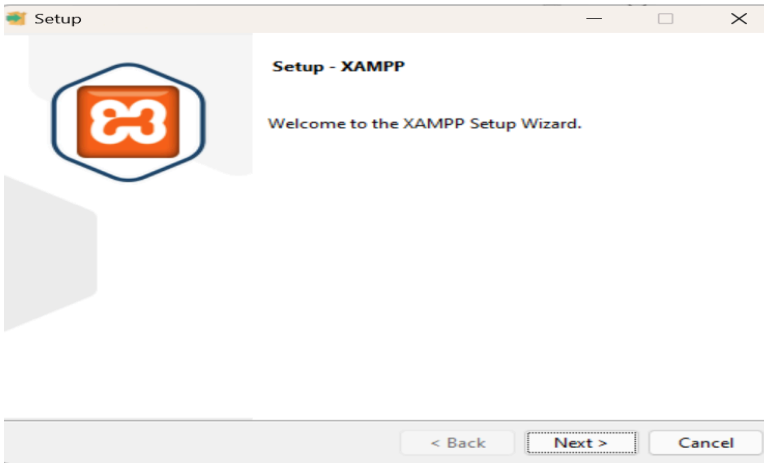
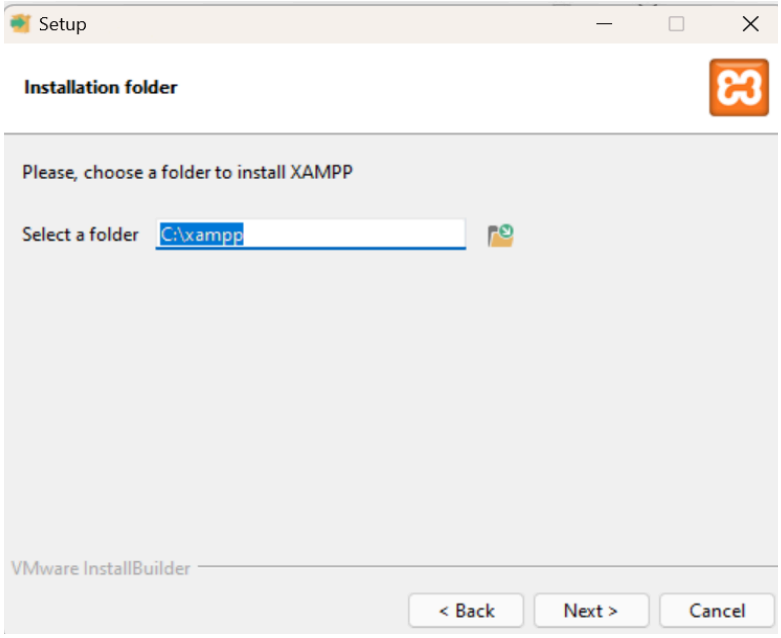
Experiment 1A

Hosting a static site using xampp

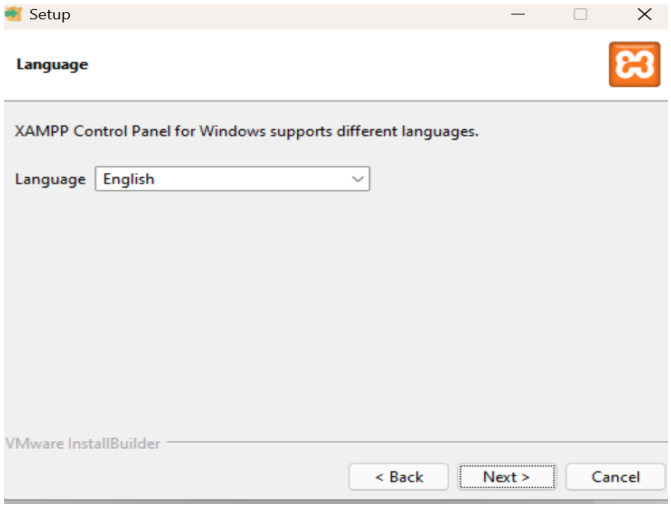
1. Download and open the xampp application and click on zip and extract it then click it.



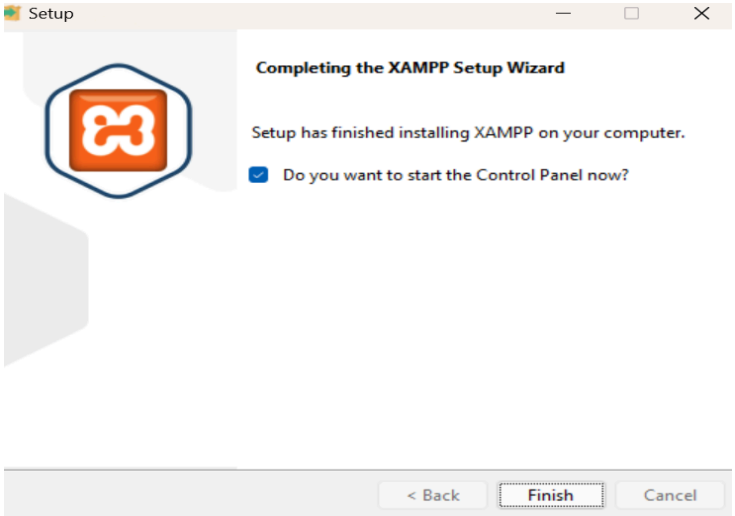
2. Then select the c directory.



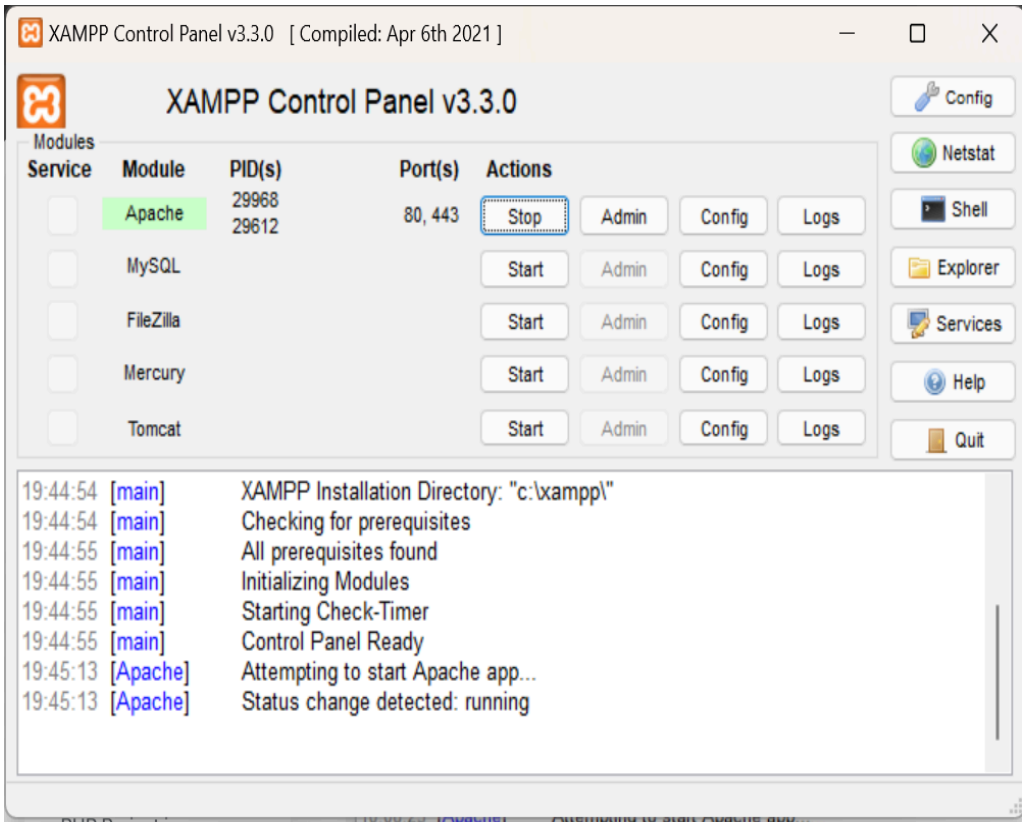
3. Then select language as english.



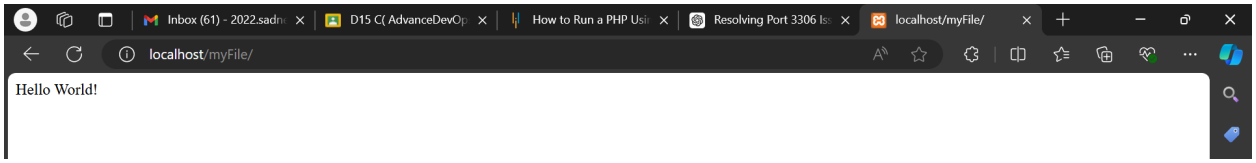
Then install and get started.



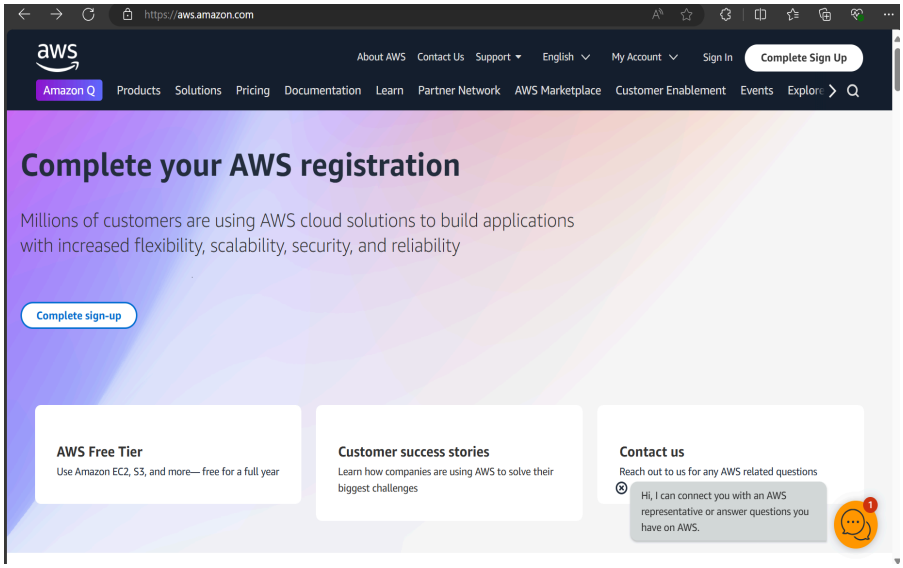
Then the final control panel opens.



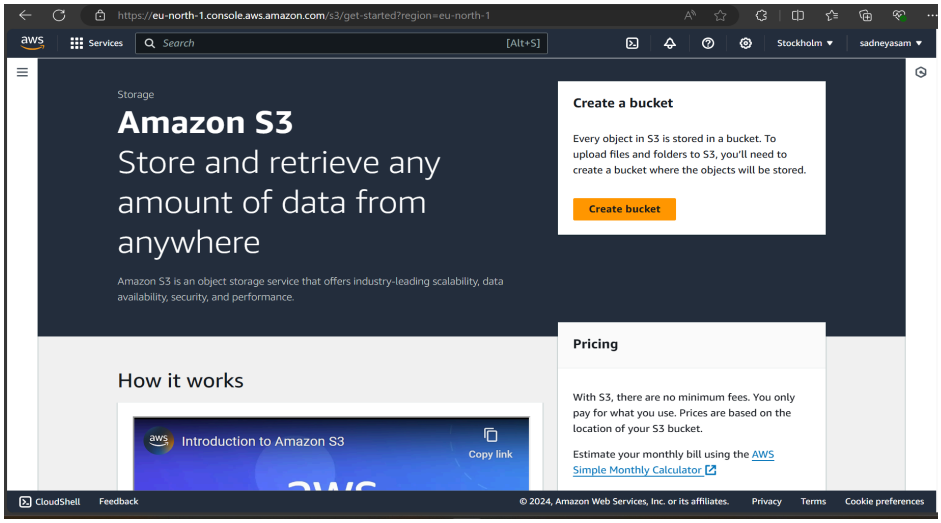
Then I created a file myFile inside htdocs folder and uploaded a sample file.and then opened https:/localhost/myFile/in.php

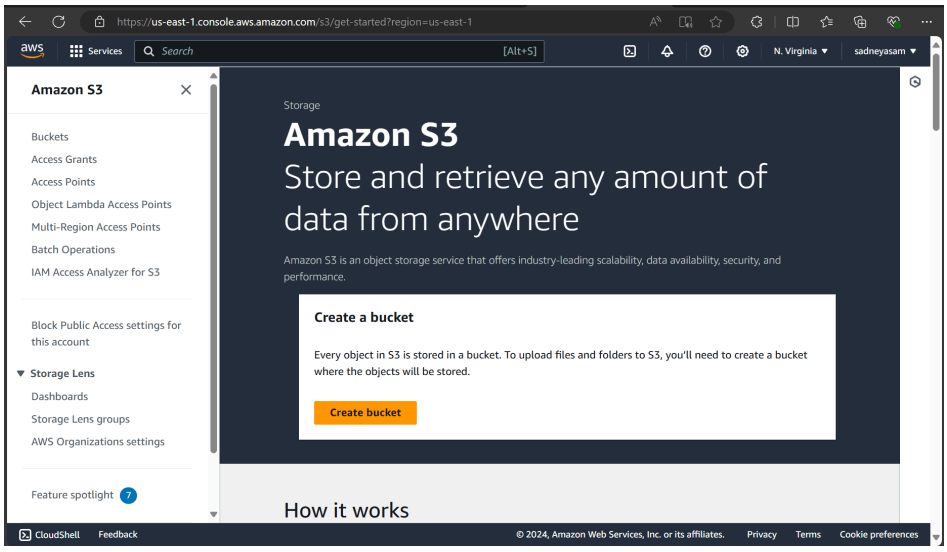


Hosting a static site using S3 bucket:



1. I clicked on create bucket.

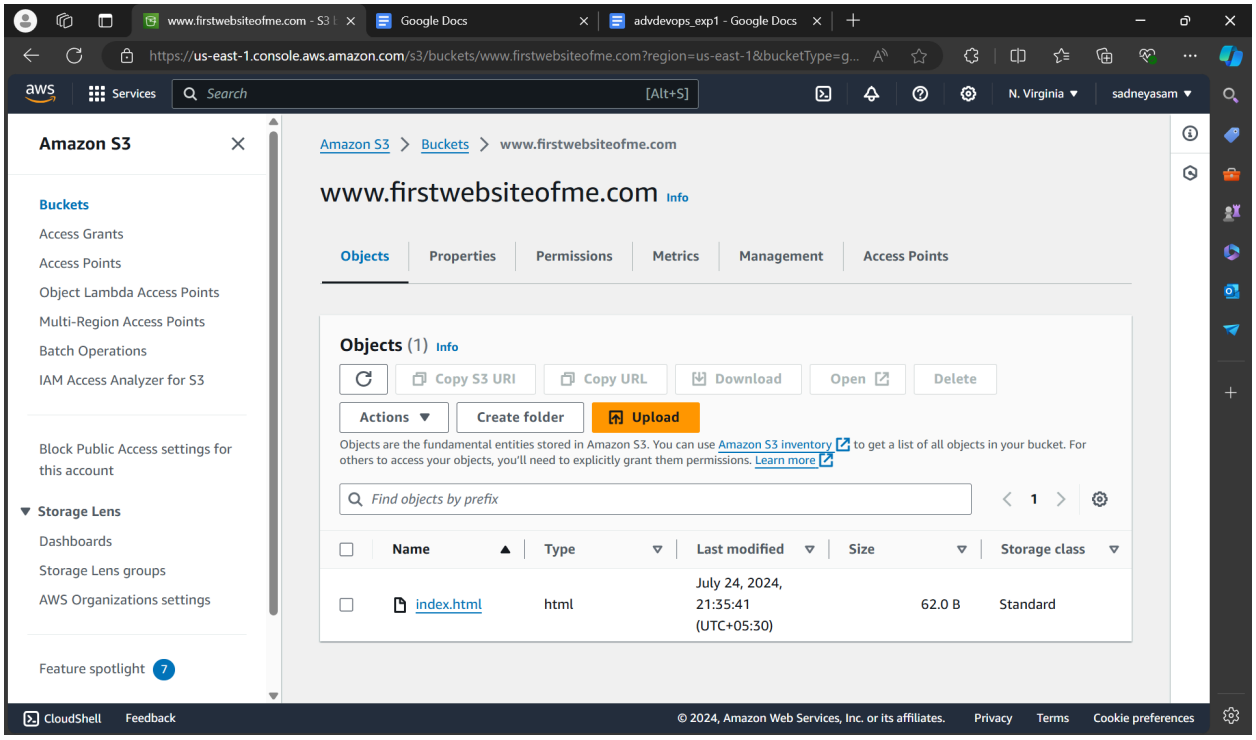




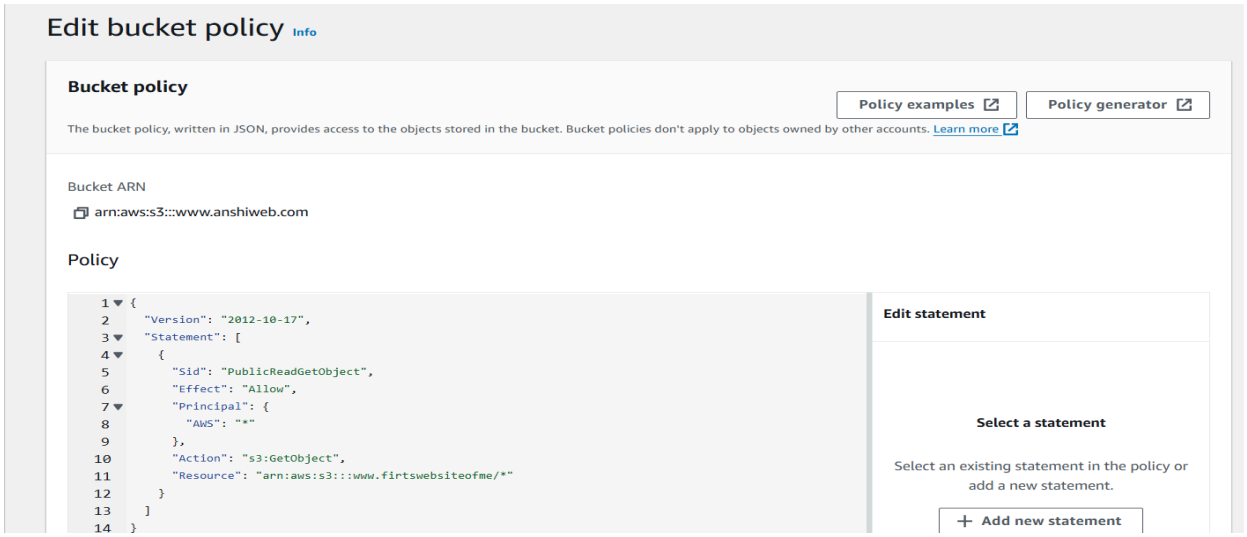
3. Then I created a bucket named www.firstwebsiteofme.com

○	www.firstwebsiteofme.com	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 24, 2024, 21:35:15 (UTC+05:30)

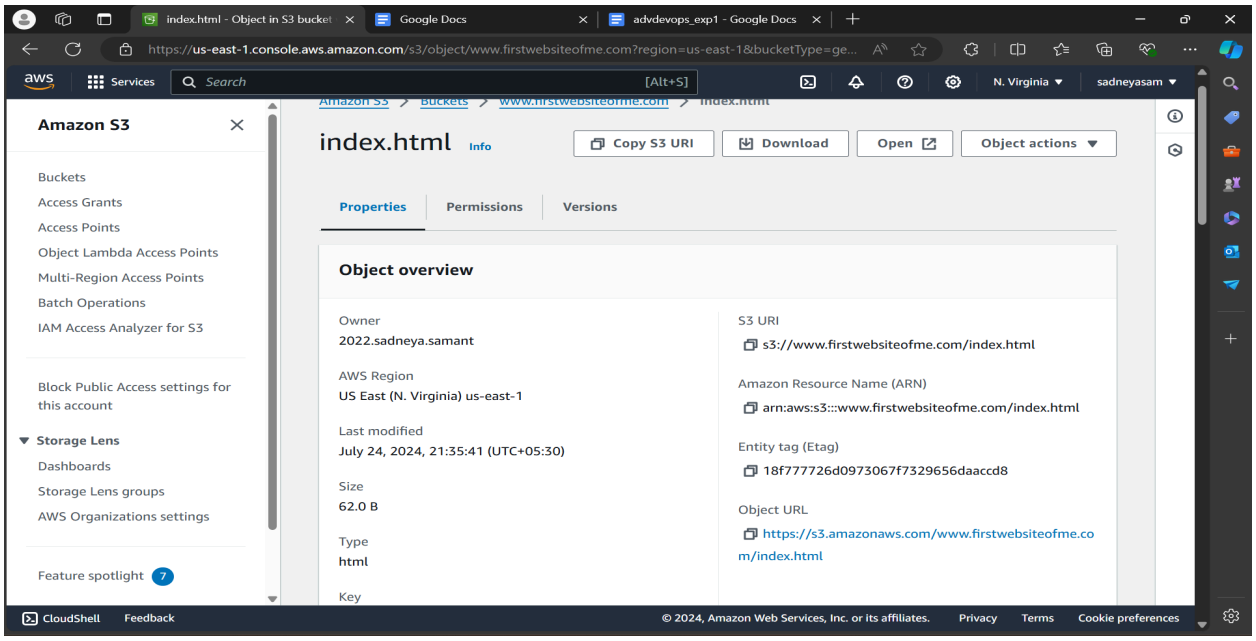
I clicked on the bucket.



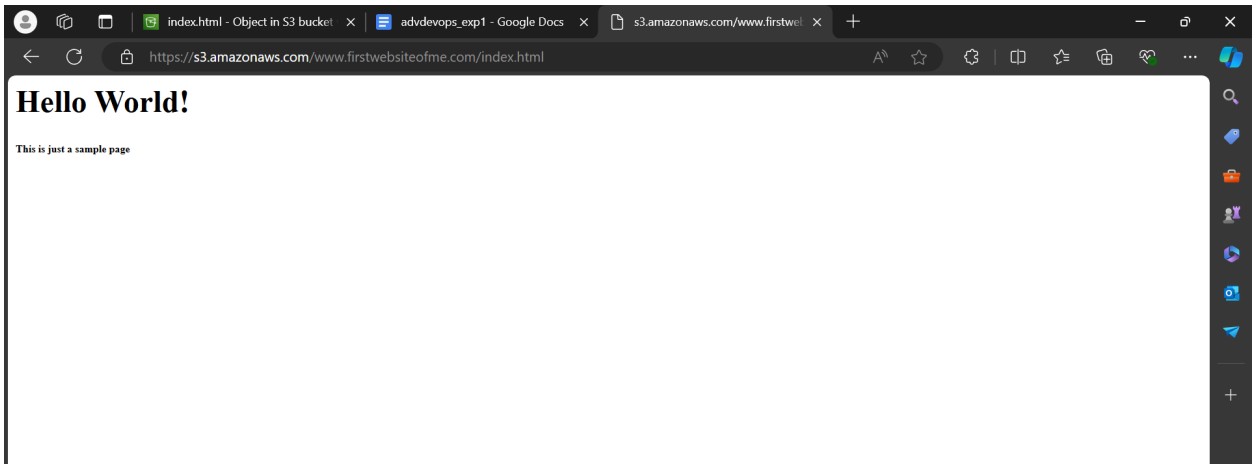
4. Then I also S3 webhosting-bucket policy with this code .I entered in the permission section there is an option edit bucket policy. Where I uploaded the code and saved it.



4. Then i clicked on index.html file and opened its properties section.



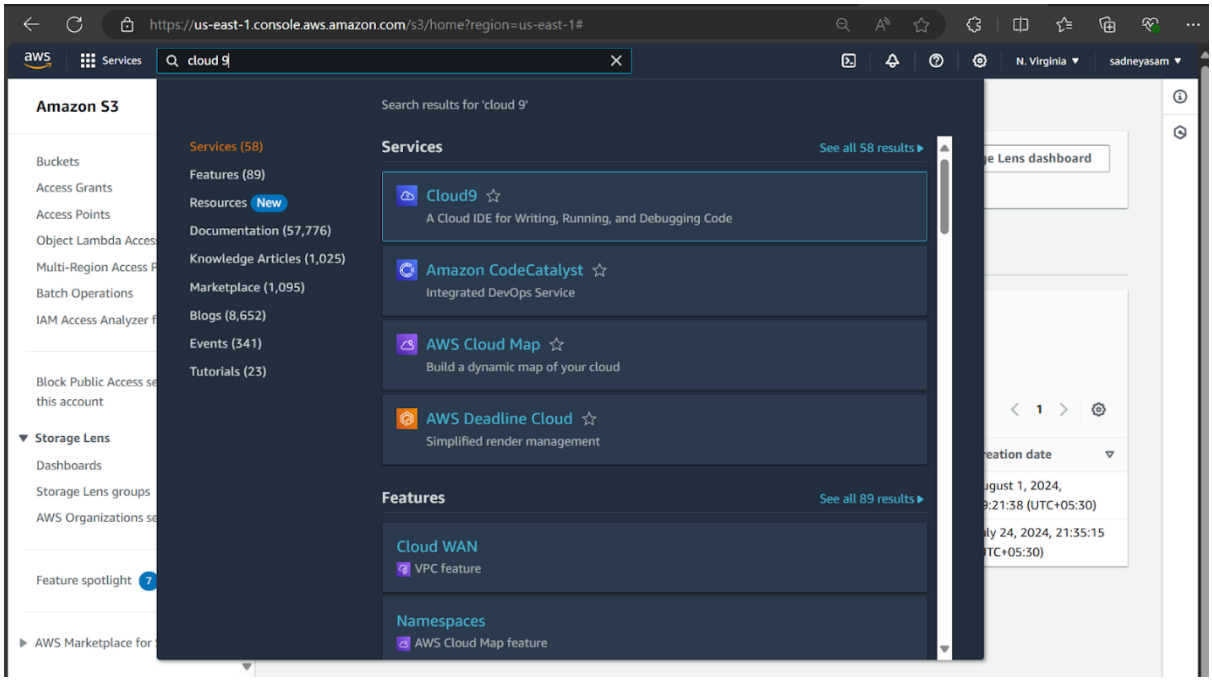
4. Then I clicked on https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html which given me the final output.



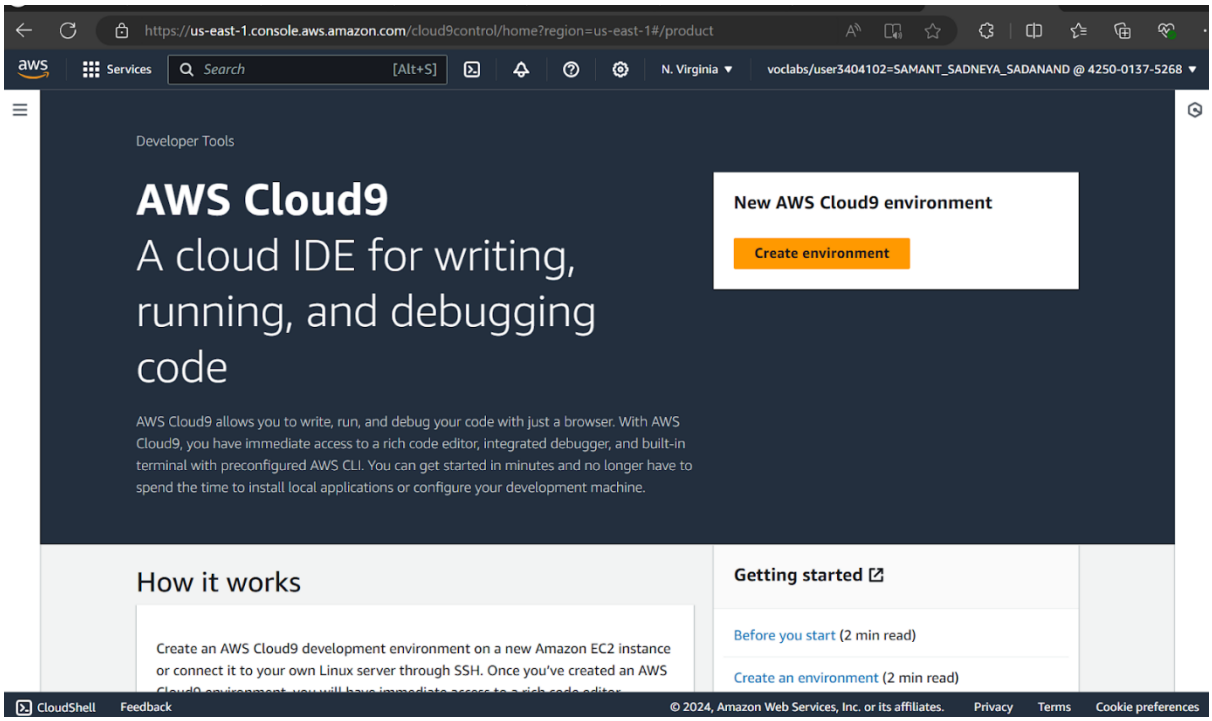
Experiment 1B

Creation of Cloud9 Environment

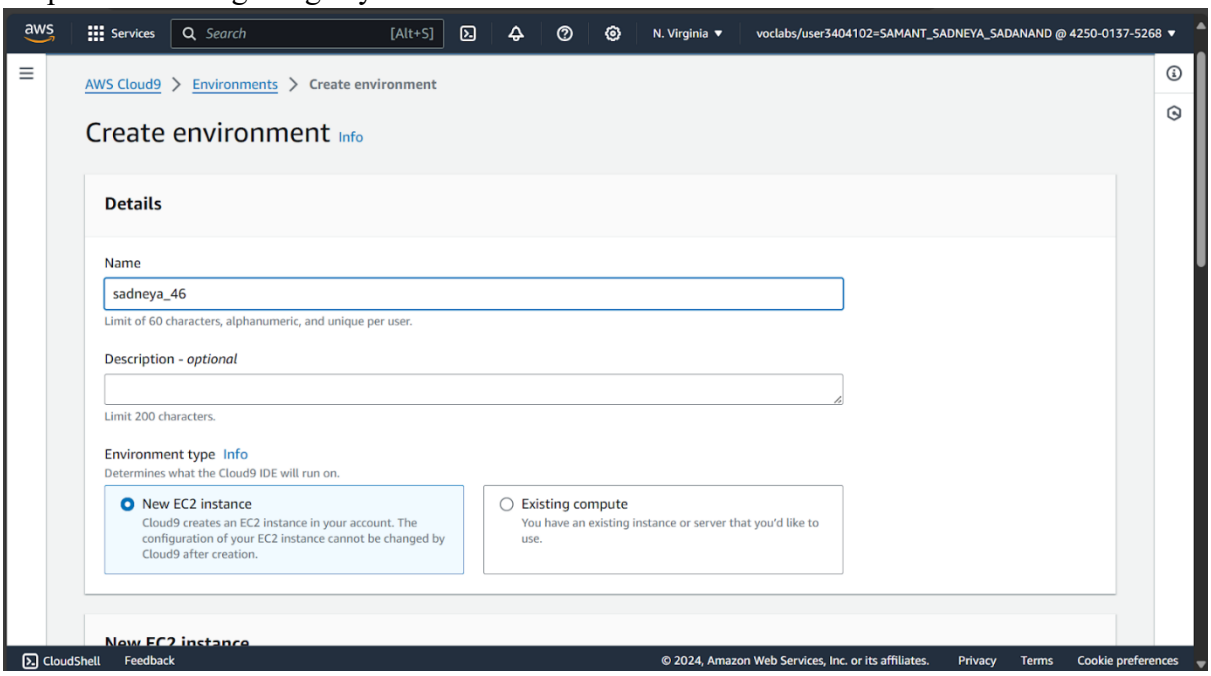
1.Go to the amazon console page.

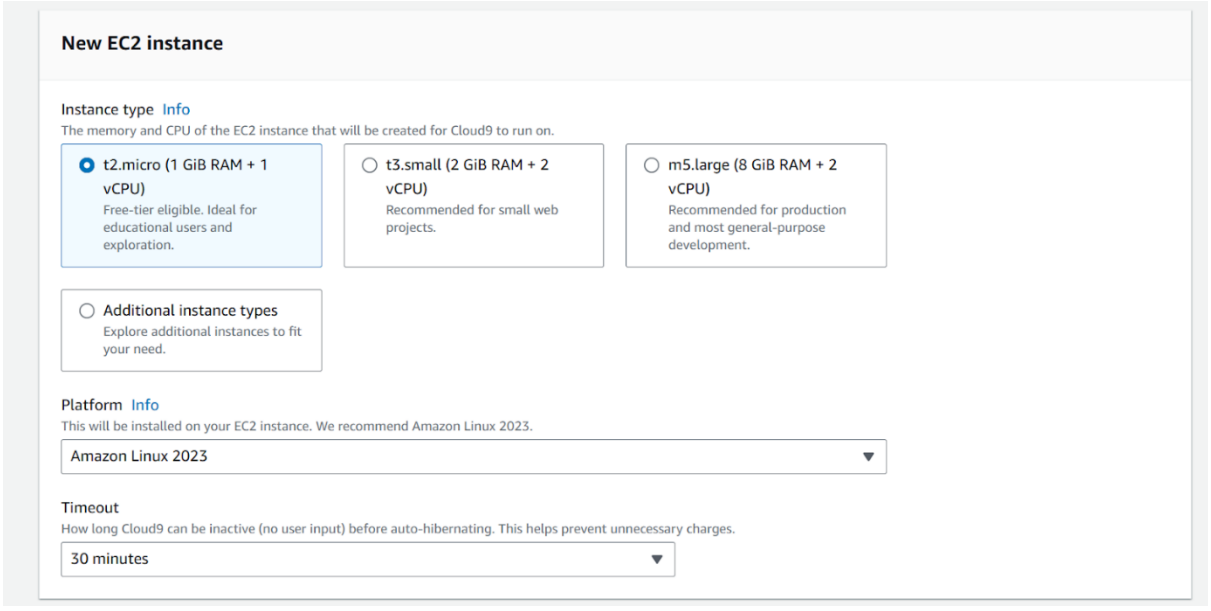


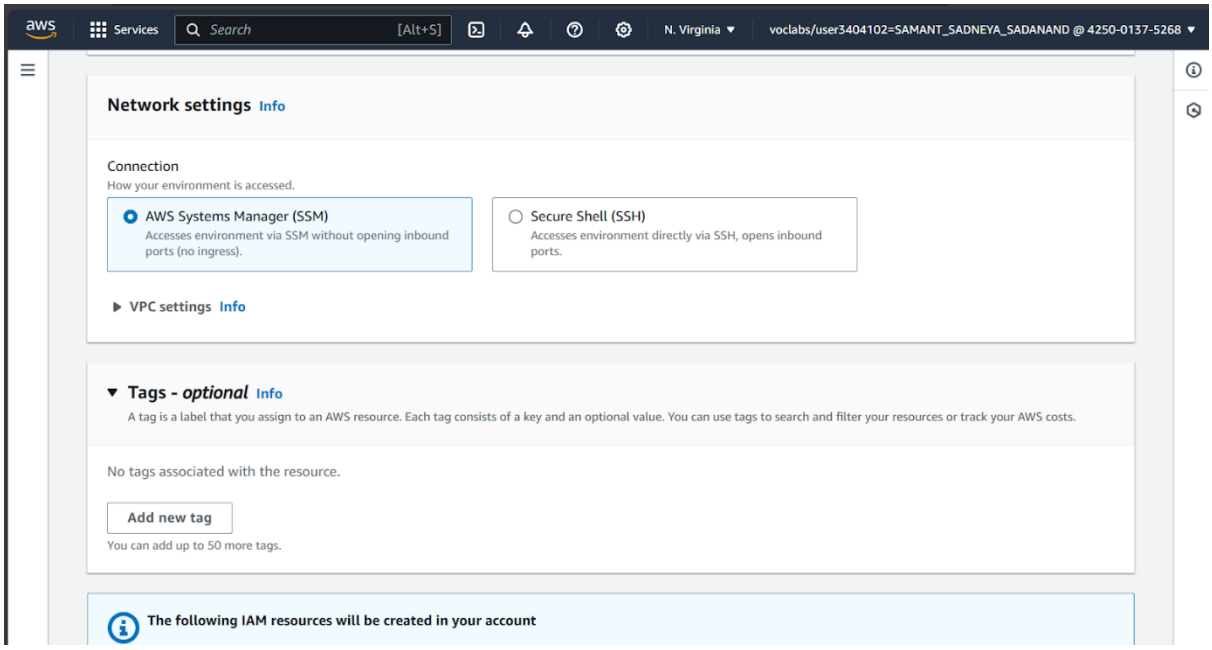
2.click on create environment.

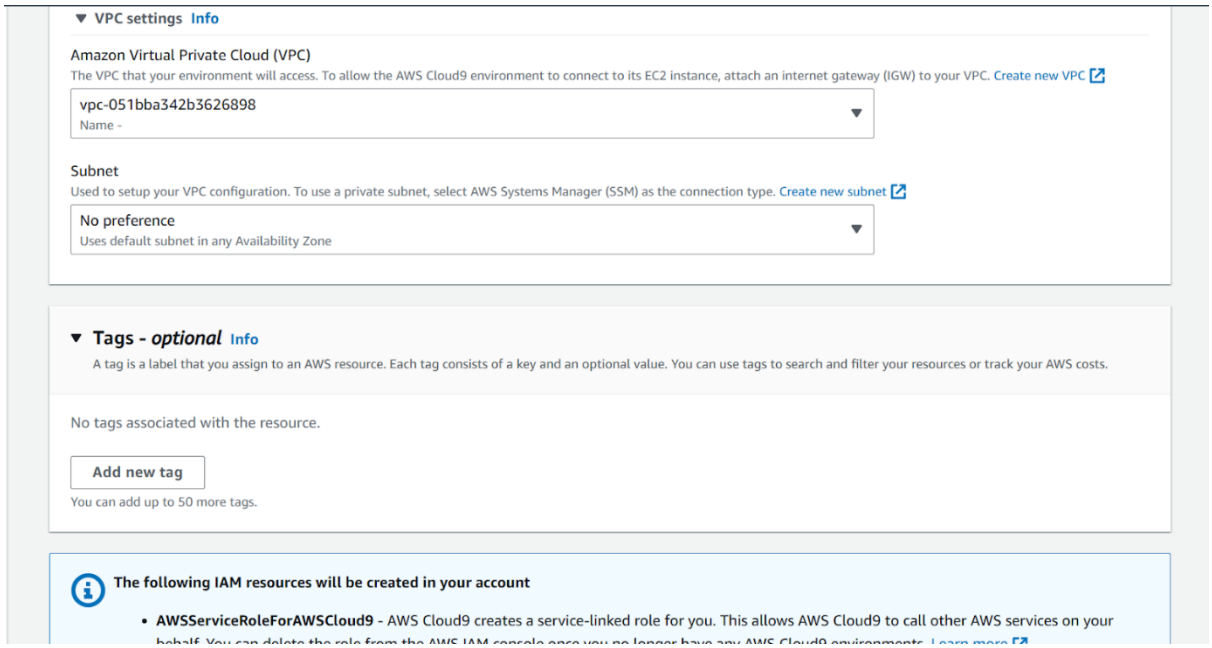


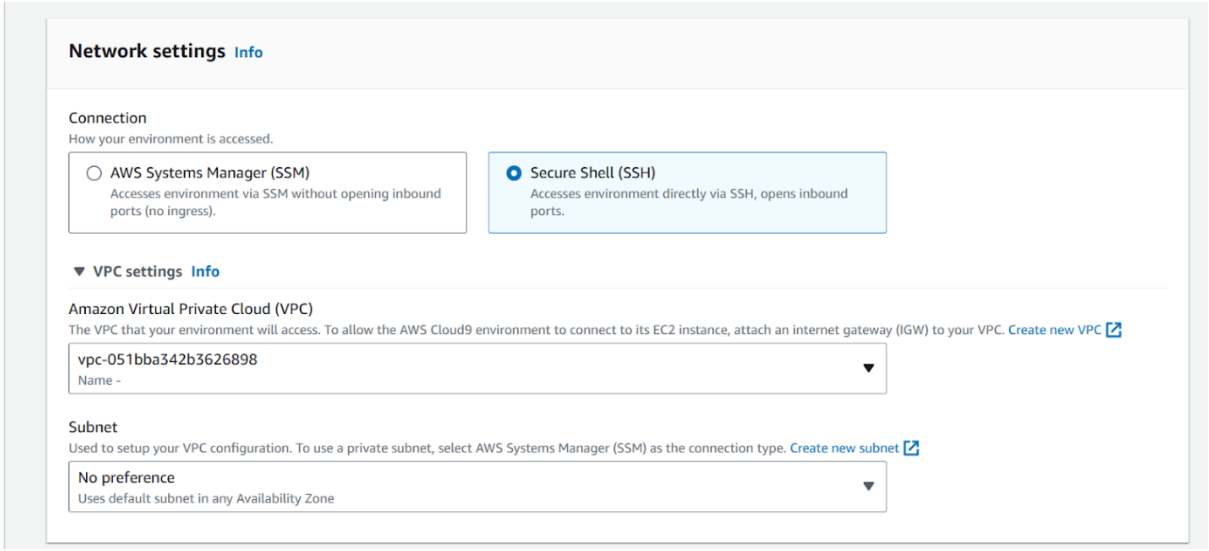
3. Give the name to your environment. Here I have given the name as sadneya_46. keep the remaining things by default.



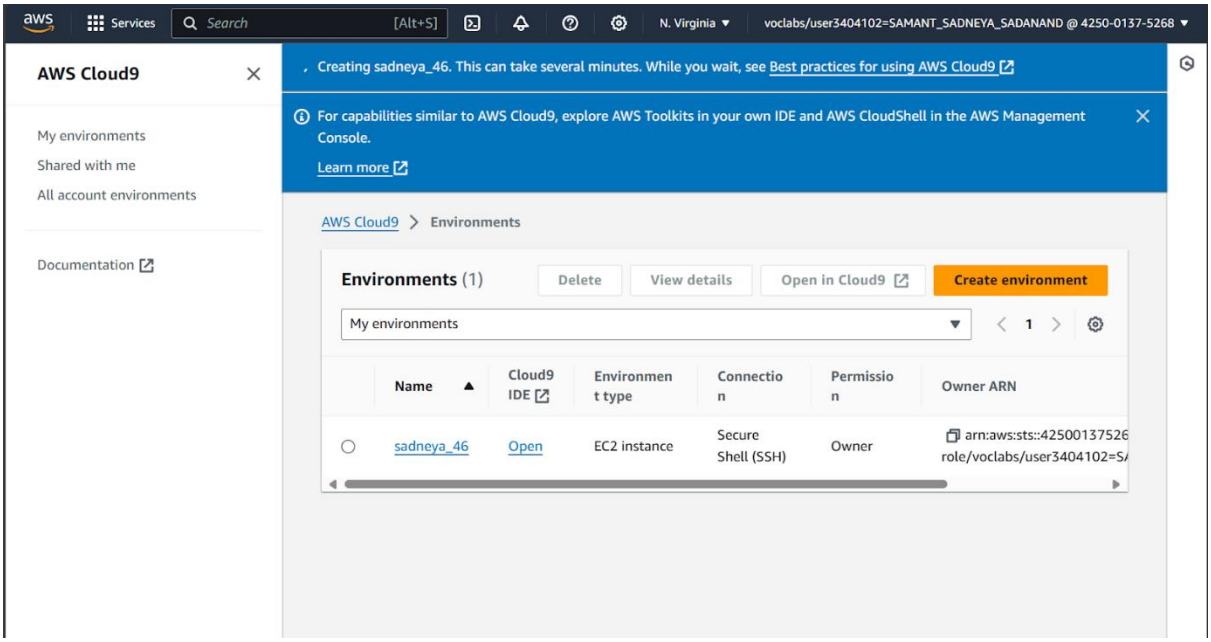



aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
Network settings Info
Connection
How your environment is accessed.
AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).
Secure Shell (SSH)
Accesses environment 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.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
The following IAM resources will be created in your account


VPC settings Info
Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. Create new VPC
vpc-051bba342b3626898
Name -
Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. Create new subnet
No preference
Uses default subnet in any Availability Zone
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.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
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

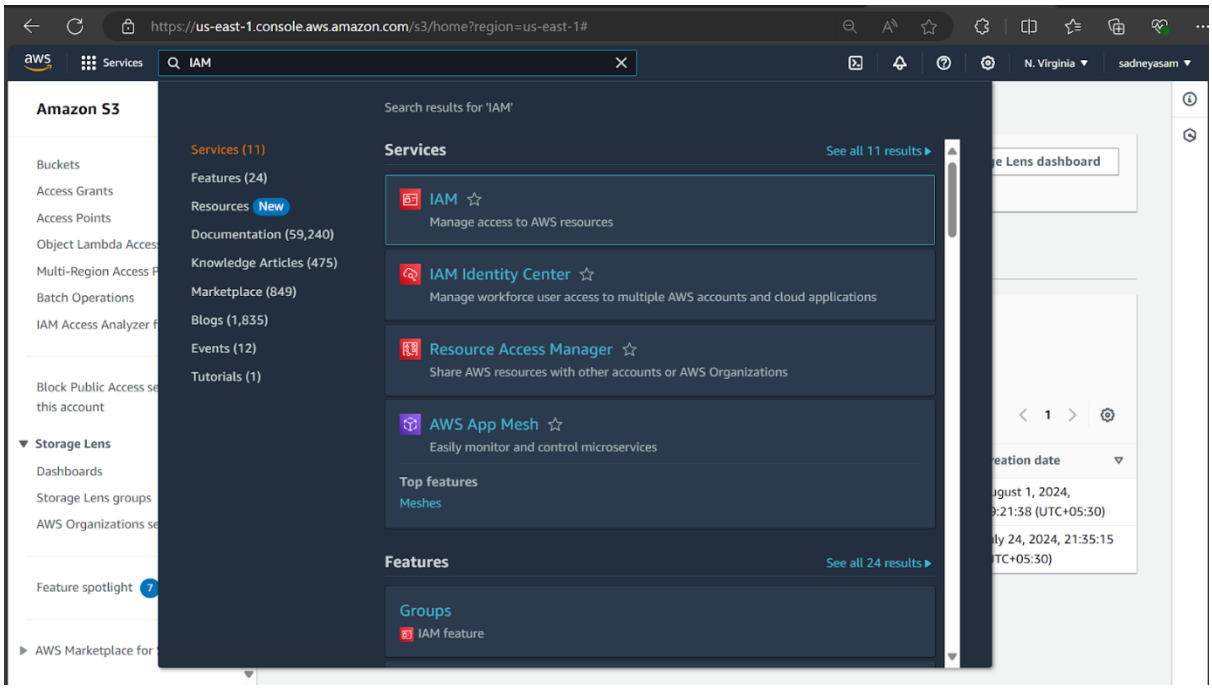


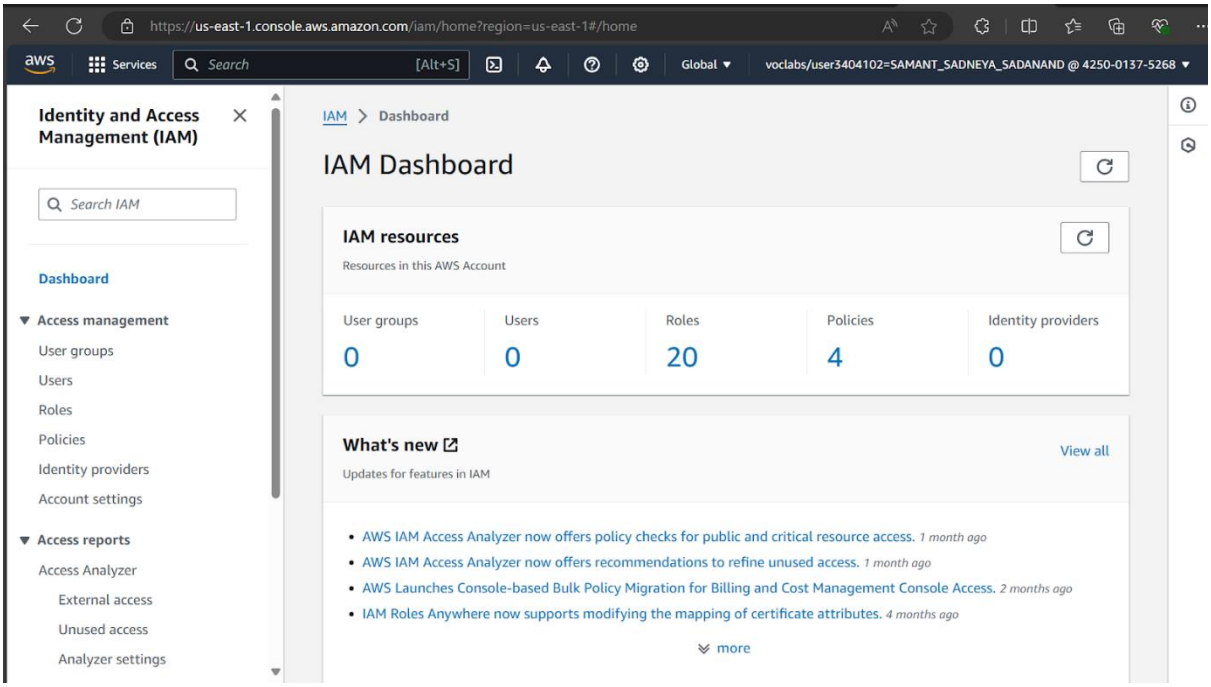
Thus, the environment was created.



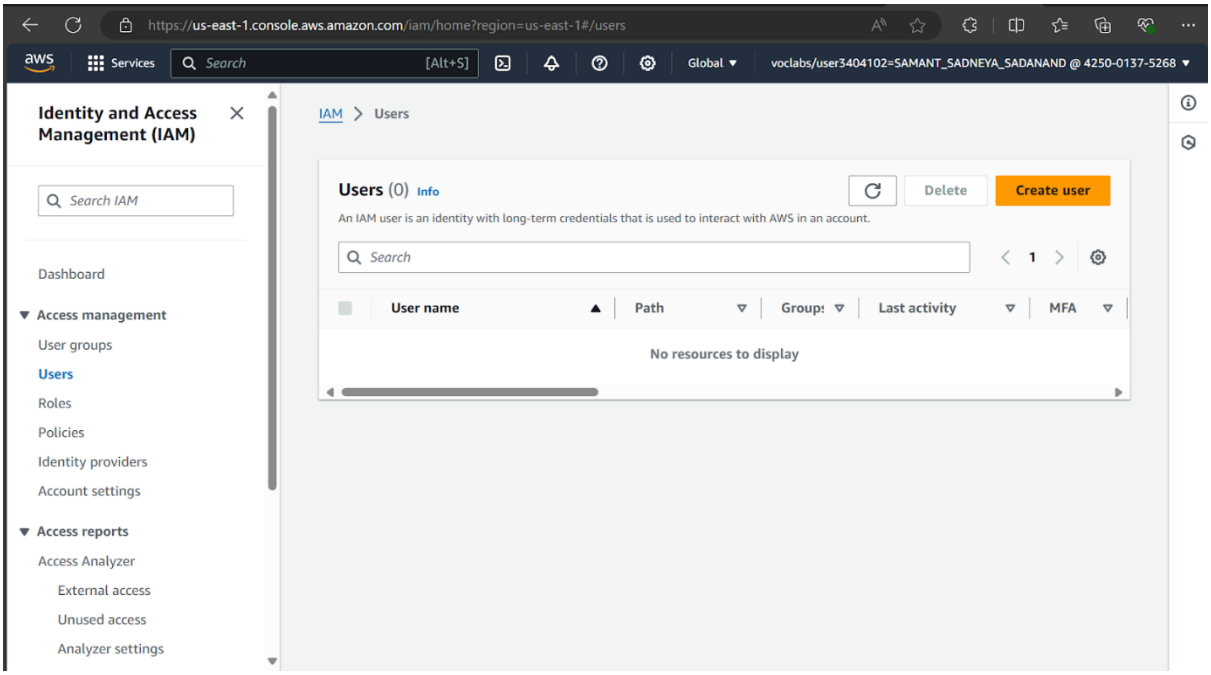
Creation of I am user

1. Go to I am.

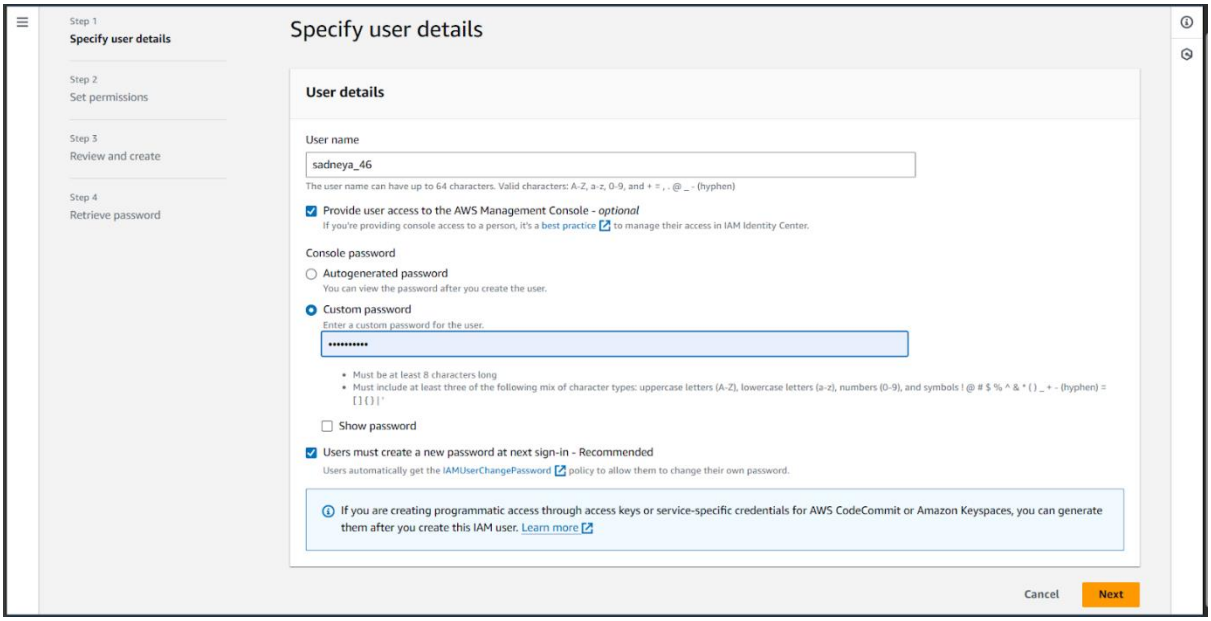




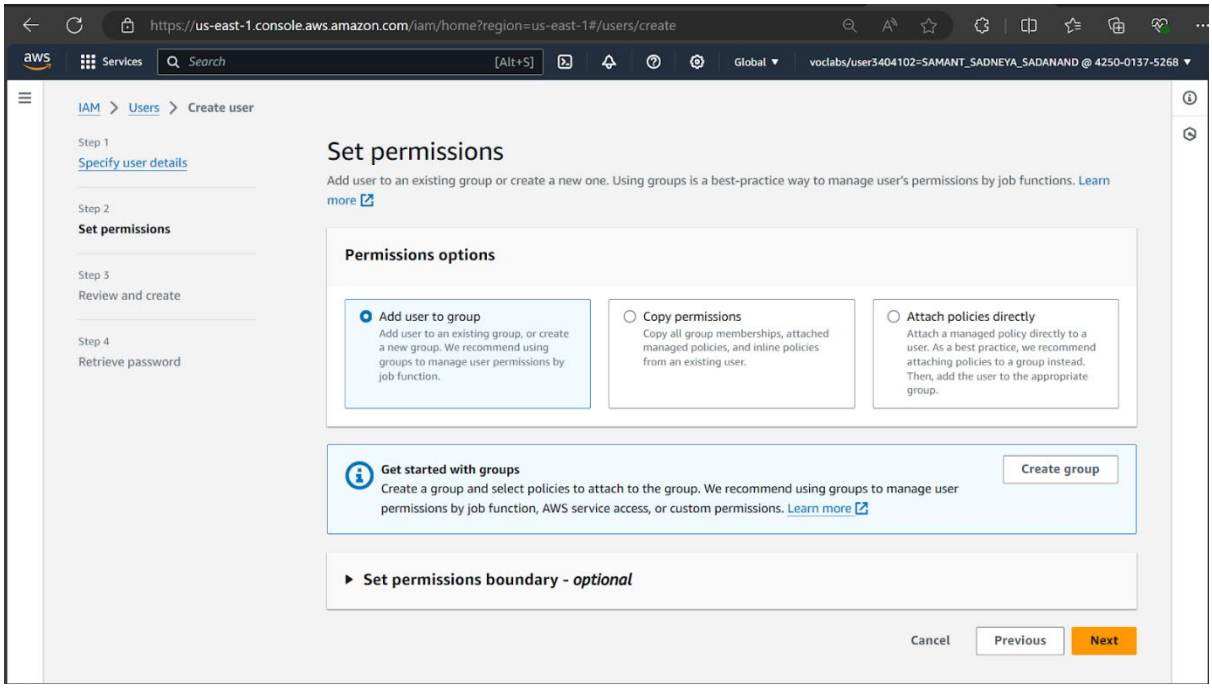
2. Select Users here.
3. Then Click on create user.



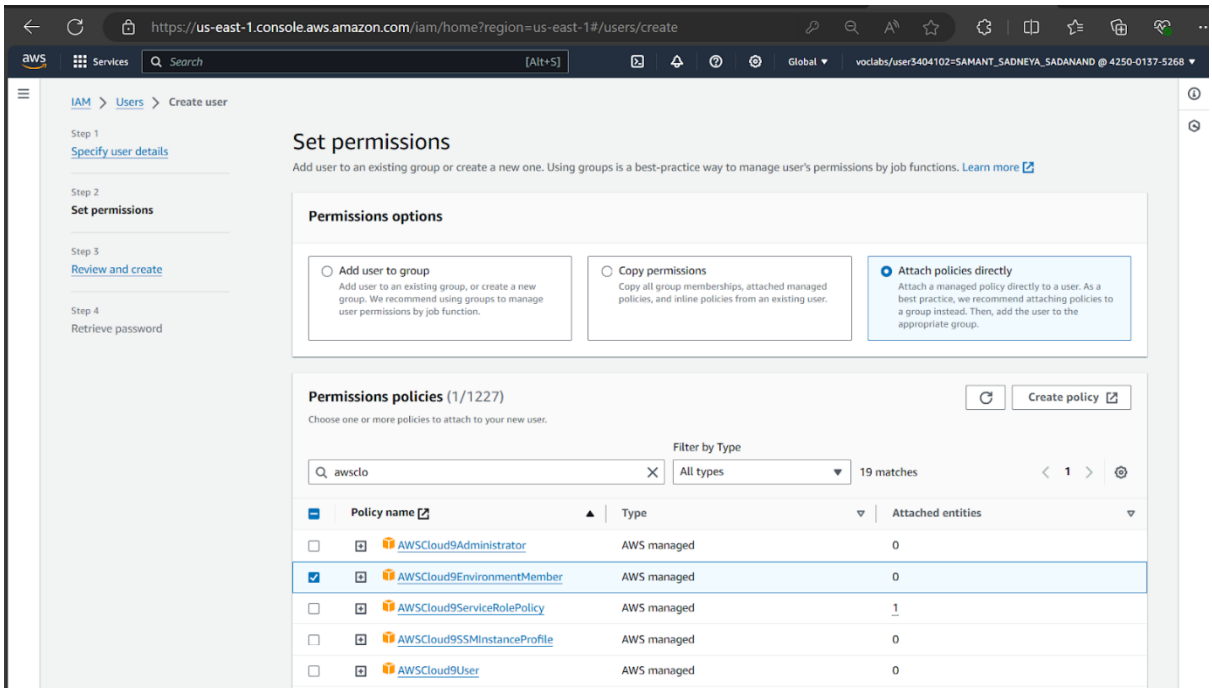
4. Give I AM user name. Here I have given it as sadneya_46. Select custom password and enter your password.



5. Select Add user to group. And click next.

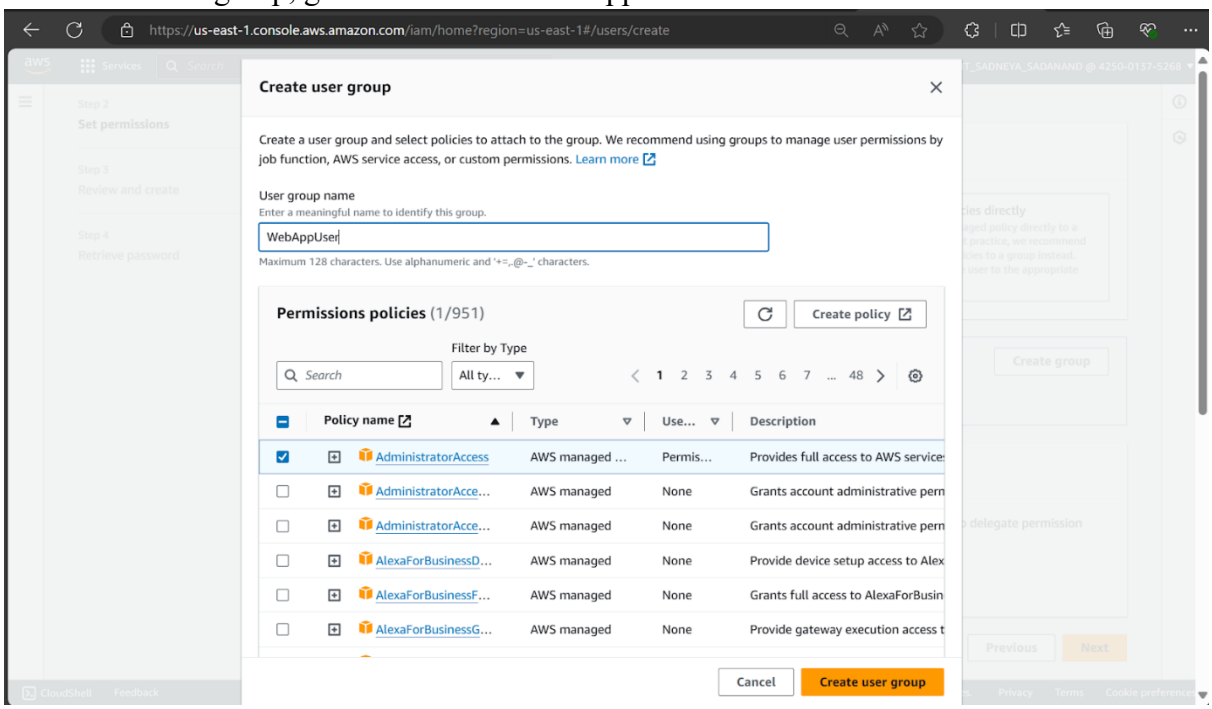


6. Also select permissions. Here I have selected AWSCloud9EnvironmentMember for cloud9 Environment. then click on next.

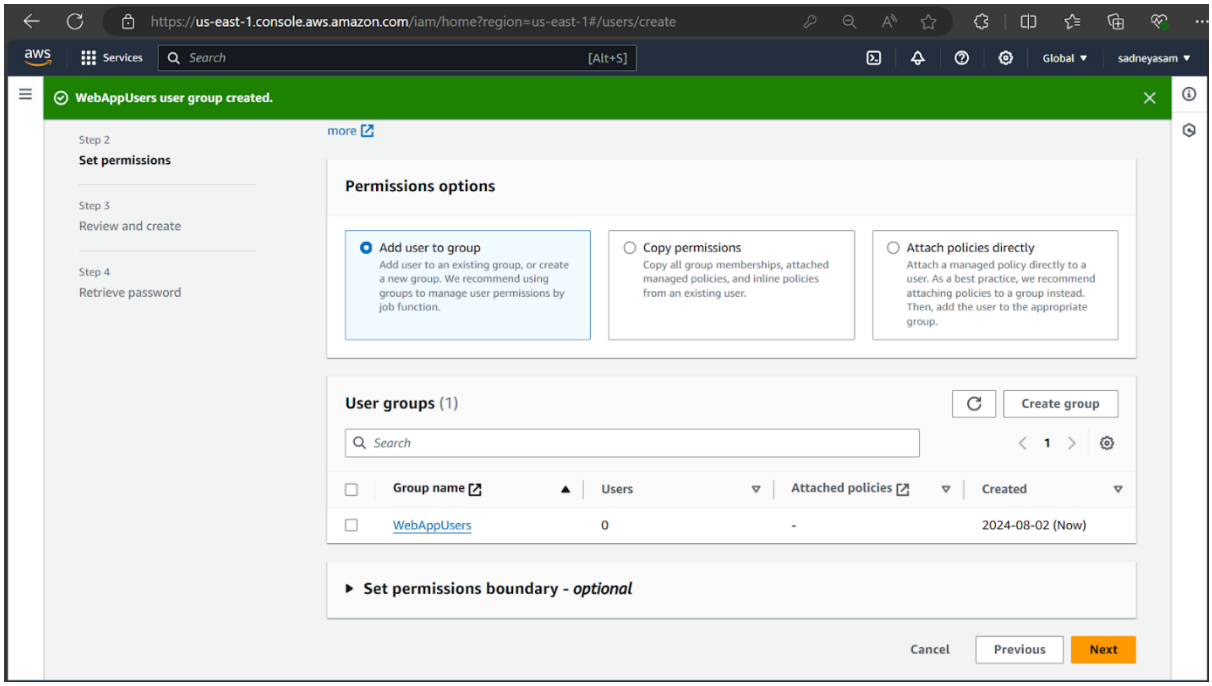


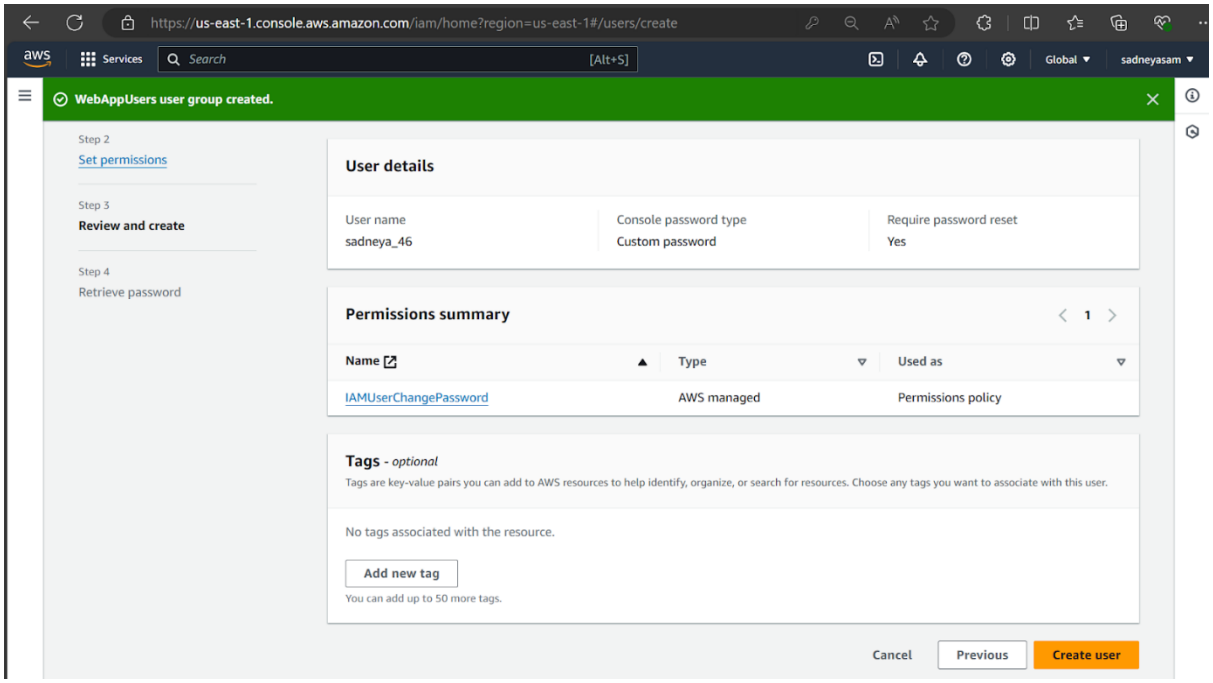
To add user, we need to create

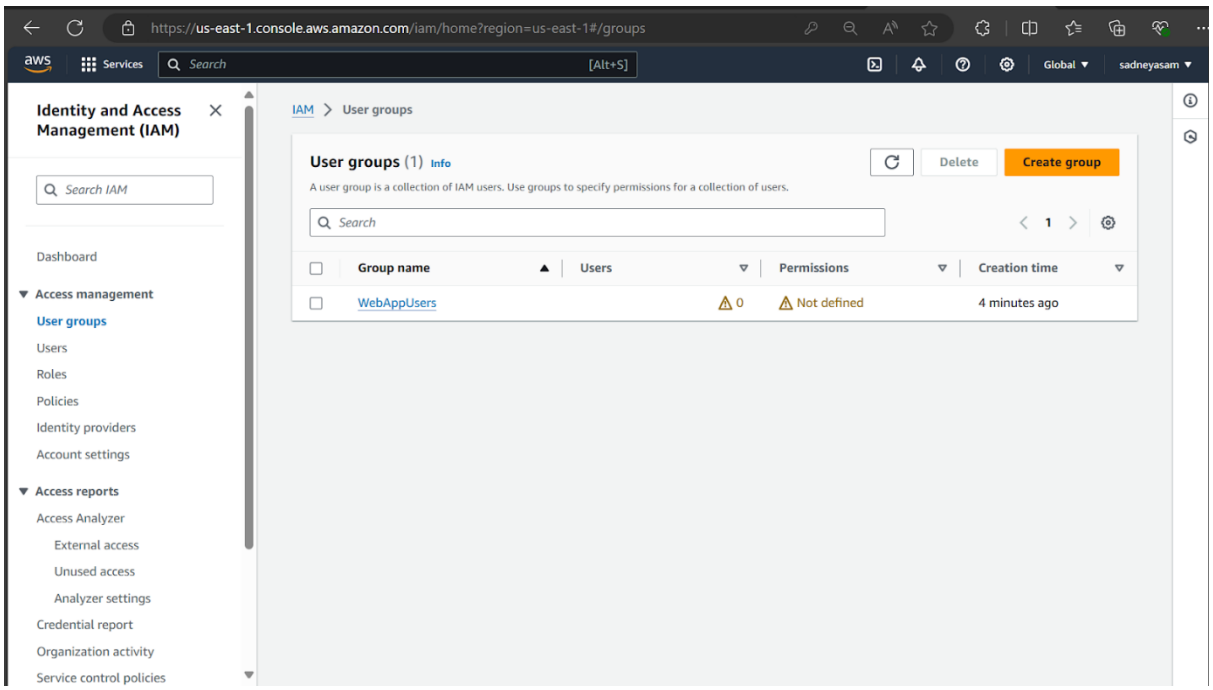
7. To create a user group, give it a name as WebAppUser.

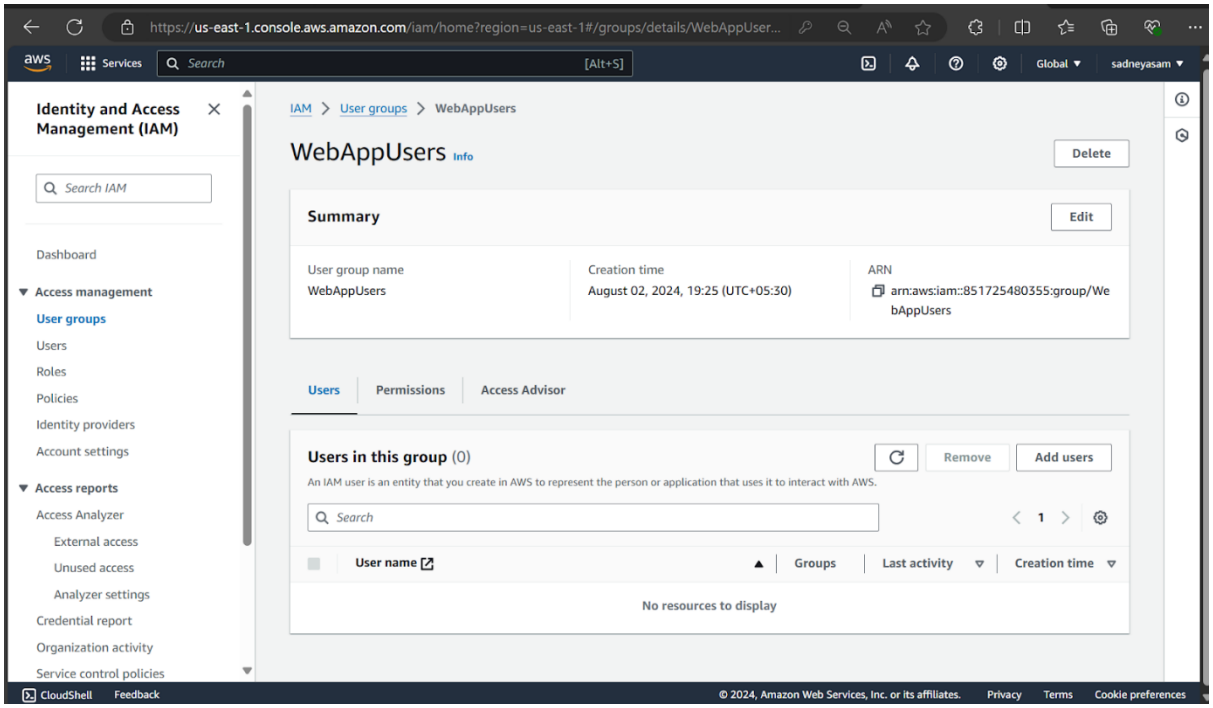


8. Thus, WebAppUser get created successfully.

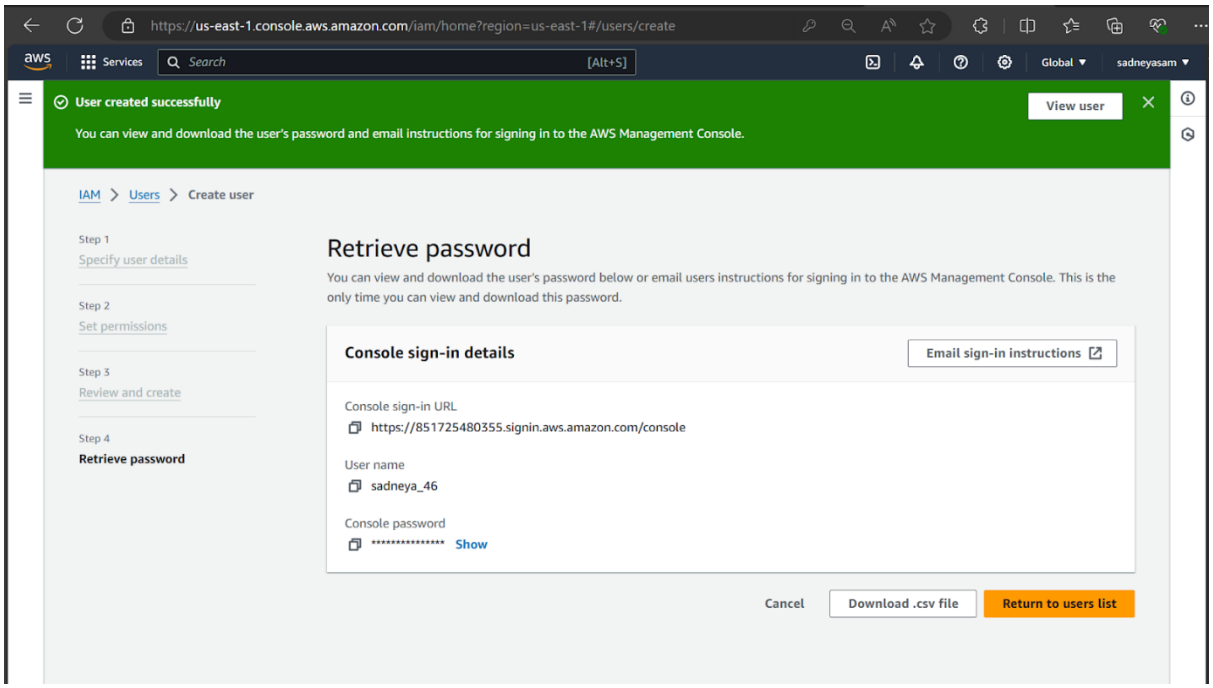



https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
aws
Services
Search
[Alt+S]
Global
sadneyasam
WebAppUsers user group created.
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
User details
User name
sadneya_46
Console password type
Custom password
Require password reset
Yes
Permissions summary
Name
Type
Used as
IAMUserChangePassword
AWS managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user

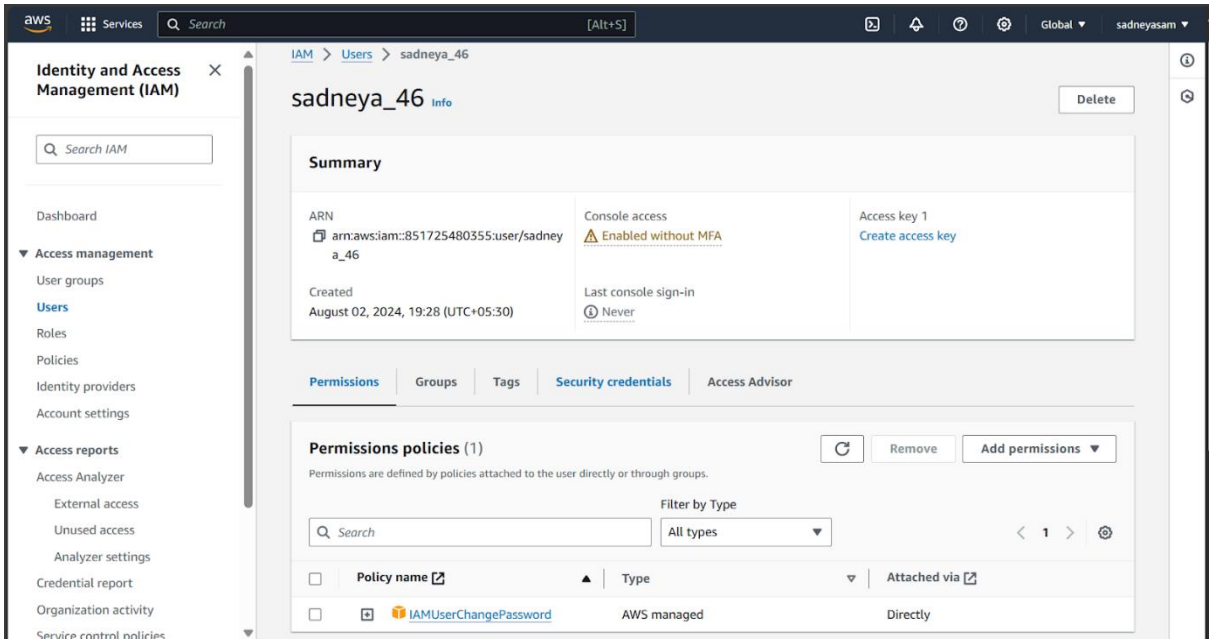

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/groups
aws
Services
Search
[Alt+S]
Global
sadneyasam
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
IAM
User groups
User groups (1) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.
Delete
Create group
Search
Group name
Users
Permissions
Creation time
WebAppUsers
0
Not defined
4 minutes ago



9. Then click on next and thus the user gets successfully created.



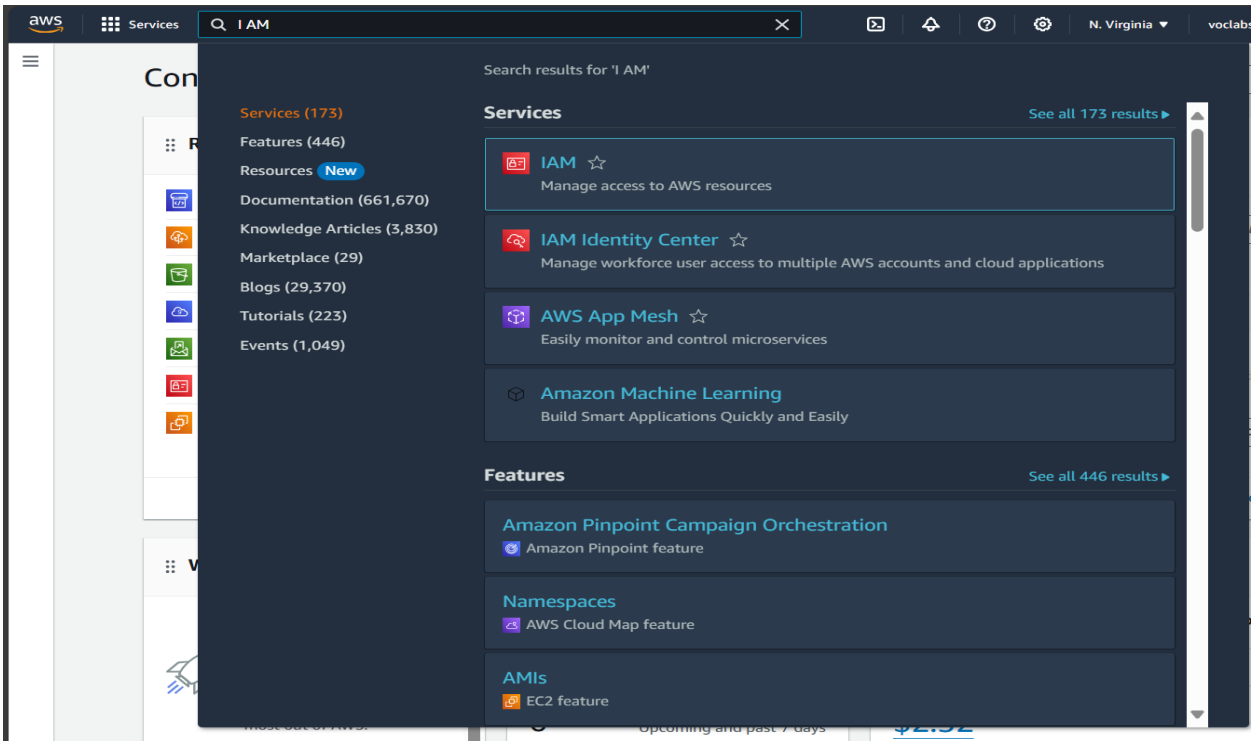
10. This is a summary of user created.


aws
Services
Search
[Alt+S]
Global
sadneyasam
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
IAM > Users > sadneya_46
sadneya_46 Info
Delete
Summary
ARN
arn:aws:iam::851725480355:user/sadneya_46
Console access
Enabled without MFA
Access key 1
Create access key
Created
August 02, 2024, 19:28 (UTC+05:30)
Last console sign-in
Never
Permissions
Groups
Tags
Security credentials
Access Advisor
Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
1
Policy name
Type
Attached via
IAMUserChangePassword
AWS managed
Directly

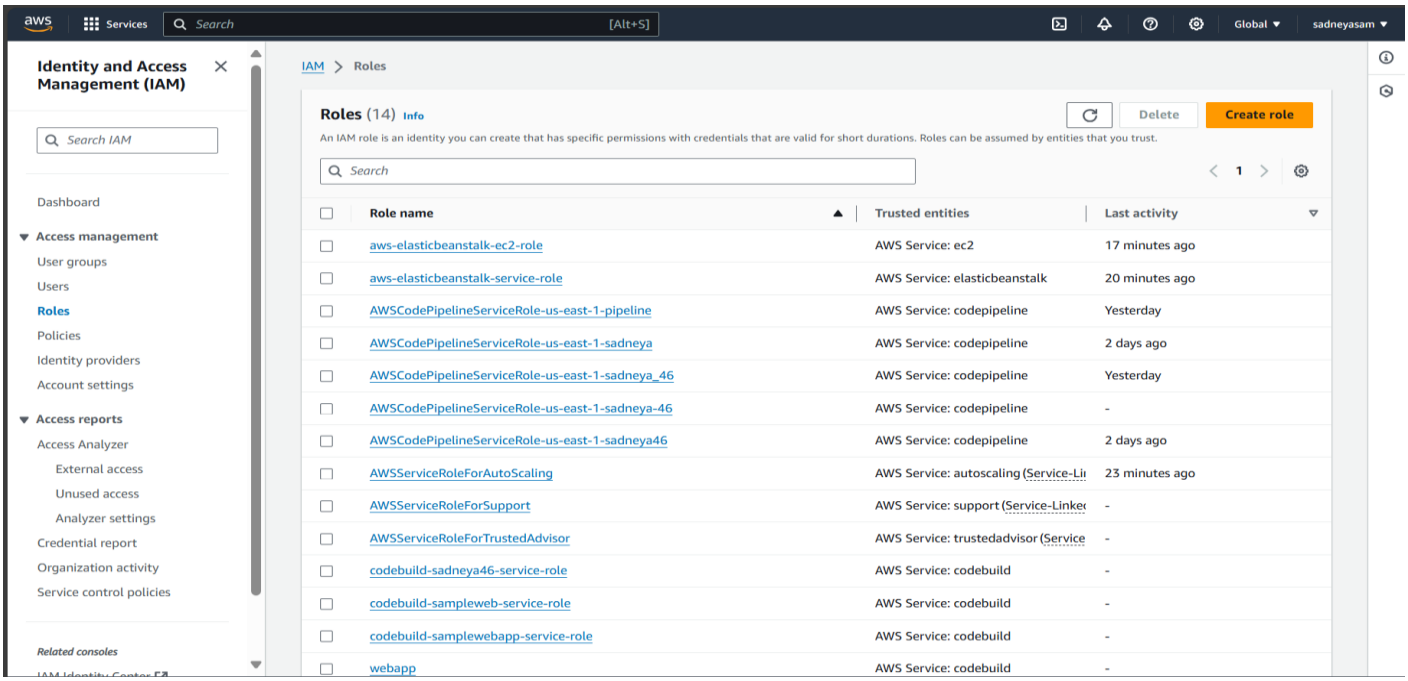
Experiment No: 2

Step1:- Creation of role:-

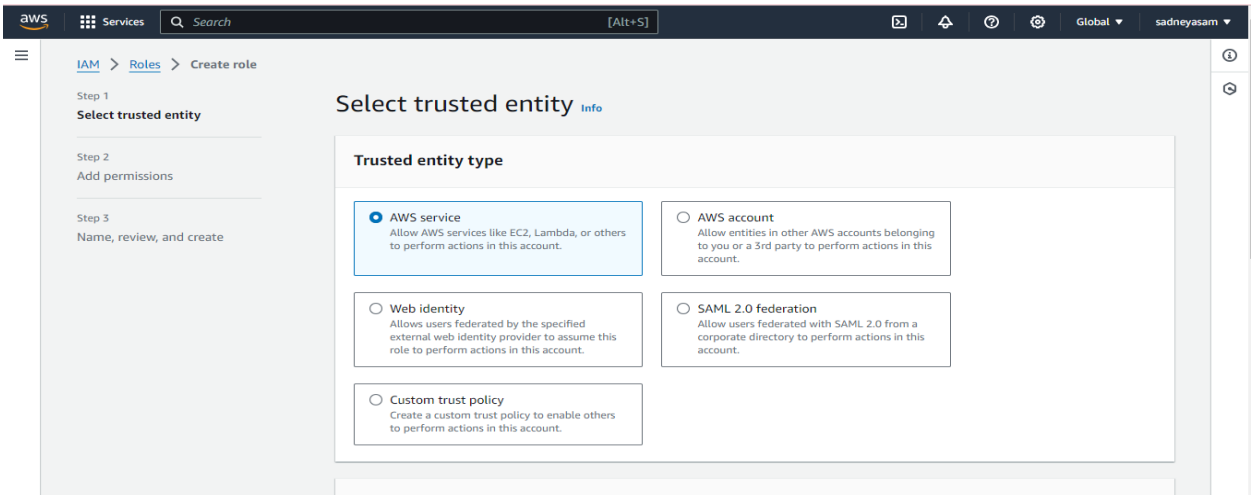
1. Login to your AWS account and search for IAM



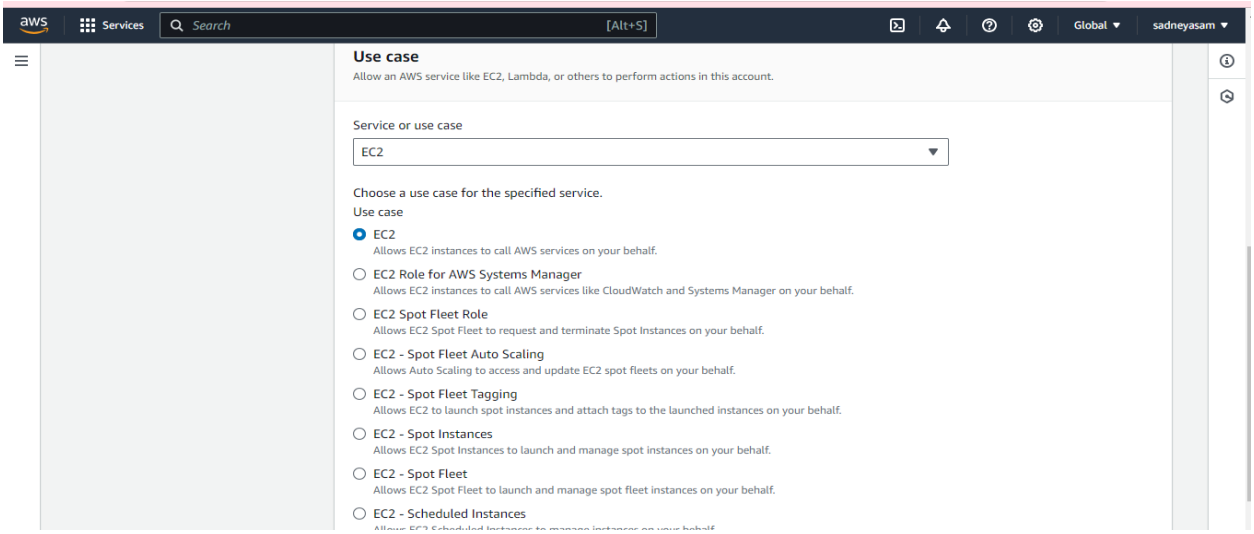
2. Then go into the role section and click on create role.



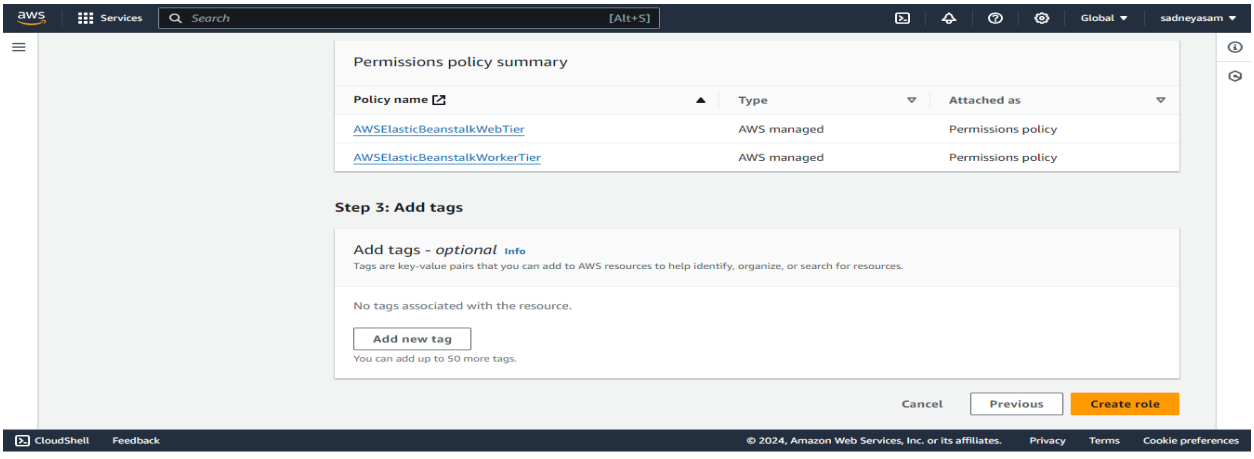
3. Then select a trusted entity as AWS service.



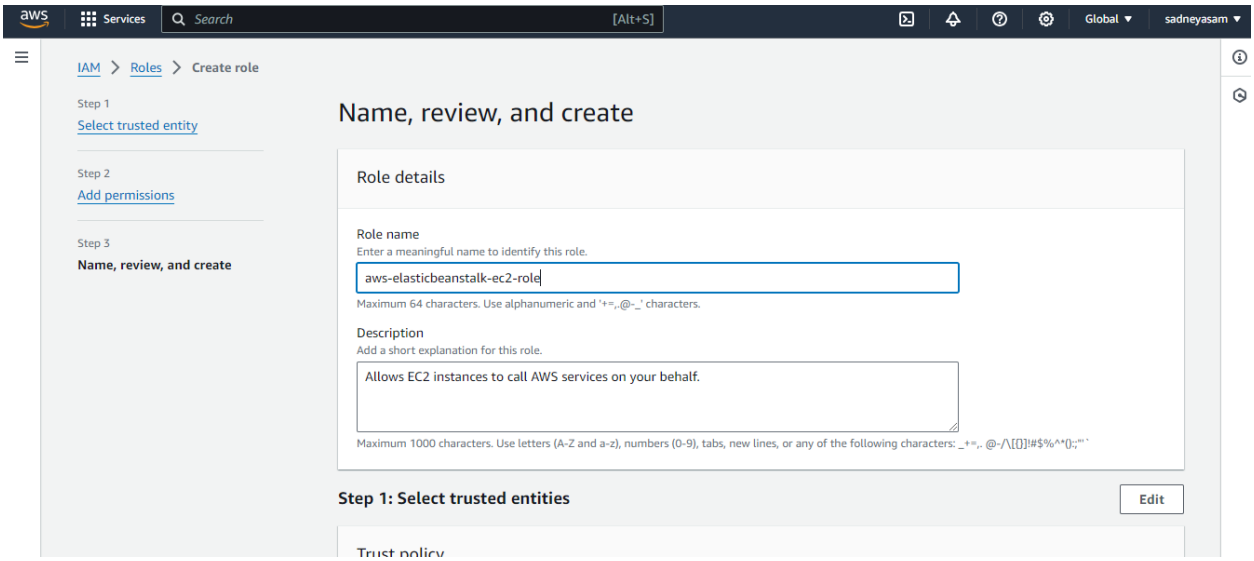
4. Select use case as EC2.



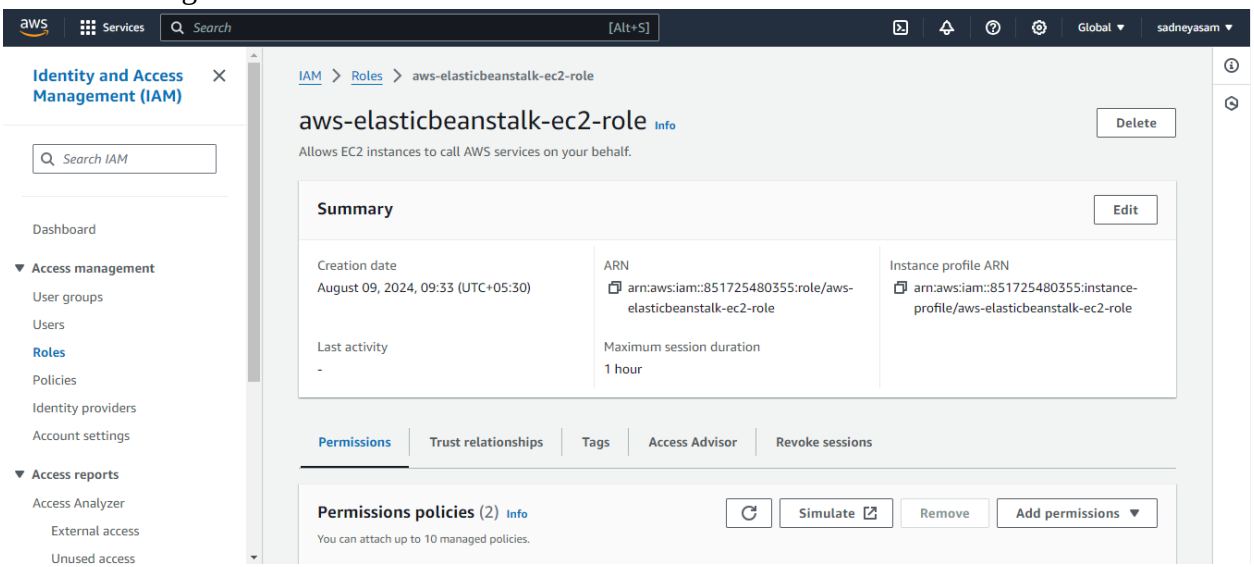
5. Select permissions as AWS Elastic Benstalk Web Tier.and AWS elastic Benstalk worker tier.



6. Give a name to Role. Here I have given my role name as aws -elasticbenstalk -ec2 -role.

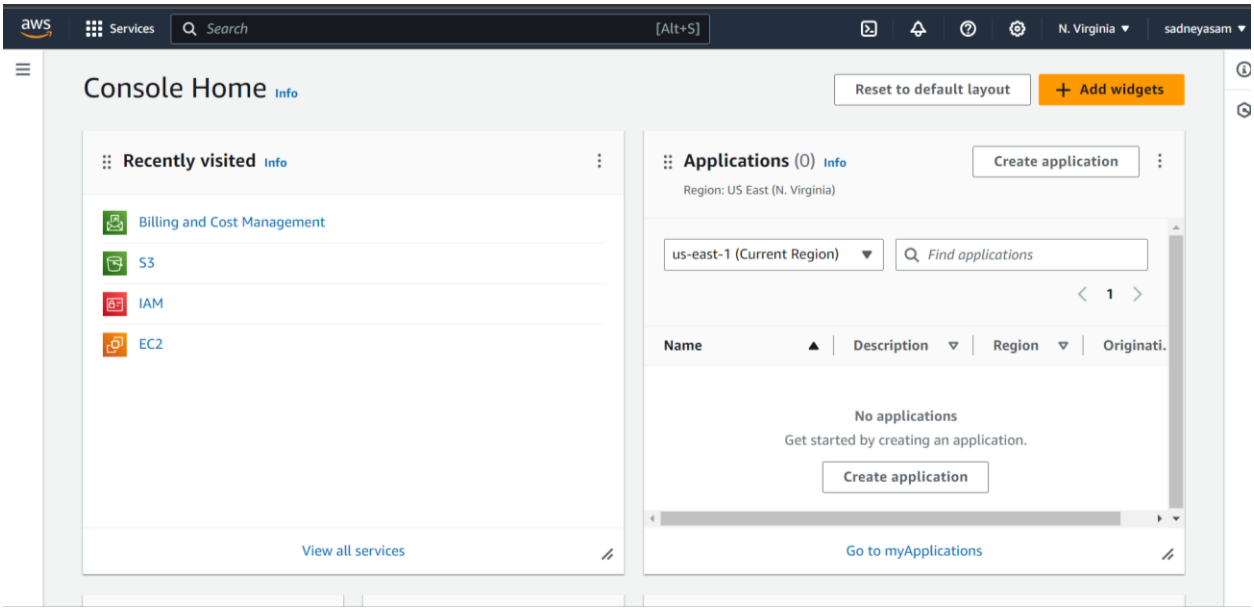


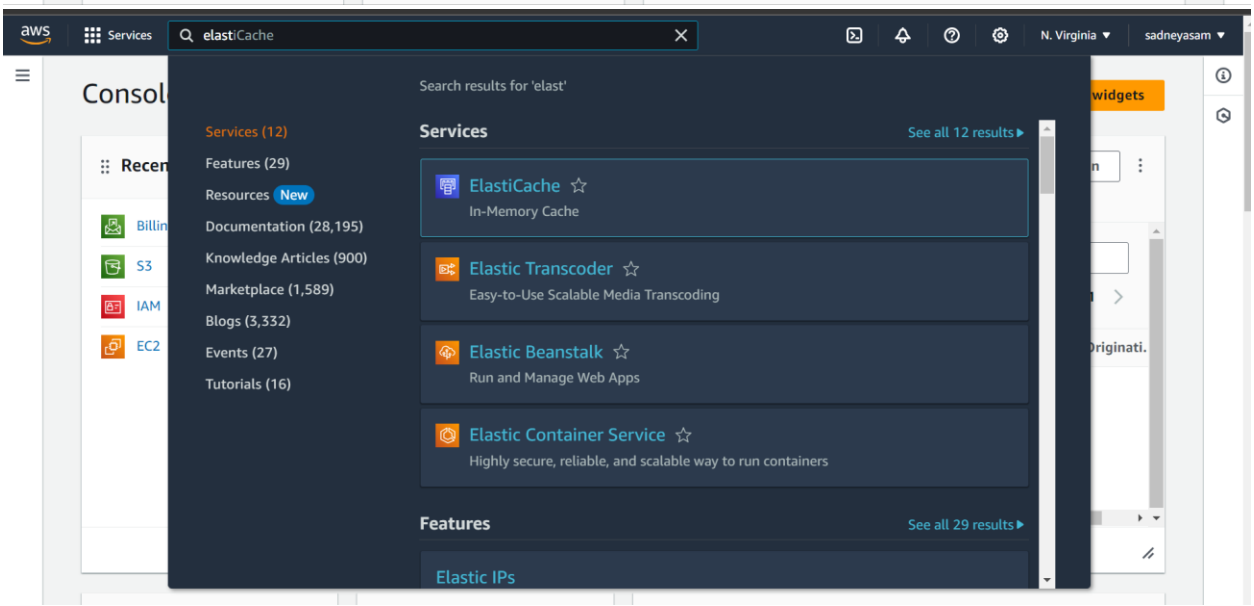
7. Then the role gets created.



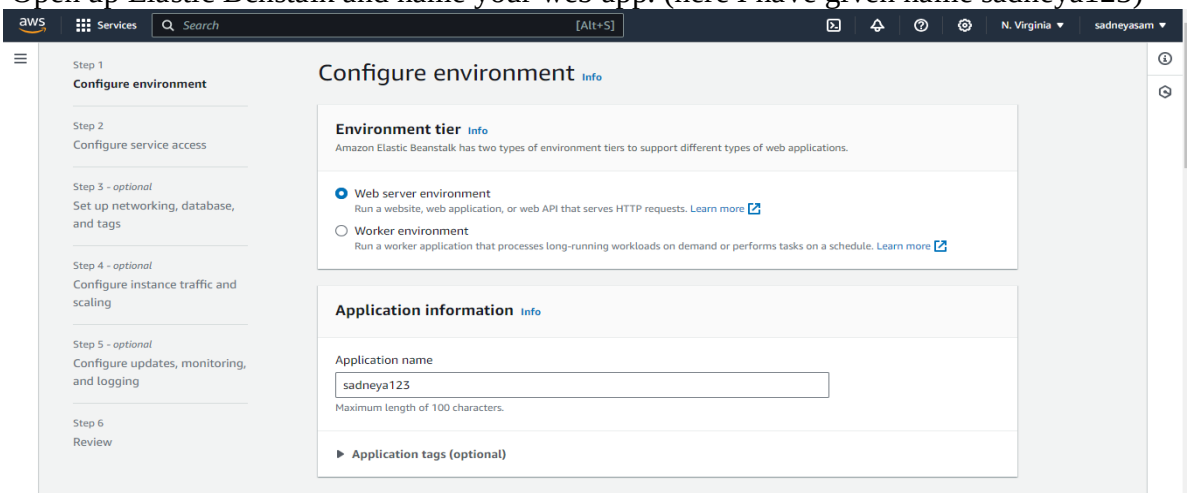
Step2:- Creation Elastic Benstalk Environment

1. search for Elastic Beanstalk in the search box.

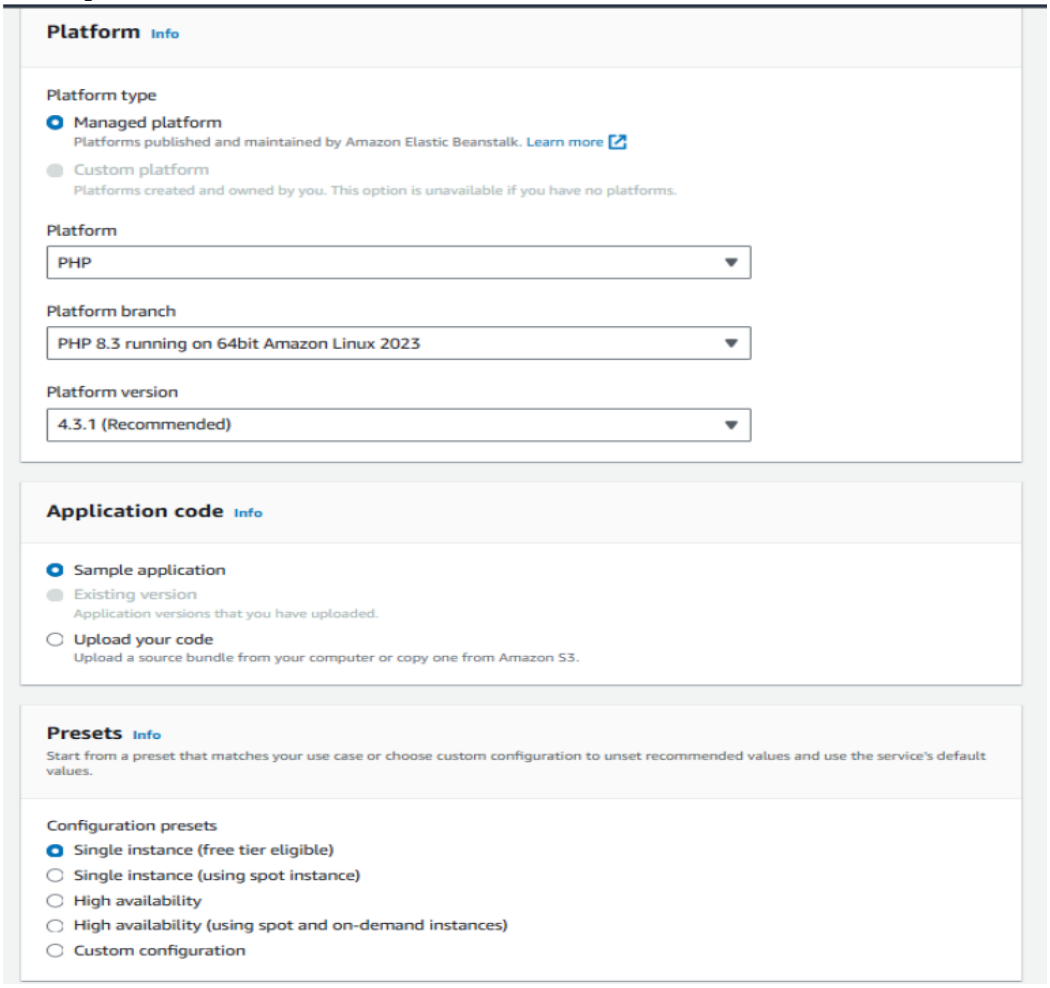




2. Open up Elastic Benstalk and name your web app. (here I have given name sadneya123)

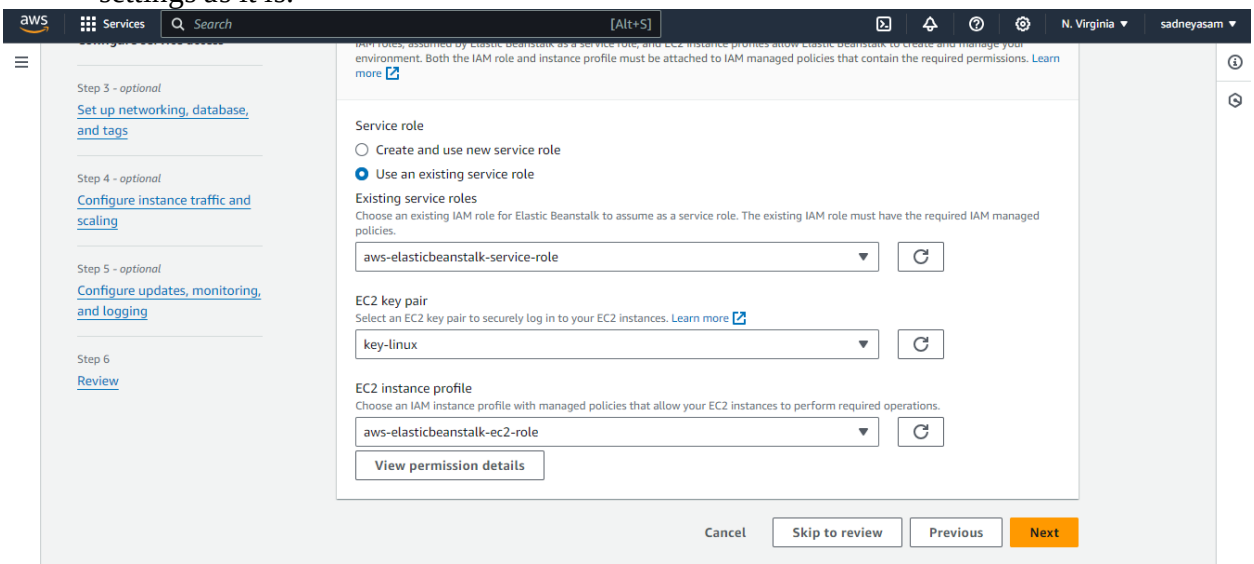


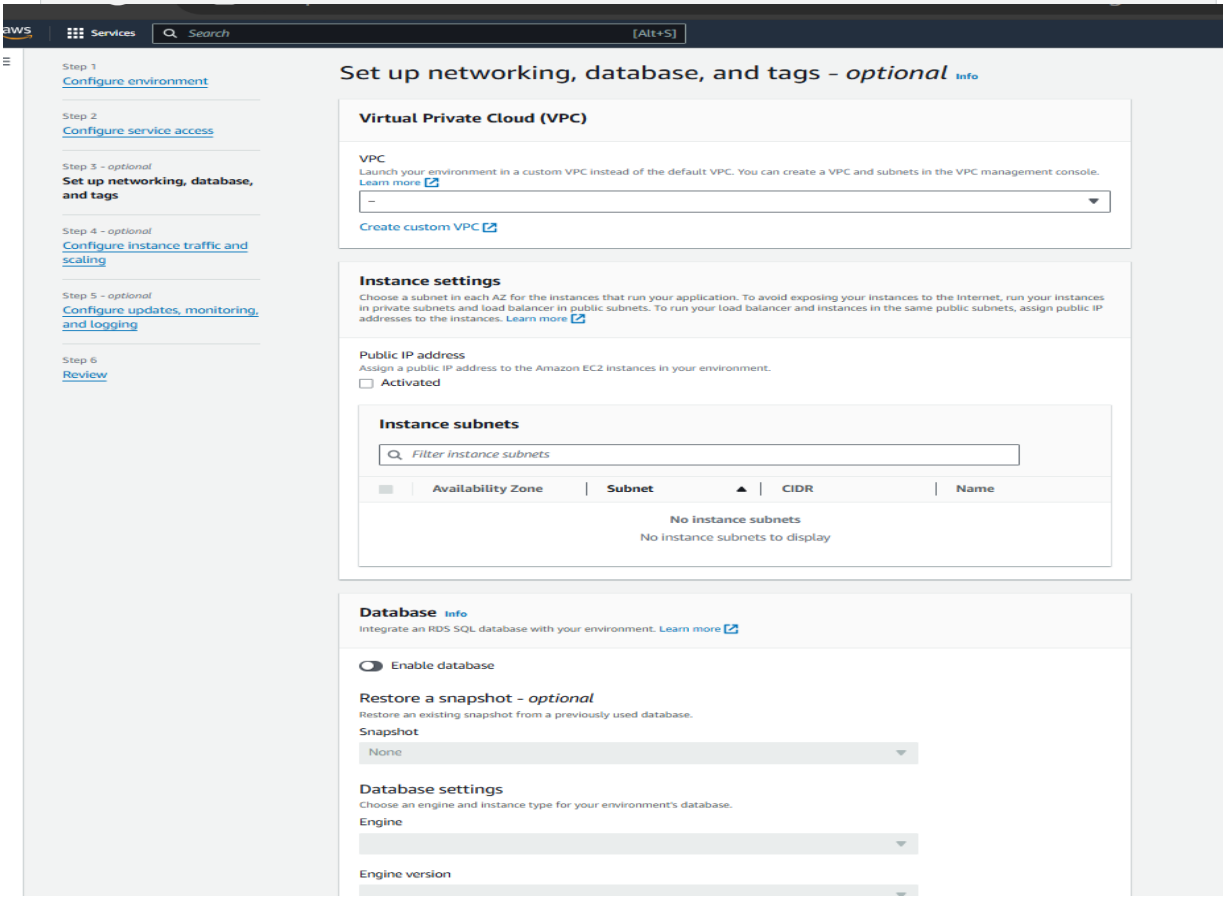
3. Select platform as PHP.

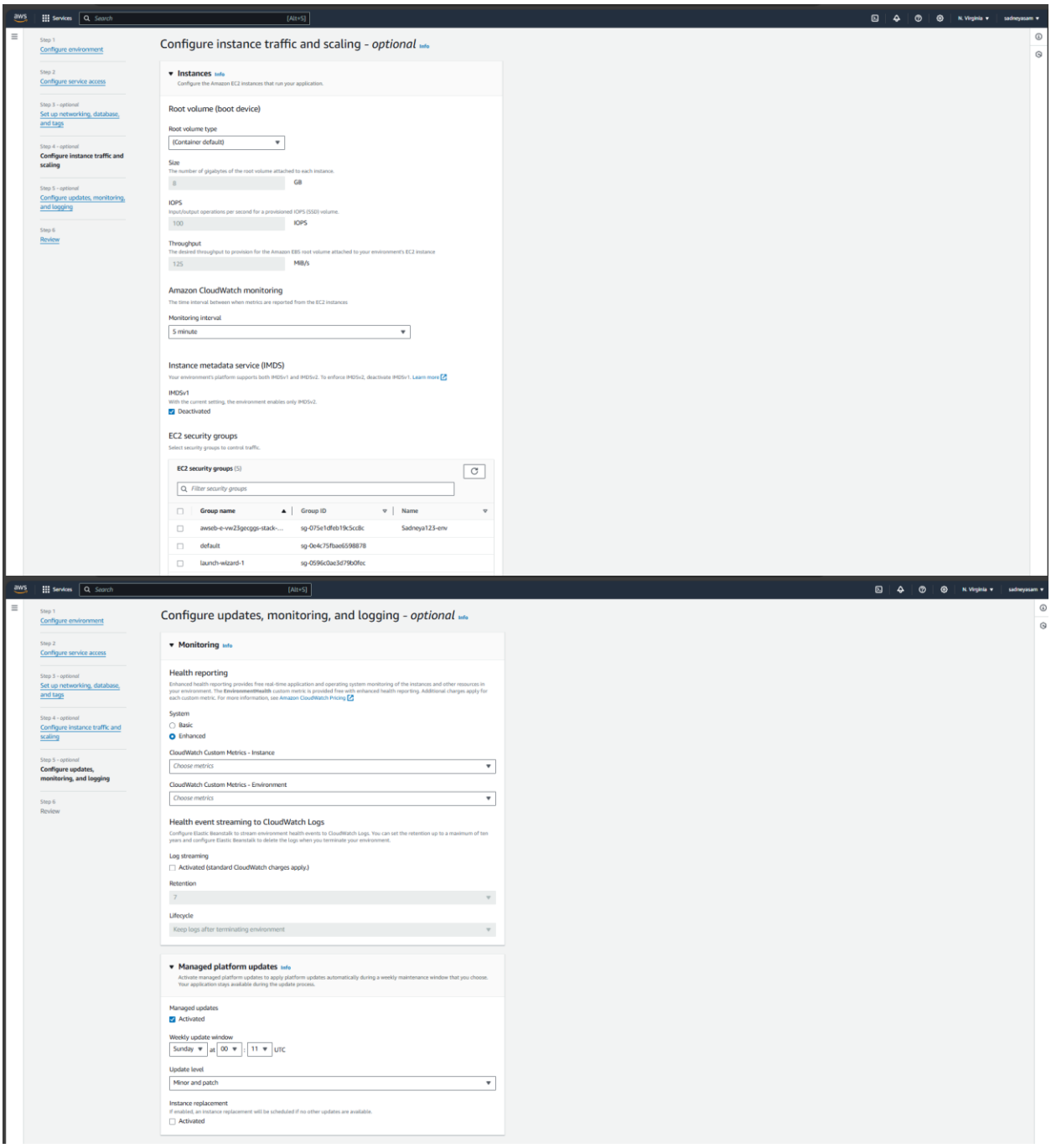


4. After clicking on next u need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created

in 1st part.Select role,then select key you have created then profile will be automatically selected according to role.then click on create application by keeping all the remaining settings as it is.

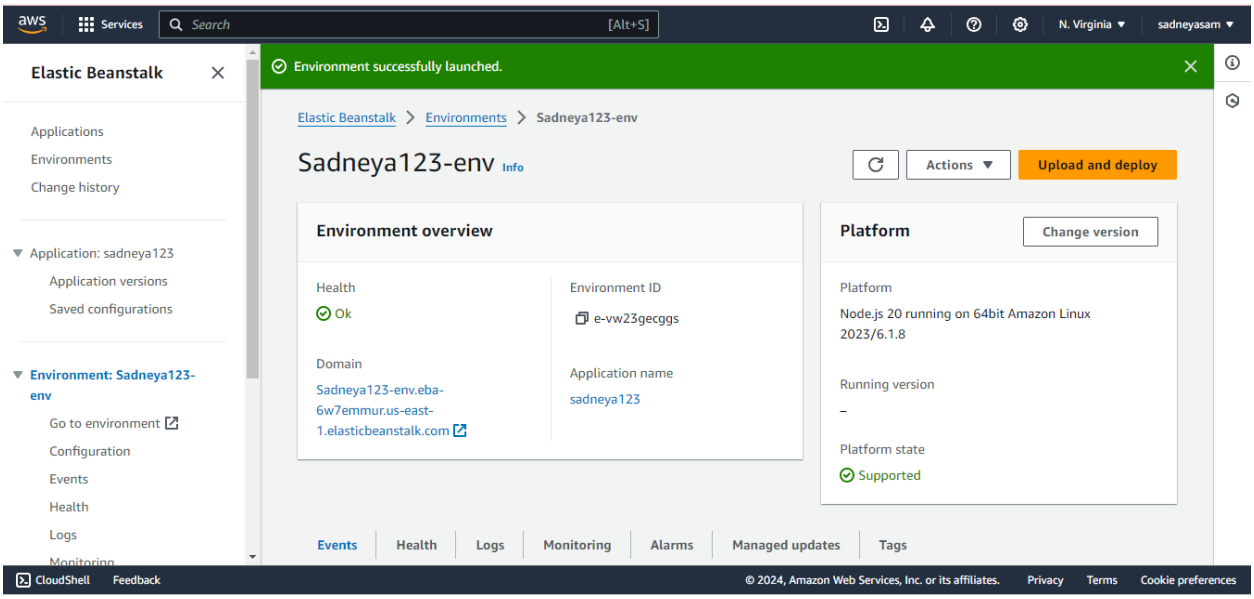




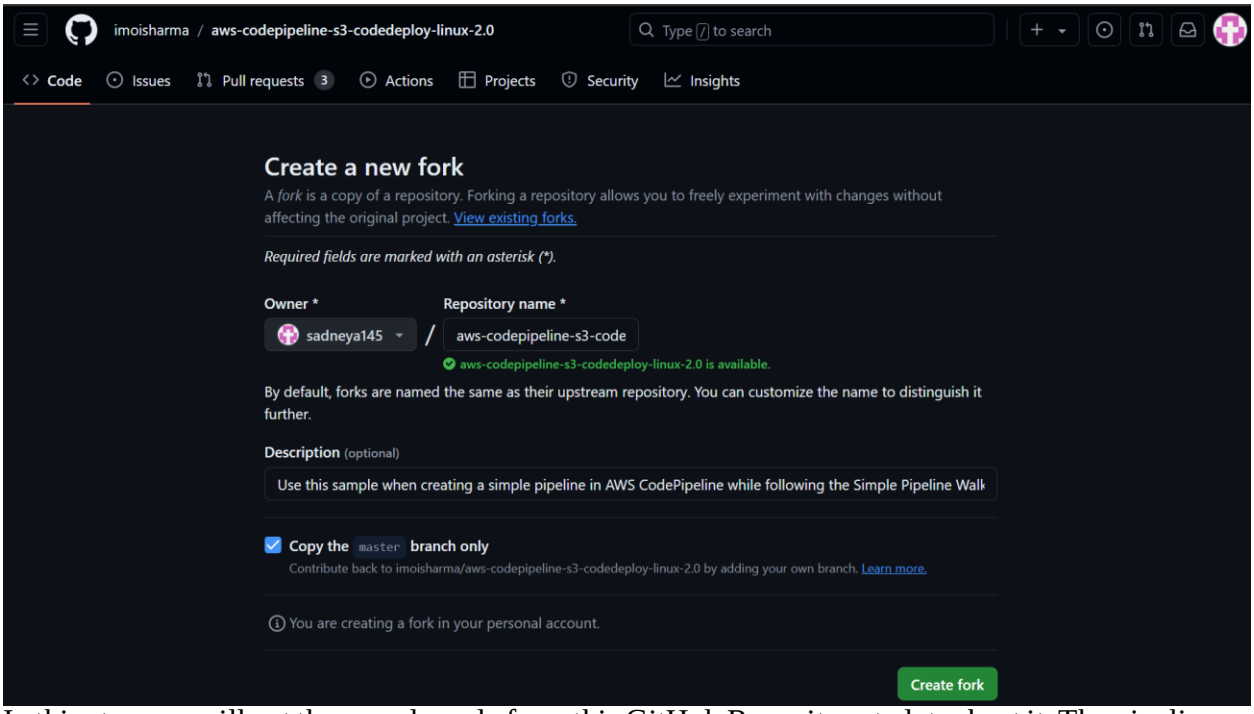


Keep Set up networking, database, and tags ,Configure instance traffic and scaling,Configure updates, monitoring, and logging all these default.

5. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.



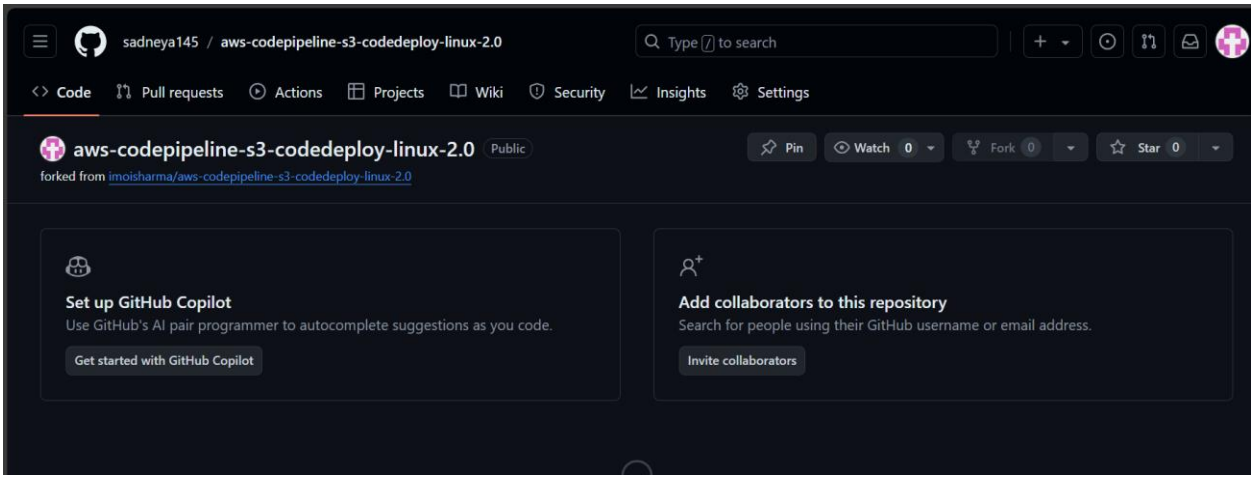
Step 3: Get a copy of your sample code



In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

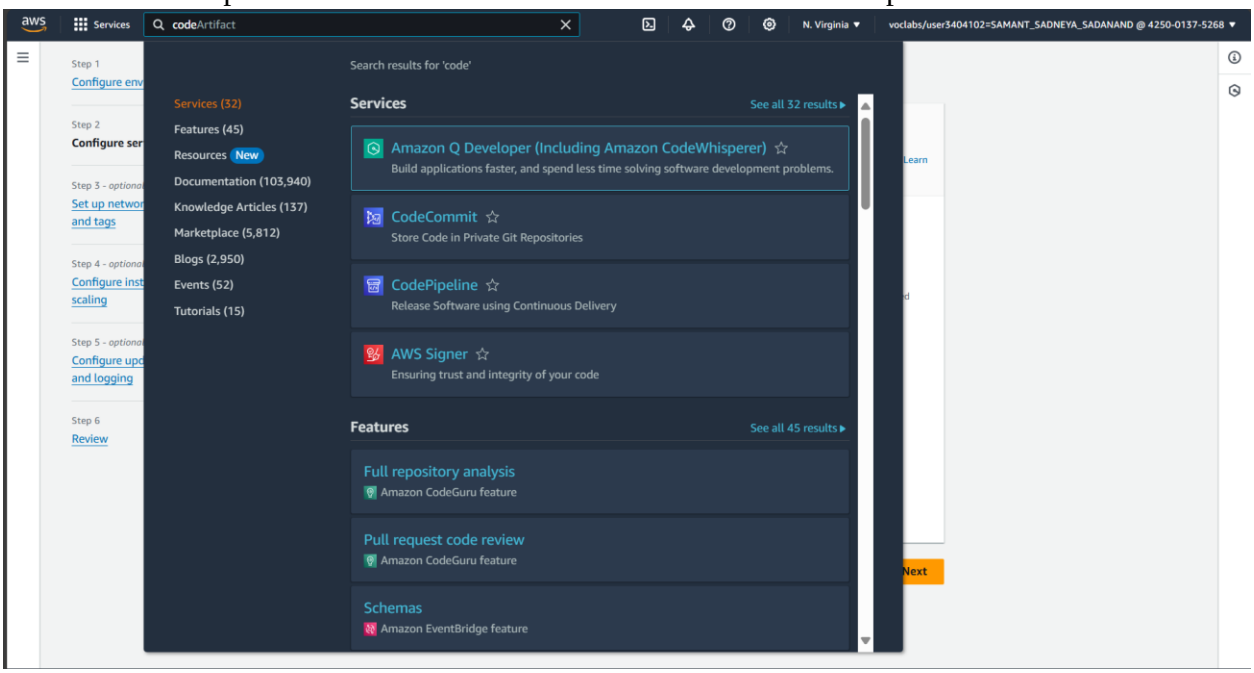
Go to the repository shared above and simply fork it.

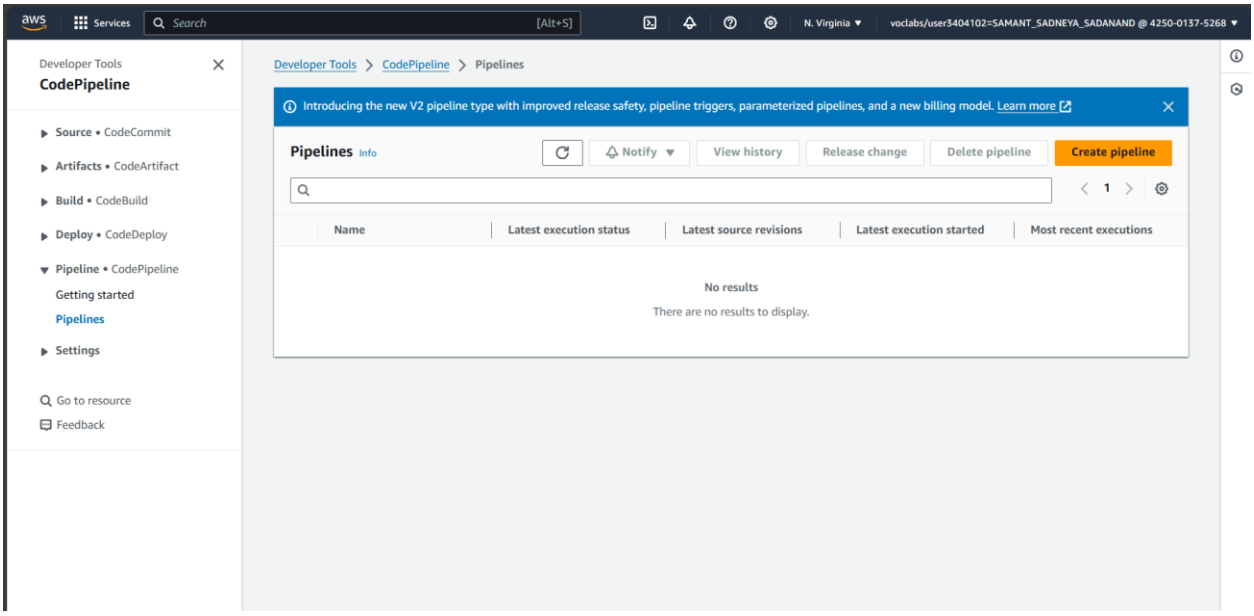


Step 4: Creating a CodePipeline

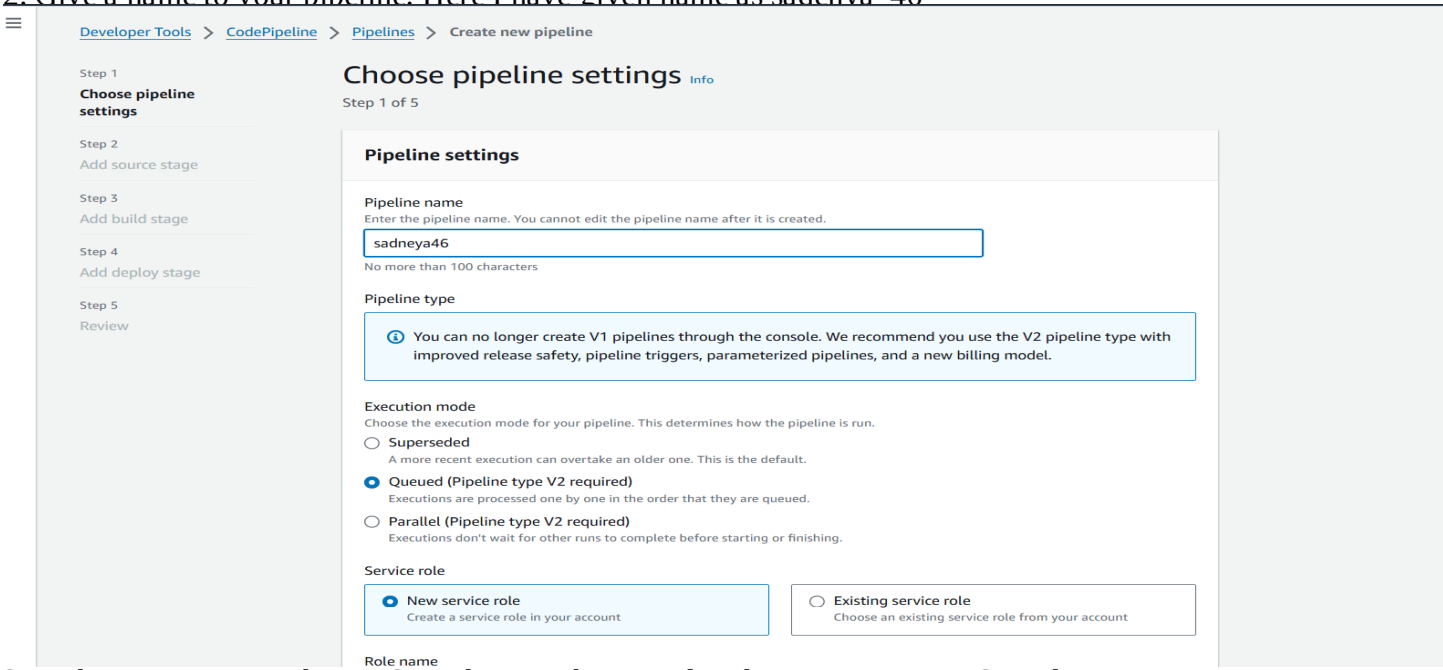
In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.

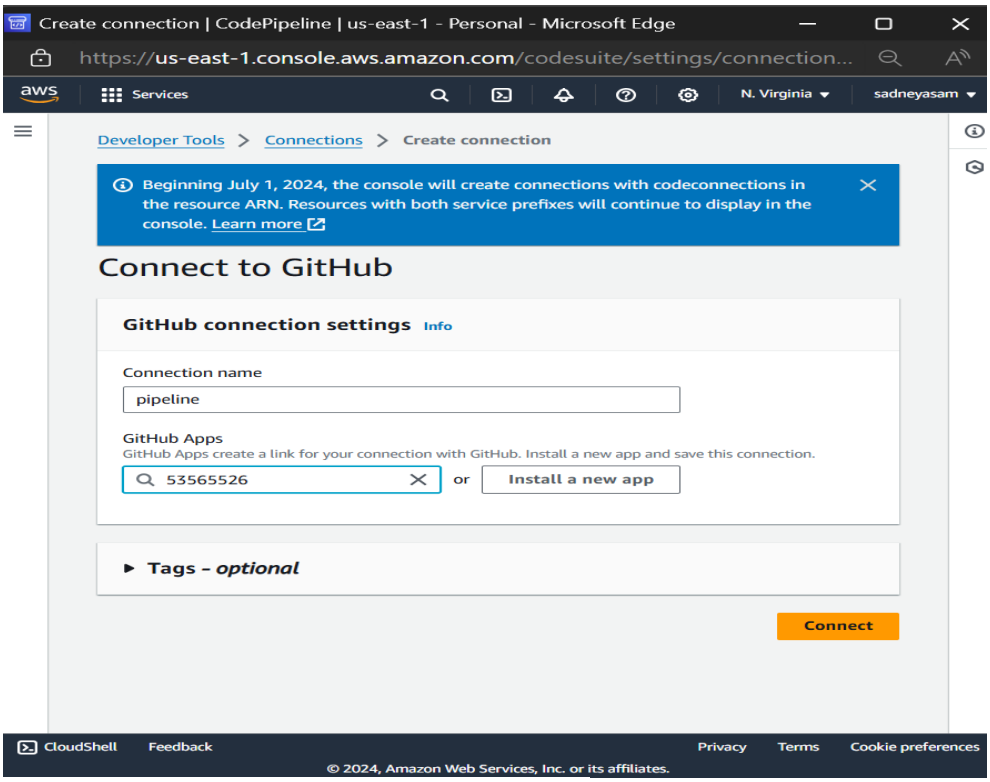


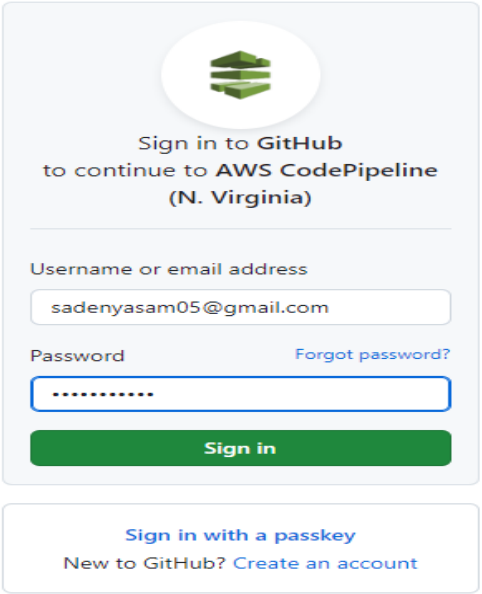


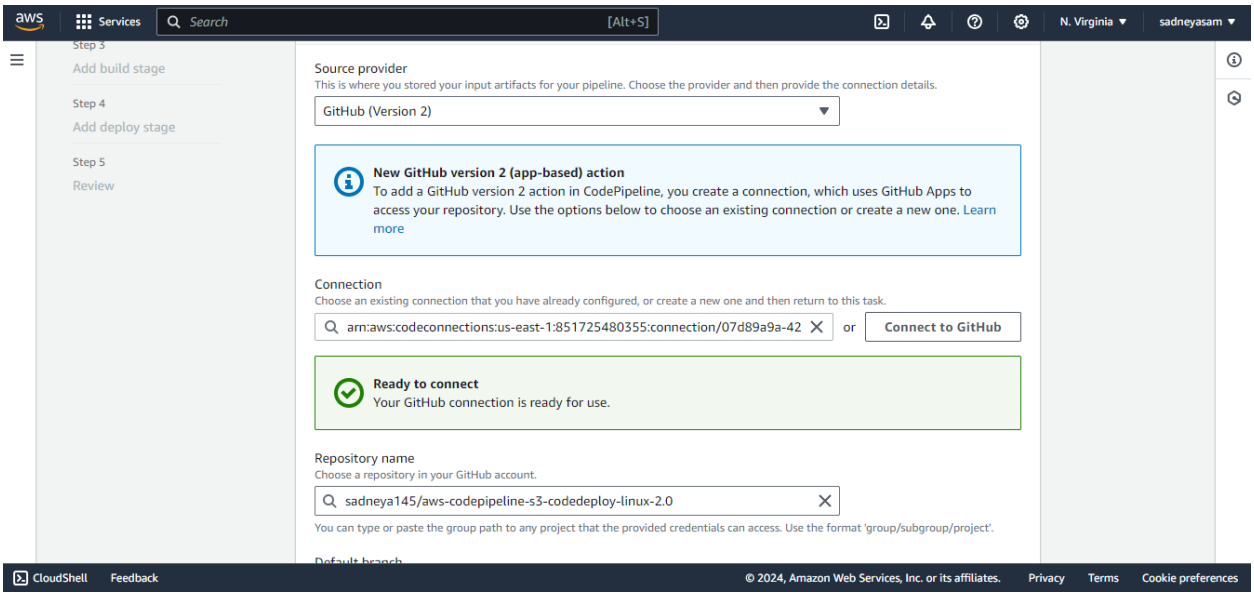
2. Give a name to your pipeline. Here I have given name as sadenya_46



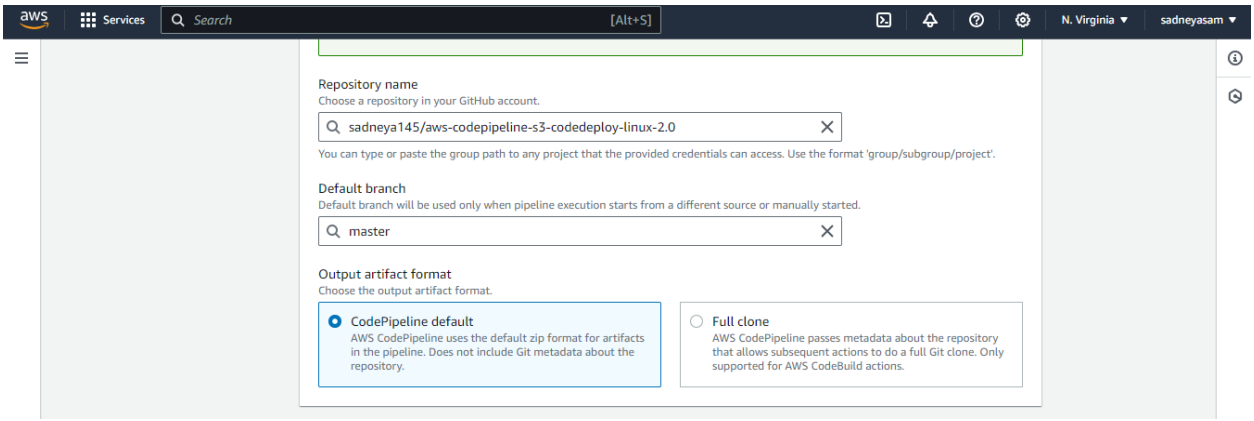
3. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.


Create connection | CodePipeline | us-east-1 - Personal - Microsoft Edge
https://us-east-1.console.aws.amazon.com/codesuite/settings/connection...
aws
Services
N. Virginia
sadneyasam
Developer Tools > Connections > Create connection
Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. Learn more
Connect to GitHub
GitHub connection settings Info
Connection name
pipeline
GitHub Apps
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.
53565526
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.

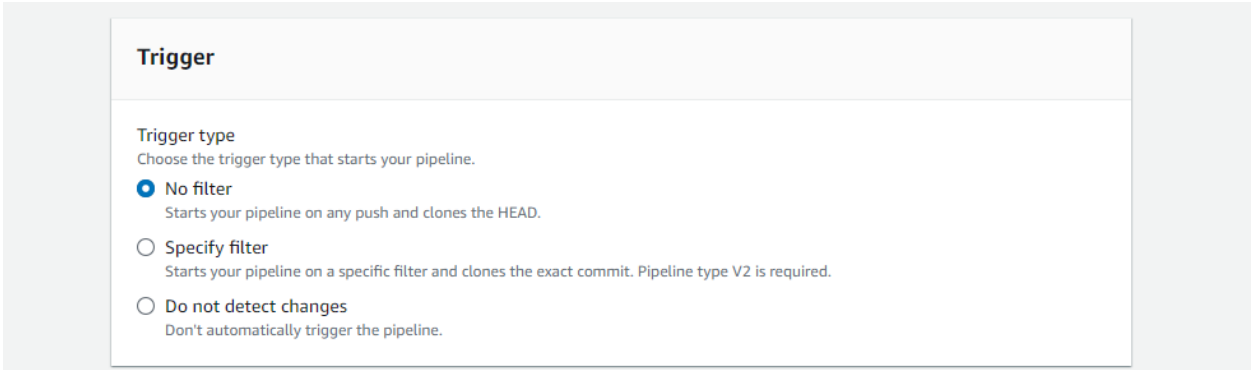

Sign in to GitHub
to continue to AWS CodePipeline
(N. Virginia)
Username or email address
sadenyasam05@gmail.com
Password
Forgot password?
Sign in
Sign in with a passkey
New to GitHub? Create an account



4. select the forked repository then select the master branch.



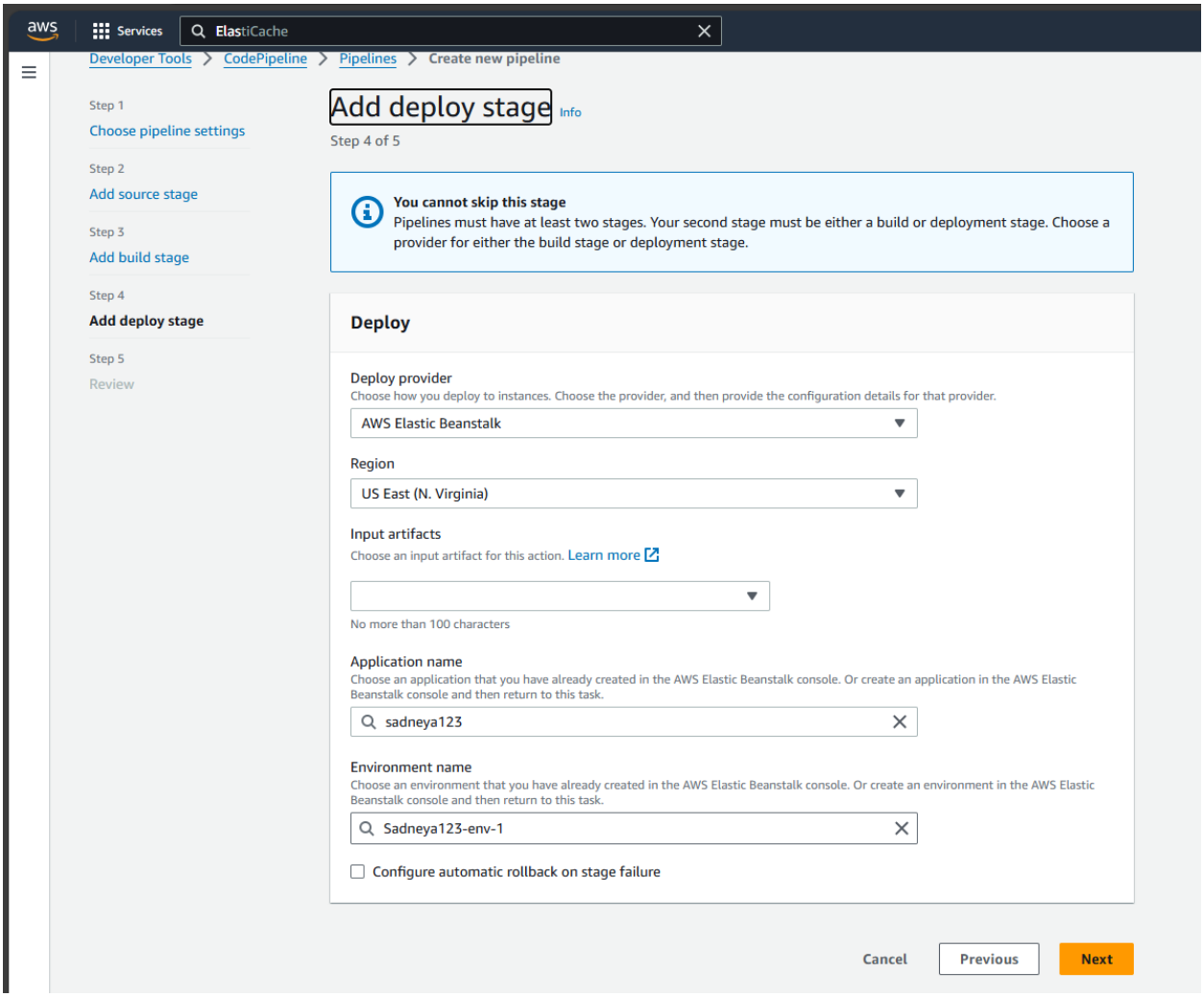
5. Then select trigger type none.



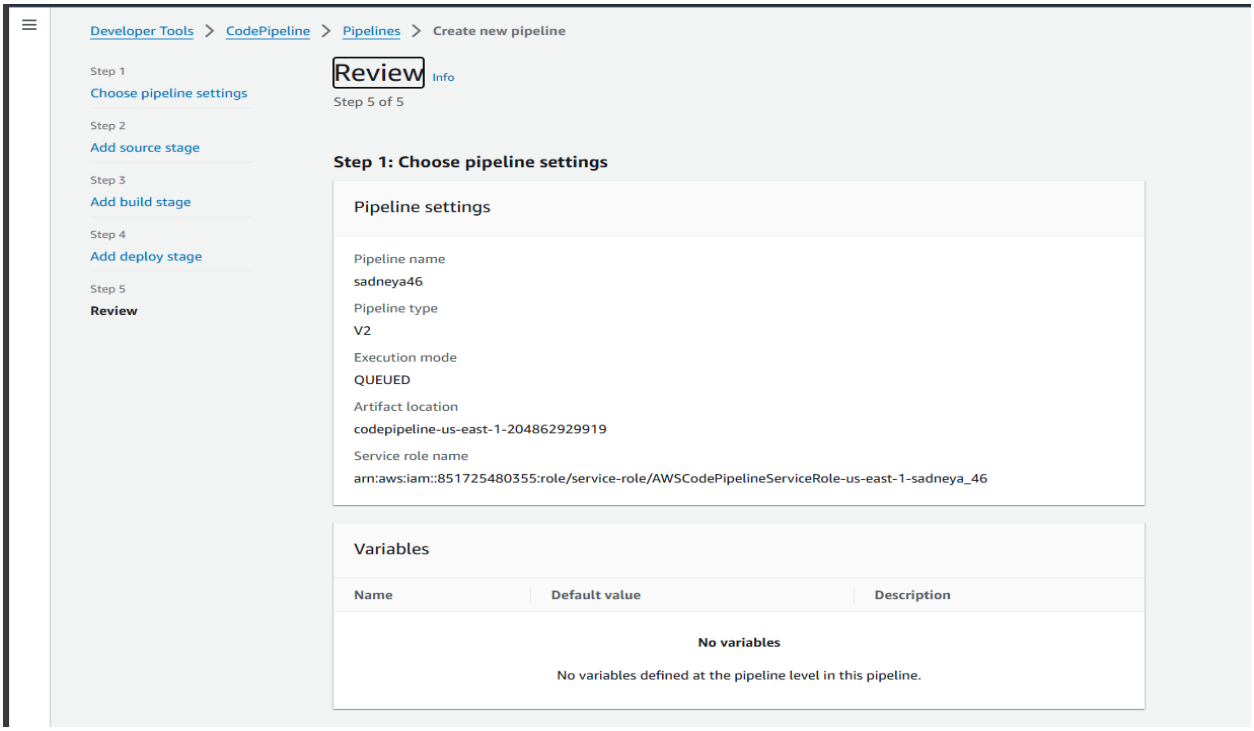
After that, click Continue and skip the build stage. Proceed to the Deployment stage.

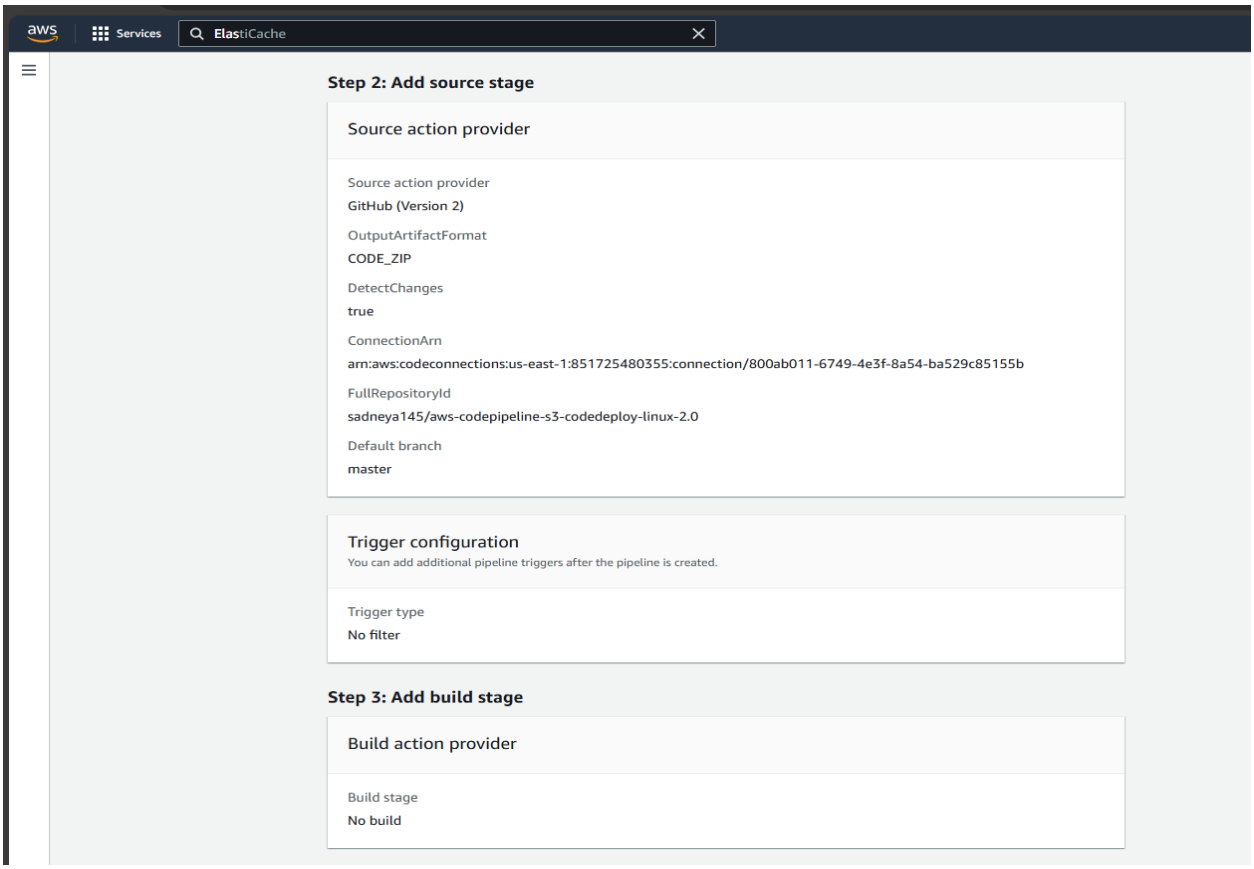
Step 5: Deployment

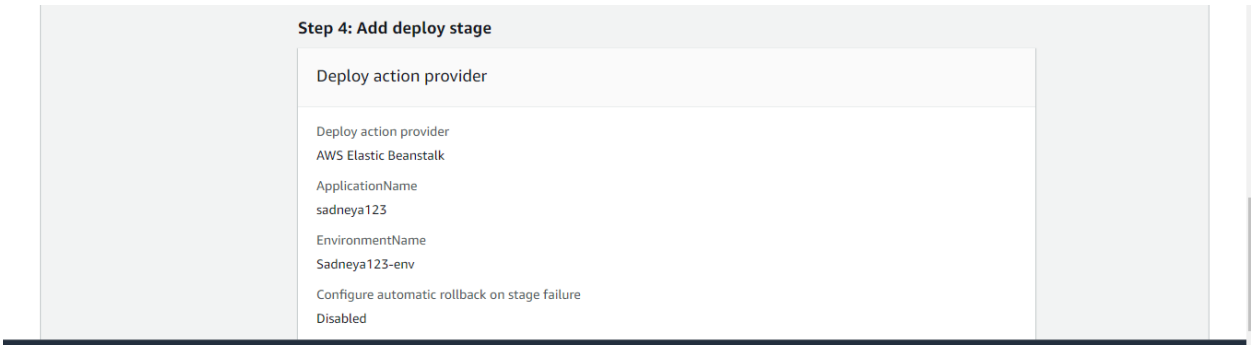
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.



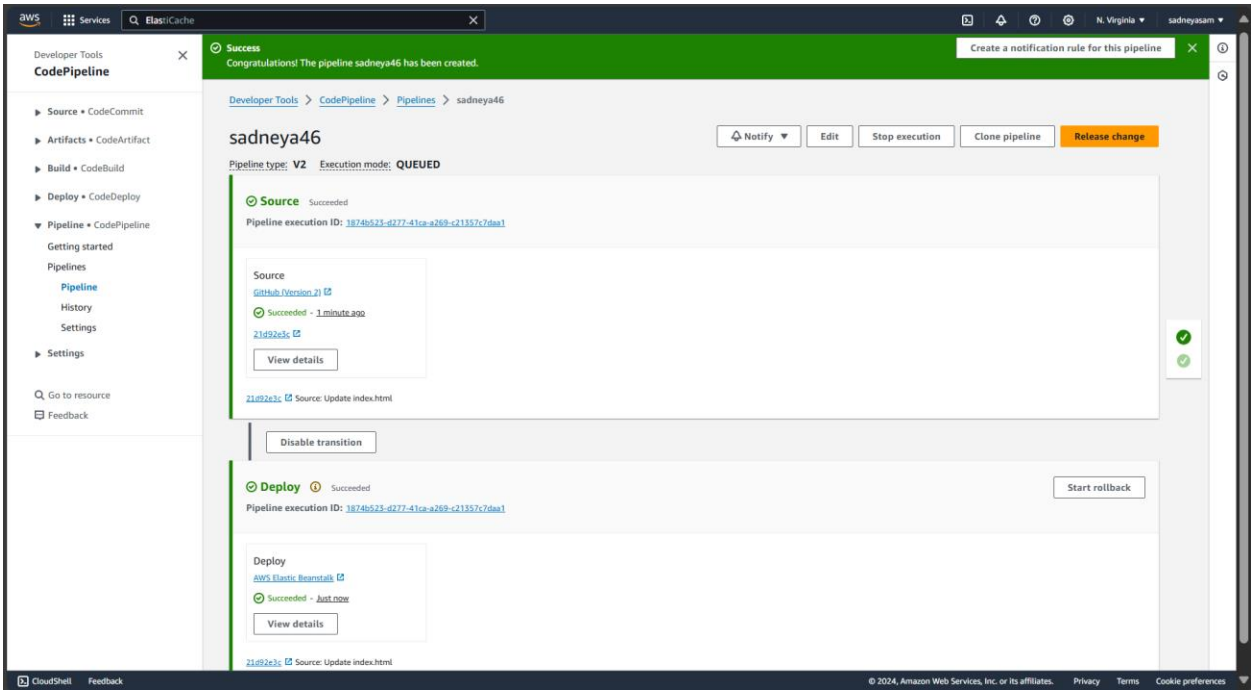
2.Click Next, Review and create the pipeline.



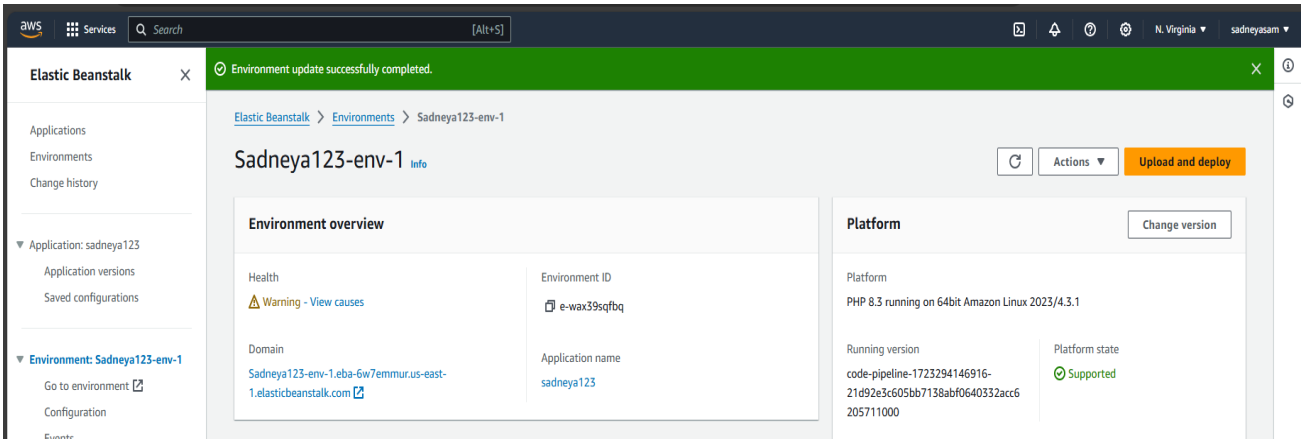


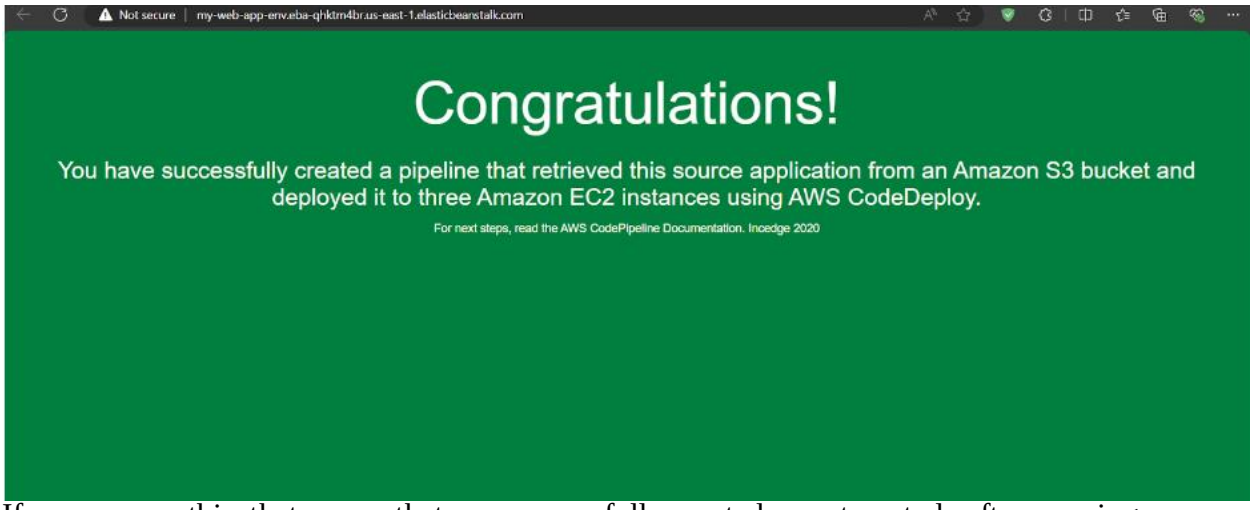


3. Then it will give you this result on screen. i.e. deployed successfully.



4. In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Benstalk.

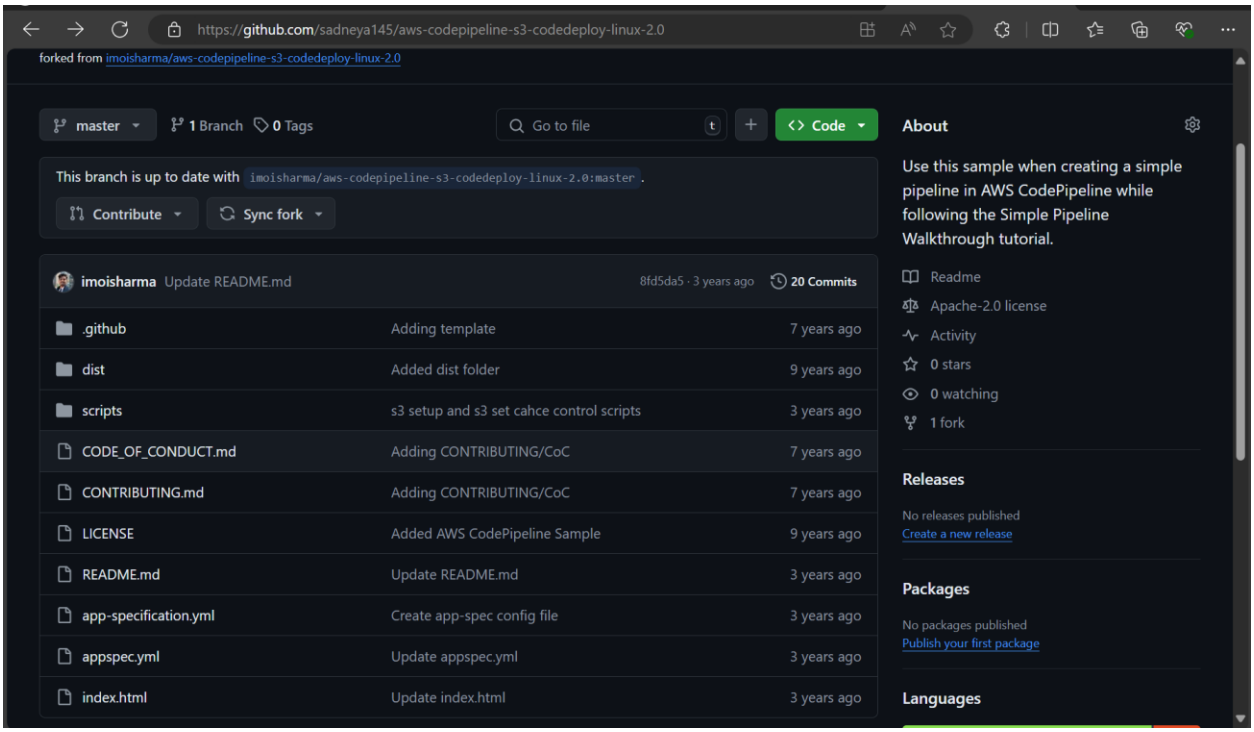


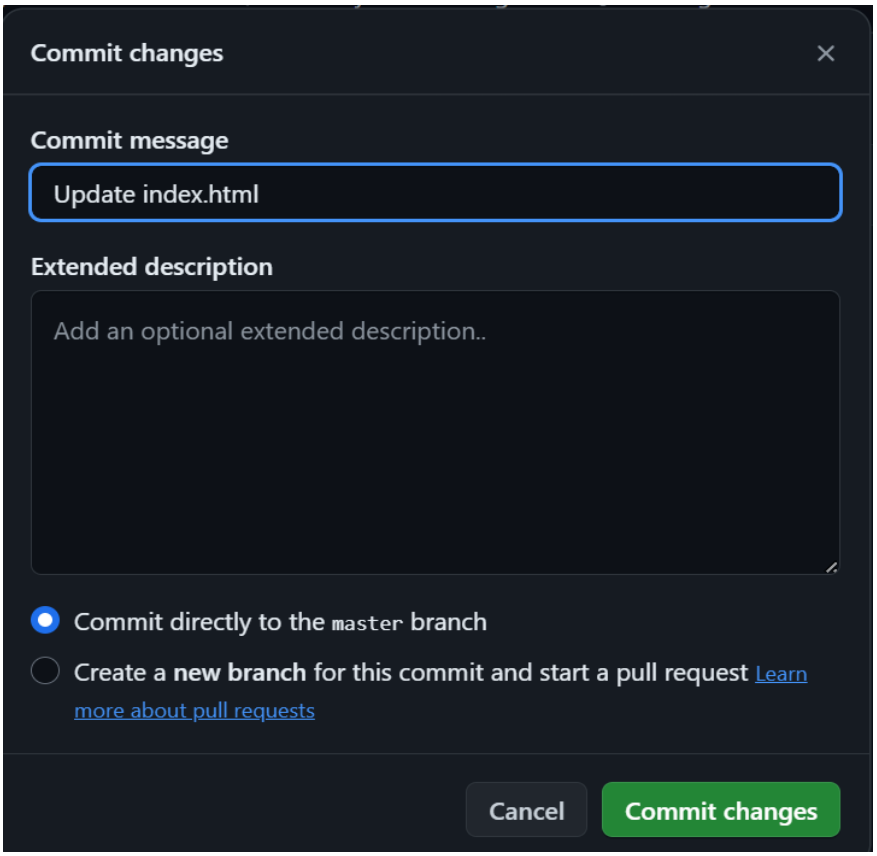


If you can see this, that means that you successfully created an automated software using CodePipeline.

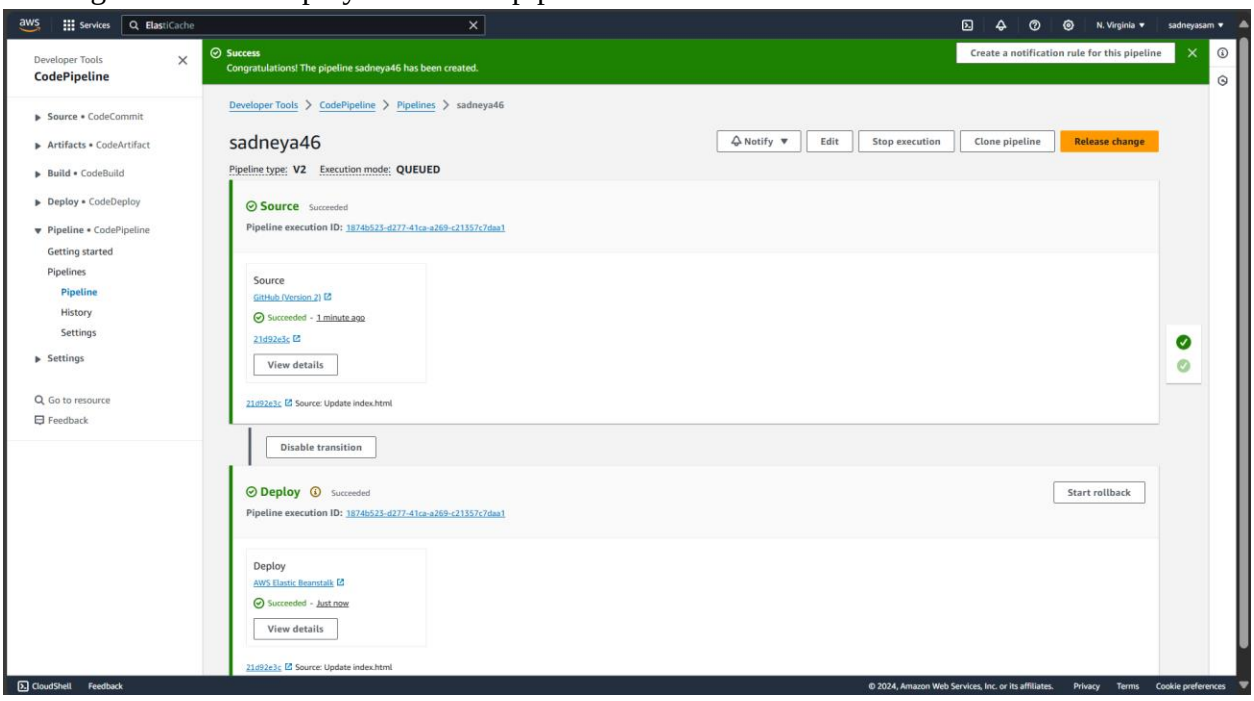
Step 6: Committing changes to update app

1. In this we make some changes in the file.Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.





2. Then again start the deployment of the pipeline.



3. Check the changes in the website , here I have added a message in h3 tag.

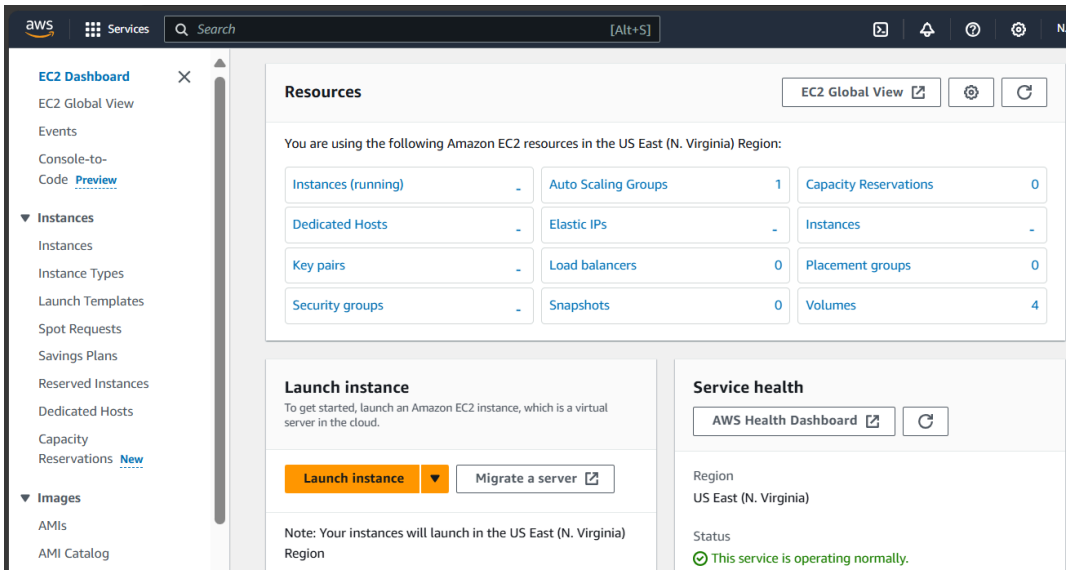

Not secure | sadneya123-env-1.eba-6w7emmur.us-east-1.elasticbeanstalk.com
Congratulations!
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.
Hello It's me D15C-46
For next steps, read the AWS CodePipeline Documentation. Incedge 2020

EXPERIMENT NO. 3

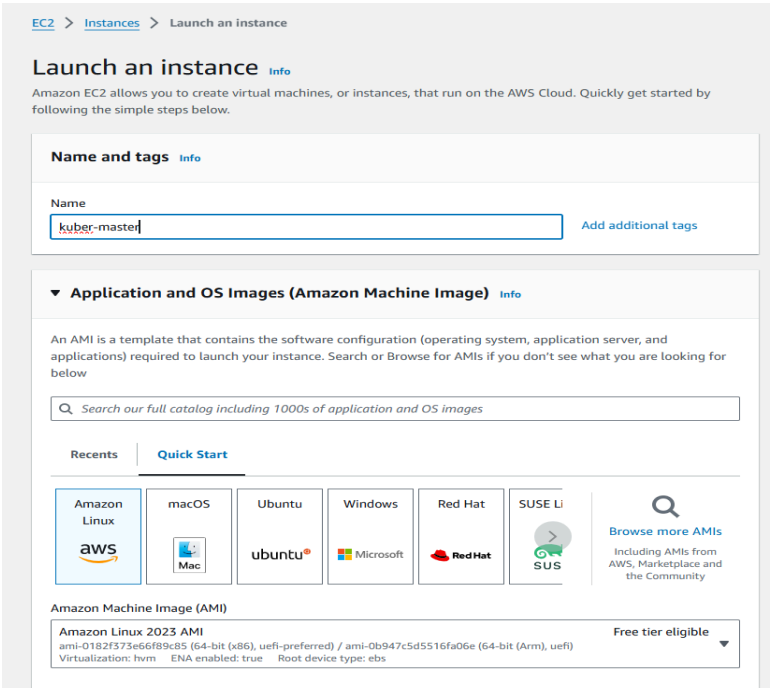
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud.

Procedure:

1. **Creation Of Instance**

- 

Search EC-2 instance. Then create three EC-2 instances and choose Amazon Linux as OS and also allow ssh traffic from anywhere.



Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0182f373e66f89c85

Verified provider

Instance type Info | Get advice

Instance type

t3.medium
Family: t3 2 vCPU 4 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0979 USD per Hour
On-Demand Windows base pricing: 0.06 USD per Hour
On-Demand Linux base pricing: 0.0416 USD per Hour
On-Demand RHEL base pricing: 0.0704 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

server

Create new key pair

Network settings Info

Edit

Network Info

vpc-051bba342b3626898

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group | Select existing security group

We'll create a new security group called '**launch-wizard-31**' with the following rules:

Allow SSH traffic from
Helps you connect to your instance
Anywhere 0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-0182f373e66f89c85

Virtual server type (instance type)
t3.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

ⓘ **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

To efficiently run kubernetes cluster select instance type of at least t3.medium as kubernetes recommends at least 2 vCPU to run smoothly on it.

Name	Instance ID	Instance state
kube-master	i-00aa79ac09d7462c0	Running
kube-worker1	i-0bab86cd3fbfcb40a	Running
kube-worker2	i-00dcfd302ffd80dda	Running

- Then for making connection through SSH into all 3 machines each in separate terminal Use this following command:
  ssh -i <keyname>.pem ubuntu@<public_ip_address> where keyname is name of the key you created here i created key server.pem and use public IP address.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 “key name.pem”. Then use the given command for making connections).

```
Sadneya@DESKTOP-IEPNL3D MINGW64 ~ (master)
$ cd downloads

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ chmod 400 "server.pem"

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ ssh -i "server.pem" ec2-user@ec2-54-174-206-93.compute-1.amazonaws.com
The authenticity of host 'ec2-54-174-206-93.compute-1.amazonaws.com (54.174.206.
93)' can't be established.
ED25519 key fingerprint is SHA256:T+tsGyI15gAvUvjeAZ7GjDIWXHOaI4EPF5g5oICrkoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-174-206-93.compute-1.amazonaws.com' (ED25519)
 to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

2. Installation Of Docker on three machines

- For installation of Docker into all three machines run the following command:
  sudo yum install docker -y

```
[ec2-user@ip-172-31-87-114 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:06:20 ago on Fri Sep 13 03:20:22 2024.
Dependencies resolved.
================================================================================
 Package                 Arch     Version                  Repository     Size
================================================================================
Installing:
 docker                  x86_64   25.0.6-1.amzn2023.0.2    amazonlinux    44 M
Installing dependencies:
 containerd              x86_64   1.7.20-1.amzn2023.0.1    amazonlinux    35 M
 iptables-libs           x86_64   1.8.8-3.amzn2023.0.2     amazonlinux   401 k
 iptables-nft            x86_64   1.8.8-3.amzn2023.0.2     amazonlinux   183 k
 libcgroup               x86_64   3.0-1.amzn2023.0.1       amazonlinux    75 k
 libnetfilter_conntrack  x86_64   1.0.8-2.amzn2023.0.2     amazonlinux    58 k
 libnfnetlink            x86_64   1.0.1-19.amzn2023.0.2    amazonlinux    30 k
 libnftnl                x86_64   1.2.2-2.amzn2023.0.2     amazonlinux    84 k
 pigz                    x86_64   2.5-1.amzn2023.0.3       amazonlinux    83 k
 runc                    x86_64   1.1.13-1.amzn2023.0.1    amazonlinux   3.2 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_ 3.6 MB/s | 401 kB     00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_6 4.6 MB/s | 183 kB     00:00
```

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
```

- Then, configure cgroup in a daemon.json file by using following commands
  cd /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
  "max-size": "100m"
  },

"storage-driver": "overlay2"
}
EOF

```
[ec2-user@ip-172-31-87-114 ~]$ cd /etc/docker
[ec2-user@ip-172-31-87-114 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json

{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.
  sudo systemctl enable docker
  sudo systemctl daemon-reload
  sudo systemctl restart docker

```
[ec2-user@ip-172-31-80-126 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

- Then check the version of docker installed.
  docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Installation Of Kubernetes on three machines

- SELinux needs to be disable before configuring kubelet thus run the following command
  sudo setenforce 0
  sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-80-126 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- Here We are adding kubernetes using the repository whose command is given below.
  cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
  [kubernetes]
  name=Kubernetes
  baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
  enabled=1
  gpgcheck=1
  gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
  exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
  EOF

```
[ec2-user@ip-172-31-80-126 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl: sudo yum update

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
Kubernetes
Dependencies resolved.
Nothing to do.
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                 Repository    Size
================================================================================
Installing:
 kubeadm                 x86_64        1.30.5-150500.1.1       kubernetes    10 M
 kubectl                 x86_64        1.30.5-150500.1.1       kubernetes    10 M
 kubelet                 x86_64        1.30.5-150500.1.1       kubernetes    17 M
Installing dependencies:
 conntrack-tools         x86_64        1.4.6-2.amzn2023.0.2    amazonlinux  208 k
 cri-tools               x86_64        1.30.1-150500.1.1       kubernetes   8.6 M
 kubernetes-cni          x86_64        1.4.0-150500.1.1        kubernetes   6.7 M
 libnetfilter_cthelper   x86_64        1.0.0-21.amzn2023.0.2   amazonlinux   24 k
 libnetfilter_cttimeout  x86_64        1.0.0-19.amzn2023.0.2   amazonlinux   24 k
 libnetfilter_queue      x86_64        1.0.5-2.amzn2023.0.2    amazonlinux   30 k

Transaction Summary
================================================================================
Install  9 Packages
```

```
Total                                                          64 MB/s |  53 MB     00:00
Kubernetes                                                     20 kB/s | 1.7 kB     00:00
Importing GPG key 0x9A296436:
 Userid     : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
 Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
 From       : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : kubernetes-cni-1.4.0-150500.1.1.x86_64                 1/9
  Installing       : cri-tools-1.30.1-150500.1.1.x86_64                     2/9
  Installing       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64         3/9
  Installing       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64    4/9
  Installing       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64     5/9
  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-1.30.5-150500.1.1.x86_64                       7/9
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64                       7/9
  Installing       : kubeadm-1.30.5-150500.1.1.x86_64                       8/9
  Installing       : kubectl-1.30.5-150500.1.1.x86_64                       9/9
  Running scriptlet: kubectl-1.30.5-150500.1.1.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-1.30.1-150500.1.1.x86_64                     5/9
  Verifying        : kubeadm-1.30.5-150500.1.1.x86_64                       6/9
  Verifying        : kubectl-1.30.5-150500.1.1.x86_64                       7/9
  Verifying        : kubelet-1.30.5-150500.1.1.x86_64                       8/9
  Verifying        : kubernetes-cni-1.4.0-150500.1.1.x86_64                 9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64  cri-tools-1.30.1-150500.1.1.x86_64  kubeadm-1.30.5-150500.1.1.x86_64
  kubectl-1.30.5-150500.1.1.x86_64             kubelet-1.30.5-150500.1.1.x86_64    kubernetes-cni-1.4.0-150500.1.1.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!
```

- After installing Kubernetes, we need to configure internet options to allow bridging.
    1. sudo swapoff -a
    2. echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
    3. sudo sysctl -p

```
[ec2-user@ip-172-31-80-126 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

4. **Perform this ONLY on the Master machine**

- Initialize kubernetes by typing below command

  sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0913 10:32:44.629146   26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
```

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk \
        --discovery-token-ca-cert-hash sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a
```

- So after initialization you will get token at the end for joining master and worker. Like here I got this :(save this token as it is required later.Then you can join any number of worker nodes by running the following on each as root.)

  kubeadm join 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk\
  --discovery-token-ca-cert-hash
  sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a

- Also,Copy the mkdir and chown commands from the top and execute them
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
[ec2-user@ip-172-31-80-126 docker]$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- Then, add a common networking plugin called flammel file as mentioned in the code.
  kubectl apply -f
  https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

```
[ec2-user@ip-172-31-80-126 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

- Now to Check the created pod use this command
  kubectl get pods

5. **Perform this ONLY on the worker machines**

Paste the below command on all 2 worker machines

- sudo yum install iproute-tc -y
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

Now use this
kubeadm join 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk\
--discovery-token-ca-cert-hash
sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a

(Optional To check the status of pods executed these commands:

Kubectl get pods -n kube-system :gives status of all pods

Kubectl get daemonstat -n kube-system: gives status of pod named daemonstat

```
[ec2-user@ip-172-31-87-114 docker]$ kubectl get pods -n kube-system
NAME                                                READY   STATUS             RESTARTS        AGE
coredns-55cb58b774-fxl2f                            1/1     Running            0               100s
coredns-55cb58b774-xnl4v                            1/1     Running            0               100s
etcd-ip-172-31-87-114.ec2.internal                  1/1     Running            1 (2m45s ago)   75s
kube-apiserver-ip-172-31-87-114.ec2.internal        1/1     Running            1 (2m15s ago)   2m11s
kube-controller-manager-ip-172-31-87-114.ec2.internal   0/1     CrashLoopBackOff   1 (8s ago)      70s
kube-proxy-4dv8m                                    1/1     Running            2 (26s ago)     100s
kube-scheduler-ip-172-31-87-114.ec2.internal        1/1     Running            1 (2m45s ago)   76s
[ec2-user@ip-172-31-87-114 docker]$ kubectl get daemonset -n kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   3m
```

)

Now to see whether master and workers get connected successfully or not run **kubectl get nodes** command on master machine

```
[ec2-user@ip-172-31-87-114 docker]$ kubectl get nodes
NAME                           STATUS   ROLES           AGE     VERSION
ip-172-31-87-114.ec2.internal   Ready    control-plane   3m21s   v1.30.5
```

Conclusion: In these EC-2 instance created successfully on AWS Linux. Then I installed docker ,kuberneted and then kubelet ,kubeadm, kubectl.Then on Master machine ,I initailized the kubernetes which given me the token which will be used for connection of master and workers.then on slave I installed iproute and enabled and restarted kubelet then i enter the token which i got from master but there was an issue in joint.that is why on output i just got of only one pc mater on performing command kubectl get nodes.

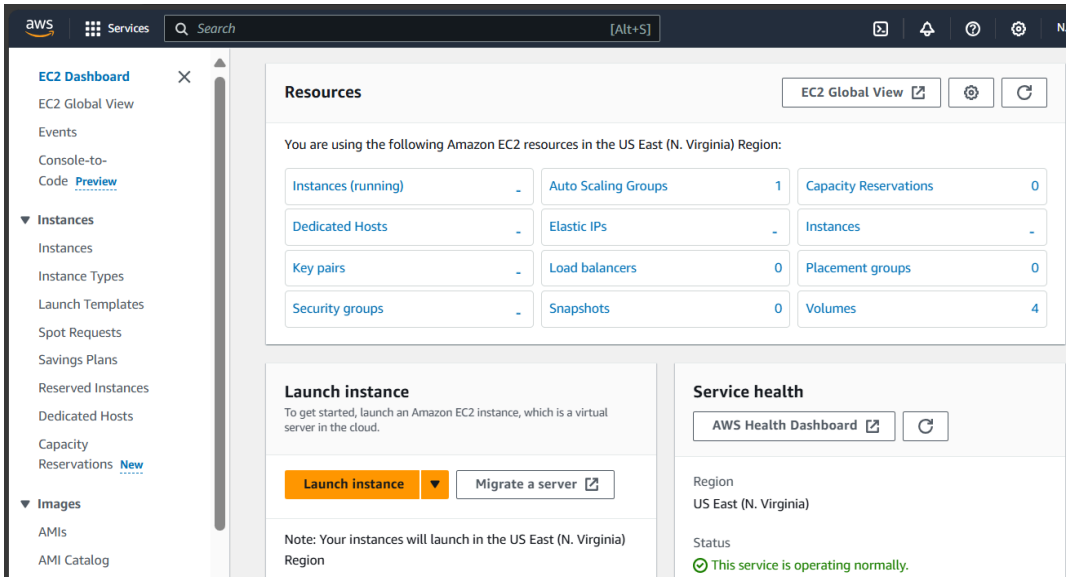
Experiment 4

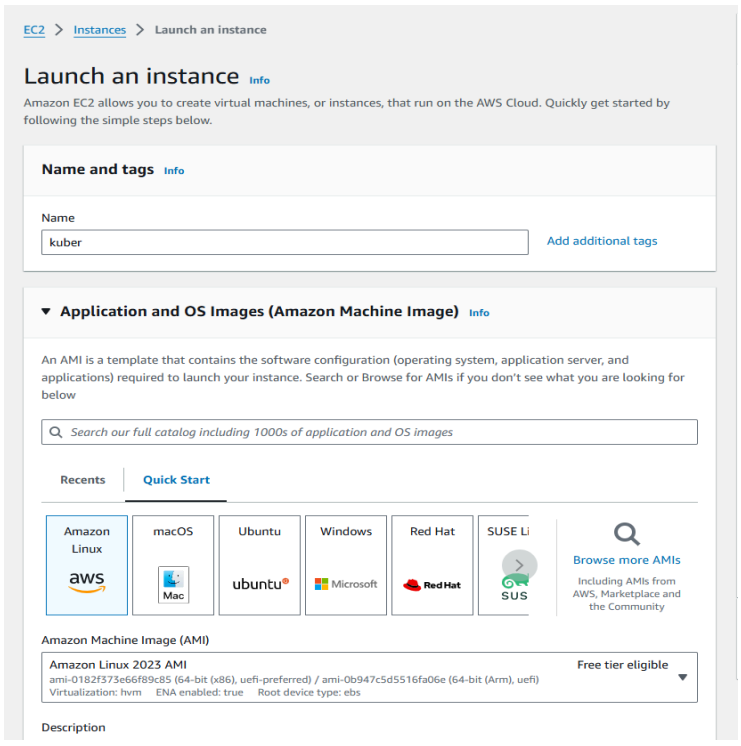
Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

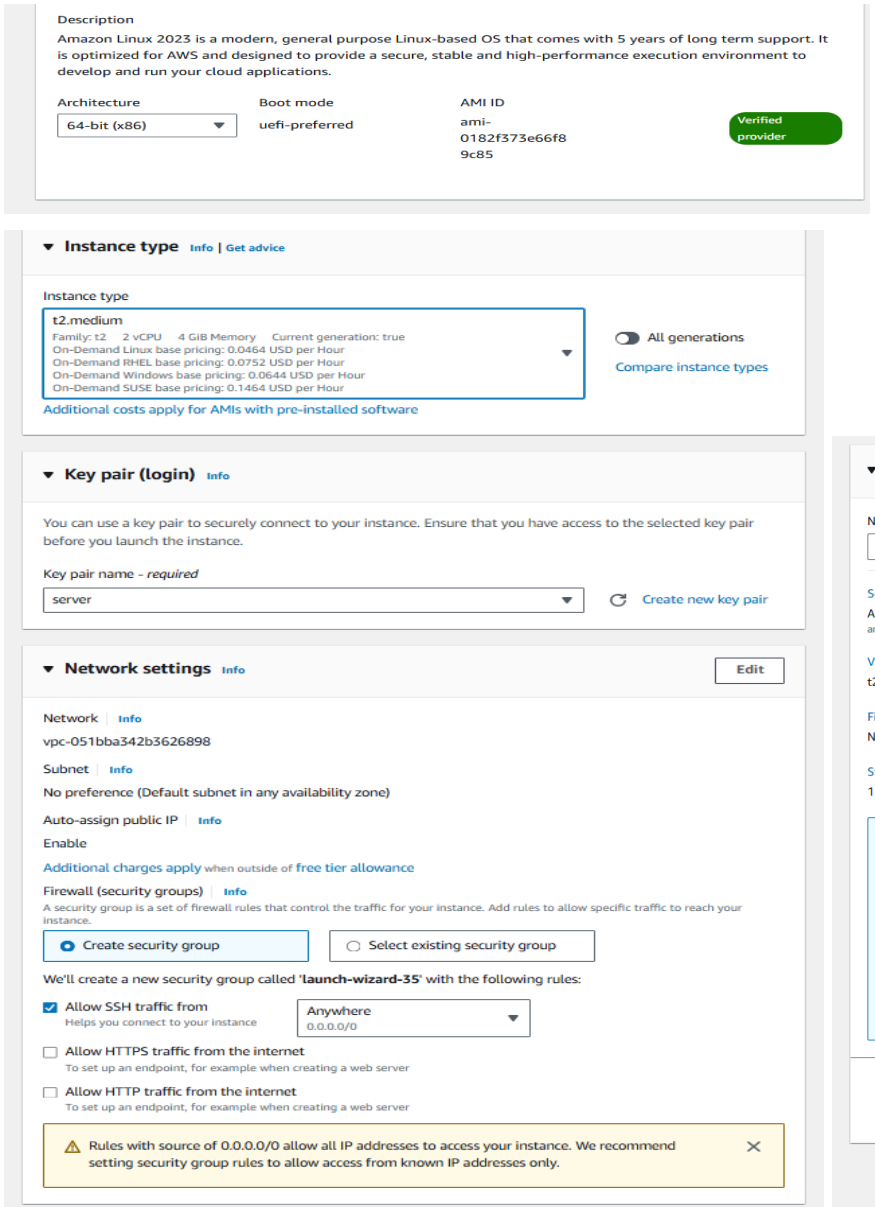
Procedure:

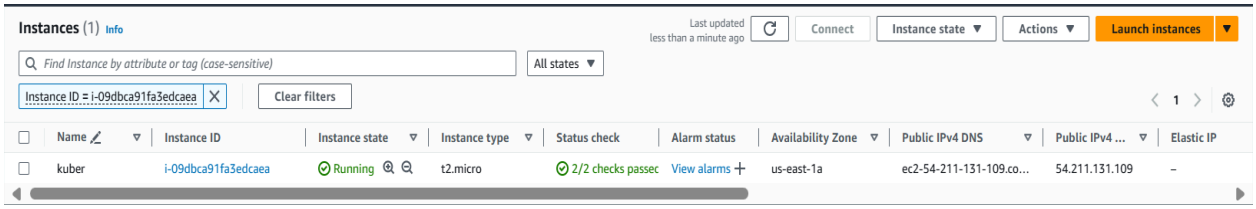
1. **Creation Of EC-2 instance**

- Create an EC2 AWS Linux instance on AWS .also edit the Security Group Inbound Rules to allow SSH. then select the t2.micro instance type

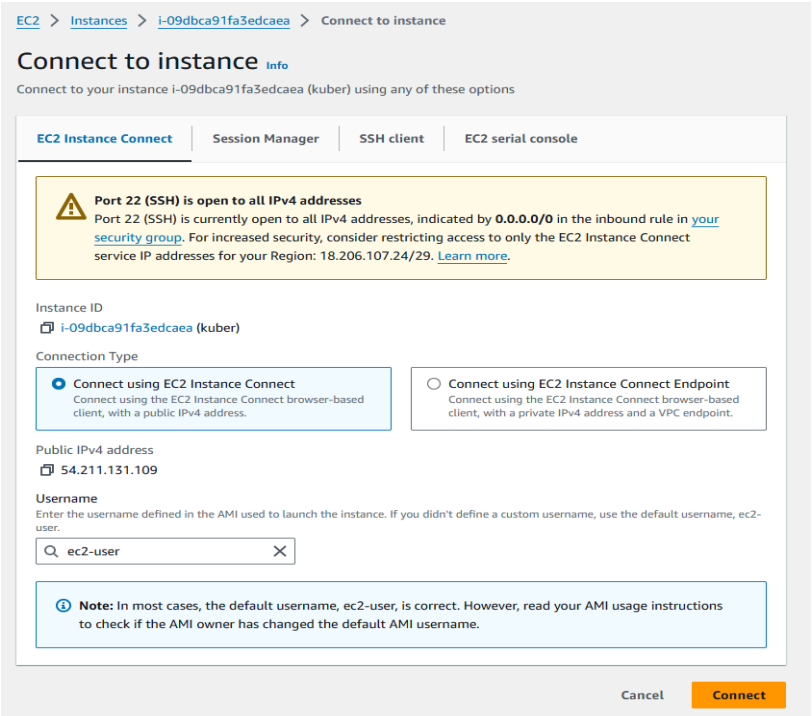








- Thus Kuber named -instance gets created.Then click on Id of that instance then click on connect button you will se this:

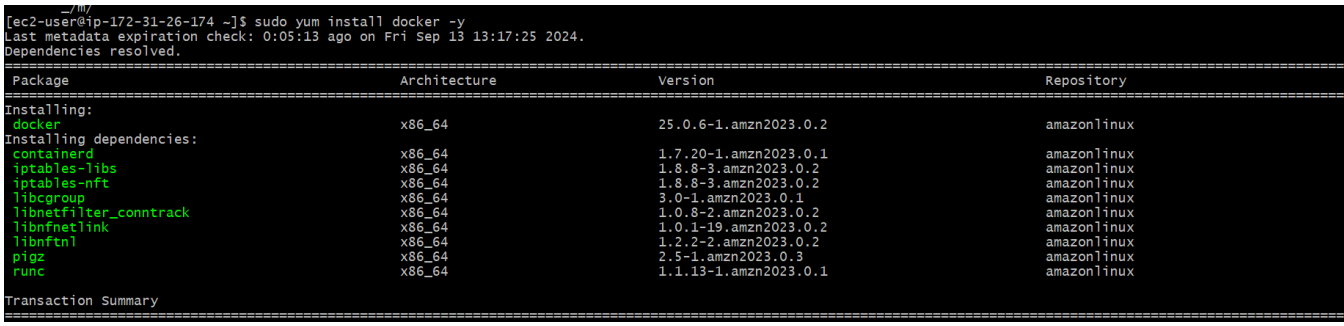


- Then go into SSH client where you will get this command
  Chmod 400 “keyname.pem”
  ssh -i <keyname>.pem ubuntu@<public_ip_address> copy it and then connect it and run the following command for establishing connection.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 “key name.pem”. Then use the given command for making connections).

```
Sadneya@DESKTOP-IEPNL3D MINGW64 ~ (master)
$ cd downloads

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ chmod 400 "server.pem"

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ ssh -i "server.pem" ec2-user@ec2-54-196-176-21.compute-1.amazonaws.com
The authenticity of host 'ec2-54-196-176-21.compute-1.amazonaws.com (54.196.176.
21)' can't be established.
ED25519 key fingerprint is SHA256:lE4CbWC3AOKdn1J+99jTmUzur4joKQmThQyRcwIJzUU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-196-176-21.compute-1.amazonaws.com' (ED25519)
 to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

2. Installation of Docker

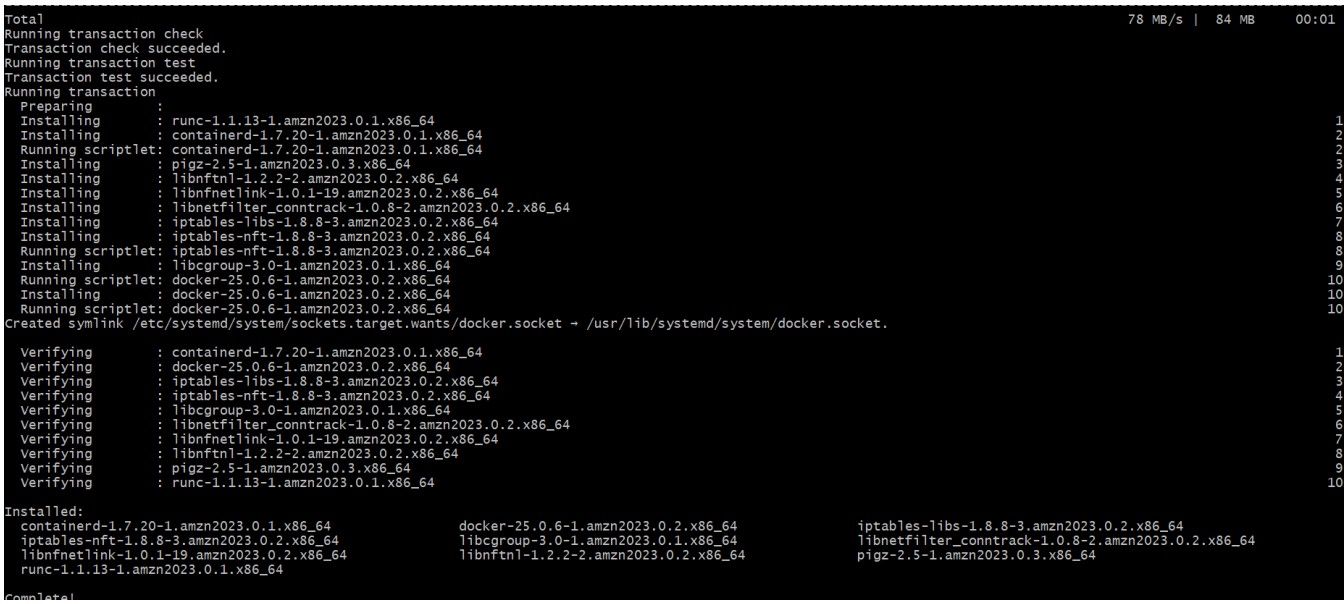
1. . For installation of Docker into the machines run the following command:
   sudo yum install docker -y



```
[ec2-user@ip-172-31-26-174 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:05:13 ago on Fri Sep 13 13:17:25 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository
Installing:			
docker	x86_64	25.0.6-1.amzn2023.0.2	amazonlinux
Installing dependencies:			
containerd	x86_64	1.7.20-1.amzn2023.0.1	amazonlinux
iptables-libs	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux
iptables-nft	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux
libcgroup	x86_64	3.0-1.amzn2023.0.1	amazonlinux
libnetfilter_conntrack	x86_64	1.0.8-2.amzn2023.0.2	amazonlinux
libnfnetlink	x86_64	1.0.1-19.amzn2023.0.2	amazonlinux
libnftnl	x86_64	1.2.2-2.amzn2023.0.2	amazonlinux
pigz	x86_64	2.5-1.amzn2023.0.3	amazonlinux
runc	x86_64	1.1.13-1.amzn2023.0.1	amazonlinux

```
Transaction Summary
```



```
Total                                                                                                78 MB/s |  84 MB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                          1
  Installing       : runc-1.1.13-1.amzn2023.0.1.x86_64                                                                        1
  Installing       : containerd-1.7.20-1.amzn2023.0.1.x86_64                                                                  2
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64                                                                  2
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                                                                           3
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                                     4
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                                5
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                                       6
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                                7
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                                 8
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                                 8
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                                      9
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                                                                      10
  Installing       : docker-25.0.6-1.amzn2023.0.2.x86_64                                                                      10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                                                                      10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.20-1.amzn2023.0.1.x86_64                                                                  1
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                                                                      2
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                                3
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                                 4
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                                      5
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                                       6
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                                7
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                                     8
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                                           9
  Verifying        : runc-1.1.13-1.amzn2023.0.1.x86_64                                                                        10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64              docker-25.0.6-1.amzn2023.0.2.x86_64              iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64             libcgroup-3.0-1.amzn2023.0.1.x86_64              libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64            libnftnl-1.2.2-2.amzn2023.0.2.x86_64             pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
```

- Then, configure cgroup in a daemon.json file by using following commands
  cd /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
  "max-size": "100m"
  },
  "storage-driver": "overlay2"
  }
  EOF

```
[ec2-user@ip-172-31-26-174 ~]$ cd /etc/docker
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.
  sudo systemctl enable docker
  sudo systemctl daemon-reload
  sudo systemctl restart docker

```
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/syste
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-26-174 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

- docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Then Install Kubernetes with the following command.

- SELinux needs to be disable before configuring kubelet thus run the following command
  sudo setenforce 0
  sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- Here We are adding kubernetes using the repository whose command is given below.
  cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
  [kubernetes]
  name=Kubernetes
  baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
  enabled=1
  gpgcheck=1
  gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
  exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
  EOF

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl:

  sudo yum update

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
Kubernetes                                                                                                  100 kB/s |  17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.
================================================================================================================================================
 Package                                 Architecture                Version                                 Repository                      Size
================================================================================================================================================
Installing:
 kubeadm                                 x86_64                      1.30.5-150500.1.1                       kubernetes                      10 M
 kubectl                                 x86_64                      1.30.5-150500.1.1                       kubernetes                      10 M
 kubelet                                 x86_64                      1.30.5-150500.1.1                       kubernetes                      17 M
Installing dependencies:
 conntrack-tools                         x86_64                      1.4.6-2.amzn2023.0.2                    amazonlinux                    208 k
 cri-tools                               x86_64                      1.30.1-150500.1.1                       kubernetes                     8.6 M
 kubernetes-cni                          x86_64                      1.4.0-150500.1.1                        kubernetes                     6.7 M
 libnetfilter_cthelper                   x86_64                      1.0.0-21.amzn2023.0.2                   amazonlinux                     24 k
 libnetfilter_cttimeout                  x86_64                      1.0.0-19.amzn2023.0.2                   amazonlinux                     24 k
 libnetfilter_queue                      x86_64                      1.0.5-2.amzn2023.0.2                    amazonlinux                     30 k

Transaction Summary
================================================================================================================================================
Install  9 Packages
```

```
Total                                                                                                       64 MB/s |  53 MB     00:00
Kubernetes                                                                                                  20 kB/s | 1.7 kB     00:00
Importing GPG key 0x9A296436:
 Userid     : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
 Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
 From       : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                      1/1
  Installing       : kubernetes-cni-1.4.0-150500.1.1.x86_64                                                                               1/9
  Installing       : cri-tools-1.30.1-150500.1.1.x86_64                                                                                   2/9
  Installing       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                                                                       3/9
  Installing       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64                                                                  4/9
  Installing       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                                                   5/9
  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-1.30.5-150500.1.1.x86_64                                                                                     7/9
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64                                                                                     7/9
  Installing       : kubeadm-1.30.5-150500.1.1.x86_64                                                                                     8/9
  Installing       : kubectl-1.30.5-150500.1.1.x86_64                                                                                     9/9
  Running scriptlet: kubectl-1.30.5-150500.1.1.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-1.30.1-150500.1.1.x86_64                                                                                   5/9
  Verifying        : kubeadm-1.30.5-150500.1.1.x86_64                                                                                     6/9
  Verifying        : kubectl-1.30.5-150500.1.1.x86_64                                                                                     7/9
  Verifying        : kubelet-1.30.5-150500.1.1.x86_64                                                                                     8/9
  Verifying        : kubernetes-cni-1.4.0-150500.1.1.x86_64                                                                               9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64              cri-tools-1.30.1-150500.1.1.x86_64              kubeadm-1.30.5-150500.1.1.x86_64
  kubectl-1.30.5-150500.1.1.x86_64                         kubelet-1.30.5-150500.1.1.x86_64                kubernetes-cni-1.4.0-150500.1.1.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!
```

- After installing Kubernetes, we need to configure internet options to allow bridging.

1. sudo swapoff -a
2. echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
3. sudo sysctl -p

```
[ec2-user@ip-172-31-26-174 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

4. Initialize the Kubecluster

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0913 10:32:44.629146   26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
```

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8 \
        --discovery-token-ca-cert-hash sha256:8293b2f6d29de466bd859007f5adbcdb3a
ecb0c446ba09033d32a5846b3d434f
```

- copy the token and save for future use .
  kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8
  \--discovery-token-ca-cert-hash
  sha256:8293b2f6d29de466bd859007f5adbcdb3aecb0c446ba09033d32a5846b3d434f
- Copy the mkdir and chown commands from the top and execute them
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
[ec2-user@ip-172-31-80-126 docker]$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- Then, add a common networking plugin called flannel as mentioned in the code.

kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

5. Now that the cluster is up and running,we can deploy our nginx server on this cluster. Apply deployment using this following command:

kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://k8s.io/examples/pods/s
imple-pod.yaml
pod/nginx created
```

Then use **kubectl get pods** to check whether the pod gets created or not.

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          12s
```

To convert state from pending to running use following command:

kubectl describe pod nginx This command will help to describe the pods it gives reason for failure as it shows the untolerated taints which need to be untainted.

- kubectl describe pod nginx

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k4lj6 (ro)
```

```
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-k4lj6:
    Type:                    Projected (a volume that contains injected data from m
ultiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 3
00s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for
 300s
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  7s    default-scheduler  0/1 nodes are available: 1 no
de(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption:
0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

- kubectl taint nodes --all node-role.kubernetes.io/control-plane-

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl taint nodes --all node-role.kubernetes.io
/control-plane-
node/ip-172-31-26-174.ec2.internal untainted
```

6. Now check pod status is is running perform **kubectl get pods** this command.

```
[ec2-user@ip-172-31-28-70 docker]$  kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          39s
[ec2-user@ip-172-31-28-70 docker]$  kubectl get pods
NAME    READY   STATUS    RESTARTS      AGE
nginx   1/1     Running   1 (45s ago)   70s
```

7.Lastly, mention the port you want to host. Here i have used localhost 8081 then check it.

kubectl port-forward nginx 8081:80

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

8. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8081

If the response is 200 OK and you can see the Nginx server name, your deployment was successful. We have successfully deployed our Nginx server on our EC2 instance.

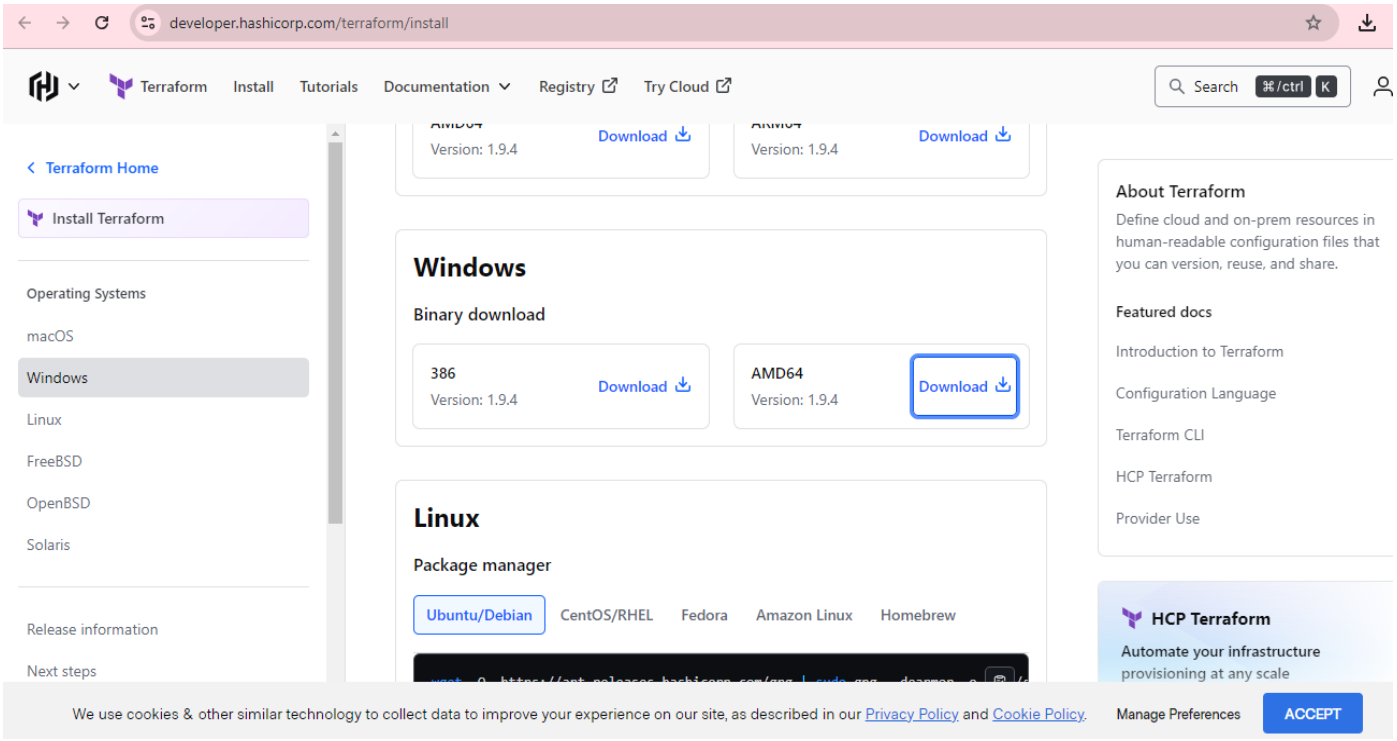
Conclusion: Firstly I created an EC2 AWS Linux instance successfully.then installed docker and

kubernetes successfully.then initialized kubernetes which given me token and chown and mkdir command. Then I execute mkdir and chown the command successfully. Then I installed a networking plugin called flannel successfully. Then I tried to deploy nginx which initially gave an error. Then I deployed (simple-pod.yml) nginx successfully and also checked by using the get pods command.then hosted it on localhost 8081 ie http://localhost:8081 successfully.

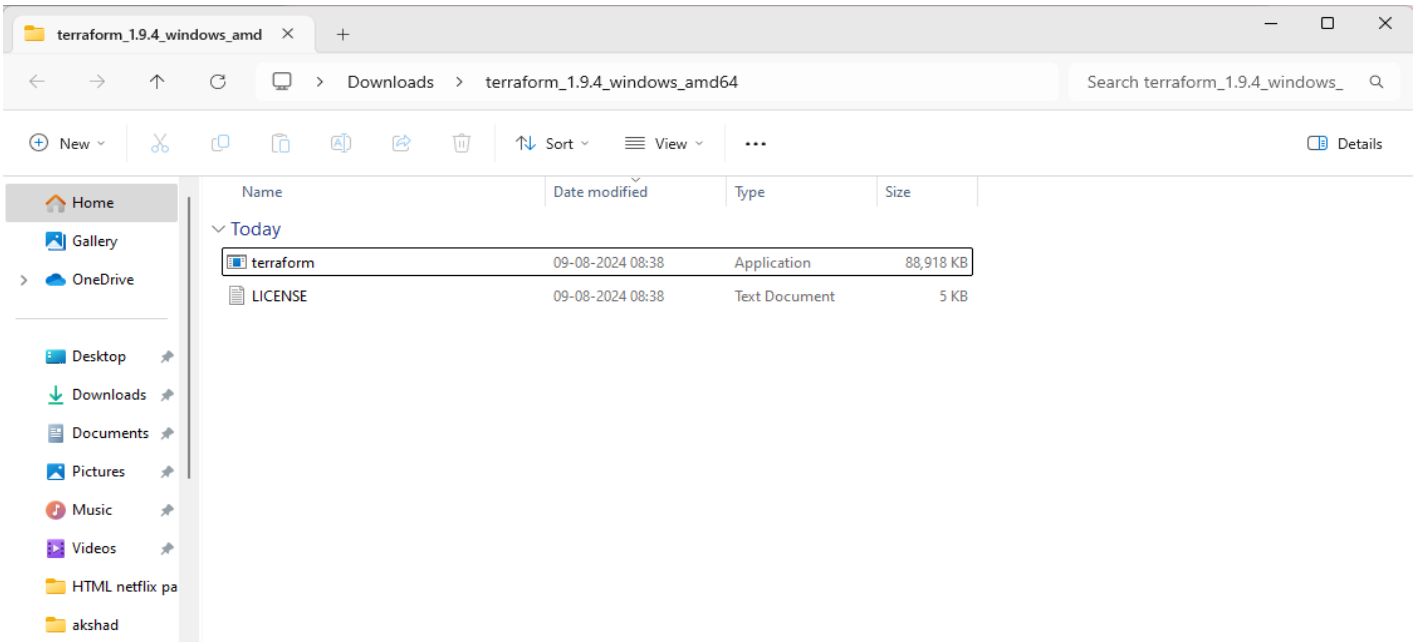
Adve_devops_5 Name: sadneya sadanand samant Roll no:46

Experiment No 5

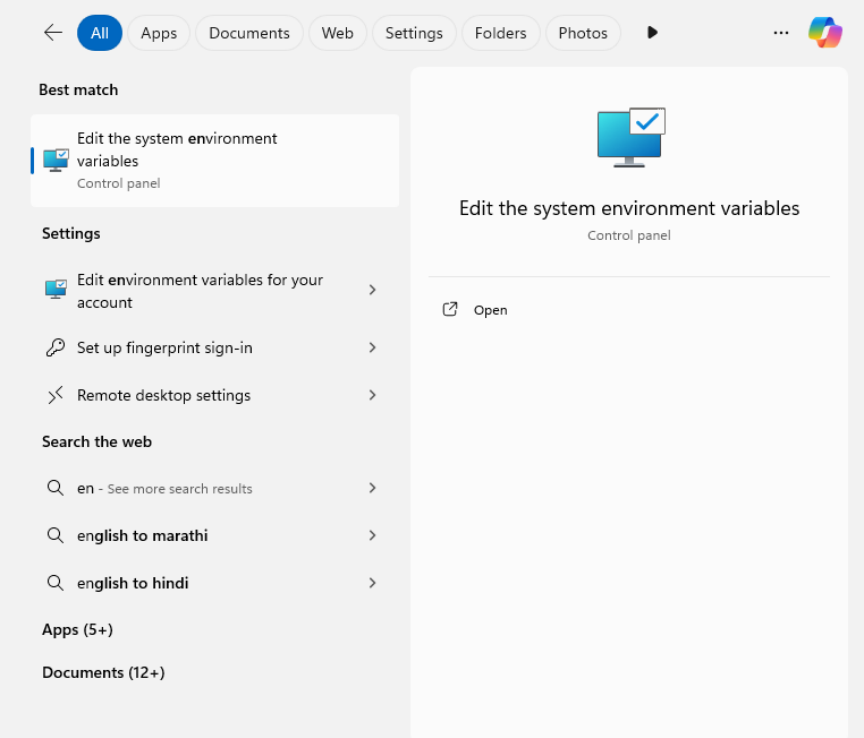
1. To install , First download Terraform Client Utility for windows from terraforms official website https://developer.hashicorp.com/terraform/install then select the operating system on which you want to download.



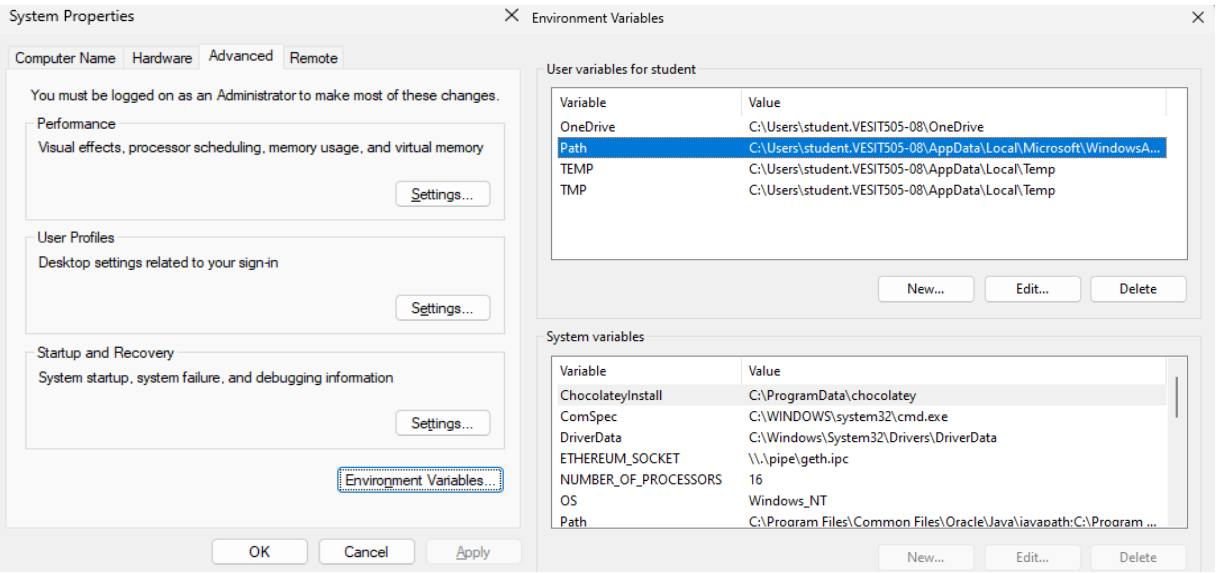
2. Extract the file from downloads. Then click on terraform Application.



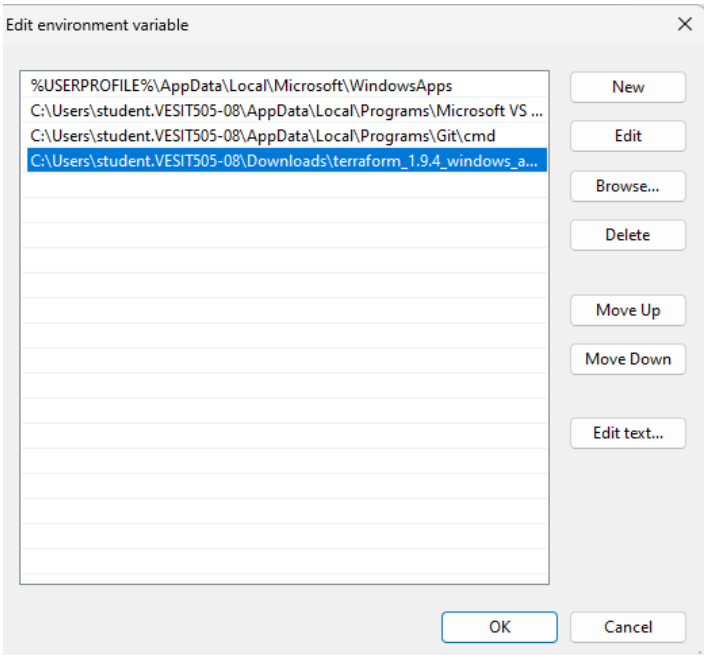
3. Then search for Edit the system environment variables.



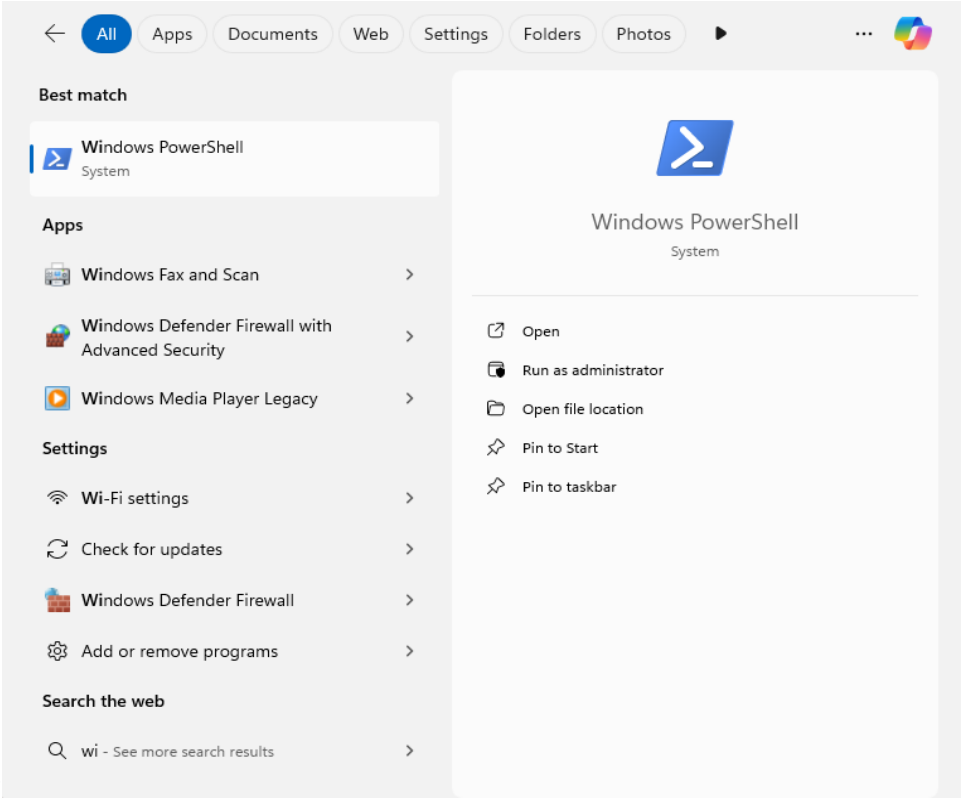
4. Then click on Environment Variables and then click on PATH.



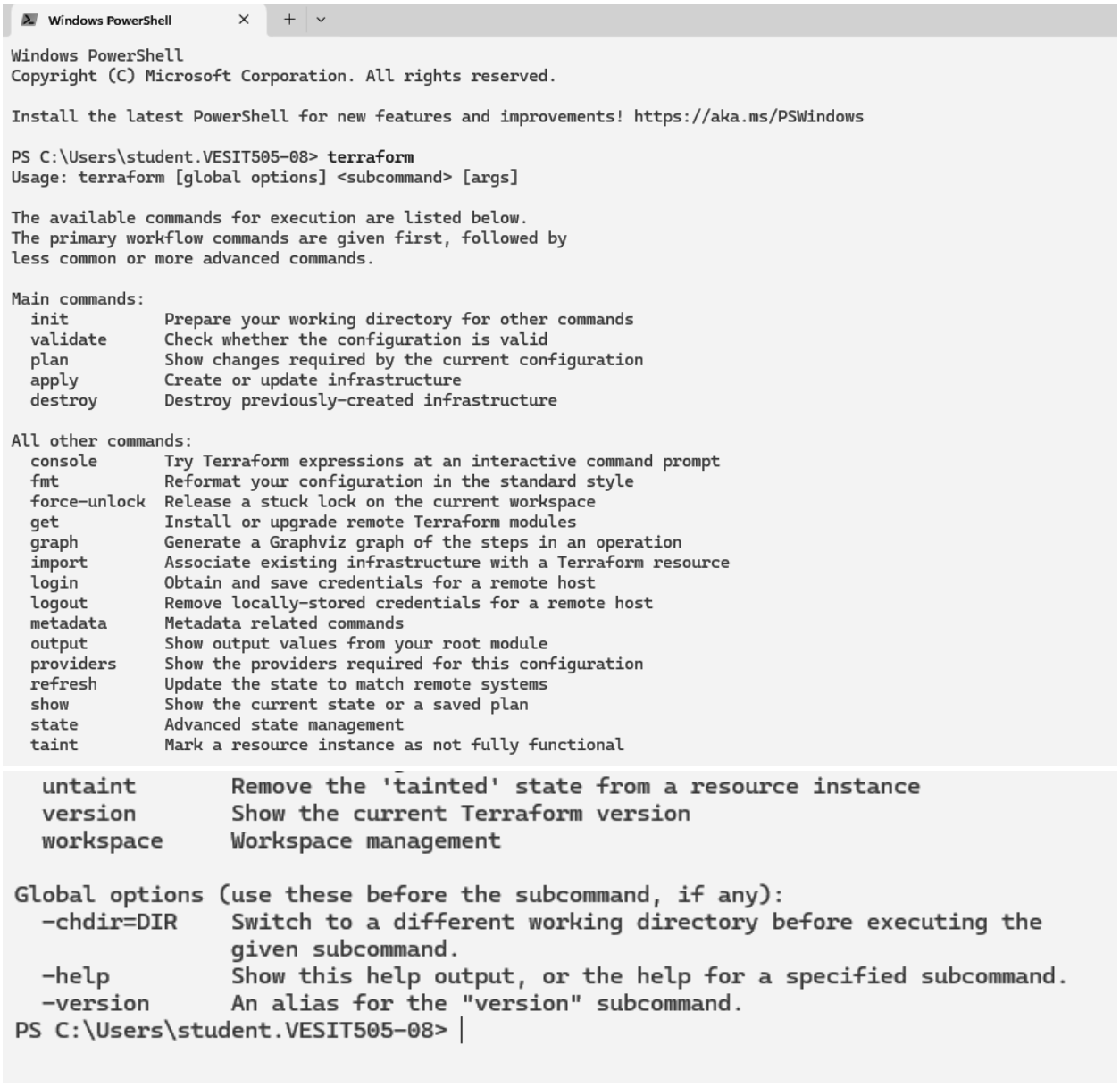
5. Then add the path of where you have stored the Terraform. Then click on ok.



6. Then search for windows powershell and open it as an administrator.



7. Then type terraform which will give information about commands ie. main commands and all other commands.



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

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

PS C:\Users\student.VESIT505-08> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output, or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.
PS C:\Users\student.VESIT505-08>
```

EXPERIMENT NO:6

A. Creating docker image using terraform

1) Download and Install Docker Desktop from https://www.docker.com/

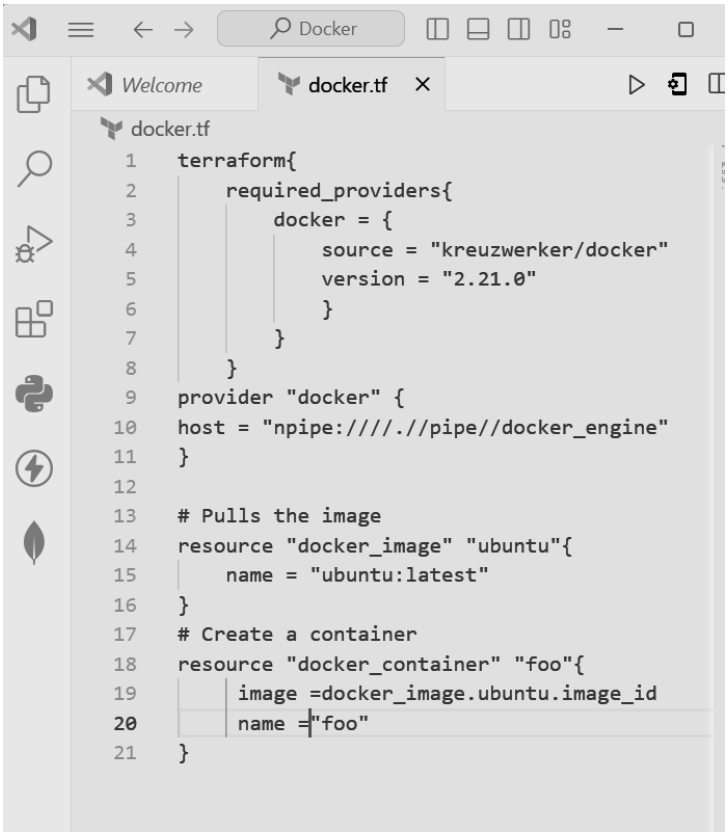
Step 1: Check the DOCKER Functionality.

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sadneya>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  pull        Download an image from a registry
  push        Upload an image to a registry
  images      List images
  login       Log in to a registry
  logout      Log out from a registry
  search      Search Docker Hub for images
  version     Show the Docker version information
  info        Display system-wide information

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx
  checkpoint  Manage checkpoints
  compose*    Docker Compose
  container   Manage containers
  context     Manage contexts
  debug*      Get a shell into any image or container
  desktop*    Docker Desktop commands (Alpha)
  dev*        Docker Dev Environments
  extension*  Manages Docker extensions
  feedback*   Provide feedback, right in your terminal!
  image       Manage images
  init*       Creates Docker-related starter files for your project
```

```
C:\Users\Sadneya>docker --version
Docker version 27.0.3, build 7d4bcd8
```

Now,create a folder and give it name as ‘Terraform Scripts’ in which we save our different types which will be further used

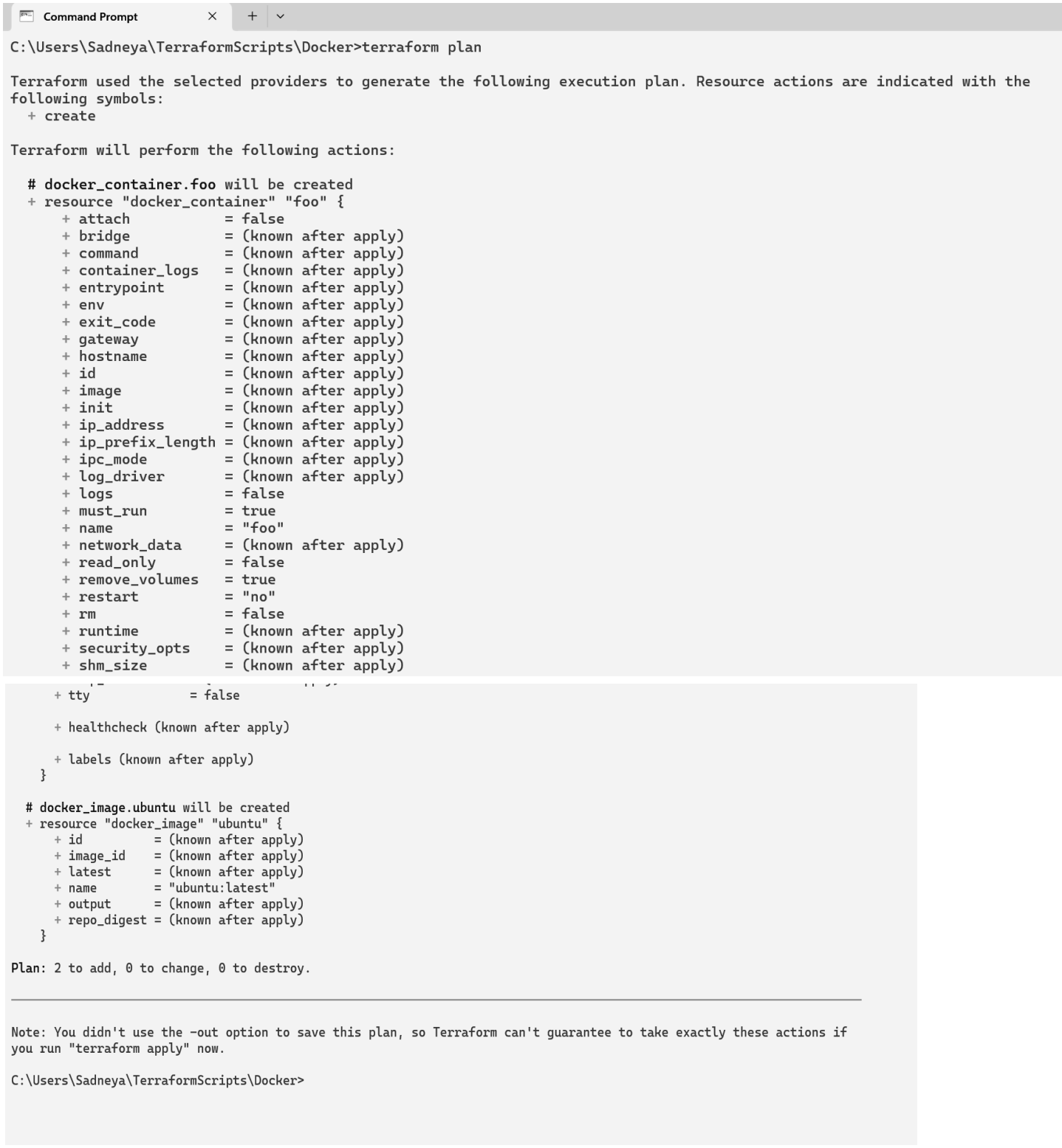
Step 2: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file in VS code and write the following script.



Step 3: Execute Terraform Init command to initialize the resources

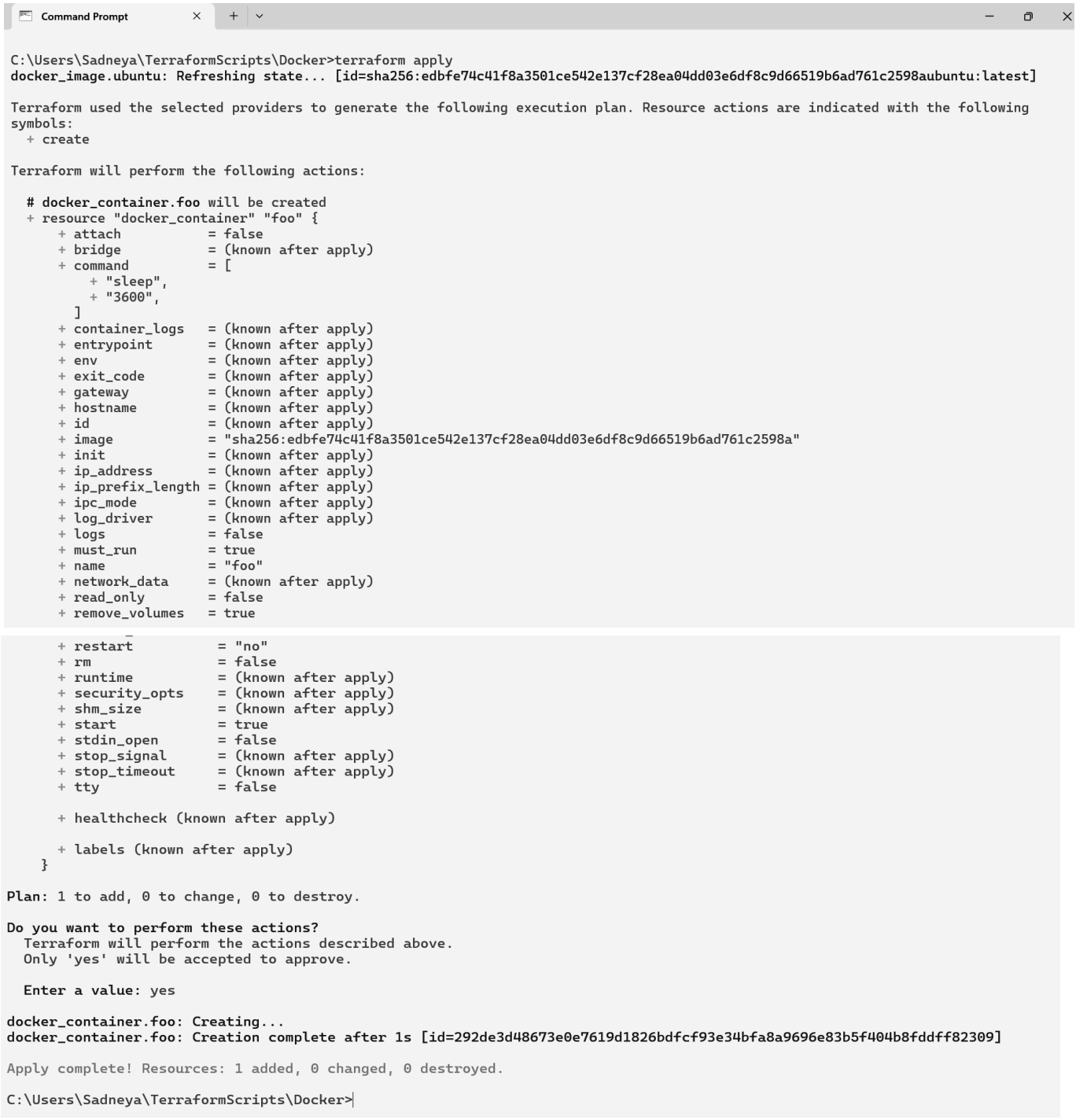
```
C:\Users\Sadneya>cd TerraformScripts

C:\Users\Sadneya\TerraformScripts>cd Docker

C:\Users\Sadneya\TerraformScripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Step 4: Execute Terraform plan to see the available resources



```
Command Prompt

C:\Users\Sadneya\TerraformScripts\Docker>terraform plan

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + 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)
      + 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.

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.

C:\Users\Sadneya\TerraformScripts\Docker>
```

Step 5: Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : "**terraform apply"**



```
Command Prompt

C:\Users\Sadneya\TerraformScripts\Docker>terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + 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            = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

Plan: 1 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_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fddff82309]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Users\Sadneya\TerraformScripts\Docker>
```

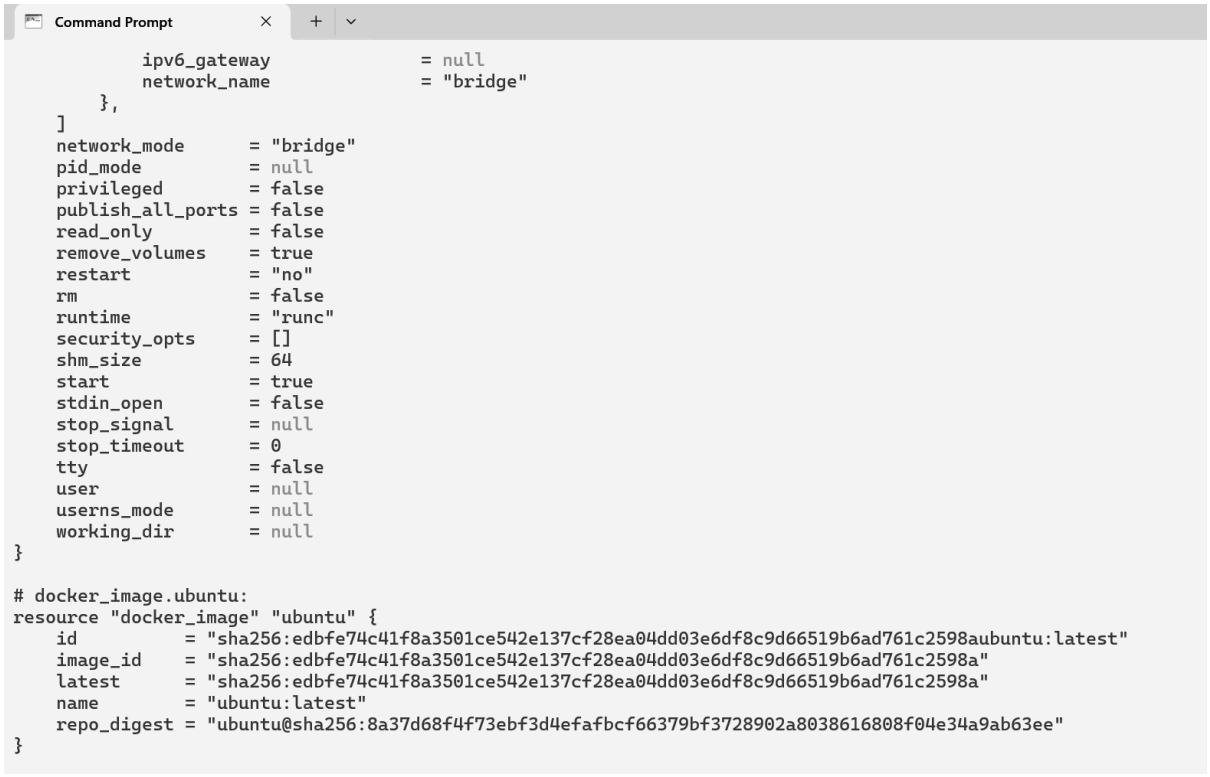
Docker images, Before Executing Apply step:

```
C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

C:\Users\Sadneya\TerraformScripts\Docker>
```

Docker images, After Executing Apply step:

```
C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    edbfe74c41f8   3 weeks ago   78.1MB
```

Step 6:terraform show it will Show the state file in a human-readable format.



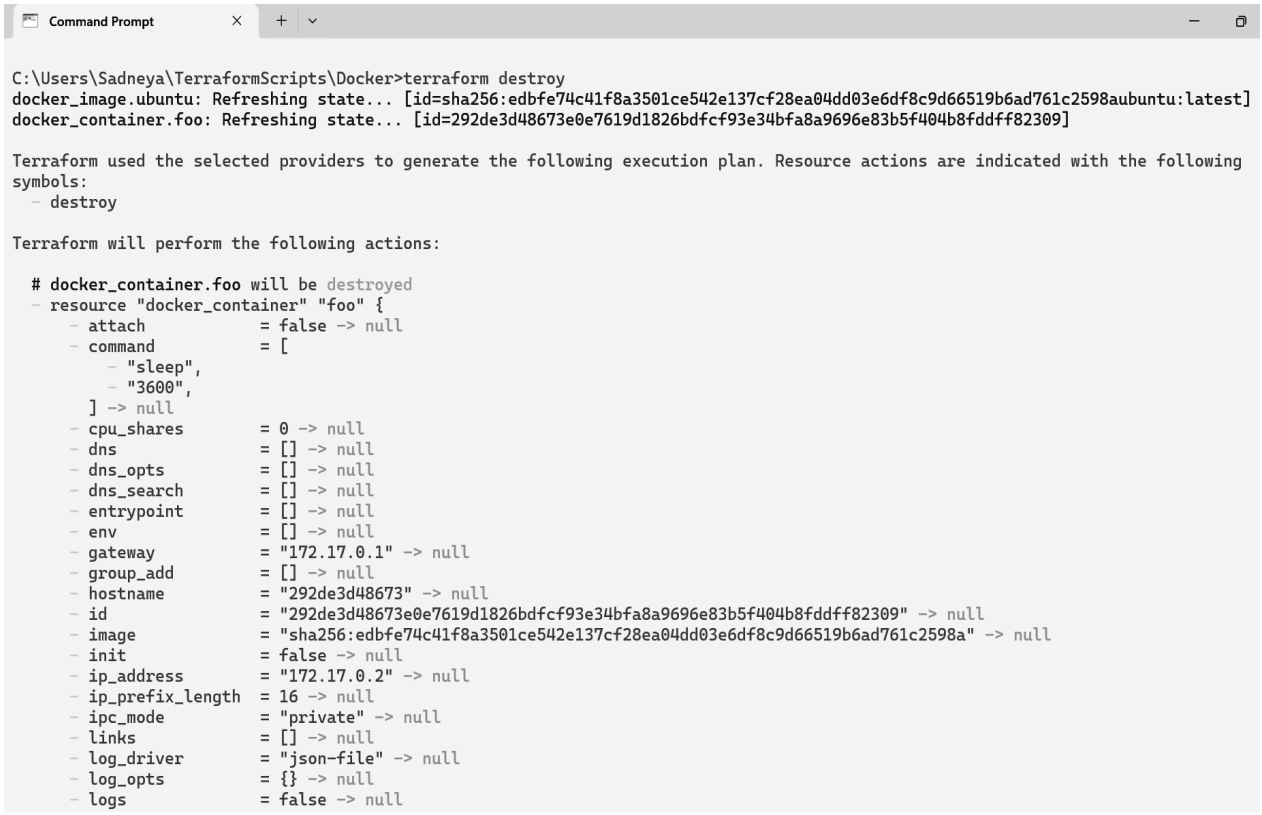
Step 7:terraform state list diplays Lists out all the resources that are tracked in the current state file.

```
C:\Users\Sadneya\TerraformScripts\Docker>terraform state list
docker_container.foo
docker_image.ubuntu
```

Step 8:terraform graph Produces a graph in DOT language showing the dependencies between objects in the state file. This can then be rendered by a program called Graphviz (amongst others).

```
C:\Users\Sadneya\TerraformScripts\Docker>terraform graph
digraph G {
  rankdir = "RL";
  node [shape = rect, fontname = "sans-serif"];
  "docker_container.foo" [label="docker_container.foo"];
  "docker_image.ubuntu" [label="docker_image.ubuntu"];
  "docker_container.foo" -> "docker_image.ubuntu";
}
```

Step 9: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.



```
Command Prompt

C:\Users\Sadneya\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fddff82309]

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         = "292de3d48673" -> null
      - id               = "292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fddff82309" -> 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     = [
          - {
              - gateway                     = "172.17.0.1"
              - global_ipv6_prefix_length = 0
              - ip_address                  = "172.17.0.2"
              - ip_prefix_length            = 16
              - network_name                = "bridge"
                # (2 unchanged attributes hidden)
            },
        ] -> null
      - network_mode     = "bridge" -> null
      - privileged       = false -> null
      - publish_all_ports = false -> null
      - read_only        = false -> null
      - remove_volumes   = true -> null
      - restart          = "no" -> null
      - rm               = false -> null
      - runtime          = "runc" -> null
      - security_opts    = [] -> null
      - shm_size         = 64 -> null
      - start            = true -> null
      - stdin_open       = false -> null
      - stop_timeout     = 0 -> null
      - storage_opts     = {} -> null
      - sysctls          = {} -> null
      - tmpfs            = {} -> null
      - tty              = false -> null
        # (8 unchanged attributes hidden)
    }

  # docker_image.ubuntu will be destroyed
  - resource "docker_image" "ubuntu" {
```

```
  # docker_image.ubuntu will be destroyed
  - resource "docker_image" "ubuntu" {
      - id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
      - image_id    = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - latest      = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - name        = "ubuntu:latest" -> null
      - repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
    }

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

docker_container.foo: Destroying... [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fddff82309]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.

C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

Docker images After Executing Destroy step

```
C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

C:\Users\Sadneya\TerraformScripts\Docker>
```

EXPERIMENT NO:7

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

Procedure:

1.Creation of project on Sonarqube

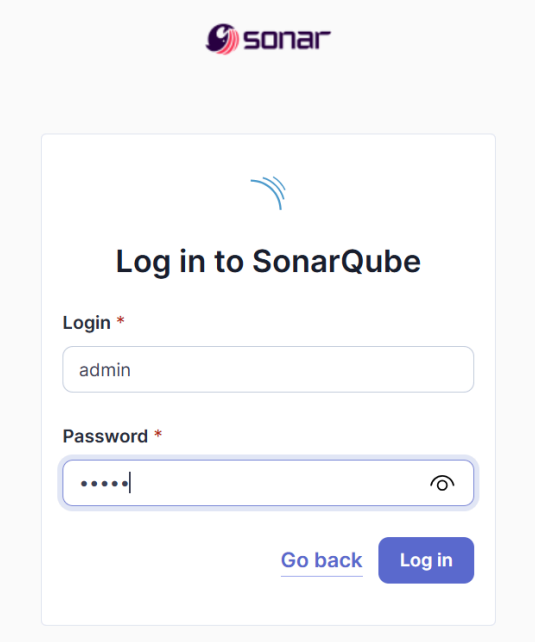
1. Open the command prompt and perform this command

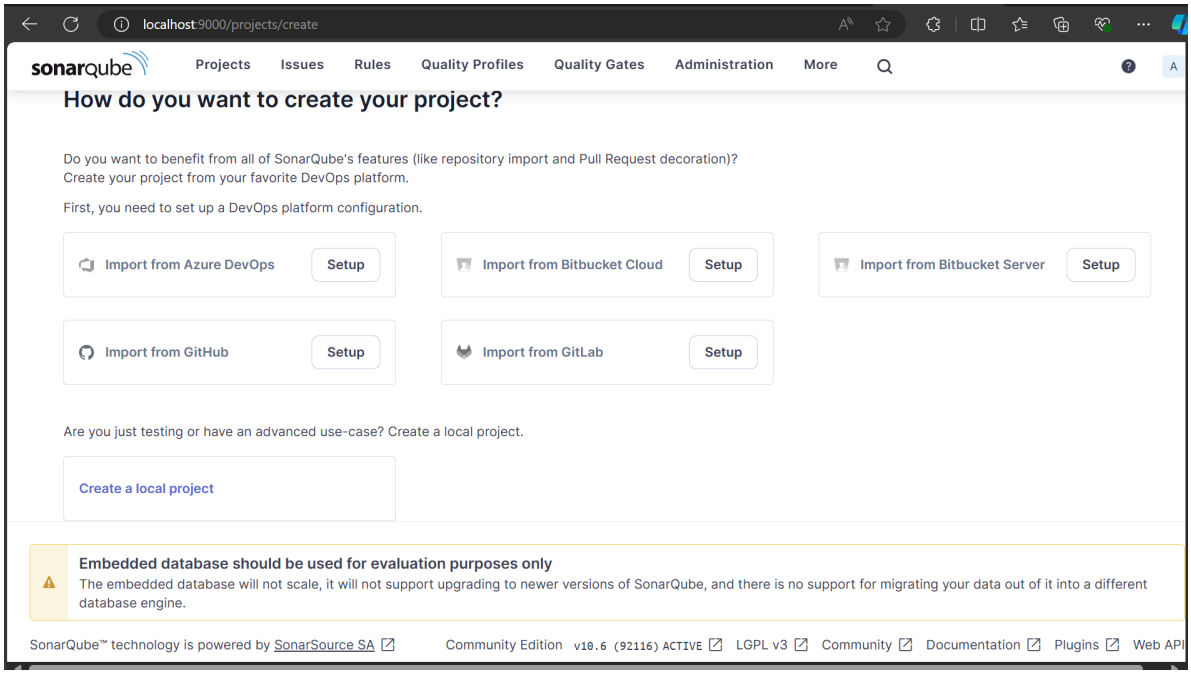
(**Docker must be Installed before running this command.**)

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

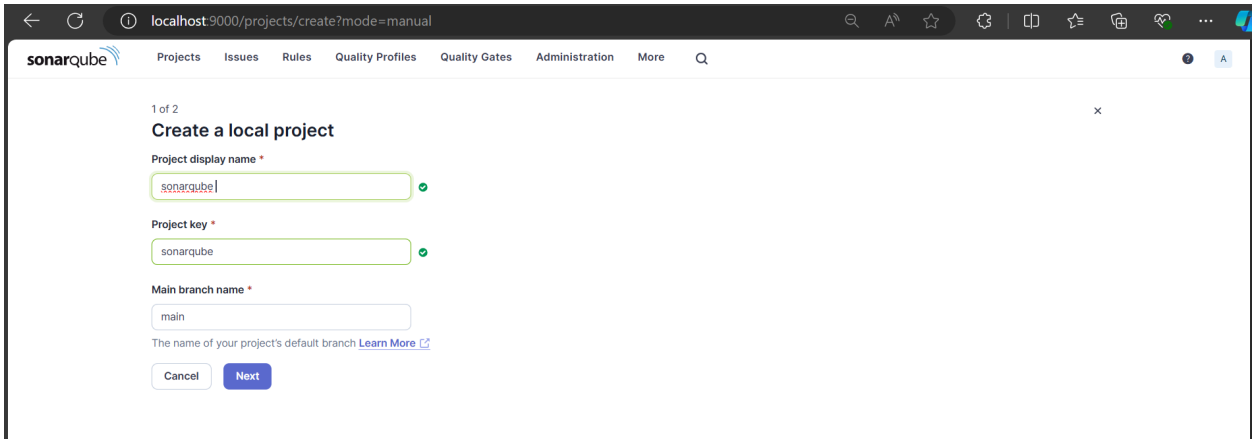
```
PS C:\Users\Sadneya> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
19948c8162691fd8be61d035e355c4fe4a6a2fd0a15237e94f75c0857ff7e2ff
```

2. Then after its successful execution, run the sonarqube at http://localhost:9000
3. Then Login as Username admin and password admin.

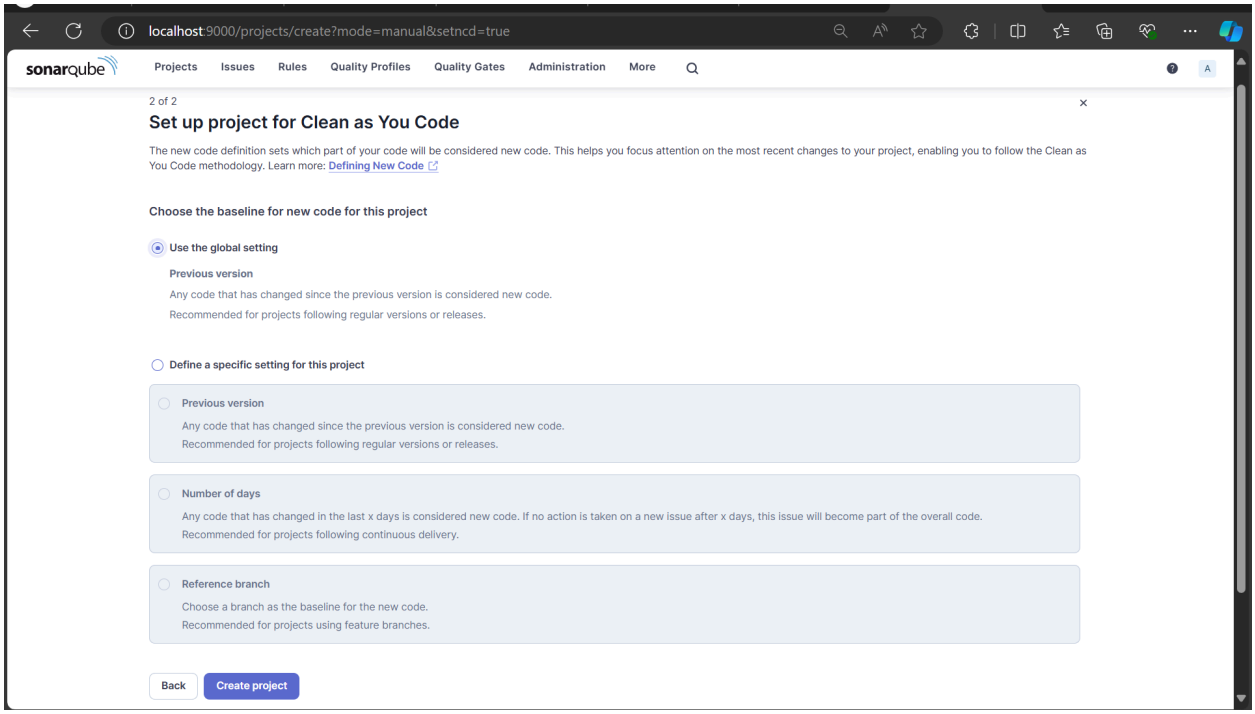




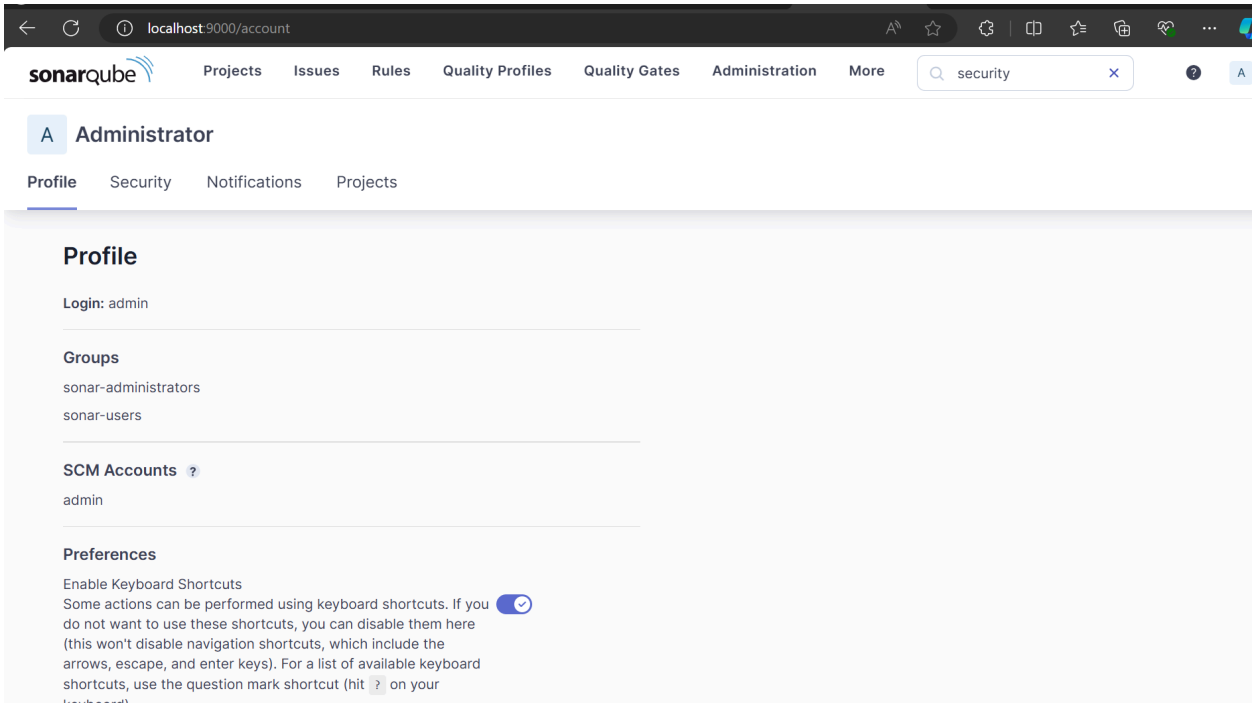
4. Then create a project. Here I have given the name “sonarqube”. Keep branch name main only and then click on next.



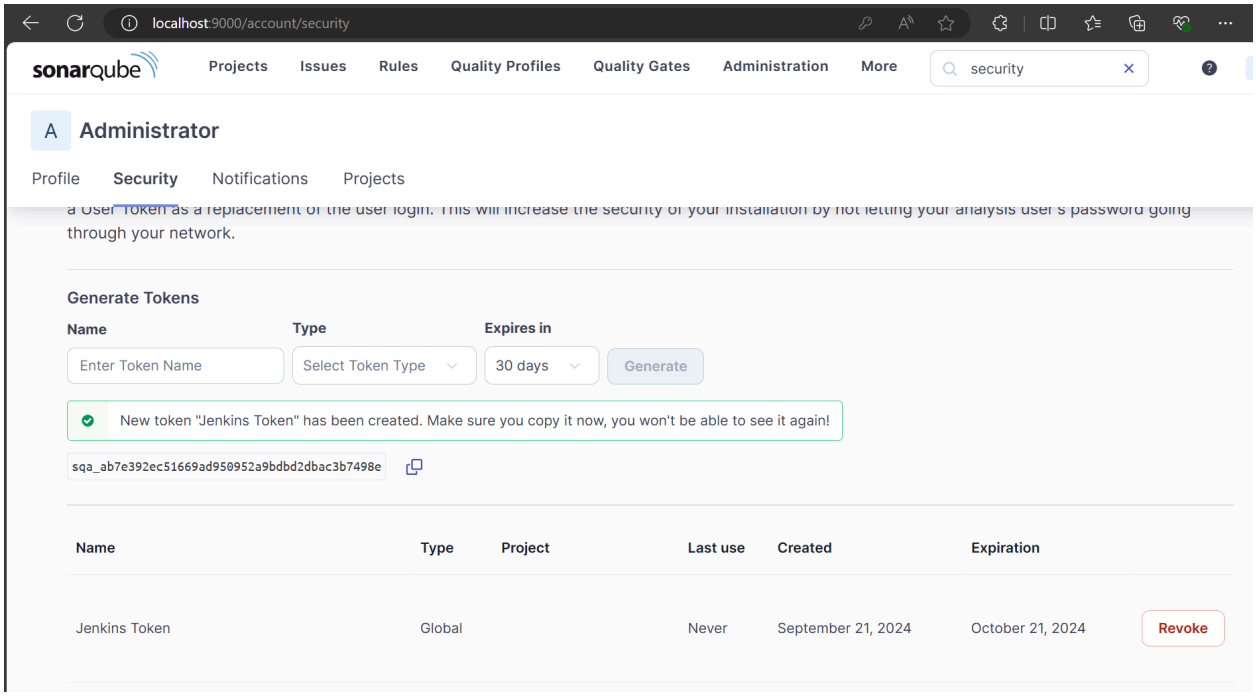
5. Then in the Setup project for clean as you code they will ask to choose the baseline for new code for this project. Choose Use the global setting.Then click on create project.



6. Then click on your account profile and select my account.Inside this click on security option.



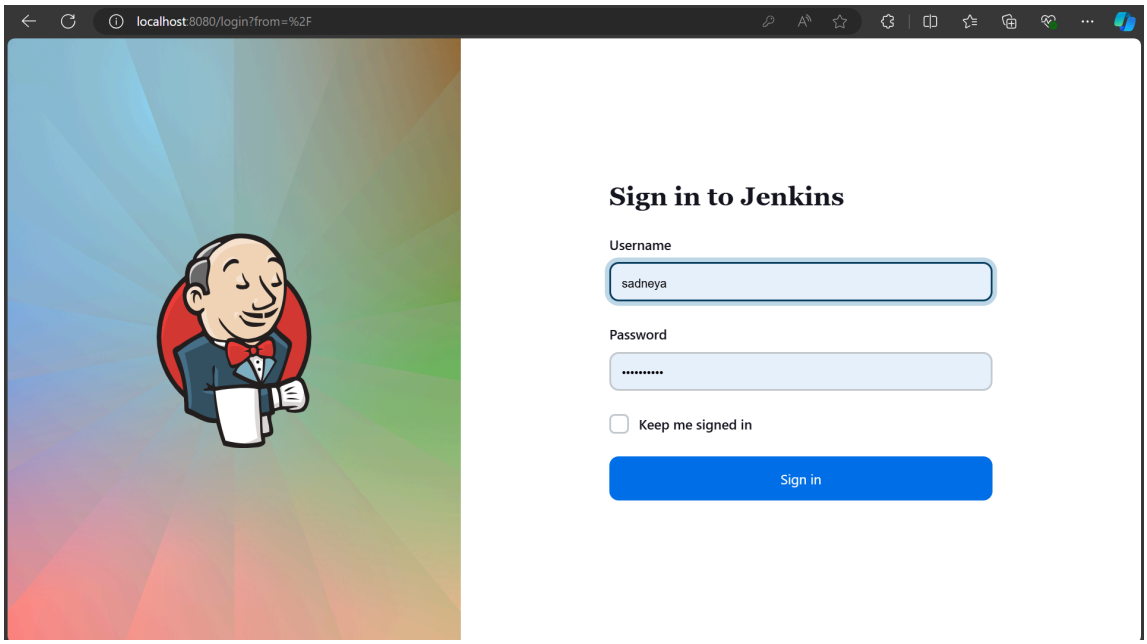
7. Then set the name of your token and here I have chosen the global scope of the token and then clicked on generate token, thus token created successfully.(Here I have given my token name as ‘jenkins token’.)



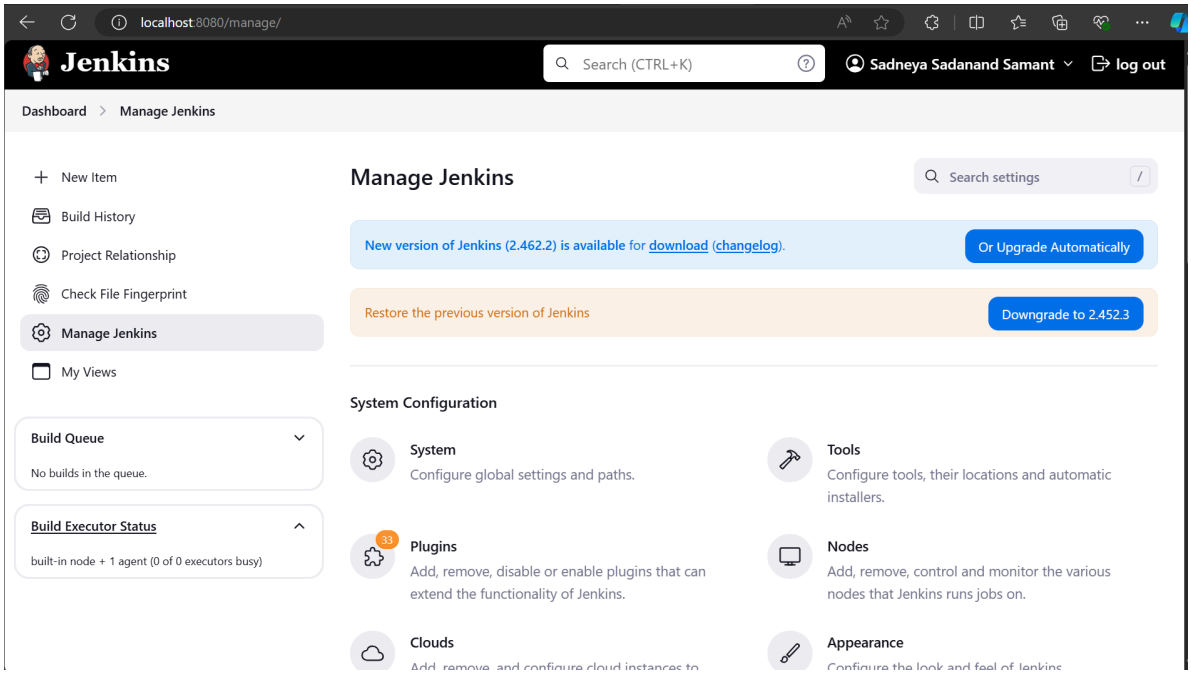
Copy and paste this token as it will be used ahead in Jenkins.

2.Creation Freestyle project on Jenkins

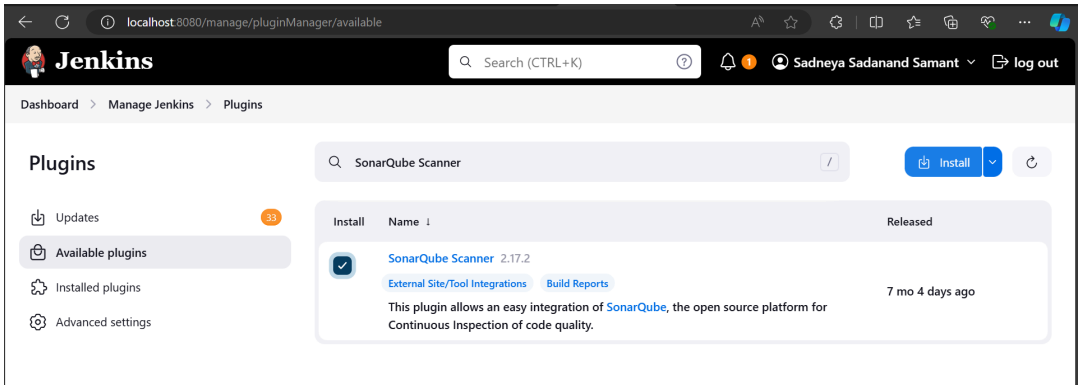
1. Run the jenkins at http://localhost:8080 and enter username and password and click on sign in

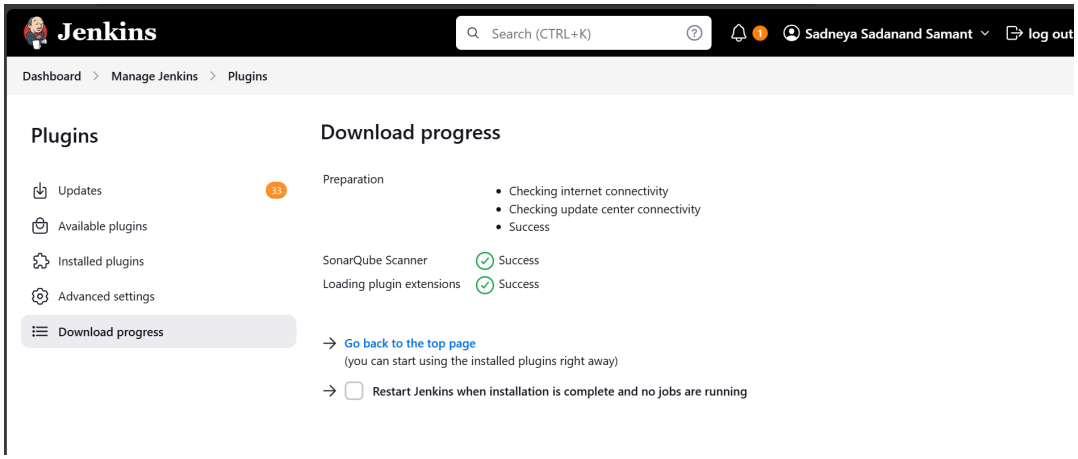


2. Then Click on manage Jenkins.

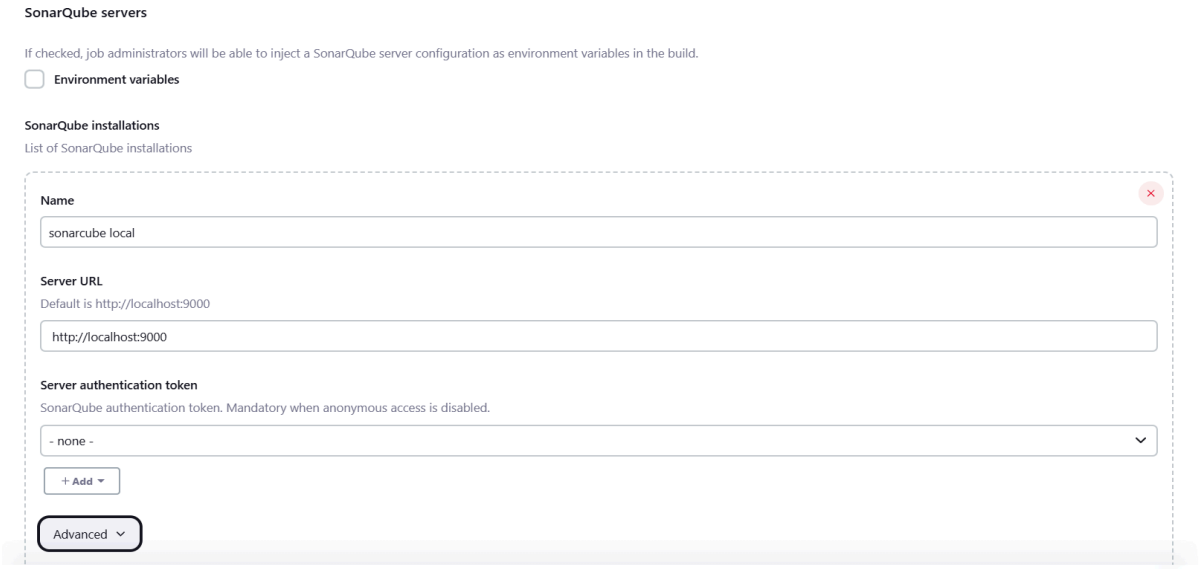


3. Here click on Plugins.Inside Available Plugins search for SonarQube Scanner and then click on Install.Then again restart Jenkins.

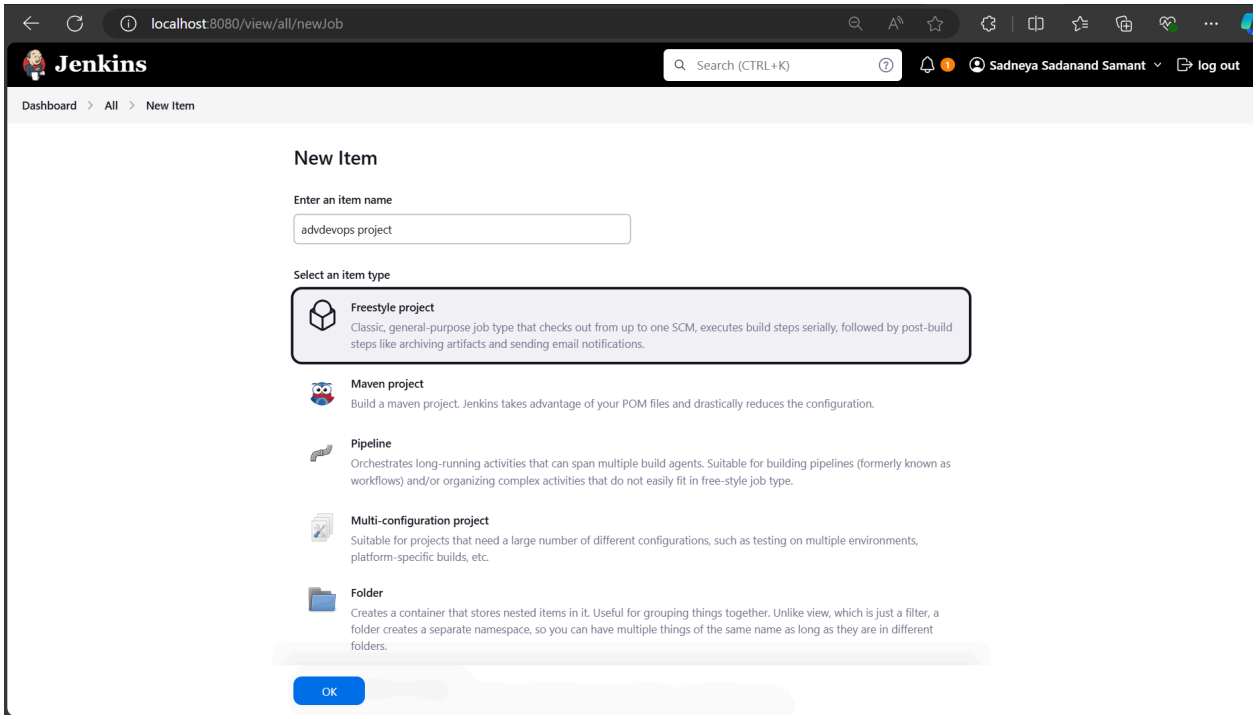




4. Then go back to Manage Jenkins and click on system and search for sonarqube server.

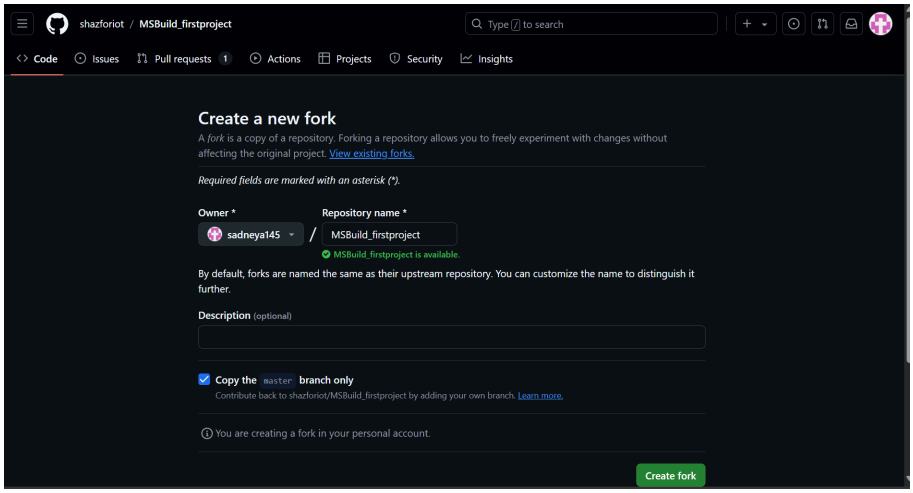


5. Click on the environment variables and give name to SonarQube installation.(here I have given the name as “sonarqube local”) then give the server URL as https://localhost:9000 then save and apply these changes.
6. Then select item type Freestyle Project and give name to your project here I have given name “advdevops project".Then click on ok.

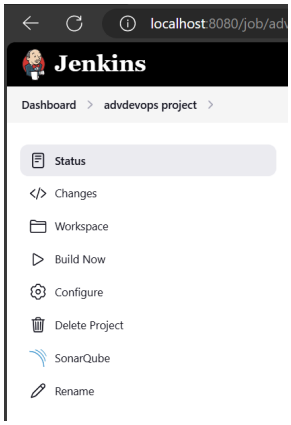


7. Choose the Github repository : https://github.com/shazforiot/MSBuild_firstproject.git

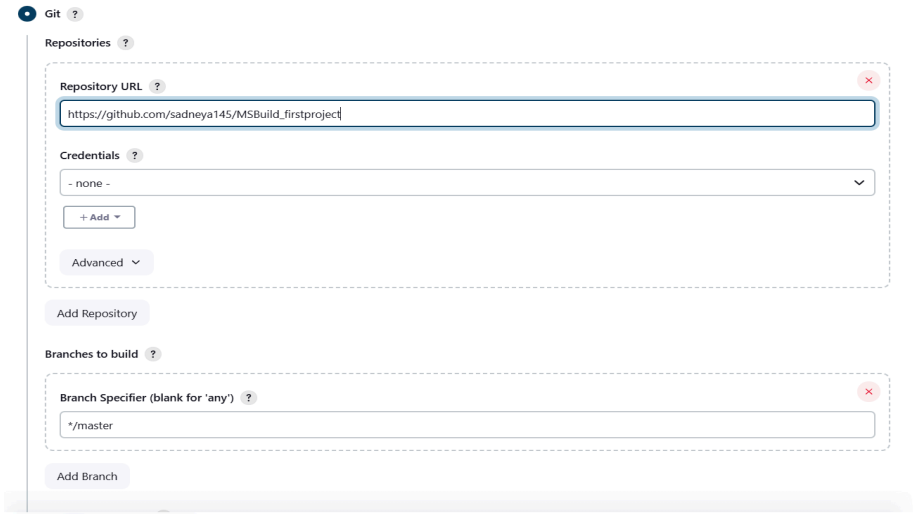
It is a sample hello-world project with no vulnerabilities and issues.It is just for testing purposes.
Fork This Repository.



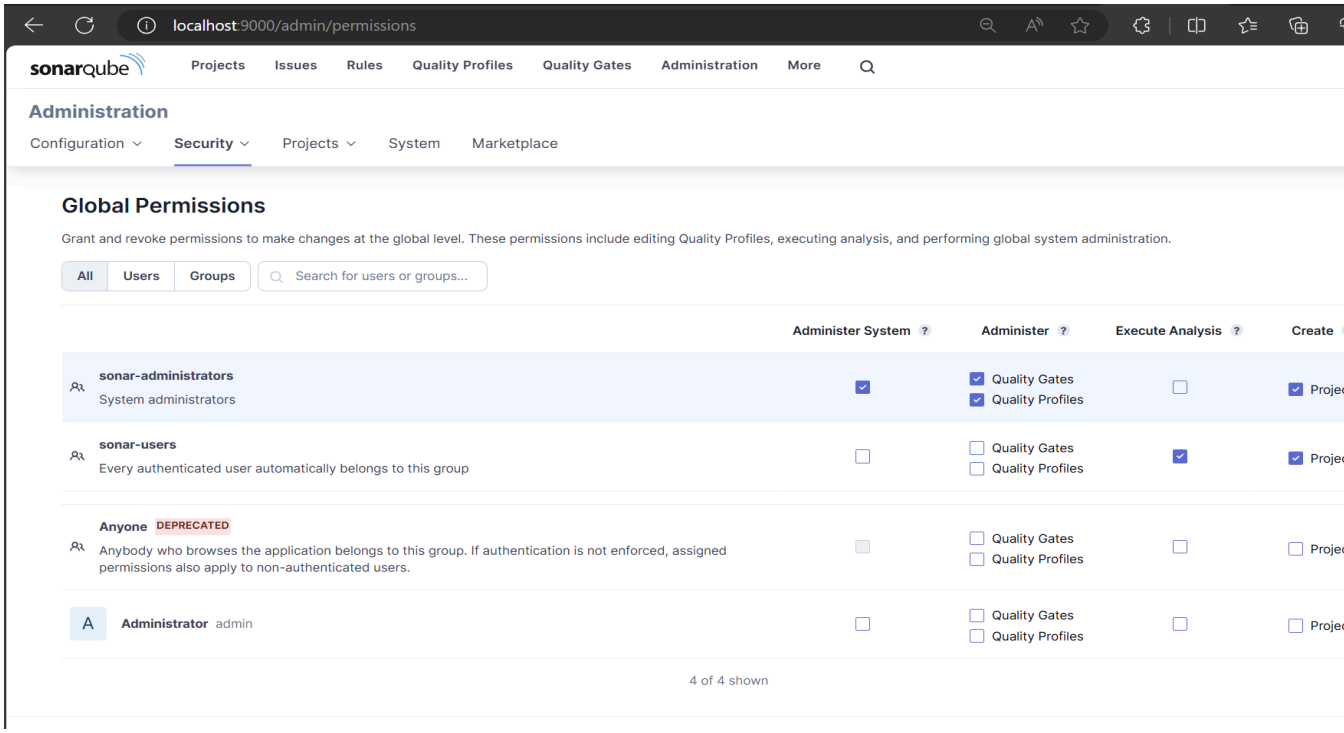
8. Then click on your project and then go to configure.



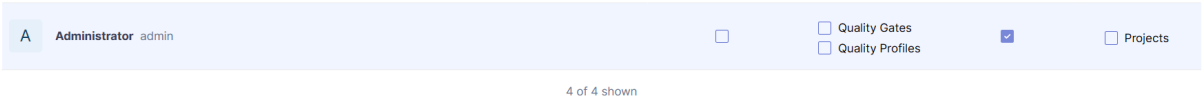
9. Then enable Git and add that github Repository URL.select */master branch.



10. Go to Sonarqube http://localhost:9000/<user_name>/permissions.



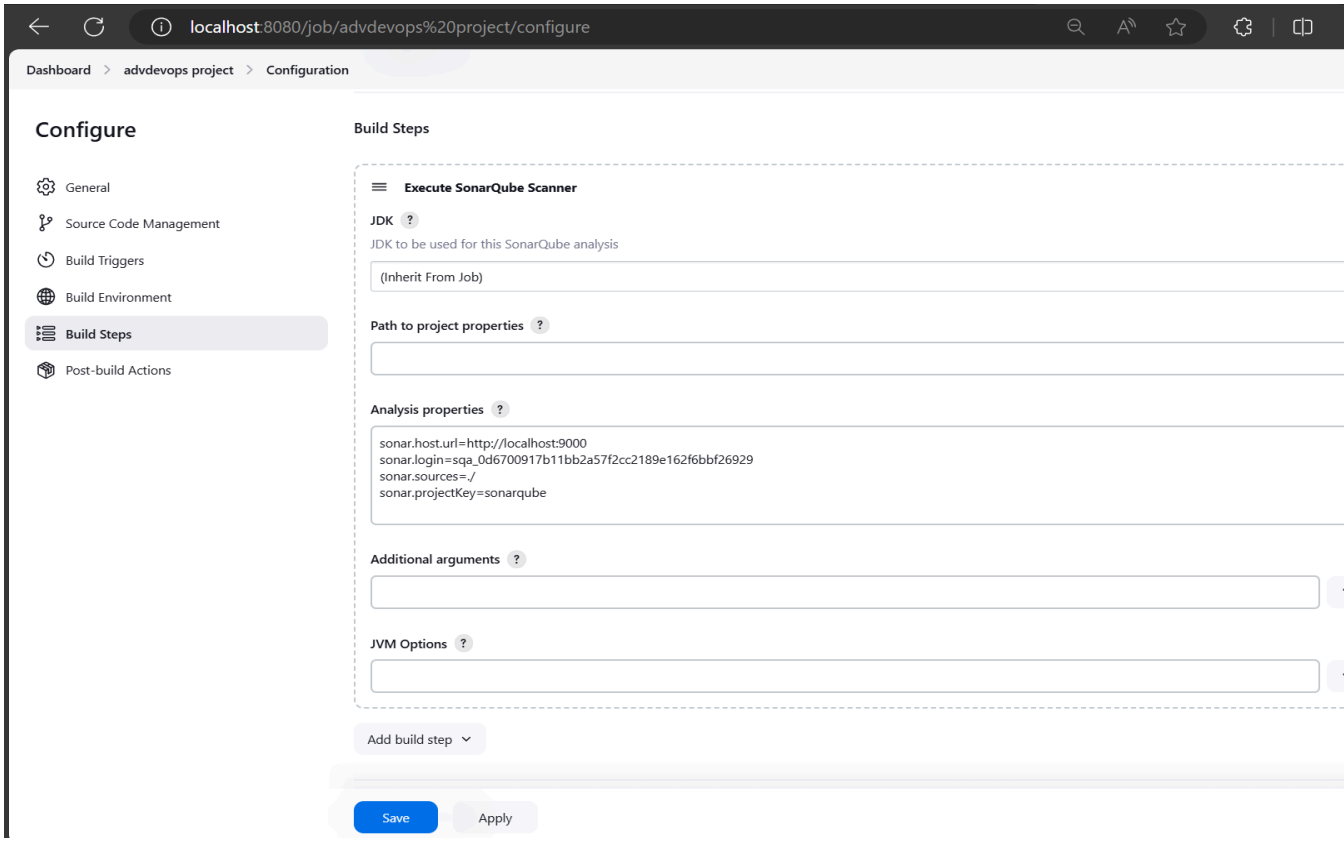
Then allow execute permissions to Admin User.



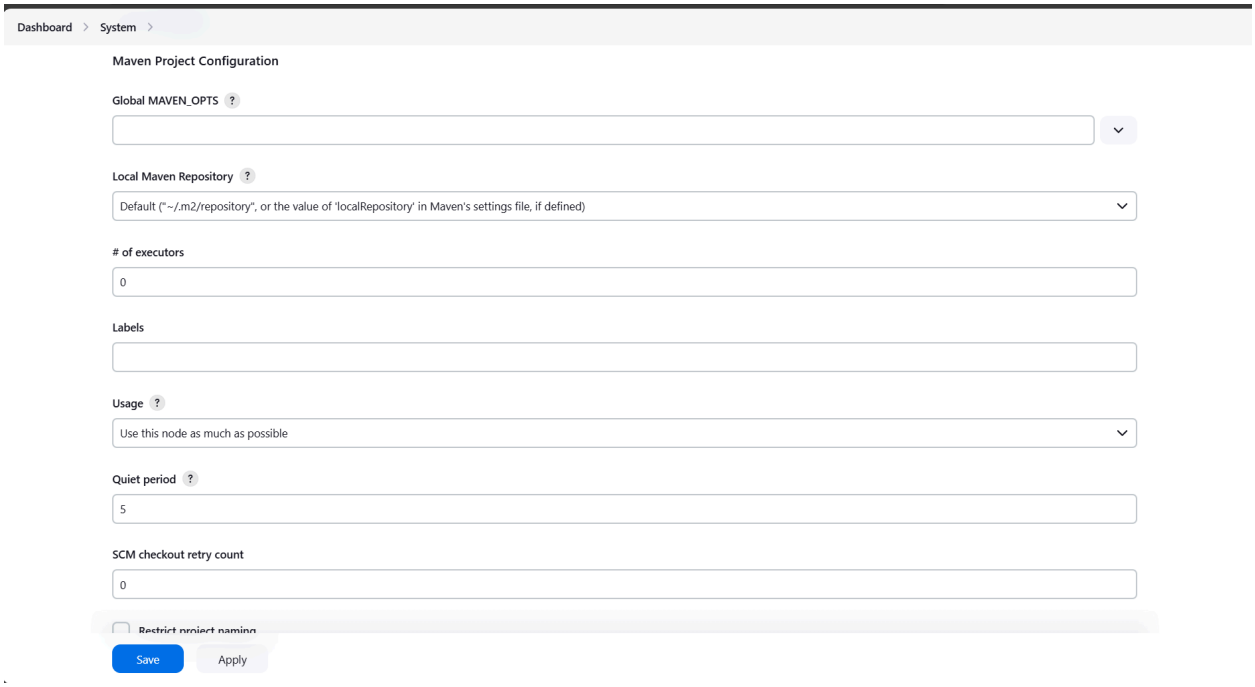
11. Then again Go inside the project and click on configure (addevops project->configure).Go there in the build steps section. Inside It add the analysis properties.

sonar.host.url=http://localhost:9000 (Your_url)
sonar.login=sqa_0d6700917b11bb2a57f2cc2189e162f6bbf26929 (login_token that You have copied on clipboard created in step 7 of creation of project of sonarqube)
sonar.sources=./
sonar.projectKey=sonarqube(project name)

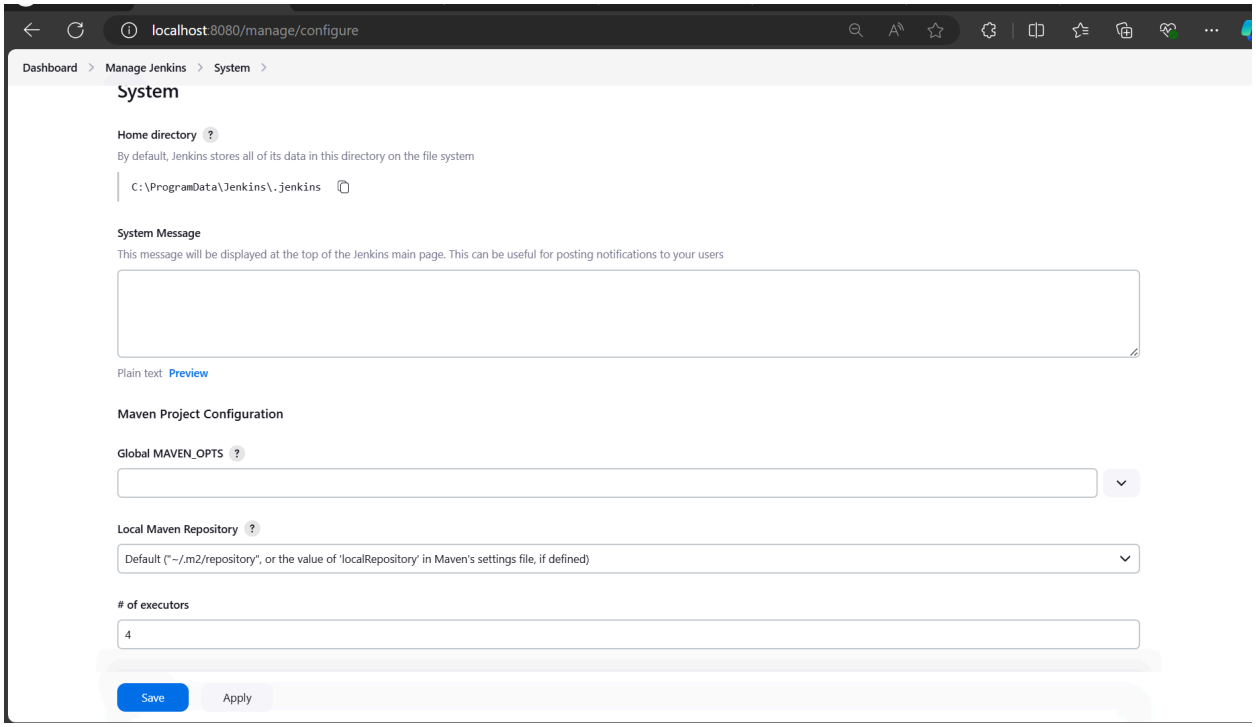
Then click on save and apply.



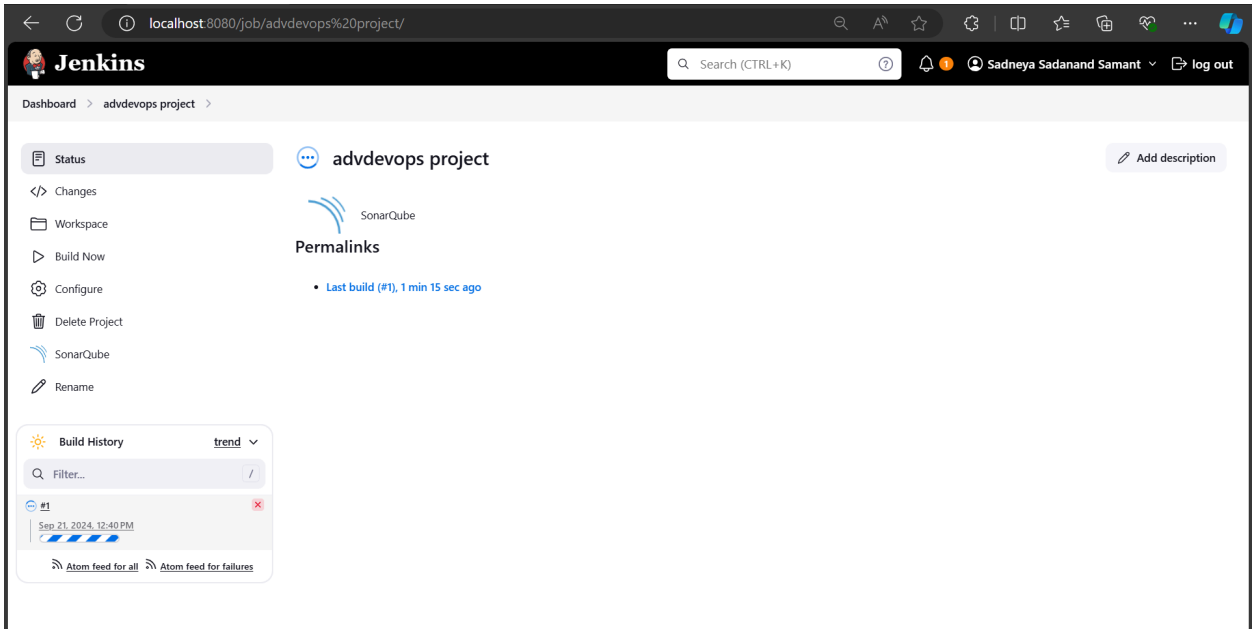
12. Then again go into manage Jenkins and check the number of executors if it is zero then set to two or more.If it is zero then it will not build successfully and give error.

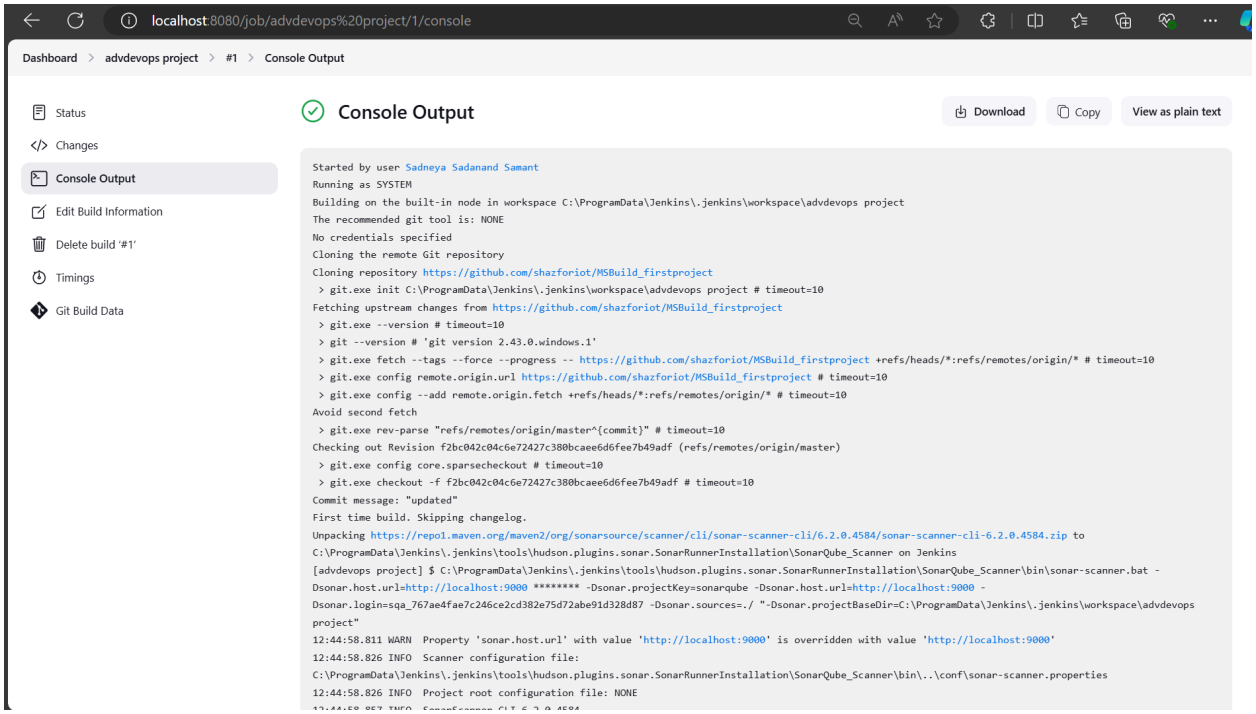


In the above image it was zero that's why my build was getting unsuccessful.Then I declared the value of # of executors to 4.



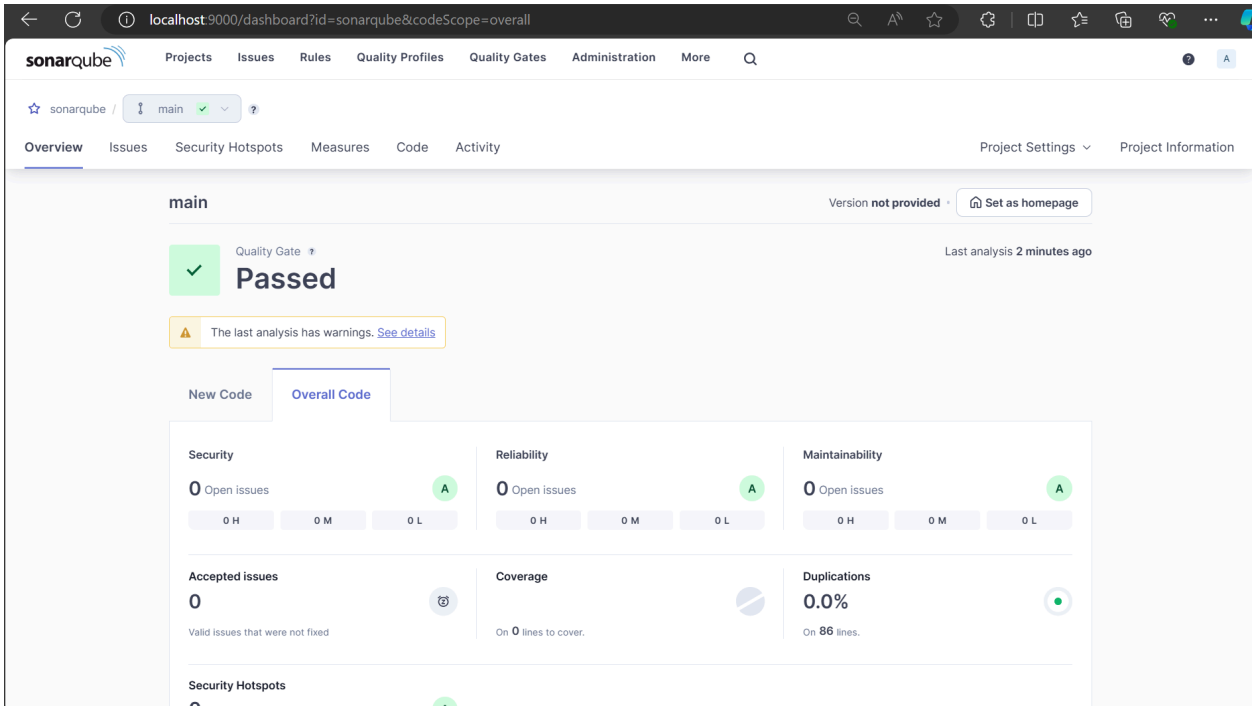
13. Then click on the build project.Thus the build is successful.see the console output.

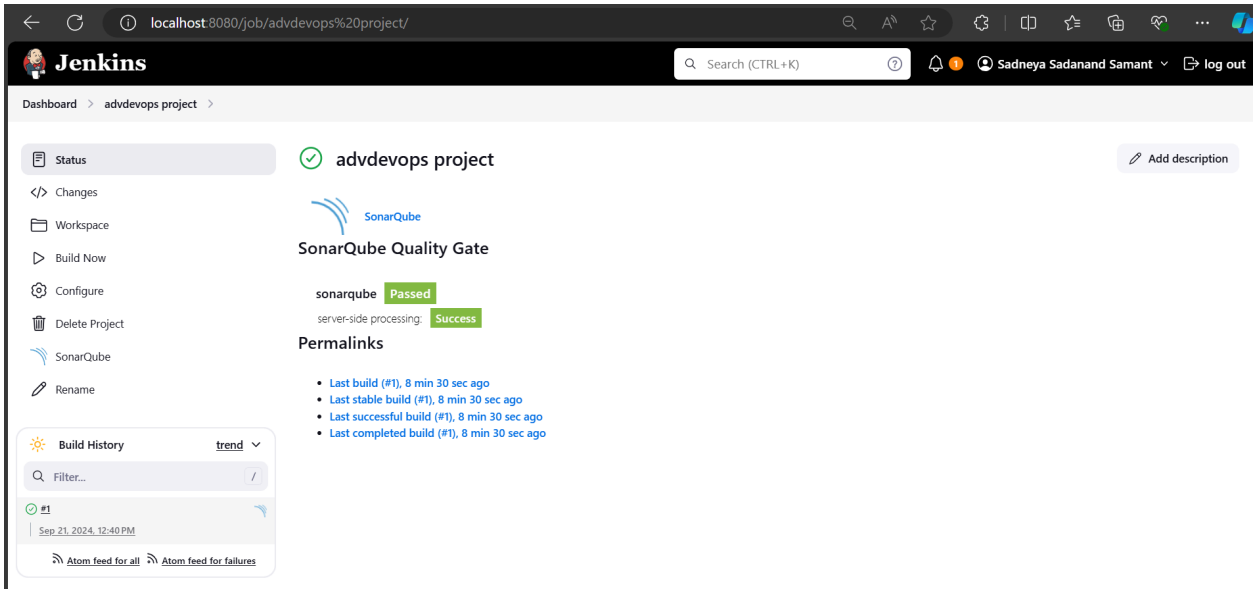





```
12:45:36.578 INFO  14/14 source files have been analyzed
12:45:36.594 INFO  Sensor TextAndSecretsSensor [text] (done) | time=2132ms
12:45:36.605 INFO  ------------ Run sensors on project
12:45:36.814 INFO  Sensor C# [csharp]
12:45:36.814 WARN  Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the
SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
12:45:36.814 INFO  Sensor C# [csharp] (done) | time=10ms
12:45:36.814 INFO  Sensor Analysis Warnings import [csharp]
12:45:36.814 INFO  Sensor Analysis Warnings import [csharp] (done) | time=0ms
12:45:36.814 INFO  Sensor C# File Caching Sensor [csharp]
12:45:36.814 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir'
property.
12:45:36.814 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=0ms
12:45:36.814 INFO  Sensor Zero Coverage Sensor
12:45:36.846 INFO  Sensor Zero Coverage Sensor (done) | time=32ms
12:45:36.846 INFO  SCM Publisher SCM provider for this project is: git
12:45:36.861 INFO  SCM Publisher 4 source files to be analyzed
12:45:39.311 INFO  SCM Publisher 4/4 source files have been analyzed (done) | time=2443ms
12:45:39.311 INFO  CPD Executor Calculating CPD for 0 files
12:45:39.420 INFO  CPD Executor CPD calculation finished (done) | time=0ms
12:45:39.436 INFO  SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
12:45:39.782 INFO  Analysis report generated in 267ms, dir size=199.9 kB
12:45:39.892 INFO  Analysis report compressed in 110ms, zip size=22.4 kB
12:45:42.561 INFO  Analysis report uploaded in 2661ms
12:45:42.563 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
12:45:42.563 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
12:45:42.564 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=b041298d-55f9-4005-844f-c6fce8ccc151
12:45:42.580 INFO  Analysis total time: 28.328 s
12:45:42.583 INFO  SonarScanner Engine completed successfully
12:45:42.637 INFO  EXECUTION SUCCESS
12:45:42.637 INFO  Total time: 43.826s
Finished: SUCCESS
```

14. Then go to SonarQube.Then go inside the project that you created.where it will show output passed.





Thus Project Build successfully.

Conclusion: Here we created the sonarqube project locally successfully.Then created a freestyle project on jenkins and installed a plugin named sonarqube scanner.Then we build that project by firstly adding github repository and mentioning Analysis properties which is nothing but name ,key, token.Then after saving and applying the changes, the build executed successfully.Then we also checked that sonarqube given response about passing of analysis.Thus Build is done successfully.

Name: Sadneya Sadanand Samant Roll No:46 Adv_devops_07

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.

1.Creation of project on Sonarqube

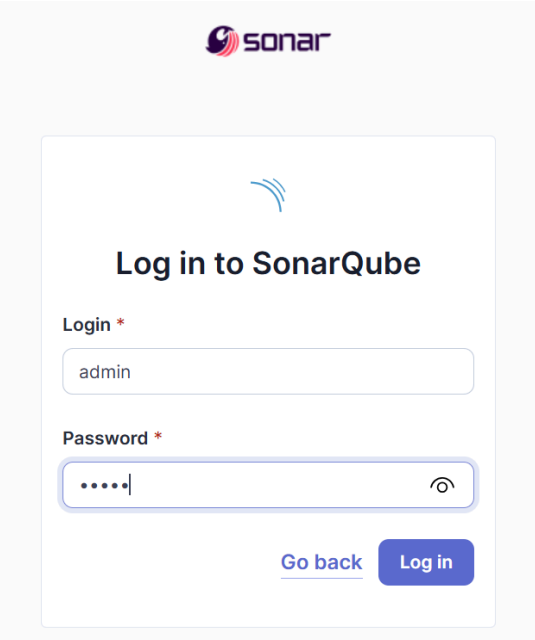
1. Open the command prompt and perform this command

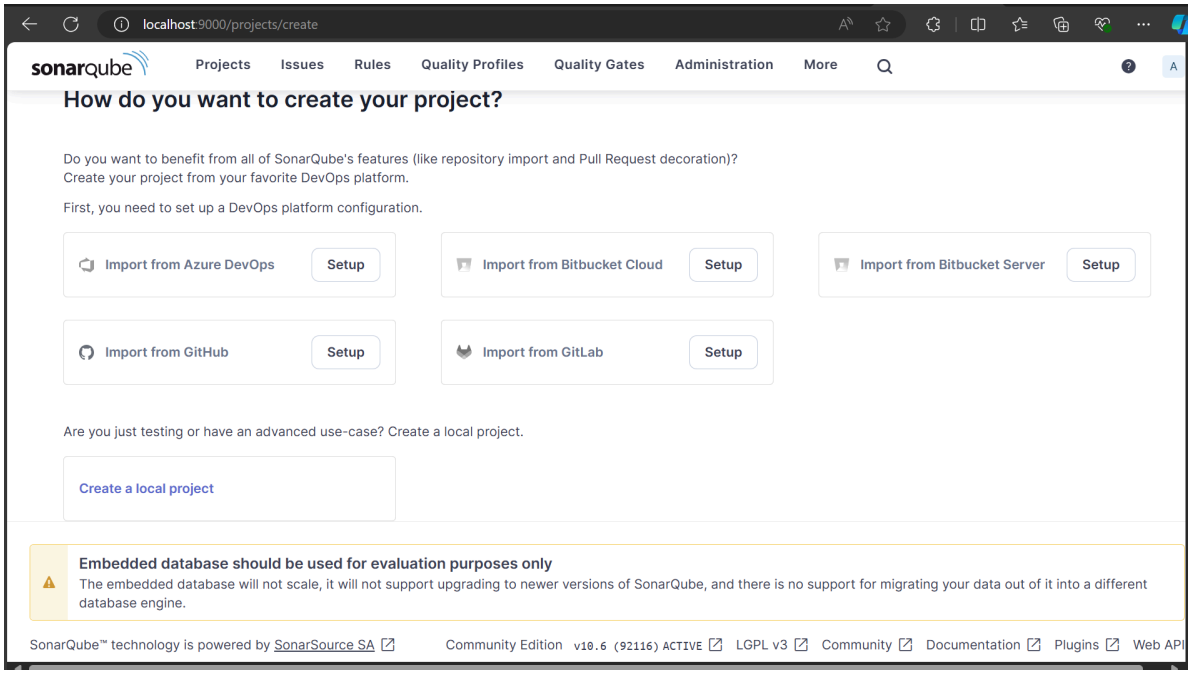
(**Docker must be Installed before running this command.**)

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

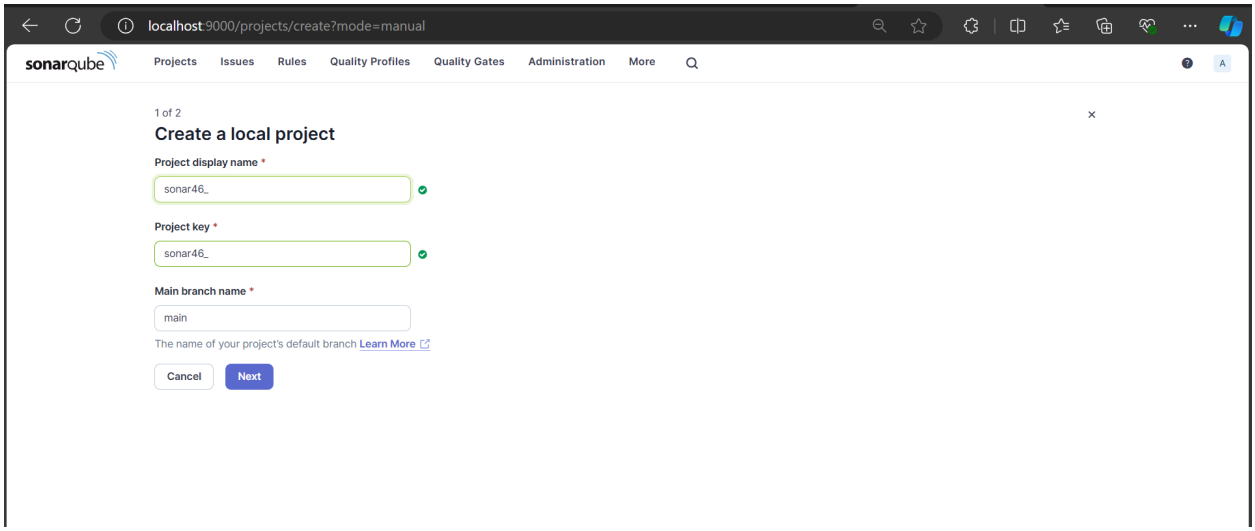
```
PS C:\Users\Sadneya> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
19948c8162691fd8be61d035e355c4fe4a6a2fd0a15237e94f75c0857ff7e2ff
```

2. Then after its successful execution, run the sonarqube at http://localhost:9000
3. Then Login as Username admin and password admin.

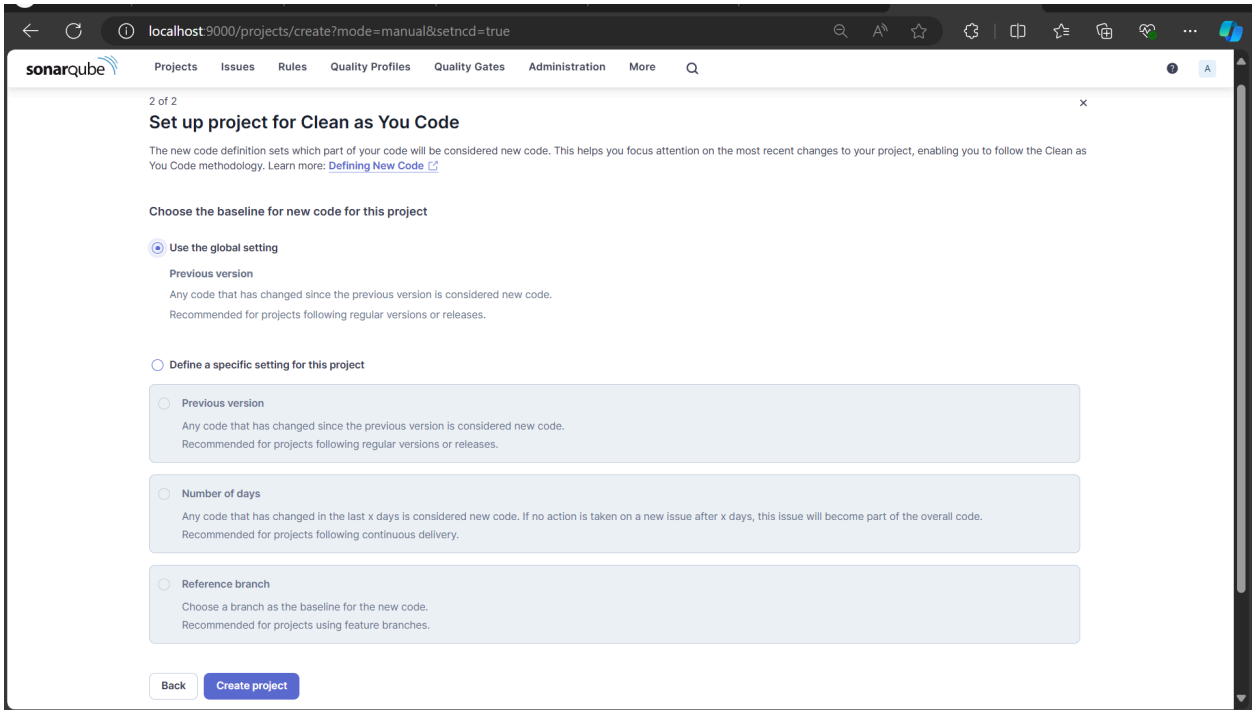




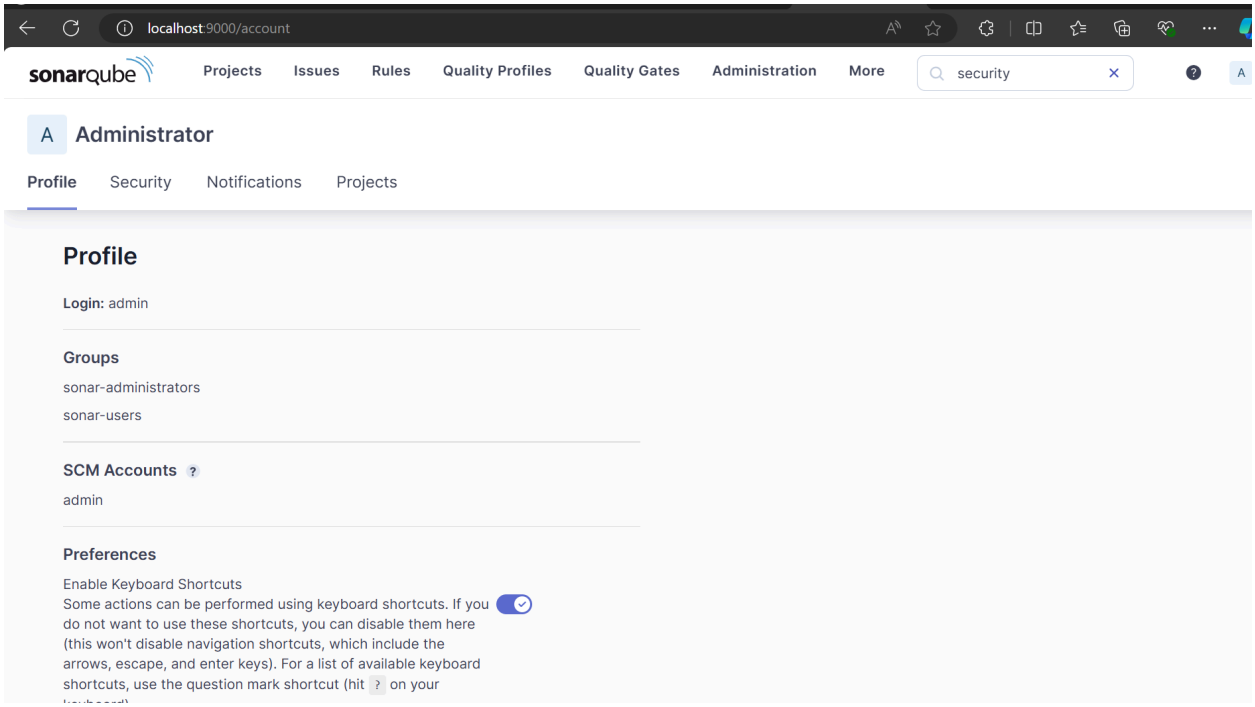
4. Then create a project. Here I have given the name as “sonar46”. Keep branch name main only and then click on next.



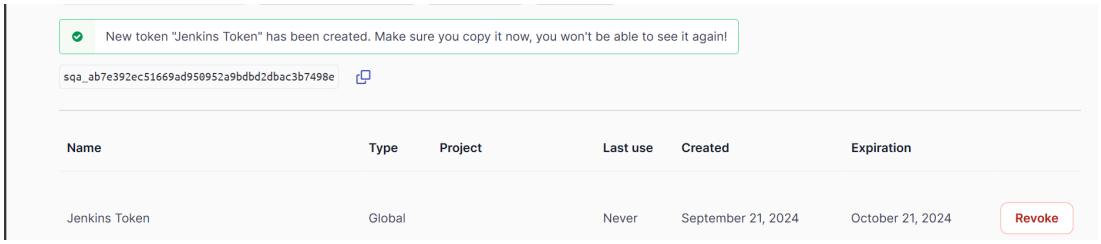
5. Then in the Setup project for clean as you code they will ask to choose the baseline for new code for this project. Choose Use the global setting.Then click on create project.



6. Then click on your account profile and select my account.Inside this click on security option.

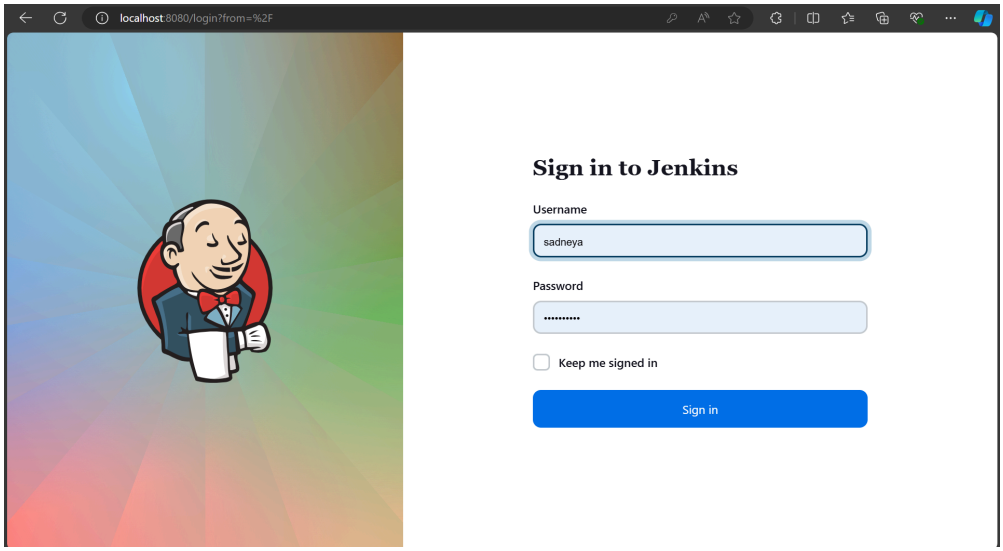


7. Then set the name of your token and here I have chosen the global scope of the token and then clicked on generate token, thus token created successfully.(Here I have given my token name as ‘jenkins token’.)

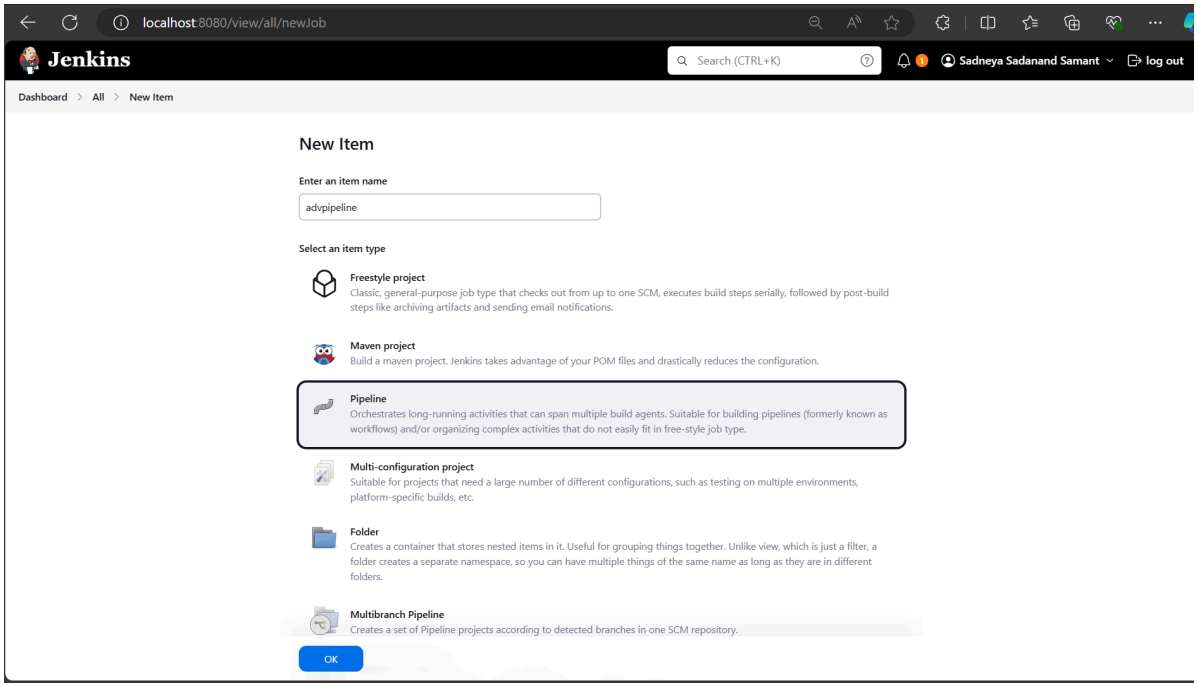


2. Creation and building of jenkins pipeline

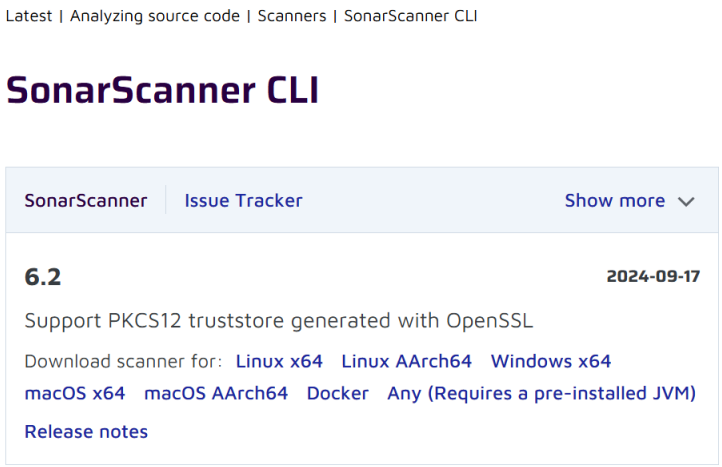
1. Run the jenkins at http://localhost:8080 and enter username and password and click on sign in



2. Then create a pipeline item type in jenkins.

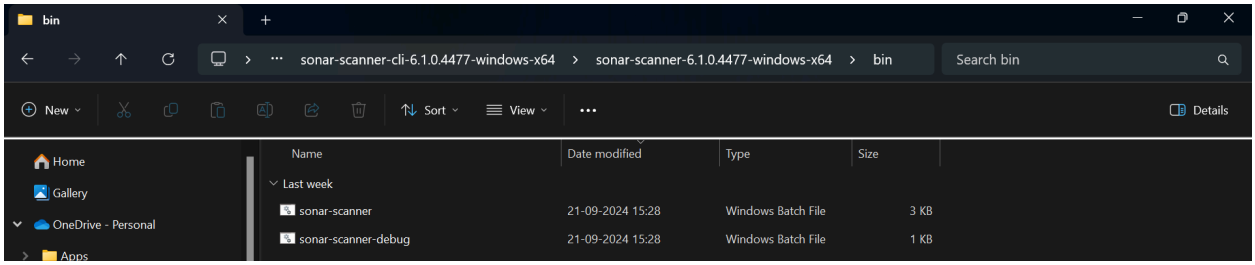


3. Go to SonarScanner CLI (sonarsource.com) this website and download the latest version of sonar scanner on the computer.
4. Then select the machine here I am using windows x64



Once it is downloaded, copy its path as it is required in the pipeline script.

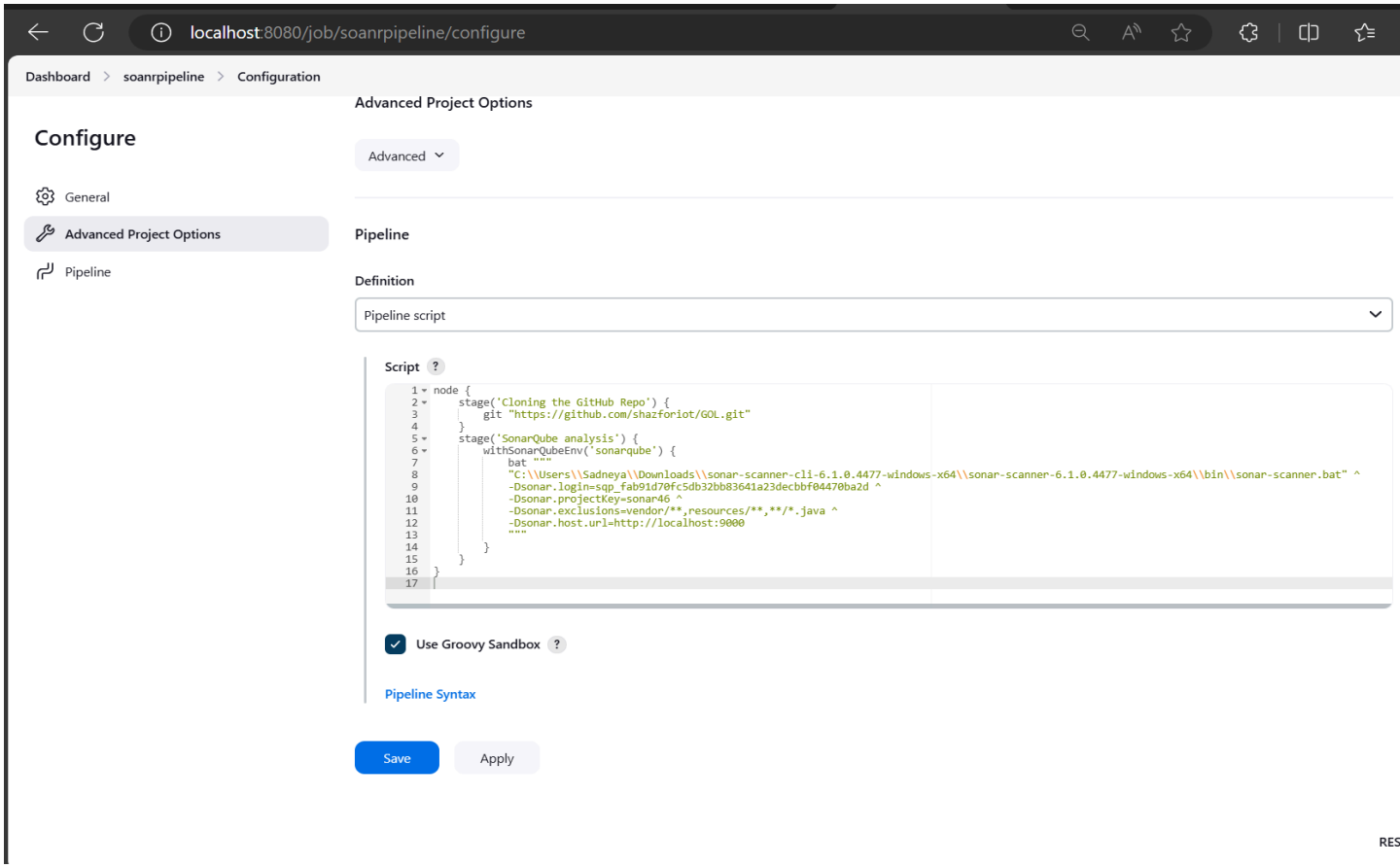
C:\\Users\\Sadneya\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.bat"



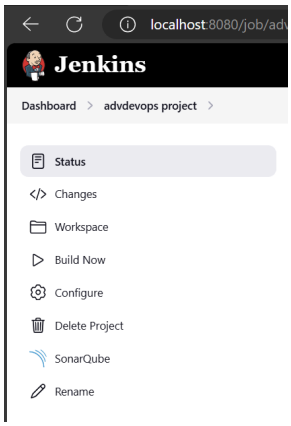
5. Inside the pipeline script section write the following script.

```
node {
   stage('Cloning the GitHub Repo') {
      git "https://github.com/shazforiot/GOL.git"   (cloning this repo)
   }
   stage('SonarQube analysis') {
      withSonarQubeEnv('sonarqube') {
         bat """

"C:\\Users\\Sadneya\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.bat" ^
          -Dsonar.login=sqp_fab91d70fc5db32bb83641a23decbbf04470ba2d ^  (login token)
```

```
            -Dsonar.projectKey=sonar46 ^  (your project name)
            -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^  (libraries u want to exclude)
            -Dsonar.host.url=http://localhost:9000 (server URL)
            """
        }
    }
}
```



6. Then save and apply the changes.Then click on build option.



Then it throws an error.

Console Output

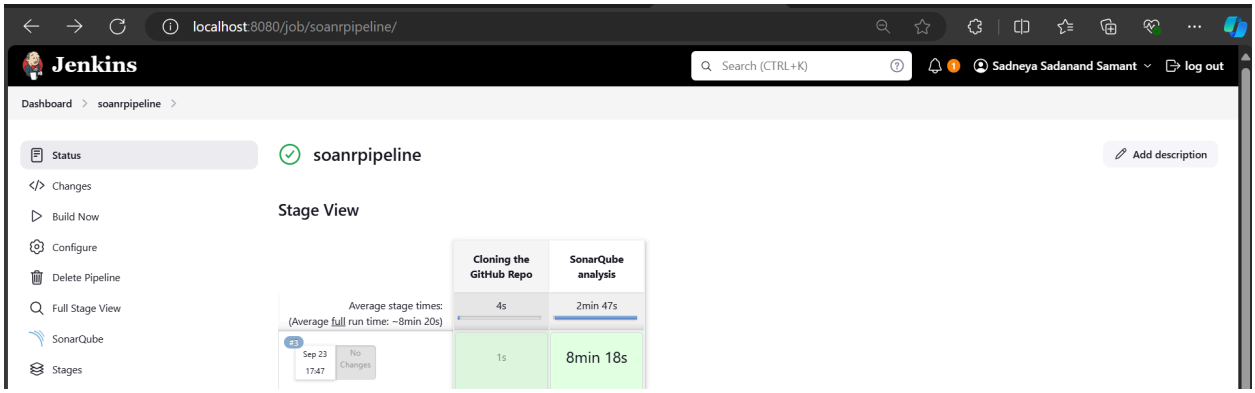
```
Started by user Sadneya Sadanand Samant
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\soanrpipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/GOL.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\soanrpipeline # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
17:42:26.066 INFO  EXECUTION FAILURE
17:42:26.066 INFO  Total time: 0.766s
17:42:26.067 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [http://localhost:9000/api/v2/analysis/jres?os=windows&arch=amd64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
17:42:26.068 ERROR
17:42:26.068 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.
[Pipeline] }
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

This error was due to failure in login and connection. So I generated the token again and replaced it in the pipeline script inside the configure of the pipeline.

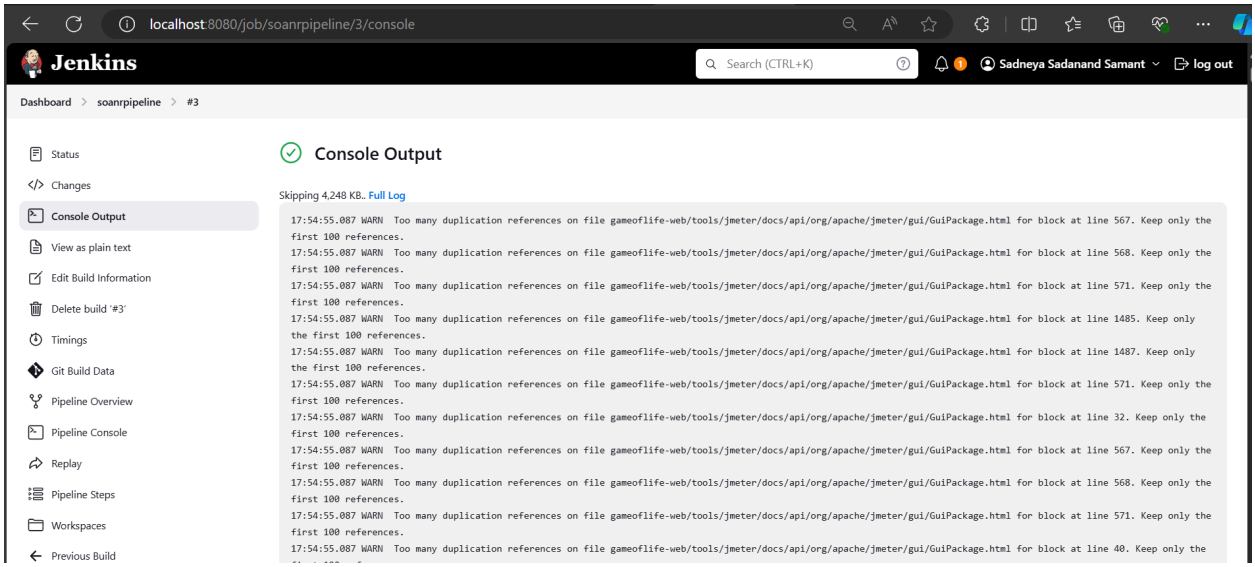
ie. -Dsonar.login=(new token)

7. Then I again clicked on build ,but this time the build got executed successfully.

3. Analysis by sonarqube

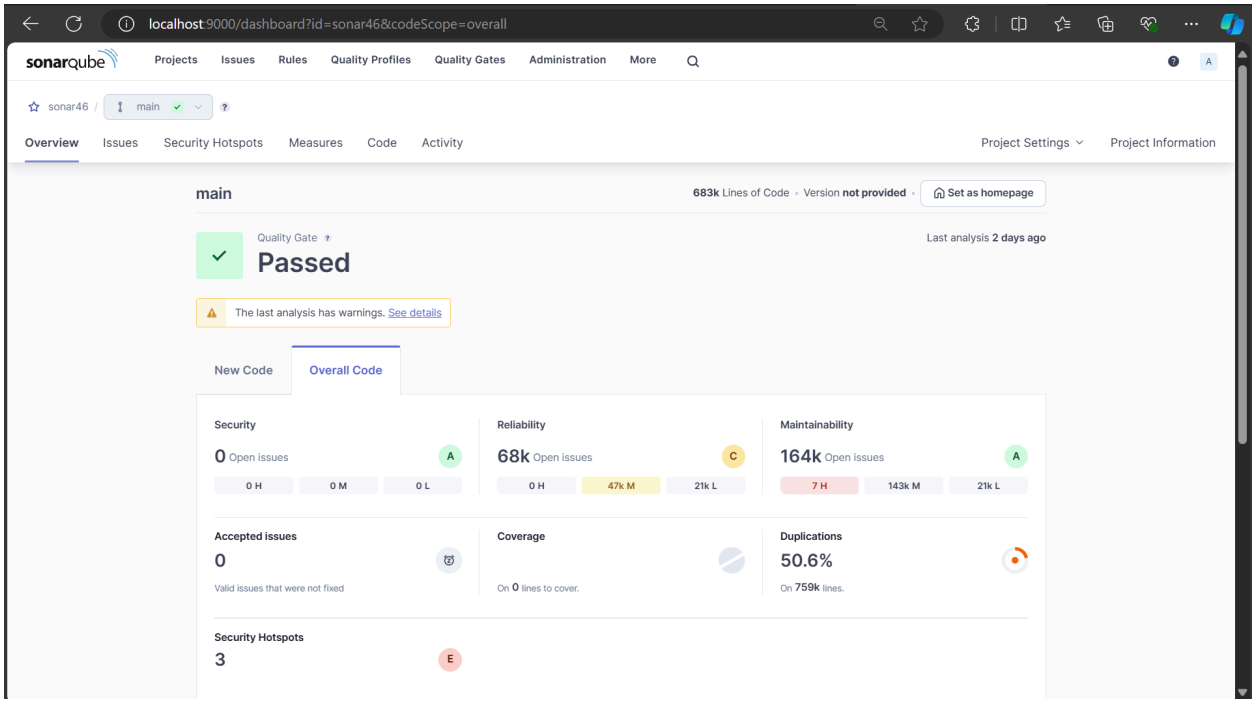


8. Check the Console Output.




```
Keep only the first 100 references.
17:55:00.057 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 154. Keep only the first 100 references.
17:55:00.057 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 555. Keep only the first 100 references.
17:55:00.057 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 154. Keep only the first 100 references.
17:55:00.057 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 555. Keep only the first 100 references.
17:55:00.072 INFO  CPD Executor CPD calculation finished (done) | time=150691ms
17:55:00.338 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
17:55:08.022 INFO  Analysis report generated in 4925ms, dir size=127.2 MB
17:55:27.419 INFO  Analysis report compressed in 19397ms, zip size=29.5 MB
17:55:30.386 INFO  Analysis report uploaded in 2962ms
17:55:30.397 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonar46
17:55:30.397 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
17:55:30.397 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=b0639869-14e8-402a-b7e6-7bd8be121405
17:55:46.751 INFO  Analysis total time: 8:11.934 s
17:55:46.771 INFO  SonarScanner Engine completed successfully
17:55:47.395 INFO  EXECUTION SUCCESS
17:55:47.520 INFO  Total time: 8:15.609s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

9. Then go to SonarQube.Then go inside the project that you created.where it will show output passed.

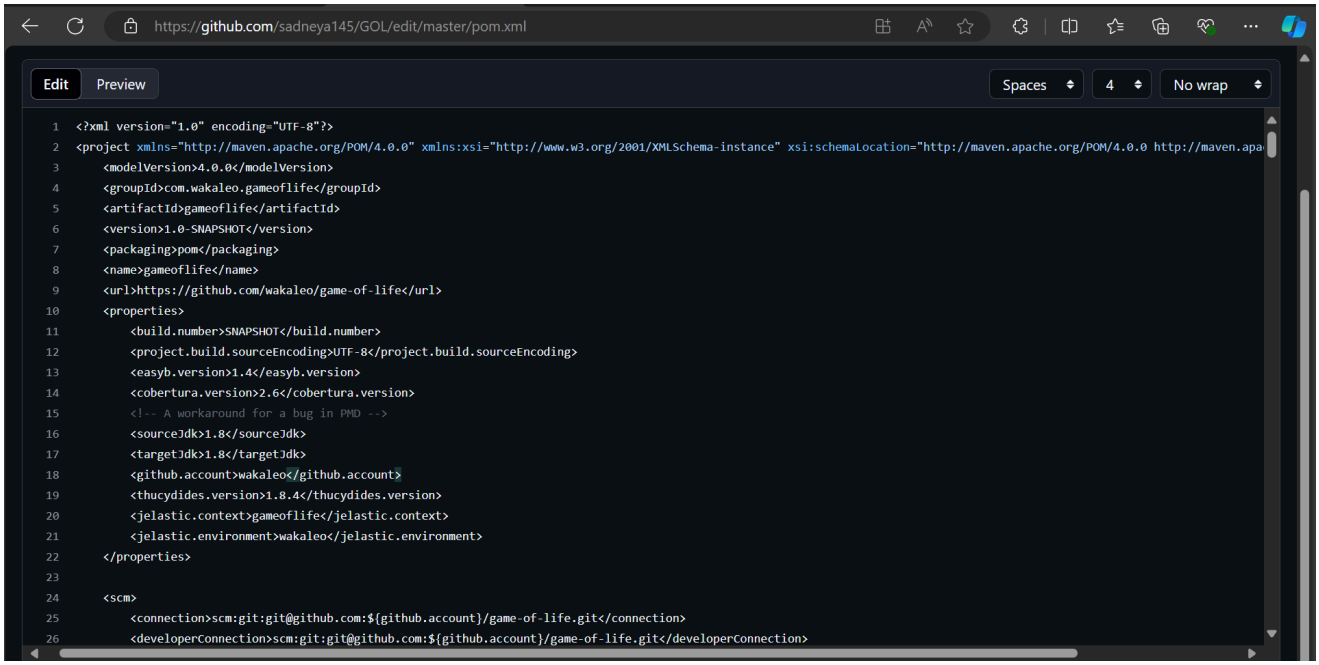


Thus Project Build successfully.

4. Analysis by sonarqube after changing code

1. Then I made the changes in the pom.xml file to see the analysis given by sonarqube. In these I have introduced the hardcoded credentials and an unused variable:

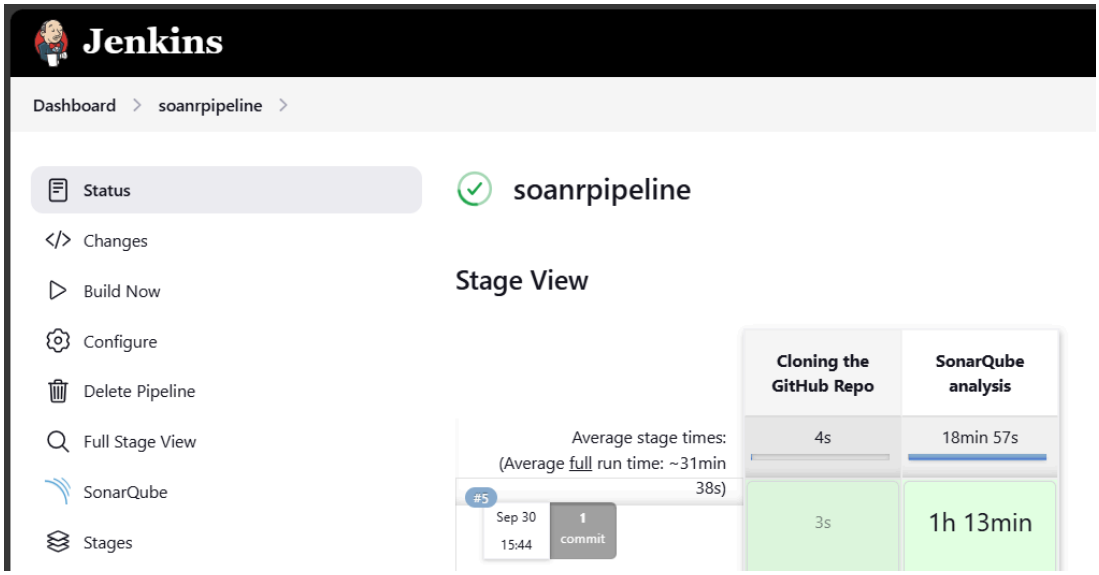
2. I have changed the properties in the pom.xml from



To this

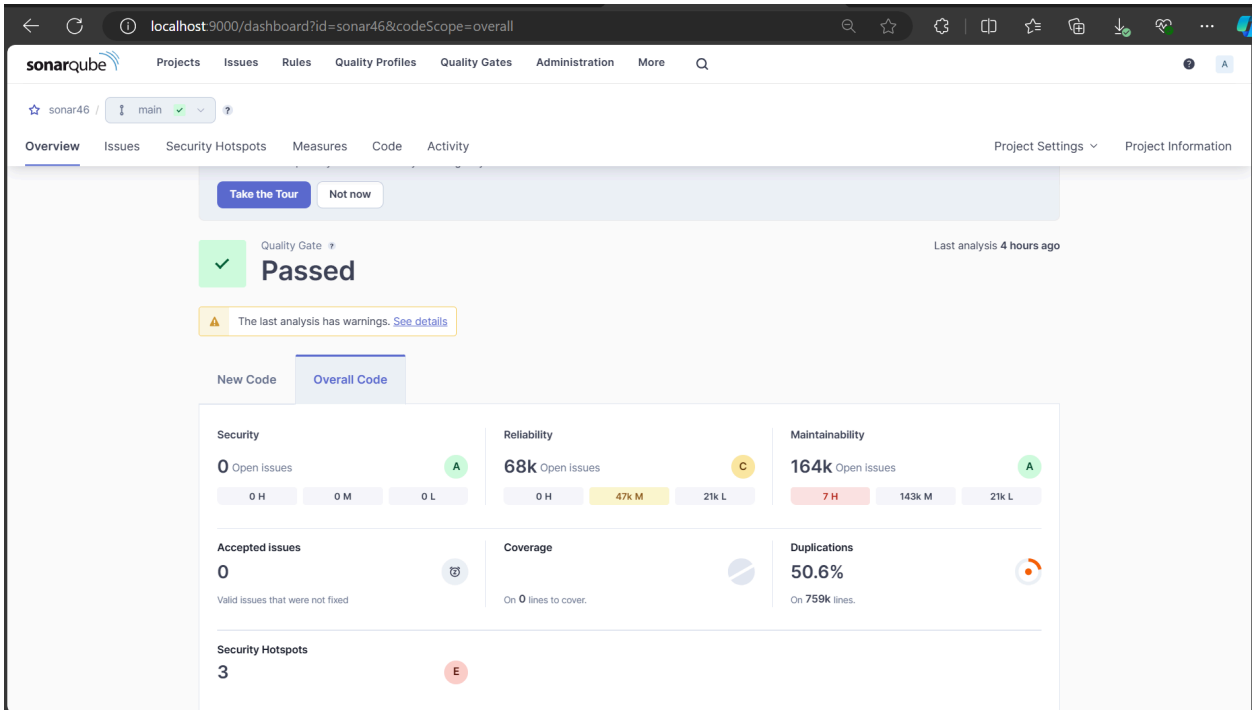
```
<properties>
    <build.number>SNAPSHOT</build.number>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <easyb.version>1.4</easyb.version>
    <cobertura.version>2.6</cobertura.version>
    <!-- A workaround for a bug in PMD -->
    <sourceJdk>1.8</sourceJdk>
    <targetJdk>1.8</targetJdk>
    <github.account>wakaleo</github.account>
    <thucydides.version>1.8.4</thucydides.version>
    <jelastic.context>gameoflife</jelastic.context>
    <jelastic.environment>wakaleo</jelastic.environment>

    <!-- Hardcoded username and password (Security issue) -->
    <jelastic.username>admin</jelastic.username>
    <jelastic.password>password123</jelastic.password>

    <!-- Unused property (Maintainability issue) -->
    <unused.property>ThisIsUnused</unused.property>
</properties>
```

3.Then again I build the code on jenkins.

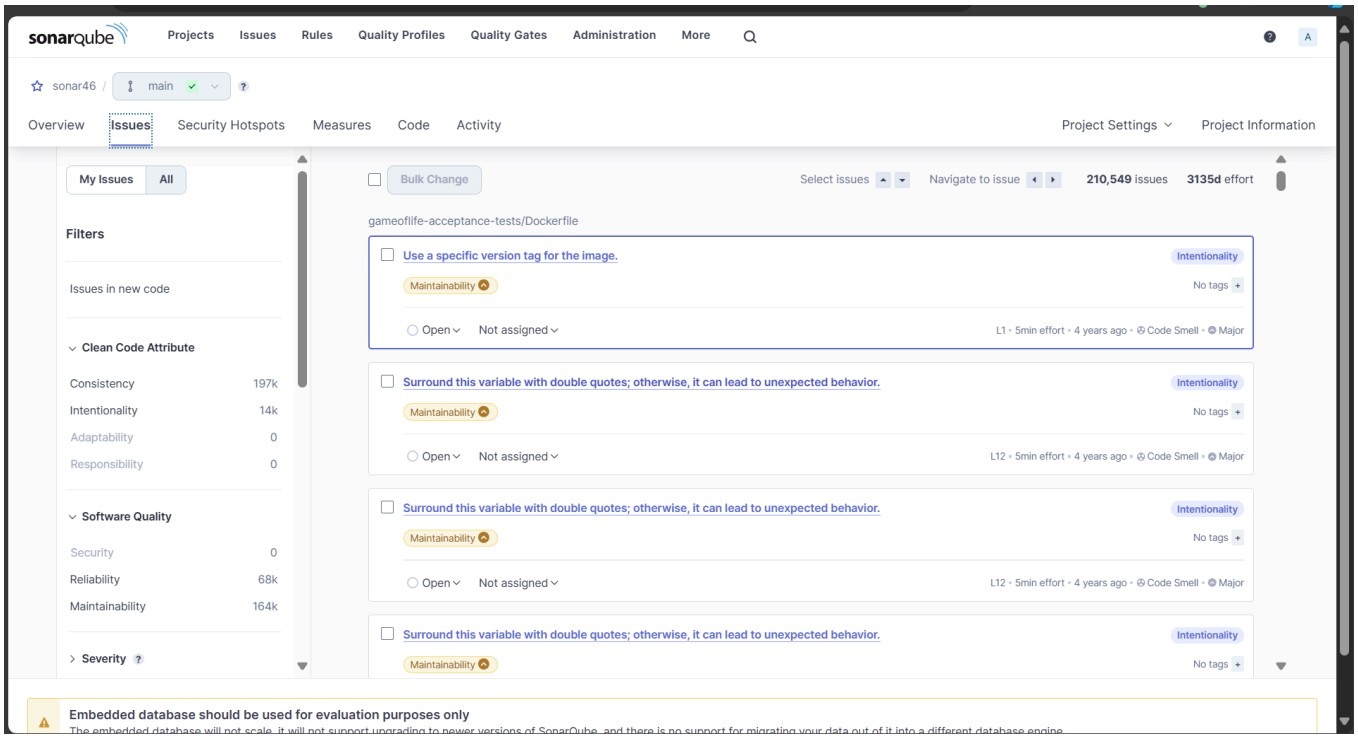


4.Then I again go to the sonarqube to see the analysis it given me output passed

Name: Sadneya Sadanand Samant Roll No:46 Adv_devops_07



But in issues section there are many issues after making changes.



Thus sonarqube analysis is done.

Conclusion: Here we created the “sonarpipeline” project locally successfully.Then created a pipeline on jenkins and installed a plugin named sonarqube scanner.Then we build that project by adding the script inside pipeline which consist of project name ,key, token and then github repository to clone. Then after saving and applying the changes, the build executed successfully.Initially there was an issue in connection to solve this I created the token again and it executed successfully.Then we also checked that sonarqube gives response as passed. Thus, Build is done successfully.Then again I made some changes in pom.xml and again built the code which was also done successfully and then I opened sonarqube which gave me detailed analysis of issues arised due to new changes in code.

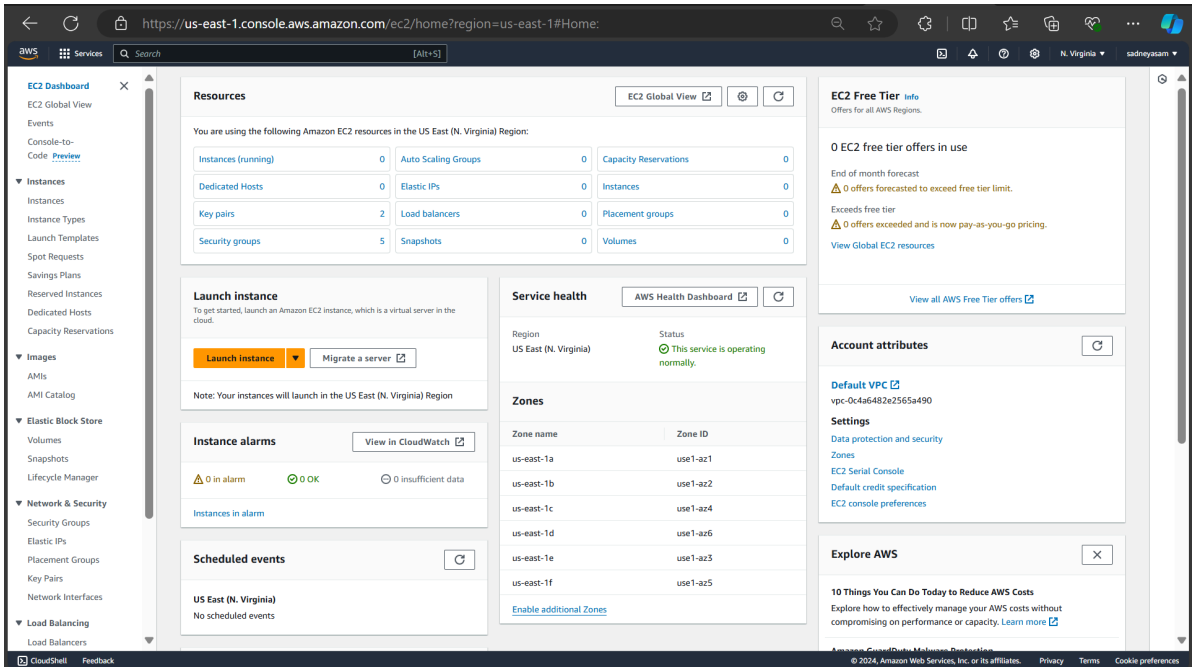
Name:Sadneya Sadanand Samant Roll No:46 Advdevops_9

Experiment No:9

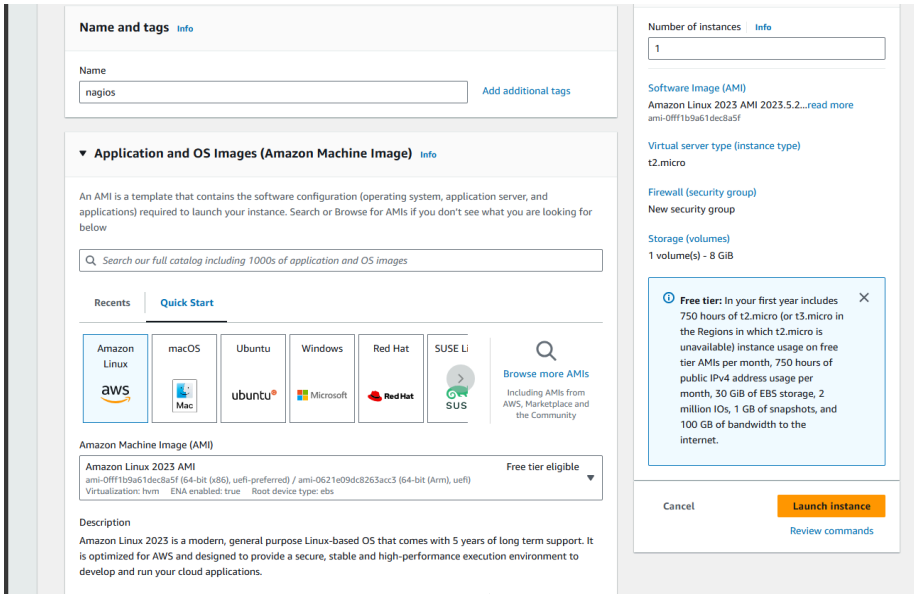
Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Steps:

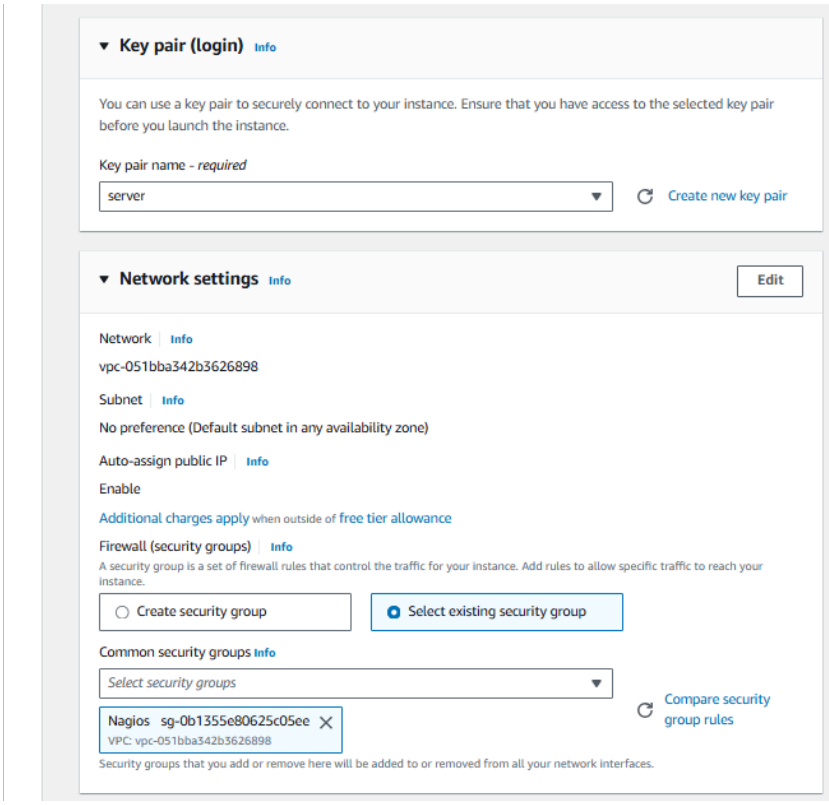
Go to AWS ACADEMY.
On Dashboard, go to EC-2 instance.



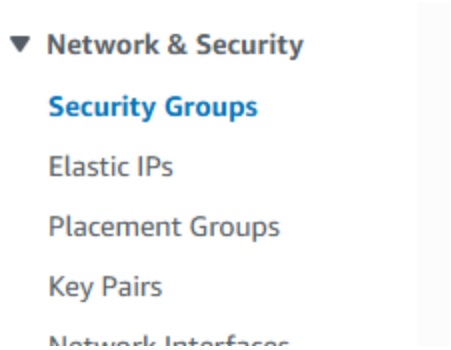
1. **Creation Of EC2 instance:**Create an Amazon Linux EC2 Instance of type t2.micro in AWS and name it - nagios-host



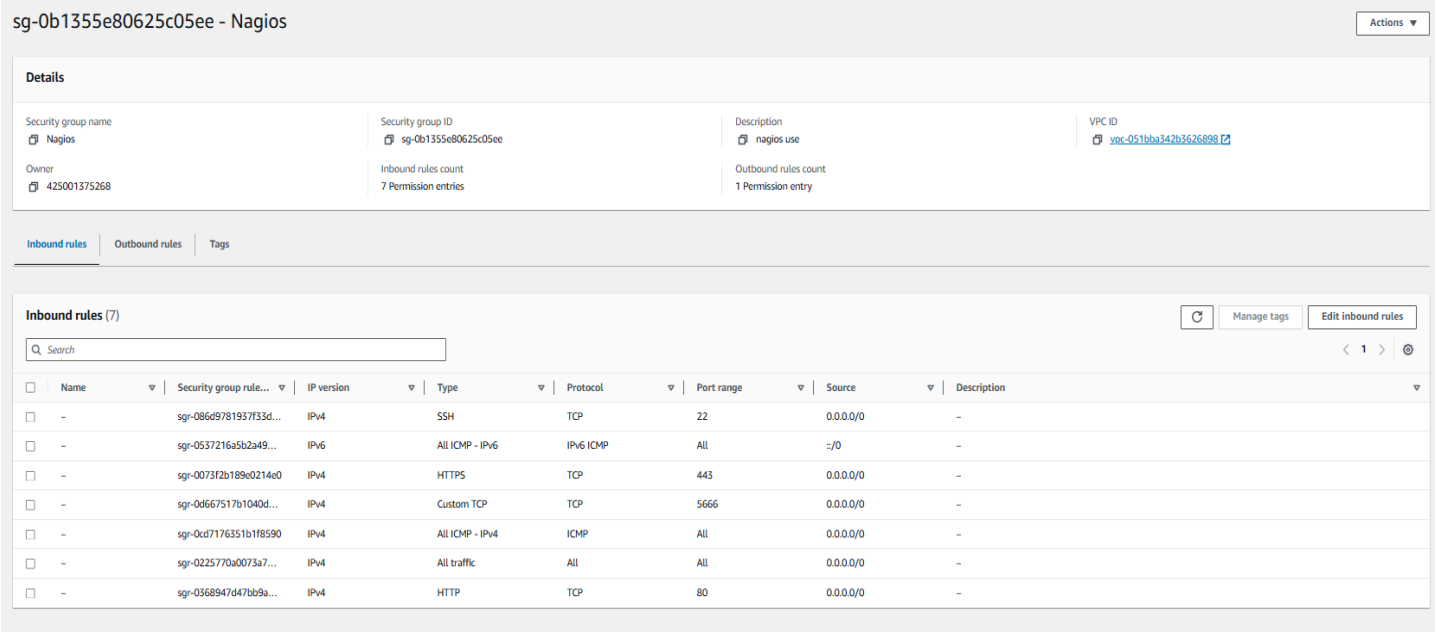
2. Select the existing Key pair or create a new pair.



3. Go back to the EC2 Dashboard there on the left pane, select the security group.



Under Security Group, create a new security group,Give a description to it.Then edit the inbound rules of the specified Security Group for this.add HTTP at port 80, HTTPS at port 443, SSH at port 22, ICMP are open from everywhere.



4.**Making the connection:** Make the connection by SSH into Your EC2 instance.the command is given by **ssh -i "key_name.pem" ec2-user@ec2-public_IP_address.compute-1.awazonaws.com**

```
PS C:\Users\Sadneya\Downloads> ssh -i "server.pem" ec2-user@ec2-18-232-155-134.compute-1.amazonaws.com
The authenticity of host 'ec2-18-232-155-134.compute-1.amazonaws.com (18.232.155.134)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tozb5yML0ekn5Xzfrw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-232-155-134.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

5. **Installing the packages:** Install the following packages:
   1. sudo yum update

```
[ec2-user@ip-172-31-39-90 ~]$ sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
Last metadata expiration check: 0:02:27 ago on Fri Oct  4 03:35:02 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

   2. sudo yum install httpd php
   3. sudo yum install gcc glibc glibc-common
   4. sudo yum install gd gd-devel

```
Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64                    brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
  bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64               cairo-1.17.6-2.amzn2023.0.1.x86_64
  cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64         fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64        fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
  freetype-2.13.2-5.amzn2023.0.1.x86_64                 freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
  gd-2.3.3-5.amzn2023.0.3.x86_64                        gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64            google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch graphite2-1.3.14-7.amzn2023.0.2.x86_64
  graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64          harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64            harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64               langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
  libICE-1.0.10-6.amzn2023.0.2.x86_64                   libSM-1.2.3-8.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64                    libX11-common-1.7.2-3.amzn2023.0.4.noarch
  libX11-devel-1.7.2-3.amzn2023.0.4.x86_64              libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64                    libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64                   libXpm-3.5.15-2.amzn2023.0.3.x86_64
  libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64             libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64                     libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64              libicu-67.1-7.amzn2023.0.3.x86_64
  libicu-devel-67.1-7.amzn2023.0.3.x86_64               libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64       libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64                libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
  libselinux-devel-3.4-5.amzn2023.0.2.x86_64            libsepol-devel-3.4-3.amzn2023.0.3.x86_64
  libtiff-4.4.0-4.amzn2023.0.18.x86_64                  libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64                   libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64                   libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64            pcre2-devel-10.40-1.amzn2023.0.3.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64               pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
  pixman-0.40.0-3.amzn2023.0.3.x86_64                   sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch               xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
```

6.**Create a new user:**Create a new User with its password by following command. Then again retype the password for confirmation.Here,I am creating a user named “nagios”.

sudo adduser -m nagios
sudo passwd nagios

```
[ec2-user@ip-172-31-39-90 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

7.**Create a new User group:** Create a new User group by following command.here I am creating a User group named “nagcmd”.
sudo groupadd nagcmd

```
[ec2-user@ip-172-31-39-90 ~]$ sudo groupadd nagcmd
```

8. Use these commands so that you don’t have to use sudo for Apache and Nagios
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-39-90 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

9. Create a new directory for Nagios downloads

mkdir ~/downloads
cd ~/downloads

```
[ec2-user@ip-172-31-39-90 ~]$ mkdir ~/downloads
cd ~/downloads
```

10. Use wget to download the nagios which is a source zip file by following command.
wget https://go.nagios.org/l/975333/2024-09-17/6kqcx

```
[ec2-user@ip-172-31-39-90 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-04 03:45:50--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.92.120.28, 34.237.219.119, 52.54.96.194, ...
Connecting to go.nagios.org (go.nagios.org)|3.92.120.28|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Dow
utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51--  http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&
=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Do
&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org
t=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: ‘6kqcx’

6kqcx                          100%[=====================================================>]   1.97M  7.32MB/s    i

2024-10-04 03:45:51 (7.32 MB/s) - ‘6kqcx’ saved [2065473/2065473]

[ec2-user@ip-172-31-39-90 downloads]$
```

11.Then install nagios plugin by following command.

Wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-39-90 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 03:47:43--  http://nagios-plugins.org/download/nagios-plugins-2.4.11.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: 2753049 (2.6M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.4.11.tar.gz’

nagios-plugins-2.4.11.tar.gz      100%[=====================================================>]   2.62M

2024-10-04 03:47:44 (6.85 MB/s) - ‘nagios-plugins-2.4.11.tar.gz’ saved [2753049/2753049]
```

12. Use tar to unzip the file and then go inside that directory.
tar zxvf 6kqcx

```
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
```

cd nagios-4.5.5

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-4.5.5
```

13. Run the configuration script with the same group name you previously created.
./configure --with-command-group=nagcmd

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
```

This given error as can’t find ssl headers

```
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

14. For this install the following packages to solve the above error.

Sudo yum install openssl-devel

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:16:59 ago on Fri Oct  4 03:35:02 2024.
Dependencies resolved.
================================================================================================================================================
 Package                       Architecture          Version                                  Repository                    Size
================================================================================================================================================
Installing:
 openssl-devel                 x86_64                1:3.0.8-1.amzn2023.0.14                  amazonlinux                  3.0 M

Transaction Summary
================================================================================================================================================
Install  1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm                                   19 MB/s | 3.0 MB     00:00
------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                            15 MB/s | 3.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                       1/1
  Installing       : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                          1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                          1/1
  Verifying        : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                          1/1

Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64

Complete!
```

15.The again Run the configuration script with the same group name you previously created. Now the error has been removed.

./configure --with-command-group=nagcmd

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for getopt.h... yes
checking for grp.h... yes
```

```
Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
            Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  /run/nagios.lock
   Check result directory:  /usr/local/nagios/var/spool/checkresults
          Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/httpd/conf.d
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                HTML URL:  http://localhost/nagios/
                 CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/bin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

16.Then Install the make (binaries), then initialize the script and sample config files. Lastly, set permissions on the external command directory.

1. sudo make install
2. sudo make install-init
3. sudo make install-config
4. sudo make install-commandmode

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
        /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
```

```
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc
```

```
*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

17.Then Edit the configuration file and change the email address.
sudo nano /usr/local/nagios/etc/objects
cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
```

18. Configure the web interface by following command.
sudo make install-webconf

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ 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 ***
```

19. Create a “nagiosadmin” account for “nagios” login along with password. You’ll have to specify the password and then again retype the password.

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

20. Then again restart apache by following command: sudo service httpd restart

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

21. Then again go back to the downloads folder and unzip the plugins zip file.
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
nagios-plugins-2.4.11/gl/m4/
```

22.To Start Nagios.Firstly,Add Nagios to the list of system services
sudo chkconfig --add nagios

This given error as it can't find the directory.

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
```

23. Firstly check the configuration by this command
sudo chkconfig nagios on
24.Then Verify the sample configuration files by following command.
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
ec2-user@ip-172-31-39-90:~
[ec2-user@ip-172-31-39-90 ~]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 1 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0
```

This command executed successfully with no errors and no warnings.

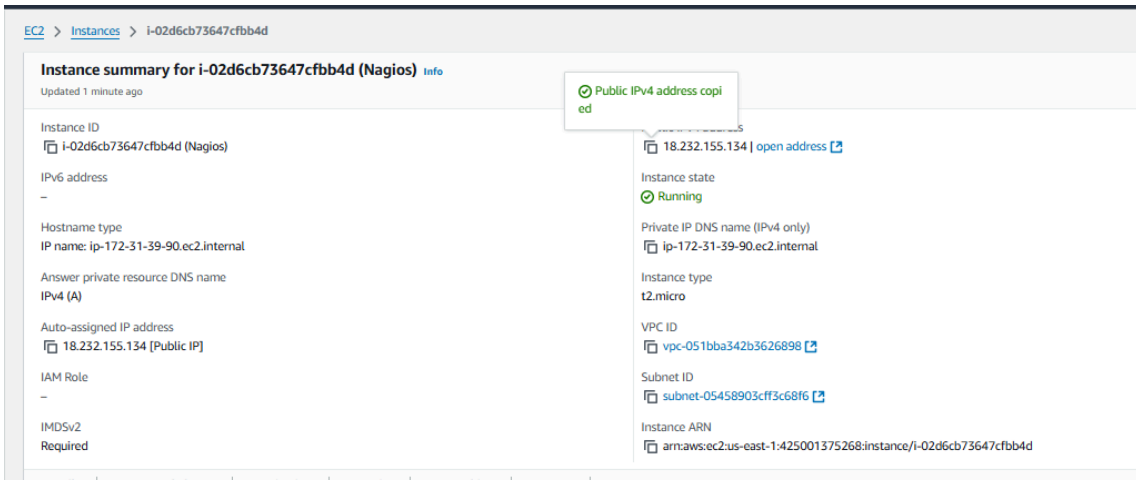
25. **Starting Nagios:** Start Nagios by following command
sudo service nagios start

```
[ec2-user@ip-172-31-39-90 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

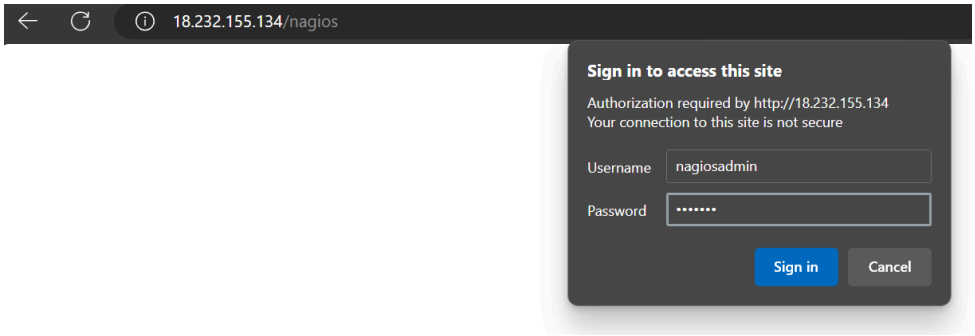
26.**Checking the status:** Then Check the status of Nagios by following command
sudo systemctl status nagios

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 1min 41s ago
       Docs: https://www.nagios.org/documentation
    Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 75300 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.6M
        CPU: 89ms
     CGroup: /system.slice/nagios.service
             ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Successfully registered manager as @wproc with query handler
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75301;pid=75301
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75302;pid=75302
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75304;pid=75304
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75303;pid=75303
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: Successfully launched command file worker with pid 75305
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(/>
Oct 04 04:15:06 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;1;(No output on stdo>
Oct 04 04:15:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(/>
Oct 04 04:15:44 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Users;CRITICAL;HARD;1;(No output on std>
[ec2-user@ip-172-31-39-90 ~]$
```

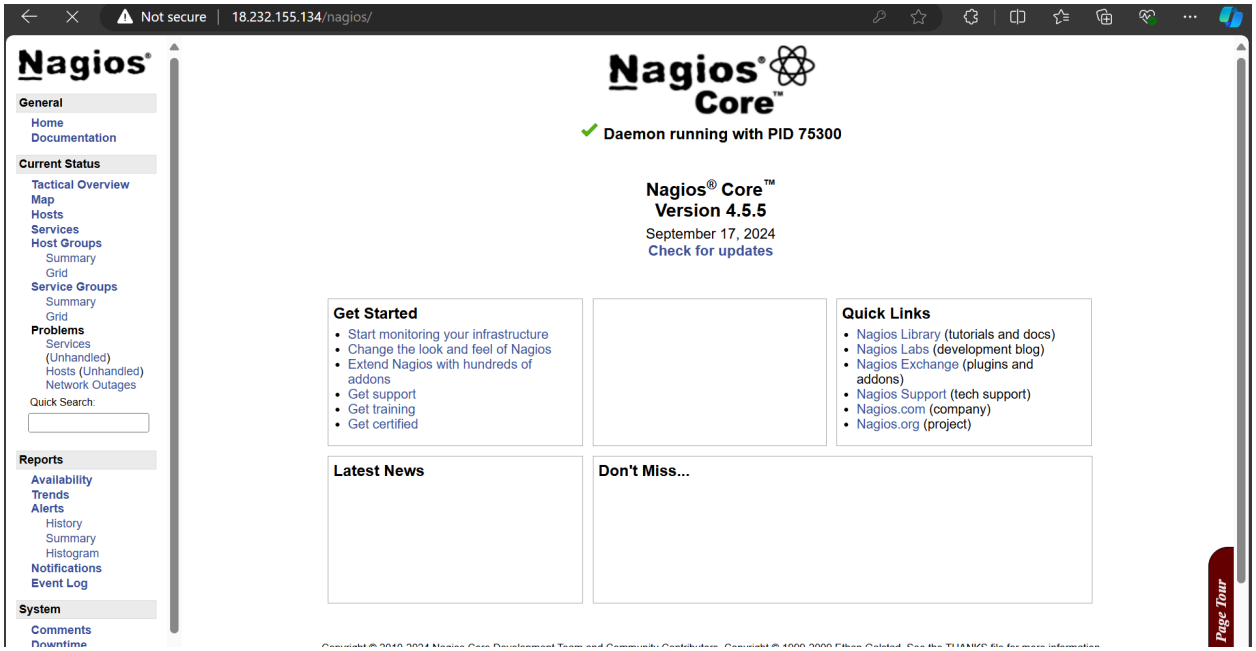
27. Then Go back to EC2 Console and copy the Public IP address of this instance.



28. Open up your browser and look for **http://<your_public_ip_address>/nagios (http://18.232.155.134/nagios)**
Enter username as “nagiosadmin” and password which you set in Step 19.



29. After entering the correct credentials, you will see this page.



This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion: Here,I have created an EC2 instance of t2.micro successfully.For EC2 instance I have created a security group nagios where I have added inbound rules.Then I have created user name “nagios” and user group “nagcmd”. Then I installed nagios and also nagios plugin successfully.Then I configured it but gave an error about can’t find ssl headers.Then I installed the required packages.Then it configured successfully.Then configured the web interfaces.Then I have started the nagios on the **http://18.232.155.134/nagios.** It given the final dashboard of nagios successfully

Experiment No:10

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

Prerequisites:
AWS Academy or Personal account.
Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

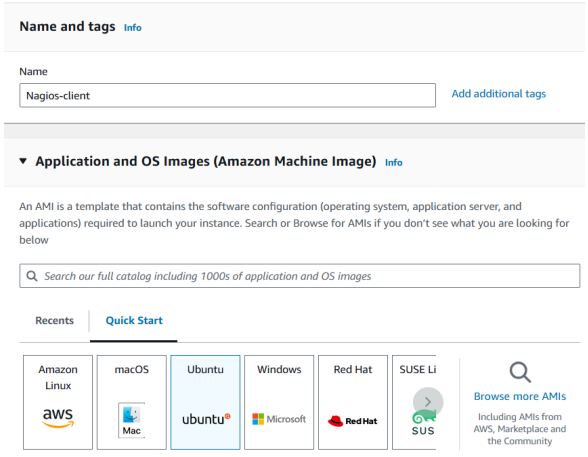
Monitoring Using Nagios:

1. Firstly,Confirm nagios Host is running or not by checking its status by following command.
   sudo systemctl status nagios

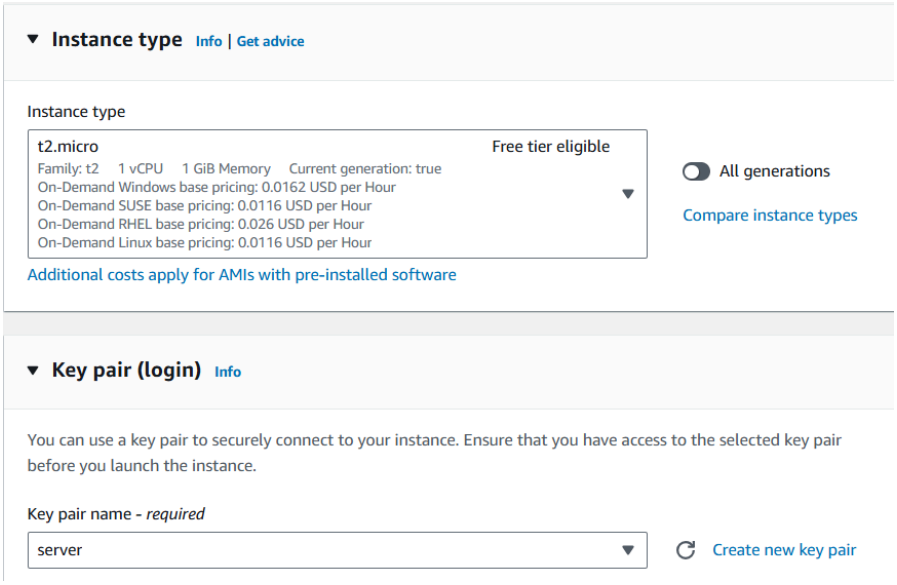
```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 9min ago
       Docs: https://www.nagios.org/documentation
    Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 75300 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.6M
        CPU: 164ms
     CGroup: /system.slice/nagios.service
             ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:17:36 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Root Partition;CRITICAL;HARD;1;(No output on st>
Oct 04 04:18:14 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;SSH;CRITICAL;HARD;1;(No output on stdout) stder>
Oct 04 04:18:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;5;(No output on stdout) stderr: execvp(/>
Oct 04 04:18:51 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;1;(No output on stdout>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Total Processes;CRITICAL;HARD;1;(No output on s>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(/>
Oct 04 04:20:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;7;(No output on stdout) stderr: execvp(/>
Oct 04 04:21:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(/>
Oct 04 04:22:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(/>
Oct 04 04:23:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(>
lines 1-28/28 (END)
```

You can now proceed ahead if you get the above message/output.If not then again create a instance (Refer Experiment No 9)

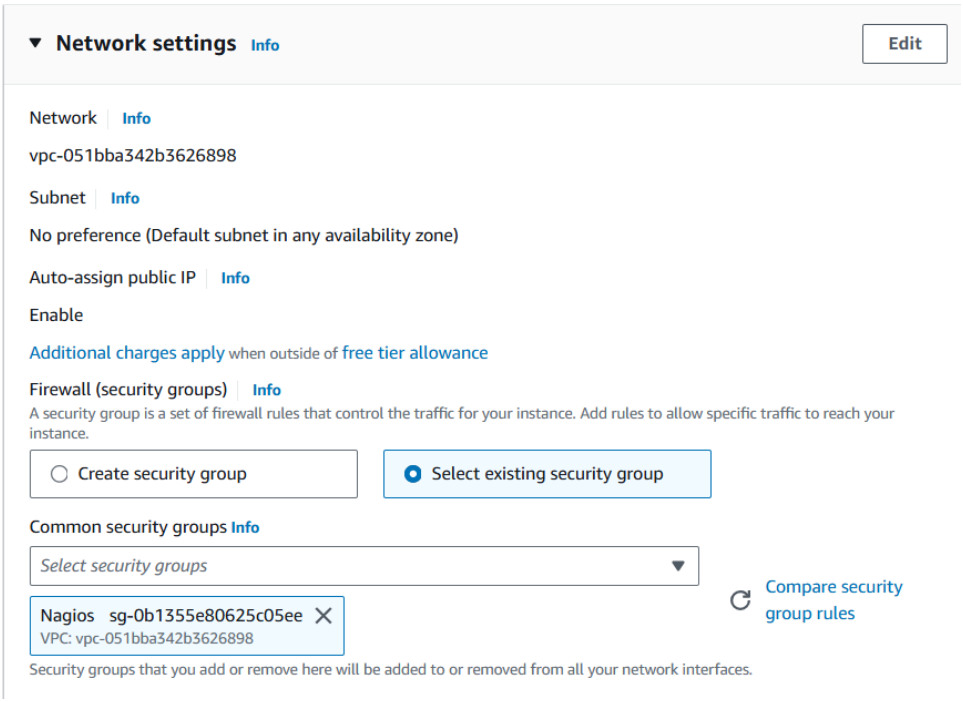
2. **Creation an EC-2 Instance:**Create an EC-2 Instance of t2.micro type on ubuntu in AWS.



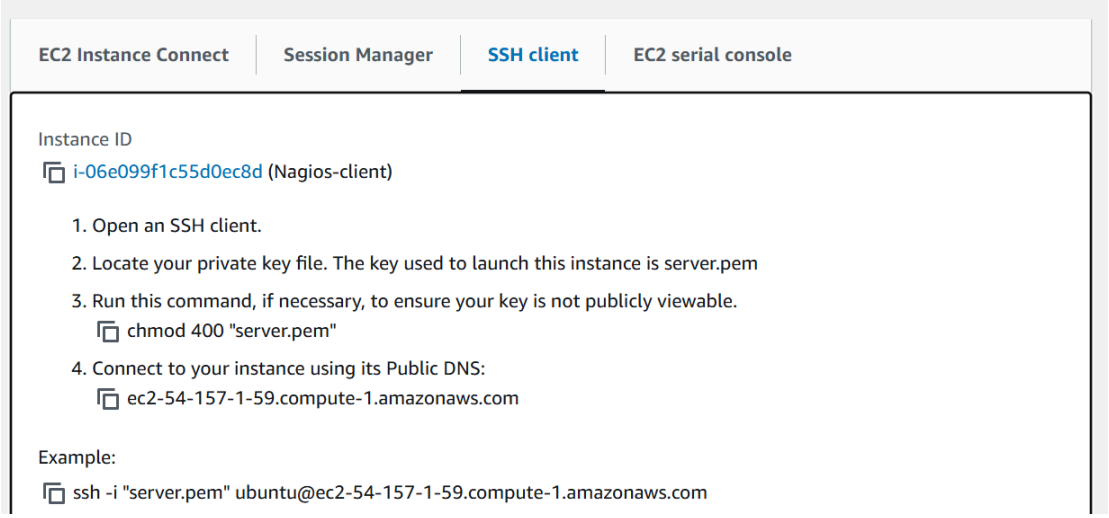
3. Then select the key pair that you have created and used in “nagios-host” EC-2 instance.



4. Then select the security group that we have created in nagios host.here I have given name to security groups as Nagios.



5. **Making the connection:**Then after instance get created successfully then click on connect then go to SSH client section there you will see connection command **ssh -i “key_pair.pem” ubuntu@ec2-Public_ip.compute-1.amazonaws.com**



Copy paste this command on command prompt.

```
PS C:\Users\Sadneya\downloads> ssh -i "server.pem" ec2-user@ec2-3-81-91-101.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-91-101.compute-1.amazonaws.com (3.81.91.101)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tozb5yML0ekn5Xzfrw.
This host key is known by the following other names/addresses:
    C:\Users\Sadneya/.ssh/known_hosts:68: ec2-18-232-155-134.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-91-101.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Oct  4 03:36:10 2024 from 125.99.93.18
```

****Now perform Following commands on Nagios-host****

6. Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-31-39-90 ~]$ ps -ef | grep nagios
ec2-user    2377    2350  0 09:15 pts/0    00:00:00 grep --color=auto nagios
```

7. Now become the root user by sudo su and create the directories.

sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[ec2-user@ip-172-31-39-90 ~]$ sudo su
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

8. Copy the localhost.cfg file to linuxserver.cfg by following command.

cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

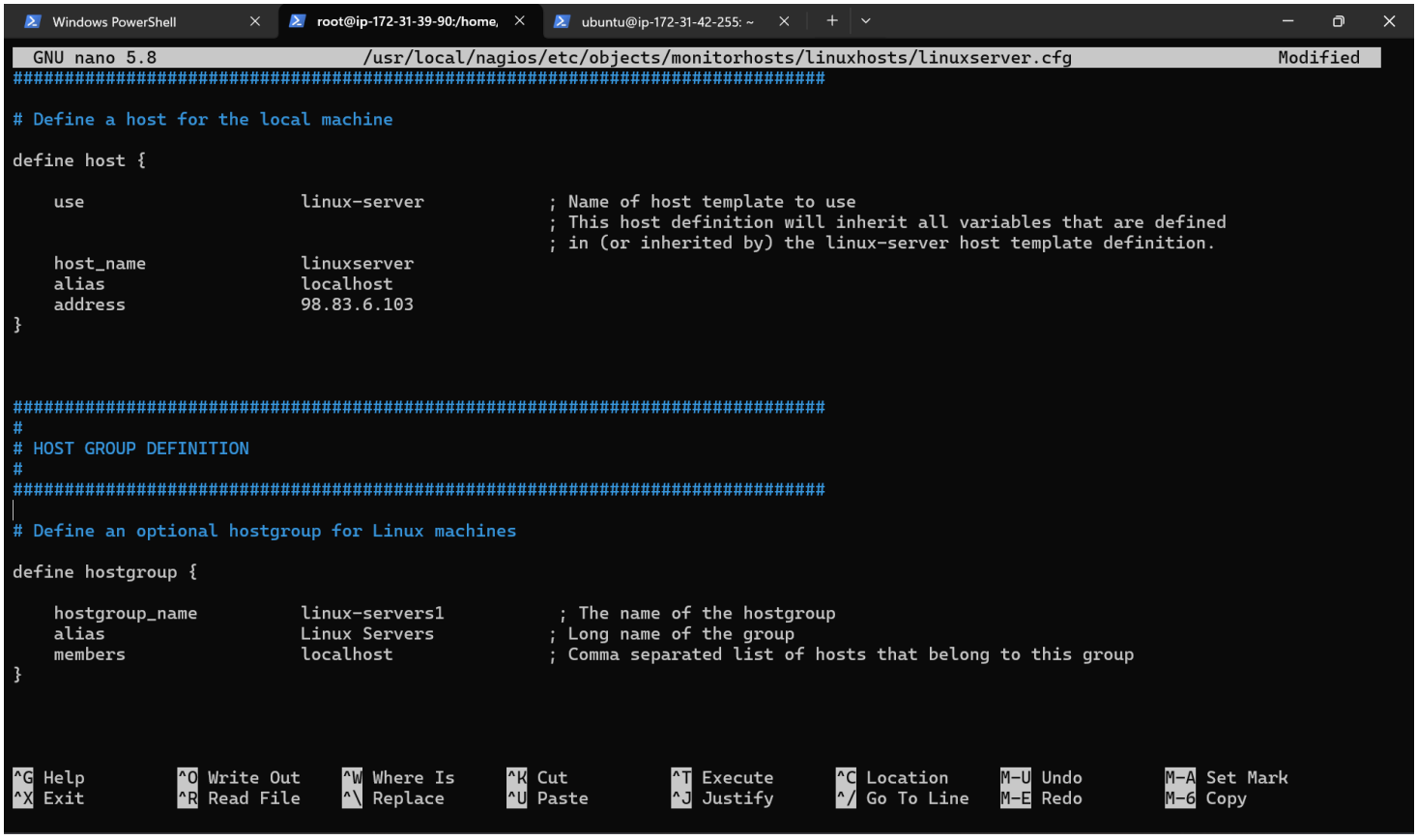
```
[root@ip-172-31-39-90 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linux
hosts/linuxserver.cfg
```

9. Open the linuxserver.cfg by nano command and then make following changes.

nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

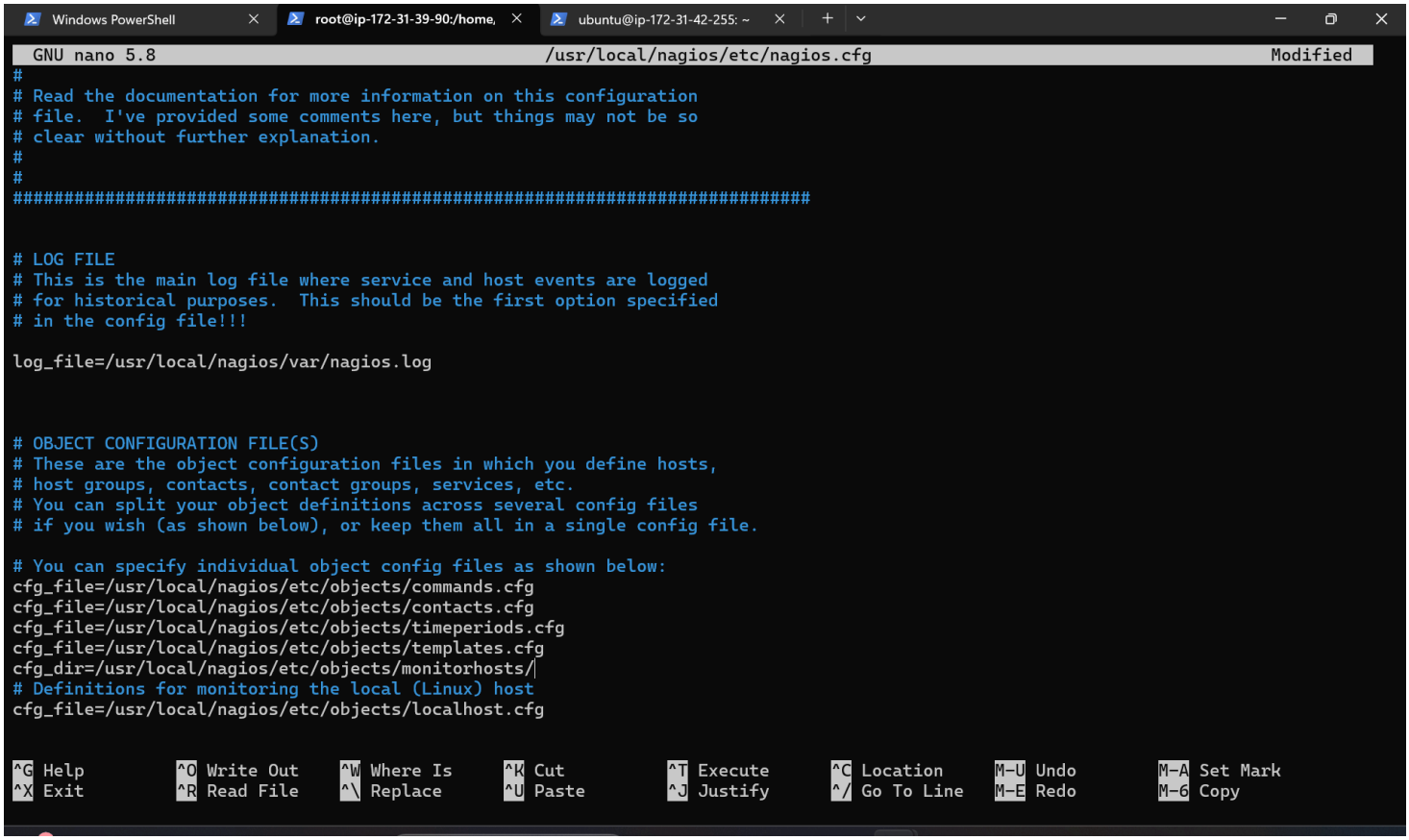
```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-39-90 ec2-user]#
```

Now change the hostname to **linuxserver**. Then Change the address to the public IP of your Linux client. Now Set hostgroup_name to **linux-servers1**.

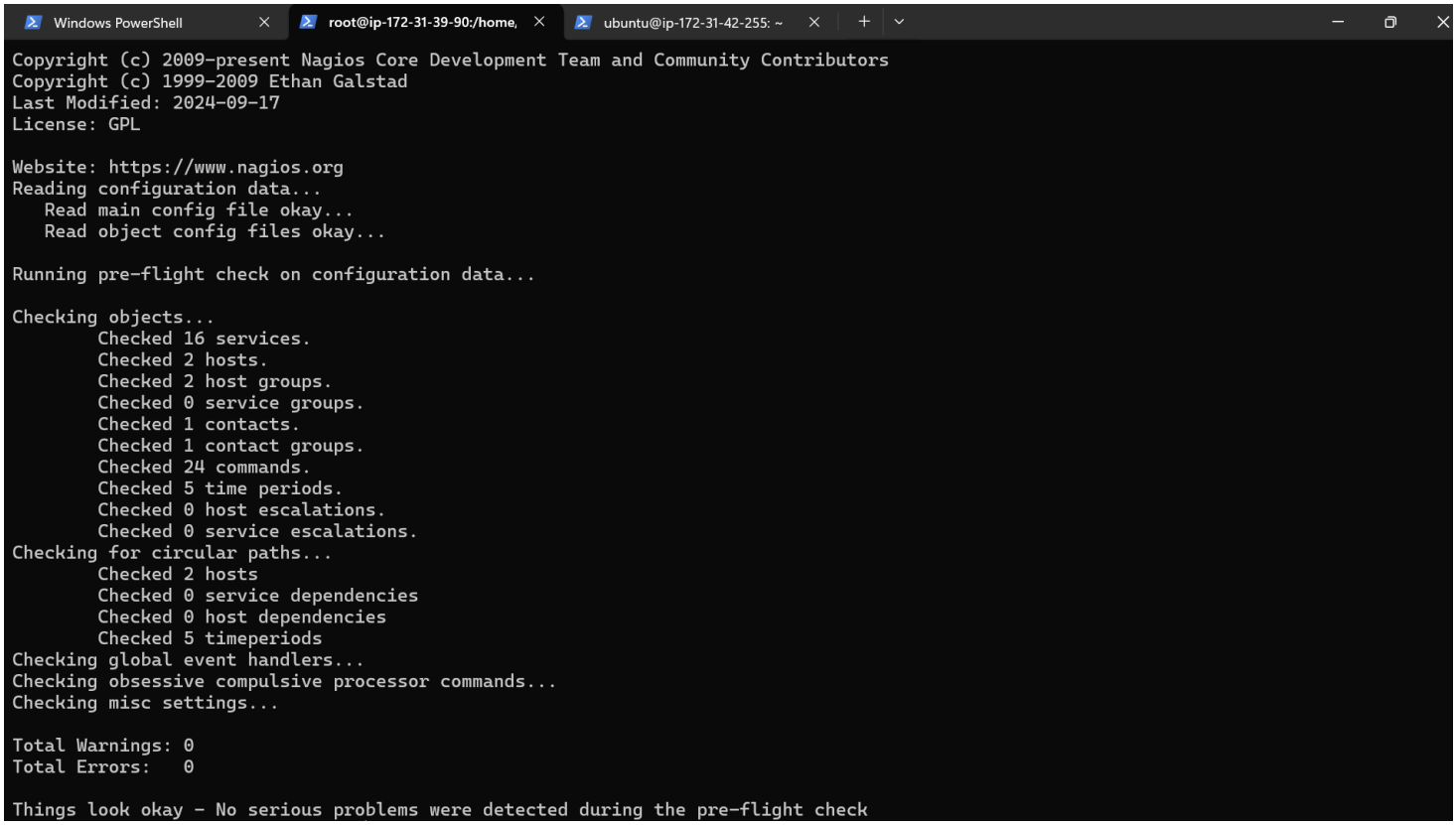


Make the hostname changes throughout in Host,Host Group,HTTP,SSH.

10. Now update the Nagios.config file add the following line in it.
    cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
    Then execute the following command : **nano /usr/local/nagios/etc/nagios.cfg**



11. Now verify the configuration by the following command.
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



12. Now restart the services of nagios by running the following command.

service nagios restart

```
[root@ip-172-31-39-90 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

****Now perform Following commands on Nagios-host****

13. Firstly update the package by following command.

sudo apt update -y

```
ubuntu@ip-172-31-42-255:~$ sudo apt update -y
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 [382 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

```
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:44 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 (4883 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

14. Now,Install the required packages by executing following commands.

sudo apt install gcc -y

```
ubuntu@ip-172-31-42-255:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
  fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1
  libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
  libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
  libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
  libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
  binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc
  gcc-13-multilib gcc-13-doc gdb-x86-64-linux-gnu glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec
  libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e
  libheif-plugin-svtenc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
  fonts-dejavu-core fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8
  libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
  libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
  libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
  libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libsframe1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbinutils amd64 2.42-4ubuntu2 [572 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]
Setting up gcc (4:13.2.0-7ubuntu1) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.15.0-1.1ubuntu2) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
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.
ubuntu@ip-172-31-42-255:~$
```

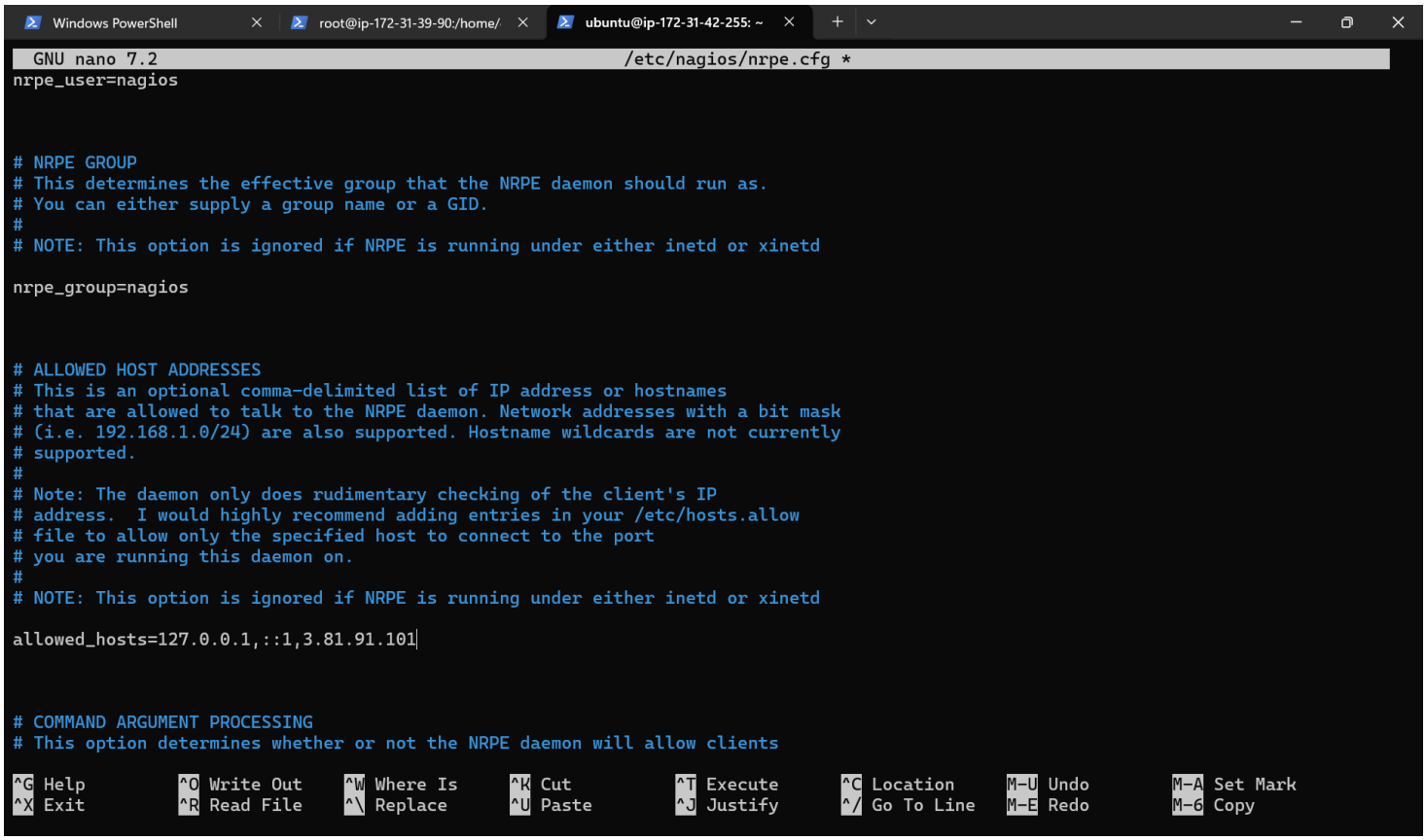
sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-42-255:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmbclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common python3-gpg python3-ldb
  python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs
  smbclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib
  fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients
  python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmbclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg
  python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules
  samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-client3 amd64 0.8-13ubuntu6 [26.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcups2t64 amd64 2.4.7-1.2ubuntu7.3 [272 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdbi1t64 amd64 0.9.0-6.1build1 [25.7 kB]
```

```
Creating config file /etc/nagios-plugins/config/netware.cfg with new version

Creating config file /etc/nagios-plugins/config/nt.cfg with new version

Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version

Creating config file /etc/nagios-plugins/config/radius.cfg with new version

Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version

Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
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.
ubuntu@ip-172-31-42-255:~$
```

15. Now open the nrpe.cfg file. Scroll down the page and find allowed_hosts.Go there, and add your nagios host IP address.

sudo nano /etc/nagios/nrpe.cfg



16. Now restart the NRPE server

sudo systemctl restart nagios-nrpe-server

17. Now again check the nagios-host status

sudo systemctl status nagios

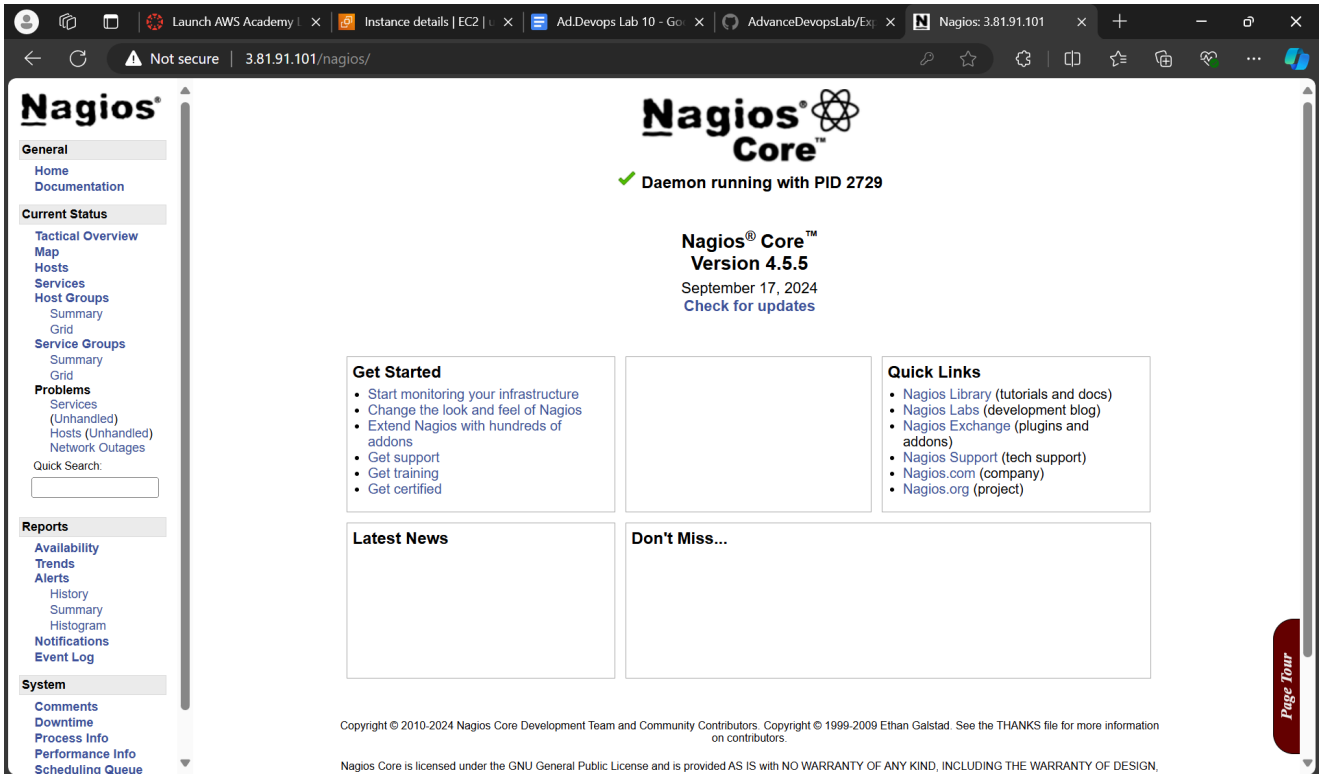
Now check the status is active or not

```
[root@ip-172-31-39-90 ec2-user]#
sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 09:24:28 UTC; 9min ago
       Docs: https://www.nagios.org/documentation
    Process: 2725 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCC>
    Process: 2727 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 2729 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.2M
        CPU: 114ms
     CGroup: /system.slice/nagios.service
             ├─2729 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─2730 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2731 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2732 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2733 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─2742 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 09:30:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;7;(No output on stdout) stderr>
Oct 04 09:31:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;8;(No output on stdout) stderr>
Oct 04 09:32:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;9;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST NOTIFICATION: nagiosadmin;linuxserver;DOWN;notify-host-by-e>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;HARD;10;(No output on stdout) stder>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: NOTIFY job 6 from worker Core Worker 2732 is a non-check >
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   host=linuxserver; service=(none); contact=nagiosadmin
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   stderr line 01: /bin/sh: line 1: /bin/mail: No such fil>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pi>
[root@ip-172-31-39-90 ec2-user]#
```

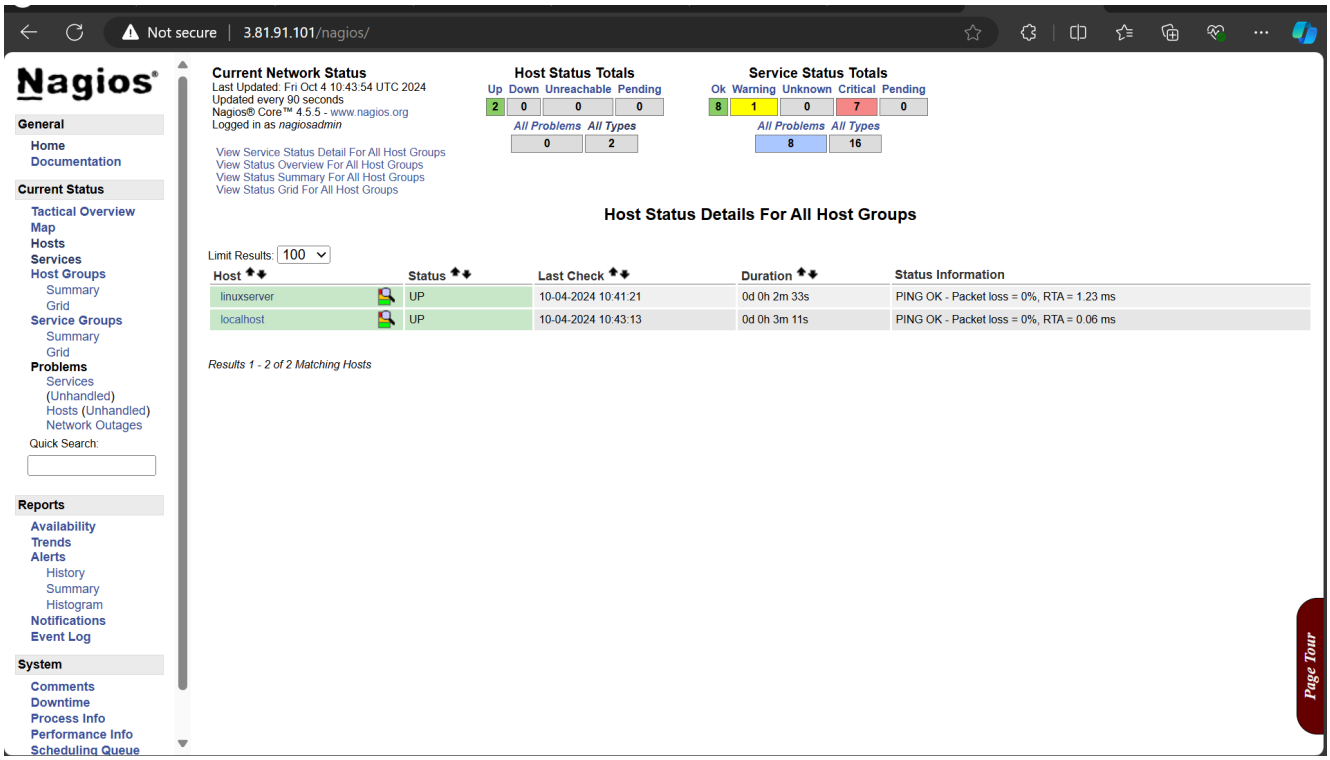
18. Now,check the status of http
    sudo systemctl status httpd
    sudo systemctl start httpd
    sudo systemctl enable httpd

```
[root@ip-172-31-39-90 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2024-10-04 09:45:14 UTC; 33min ago
       Docs: man:httpd.service(8)
   Main PID: 4146 (httpd)
     Status: "Total requests: 19; Idle/Busy workers 100/0;Requests/sec: 0.0095; Bytes served/sec:  70 B/sec"
      Tasks: 230 (limit: 1112)
     Memory: 17.3M
        CPU: 1.428s
     CGroup: /system.slice/httpd.service
             ├─4146 /usr/sbin/httpd -DFOREGROUND
             ├─4148 /usr/sbin/httpd -DFOREGROUND
             ├─4149 /usr/sbin/httpd -DFOREGROUND
             ├─4150 /usr/sbin/httpd -DFOREGROUND
             ├─4151 /usr/sbin/httpd -DFOREGROUND
             └─4533 /usr/sbin/httpd -DFOREGROUND

Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal httpd[4146]: Server configured, listening on: port 80
[root@ip-172-31-39-90 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-39-90 ec2-user]# sudo systemctl enable httpd
```

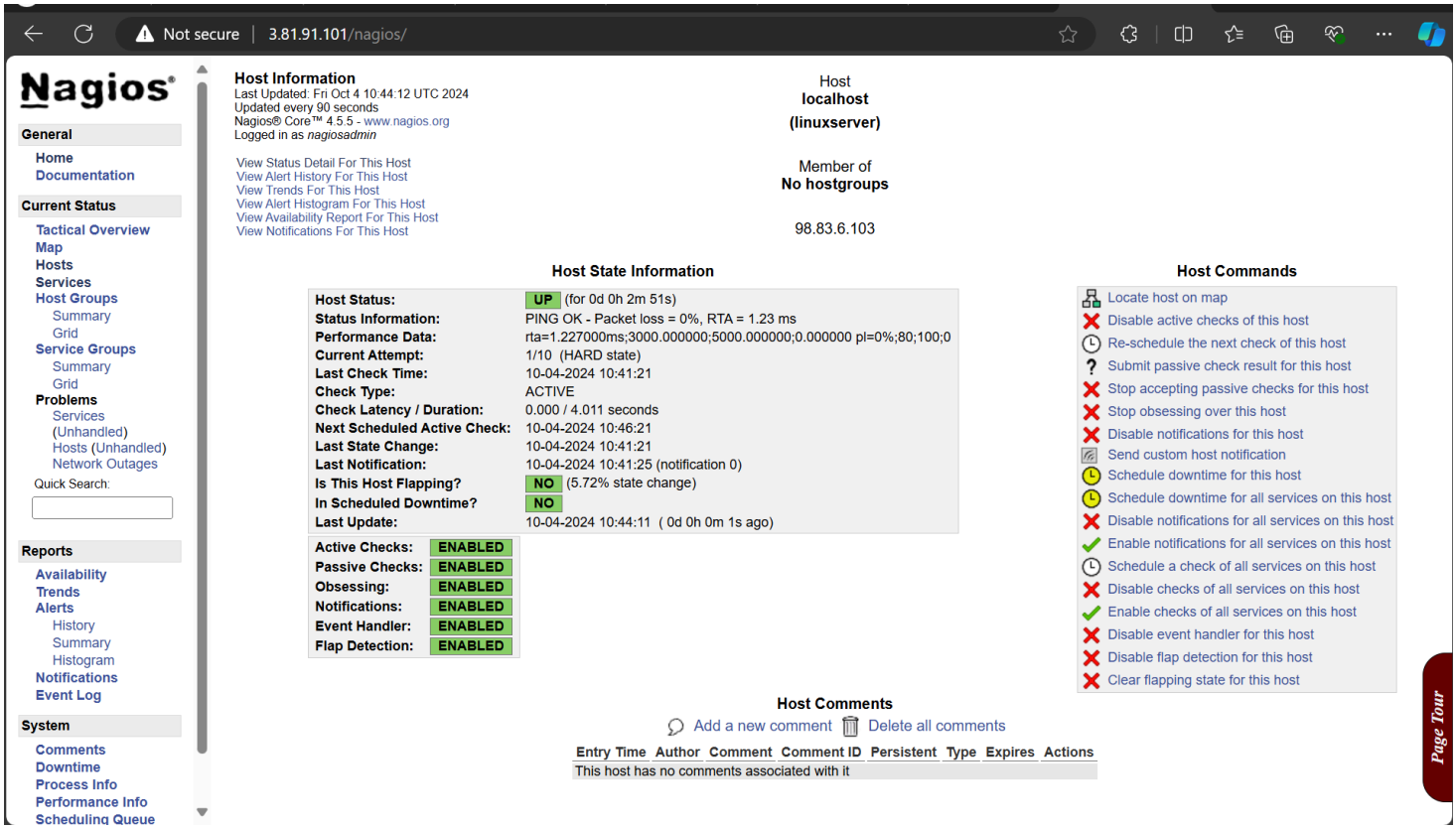
19. Now to check Nagios dashboard go to http://<Nagios-host ip>/nagios.
    Take host public IP address to add in “Nagios-host ip”.



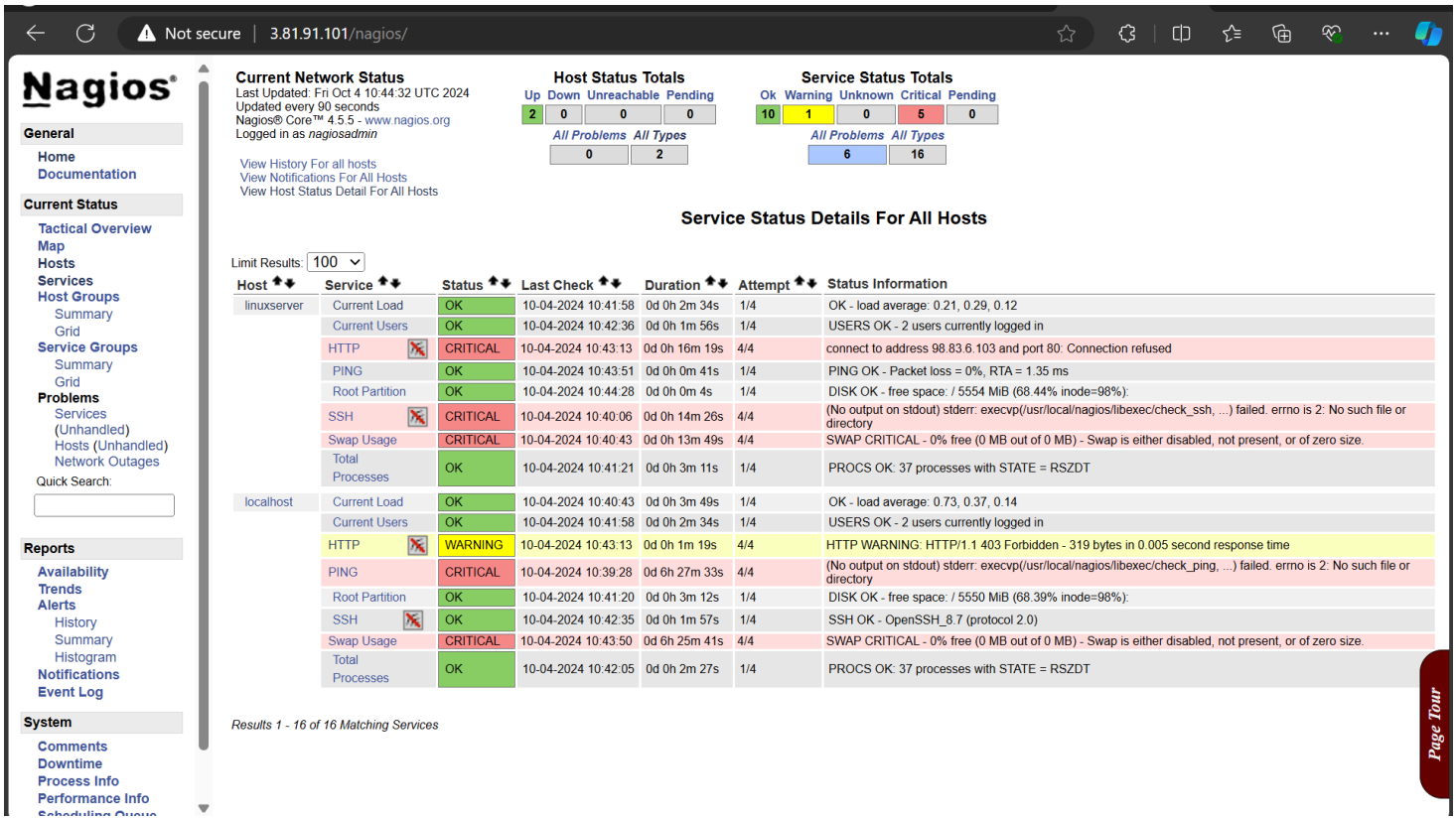
20. Now Click on Hosts from the left side panel there you will get the status of linuxserver and localhost. Check whether its status is up or down.If it is up then it is successful.



21. Now click on linuxserver. There you will get Host state information.



22. Then click on Services presnet on left sidebar where you will get current network status.



Conclusion: In this experiment, we created a nagios client EC-2 instance of t2.micro ubuntu instance. Then after updating hostname, hostgroup_name.Then adding the address to the public IP of your Linux client.Then verify the configuration and then check the status of nagios-host and then start and enable it then open http://<Nagios-host ip>/nagios. There you will get the status of linuxserver and localhost.Thus we can monitor essential network services on the Linux server.

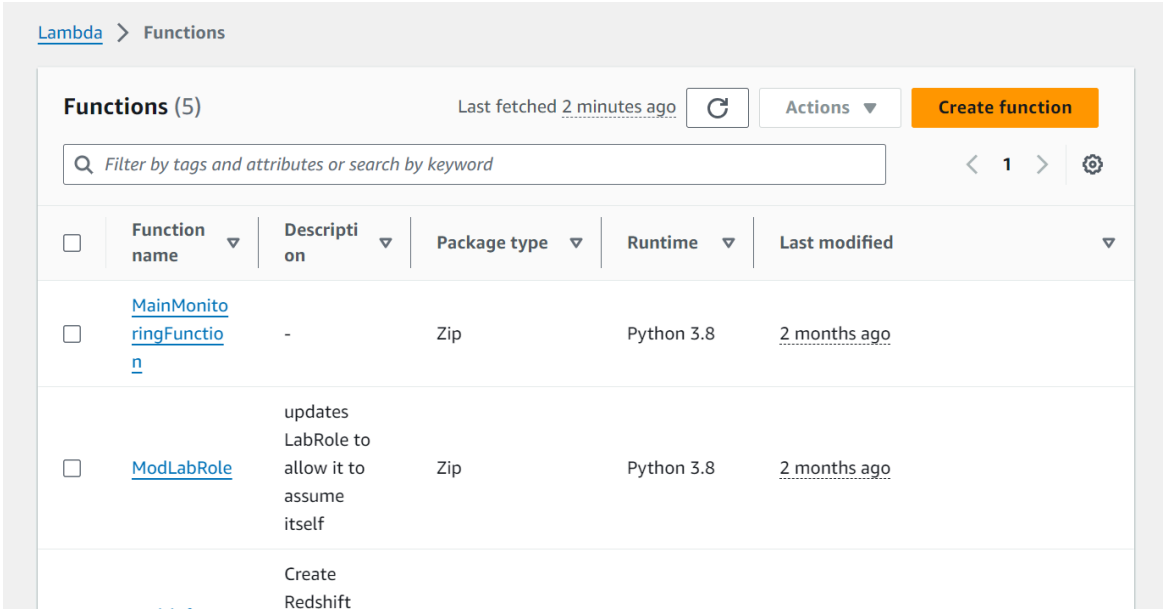
Experiment No:11

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

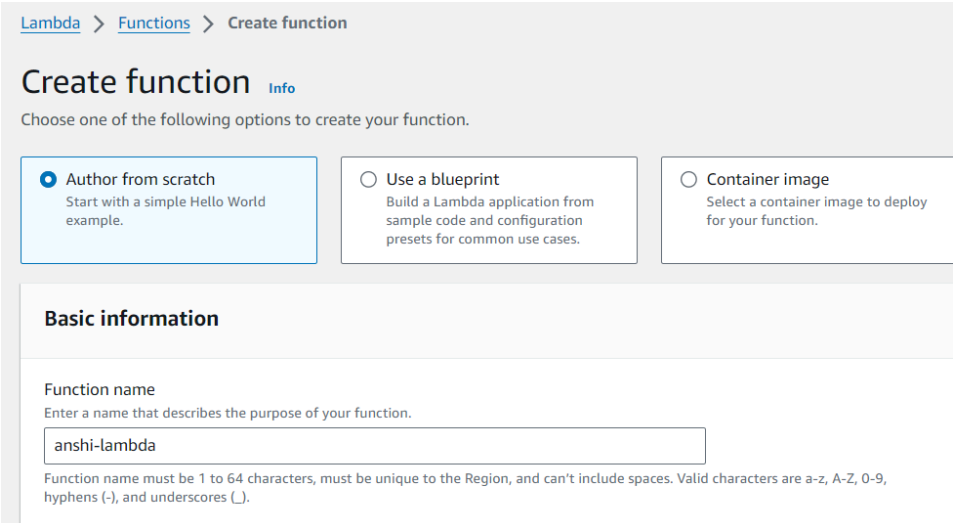
Steps:

1. Go to AWS ACADEMY.
2. **Create the lambda function:**

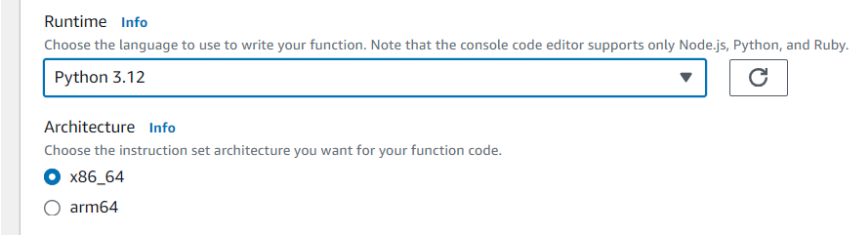
Firstly, Search lambda, then Open lambda and then click on create function button.



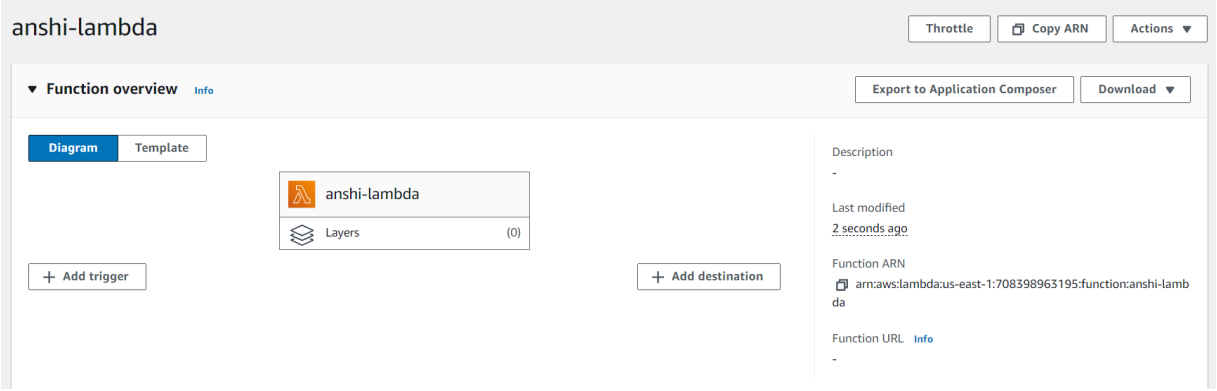
3. Now Give a name to your Lambda function,



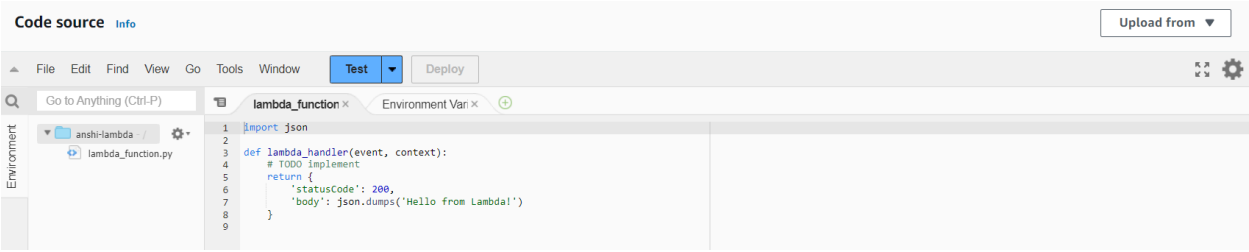
4. Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12, Architecture as x86, and Exceution role to Create a new role with basic Lambda permissions.



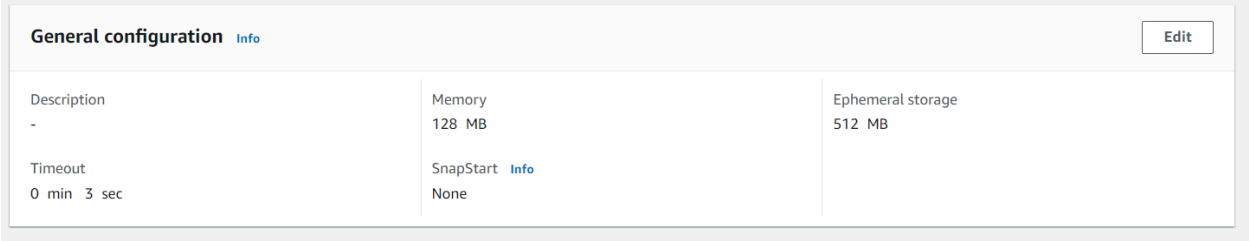
Thus the Lambda function was created successfully.



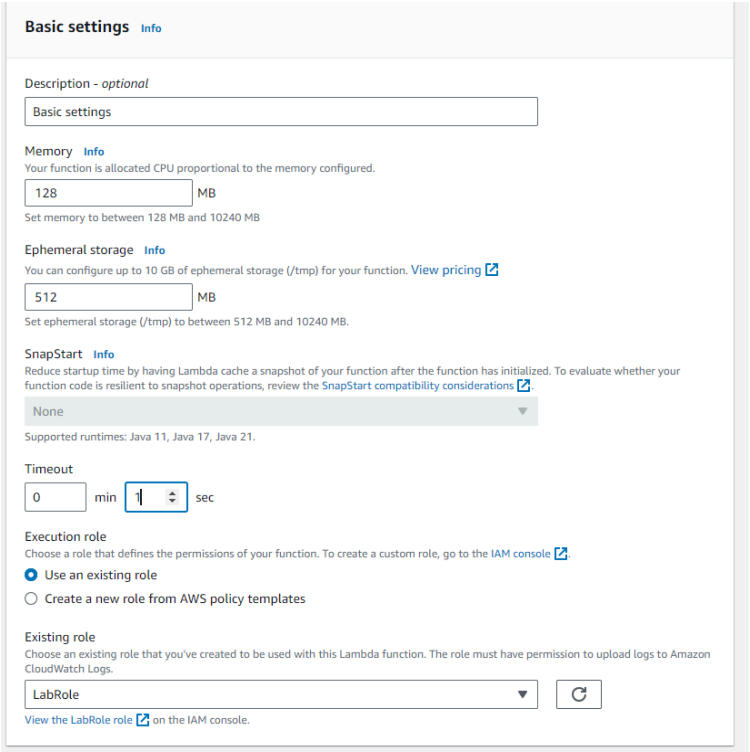
5. Then go to code section.



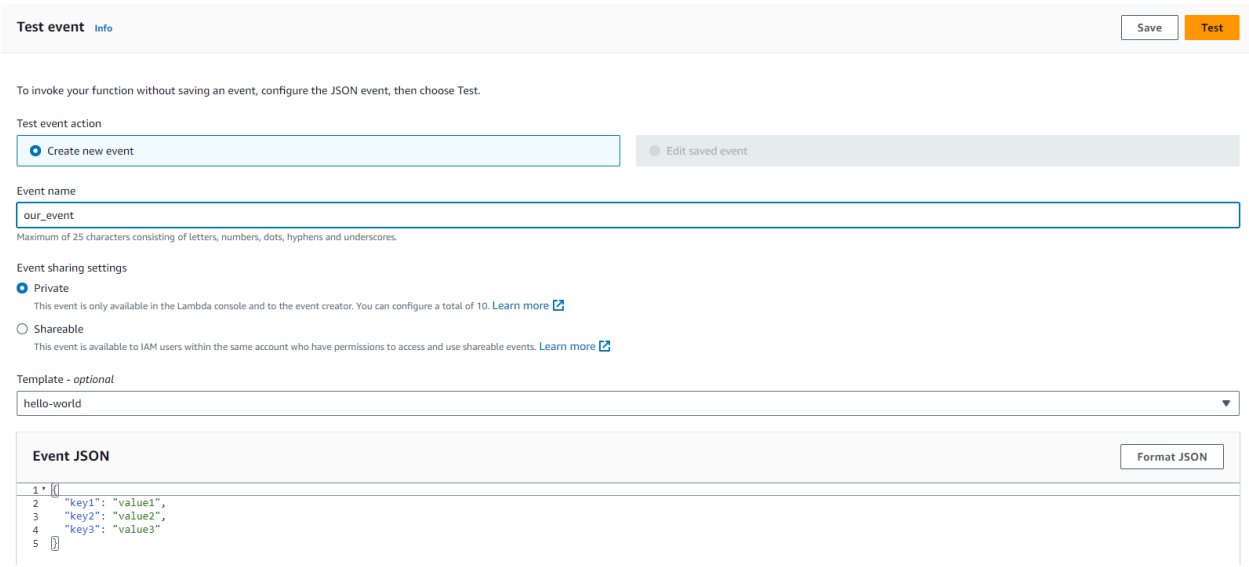
So to Edit the basic settings, go to configuration then click on edit.



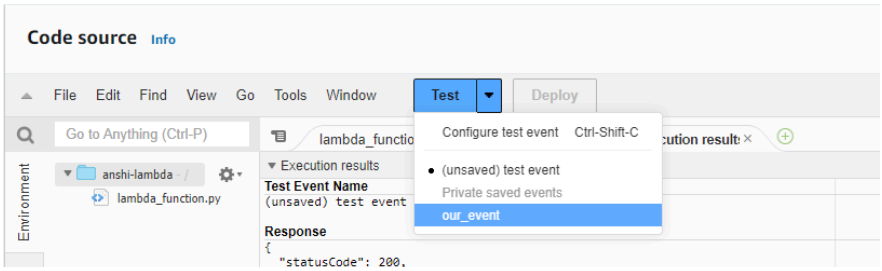
Now enter a description and change Memory and Timeout.Here, I've changed the Timeout period to 1 sec.



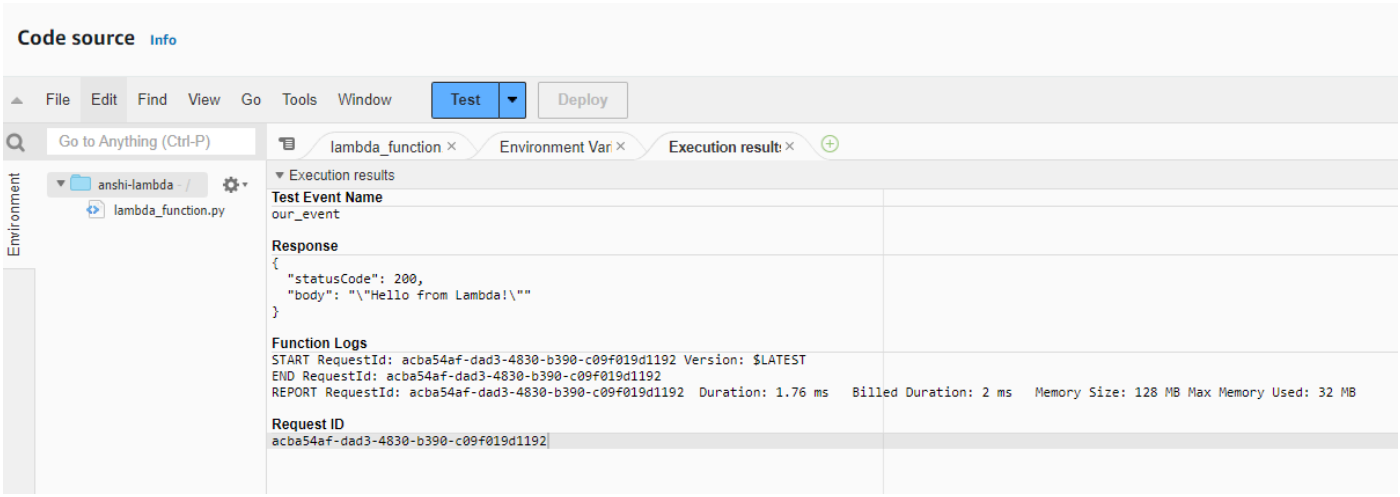
6. Now Click on the Test tab then select Create a new event, give a name to the event here i have given name as “our_event” and then select Event Sharing to private, and select hello-world template.



7. **Testing & Deployment:** Now In Code section select the created event (our_event) from the dropdown of test ,then click on test .

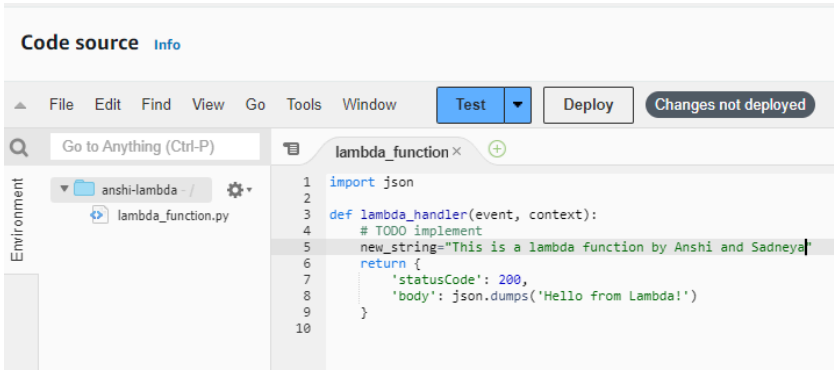


8. Now you will see the following output.

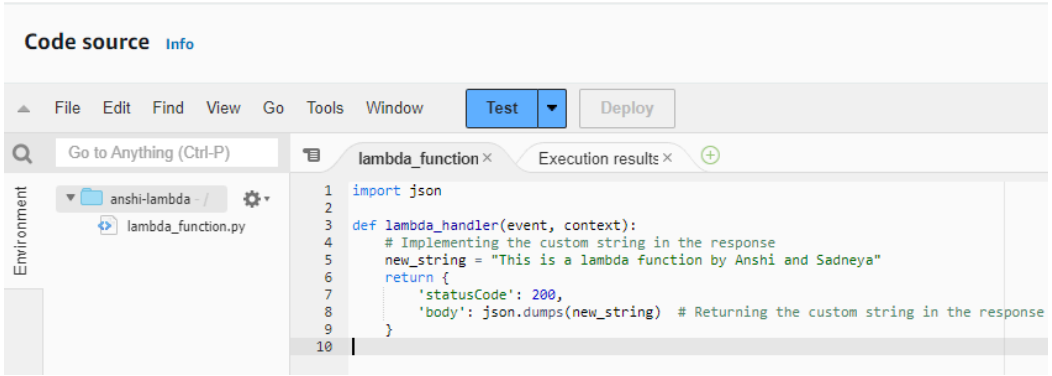


9. **Making changes**

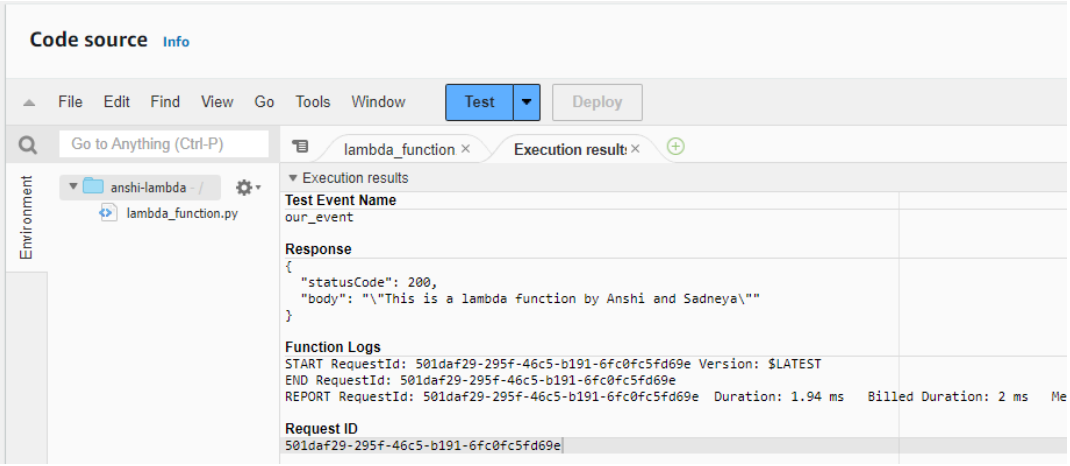
You can edit your lambda function code.Here I have created a new string name “new_string” and assigned a string to it.



Now save it by ctrl+s and then click finally on deploy to deploy the changes.



10. **Testing and redeploying changes** Now click on the test and observe the output. Thus Output gives status code 200. Thus deployment is done successfully.



Conclusion: In this experiment, we have successfully created an AWS Lambda function. After we have chosen the python language for writing the function. Then we edited the basic settings, including adjusting the timeout to 1 second.Then we tested it and finally deployed it.Thus it got deployed successfully.Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time.This provided information about simplicity of AWS Lambda in creating serverless application.

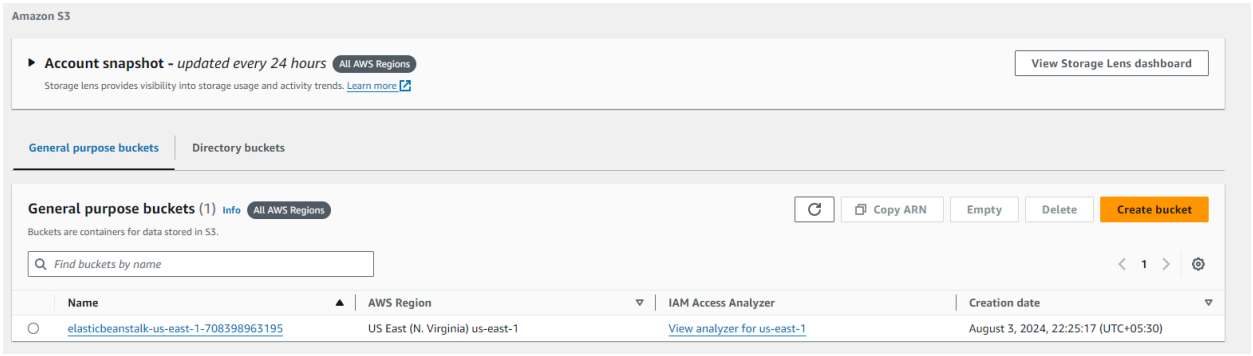
Experiment No:12

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

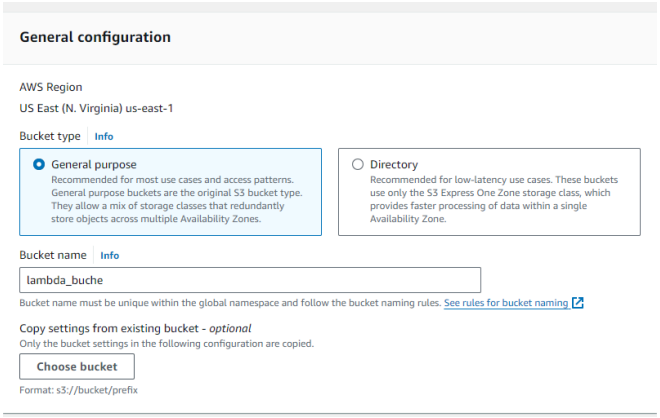
Steps :

1. **Creating the lambda function:**

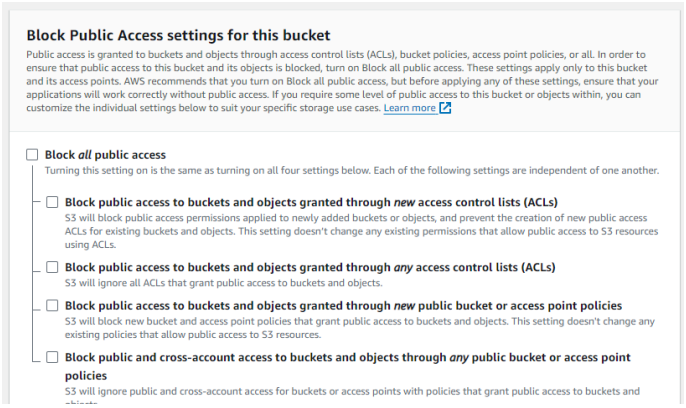
Go to AWS ACADEMY. Now search and open S3 from services. Then click on create bucket.



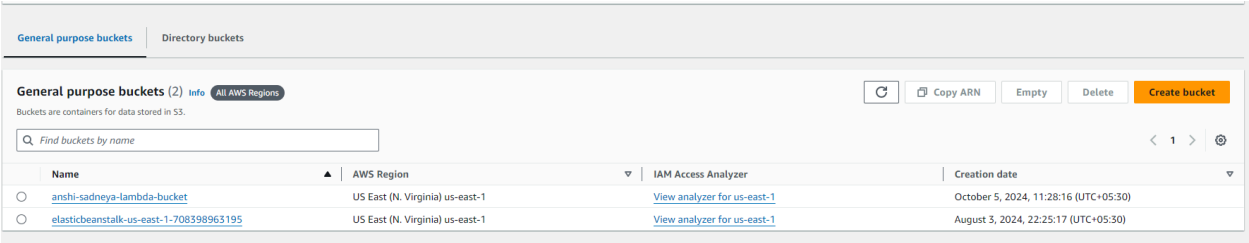
2. Now Give a name to your Bucket.Here I have given the name as lambda_buche.



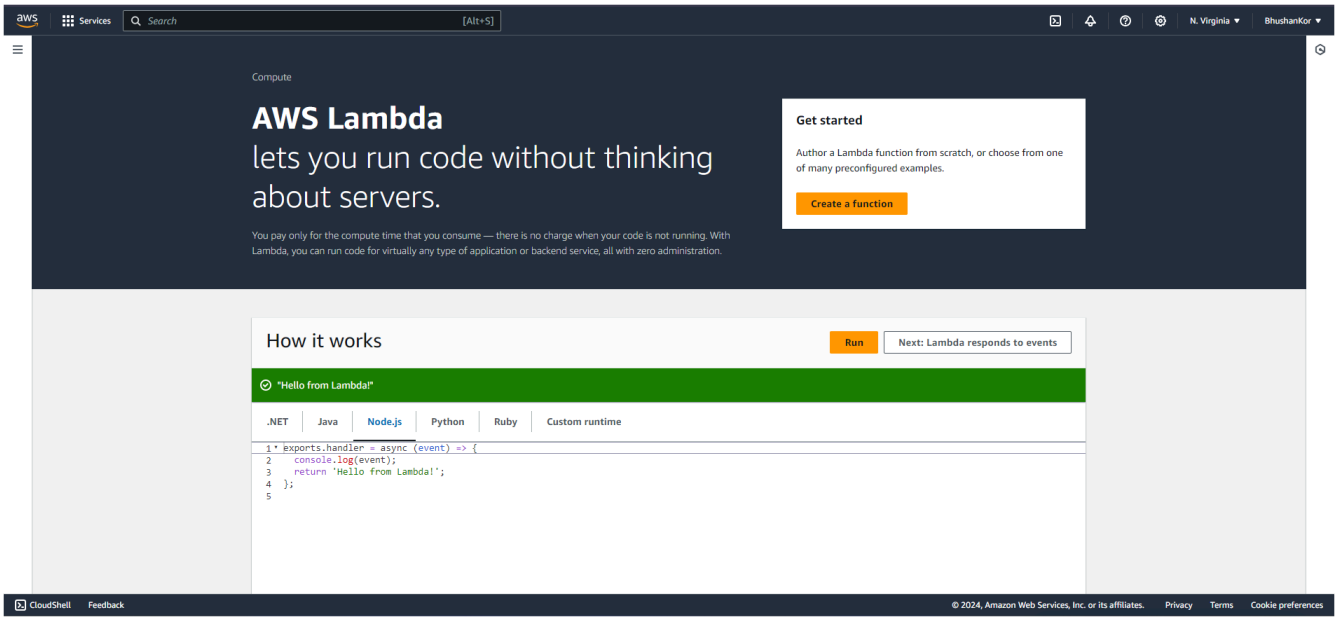
3. then remove the Block public access and keep other settings default.



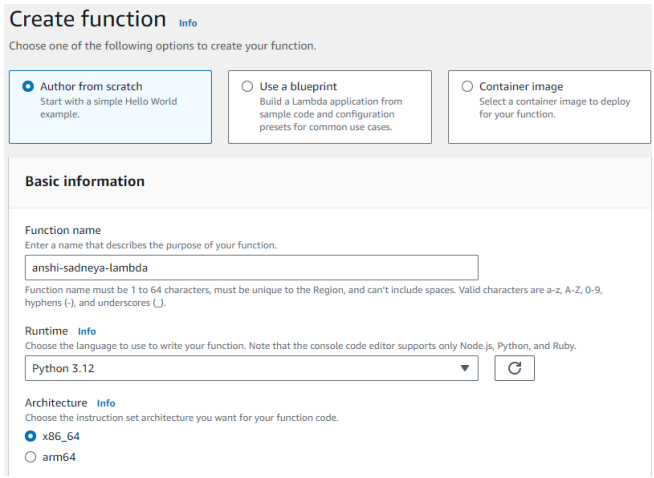
4. Thus,the S3 bucket was created successfully.



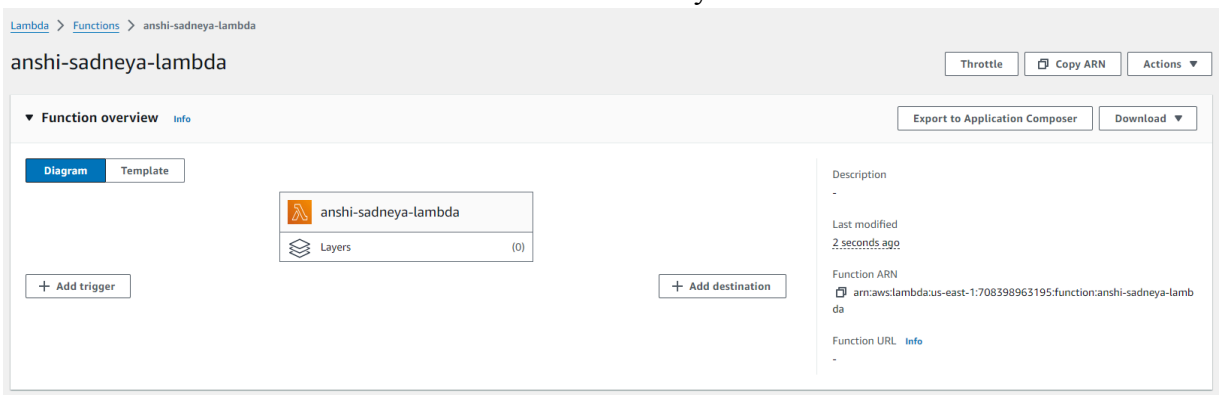
5. Search and Open lambda console and click on create function button.



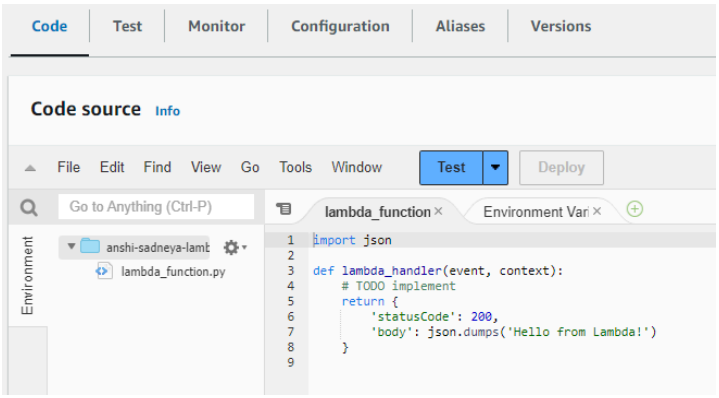
6. Now Give a name to your Lambda function, Select the language to write your function.Here I have chosen python3.12, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.Note that the console code editor supports only Node.js, Python, and Ruby



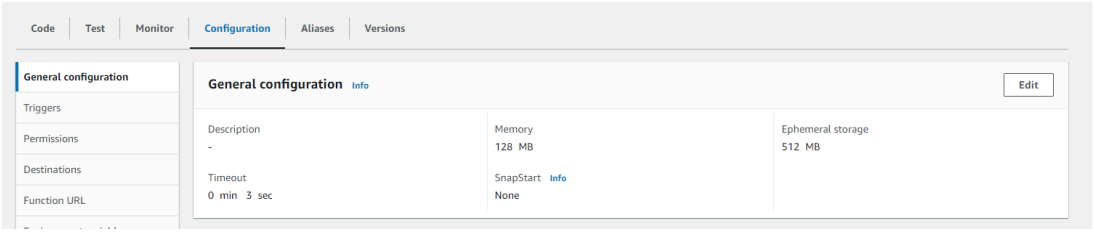
7. Thus the Lambda function was created successfully.



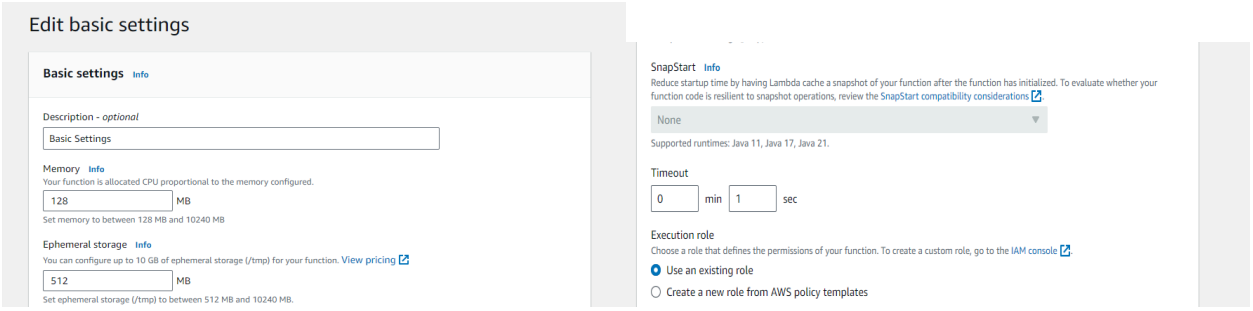
8. Then Go into the code section.You will see some default code there.



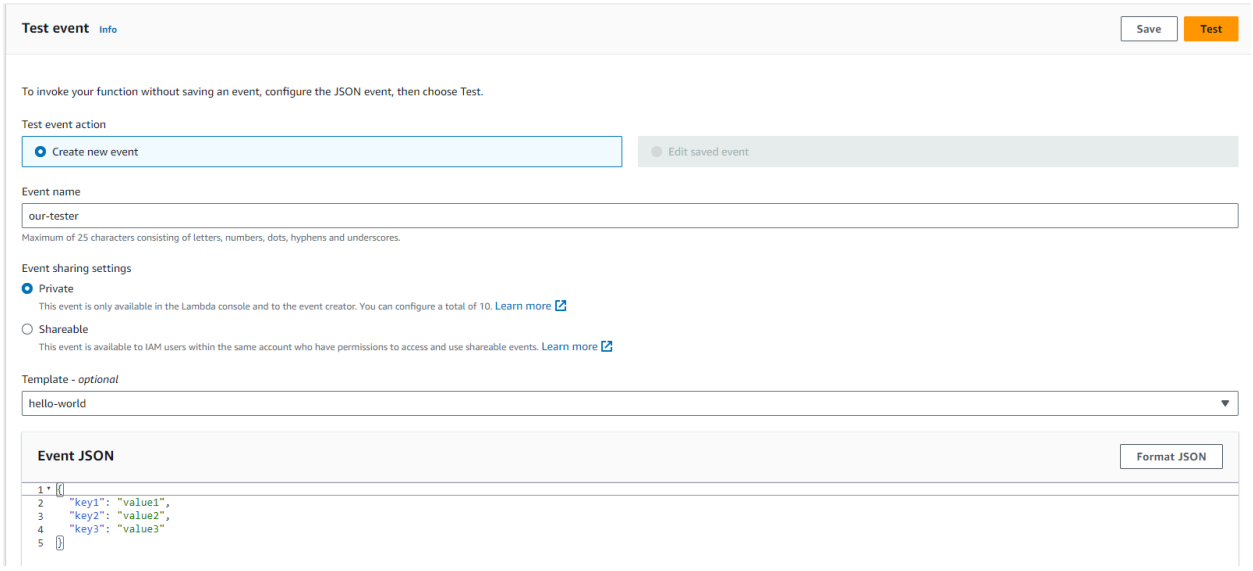
9. To Edit the basic settings go to configuration then click on edit setting.



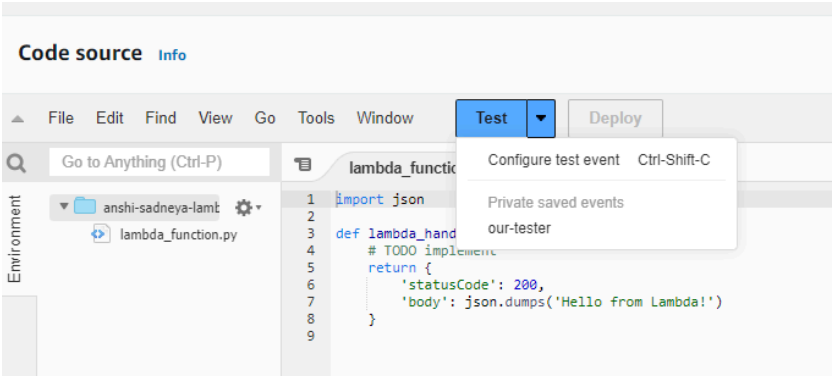
10. Here,enter a description which is optional and change Memory and Timeout. I've changed the Timeout period to 1 sec.



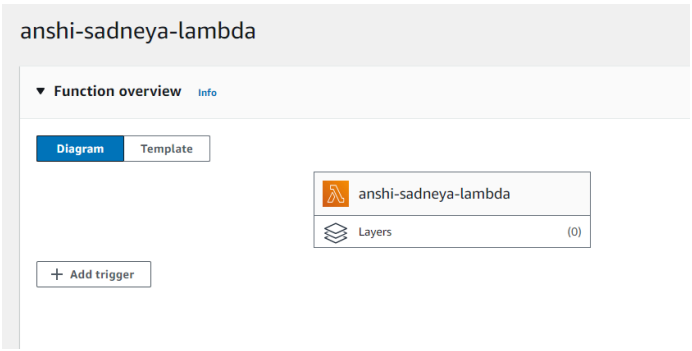
11. Now Click on the Test then select Create a new event, give a name to the event.Here I have given name as 'our-tester' and then select Event Sharing to private, and select s3 put template.



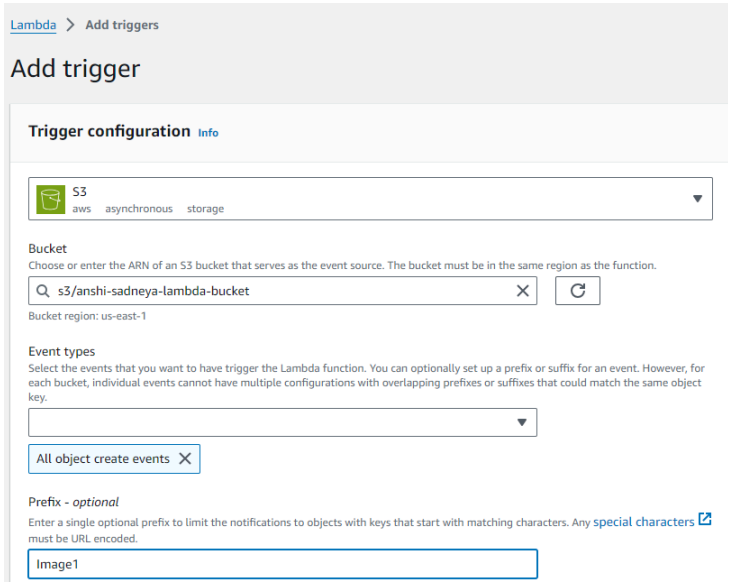
12. Now go to the Code section.Then click on the Test dropdown icon and select the event which we have created now('our-tester').



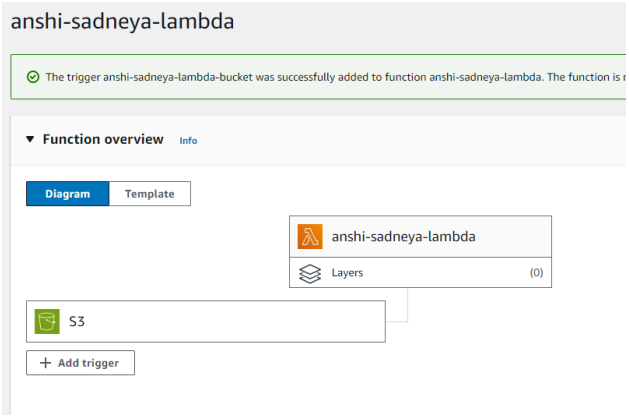
13. Now go into the Lambda function and then click on add tigger.



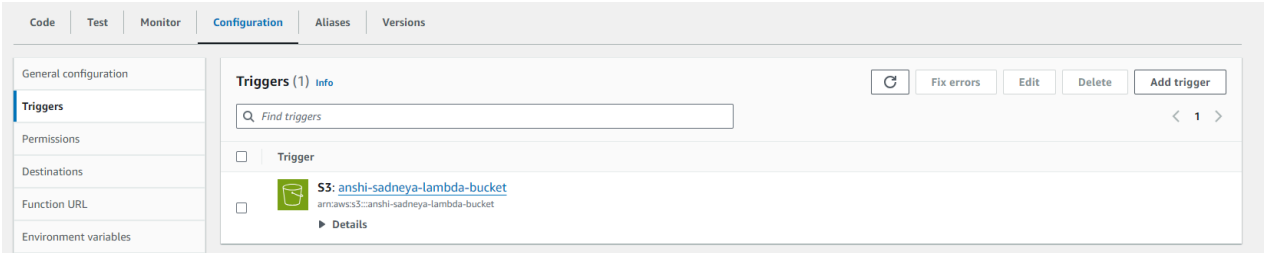
14. Now in the Trigger information.Select the source as S3.Then select the bucket which we have created now (lambda_buche), keep other things default and also you can add prefix to image.



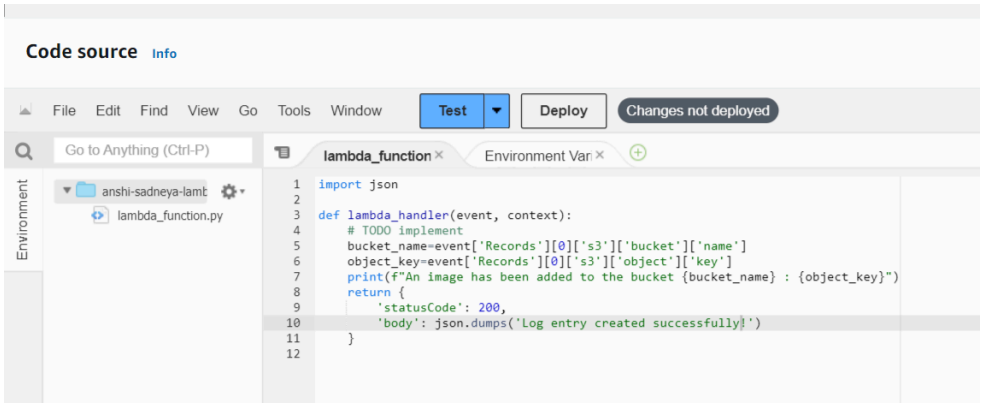
15. Thus,Trigger created successfully.



16. You can also check it in the configuration section.

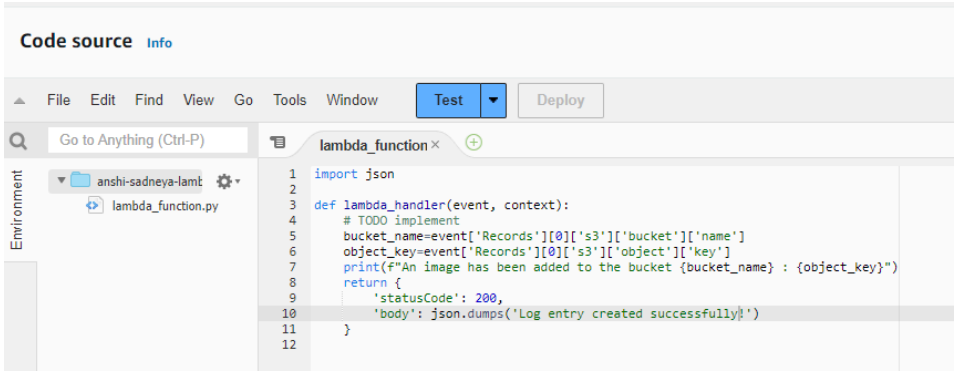


17. Now write a code which logs a message "Log entry crested successfully" when triggered.

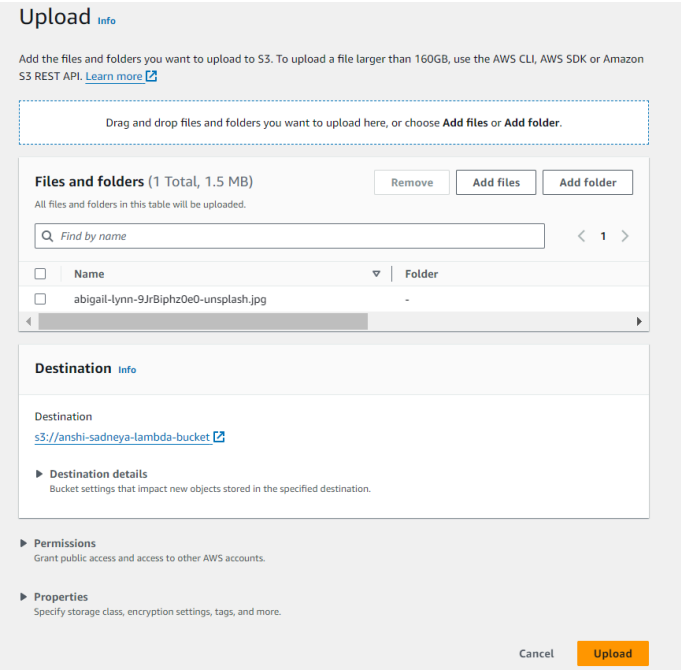


Here changes are not deployed.

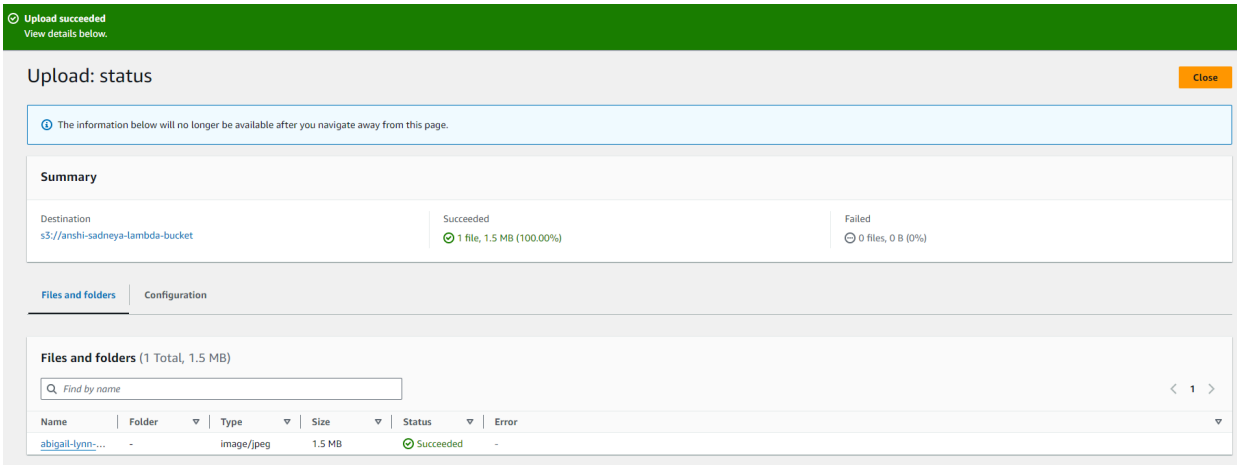
18. So now,Save the file by ctrl+s and then click on deploy.



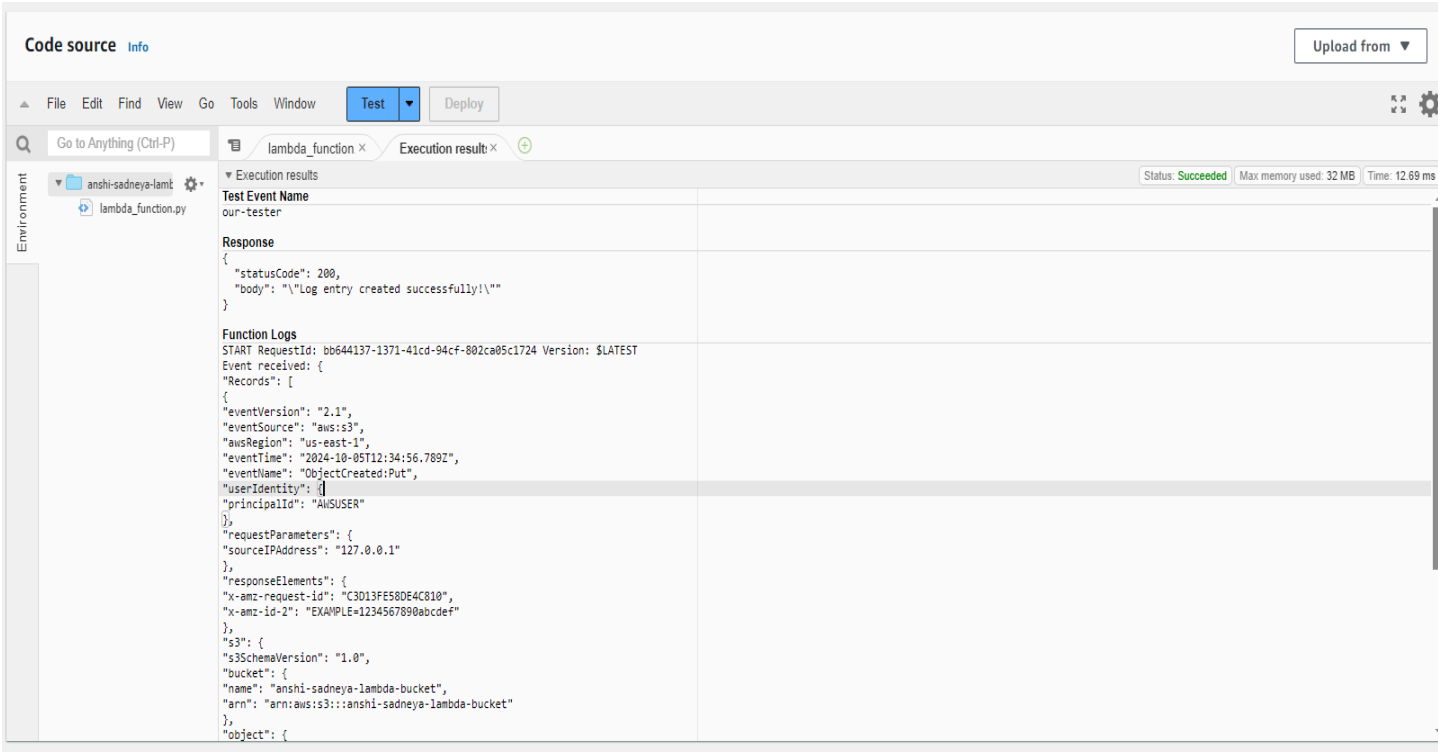
19. Go to S3 bucket,and there upload any image to the bucket.



20. Thus image uploaded successfully.

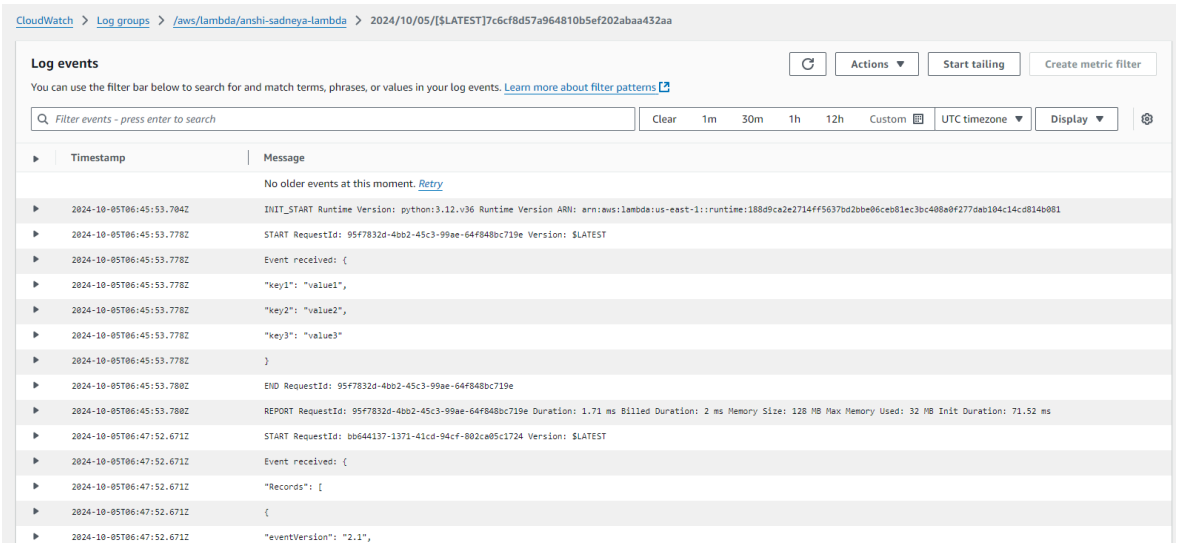


21. Now goto lambda function.Then click on test.This will give you log about the image that we have uploaded in S3 bucket.



In response,It gives status 200 and also the message “Log entry created successfully” and also contains function Logs.

22. Now go to cloudwatch.Then go into log groups.Inside that you will get the lambda function name that we have created click on it.Here,you will get a detailed log of events.



Conclusion: In this,we have created lambda function successfully.Then we have also created s3 bucket successfully.Then I have edited the setting by setting timeout for 1 sec and adding a description.Then created a event name ‘our-tester’.then selected that event for test. Then deployed it.Thus deployed successfully.Then we have added a trigger in which we added a s3 bucket which we created.Then we have added a print message.Then again deployed the code after uploading an image in a s3 bucket.This given a status code 200 in response with that message which we added in code.Then in cloudwatch it given detailed logs of it.