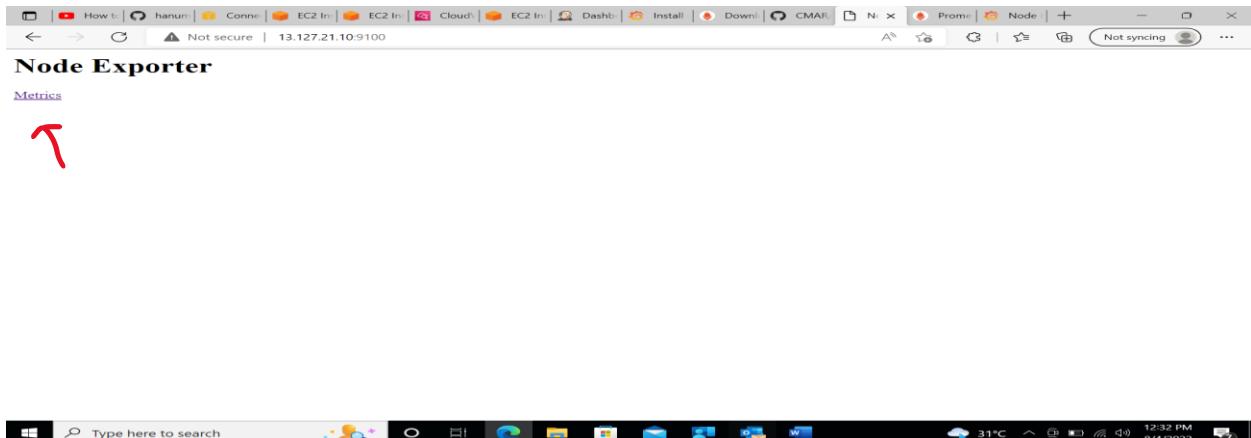
Git URL: <https://github.com/mhanumanth/CICD-PIPELINE.git>

## TASK6: BUILDING MONITORING TASK

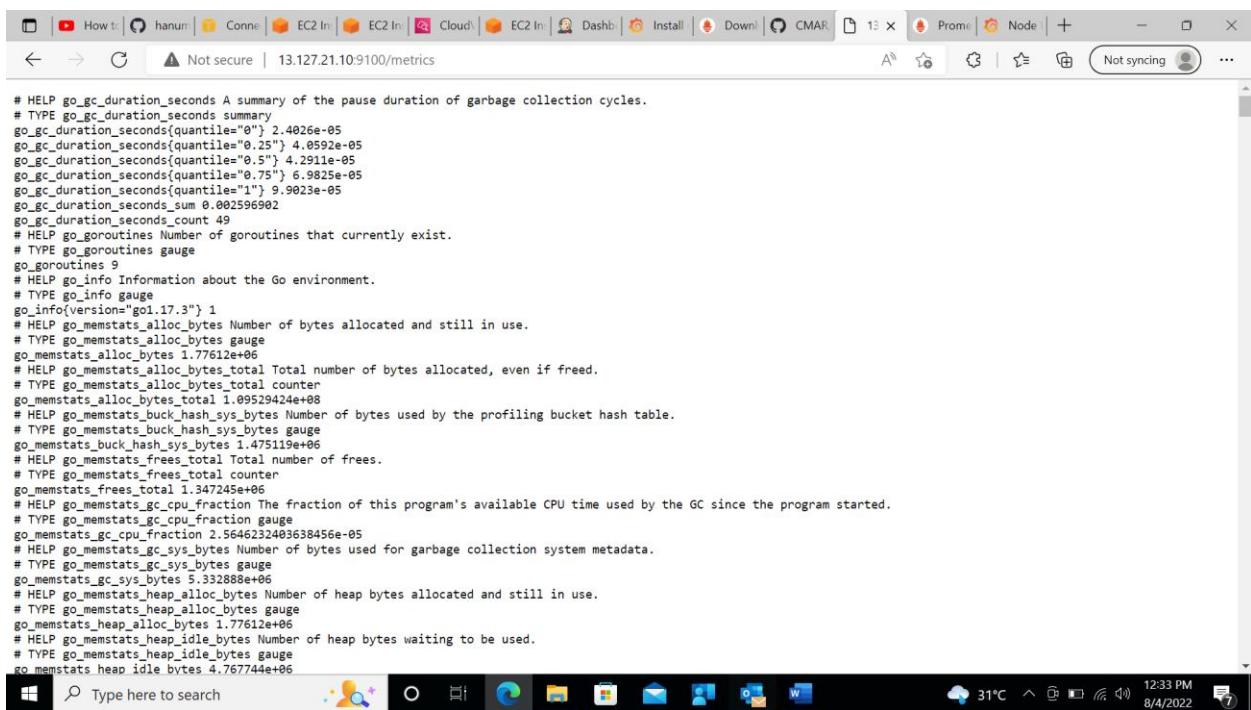
Monitoring tools like Prometheus and Grafana are both built for time-series data. **Prometheus excels in metric data collection, whereas Grafana champions metric visualizations.** Both tools are open source, free and have vibrant communities

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.

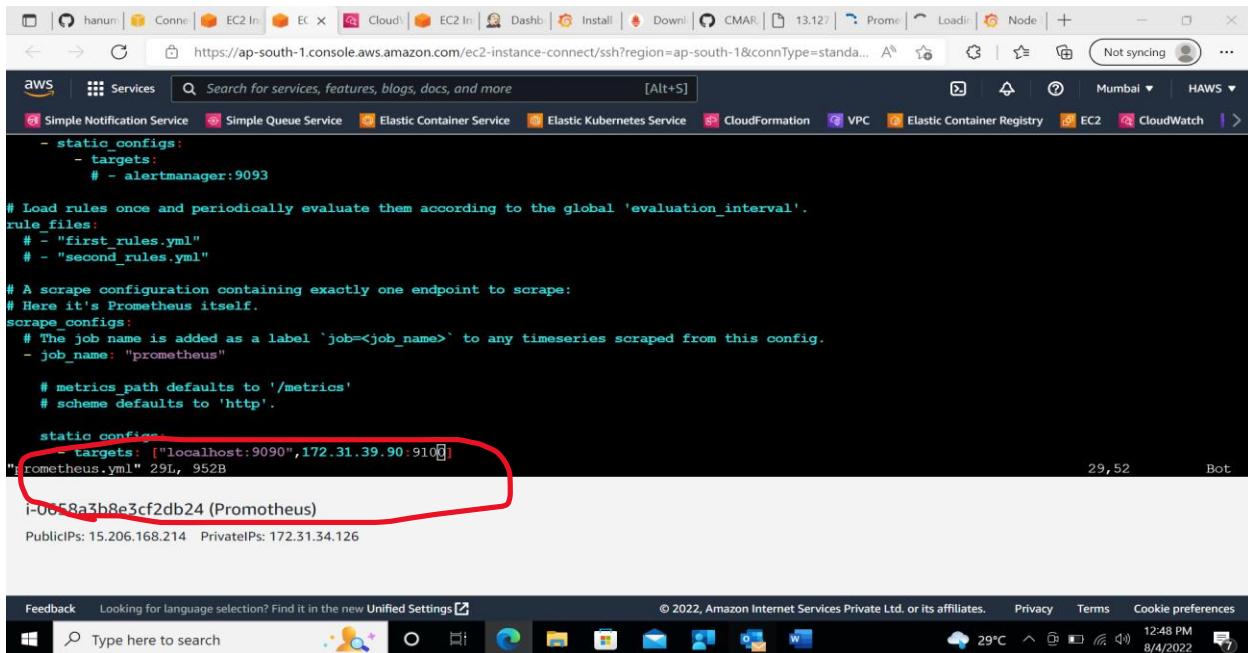
I had installed node\_port on the VM to be monitored, which will export all the metrics of this VM



After clicking on the metrics



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



```
static_configs:
  - targets:
    # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

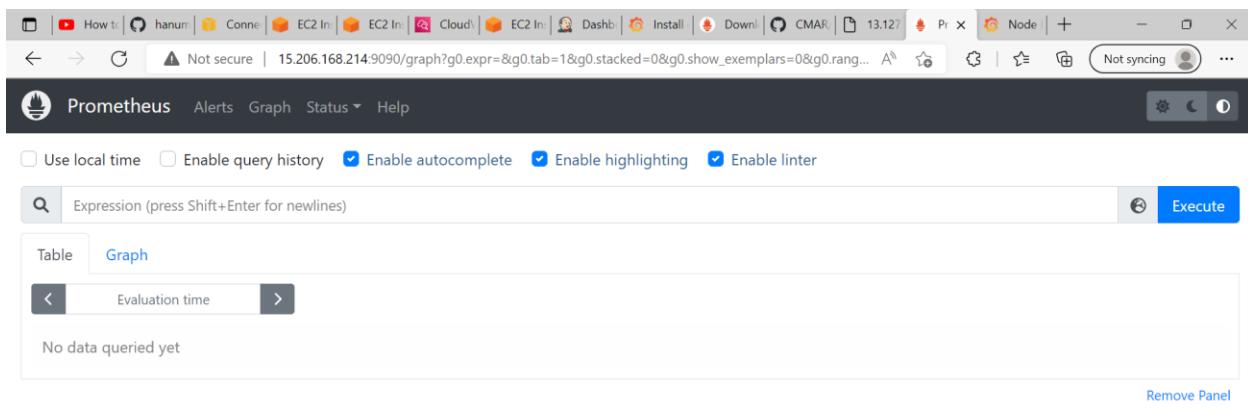
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090", 172.31.39.90:9100]
        file: "prometheus.yml"
        29L, 952B

i-0e58a3b8e3cf2db24 (Prometheus)
Public IPs: 15.206.168.214 Private IPs: 172.31.34.126
```

Prometheus runs on the port 9090



Not secure | 15.206.168.214:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.show\_exemplars=0&g0.range.end=now A Not syncing ...

Prometheus Alerts Graph Status Help

Use local time  Enable query history  Enable autocomplete  Enable highlighting  Enable linter

Expression (press Shift+Enter for newlines) Execute

Table Graph

Evaluation time < >

No data queried yet Remove Panel

Add Panel



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

The screenshot shows the Prometheus Targets page. The top navigation bar has tabs for Prometheus, Alerts, Graph, Status (which is highlighted), and Help. Below the tabs is a search bar with the placeholder "Filter by endpoint or labels". A yellow sticky note is placed over the Status tab. The main section is titled "Targets" and shows a table with two entries:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	instance="localhost:9090" job="prometheus"	6.504s ago	4.956ms	
<a href="http://172.31.39.90:9100/metrics">http://172.31.39.90:9100/metrics</a>	UP	instance="172.31.39.90:9100" job="prometheus"	5.733s ago	11.952ms	



In Graphana we can visualize the different metrics as shown below

