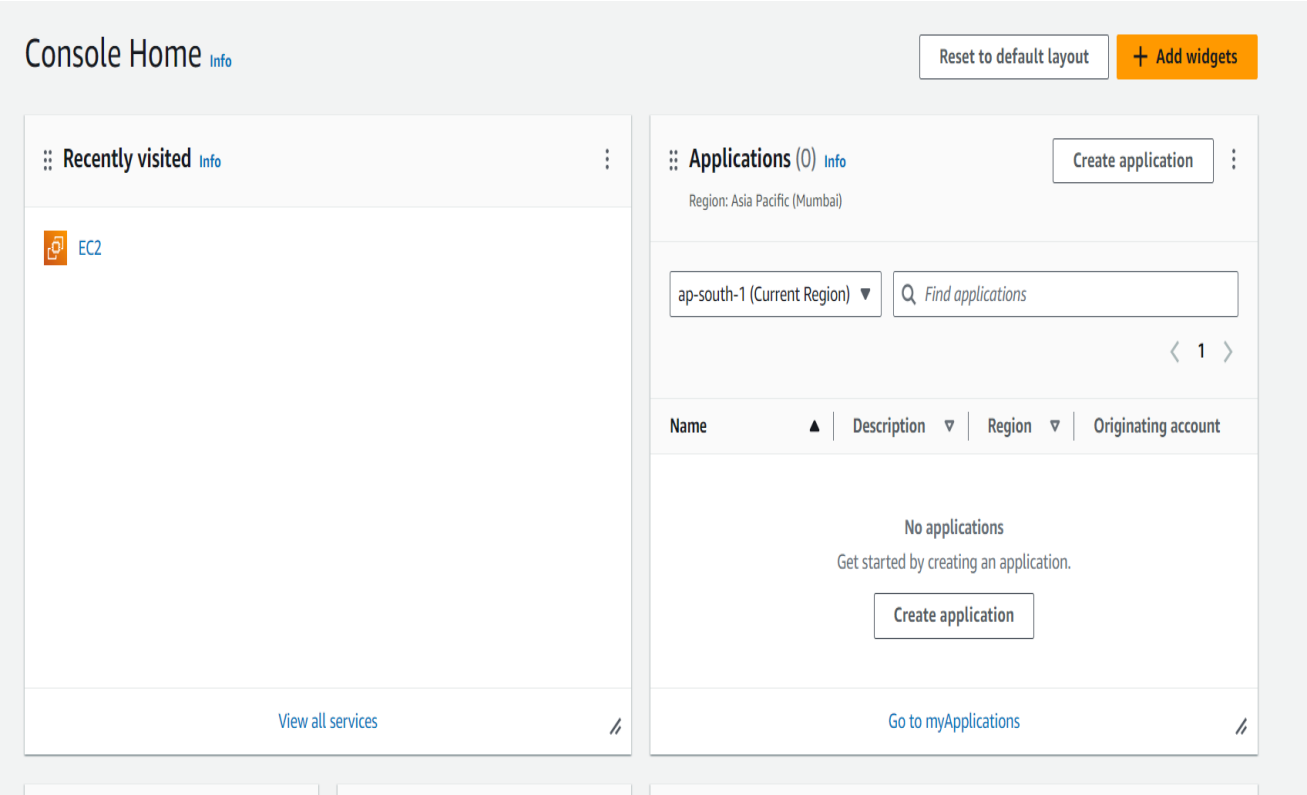
EXPERIMENT NO. 1

NAME: MOHIT PATIL CLASS:D15A ROLLNO.37

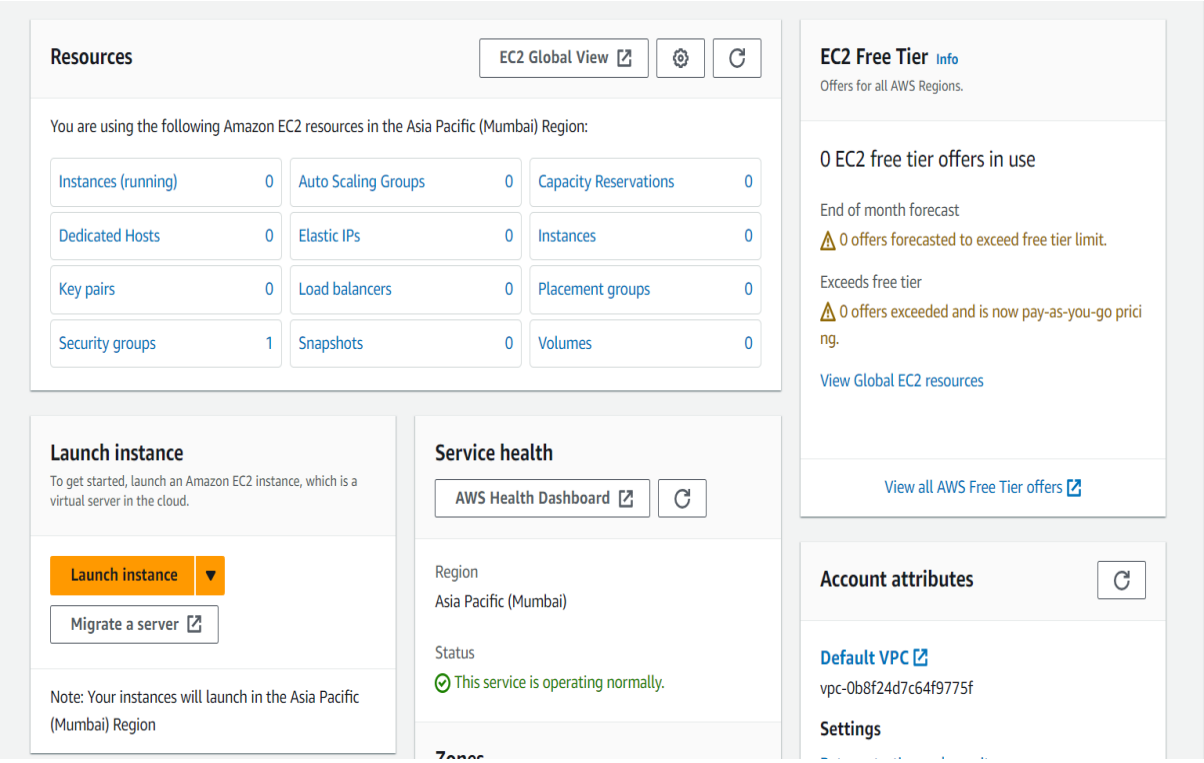
Aim : To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

EC2 Instance Creation and static site hosting

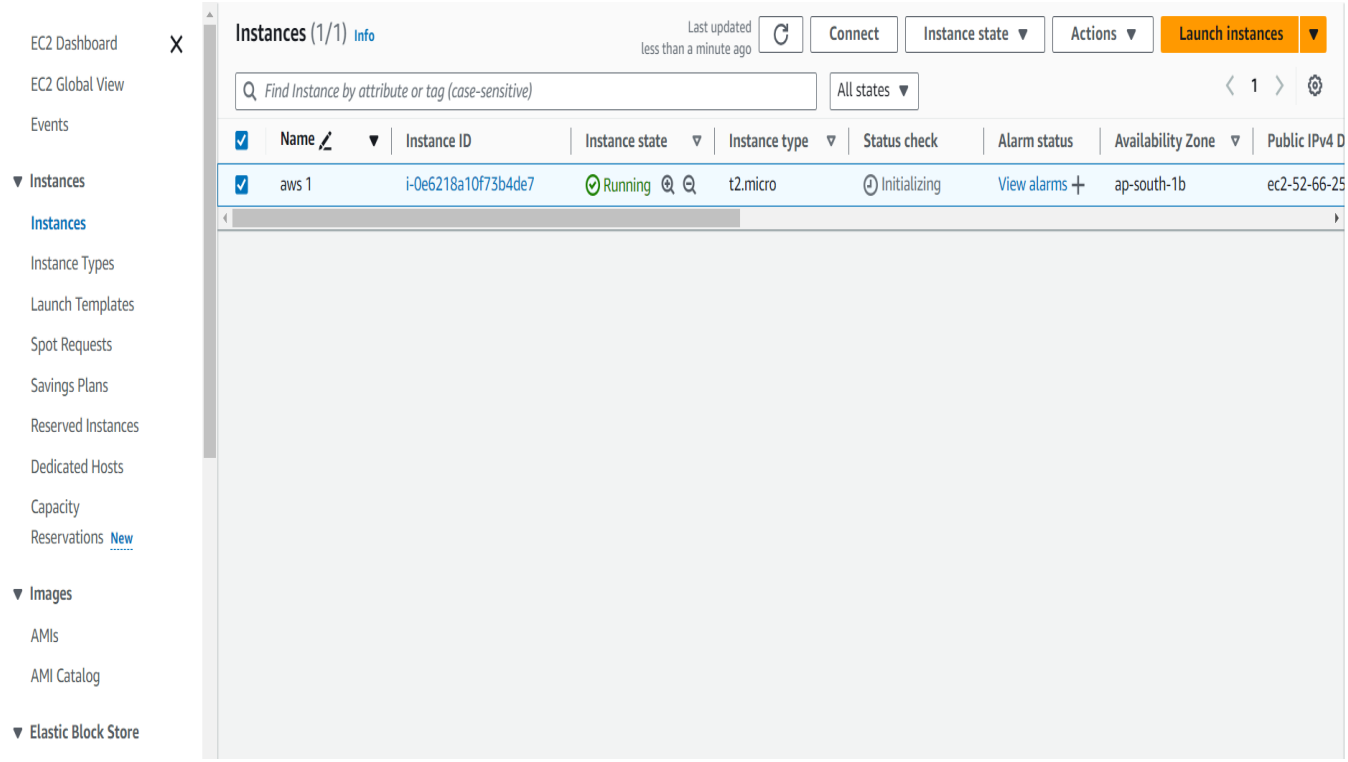
1. Login to your AWS account



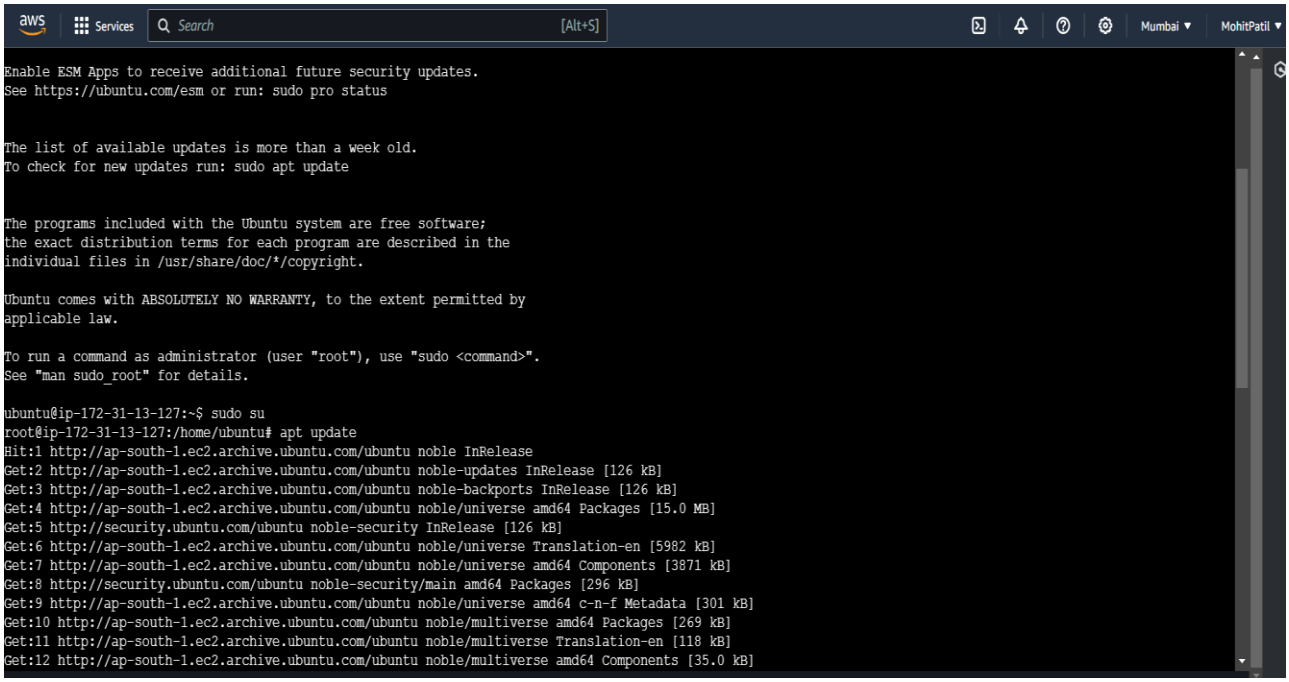
2. Click on EC2 and then create an instance by clicking on instances



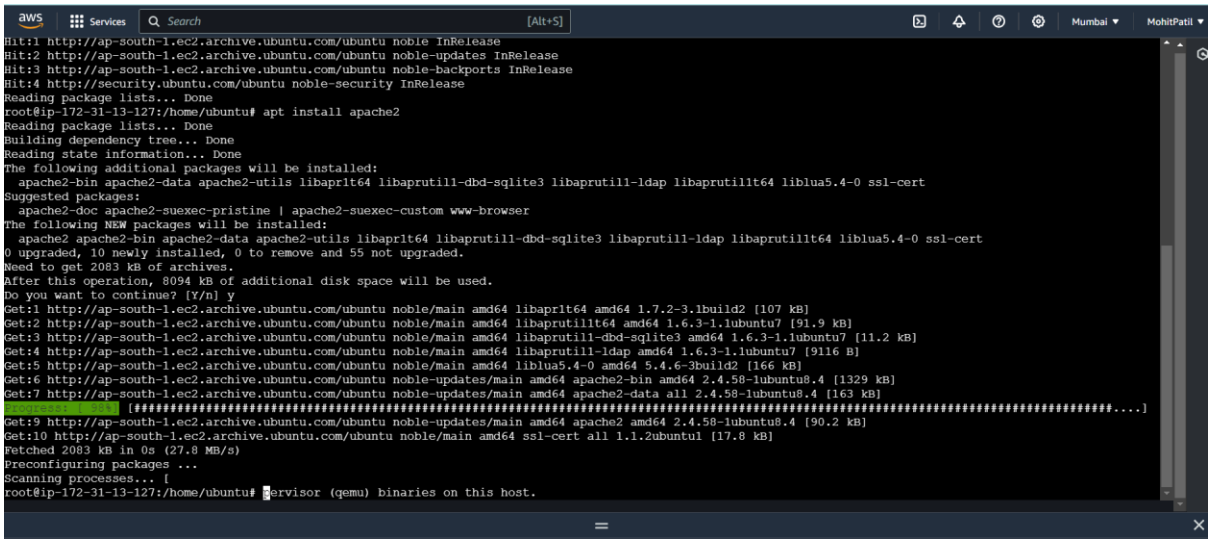
3. After an instance is created wait for it to come to Running state



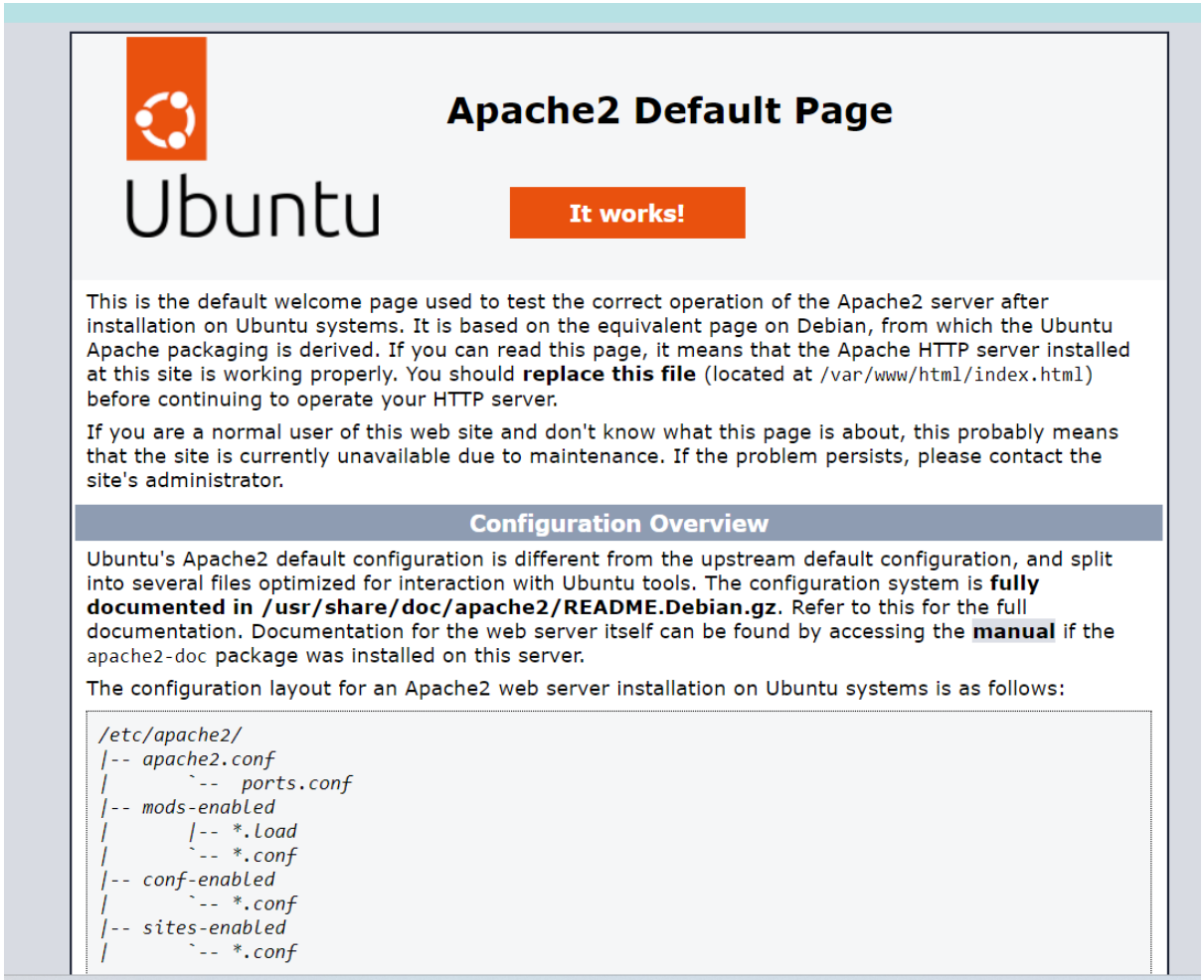
4. After doing that you will see this UI



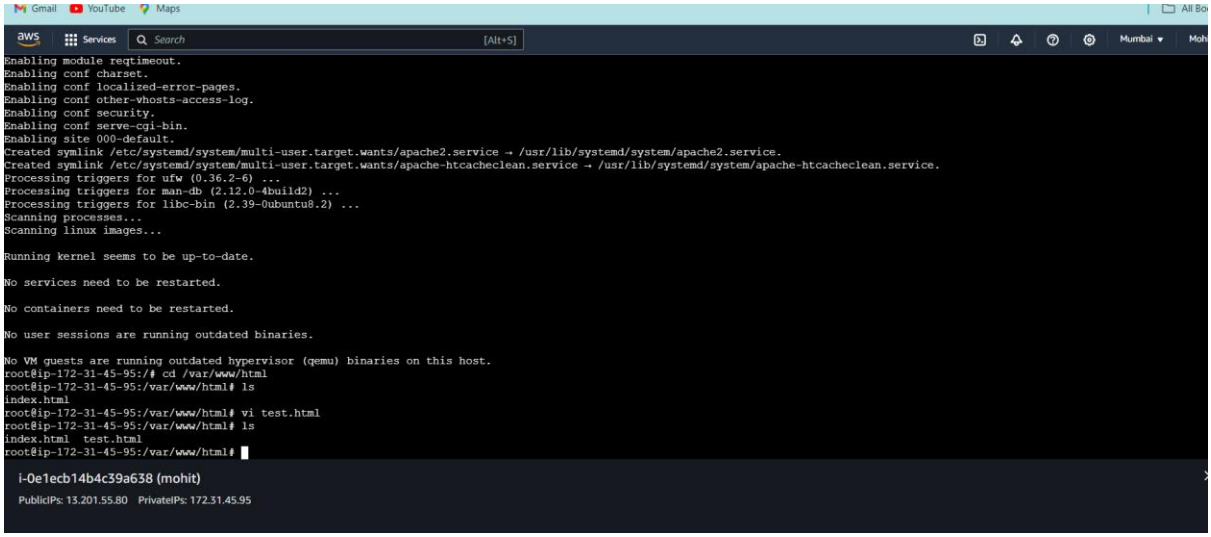
5. Follow these steps and then run these commands



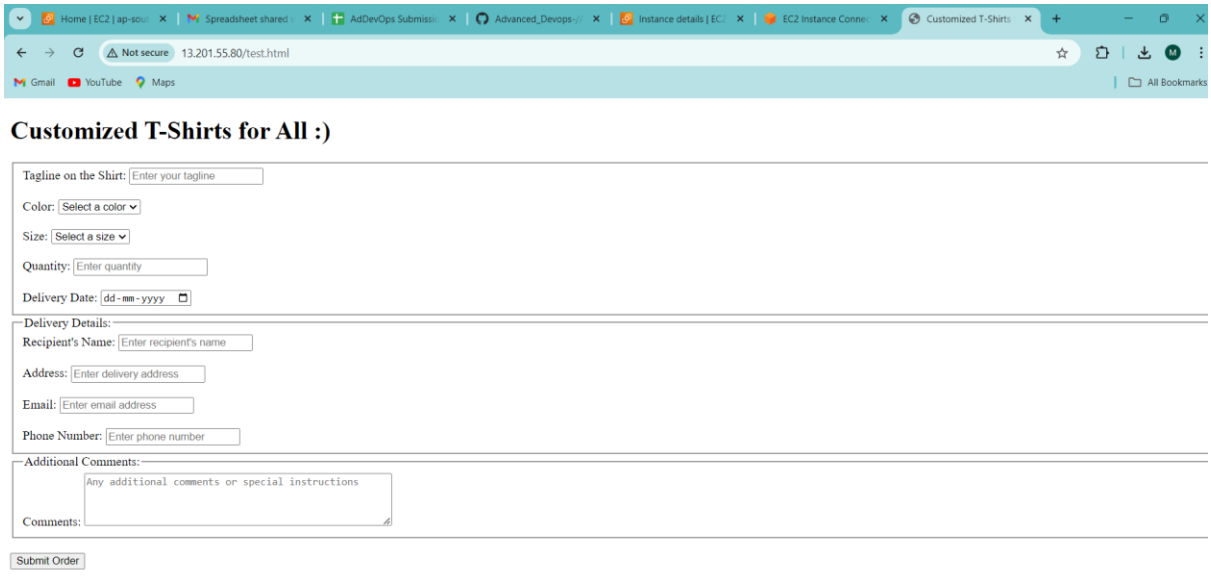
6. After that the ip-address which was given while running the instance, copy that and paste that on chrome, make sure that it is http and not https



7. Create a file using vi command and save it using : wq

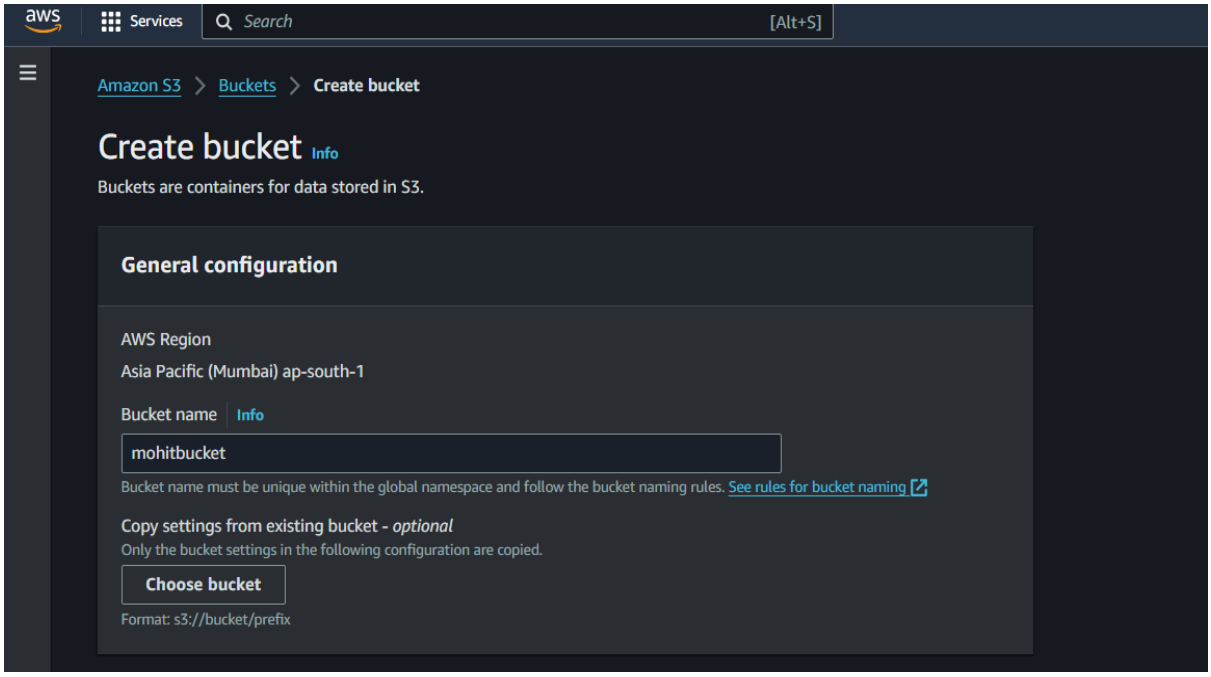


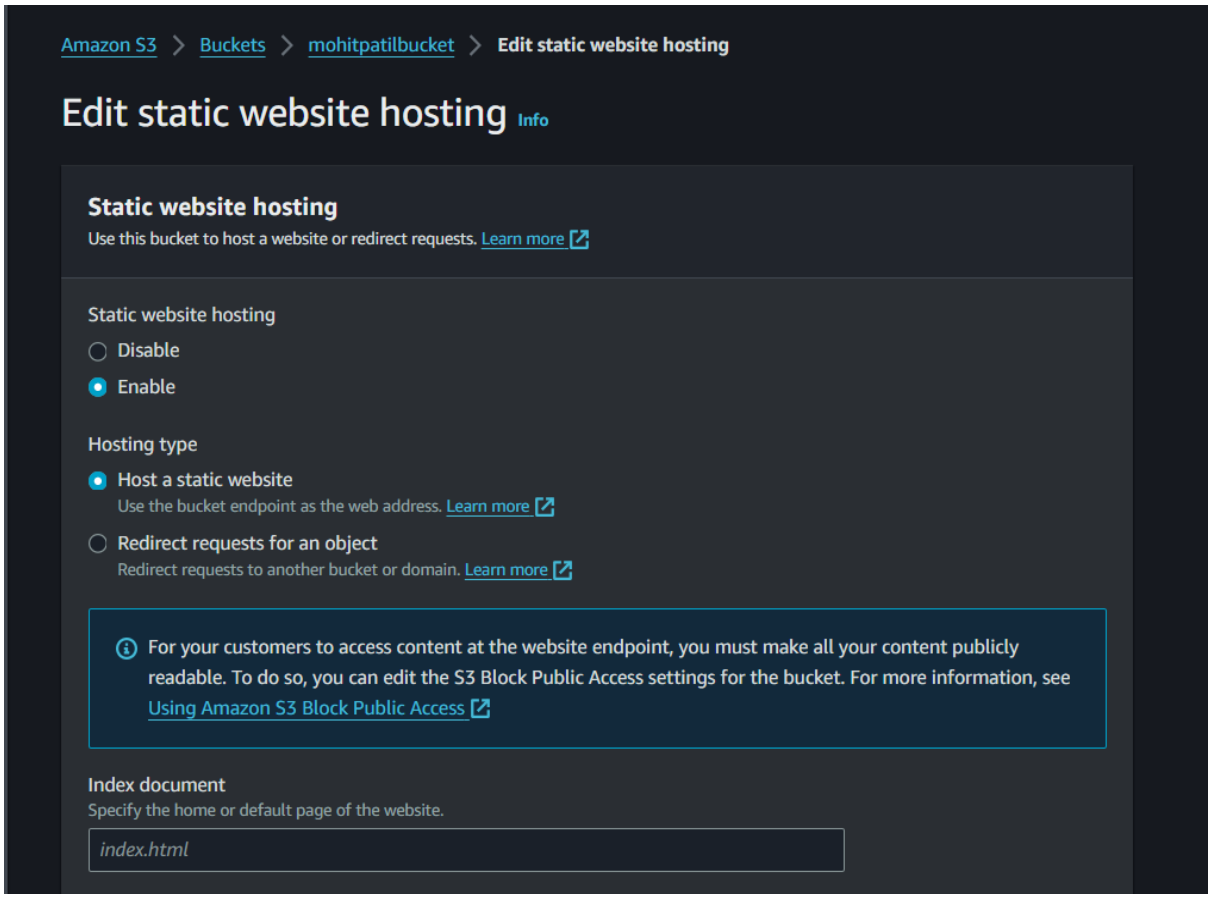
8. After saving that file go that page where ubuntu is listed and then on the link add “/your_file_name.html” and then whatever you saved on that file will be displayed



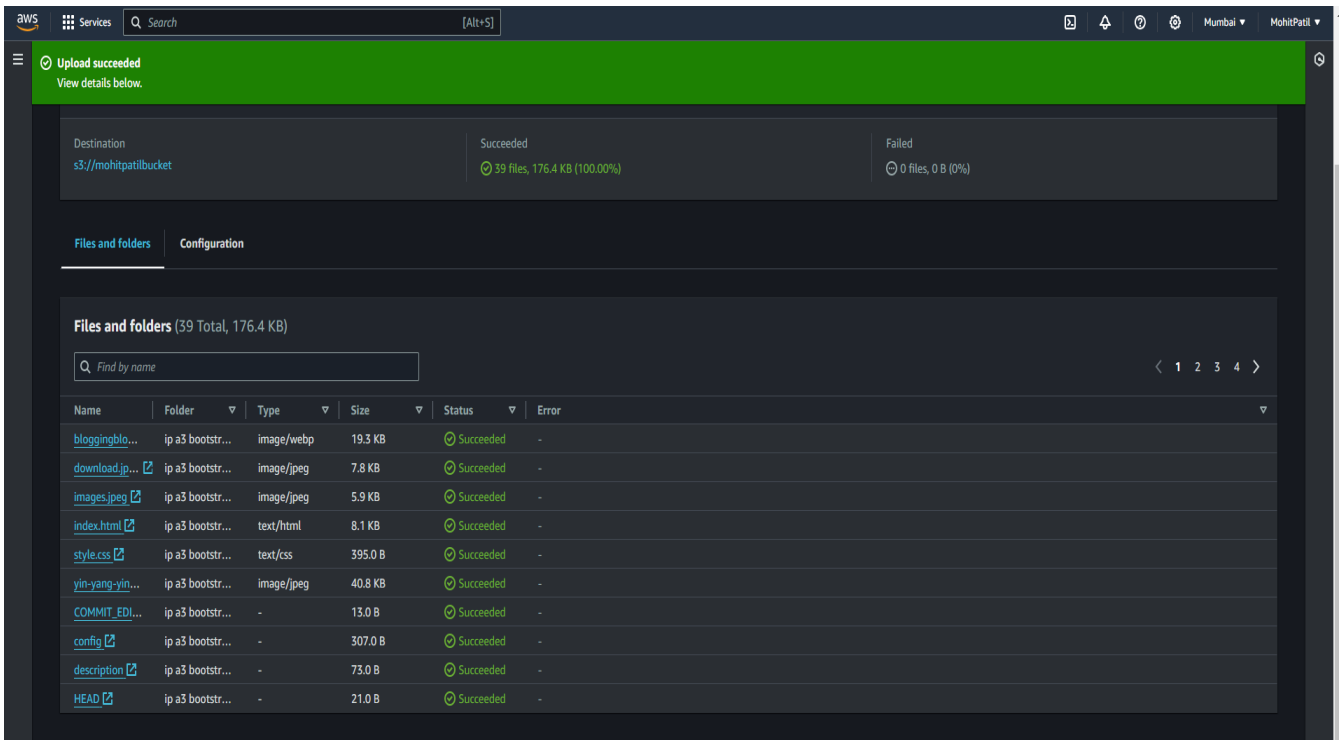
Static Hosting using S3 bucket

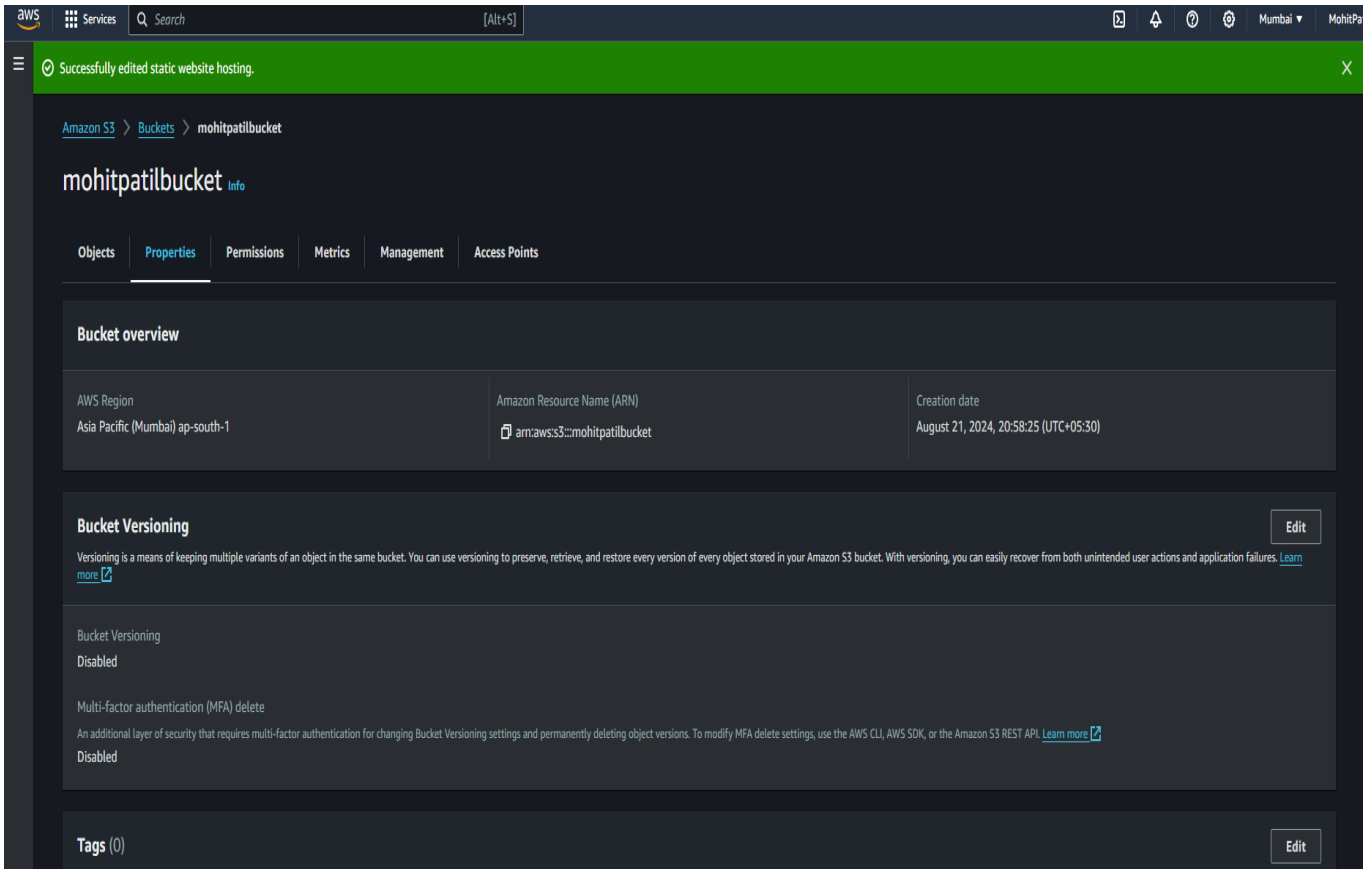
Step1: Create bucket

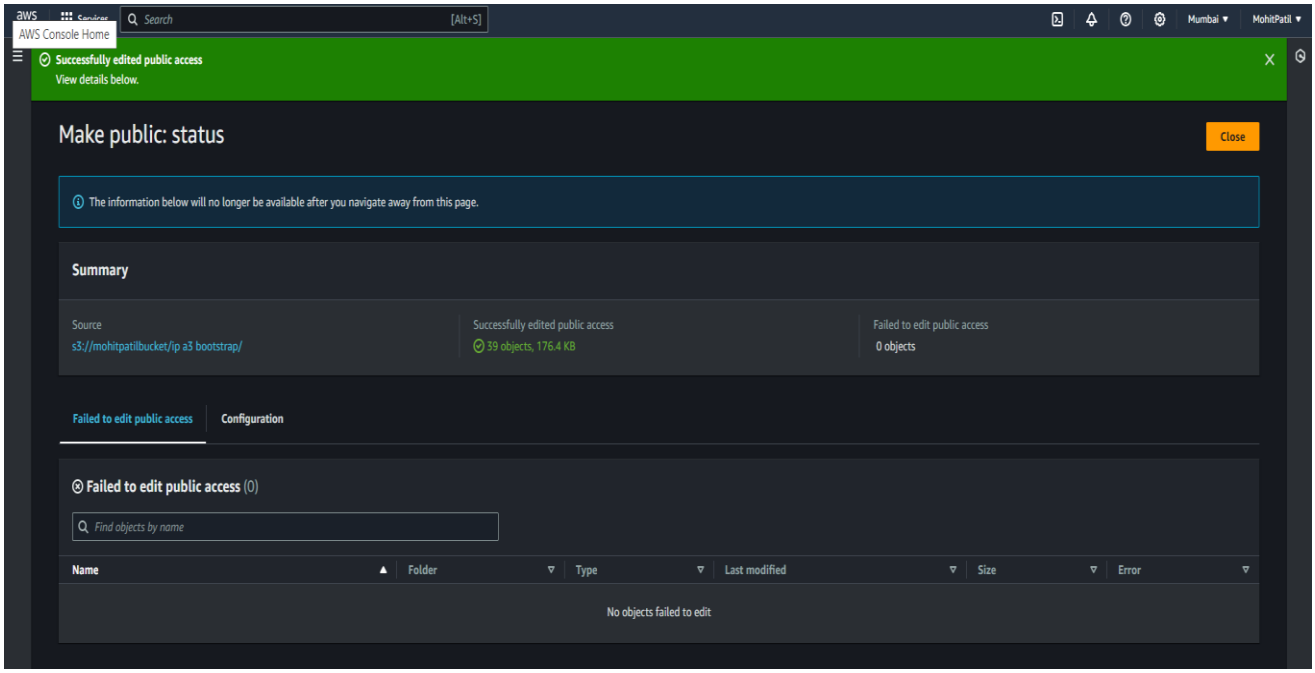




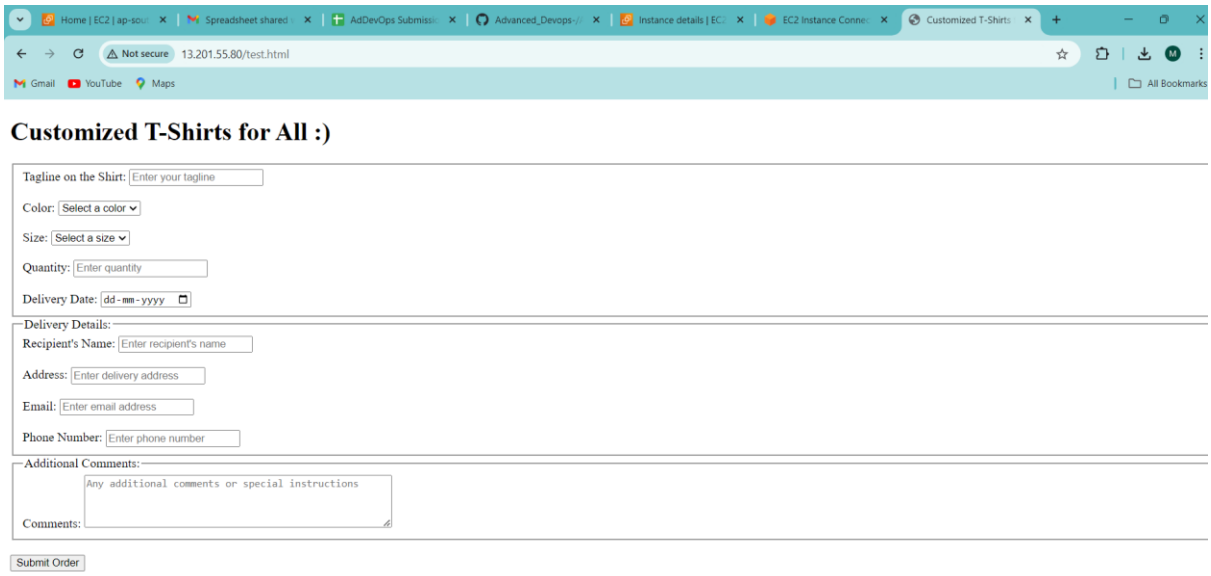
Step 2: Add resources





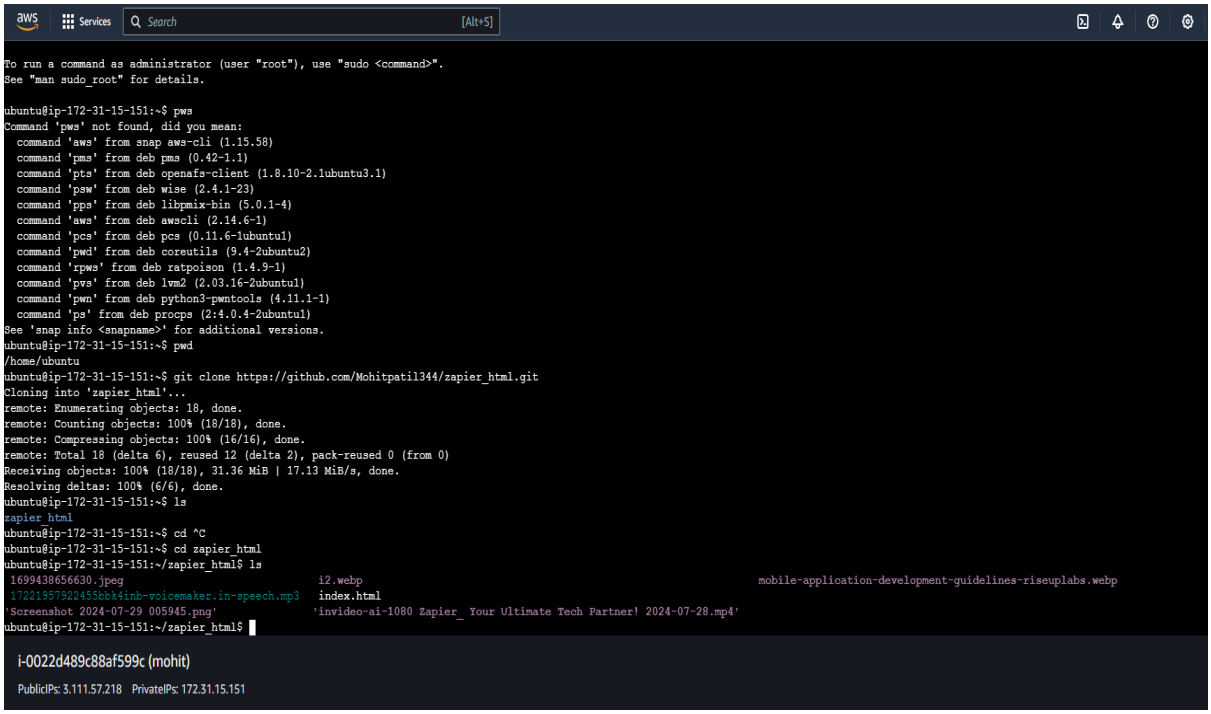


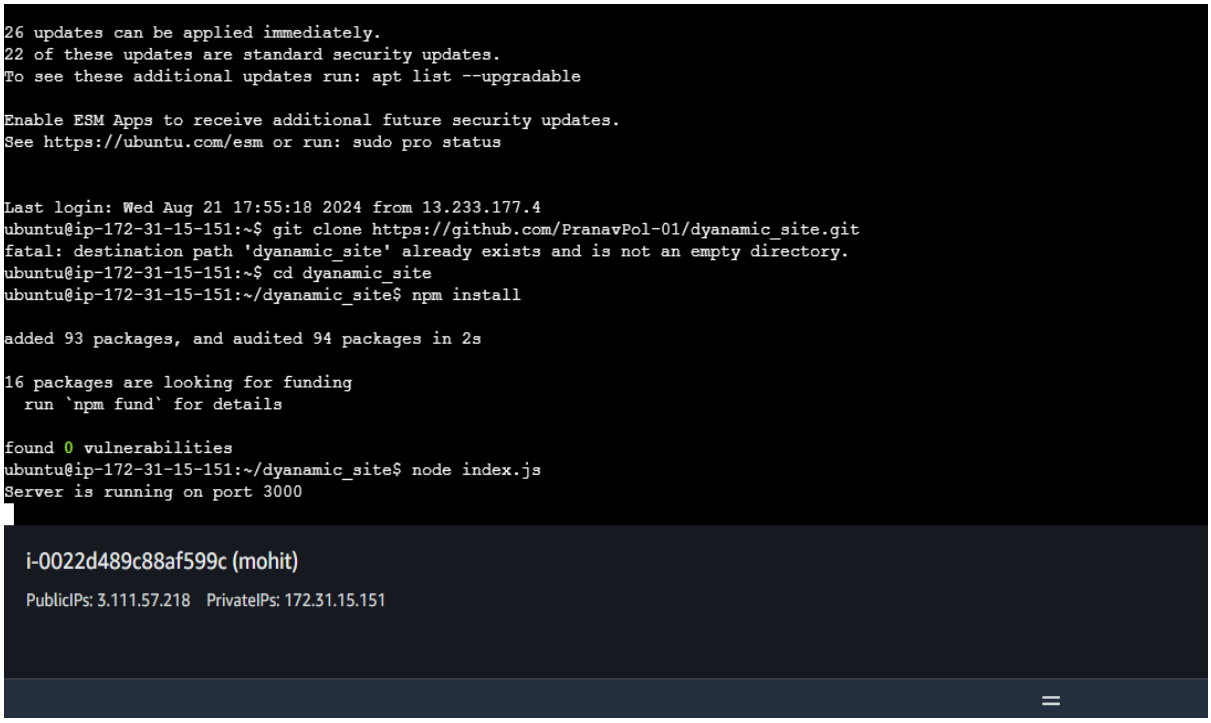
Step 3 : visit hosted website



EC2 Dynamic Site Hosting

Step 1 : Open Console and clone the github repository



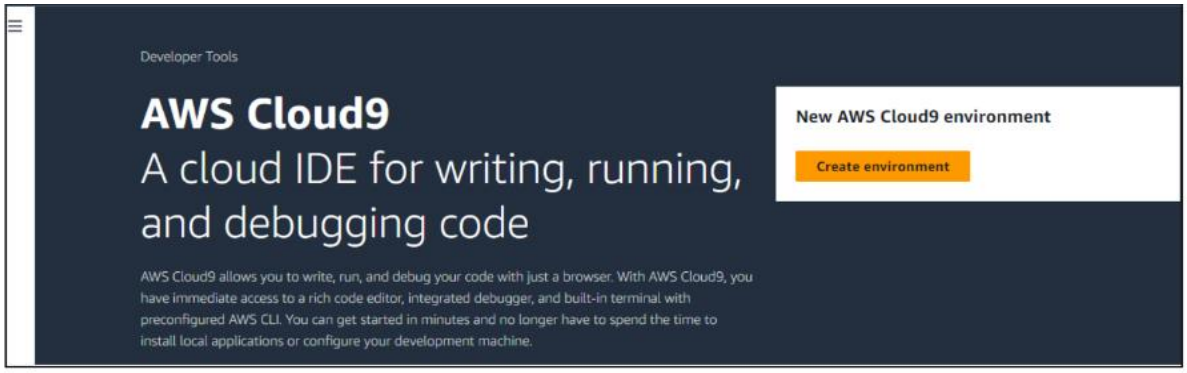


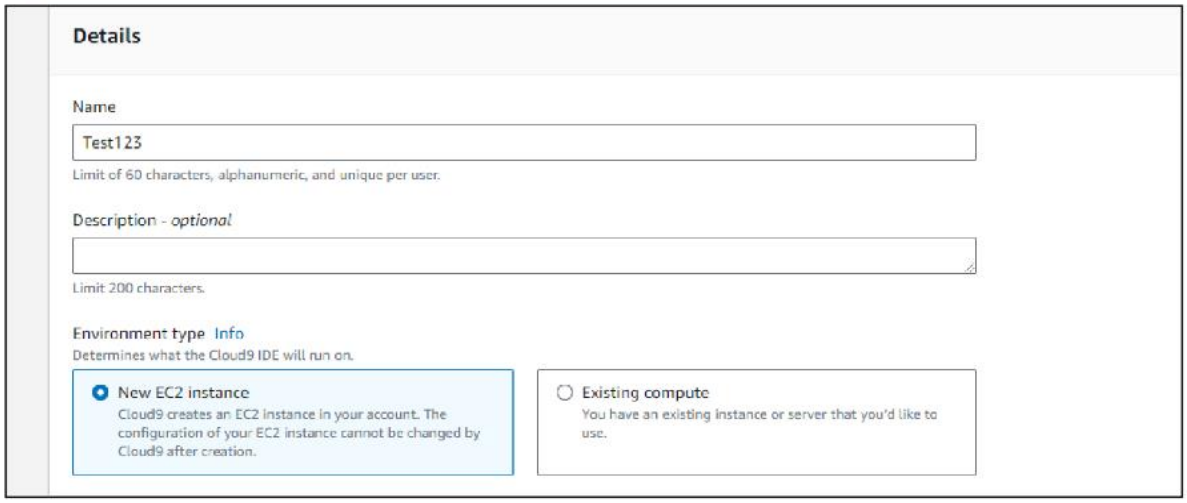
Step 2 : Install necessary Packages and run website on port 3000

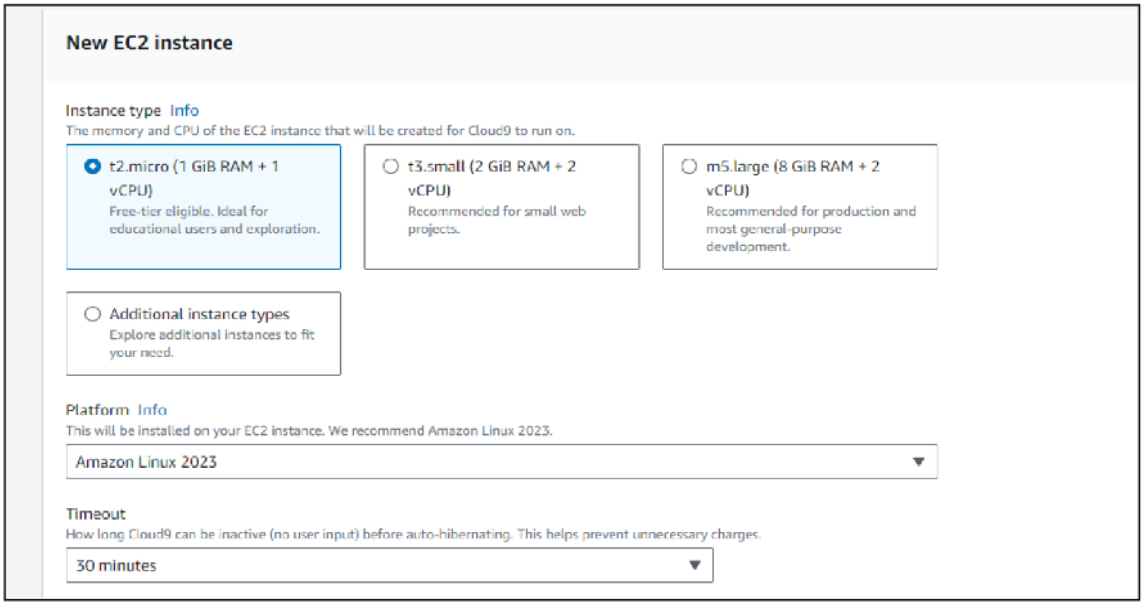
Hey this is Dynamic Website.

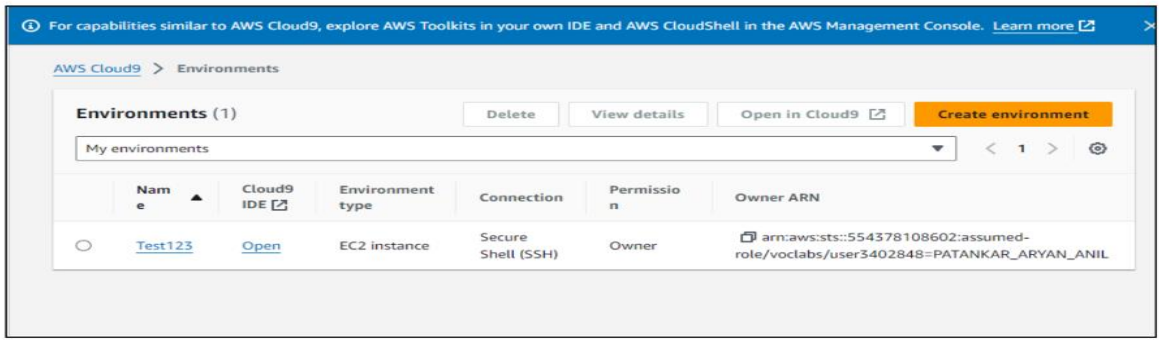
Hey this is about page.

Cloud 9 IDE Site Hosting

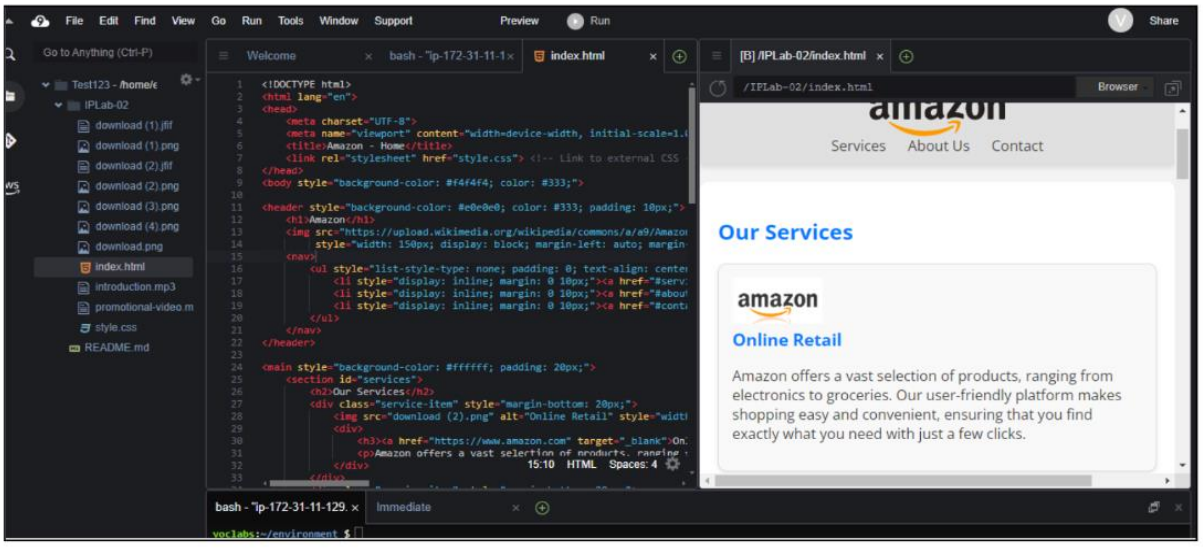







For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. Learn more
AWS Cloud9 > Environments
Environments (1)
Delete
View details
Open in Cloud9
Create environment
My environments
Name
Cloud9 IDE
Environment type
Connection
Permission
Owner ARN
Test123
Open
EC2 instance
Secure Shell (SSH)
Owner
arn:aws:sts::554378108602:assumed-role/voclabs/user3402848=PATANKAR_ARYAN_ANIL


File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Welcome
Developer Tools
AWS Cloud9
Welcome to your development environment
AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.
Getting started
Create File
Upload Files...
Clone from GitHub
Toolkit for AWS Cloud9
The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications
bash - "ip-172-31-11-129.
Immediate
voclabs:~/environment $


File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Welcome
bash - "ip-172-31-11-1
index.html
[B] /IPLab-02/index.html
/IPLab-02/index.html
Browser
Services
About Us
Contact
Our Services
amazon
Online Retail
Amazon offers a vast selection of products, ranging from electronics to groceries. Our user-friendly platform makes shopping easy and convenient, ensuring that you find exactly what you need with just a few clicks.
15:10 HTML Spaces: 4
bash - "ip-172-31-11-129.
Immediate
voclabs:~/environment $

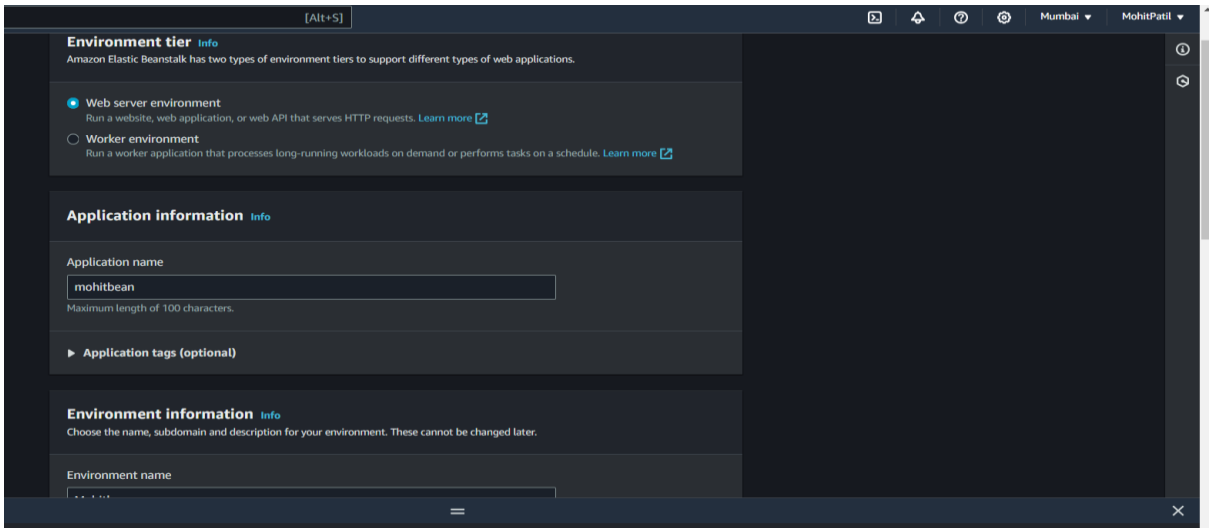
EXPERIMENT 2

NAME:MOHIT PATIL CLASS:D15A ROLL NO:37

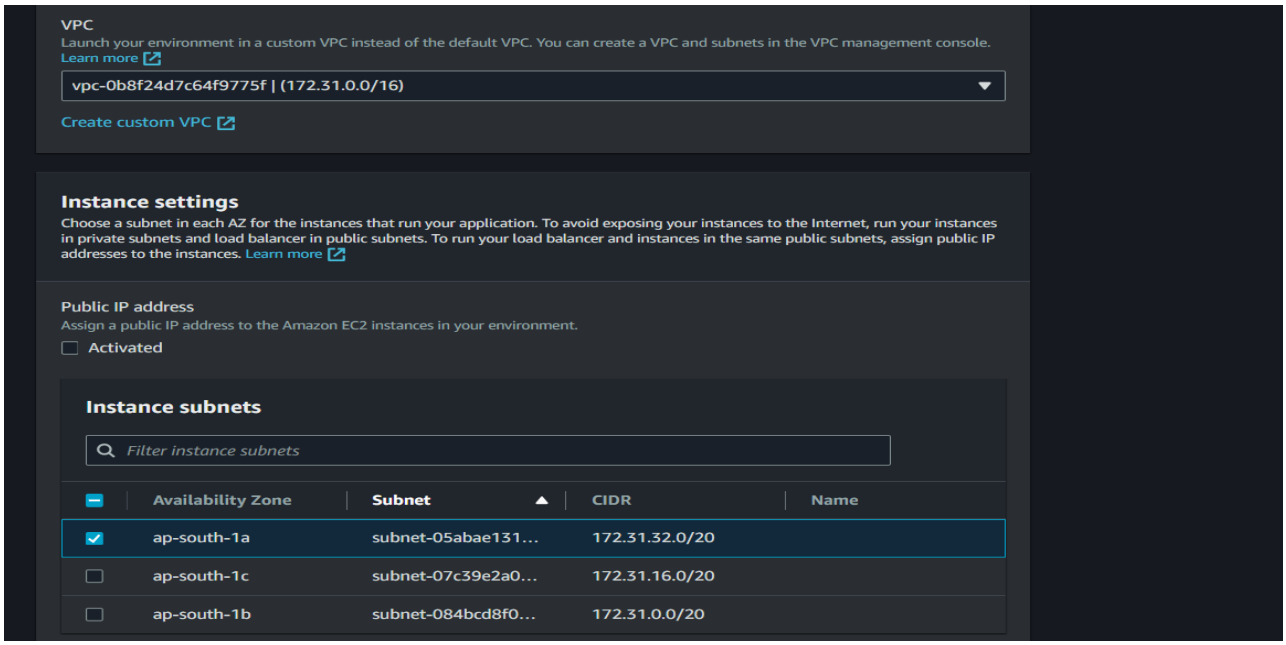
Aim :To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Elastic Beanstalk

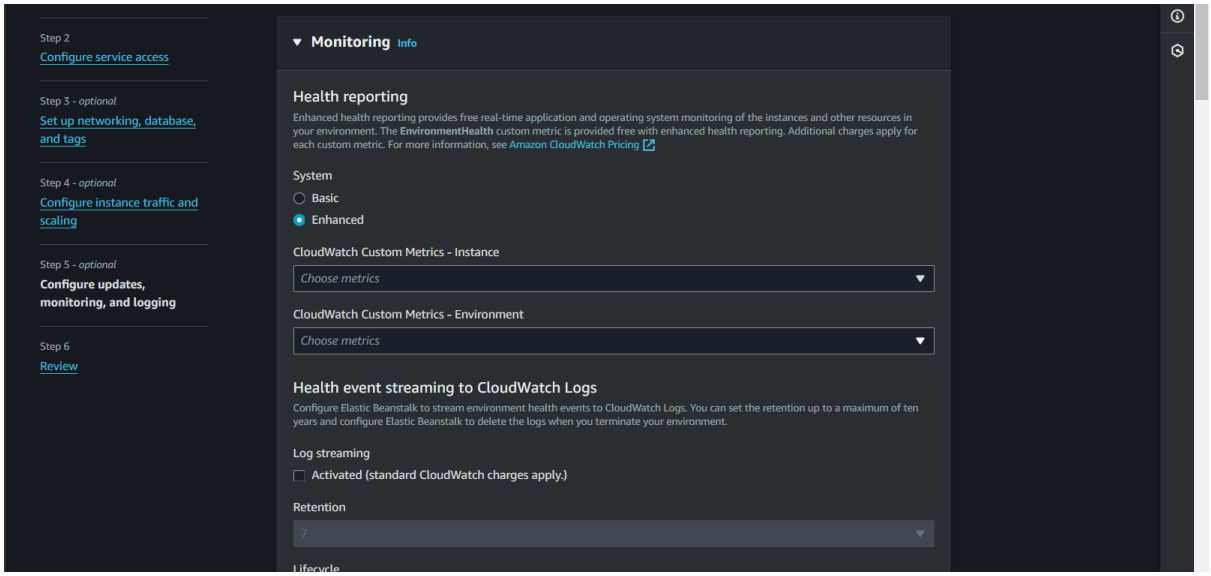
Step 1: create environment

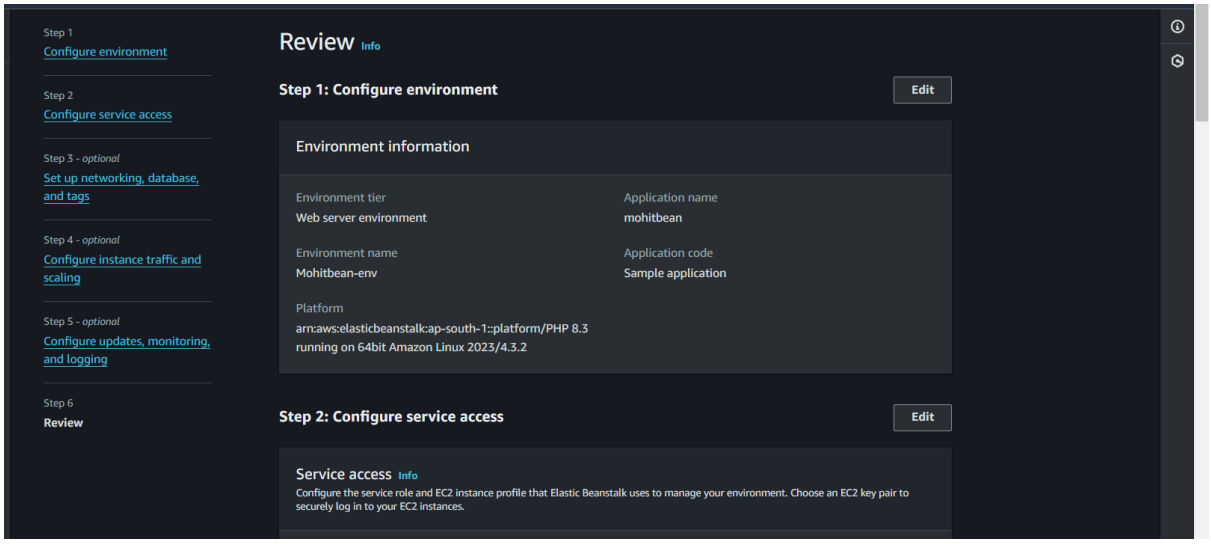


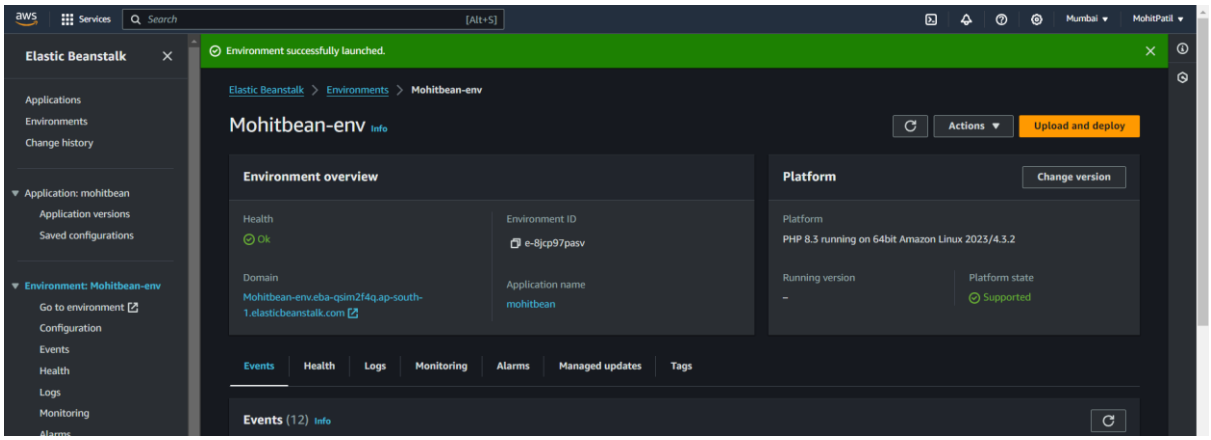
Step 2 : add your Ec2 key pair and instance profile



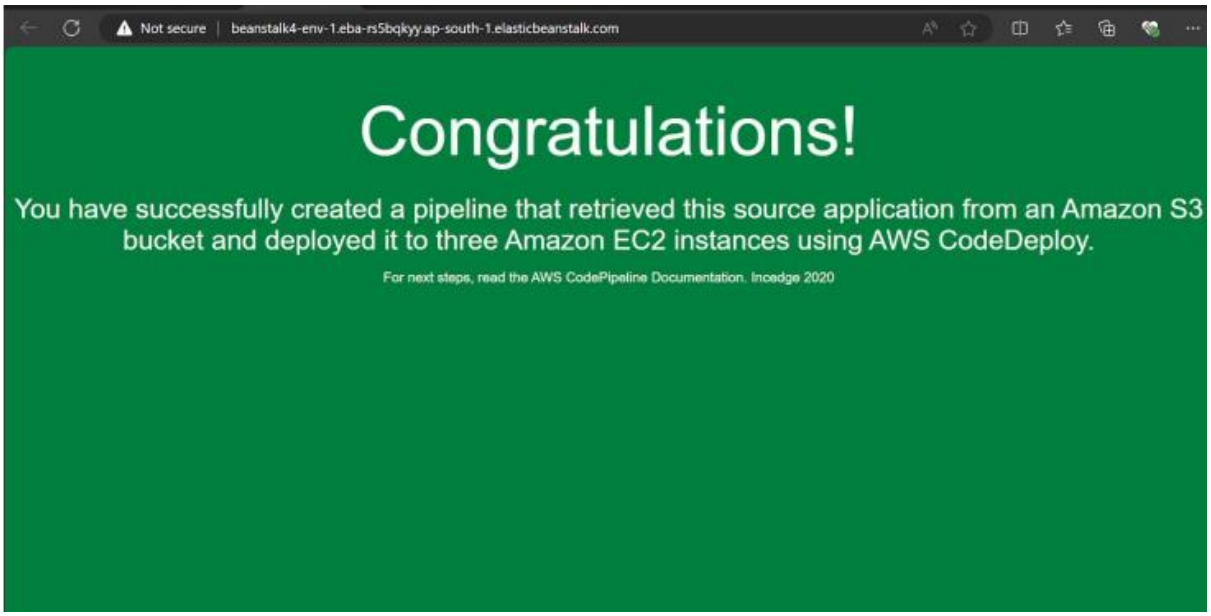
Step 3 : add security config and review all settings



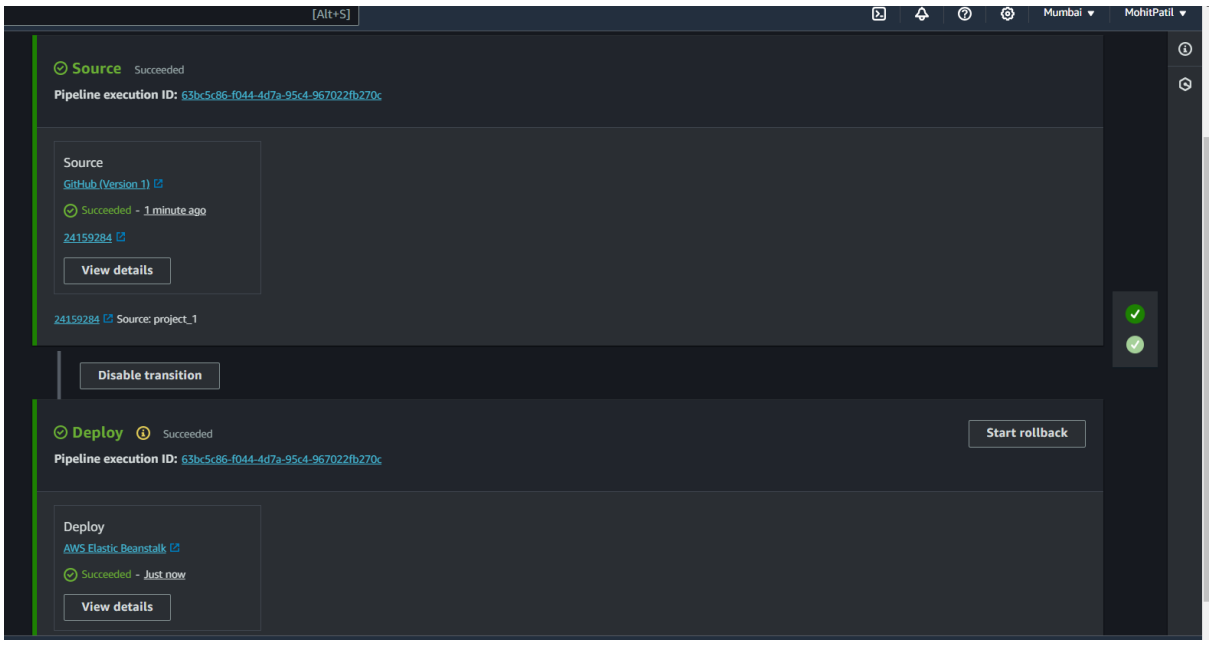




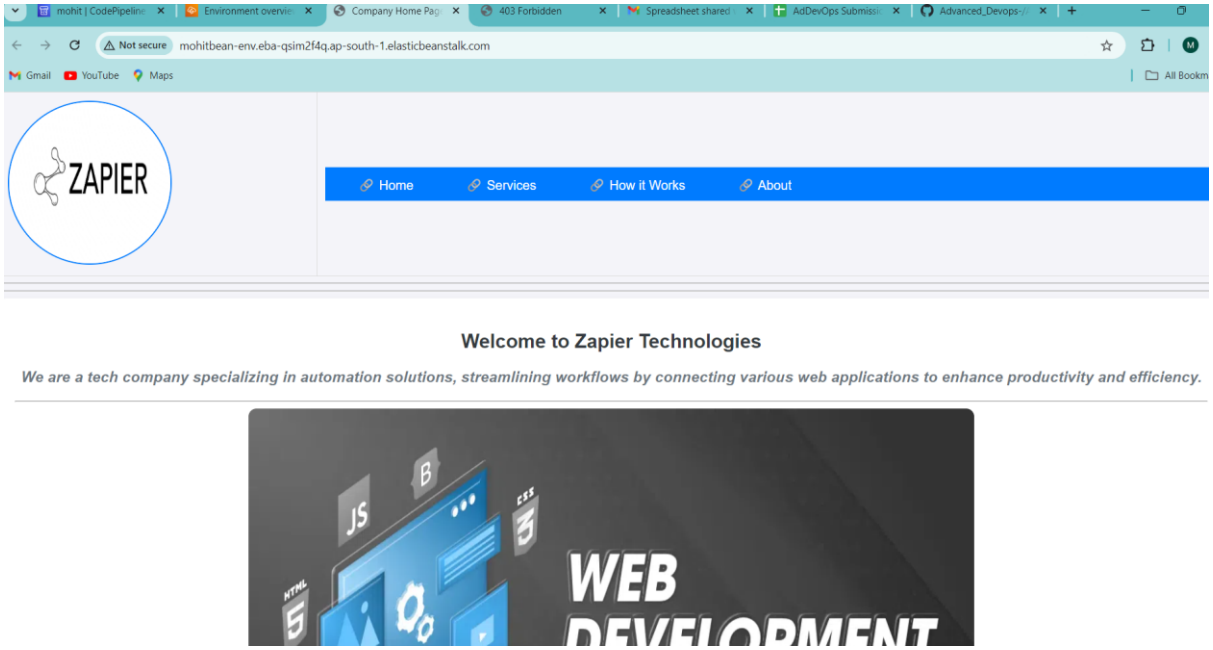
Step 4 : Beanstalk environment is created



view the pipeline build and deployment



Check the deployed website at beanstalk link

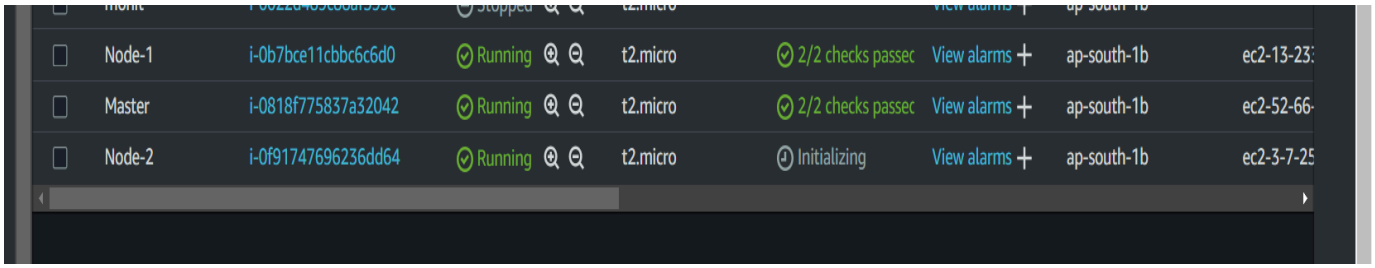


Experiment:3

Name:Mohit Patil Class:D15A Roll no.37

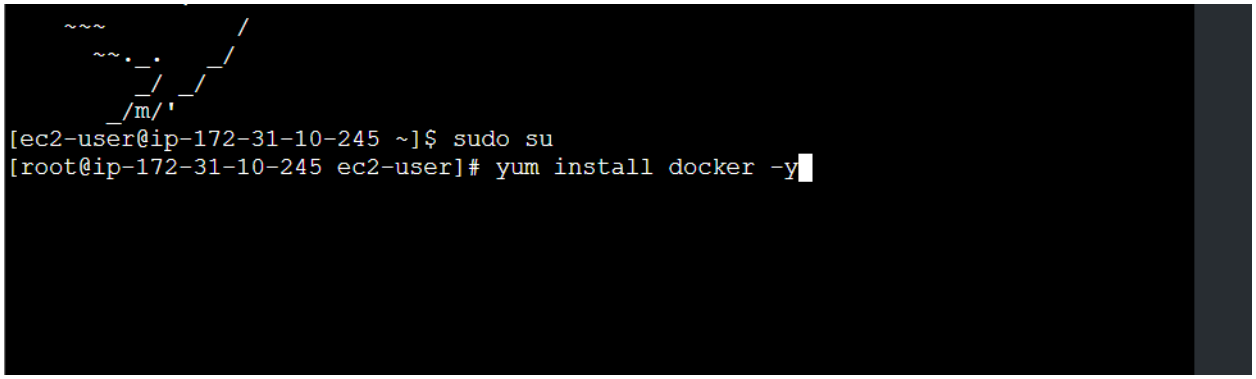
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

1. Create 3 EC2 Ubuntu Instances on AWS.

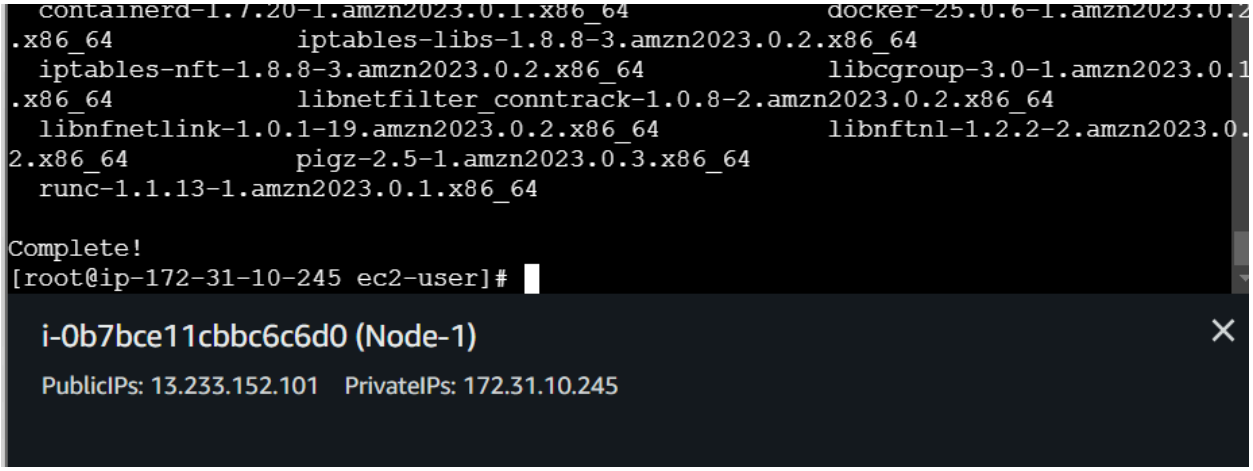


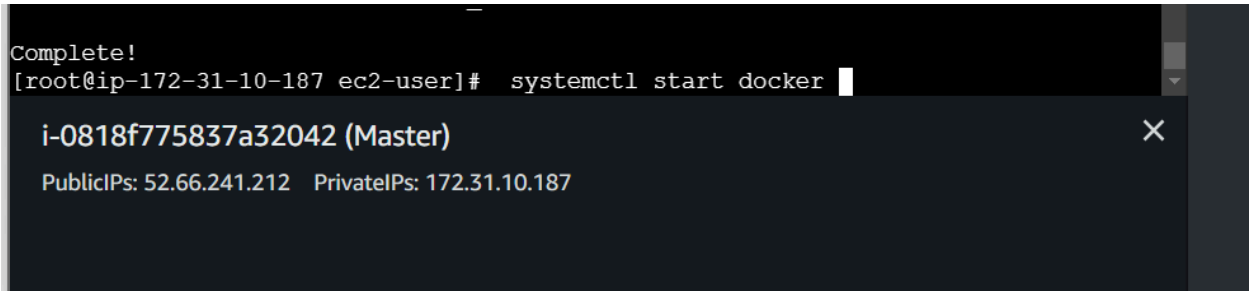
1. Now click on connect to instance, then click on SSH client.
2. Now copy the ssh from the example and paste it on command prompt.(I used gitbash)

3. After this type on all 3 machines Yum install docker -y



4. To start the docker on master and slave perform this command: Systemctl start docker





EXTRA:
To check if docker is installed or not

Docker -v

```
[root@ip-172-31-84-37 ec2-user]# systemctl start docker
[root@ip-172-31-84-37 ec2-user]# sudo su
[root@ip-172-31-84-37 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-84-37 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
```

5. Now to install kubeadm on master and Nodes :

Installing kubeadm:Go the official documentation

off kubeadm.

```
[ec2-user@ip-172-31-14-163 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-14-163 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-14-163 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-14-163 ~]$
```

/ Installing kubeadm

Installing kubeadm

This page shows how to install the `kubeadm` toolbox. For information on how to create a cluster with kubeadm once you have performed this installation process, see the Creating a cluster with kubeadm page.

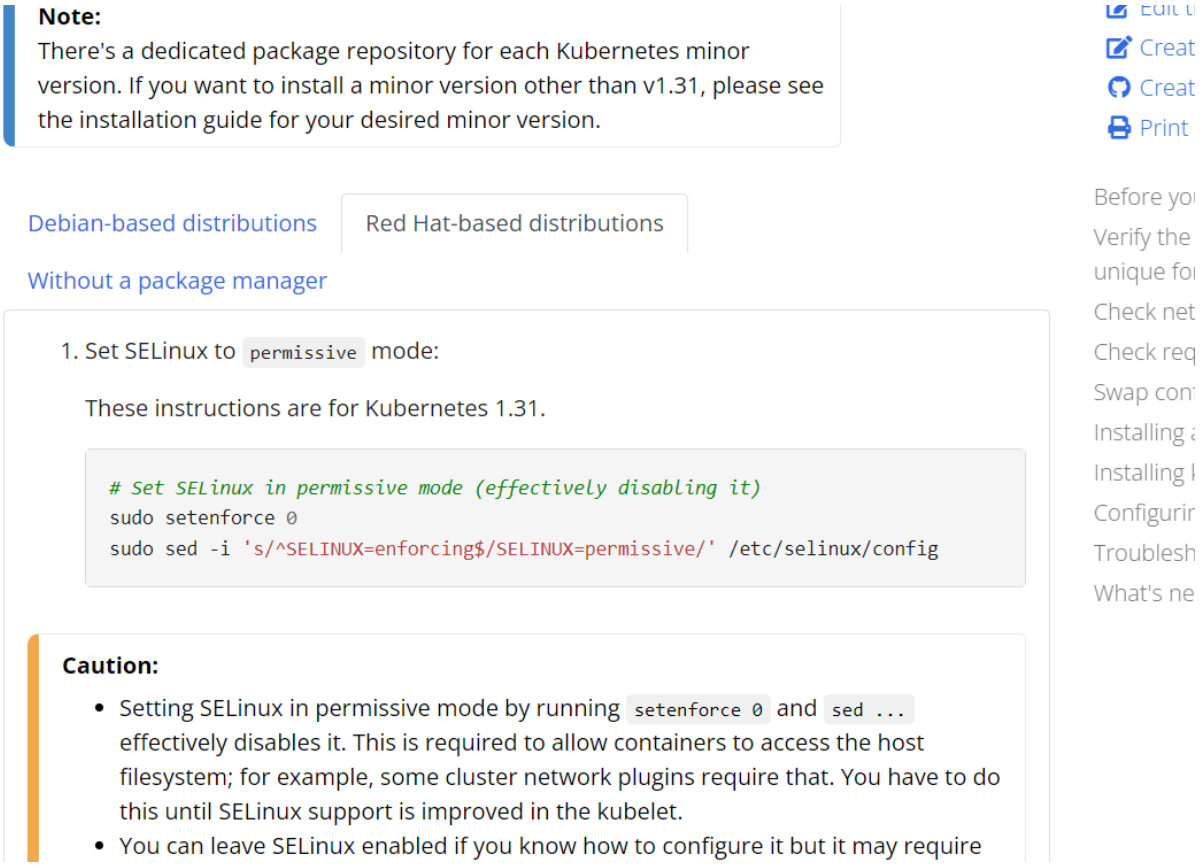
This installation guide is for Kubernetes v1.31. If you want to use a different Kubernetes version, please refer to the following pages instead:

- Installing kubeadm (Kubernetes v1.30)
- Installing kubeadm (Kubernetes v1.29)
- Installing kubeadm (Kubernetes v1.28)
- Installing kubeadm (Kubernetes v1.27)

Before you begin

- A compatible Linux host. The Kubernetes project provides generic instructions for Linux distributions based on Debian and Red Hat, and

6. Scroll down and select Red Hat based distributions:



Note:
There's a dedicated package repository for each Kubernetes minor version. If you want to install a minor version other than v1.31, please see the installation guide for your desired minor version.

Debian-based distributions | Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

Caution:

- Setting SELinux in permissive mode by running `setenforce 0` and `sed ...` effectively disables it. This is required to allow containers to access the host filesystem; for example, some cluster network plugins require that. You have to do this until SELinux support is improved in the kubelet.
- You can leave SELinux enabled if you know how to configure it but it may require

Edit t
Creat
Creat
Print

Before yo
Verify the
unique fo
Check net
Check req
Swap con
Installing
Installing
Configurir
Troublesh
What's ne

7.Now copy the command on all 3 machines:

1. Set SELinux to `permissive` mode:

   These instructions are for Kubernetes 1.31.

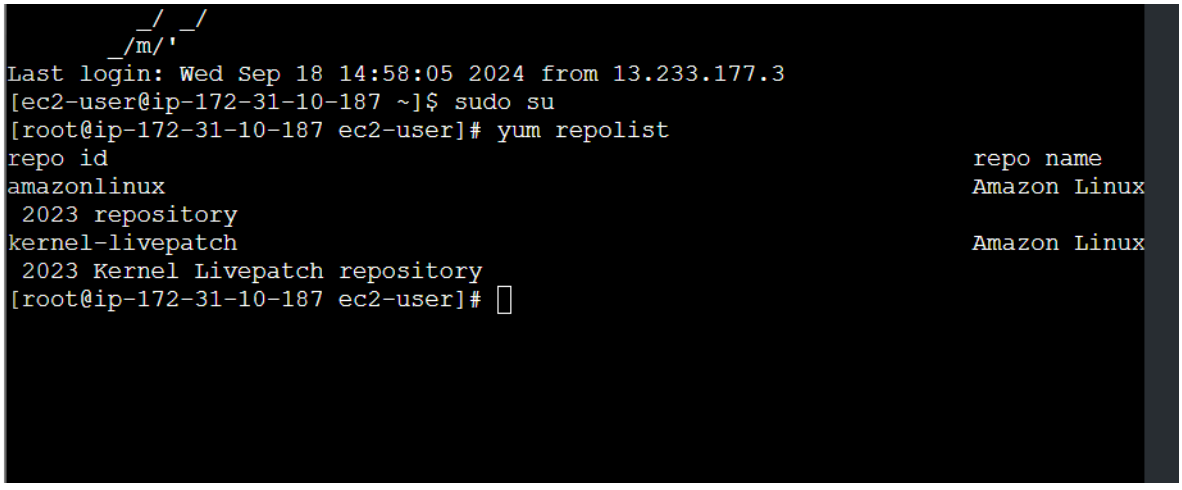
```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

3. Install kubelet, kubeadm and kubectl:

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

8.Run yum repolist command for checking the repositorires



```
8/9
  Verifying         : kubernetes-cni-1.5.1-150500.1.1.x86_64

9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64            cri-tools-1.31.1-150500
.1.1.x86_64                        kubeadm-1.31.1-150500.1.1.x86_64

  kubectl-1.31.1-150500.1.1.x86_64                       kubelet-1.31.1-150500.1
.1.x86_64                          kubernetes-cni-1.5.1-150500.1.1.x86_64

  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64     libnetfilter_cttimeout-
1.0.0-19.amzn2023.0.2.x86_64     libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64


Complete!
[root@ip-172-31-10-187 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /
usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-10-187 ec2-user]#
```

i-0818f775837a32042 (Master)

PublicIPs: 52.66.241.212 PrivateIPs: 172.31.10.187

```
Complete!
[root@ip-172-31-10-187 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-10-187 ec2-user]# yum repolist
repo id                                                repo name
amazonlinux                                            Amazon Linux 2023 repository
kernel-livepatch                                       Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                             Kubernetes
[root@ip-172-31-10-187 ec2-user]#
```

i-0818f775837a32042 (Master)

PublicIPs: 52.66.241.212 PrivateIPs: 172.31.10.187

EXTRA :

Got error in initializing Kubernetes

```
[root@ip-172-31-31-240 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0908 11:25:45.820964    2320 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking t
RI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containe
ble desc = connection error: desc = "transport: Error while dialing: dial unix /var/run/containerd/containerd.sock:
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
```

9.Copy paste the commands in all three instances

```
[root@ip-172-31-10-187 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
       [WARNING FileExisting-socat]: socat not found in system path
       [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0918 15:44:45.531044    3429 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used b
y kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
```

```
  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.10.187:6443 --token v5qt9c.8kc71gb6xwil18co \
        --discovery-token-ca-cert-hash sha256:b9765a8ba4dc546e8263bd7b7531f78b4b8582c9757a9371e00ab90a71a38785
[root@ip-172-31-10-187 ec2-user]#    mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-10-187 ec2-user]#    export KUBECONFIG=/etc/kubernetes/admin.conf
[root@ip-172-31-10-187 ec2-user]#
```

i-0818f775837a32042 (Master)

PublicIPs: 13.126.147.65 PrivateIPs: 172.31.10.187

10.After pasting the connection link in the nodes run the kublect get nodes command to view the connected nodes successfully

```
ubuntu@ip-172-31-17-23:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-17-23    Ready    control-plane   3m56s   v1.29.0
ip-172-31-18-12    Ready    <none>          37s     v1.29.0
ip-172-31-26-153   Ready    <none>          24s     v1.29.0
ubuntu@ip-172-31-17-23:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-17-23    Ready    control-plane   9m34s   v1.29.0
ip-172-31-18-12    Ready    <none>          6m15s   v1.29.0
ip-172-31-26-153   Ready    <none>          6m2s    v1.29.0
ubuntu@ip-172-31-17-23:~$
```

Experiment:4

Name:Mohit Patil Class:D15A Roll no.37

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

What is Kubernetes?

Kubernetes, often referred to as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Originally developed by Google, it has become the industry standard for managing container workloads due to its flexibility and robust features.

- **Core Concepts of Kubernetes**
  - Containers: These are lightweight, portable packages that include everything needed to run an application, ensuring consistency across different environments.
-
  - Pods: The smallest deployable units in Kubernetes, pods can contain one or more containers that share storage and network resources.
-
  - Nodes: A node is a worker machine in the Kubernetes cluster that runs at least one pod. Nodes can be either physical or virtual machines.
-
  - Clusters: A cluster comprises multiple nodes that run containerized applications. The control plane manages the cluster's state.
-
  - Services: Services provide stable endpoints for accessing pods and facilitate load balancing and service discovery.
-
  - Deployments: A deployment manages the lifecycle of pods, allowing users to specify the number of replicas and facilitating rolling updates and rollbacks.

Role of Kubernetes

What is Kubectl?

Kubectl is the command-line interface used to interact with the Kubernetes API server. It enables users to manage resources within a Kubernetes cluster effectively.

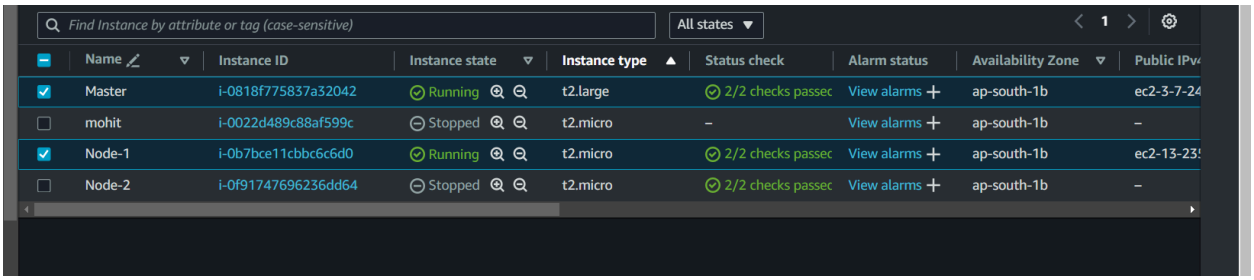
Configuration Files

Configuration files are essential for defining how resources should be created or modified within Kubernetes. Users can employ declarative configurations (using YAML/JSON files) or imperative commands directly in the terminal.

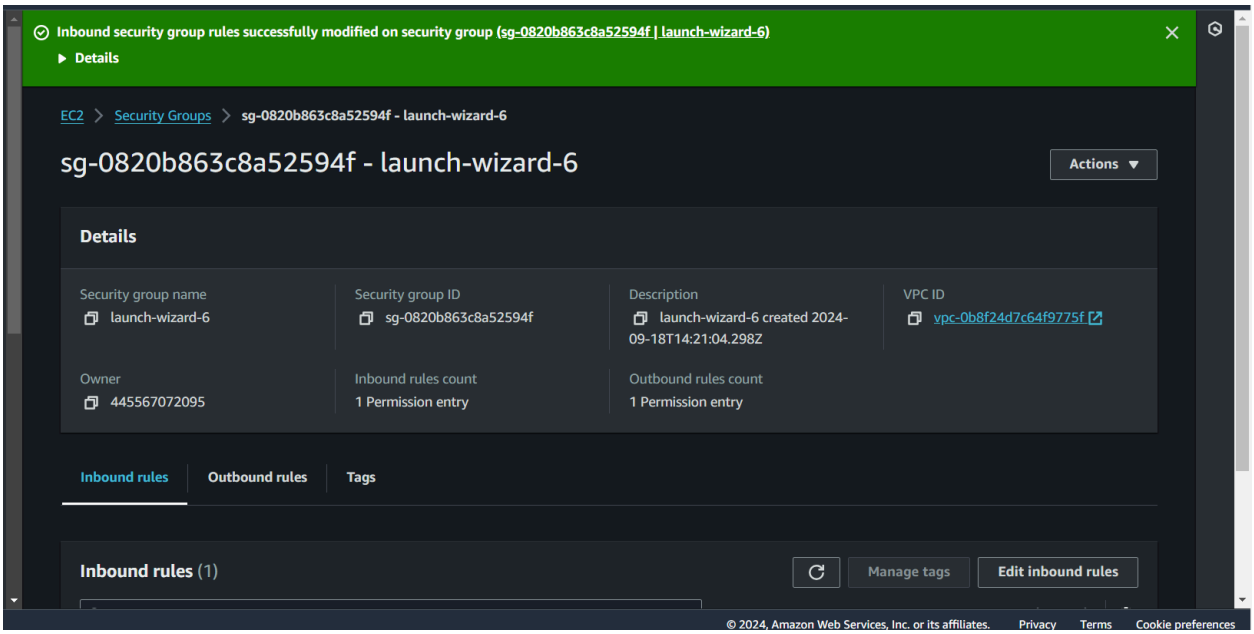
Application Deployment on Kubernetes

- Define Application Requirements: Identify necessary resources such as CPU, memory, storage, etc.
- Create Deployment Configurations: Write deployment manifests specifying container images, replicas for scaling, health checks, etc.
- Deploying with Kubectl: Use kubectl commands like kubectl apply to deploy applications based on these configurations.
- Monitoring and Scaling Applications: Monitor performance metrics and adjust deployments based on traffic demands.
- Updating Applications: Modify deployment configurations for updates; Kubernetes supports rolling updates by default.
- Rollback Capabilities: If an update causes issues, kubectl allows easy rollback to previous versions using commands like kubectl rollout undo.

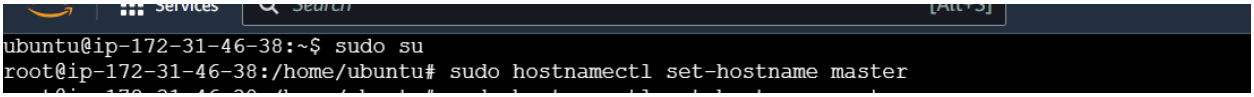
Step 1.Creation of 2 EC2 Ubuntu Instances on AWS.



Step 2.Edit inbound rules of security group ‘launch-wizard-1’ and set ‘All Traffic’



Step 3. Set master and worker as hostname on respective servers



Step 4.Installation of docker

```
root@ip-172-31-46-38:/home/ubuntu# sudo hostnamectl set-hostname master
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [528 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [127 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8352 B]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [368 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [151 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
```

```
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 133 not upgraded.
```

```
sudo: systmectl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2364 (dockerd)
      Tasks: 9
     Memory: 25.7M (peak: 26.0M)
        CPU: 203ms
     CGroup: /system.slice/docker.service
             └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 versio
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

i-0bd5556e5332ef8aa (master-1)

Step 5.Installation of Kubernetes-


```
sudo: systmectl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2364 (dockerd)
      Tasks: 9
     Memory: 25.7M (peak: 26.0M)
        CPU: 203ms
     CGroup: /system.slice/docker.service
             └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 versio
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

i-0bd5556e5332ef8aa (master-1)

```
root@ip-172-31-46-38:/home/ubuntu#       install ca certificate
install: cannot stat 'ca': No such file or directory
root@ip-172-31-46-38:/home/ubuntu# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user]
            [-u user] [command [arg ...]]
usage: sudo [-ABbEHkNnPS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNnS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] file ...
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
```

Step.6 Kubernetes Deployment

```
E: Unable to locate package kubeadm
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu10.4 [341 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2ubuntu10.4 [333 kB]
Fetched 904 kB in 0s (26.6 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68108 files and directories currently installed.)
```

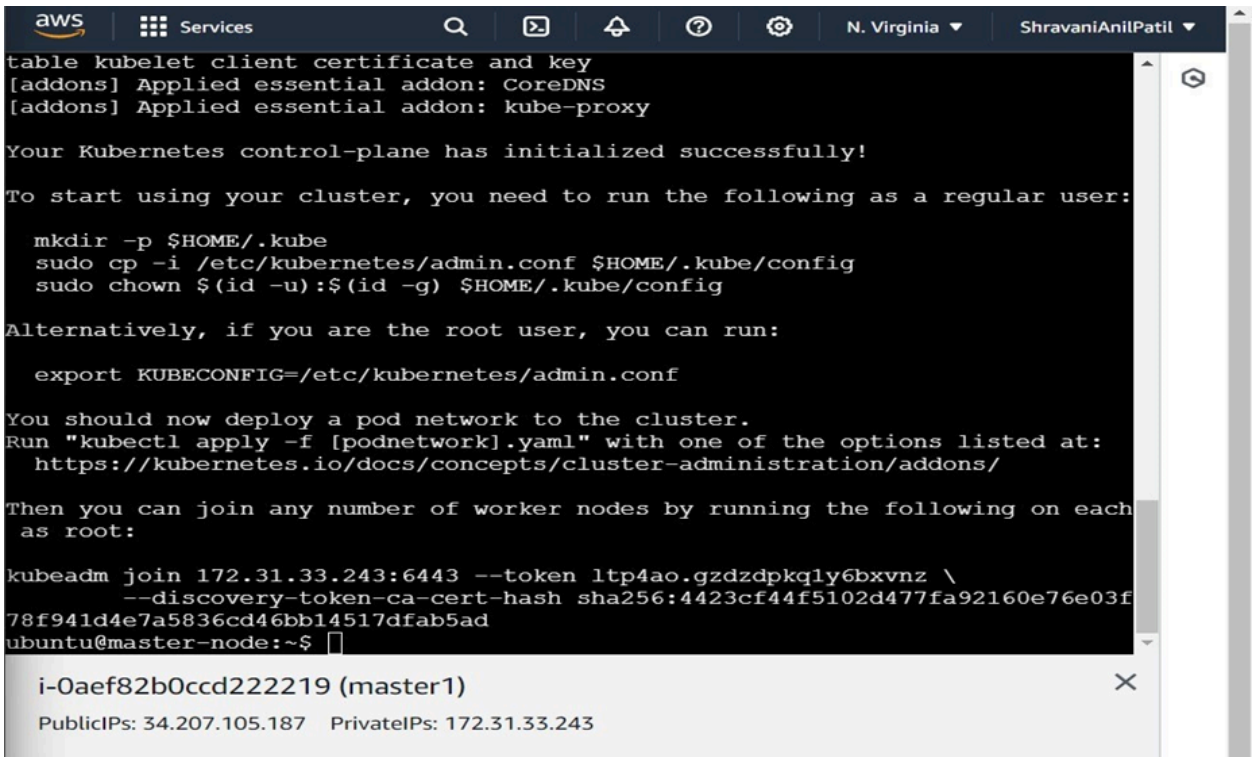
```
bash: https://packages.cloud.google.com/apt/doc/apt-key.gpg: No such file or directory
root@ip-172-31-46-38:/home/ubuntu# sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
root@ip-172-31-46-38:/home/ubuntu#
```

```
bash: /etc/apt/sources.list.d/kubernetes.list: Permission denied
root@ip-172-31-46-38:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | su
do tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
root@ip-172-31-46-38:/home/ubuntu# sudo apt get update
E: Invalid operation get
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 142.251.42.14 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu#
```

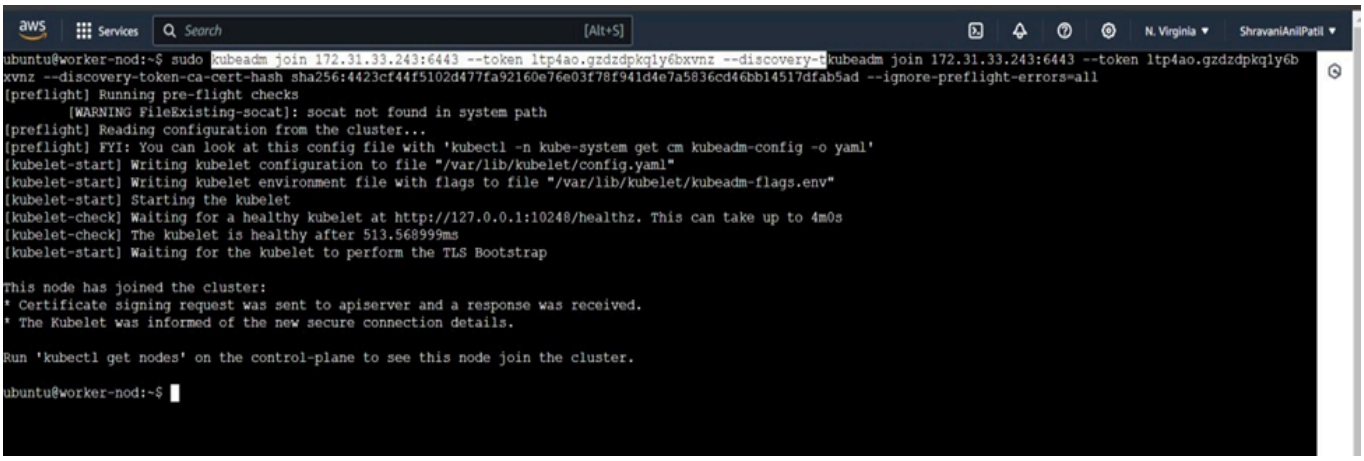
```
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 142.250.192.142 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu#
```

Extra:

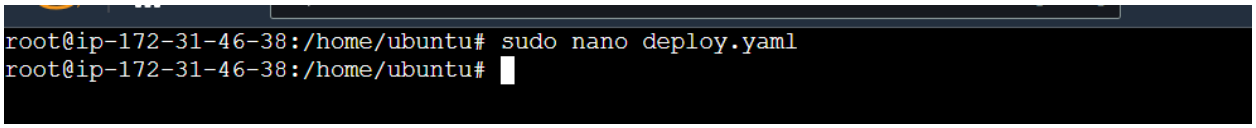
```
kubectl 1.31.1 from Canonical✓ installed
root@ip-172-31-46-38:/home/ubuntu# kubectl version --client
Client Version: v1.31.1
Kustomize Version: v5.4.2
root@ip-172-31-46-38:/home/ubuntu#
```



Step 7. Deploy Pod Network to Cluster and Join Worker Node to Cluster



Step 8. Create one file deploy.yaml



```
ubuntu@master-node:~$ cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@master-node:~$
```

Step 9 : Create Deployment

```
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@ip-172-31-46-38:/home/ubuntu# kubectl create -f deploy.yaml
```

```
service/nginx-deployment exposed
ubuntu@master-node:~$ kubectl get svc
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes         ClusterIP      10.96.0.1      <none>        443/TCP        4h43m
nginx-deployment   LoadBalancer   10.101.59.94   <pending>     80:31041/TCP   4m34s
ubuntu@master-node:~$
```

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

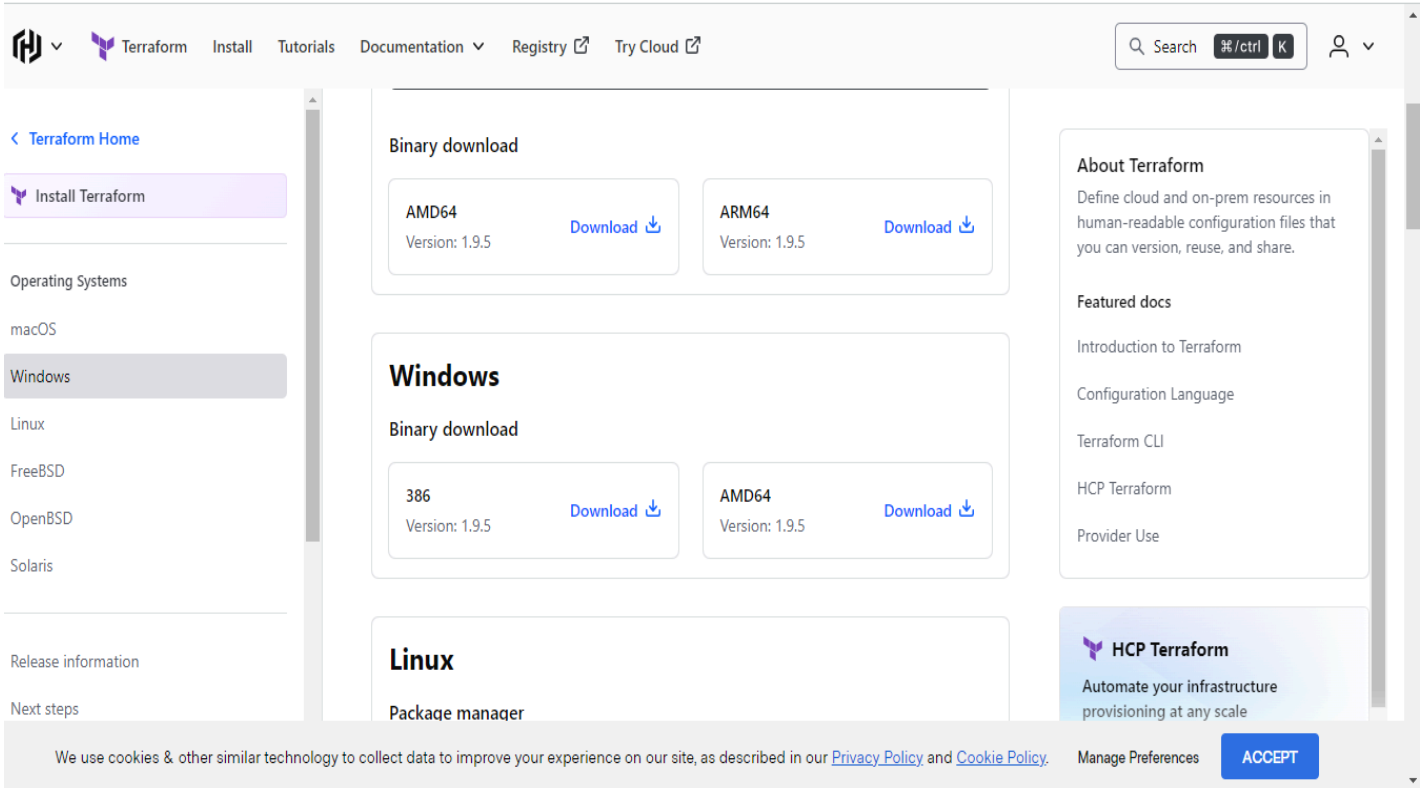
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

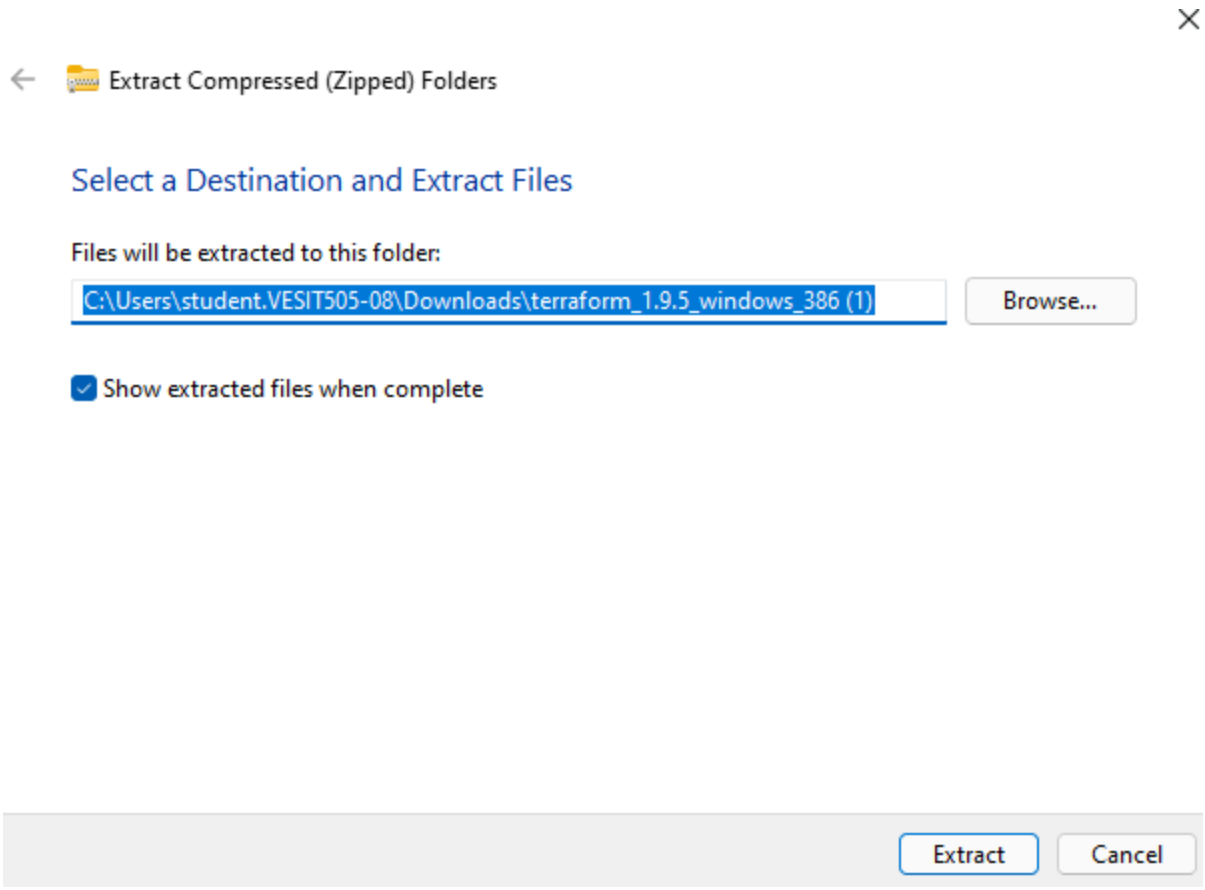
Thank you for using nginx.

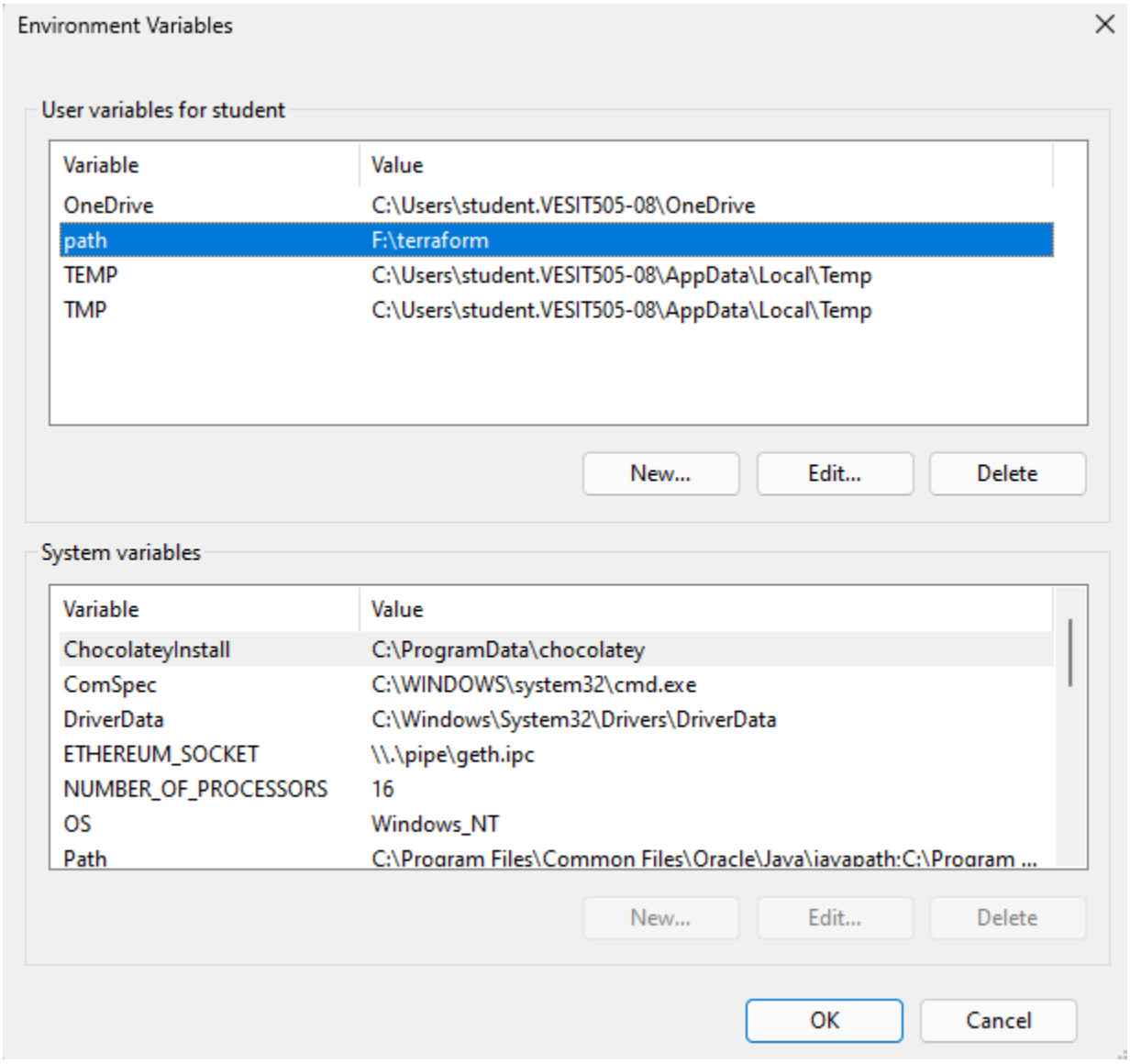
EXPERIMENT 5

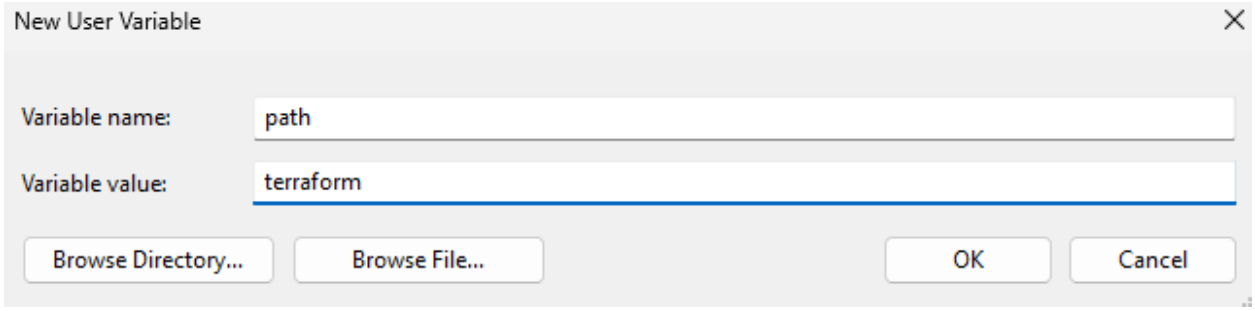
NAME:MOHIT PATIL DIV:D15A ROLL NO:37

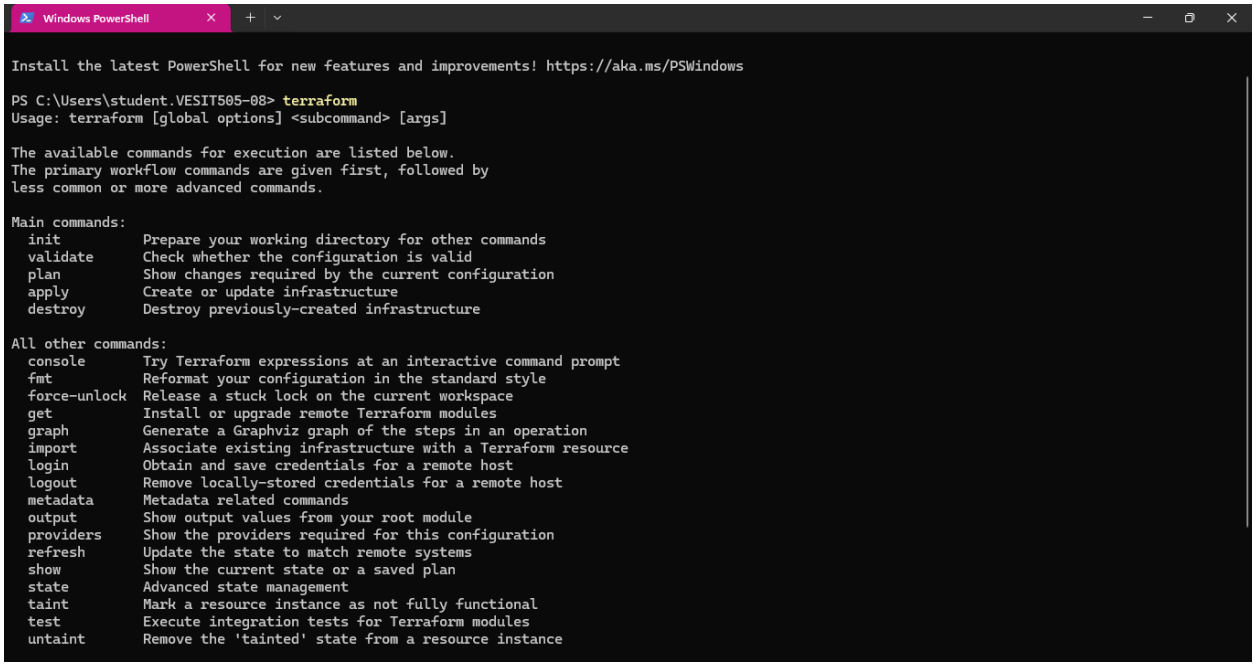
Aim:Installation and configuration of Terraform in windows




Extract Compressed (Zipped) Folders
Select a Destination and Extract Files
Files will be extracted to this folder:
C:\Users\student.VESIT505-08\Downloads\terraform_1.9.5_windows_386 (1)
Browse...
Show extracted files when complete
Extract
Cancel


Environment Variables
User variables for student
Variable
Value
OneDrive
C:\Users\student.VESIT505-08\OneDrive
path
F:\terraform
TEMP
C:\Users\student.VESIT505-08\AppData\Local\Temp
TMP
C:\Users\student.VESIT505-08\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ChocolateyInstall
C:\ProgramData\chocolatey
ComSpec
C:\WINDOWS\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
ETHEREUM_SOCKET
\\.\pipe\geth.ipc
NUMBER_OF_PROCESSORS
16
OS
Windows_NT
Path
C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
New...
Edit...
Delete
OK
Cancel


New User Variable
Variable name:
path
Variable value:
terraform
Browse Directory...
Browse File...
OK
Cancel

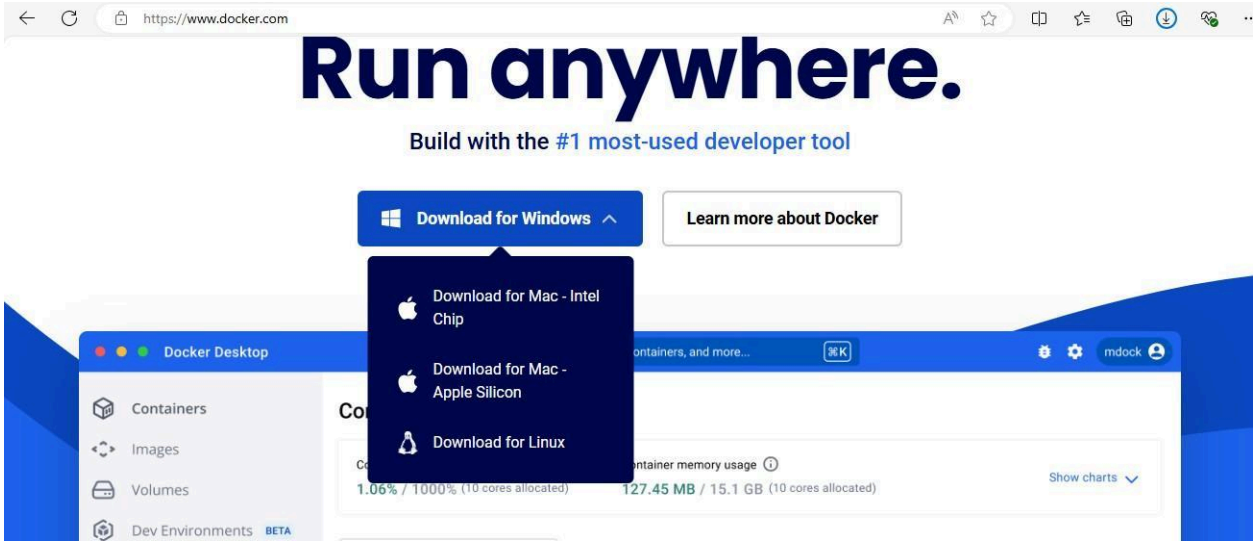

Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\student.VESIT505-08> terraform
Usage: terraform [global options] <subcommand> [args]
The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.
Main commands:
init Prepare your working directory for other commands
validate Check whether the configuration is valid
plan Show changes required by the current configuration
apply Create or update infrastructure
destroy Destroy previously-created infrastructure
All other commands:
console Try Terraform expressions at an interactive command prompt
fmt Reformat your configuration in the standard style
force-unlock Release a stuck lock on the current workspace
get Install or upgrade remote Terraform modules
graph Generate a Graphviz graph of the steps in an operation
import Associate existing infrastructure with a Terraform resource
login Obtain and save credentials for a remote host
logout Remove locally-stored credentials for a remote host
metadata Metadata related commands
output Show output values from your root module
providers Show the providers required for this configuration
refresh Update the state to match remote systems
show Show the current state or a saved plan
state Advanced state management
taint Mark a resource instance as not fully functional
test Execute integration tests for Terraform modules
untaint Remove the 'tainted' state from a resource instance

Mohit Patil **D15A - 37**

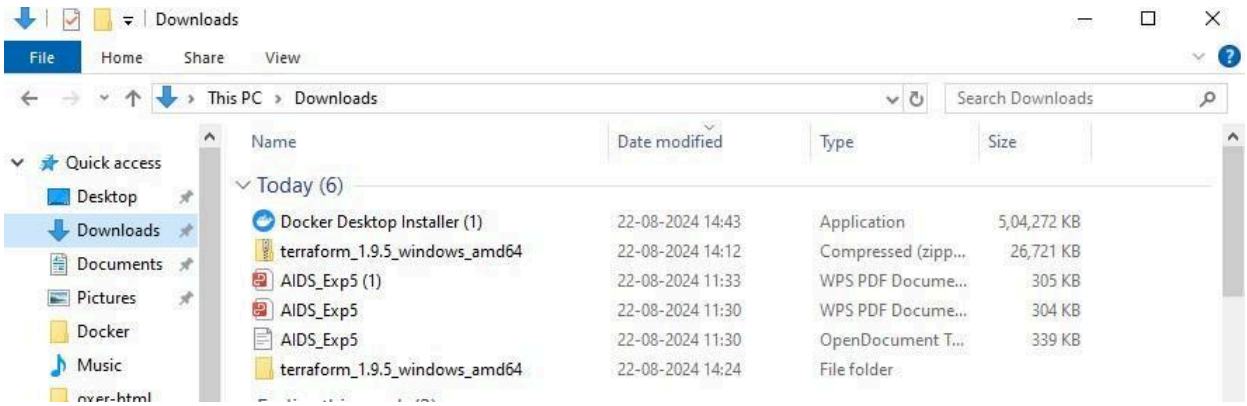
Experiment 6

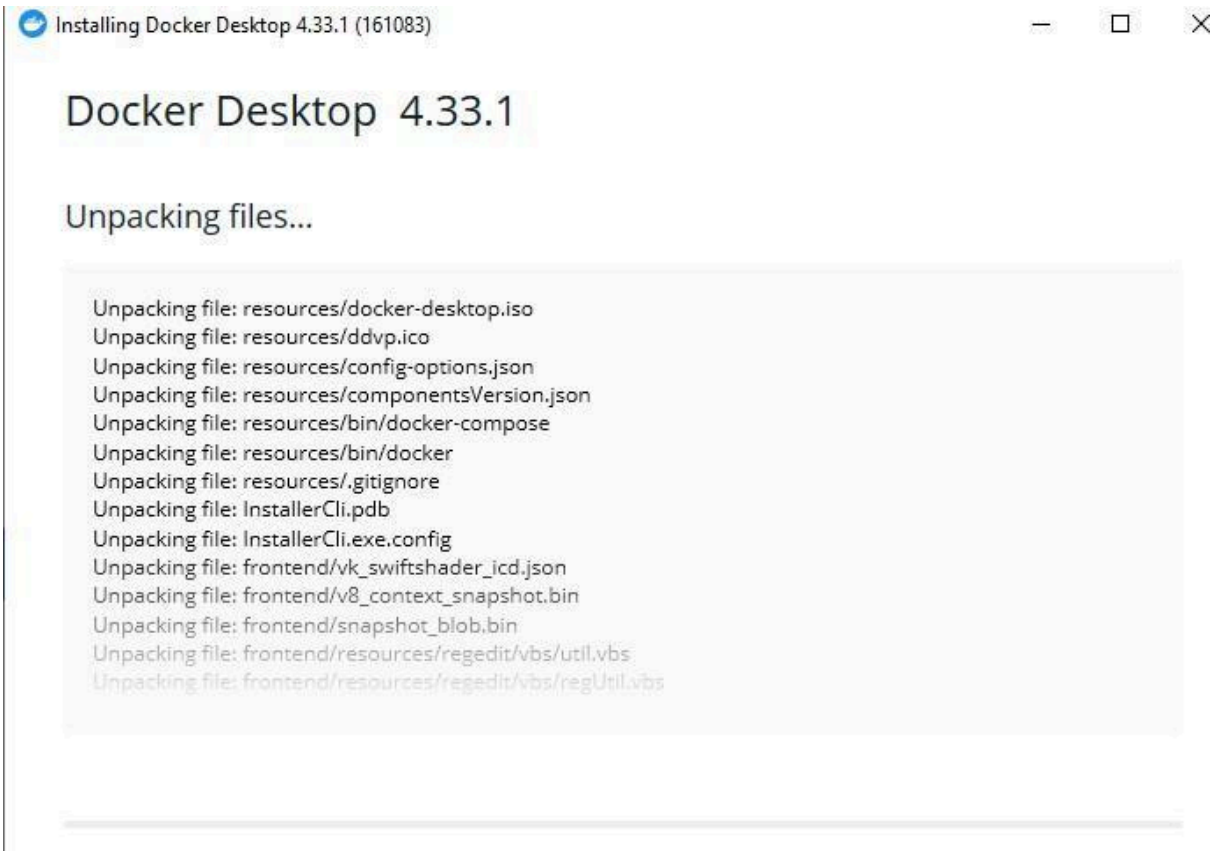
Aim:
To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.(S3 bucket or Docker)

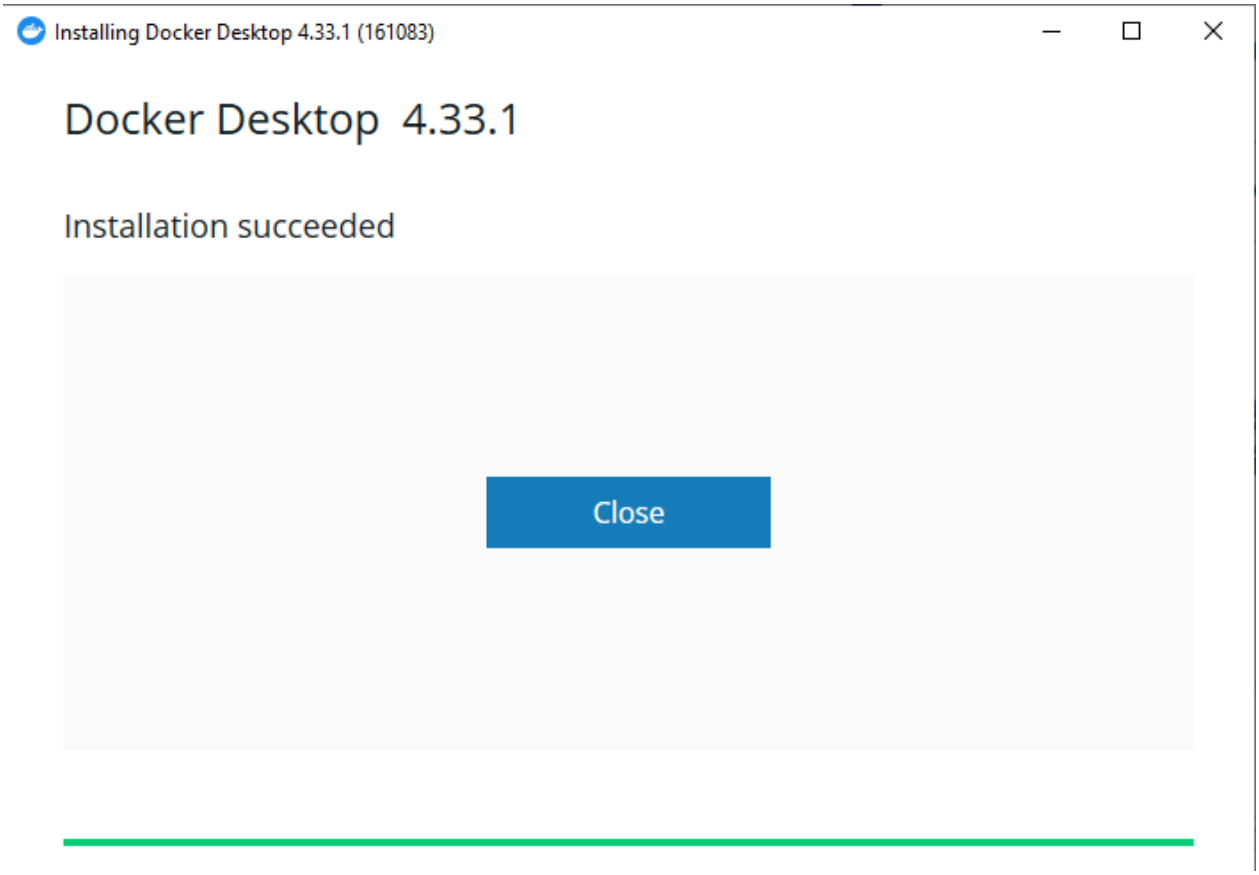
Step 1: Download Docker form www.docker.com



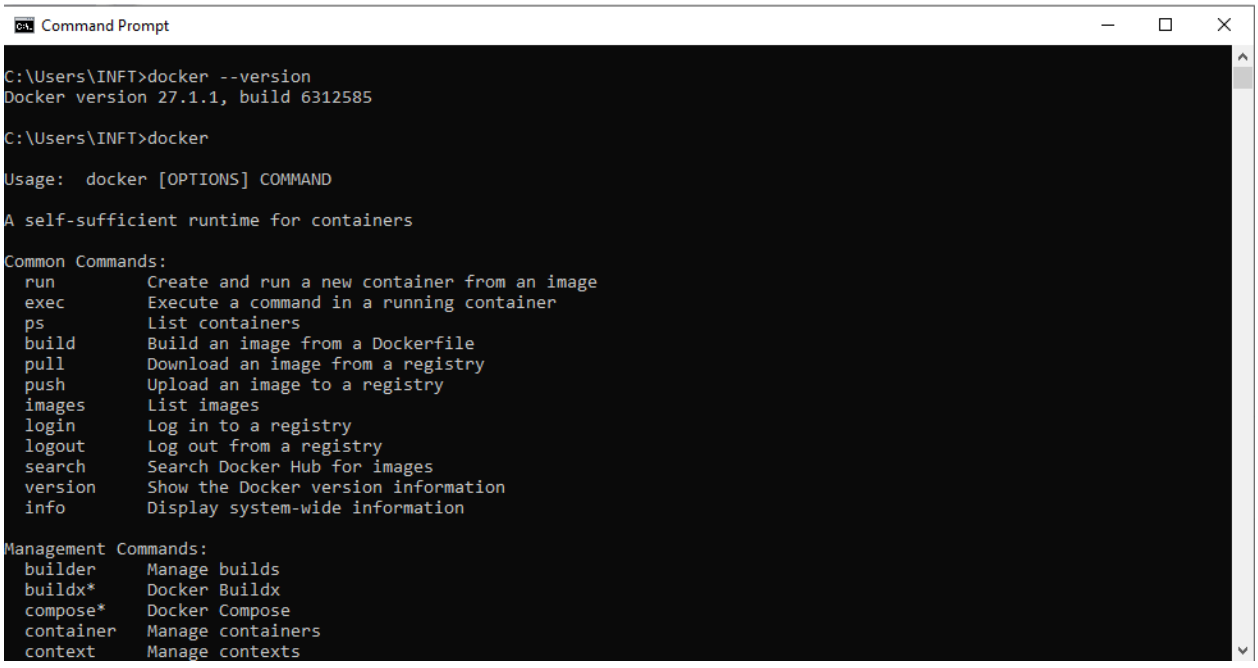
Step 2: The Docker is successfully downloaded. Now, run the docker installer and complete the installation.



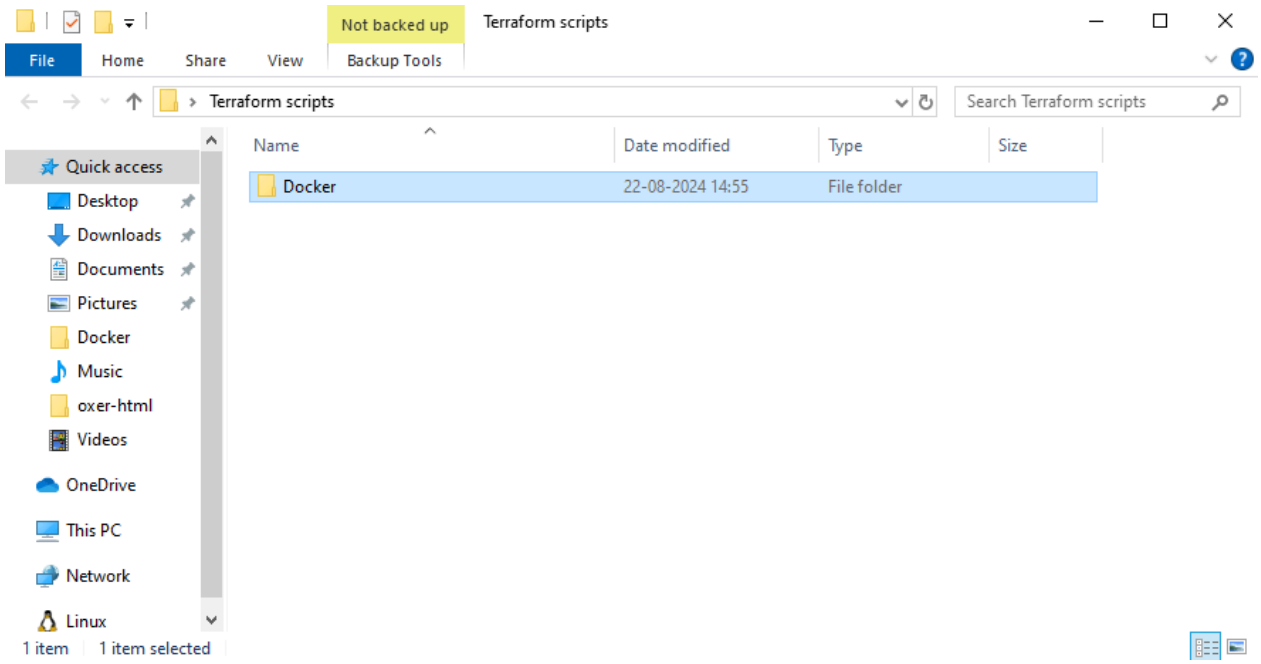

Installing Docker Desktop 4.33.1 (161083)
Docker Desktop 4.33.1
Unpacking files...
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/util.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs



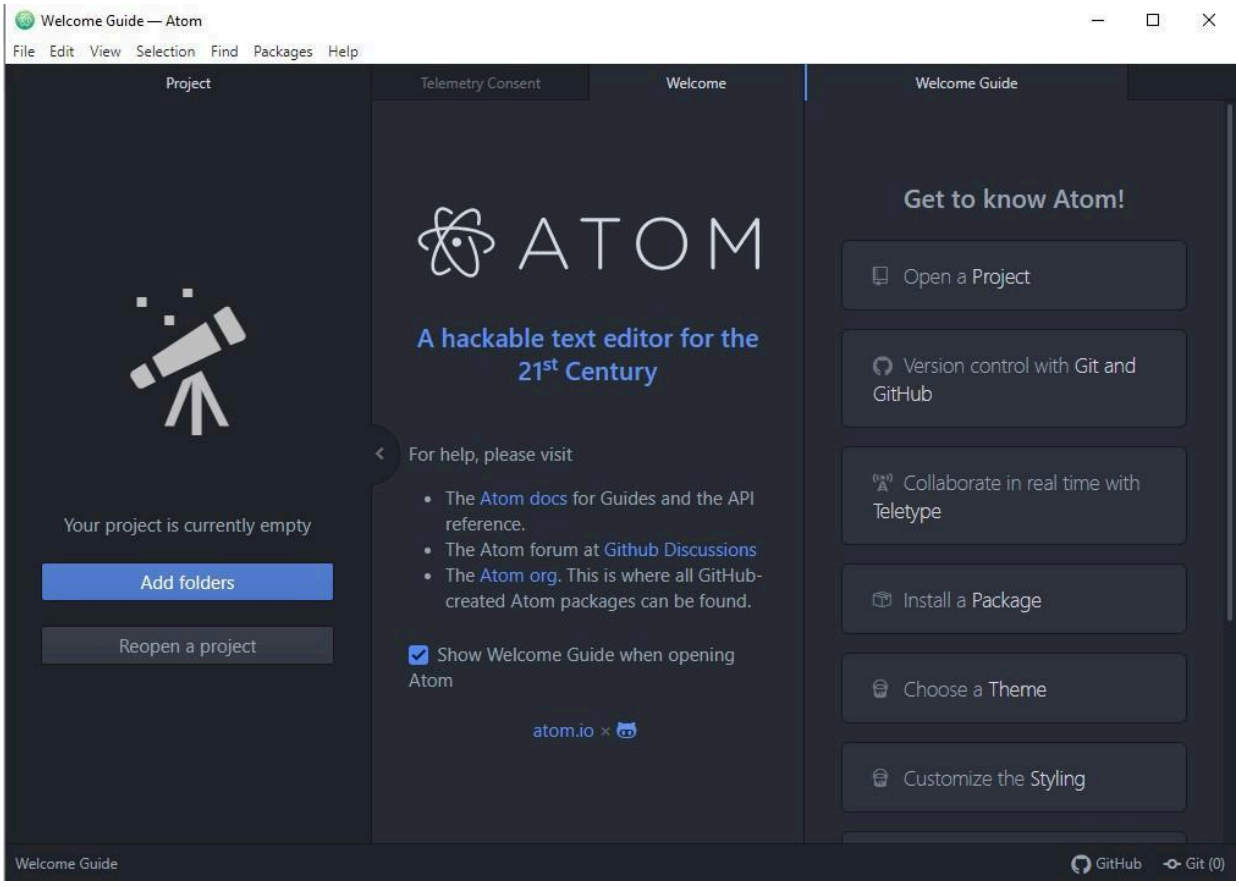
Step 3: Open Command Prompt and run as administrator. Enter the command docker –version, to check whether the docker is successfully installed.



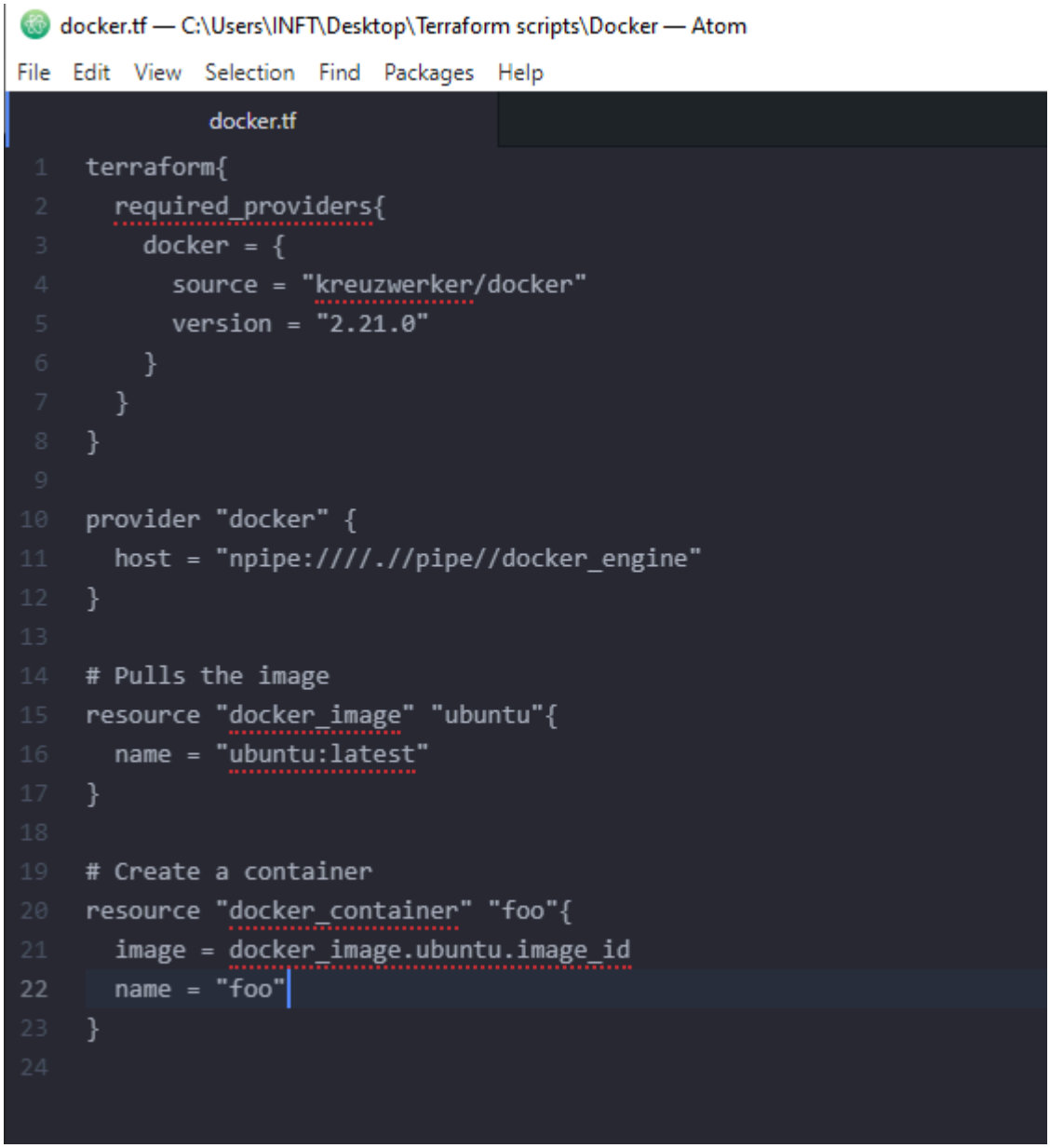
Step 4: Create a folder Terraform_scripts and inside it create a folder named Docker.



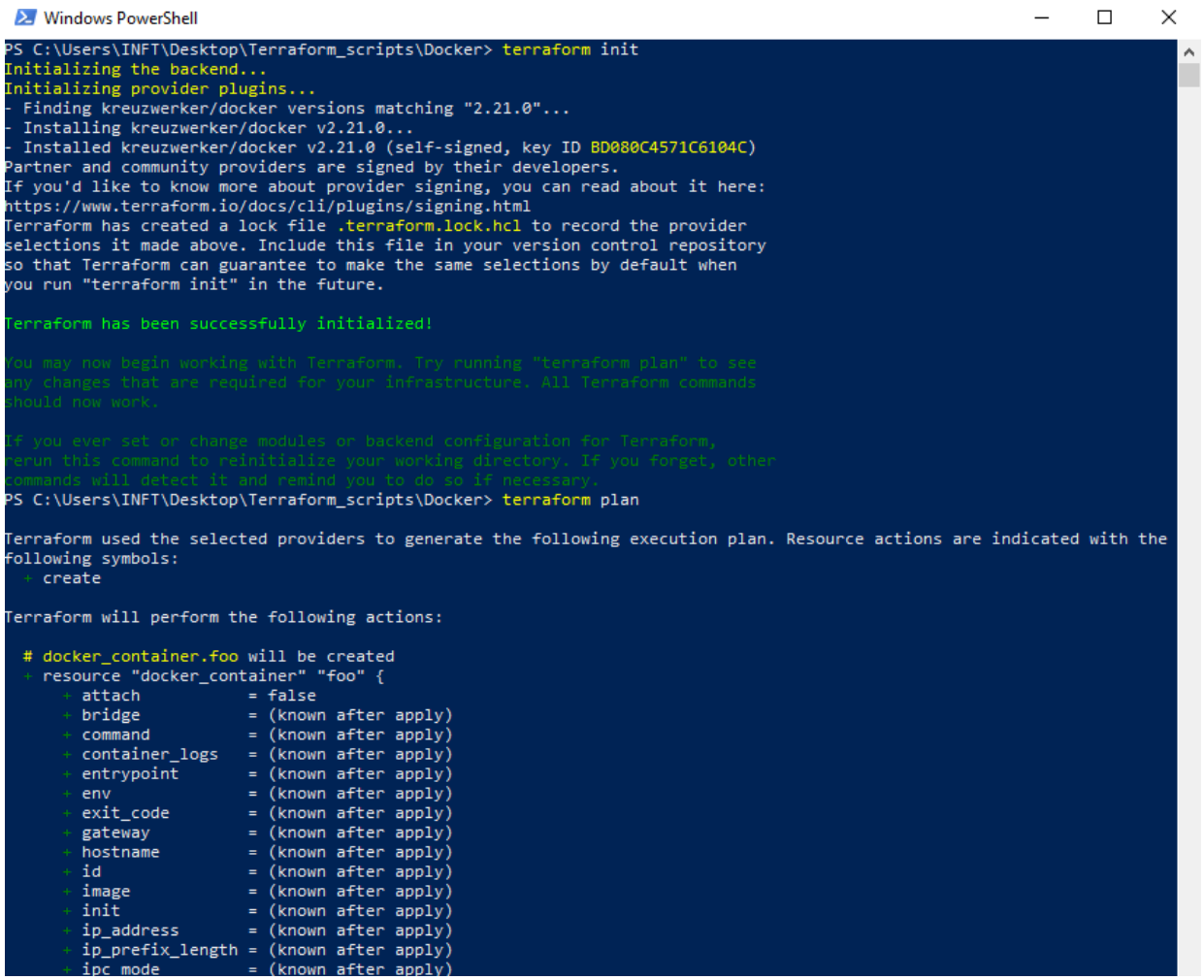
Step 5: Download Atom Editor.

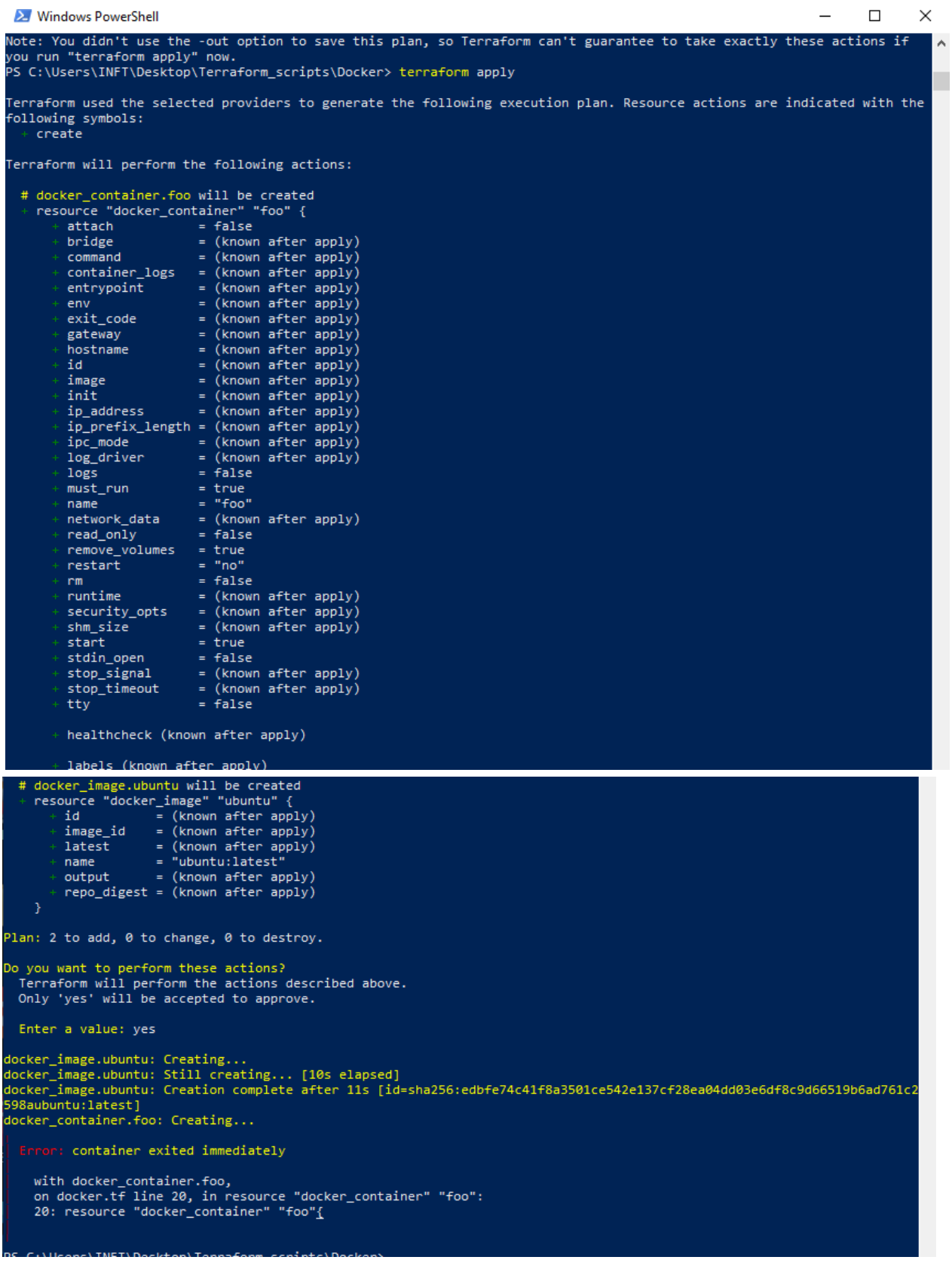


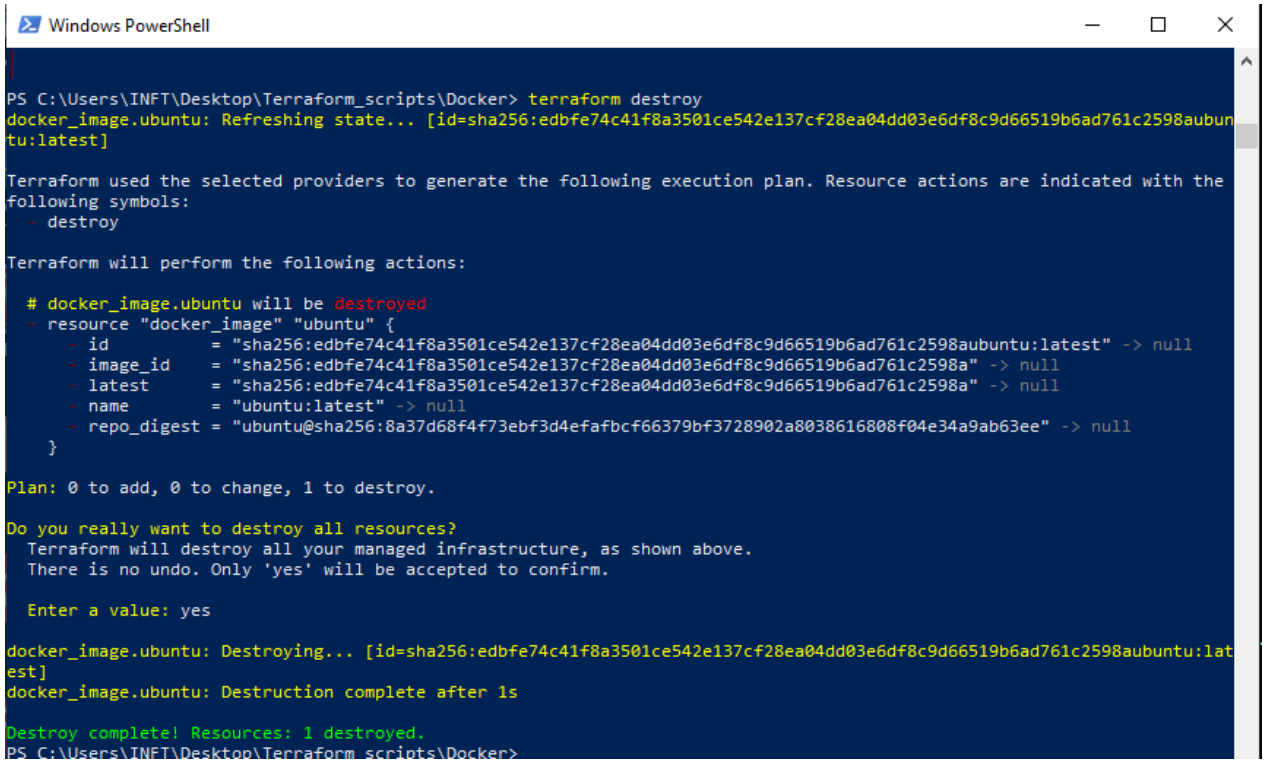
Step 6: Run the following script in the Atom Editor

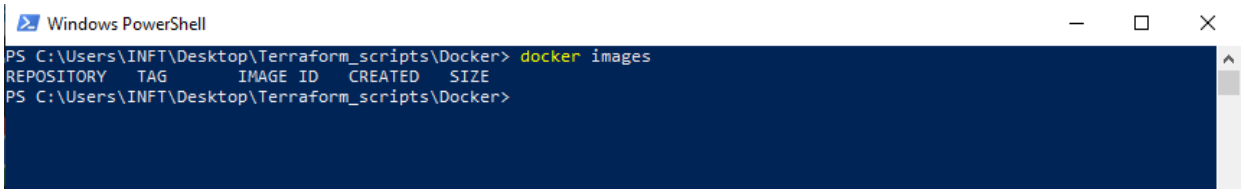


Step 7: Open Windows Explorer and run the following command terraform init, terraform plan, terraform apply, terraform destroy and docker images.


Windows PowerShell
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
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.
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> 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:
# docker_container.foo will be created
+ resource "docker_container" "foo" {
+ attach = false
+ bridge = (known after apply)
+ command = (known after apply)
+ container_logs = (known after apply)
+ entrypoint = (known after apply)
+ env = (known after apply)
+ exit_code = (known after apply)
+ gateway = (known after apply)
+ hostname = (known after apply)
+ id = (known after apply)
+ image = (known after apply)
+ init = (known after apply)
+ ip_address = (known after apply)
+ ip_prefix_length = (known after apply)
+ ipc_mode = (known after apply)


Windows PowerShell
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.
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> terraform apply
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:
# docker_container.foo will be created
+ resource "docker_container" "foo" {
+ attach = false
+ bridge = (known after apply)
+ command = (known after apply)
+ container_logs = (known after apply)
+ entrypoint = (known after apply)
+ env = (known after apply)
+ exit_code = (known after apply)
+ gateway = (known after apply)
+ hostname = (known after apply)
+ id = (known after apply)
+ image = (known after apply)
+ init = (known after apply)
+ ip_address = (known after apply)
+ ip_prefix_length = (known after apply)
+ ipc_mode = (known after apply)
+ log_driver = (known after apply)
+ logs = false
+ must_run = true
+ name = "foo"
+ network_data = (known after apply)
+ read_only = false
+ remove_volumes = true
+ restart = "no"
+ rm = false
+ runtime = (known after apply)
+ security_opts = (known after apply)
+ shm_size = (known after apply)
+ start = true
+ stdin_open = false
+ stop_signal = (known after apply)
+ stop_timeout = (known after apply)
+ tty = false
+ healthcheck (known after apply)
+ labels (known after apply)
# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
+ id = (known after apply)
+ image_id = (known after apply)
+ latest = (known after apply)
+ name = "ubuntu:latest"
+ output = (known after apply)
+ repo_digest = (known after apply)
}
Plan: 2 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Creation complete after 11s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
Error: container exited immediately
with docker_container.foo,
on docker.tf line 20, in resource "docker_container" "foo":
20: resource "docker_container" "foo"{


Windows PowerShell
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
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:
# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
- id = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
- image_id = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
- latest = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
- name = "ubuntu:latest" -> null
- repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
}
Plan: 0 to add, 0 to change, 1 to destroy.
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s
Destroy complete! Resources: 1 destroyed.
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker>

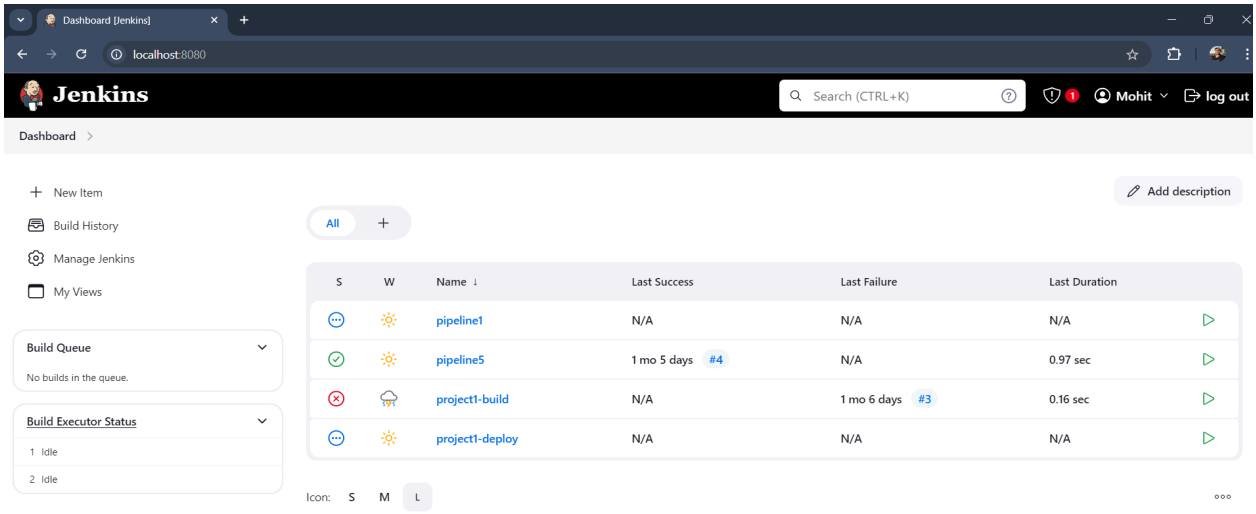

Windows PowerShell
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker>

Mohit Patil D15A 37

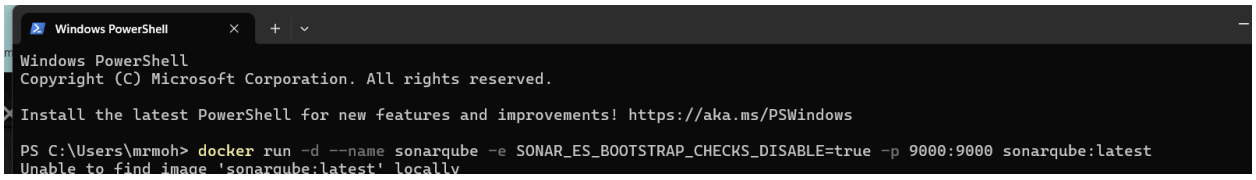
EXPERIMENT 7

Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

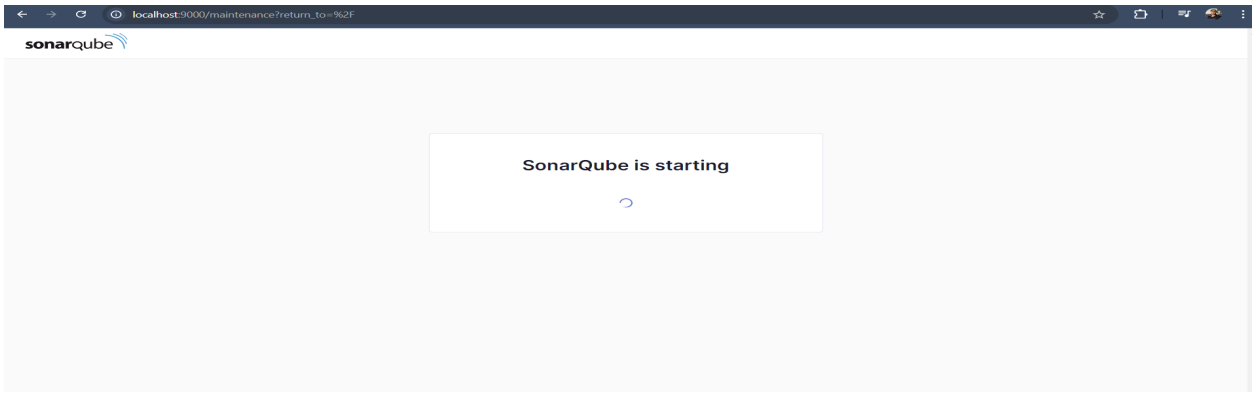
- Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



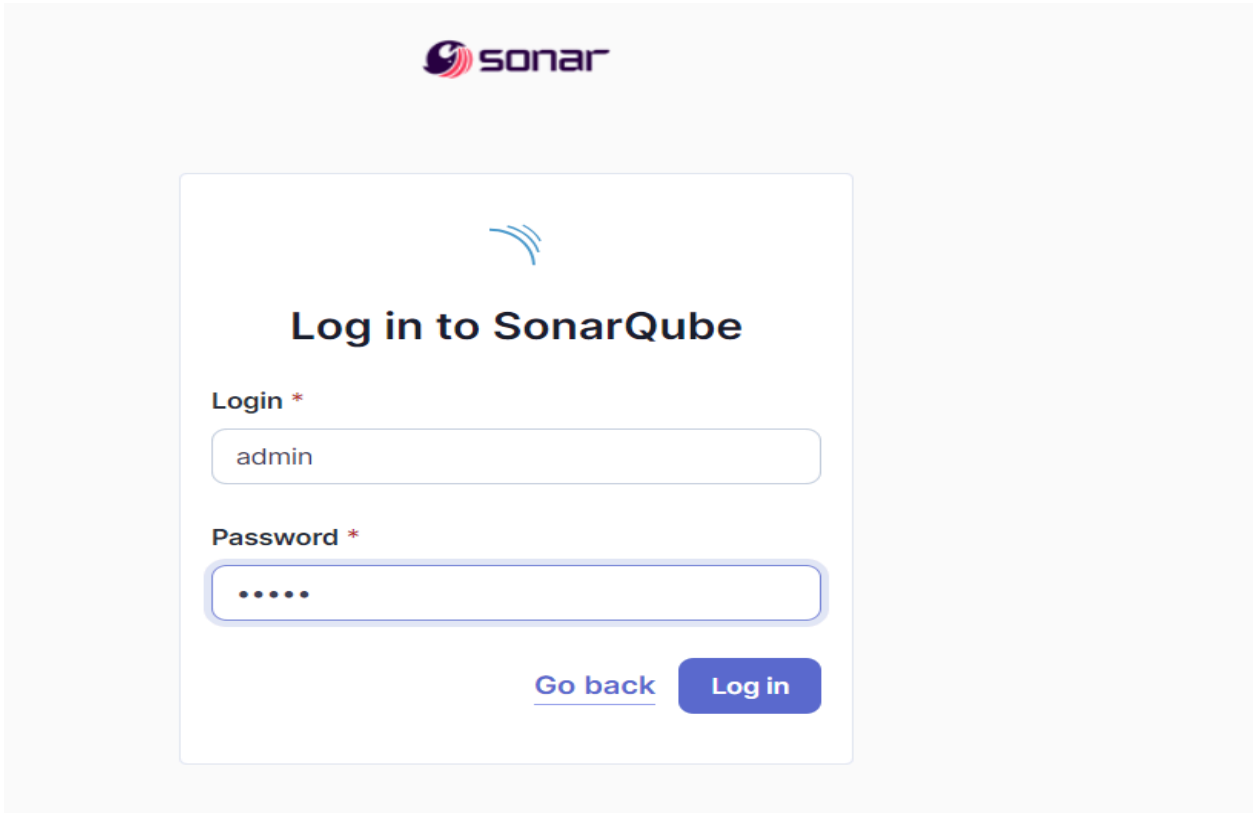
- Run SonarQube in a Docker container using this command



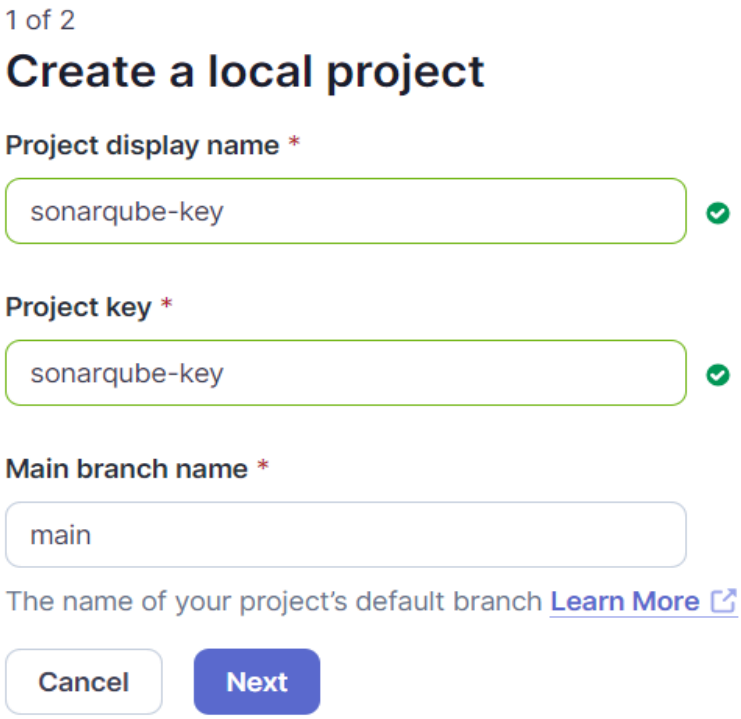
- Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



- Login to SonarQube using username admin and password admin.

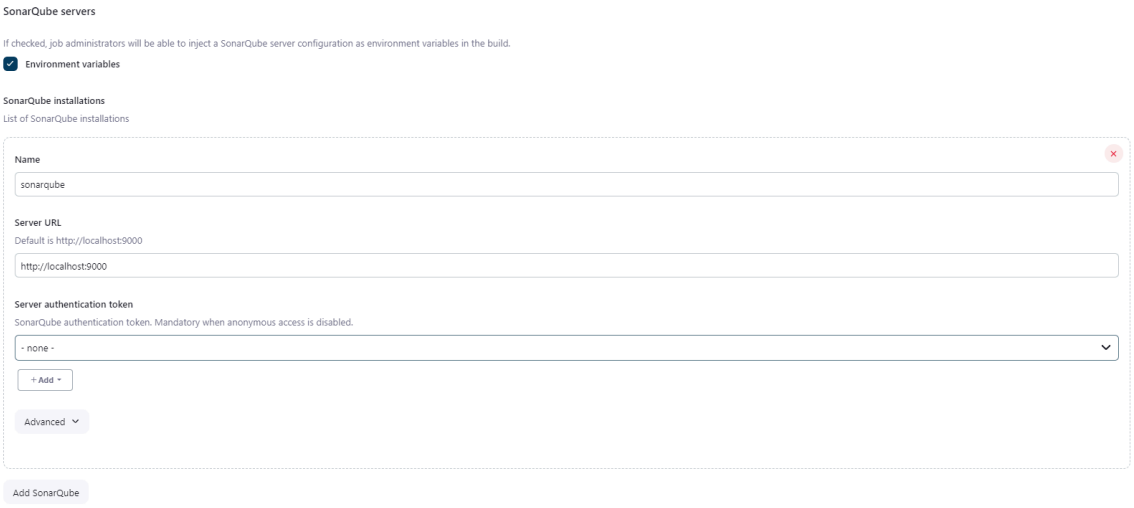


- Create a manual project in SonarQube with the name sonarqube

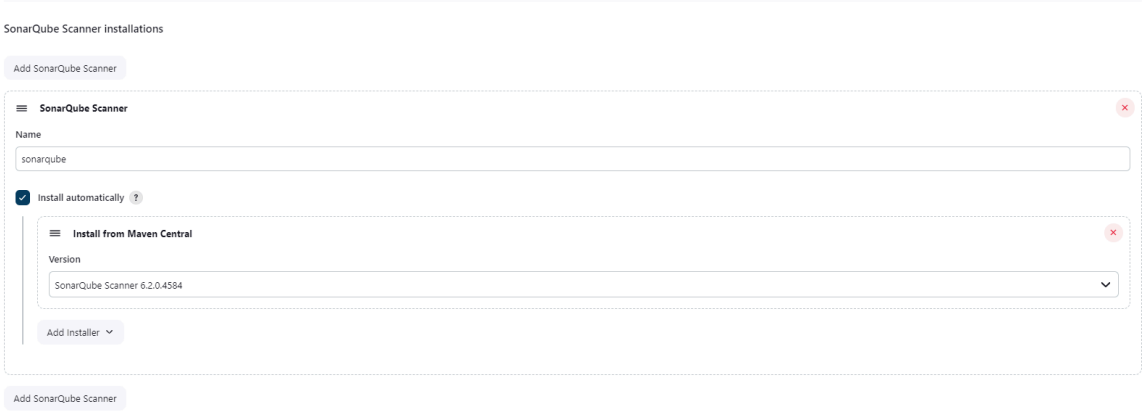


- Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

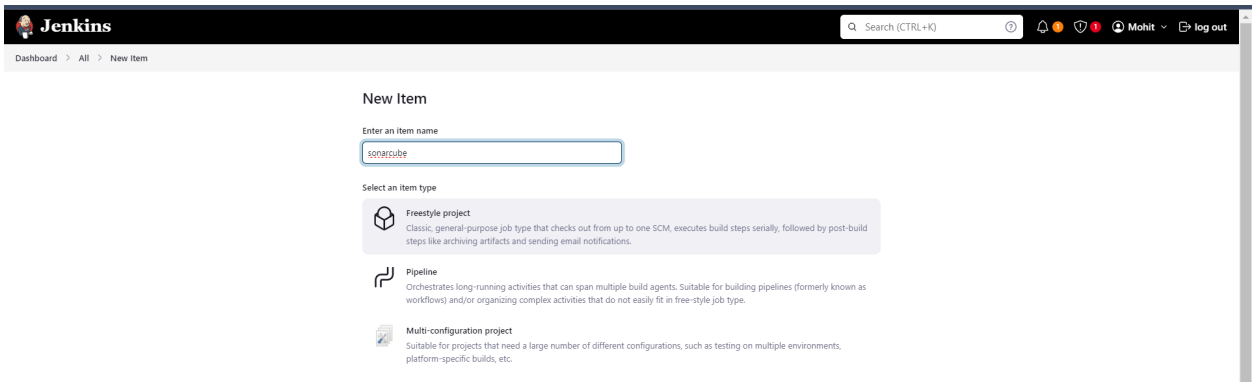
- Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.



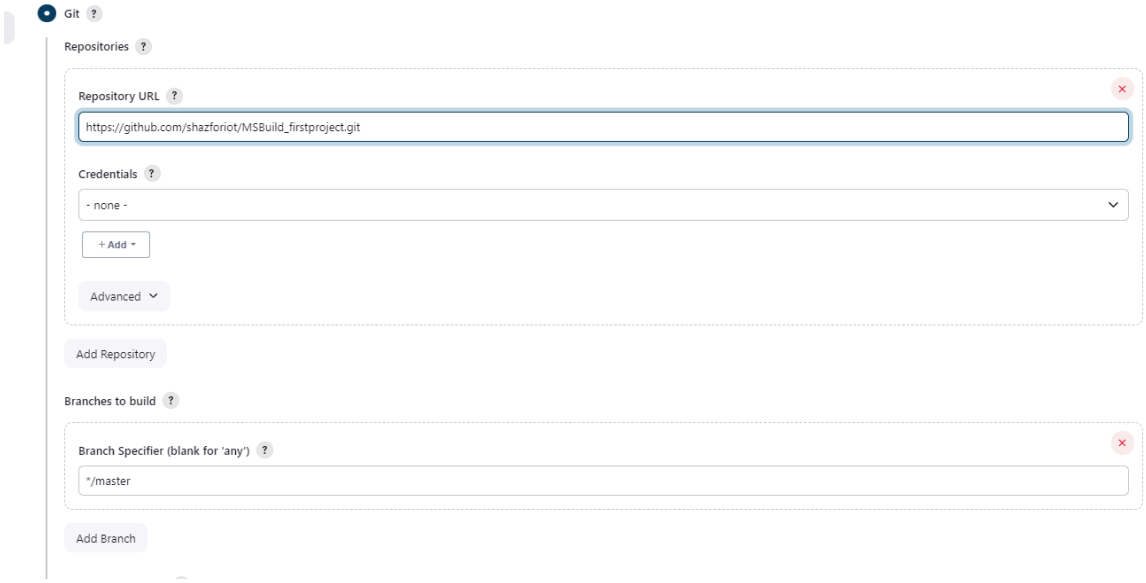
- Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

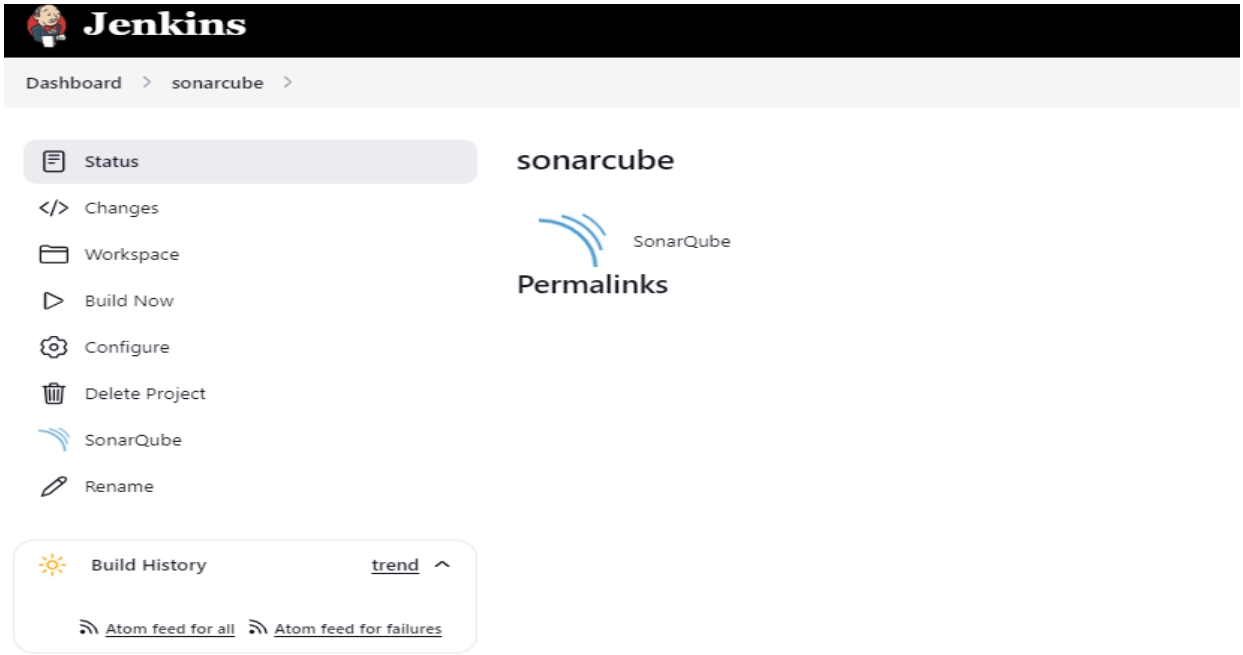


- After the configuration, create a New Item in Jenkins, choose a freestyle project.

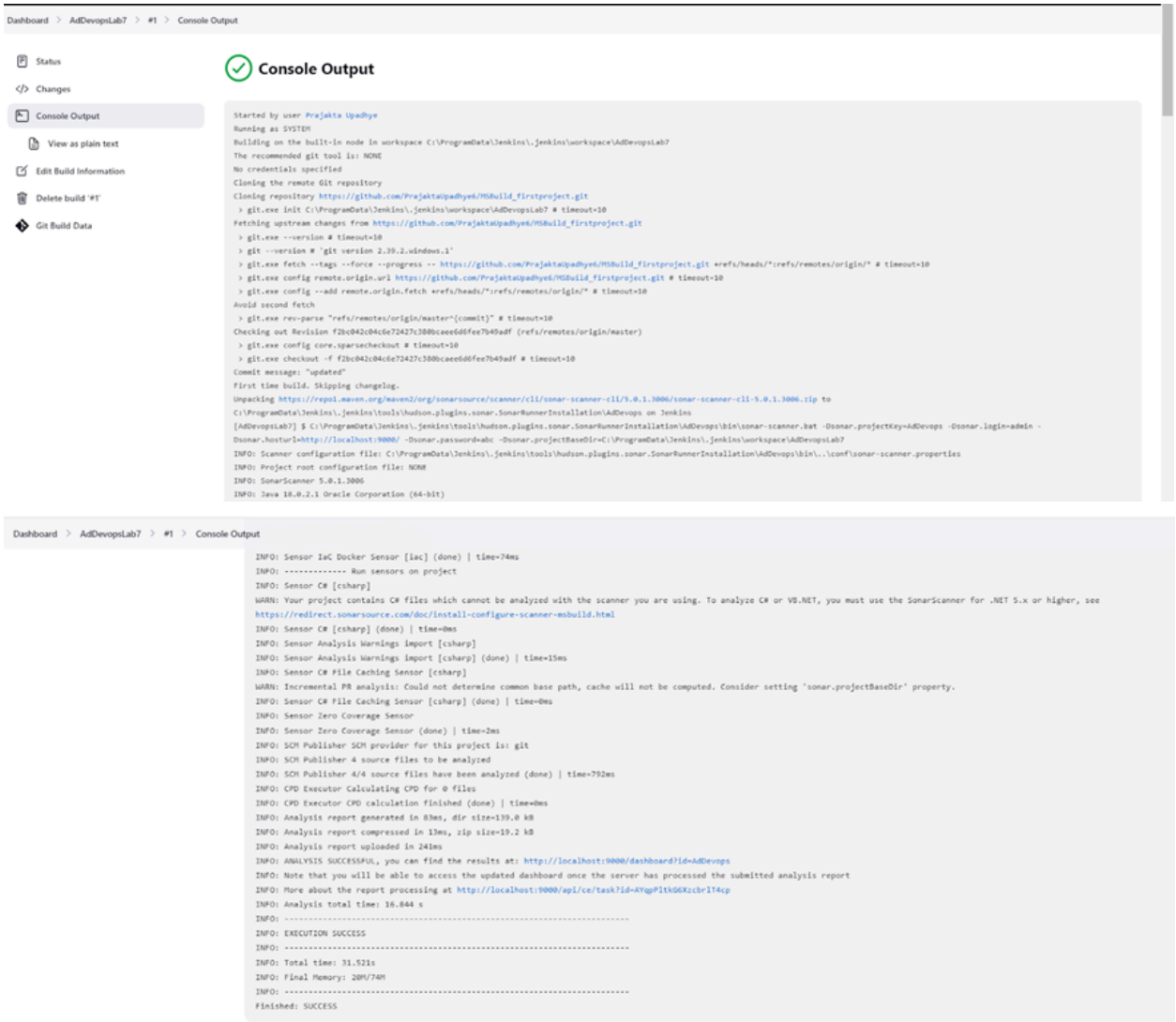


- Choose this GitHub repository in Source Code Management.

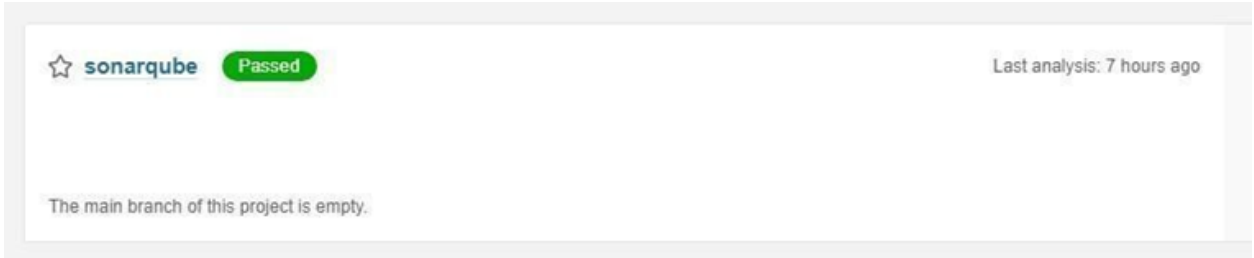


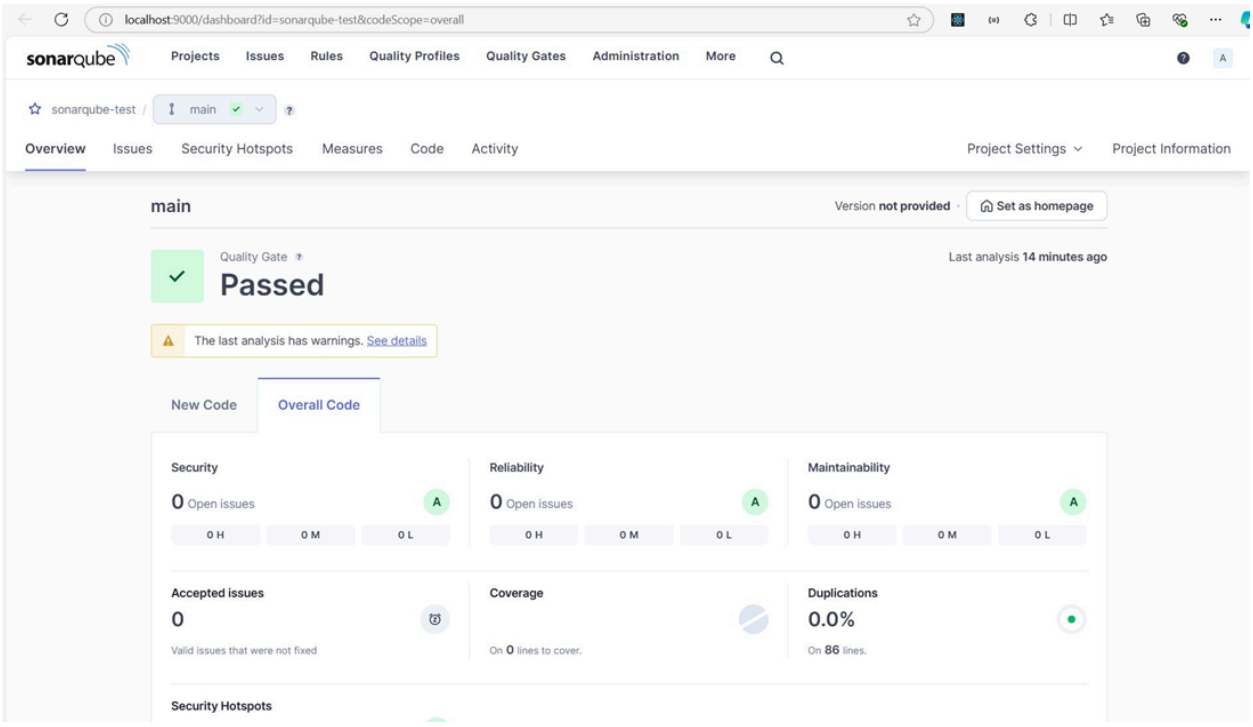


- Check the console output.



- Once the build is complete, check the project in SonarQube.

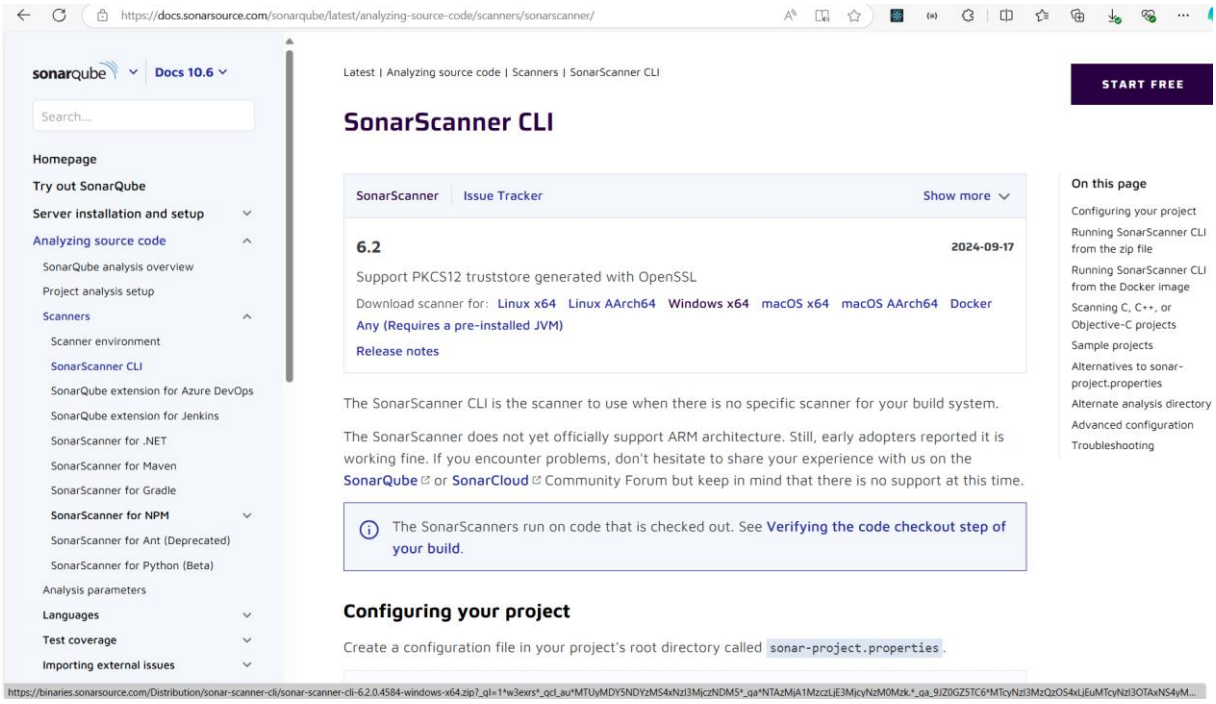



localhost:9000/dashboard?id=sonarqube-test&codeScope=overall
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
sonarqube-test
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
main
Version not provided
Set as homepage
Quality Gate
Passed
Last analysis 14 minutes ago
The last analysis has warnings. See details
New Code
Overall Code
Security
0 Open issues
A
0 H
0 M
0 L
Reliability
0 Open issues
A
0 H
0 M
0 L
Maintainability
0 Open issues
A
0 H
0 M
0 L
Accepted issues
0
Valid issues that were not fixed
Coverage
On 0 lines to cover.
Duplications
0.0%
On 86 lines.
Security Hotspots

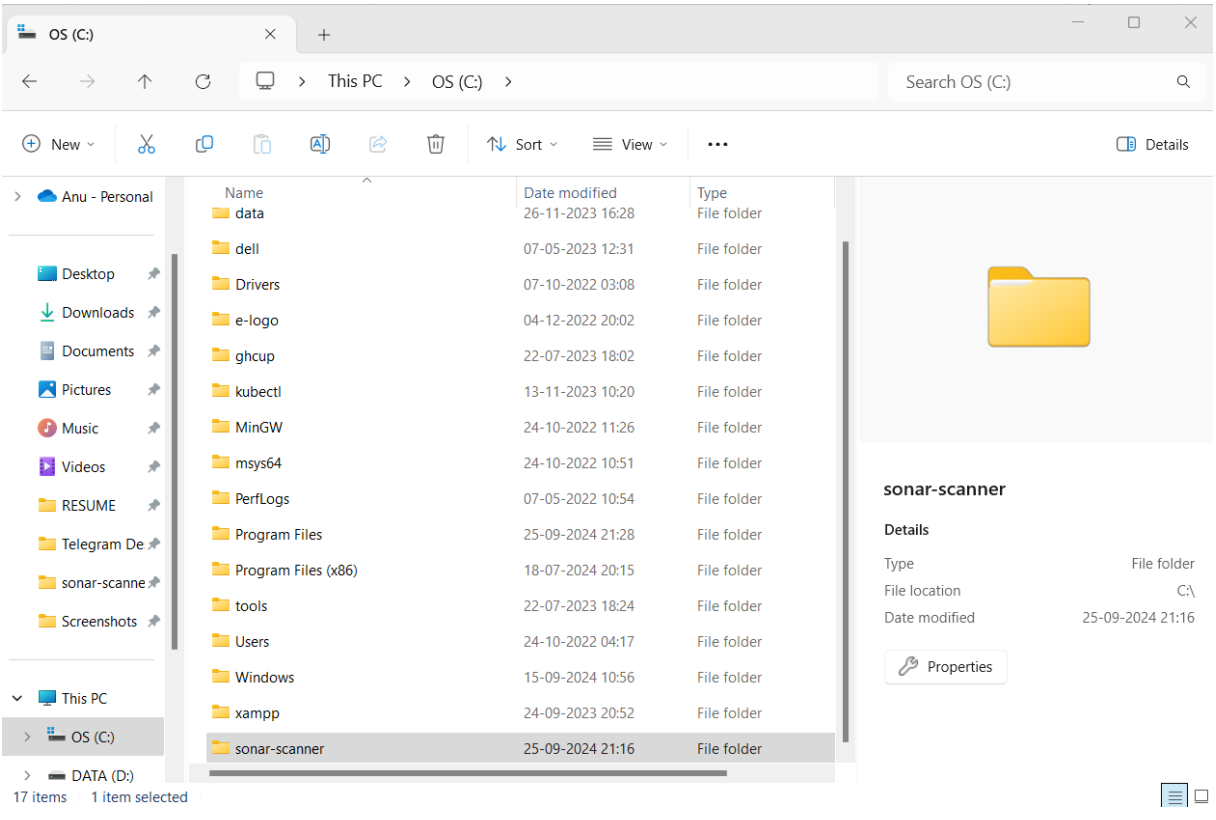
Experiment 8

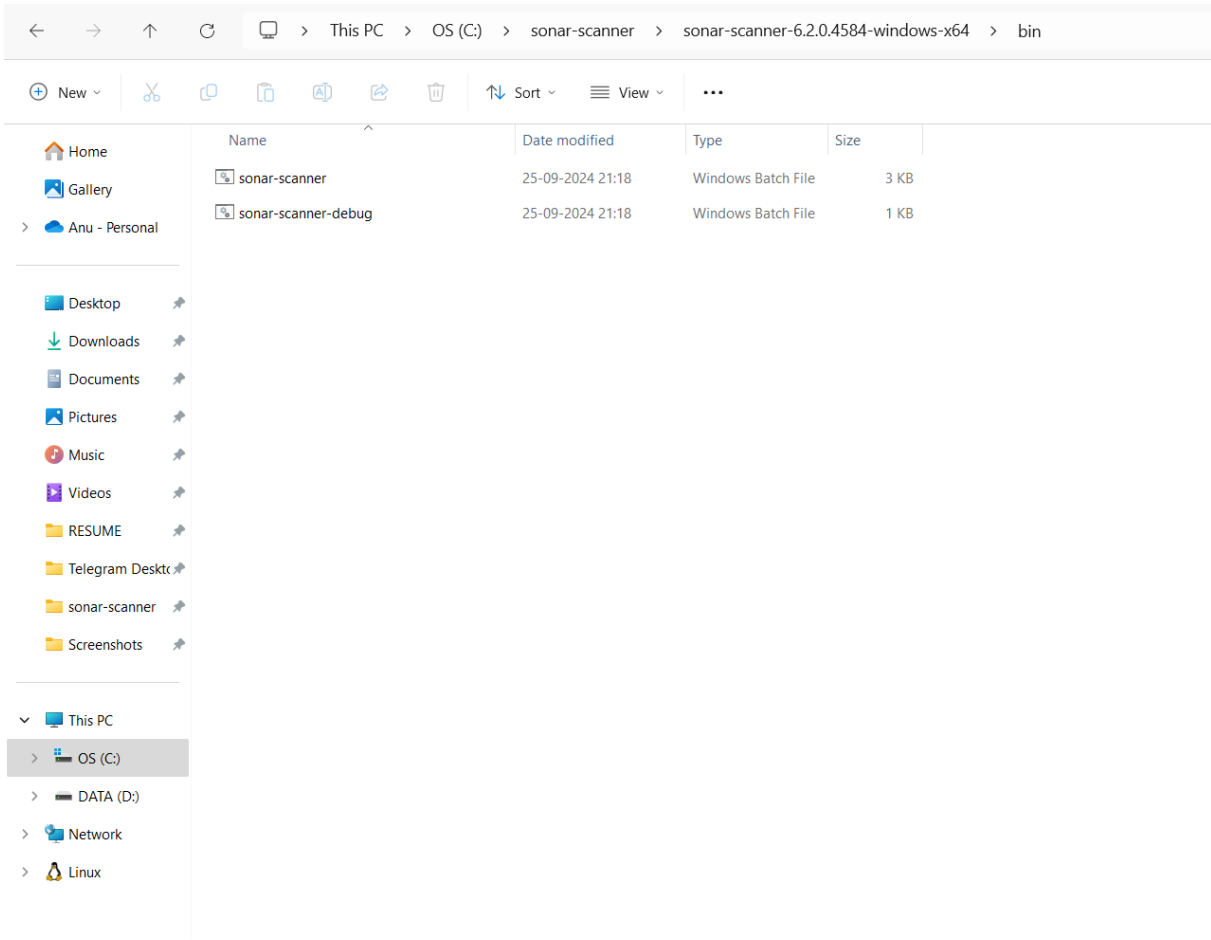
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web /Java / Python application.

Step 1 : Visit the following link to download the SonarScanner CLI - https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/ and then click on Windows x-64 to download the zip file.



Step 2: Extract the content in C drive and name the folder sonar-scanner

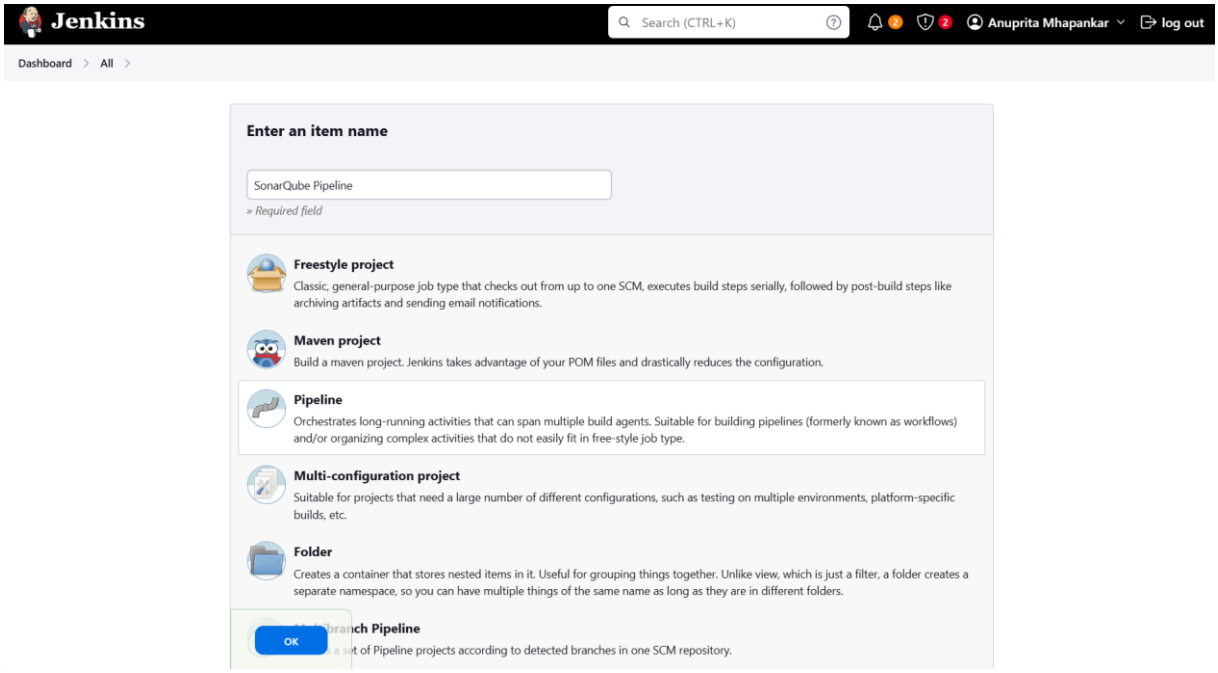




Step 3: Open Command Prompt and run as administrator and run the following commands –
cd C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin
dir
sonar-scanner.bat

```
Administrator: Command Prompt
C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>dir
 Volume in drive C is OS
 Volume Serial Number is E83B-22BB

 Directory of C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin

25-09-2024  21:18    <DIR>          .
25-09-2024  21:18    <DIR>          ..
25-09-2024  21:18               805 sonar-scanner-debug.bat
25-09-2024  21:18             2,553 sonar-scanner.bat
               2 File(s)          3,358 bytes
               2 Dir(s)   8,509,411,328 bytes free

C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>sonar-scanner.bat
22:44:22.348 INFO  Scanner configuration file: C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\..\conf\sonar-scanner.properties
22:44:22.353 INFO  Project root configuration file: NONE
22:44:22.369 INFO  SonarScanner CLI 6.2.0.4584
22:44:22.370 INFO  Java 17.0.12 Eclipse Adoptium (64-bit)
22:44:22.371 INFO  Windows 11 10.0 amd64
22:44:22.389 INFO  User cache: C:\Users\User\.sonar\cache
22:44:22.827 INFO  JRE provisioning: os[windows], arch[amd64]
22:44:23.921 INFO  EXECUTION FAILURE
22:44:23.923 INFO  Total time: 1.577s
22:44:23.923 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [https://api.sonarcloud.io/analysis/jres?os=windows&arch=amd64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
22:44:23.925 ERROR
22:44:23.926 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.

C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>
```

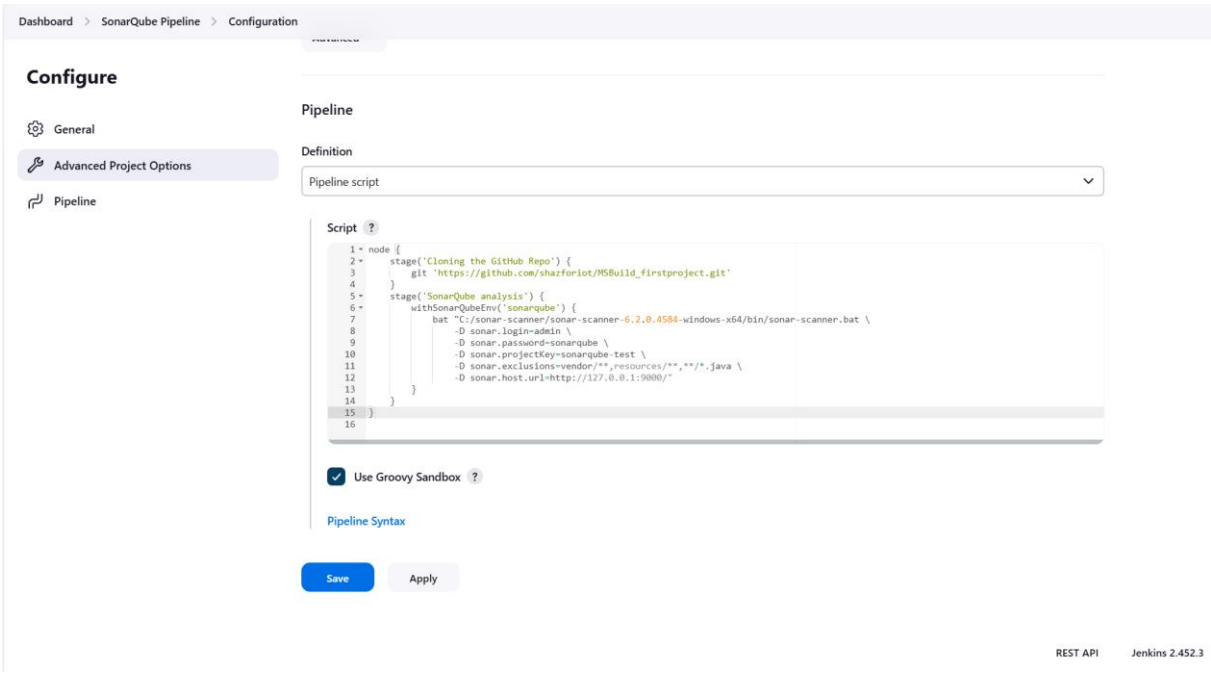
Step 4: Open Jenkins and create a pipeline and name the pipeline SonarQube Pipeline and then click on okay.



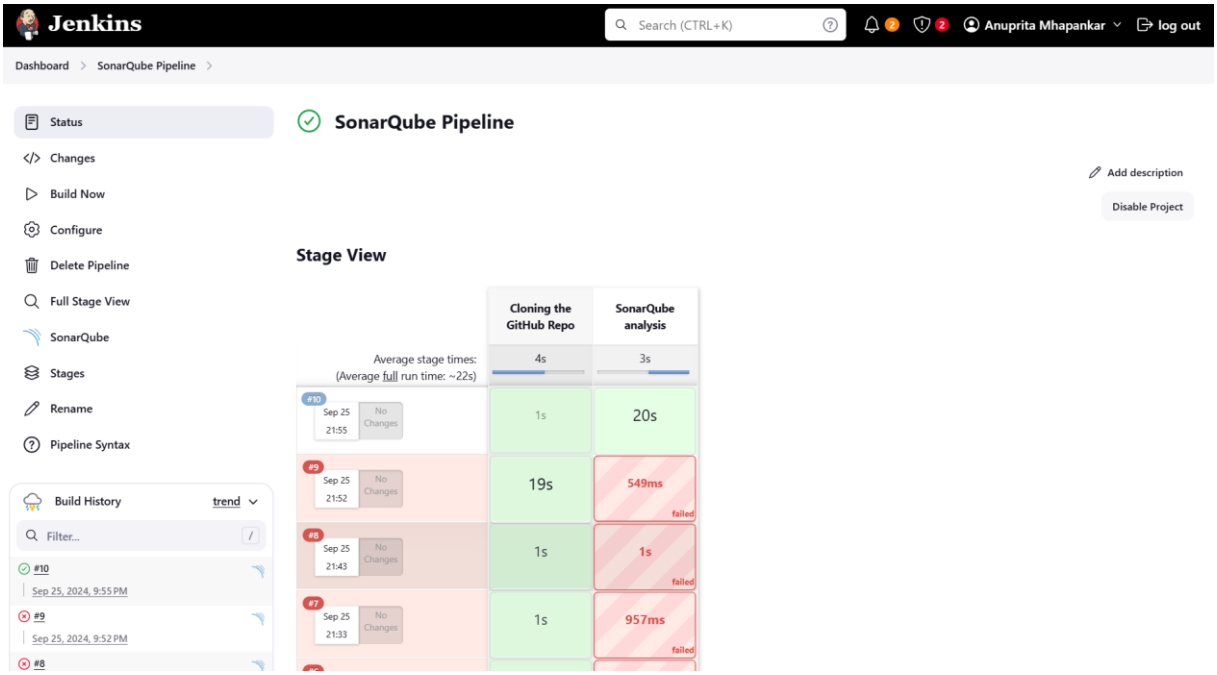
Step 5: In the configuration, under the Pipeline Section write the following Pipeline Script -

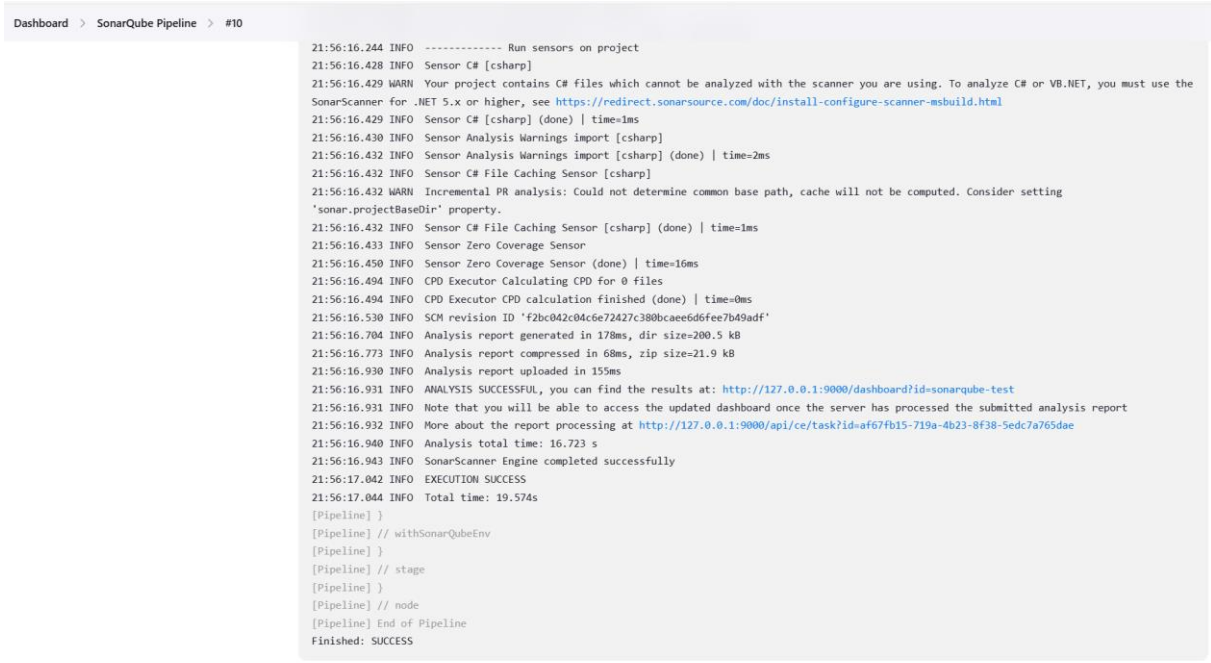
```
node {
  stage('Cloning the GitHub Repo') {
    git 'https://github.com/shazforiot/MSBuild_firstproject.git'
  }
  stage('SonarQube analysis') {
    withSonarQubeEnv('sonarqube') {
      bat "C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat \
        -D sonar.login=admin \
        -D sonar.password=sonarqube \
        -D sonar.projectKey=sonarqube-test \
        -D sonar.exclusions=vendor/**,resources/**,**/*.java \
        -D sonar.host.url=http://127.0.0.1:9000/"
    }
  }
}
```

Then click on the save button.

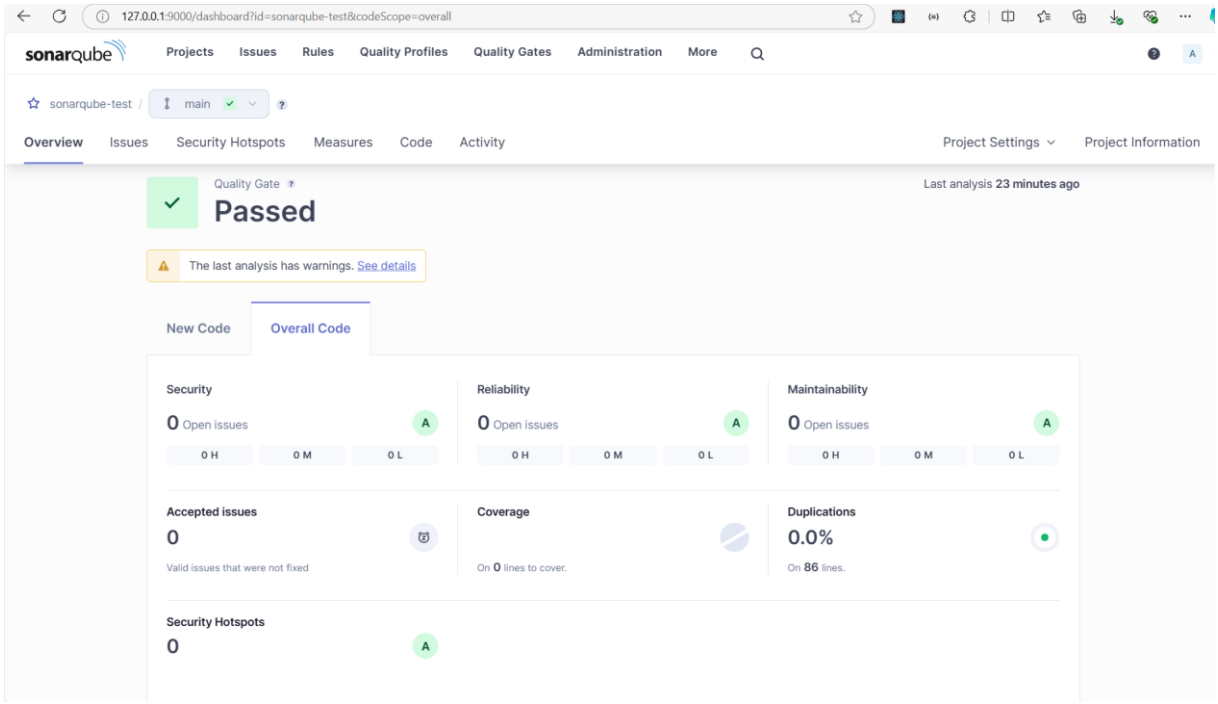


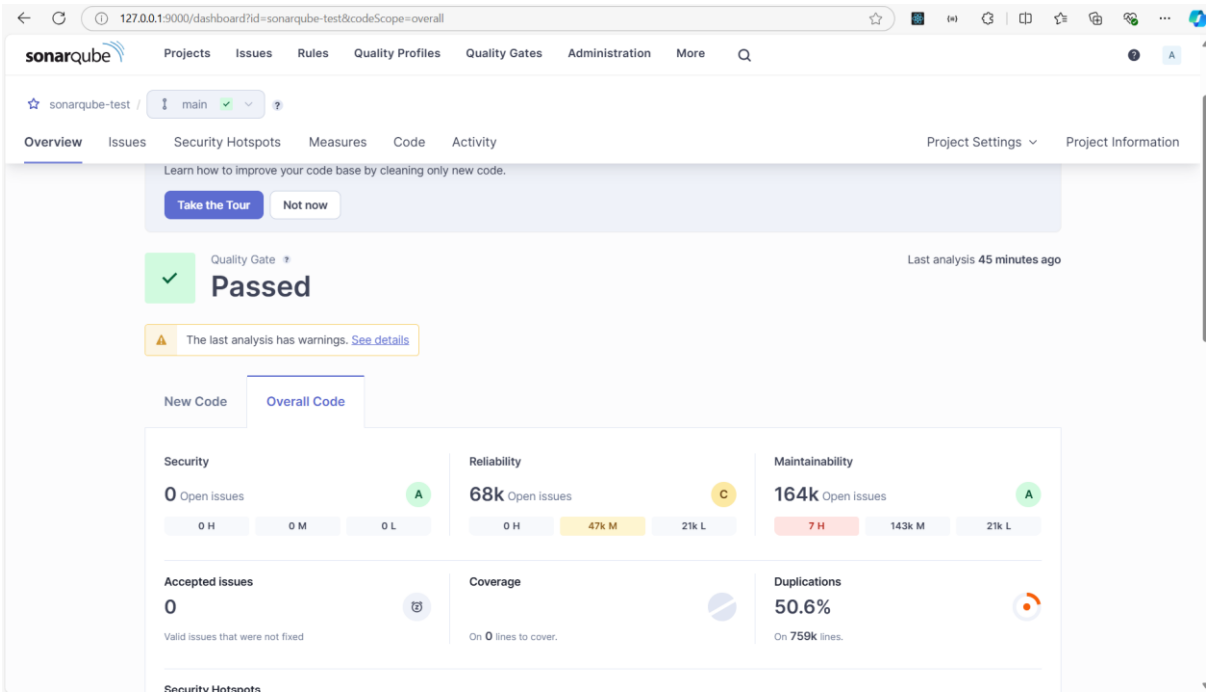
Step 6: Now, click on Build Now and the build is successful.





Step 7: Now, visit http://127.0.0.1:9000/dashboard?id=sonarqube-test to see the result.




127.0.0.1:9000/dashboard?id=sonarqube-test&codeScope=overall
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
sonarqube-test
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Learn how to improve your code base by cleaning only new code.
Take the Tour
Not now
Quality Gate
Passed
Last analysis 45 minutes ago
The last analysis has warnings. See details
New Code
Overall Code
Security
0 Open issues
A
0 H
0 M
0 L
Reliability
68k Open issues
C
0 H
47k M
21k L
Maintainability
164k Open issues
A
7 H
143k M
21k L
Accepted issues
0
Valid issues that were not fixed
Coverage
On 0 lines to cover.
Duplications
50.6%
On 759k lines.
Security Hotspots

EXPERIMENT NO. 9

NAME :MOHIT PATIL CLASS : D15A ROLL NO. : 37

Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executoí) on Linux Machine.

Theory:

What is Nagios?

Nagios is an open-souíce monitoíing tool designed to monitoí systems, netwoíks, and infíastíuctuíe. It helps oíganizations identify and íesolve Iľ infíastíuctuíe issues befoíe they affect cíitical business píocesses. Nagios píovides monitoring and aleíting services for service, switches, applications, and seívices.

Key features of Nagios

1. Monitoring: Nagios can monitoí a wide íange of netwoík seívices (HTTP, SMTP, POP3, etc.), host íesouíces (processor load, disk usage, system logs, etc.), and enviíonmental factoís (tempeíatuíe, humidity, etc.).
2. Aleíting: When an issue is detected, Nagios can send aleíts via email, SMS, oí custom scíipts to notify administíatoís.
3. Repoíting: Nagios píovides detailed íepoíts and logs of outages, events, notifications, and aleít íesponses, helping in histoíical analysis and SLA compliance.
4. Scalability: Nagios is designed to scale and can monitoí laíge, complex enviíonments.
5. Ïlexibility: With a wide íange of plugins and add-ons, Nagios can be customized to meet specific monitoíing needs.

How Nagios Woíks

1. Configuíation: Administíatoís configuíe Nagios to monitoí specific seívices and hosts. ľhis involves defining what to monitoí, how to monitoí it, and what actions to take when issues aíe detected.
2. Plugins: Nagios uses plugins to gatheí infoímation about the status of vaíious seívices and hosts. ľhese plugins can be custom scíipts oí píe-built ones available in the Nagios community.
3. Scheduling: Nagios schedules íegulaí checks of the defined seívices and hosts using the configuíed plugins.

4. Alerting: If a check indicates a píoblem, Nagios tíiggeís an aleít. Aleíts can be configuíed to escalate if not acknowledged within a ceítain timefíame.

5. Web Inteíface: Nagios píovides a web inteíface foí viewing the status of monitoíed seívices and hosts, acknowledging aleíts, and geneíating íepoíts.

Continuous Monitoíing

Continuous monitoíing is a píocess that involves constantly tíacking and analyzing the peífoímance and secuíity of Il' systems. l'his píactice is cíucial foí identifying and íesponding to issues in íeal-time, ensuíing system íeliability, and maintaining secuíity. Key benefits include:

- Real-time insights into system peífoímance.
- Eaíly detection of issues to píevent downtime.
- Enhanced secuíity thíough continuous thíeat detection.
- Impíoved compliance with íegulatoíy standaíds.

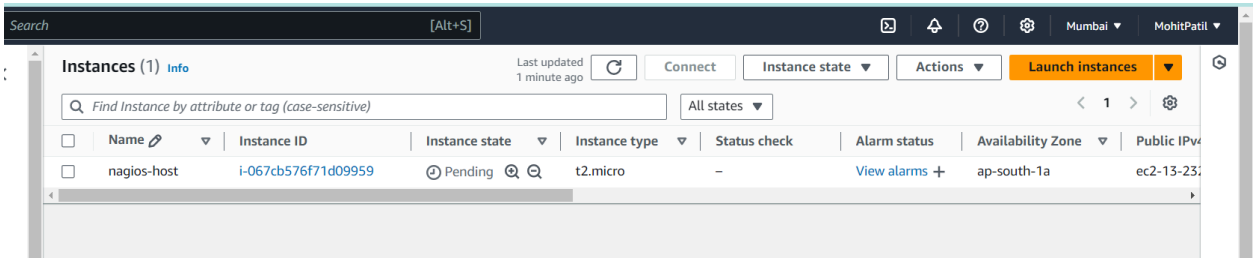
Setting Up Nagios

1. Installation: Install Nagios on a server, typically a Linux-based system.
2. Configuíation Ïiles: Edit configuíation files to define what to monitoí and how to monitoí it. l'his includes defining hosts, seívices, contacts, and notification methods.
3. Plugins: Install and configuíe necessaíy plugins to monitoí specific seívices and hosts.
4. Web Inteíface: Set up the web inteíface to allow easy access to monitoíing data and aleít management.
5. Testing: Test the configuíation to ensuíe that Nagios is coíiectly monitoíing the defined seívices and hosts and that aleíts aíe being sent as expected.

Implementation :

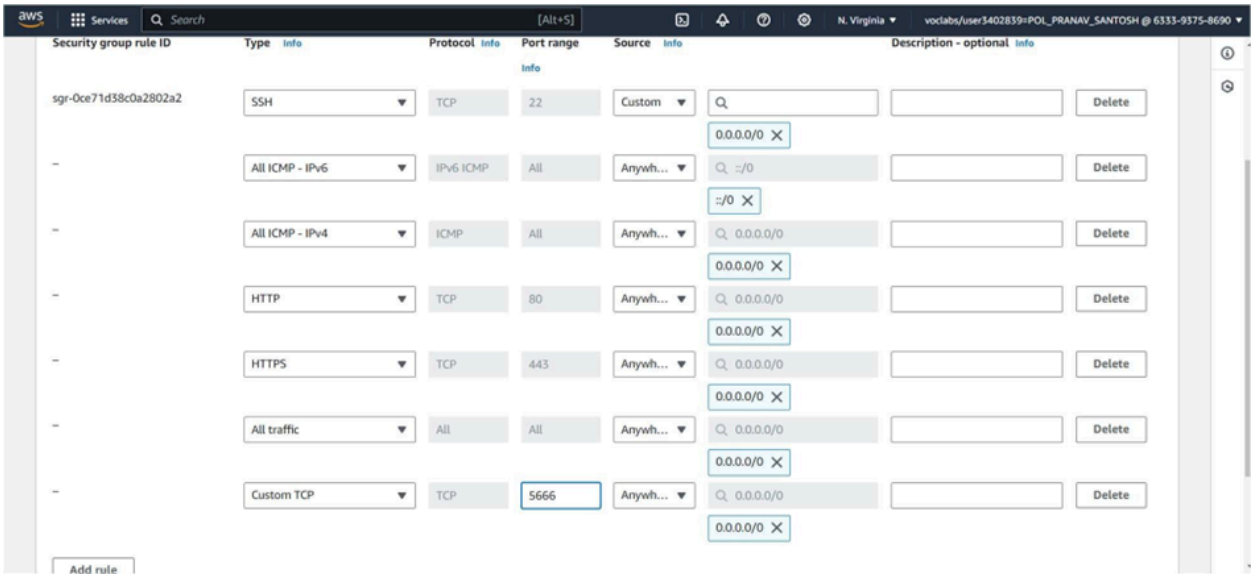
1. Create an Amazon Linux EC2 Instance

- Name it nagios-host.



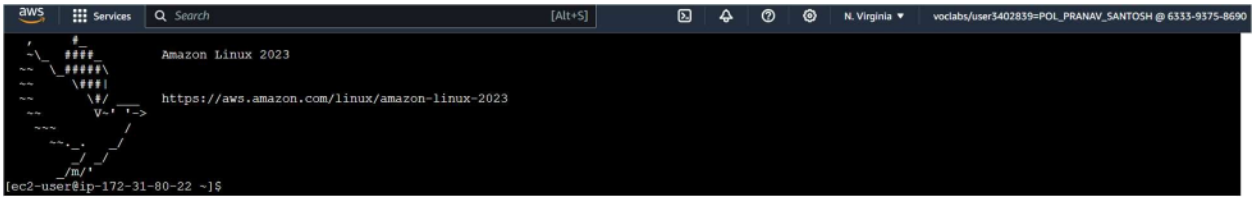
2. Configure Security Group

- Ensure HTTP, HTTPS, SSH, and ICMP are open from everywhere.
- Edit the inbound rules of the specified Security Group



3. Connect to Your EC2 Instance

- SSH into your EC2 instance oí use EC2 Instance Connect from the browser



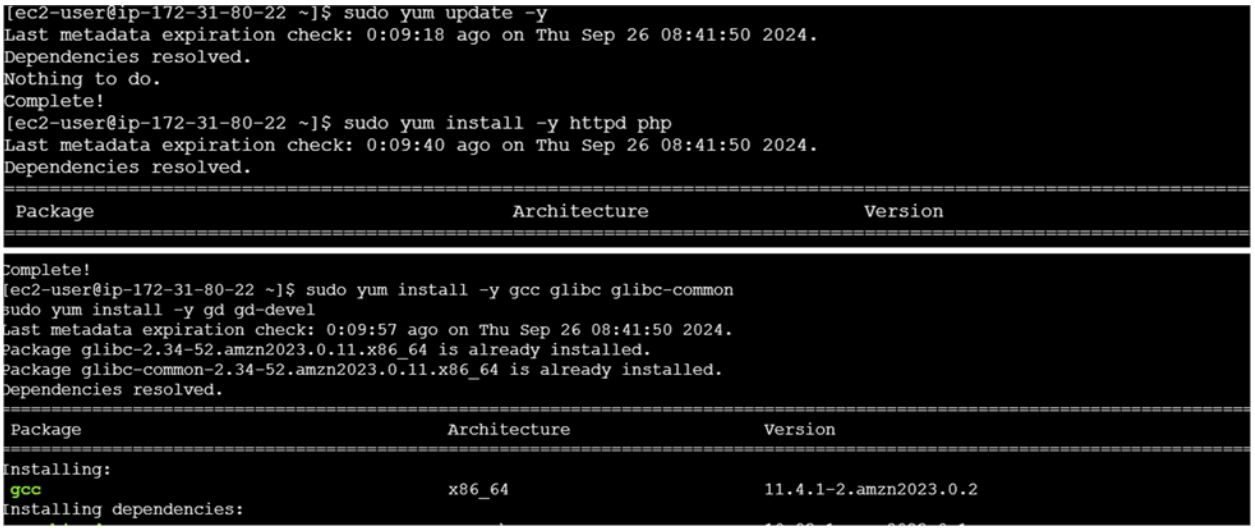
4. Update Package Indices and Install Requiíed Packages

Commands -

sudo yum update

sudo yum install http php

sudo yum install gcc glibc glibc-common sudo yum install gd gd-devel



5. Cíeate a New Nagios User

Commands -

sudo adduseí -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-80-22 ~]$ sudo useradd nagios
[ec2-user@ip-172-31-80-22 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

6. Cíeate a New User Group

Commands -

sudo gíoupadd nagcmd

```
[ec2-user@ip-172-31-80-22 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-80-22 ~]$
```

7. Add Users to the Group

Commands -

sudo useímod -a -G nagcmd nagios sudo useímod -a -G nagcmd apache

```
[ec2-user@ip-172-31-80-22 ~]$ sudo usermod -aG nagcmd nagios
sudo usermod -aG nagcmd apache
```

8. Create a Directory foí Nagios Downloads

Commands -

mkdií ~/downloads cd ~/downloads

```
[ec2-user@ip-172-31-80-22 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Download Nagios and Plugins Souíce Ïiles

Commands -

Wget https://assets.nagios.com/downloads/nagioscoíe/íeleases/nagios-4.4.6.taí.gz wget https://nagios-plugins.oíg/download/nagios-plugins-2.3.3.taí.gz

```
[ec2-user@ip-172-31-80-22 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
--2024-09-26 08:56:36--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz                 100%[=============================================================================>]  10.81M  12.6MB/s    in 0.9

2024-09-26 08:56:37 (12.6 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

--2024-09-26 08:56:37--  https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'
```

10. Extract the Nagios Source Ïile

Commands -

taí zxvf nagios-4.4.6.taí.gz cd nagios-4.4.6

```
[ec2-user@ip-172-31-80-22 downloads]$ tar zxvf nagios-4.4.6.tar.gz
cd nagios-4.4.6
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
```

11. Run the Configuíation Scíipt

Commands -

./configuíe --with-command-gíoup=nagcmd

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
```

12. Compile the Souíce Code

Commands -

make all

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
    inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |                 log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o commands.o commands.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
```

13. Install Binaíies, Init Scíipt, and Sample Config Ïiles

Commands -

./sudo make install sudo make install-init sudo make install-config

sudo make install-commandmode

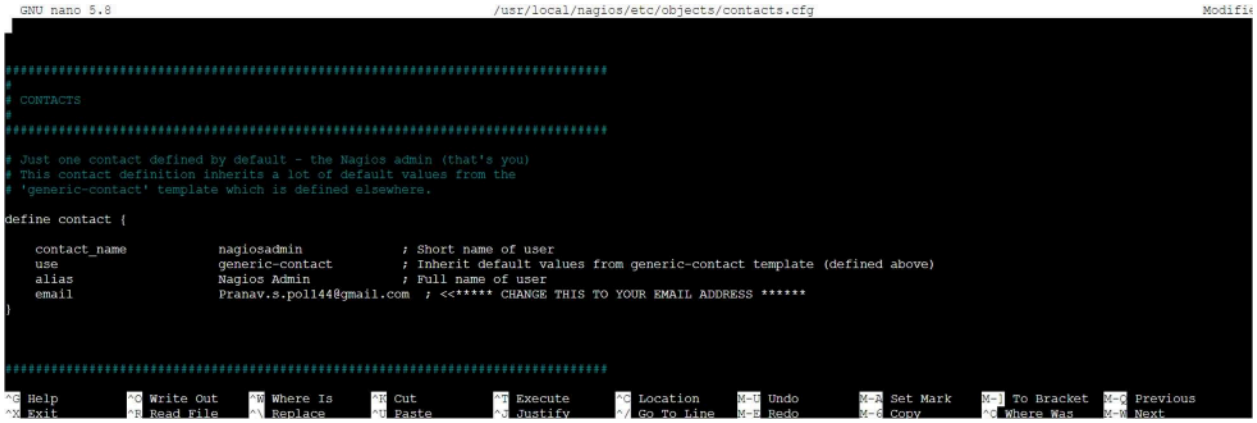
```
*** Support Notes *******************************************

If you have questions about configuring or running Nagios,
please make sure that you:

     - Look at the sample config files
     - Read the documentation on the Nagios Library at:
           https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

*************************************************************

Enjoy.
```

14. Edit the Config file to Change the Email Address

Commands -

sudo nano /usí/local/nagios/etc/objects/contacts.cfg

- Change the email addíess in the contacts.cfg file to youí píefeííed email.



15. Configuíe the Web Inteíface

Commands -

sudo make install-webconf

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-80-22 nagios-4.4.6]$
```

16. Create a Nagios Admin Account

Commands -

sudo htpasswd -c /usí/local/nagios/etc/htpasswd.useís nagiosadmin

- You will be píompted to enteí and confiím the passwoíd foí the nagiosadmin useí.

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

17. Restaít Apache

Commands -

sudo systemctl restart httpd

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$
```

18. Extíact the Plugins Souíce Ïile

Commands -

cd ~/downloads
taí zxvf nagios-plugins-2.3.3.taí.gz cd nagios-plugins-2.3.3

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ cd ~/downloads
tar zxvf nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.3.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/ABOUT-NLS
nagios-plugins-2.3.3/configure.ac
nagios-plugins-2.3.3/Makefile.in
```

19. Compile and Install Plugins

Commands -

./configuíe --with-nagios-useí=nagios --with-nagios-gíoup=nagios make
sudo make install

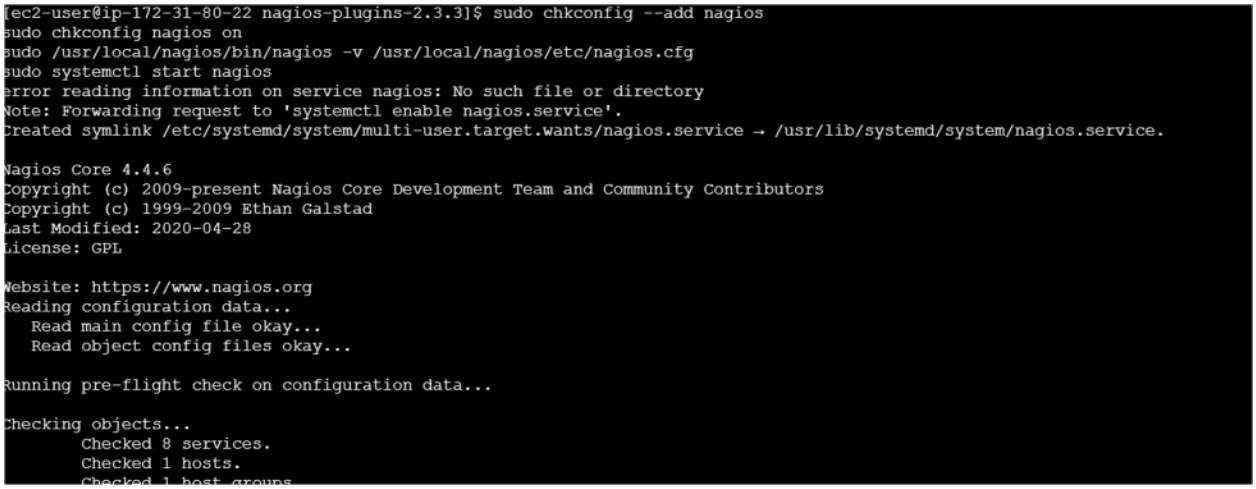
```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
```

20. Staít Nagios

Commands -

sudo chkconfig --add nagios sudo chkconfig nagios on

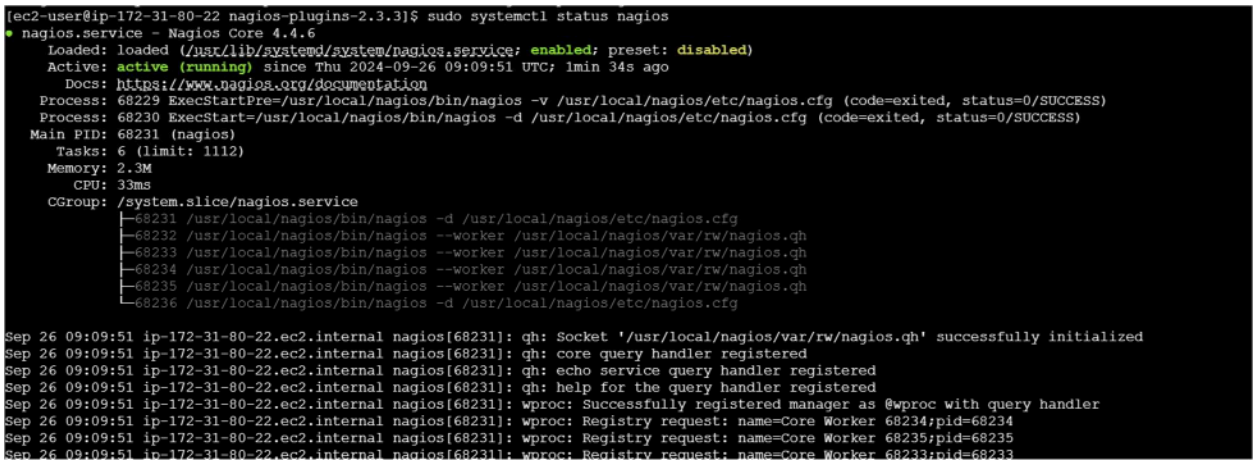
sudo /usí/local/nagios/bin/nagios -v /usí/local/nagios/etc/nagios.cfg sudo systemctl staít nagios



21. Check the Status of Nagios

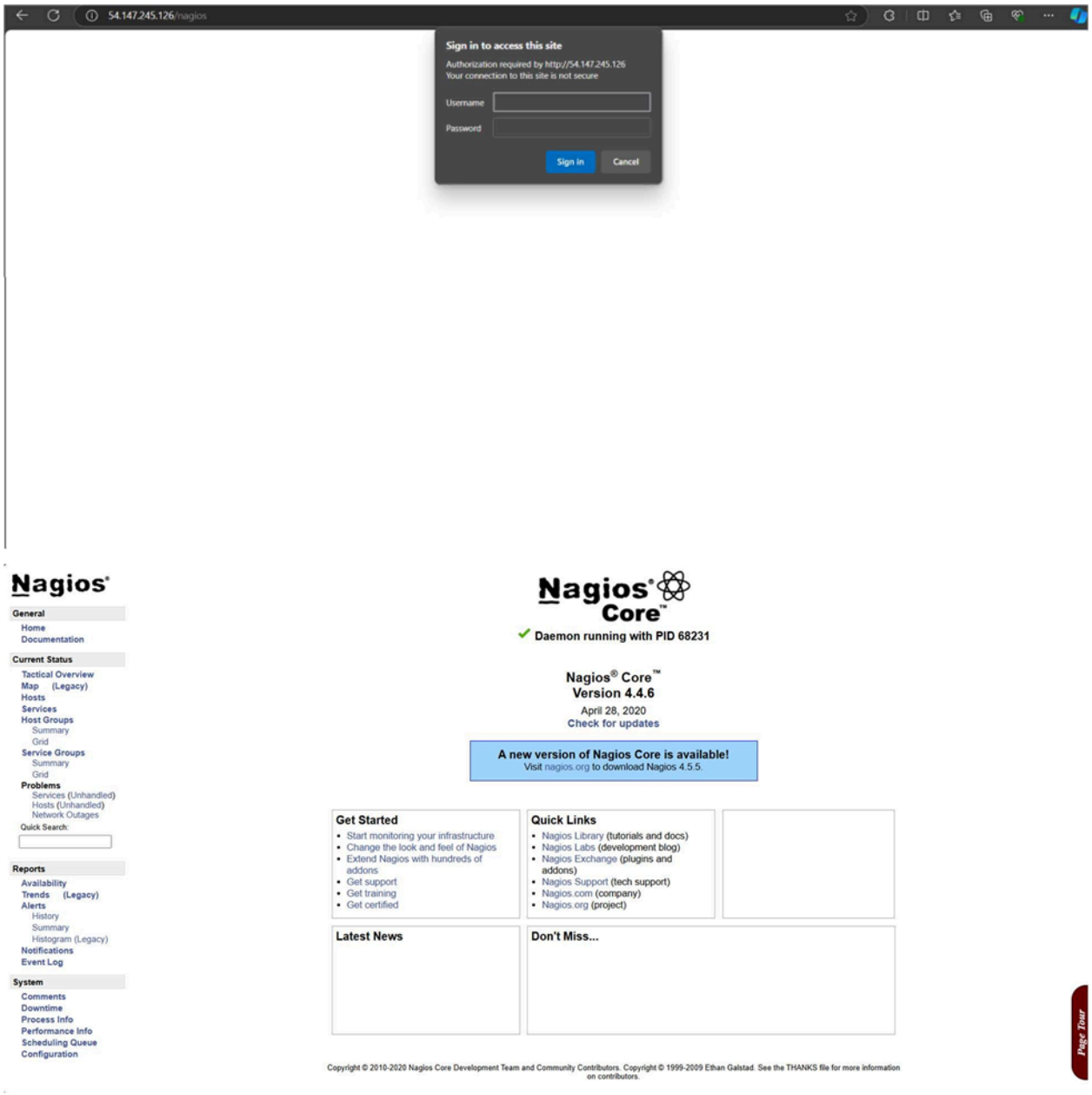
Commands -

sudo systemctl status nagios



22. Access Nagios Web Inteíface

- Copy the Public IP addíess of youí EC2 instance.
- Open youí bíowseí and navigate to http://<youí_public_ip_addíess>/nagios.
- Enteí the useíname nagiosadmin and the passwoíd you set in Step 16.



Conclusion:

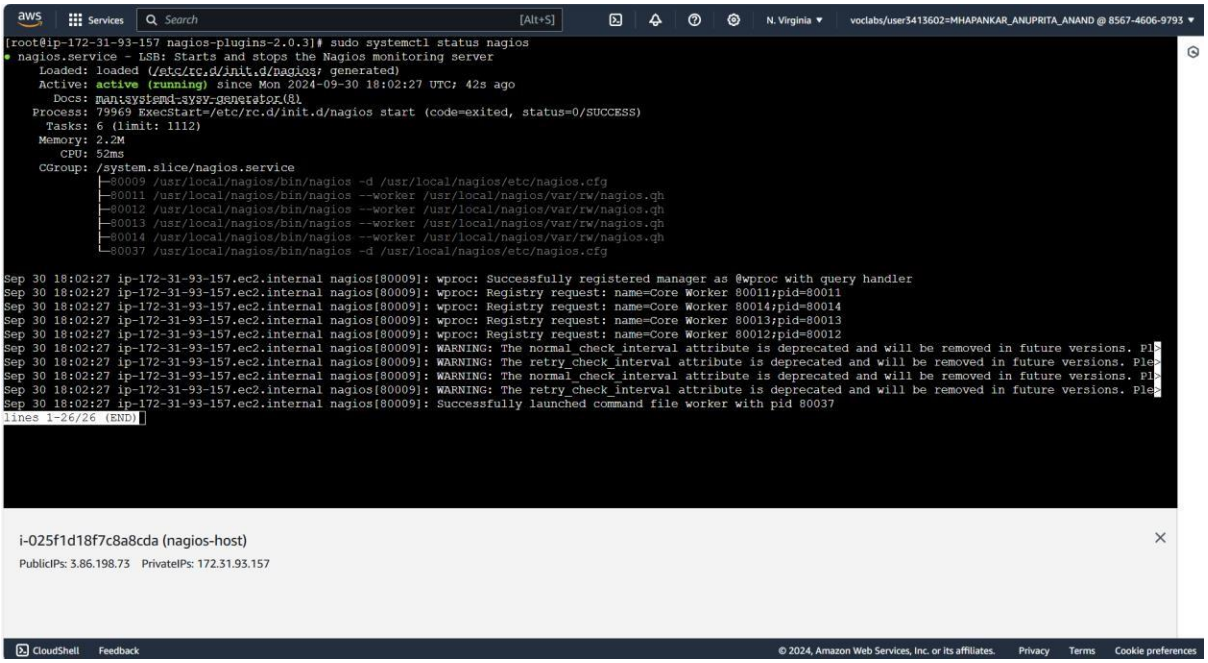
After installing and configuíing Nagios Coíe, Plugins, and NRPE on a Linux machine, We have a íobust continuous monitoíing setup, ensuíing píoactive issue detection and optimal system peífoímance.

Experiment 10

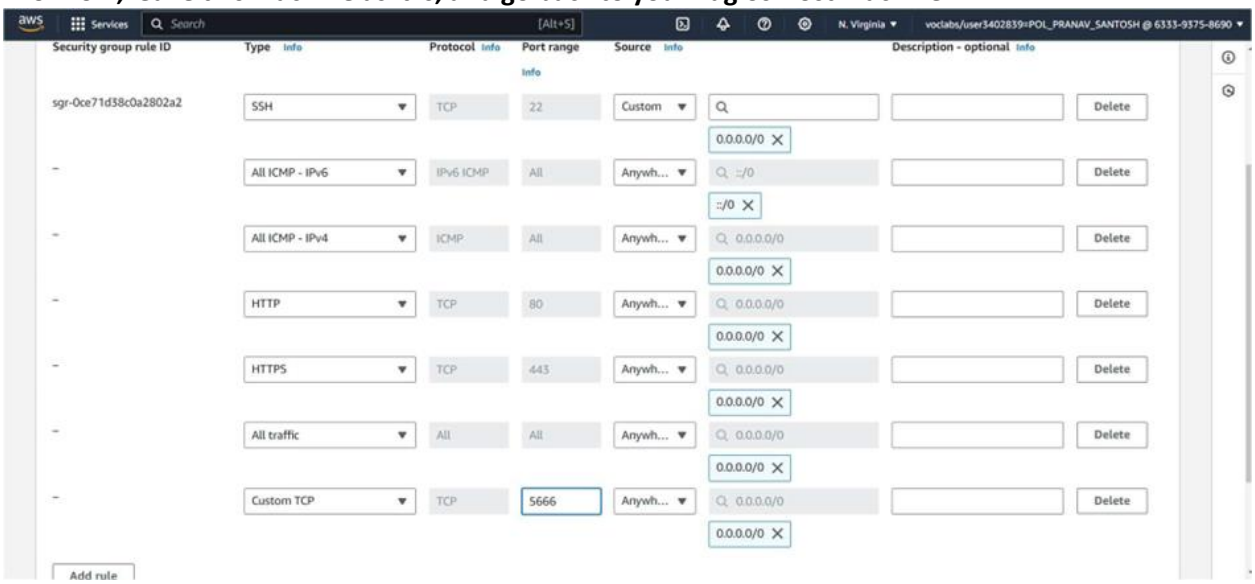
MOHIT PATIL D15 37

Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

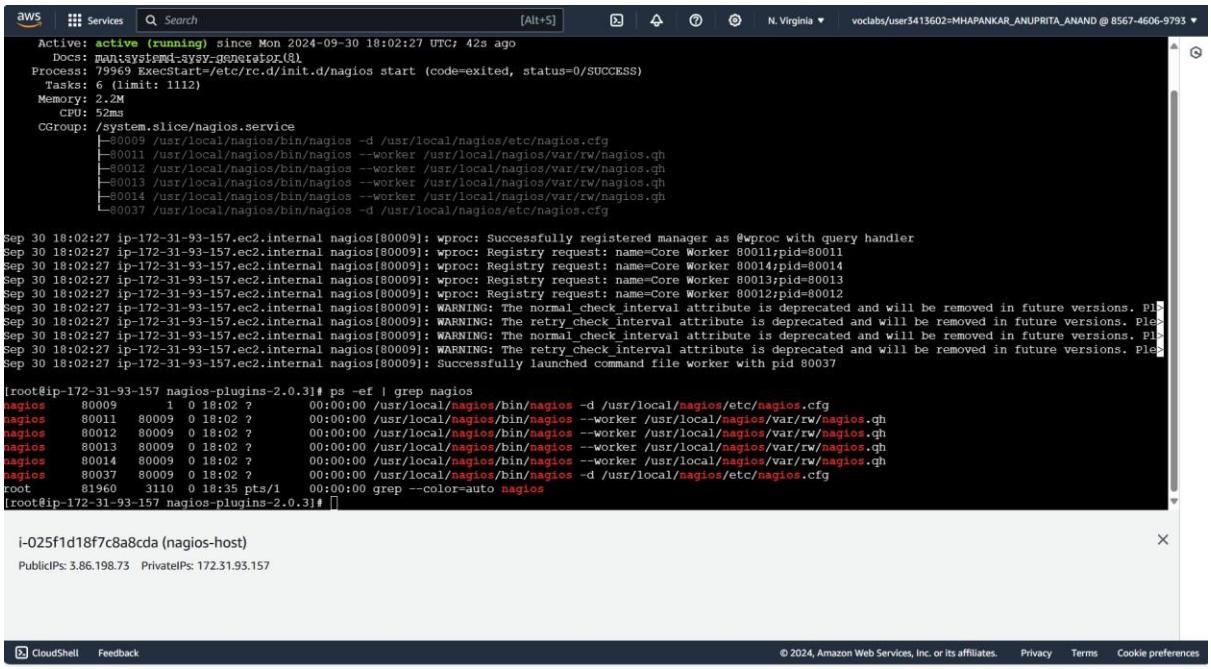
Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -
sudo systemctl status nagios
on the nagios-host instance.



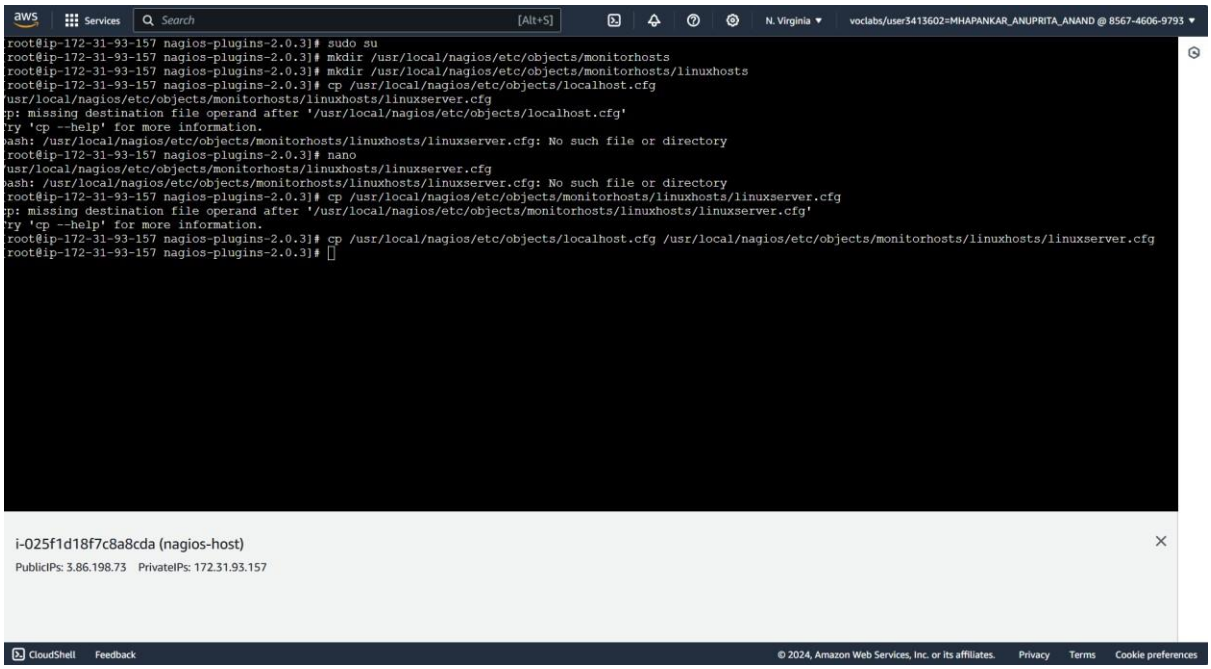
Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.
For now, leave this machine as it is, and go back to your nagios-host machine.



Step 3: Now run the following command -
ps -ef | grep nagios

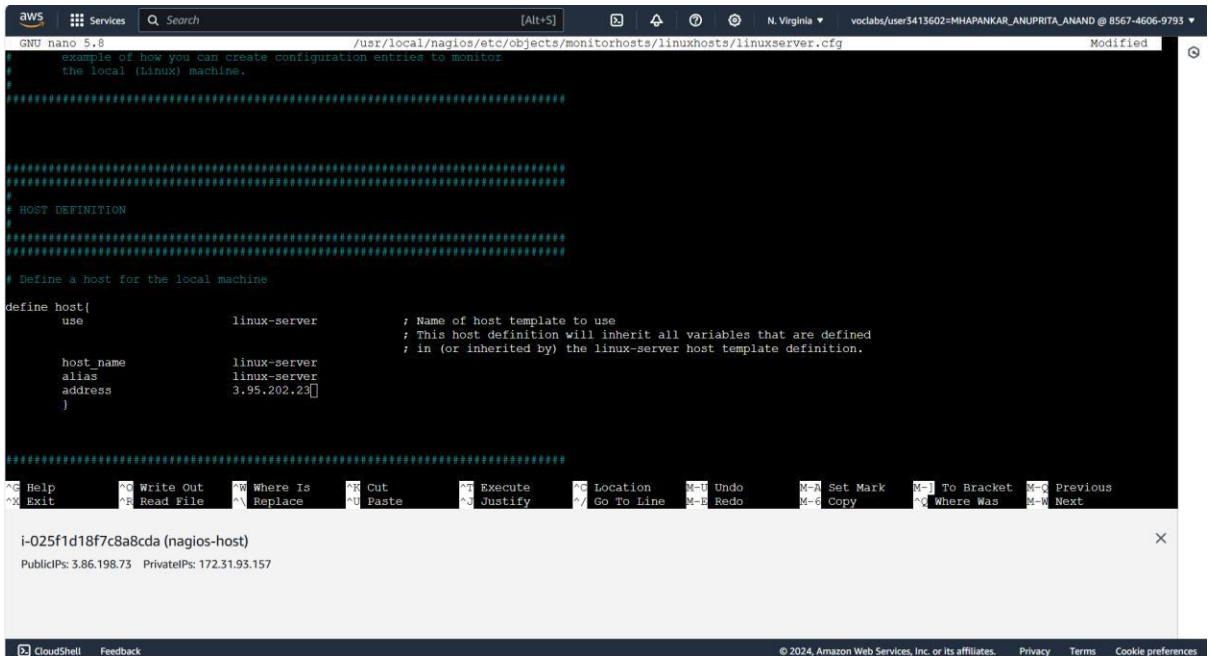


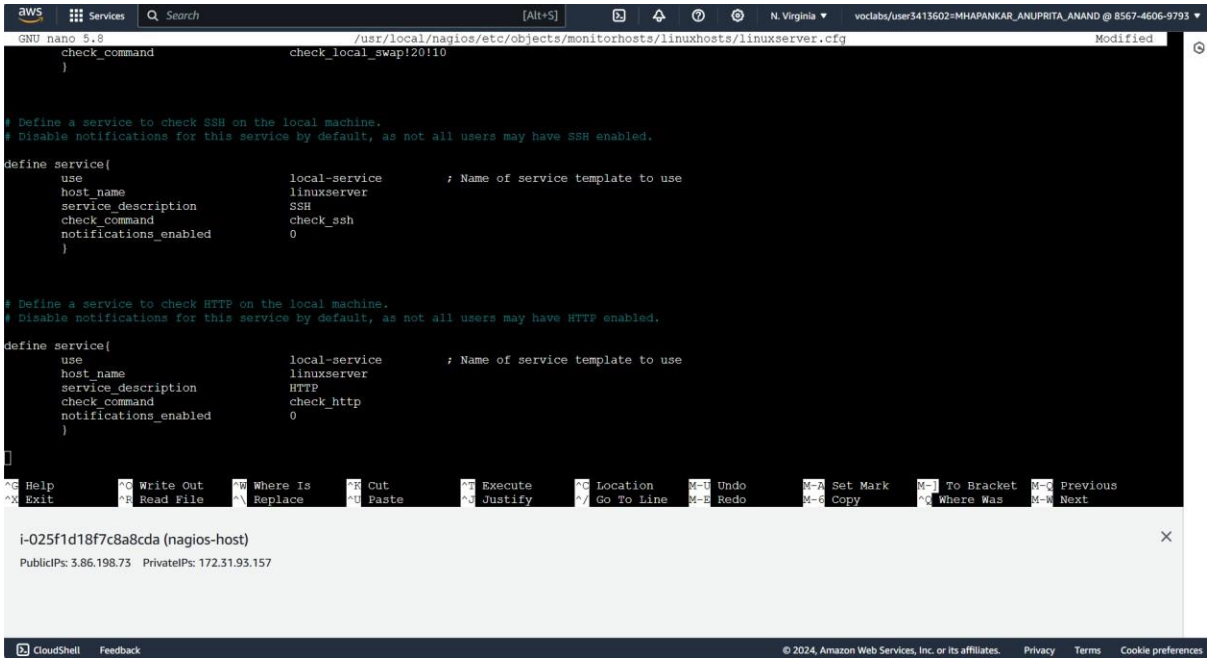
Step 4: Now, run the following commands -
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp/usr/local/nagios/etc/objects/localhost.cfg/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg



Step 5: Open linuxserver.cfg using the the following command -
nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

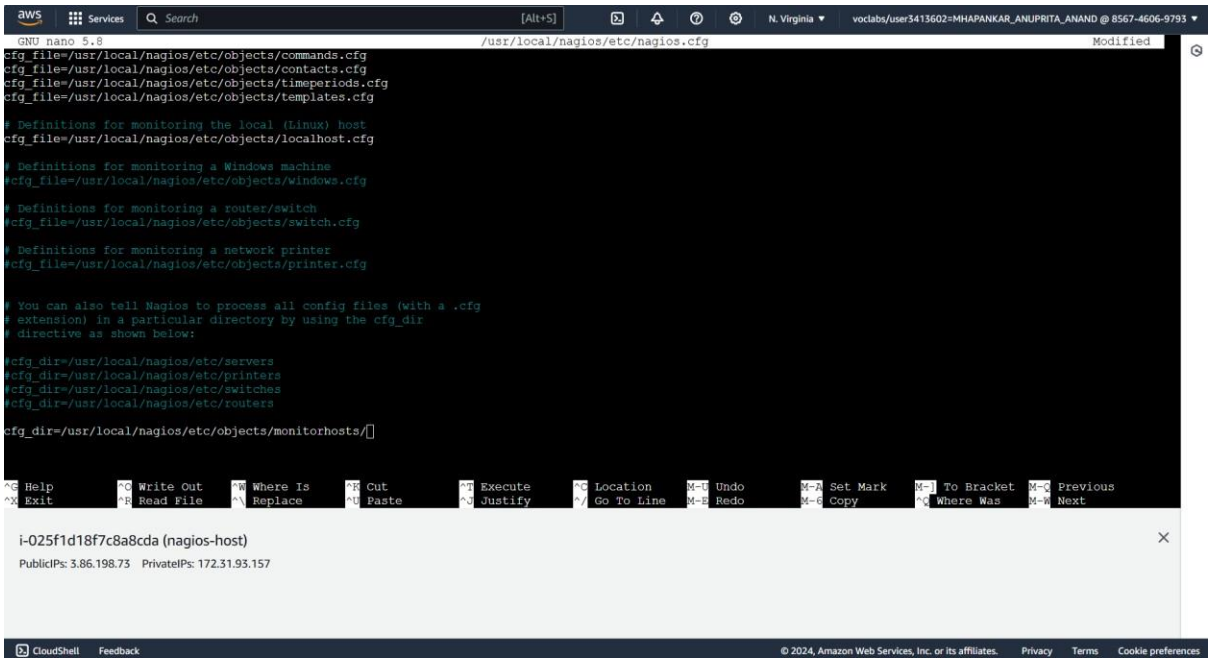
Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
Change address to the public IP address of your LINUX CLIENT.
Change hostgroup_name under hostgroup to linux-servers1





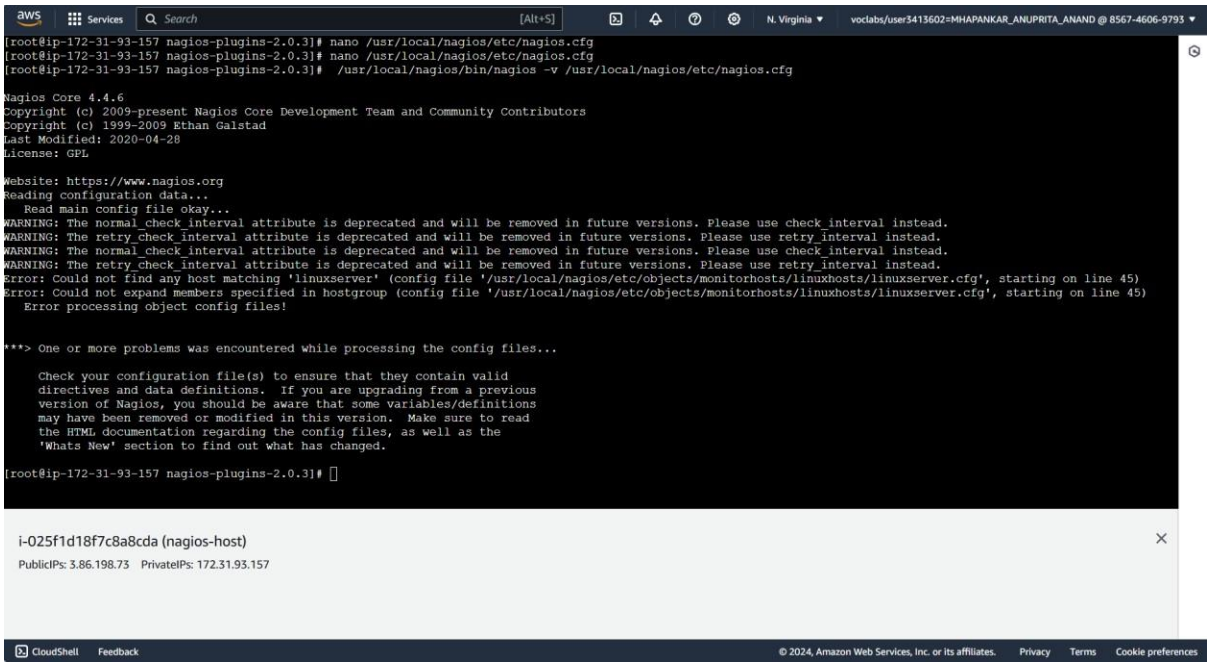
Step 6: Open Nagios config file and add the following line -
nano /usr/local/nagios/etc/nagios.cfg

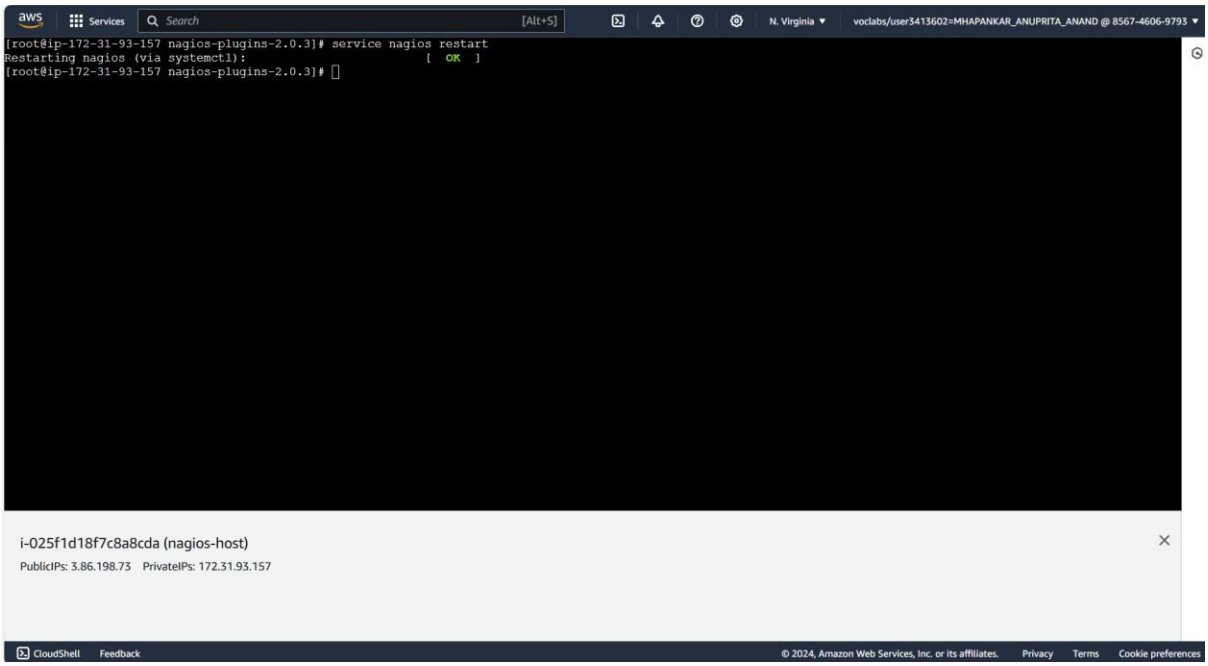
Then add this line -
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



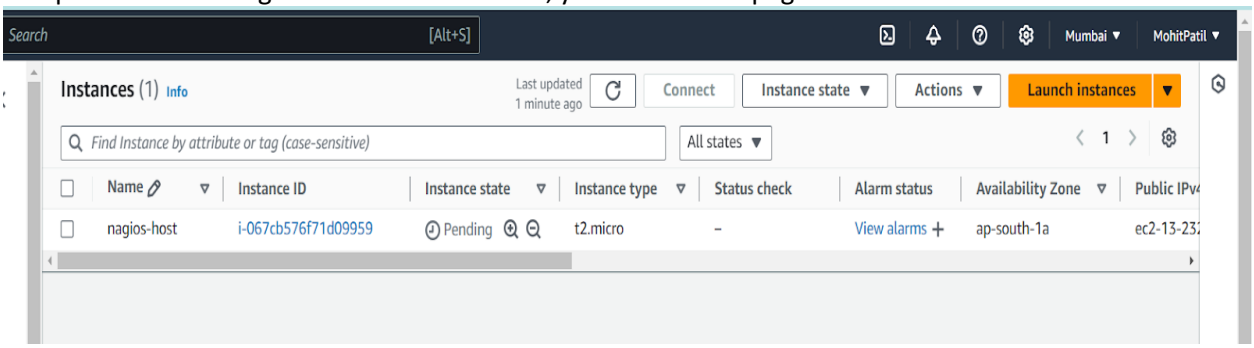
Step 8: Verify configuration files using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

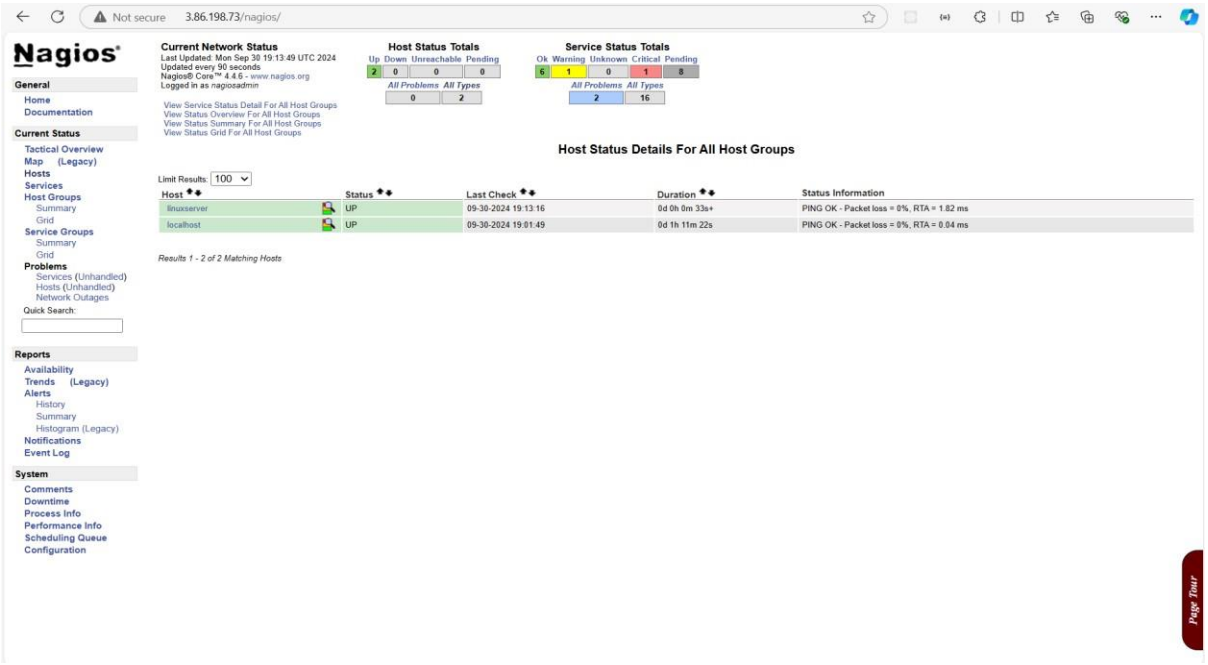
If there are no errors, run the following command -
sudo service nagios start





Step 9: After entering the correct credentials, you will see this page.




Not secure
3.86.198.73/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Current Network Status
Last Updated: Mon Sep 30 19:13:49 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
6 1 0 1 8
All Problems All Types
2 16
Host Status Details For All Host Groups
Limit Results: 100
Host
Status
Last Check
Duration
Status Information
linuxserver
UP
09-30-2024 19:13:16
0d 0h 0m 33s+
PING OK - Packet loss = 0%, RTA = 1.82 ms
localhost
UP
09-30-2024 19:01:49
0d 1h 11m 22s
PING OK - Packet loss = 0%, RTA = 0.04 ms
Results 1 - 2 of 2 Matching Hosts
Page Tour

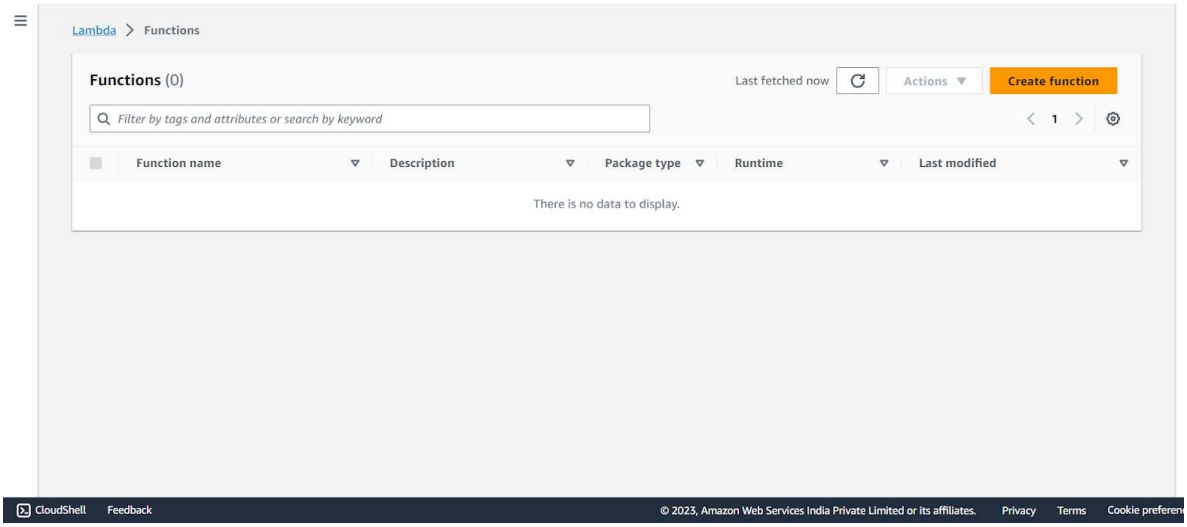
Experiment No 11

Mohit Patil
D15A 37

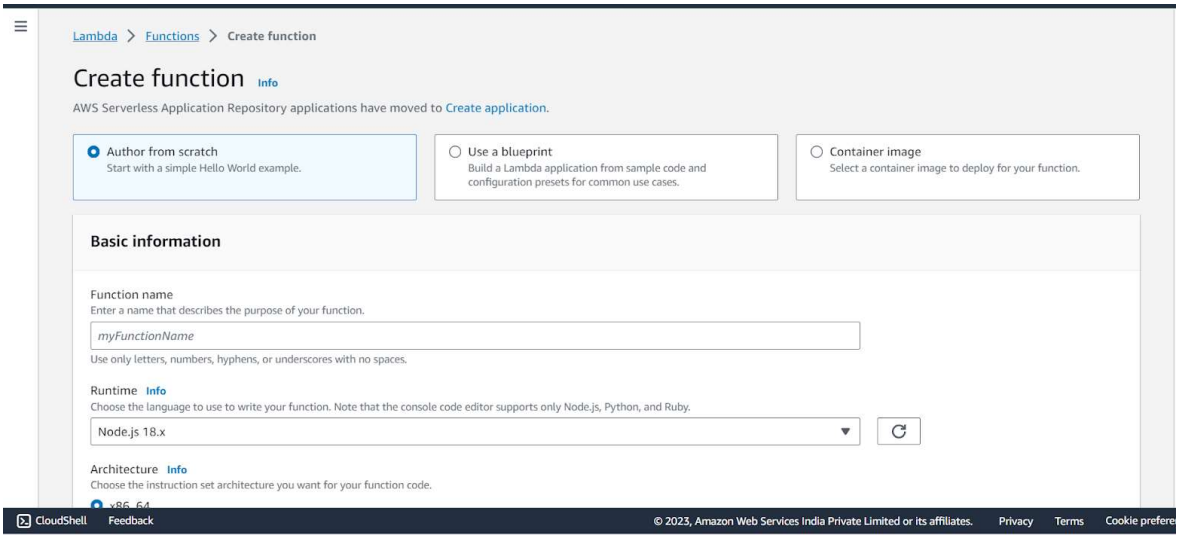
AIM: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

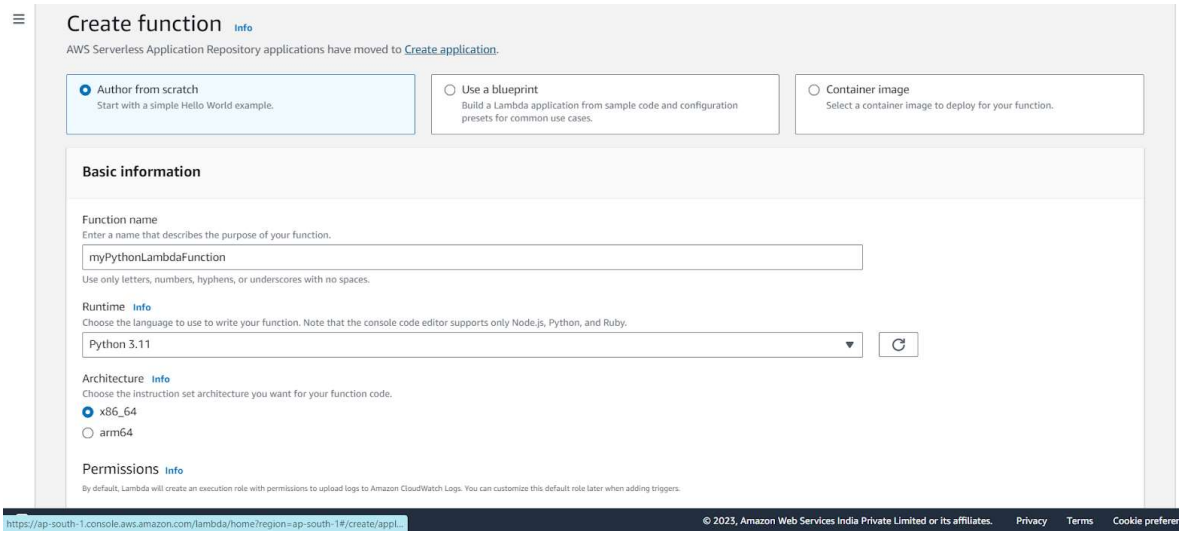
Steps to create an AWS Lambda function

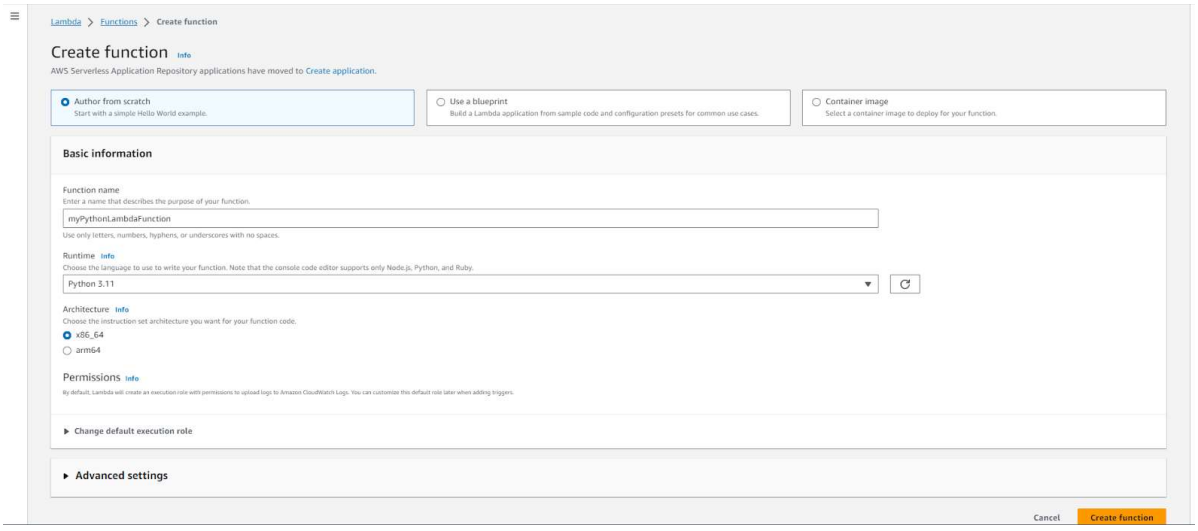
Step 1:Open up the Lambda Console and click on the Create button.
Be mindful of where you create your functions since Lambda is region-dependent.



2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.
Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.
After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

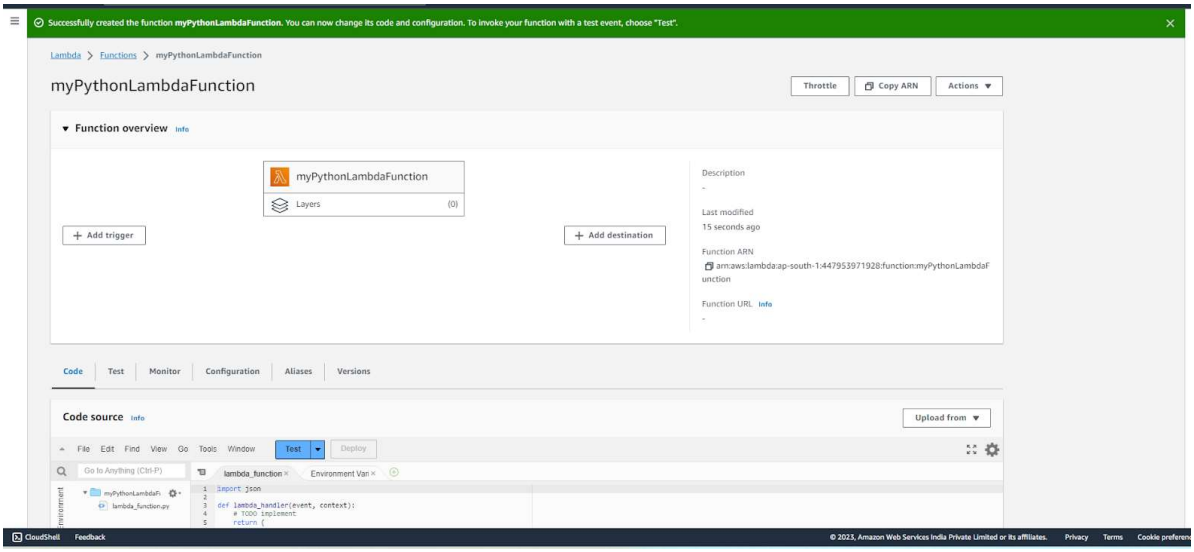


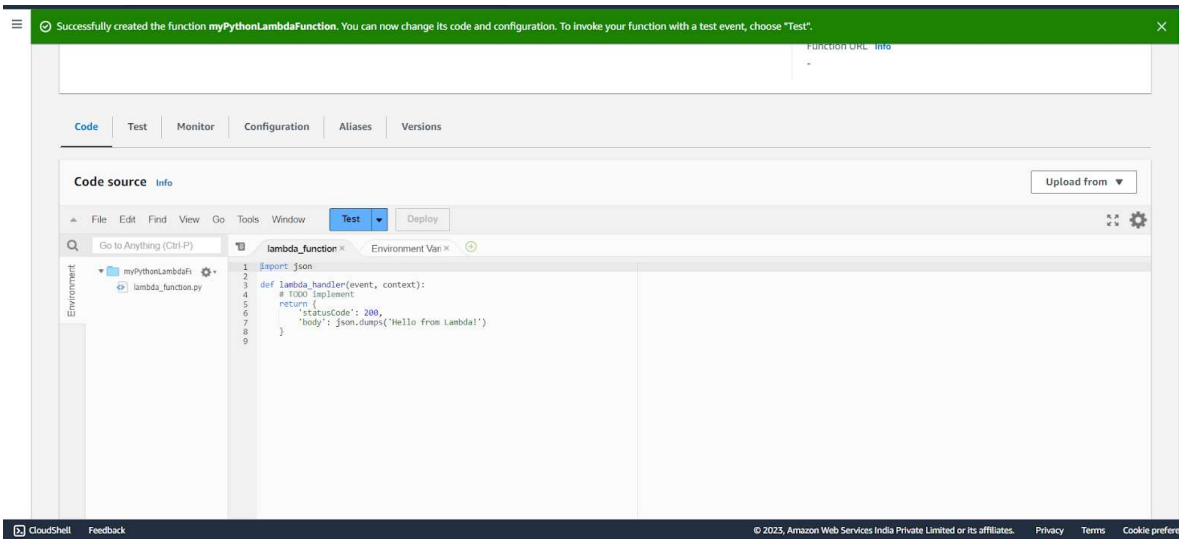




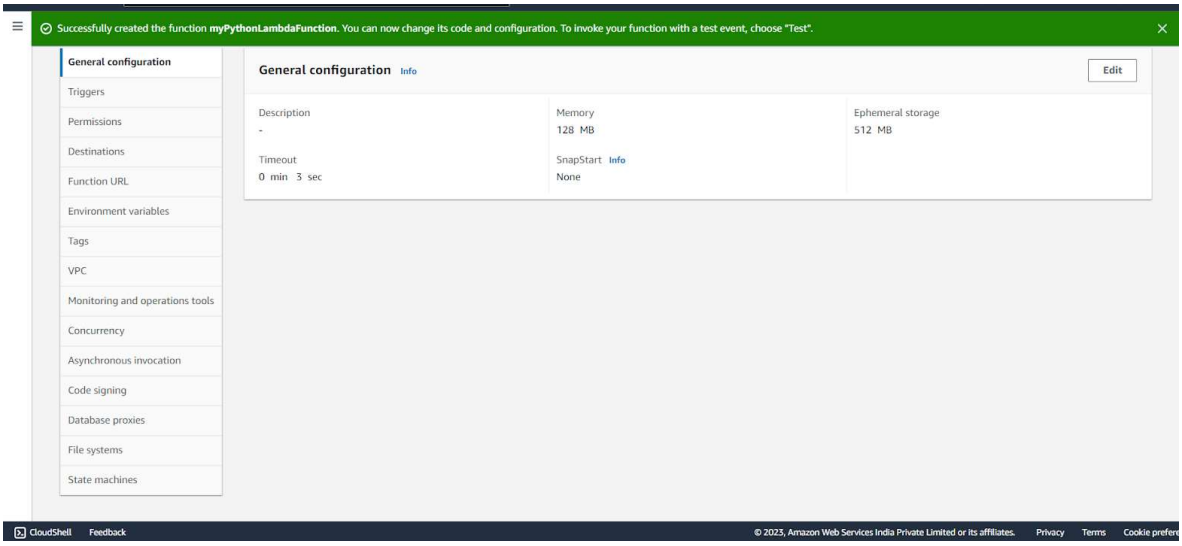
Click on the Create button.

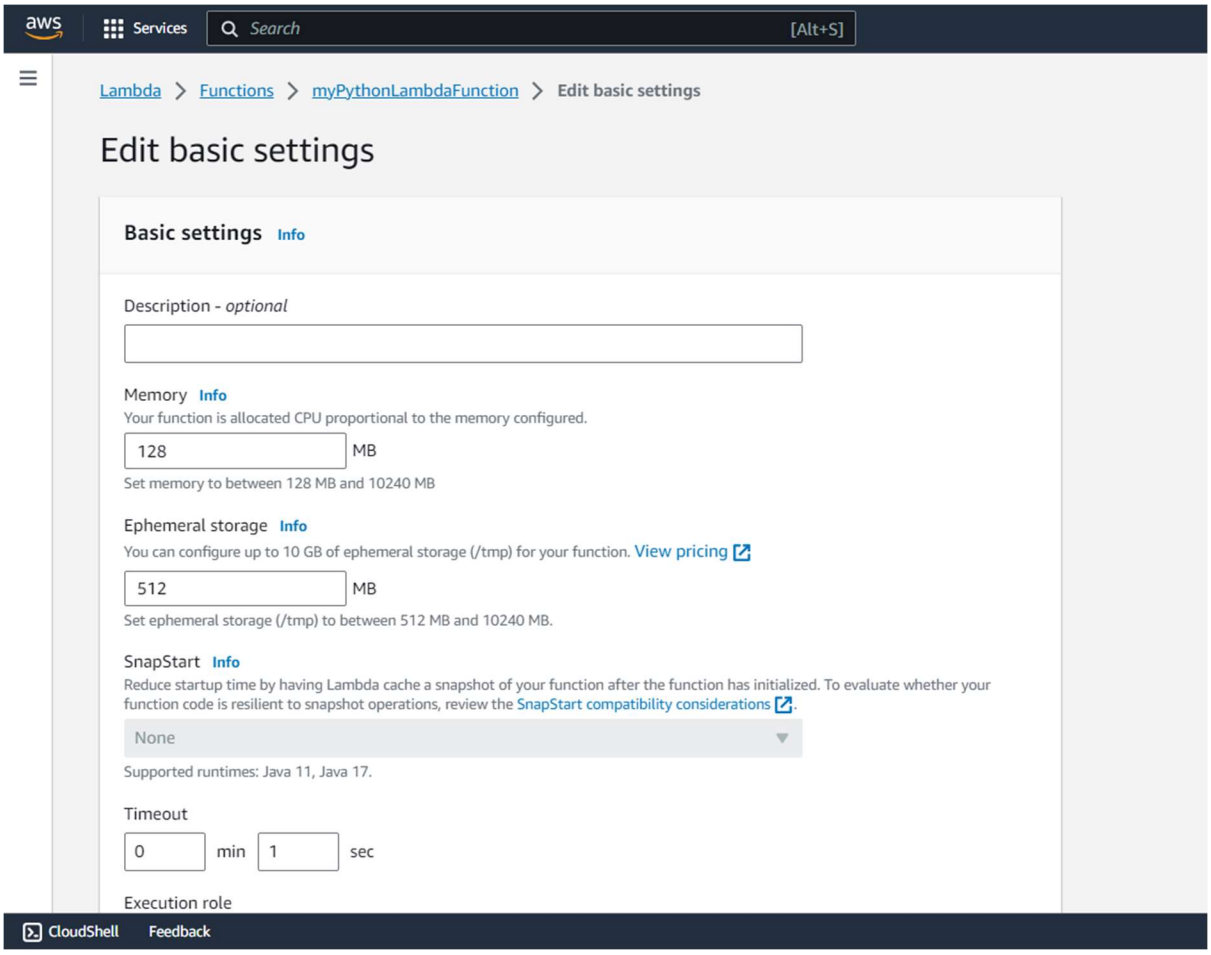
3. This process will take a while to finish and after that, you'll get a message that your function was successfully created.



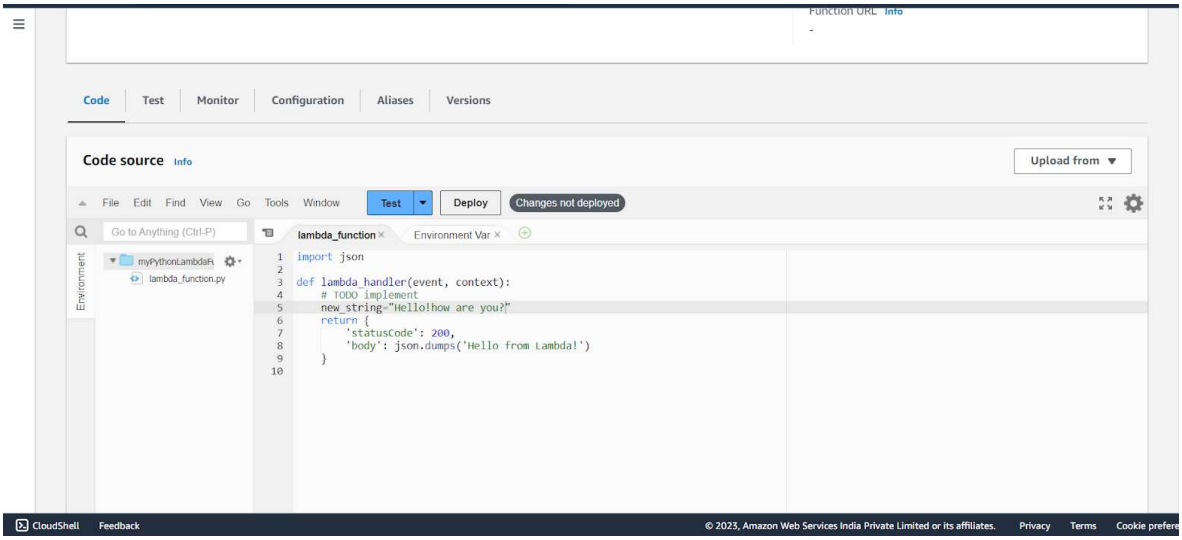


4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

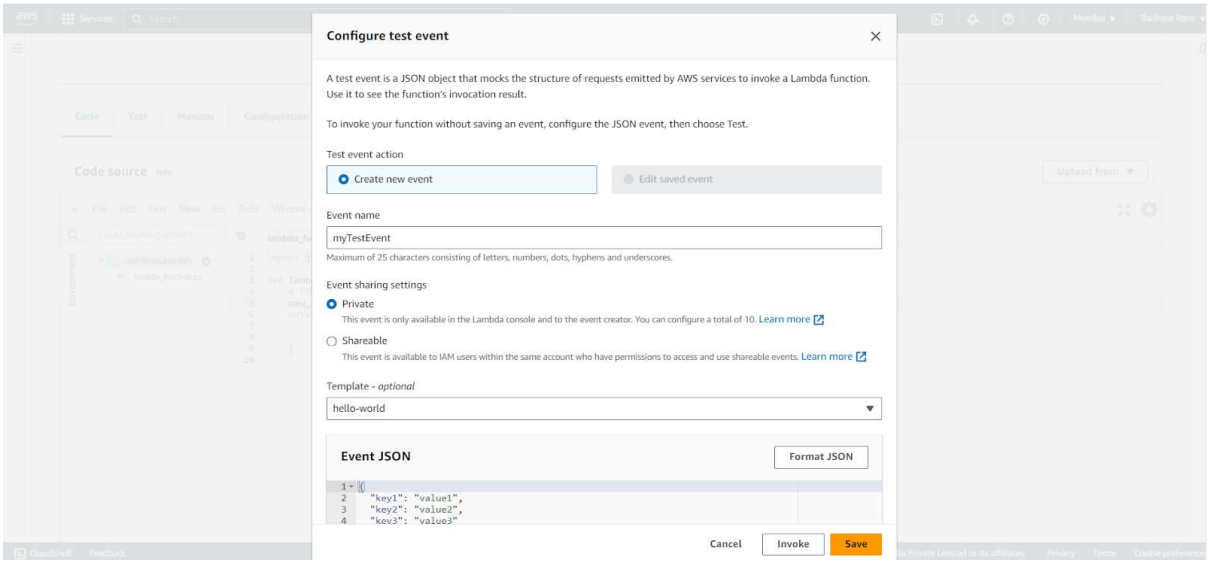




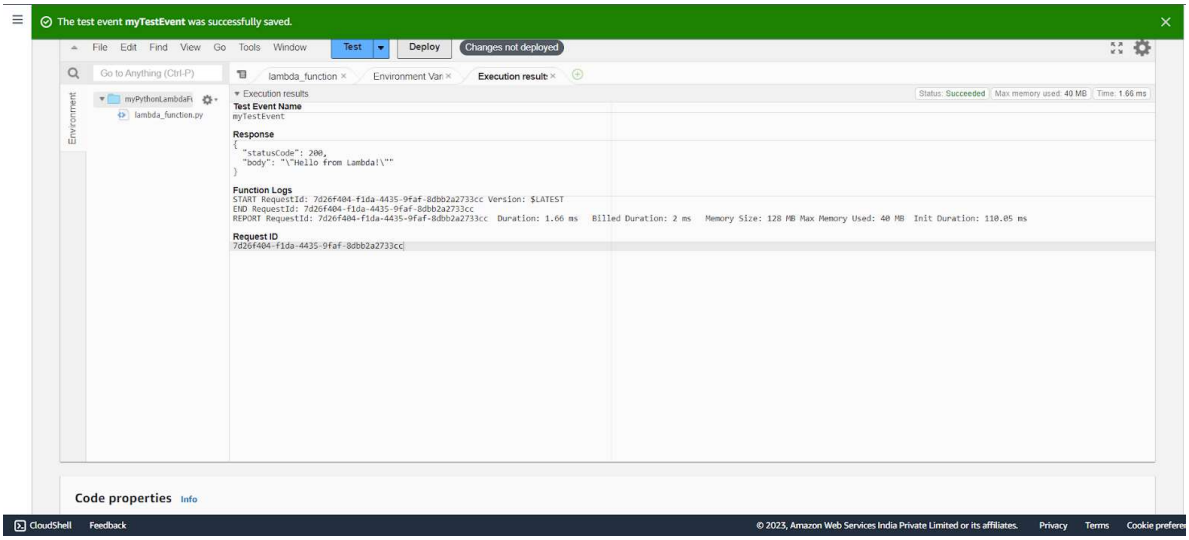
5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed.
Press Ctrl + S to save the file and click Deploy to deploy the changes.

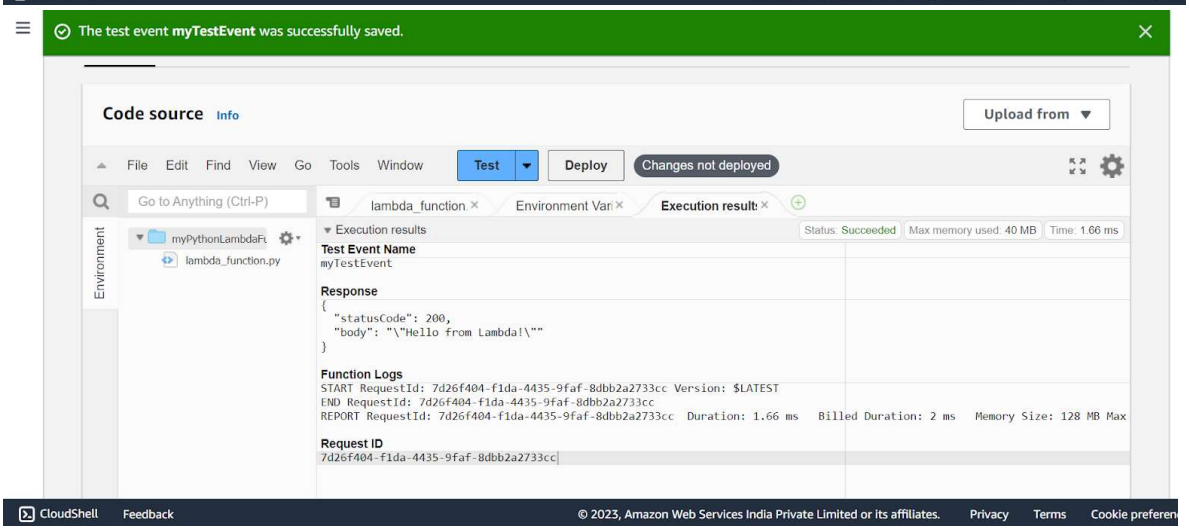


6. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



7. Now click on Test and you should be able to see the results.



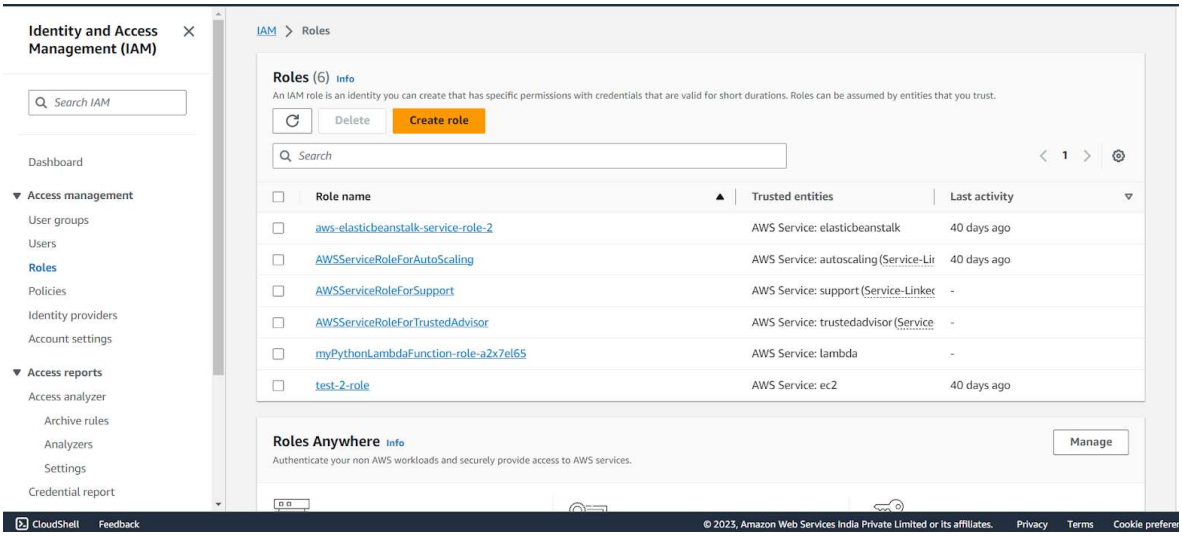


Conclusion: Thus, we understood AWS Lambda, its workflow, various functions and created our first Lambda functions using Python / Java / Nodejs.

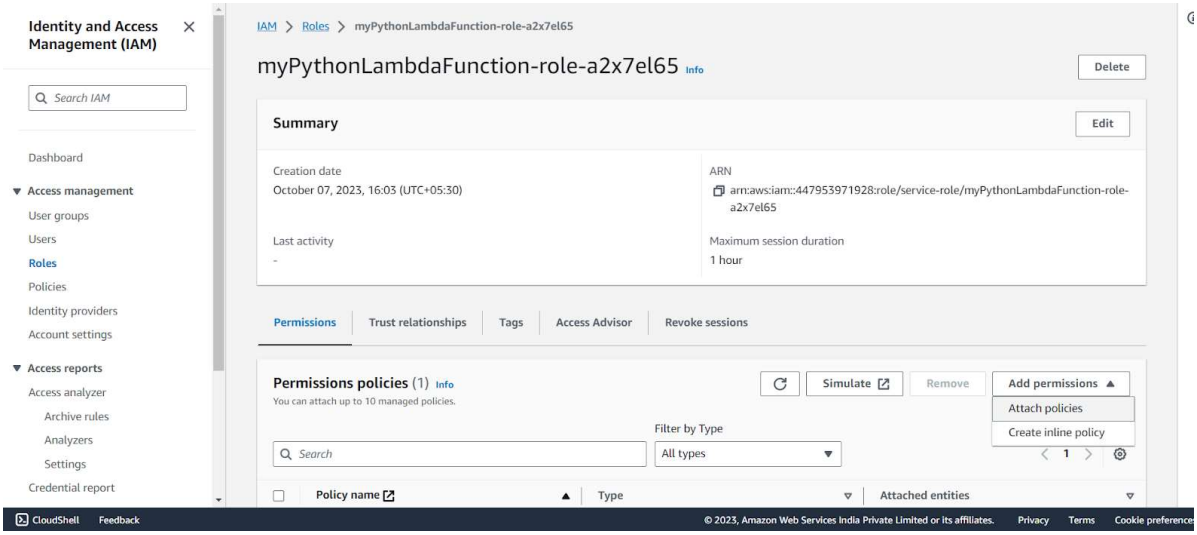
Experiment No 12

Mohit Patil
D15A 37

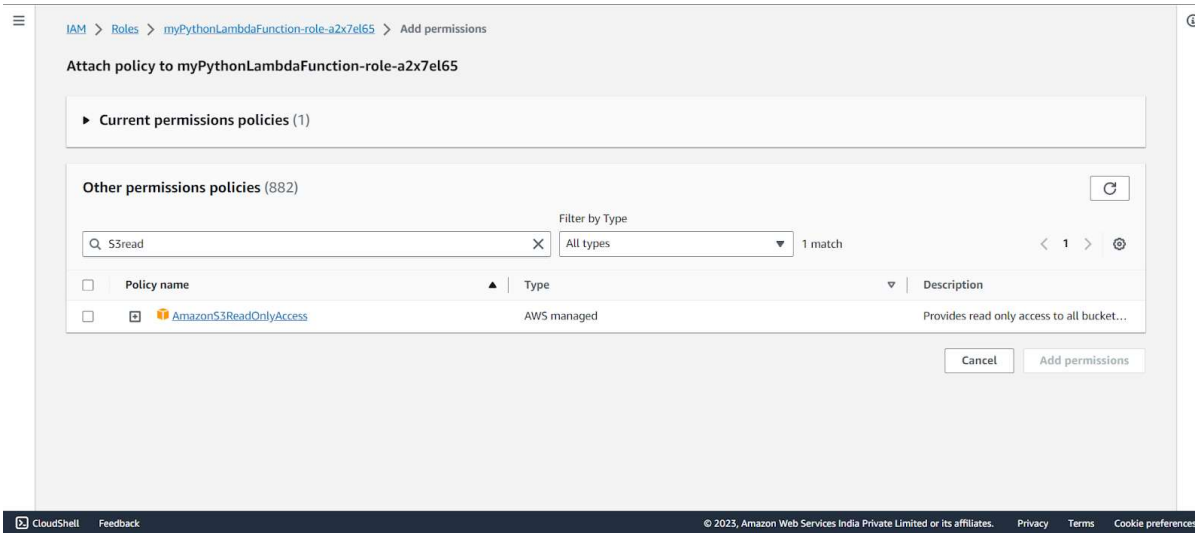
Step 1: Open up the IAM Console and under Roles, choose the Role we previously created for the Python Lambda Function (You can find your role name configuration of your Lambda function).



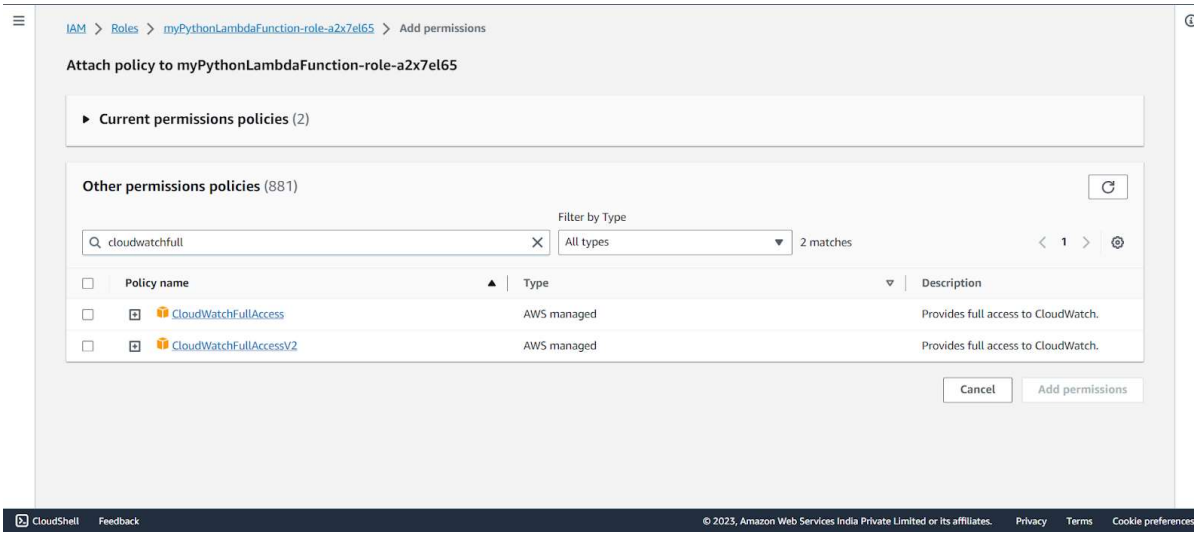
Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.



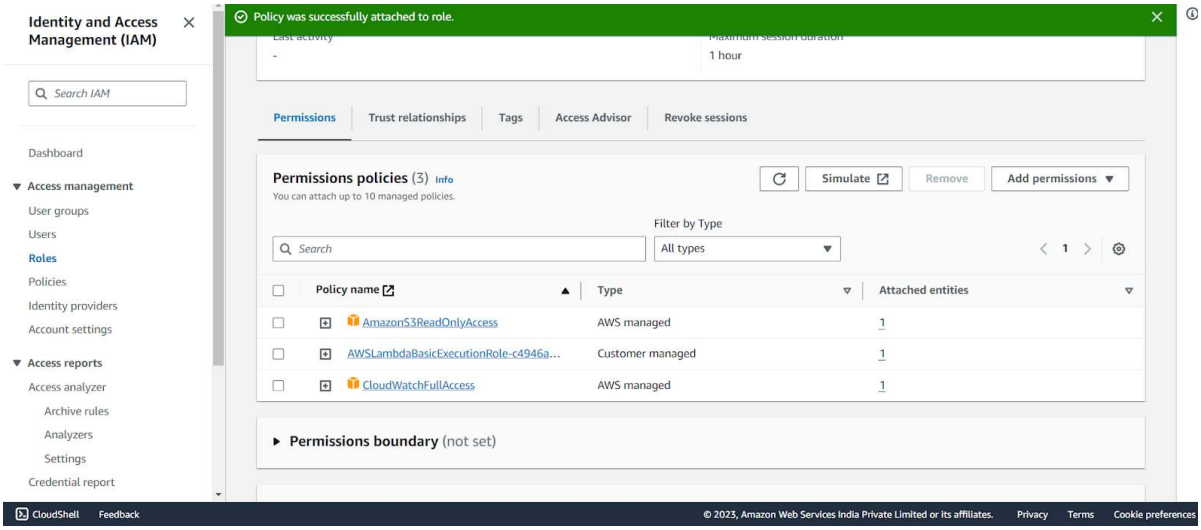
S3-ReadOnly



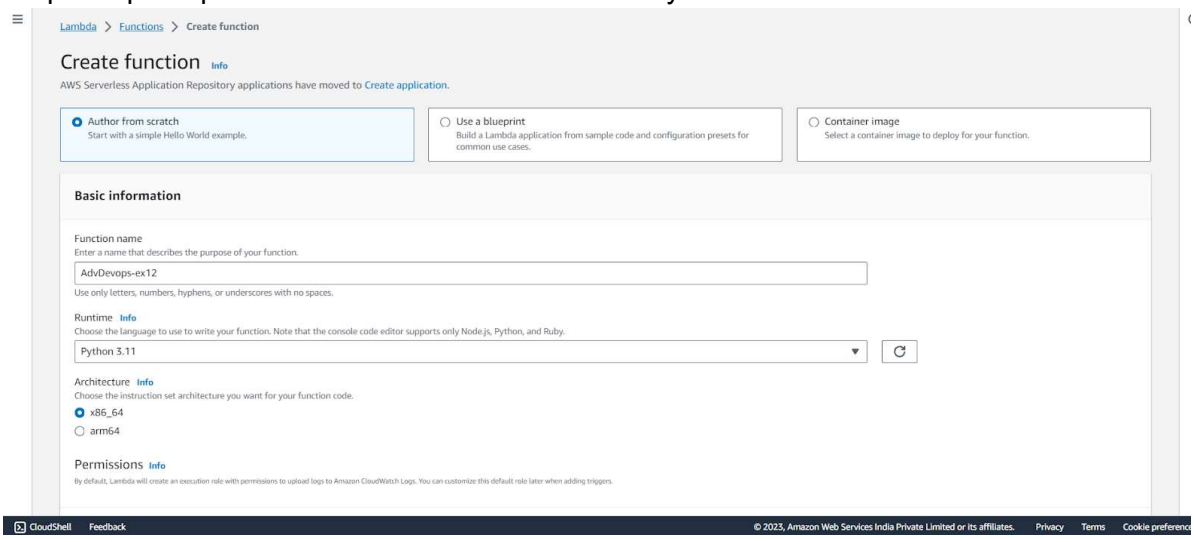
CloudWatchFull



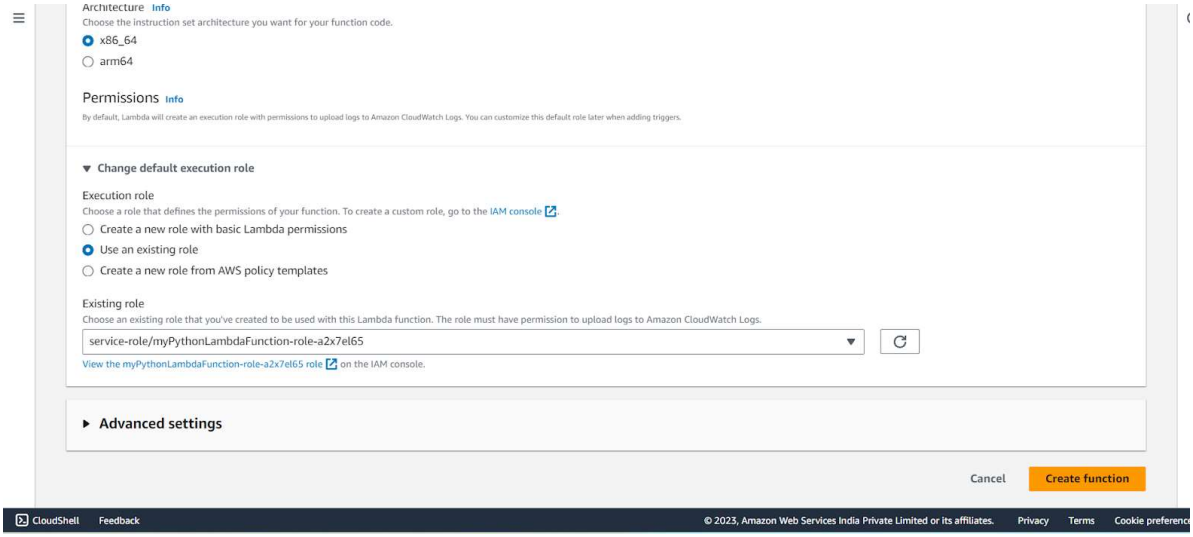
After successful attachment of policy you will see something like this you will be able to see the updated policies.



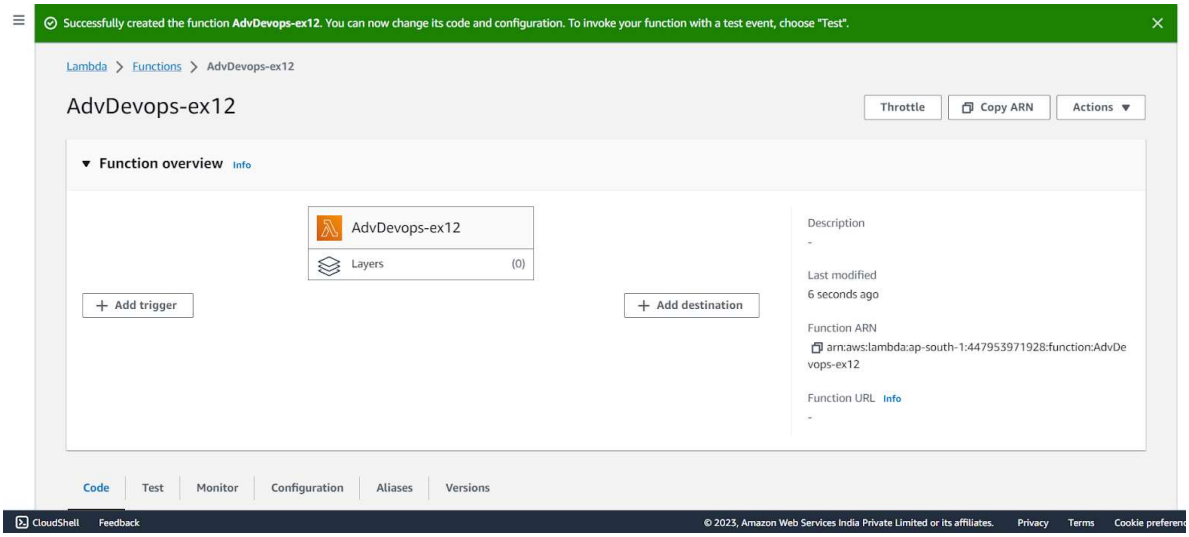
Step 3: Open up AWS Lambda and create a new Python function.



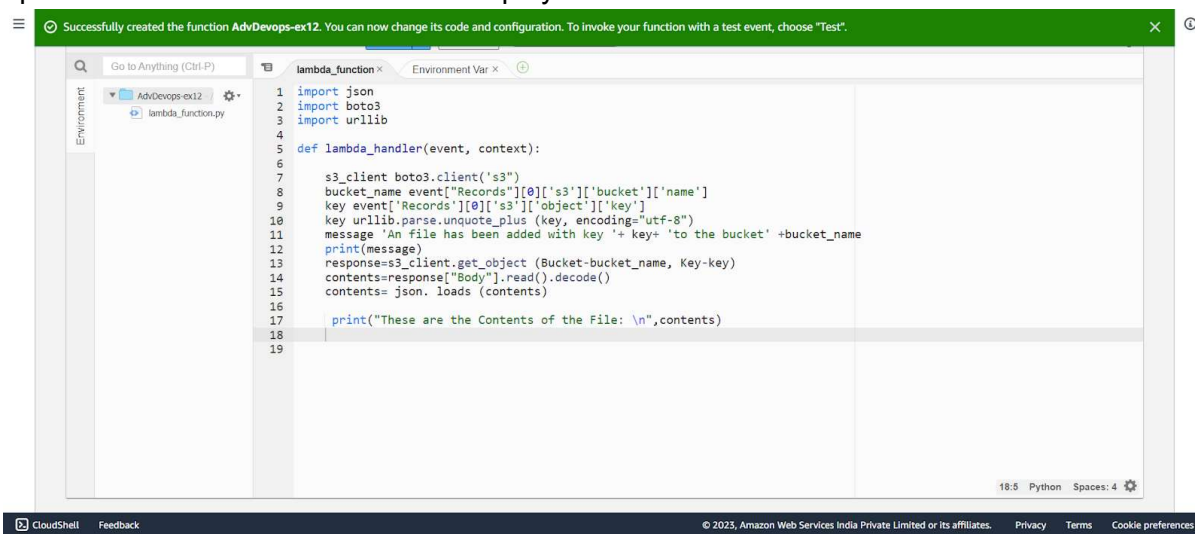
Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.



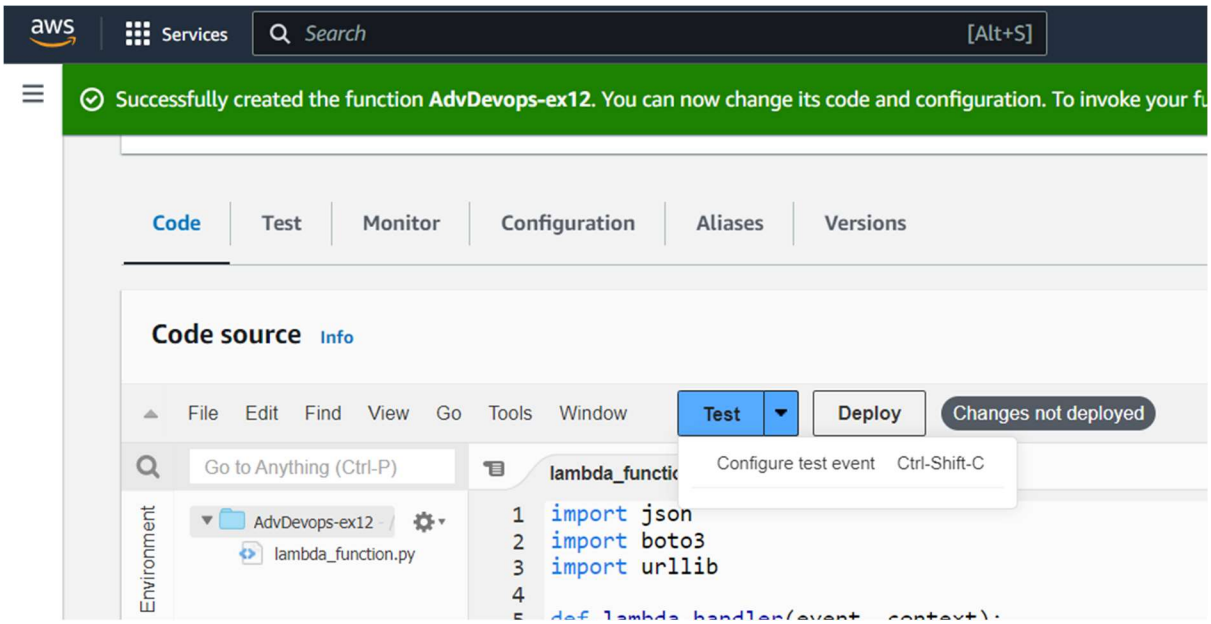
Step 4: The function is up and running.

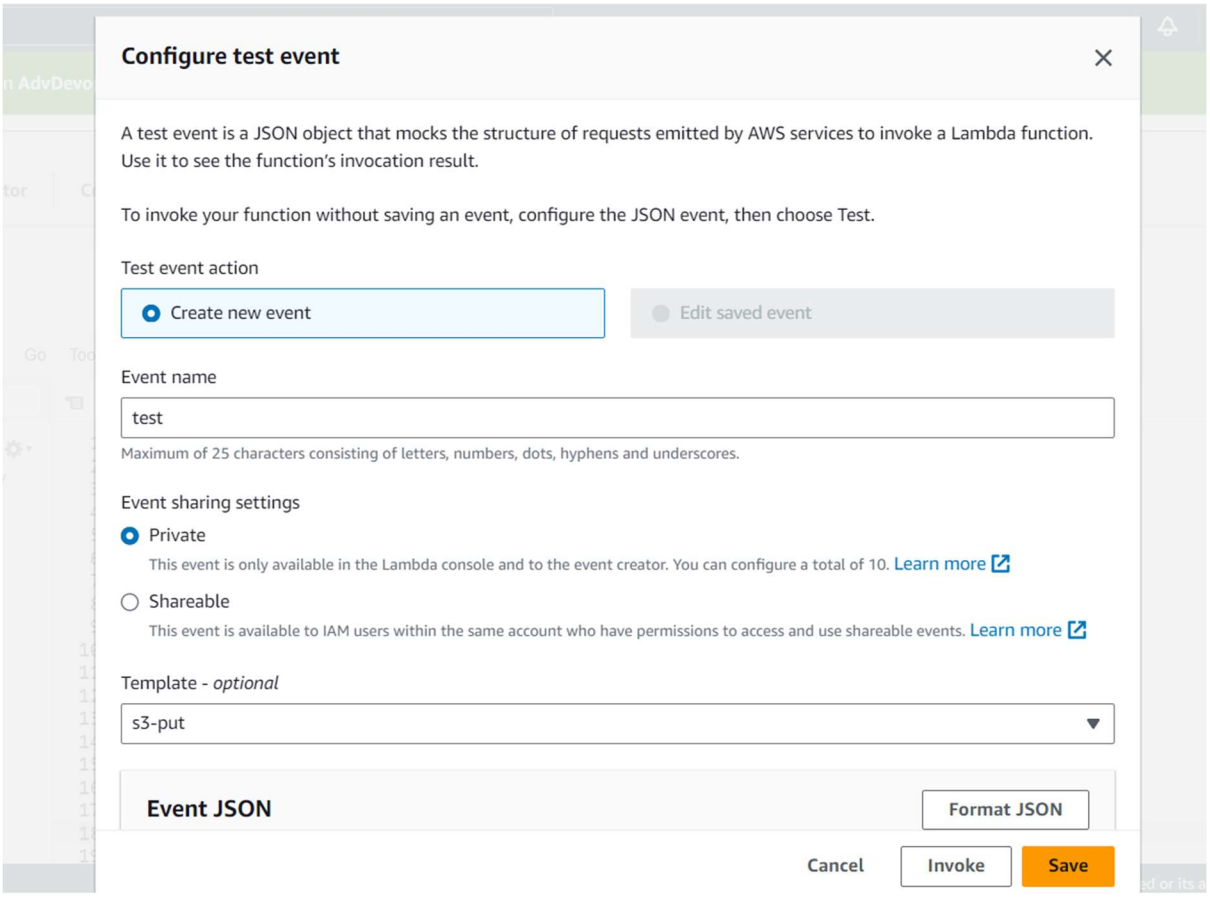


Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.



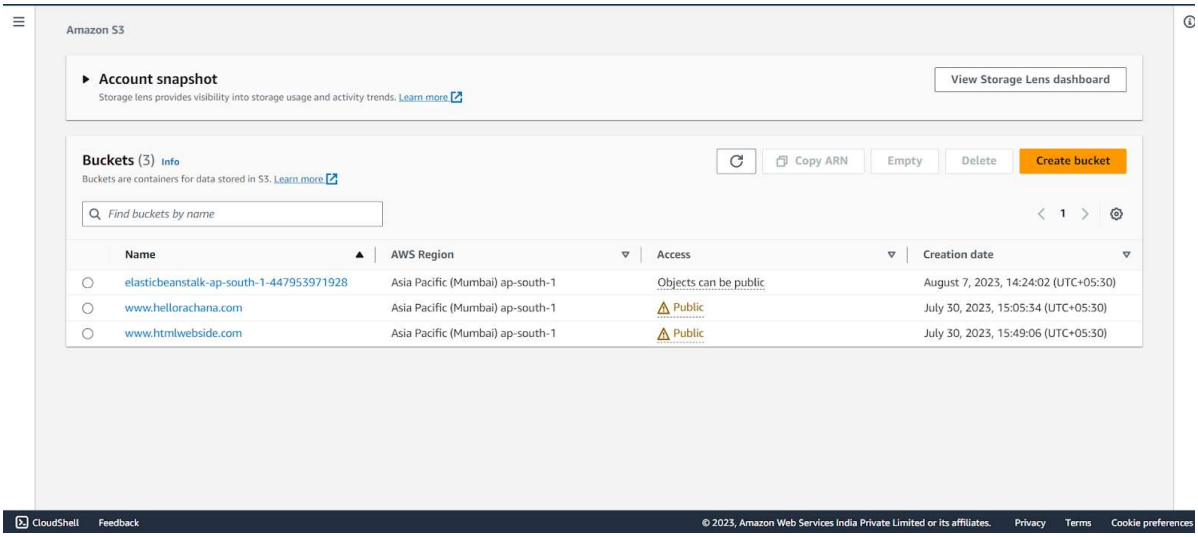
Step 6: Click on Test and choose the ‘S3 Put’ Template.



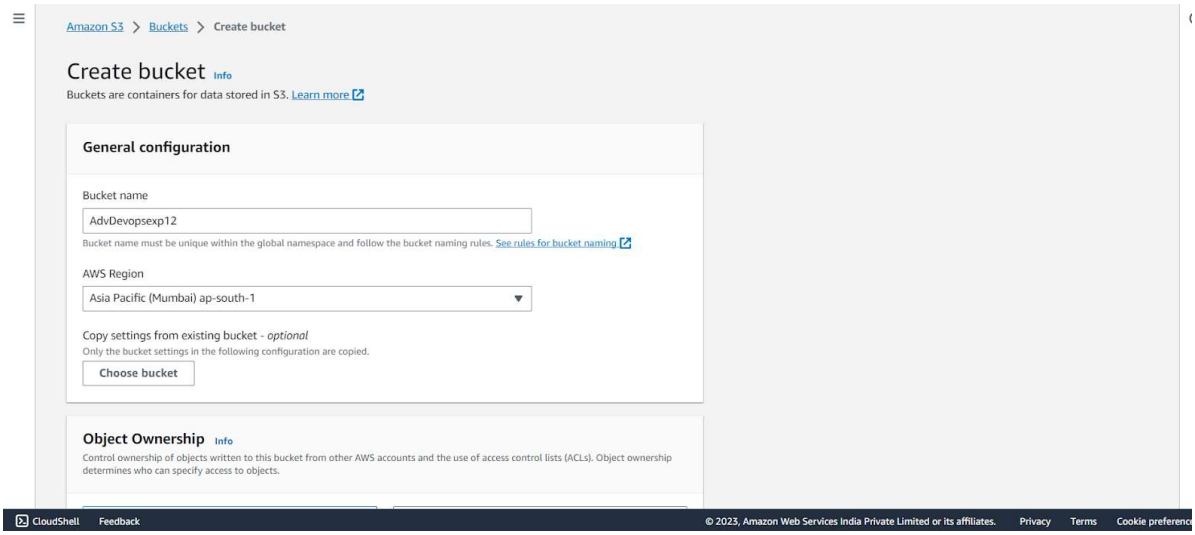


And Save it.

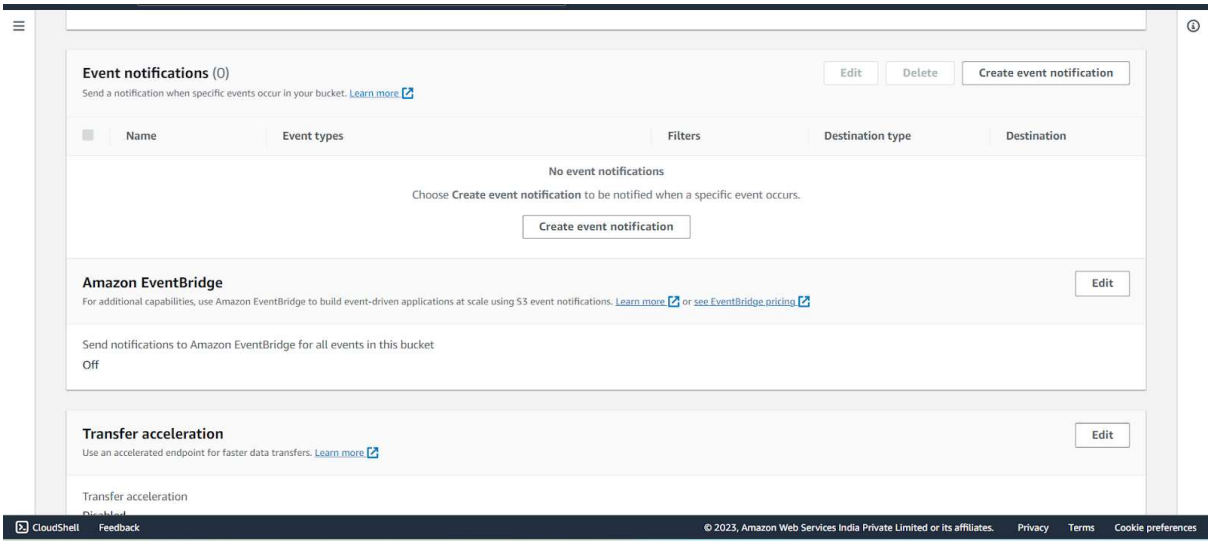
Step 7: Open up the S3 Console and create a new bucket.



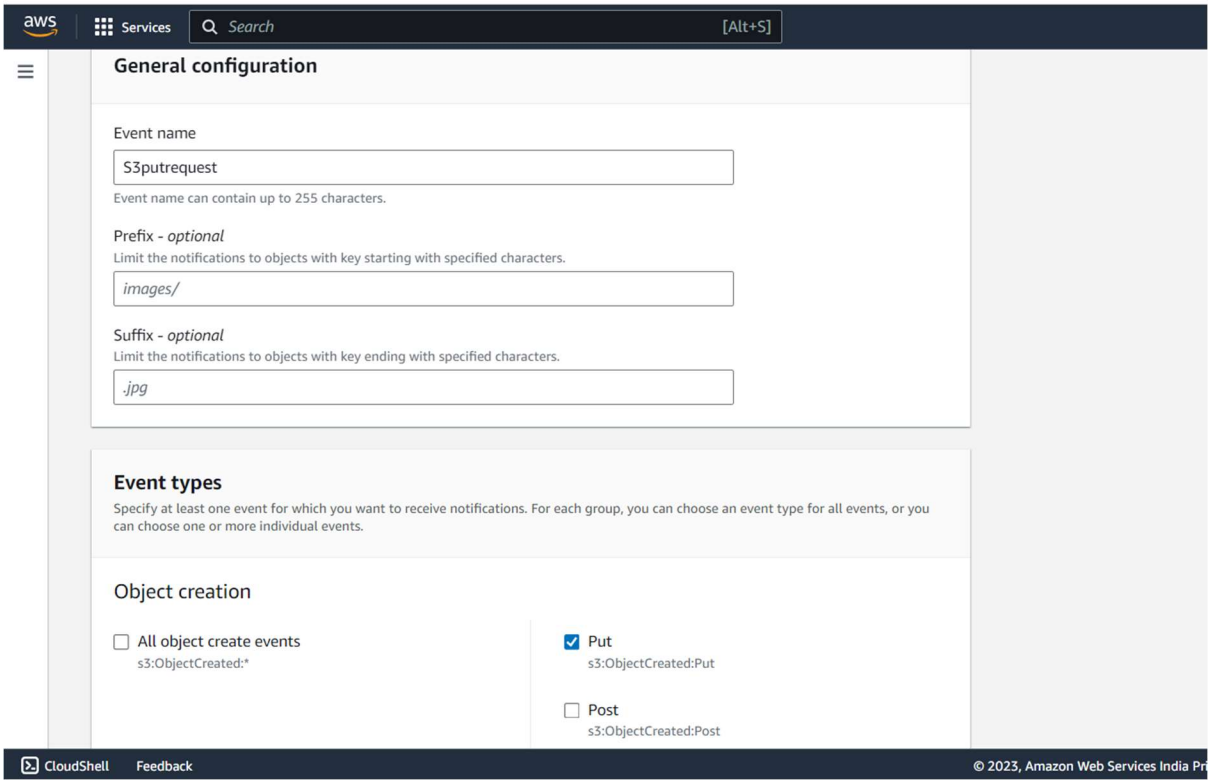
Step 8: With all general settings, create the bucket in the same region as the function.



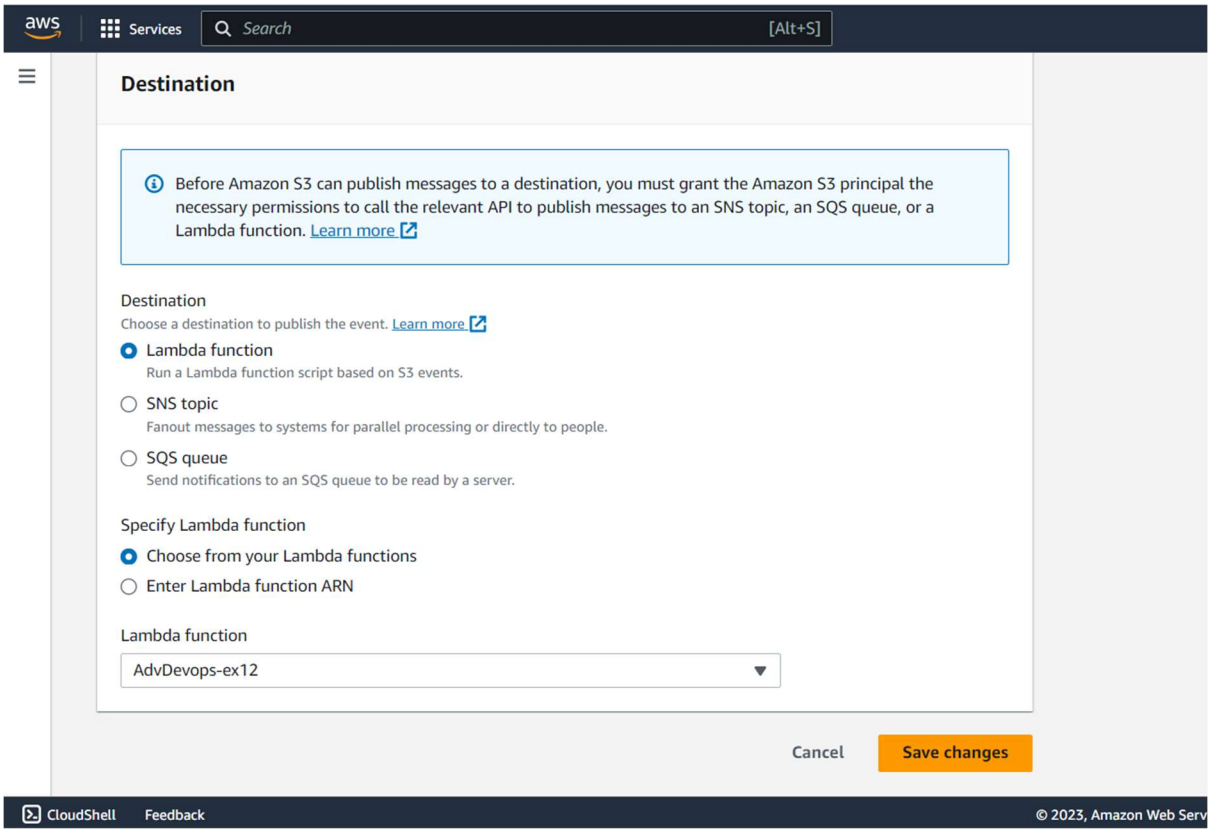
Step 9: Click on the created bucket and under properties, look for events.



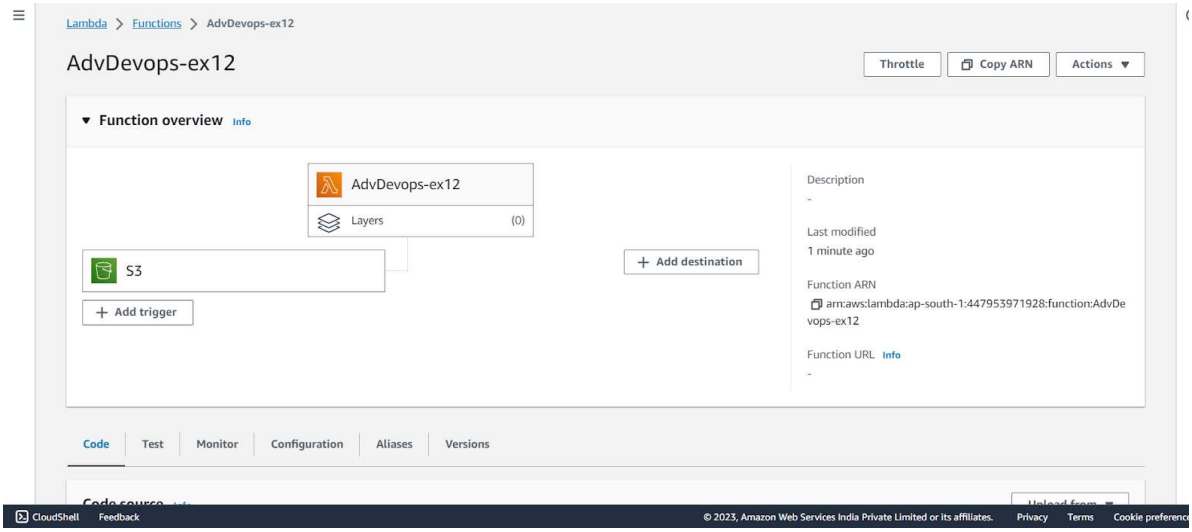
Click on Create Event Notification.
Step 10: Mention an event name and check Put under event types.



Choose Lambda function as destination and choose your lambda function and save the changes.

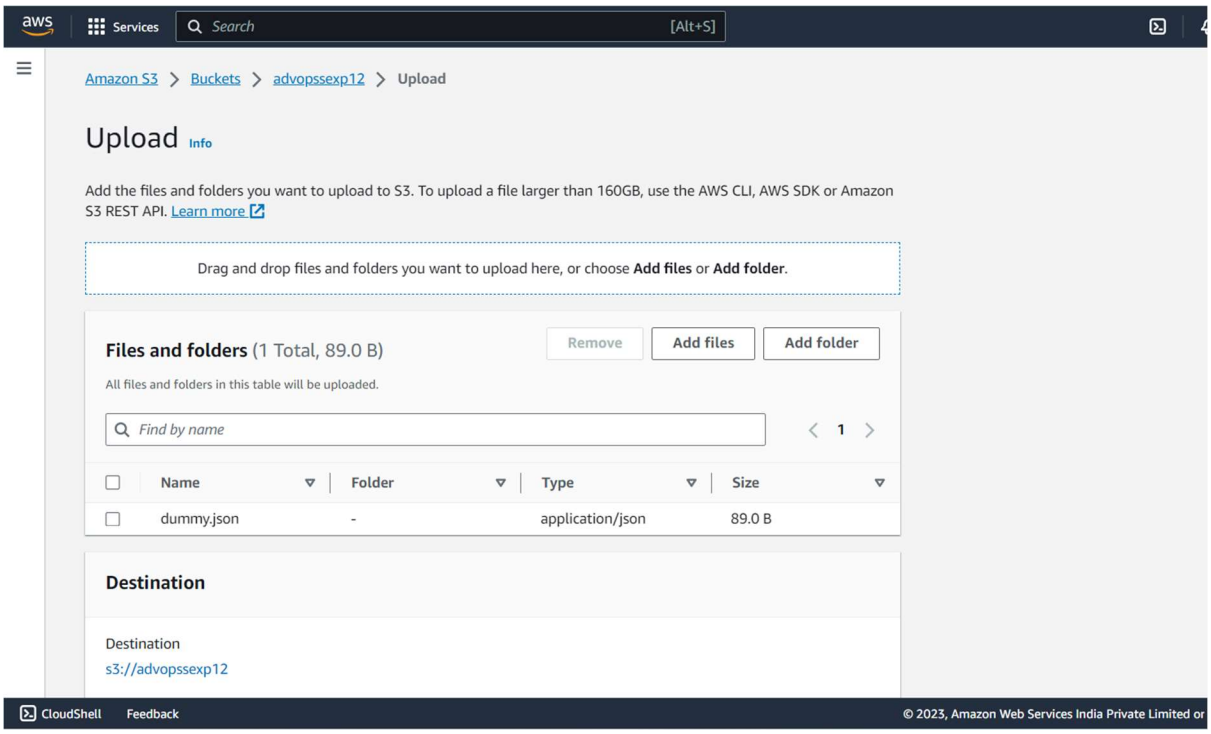


Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



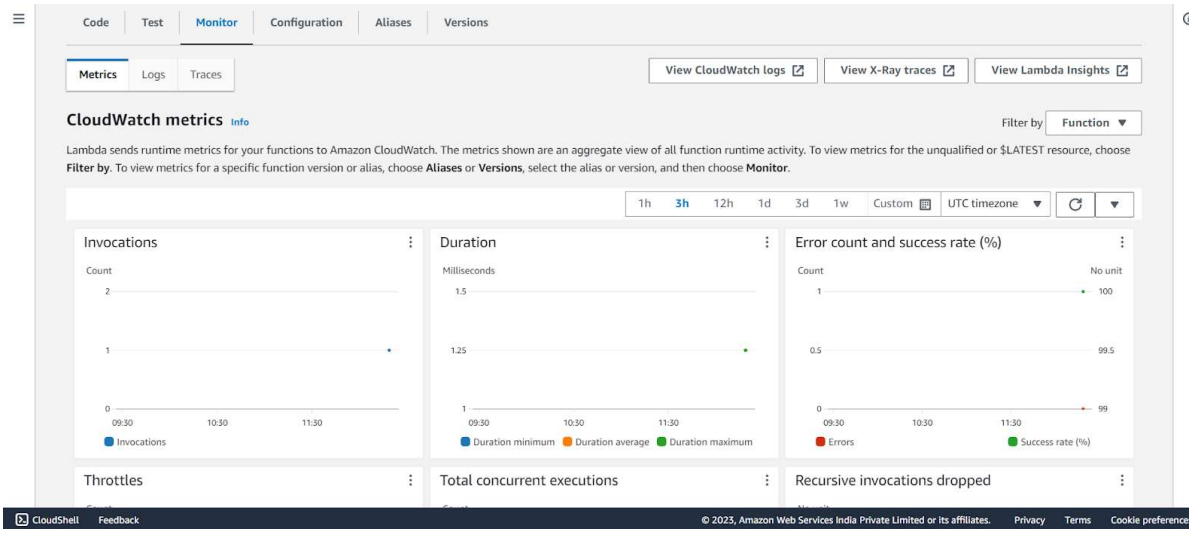
Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.
Step 14: Select the dummy data file from your computer and click Upload.

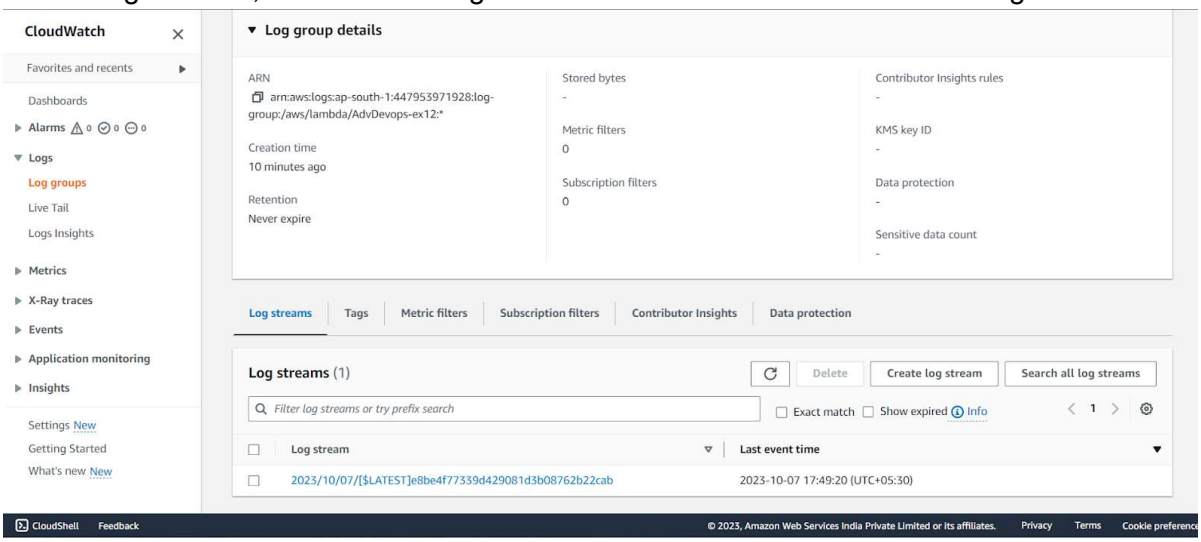


Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

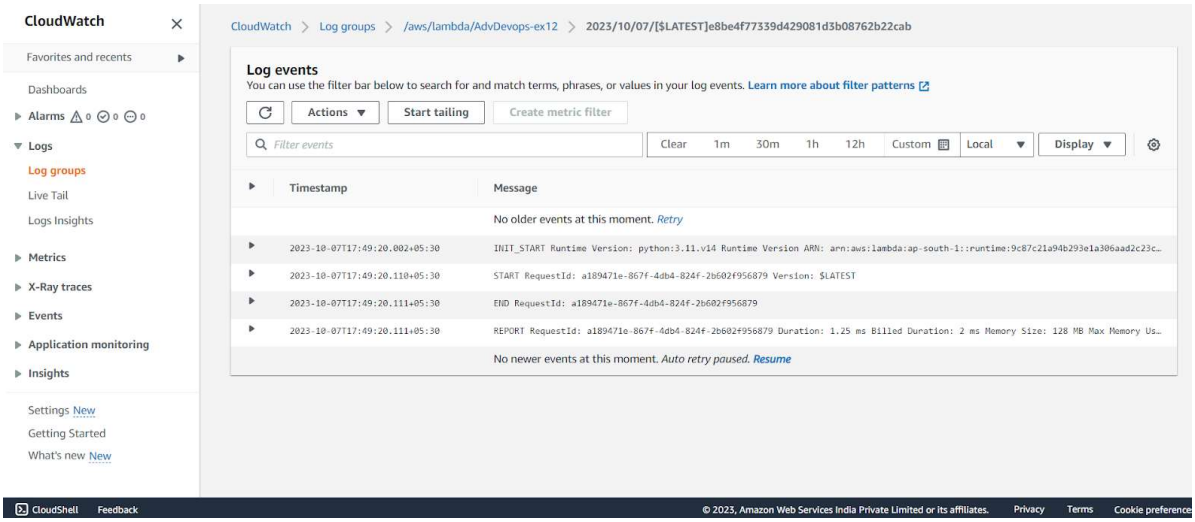
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.



Conclusion: Thus, we have created a Lambda function which logs “An Image has been added” once you add an object to a specific bucket in S3.