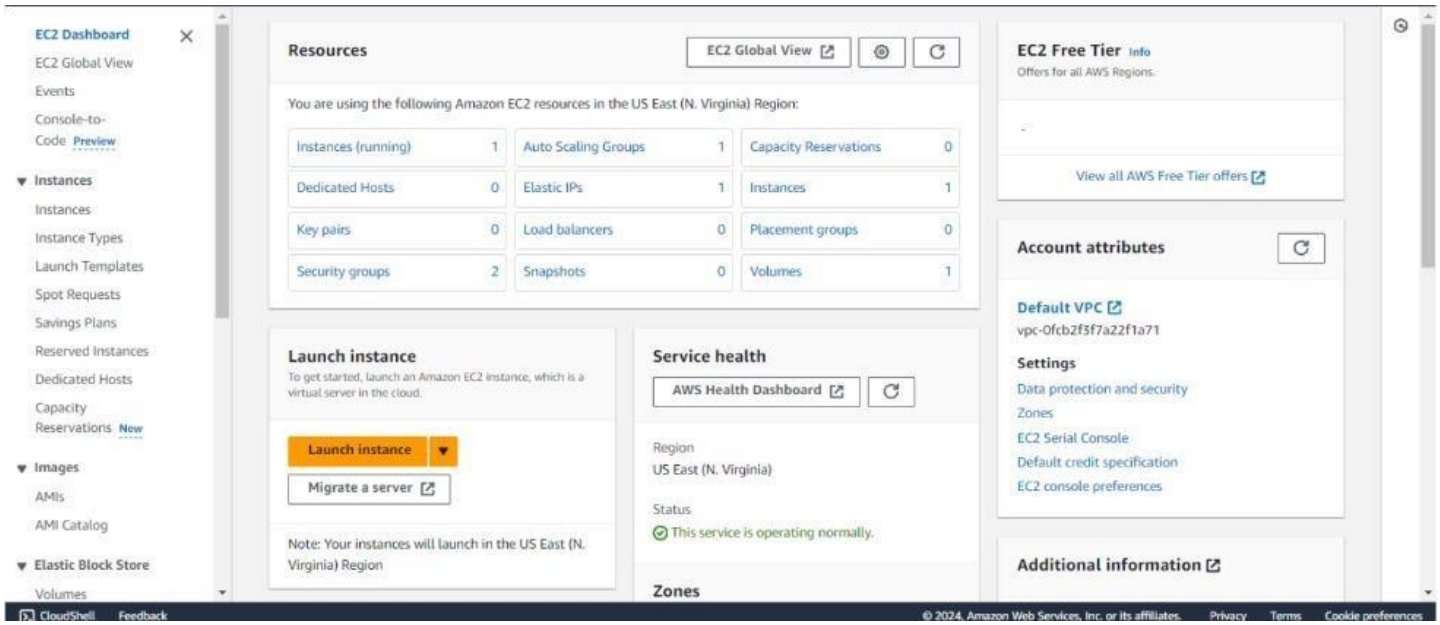
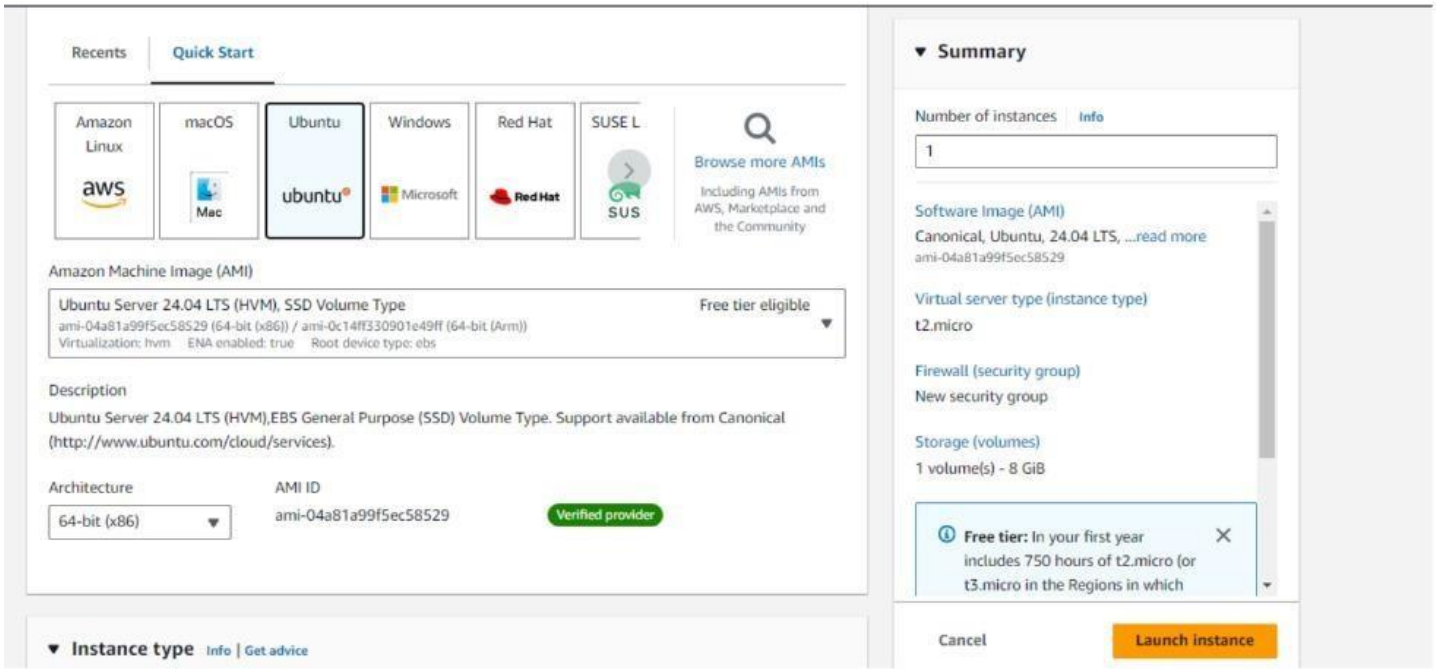
Experiment 1

Aim: To develop a website and host it on your local machine on a VM

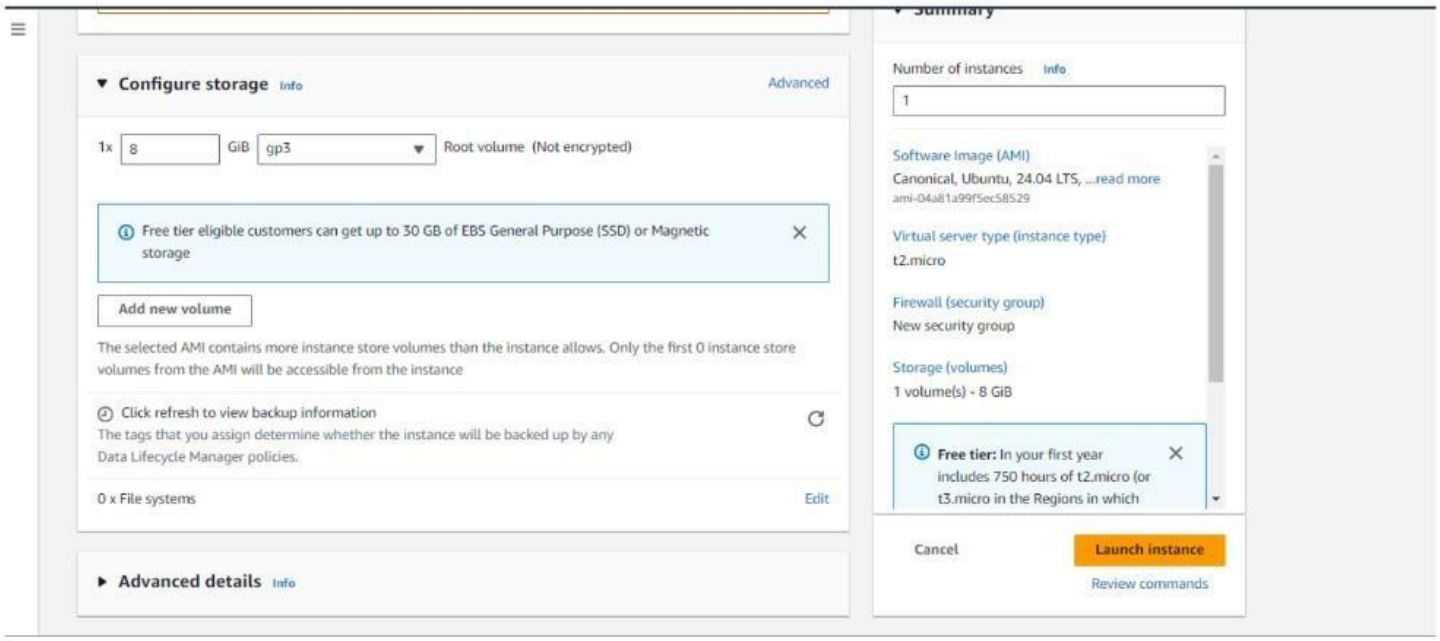
1. Open AWS Academy and select launch instance



2. And then select Ubuntu or Linux.



3. Set the configuration and then connect to the instance which you created.



4. Execute the following commands in the aws console.

<u>Commands :</u>

sudo su

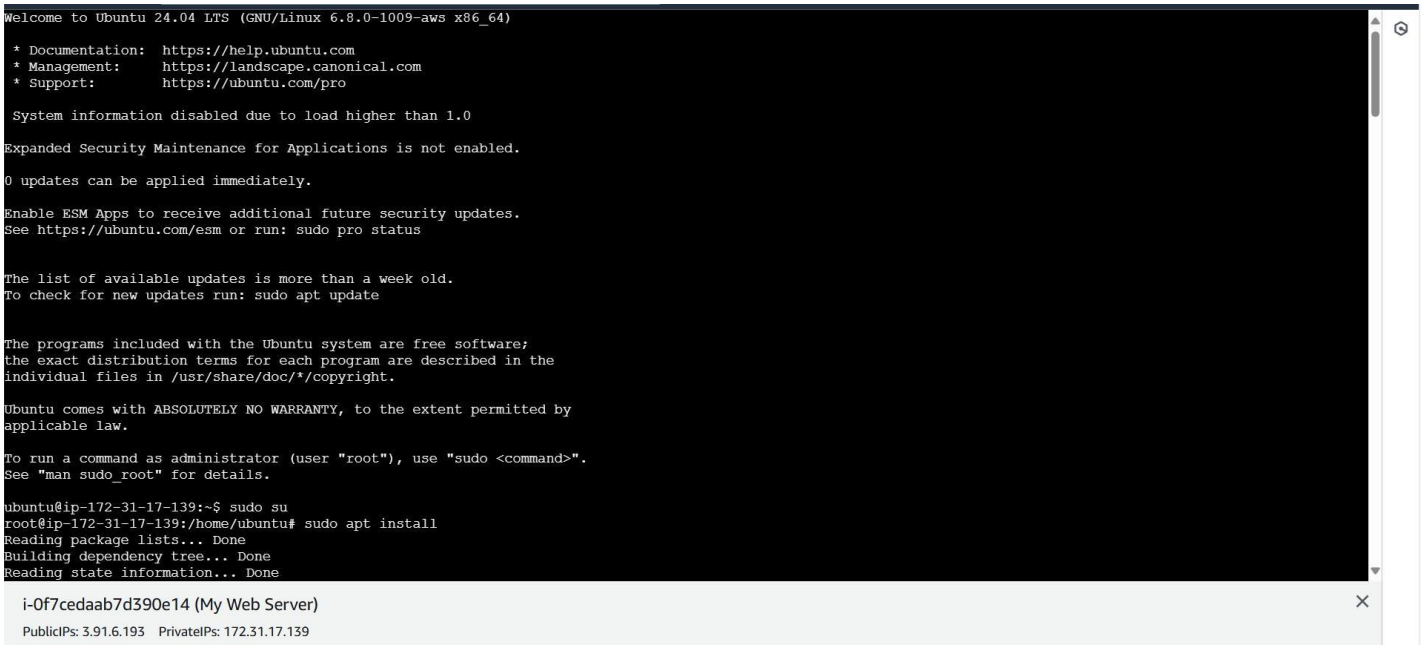
sudo apt install

sudo apt-get update

apt install apache2

systemctl status apache2

cd /var/www/html/



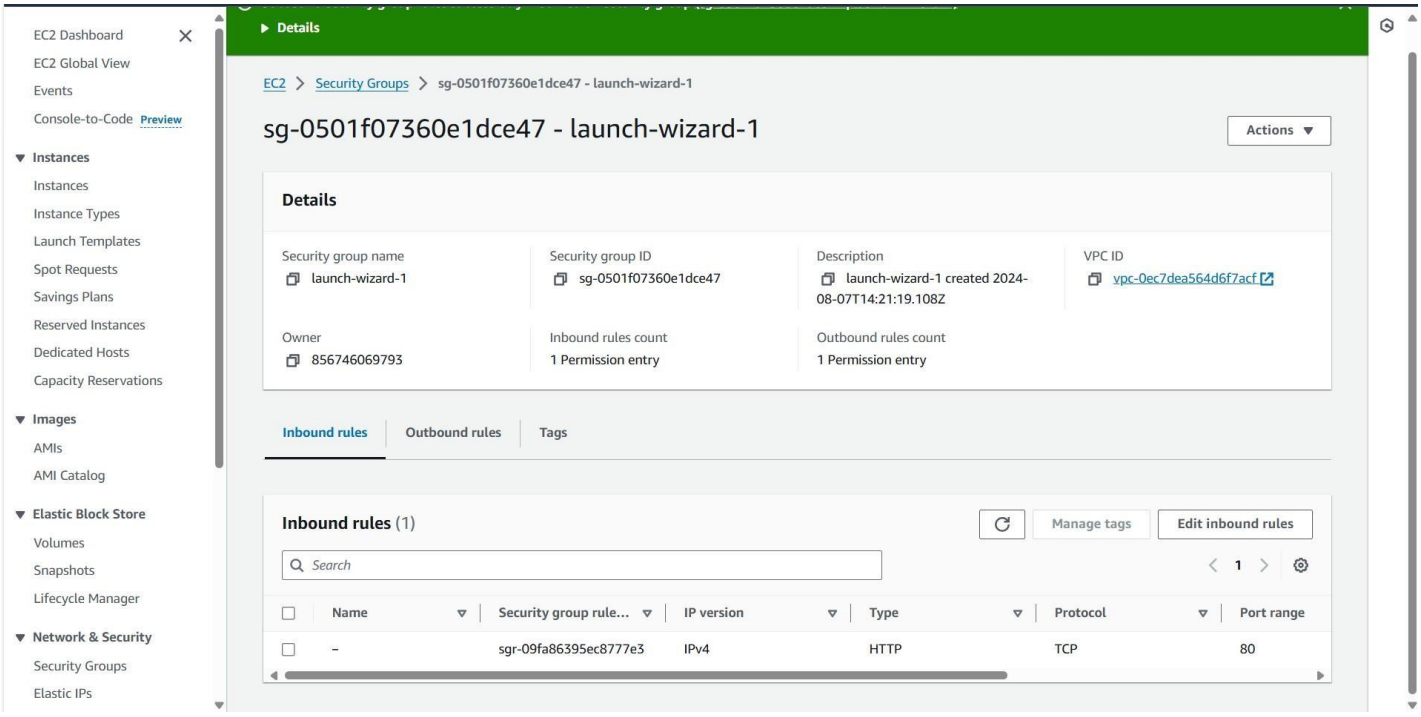
```
root@ip-172-31-17-139:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [265 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [63.3 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [3668 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [247 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [107 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9220 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [208 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [40.7 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [420 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [318 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [82.9 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5676 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [319 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [134 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.6 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [208 kB]
```

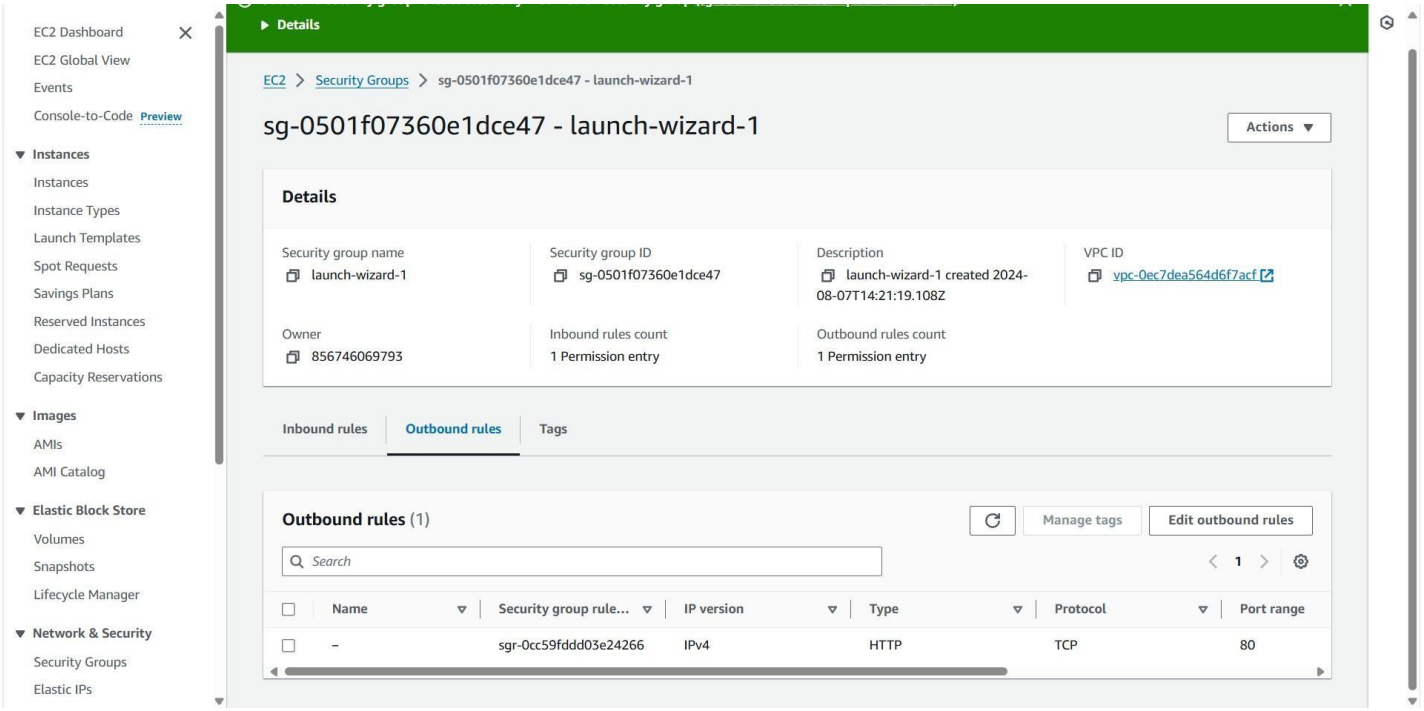
```
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [208 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [40.7 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 6s (5073 kB/s)
Reading package lists... Done
root@ip-172-31-17-139:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 42 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
```

```
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
Fetched 2083 kB in 0s (25.8 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1t64:amd64.
(Reading database ... 67739 files and directories currently installed.)
Preparing to unpack .../0-libapr1t64_1.7.2-3.1build2_amd64.deb ...
Unpacking libapr1t64:amd64 (1.7.2-3.1build2) ...
Selecting previously unselected package libaprutil1t64:amd64.
Preparing to unpack .../1-libaprutil1t64_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
Preparing to unpack .../2-libaprutil1-dbd-sqlite3_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../3-libaprutil1-ldap_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package liblua5.4-0:amd64.
Preparing to unpack .../4-liblua5.4-0_5.4.6-3build2_amd64.deb ...
Unpacking liblua5.4-0:amd64 (5.4.6-3build2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../5-apache2-bin_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2-bin (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../6-apache2-data_2.4.58-1ubuntu8.4_all.deb ...
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-17-139:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-08-07 14:32:03 UTC; 32s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2487 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.4M (peak: 5.6M)
        CPU: 37ms
     CGroup: /system.slice/apache2.service
             ├─2487 /usr/sbin/apache2 -k start
             ├─2490 /usr/sbin/apache2 -k start
             └─2491 /usr/sbin/apache2 -k start

Aug 07 14:32:03 ip-172-31-17-139 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 07 14:32:03 ip-172-31-17-139 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-17-139:/home/ubuntu# cd /var/www/html/
root@ip-172-31-17-139:/var/www/html#
```

5. Edit the inbound and outbound rules.

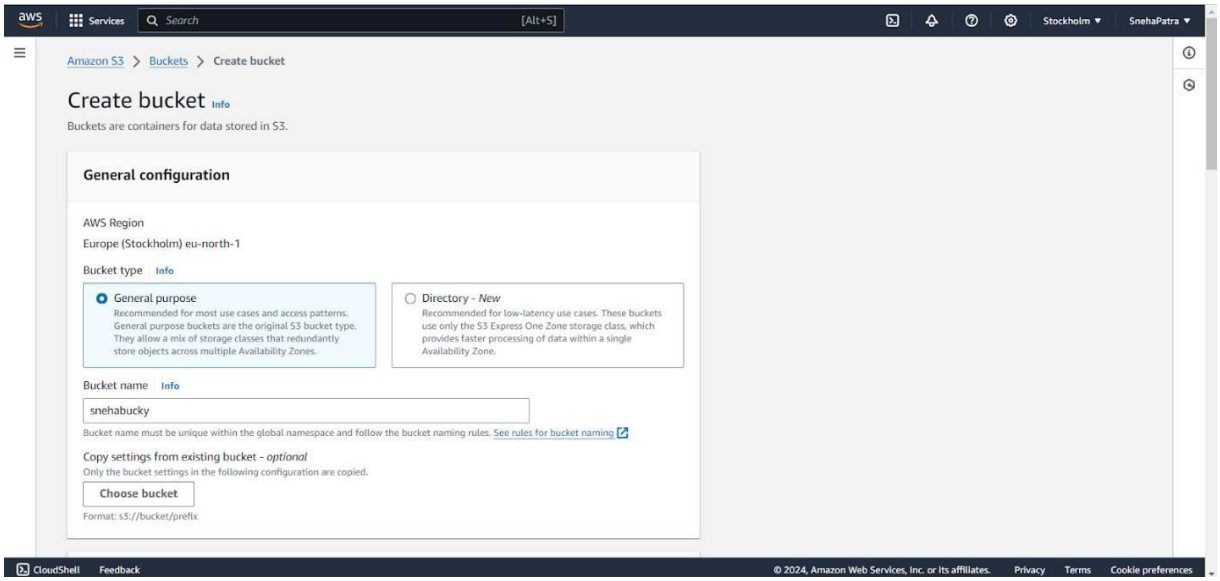


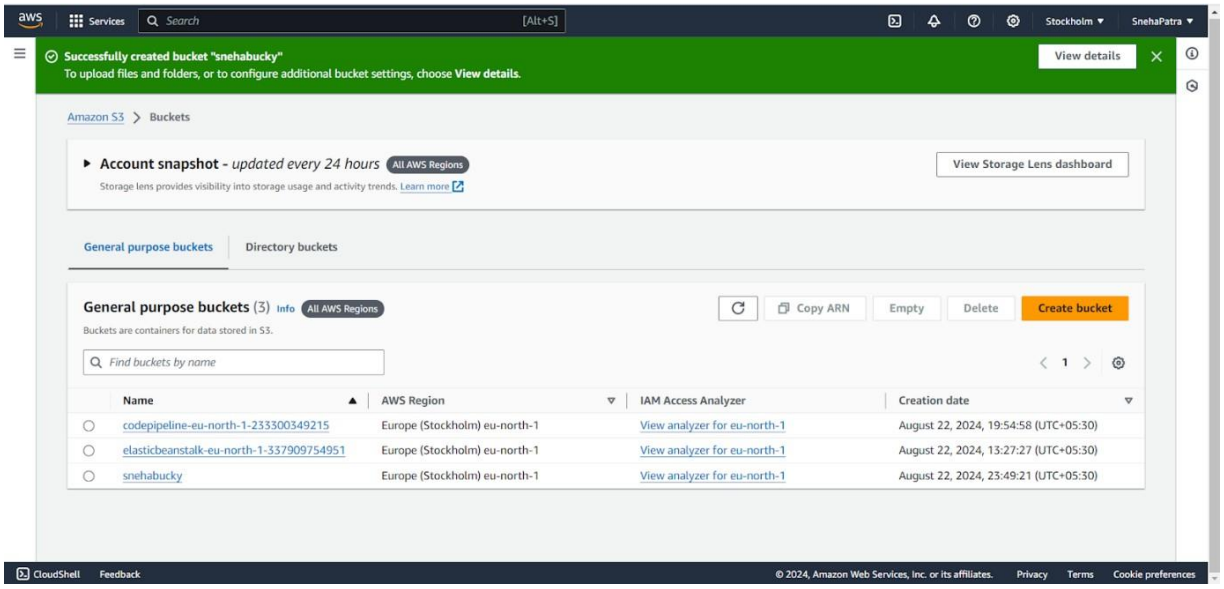


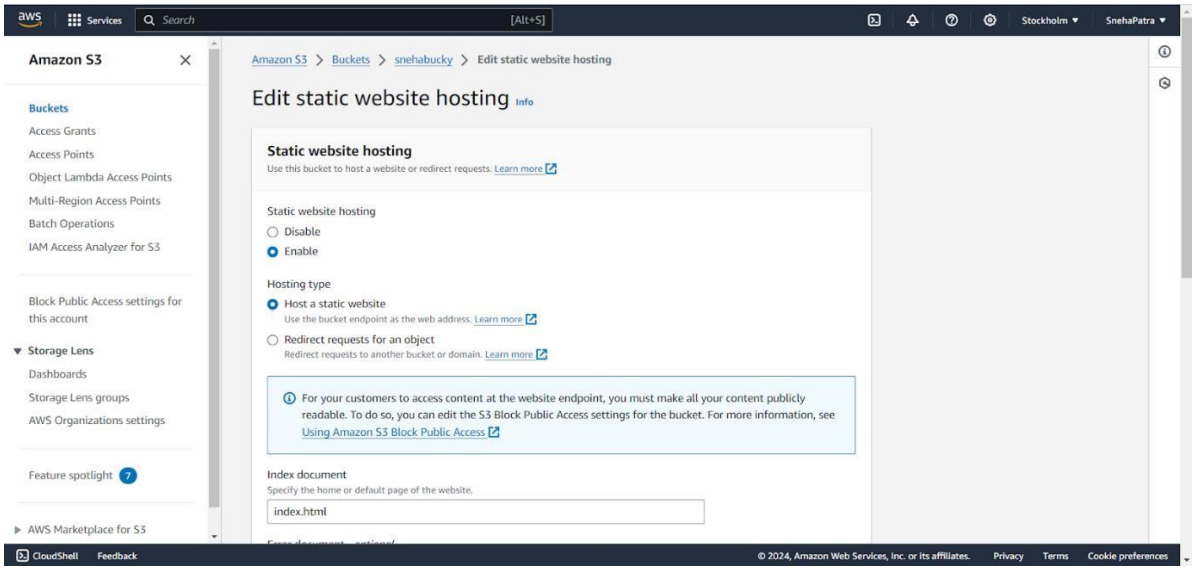
6. This is the hosted Static Website.

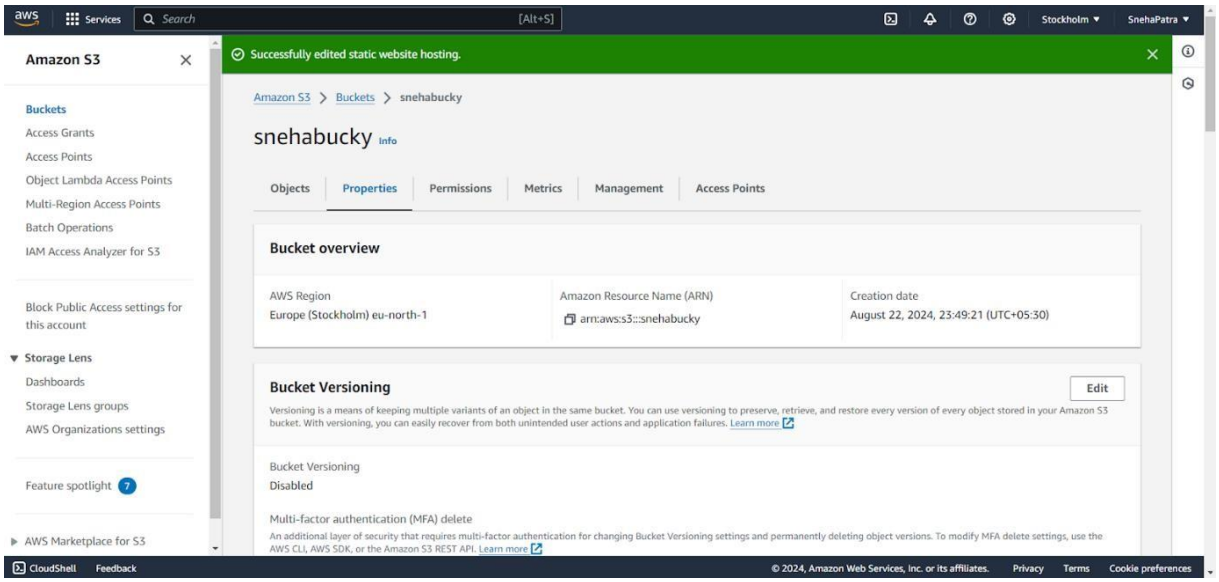
Using S3

1. Now visit S3 under the developer tools and create a Bucket and then Click on the Edit Static Website Hosting under the properties tab

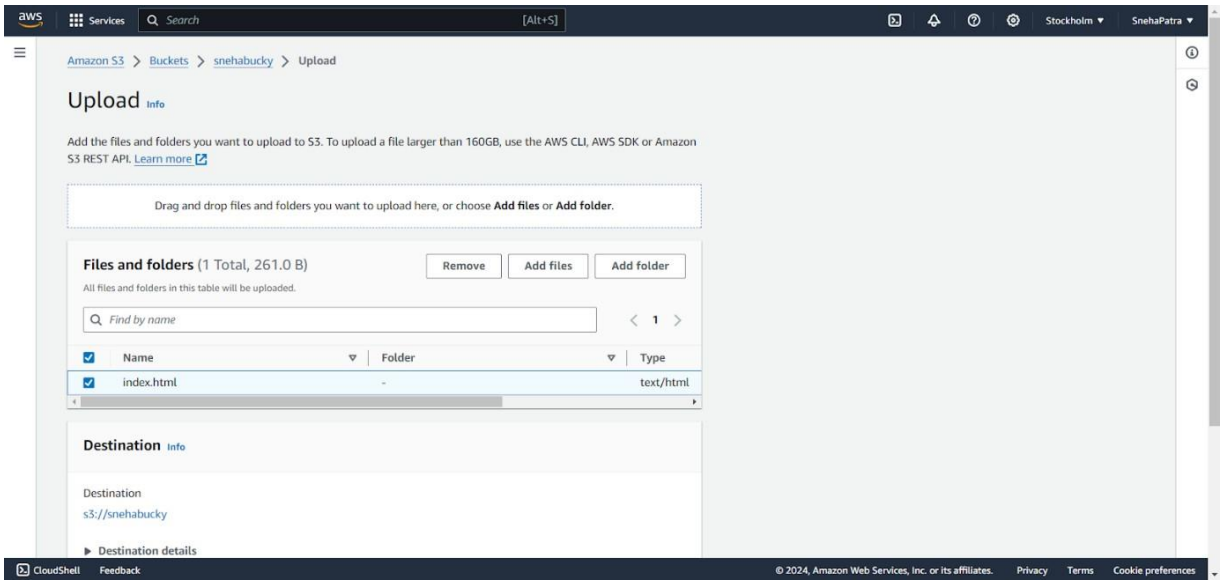

Services
Search
[Alt+S]
Stockholm
SnehaPatra
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
Europe (Stockholm) eu-north-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
snehabucky
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

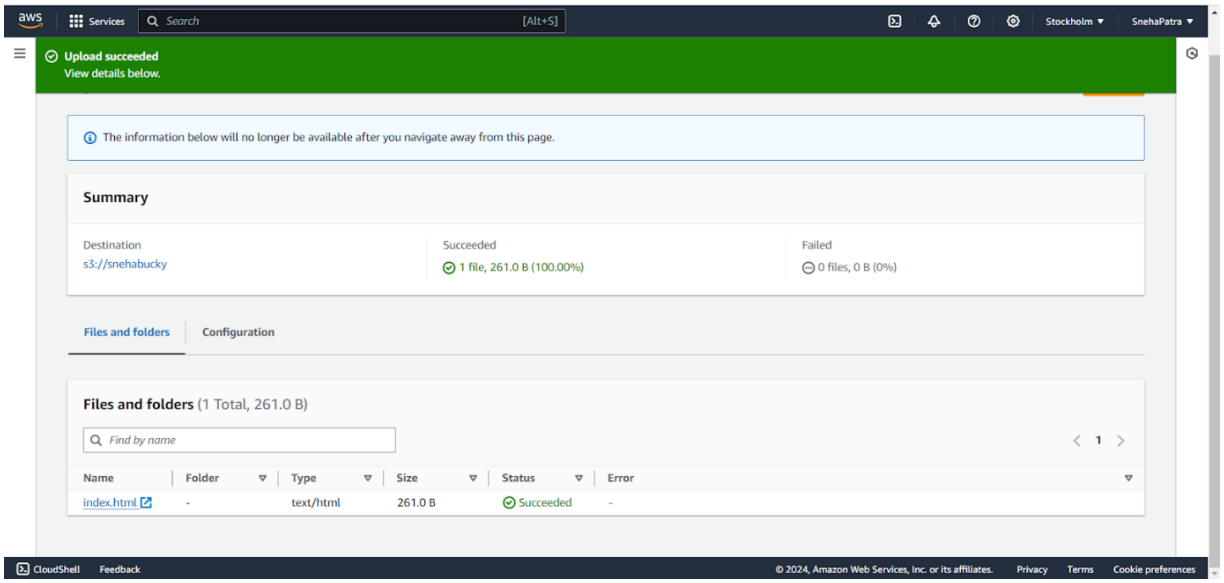

Services
Search
[Alt+S]
Stockholm
SnehaPatra
Successfully created bucket "snehabucky"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
Amazon S3 > Buckets
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (3) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
codepipeline-eu-north-1-233300349215
Europe (Stockholm) eu-north-1
View analyzer for eu-north-1
August 22, 2024, 19:54:58 (UTC+05:30)
elasticbeanstalk-eu-north-1-337909754951
Europe (Stockholm) eu-north-1
View analyzer for eu-north-1
August 22, 2024, 13:27:27 (UTC+05:30)
snehabucky
Europe (Stockholm) eu-north-1
View analyzer for eu-north-1
August 22, 2024, 23:49:21 (UTC+05:30)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Stockholm
SnehaPatra
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
Amazon S3 > Buckets > snehabucky > Edit static website hosting
Edit static website hosting Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
Index document
Specify the home or default page of the website.
index.html
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

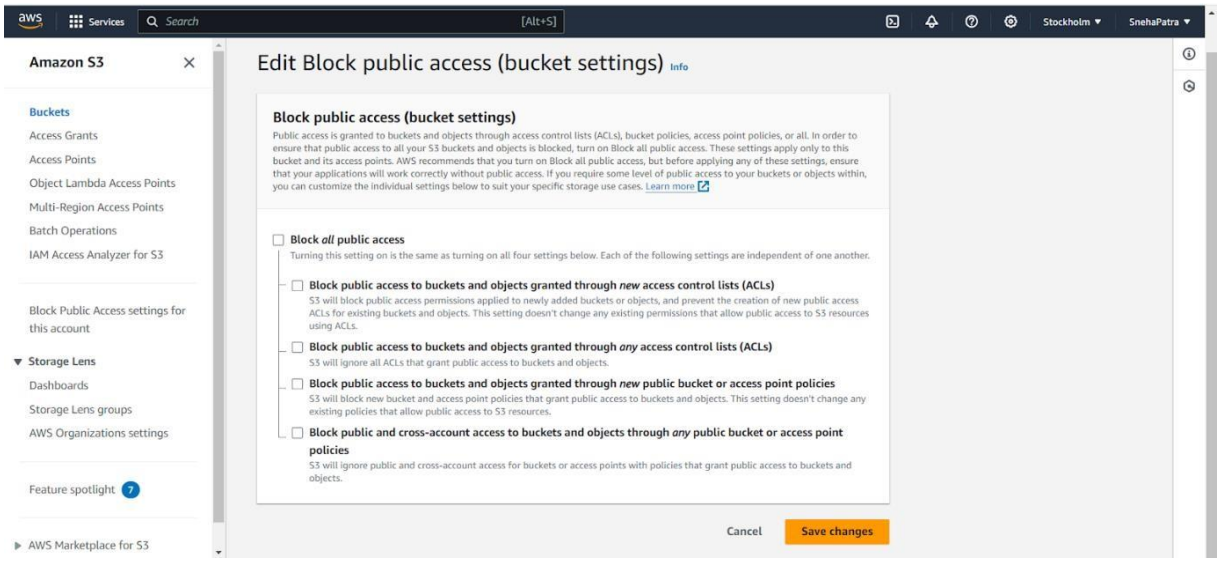


2. Upload your html file.

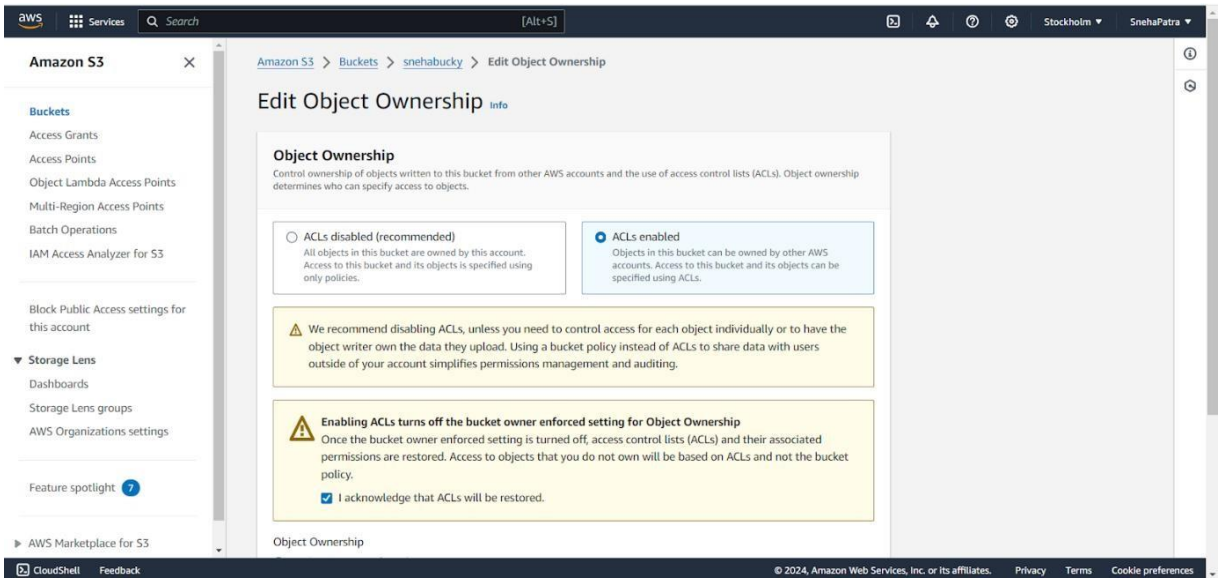




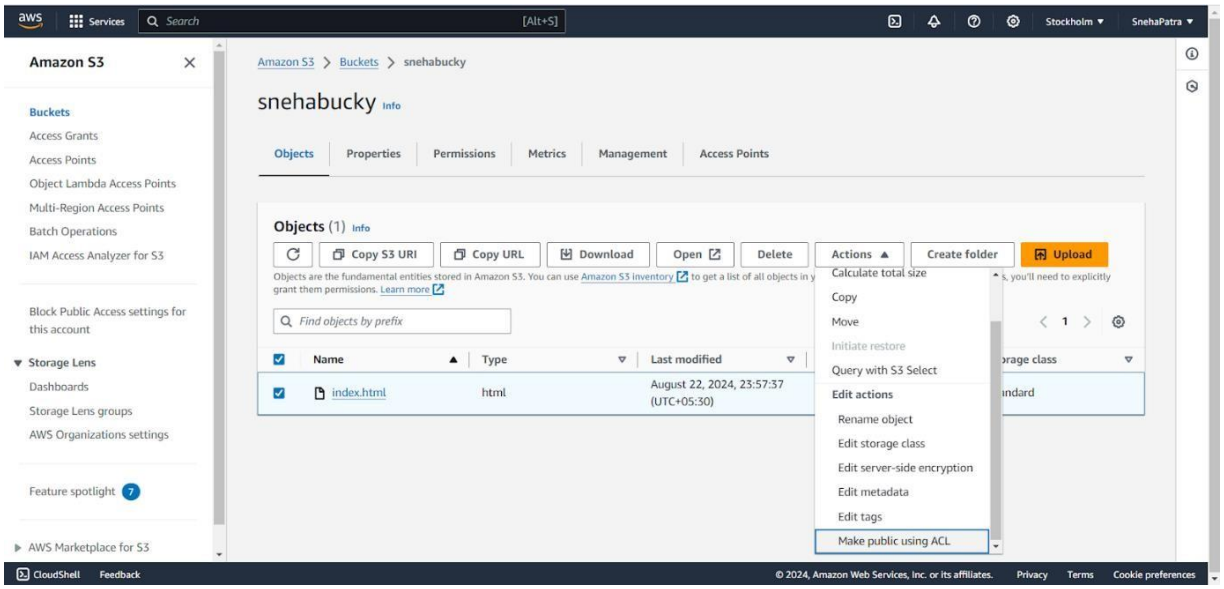
3. And then select on the Edit block public access under the Permissions tab.

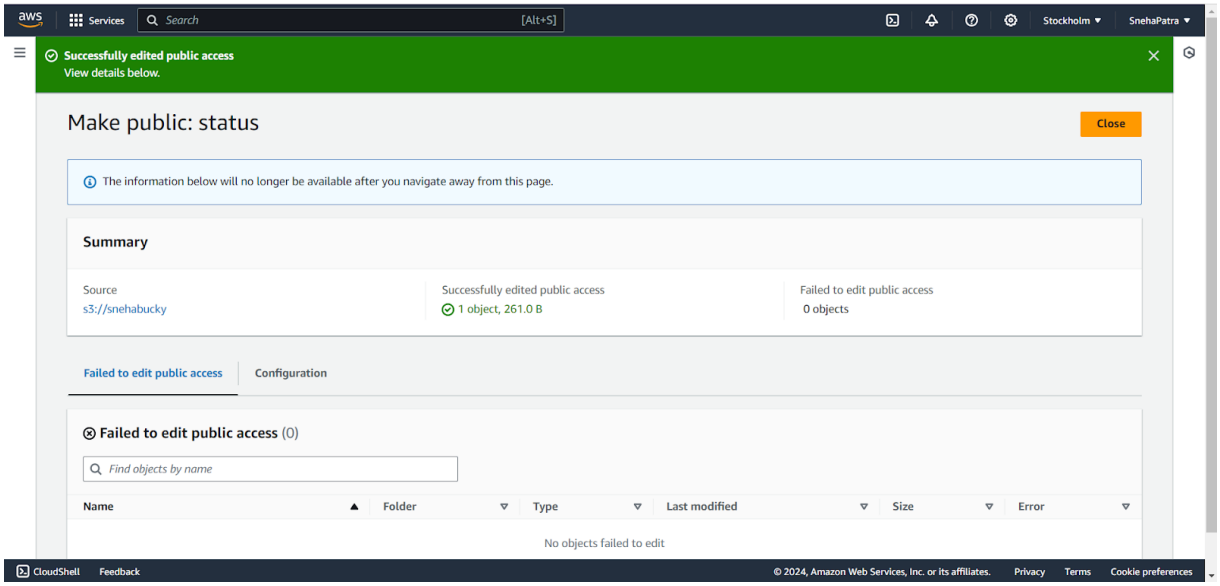


4. Select Object Ownership under Permission Tab

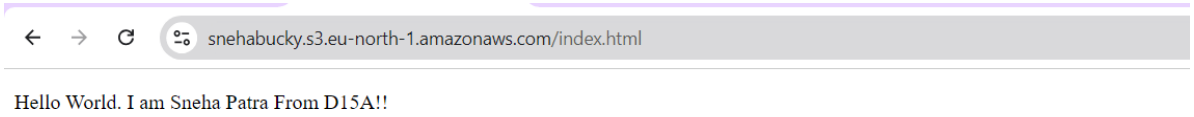


5. Select the file and click on Actions and select the option Make Public using ACL from the dropdown





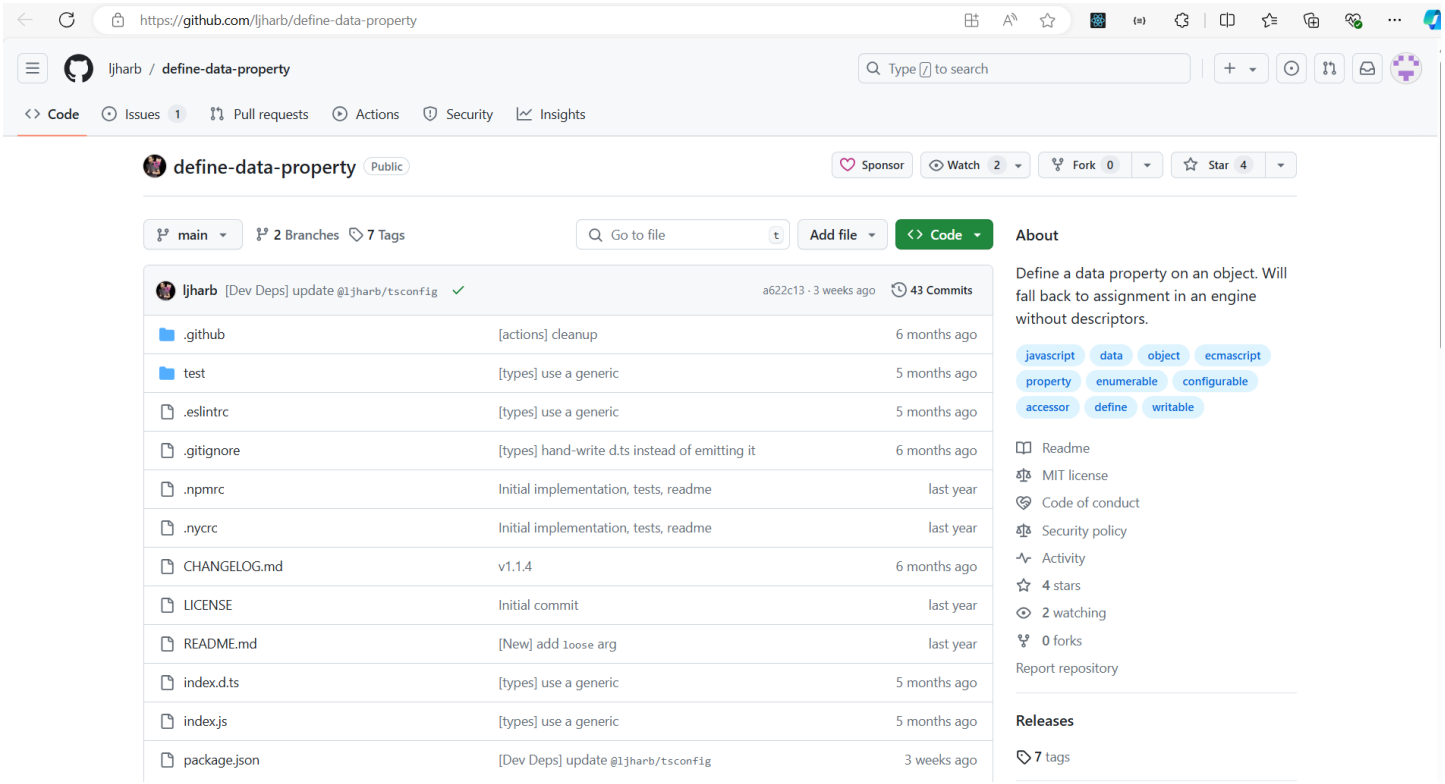
6. Now you can visit the domain and the website hosted.



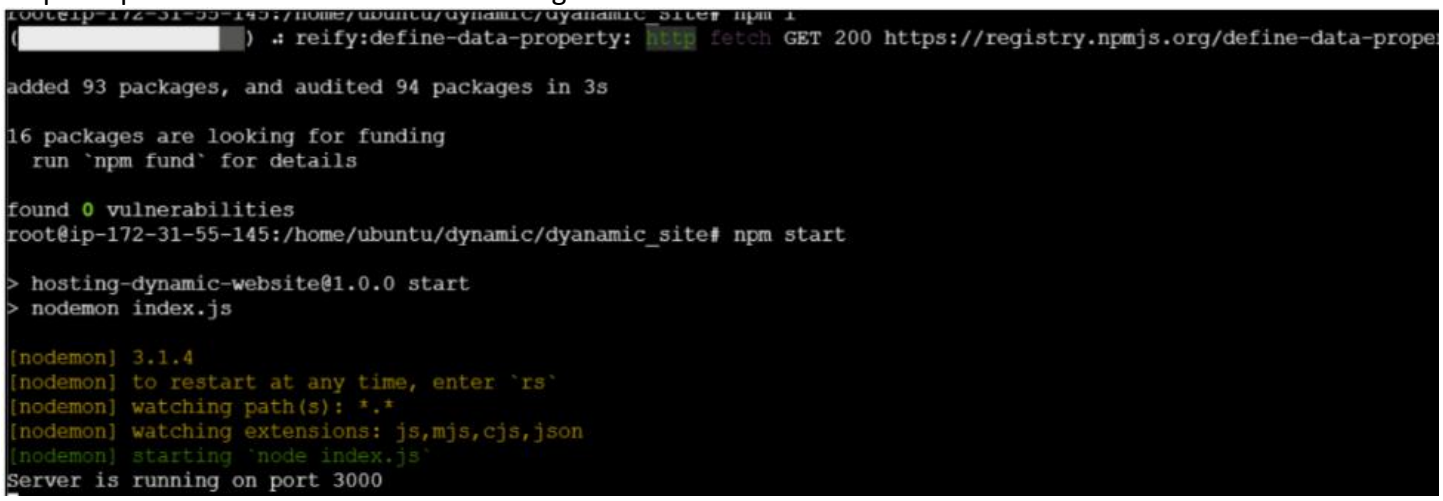
Hello World. I am Sneha Patra From D15A!!

Dynamic Hosting :

Step 1: Clone the following Github repository: https://github.com/ljharb/define-data-property



Step 2: Open Console and run the following command

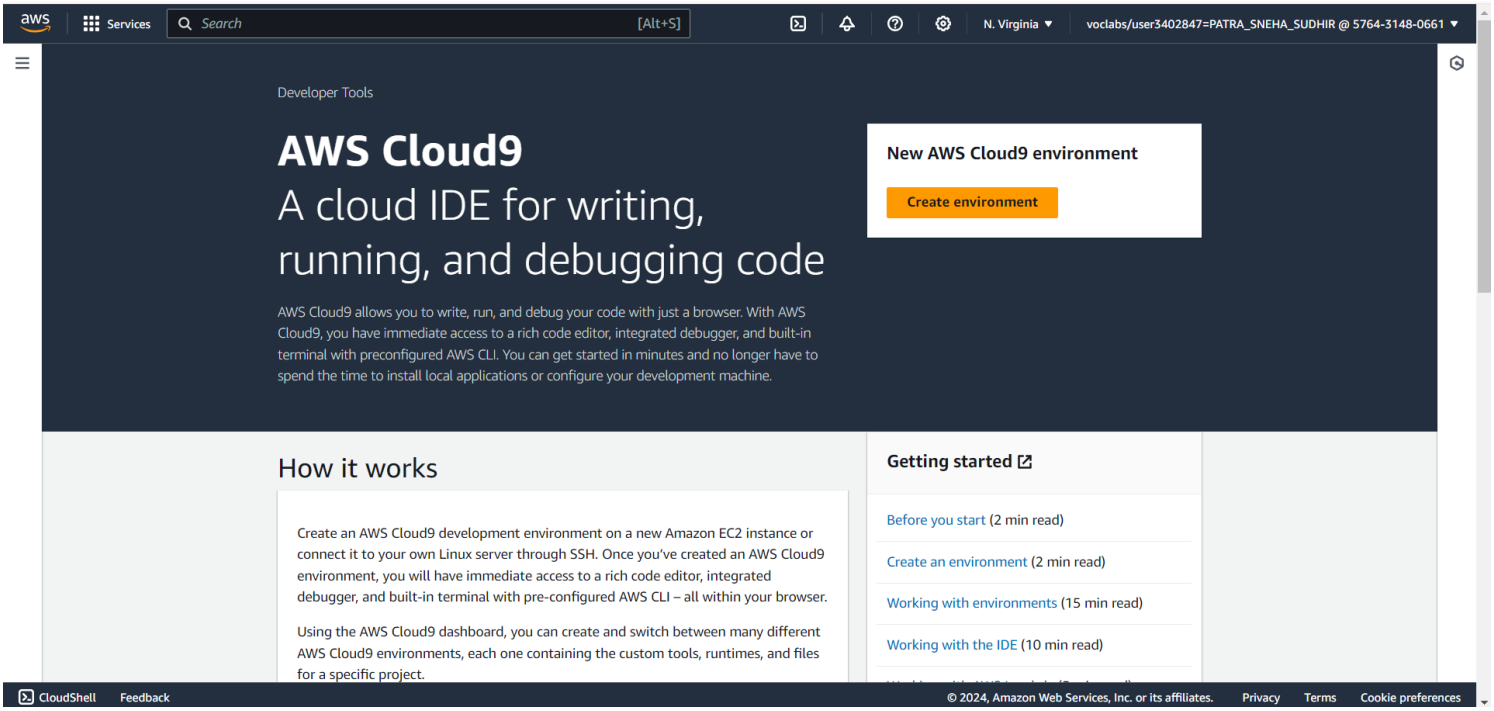


Step 3: Install necessary packages and run the website on port number 3000.

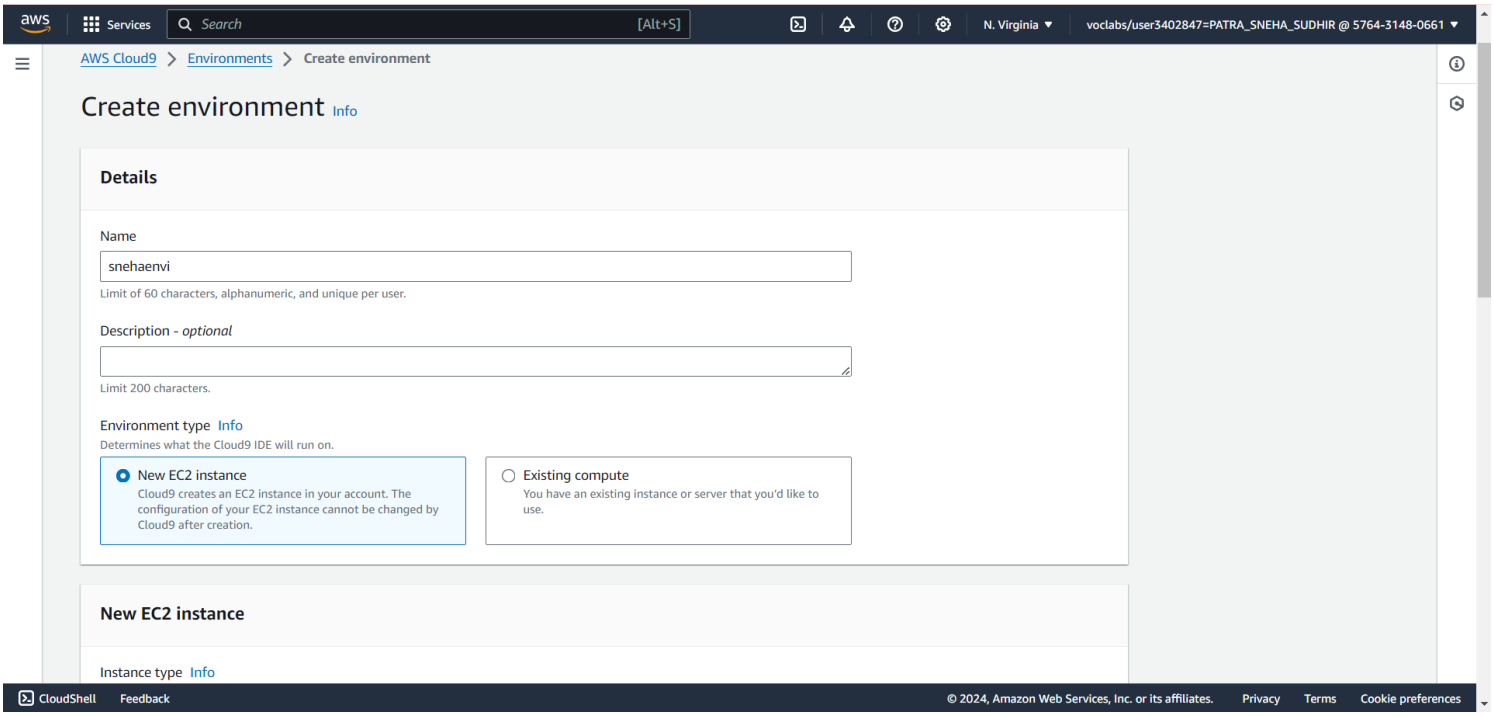
Hey this is about page.

IDE Hosting :

Step 1: Go to AWS Academy and open AWS Cloud9 from developer Tools and select create environment.



Step 2: Create a environment



30 minutes

Network settings Info

Connection

How your environment is accessed.

- (•) AWS Systems Manager (SSM)
  Accesses environment via SSM without opening inbound ports (no ingress).
- ( ) Secure Shell (SSH)
  Accesses environment directly via SSH, opens inbound ports.

▶ VPC settings Info

▶ Tags - *optional* Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
- **AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. Learn more

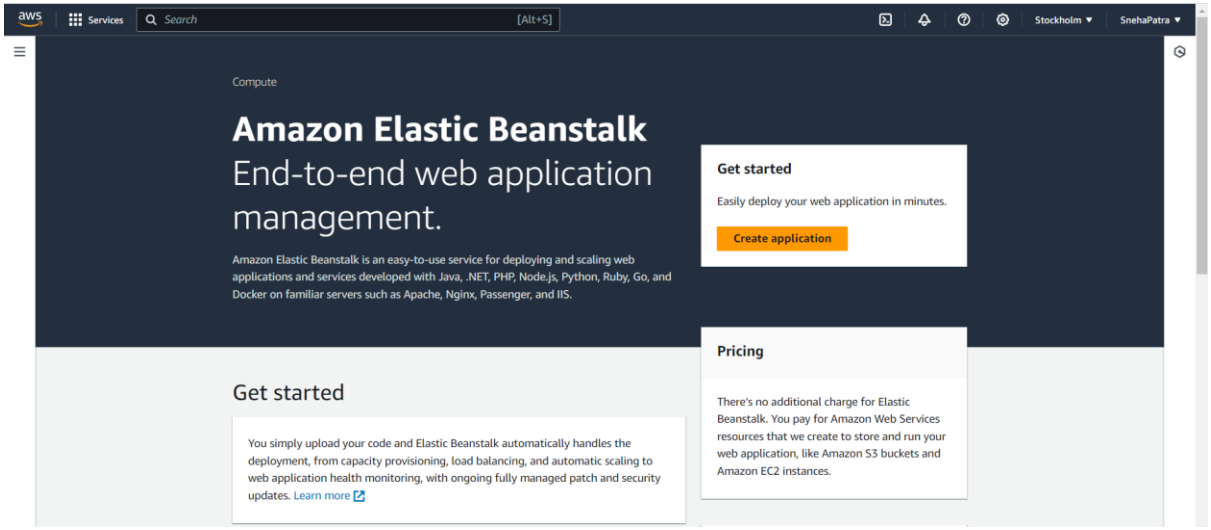
Cancel Create

Sneha Patra **D15A** **41**

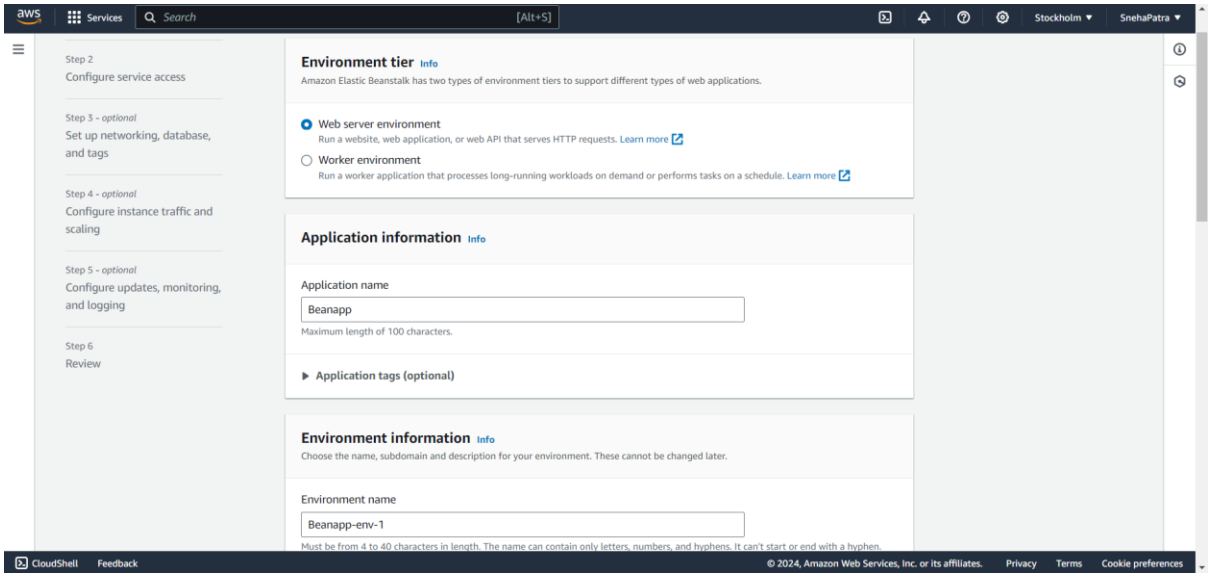
EXPERIMENT 2- BEANSTALK

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.

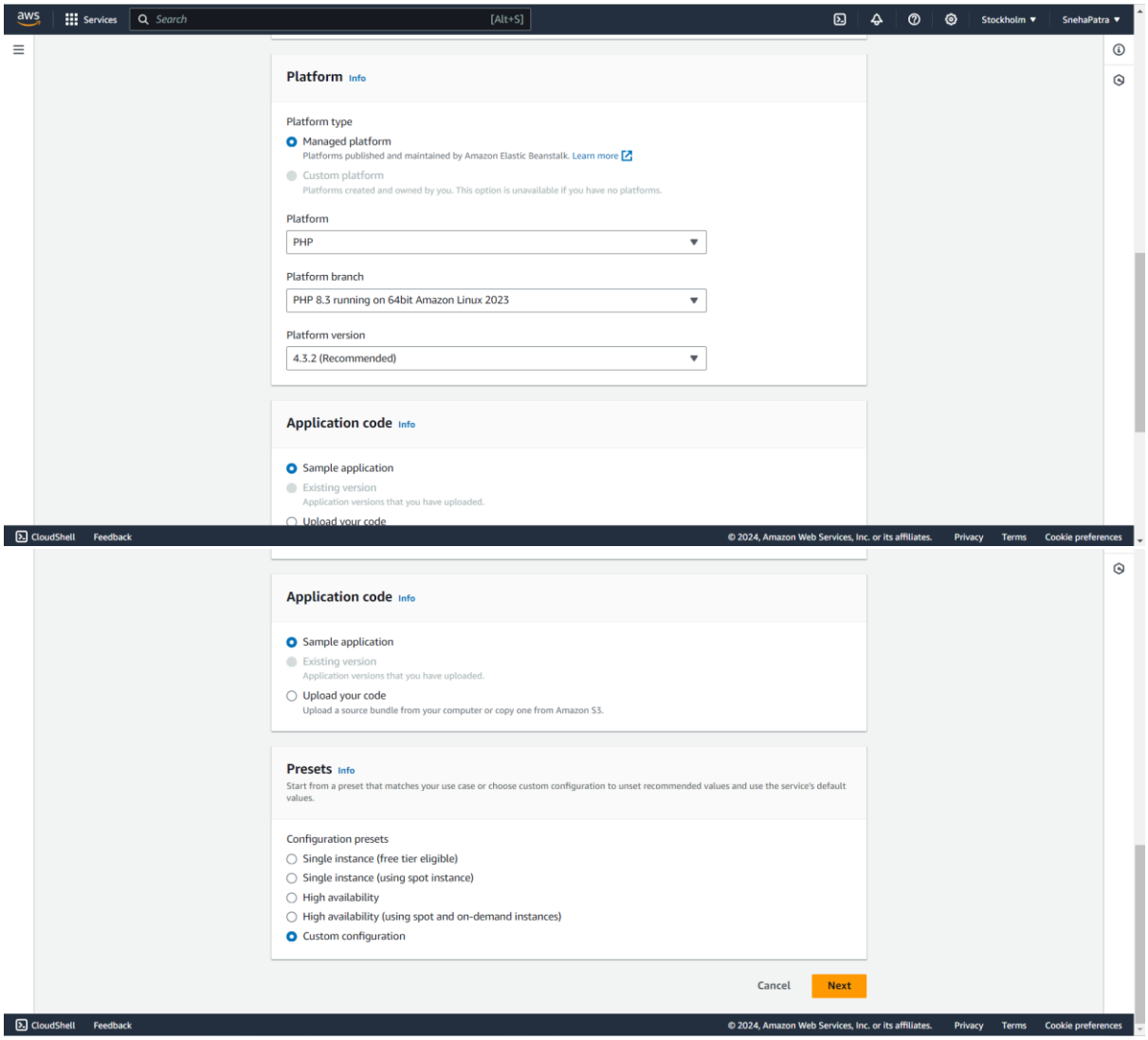
1. Open the aws console and then search Elastic Beanstalk and click on create application.



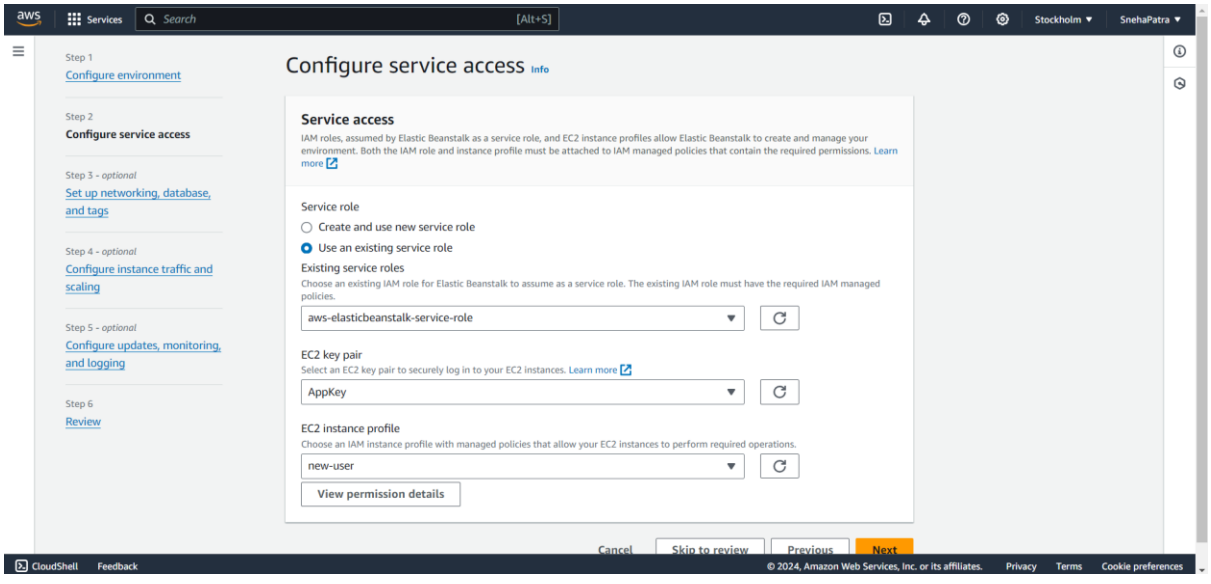
2. Configure the environment by adding your application name.



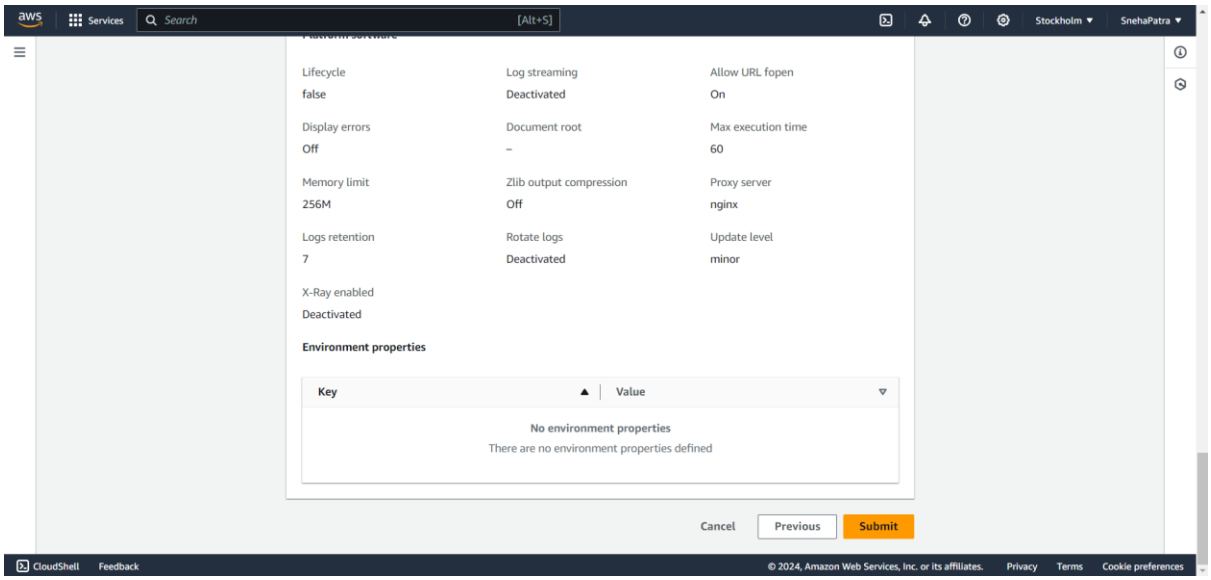
3. Choose PHP and click next.



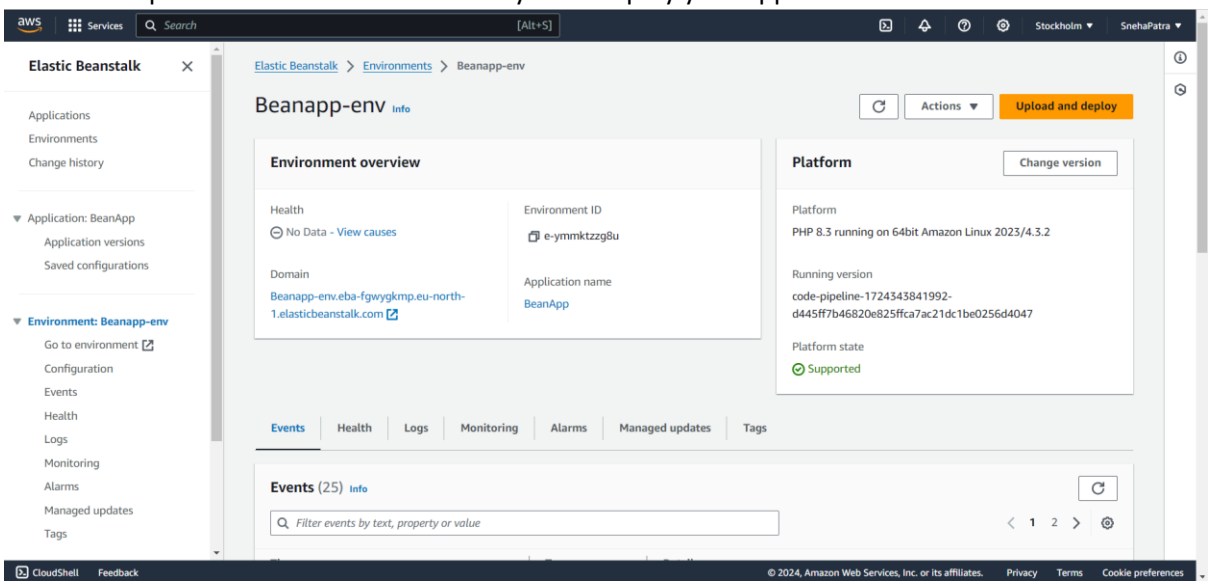
4. Add your key pair and also select EC2 instance profile.



5. After selecting skip to review then select submit.

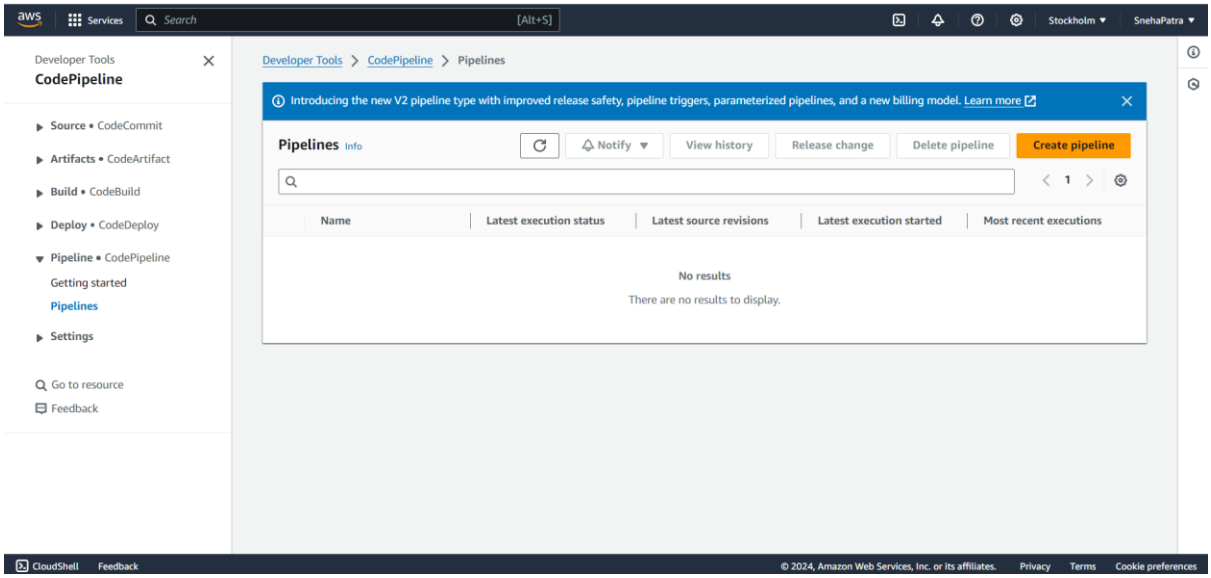


6. Your sample environment is created for you to deploy your application.

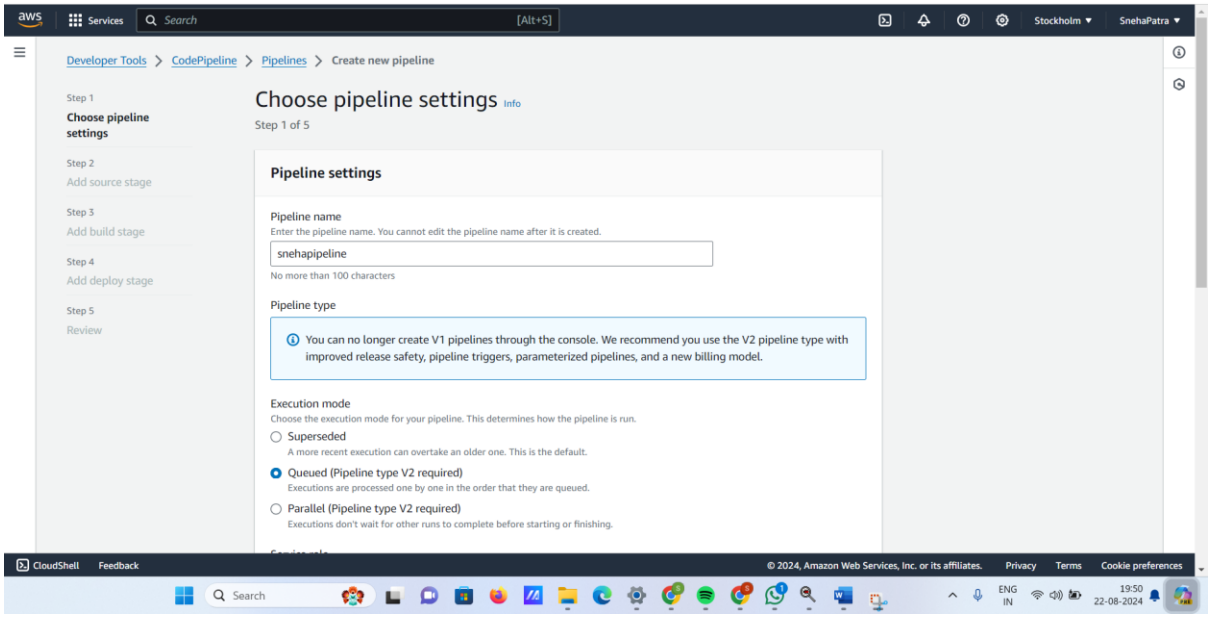


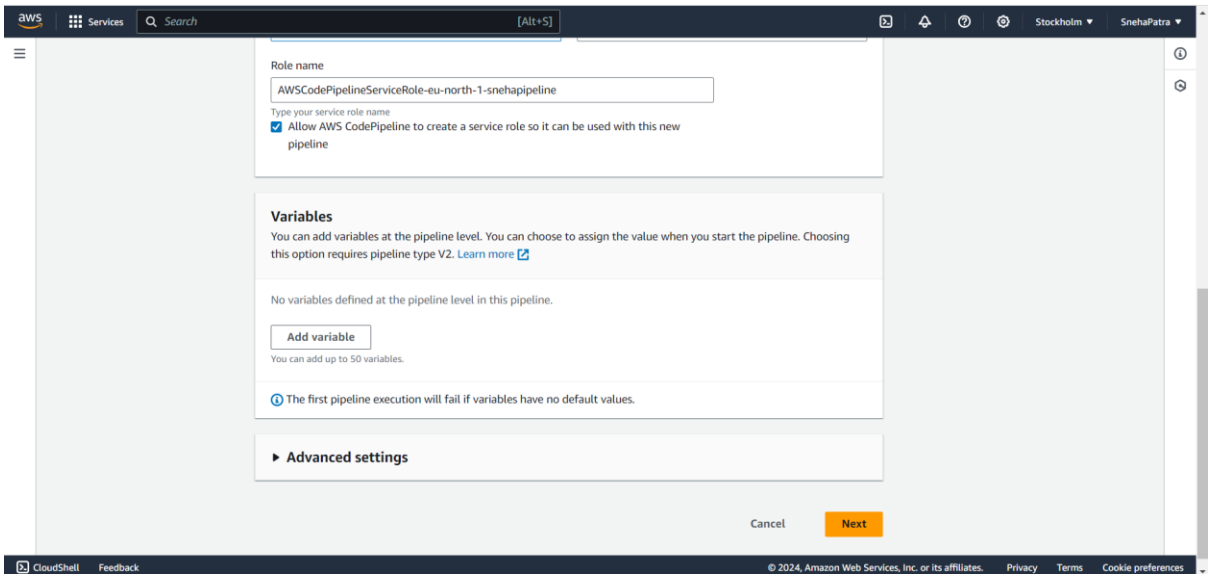
NOW LET US CREATE PIPELINE:

1.Select CodePipeline and create a new pipeline

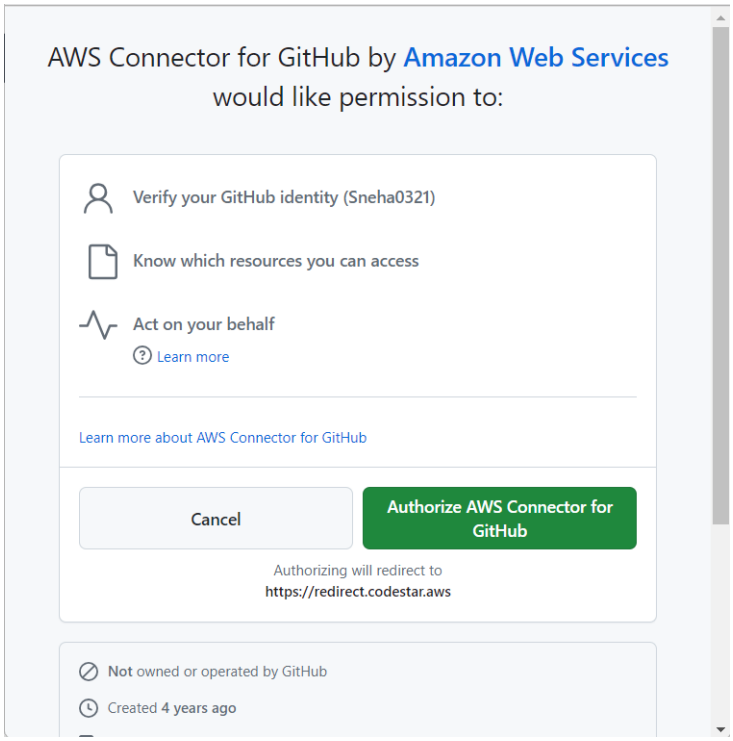


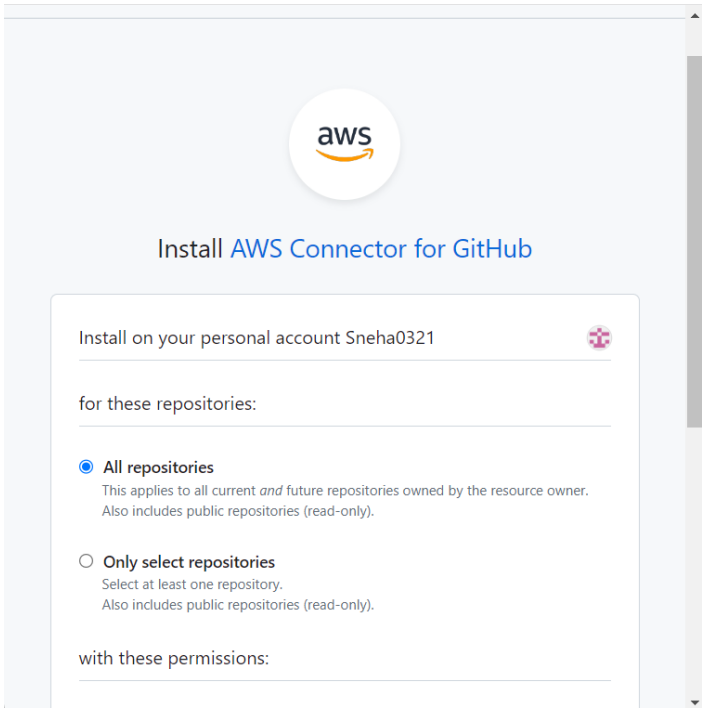
2.Name Your pipeline.

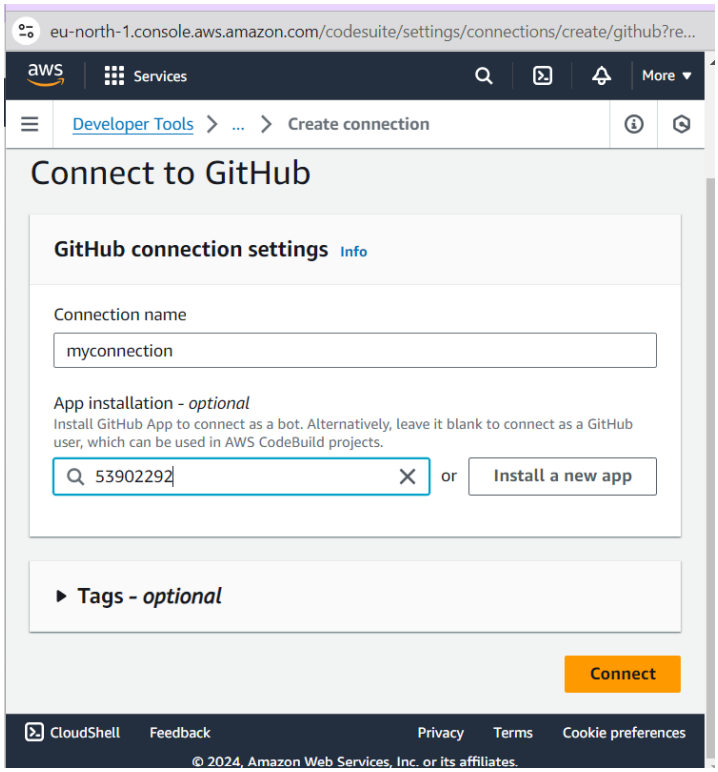




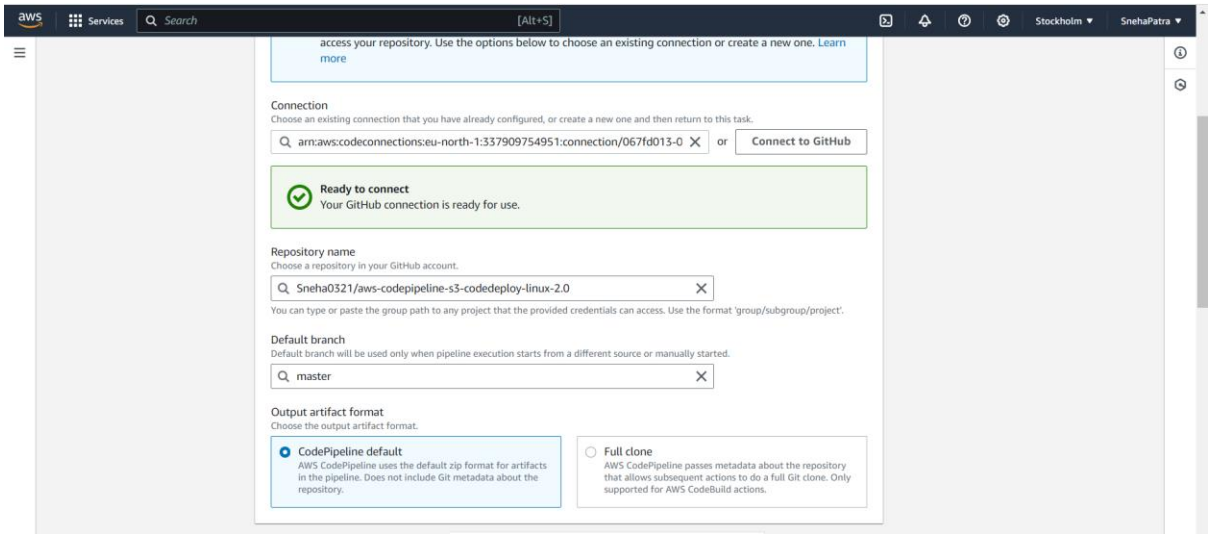
4. In the source stage select Github v2 as the provider and then connect your github.



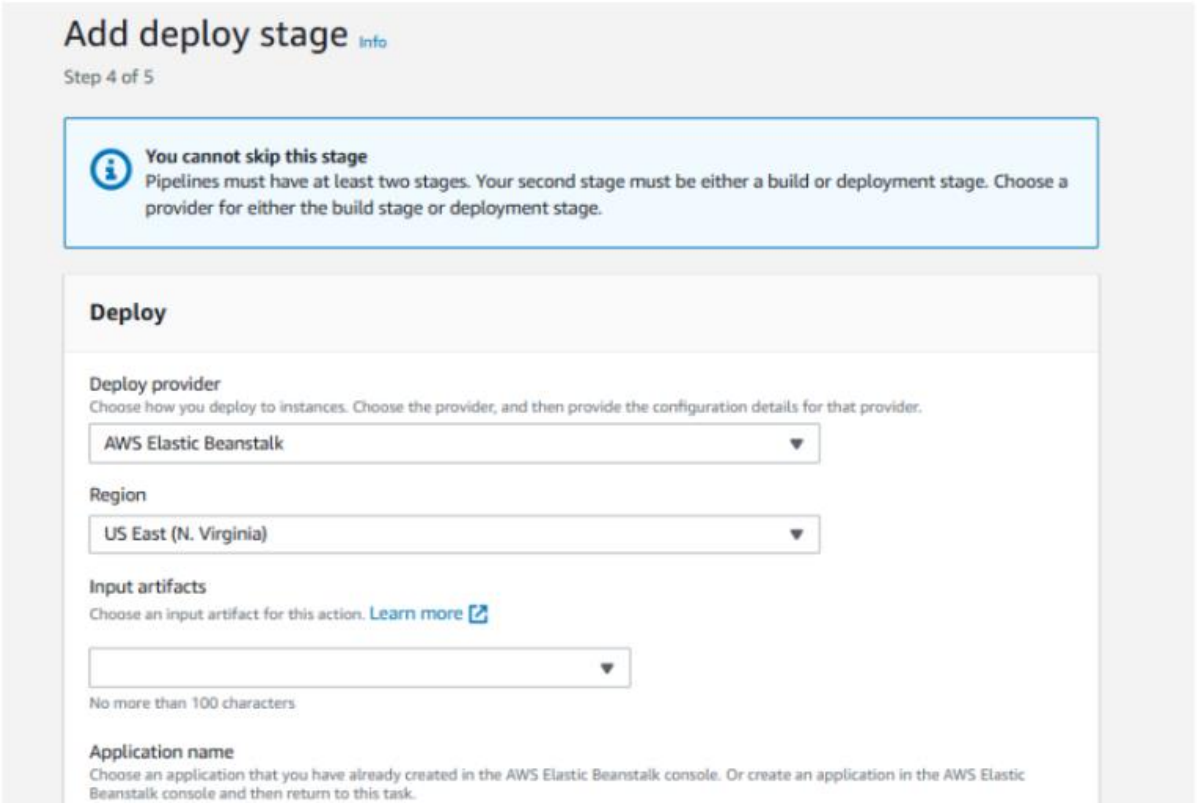

aws
Install AWS Connector for GitHub
Install on your personal account Sneha0321
for these repositories:
All repositories
This applies to all current and future repositories owned by the resource owner.
Also includes public repositories (read-only).
Only select repositories
Select at least one repository.
Also includes public repositories (read-only).
with these permissions:


eu-north-1.console.aws.amazon.com/codesuite/settings/connections/create/github?re...
aws
Services
More
Developer Tools
...
Create connection
Connect to GitHub
GitHub connection settings
Info
Connection name
myconnection
App installation - optional
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.
53902292
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.

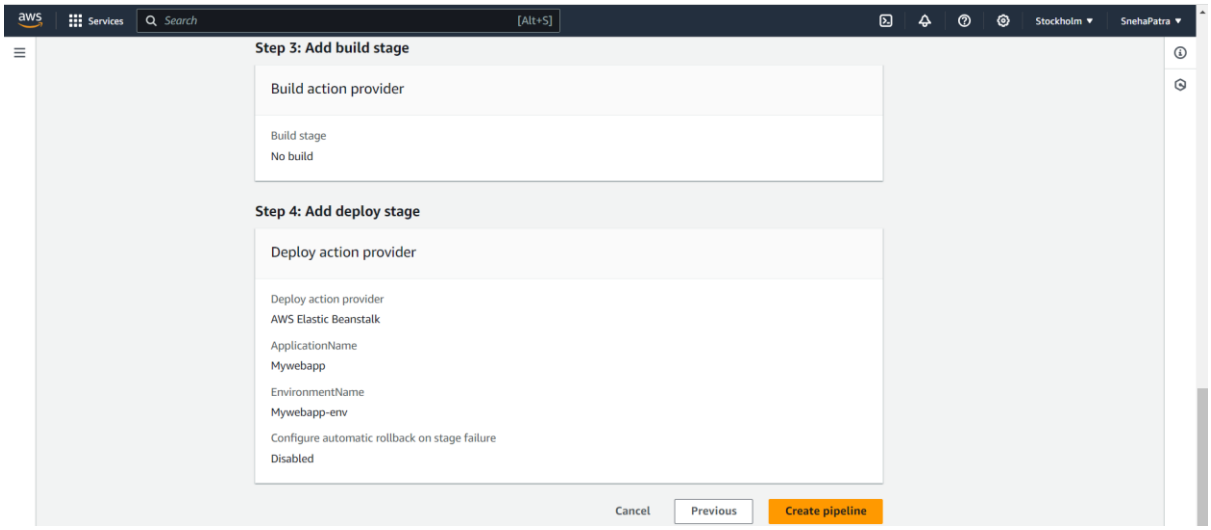
5. After establishing connection select repository and branch name.



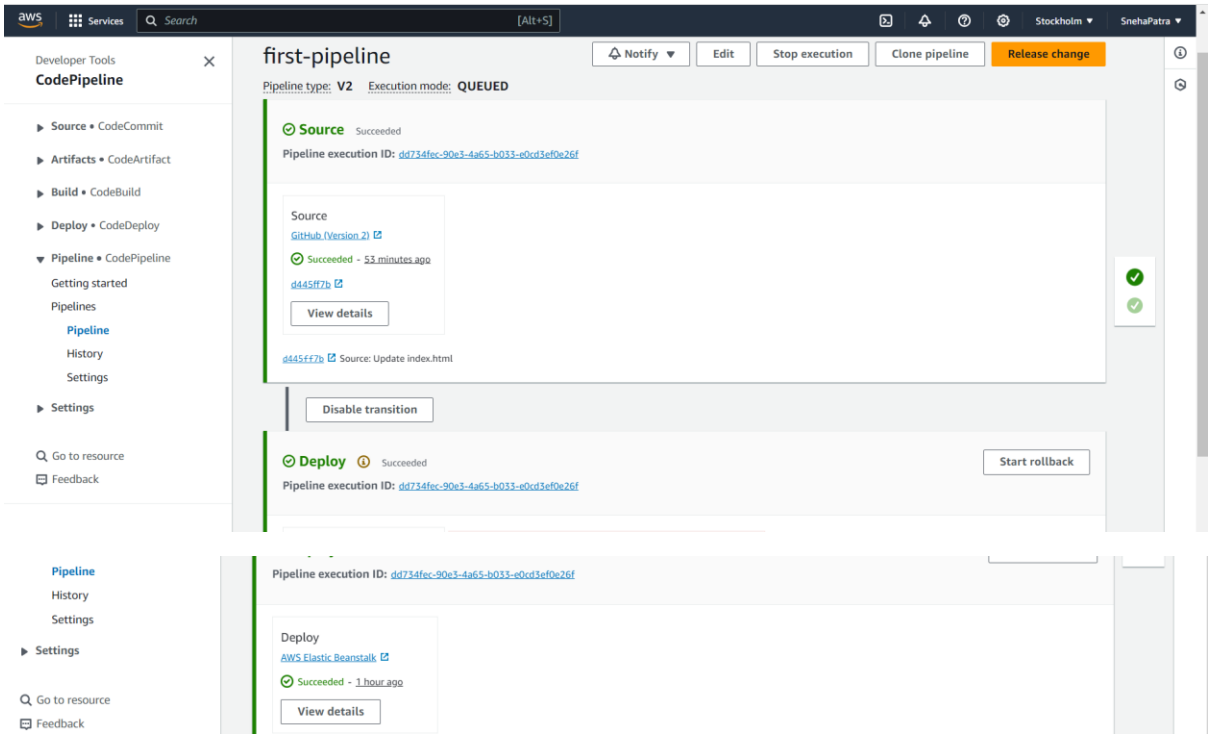
6. Add deploy stage.



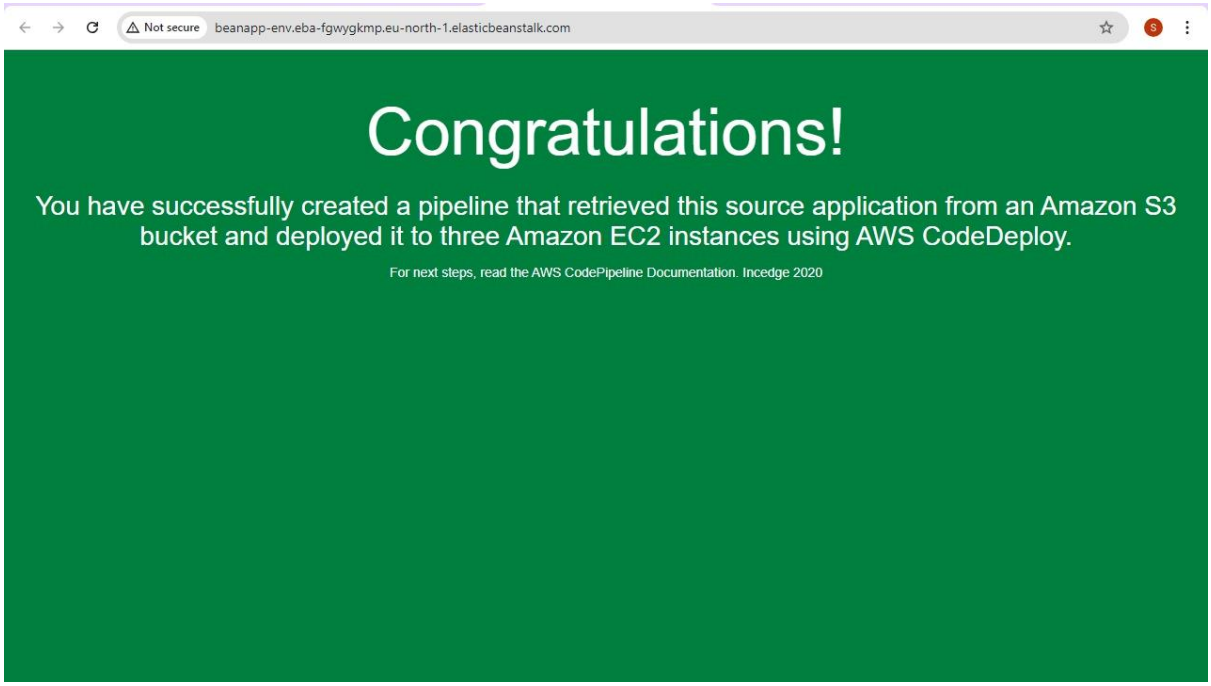
7. Review your pipeline.



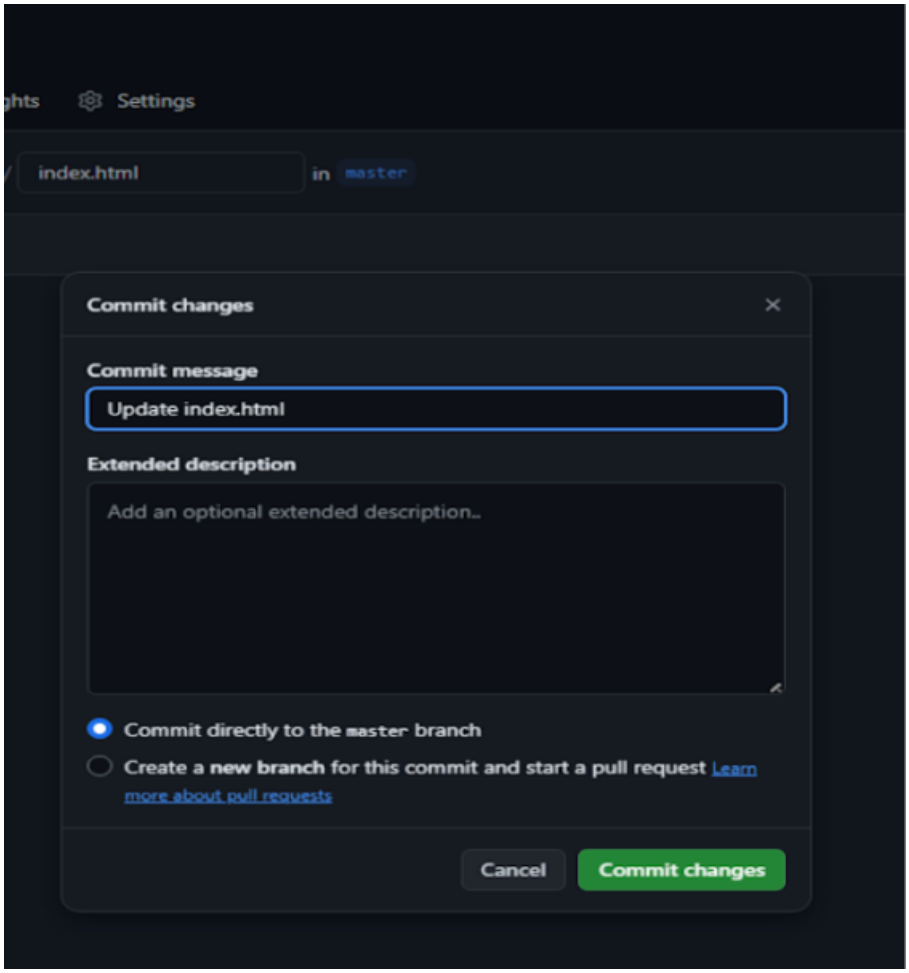
8. Now your pipeline is created.



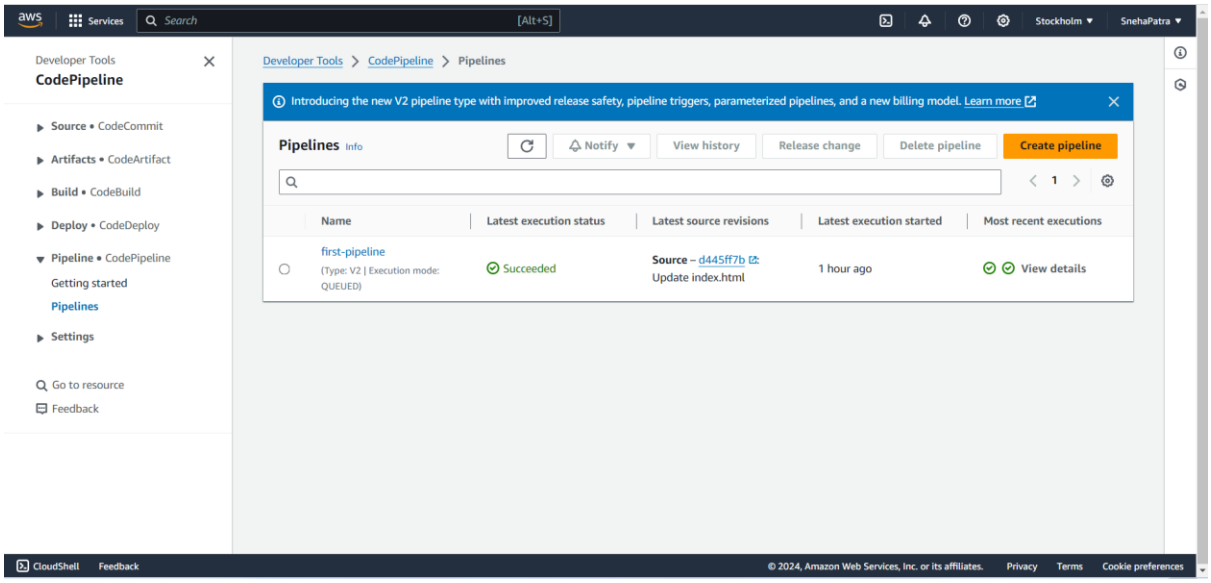
9. This is the website hosted in our beanstalk environment.

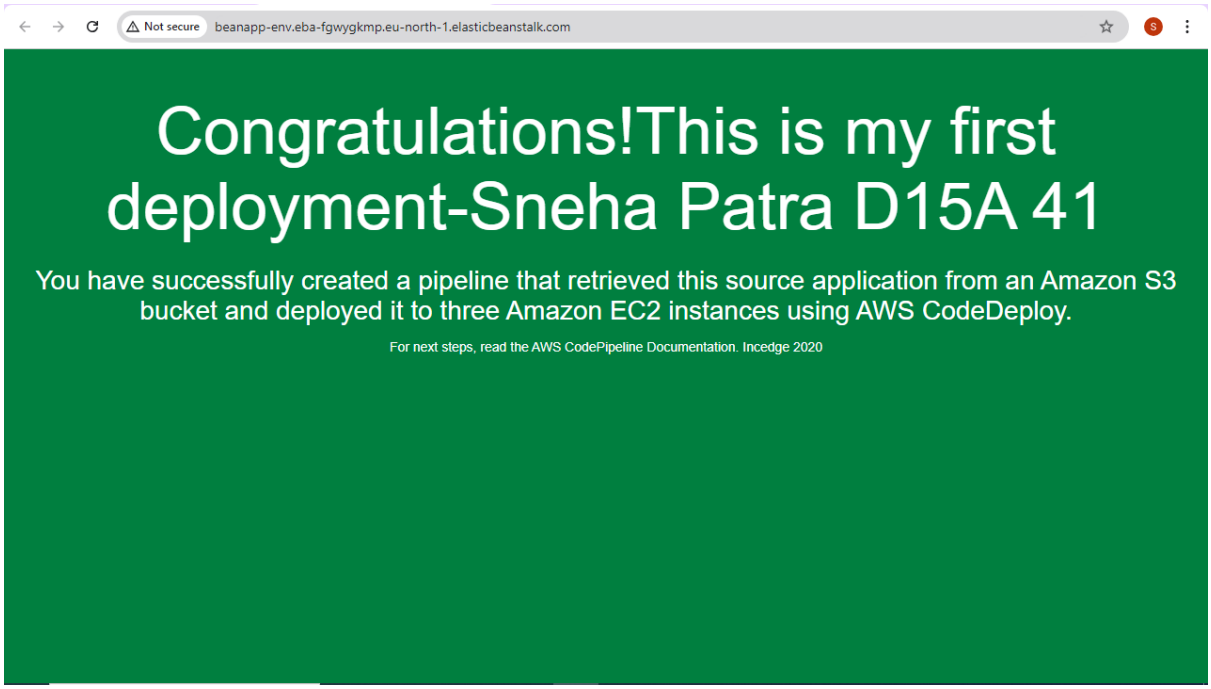


10. Go to the repository and make the changes in the index.html file and commit them



11.The changes that are committed will reflect on url once you refreshed it.

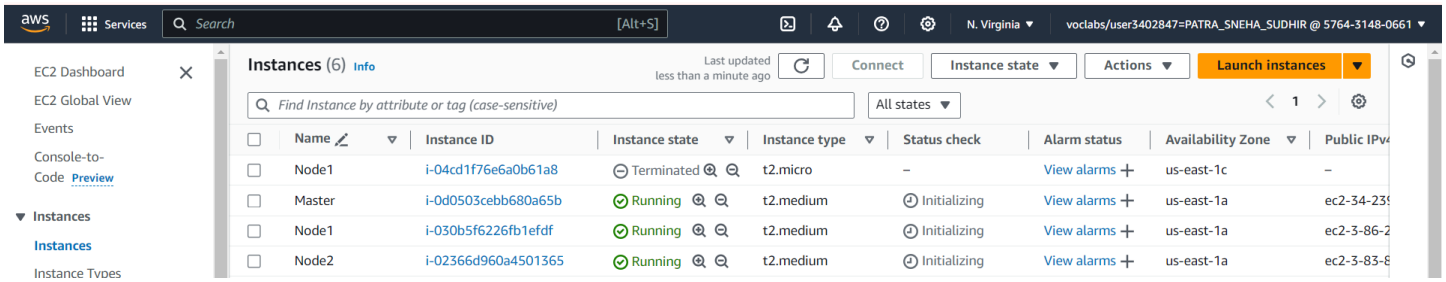


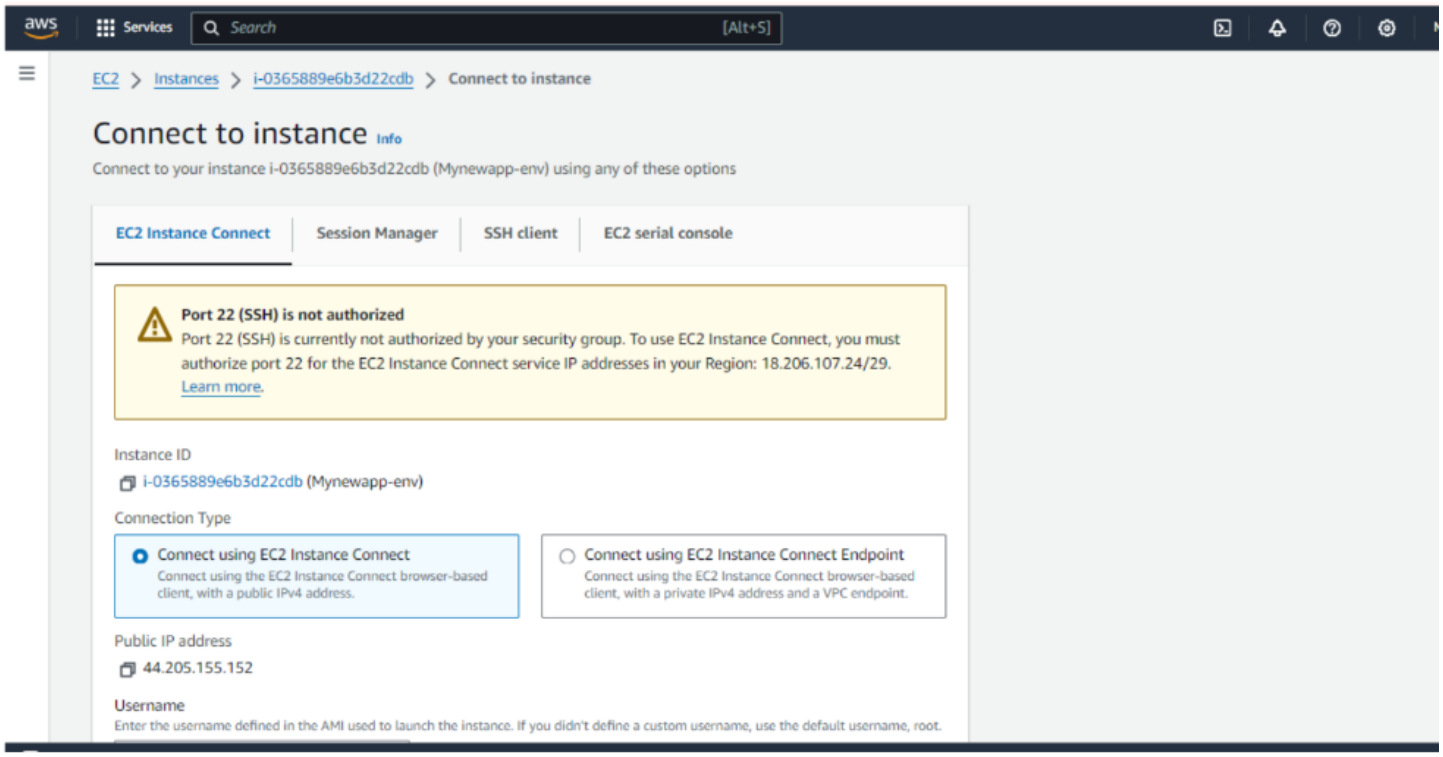


Advanced DevOps Experiment 3

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

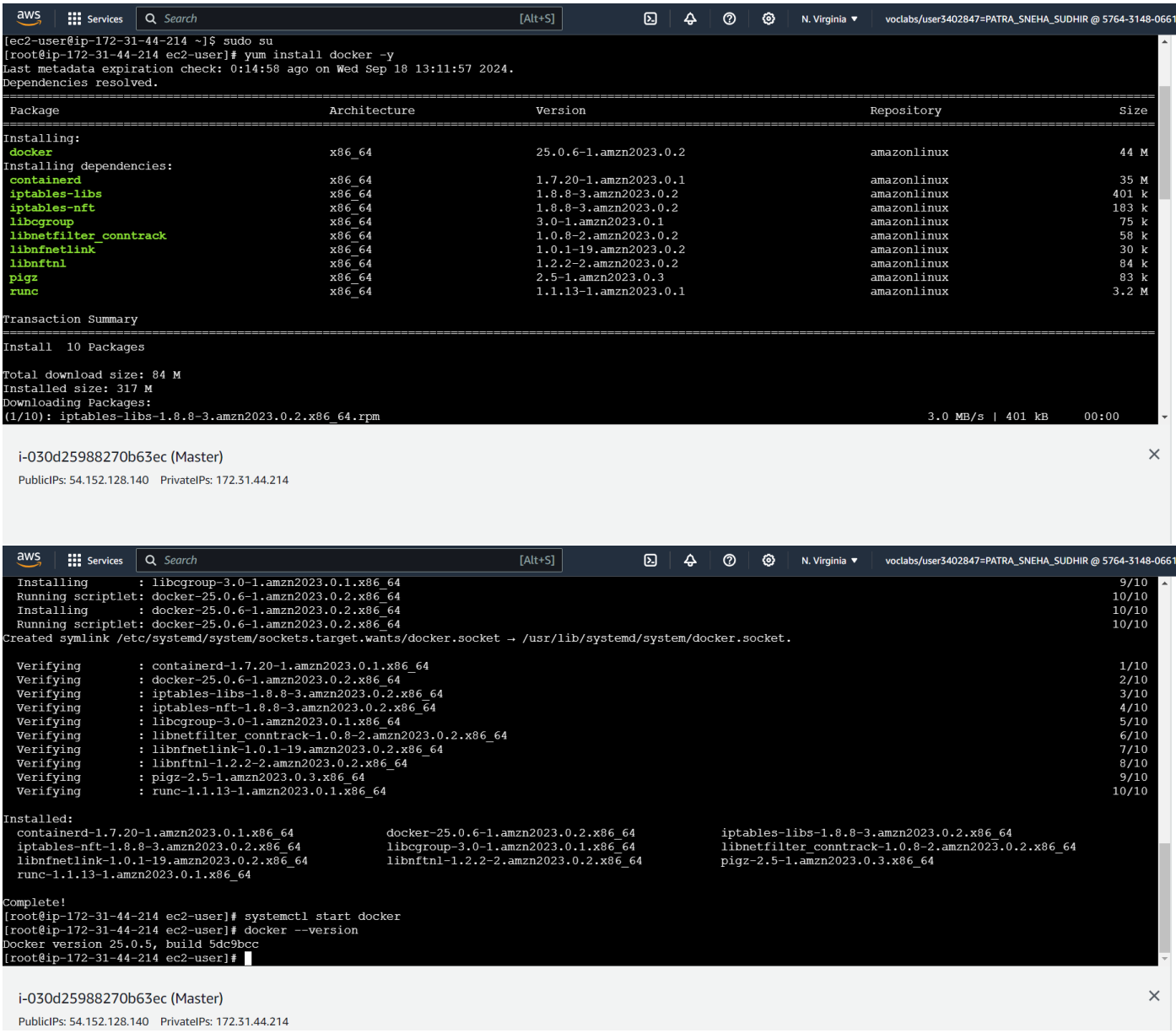
Step 1: Go to AWS Academy in services select EC2 and create 3 instance with and name them as master, node1, node2 and remember to select instance type as t2.medium.



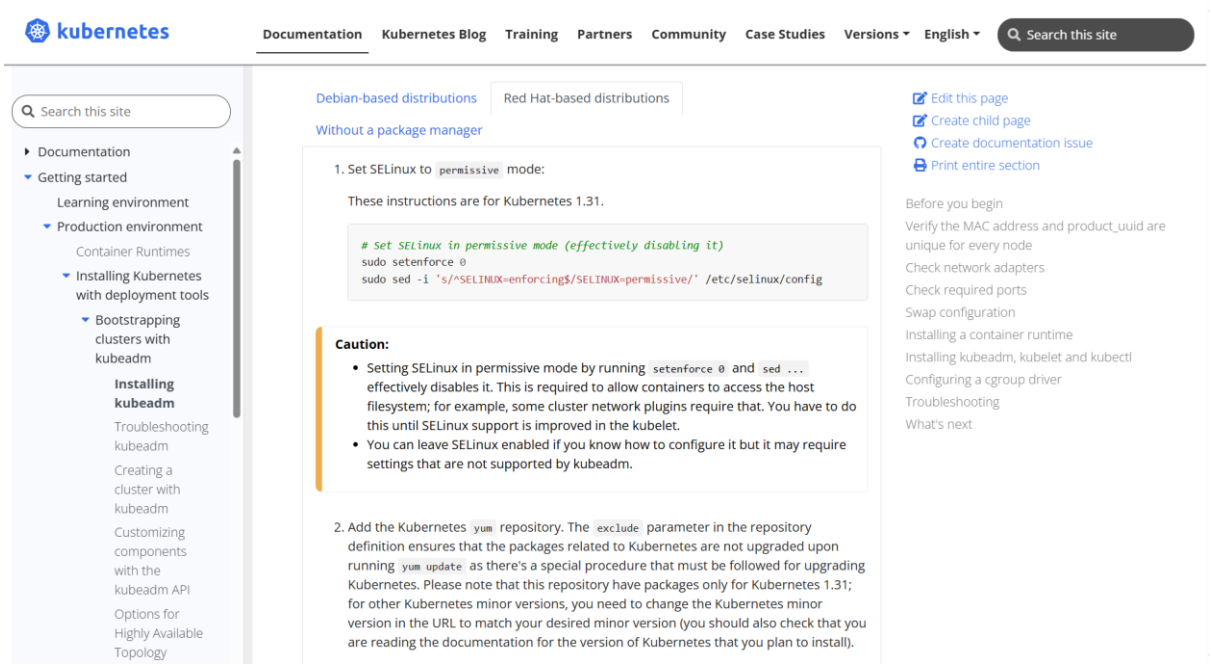


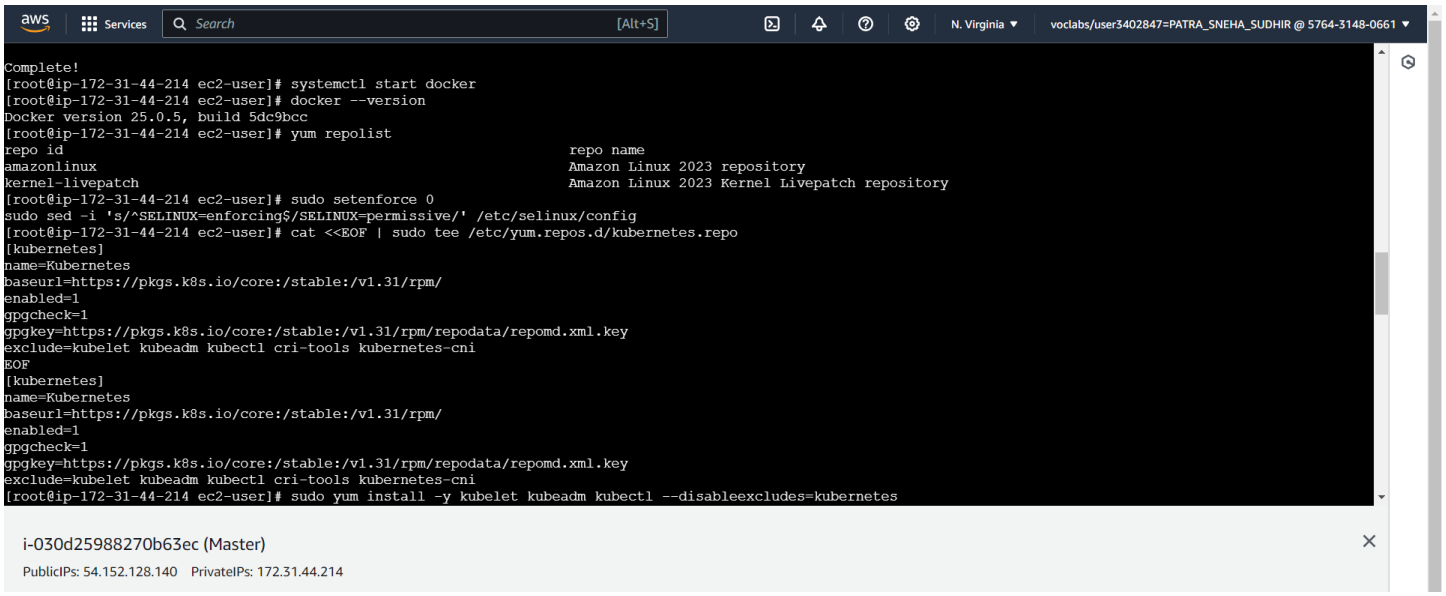
Step 2: Then, Select and connect each instance and run the following commands inside the console of each instance.

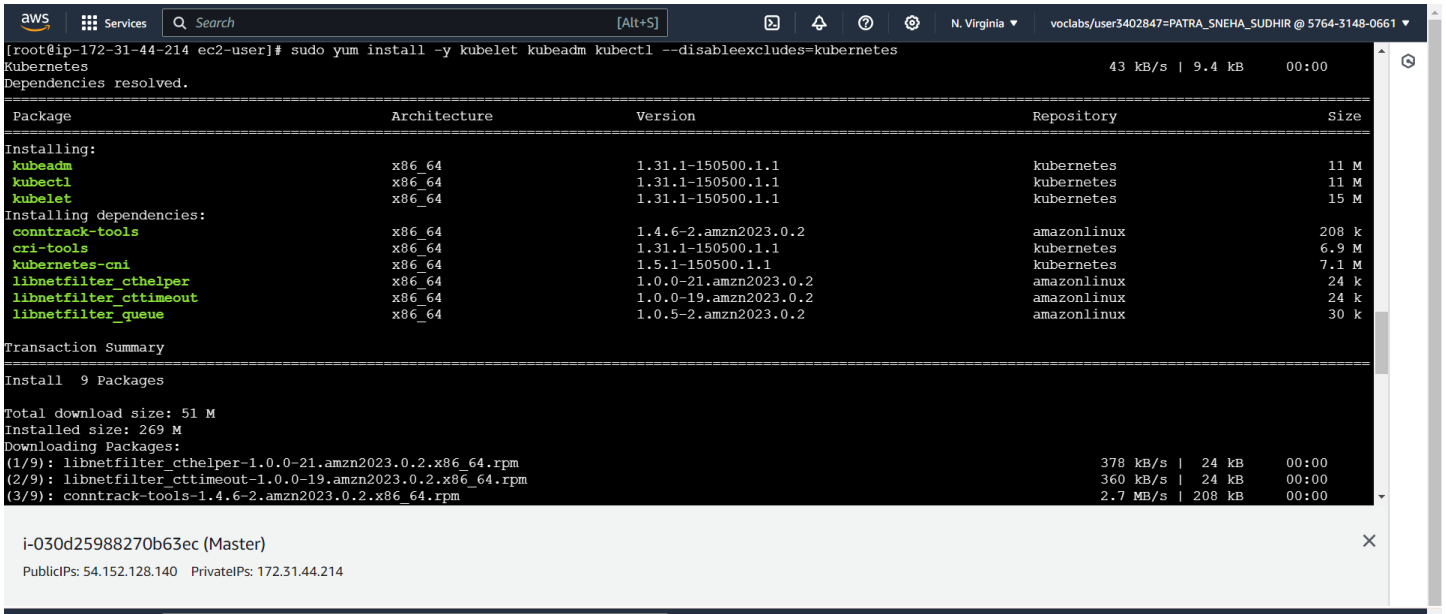
- sudo su
- yum install docker -y
- systemctl start docker
- docker –version
- yum repolist

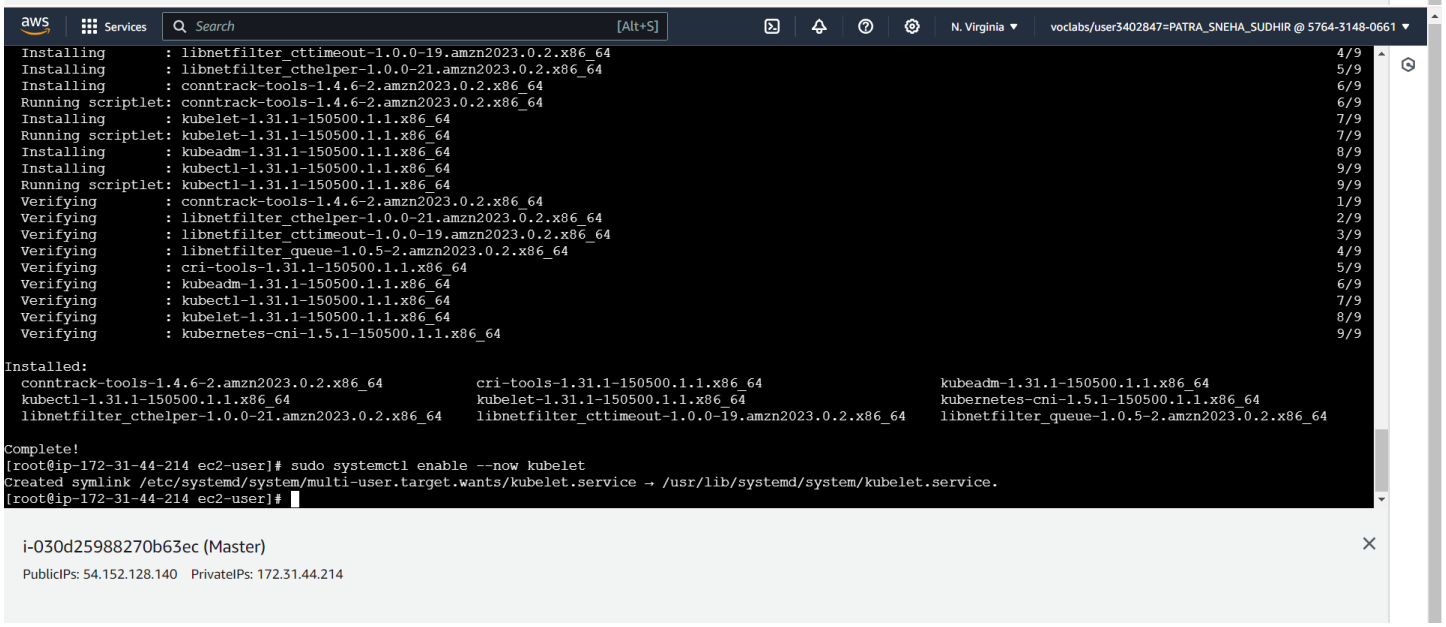


Step 3: Now, visit the following link https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/ and scroll down till you find Red-Hat and then select Red-Hat based distributions tab copy all the commands one by one in each console of instance.

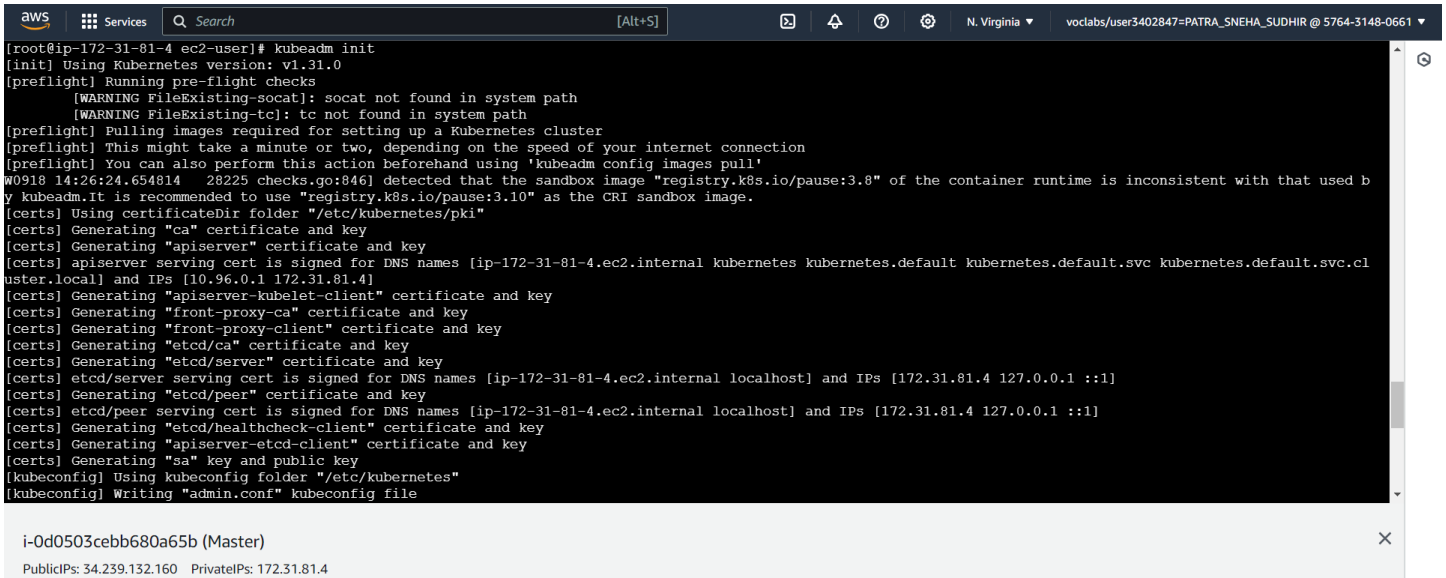

kubernetes
Documentation
Kubernetes Blog
Training
Partners
Community
Case Studies
Versions
English
Search this site
Search this site
Documentation
Getting started
Learning environment
Production environment
Container Runtimes
Installing Kubernetes with deployment tools
Bootstrapping clusters with kubeadm
Installing kubeadm
Troubleshooting kubeadm
Creating a cluster with kubeadm
Customizing components with the kubeadm API
Options for Highly Available Topology
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 settings that are not supported by kubeadm.
2. Add the Kubernetes yum repository. The exclude parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running yum update as there's a special procedure that must be followed for upgrading Kubernetes. Please note that this repository have packages only for Kubernetes 1.31; for other Kubernetes minor versions, you need to change the Kubernetes minor version in the URL to match your desired minor version (you should also check that you are reading the documentation for the version of Kubernetes that you plan to install).
Edit this page
Create child page
Create documentation issue
Print entire section
Before you begin
Verify the MAC address and product_uuid are unique for every node
Check network adapters
Check required ports
Swap configuration
Installing a container runtime
Installing kubeadm, kubelet and kubectl
Configuring a cgroup driver
Troubleshooting
What's next


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Complete!
[root@ip-172-31-44-214 ec2-user]# systemctl start docker
[root@ip-172-31-44-214 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-44-214 ec2-user]# yum repolist
repo id repo name
amazonlinux Amazon Linux 2023 repository
kernel-livepatch Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-44-214 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-44-214 ec2-user]# 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
[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
[root@ip-172-31-44-214 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
i-030d25988270b63ec (Master)
PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214



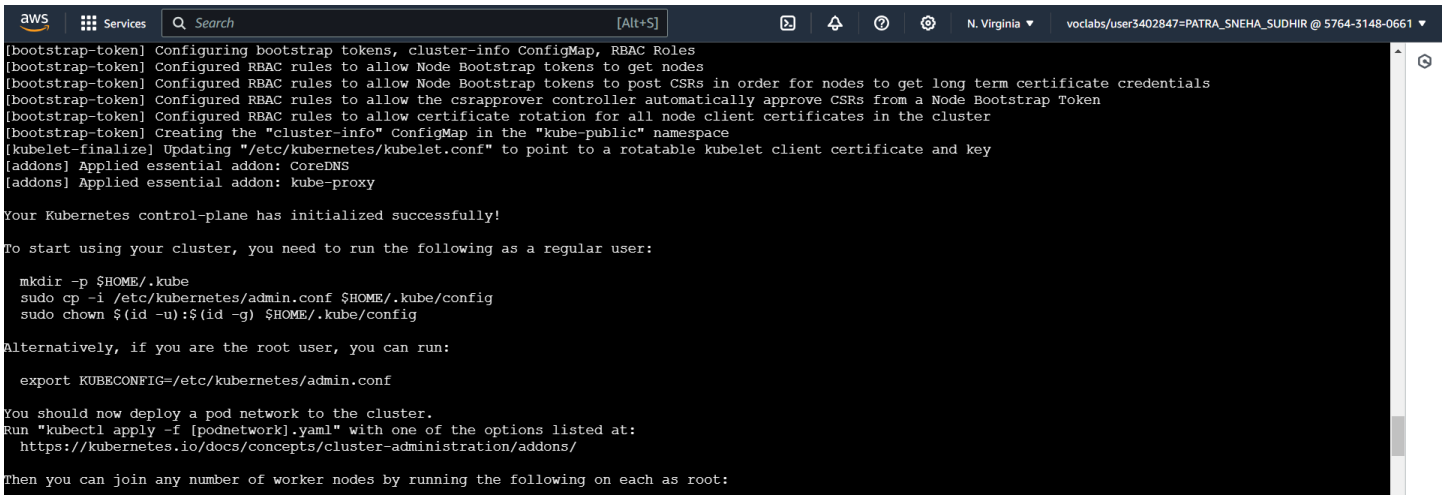


Step 4: Now, run the following command in the mater instance - kubeadm init

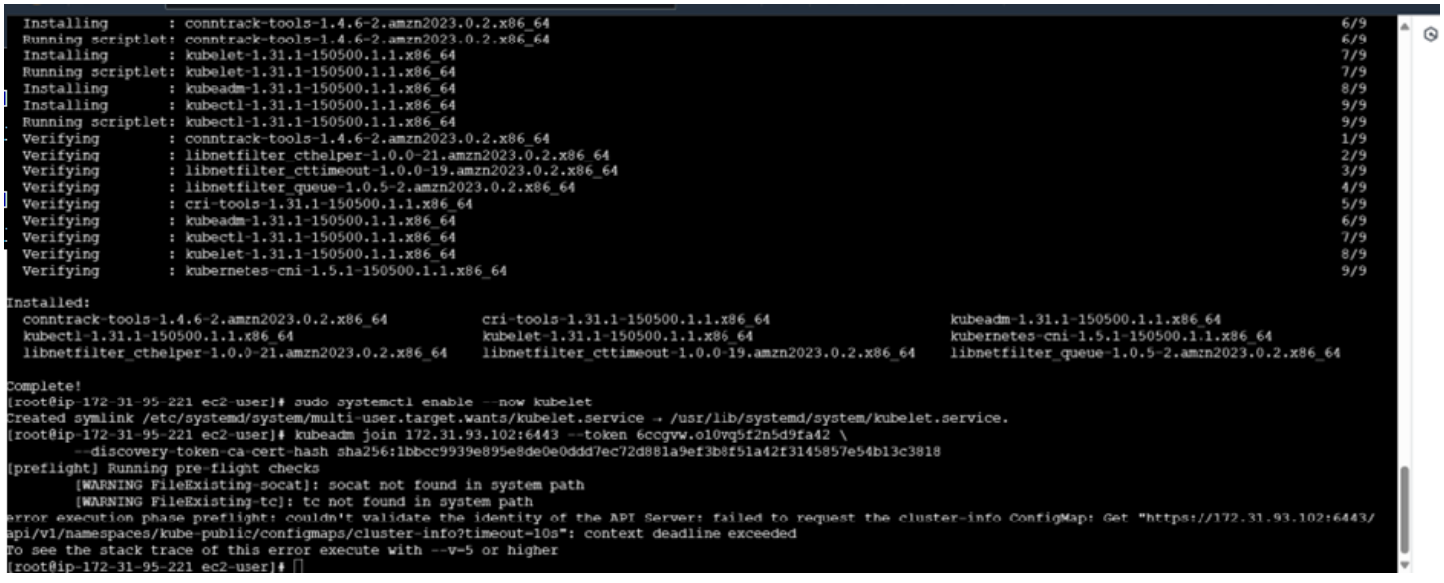


Step 5: Now, run the following commands in master instance's console –

a. mkdir -p $HOME/.kube
   sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   sudo chown $(id -u):$(id -g) $HOME/.kube/config
b. export KUBECONFIG=/etc/kubernetes/admin.conf
c. kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
   --discovery-token-ca-cert-hash
   sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818



Step 6: Run this command in node1 and node2 -
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818



```
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                  6/9
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                  6/9
  Installing       : kubelet-1.31.1-150500.1.1.x86_64                                                             7/9
  Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64                                                             7/9
  Installing       : kubeadm-1.31.1-150500.1.1.x86_64                                                             8/9
  Installing       : kubectl-1.31.1-150500.1.1.x86_64                                                             9/9
  Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64                                                             9/9
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                  1/9
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                           2/9
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64                                          3/9
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                                               4/9
  Verifying        : cri-tools-1.31.1-150500.1.1.x86_64                                                           5/9
  Verifying        : kubeadm-1.31.1-150500.1.1.x86_64                                                             6/9
  Verifying        : kubectl-1.31.1-150500.1.1.x86_64                                                             7/9
  Verifying        : kubelet-1.31.1-150500.1.1.x86_64                                                             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-172-31-94-95 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-172-31-94-95 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.ol0vq5f2n5d9fa42 \
        --discovery-token-ca-cert-hash sha256:1bbcc9939e095e8de0e0ddd7ec72d081a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.93.102:6443/
api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-94-95 ec2-user]#
```

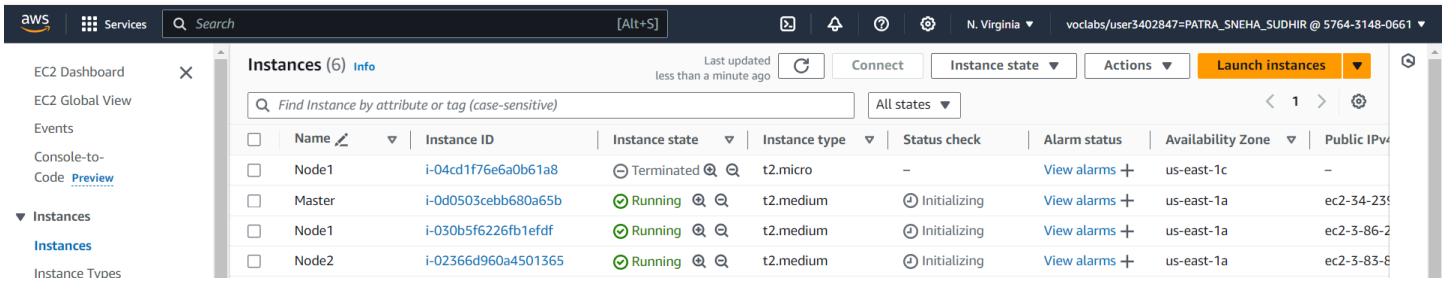
Step 7: Run the following command in master instance console -
kubectl get nodes

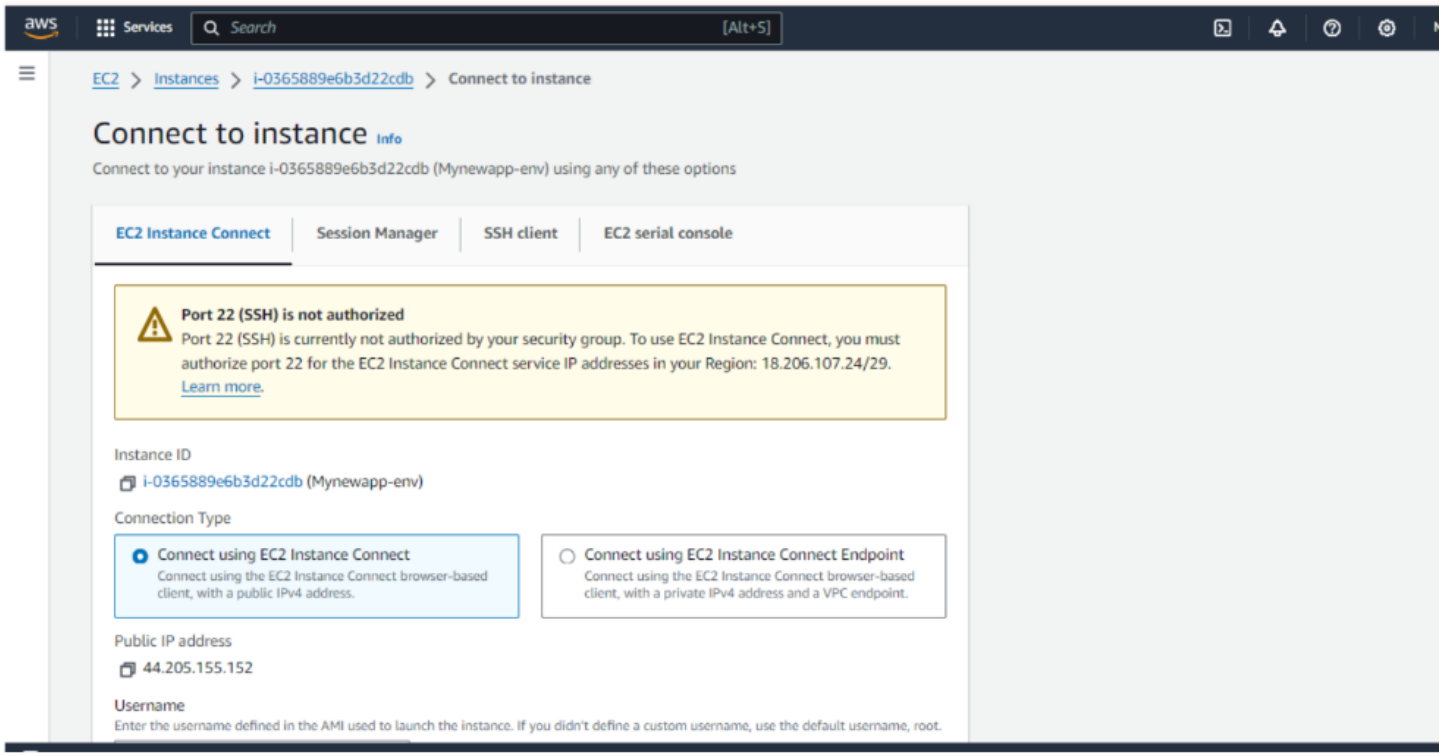
```
[root@ip-172-31-81-4 ec2-user]# kubectl get nodes
NAME                         STATUS     ROLES           AGE   VERSION
ip-172-31-81-4.ec2.internal  NotReady   control-plane   26m   v1.31.1
[root@ip-172-31-81-4 ec2-user]# kubectl get nodes
NAME                         STATUS     ROLES           AGE   VERSION
ip-172-31-81-4.ec2.internal  NotReady   control-plane   26m   v1.31.1
ip-172-31-94-95.ec2.internal NotReady   <none>          17s   v1.31.1
ip-172-31-95-221.ec2.internal NotReady  <none>          13s   v1.31.1
[root@ip-172-31-81-4 ec2-user]#
```

Advanced DevOps Experiment 4

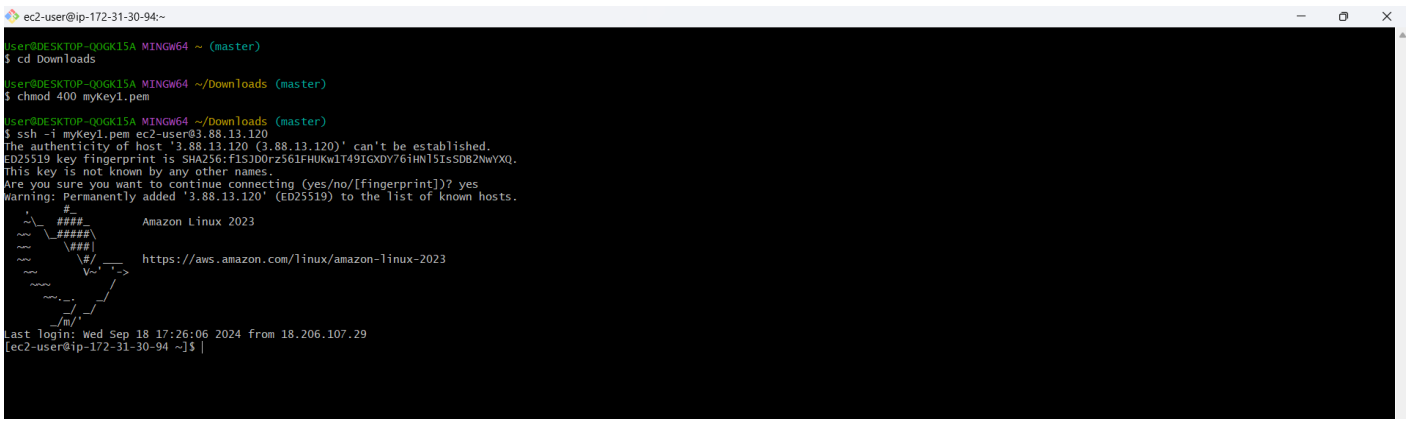
Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Step 1: Go to AWS Academy in services select EC2 and create 3 instance with and name them as master, node1, node2 and remember to select instance type as t2.medium.



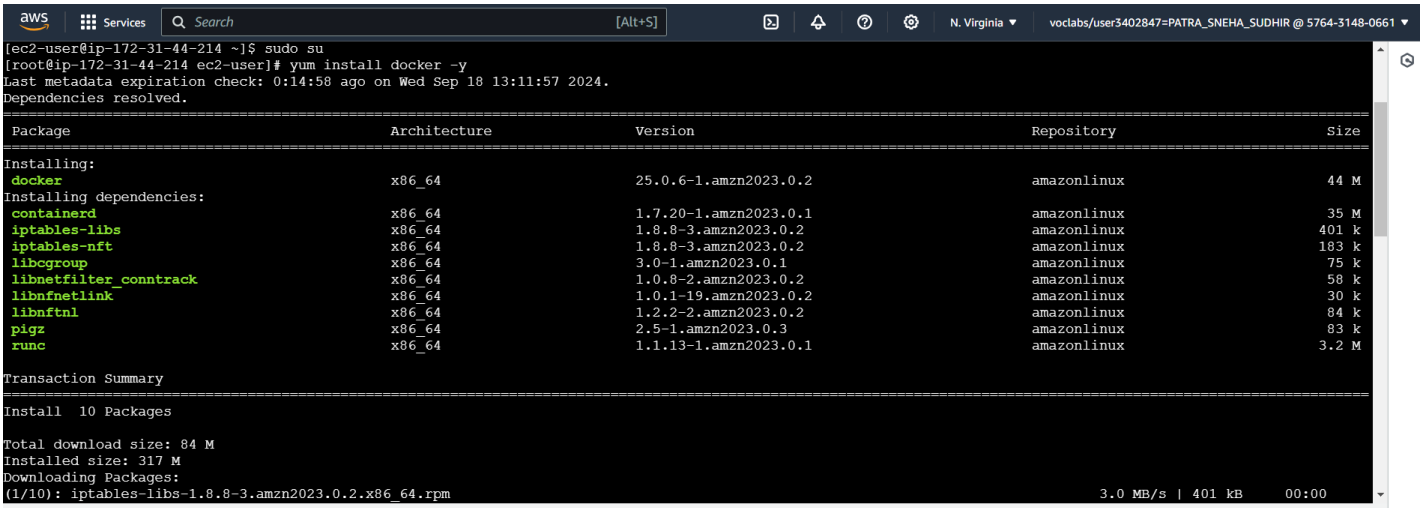


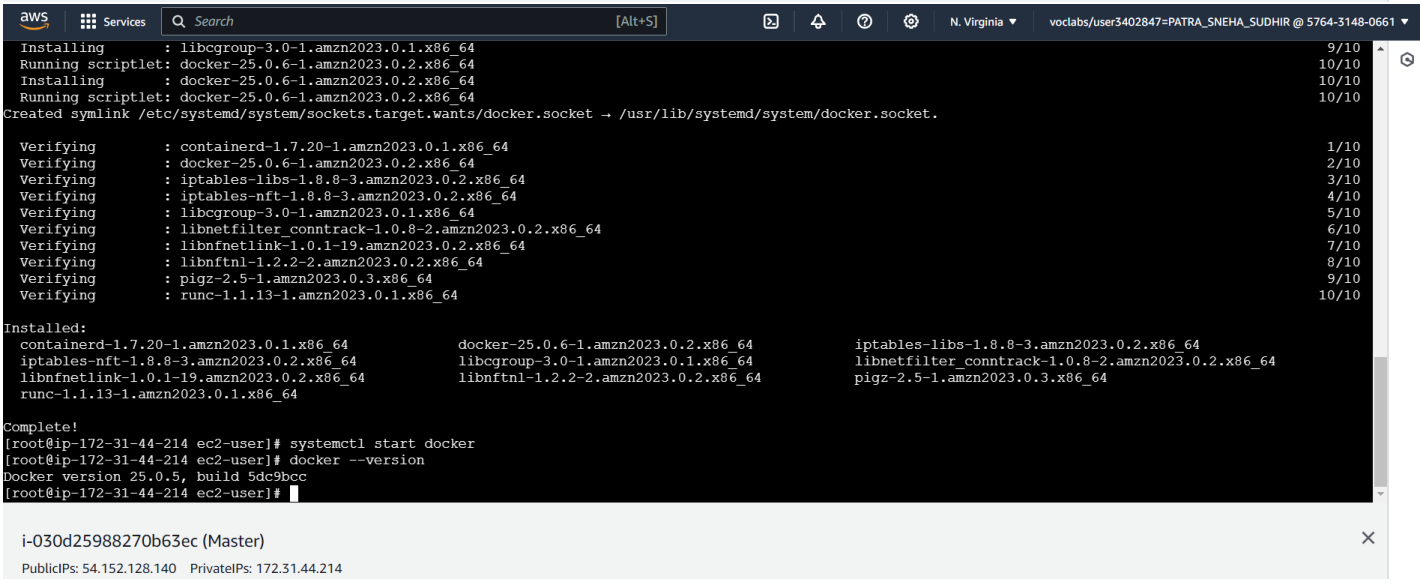
Step 2: Create a new key pair and name it as myKey1 and download as .pem file. Then, open command promt and go to the directory where the key is downloaded and run the following command
chmod 400 myKey1.pem
ssh -i myKey1.pem ec2-user@3.88.13.120
Repeat the steps for node1, master and node2



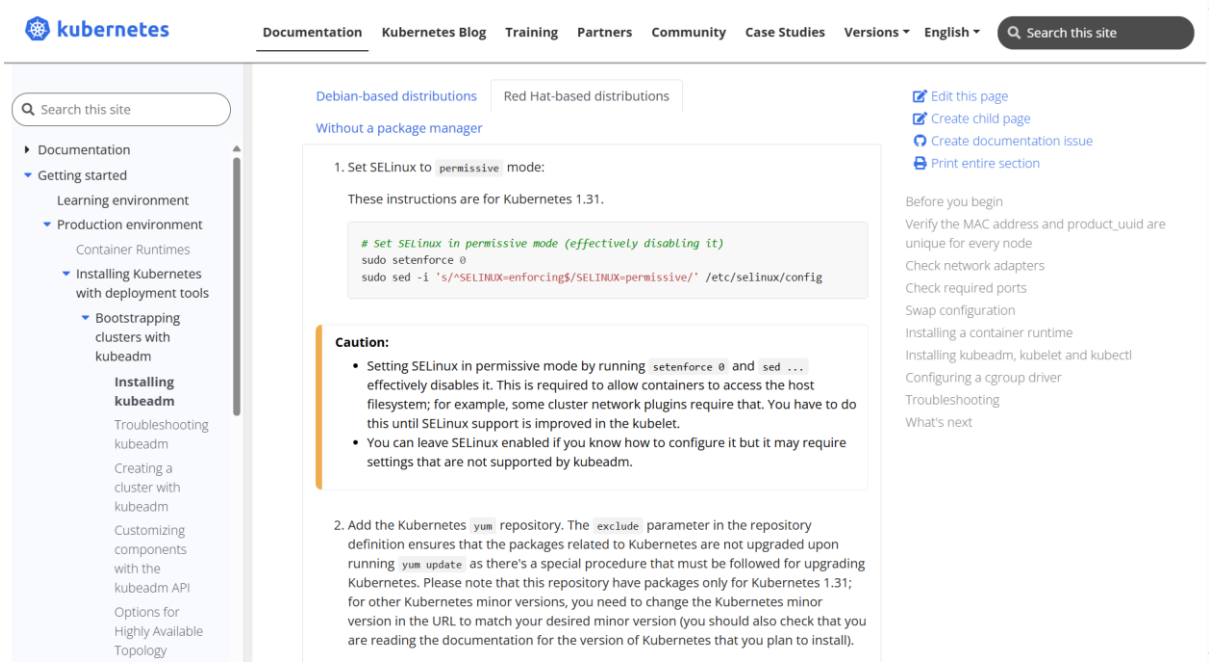
Step 3: Then, Select and connect each instance and run the following commands inside the console of each instance.

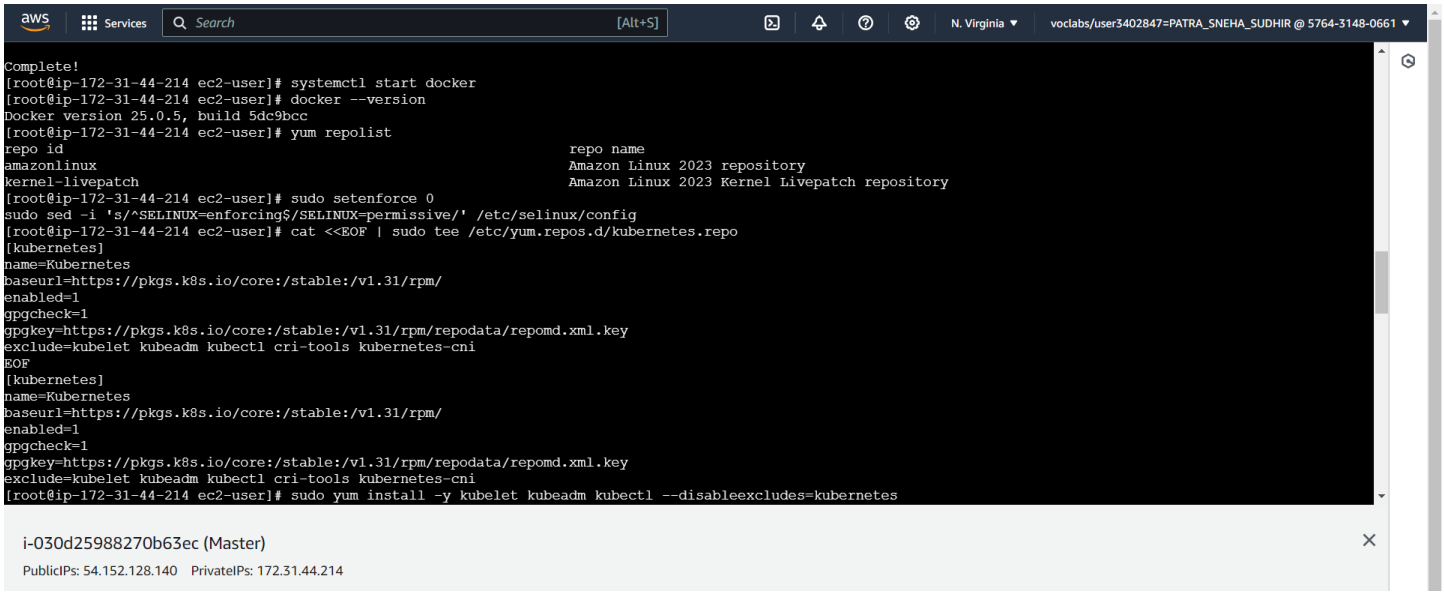
- sudo su
- yum install docker -y
- systemctl start docker
- docker –version
- yum repolist

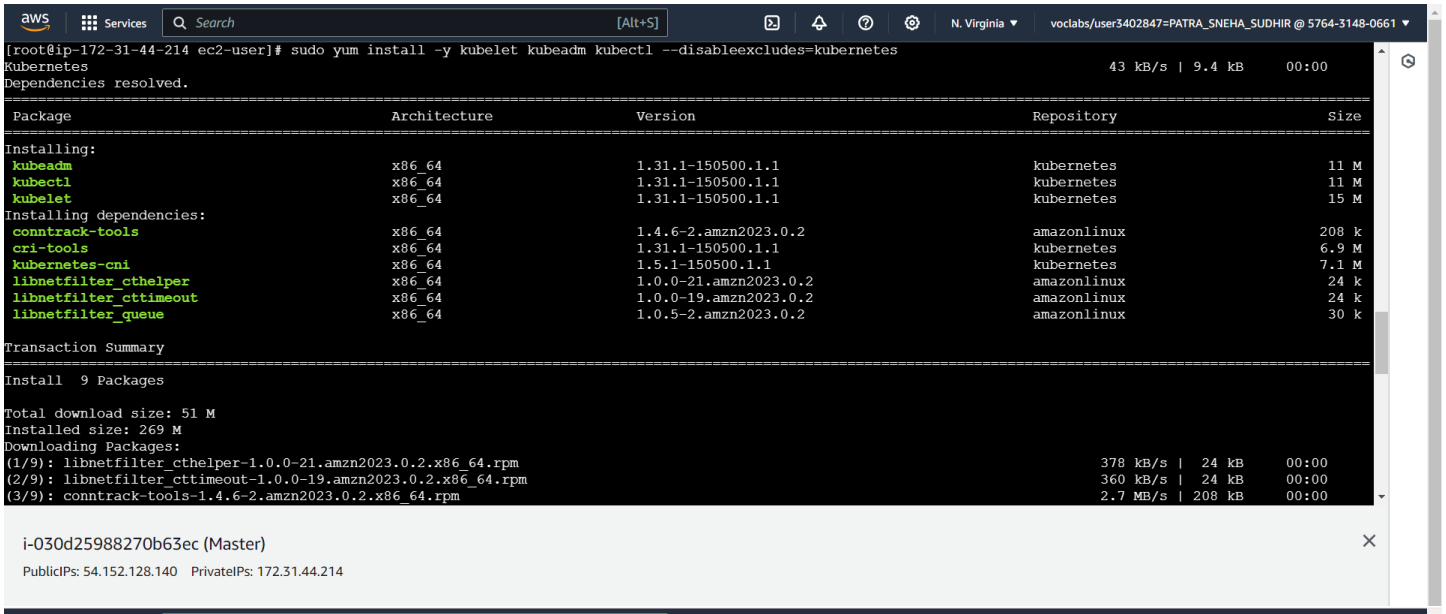


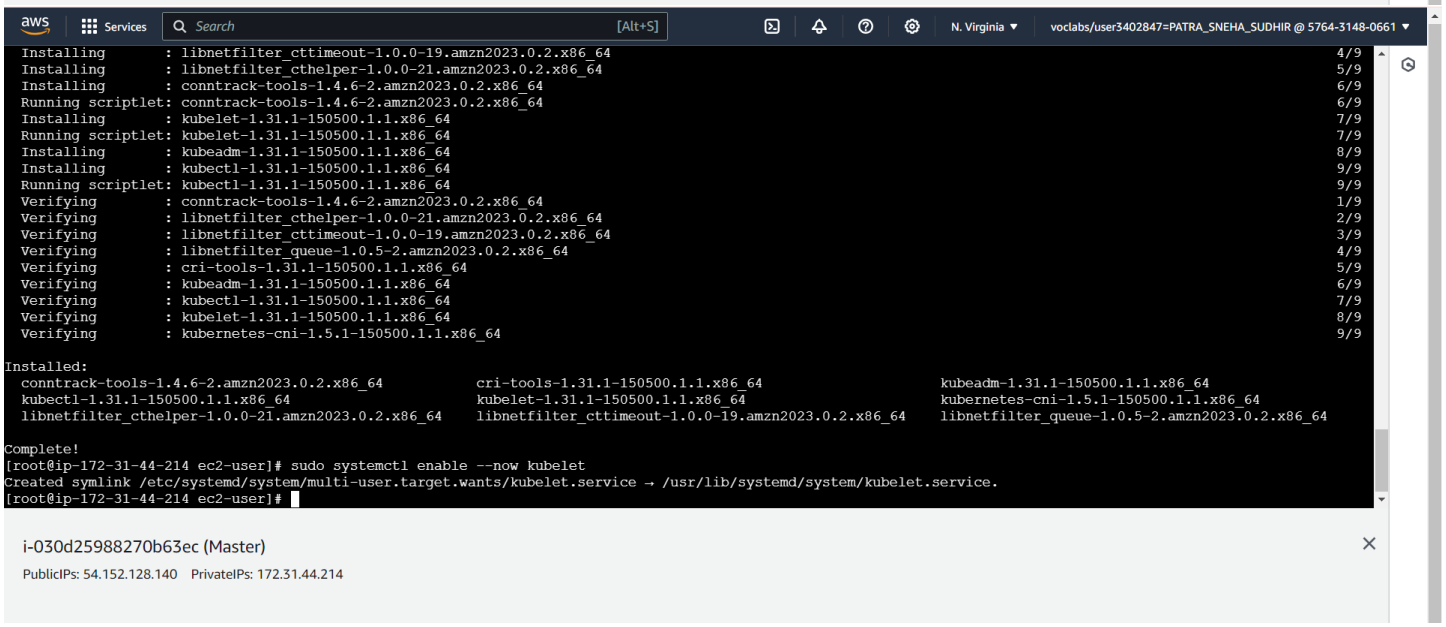


Step 4: Now, visit the following link https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/ and scroll down till you find Red-Hat and then select Red-Hat based distributions tab copy all the commands one by one in each console of instance.

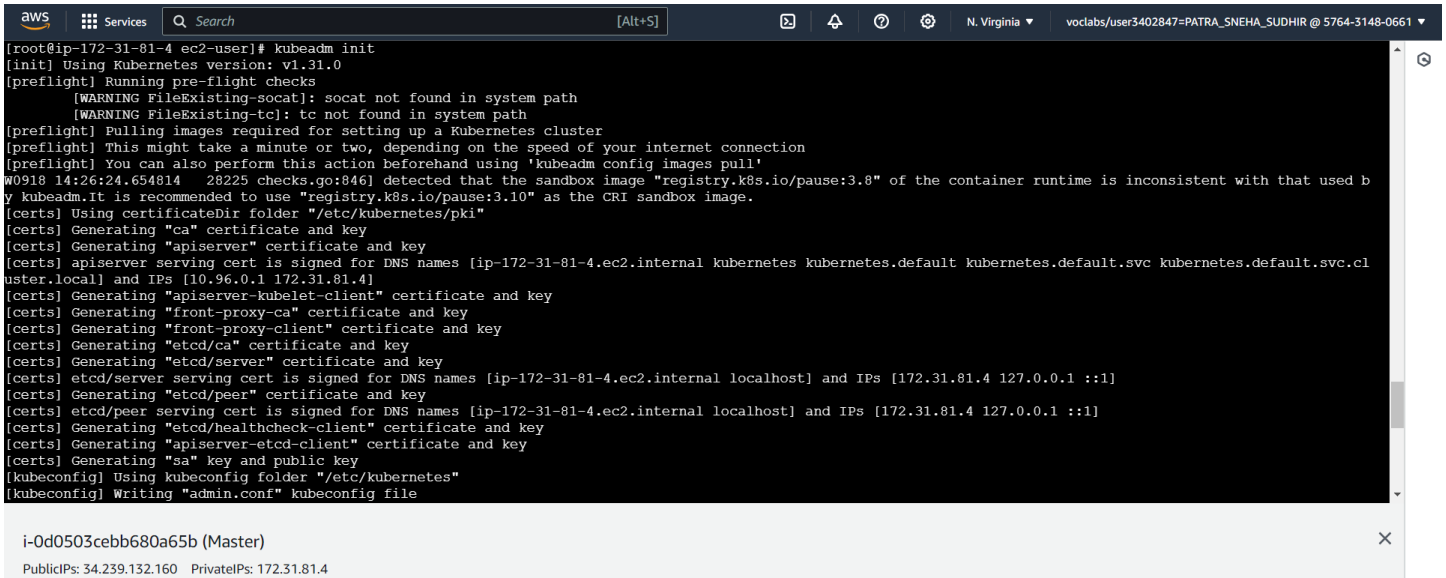






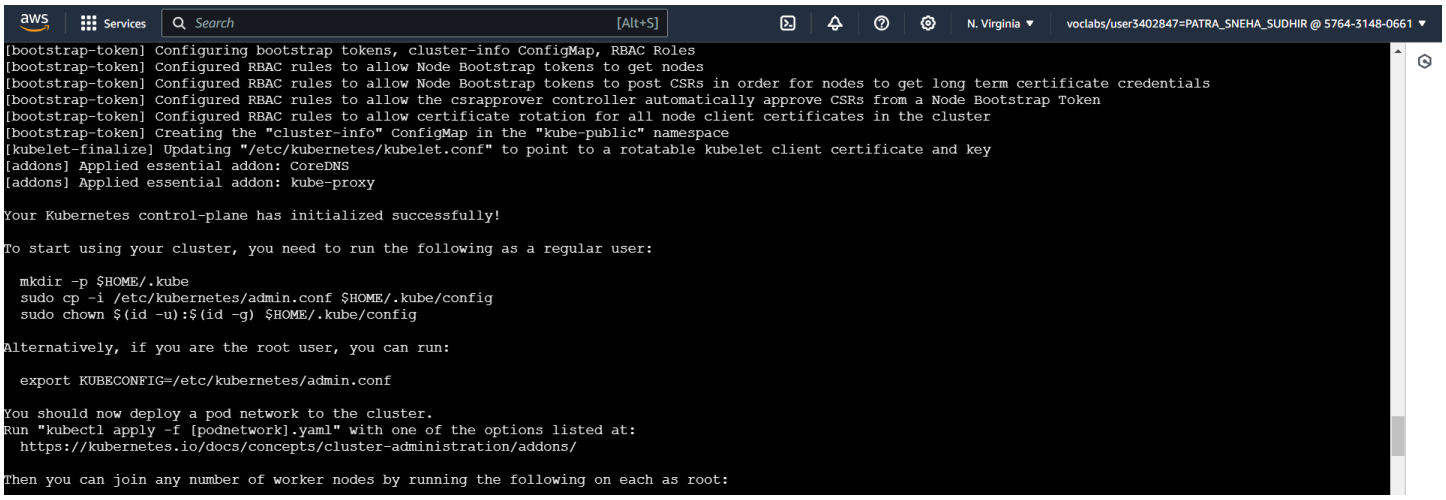


Step 5: Now, run the following command in the mater instance - kubeadm init

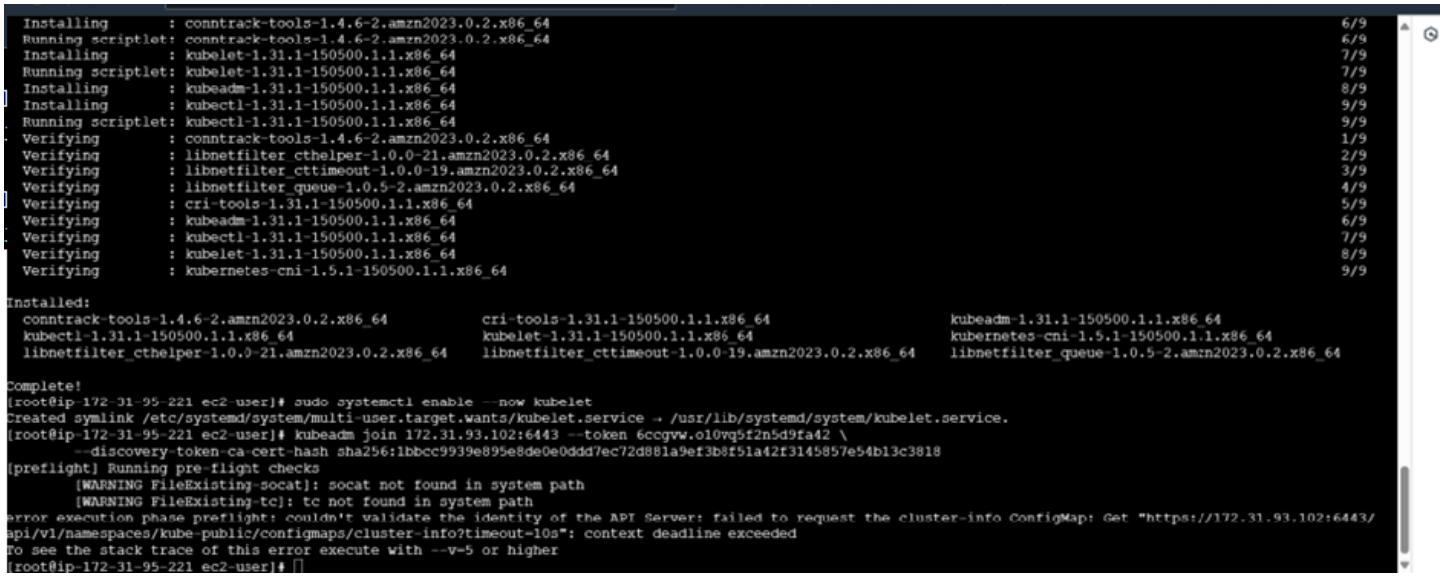


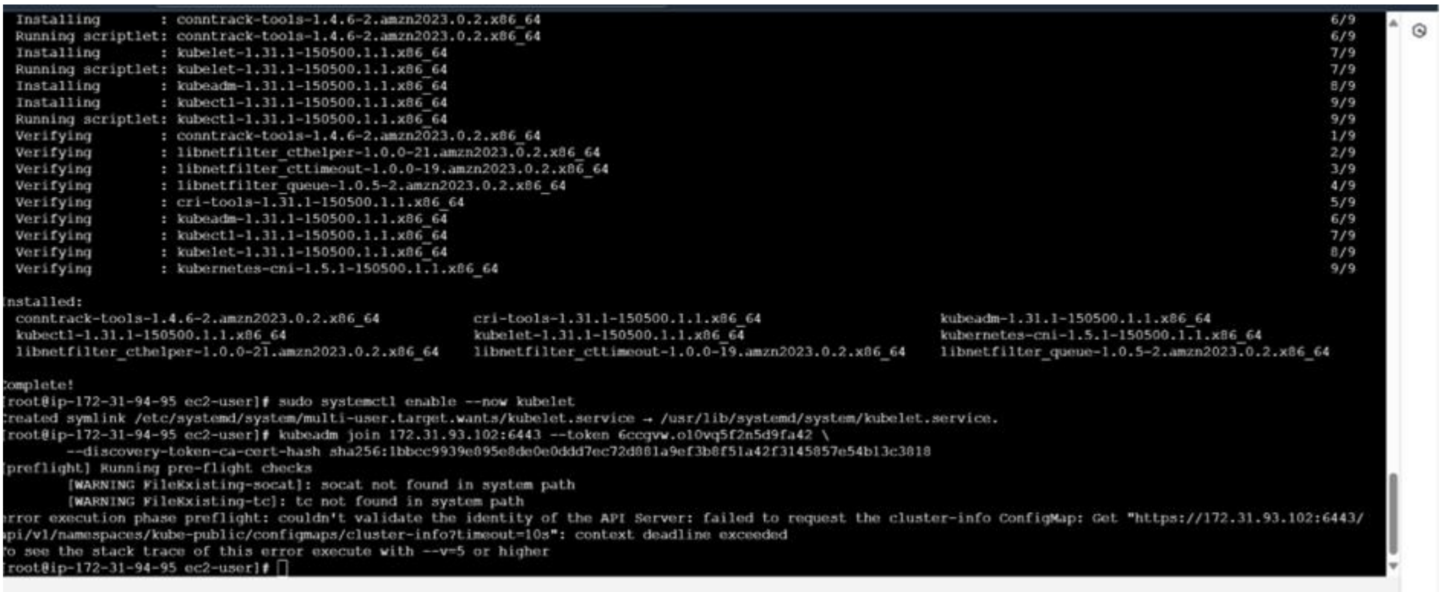
Step 6: Now, run the following commands in master instance's console –

a. mkdir -p $HOME/.kube
   sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   sudo chown $(id -u):$(id -g) $HOME/.kube/config
b. export KUBECONFIG=/etc/kubernetes/admin.conf
c. kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
   --discovery-token-ca-cert-hash
   sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818

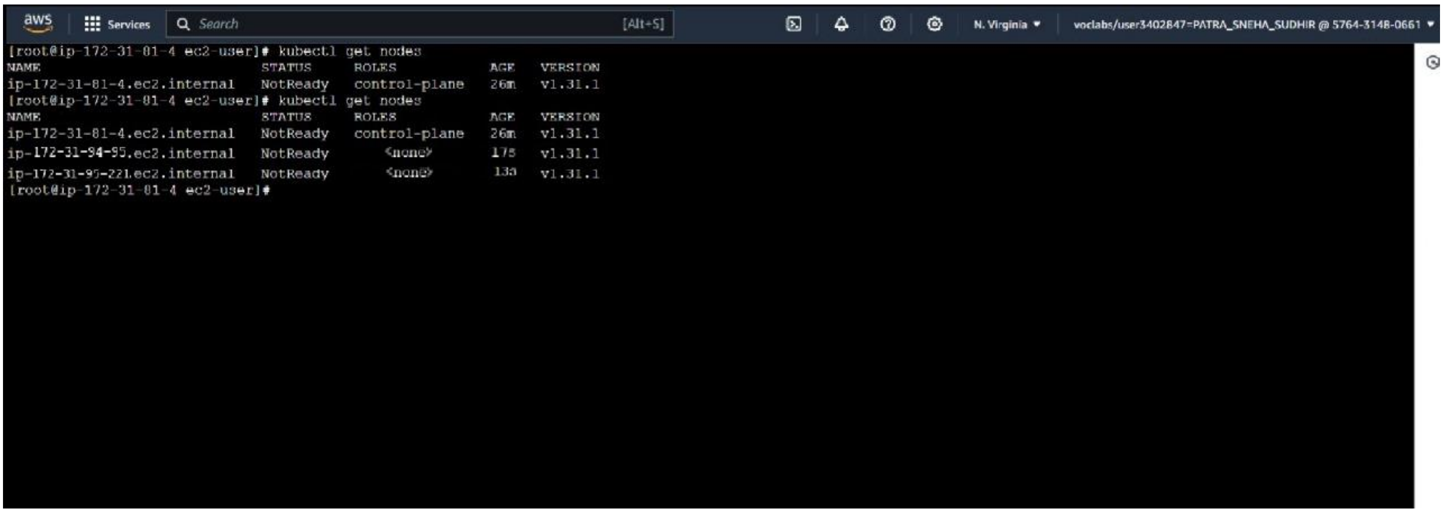


Step 7: Run this command in node1 and node2 -
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818

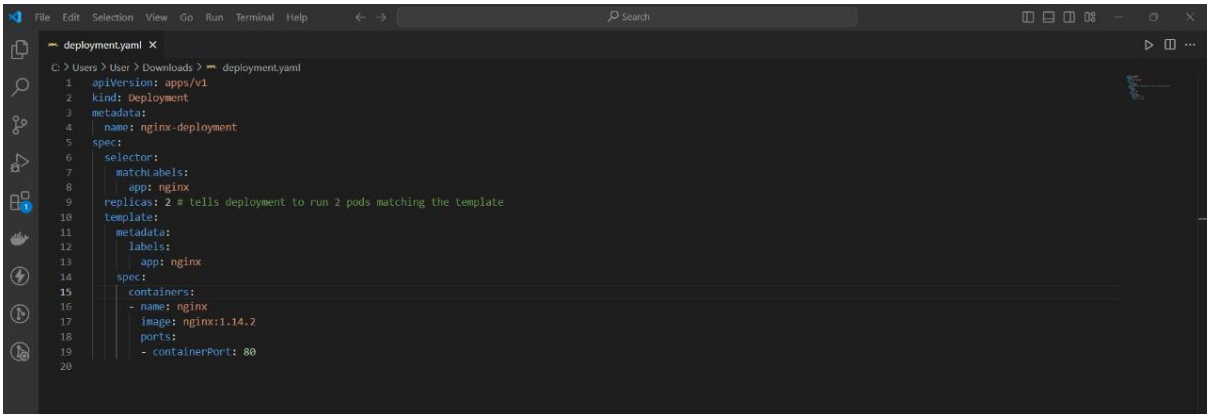




Step 8: Run the following command in master instance console -
kubectl get nodes

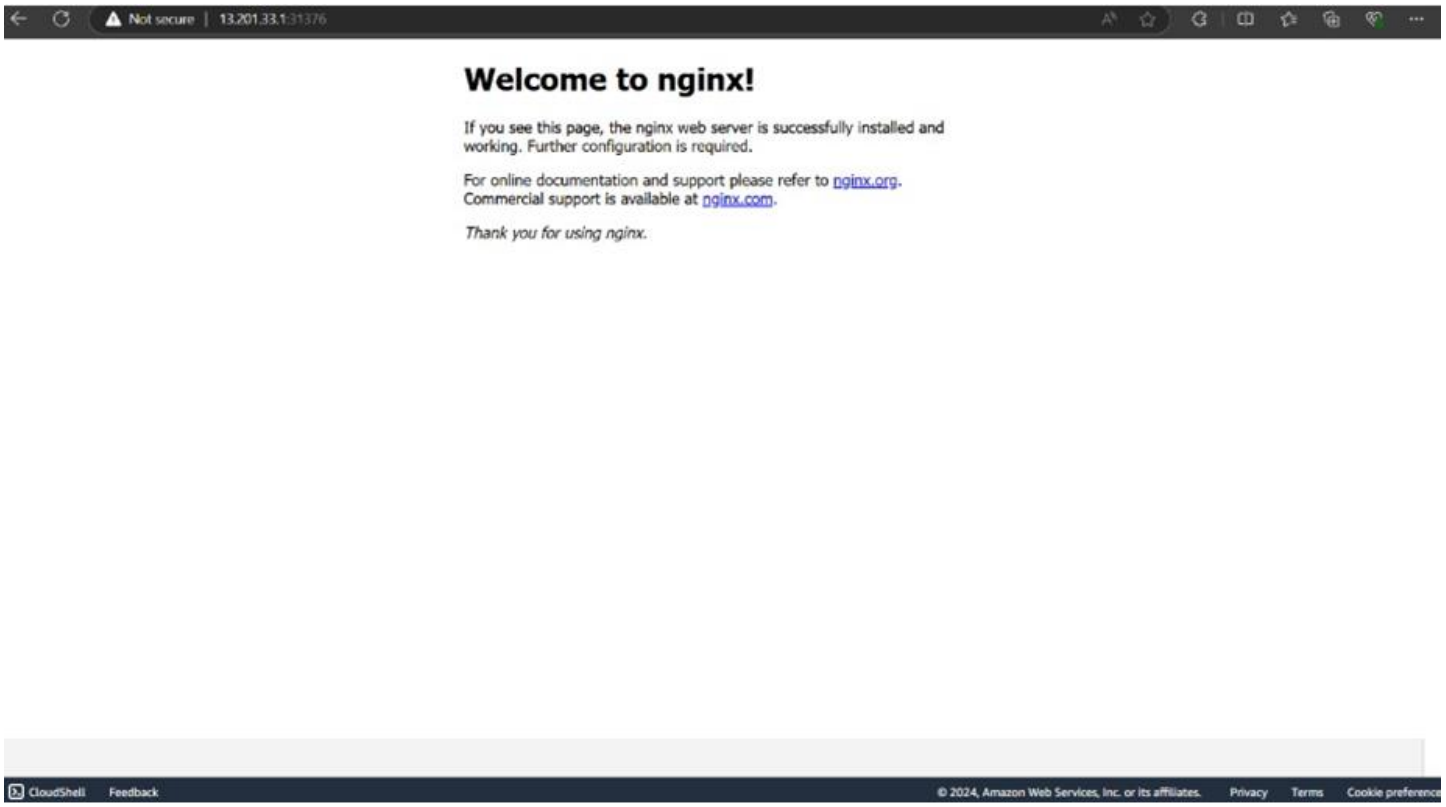


Step 9: Once the cluster is set up and running, deploy an Nginx application:
kubectl apply -f https://k8s.io/examples/application/deployment.yaml
Forward the Nginx service to your localhost so that you can access it using the following command
kubectl port-forward deployment/nginx-deployment 8080:80



Step 10: In a new terminal of Git Bash, run the following command:
curl --head http://127.0.0.1:8080

Step 11: The website is deployed.

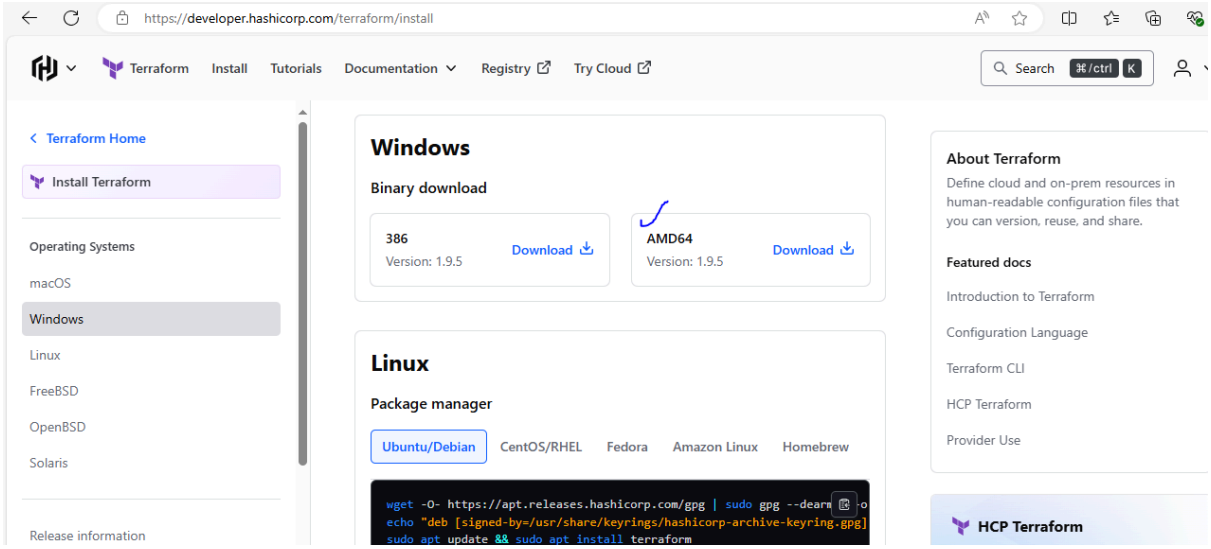


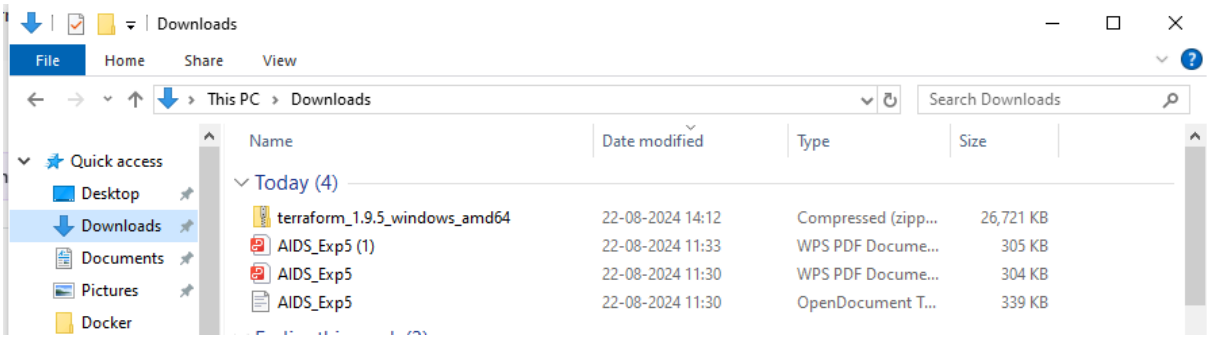
EXPERIMENT 5:TERRAFORM

AIM: To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine and Windows.

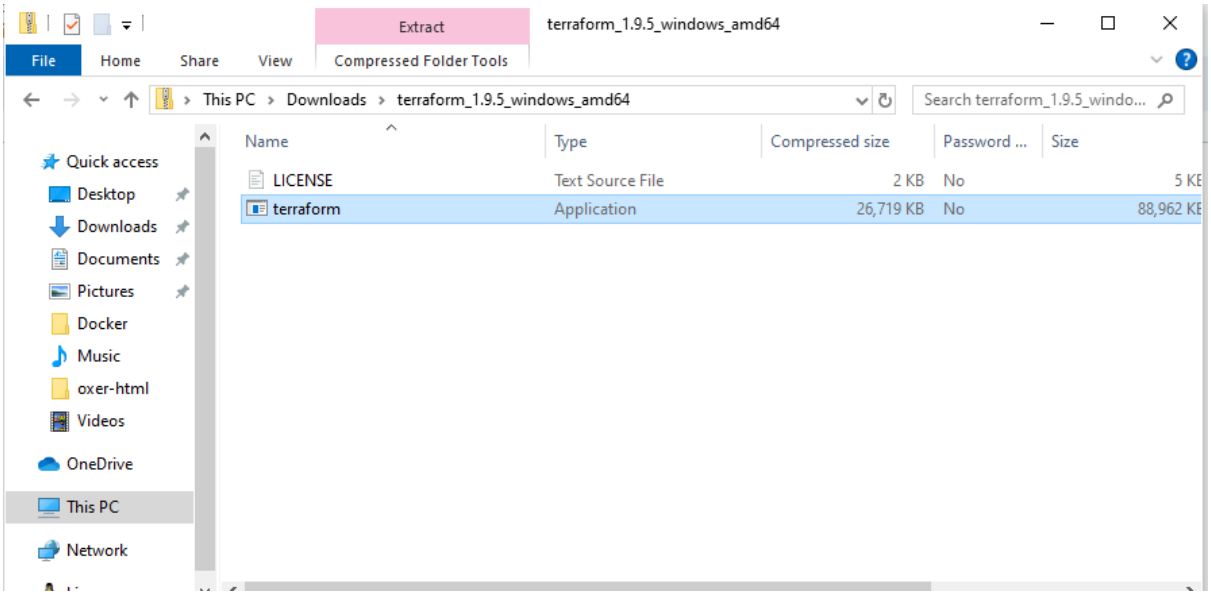
Installation and Configuration of Terraform in Windows:

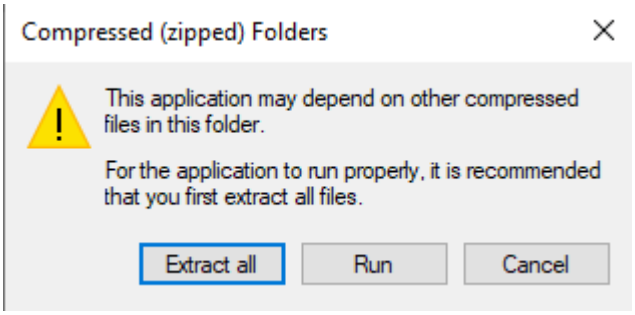
1: To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website: :https://www.terraform.io/downloads.html



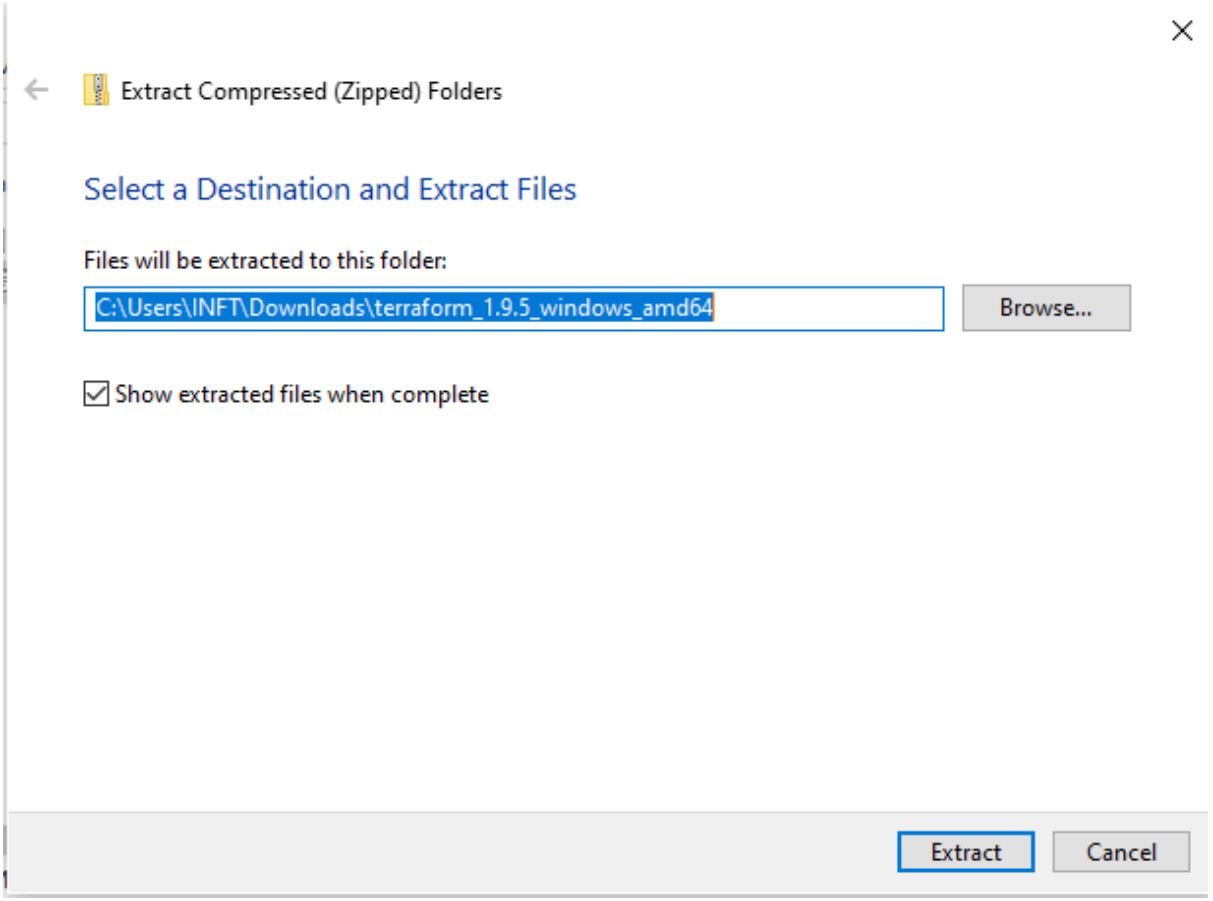


2:Compress the zip file.

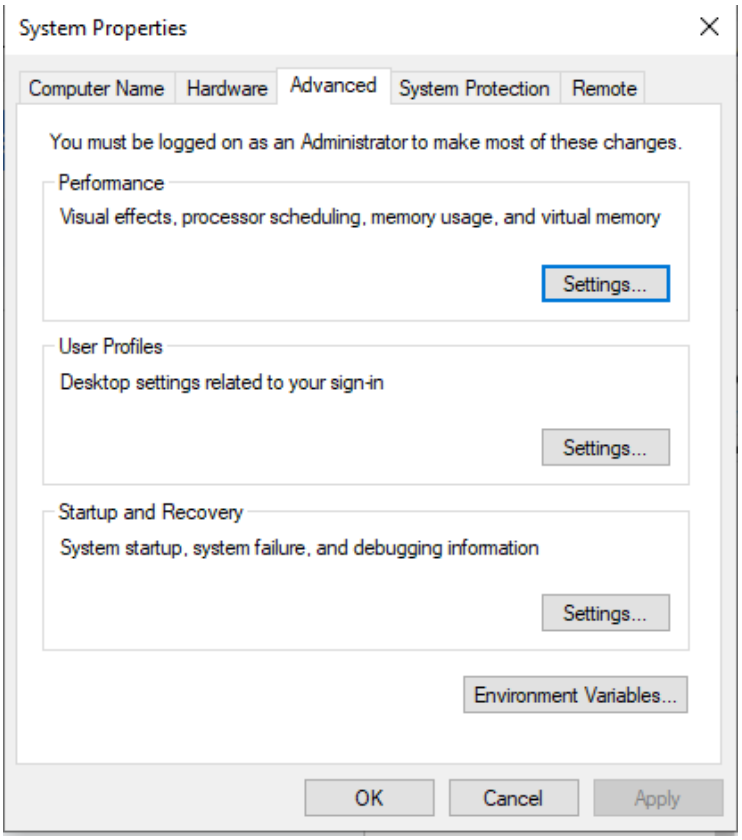


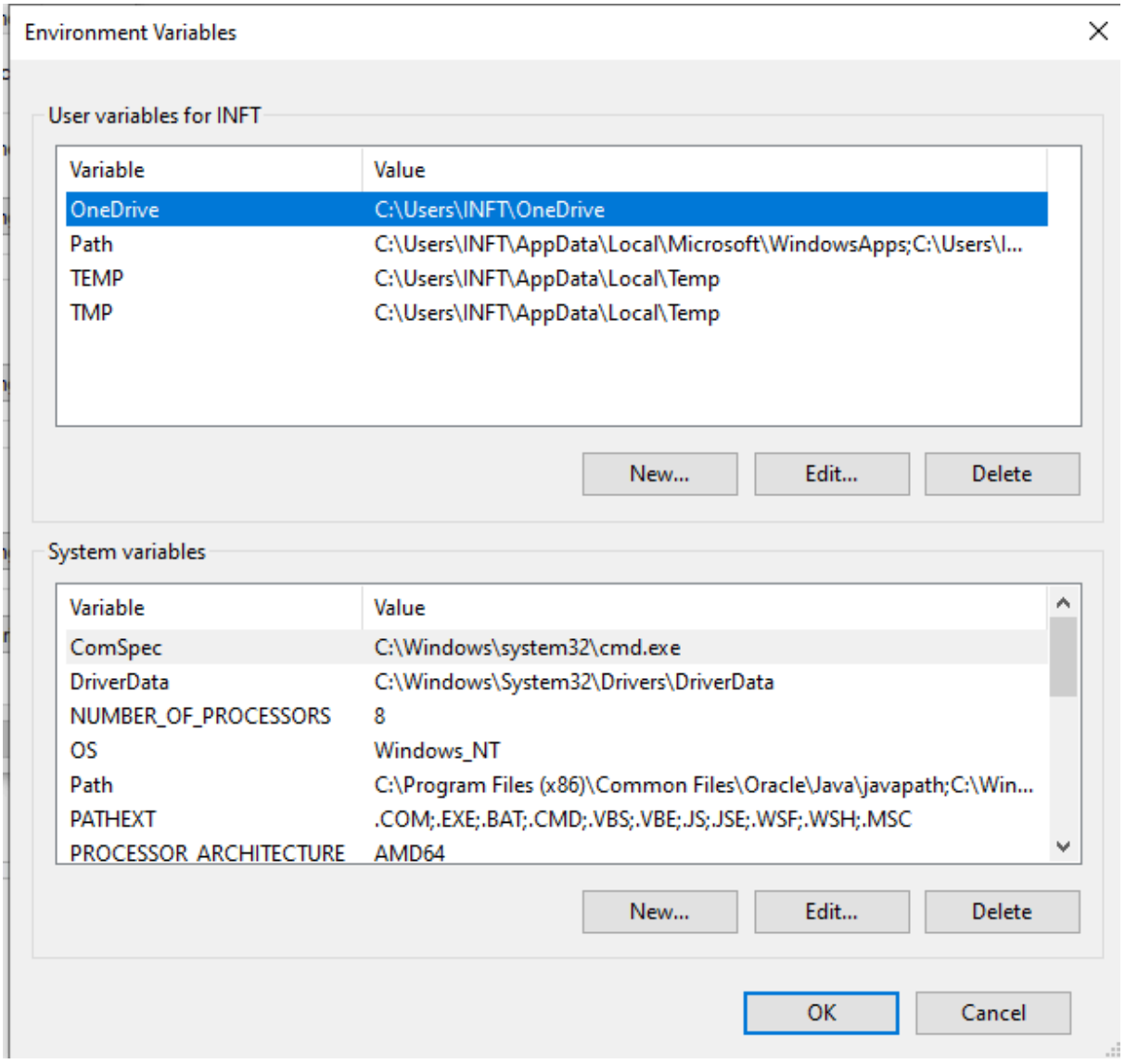


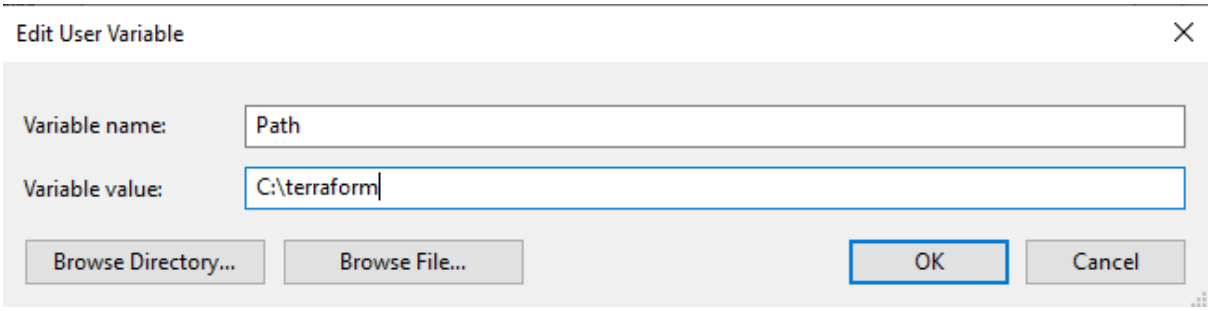
2: Extract the downloaded setup file Terraform.exe.



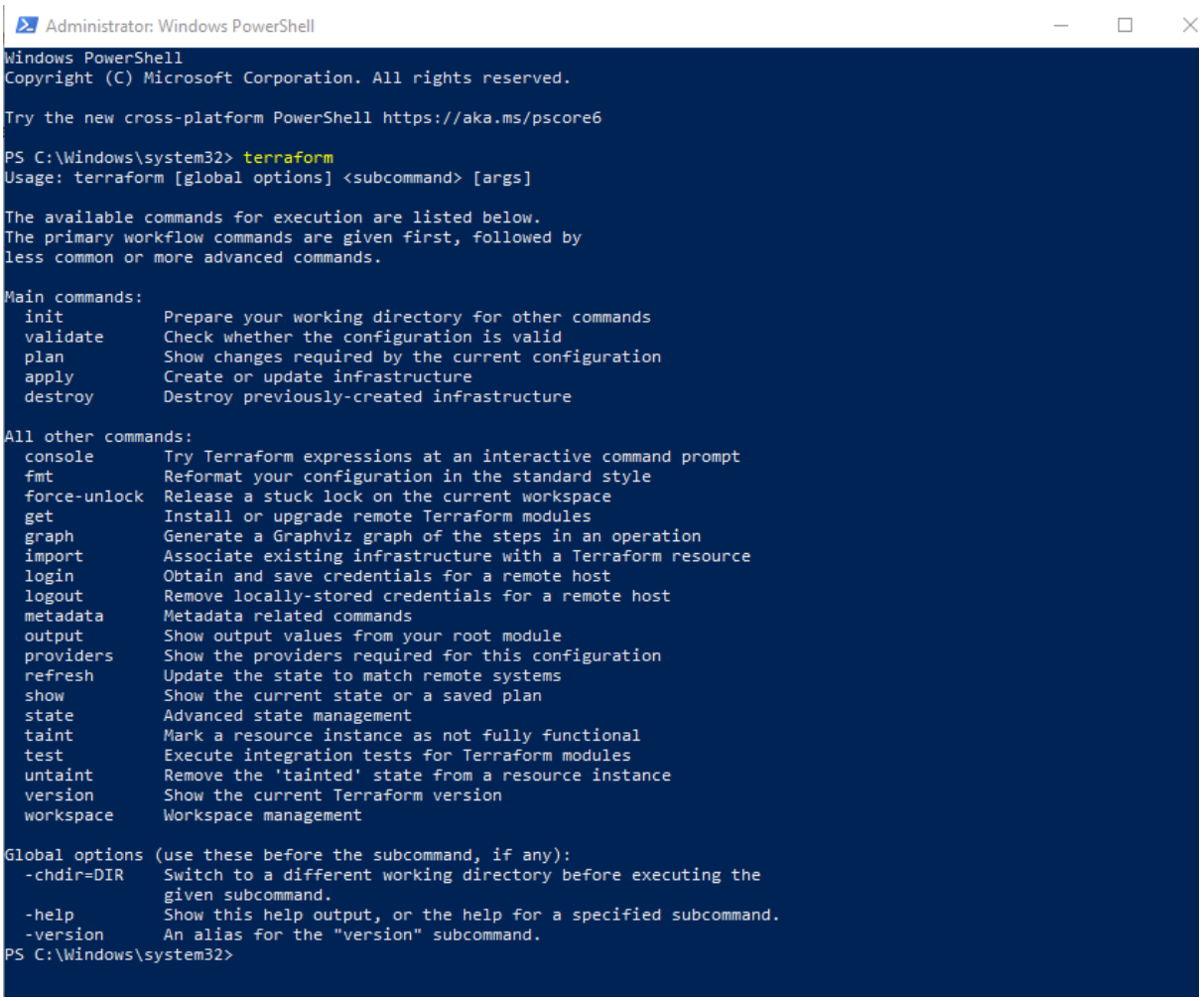
4: Set the System path for Terraform in Environment Variables.




Environment Variables
User variables for INFT
Variable
Value
OneDrive
C:\Users\INFT\OneDrive
Path
C:\Users\INFT\AppData\Local\Microsoft\WindowsApps;C:\Users\I...
TEMP
C:\Users\INFT\AppData\Local\Temp
TMP
C:\Users\INFT\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\Windows\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
8
OS
Windows_NT
Path
C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE
AMD64
New...
Edit...
Delete
OK
Cancel


Edit User Variable
Variable name:
Path
Variable value:
C:\terraform
Browse Directory...
Browse File...
OK
Cancel

5 : Open Terraform in PowerShell and check its functionality

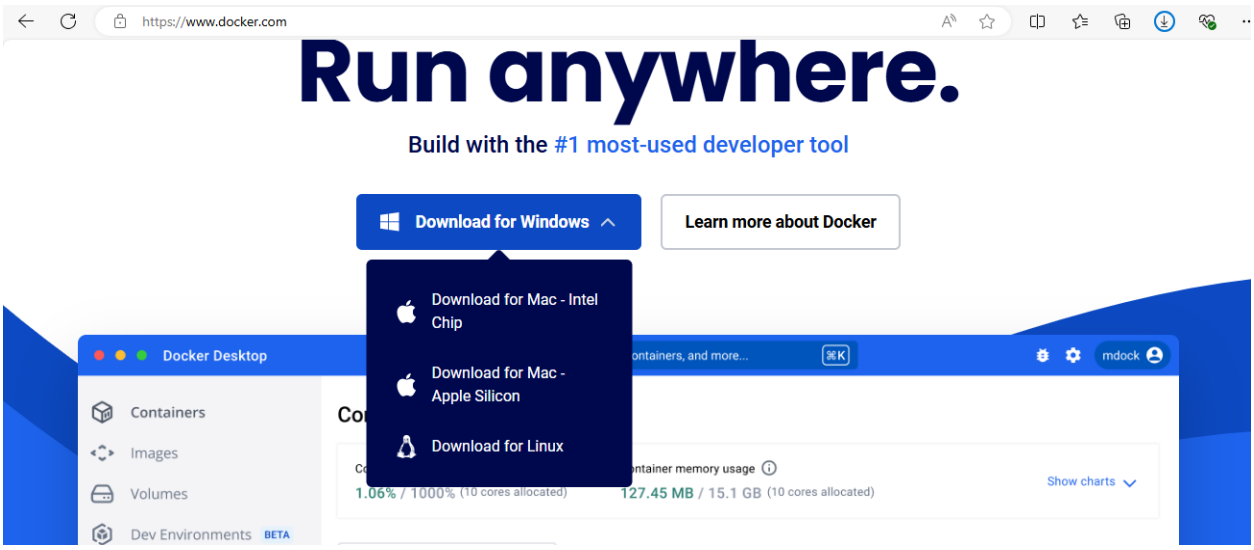


Sneha Patra **D15A** **41**

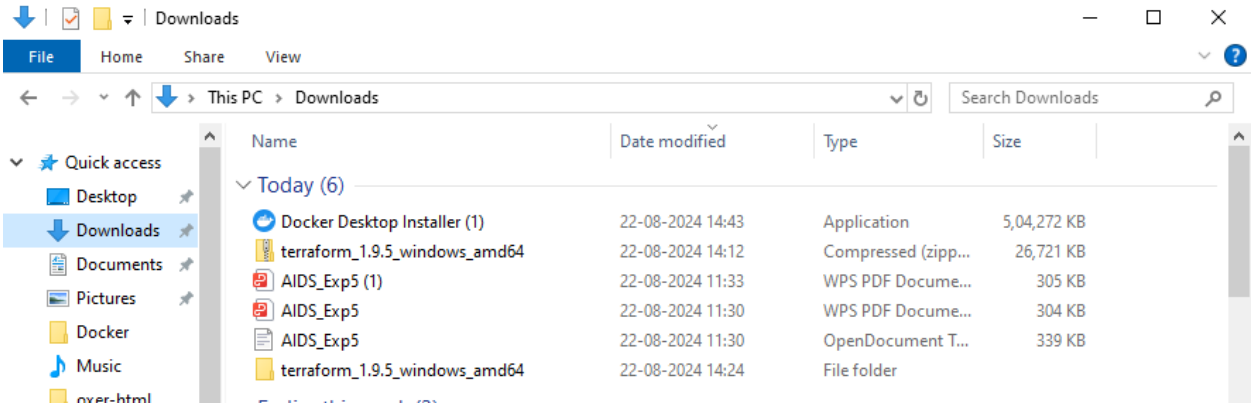
Experiment 6

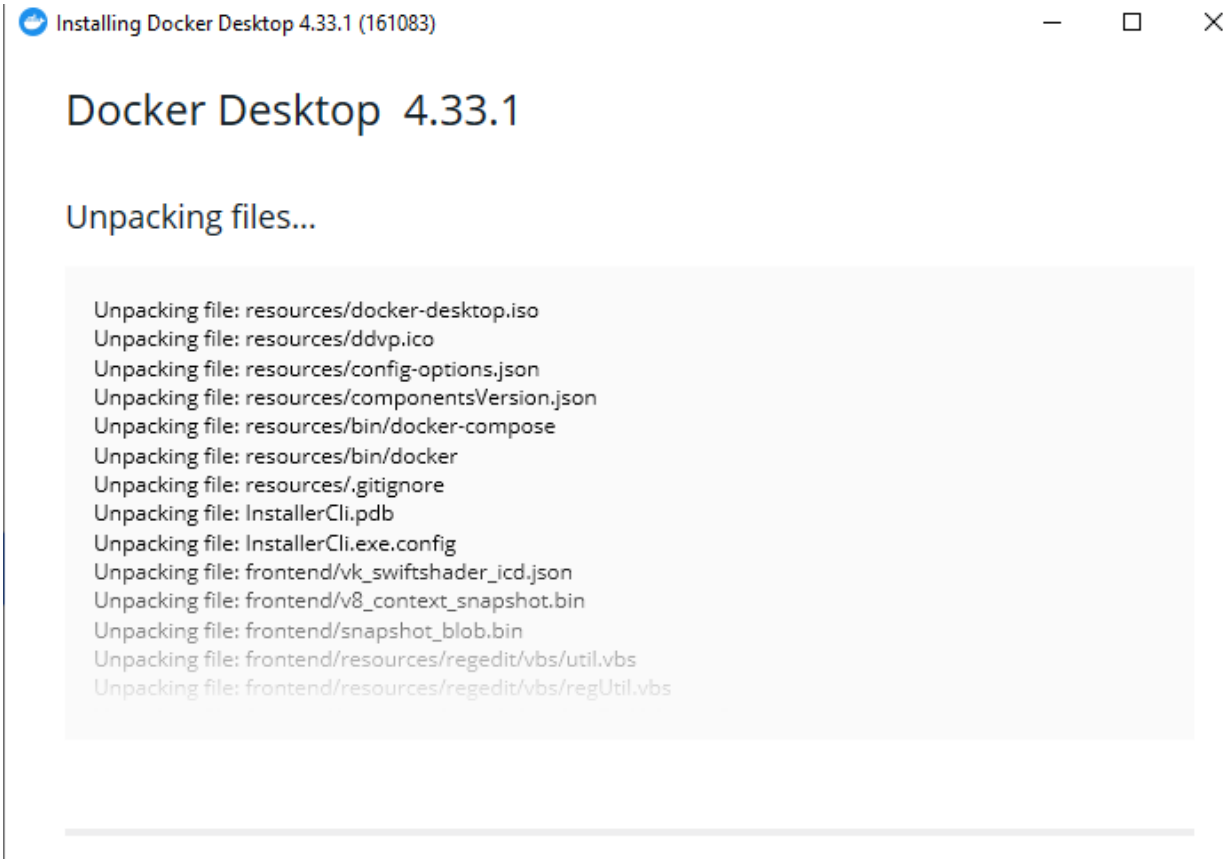
Aim: To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker)

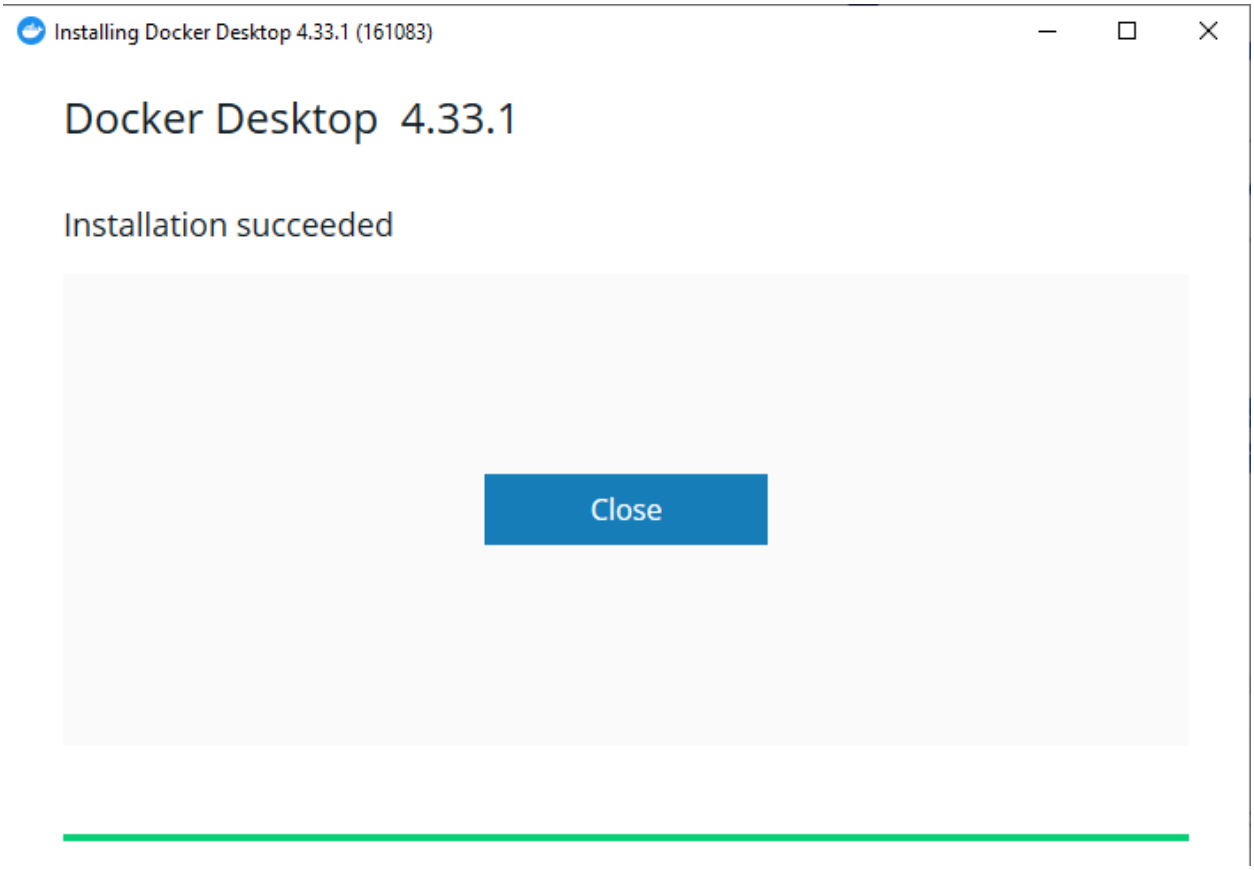
Step 1: Download Docker form www.docker.com



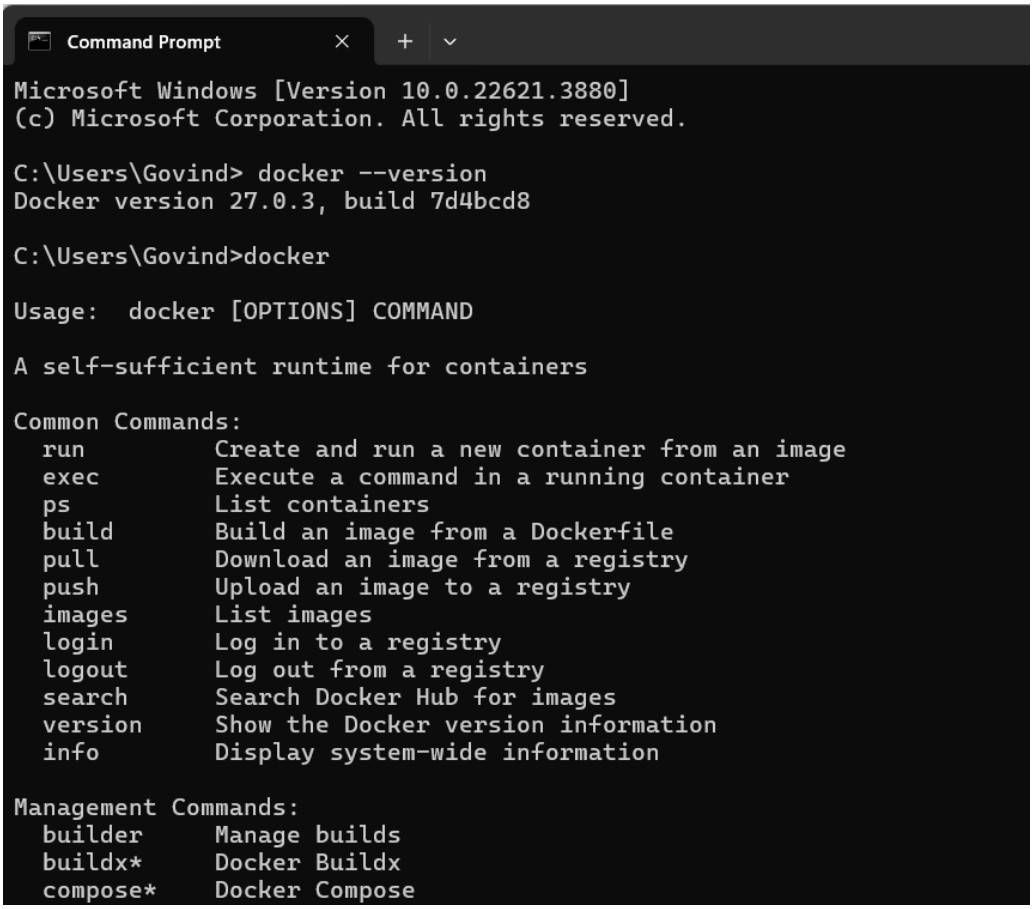
Step 2: Now, Docker is successfully downloaded.



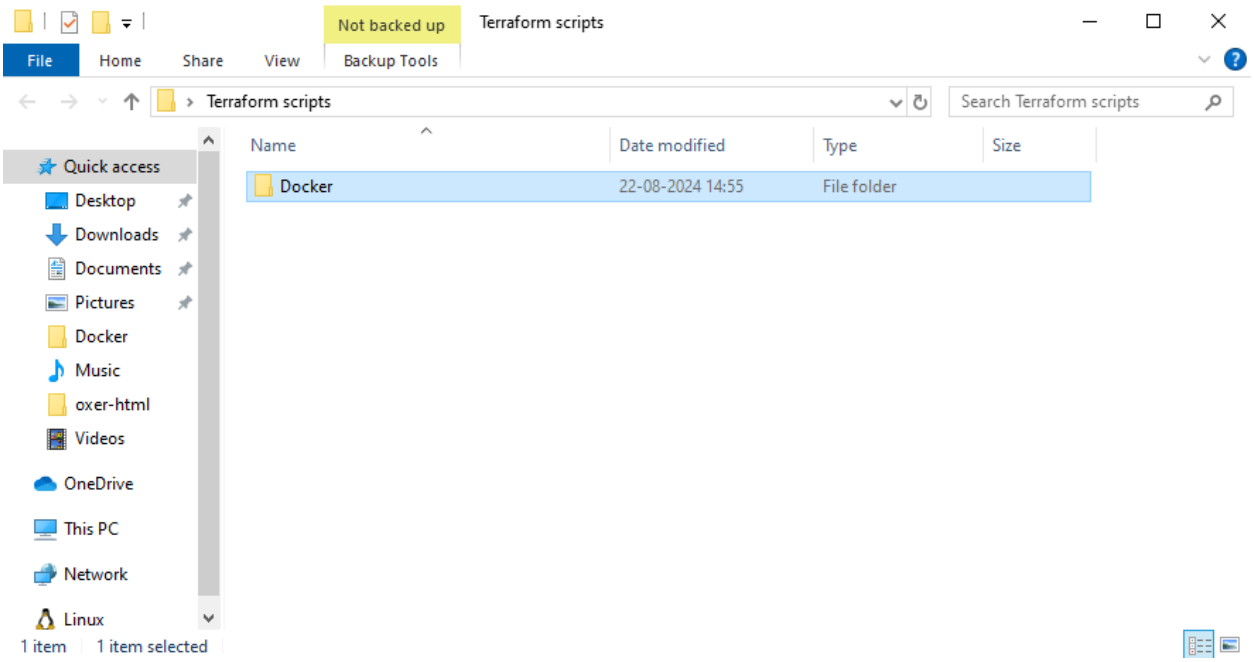

Installing Docker Desktop 4.33.1 (161083)
Docker Desktop 4.33.1
Unpacking files...
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
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


Installing Docker Desktop 4.33.1 (161083)
Docker Desktop 4.33.1
Installation succeeded
Close

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



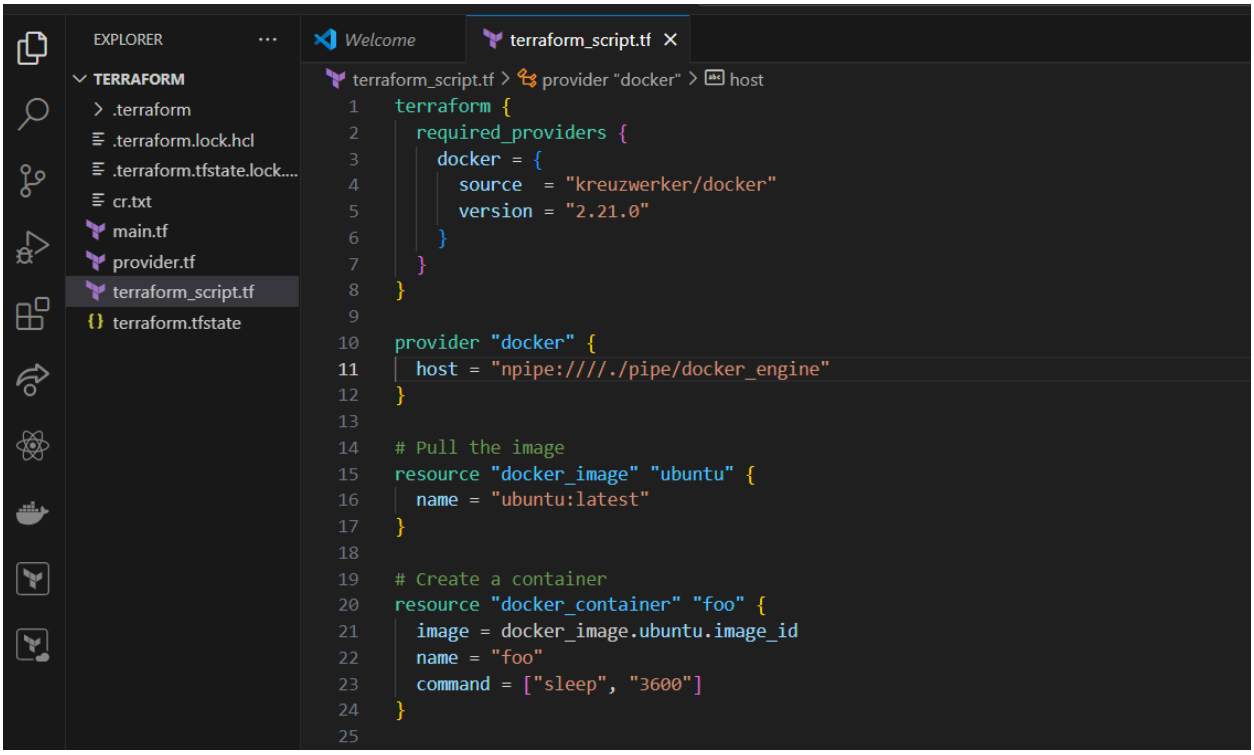
Step 4: Create a folder Terraform_scripts and inside it create a folder named Docker.



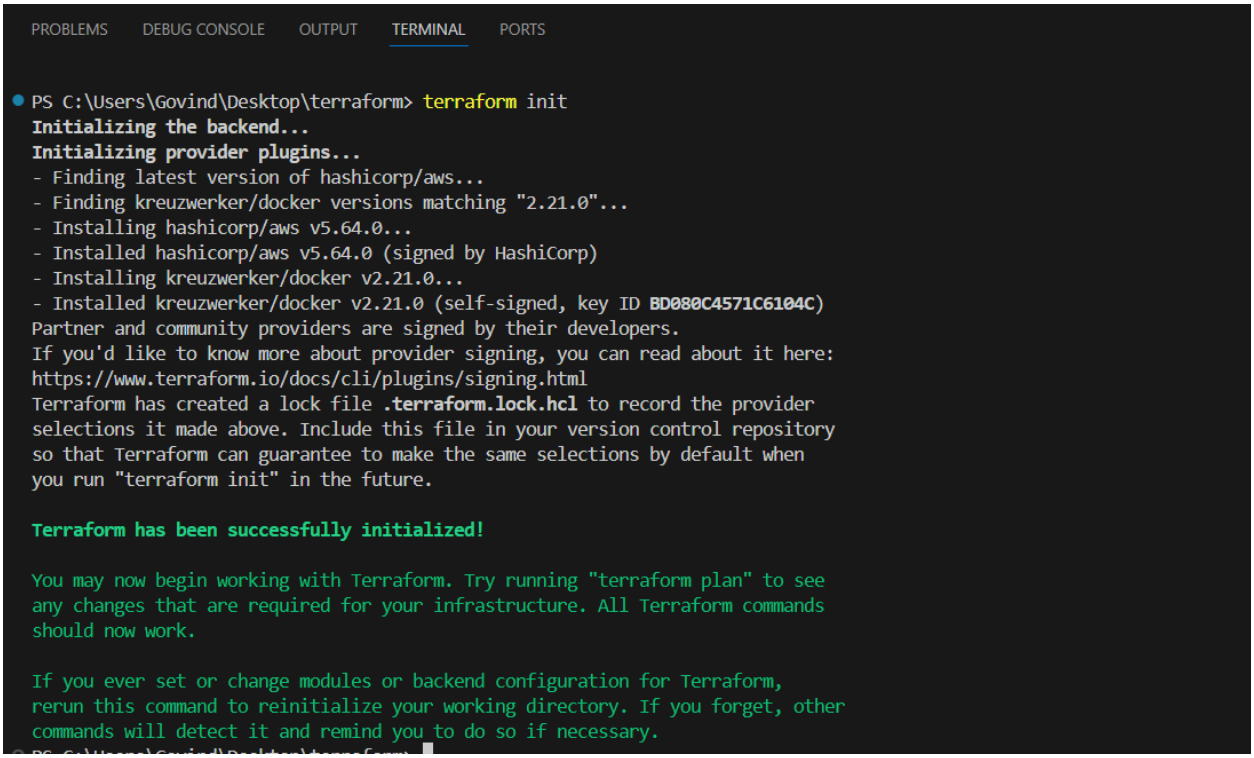
Step 5: create a new folder named ‘Terraform’ in the ‘TerraformScripts’ folder. Then create a new terraform_script.tf file using vs code.
Run the following script in the VS Code.

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

provider "docker" {
  host = "npipe:////./pipe/docker_engine"
}

# Pull the image
```

```
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

# Create a container
resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name = "foo"
  command = ["sleep", "3600"]
}
```



Step 6: Open Windows Explorer and run the following command terraform init, terraform plan, terraform apply, terraform destroy, terraform provider, terraform validate, terraform state list and docker images.



```
PS C:\Users\Govind\Desktop\terraform> 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          = [
          + "sleep",
          + "3600",
        ]
      + 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)
```

Ln 25, Col 1 (439 selected) Spaces: 2 UTF-8 CRLF

```
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\Govind\Desktop\terraform> 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          = [
          + "sleep",
          + "3600",
        ]
      + 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)
```

Ln 25, Col 1 (439 selected) Spaces: 2 UTF-8 CRLF

```
      + 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: Still creating... [20s elapsed]
docker_image.ubuntu: Creation complete after 29s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=2f7e8bcaf7e5f75f04f53be0aa80e74a915285dddb826402ebfc7f569e571ebd]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

```
PS C:\Users\Govind\Desktop\terraform> terraform providers

Providers required by configuration:
.
├── provider[registry.terraform.io/kreuzwerker/docker] 2.21.0
└── provider[registry.terraform.io/hashicorp/aws]

Providers required by state:

    provider[registry.terraform.io/kreuzwerker/docker]
```

```
PS C:\Users\Govind\Desktop\terraform> terraform validate
Success! The configuration is valid.

PS C:\Users\Govind\Desktop\terraform> terraform state list
docker_container.foo
docker_image.ubuntu
```

```
PS C:\Users\Govind\Desktop\terraform> docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
ubuntu                  latest    edbfe74c41f8   3 weeks ago     78.1MB
nginx                   latest    448a08f1d2f9   15 months ago   142MB
nginx/docker-extension  0.0.3     41d3d0d7d940   16 months ago   7.53MB
```

```
PS C:\Users\Govind\Desktop\terraform> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=2f7e8bcaf7e5f75f04f53be0aa80e74a915285dddb826402ebfc7f569e571ebd]

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_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach            = false -> null
      - command           = [
          - "sleep",
          - "3600",
        ] -> null
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      - entrypoint        = [] -> null
      - env               = [] -> null
      - gateway           = "172.17.0.1" -> null
      - group_add         = [] -> null
      - hostname          = "2f7e8bcaf7e5" -> null
      - id                = "2f7e8bcaf7e5f75f04f53be0aa80e74a915285dddb826402ebfc7f569e571ebd" -> null
      - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - init              = false -> null
      - ip_address        = "172.17.0.2" -> null
      - ip_prefix_length  = 16 -> null
      - ipc_mode          = "private" -> null
      - links             = [] -> null
      - log_driver        = "json-file" -> null
      - log_opts          = {} -> null
      - logs              = false -> null
      - max_retry_count   = 0 -> null
      - memory            = 0 -> null
      - memory_swap       = 0 -> null
      - must_run          = true -> null
      - name              = "foo" -> null
      - network_data      = [
```

Experiment No 7

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

Step 1: Make sure you download all the Prerequisites required :

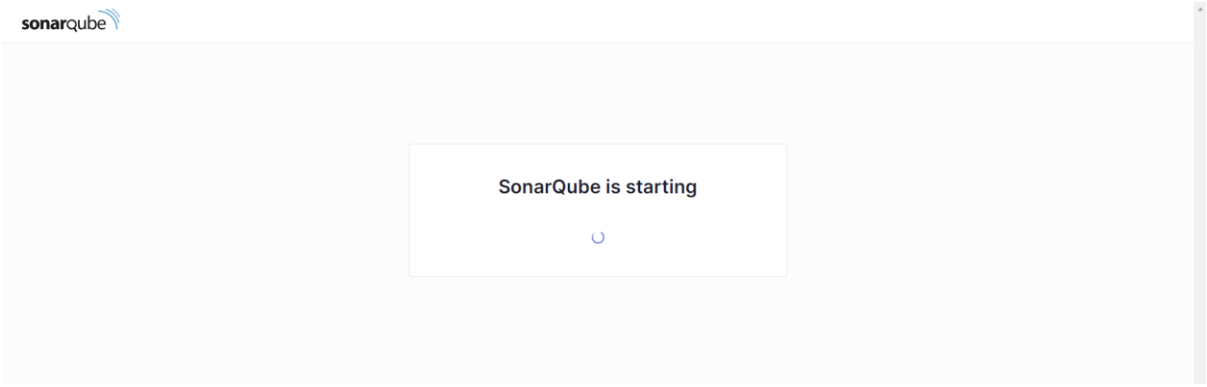
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

And then run the following command on your window powershell/cmd prompt only once:

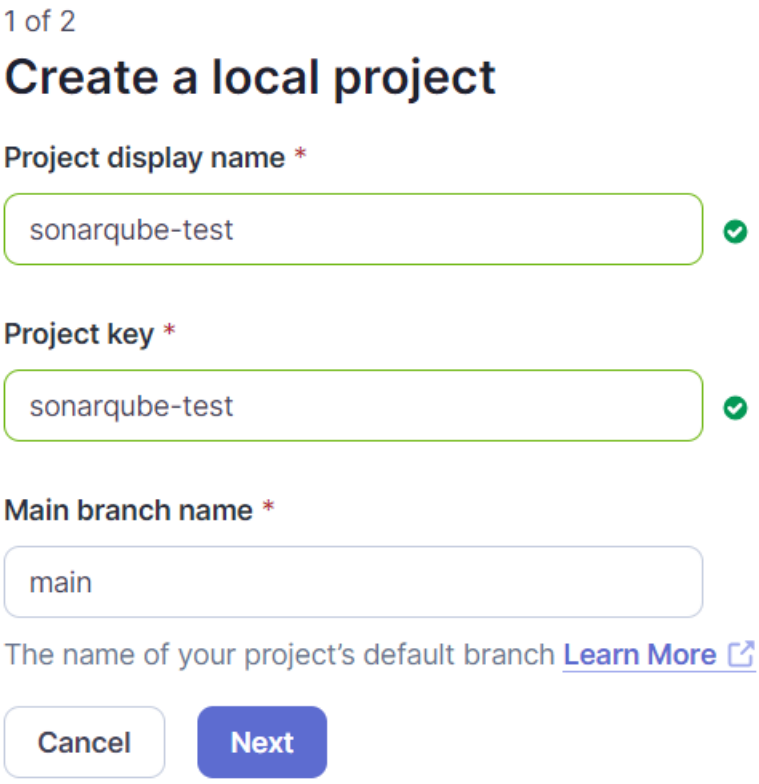
$ docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 Sonarqube:latest

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindow
s

PS C:\Users\Govind> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=
true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
f7b8d5f7157e7cb2643b22c497a78368667800258325d55d236330b6b6286f85
```

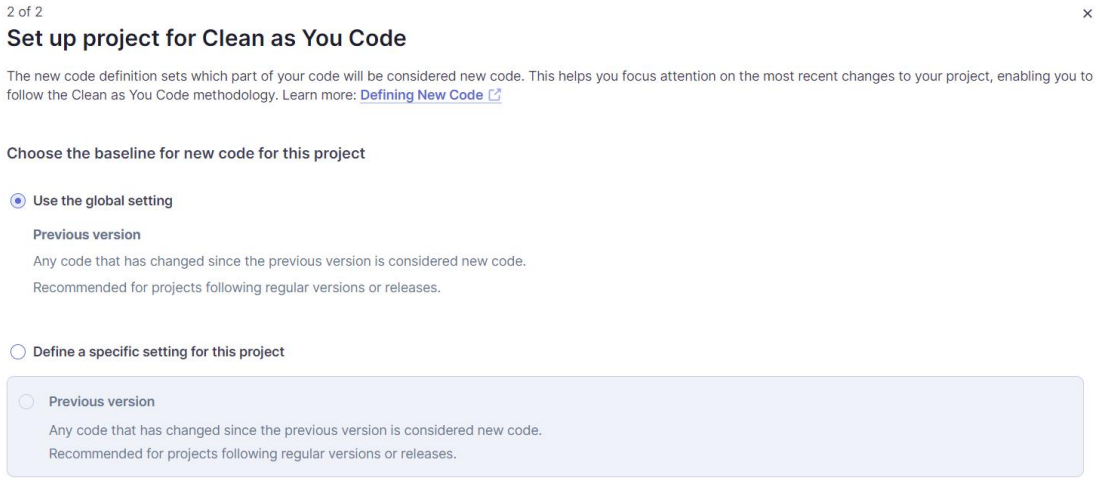
Step 2: Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



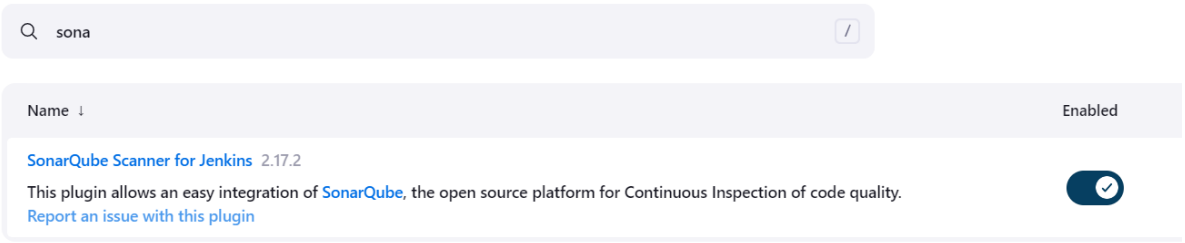
Step 3: Login to SonarQube using username admin and password admin then create a local project in SonarQube with the name sonarqube-test.

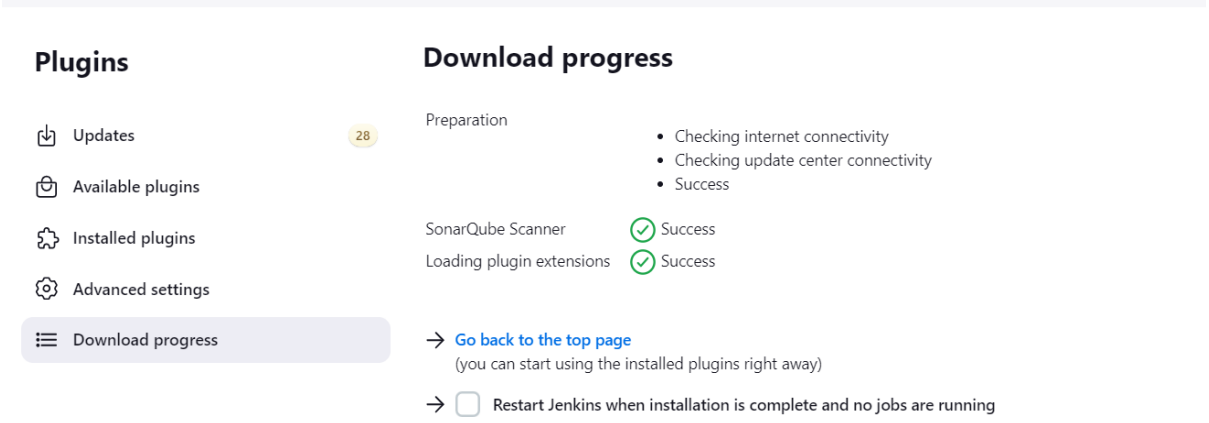


Step 4: Setup the project to use the global setting and open Jenkins Dashboard.

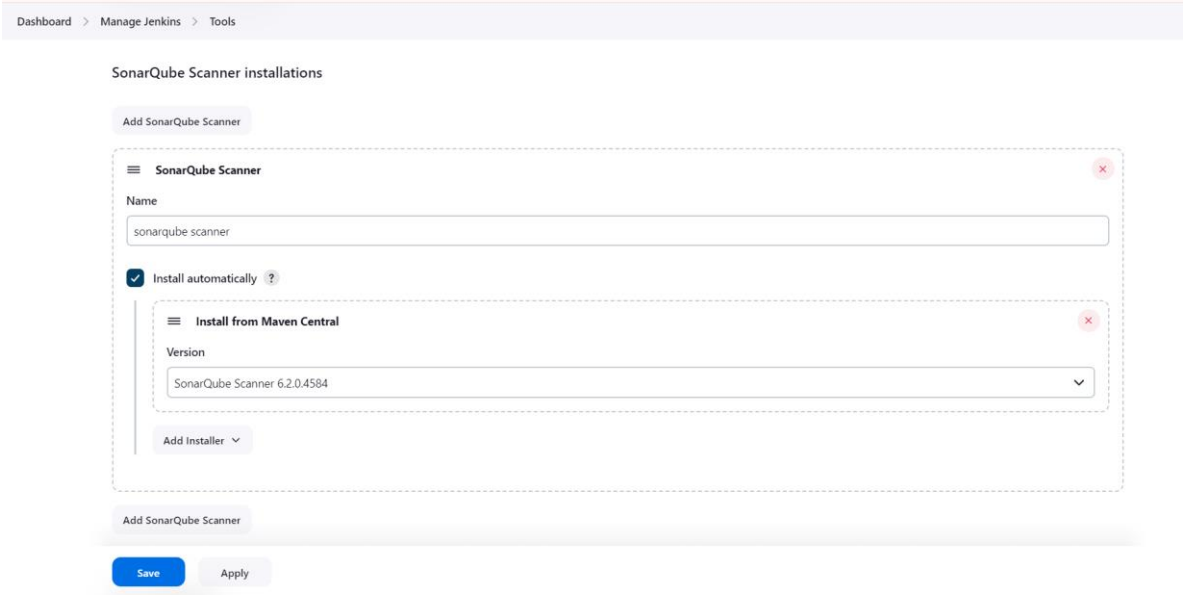


Step 5: Now, Go to Dashboard ->Manage Jenkins -> Plugin Manager and search for SonarQube Scanner under Available plugins for Jenkins and install it.

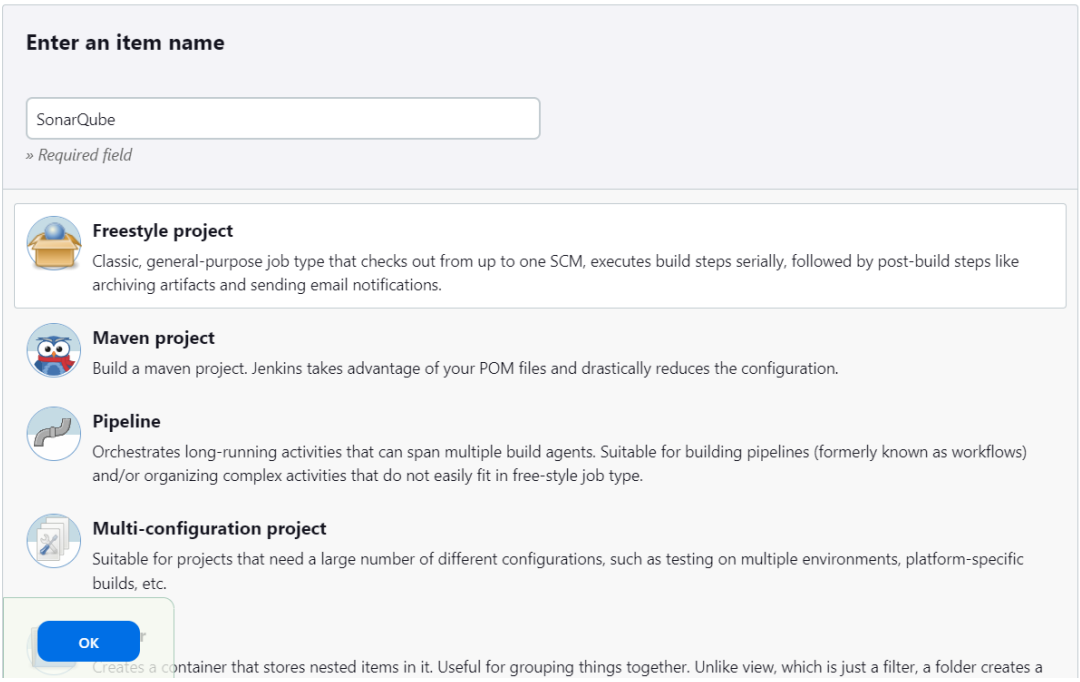




Step 6: Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details then Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.



Step 7: Create a New Item in Jenkins and choose a freestyle project.



Step 7: Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git

Under Build ->Execute SonarQube Scanner, enter these Analysis

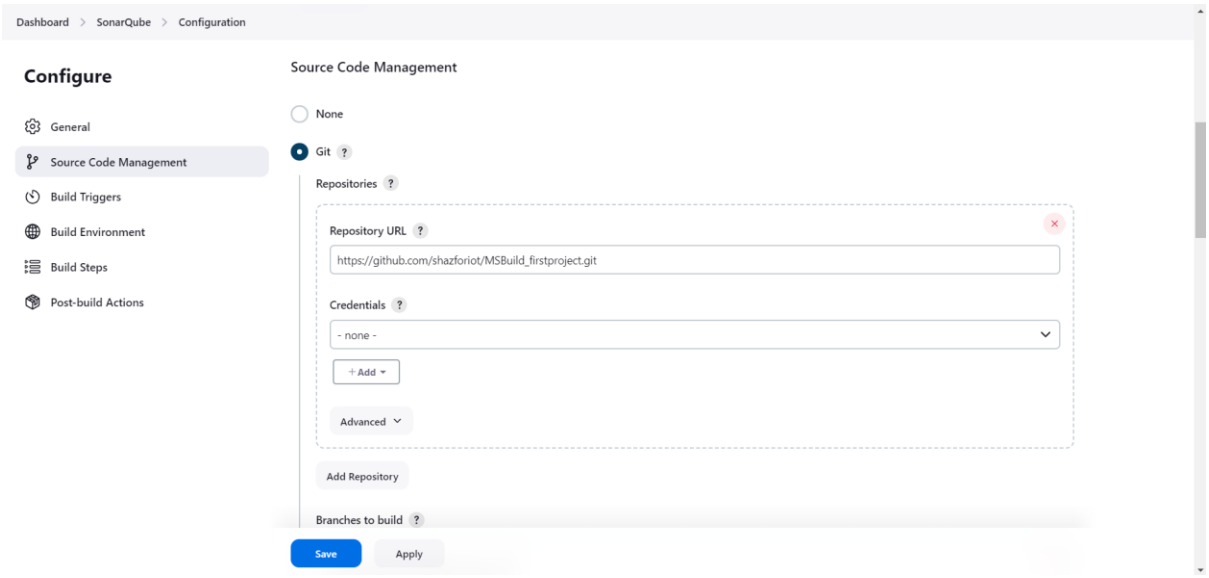
properties. Mention the SonarQube Project Key, Login, Password, and Host URL.

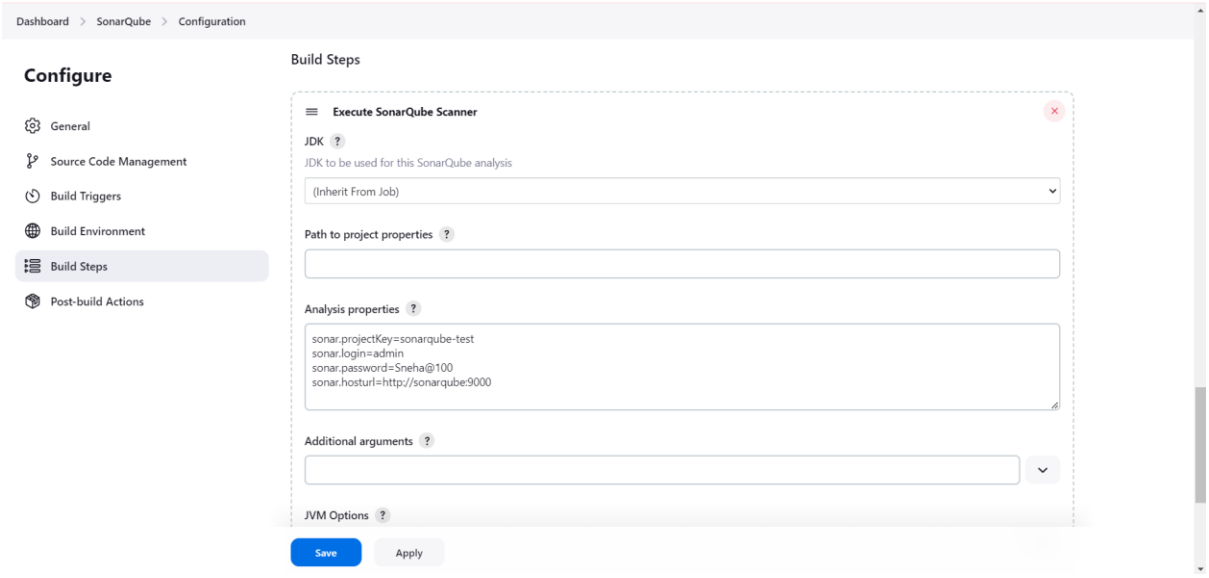
sonar.projectKey=sonarqube-test

sonar.login=admin

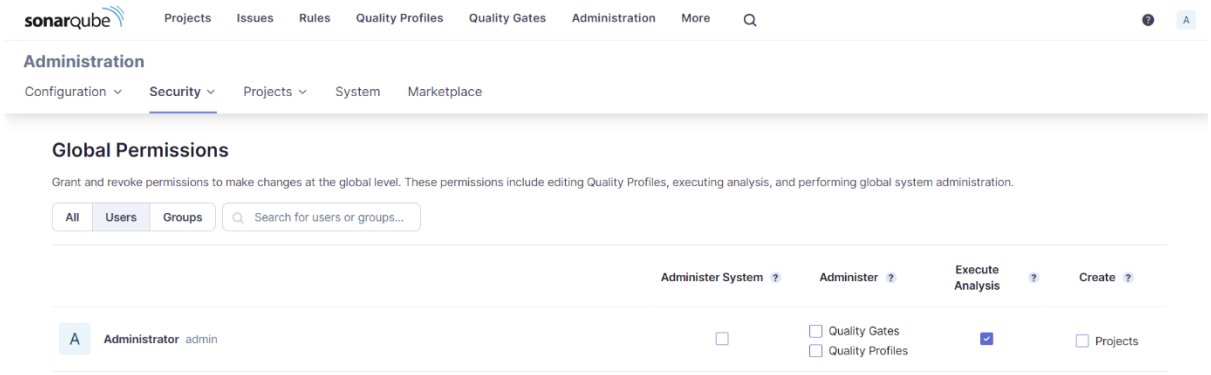
sonar.password=Sneha@100

sonar.hosturl=http://localhost:9000/

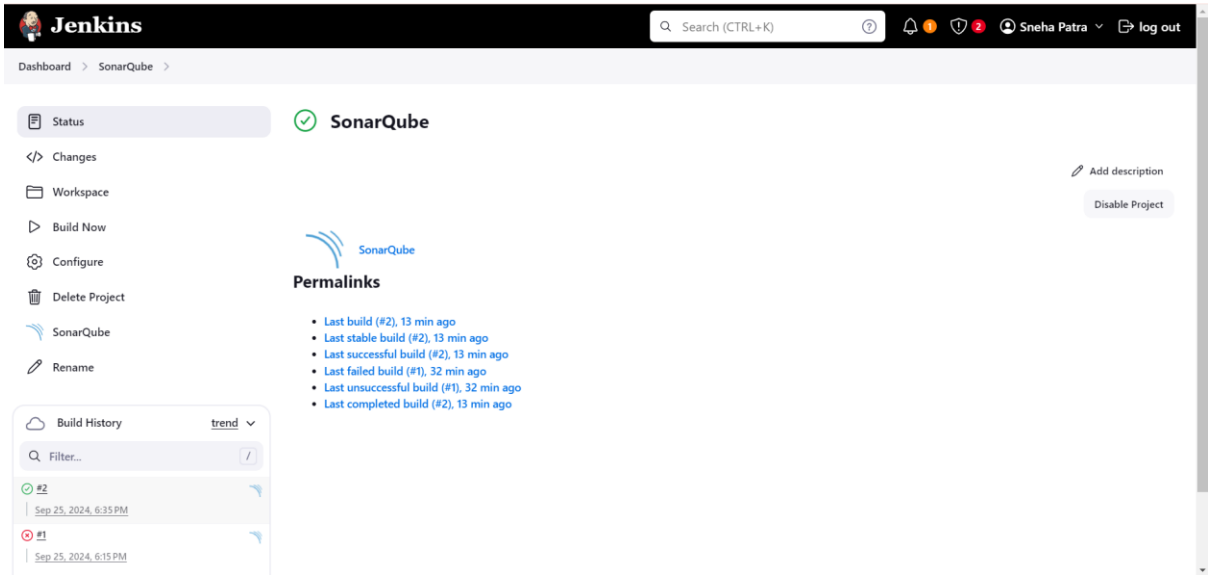


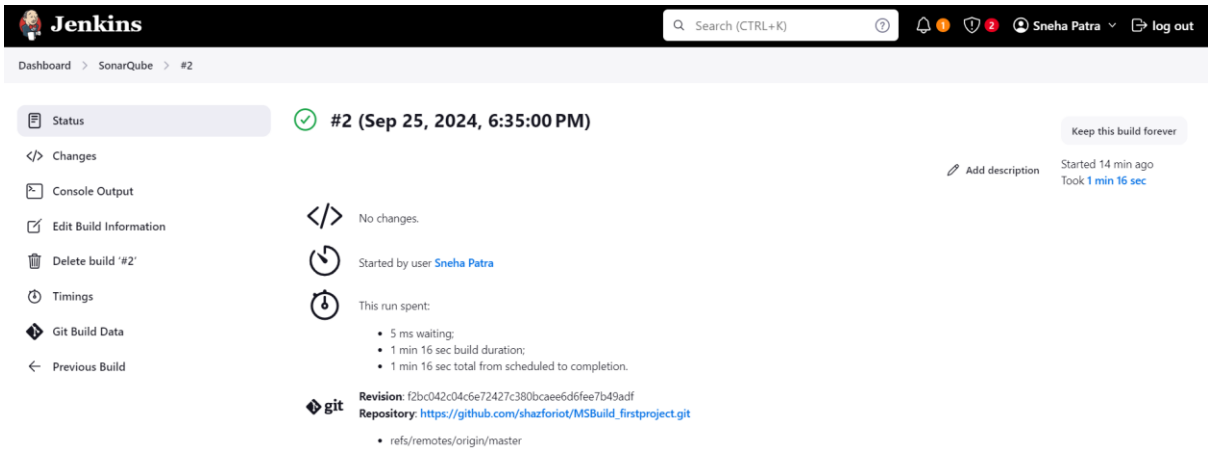


Step 8: Go to http://localhost:9000/ and enter your previously created username and password then Go to Permissions and grant the Admin user Execute Permissions.

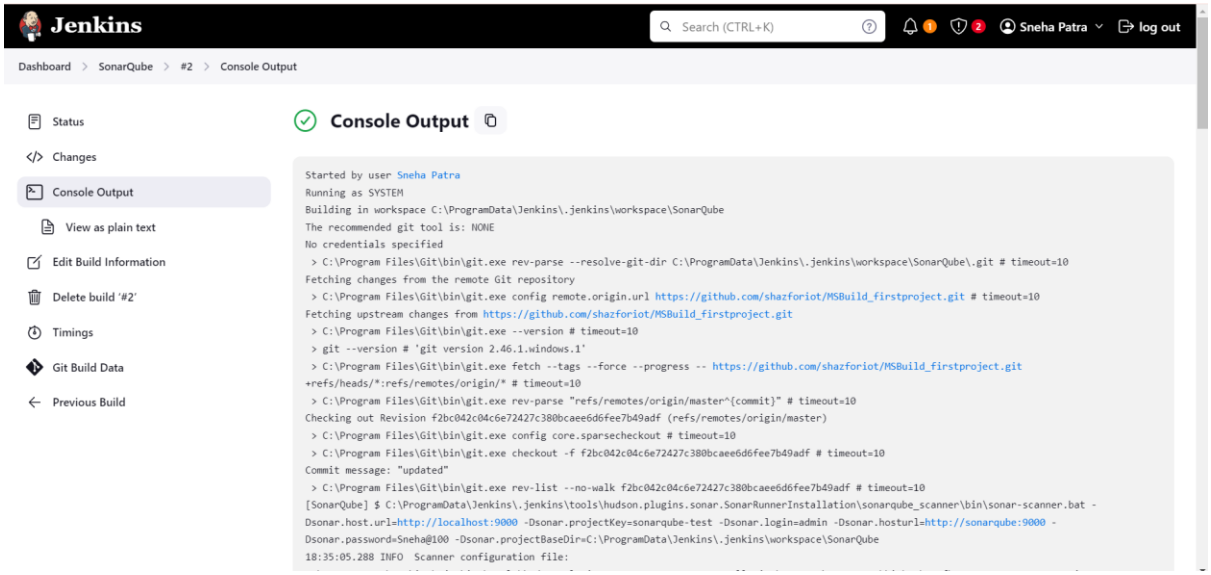


Step 9: Now Build and Run.

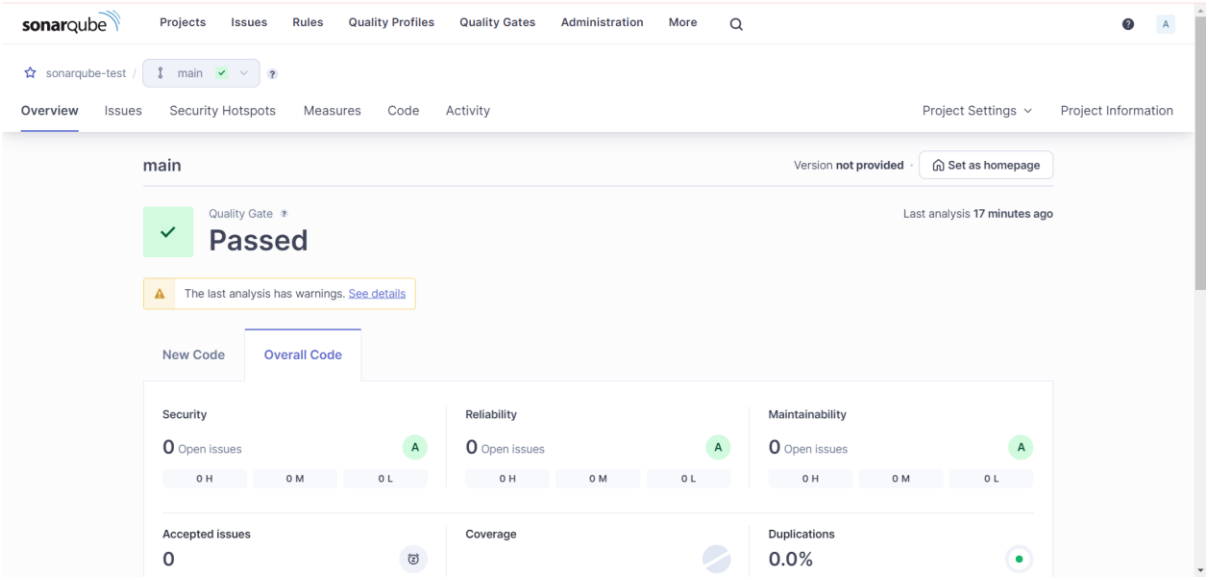




Step 10: Check the console output.



Step 11: Once the build is successful, check the project in SonarQube. In this way, we have integrated Jenkins with SonarQube for SAST.



Conclusion:

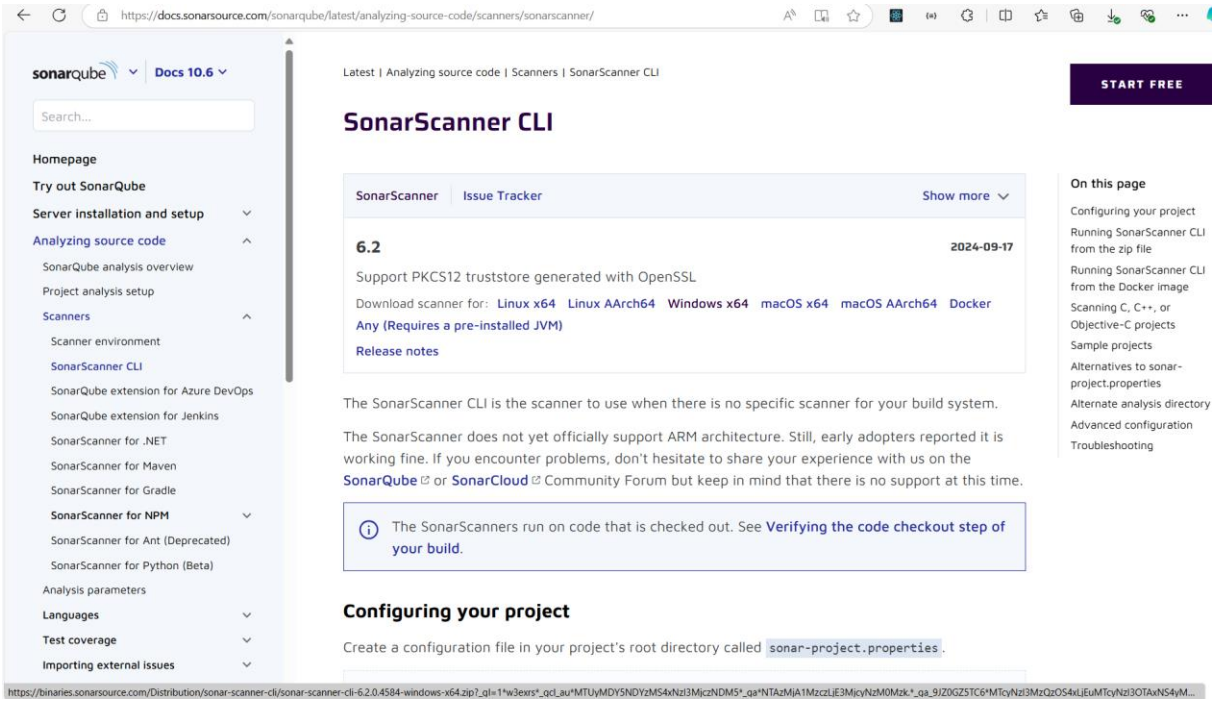
In this experiment, we have understood the importance of SAST and have successfully

integrated Jenkins with SonarQube for Static Analysis and Code Testing.

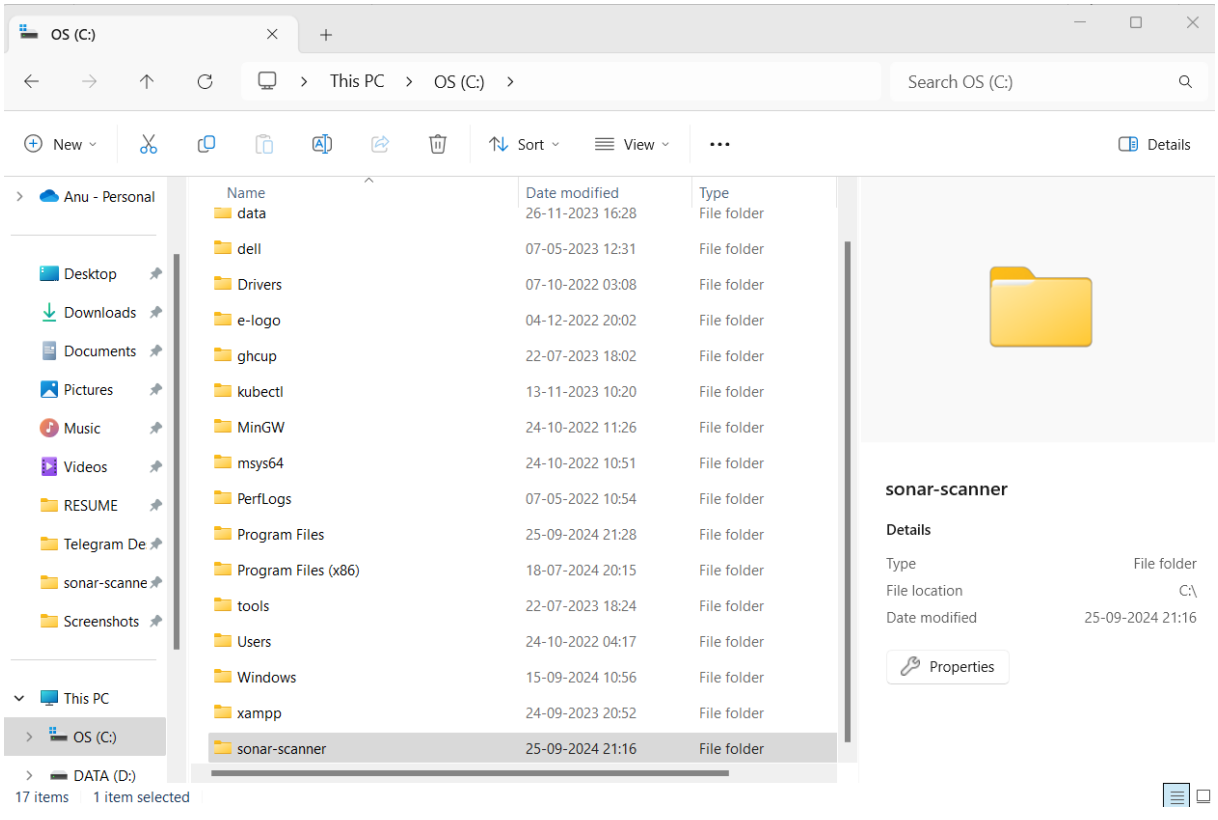
Experiment 8

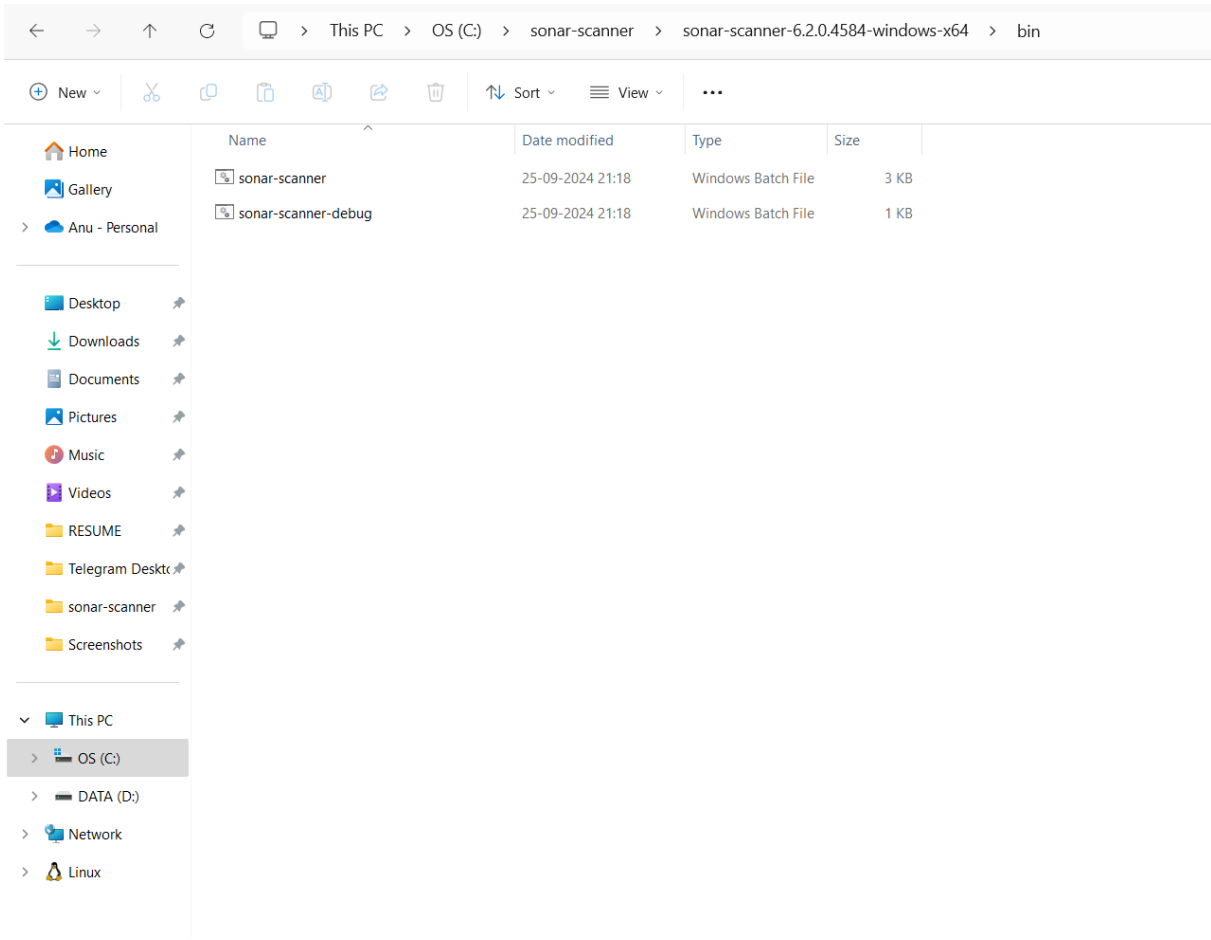
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web /Java / Python application.

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.

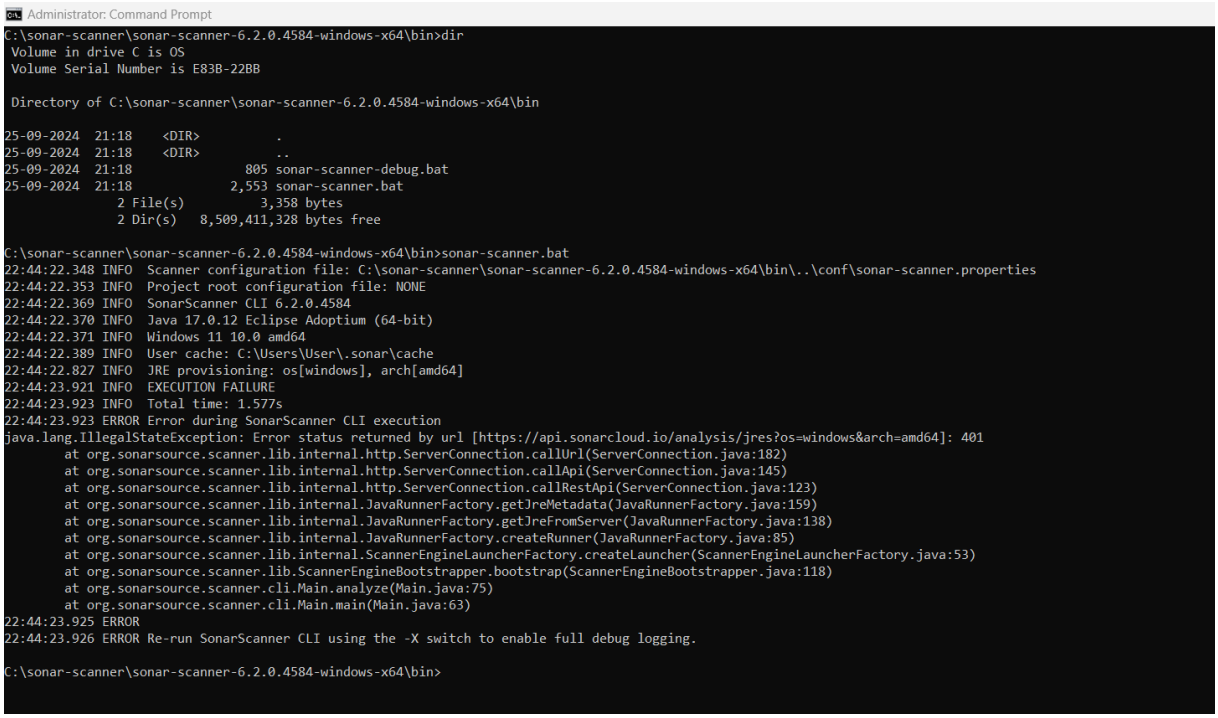


Step 2: Then extract the content in C drive and name the folder sonar-scanner

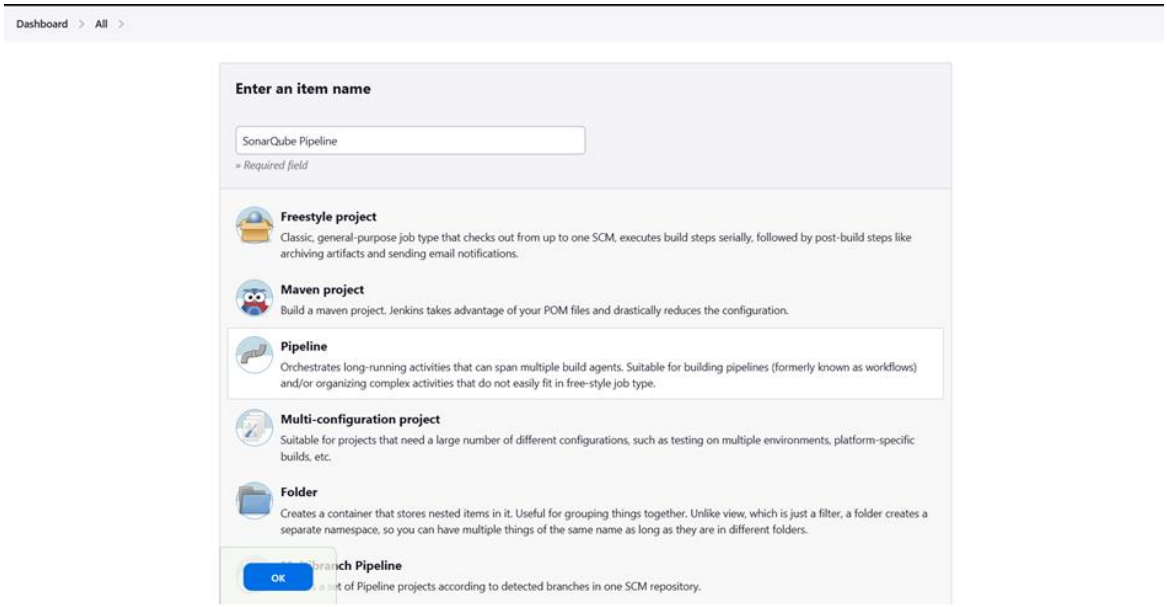




Step 3: Then now open Cmd Prompt and run as administrator and run the following commands –
cd C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin
dir
sonar-scanner.bat



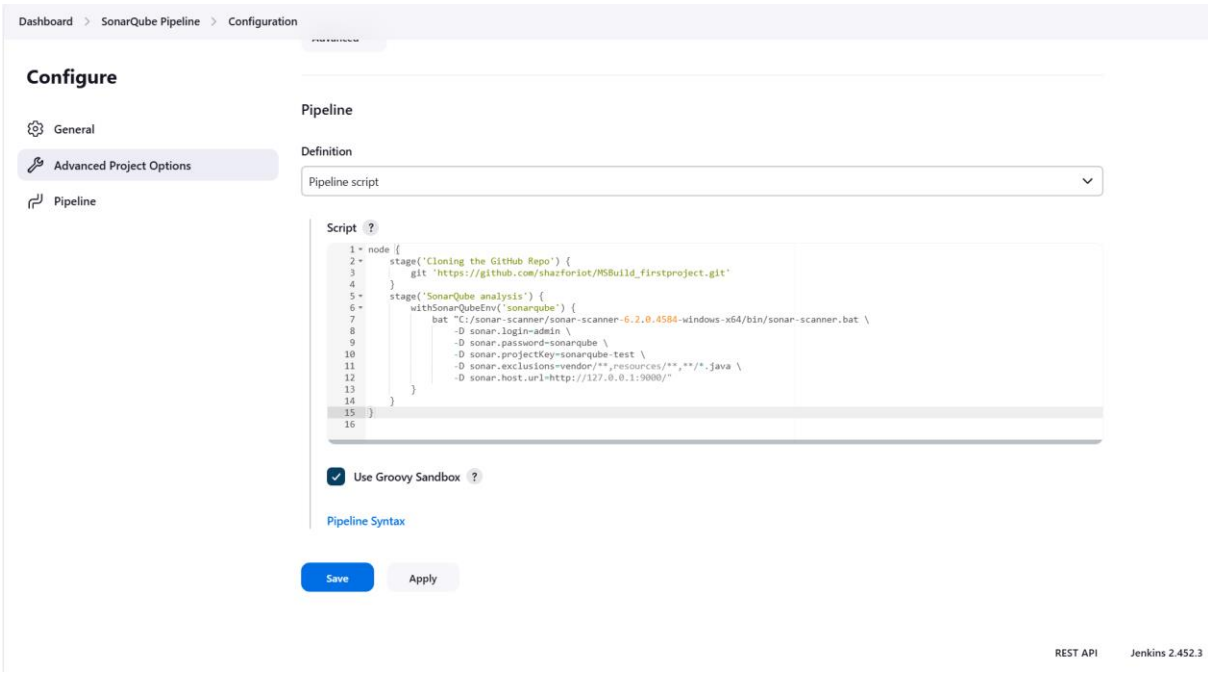
Step 4: Open Jenkins and create a pipeline and name the pipeline SonarQube Pipeline



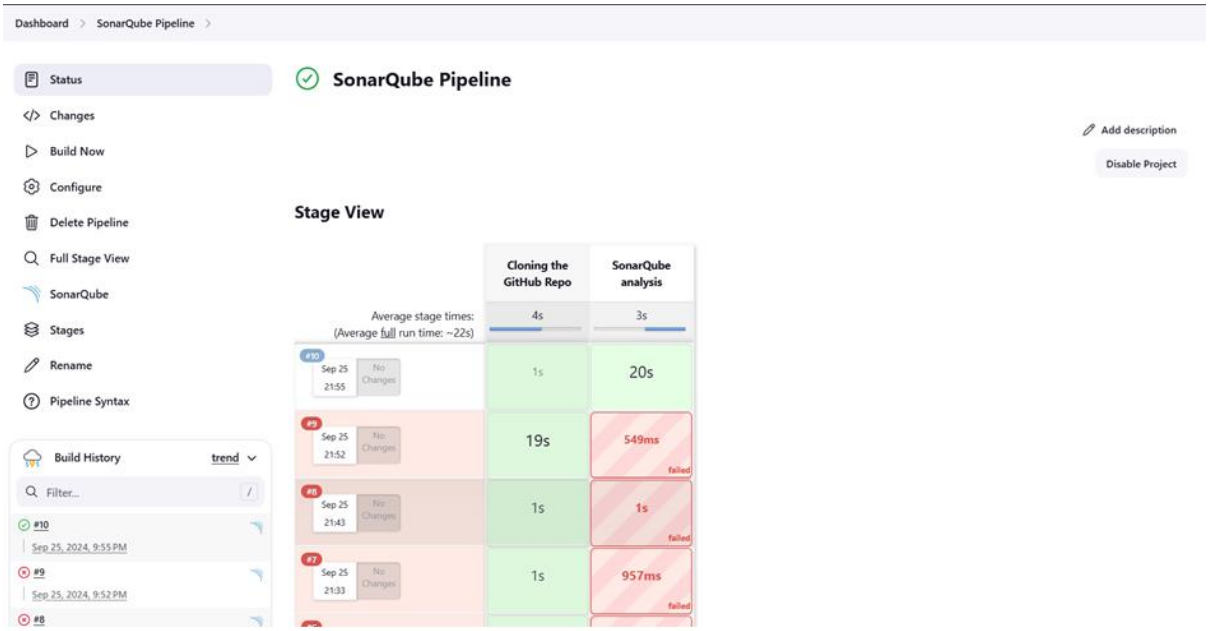
Step 5: In the configuration, under the Pipeline Section write the following Pipeline Script -

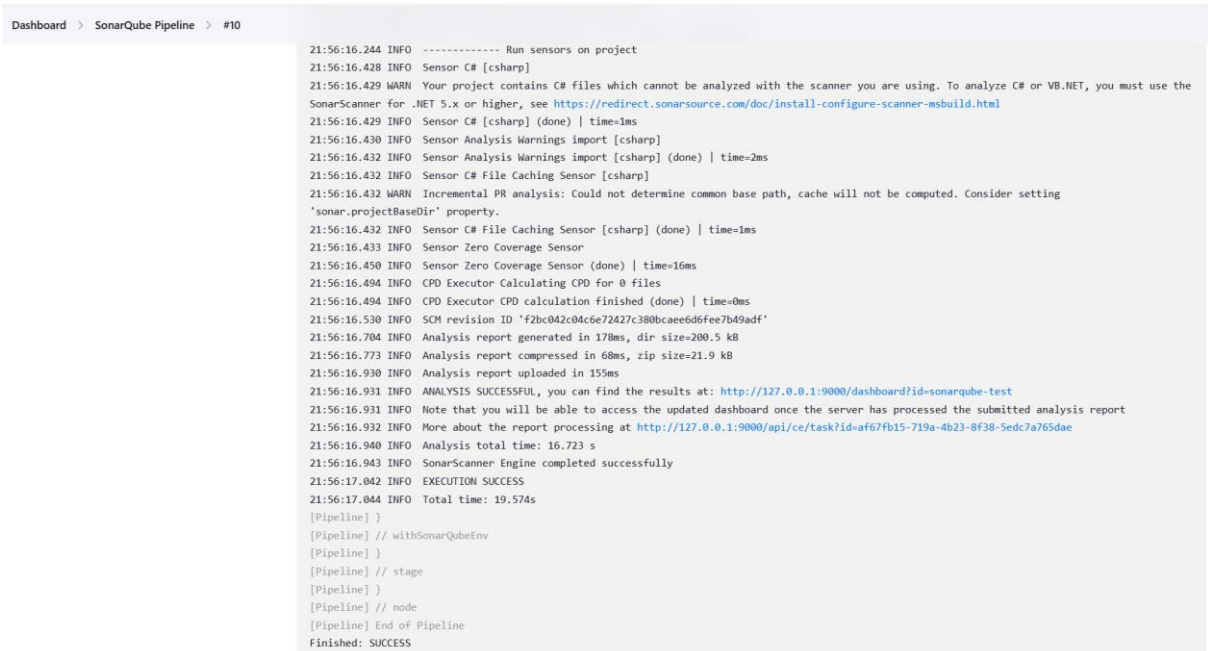
```
node {
   stage('Cloning the GitHub Repo') {
      git 'https://github.com/shazforiot/MSBuild_firstproject.git'
   }
   stage('SonarQube analysis') {
      withSonarQubeEnv('sonarqube') {
         bat "C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat \
            -D sonar.login=admin \
            -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.

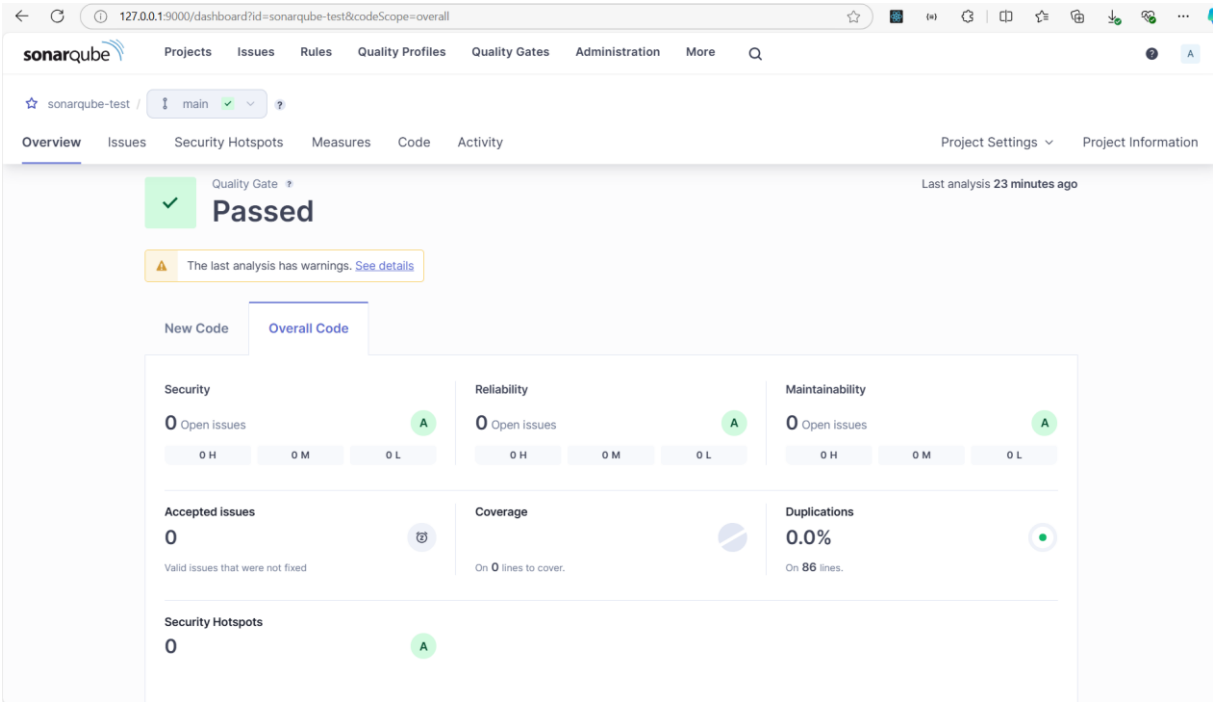


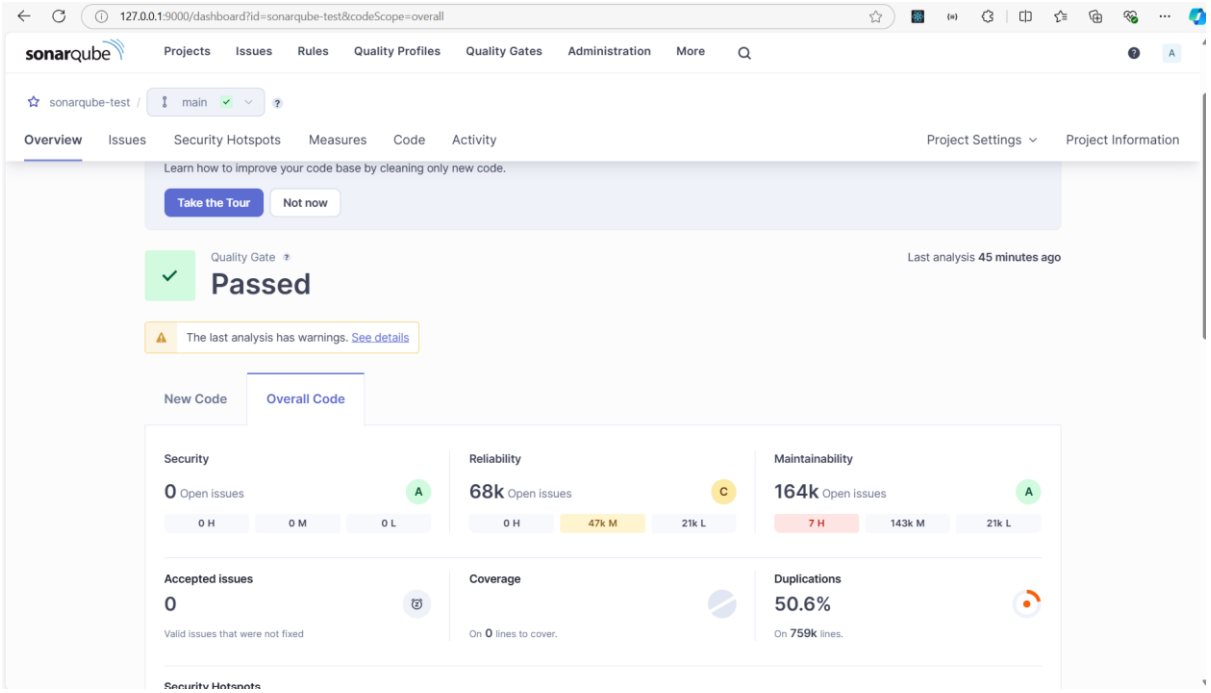
Step 6: Now, click on Build Now and the build is successful.





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





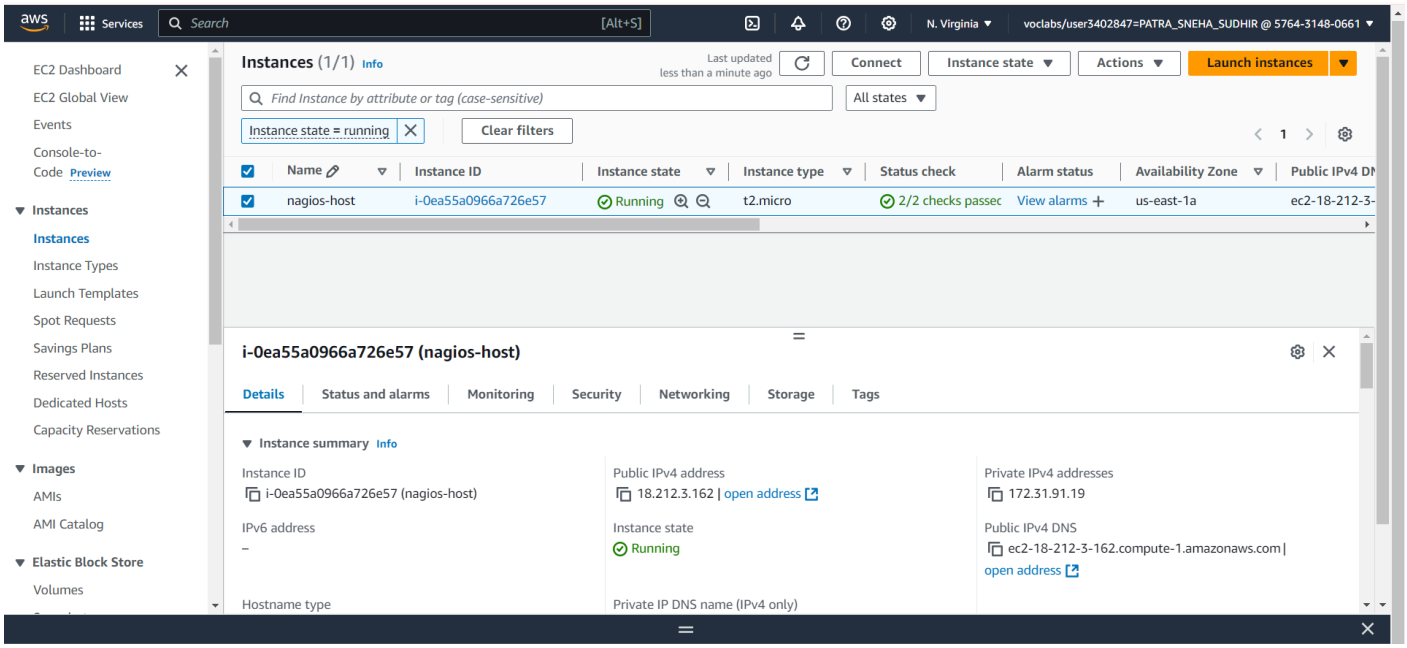
Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, andsecurity vulnerabilities on our sample Java application.

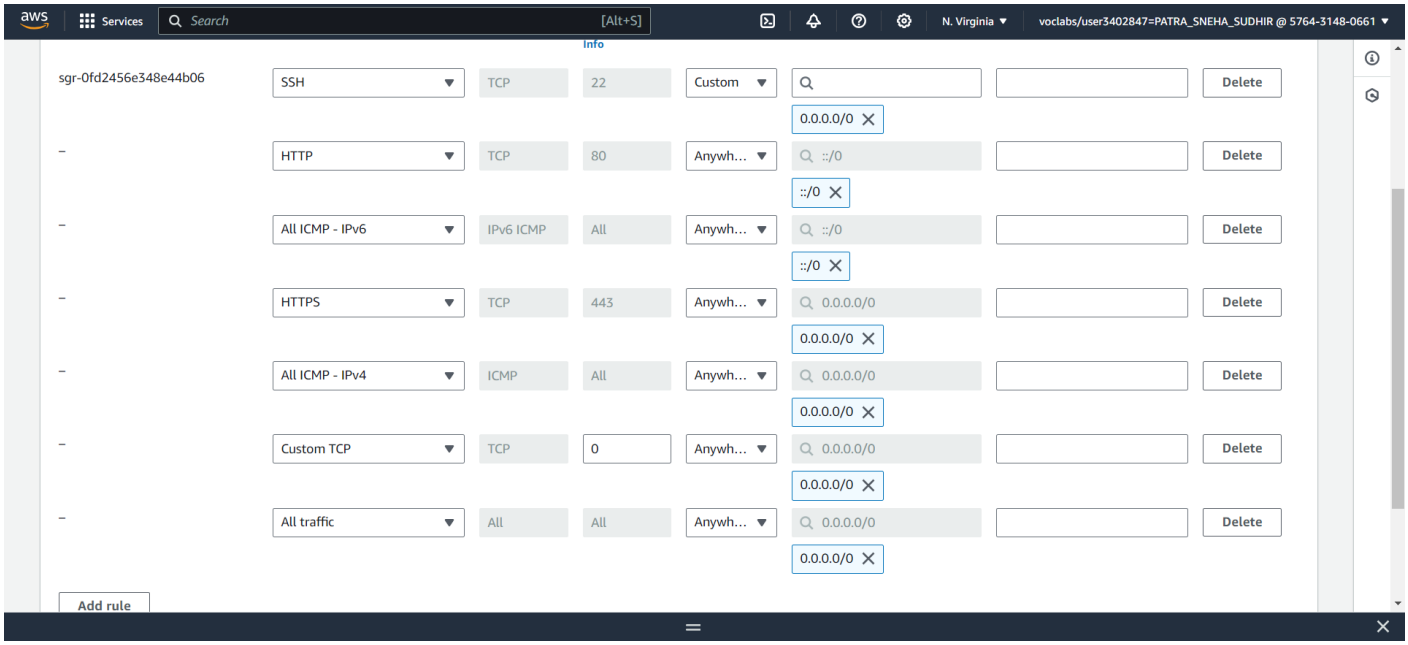
Experiment 9

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

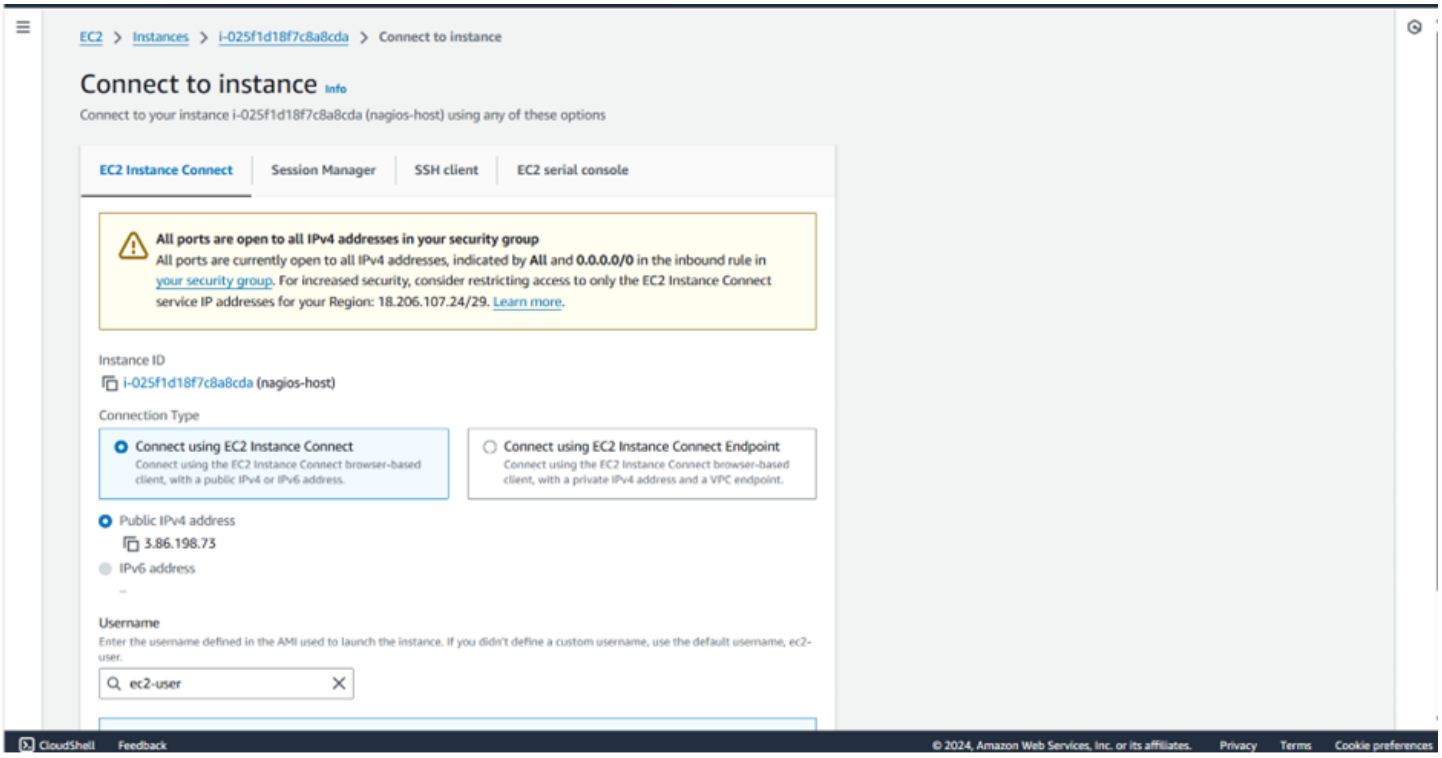
Step 1: Create an EC2 Instance and name it as nagios-host



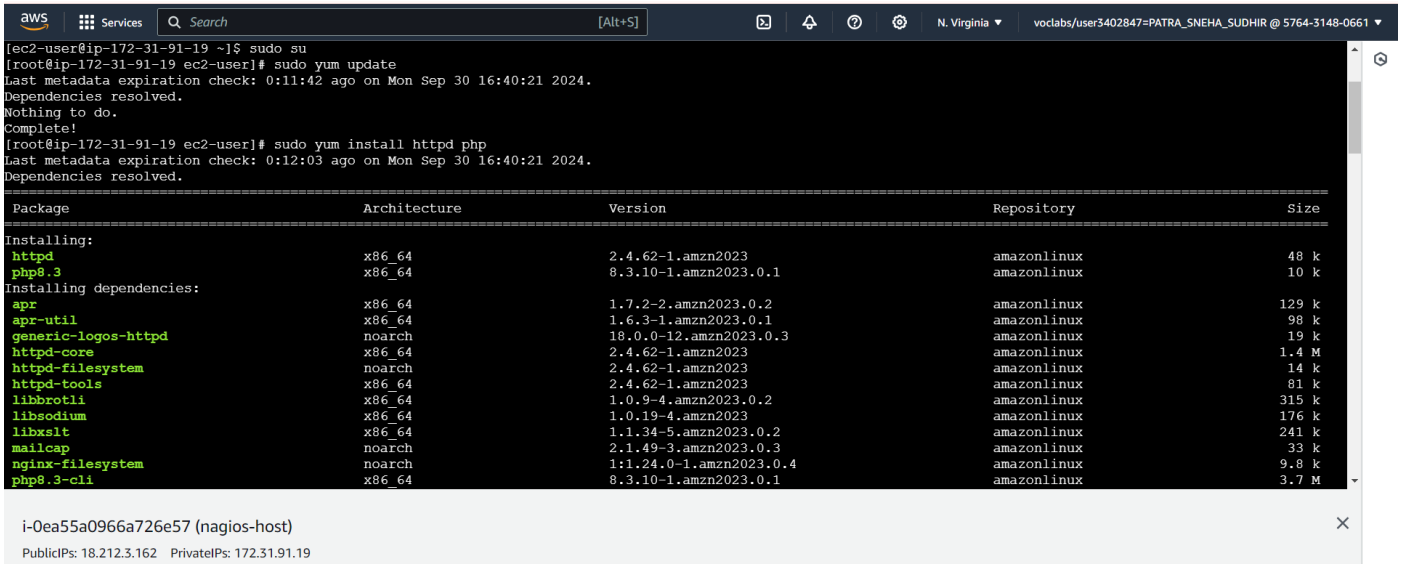
Step 2: Under the security groups, click on edit inbound rules and set as shown in the figure below

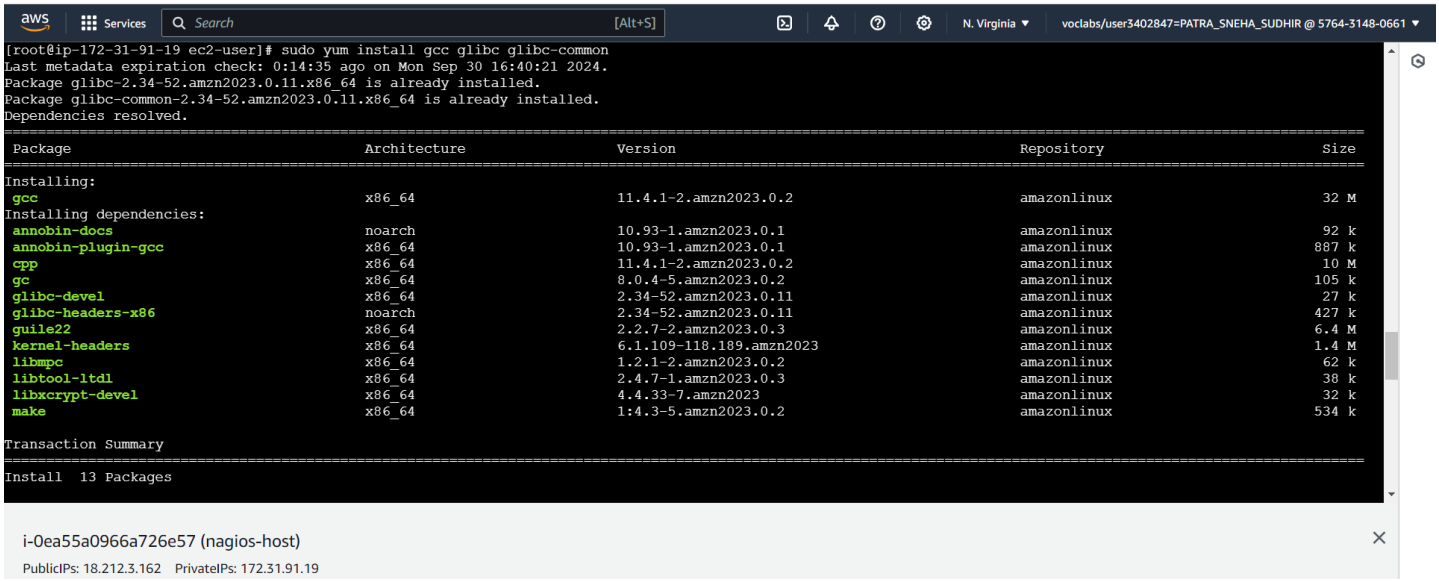


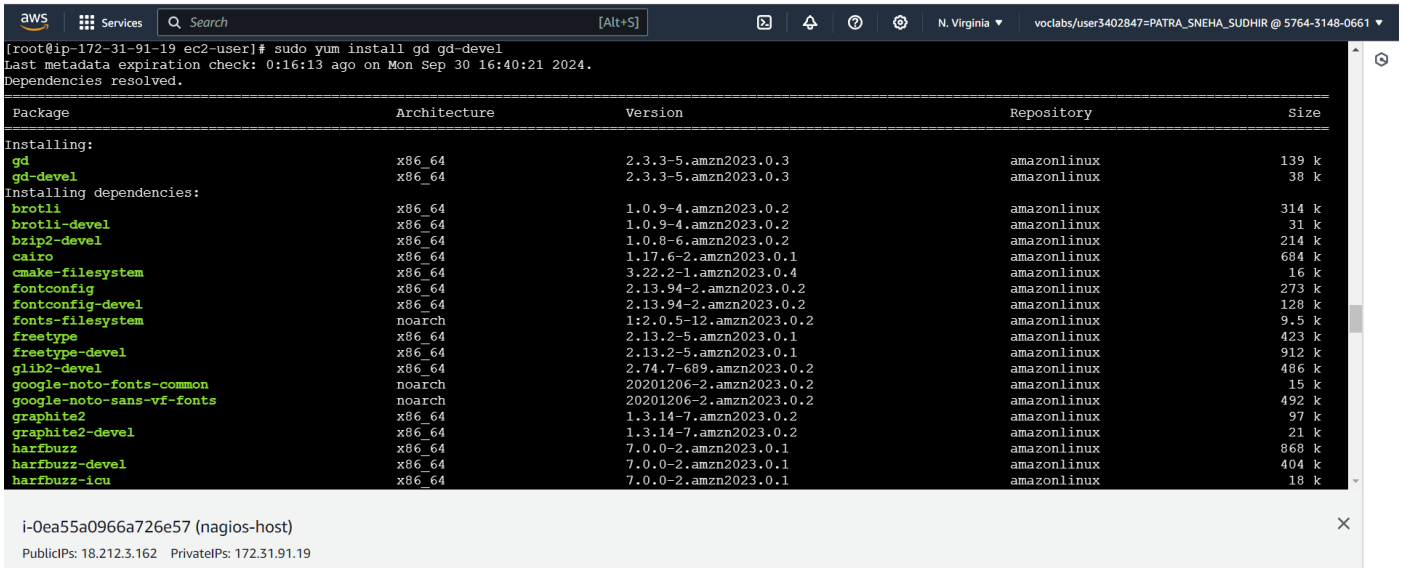
Step 3: Then select the instance nagios-host and then connect the instance.



Step 3: Now, run the following commands -
sudo su
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel







Step 4: Create a new nagios user with its password.

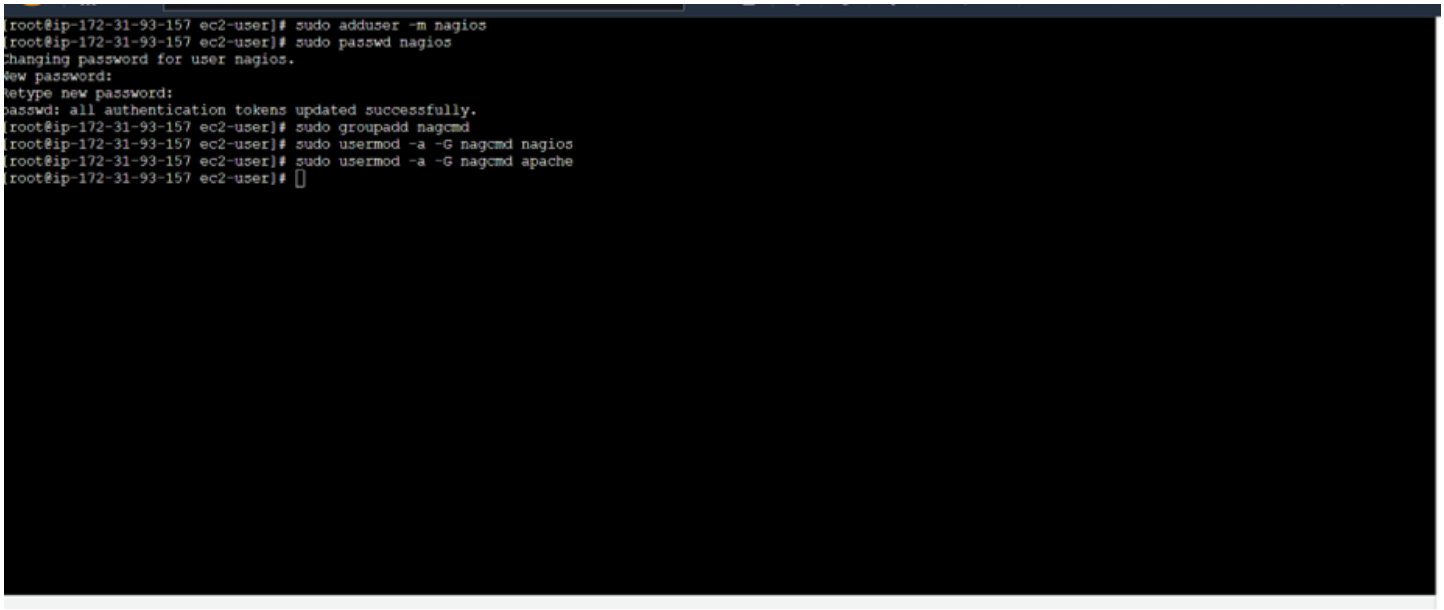
sudo adduser -m nagios

sudo passwd nagios

sudo groupadd nagcmd

sudo usermod -a -G nagcmd nagios

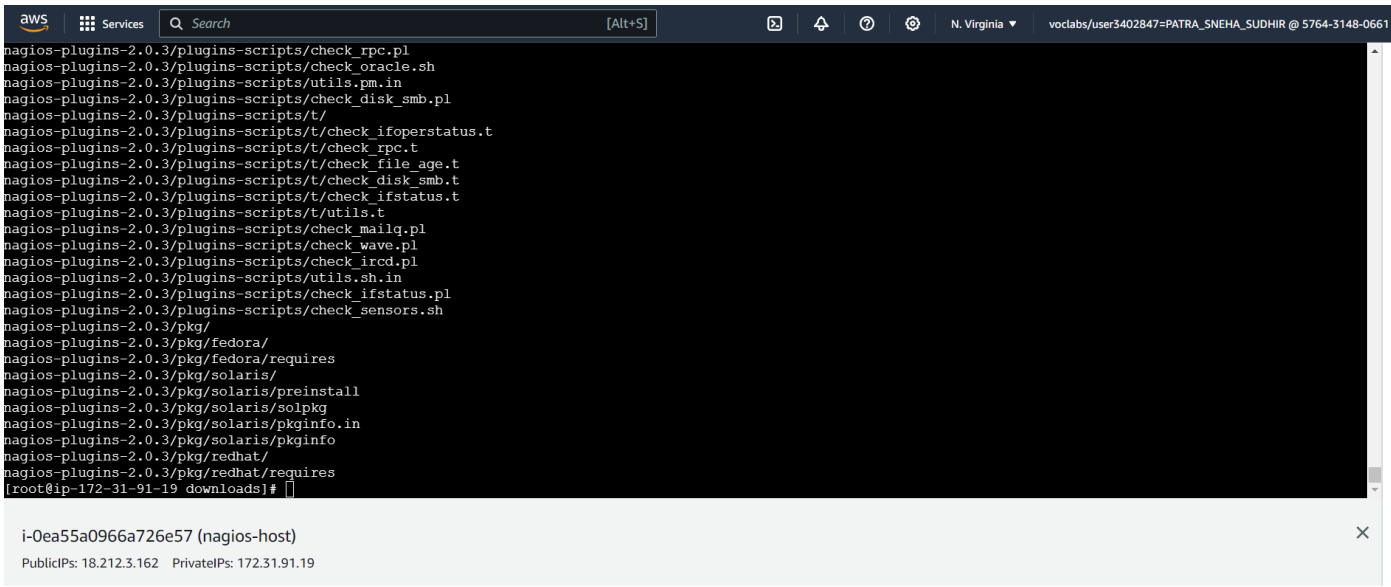
sudo usermod -a -G nagcmd apache



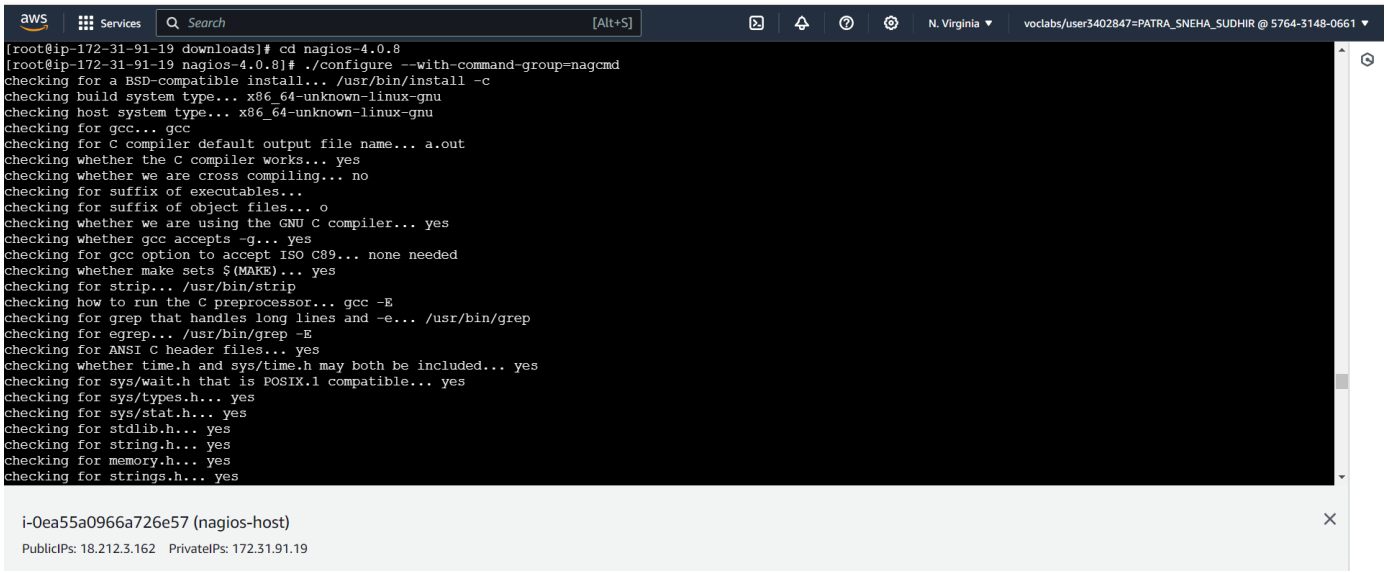
Step 5: Now, run the following commands -
mkdir ~/downloads
cd ~/downloads
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz

To resolve the error run the following commands -
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz
tar zxvf nagios-plugins-2.0.3.tar.gz
cd nagios-4.0.8

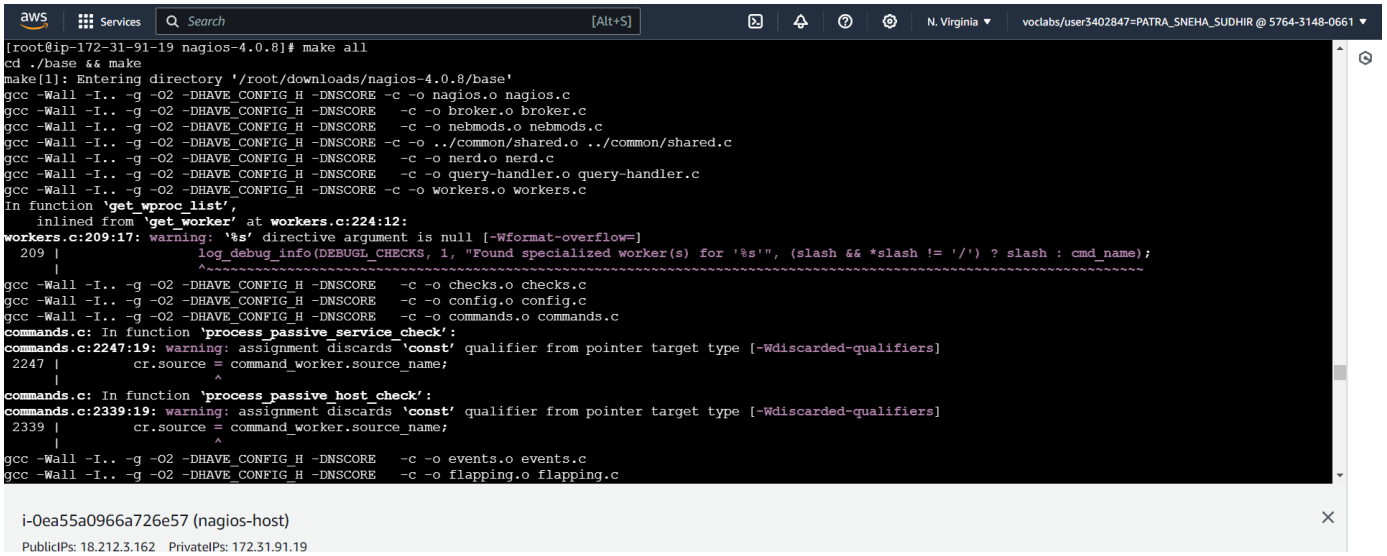
```
[root@ip-172-31-93-157 downloads]# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-09-30 17:03:04--  http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net)... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-30 17:03:04--  http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Reusing existing connection to prdownloads.sourceforge.net:80.
HTTP request sent, awaiting response... 302 Found
Location: http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1 [following]
--2024-09-30 17:03:04--  http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1
Resolving versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)... 162.251.232.173
Connecting to versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)|162.251.232.173|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz                 100%[=============================================================================>]   1.72M  2.21MB/s    in 0.8s

2024-09-30 17:03:05 (2.21 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

--2024-09-30 17:03:05--  http://nagios-plugins.org/download/nagios-plugins-2.0.3.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|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz.1'

nagios-plugins-2.0.3.tar.gz.1       100%[=============================================================================>]   2.54M  7.26MB/s    in 0.3s

2024-09-30 17:03:05 (7.26 MB/s) - 'nagios-plugins-2.0.3.tar.gz.1' saved [2659772/2659772]
```

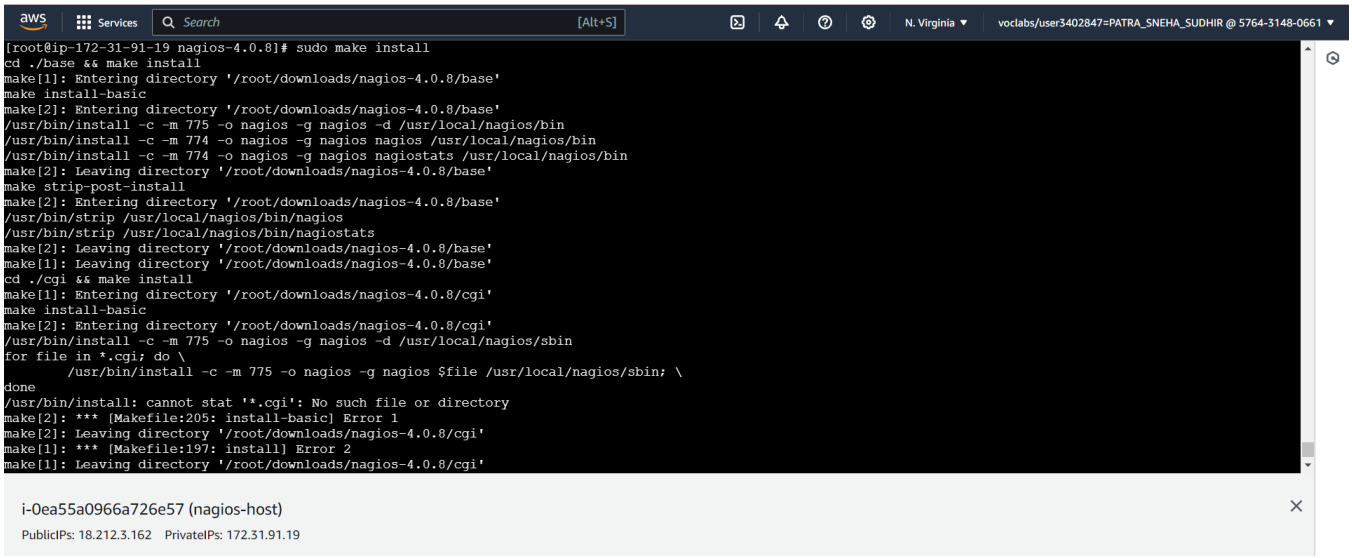


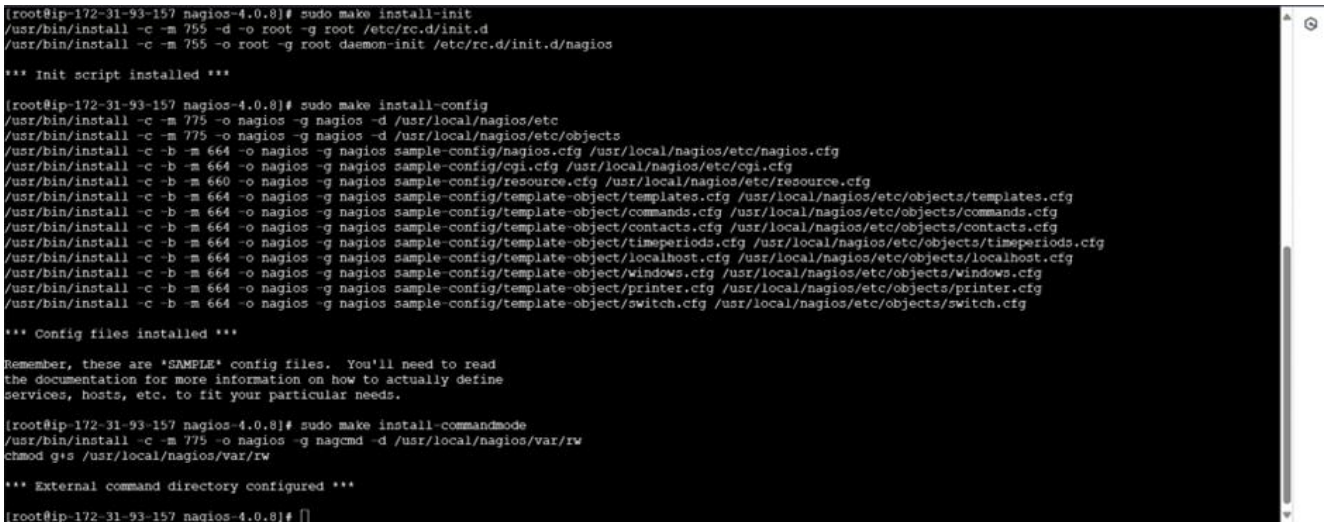
Step 6: Now to run the configuration script run the following command.
./configure --with-command-group=nagcmd



Step 7: Now, to compile the source code run the following command -
make all



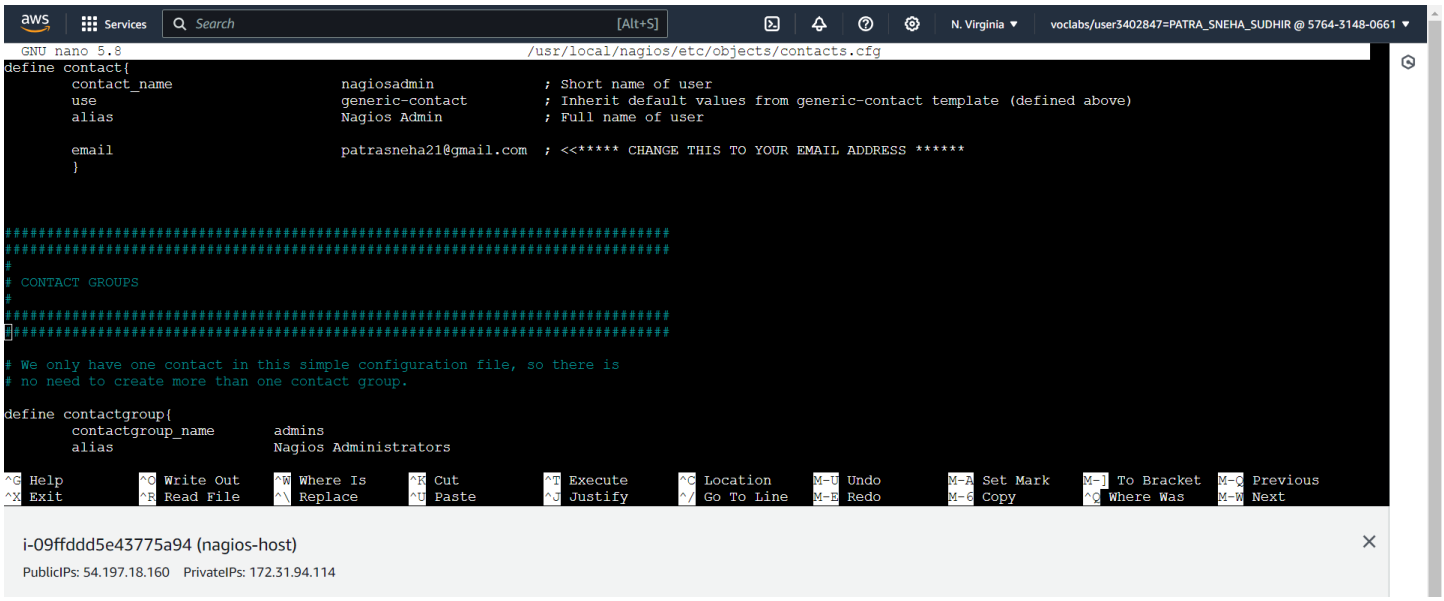




To resolve the errors run the following commands -
sudo yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel
rm -rf nagios-4.0.8
cd ~/downloads/nagios-4.4.6
./configure --with-command-group=nagcmd
make all
sudo make install

```
      web interface

  make install-classicui
     - This installs the classic theme for the Nagios
       web interface


*** 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.
```

Step 8: Edit the config file and change the email address.
sudo nano /usr/local/nagios/etc/objects/contacts.cfg



Step 9: Now run the following commands –
sudo make install-webconf
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
sudo service httpd restart
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz

```
    - 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.

[root@ip-172-31-93-157 nagios-4.4.6]# sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[root@ip-172-31-93-157 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 ***

[root@ip-172-31-93-157 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
```

```
[root@ip-172-31-93-157 nagios-4.4.6]# sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ip-172-31-93-157 nagios-4.4.6]# cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
nagios-plugins-2.0.3/
nagios-plugins-2.0.3/perlmods/
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.in
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.am
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.0.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile
nagios-plugins-2.0.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.0.3/perlmods/install_order
nagios-plugins-2.0.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.0.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.0.3/ABOUT-NLS
nagios-plugins-2.0.3/configure.ac
nagios-plugins-2.0.3/Makefile.in
nagios-plugins-2.0.3/config.h.in
nagios-plugins-2.0.3/ChangeLog
nagios-plugins-2.0.3/AUTHORS
nagios-plugins-2.0.3/lib/
nagios-plugins-2.0.3/lib/parse_ini.h
nagios-plugins-2.0.3/lib/extra_opts.c
nagios-plugins-2.0.3/lib/Makefile.in
```

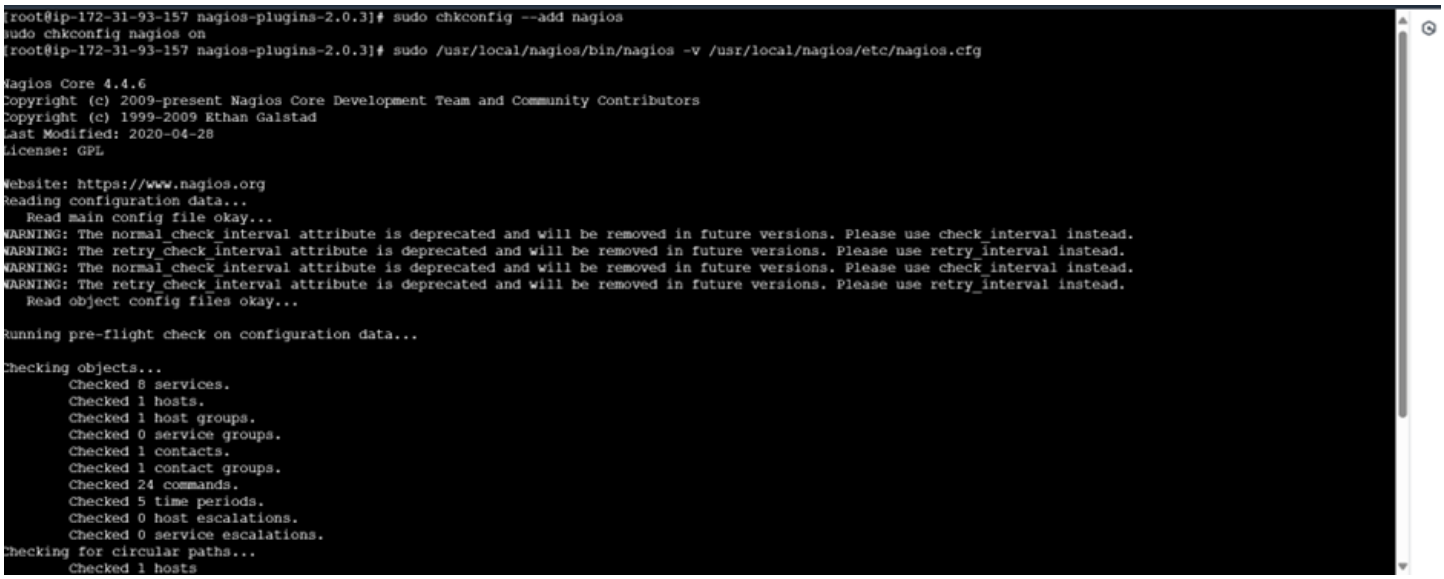
Step 10: Compile and install plugins
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install

```
/usr/bin/install -c -o nagios -g nagios check_dhcp /usr/local/nagios/libexec/check_dhcp
chown root /usr/local/nagios/libexec/check_dhcp
chmod ug=rx,u+s /usr/local/nagios/libexec/check_dhcp
/usr/bin/install -c -o nagios -g nagios check_icmp /usr/local/nagios/libexec/check_icmp
chown root /usr/local/nagios/libexec/check_icmp
chmod ug=rx,u+s /usr/local/nagios/libexec/check_icmp
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/plugins-root'
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/plugins-root'
Making install in po
make[1]: Entering directory '/root/downloads/nagios-plugins-2.0.3/po'
/usr/bin/mkdir -p /usr/local/nagios/share
installing fr.gmo as /usr/local/nagios/share/locale/fr/LC_MESSAGES/nagios-plugins.mo
installing de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo
if test "nagios-plugins" = "gettext-tools"; then \
  /usr/bin/mkdir -p /usr/local/nagios/share/gettext/po; \
  for file in Makefile.in.in remove-potcdate.sin    Makevars.template; do \
    /usr/bin/install -c -o nagios -g nagios -m 644 ./$file \
                  /usr/local/nagios/share/gettext/po/$file; \
  done; \
  for file in Makevars; do \
    rm -f /usr/local/nagios/share/gettext/po/$file; \
  done; \
else \
  : ; \
fi
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/po'
make[1]: Entering directory '/root/downloads/nagios-plugins-2.0.3'
make[2]: Entering directory '/root/downloads/nagios-plugins-2.0.3'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/root/downloads/nagios-plugins-2.0.3'
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3'
[root@ip-172-31-93-157 nagios-plugins-2.0.3]#
```

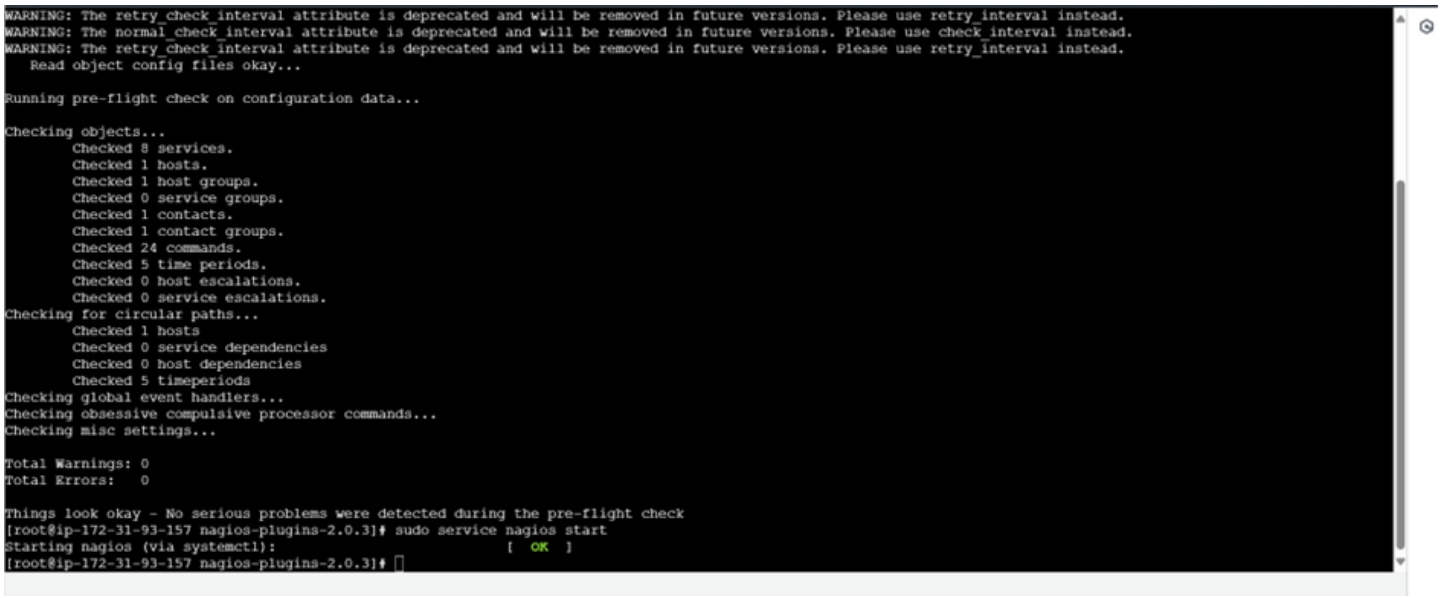
Step 11: To start nagios run the following commands –
sudo chkconfig --add nagios

sudo chkconfig nagios on

Verify using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



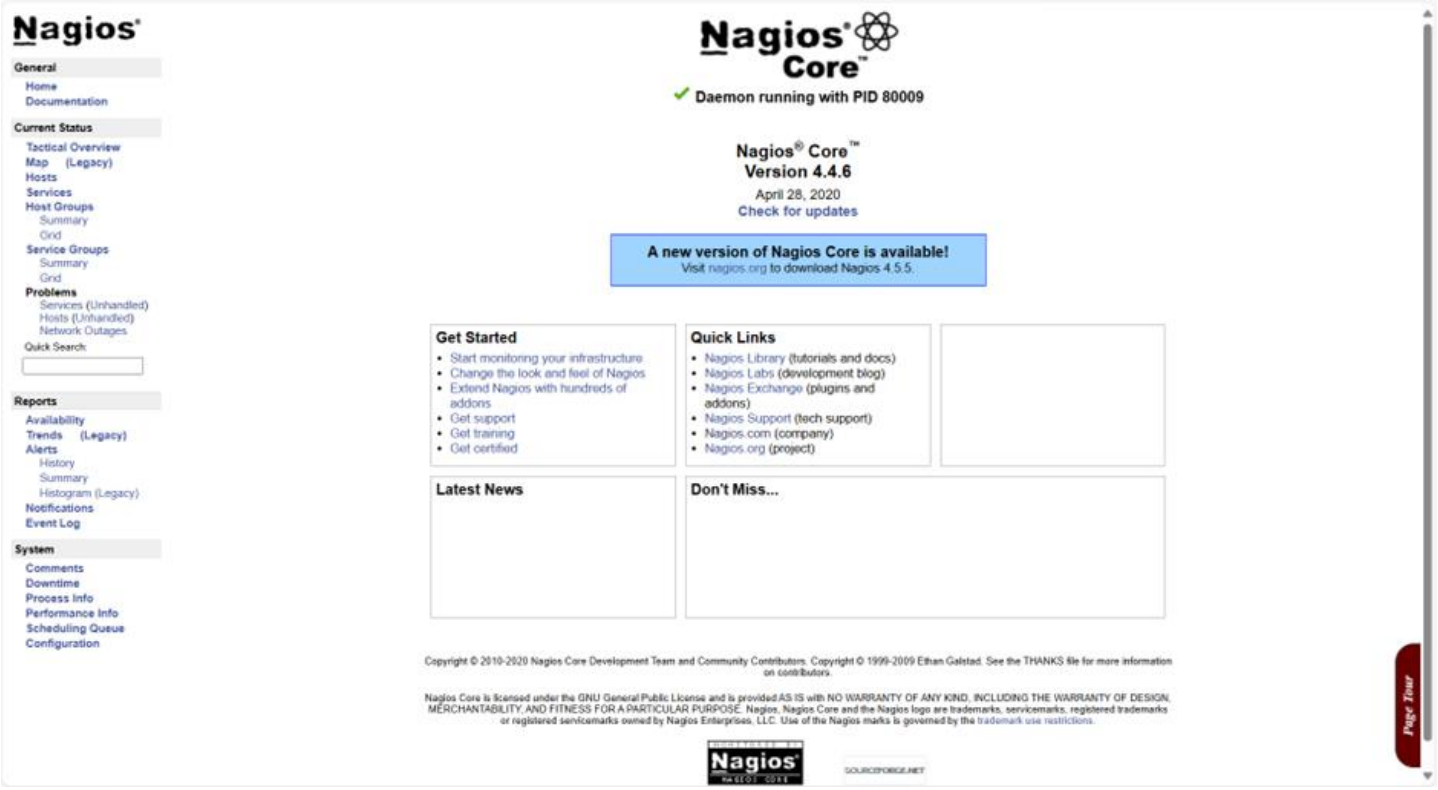
If there are no errors run the following command –
sudo service nagios start



Check status using the following command -
sudo systemctl status nagios

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
     Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
     Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1112)
     Memory: 2.2M
        CPU: 52ms
     CGroup: /system.slice/nagios.service
             ├─80009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─80011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─80037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Pl
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Pl
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file worker with pid 80037
lines 1-26/26 (END)
```

Step 12: Go to EC2 instance and copy the public IP address of the instance. Now visit http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.



Conclusion: Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) was installed and configured.

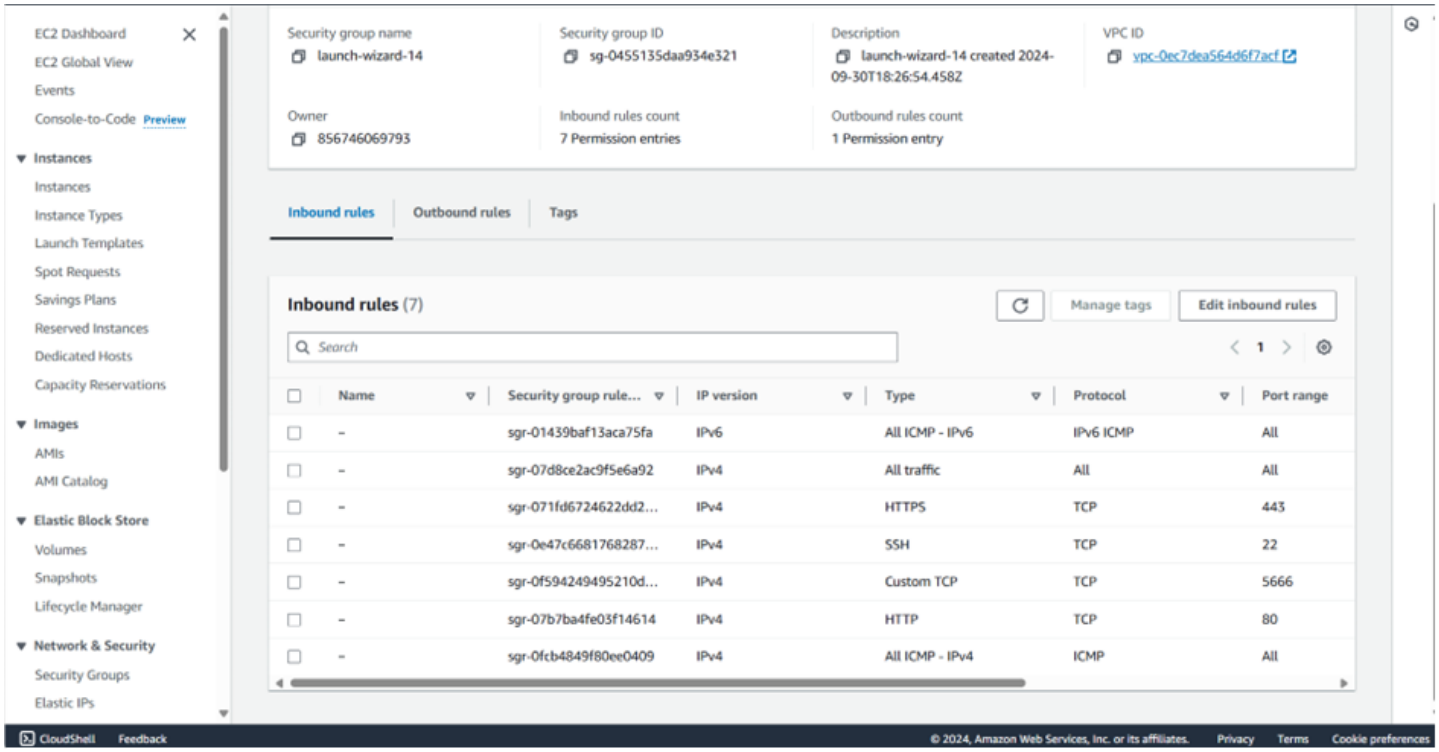
Experiment 10

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

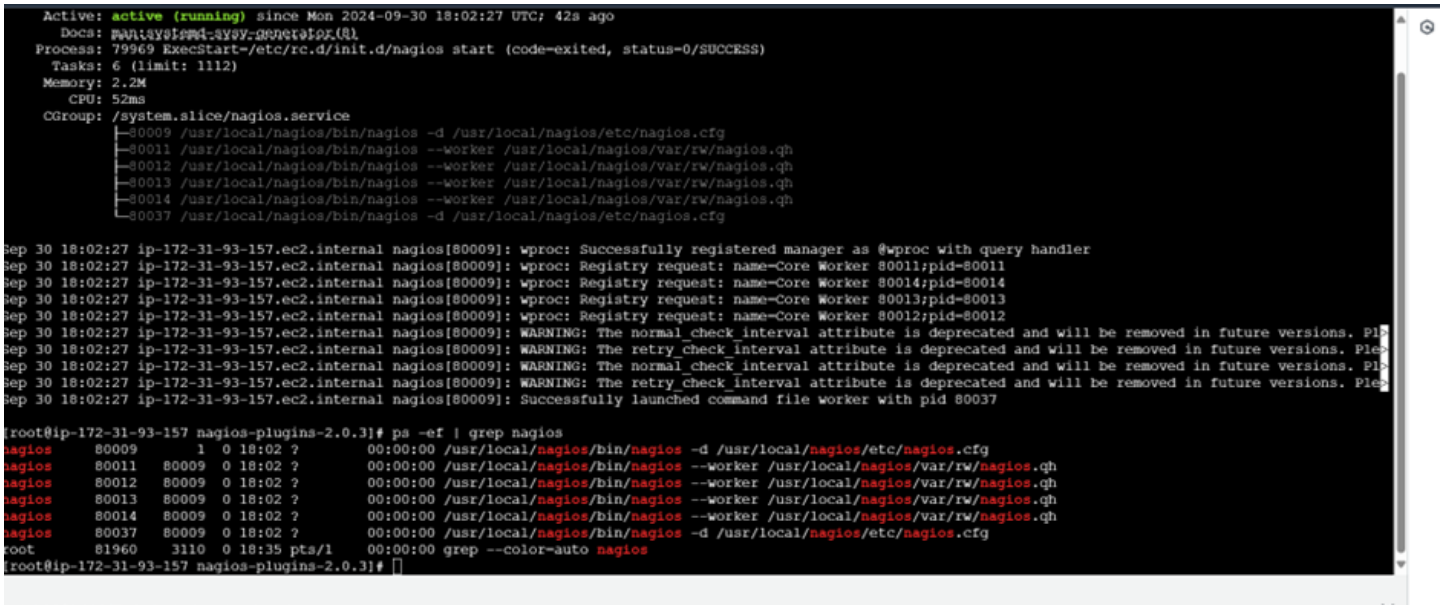
Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -
sudo systemctl status nagios
on the nagios-host instance.

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
     Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
     Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1112)
     Memory: 2.2M
        CPU: 52ms
     CGroup: /system.slice/nagios.service
             ├─80009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─80011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─80037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Ple
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file worker with pid 80037
lines 1-26/26 (END)
```

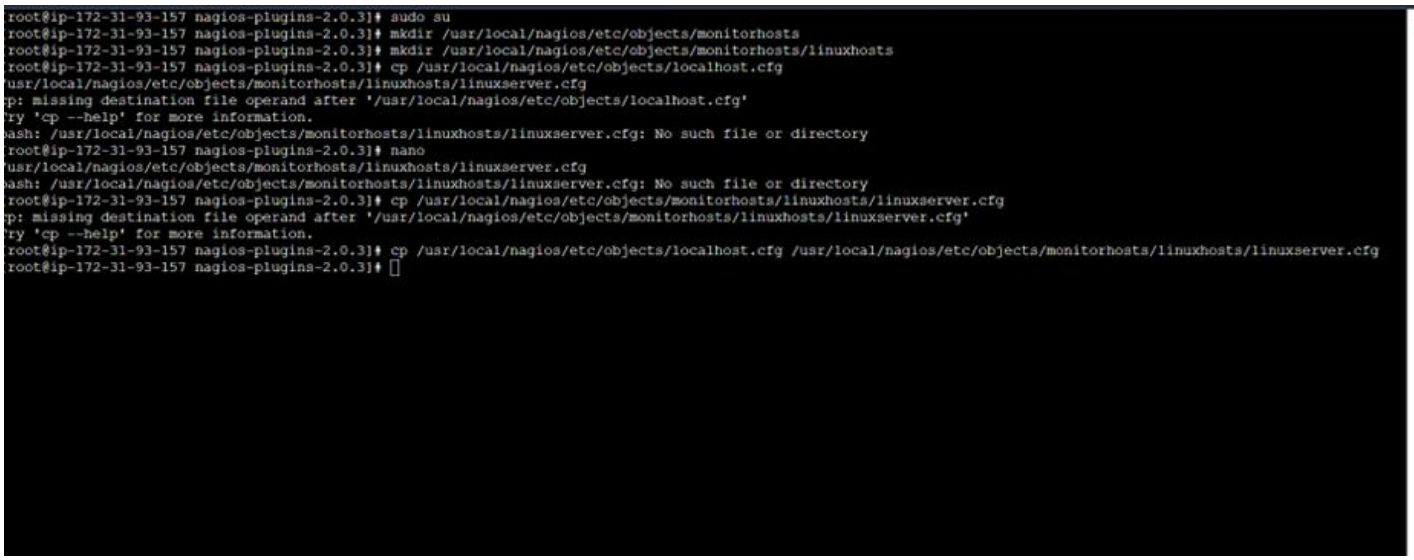
Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.
For now, leave this machine as it is, and go back to your nagios-host machine.



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

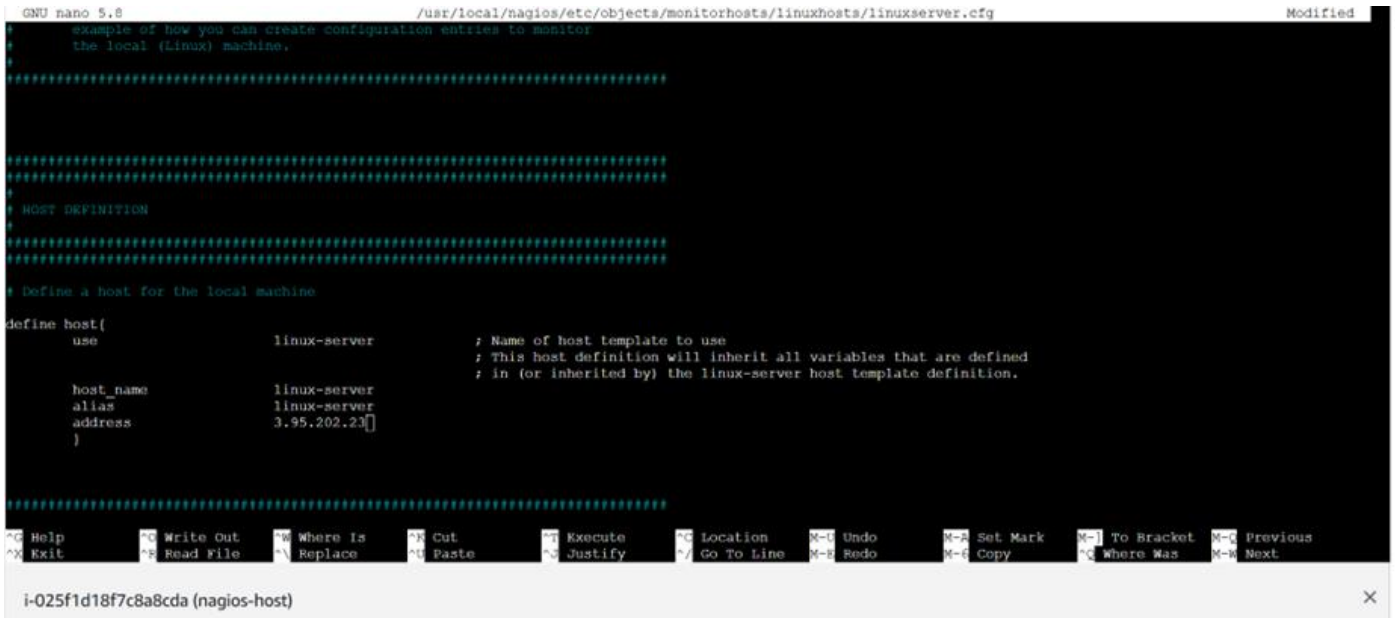


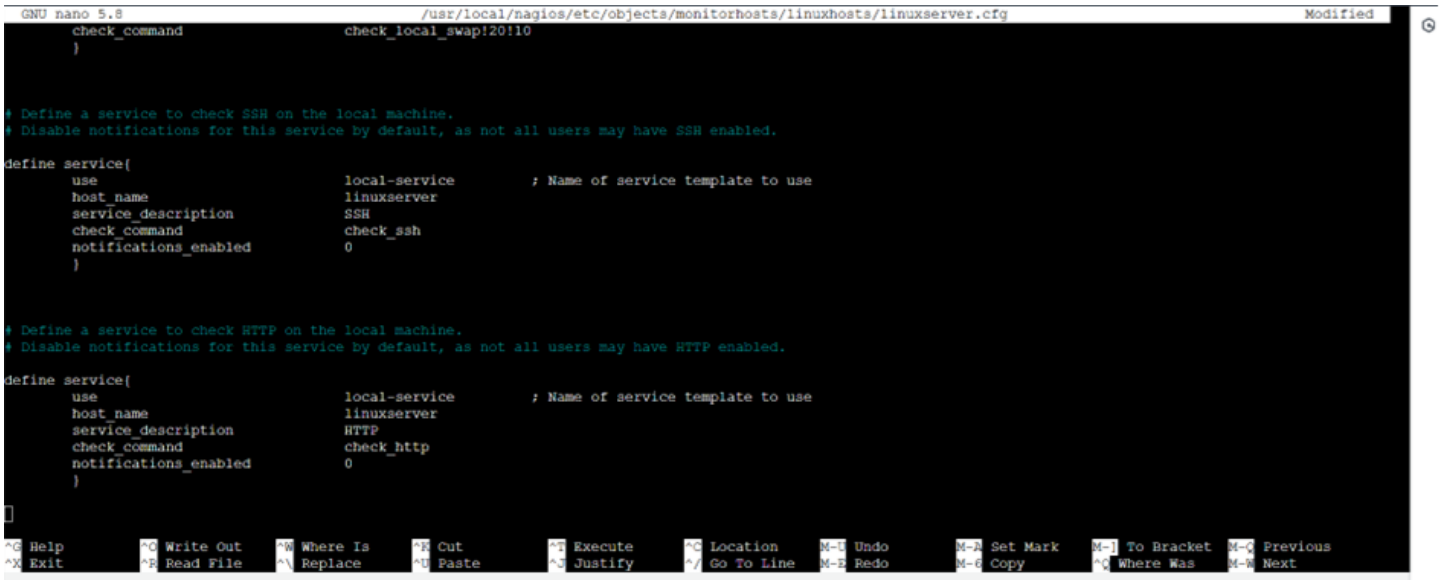
Step 4: Now, run the following commands -
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp/usr/local/nagios/etc/objects/localhost.cfg/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg



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

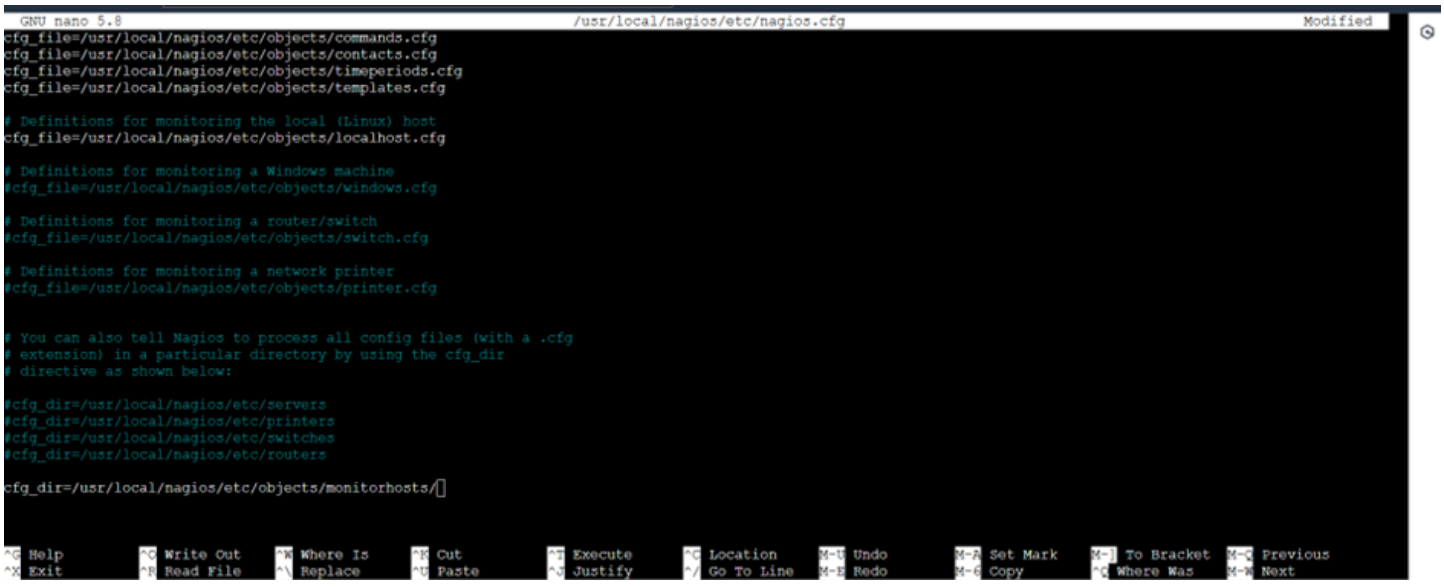
Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
Change address to the public IP address of your LINUX CLIENT.
Change hostgroup_name under hostgroup to linux-servers1





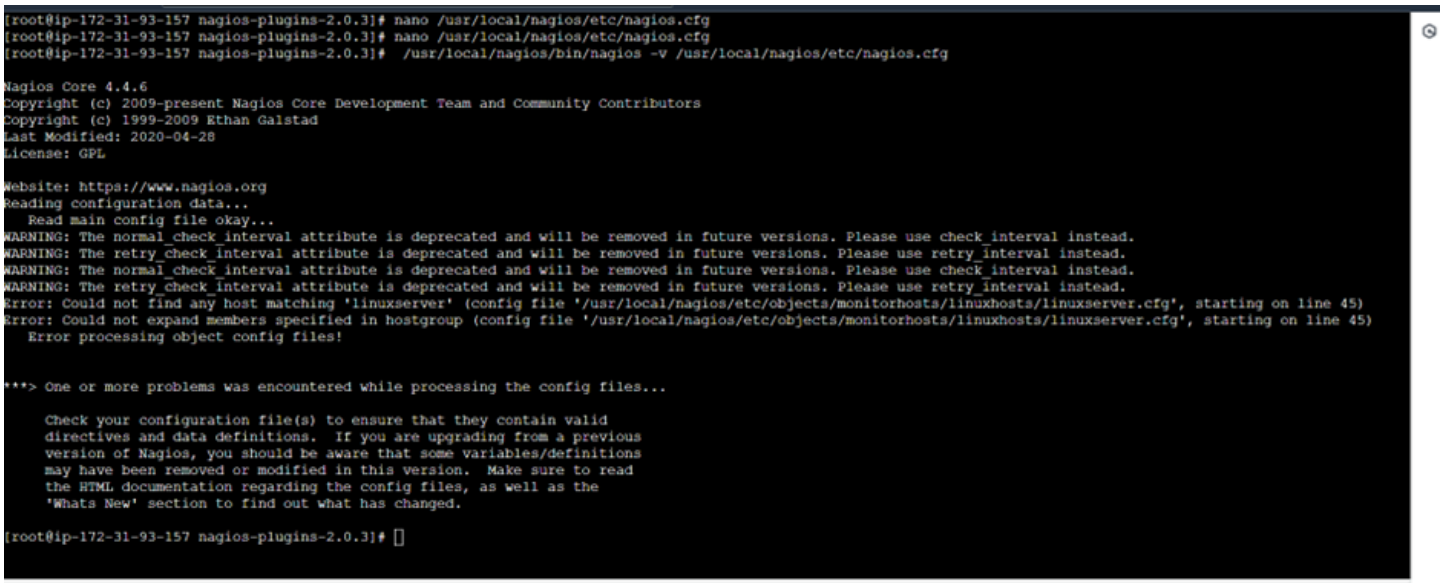
Step 6: Open Nagios config file and add the following line -
nano /usr/local/nagios/etc/nagios.cfg

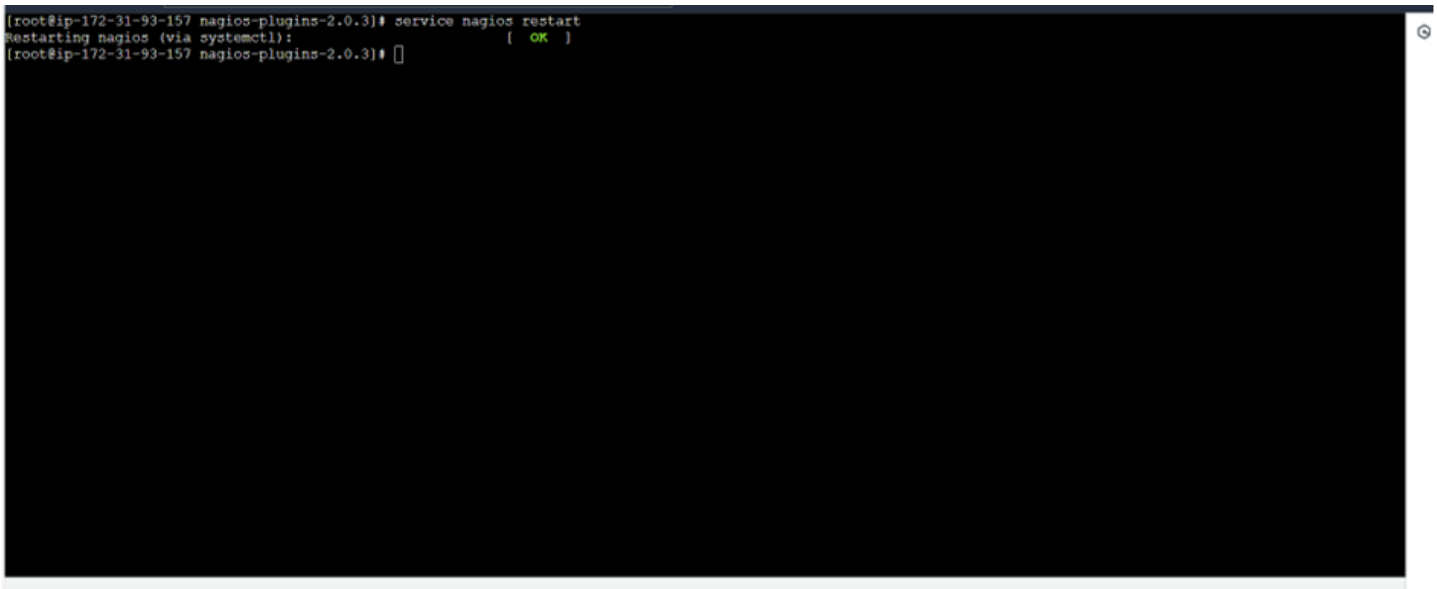
Then add this line -
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



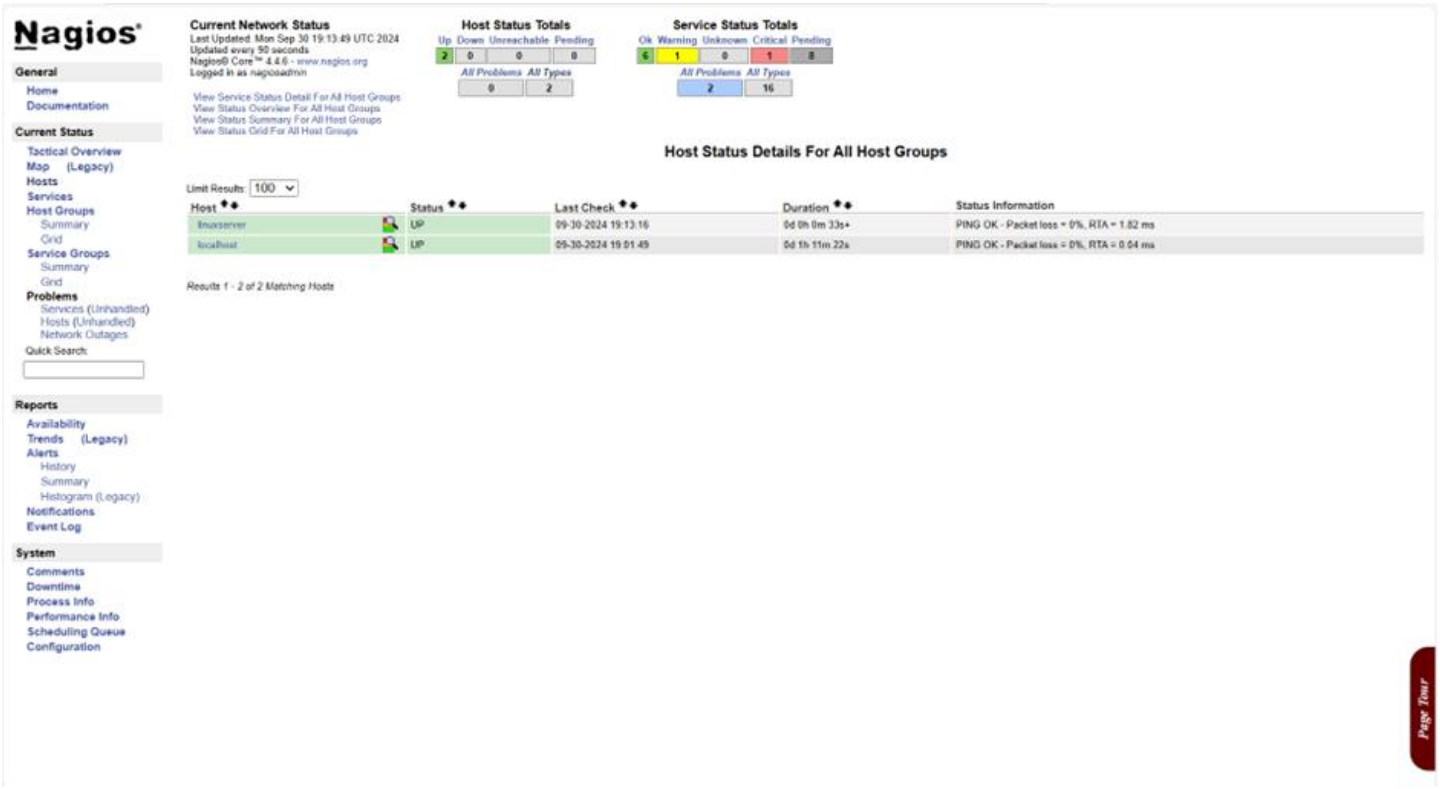
Step 8: Verify configuration files using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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.

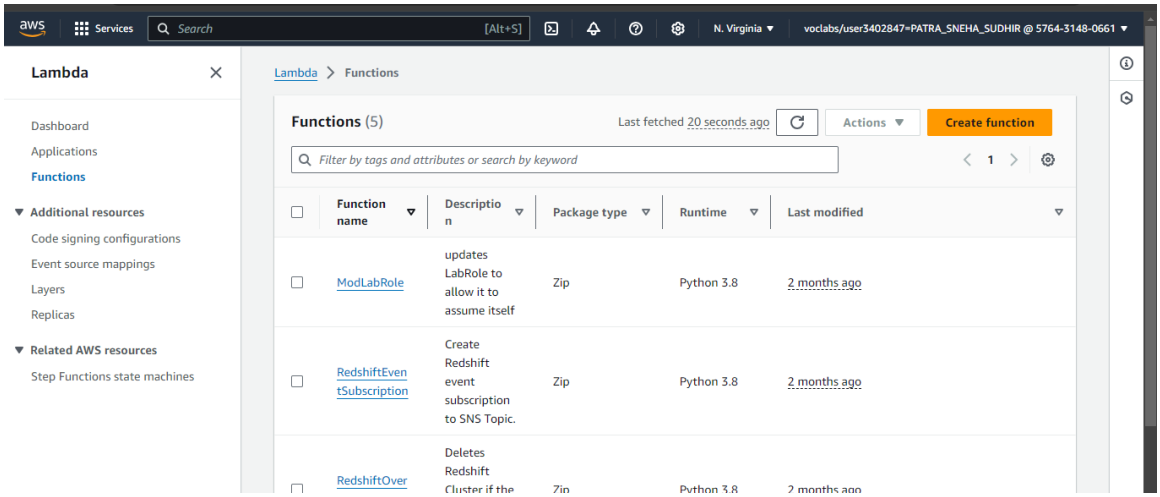


Conclusion: Port, Service monitoring, Windows/Linux server monitoring using Nagios was performed.

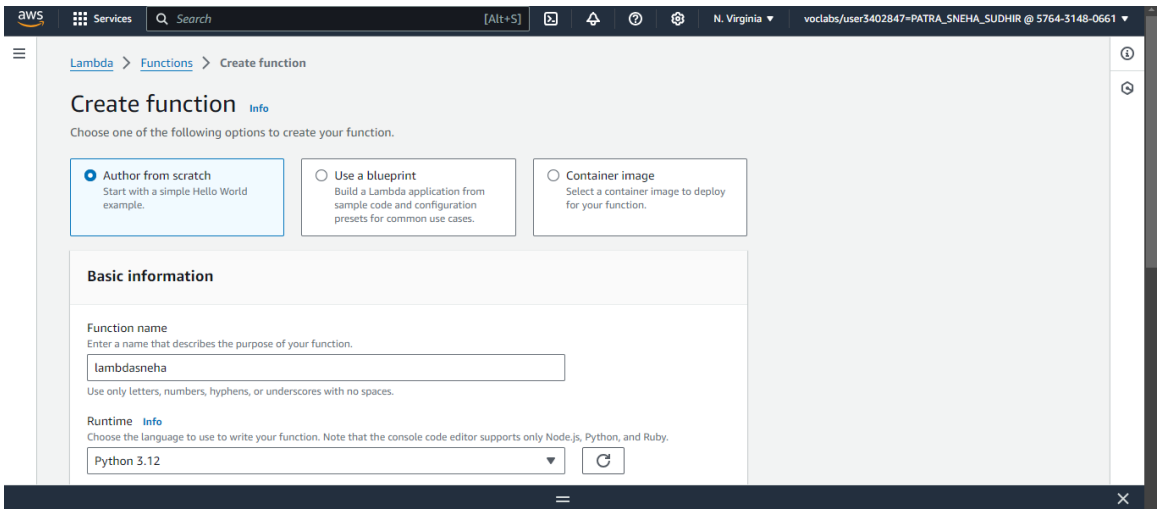
Experiment 11

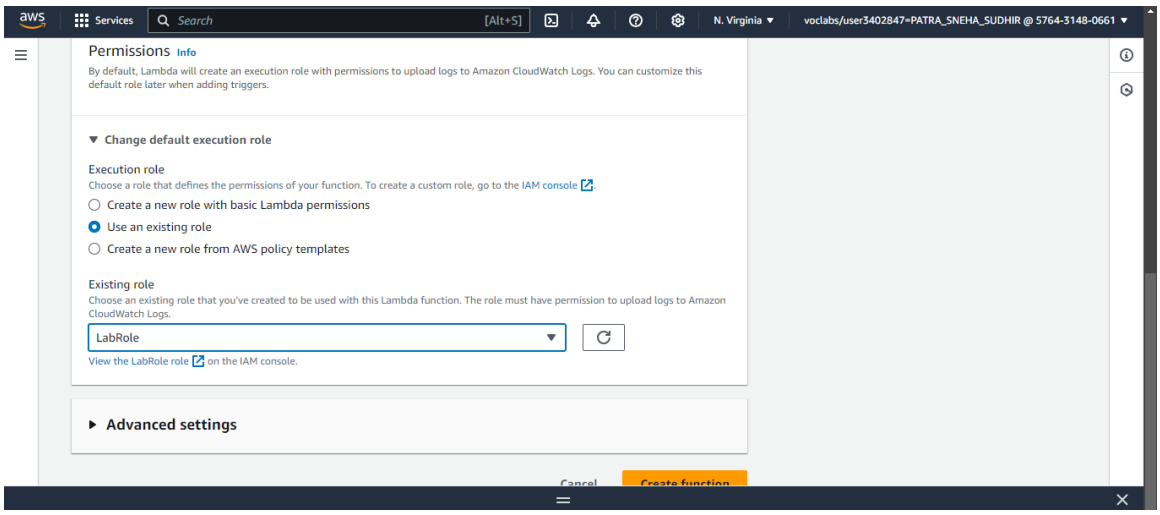
Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

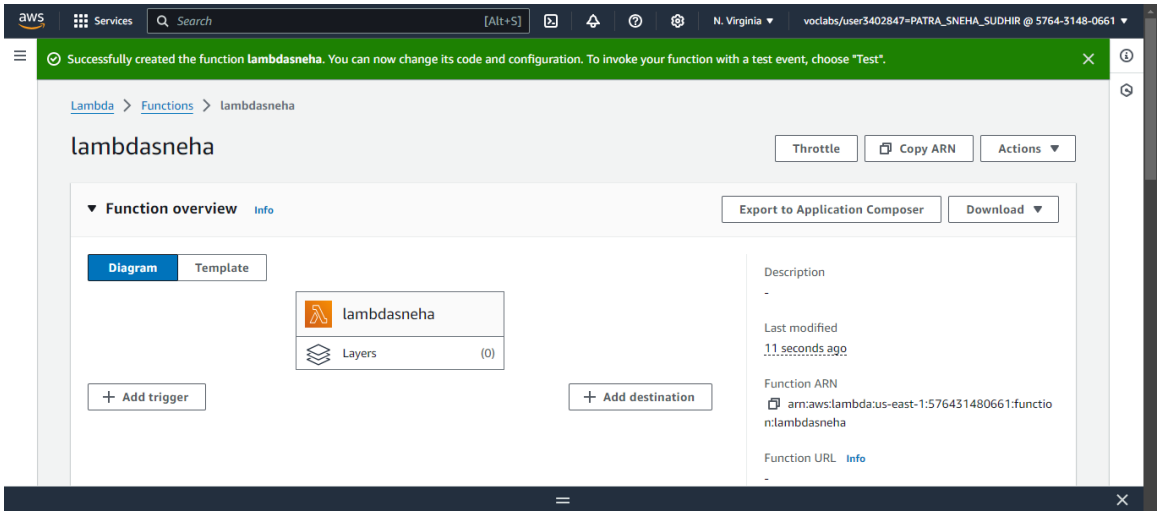
Step1: Open up the Lambda Console and click on the Create button.



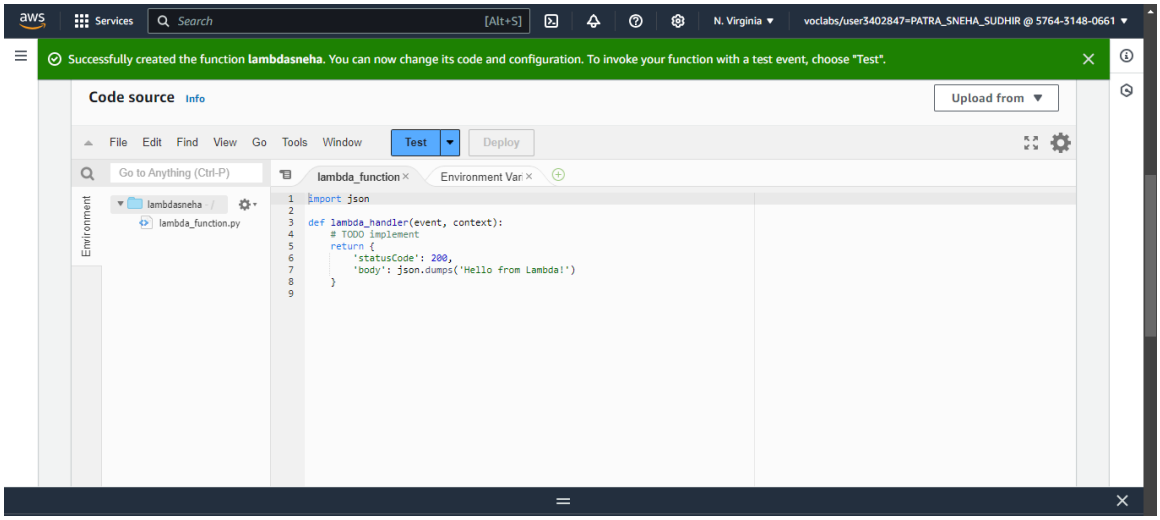
Step 2: Create a function and name it and select the runtime environment as Python 3.12 and for the role select the existing role LabRole.



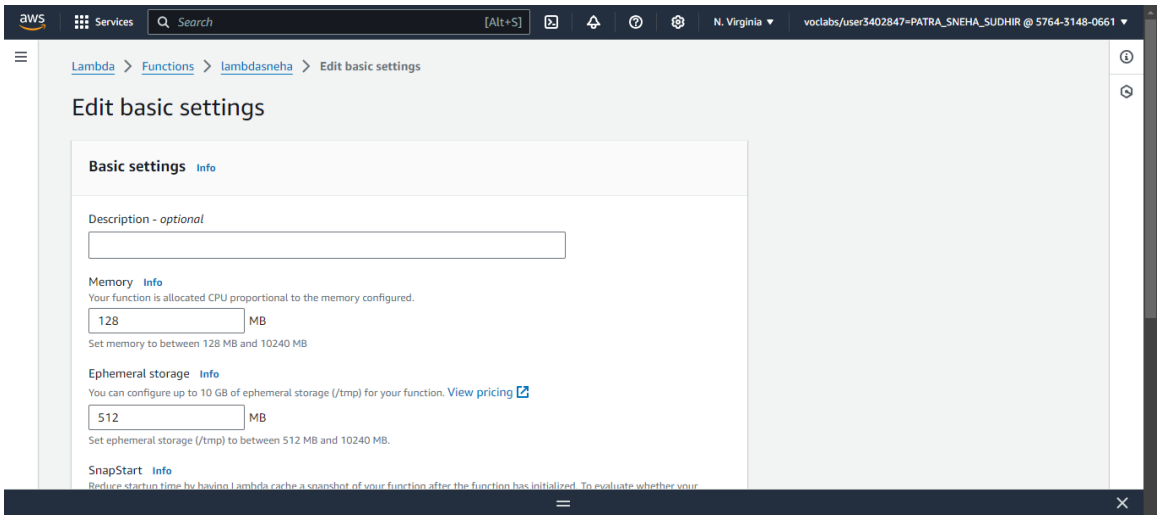




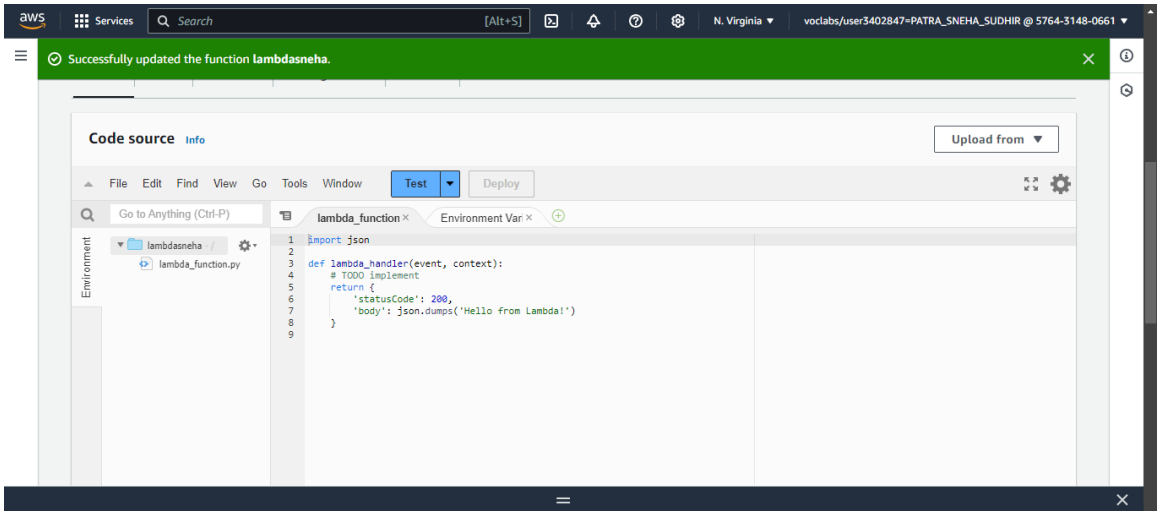
Step 3: Scroll down to the code source section and then visit Configuration section and click on edit.

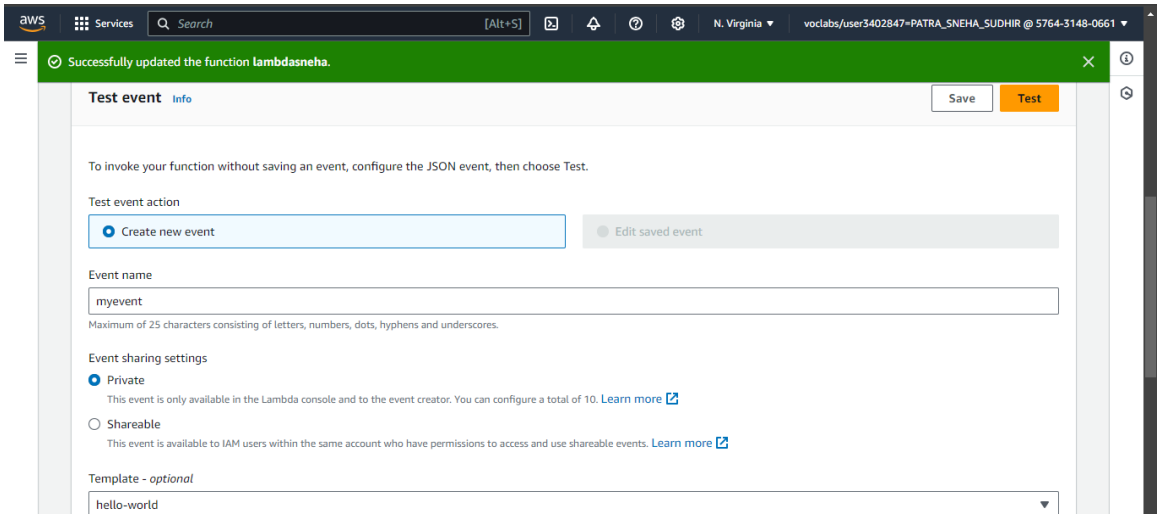


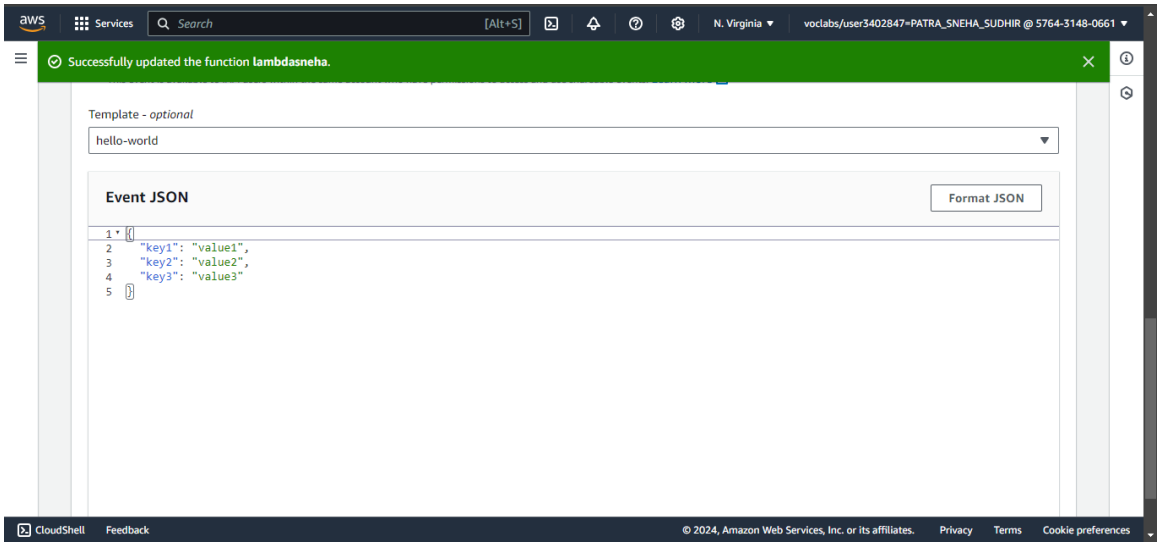
Step 4: Make the following changes and then click on the save button.



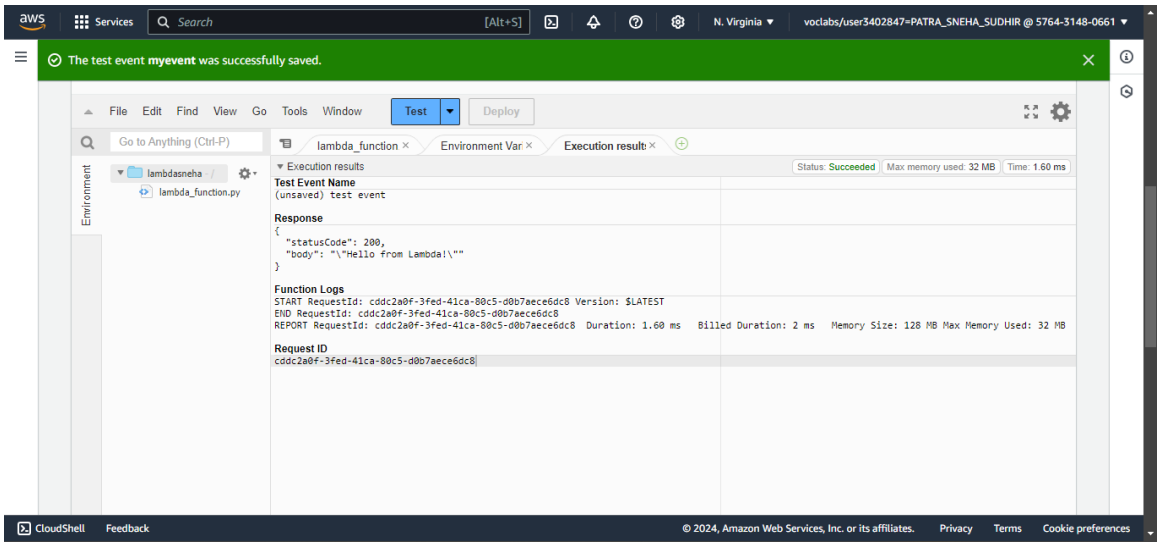
Step 5: Scroll down to the code source section and then visit Test section. Create a new event and then name the event. Now, click on Test.


Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Successfully updated the function lambdasneha.
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Environment
lambdasneha
lambda_function.py
import json
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')
}


Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Successfully updated the function lambdasneha.
Test event Info
Save
Test
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
myevent
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more
Template - optional
hello-world


Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Successfully updated the function lambdasneha.
Template - optional
hello-world
Event JSON
Format JSON
{
"key1": "value1",
"key2": "value2",
"key3": "value3"
}
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

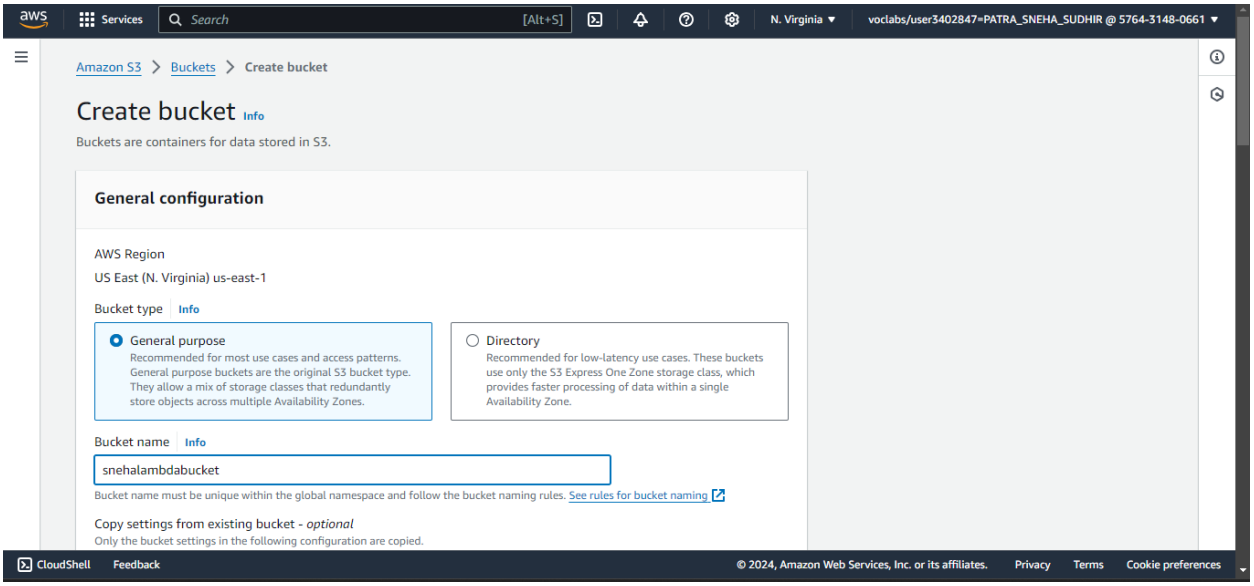
Step 6: The test results are as shown below.

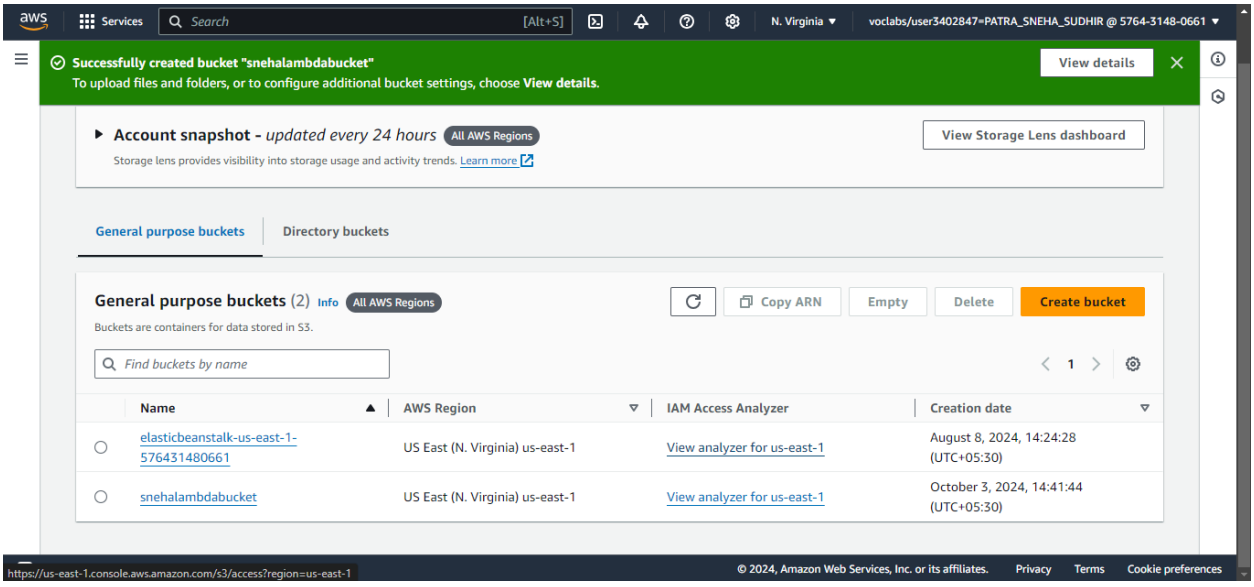


Experiment-12

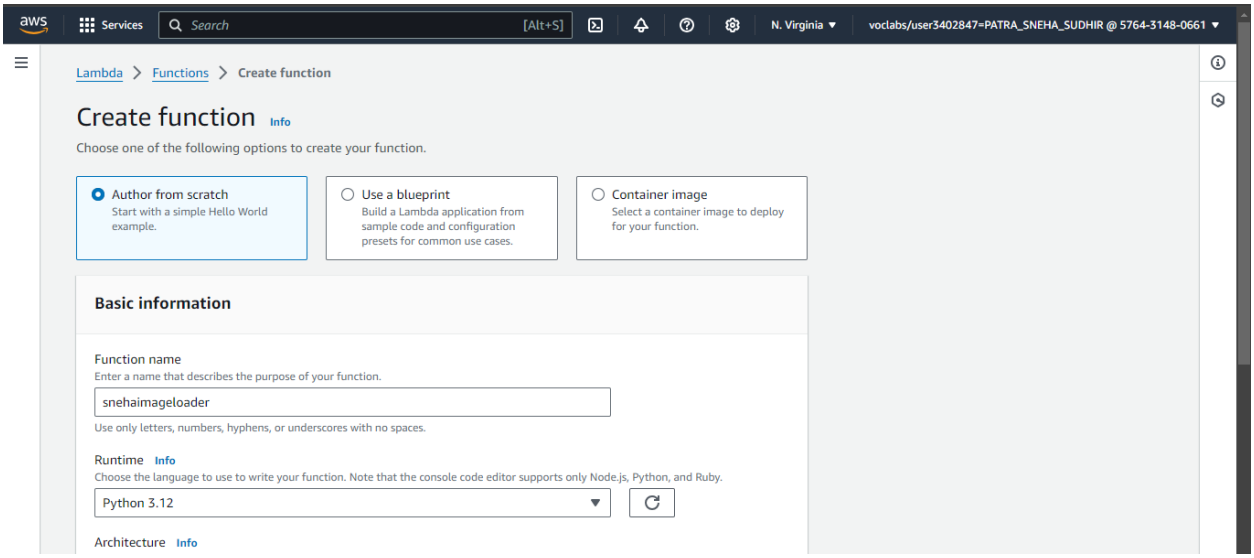
AIM: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

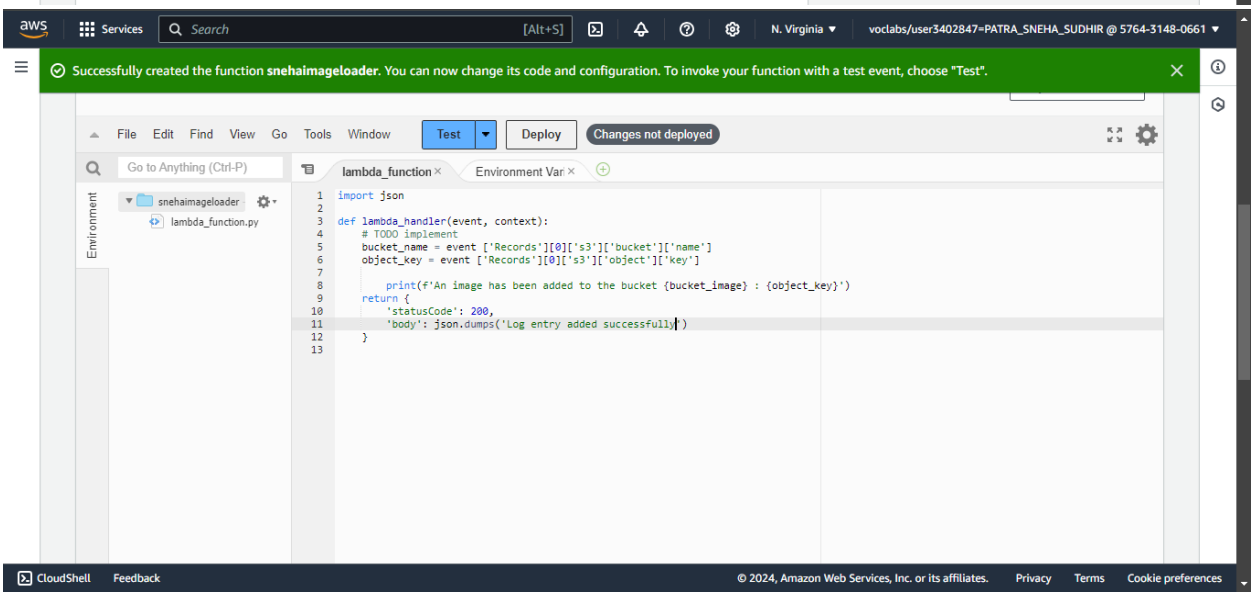
Step 1: First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.



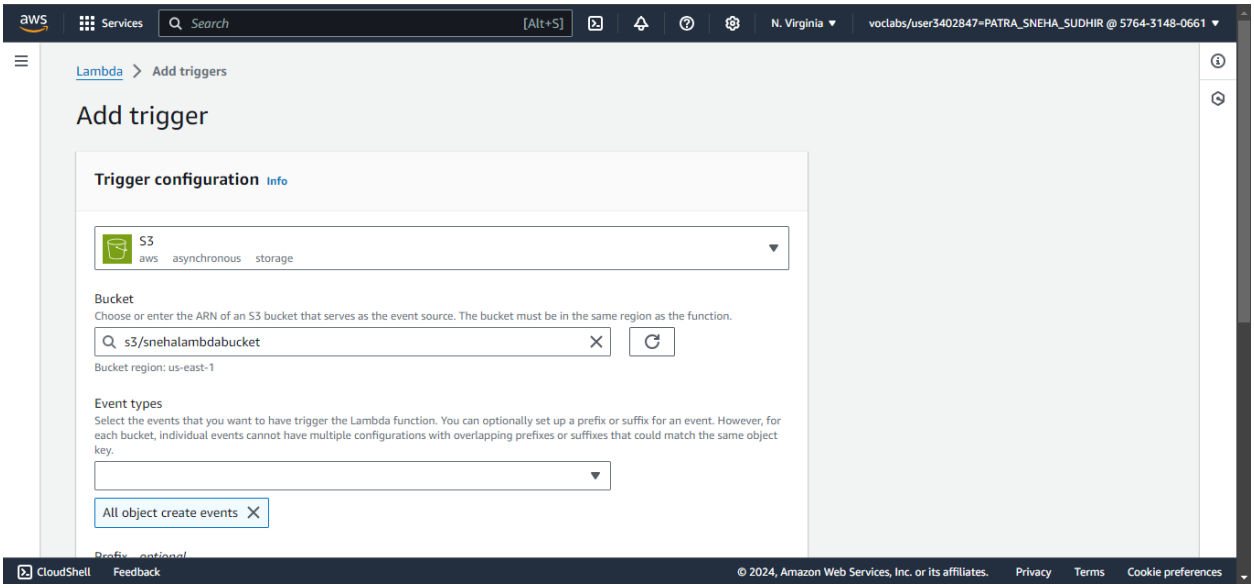


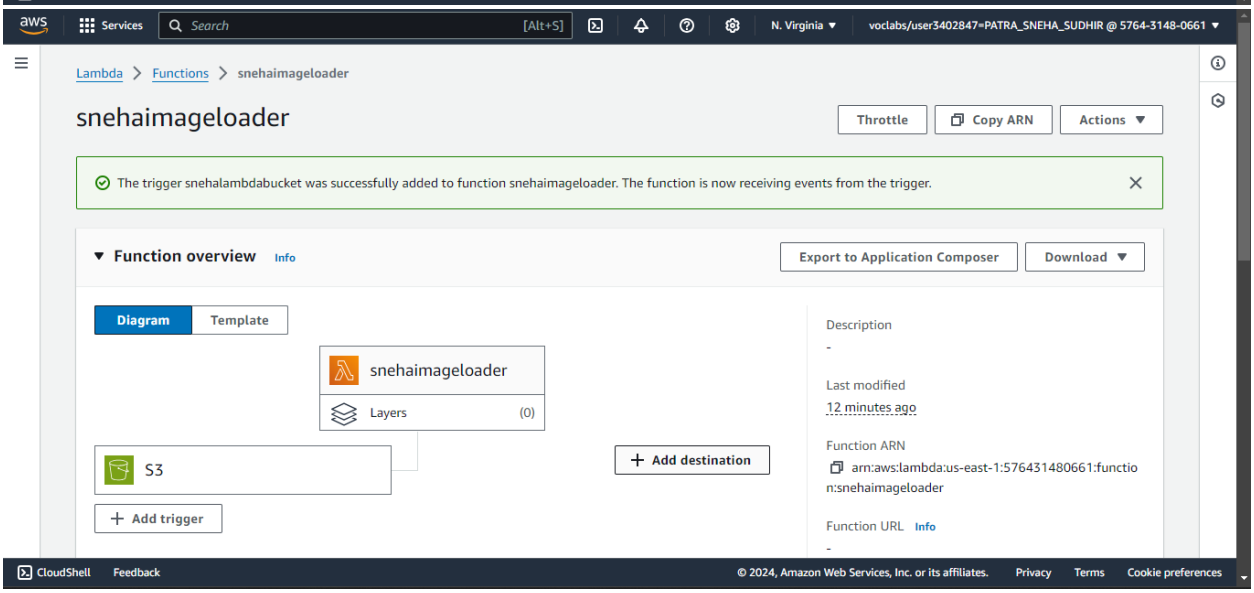
Step 2: Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java. I have selected python environment.

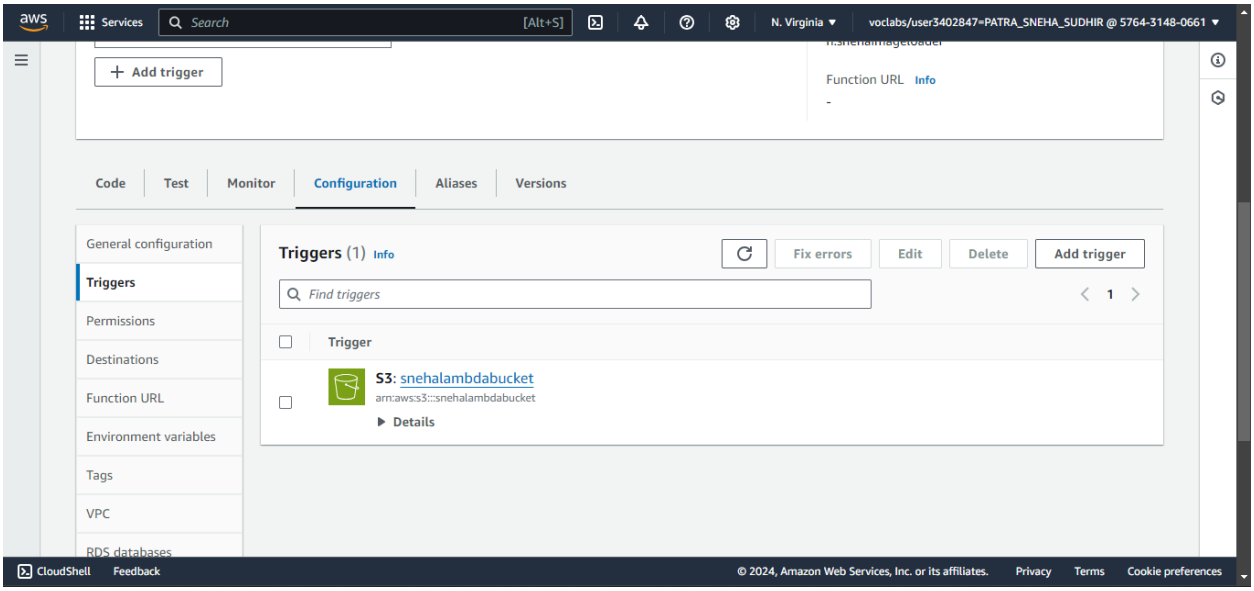




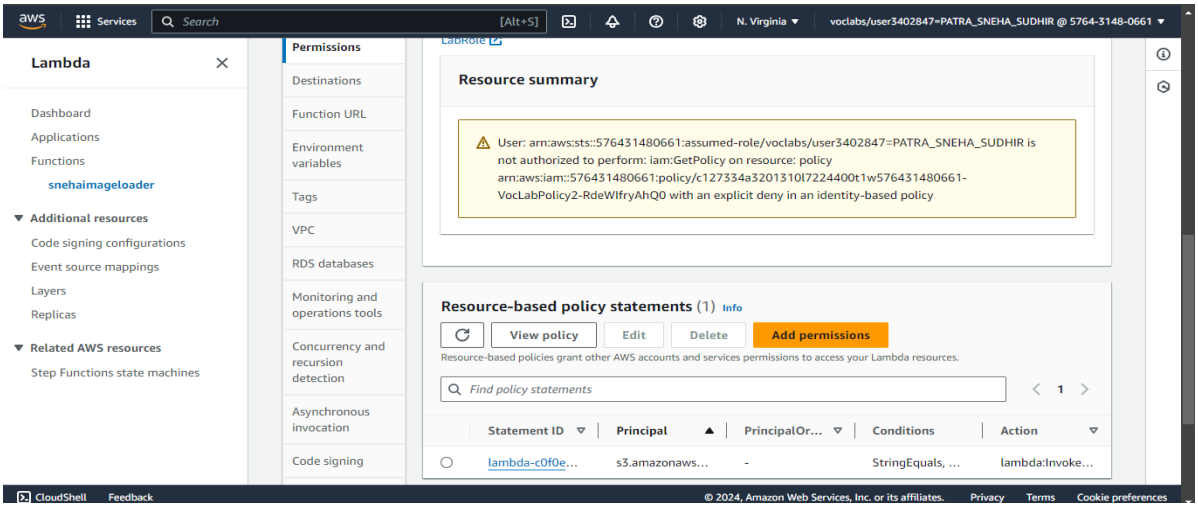
Step 3:Link the S3 bucket to the Lambda function by setting up a trigger.


Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Lambda
Add triggers
Add trigger
Trigger configuration
Info
S3
aws
asynchronous
storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/snehalambdabucket
Bucket region: us-east-1
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

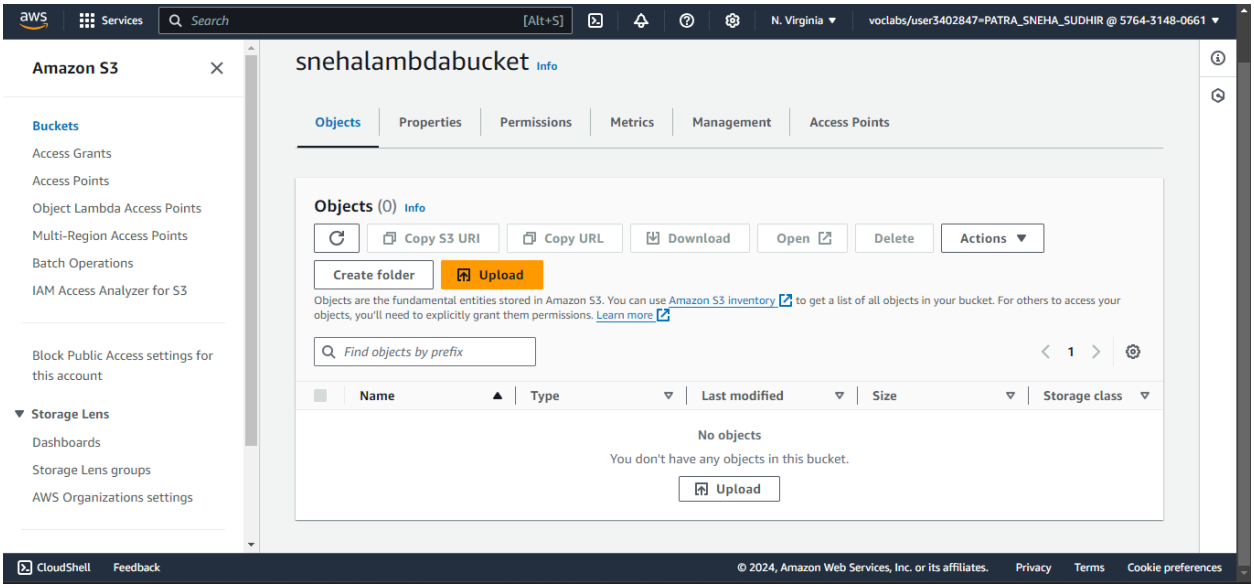

Services
Search
[Alt+S]
N. Virginia
voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661
Lambda
Functions
snehaimageloader
snehaimageloader
Throttle
Copy ARN
Actions
The trigger snehalambdabucket was successfully added to function snehaimageloader. The function is now receiving events from the trigger.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
snehaimageloader
Layers
(0)
S3
Add trigger
Add destination
Description
-
Last modified
12 minutes ago
Function ARN
arn:aws:lambda:us-east-1:576431480661:function:snehaimageloader
Function URL
Info
-
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

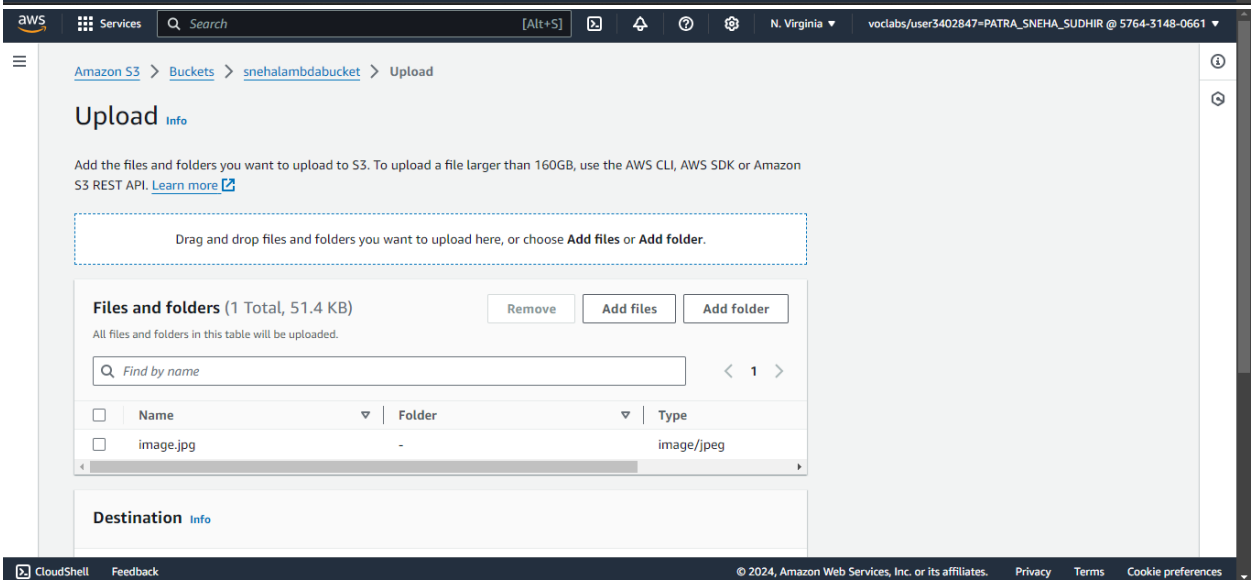


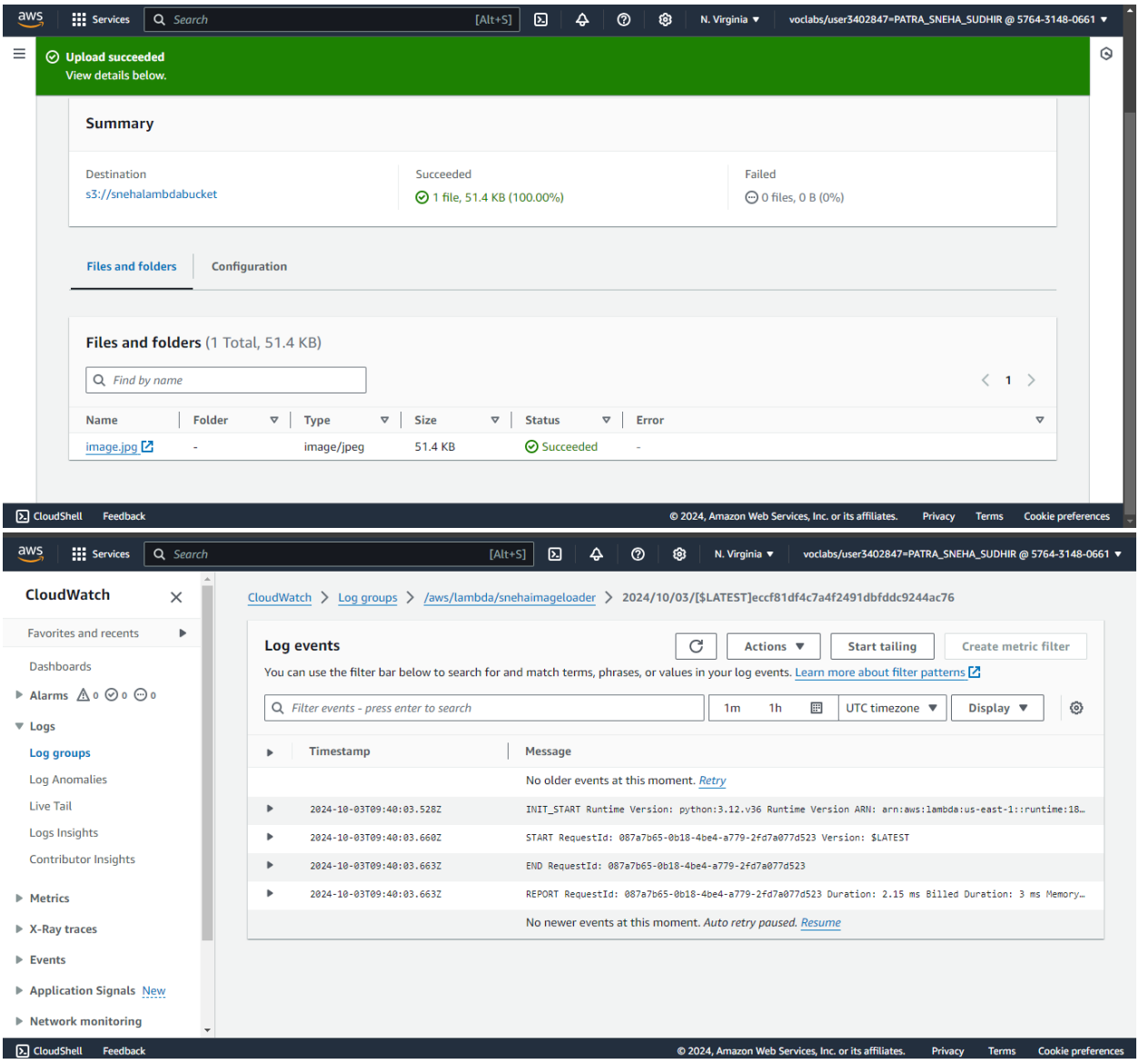
Step 4:Setup rthe required permissions.



Step 5: Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.







Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.

SNEHA PATRA
DISA '41'
DATE 05/05

Advance Devops

ASSIGNMENT NO: 1

Q1] Use S3 bucket and host video streaming

Ans: Steps to host video streaming on S3 bucket

Prerequisites:

Before you start to host video streaming through S3 bucket, you must register and configure a custom domain (for example, example.com) with Route 53 so that you can configure your cloudfront to use a custom domain name later. Without a custom domain name, your S3 video is publicly and hosted through Cloudfront at a URL that looks similar to the following https://cloudfront distribution domain name/path to an S3 video

- Step 1: Create Bucket

Sign in to Aws console and open Amazon S3 console. Then Select Bucket → Create Bucket > Enter Bucket name > Select Region → Block Public Access settings for this Bucket → Remaining setting Defaults → Then choose Create Bucket.

Step 2: Upload a video to the S3 bucket

- Now, in buckets list, choose the name of the bucket that you created in step 1 to upload your file to
- On the objects tab for your bucket, choose Upload
- Then on upload page, unde Files and folders, choose Add files
- Then select a file to upload, and then choose Open for example, you can upload a video file named
- Choose upload

Then open cloudfront and follow next steps.

Step 3: Create a cloudfront origin access identity

- Sign in to AWS and open the cloudfront console
- Select security section and choose Origin access
- Now under Identities tab, choose origin access identit
- Enter name and choose create

Step 4: Create a cloudfront distribution

a) Create a cloudfront distribution

- choose Distribution → create distribution
- choose the domain name → starts the name of S3 bucket you created
- for origin access, select legacy access identities
- Under Orgin access identity → origin access identity
- Under Bucket policy, choose Yes update the bucket policy → select Default cache behavior → Viwer protocal policy → Redirect HTTP to HTTPS
- Keep the remaining setting to Default.
- Now, choose Create distribution

b) Review the bucket policy.

Step 5: Access the video through the cloudfront distribution

- Go to Distribution in the left panel
- find the distribution by matching the S3 origin name and then copy the Domain Name and now Open a new tab and paste the copied Domai name
- Now, Return the previous tab, open the S3 conso and select the bucket created in step1 and then choose the video object uploaded in step2 and the copy the key from The Object overview.

- In the new tab, append / and paste the key to the domain Name
- Your video is now accessible via Cloudfront

Step 6: Configure your cloudfront distribution to use your custom domain name

a) Request an SSL certificate
b) Add the alternate DNS to your Cloudfront.
c) Create a DNS record to route traffic from your alternate domain name to your cloudfront distribution's Domain name.
d) Check whether IPV6 is enabled for your distribution and create another DNS record if needed

Step 7: Access the S3 video through the Cloudfront distribution with the custom domain name

- Select Distribution → Find distribution by matching S3 origin name → Copy the alternate domain name → Past the domain name.
- Open S3 → find path to your S3video → Return to the tab with domain, add /<s3video path>
- Access your video at https://Cloudfront_domain/S3_video_path.

Step 8: View data about requests received by your Cloudfront distribution (Optional)

Now, your video streaming is live through S3 bucket.

Q2] Discuss BMW and Hot Star case studies using AWS

Ans:
- The BMW Group whose headquater is in Munich, Germany, is a Global manufacturer of premium automobiles and motorcycles.
- The company needed to scale its data lake to support the growing demands of internal and external stakeholders. As data wasn't easily accesible..:
- The BMW Group's innovation was slowed down by their own IT infrastructure and the long lead times required to support new initiative. The BMW needed to develop a solution agile enough to both support the data needs of all the various internal buisness units and allow this company to move quicky to address the array of emerging use cases its customers demand.
- The BMW Group re-architected its on-premises data lake to the AWS Cloud, creating a Cloud Data Hub (CDH) that integrates anonymized data from vechile sensors and other sources.
- Using AWS services like Amazon S3, Athena, Kinesis Data Firehose, and Glue, BMW streamlined data management and enabled scalable, agile operations for data engineers. The setup also allowed teams to maintain their own DevOps processes, fostering innovation. A modern web portal was implemented to help users discover and query trusted datasets, facilitating new insights.

- The company uses this data to monitor ~~vechle~~ vehicle health indicators such as check control error to identify potential issues across vechicle lines
- This enable the BMW Groups to leverage fleet data ingested, collected and refined from the CDH to better resolve issues, even before they impact customers.

HOTSTAR

- Hotstar is an indian subscription video on-demand streaming service owned by and operated by Star India, a subsidiary of The Walt Disney Company India.
- In 2019, During ICC world cup Semi-final between India and New-Zealand, Hotstar sets a new record of 25.3 million viewers. So, on the game day, the first spike witnessed was from 1.5M to 15m, as India started batting. Then, in between it was usual (10-12m) then Dhoni came to bat, and again sudden spike was noticed in traffic, taking it to 25.3m viewers. But then Dhoni got out, and suddenly there was drastic viewers drop to 4.1M viewers.
- The very first challenge was handling 25.3m viewers. Secondly, when users dropped off the match some of them exited from the app entirely and others returned to the homepage and started exploring other content. That leads to an increase in load on homepage services.
- Hotstar does not use traditional autoscaling from

Aws because there were a lot of challenges like
- Insufficient capacity errors.
- Step size, Autoscaling groups.
- They built their own scaling strategy. and At the Back end side, Hotstar uses Amazon Route 53 and Amazon CloudFront are services for the Hotstar streaming video

Q3] Why kubernetes and advantages and disadvantages of kubernetes. Explain how adidas uses kubernets?

Ans. Imagine you have a bunch of different programs running on your computer. They need to work together and sometimes you want to run more copies of a program when things are busy. This can be hard to do manually, thats where kubernets comes in

Advantages of kubernets

- Scalability - kubernets allows developers to easily scale their application up or down as demand fluctuates
- Resource efficiency
kubernets helps optimize the use of resources by scheduling containers to run on the most appropriate node based on their resource requirement.
- High availability
kubernetes provides mechanisms for ensuring that application are alway available.
- Other advantages are Portability, Self healing,

service discovery and load balacing, Extensi

Disadvantages of kubernetes.

- Complexity:
  kubernets can be complex to set up and manage.
- Learning curve
  Developers and operations teams need to learn how to use kubernets effectively
- Performance overhead
  kubernets introduces some overhead in terms of CPU and memory usage, which can impact application performance.
- Some other disadvantages are Security, Dependency on external services, lack of maturity Complexity of networking.

Case Study: Adidas.

- Adidas is a globally renowned sportswear and athletic footwear company headquartered in Germany
- In recent years, the adidas team was happy with the software choices from a technology perspective. But accessing all of the tools was a problem. for example, just to get a developer VM, you had to send a request, give the purpose, who's responsible, internal cost center and so on. So the best case is you got your machine in half an hour and the worst case in half week or sometimes even a week was a challenge faced by the adidas.
- They found the solution with containerization,

agile development, continuous delivery, and a cloud native platform that includes kubernetes and Prometheus.

- Just six months after the project began, 100% of the adidas e-commerce site was running on kubernetes. load time for the e-commerce site was reduced by Half Release went from every 4-6 weeks to 3-4 times a day. with 4000 pods, 200 nodes, and 80,000 builds per month, adidas is now running 40% of its most critical, impactful systems on its cloud native platform.

Q4] what are Nagios and explain how Nagios are used in E-services.

Ans. • Nagios is a powerful monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes.

- E-services S.R.L is an innovative energy company offering solutions in monitoring, VOIP, call centers and IT solutions. E-services S.R.L. chose to partner with Nagios and become the officid representative of Nagios XI in Paraguay.
- To set up a monitoring system across Paraguay one would have to carefully consider monitoring band width, setting up different levels of support for hosts that are only being watched, a high availability system with a failover system,

an a system that can be accessed from afar.

- Nagios provided E-services and ANDE with real-time solution including:
  - A centralized monitoring system for their entire infrastructure, easing the sysAdmin workload
  - Helpful and intutive statics that simplify decision making and aid troubleshooting
  - Easy-to-understand graphics and displays
  - Excellent availability, thanks to the mirrored servers with the failover system.

Sneha Patra
D15A '41'

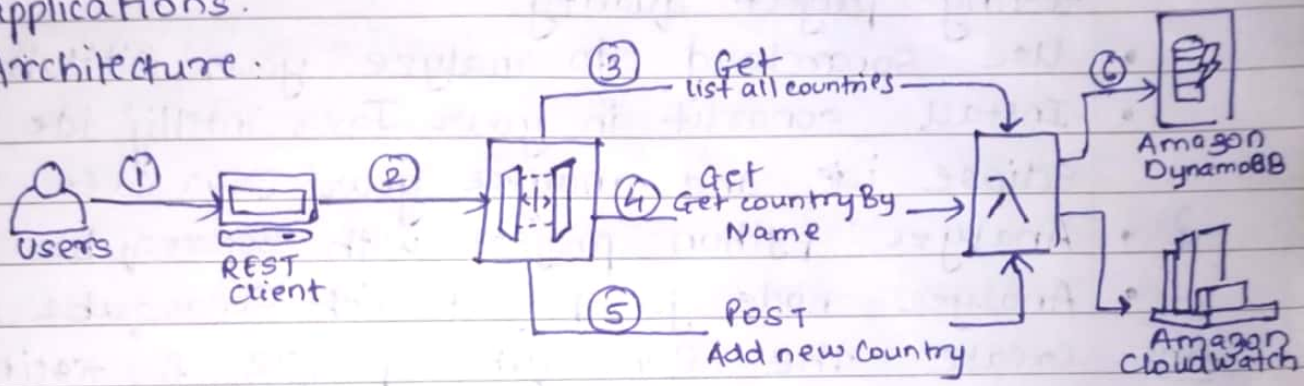
Advance DevOps

Assignment No: 2

Q1] Create a REST API with the Serverless framework.

Ans. Aws is a popular cloud provider that offers a myriad of services to support serverless architectures One of the most common use cases involves the combination of Lambda, API Gateway, and DynamoDB to build scalable, efficient, and cost-effective applications.

Architecture:



Setting Up the Environment.

- To install the Serverless framework, you need to have Node.js installed on your machine. Once you have installed the serverless framework you can create a new serverless project using a template Here we will use aws-node.js template
- The handler.js file is where we will write our Lambda function code.
- The serverless.yml file is where we will define our infrastructure. In this we will define DynamoDB table to store Data
- Lambda function will be triggered by different endpoints of our API.
- After defining the serverless.yml file and

all of the functions, we can deploy our serverless application using following commands SLS deploy
- After the deployment is complete, the serverless framework will output the URLs of API endpoint.

Q2] Case Study for Sonarqube.
- Create your own profile in sonarqube for testing project quality
- Use Sonarcloud to analyze your GitHub code
- Install sonarlit in your Java intellij ide or eclipse ide and analyze your Java code
- Analyze python project with Sonarqube
- Analyze node js project with Sonarqube.

Ans: Create the Sonarqube profile for testing project quality
- Then open Intelij setting, find Tools > Sonarlint entry and select + to open the connection wizard
- Enter a name for this connection, select SonarCloud or SonarQube.
- Choose the authentication method.

(a) Generate token on sonarqube or SonarCloud
(b) Now add username + Password: This can be used on Sonarqube connection only
- for SonarCloud only select organisation that you want to connect to.
- SonarQube and SonarCloud can push notification to developers.
- Validate the connection creating by selecting

Finishing at the end of the wizard and save the connection in global setting by clicking OK.

→ BIND Python Project to Sonarqube

- Select SonarLint > Bind project to SonarQube
- → choose the correct project from SonarQube
- → Analyze the project (Python Project)
- Trigger an analysis by going to Code > Analyze code > SonarLint.

→ Analyze Noje.js project.

- Make sure your Node.js project is properly configured with sonar-project.properties file or equivalent for the analysis to run.

Q3] At a large organization, your centralized operations team may get many repetitive infrastructure requests. You can use Terraform to build a "self-serve" infrastructure model that lets product teams manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying and manging services in your organization, allowing teams to efficiently deploy services in compliance with your organization's practices. Terraform Cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

Sol^n At a large organization, implementing a self-serve infrastructure model using Terraform can significantly streamline the process of maniging infrastructure across different teams. This approach allows product teams to manage their own infrastructure independently while adhering to organizational standards and best practices.

Key aspects of this self-serve model include:

1.] Standardization through Terraform Modules.
By creating and utilizing Terraform modules, organizations can codify their infrastructure deployment and management standards. These modules serve as reusable packages of Terraform configurations that encapsulate common patterns and best practices.

2.] Efficient Deployment:
Product team can leverage these standardized modules to quickly deploy services without needing to reinvent the wheel or wait for the centralized operations team to handle every request.

3.] Compliance
By using predefined modules, teams ensure that their deployments comply with the organization's established practices and security guidelines.

4.] Automation.
The use of Terraform modules promotes

automation, reducing manual intervention and potential human errors in infrastructure management.

5.] Version Control.

With modules stored in version control systems like Git, teams can track changes, collaborate on improvements, and maintain a history of infrastructure configuration.

Integration with Ticketing Systems

Terraform Cloud offers integration capabilities that further enhance the self-serve model:

1.] Automatic Infrastructure Requests.

Terraform Cloud can integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests. This automation streamlines the process of submitting and tracking infrastructure changes.

2.] Centralized Management.

By centralizing infrastructure management through Terraform Cloud, organizations can maintain better control over who can request and approve infrastructure changes.

3.] Governance.

The integration with ticketing systems allows for better governance of infrastructure requests, ensuring that all changes go through proper approval processes before deployment.

Collaborative infrastructure Management.

As organizations grow and adopt Terraform at scale, they often move towards a collabrative infrastructure management approach:

1.] Team-Based Permissions and Team-Based Implementing
2.] State and Run History.
3.] Sensitive information protection.
4.] Module Registry.

By implementing these features and practices large organizations can effectively leverage Terraform to build a robust, scalable, and compliant infrastructure management system that supports both centralized control and decentralized team autonomy.