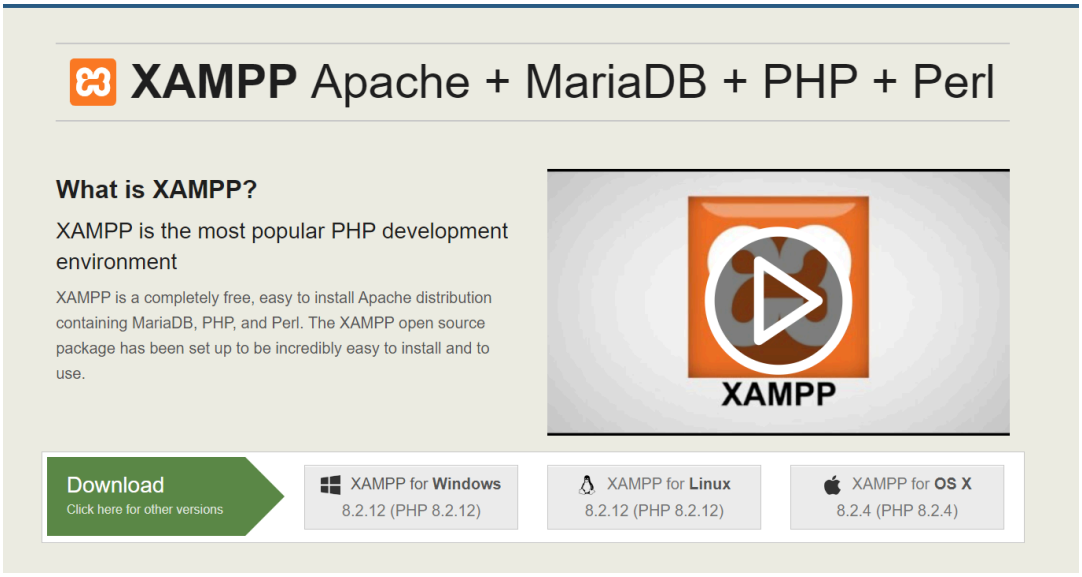
Name:Atharv Nikam **Div:D15C** **Roll:36**

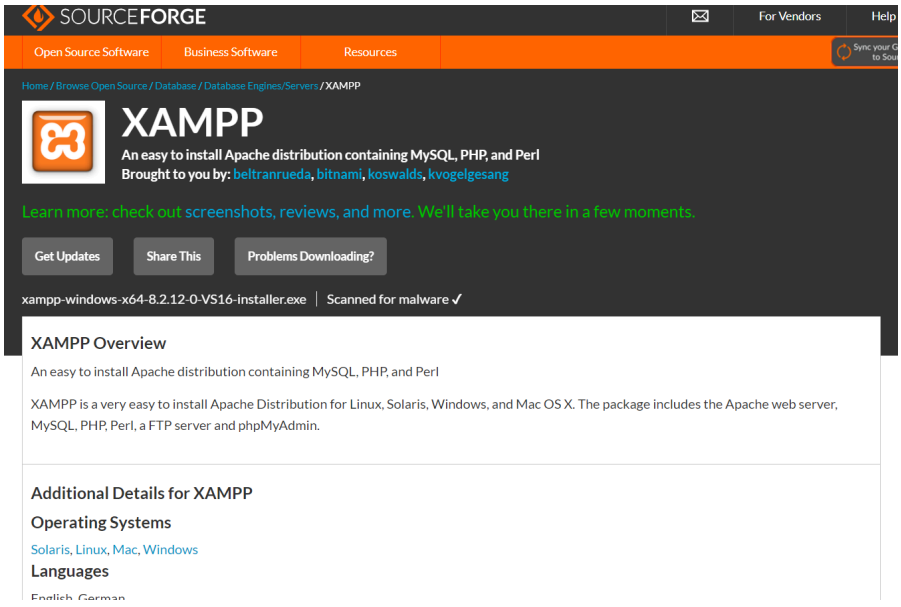
Exp :1A

Aim : AWS (EC2) Installation steps for Linux instance Hosting a website on Local Virtual Machine using Xampp

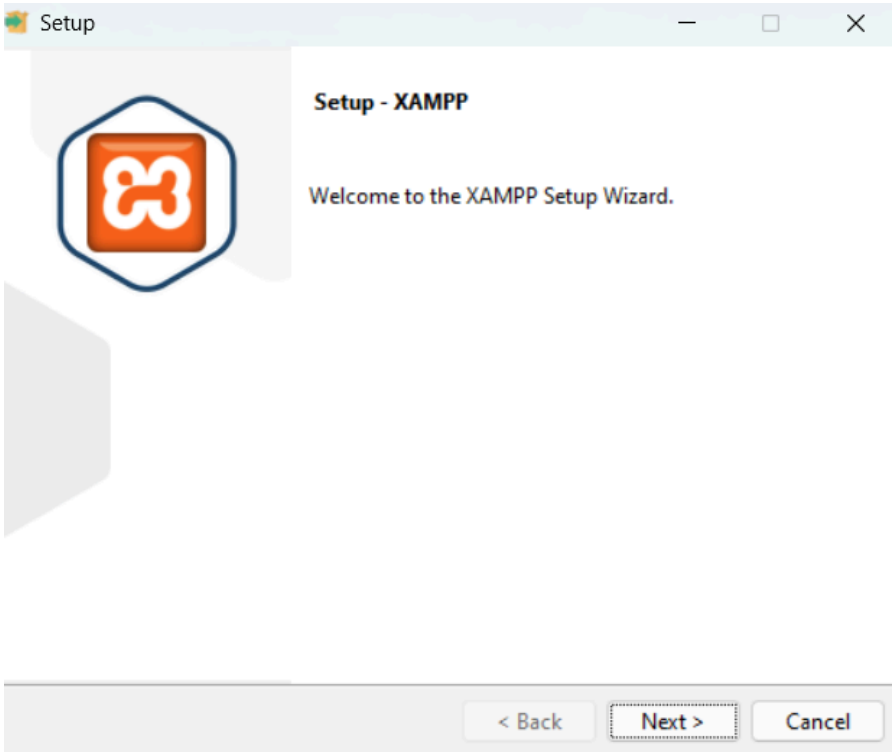
1) Go to official website of xampp

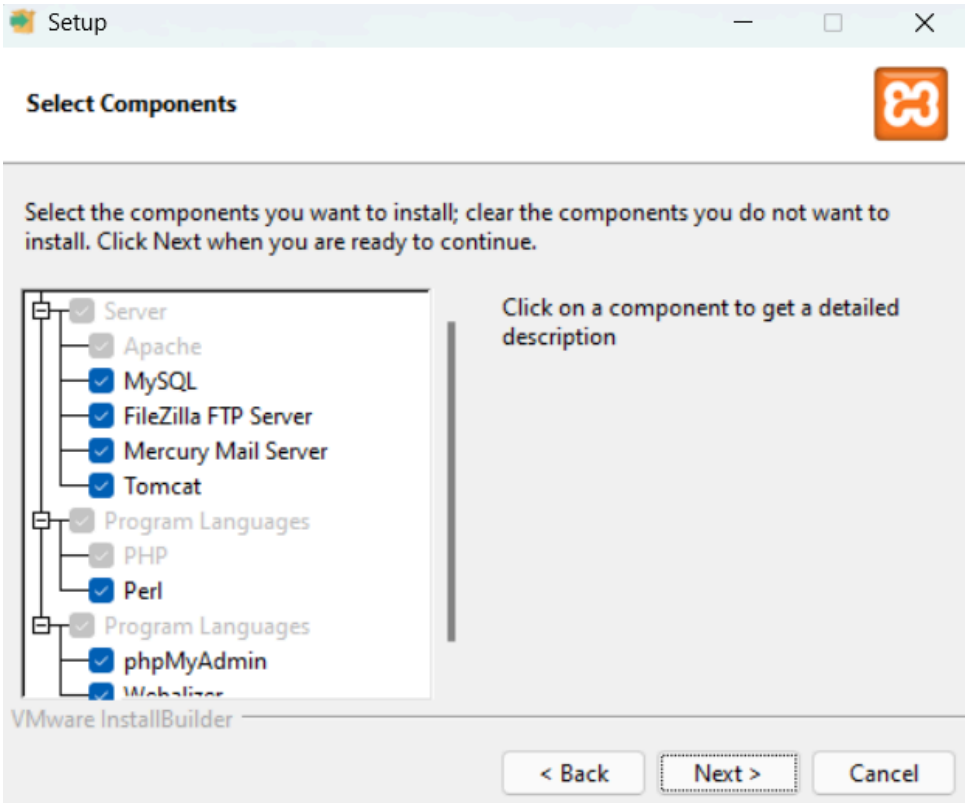


2) click on download and it will automatically get downloaded

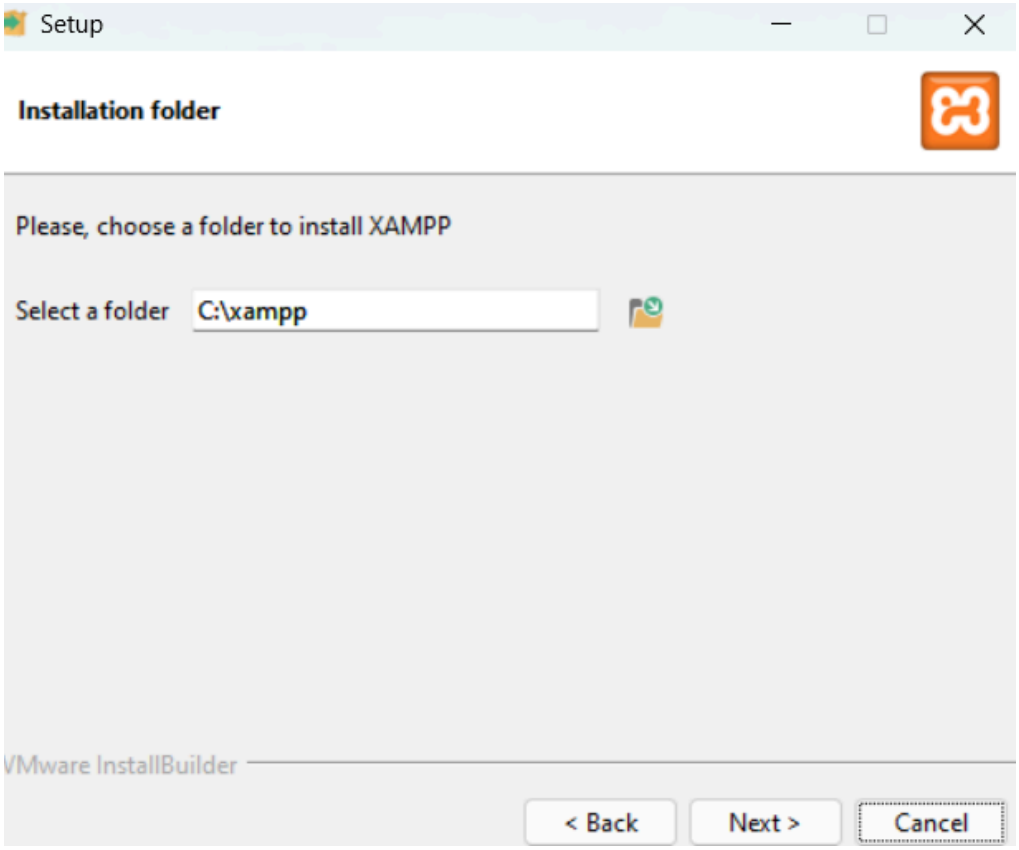


3)click next

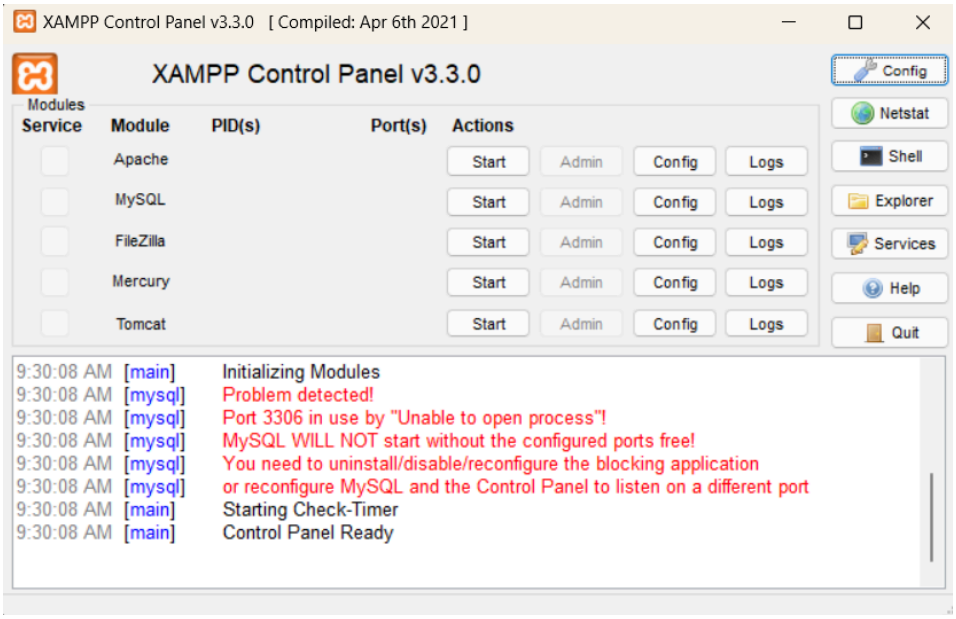

Setup
Setup - XAMPP
Welcome to the XAMPP Setup Wizard.
< Back
Next >
Cancel


Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Server
Apache
MySQL
FileZilla FTP Server
Mercury Mail Server
Tomcat
Program Languages
PHP
Perl
Program Languages
phpMyAdmin
Click on a component to get a detailed description
VMware InstallBuilder
< Back
Next >
Cancel

4)click on next till the setup gets complete



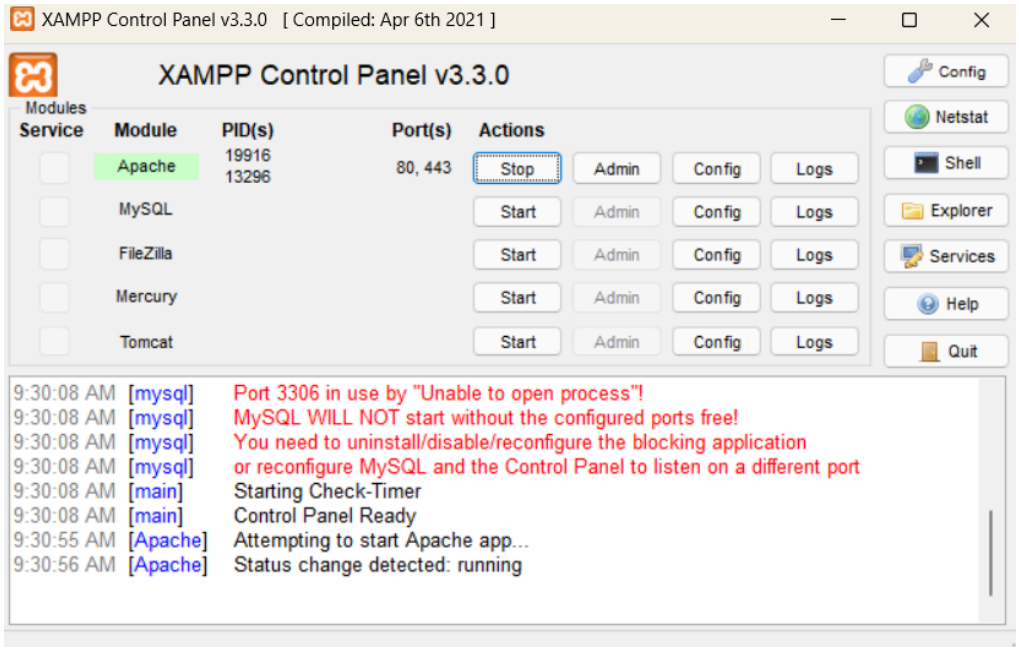
5)Open Xampp



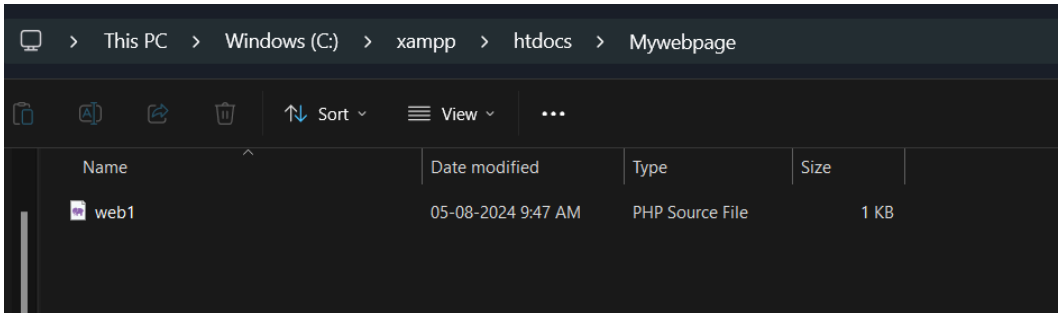
6)Write a php code

```php
<?php
echo "Hello, My Name is Atharv Nikam";

echo "<br>";
echo "My roll no is 36";
echo "<br>";
echo "Welcome to Adv Devops Lab";
?>
```

7)Starting Xampp

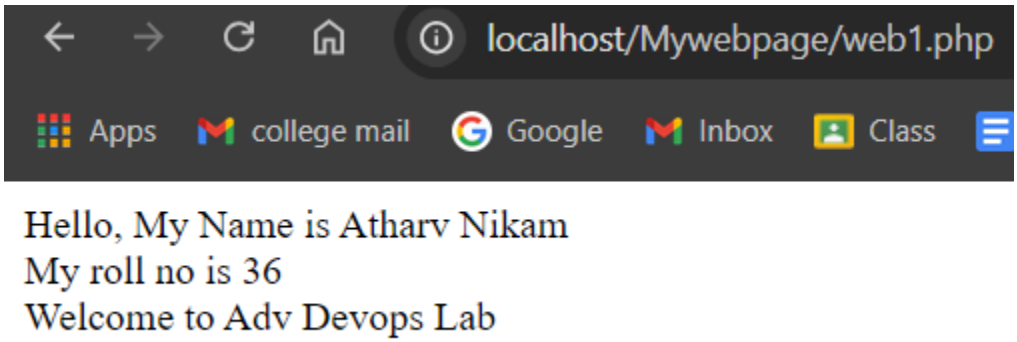


8)put your php file in the xampp ->htdocs



9)Open this

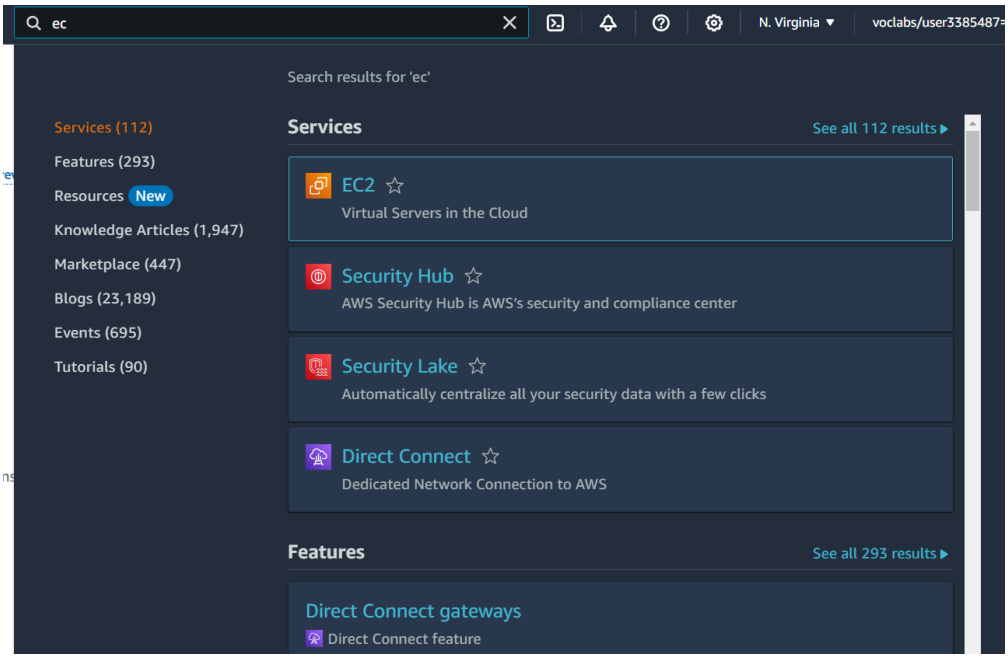
Click on your file and your website will open



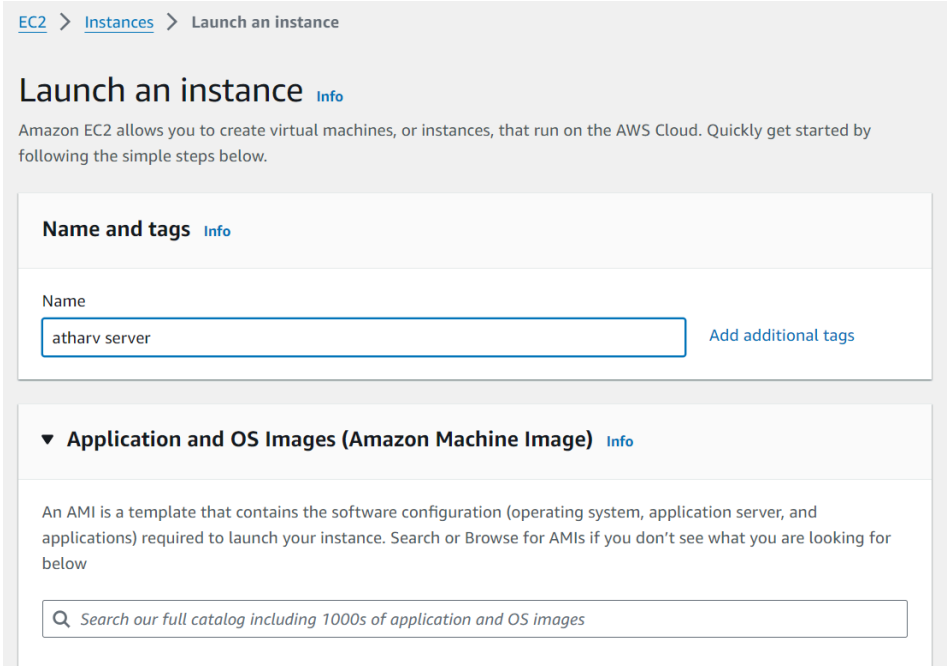
Name:Atharv Nikam Div D15C Roll No:36

Aim : AWS (EC2) Installation steps for Linux instance

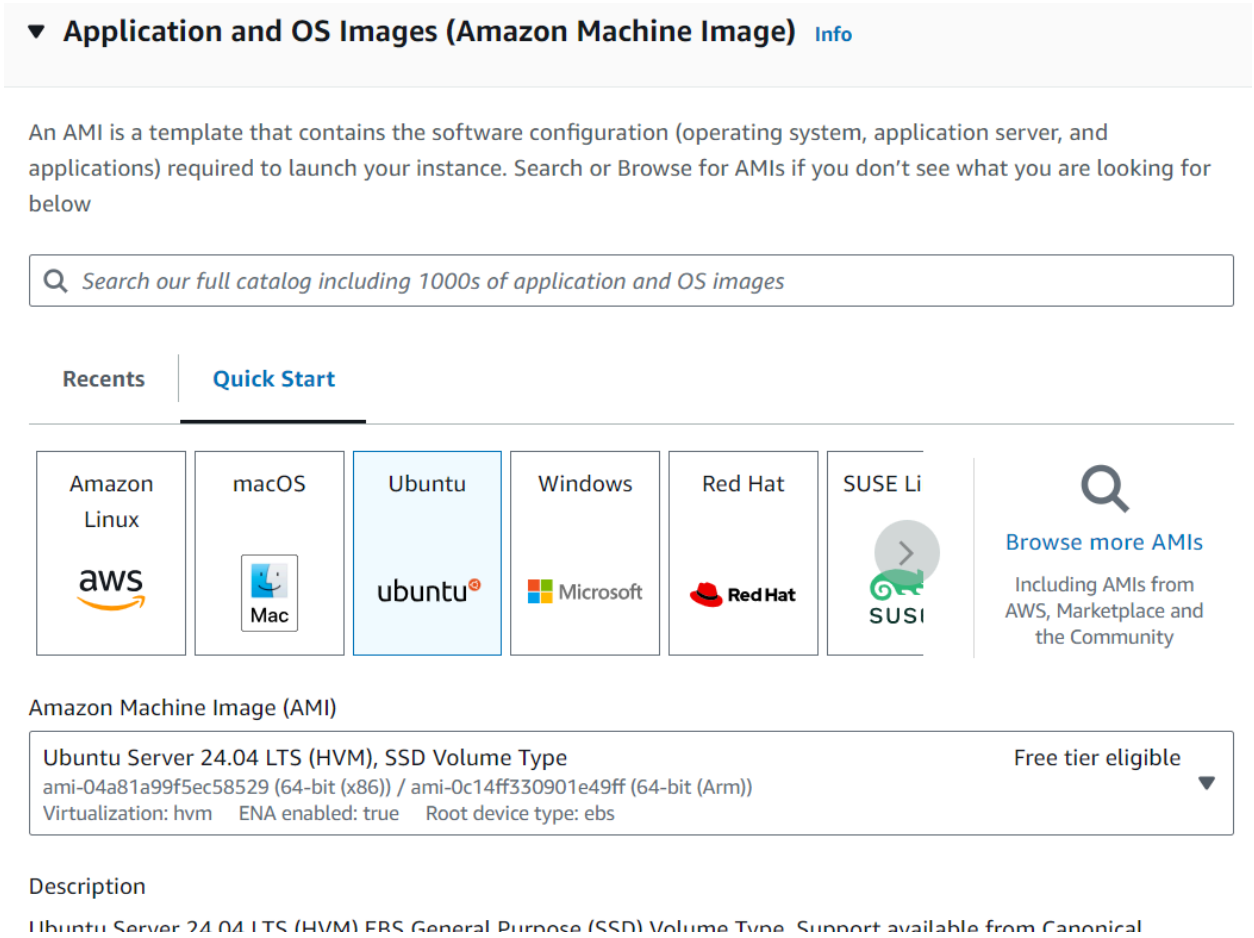
1)Go to aws homepage and click on ec2



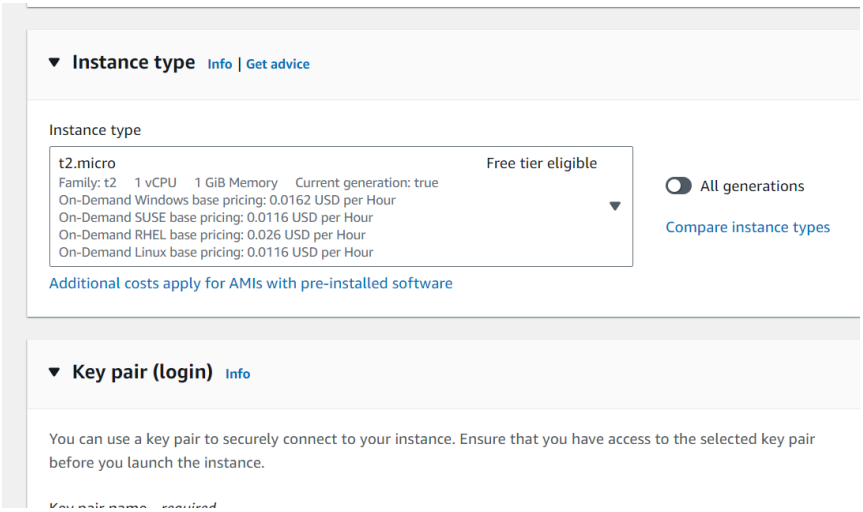
2)click on ec and give a name



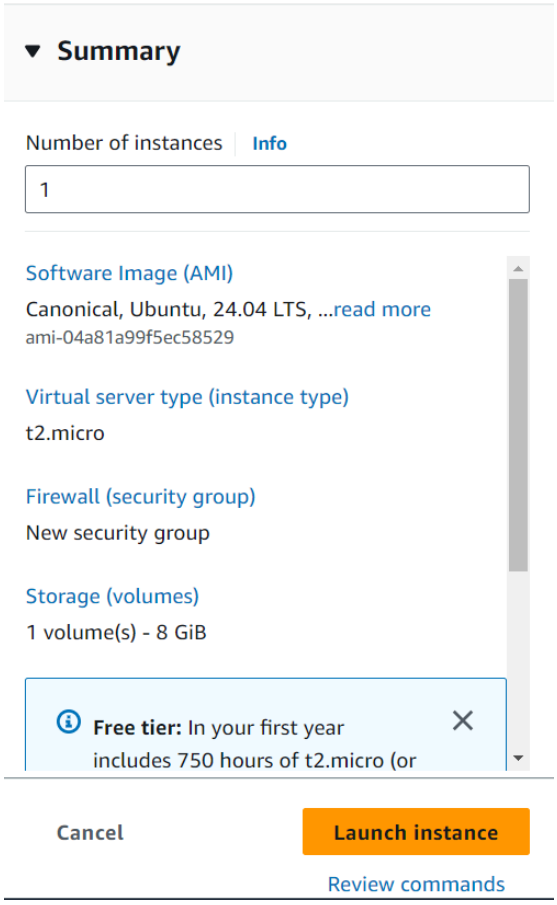
3)select on ubuntu



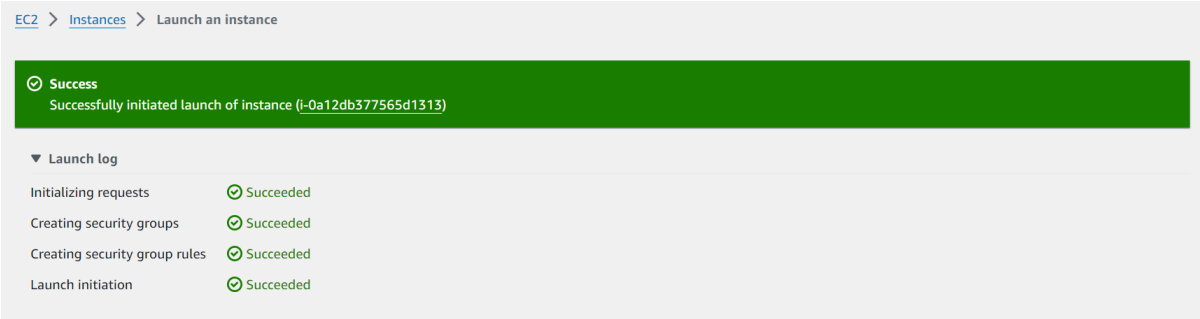
4)select instance type t2



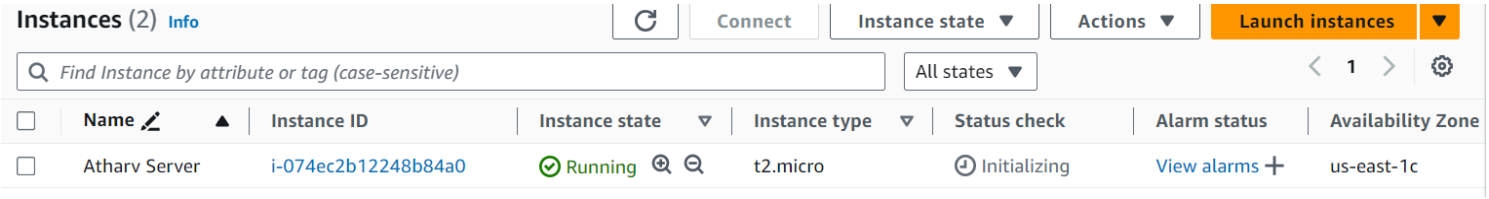
5)see the summary and launch the instance



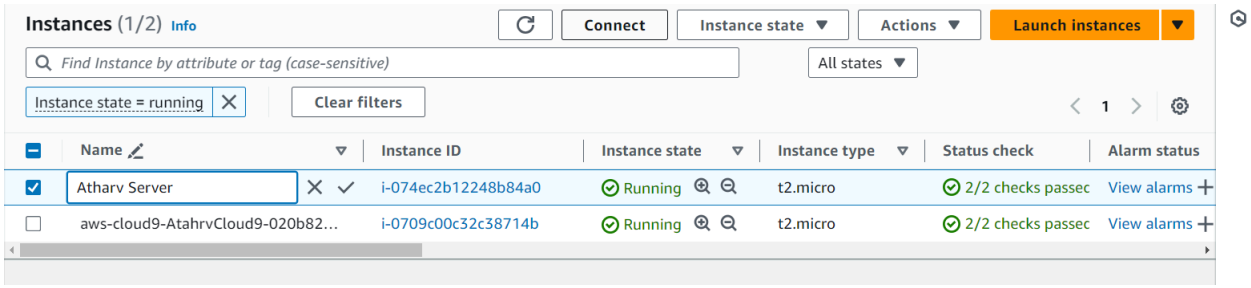
6)Successfully instance created



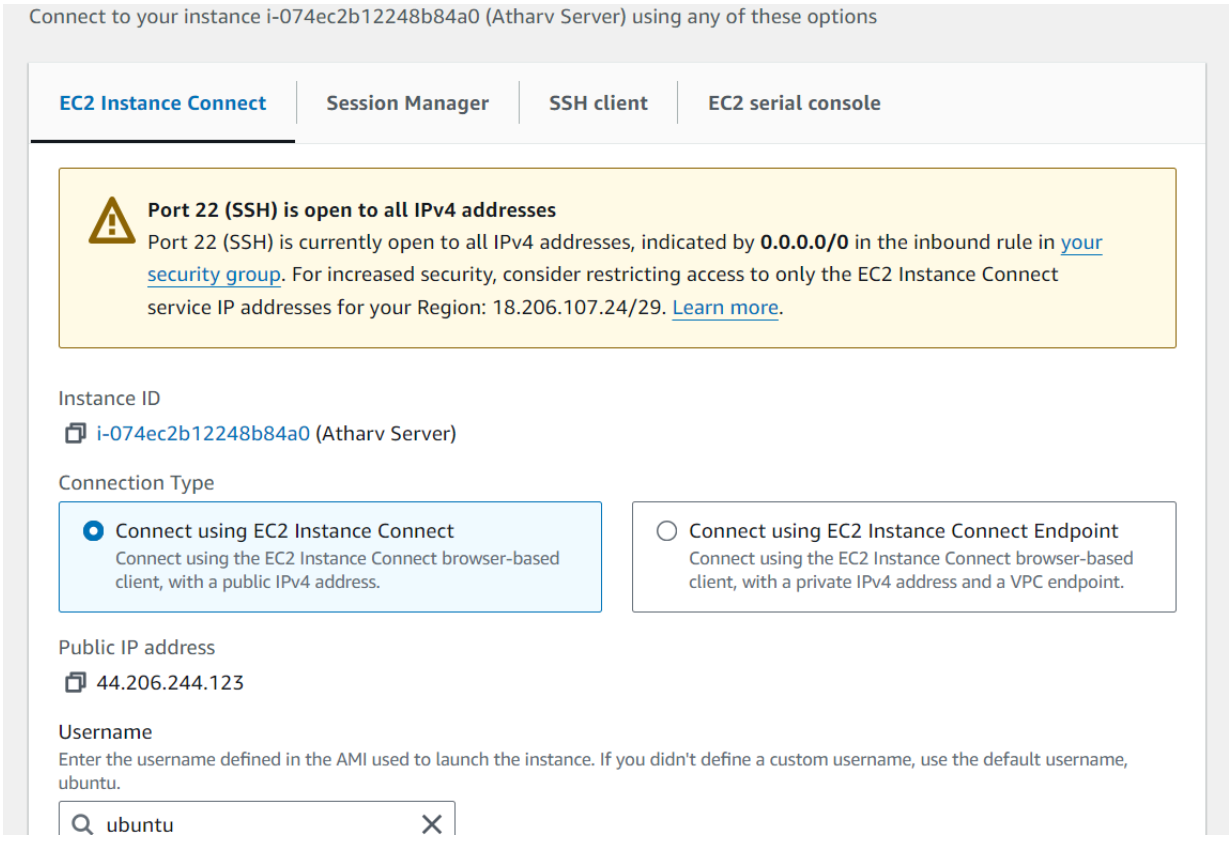
7)see your running instances



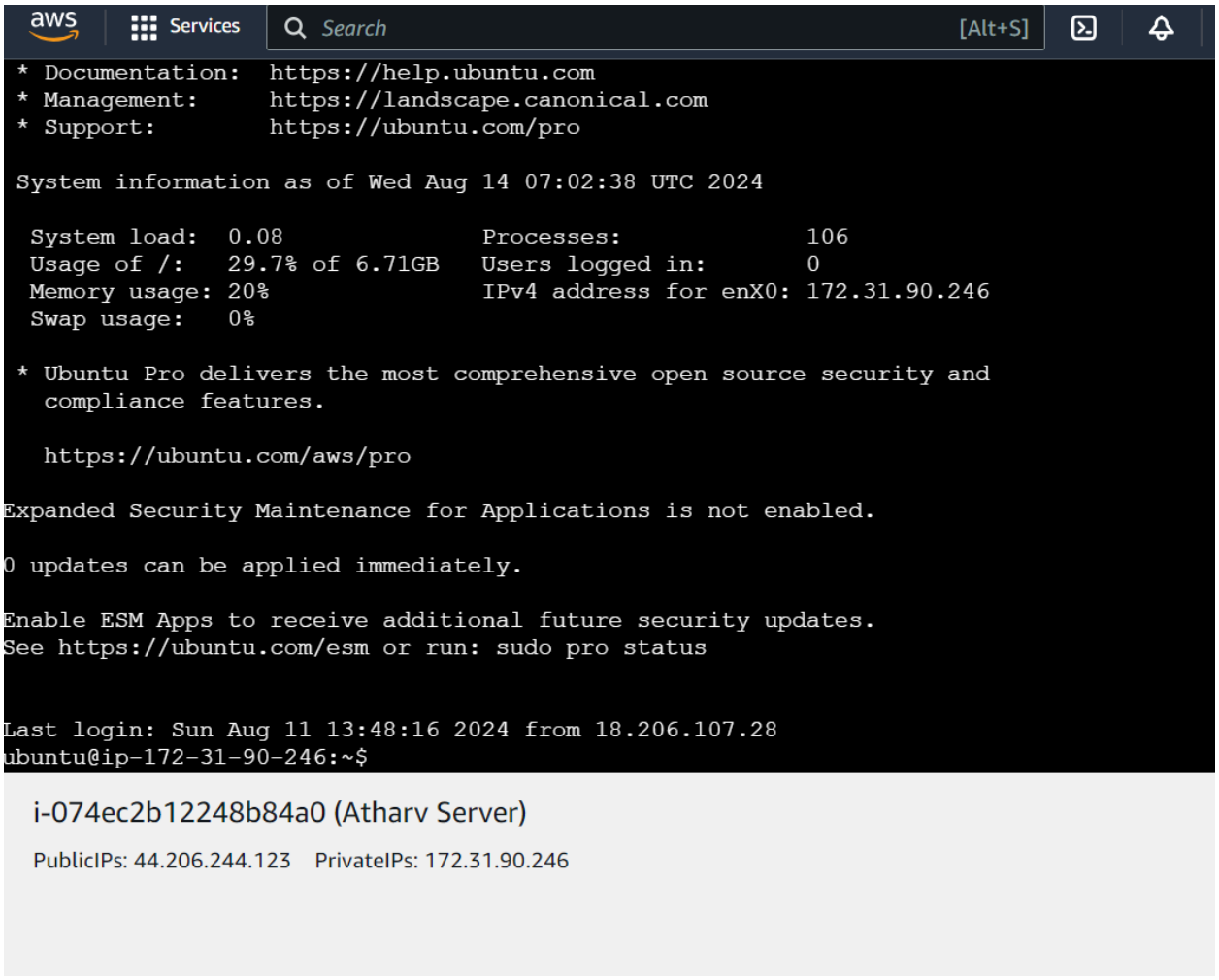
8)click on connect



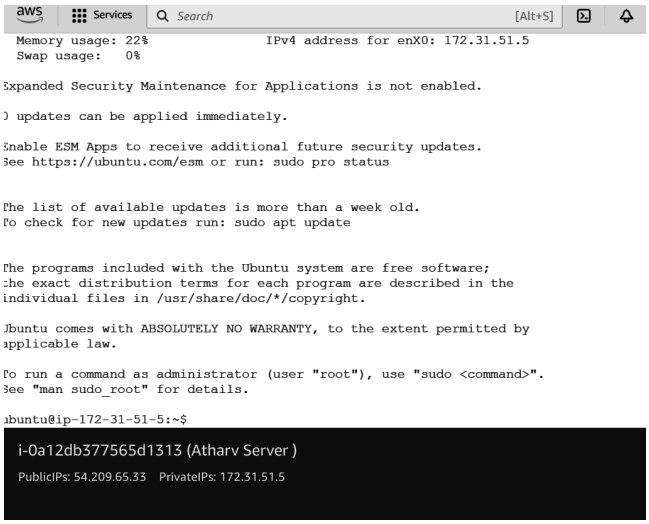
9)you will see this page



10)this console will open

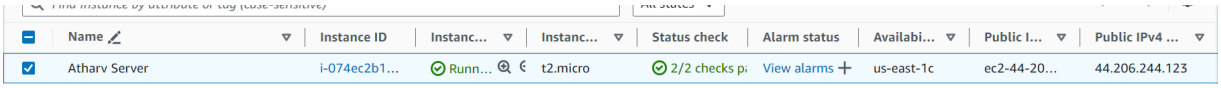


11)Run all the commands



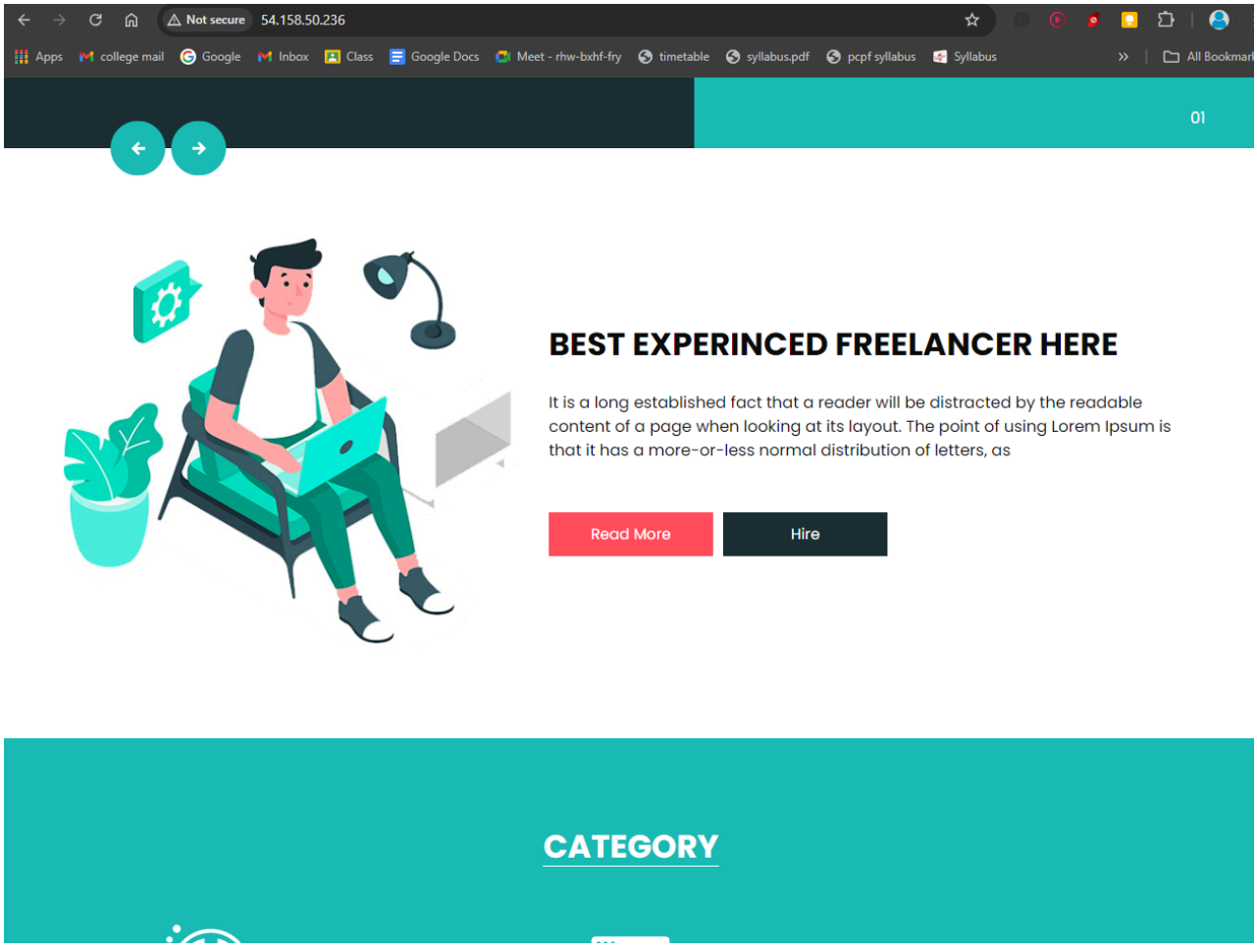
```
root@ip-172-31-90-246:~/temp# ls
spering-html  spering.zip
root@ip-172-31-90-246:~/temp# ls -lrt
total 552
drwxr-xr-x 5 root root   4096 Sep 16  2020 spering-html
-rw-r--r-- 1 root root 557415 Aug 20  2021 spering.zip
root@ip-172-31-90-246:~/temp# cd spering-html
root@ip-172-31-90-246:~/temp/spering-html# ls -lrt
total 72
-rw-r--r-- 1 root root 23212 Jul 28  2020 index.html
-rw-r--r-- 1 root root 10108 Jul 28  2020 about.html
-rw-r--r-- 1 root root  9824 Jul 28  2020 category.html
-rw-r--r-- 1 root root 11825 Jul 28  2020 work.html
drwxr-xr-x 2 root root  4096 Sep 16  2020 js
drwxr-xr-x 2 root root  4096 Sep 16  2020 images
drwxr-xr-x 2 root root  4096 Sep 16  2020 css
root@ip-172-31-90-246:~/temp/spering-html# mv * /var/www/html/
root@ip-172-31-90-246:~/temp/spering-html# cd /var/www/html/
root@ip-172-31-90-246:/var/www/html# ls -lrt
total 72
-rw-r--r-- 1 root root 23212 Jul 28  2020 index.html
-rw-r--r-- 1 root root 10108 Jul 28  2020 about.html
-rw-r--r-- 1 root root  9824 Jul 28  2020 category.html
-rw-r--r-- 1 root root 11825 Jul 28  2020 work.html
drwxr-xr-x 2 root root  4096 Sep 16  2020 js
drwxr-xr-x 2 root root  4096 Sep 16  2020 images
drwxr-xr-x 2 root root  4096 Sep 16  2020 css
root@ip-172-31-90-246:/var/www/html#
```

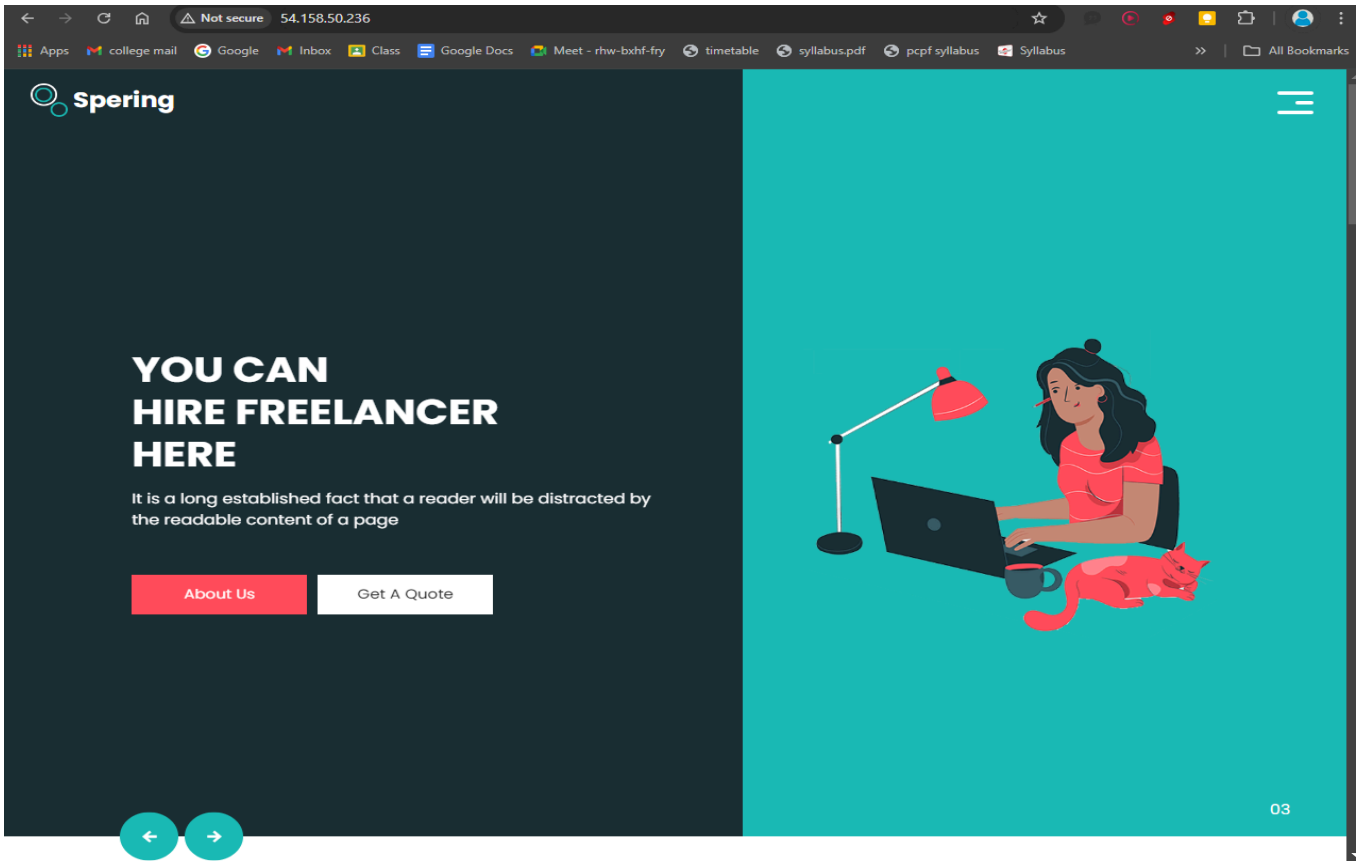
12)Enter the public domain from here



Name	Instance ID	Instanc...	Instanc...	Status check	Alarm status	Availabi...	Public I...	Public IPv4 ...
Atharv Server	i-074ec2b1...	Runn...	t2.micro	2/2 checks p	View alarms	us-east-1c	ec2-44-20...	44.206.244.123

13)Enter the domain and open it on your browser and you will see the website

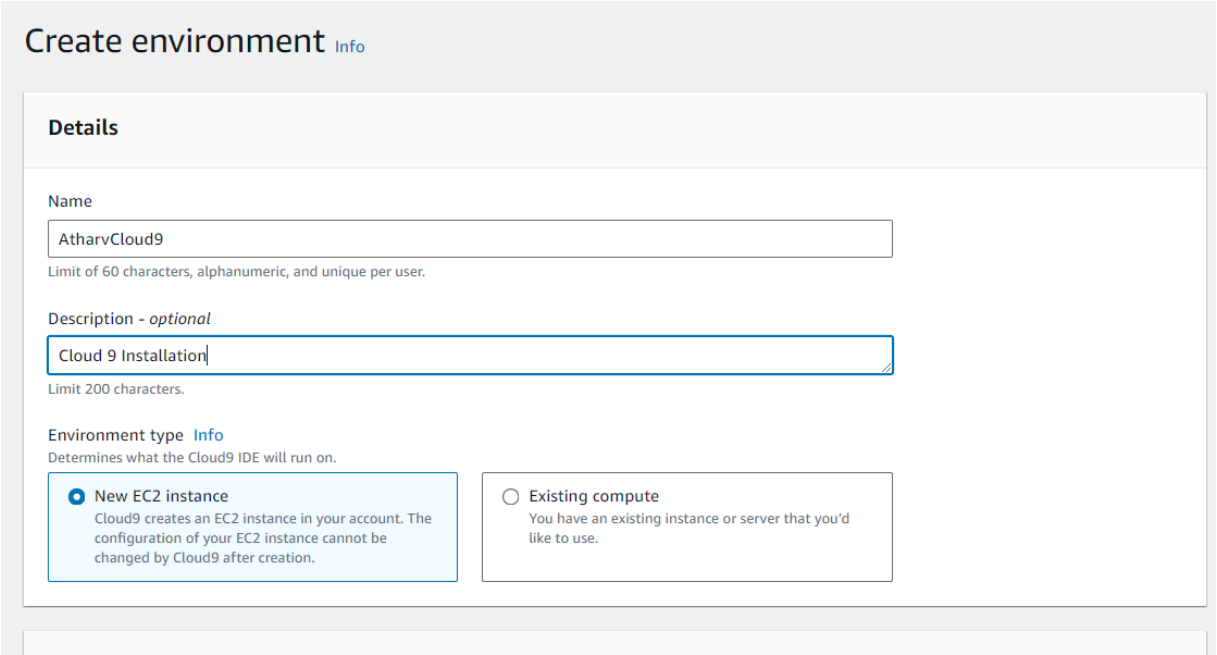



Not secure
54.158.50.236
Apps
college mail
Google
Inbox
Class
Google Docs
Meet - rhw-bxhf-fry
timetable
syllabus.pdf
pcpf syllabus
Syllabus
All Bookmarks
Spering
YOU CAN
HIRE FREELANCER
HERE
It is a long established fact that a reader will be distracted by the readable content of a page
About Us
Get A Quote
03

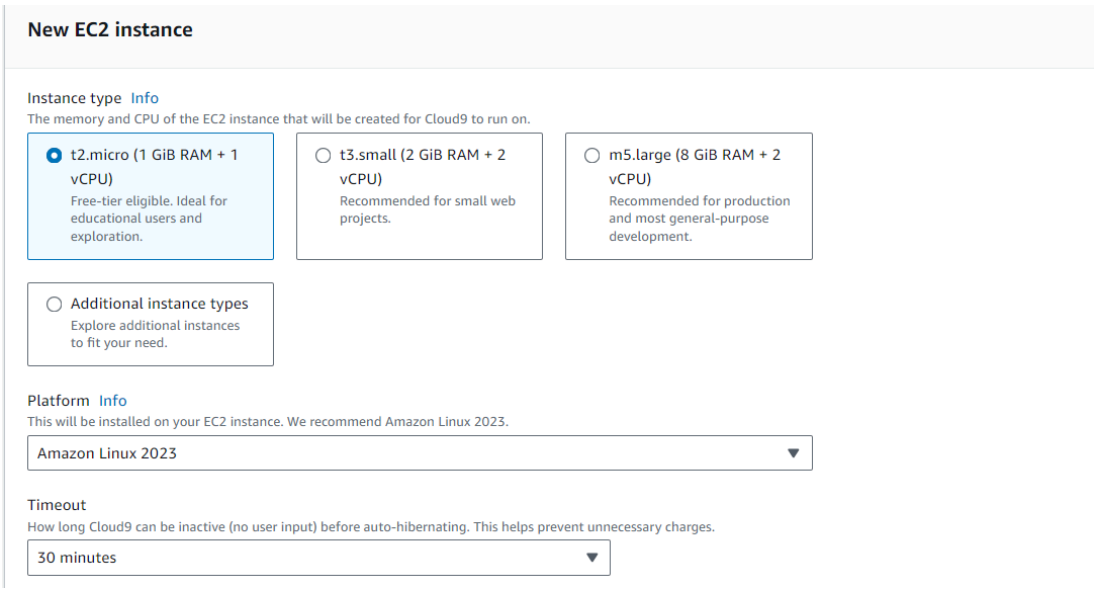
Name:Atharv Nikam **Div:D15C** **Roll:36**

Experiment 1B
Aim : AWS (EC2) Installation steps for Linux instance

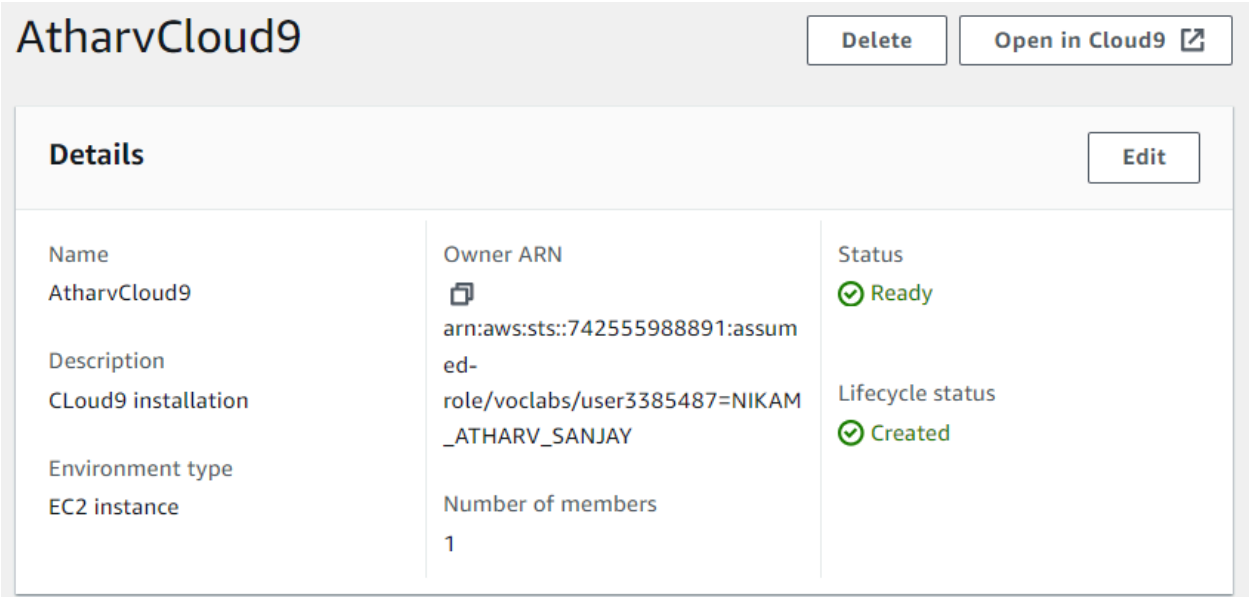
1. Open your AWS account and search for Cloud9 service inside Developer tools. Create a new Cloud9 environment by filling in the required details. Make sure you use an EC2 instance to create your environment.

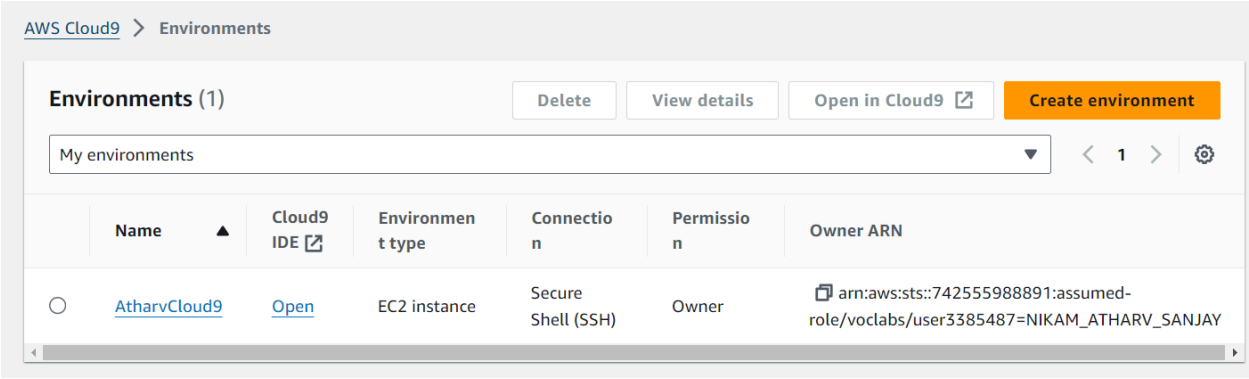


2)Select T2 Micro

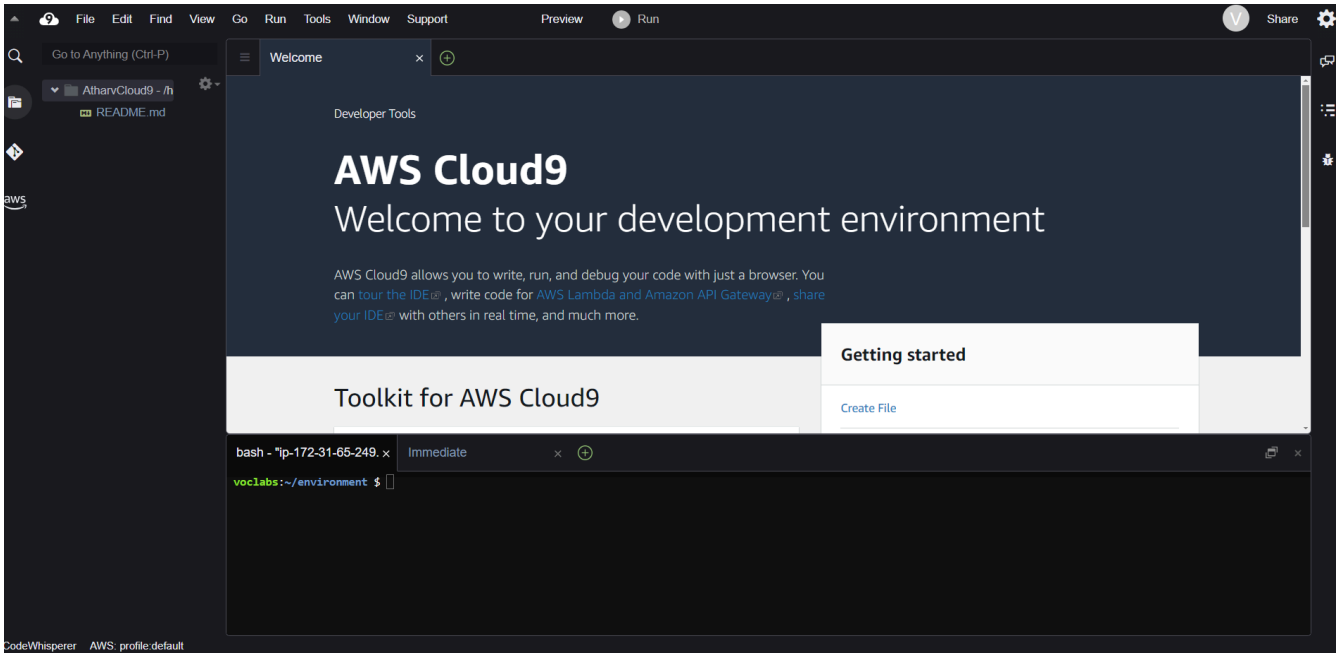


3)See your summary

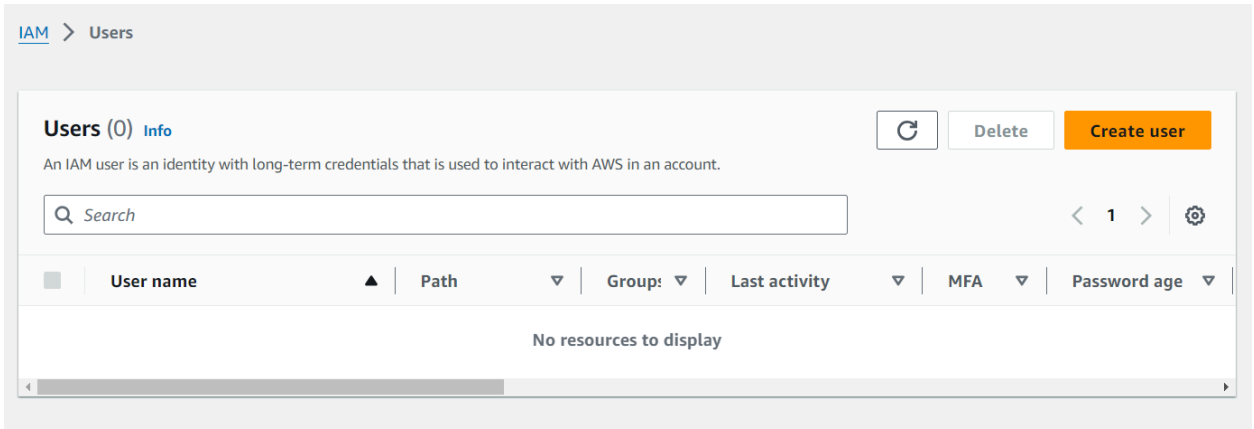




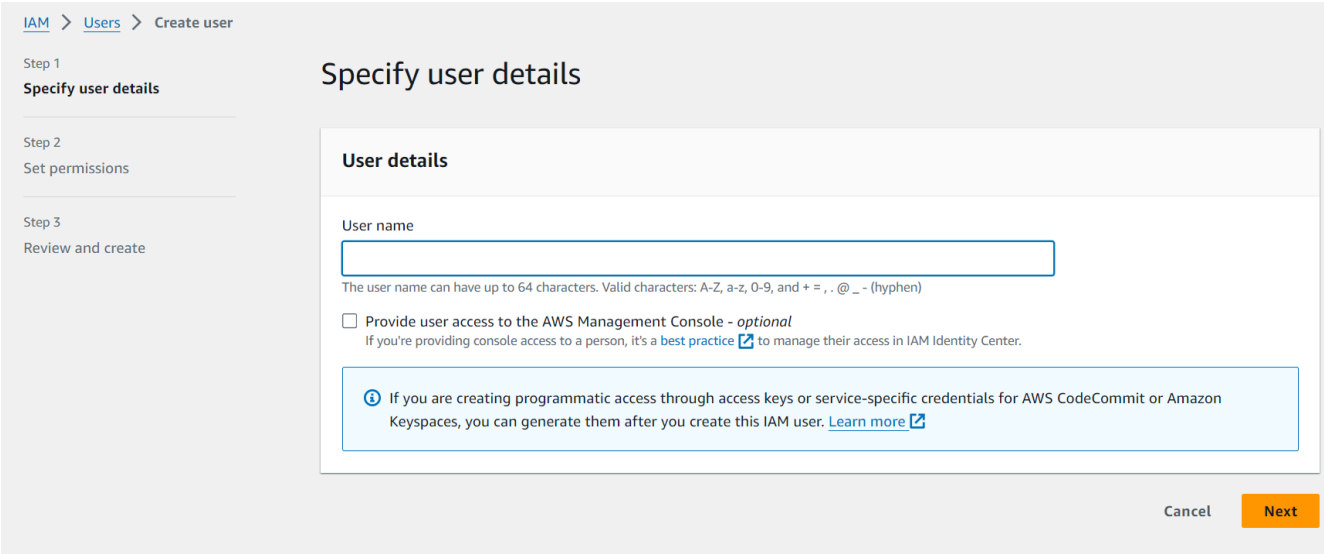
4)Your Aws Cloud9 Console will open



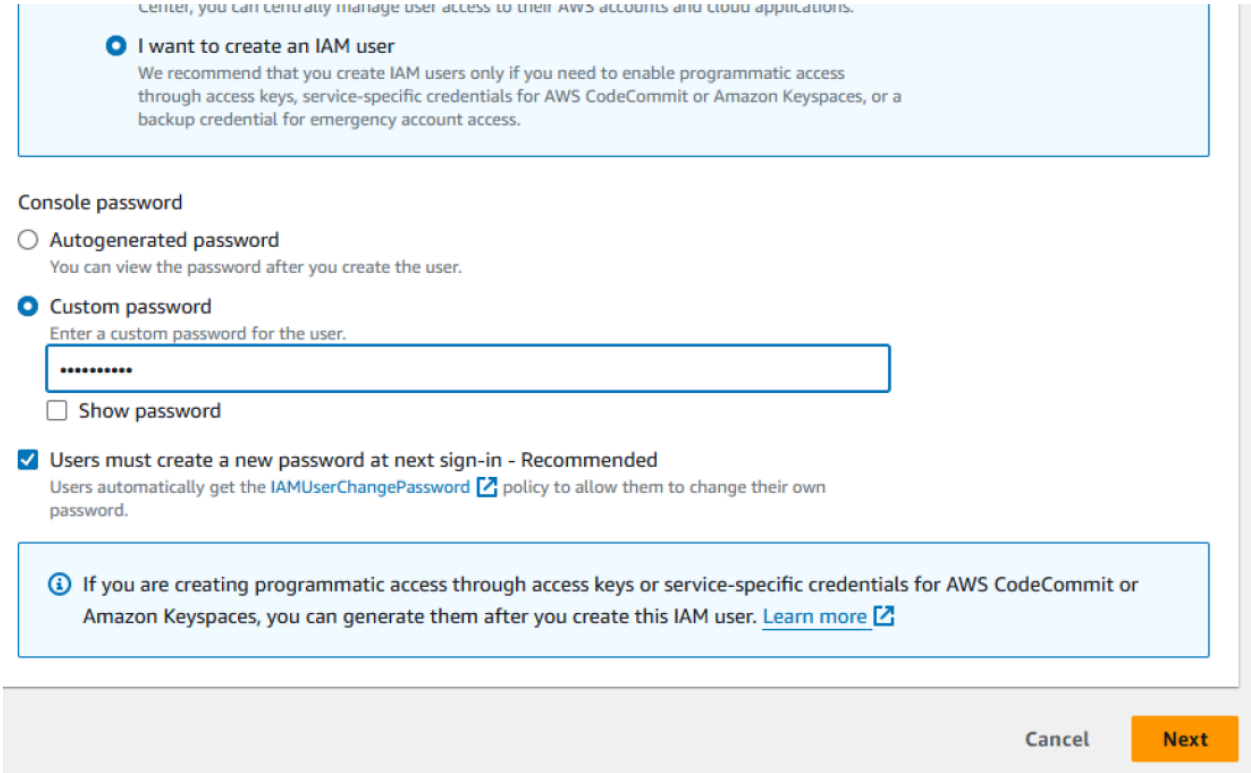
5)Click On IAM and create a new user



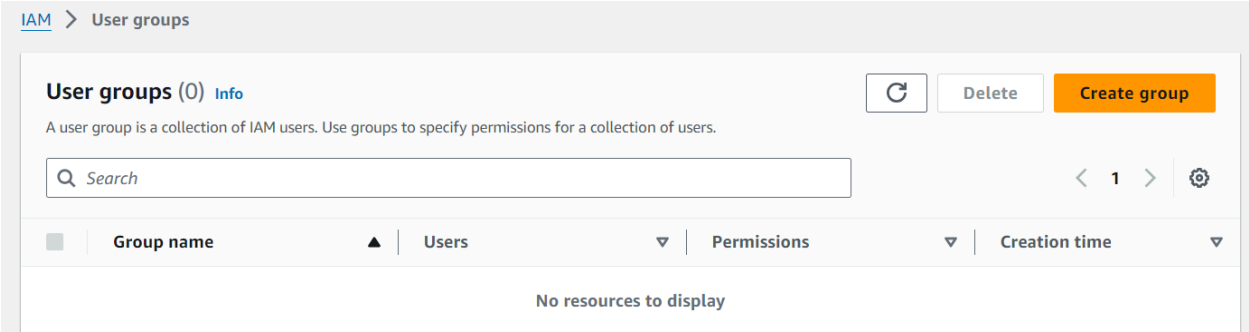
6)Enter your userName



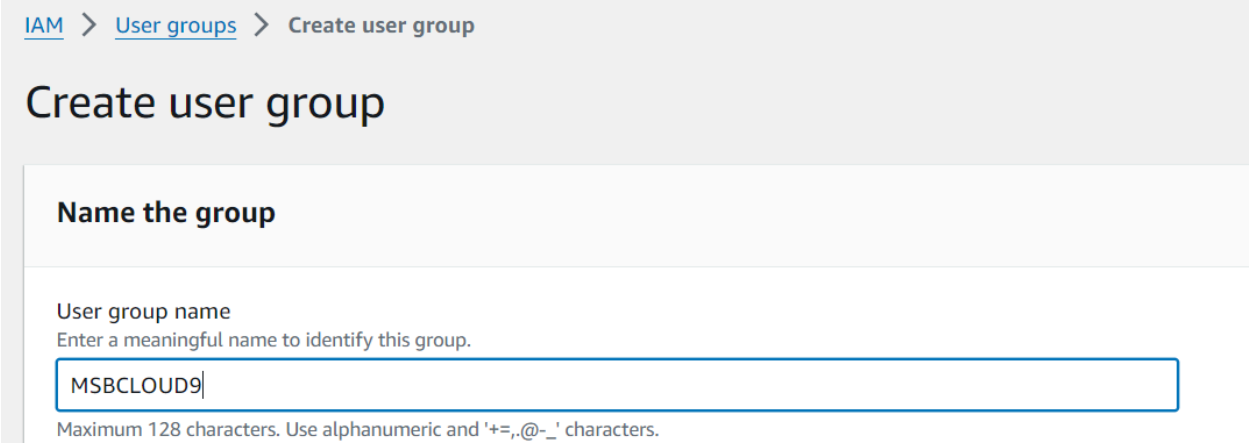
7)Enter a Password



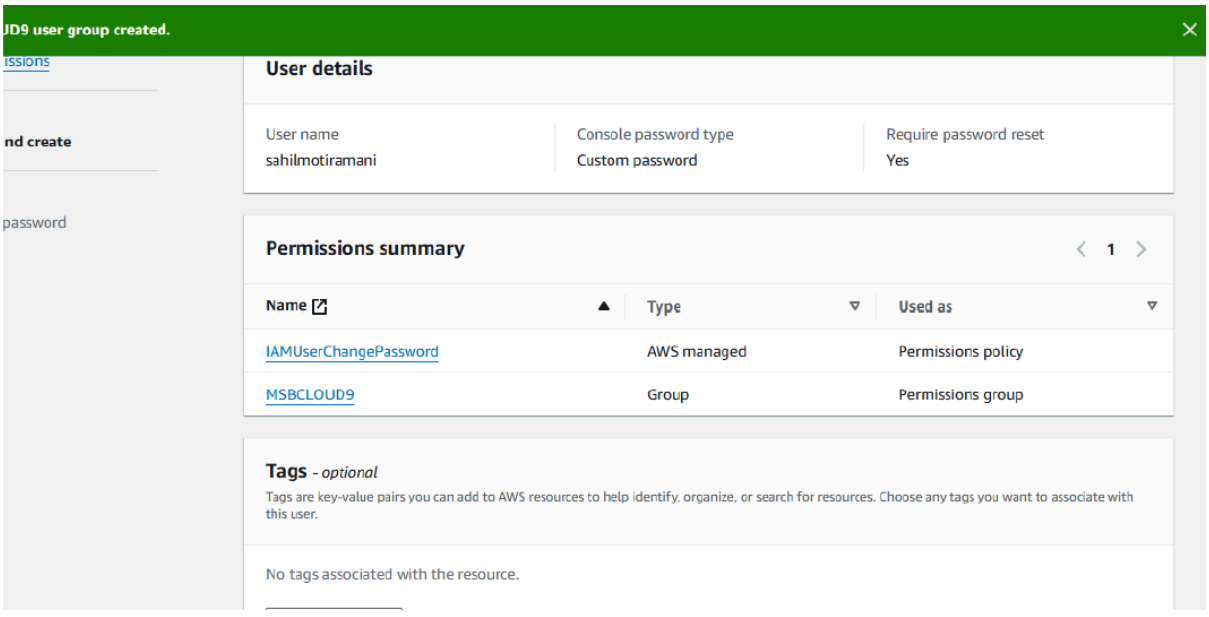
8)Now similarly Create a group



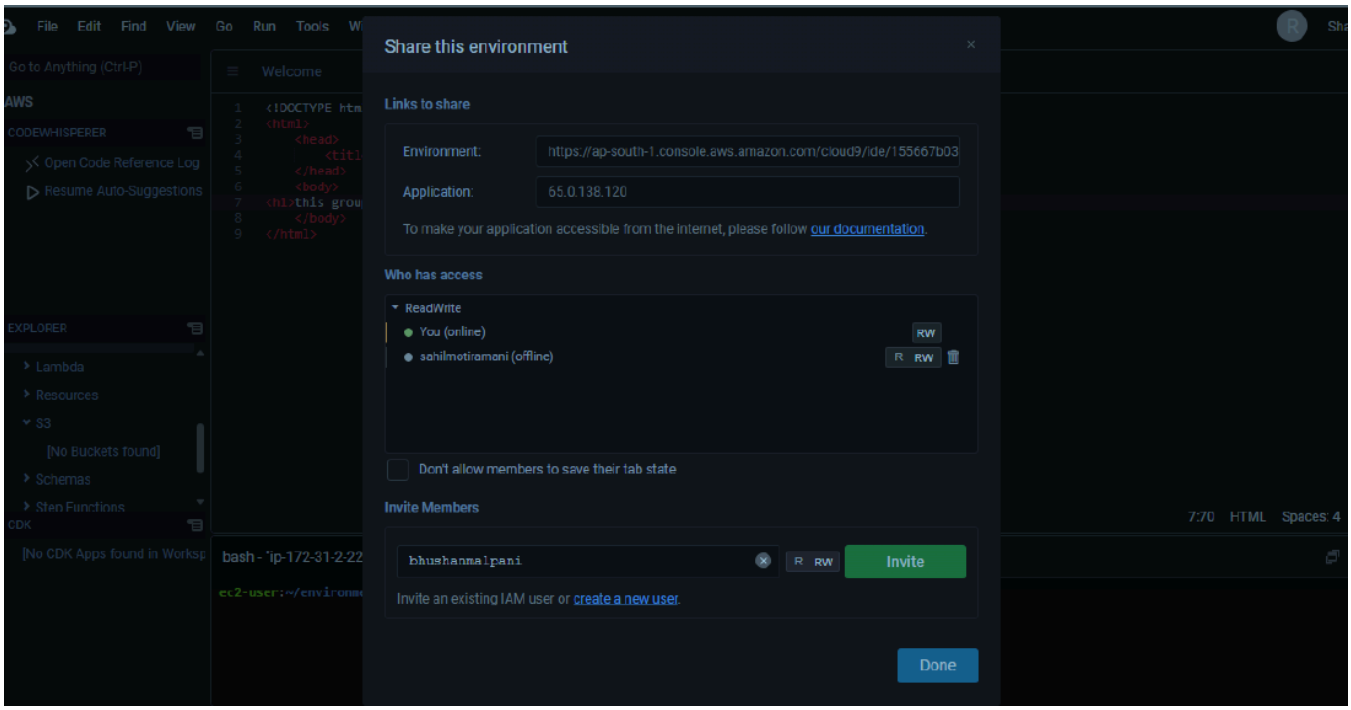
9)give a name



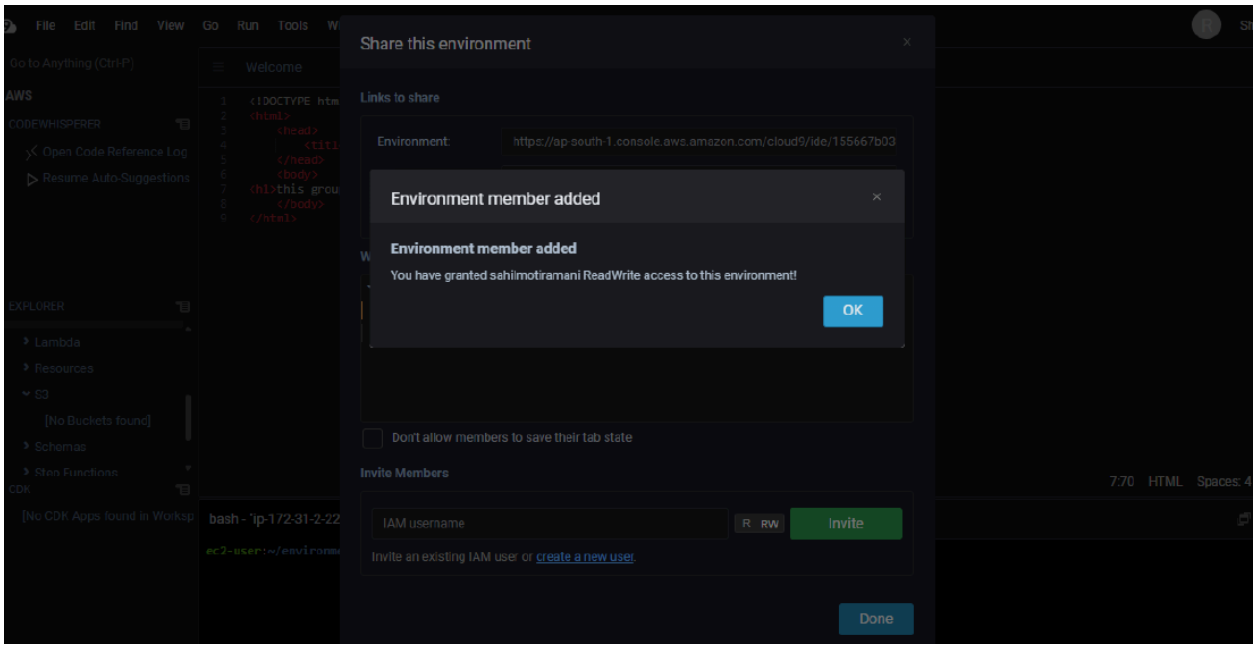
10)The user has successfully been created i.e There is a custom made username and a password for the IAM user.



11)Now you can share your environment now you can add collaborators



12)New Member added



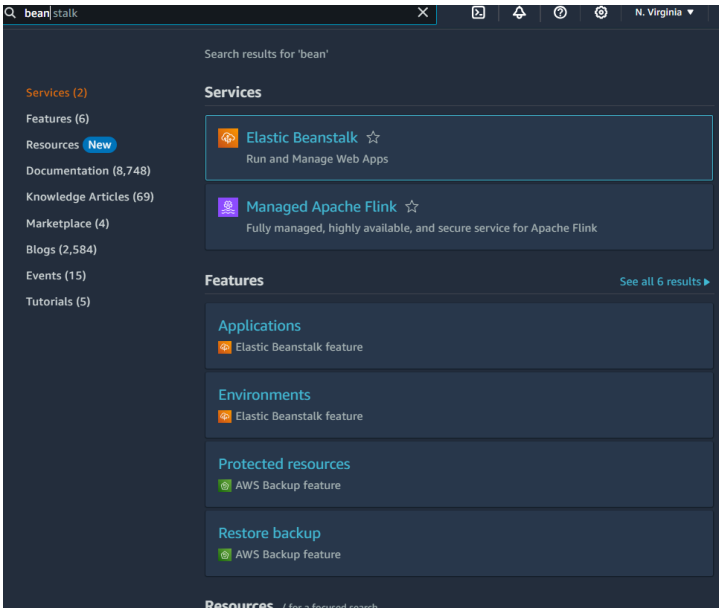
We were required to log in from another browser using the IAM user's credentials to gain access to the shared Cloud9 environment. Unfortunately, we were unable to complete these steps because the Cloud9 services were disrupted, which also blocked remote access to the IAM user account. This disruption has prevented us from performing the necessary actions, leaving us unable to access the shared environment as intended.

Name:Atharv Sanjay Nikam **Div:15C** **Roll no : 36**

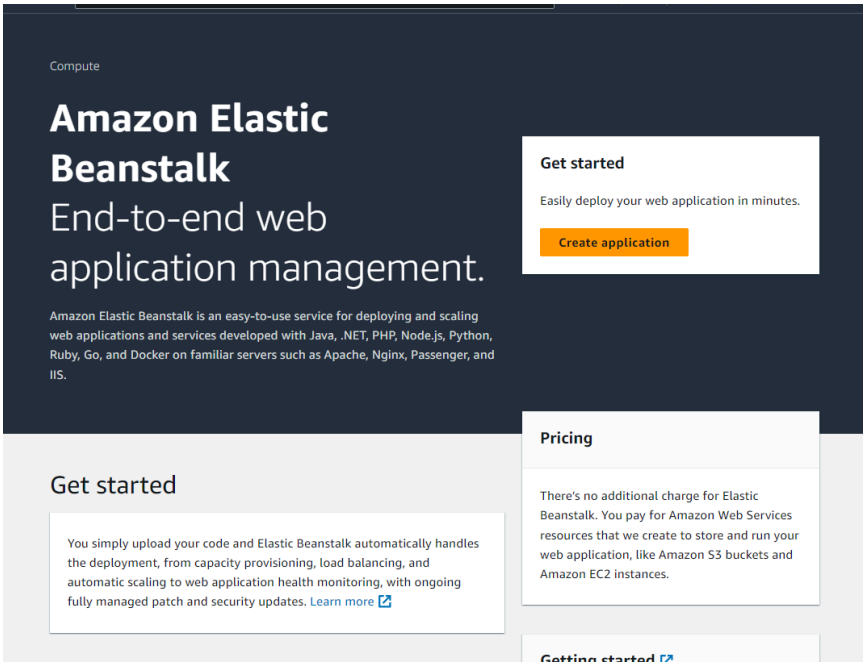
Exp 2

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

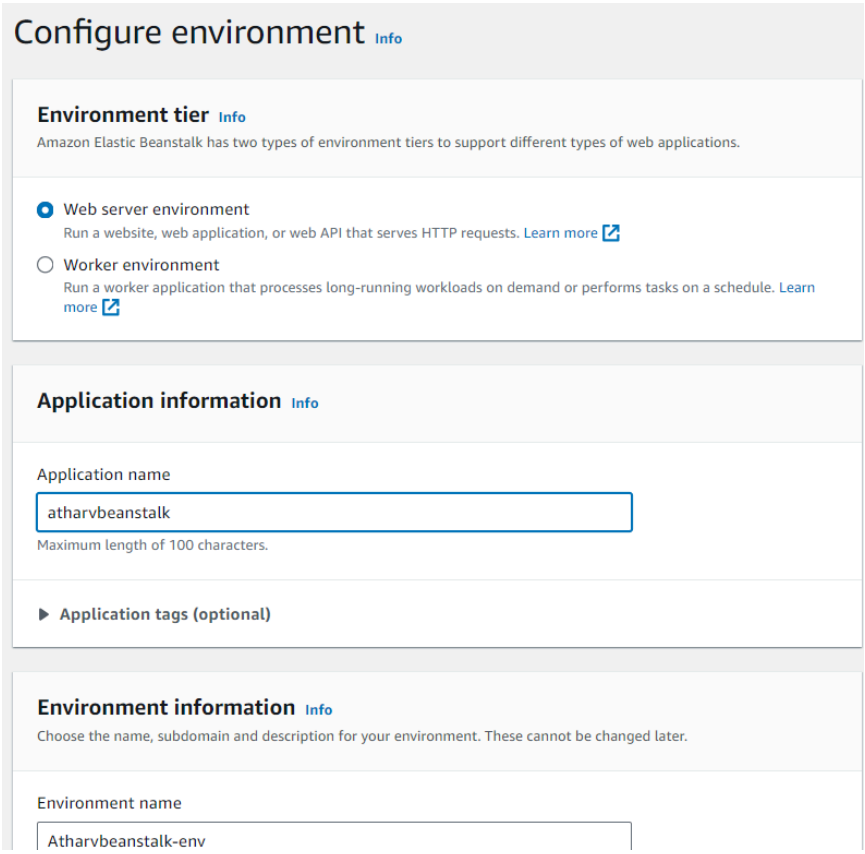
1)Login to your account and search for Elastic Beanstalk



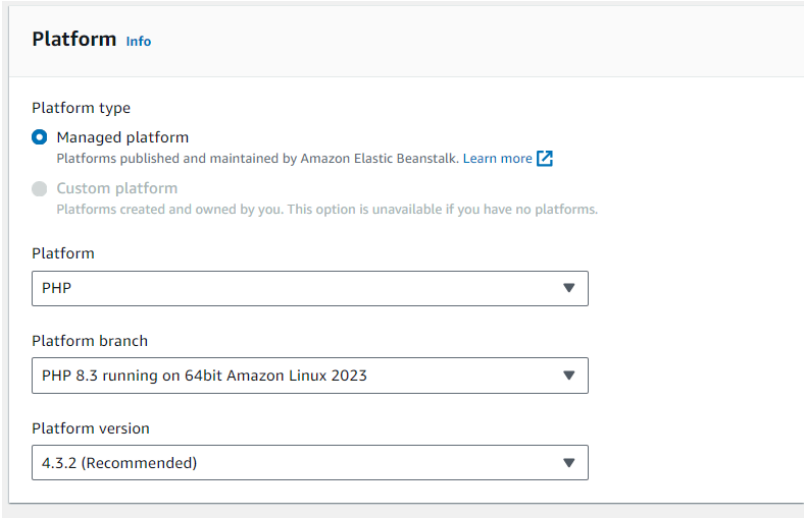
2)Click on it and select create application



3)Give a appropriate name to you application

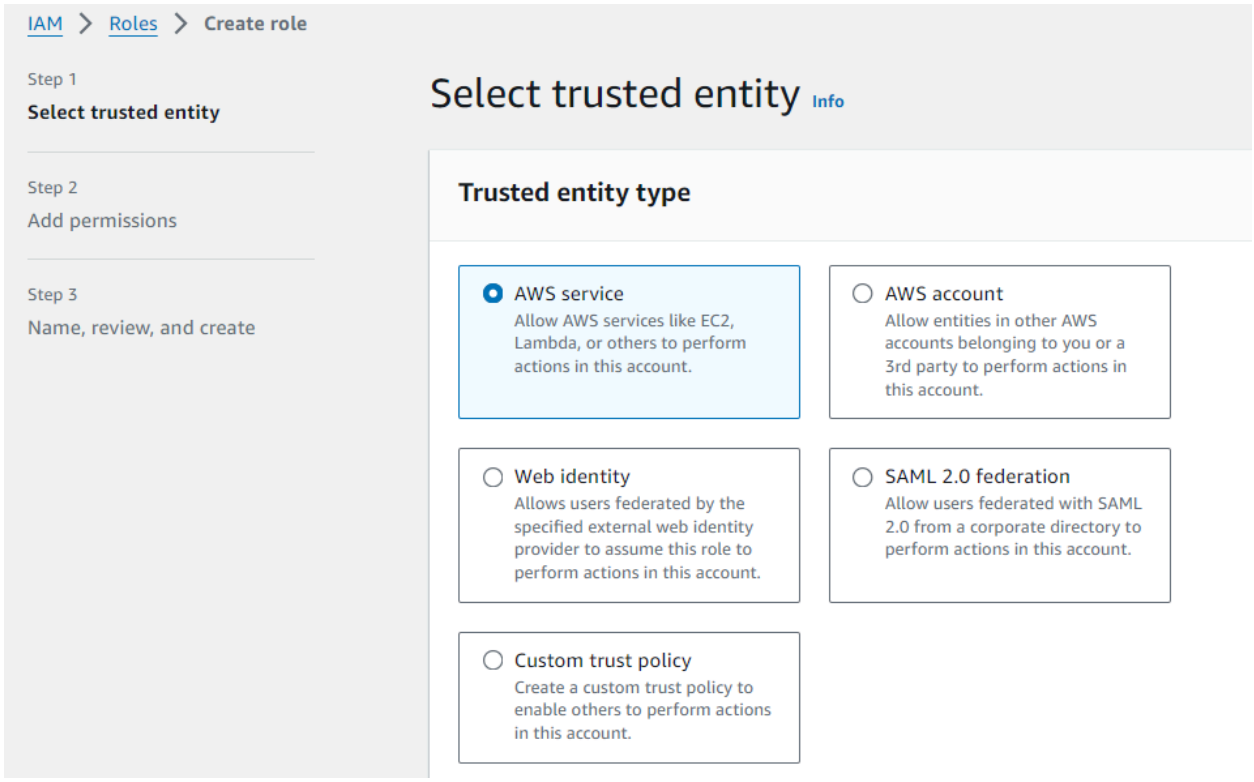


4)Select php and other will automatically get filled

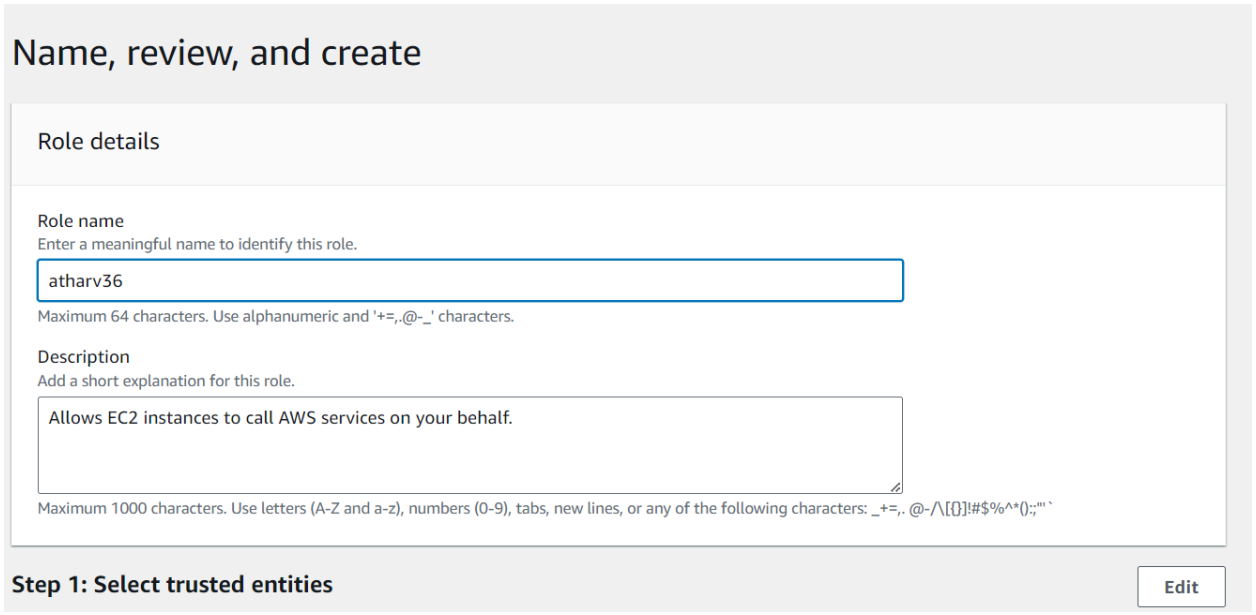


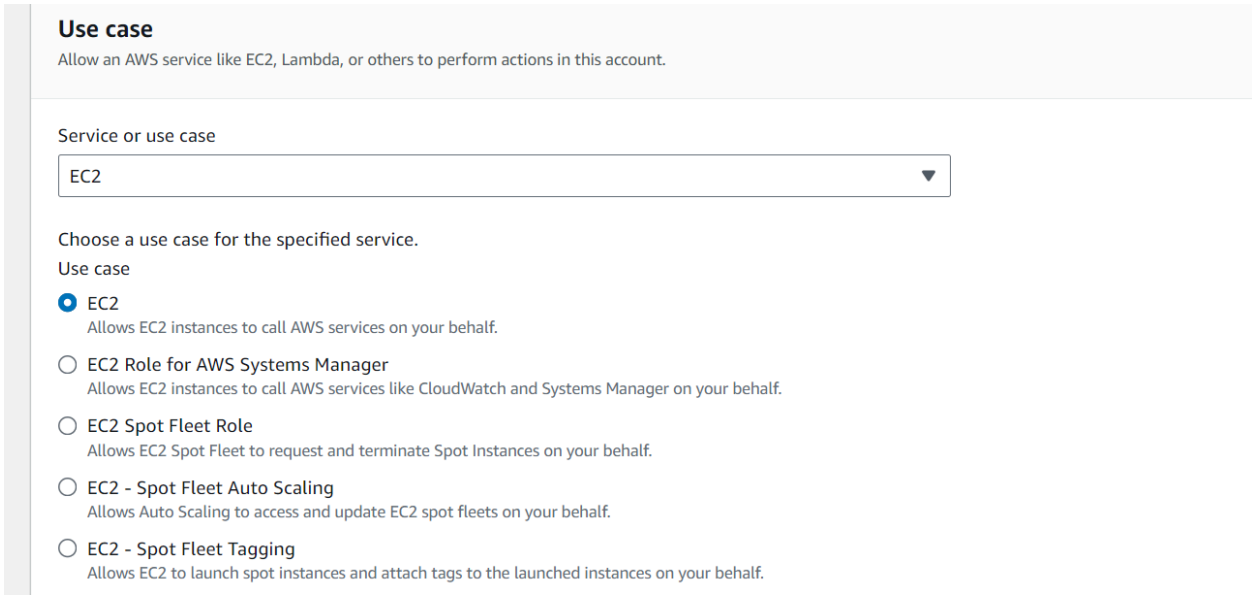
5)Now, while creating the environment, we are asked to provide an IAM role with the necessary
EC2 permissions. We are supposed to make sure that we have made an existing IAM role with
the following set of permissions:
1. AWSElasticBeanStalkWebTier
2. AWSElasticBeanStalkWorkerTier
3. AWSElasticBeanStalkMulticontainerDocker

Select iam and then create role

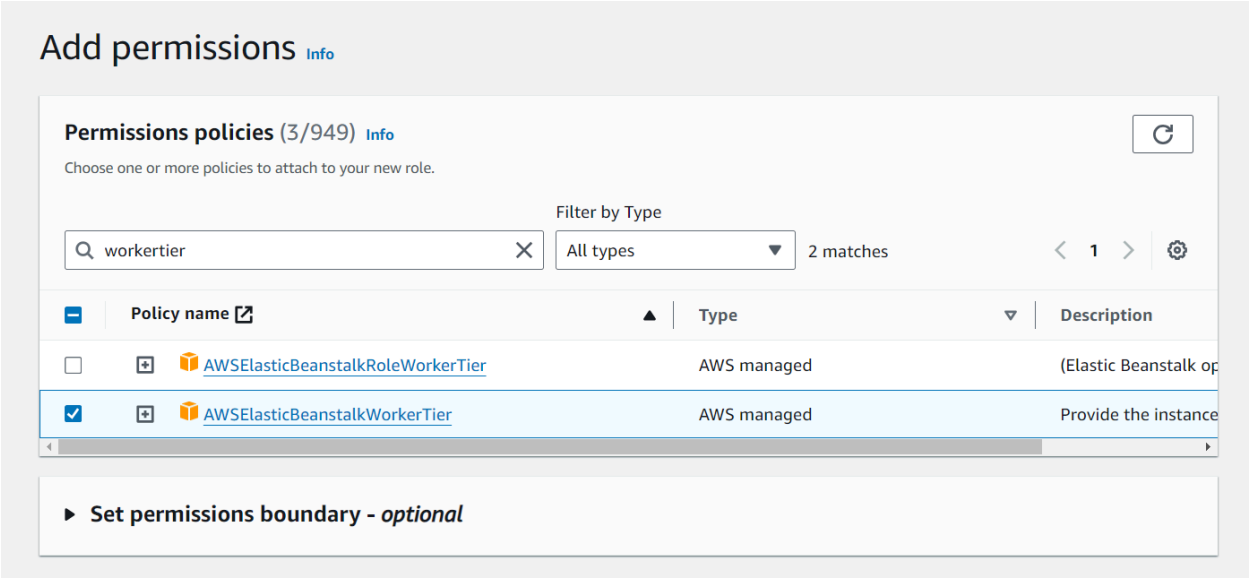


5)give a name and then select ec2

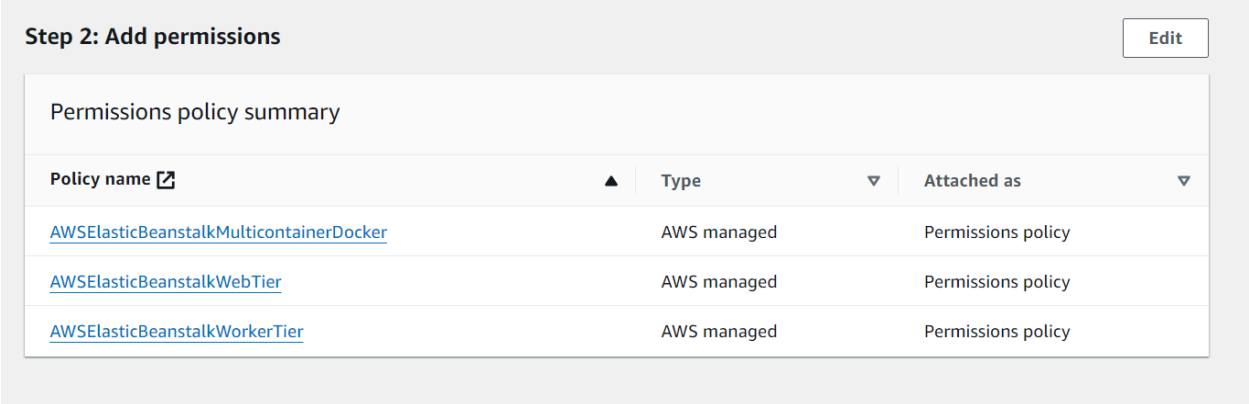

Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
atharv36
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"' `
Step 1: Select trusted entities
Edit


Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
EC2
Choose a use case for the specified service.
Use case
EC2
Allows EC2 instances to call AWS services on your behalf.
EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

Add roles

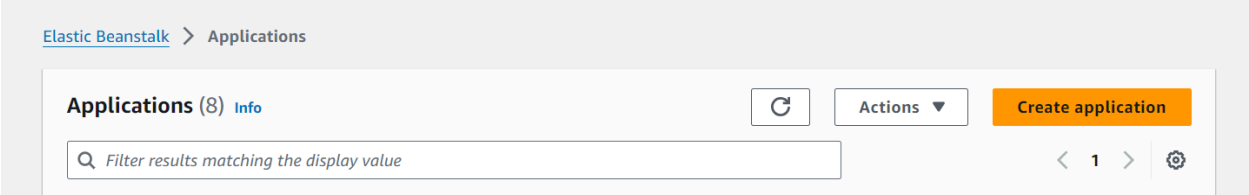


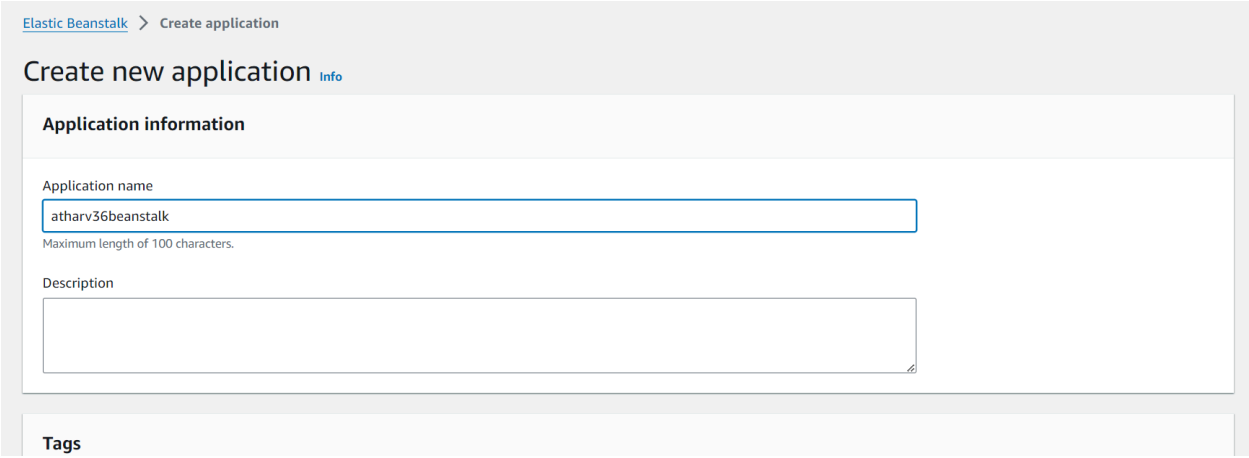
These 3 roles should be added



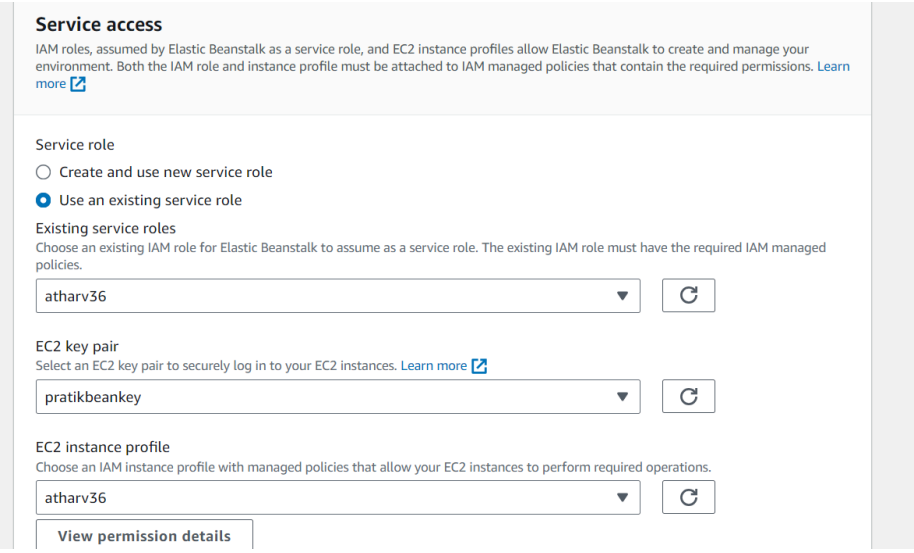
6)Now create an Application

Click on create application

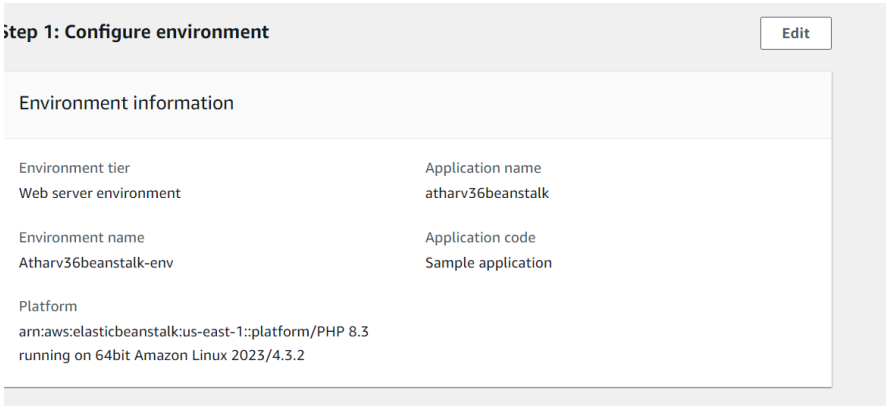




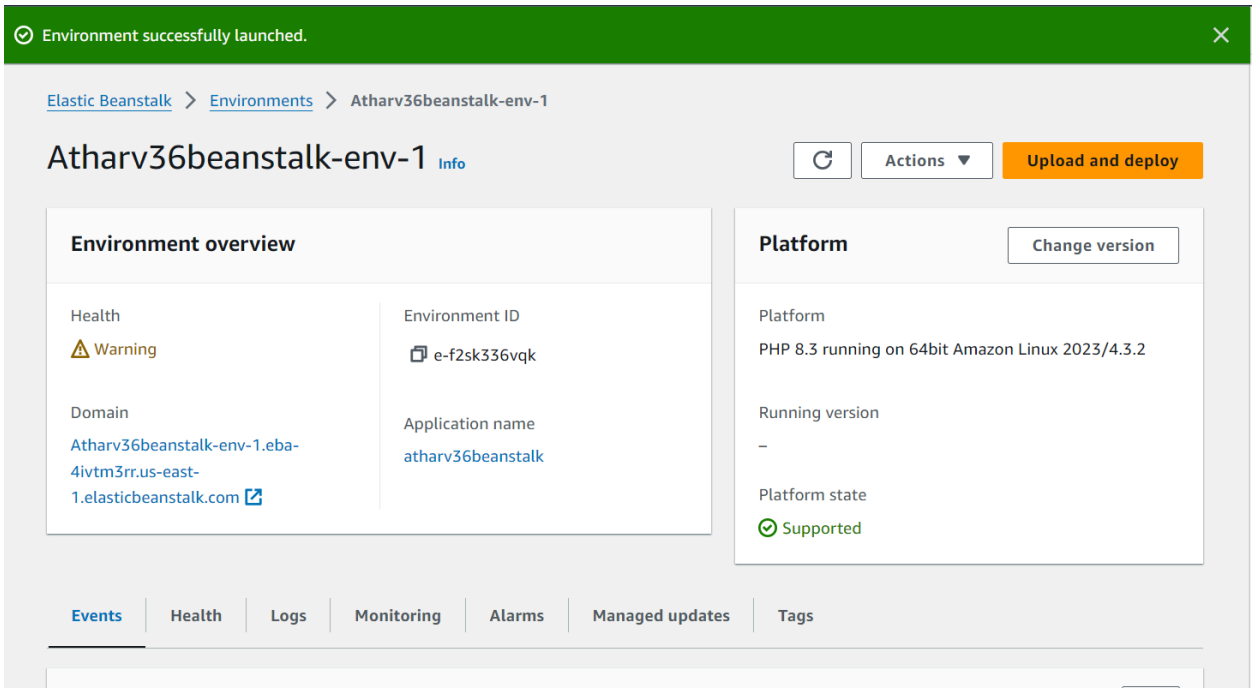
7) in services access give the role we just created



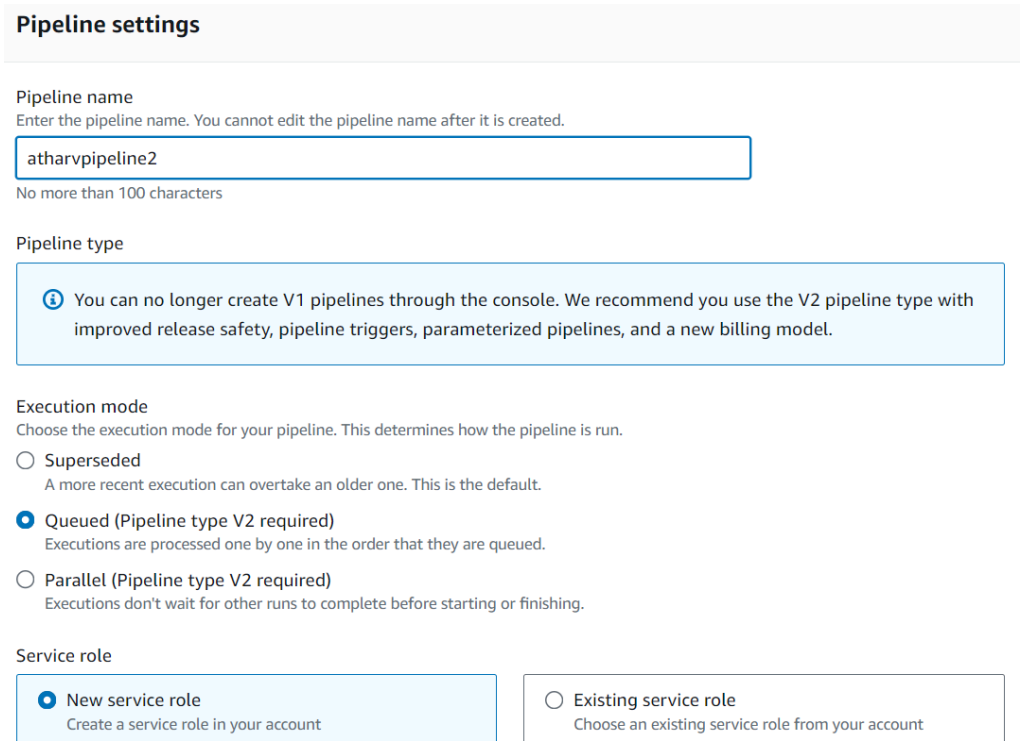
8)See the summary and click on create application



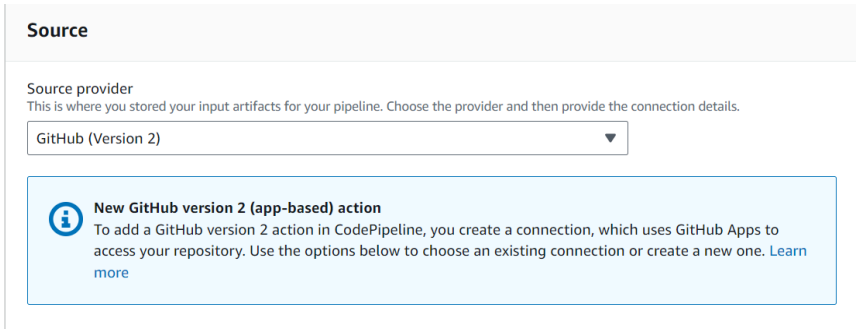
9)then you will get a message that you environment is created



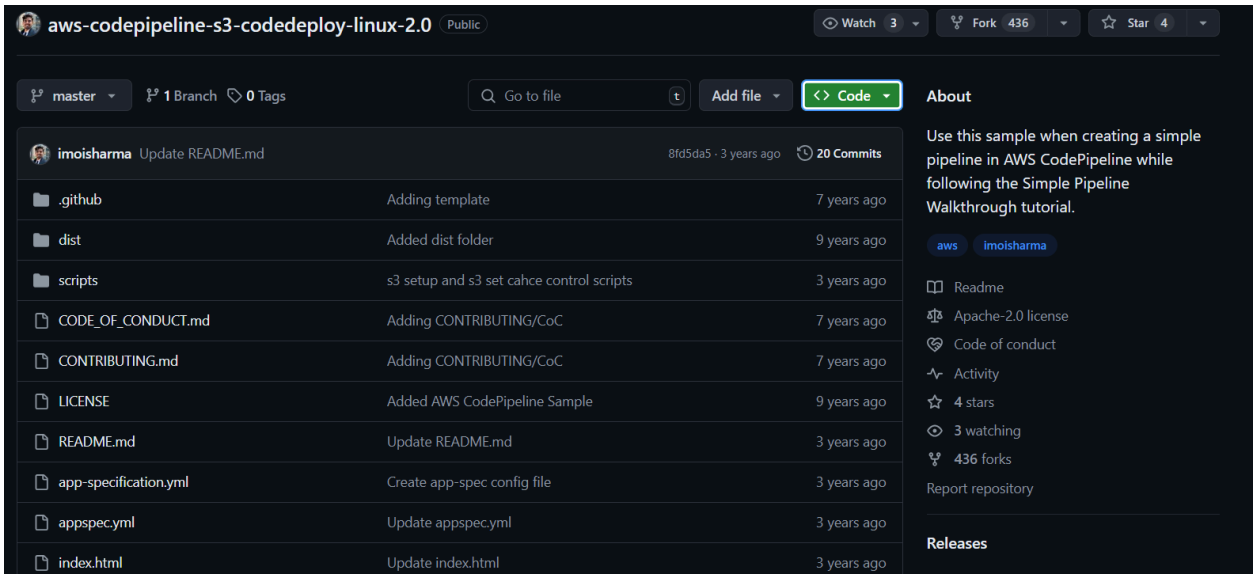
10)Create a pipeline



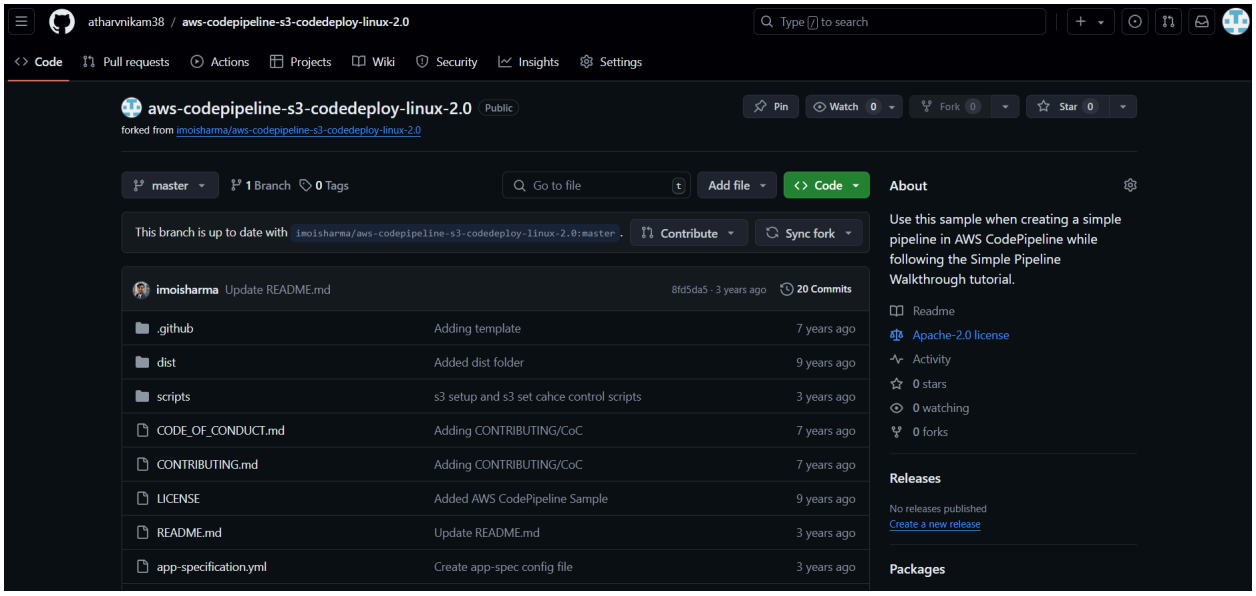
11)select source and select Github version 2



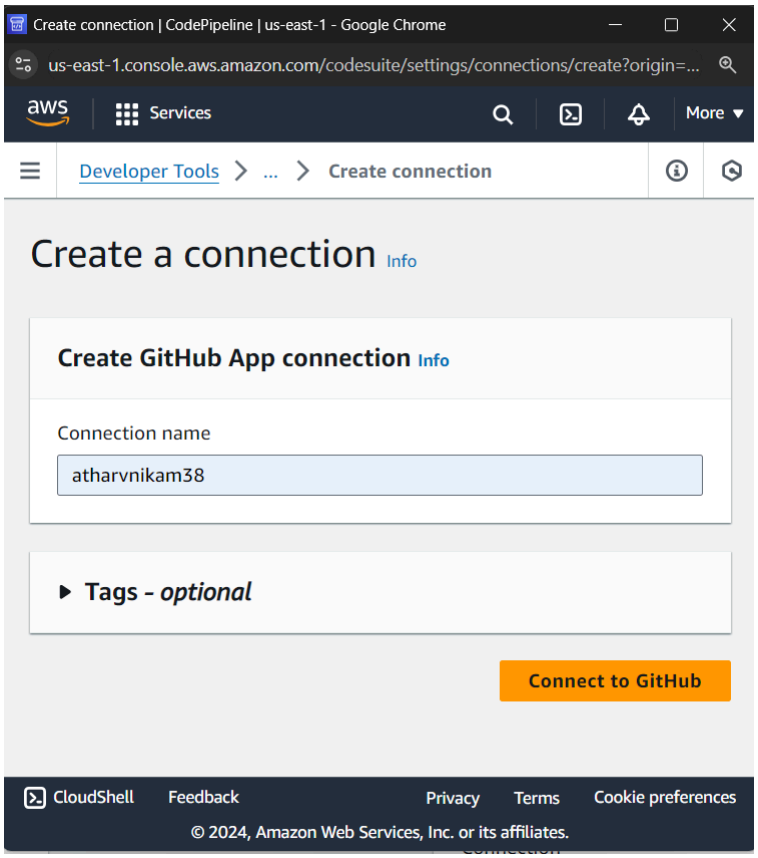
12)First you have to fork a github repo



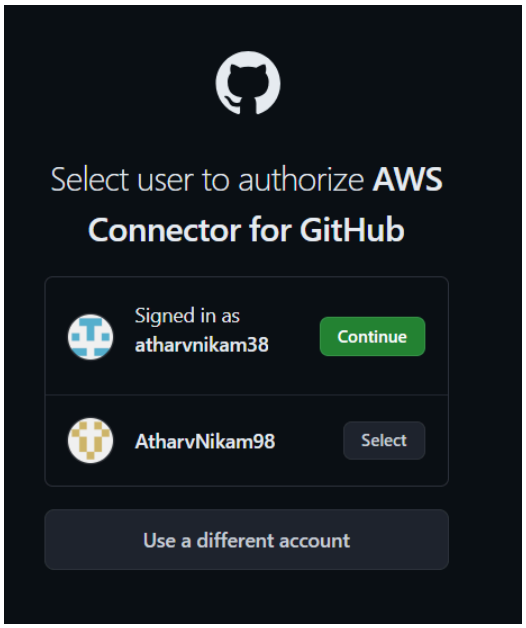
13)forked this image on the your account

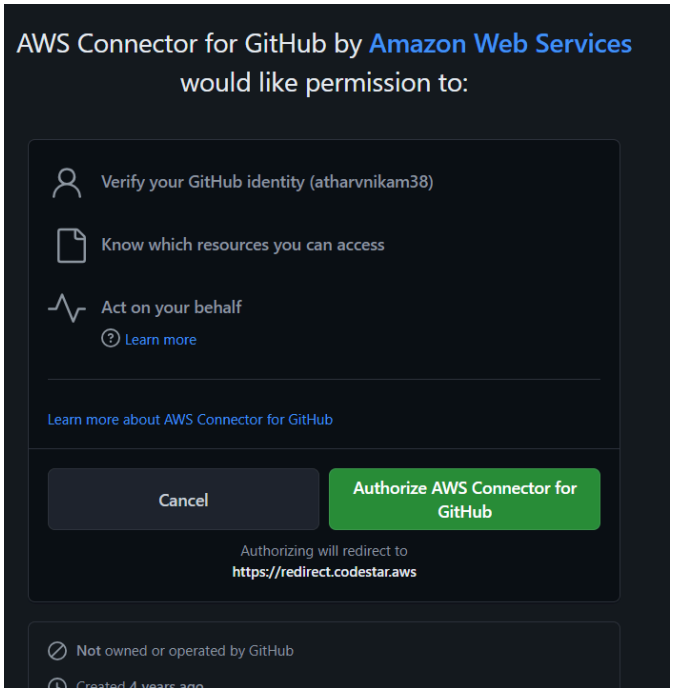


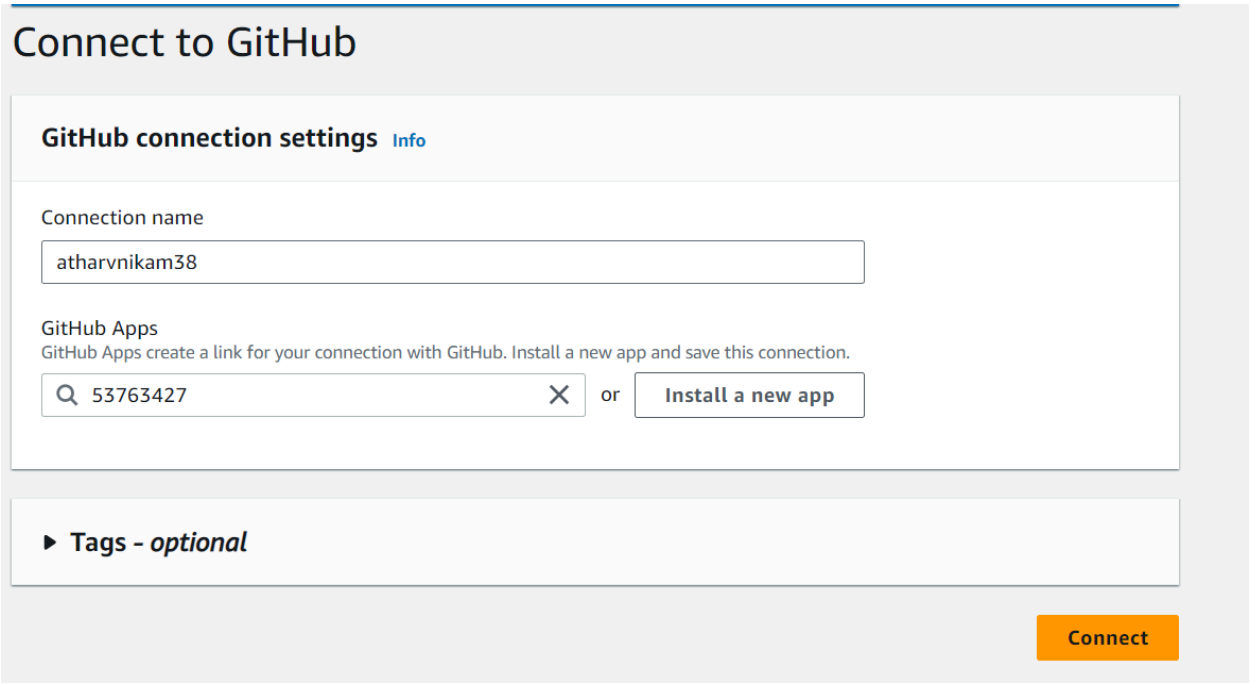
14)Now connect your github repo here



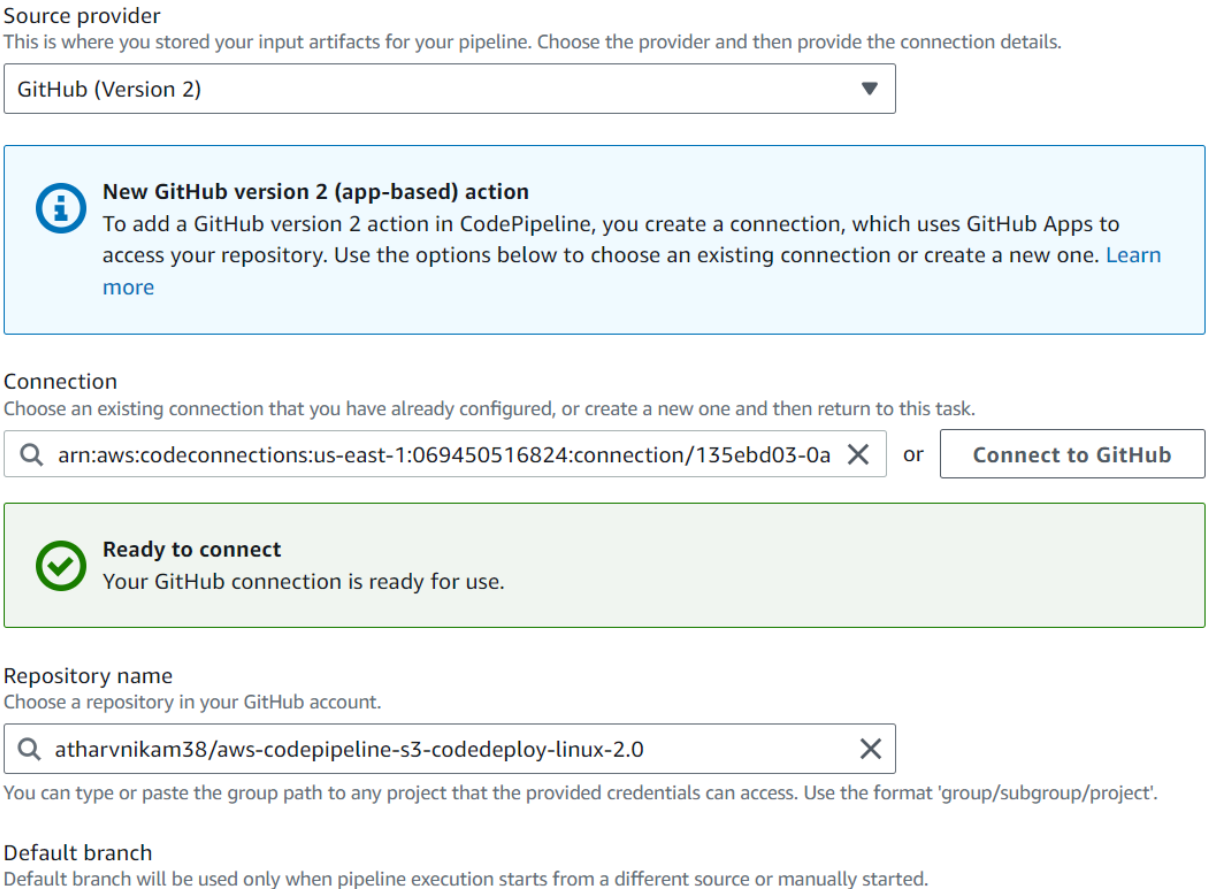
15)Authorize it



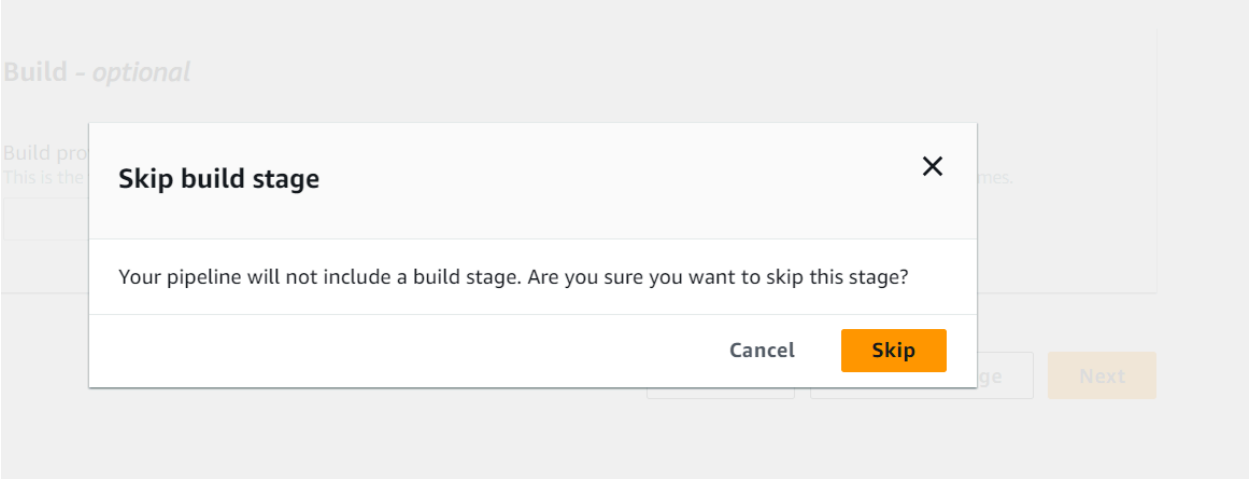

AWS Connector for GitHub by Amazon Web Services would like permission to:
Verify your GitHub identity (atharvnikam38)
Know which resources you can access
Act on your behalf
Learn more
Learn more about AWS Connector for GitHub
Cancel
Authorize AWS Connector for GitHub
Authorizing will redirect to
https://redirect.codestar.aws
Not owned or operated by GitHub


Connect to GitHub
GitHub connection settings Info
Connection name
atharvnikam38
GitHub Apps
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.
53763427
or
Install a new app
Tags - optional
Connect

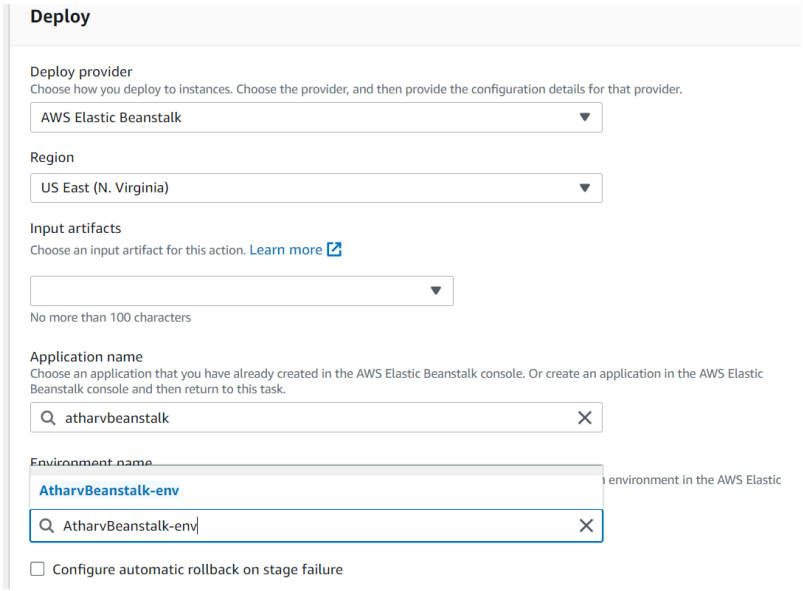
16)now you have connect the repo



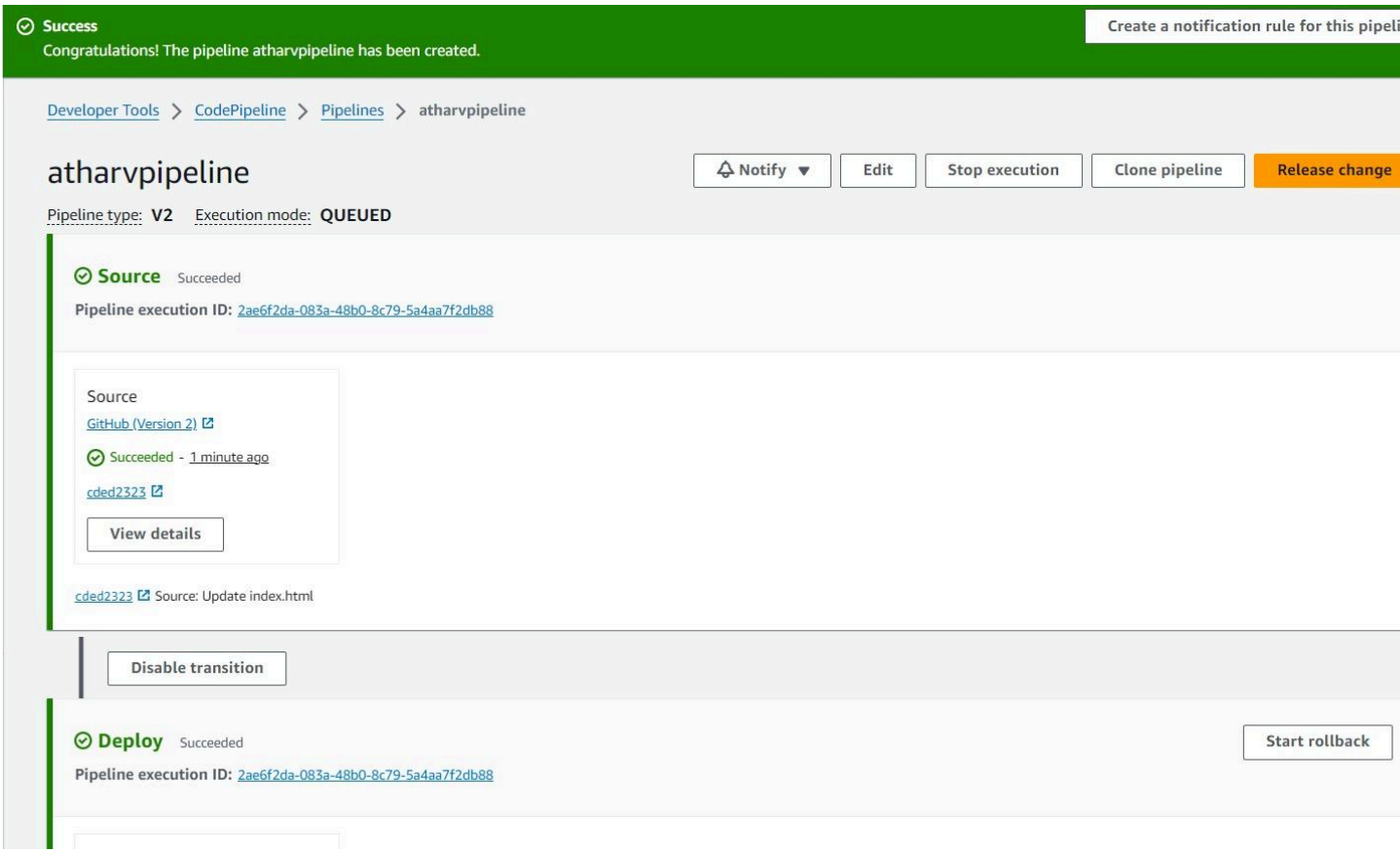
17)click next and click skip build test



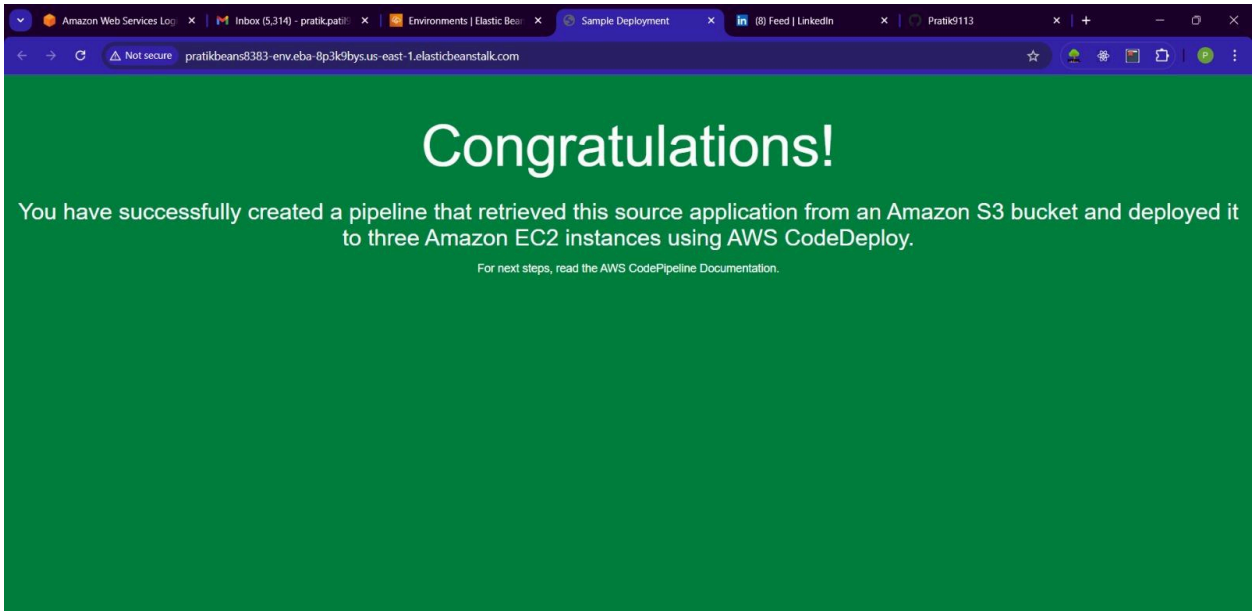
18)enter final info



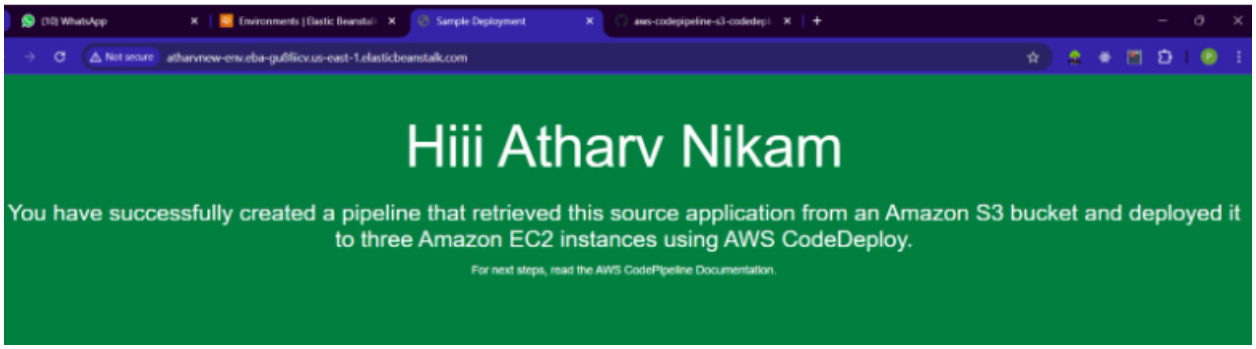
19)you have successfully created the pipeline



20)you will get this message



21)make changes to your repo and rerun the pipeline



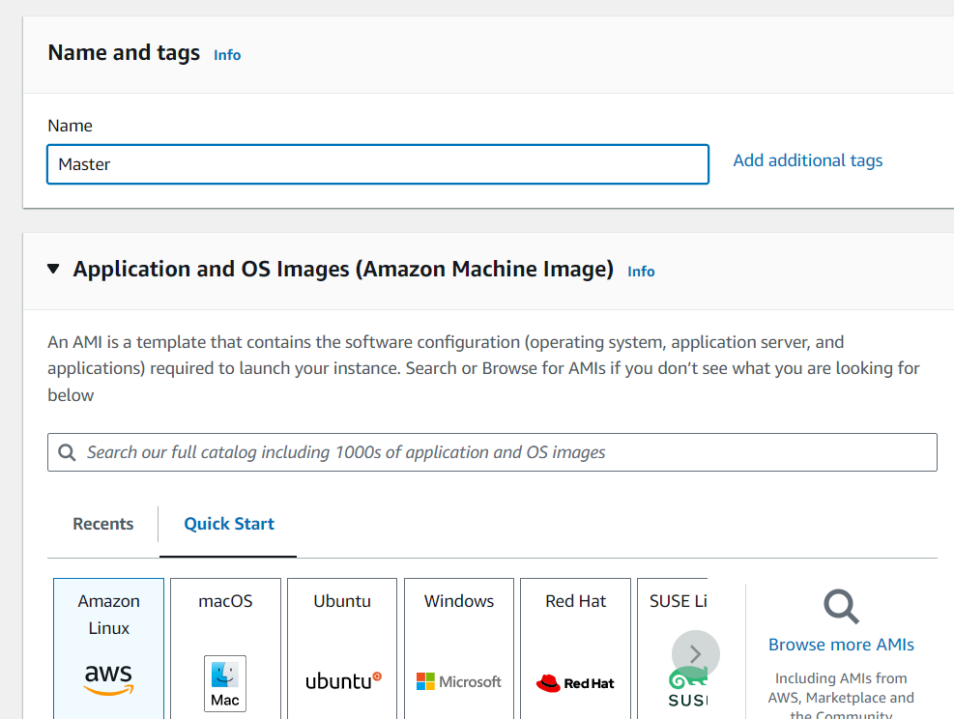
Name:Atharv Sanjay Nikam **Div:D15C** **Roll:36**

Experiment 3

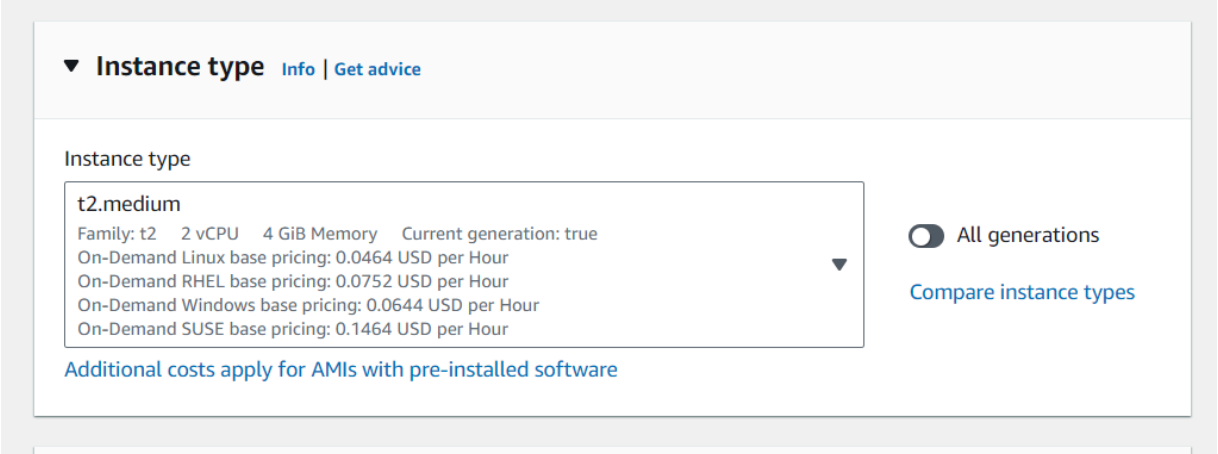
Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

1)Launch an EC2 Instance:

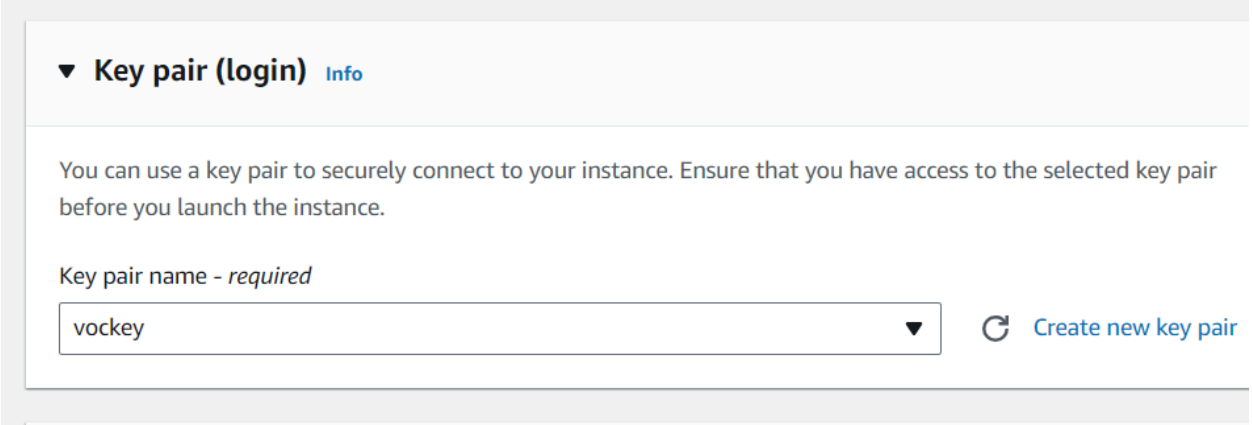
Choose Amazon Linux as the operating system for your instance.



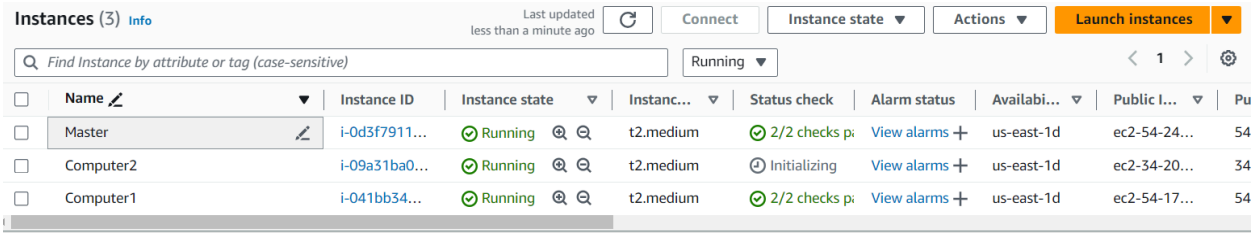
Note: The AWS free tier provides a t2.micro instance (1 CPU, 1 GiB RAM), but Kubernetes requires a minimum of 2 CPUs and 2 GiB RAM. So, make sure to select **t2.medium**

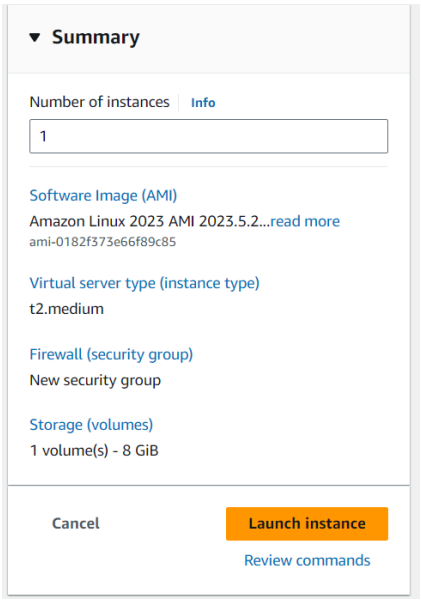


2)select a Key Pair:
You can either use the default key pair provided by AWS or create a new one for better security. Click on **Create**.



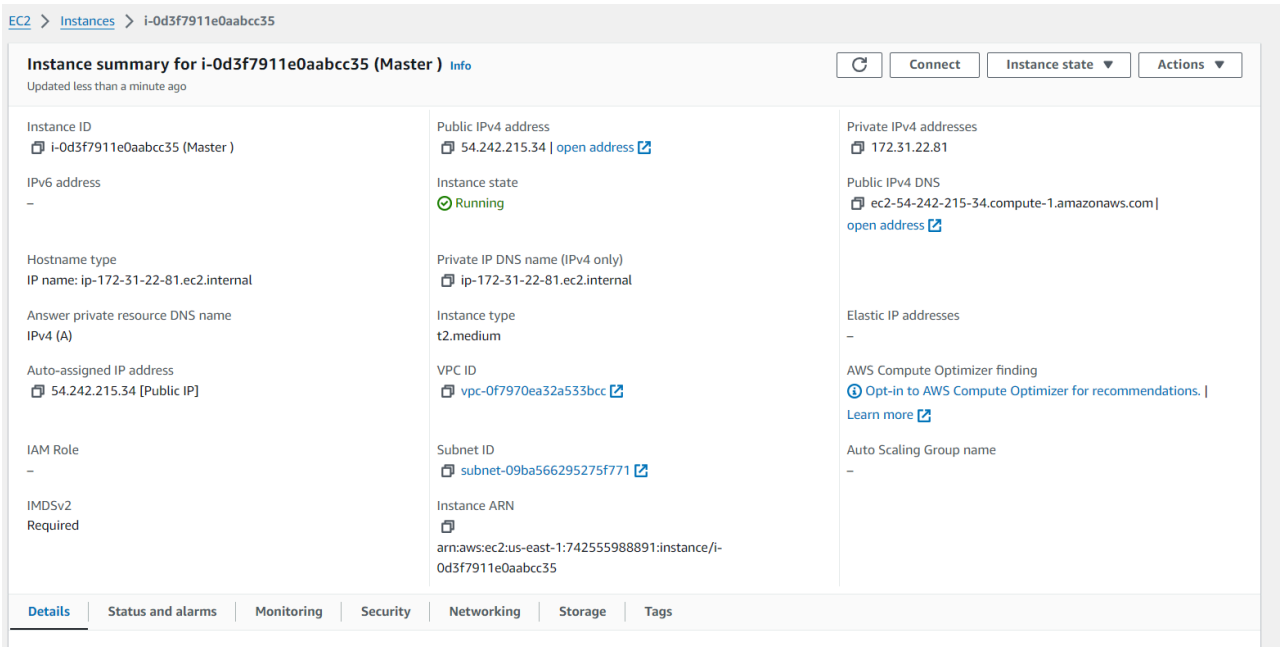
All your instances will appear here

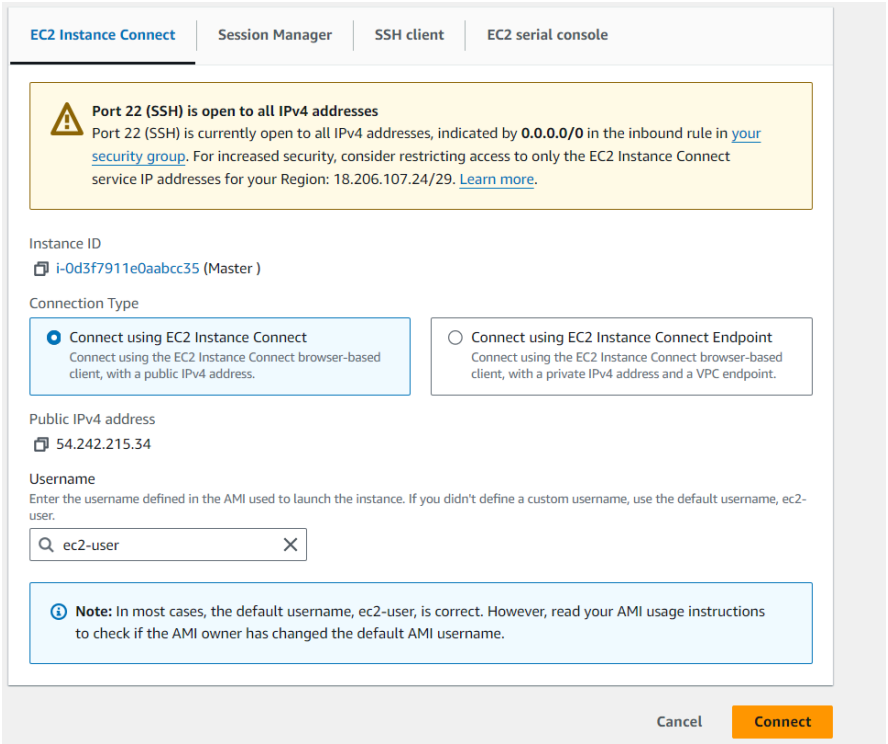




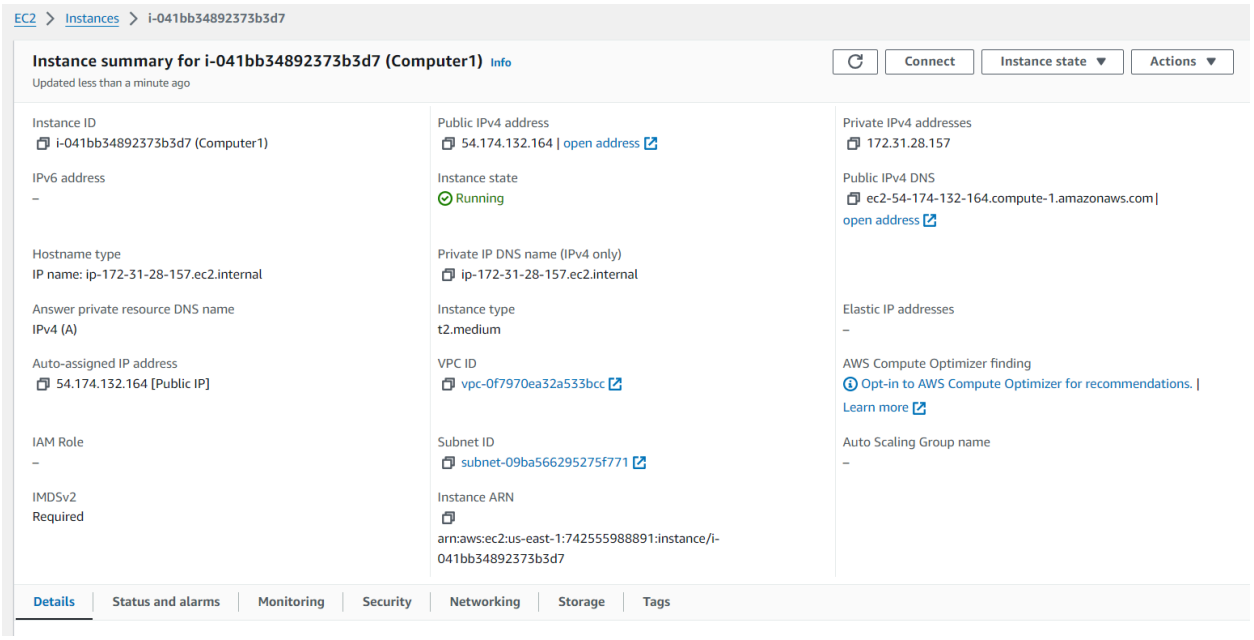
3)Connect to the Instance:

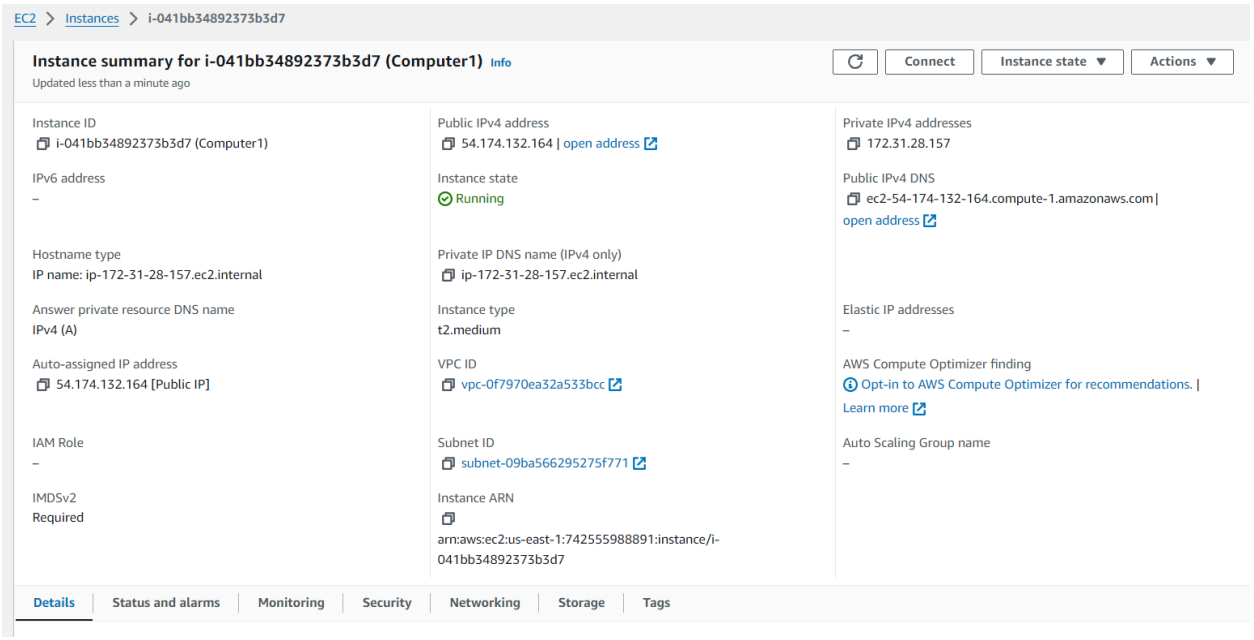
Navigate to the instances page, locate your instance, and click on its ID. Then, select **Connect** and keep the default connection settings. Finally, click **Connect** to start your session.





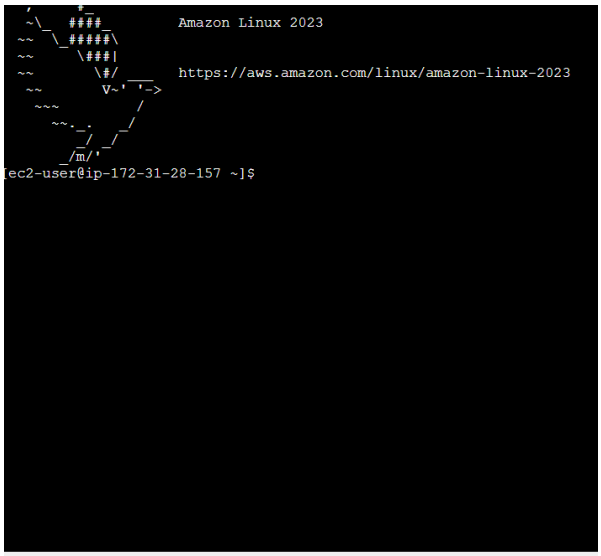
Name:Atharv Sanjay Nikam **Div:D15C** **Roll:36**



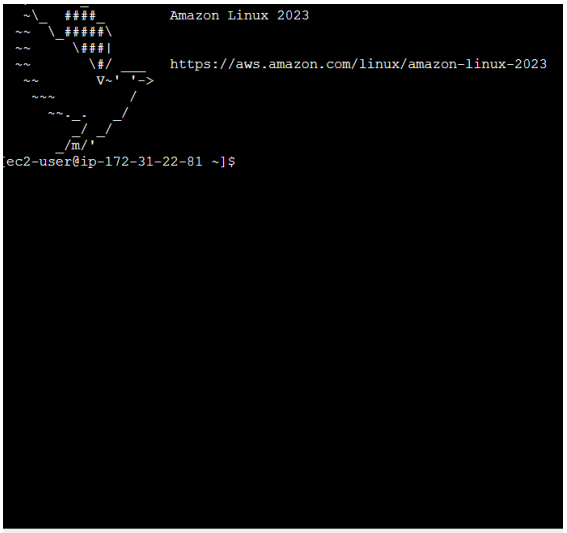


Name:Atharv Sanjay Nikam **Dlv:D15C** **Roll:36**

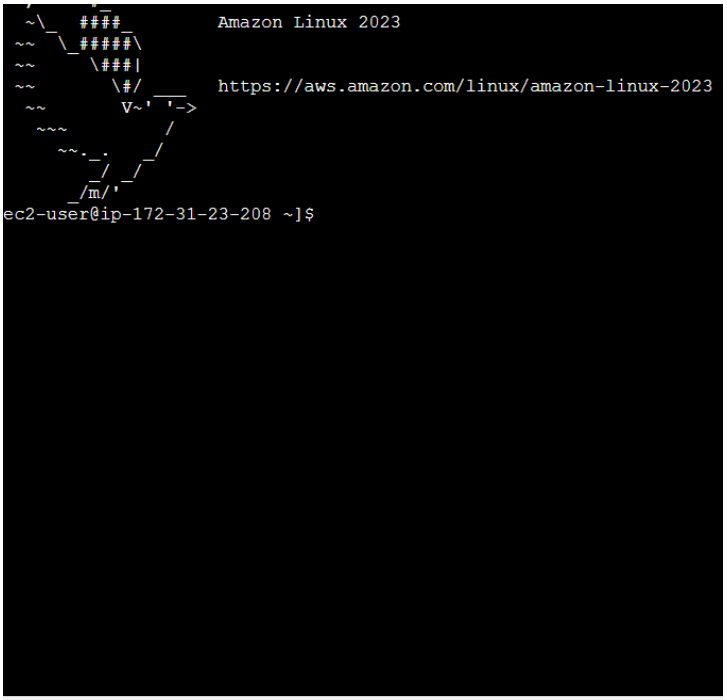
All the three server



i-041bb34892373b3d7 (Computer1)
PublicIPs: 54.174.132.164 PrivateIPs: 172.31.28.157



i-0d3f7911e0aabcc35 (Master)
PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81



i-09a31ba0572c10bdb (Computer2)
PublicIPs: 34.204.51.153 PrivateIPs: 172.31.23.208

Name:Atharv Sanjay Nikam **Div:D15C** **Roll:36**

Step 2: Installation of Docker

1)Switch to Root User:

Run `sudo su` to get root-level access in the terminal.

```
[ec2-user@ip-172-31-22-81 ~]$ sudo su
[root@ip-172-31-22-81 ec2-user]#
```

2)Install Docker:

Use the YUM package manager to install Docker by running:

```
yum install docker -y
```

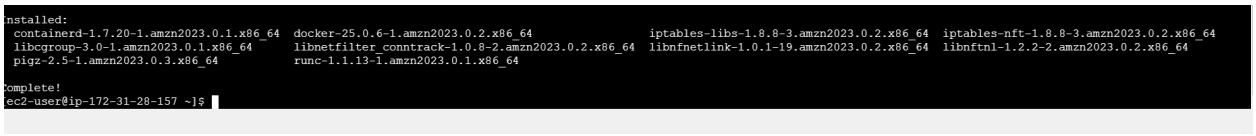
```
[root@ip-172-31-22-81 ec2-user]# yum install docker -y
Last metadata expiration check: 0:11:19 ago on Sun Sep 15 06:50:06 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    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_64.rpm      6.3 MB/s | 401 kB     00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm       6.0 MB/s | 183 kB     00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm            2.8 MB/s |  75 kB     00:00
```

```
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_64.rpm      6.3 MB/s | 401 kB     00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm       6.0 MB/s | 183 kB     00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm            2.8 MB/s |  75 kB     00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.6 MB/s |  58 kB     00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm      1.4 MB/s |  30 kB     00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm           1.9 MB/s |  84 kB     00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                 1.1 MB/s |  83 kB     00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm               22 MB/s | 3.2 MB     00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm         41 MB/s |  35 MB     00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm            38 MB/s |  44 MB     00:01
--------------------------------------------------------------------------------
Total                                                       70 MB/s |  84 MB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : runc-1.1.13-1.amzn2023.0.1.x86_64                      1/10
  Installing       : containerd-1.7.20-1.amzn2023.0.1.x86_64                2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64                2/10
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                         3/10
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                   4/10
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64              5/10
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64     6/10
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64              7/10
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64               8/10
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64               8/10
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                    9/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                    10/10
  Installing       : docker-25.0.6-1.amzn2023.0.2.x86_64                    10/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                    10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
```

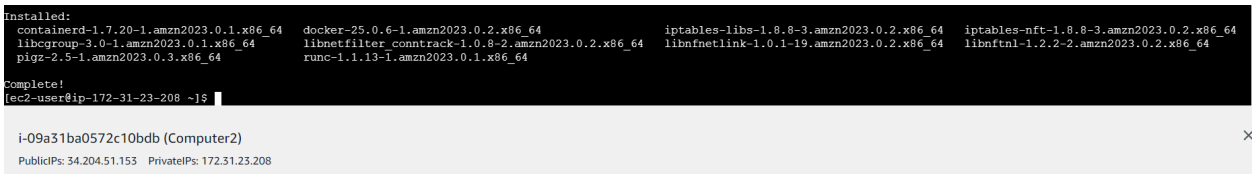
Name:Atharv Sanjay Nikam **Div:D15C** **Roll:36**

3)Install in all devices

Computer 1



Computer2



4)Configure Docker Daemon:

You need to configure Docker to use the systemd cgroup driver.

Run the 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
```

```
[root@ip-172-31-22-81 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"
}
```

5)Enable and Start Docker:

Start Docker by running:

bash

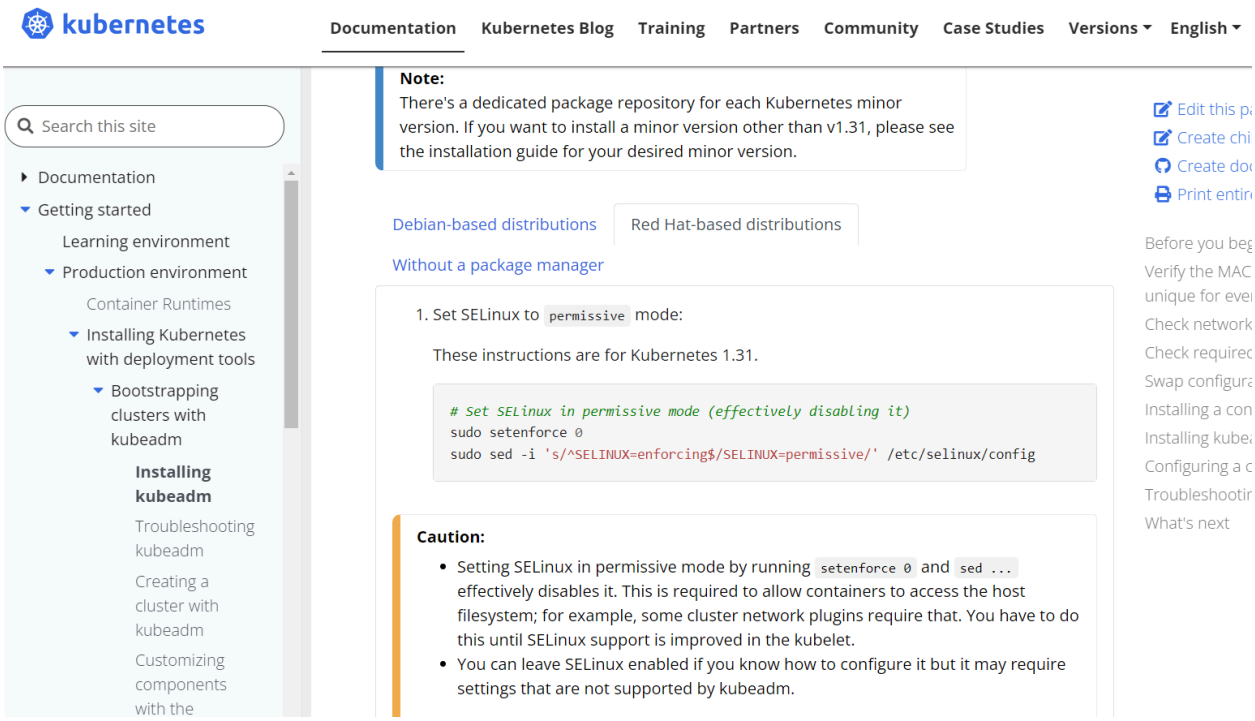
Copy code

```
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
[root@ip-172-31-22-81 docker]# sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-22-81 docker]#
```

Step 3: Installing Kubernetes

For installing kubernetes, we will be using kubeadm, a framework used for creating kubernetes clusters using command line.
https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm
The following will be visible when you visit the website.



1)Prepare the System:

Configure SELinux to run in permissive mode to avoid permission issues during the Kubernetes setup:

bash
Copy code

```
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
/etc/selinux/config
```

```
[root@ip-172-31-22-81 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-22-81 ec2-user]#
```

Computer 1

```
[ec2-user@ip-172-31-28-157 ~]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-28-157 ~]$
```

Computer 2

```
[ec2-user@ip-172-31-23-208 ~]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-23-208 ~]$
```

2)Add the Kubernetes Repo:

Create a Kubernetes repository by running:

bash

Copy code

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

```
[root@ip-172-31-22-81 ec2-user]# 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
[root@ip-172-31-22-81 ec2-user]#
```

i-0d3f7911e0aabcc35 (Master)

PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81

```
[root@ip-172-31-22-81 ec2-user]# sudo yum update
Kubernetes                                              [===                                                ] ---  B/s |   0  B     --:-- ETA
Kubernetes                                                                                                    90 kB/s |  17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-22-81 ec2-user]#
```

yum repolist

This command shows the repositories created on the machine.

```
[root@ip-172-31-22-81 ec2-user]# yum repolist
repo id                                                                repo name
amazonlinux                                                            Amazon Linux 2023 repository
kernel-livepatch                                                       Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                             Kubernetes
[root@ip-172-31-22-81 ec2-user]#
```

3)Install Kubernetes Components:

Now, install the required Kubernetes tools by executing:

bash

Copy code

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

```
[root@ip-172-31-22-81 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:06 ago on Sun Sep 15 07:47:07 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 download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm  432 kB/s |  24 kB  00:00
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm   374 kB/s |  24 kB  00:00
```

```
[ec2-user@ip-172-31-28-157 ~]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:08 ago on Sun Sep 15 07:47:52 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 download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm  426 kB/s |  24 kB  00:00
```

```
[ec2-user@ip-172-31-23-208 ~]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:07 ago on Sun Sep 15 07:47:58 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 download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm  455 kB/s |  24 kB  00:00
```

Configure System Settings:

Run the following to configure a network bridge:

bash

Copy code

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

```
[root@ip-172-31-22-81 ec2-user]# 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
[root@ip-172-31-22-81 ec2-user]#
```

i-0d3f7911e0aabcc35 (Master)

PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81

PERFORM THE FOLLOWING ON ONLY THE MASTER MACHINE

Initialize Kubernetes:
Initialize your Kubernetes cluster with the following command:

bash
Copy code

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=NumCPU,Mem
```

```
[root@ip-172-31-22-81 ec2-user]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=all
I0915 07:56:11.461747   30220 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
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0915 07:56:11.697643   30220 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the contai
mended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal kubernetes kubernetes.default kub
 [10.96.0.1 172.31.22.81]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal localhost] and IPs [172.31.22.8
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal localhost] and IPs [172.31.22.81
[certs] Generating "etcd/healthcheck-client" certificate and key
```

Configure `kubectl` Access:

To set up `kubectl` for your non-root user, run:

bash
Copy code

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

```
[root@ip-172-31-22-81 ec2-user]# mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-22-81 ec2-user]#
```

To check whether nodes are connected, run the command

- kubectl get nodes

This output shows only master is connected right now.

Kubtectl get nodes

```
[root@ip-172-31-22-81 ec2-user]# kubectl get nodes
NAME                           STATUS     ROLES           AGE     VERSION
ip-172-31-22-81.ec2.internal   NotReady   control-plane   4m52s   v1.30.5
```

Perform this Only on node Machines

```
[root@ip-172-31-22-81 ec2-user]# sudo yum install socat -y
Last metadata expiration check: 0:19:33 ago on Sun Sep 15 07:47:07 2024.
Dependencies resolved.
================================================================================
 Package          Architecture        Version                  Repository    Si
================================================================================
Installing:
 socat            x86_64              1.7.4.2-1.amzn2023.0.2   amazonlinux   303

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

Total download size: 303 k
Installed size: 1.1 M
Downloading Packages:
socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm          2.4 MB/s | 303 kB     00:00
--------------------------------------------------------------------------------
```

This is the token which we got
172.31.22.81:6443 --token 9wf1pg.1xgp88wof0wpslc9 \
--discovery-token-ca-cert-hash
sha256:b4efc86172e4999d3d1e530147cc26705f6543cd48689416874811c98b2a325f

Put this command on the node machine to connect

Now go Back to the master machine and write ‘kubectl get nodes’

```
[root@ip-172-31-22-81 docker]# kubectl get nodes
NAME                          STATUS     ROLES           AGE     VERSION
ip-172-31-22-81.ec2.internal  Ready      control-plane   4m13s   v1.30.5
ip-172-31-28-157.ec2.internal Not Ready  computer 1      2m24s   v1.30.5
ip-172-31-23-208.ec2.internal Not Ready  computer 2      2m17s   v1.30.5
```

We will see nodes are <NOT READY> to change it install a CNI plugin on Master Machine
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

Now ‘Run kubectl get nodes’

```
[root@ip-172-31-22-81 docker]# kubectl get nodes
NAME                          STATUS   ROLES           AGE     VERSION
ip-172-31-22-81.ec2.internal  Ready    control-plane   9m33s   v1.30.5
ip-172-31-28-157.ec2.internal Ready    computer 1      7m24s   v1.30.5
ip-172-31-23-208.ec2.internal Ready    computer 2      7m17s   v1.30.5
```

Conclusion

In this experiment, we explored the architecture of a Kubernetes cluster and successfully deployed it on AWS EC2 instances, consisting of a master node and two worker nodes. After setting up Docker, Kubernetes components (kubelet, kubeadm, kubectl), and containerd across all nodes, the cluster was initialized by configuring the master node and joining the worker nodes. Initially, the nodes showed a "NotReady" status, which was corrected by implementing the Calico networking solution. Additionally, nodes were labeled appropriately based on their roles. By the end of the process, all nodes were fully operational, confirming that the Kubernetes cluster was configured correctly .

Name:Atharv Nikam Div:D15C Roll:36

Experiment 4

Aim: To set up Kubectl for managing a Kubernetes cluster and deploy a basic application.

Theory: Kubernetes, which originated from Google, is a popular open-source platform for managing containerized applications. It streamlines the scaling, deployment, and maintenance of containers, ensuring resilience and flexibility. It's become a standard in the industry for orchestrating containers, with contributions from top technology companies through the Cloud Native Computing Foundation (CNCF).

Kubernetes Deployment:
This refers to a resource in Kubernetes that allows for rolling updates and rollbacks of applications. It ensures that the correct number of pods are running, maintaining desired configurations at all times.

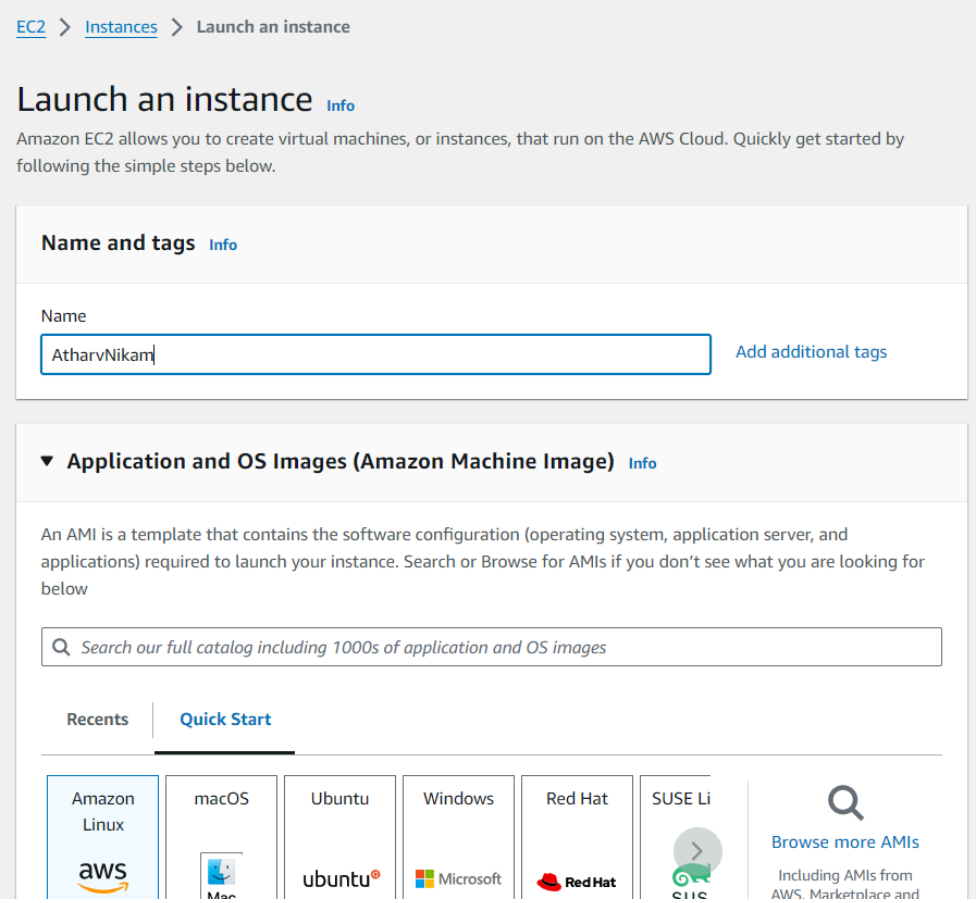
Requirements:

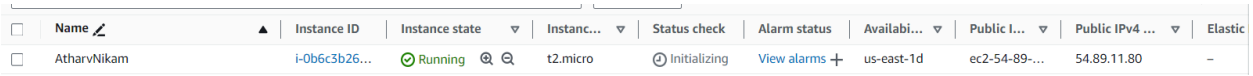
- **EC2 Instance:** A t2.medium instance with at least 2 CPUs is necessary to accommodate Kubernetes' resource needs.
    - **Minimum configuration:**
        - Instance Type: t2.medium
        - CPUs: 2
        - Memory: Suitable for container operations

Name:Atharv Nikam Div:D15C Roll:36

Step 1:

Log in to your AWS account and launch an EC2 instance. Choose Ubuntu as the AMI and set the instance type to t2.medium. Generate an RSA key in `.pem` format, and move it to a secure folder on your system.
Note: Kubernetes requires at least 2 CPUs, so ensure you select the t2.medium instance. Be mindful to terminate the instance after the task is complete, as it isn't covered under the free tier.

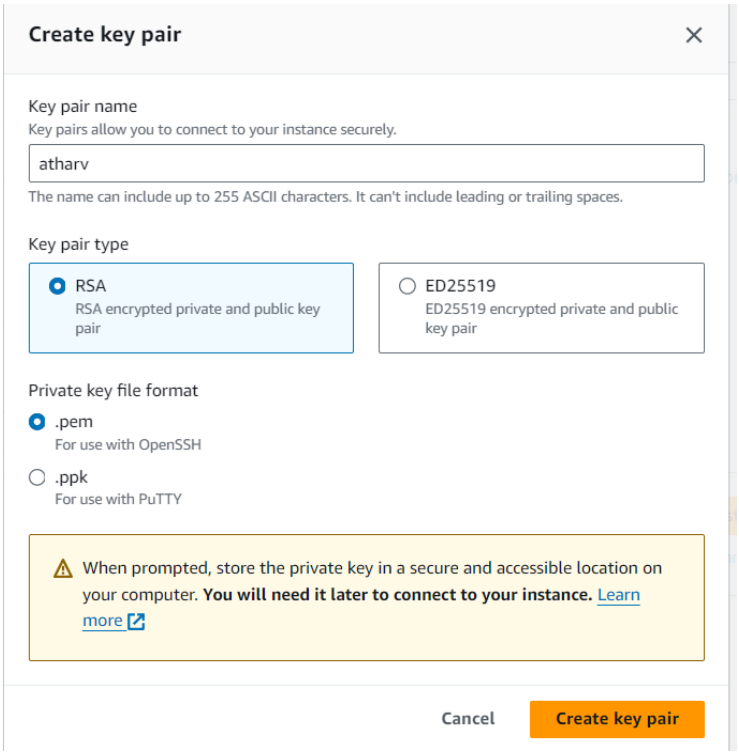




Name:Atharv Nikam Div:D15C Roll:36

Step 2:

Connect to your EC2 instance via SSH. Open the terminal in the directory where the `.pem` key is stored, and run the SSH command to access the instance.




```
PS D:\Key> ssh -i "atharv.pem" ubuntu@ec2-54-83-70-136.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Sep 16 05:30:31 UTC 2024

  System load:  0.24              Processes:             159
  Usage of /:   55.6% of 6.71GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.30.208
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

135 updates can be applied immediately.
41 of these updates are standard security updates.
```

Name:Atharv Nikam Div:D15C Roll:36

Step 3:
Install Docker on your EC2 instance using the following commands:
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-30-208:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/nul
ubuntu@ip-172-31-30-208:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports 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 InRelease [126 kB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [123 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8264 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [365 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [150 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.3 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [317 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [61.5 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

Name:Atharv Nikam Div:D15C Roll:36

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.1-1~ubuntu.24.04~noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.1-1~ubuntu.24.04~noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.1-1~ubuntu.24.04~noble [9571 kB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 MB]
Fetched 122 MB in 2s (66.5 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
```

```
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam Div:D15C Roll:36

Configure Docker to use the systemd cgroup driver:
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"]
} EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}

```
ubuntu@ip-172-31-30-208:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"] } EOF
{
"exec-opts": ["native.cgroupdriver=systemd"] }ubuntu@ip-172-31-30-208:~$
```

Enable and restart Docker:

sudo systemctl enable docker

sudo systemctl daemon-reload

sudo systemctl restart docker

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-30-208:~$
```

Install Kubernetes using the following commands:

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-30-208:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-30-208:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-30-208:~$
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 132 kB in 1s (218 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 133 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (81.3 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68007 files and directories currently installed.)
ubuntu@ip-172-31-30-208:~$
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-30-208:~$
```

Step 5:

To initialize the Kubernetes cluster, run:

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

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-30-208:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0916 04:01:46.391802    6909 checks.go:1080] [preflight] WARNING: Couldn't create the interfac
validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc erro
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime
e runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-30-208:~$
```

If you encounter any errors, such as missing container runtimes, install containerd:

```
sudo apt-get install -y containerd
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 133 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (82.5 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4.1) ...
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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-172-31-30-208:~$
```

Name:Atharv Nikam Div:D15C Roll:36

```
ubuntu@ip-172-31-30-208:~$ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

[metrics]
  address = ""
  grpc_histogram = false

[plugins]

  [plugins."io.containerd.gc.v1.scheduler"]
    deletion_threshold = 0
    mutation_threshold = 100
    pause_threshold = 0.02
    schedule_delay = "0s"
    startup_delay = "100ms"

  [plugins."io.containerd.grpc.v1.cri"]
    cdi_spec_dirs = ["/etc/cdi", "/var/run/cdi"]
```

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl restart containerd
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable containerd
ubuntu@ip-172-31-30-208:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-09-16 04:11:07 UTC; 22s ago
       Docs: https://containerd.io
   Main PID: 7659 (containerd)
      Tasks: 7
     Memory: 14.0M (peak: 14.7M)
        CPU: 107ms
     CGroup: /system.slice/containerd.service
             └─7659 /usr/bin/containerd

Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807160348Z" level=info msg="Start subscribing containerd event"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807213129Z" level=info msg="Start recovering state"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807163270Z" level=info msg=serving... address=/run/containerd/containerd.sock.ttrpc
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807351018Z" level=info msg=serving... address=/run/containerd/containerd.sock
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807469775Z" level=info msg="Start event monitor"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807526625Z" level=info msg="Start snapshots syncer"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807580042Z" level=info msg="Start cni network conf syncer for default"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807593159Z" level=info msg="Start streaming server"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807763589Z" level=info msg="containerd successfully booted in 0.028456s"
Sep 16 04:11:07 ip-172-31-30-208 systemd[1]: Started containerd.service - containerd container runtime.
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam Div:D15C Roll:36

Set up your Kubernetes configuration:
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-172-31-25-17 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.x
ml.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.x
ml.key exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[ec2-user@ip-172-31-25-17 docker]$
```

Install a networking plugin:

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

```
ubuntu@ip-172-31-30-208:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-30-208:~$
```

Step 7:

To deploy an Nginx server, use the following commands:

```
ubuntu@ip-172-31-30-208:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam Div:D15C Roll:36

```
ubuntu@ip-172-31-30-208:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fzkg4    0/1     Pending   0          30s
nginx-deployment-d556bf558-qvghq    0/1     Pending   0          30s
ubuntu@ip-172-31-30-208:~$
```

Forward the port to access the server:

POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80

```
ubuntu@ip-172-31-30-208:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
ubuntu@ip-172-31-30-208:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-30-208:~$
```

If the pod is in a pending state, remove the control-plane taint:

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

```
ubuntu@ip-172-31-30-208:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
```

```
ubuntu@ip-172-31-30-208:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-30-208   Ready    control-plane   39m   v1.31.1
ubuntu@ip-172-31-30-208:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fzkg4    1/1     Running   0          23m
nginx-deployment-d556bf558-qvghq    1/1     Running   0          23m
ubuntu@ip-172-31-30-208:~$
```

Step 8:

Finally, verify the Nginx server is running:

```
ubuntu@ip-172-31-30-208:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Mon, 16 Sep 2024 05:04:04 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

If the response is `200 OK`, your Nginx deployment is successful.

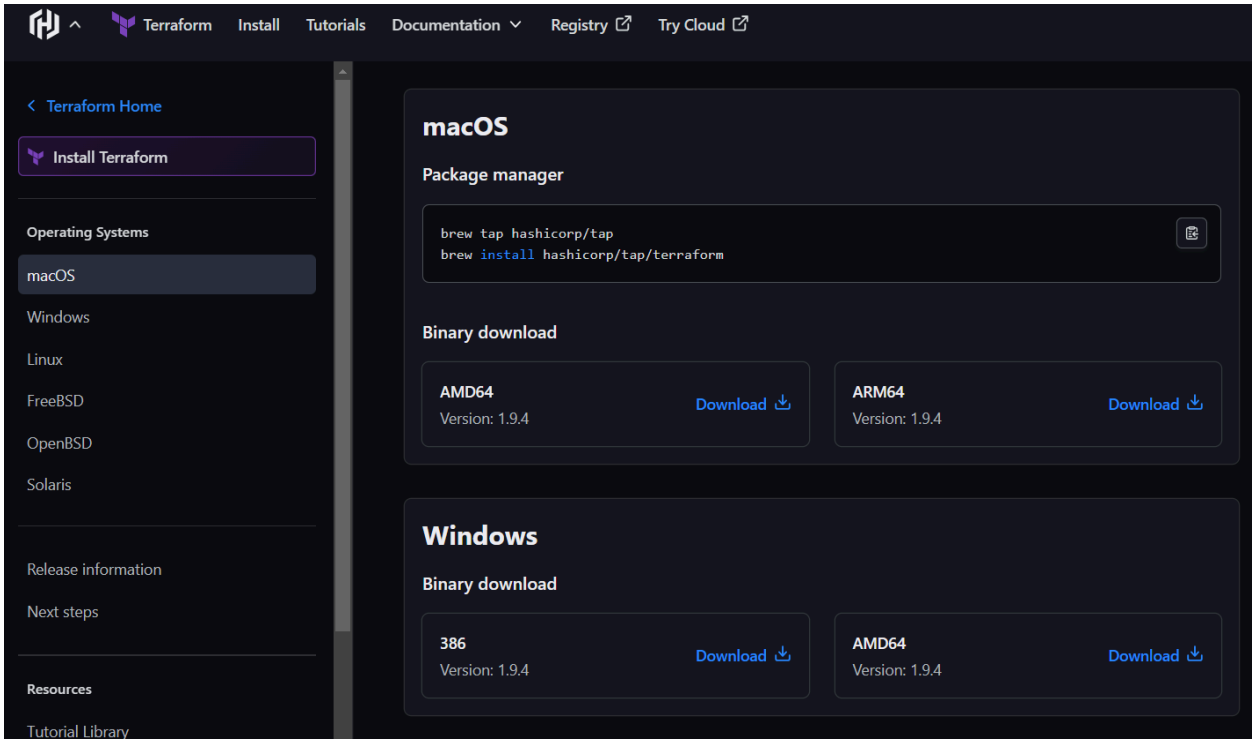
Conclusion:

Through this experiment, Kubernetes was successfully installed on an EC2 instance, and an Nginx server was deployed. The process included troubleshooting common issues like pod states and container runtimes, ensuring a stable deployment environment

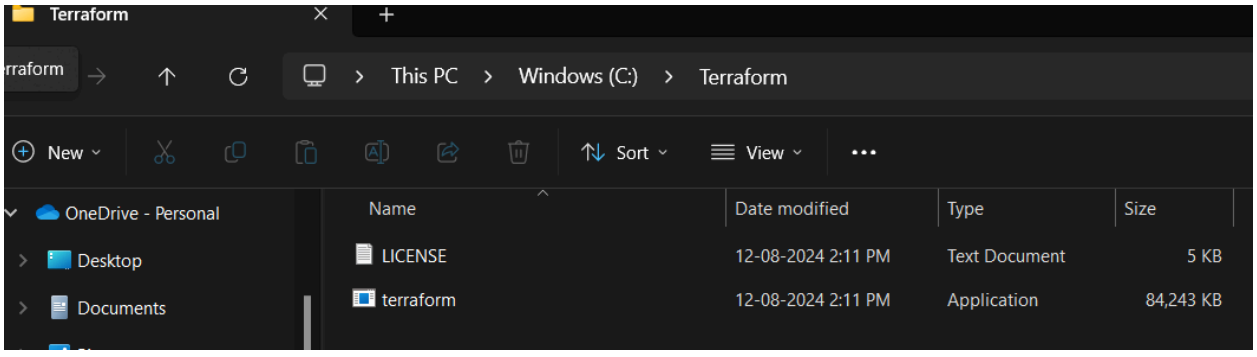
Name:Atharv Nikam Div:D15C Roll:36

Experiment 5

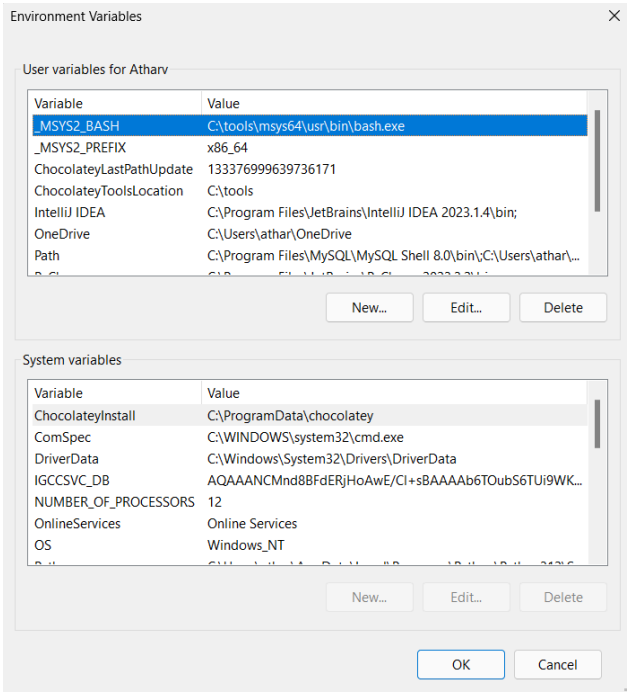
1)Downloading Terraform from official website
https://developer.hashicorp.com/terraform/install



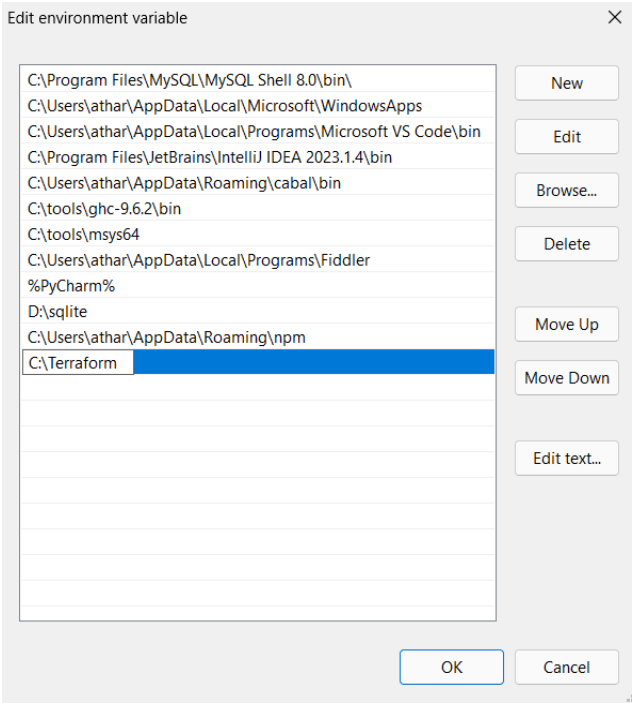
2) Save it into Windows C



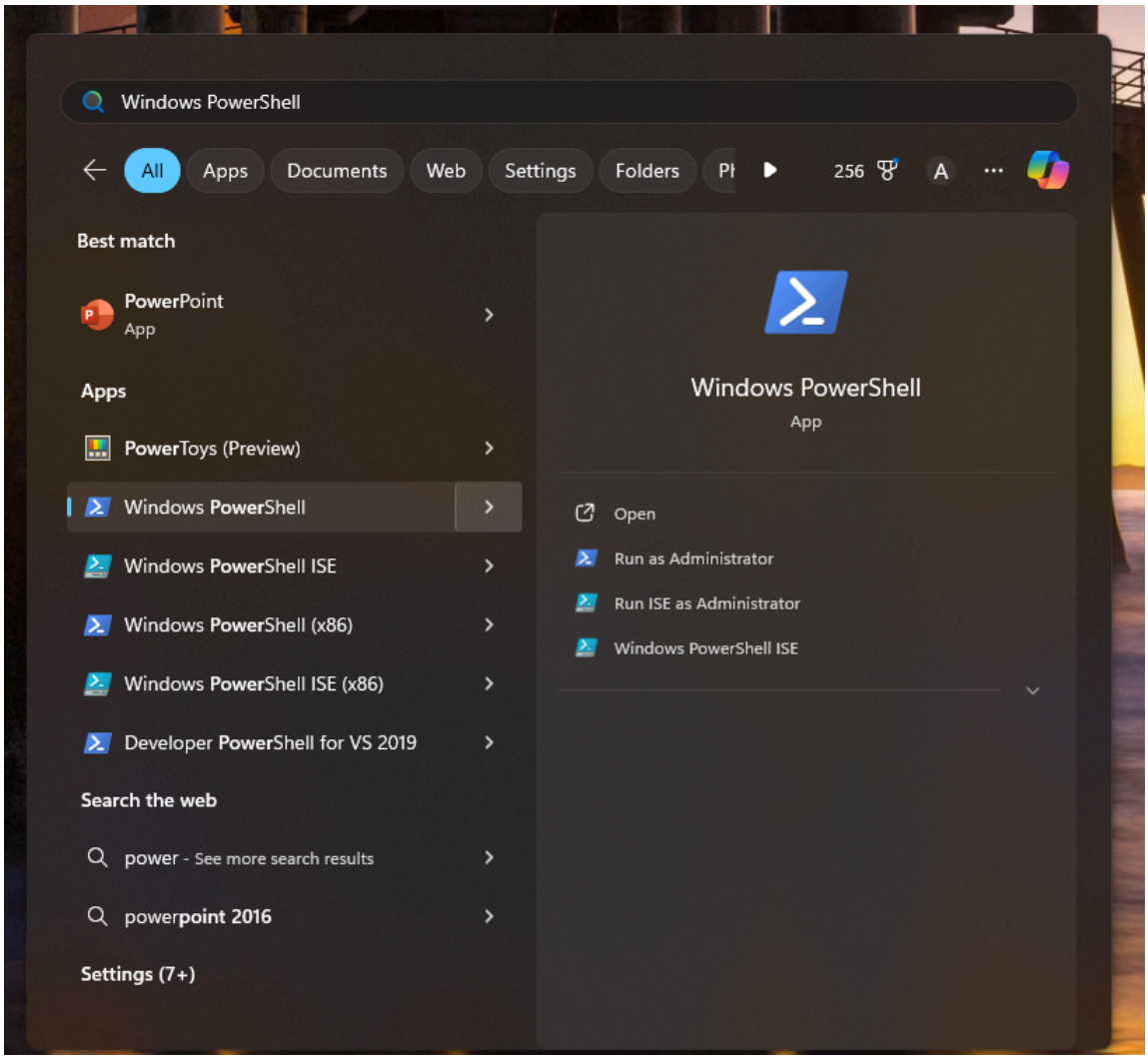
3)Change Environment Variables



4)Click on path and add new C:\Terraform



5)search for powershell and open it



5)To Check if terraform is installed open powershell and type Terraform —version

```
PS C:\Users\athar> Terraform --version
Terraform v1.9.4
on windows_386
PS C:\Users\athar>
```

6)To see all terraform commands type Terraform –help

```
PS C:\Users\athar> Terraform --help
Usage: terraform [global options] <subcommand> [args]

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

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

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

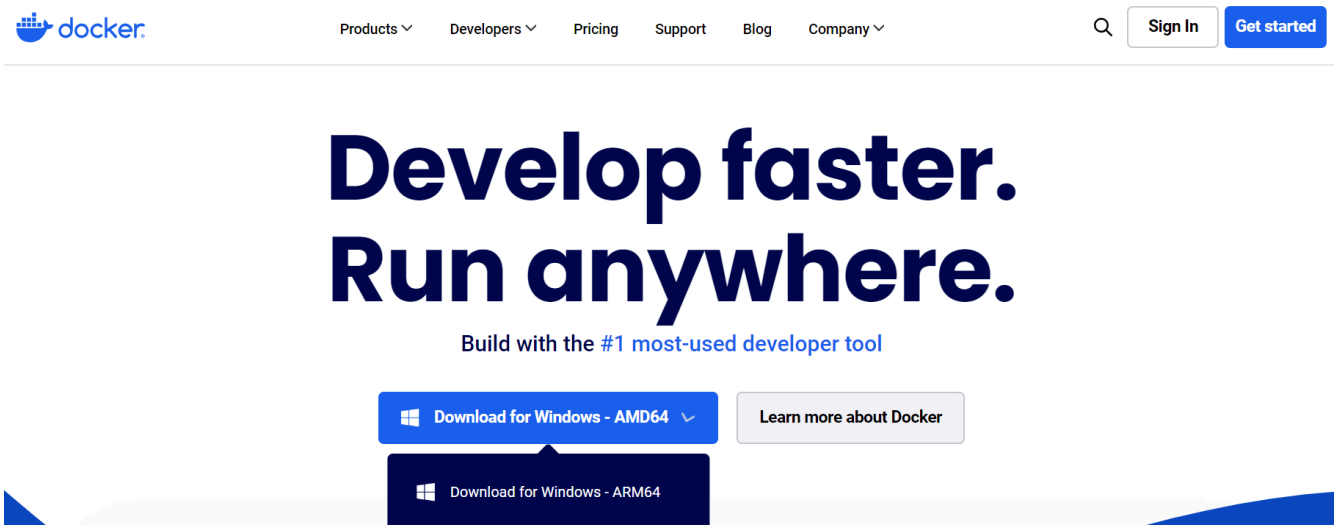
Name:Atharv Nikam **Div:D15C** **Roll No :36**

Exp 6

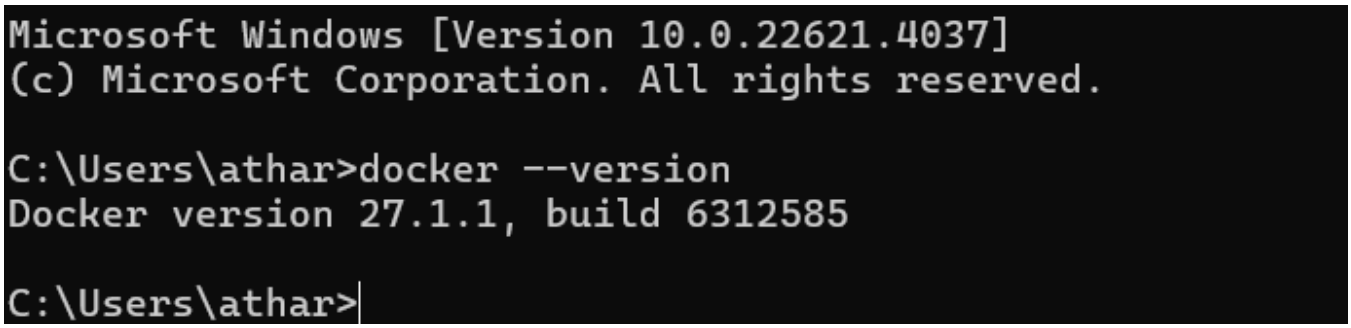
Aim : To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.(S3 bucket or Docker)

Step1: check if Docker is installed in your system to check open your powershell and type docker

If not then install docker from https://www.docker.com/ and download it



Check if docker has installed by opening command prompt and typing docker —version



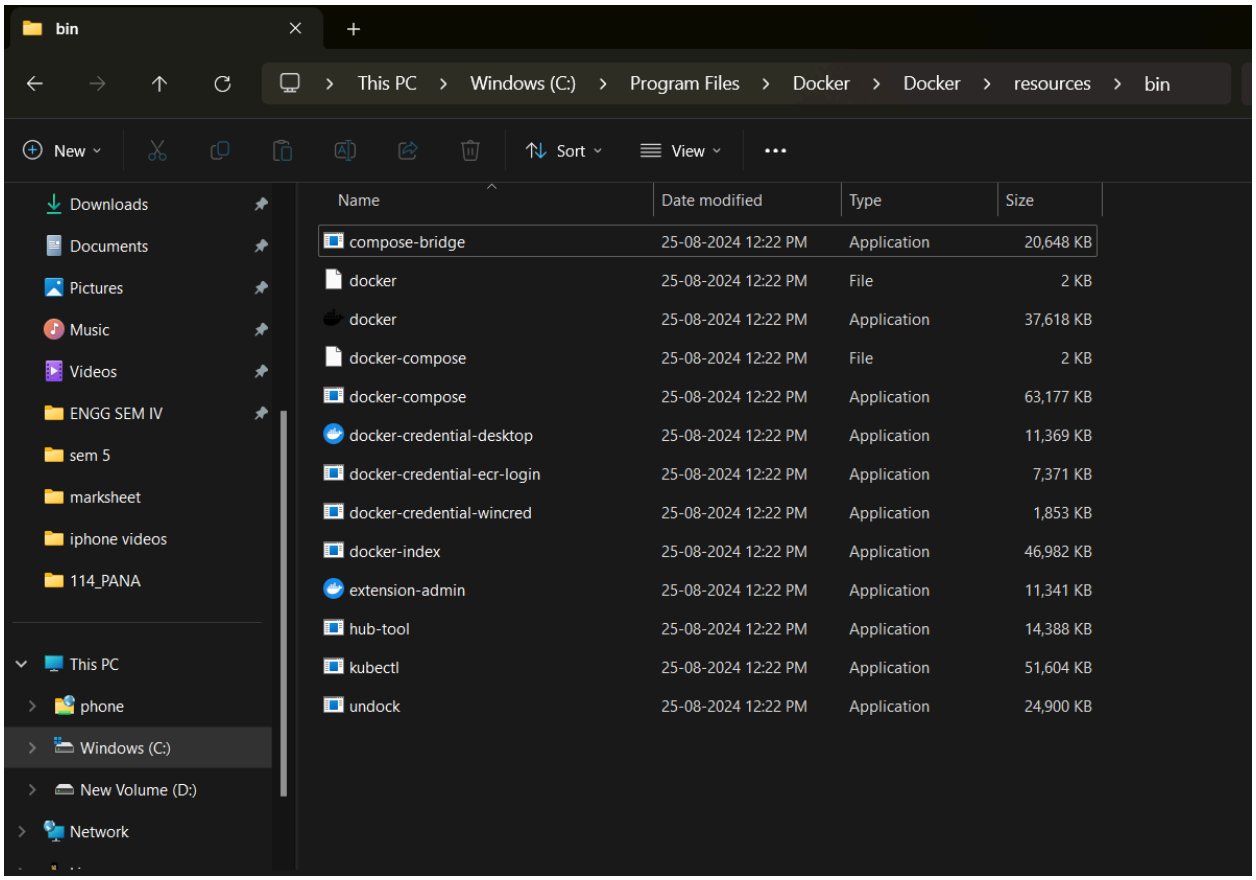
Name:Atharv Nikam **Div:D15C** **Roll No :36**

Type docker to see all the commands

```
C:\Users\athar>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
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
  plugin      Manage plugins
  sbom*       View the packaged-based Software Bill Of Materials (SBOM) for an image
  scout*      Docker Scout
  system      Manage Docker
  trust       Manage trust on Docker images
  volume      Manage volumes

Swarm Commands:
  config      Manage Swarm configs
  node        Manage Swarm nodes
  secret      Manage Swarm secrets
  service     Manage Swarm services
  stack       Manage Swarm stacks
  swarm       Manage Swarm

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  logs        Fetch the logs of a container
```

Name:Atharv Nikam **Div:D15C** **Roll No :36**

If you are not getting the version try these steps
Go to your file explorer and go to bin folder of your docker file or just paste this
C:\Program Files\Docker\Docker\resources\bin



Access the 'Edit the System Environment Variables' option on your computer. Then, select Environment Variables.

Look for a 'Path' variable under System variables. If it’s there, select it and click on Edit.

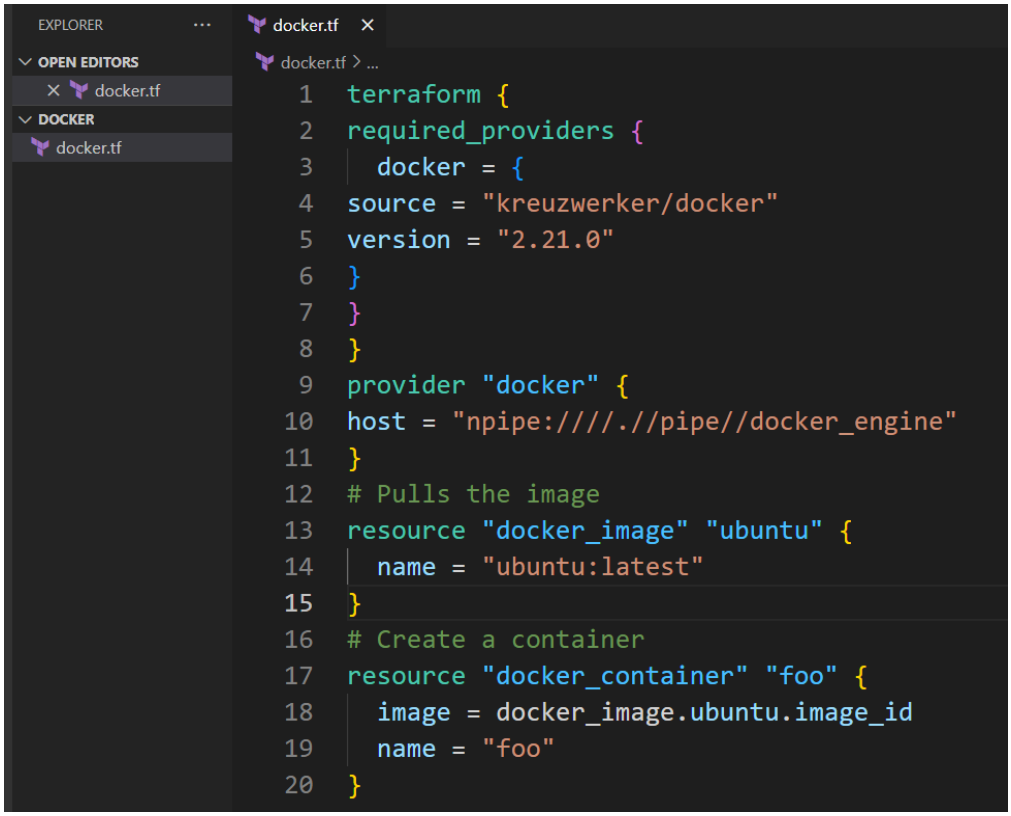
If it’s not there, click on New and create a 'Path' variable.
If the variable already exists, click on Edit and then on New to open a text box. Paste the path you copied into the box and click OK to close all the windows.

Now do step 1 again to check the docker version

Step 2: Create a folder Terraform scripts and then inside that create docker file and at last create a docker.ts file
After that open that folder in visual studio and paste this code

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

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

# Pulls the image
resource "docker_image" "ubuntu"
  {name = "ubuntu:latest"
}

# Create a container
resource "docker_container" "foo"
  { image =
 docker_image.ubuntu.image_idname =
 "foo"
}
```

Name:Atharv Nikam **Div:D15C** **Roll No :36**



Step 3:Open the terminal and go to folder where docker.tf is present

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

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS D:\all code\Terraform Scripts\Docker>
```

Name:Atharv Nikam **Div:D15C** **Roll No :36**

Step 4:Run the Command ‘terraform plan’ This would create a execution plan and you could see an overview of your plan

```
PS D:\all code\Terraform Scripts\Docker> terraform plan

Planning failed. Terraform encountered an error while generating this plan.


  Error: Error pinging Docker server: error during connect: Get "http://%2F%2F.%2F%2Fpipe%2F%2Fdocker_engine/_ping": open //.//pipe//docker_engine: The system cannot find the fil
e specified.

    with provider["registry.terraform.io/kreuzwerker/docker"],
    on docker.tf line 9, in provider "docker":
     9: provider "docker" {
```

Sometimes the docker engine os not working so on the docker Docker Desktops and try again

Step 5 :Run terraform plan again

```
PS D:\all code\Terraform Scripts\Docker> terraform plan

Terraform used the selected providers to generate the following execution p
  + 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)
```

Name:Atharv Nikam **Div:D15C** **Roll No :36**

Step 6:run command terraform apply

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Creation complete after 12s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...

Error: container exited immediately

  with docker_container.foo,
  on docker.tf line 17, in resource "docker_container" "foo":
  17: resource "docker_container" "foo" {
```

The script will give an error because this script took very less time to execute to resolve this issue we have to add this command

‘command=[“sleep”,”infinity”]’
Now we will get this

```
PS D:\all code\Terraform Scripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6

Terraform used the selected providers to generate the following execution plan. Resource actions are indi
  + 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",
          + "infinity",
        ]
      + 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
```

Name:Atharv Nikam **Div:D15C** **Roll No :36**

```
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 0s [id=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\all code\Terraform Scripts\Docker>
```

Docker images before terraform apply

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
PS D:\all code\Terraform Scripts\Docker>
```

Docker images after terraform apply

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
ubuntu       latest    edbfe74c41f8   3 weeks ago    78.1MB
PS D:\all code\Terraform Scripts\Docker>
```

Step 7: Now the image is created to destroy that we have to use the command Terraform destroy

```
PS D:\all code\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]

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",
          - "infinity",
        ] -> 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         = "305bdd507338" -> null
      - id               = "305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad" -> 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
```

Enter yes

```
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=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
PS D:\all code\Terraform Scripts\Docker>
```

To check if the images are destroyed check run docker images

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
PS D:\all code\Terraform Scripts\Docker>
```

Name:Atharv Nikam **Div:D15C** **Roll no: 36**

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

Prerequisites:

1. **Docker:**
   Confirm Docker is installed and functioning by running the command:

```
PS C:\Users\athar> docker -v
Docker version 27.1.1, build 6312585
PS C:\Users\athar>
```

2 . SonarQube Image Installation:

Use the following command to download the SonarQube image via Docker:

```
PS C:\Users\athar>  docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

3.Jenkins Installation:

Ensure Jenkins is already installed and properly configured on your system

Name:Atharv Nikam Div:D15C Roll:36

Experiment Steps:

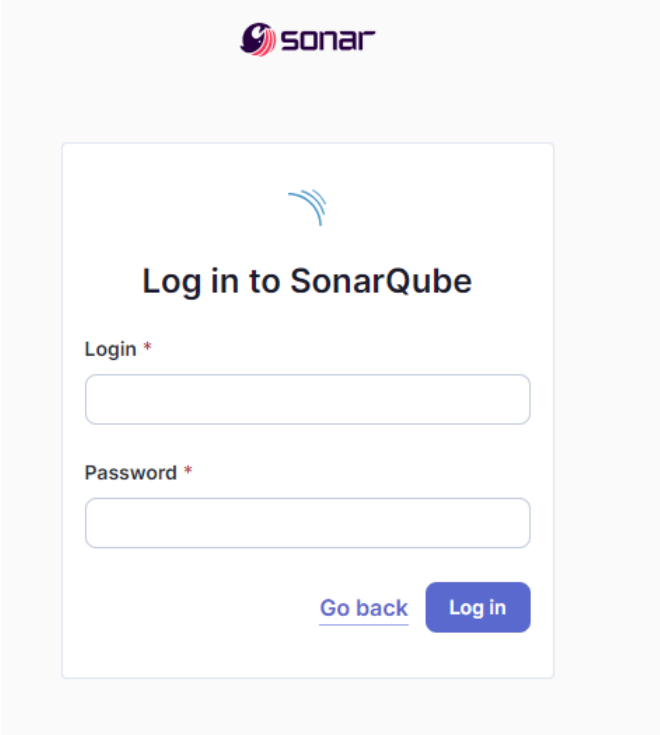
Step 1:

Run the SonarQube Docker container using the command:

```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DI
SABLE=true -p 9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar>
```

Step 2:

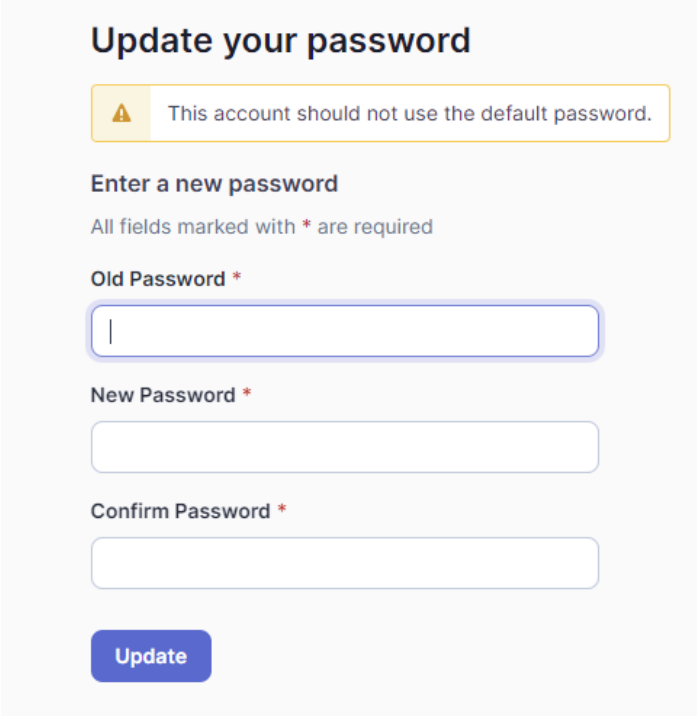
After SonarQube starts, navigate to `http://localhost:9000` in your browser.



Name:Atharv Nikam Div:D15C Roll:36

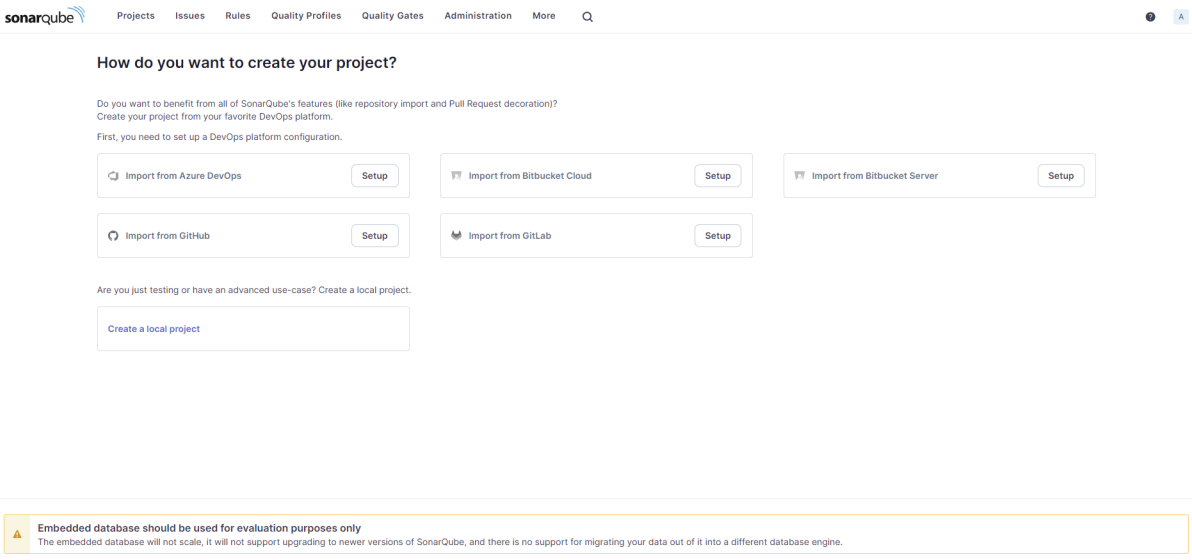
Step 3:

Log in using the default credentials (`admin`/`admin`). After logging in, you will be prompted to change the password. Ensure to note the new password.

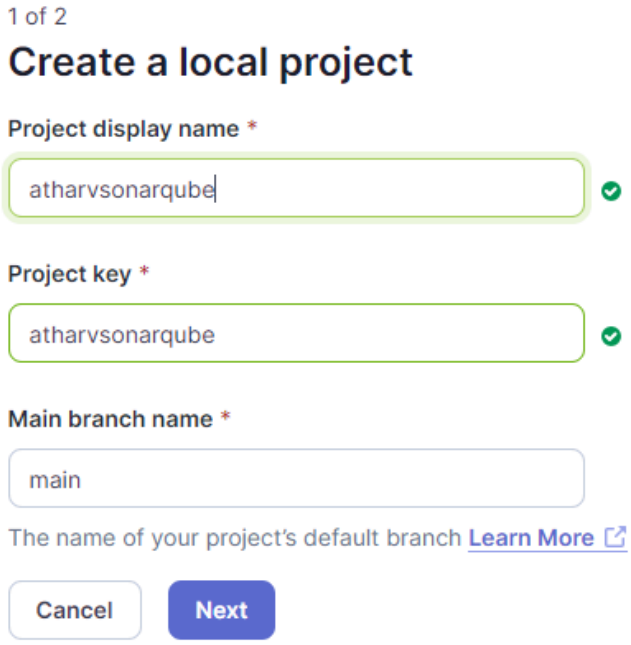


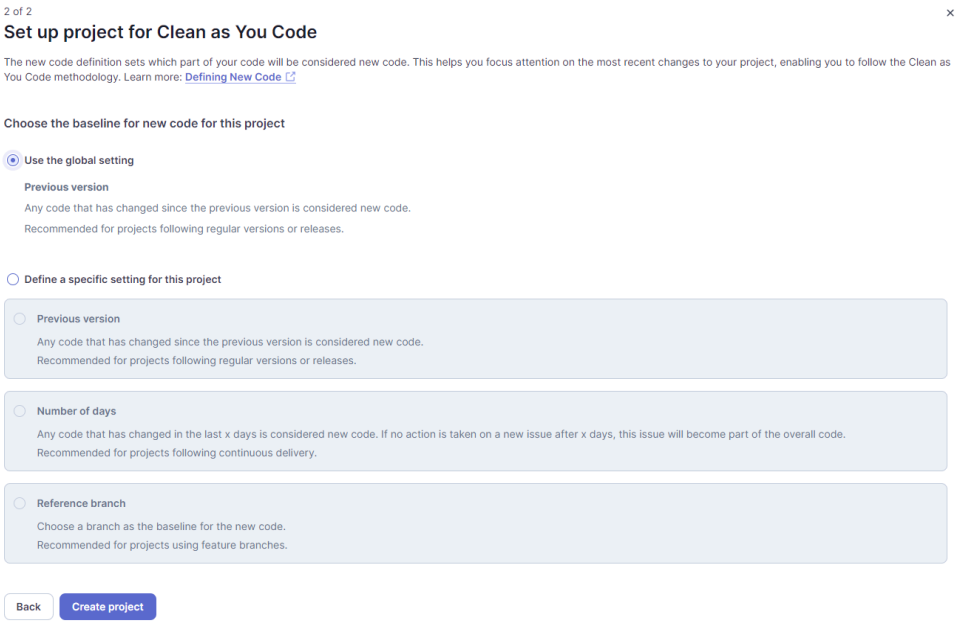
Step 4:

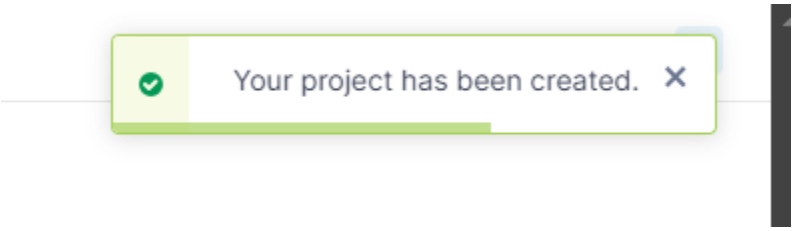
Once logged in, create a new project by clicking on "Create a Local Project." Provide a project name and a project key, then proceed.



Name:Atharv Nikam Div:D15C Roll:36



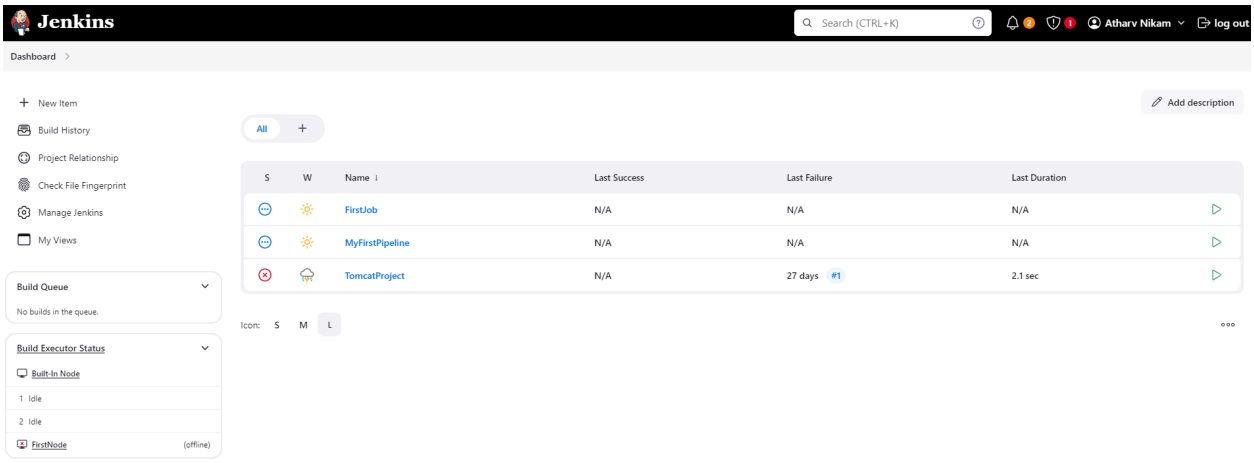




Name:Atharv Nikam Div:D15C Roll:36

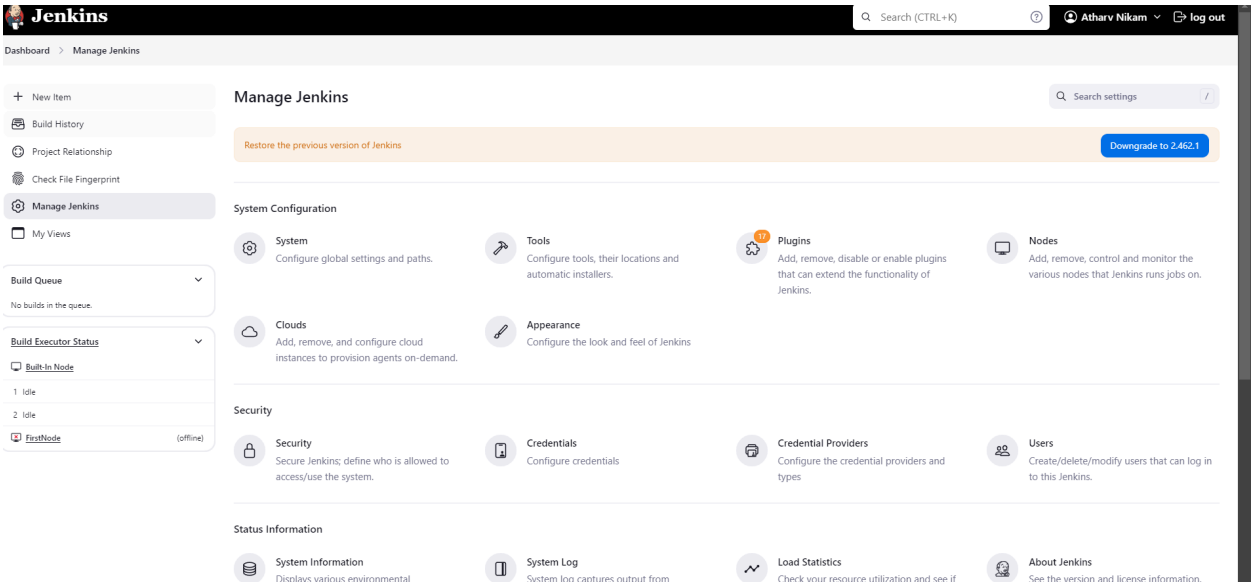
Step 5:

Open Jenkins on its assigned port (`http://localhost:<port_number>`).



Step 6:

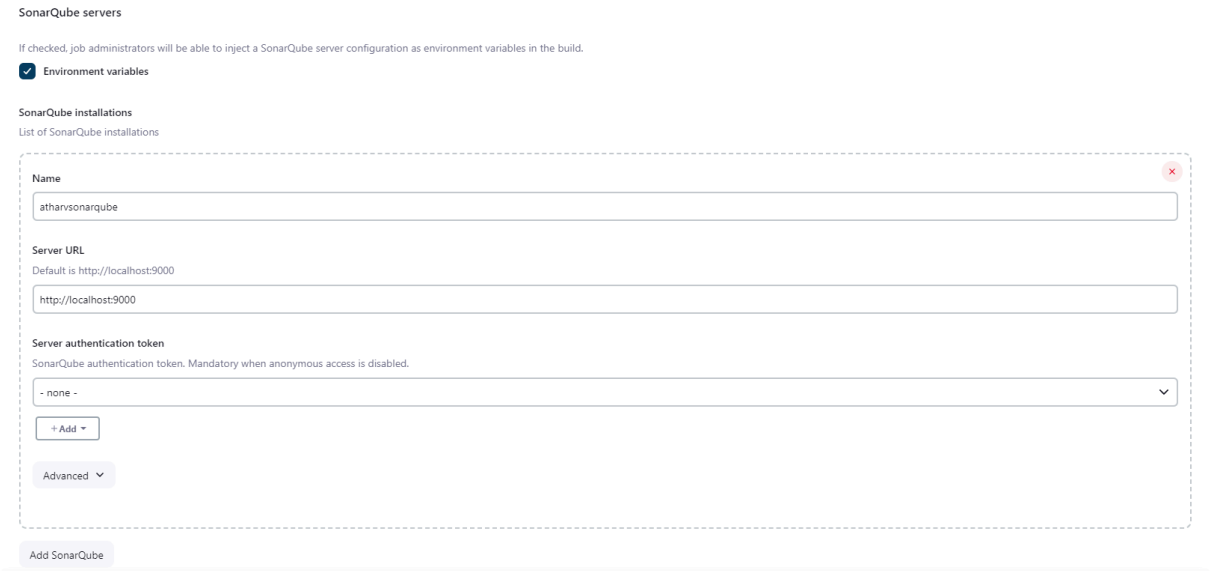
In Jenkins, go to **Manage Jenkins**, search for **SonarQube Scanner for Jenkins**, and install the plugin.



Name:Atharv Nikam Div:D15C Roll:36

Step 7:

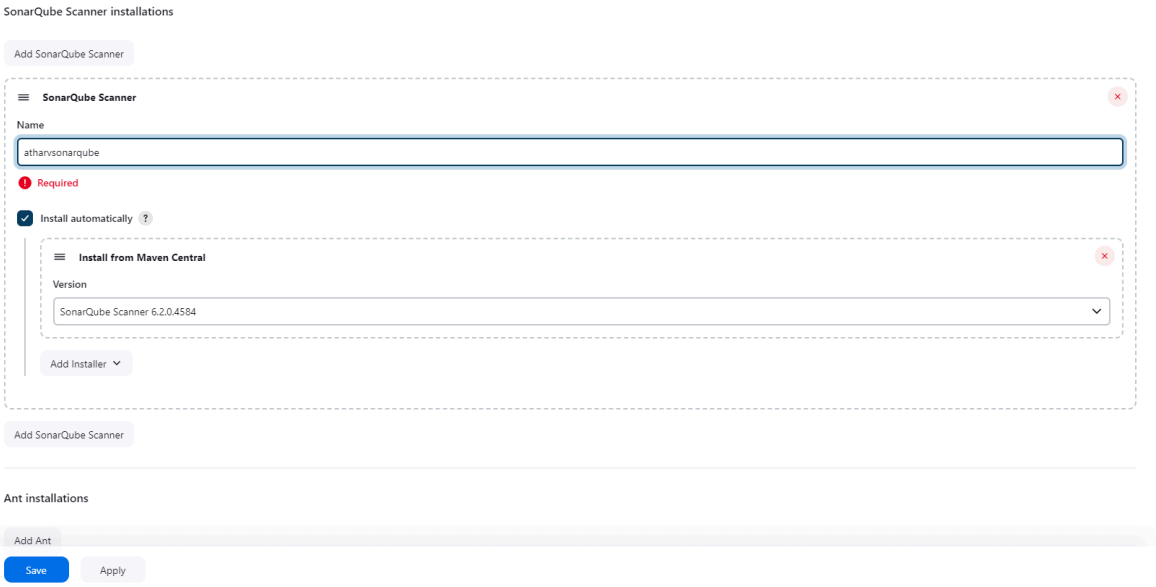
After installation, navigate to **Manage Jenkins** → **System**. Under **SonarQube servers**, add your SonarQube server and configure it with the necessary authentication token.



Name:Atharv Nikam Div:D15C Roll:36

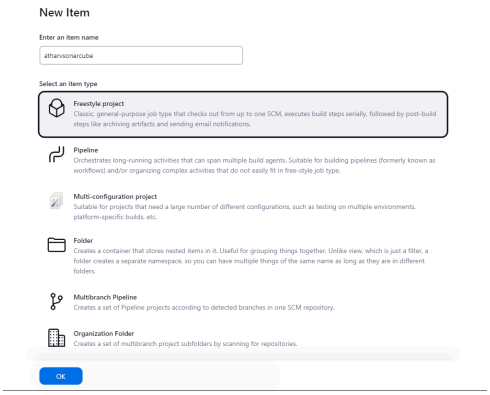
Step 8:

Next, go to **Manage Jenkins → Tools**. Under **SonarQube Scanner**, choose the latest configuration and enable automatic installation.



Step 9:

Create a new item in Jenkins, selecting a **Freestyle Project**.



Name:Atharv Nikam Div:D15C Roll:36

Step 10:
In the Source Code Management section, use this GitHub repository:

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

This repository contains a sample project to test.

Step 11:

In the Build section, add the SonarQube Scanner. Enter the required SonarQube project details such as project key, login credentials, source path, and the server URL.

sonar.projectKey=atharvsonarqube

sonar.login=admin

sonar.password=atharv@123

sonar.sources=.

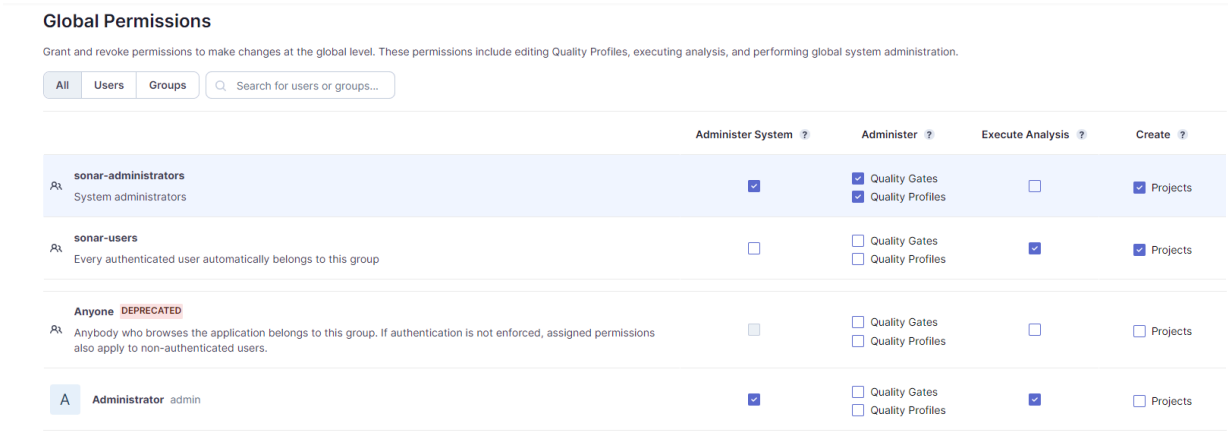
sonar.host.url=http://localhost:9000

Step 12:
Grant the local user (e.g., admin) permission to execute analysis on SonarQube by navigating to:

```
http://localhost:<port_number>/admin/permissions
```

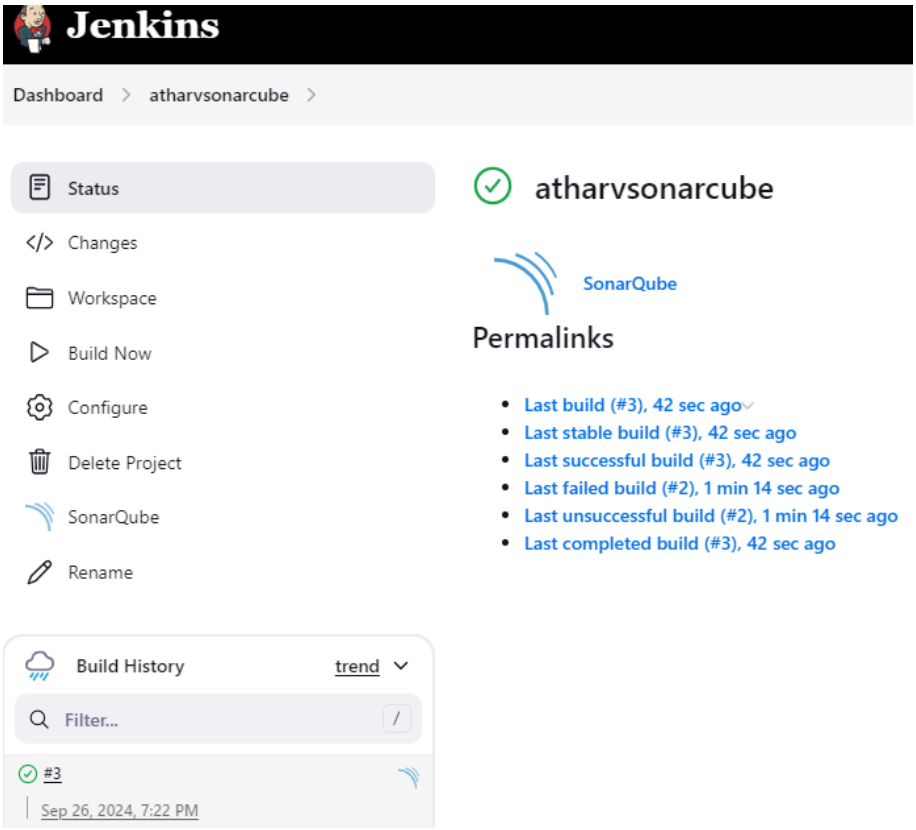
Check the "Execute Analysis" box.

Name:Atharv Nikam Div:D15C Roll:36



Step 13:

Go back to Jenkins and trigger a build by selecting **Build Now** for the project you created.



Name:Atharv Nikam Div:D15C Roll:36

Console Output

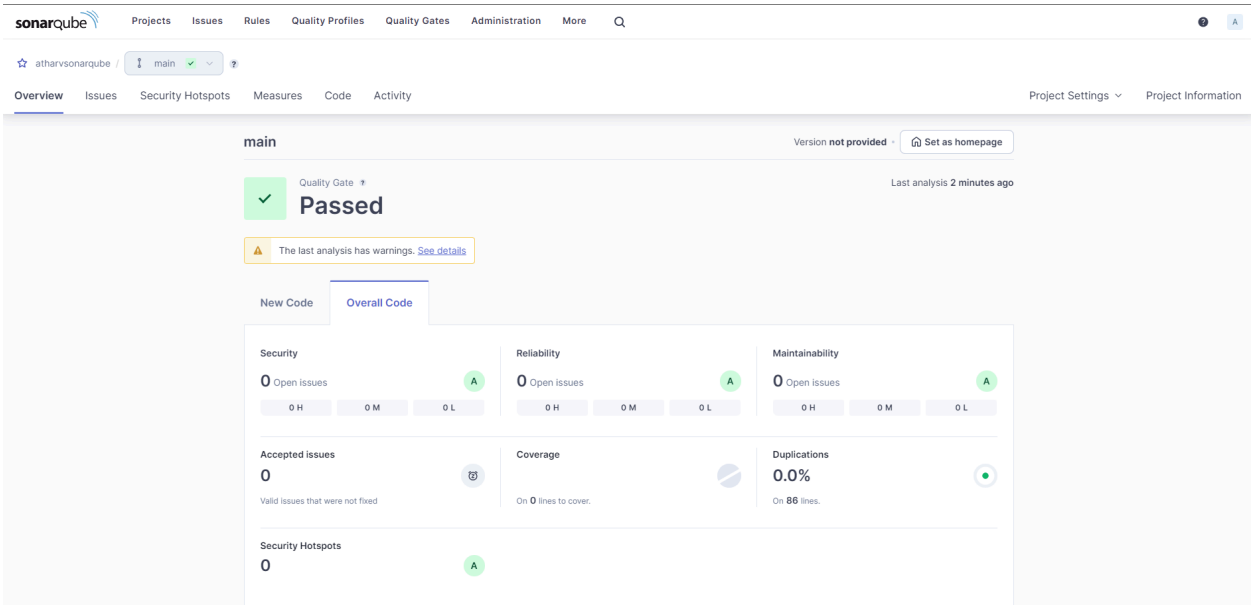
```
Started by user Atharv Nikam
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.46.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
 > git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[atharvsonarcube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarqube\bin\sonar-scanner.bat -Dsonar.hos
Dsonar.projectKey=atharvsonarqube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=atharv@123 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
19:23:00.321 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
19:23:00.321 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarqube\bin\..
19:23:00.321 INFO  Project root configuration file: NONE
19:23:00.339 INFO  SonarScanner CLI 6.2.0.4584
19:23:00.339 INFO  Java 20 Oracle Corporation (64-bit)
19:23:00.339 INFO  Windows 11 10.0 amd64
19:23:00.354 INFO  User cache: C:\WINDOWS\system32\config\systemprofile\.sonar\cache
19:23:01.038 INFO  JRE provisioning: os[windows], arch[amd64]
19:23:04.310 INFO  Communicating with SonarQube Server 10.6.0.92116
19:23:04.678 INFO  Starting SonarScanner Engine...
19:23:04.678 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
19:23:05.407 INFO  Load global settings
```

```
19:23:21.882 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
19:23:21.882 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=0ms
19:23:21.882 INFO  Sensor Zero Coverage Sensor
19:23:21.882 INFO  Sensor Zero Coverage Sensor (done) | time=0ms
19:23:21.882 INFO  SCM Publisher SCM provider for this project is: git
19:23:21.882 INFO  SCM Publisher 4 source files to be analyzed
19:23:22.262 INFO  SCM Publisher 4/4 source files have been analyzed (done) | time=380ms
19:23:22.262 INFO  CPD Executor Calculating CPD for 0 files
19:23:22.262 INFO  CPD Executor CPD calculation finished (done) | time=0ms
19:23:22.269 INFO  SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
19:23:22.409 INFO  Analysis report generated in 53ms, dir size=201.0 kB
19:23:22.441 INFO  Analysis report compressed in 16ms, zip size=22.5 kB
19:23:22.615 INFO  Analysis report uploaded in 171ms
19:23:22.615 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarqube
19:23:22.615 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
19:23:22.615 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=3cfcbe0c-0055-497d-a657-1cad4d6e2a05
19:23:22.615 INFO  Analysis total time: 14.707 s
19:23:22.615 INFO  SonarScanner Engine completed successfully
19:23:22.661 INFO  EXECUTION SUCCESS
19:23:22.661 INFO  Total time: 22.340s
Finished: SUCCESS
```

Name:Atharv Nikam Div:D15C Roll:36

Step 14:

Once the build process is complete, head back to SonarQube and verify the analysis results linked to your project.



Conclusion:

Through this experiment, we explored how to set up and perform Static Application Security Testing (SAST) using Jenkins in combination with SonarQube. By leveraging Docker, we utilized a SonarQube container without needing a local installation. Following the setup of Jenkins and SonarQube, we analyzed a sample project from GitHub for potential vulnerabilities. Once the project was built, the analysis confirmed the security status, providing valuable insight into the effectiveness of SAST in the CI/CD pipeline.

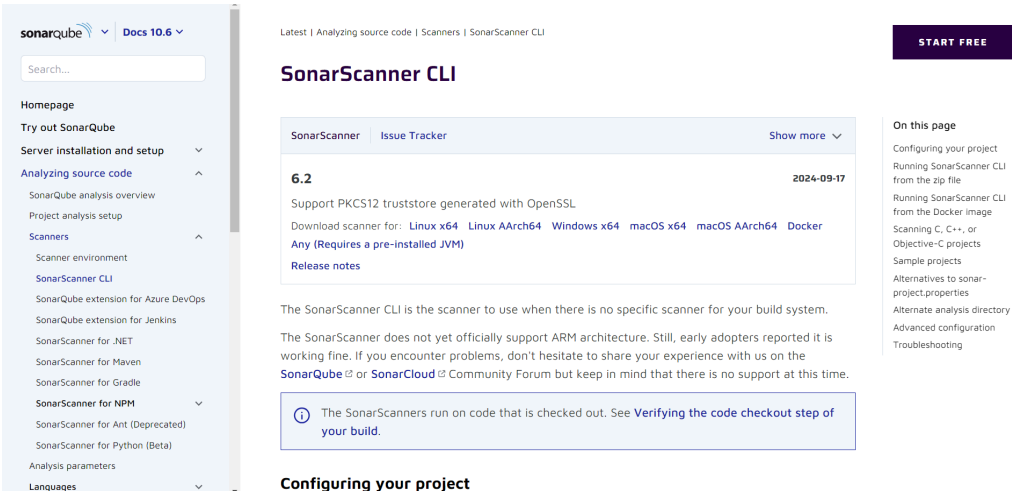
Name:Atharv Nikam **Div:D15C** **Roll No:36**

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.

Prerequisites:

1. **Download Sonar Scanner:**
   Access the SonarQube documentation and download the SonarQube scanner CLI from this link:

https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/



2. After downloading, extract the zip file into a designated folder.

Install Docker:
Run the following command to verify Docker is installed:

```
PS C:\Users\athar> docker -v
Docker version 27.1.1, build 6312585
PS C:\Users\athar>
```

3 .Pull SonarQube Docker Image:
Install the SonarQube image by executing:

Copy code
```
docker pull sonarqube
```

```
PS C:\Users\athar>  docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

4. Ensure Jenkins is installed:
Confirm that Jenkins is installed and configured on your system.

Experiment Steps:

Step 1:
Run the SonarQube Docker container by entering the command below:

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

```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DI
SABLE=true -p 9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar>
```

Step 2:

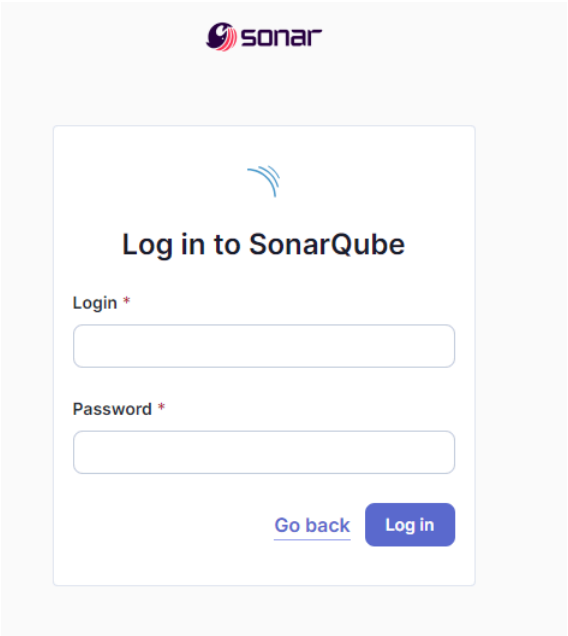
After SonarQube is running, open your browser and go to `http://localhost:9000`.

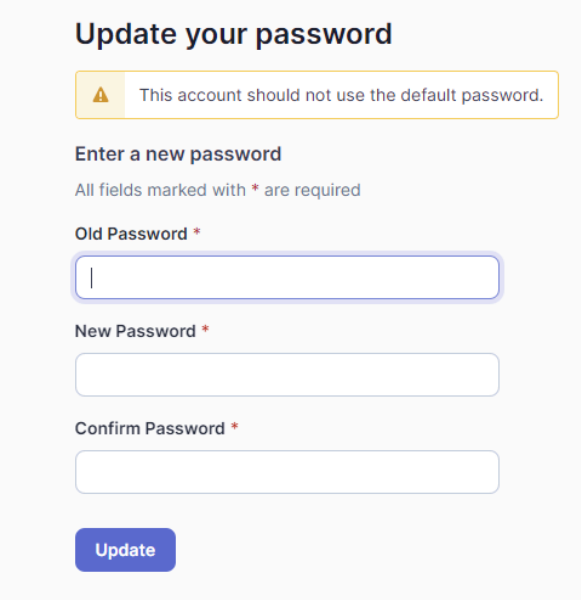
Step 3:
Log in to SonarQube using the default credentials:

`Username: admin  Password: admin`

`You will be asked to reset the password after logging in for the first time. Set a new password and remember it.`

Name:Atharv Nikam Div:D15C Roll :36

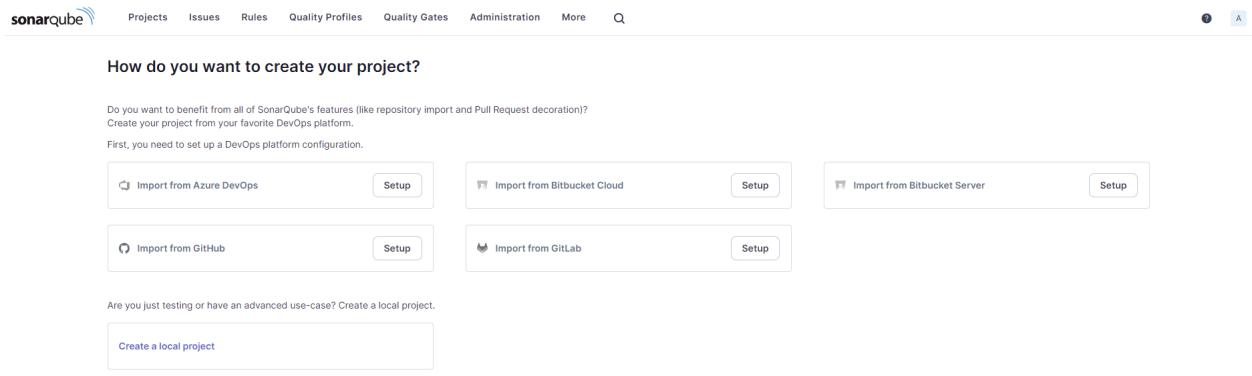


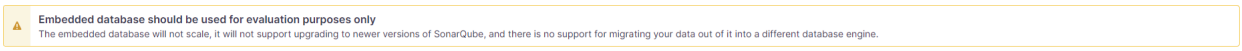


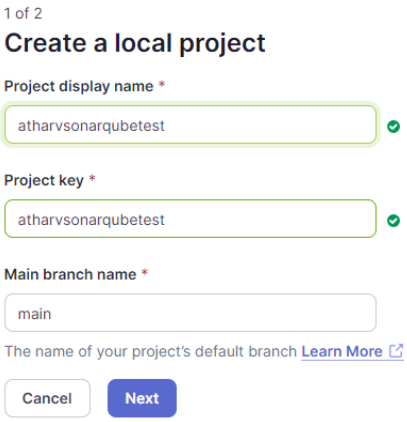
Name:Atharv Nikam Div:D15C Roll :36

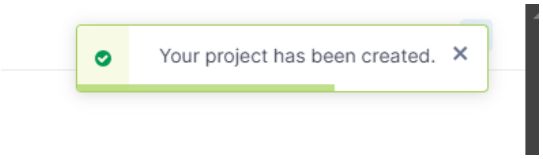
Step 4:

On the SonarQube dashboard, click **Create a Local Project**. Provide a project name and a unique project key.

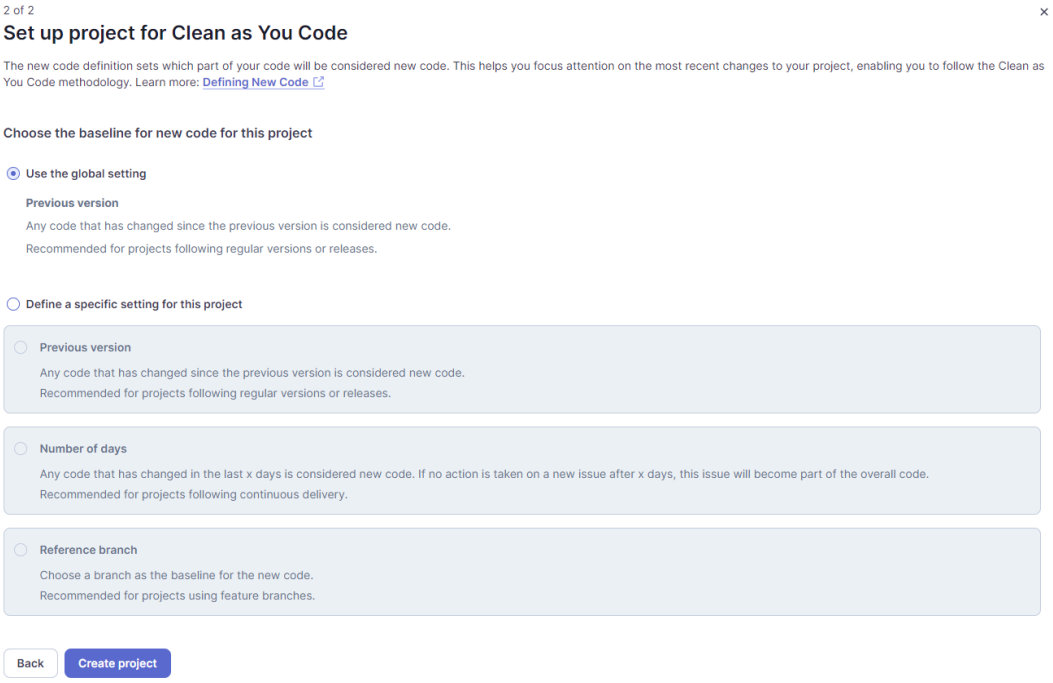








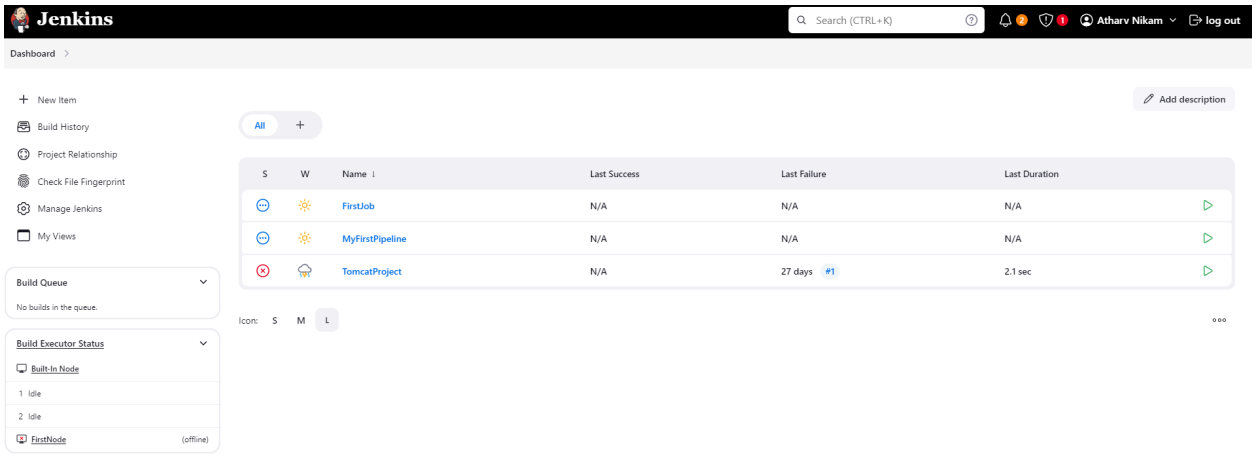
Name:Atharv Nikam Div:D15C Roll :36



Step 5:

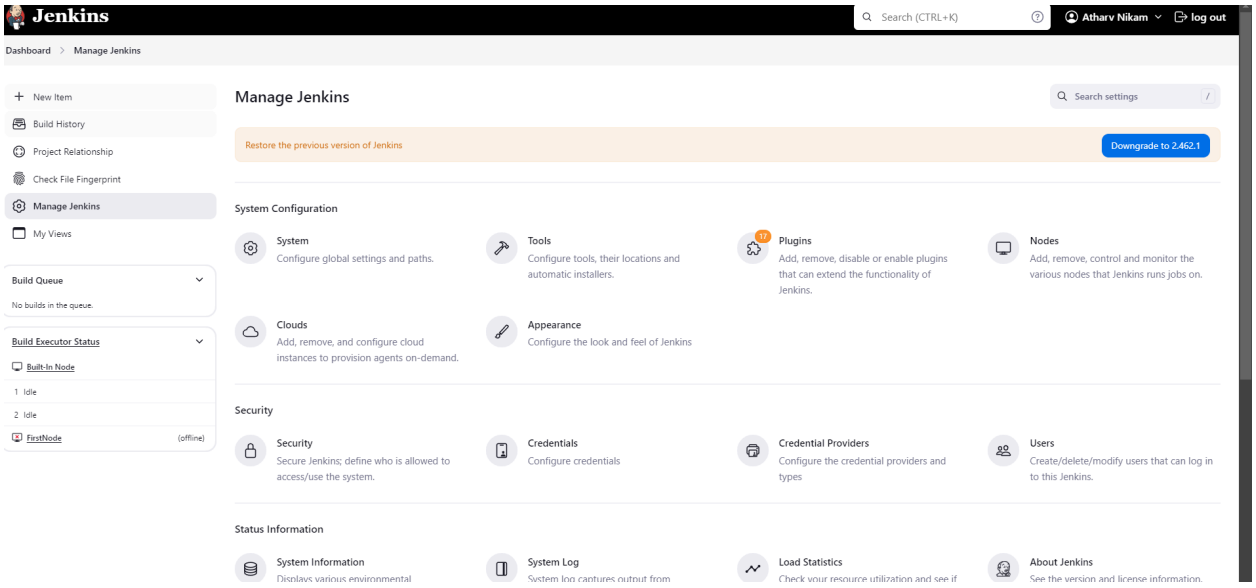
Open Jenkins by navigating to the port on which it is installed:

http://localhost:<port_number>



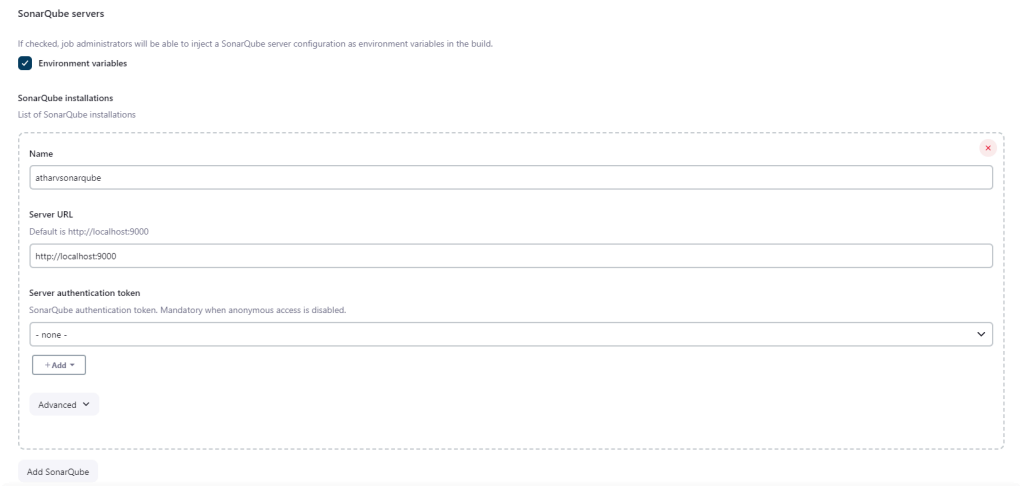
Name:Atharv Nikam Div:D15C Roll :36

Step 6:
In Jenkins, go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner for Jenkins**. Install the plugin.

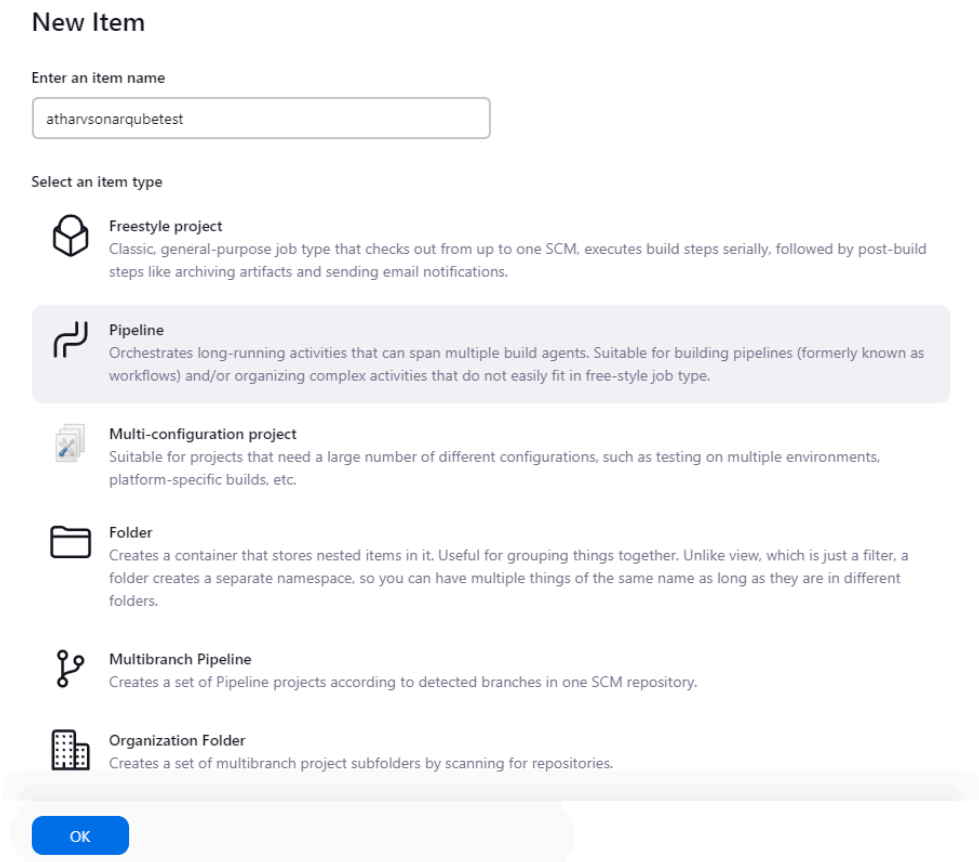


Step 7:
Once installed, head to **Manage Jenkins → System**. Under **SonarQube Servers**, add your SonarQube server, and provide any necessary authentication tokens.

Step 8:
Next, under **Manage Jenkins → Tools**, navigate to **SonarQube Scanner** and configure it to automatically install the latest version.



Step 9:
Create a new pipeline item in Jenkins



Name:Atharv Nikam Div:D15C Roll :36

Step 10:

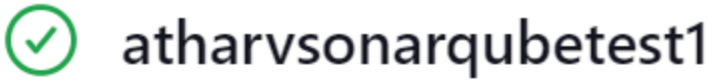
In the pipeline script section, input the following:

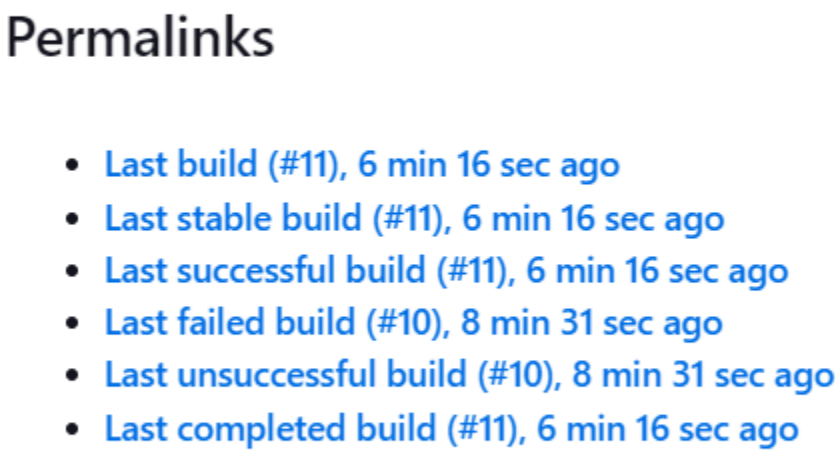
```
node {
  stage('Cloning the GitHub Repo') {
     git 'https://github.com/shazforiot/GOL.git'
  }

  stage('SonarQube Analysis') {
     withSonarQubeEnv('atharvsonarqube') {
        bat """
        <PATH_TO_SONARSCANNER_FOLDER>\\bin\\sonar-scanner.bat ^
        -D sonar.login=<SONARQUBE_LOGIN> ^
        -D sonar.password=<SONARQUBE_PASSWORD> ^
        -D sonar.projectKey=<PROJECT_KEY> ^
        -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
        -D sonar.host.url=http://localhost:9000/
        """
     }
  }
}
```

This script clones a sample Java project from GitHub, which has several issues that SonarQube will detect.

Name:Atharv Nikam Div:D15C Roll :36

Step 11:
Go back to Jenkins, select the job you just created, and click **Build Now** to run the pipeline.



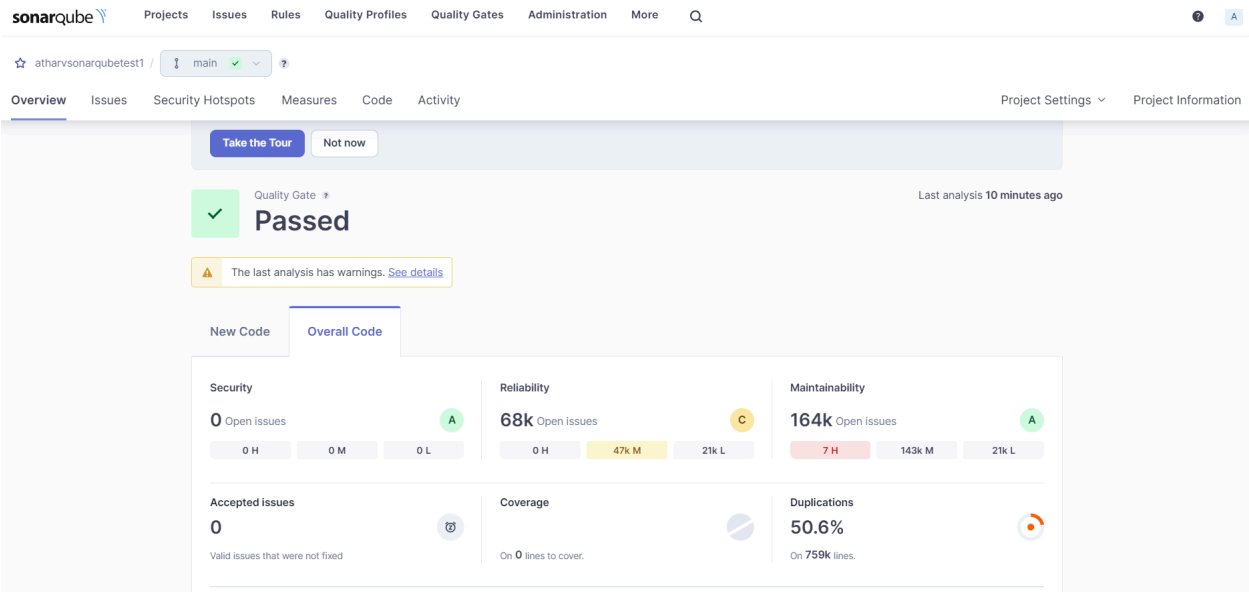



```
20:29:07.725 INFO  CPD Executor CPD calculation finished (done) | time=71153ms
20:29:07.910 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:30:07.200 INFO  Analysis report generated in 3510ms, dir size=126.4 MB
20:30:16.243 INFO  Analysis report compressed in 9029ms, zip size=29.5 MB
20:30:18.716 INFO  Analysis report uploaded in 2466ms
20:30:18.723 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarqubetest1
20:30:18.723 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:30:18.723 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=e3321ac6-d5a9-4e20-be13-d9d71fb2c392
20:30:31.564 INFO  Analysis total time: 5:54.794 s
20:30:31.583 INFO  SonarScanner Engine completed successfully
20:30:32.216 INFO  EXECUTION SUCCESS
20:30:32.278 INFO  Total time: 5:59.482s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Name:Atharv Nikam Div:D15C Roll :36

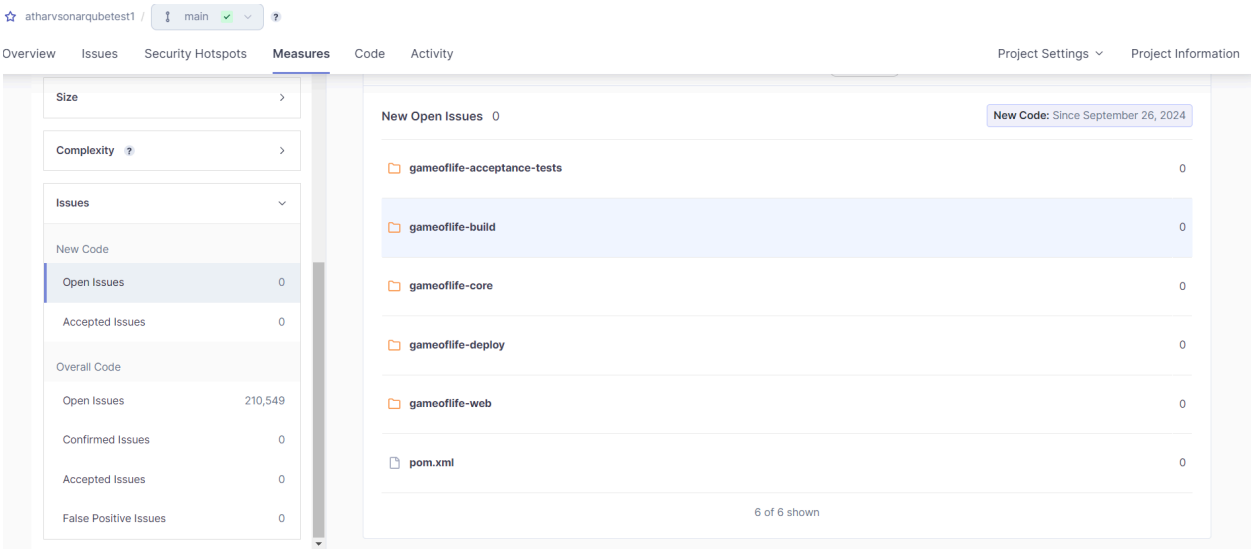
Step 12:

Once the build is complete, return to SonarQube to view the analysis of your project. Check for bugs, code smells, duplications, and other metrics related to the quality of your code.



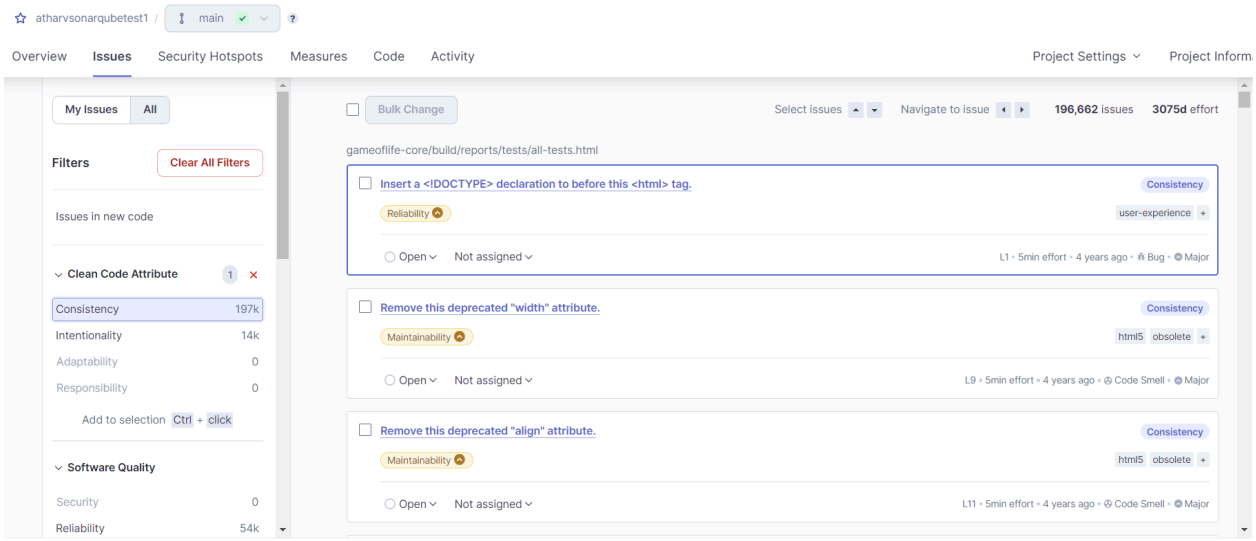
Under different tabs, check all the issues with the code.

- Code Problems

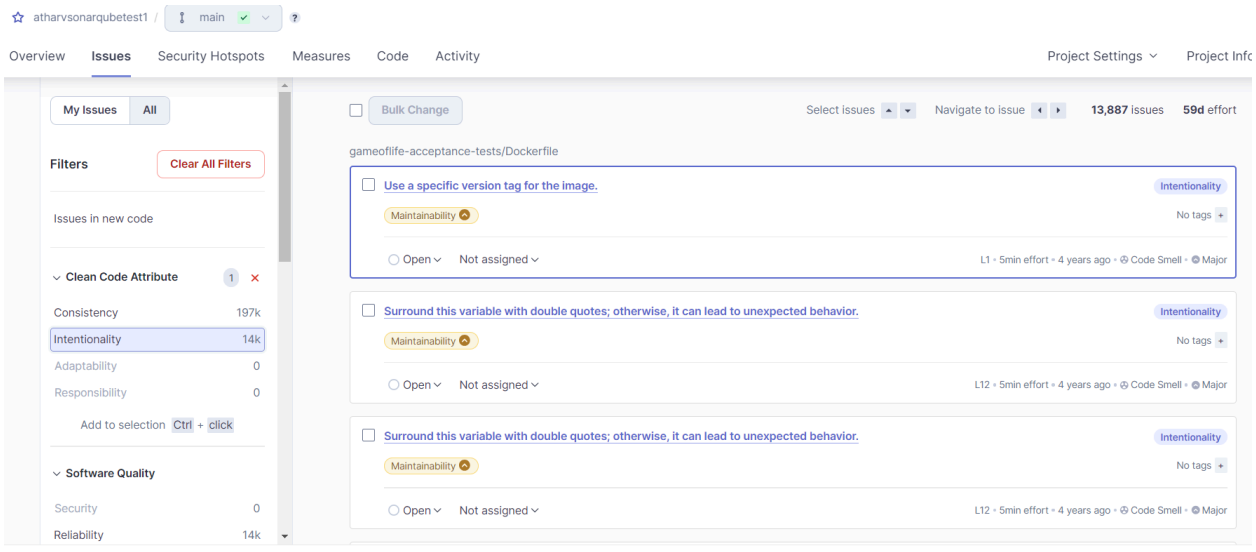


Name:Atharv Nikam Div:D15C Roll :36

- Consistency

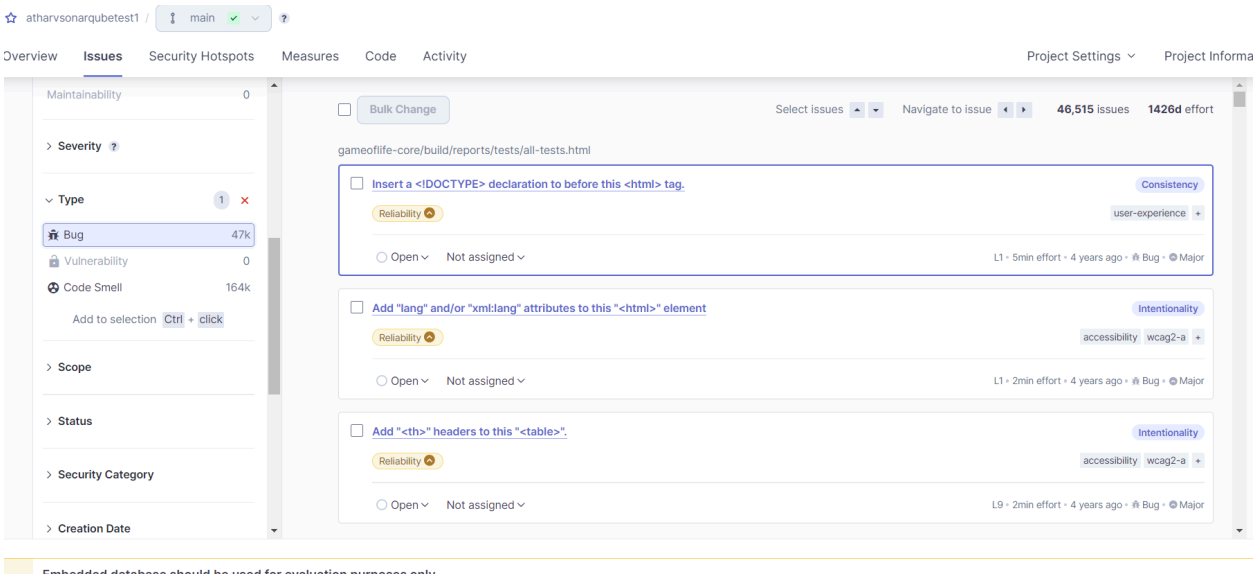


- Intentionality

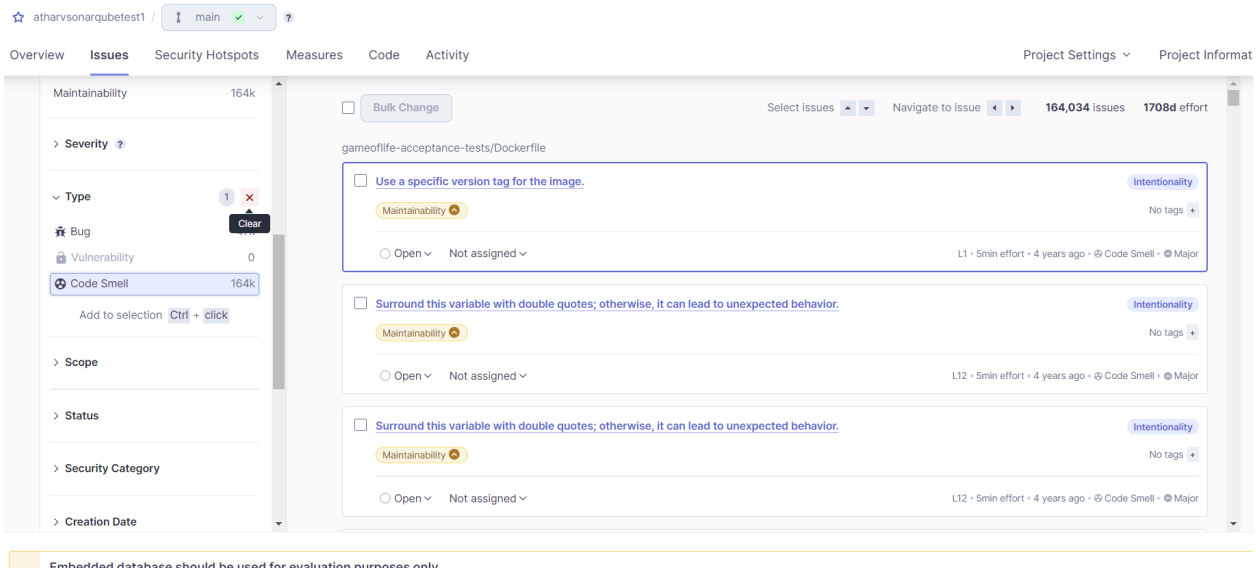


Name:Atharv Nikam Div:D15C Roll :36

- Bugs

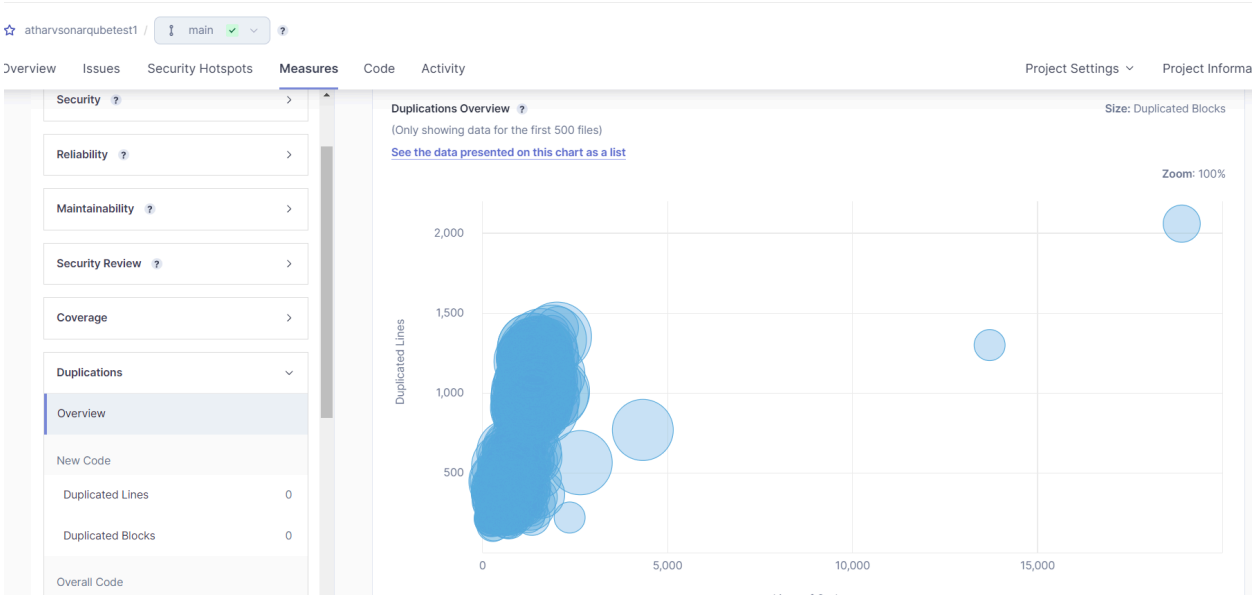


- Code Smells

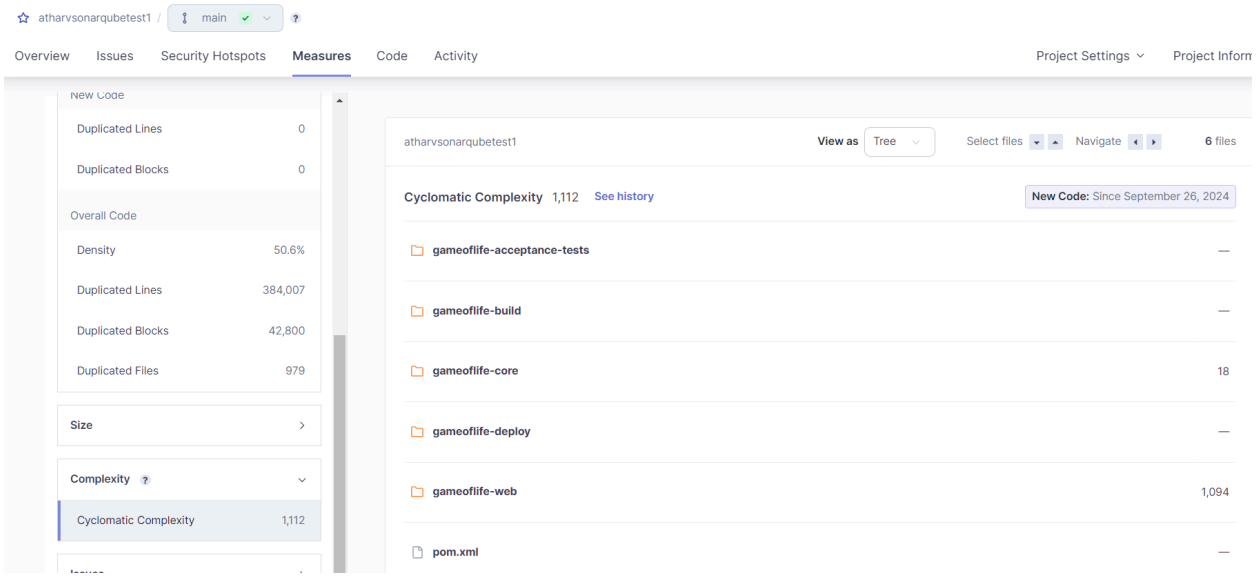


Name:Atharv Nikam Div:D15C Roll :36

- Duplications



- Cyclomatic Complexities



Conclusion:

This experiment allowed us to integrate Jenkins and SonarQube to set up a CI/CD pipeline capable of performing static analysis on Java code. Through this process, we automated the detection of common code issues such as bugs, code smells, and duplications. By leveraging Docker for SonarQube and the Jenkins pipeline, we streamlined the code scanning process, ensuring any issues were highlighted during the build phase. This integration demonstrates the importance of automated code quality checks in a continuous delivery environment

Name:Atharv Nikam **Div:D15C** **Roll:36**

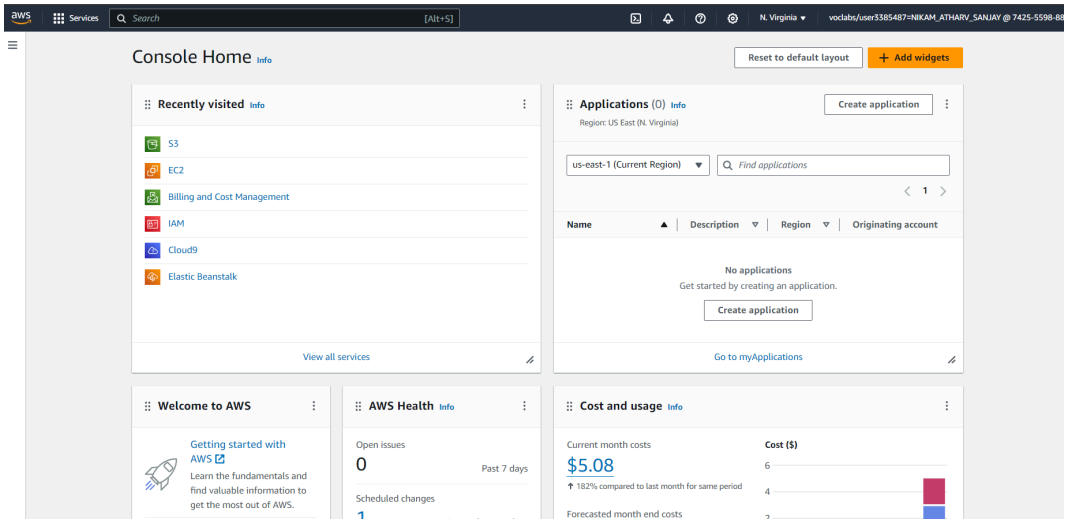
Experiment 9

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

STEPS

Step 1: Accessing AWS EC2

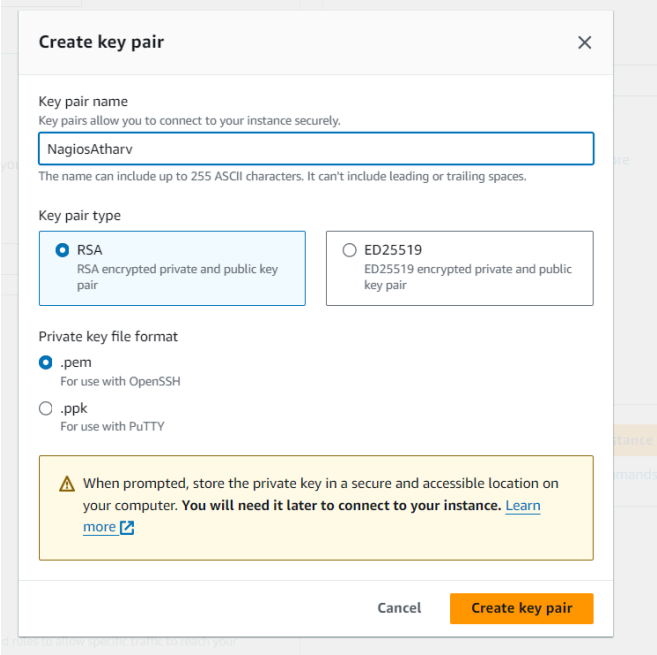
Log in to your AWS account, search for EC2 in the services section, and open the interface. Click on "Create Instance" and select **Amazon Linux** as the OS image for the instance.



Step 2: Setting Up Key Pair

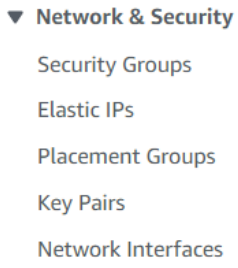
If you don't already have a key pair or `.pem` file, generate a new key pair. If one exists, select it. Ensure that the `.pem` file is saved securely on your system for future use.

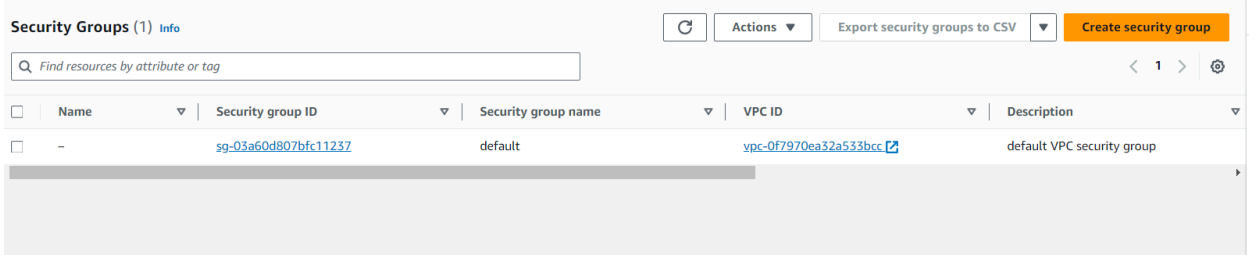
Name:Atharv Nikam Div :D15C Roll: 36



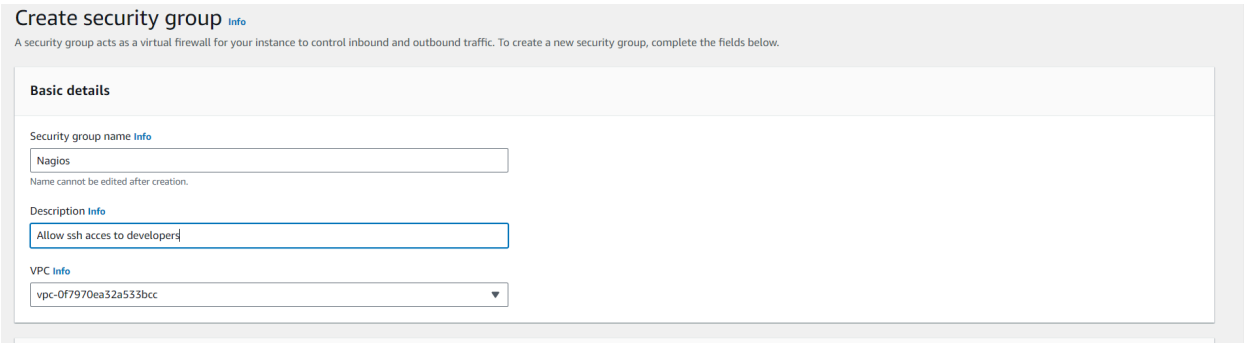
Step 3: Configuring Security Group

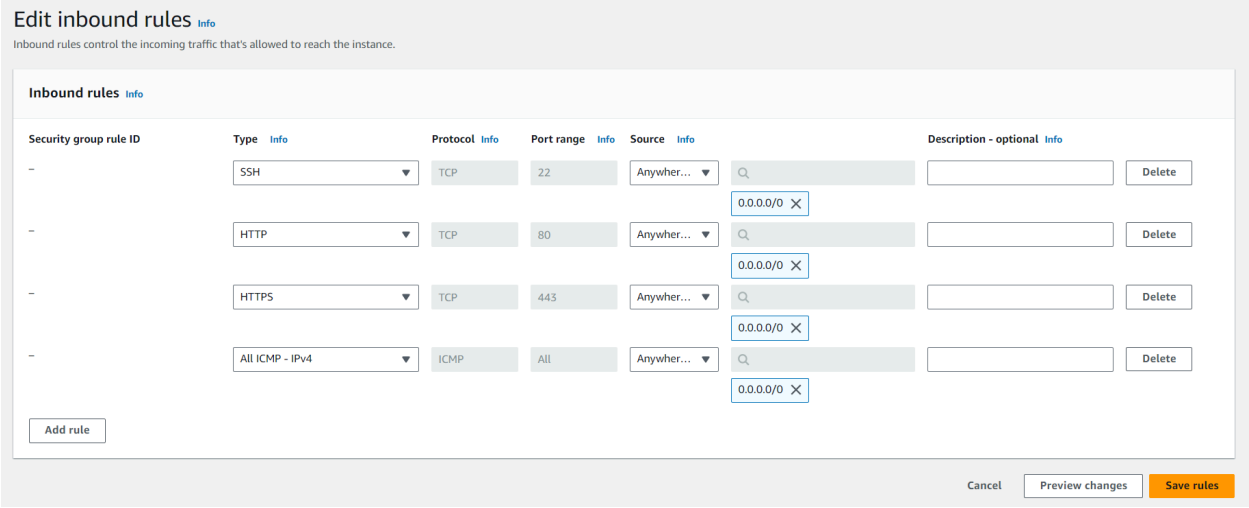
Navigate to the **Security Groups** section from the left panel. Locate the security group linked to your instance, click on the instance ID, and edit the **Inbound Rules**. Add the following rules: HTTP, HTTPS, Custom TCP (port 5666), All ICMP (IPv4 and IPv6), and All traffic. Save the rules.

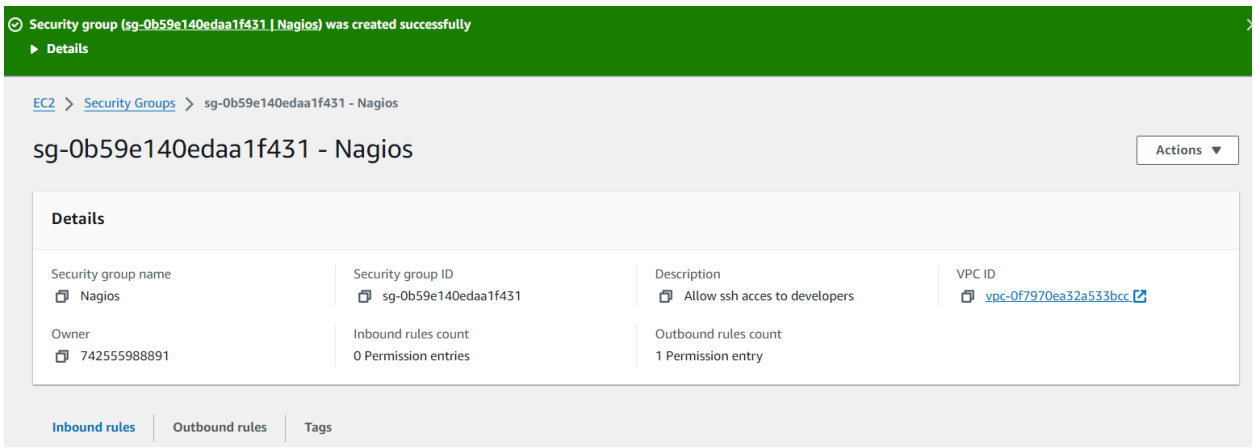




Name:Atharv Nikam Div :D15C Roll: 36



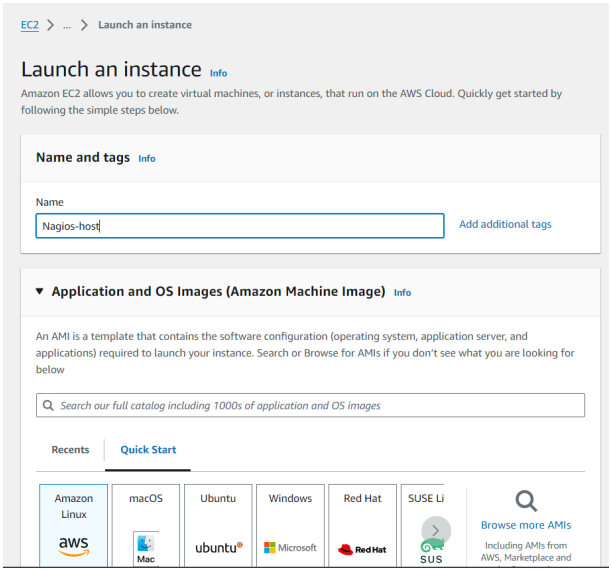


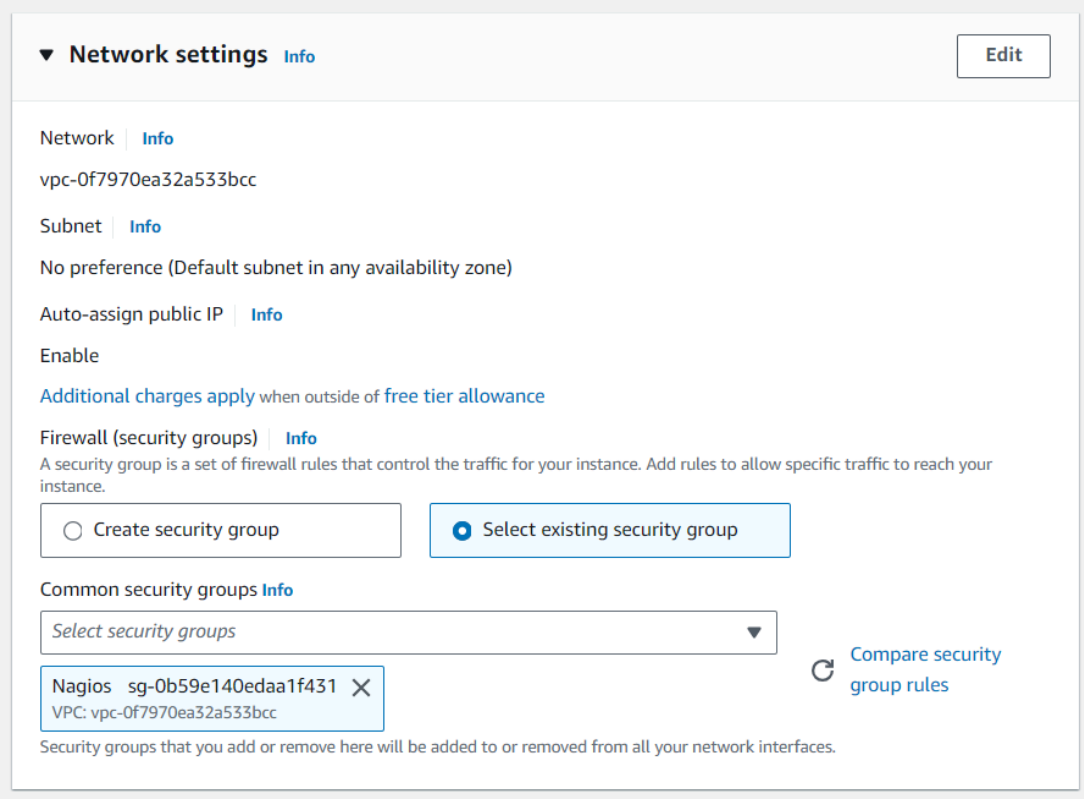


Name:Atharv Nikam Div :D15C Roll: 36

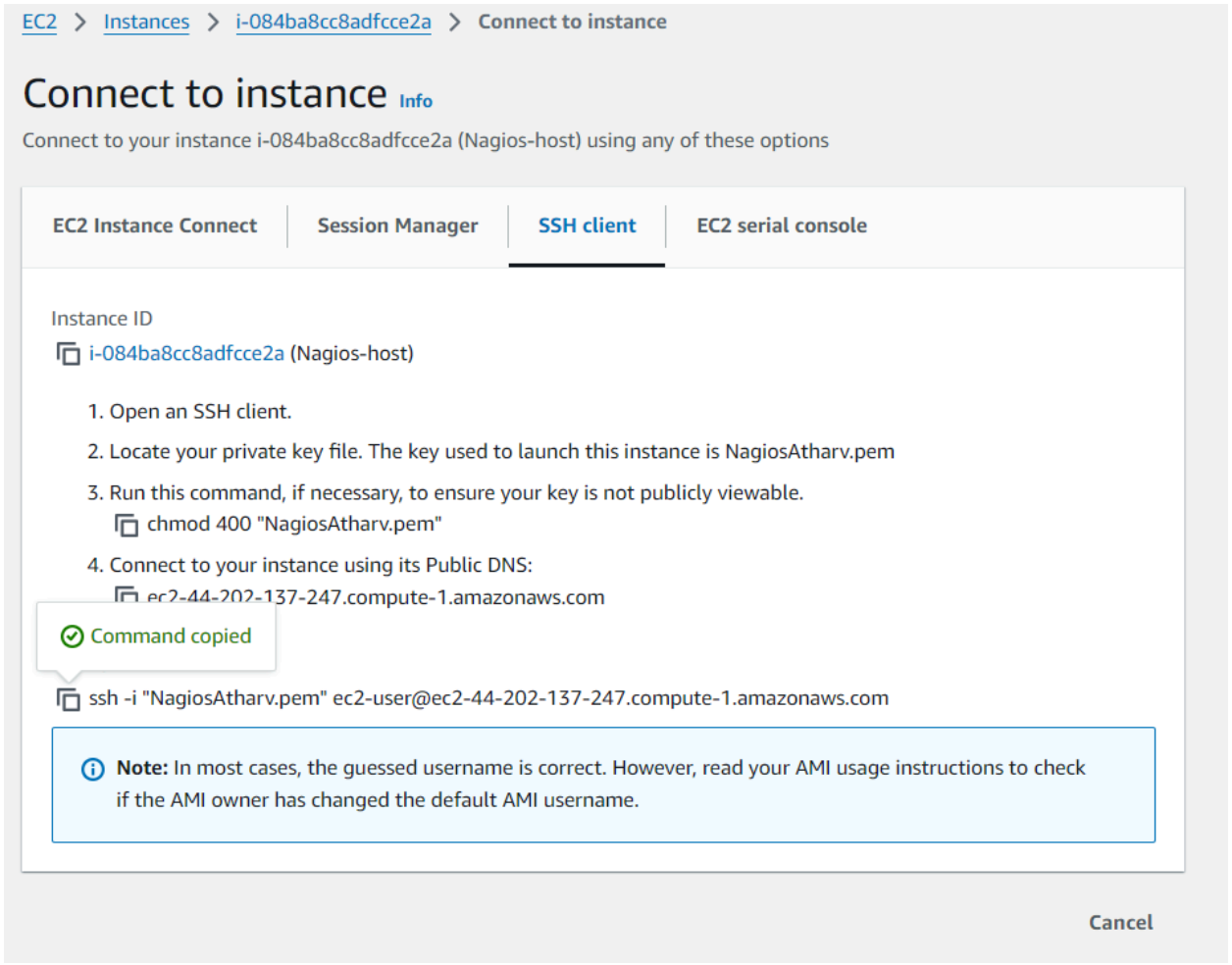
Step 4: Connecting to EC2 Instance

Back in the EC2 dashboard, click on the instance ID and choose **Connect**. Under **SSH Client**, copy the provided command. Open your terminal where the `.pem` file is located, paste the SSH command, and execute it to connect.





Name:Atharv Nikam Div :D15C Roll: 36

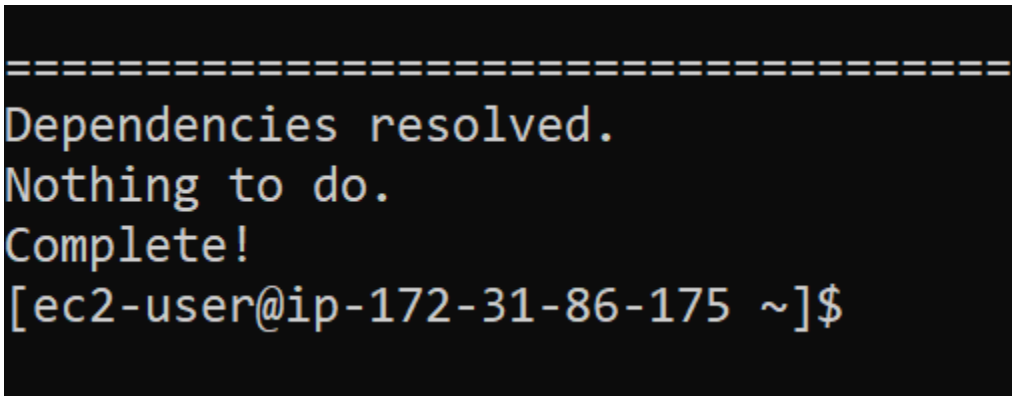



```
PS D:\Advancedevops key> ssh -i "NagiosAtharv.pem" ec2-user@ec2-44-201-253-108.compute-1.amazonaws.com

A newer release of "Amazon Linux" is available.
  Version 2023.5.20241001:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-86-175 ~]$
```

Name:Atharv Nikam Div :D15C Roll: 36

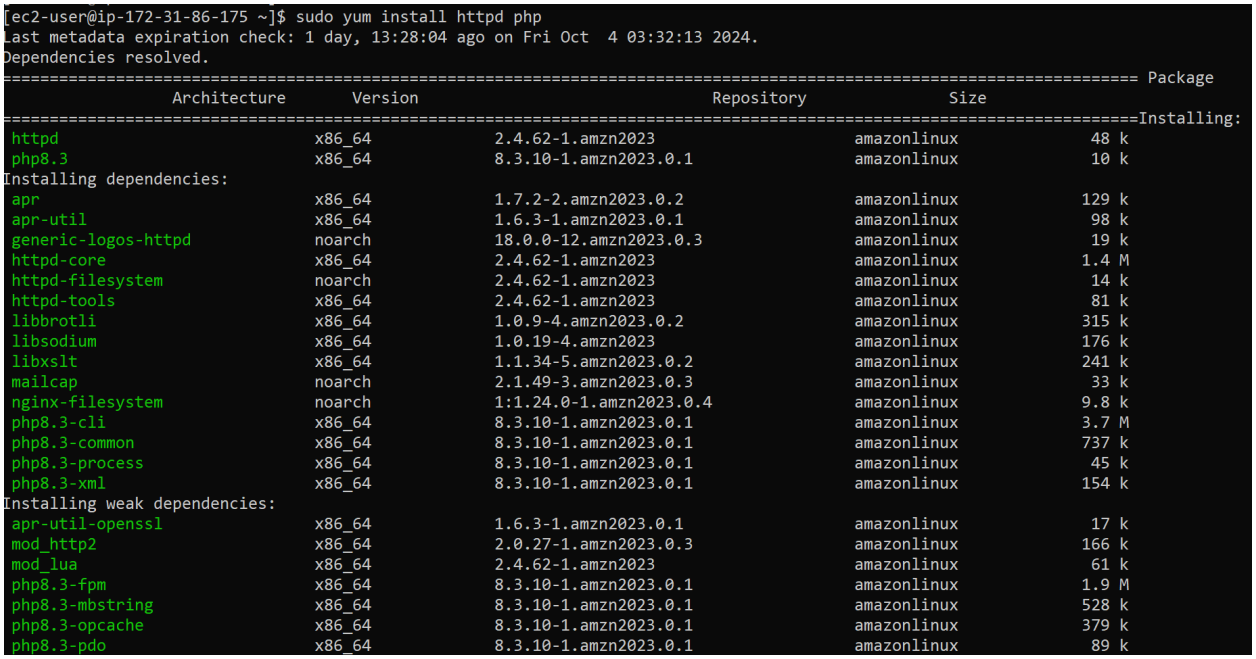
Step 5: Updating YUM Packages

Run `sudo yum update` to update the system's YUM package manager. This ensures the latest updates and security patches are installed



Step 6: Installing Apache and PHP

Install the necessary web server components, Apache and PHP, by running: `sudo yum install httpd php`.



Name:Atharv Nikam Div :D15C Roll: 36

```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                          apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64             generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.62-1.amzn2023.x86_64                           httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch                httpd-tools-2.4.62-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64                    libsodium-1.0.19-4.amzn2023.x86_64
  libxslt-1.1.34-5.amzn2023.0.2.x86_64                     mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64                   mod_lua-2.4.62-1.amzn2023.x86_64
  nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch          php8.3-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64                  php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64                  php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64              php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-process-8.3.10-1.amzn2023.0.1.x86_64              php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

Complete!
```

Step 7: Installing C/C++ Compiler

Install the GCC compiler along with C libraries using:
`sudo yum install gcc glibc glibc-common`.

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 1 day, 13:29:48 ago on Fri Oct  4 03:32:13 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
================================================================================================================================
 Package                        Architecture          Version                                Repository                   Size
================================================================================================================================
Installing:
 gcc                            x86_64                11.4.1-2.amzn2023.0.2                  amazonlinux                  32 M
Installing dependencies:
 annobin-docs                   noarch                10.93-1.amzn2023.0.1                   amazonlinux                  92 k
 annobin-plugin-gcc             x86_64                10.93-1.amzn2023.0.1                   amazonlinux                 887 k
 cpp                            x86_64                11.4.1-2.amzn2023.0.2                  amazonlinux                  10 M
 gc                             x86_64                8.0.4-5.amzn2023.0.2                   amazonlinux                 105 k
 glibc-devel                    x86_64                2.34-52.amzn2023.0.11                  amazonlinux                  27 k
 glibc-headers-x86              noarch                2.34-52.amzn2023.0.11                  amazonlinux                 427 k
 guile22                        x86_64                2.2.7-2.amzn2023.0.3                   amazonlinux                 6.4 M
 kernel-headers                 x86_64                6.1.109-118.189.amzn2023               amazonlinux                 1.4 M
 libmpc                         x86_64                1.2.1-2.amzn2023.0.2                   amazonlinux                  62 k
 libtool-ltdl                   x86_64                2.4.7-1.amzn2023.0.3                   amazonlinux                  38 k
 libxcrypt-devel                x86_64                4.4.33-7.amzn2023                      amazonlinux                  32 k
 make                           x86_64                1:4.3-5.amzn2023.0.2                   amazonlinux                 534 k

Transaction Summary
================================================================================================================================
Install  13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]:
```

```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch                 annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  cpp-11.4.1-2.amzn2023.0.2.x86_64                         gc-8.0.4-5.amzn2023.0.2.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64                         glibc-devel-2.34-52.amzn2023.0.11.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch           guile22-2.2.7-2.amzn2023.0.3.x86_64
  kernel-headers-6.1.109-118.189.amzn2023.x86_64           libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64                 libxcrypt-devel-4.4.33-7.amzn2023.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64

Complete!
```

Name:Atharv Nikam Div :D15C Roll: 36

Step 8: Installing GD Library

For rendering images in Nagios, install the GD library with:
`sudo yum install gd gd-devel`.

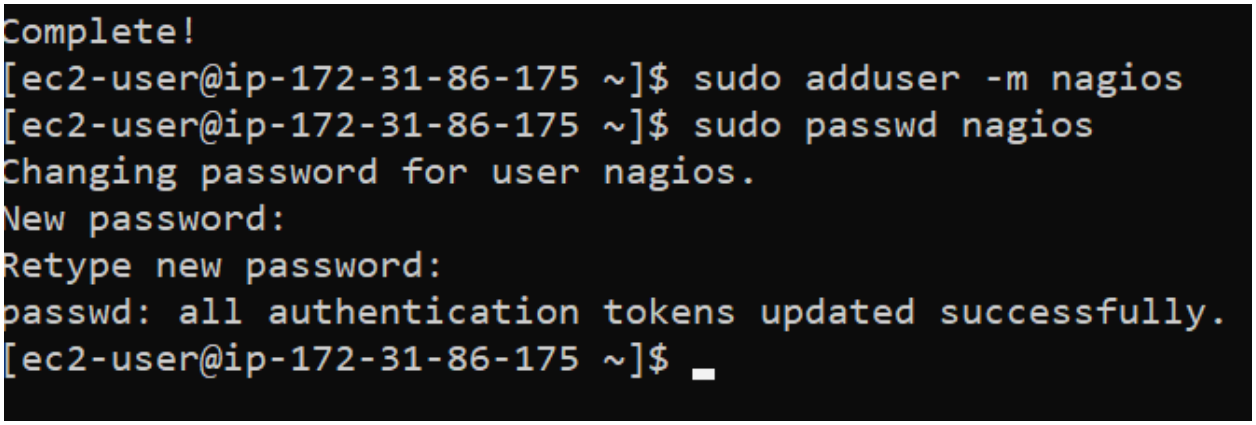
```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 1 day, 13:31:19 ago on Fri Oct  4 03:32:13 2024.
Dependencies resolved.
================================================================================================================================
 Package                          Architecture        Version                              Repository                Size
================================================================================================================================
Installing:
 gd                               x86_64              2.3.3-5.amzn2023.0.3                 amazonlinux              139 k
 gd-devel                         x86_64              2.3.3-5.amzn2023.0.3                 amazonlinux               38 k
Installing dependencies:
 brotli                           x86_64              1.0.9-4.amzn2023.0.2                 amazonlinux              314 k
 brotli-devel                     x86_64              1.0.9-4.amzn2023.0.2                 amazonlinux               31 k
 bzip2-devel                      x86_64              1.0.8-6.amzn2023.0.2                 amazonlinux              214 k
 cairo                            x86_64              1.17.6-2.amzn2023.0.1                amazonlinux              684 k
 cmake-filesystem                 x86_64              3.22.2-1.amzn2023.0.4                amazonlinux               16 k
 fontconfig                       x86_64              2.13.94-2.amzn2023.0.2               amazonlinux              273 k
 fontconfig-devel                 x86_64              2.13.94-2.amzn2023.0.2               amazonlinux              128 k
 fonts-filesystem                 noarch              1:2.0.5-12.amzn2023.0.2              amazonlinux              9.5 k
 freetype                         x86_64              2.13.2-5.amzn2023.0.1                amazonlinux              423 k
 freetype-devel                   x86_64              2.13.2-5.amzn2023.0.1                amazonlinux              912 k
 glib2-devel                      x86_64              2.74.7-689.amzn2023.0.2              amazonlinux              486 k
 google-noto-fonts-common         noarch              20201206-2.amzn2023.0.2              amazonlinux               15 k
 google-noto-sans-vf-fonts        noarch              20201206-2.amzn2023.0.2              amazonlinux              492 k
 graphite2                        x86_64              1.3.14-7.amzn2023.0.2                amazonlinux               97 k
 graphite2-devel                  x86_64              1.3.14-7.amzn2023.0.2                amazonlinux               21 k
 harfbuzz                         x86_64              7.0.0-2.amzn2023.0.1                 amazonlinux              868 k
```

```
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!
[ec2-user@ip-172-31-86-175 ~]$
```

Step 9: Creating Nagios User

Create a user called `nagios`, provide it with a home directory, and set a password:

```
sudo adduser -m nagios
sudo passwd nagios.
```



Step 10: Creating Nagcmd Group

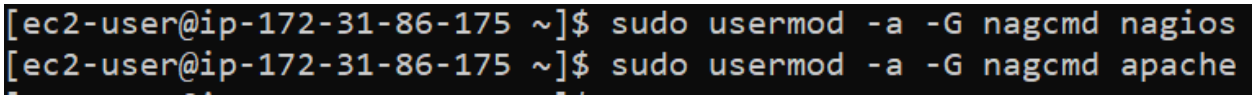
Create a group called `nagcmd` to manage Nagios command executions:

```
sudo groupadd nagcmd.
```

Step 11: Adding Users to Nagcmd Group

Add both `nagios` and `apache` users to the `nagcmd` group:

```
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache.
```



Name:Atharv Nikam Div :D15C Roll: 36

Step 12: Creating Download Directory

Create a directory to store Nagios files:
`mkdir ~/downloads`
Navigate to the directory using:
`cd ~/downloads`.

```
[ec2-user@ip-172-31-86-175 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-86-175 ~]$ cd ~/downloads
[ec2-user@ip-172-31-86-175 downloads]$
```

Step 13: Downloading Nagios Core and Plugins

Download the latest versions of Nagios Core and Plugins. If no newer versions are available, use:
`wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz`
`wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz`.

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-05 17:11:46--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: ‘nagios-4.5.5.tar.gz’

nagios-4.5.5.tar.gz             100%[=============================================================>]   1.97M  5.10MB/s    in 0.4s

2024-10-05 17:11:46 (5.10 MB/s) - ‘nagios-4.5.5.tar.gz’ saved [2065473/2065473]

[ec2-user@ip-172-31-86-175 downloads]$
```

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-05 17:12:19--  https://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|:443... 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  6.37MB/s    in 0.4s

2024-10-05 17:12:19 (6.37 MB/s) - ‘nagios-plugins-2.4.11.tar.gz’ saved [2753049/2753049]

[ec2-user@ip-172-31-86-175 downloads]$
```

Step 14: Extracting Nagios Core

Extract the Nagios Core files in the same directory using:
`tar zxvf nagios-4.5.5.tar.gz`.

```
[ec2-user@ip-172-31-86-175 downloads]$ tar zxvf nagios-4.5.5.tar.gz
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
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
```

```
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-86-175 downloads]$
```

Step 15: Configuring Nagios Core

```
[ec2-user@ip-172-31-86-175 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-86-175 downloads]$
```

If there's an error regarding SSL headers, install OpenSSL using:
`sudo yum install openssl-devel`.

```
[ec2-user@ip-172-31-86-175 downloads]$ sudo yum install openssl-devel
Last metadata expiration check: 1 day, 13:46:43 ago on Fri Oct  4 03:32:13 2024.
Dependencies resolved.
================================================================================
 Package                Architecture           Version                    Repos
================================================================================
Installing:
 openssl-devel          x86_64                 1:3.0.8-1.amzn2023.0.14    amazo

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

Total download size: 3.0 M
Installed size: 4.7 M
```

To ensure Nagios uses the `nagcmd` group for external commands, run:
`./configure --with-command-group=nagcmd`.

```
[ec2-user@ip-172-31-86-175 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
```

Name:Atharv Nikam Div :D15C Roll: 36

Step 16: Compiling and Installing Nagios

Compile Nagios by running:

```
make all
```

Install with:

```
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
```

```
sudo make install
```

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install
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'
```

`sudo make install-init`

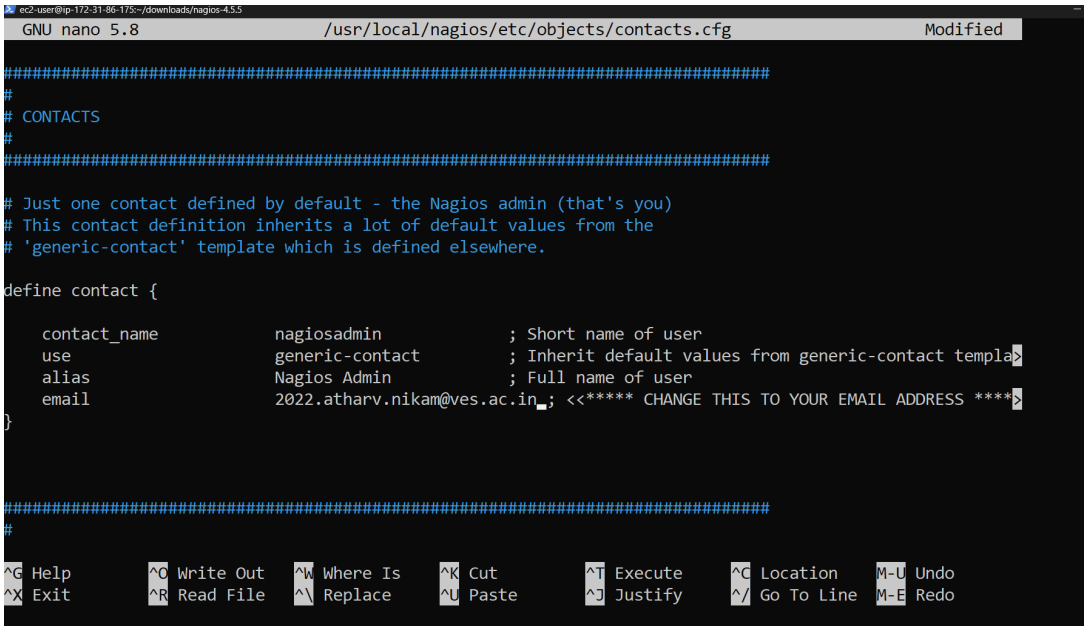
```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.c
fg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resour
ce.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/
nagios/etc/objects/templates.cfg
```

`sudo make install-commandmode`.

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-commandmode
/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 ***

[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 17: Configuring Email for Notifications

To configure notifications, open the `contacts.cfg` file and set your email address under `define contact{}` using:
`sudo nano /usr/local/nagios/etc/objects/contacts.cfg`.
Save your changes with CTRL+O and exit with CTRL+X.



Step 18: Installing Nagios Web Interface

Install the web interface configuration by running:
`sudo make install-webconf`.

```
[ec2-user@ip-172-31-86-175 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 ***
```

Step 19: Creating Web Interface User

Create a user called `nagiosadmin` to access the Nagios web interface:
`sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin`.

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadm
in
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 20: Restarting Apache Server

Restart the Apache server to apply the new configurations:
`sudo service httpd restart`.

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Name:Atharv Nikam Div :D15C Roll: 36

Step 21: Extracting Nagios Plugins

Extract the Nagios plugins in the `downloads` directory using:
`tar zxvf nagios-plugins-2.4.11.tar.gz`.

```
[ec2-user@ip-172-31-86-175 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/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-86-175 downloads]$
```

```
[ec2-user@ip-172-31-86-175 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/ltmain.sh
```

Step 22: Configuring Nagios Plugins

Navigate to the extracted folder and configure Nagios plugins with:
`./configure --with-nagios-user=nagios --with-nagios-group=nagios`.

```
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged


config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

```
gins/utils.o  -L. ../lib/libnagiosplug.a ../gl/libgnu.a -lresolv -lssl -lcrypto -lpthre
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/plugins-root
Making all in po
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 23: Compiling and Installing Plugins

Compile and install the plugins using:

```
make
sudo make install.
```

```
fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 24: Registering Nagios as a Service

To make Nagios manageable as a system service, run:

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagi
os.service.
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

```
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...
```

```
Checking misc settings...

Total Errors:   0

[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

`sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Check its status with:

`sudo systemctl status nagios`.

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sat 2024-10-05 19:18:16 UTC; 14s ago
       Docs: https://www.nagios.org/documentation
    Process: 68151 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=e>
    Process: 68152 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exit>
   Main PID: 68153 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.5M
        CPU: 79ms
     CGroup: /system.slice/nagios.service
             ├─68153 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─68154 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68155 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68156 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68157 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh

Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
```

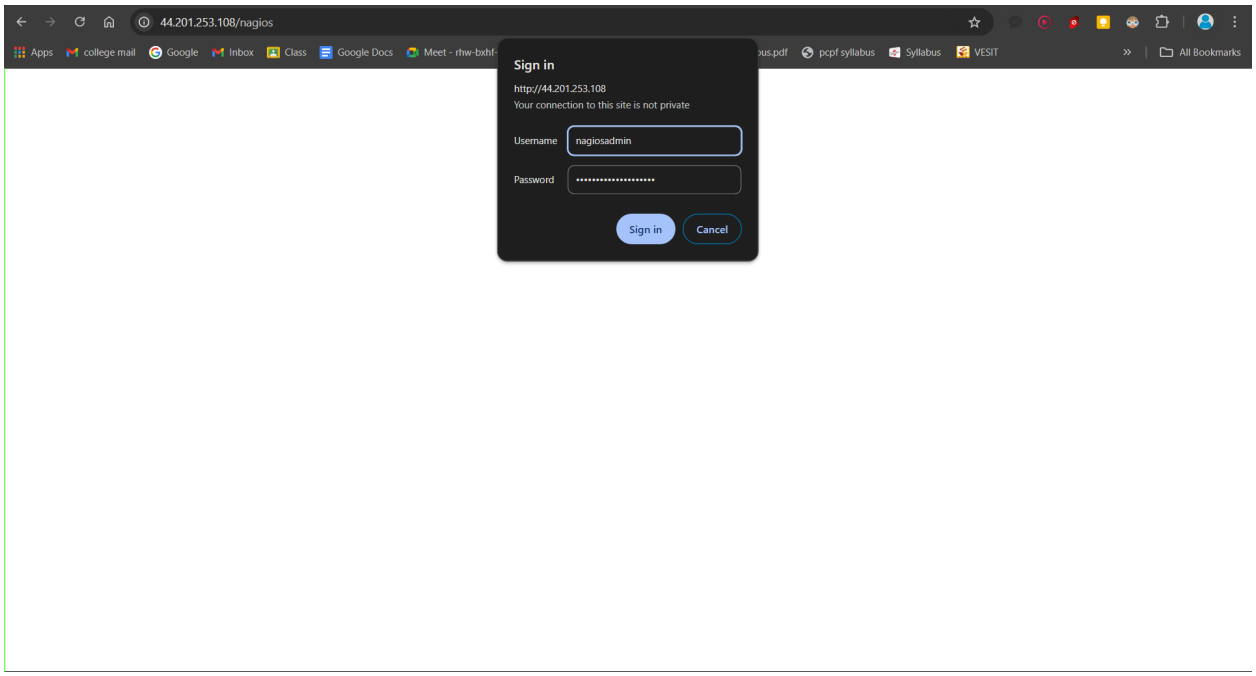
```
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: Successfully launched command file worker w>
lines 3-28/28 (END)
```

Step 27: Accessing Nagios Dashboard

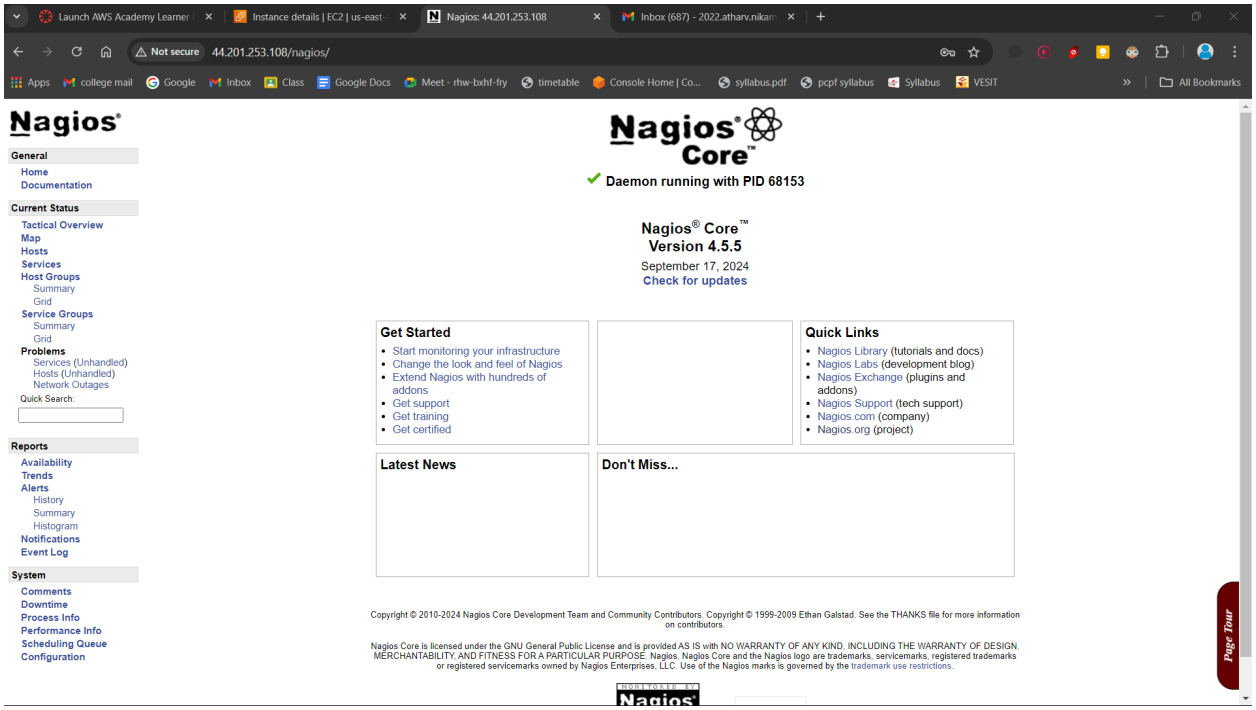
Copy the public IP address of your EC2 instance and type it in your browser, appending `/nagios` to access the dashboard: `http://<public-ip-address>/nagios`.

Username:nagiosadmin

password:(the password you have set)



Name:Atharv Nikam Div :D15C Roll: 36



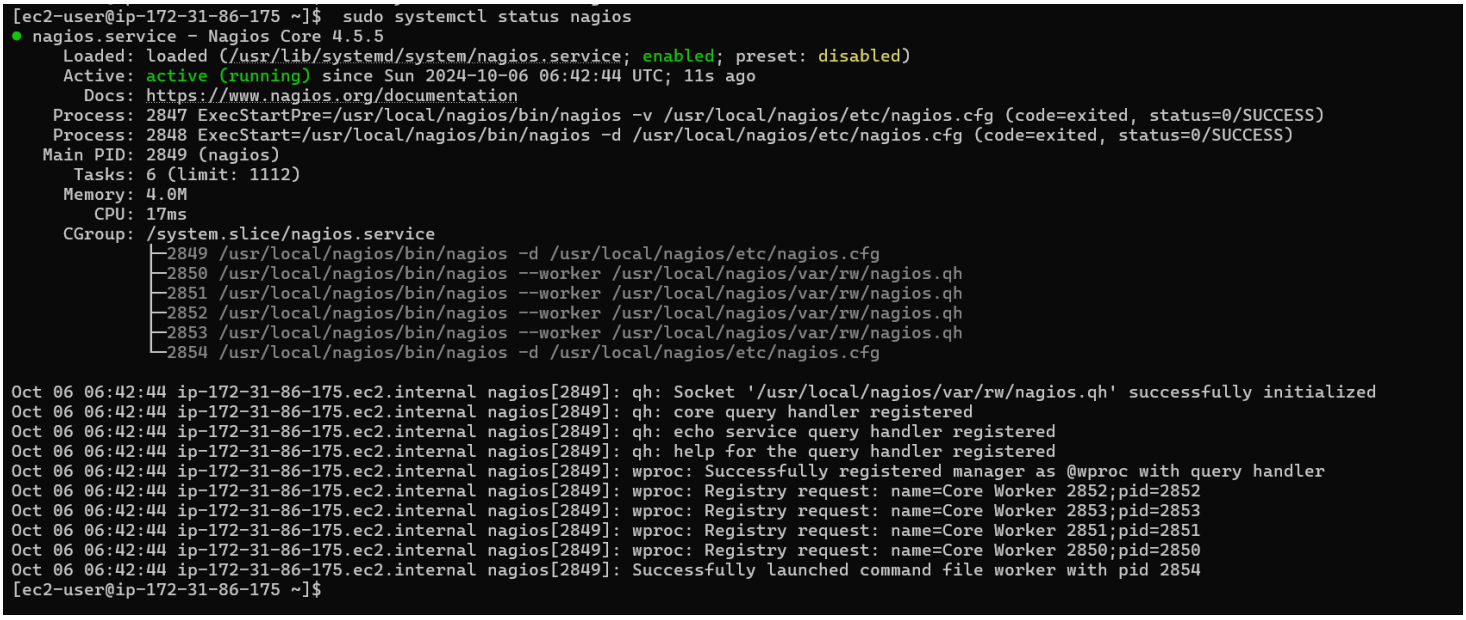
Conclusion:

In this experiment, I successfully explored the process of installing and configuring Nagios Core, Plugins, and NRPE on a Linux-based system. By carefully configuring the necessary security settings, installing dependencies, and resolving common issues like permission errors and missing directories, I ensured a smooth setup. This setup allows for efficient monitoring using Nagios, and I accessed the monitoring interface through the EC2 instance's public IP, demonstrating the effectiveness of continuous monitoring in a cloud environment.

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

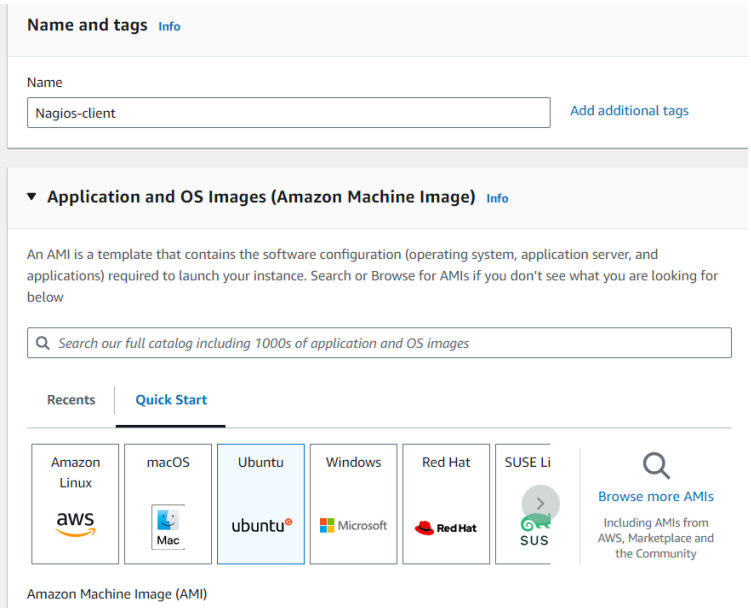
Step 1: Confirming Nagios on the Server

First, ensure that Nagios is running on your server by executing the following command on your Amazon Linux machine (Nagios-host):
`sudo systemctl status nagios`.



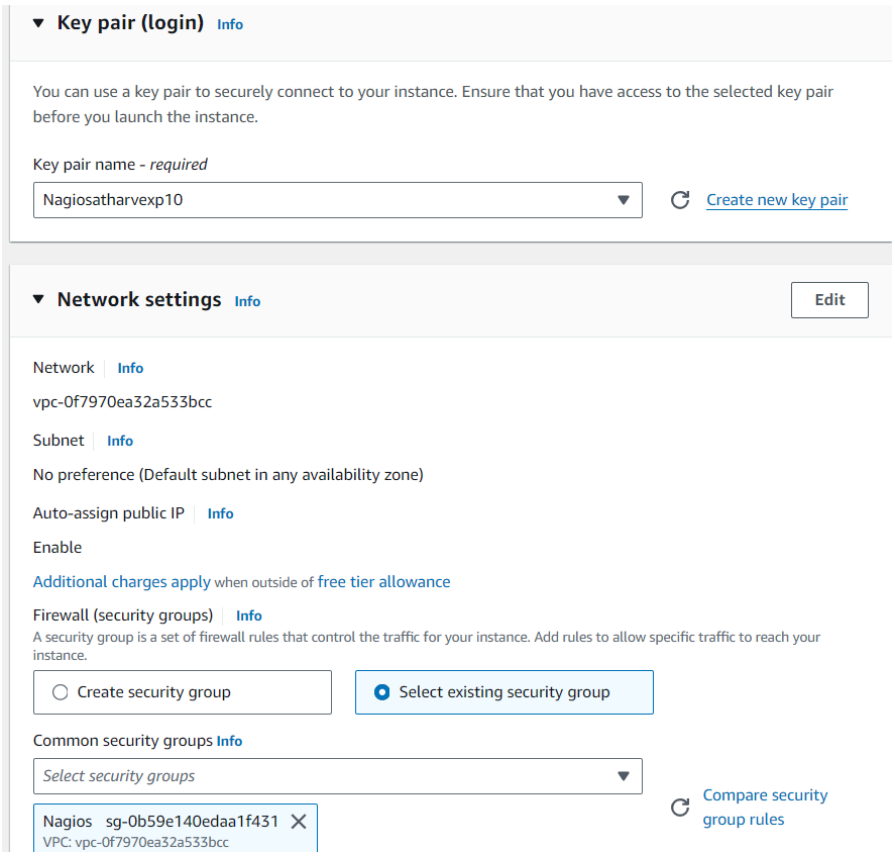
Step 2: Creating EC2 Instance

Next, create a new EC2 instance named **Nagios-client** with the Ubuntu AMI and **t2.micro** instance type. Generate an RSA key pair (.pem file), or use an existing one if available. Make sure to select the security group used in your previous Nagios-host setup.



Step 3: Connecting to the Instance

After creating the EC2 instance, connect to it. Navigate to the folder where the key (.pem) is stored on your local machine. Copy the provided SSH command from the instance's **SSH Client** section and paste it into your terminal.



Name:Atharv Nikam Div:D15C Roll:36

```
PS D:\Advancedevops key\exp10>  ssh -i "Nagiosatharvexp10.pem" ubuntu@ec2-3-80-172-58.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Oct  6 06:55:47 UTC 2024

  System load:  0.08              Processes:             106
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.46.49
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-49:~$
```

Step 4: Checking Nagios Status

To confirm Nagios is running correctly on the Nagios-host, execute the following: `ps -ef | grep nagios`.

```
[ec2-user@ip-172-31-86-175 ~]$ ps -ef | grep nagios
nagios      2849       1  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2850    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2851    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2852    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2853    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2854    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    3397    2384  0 06:56 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-86-175 ~]$
```

Step 5: Creating Root Directories

Switch to the root user and create necessary directories for monitoring hosts:
`sudo su`
`mkdir /usr/local/nagios/etc/objects/monitorhosts`
`mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts`.

Step 6: Configuring Monitoring for Linux Server

Copy the sample Nagios configuration file for localhost and create a new configuration file for the Linux server:

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

Step 7: Editing Linux Server Configuration

Open the `linuxserver.cfg` file and modify the hostname, IP address, and hostgroup as follows:

- **hostname:** linuxserver
- **address:** Public IP of the Linux client
- **hostgroup_name:** linux-servers1
  Use the command:
  `nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg`.

```
define host {

    use                     linux-server            ; Name of host template to use
                                                    ; This host definition will inherit all variables that are defined
                                                    ; in (or inherited by) the linux-server host template definition.
    host_name               linuxserver
    alias                   localhost
    address                 172.31.46.49
}



###############################################################################
#
# HOST GROUP DEFINITION
#
###############################################################################

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name          linux-servers1          ; The name of the hostgroup
    alias                   Linux Servers           ; Long name of the group
    members                 localhost               ; Comma separated list of hosts that belong to this group
}
```

Step 8: Updating Nagios Configuration

Add the following line to Nagios' main configuration file to include the monitoring hosts directory:
`cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/`.
Edit the file using:
`nano /usr/local/nagios/etc/nagios.cfg`.

```
#
# Read the documentation for more information on this configuration

# file.  I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
##############################################################################


# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes.  This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log



# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Step 9: Verifying Nagios Configuration

To check for any syntax errors in your configuration, run:
`/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`.

```
Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 2 hosts.
        Checked 2 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 2 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

Things look okay - No serious problems were detected during the pre-flight check
```

Step 10: Restarting Nagios

Restart the Nagios service to apply the configuration changes:
`sudo service nagios restart`.

```
[root@ip-172-31-86-175 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-86-175 ec2-user]#
```

Step 11: Installing NRPE on Nagios Client

Connect to the Nagios-client instance and update the system. Then install NRPE and necessary Nagios plugins using:

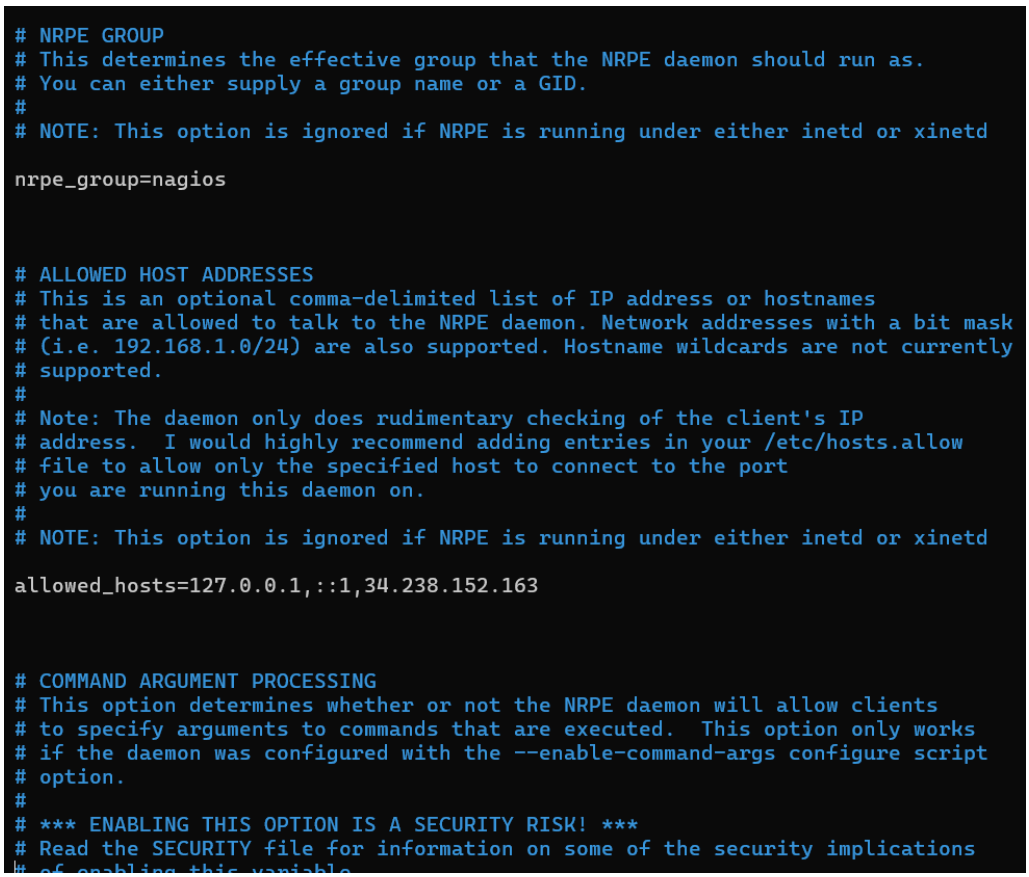
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-46-49:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:24 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
```

```
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-172-31-46-49:~$
```

Step 12: Configuring NRPE

Edit the NRPE configuration file to allow the Nagios-host to communicate with the client. Add the Nagios-host's IP address under `allowed_hosts`: `sudo nano /etc/nagios/nrpe.cfg`.



```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_group=nagios



# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,::1,34.238.152.163



# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed.  This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable
```

Step 13: Restarting NRPE

After editing the NRPE configuration, restart the NRPE server: `sudo systemctl restart nagios-nrpe-server`.

```
ubuntu@ip-172-31-46-49:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-46-49:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-46-49:~$
```

Step 14: Checking Nagios and HTTPD Services

On the Nagios-host, check the status of Nagios and ensure that the HTTPD service is active:

`sudo systemctl status nagios`

`sudo systemctl status httpd`.

```
[root@ip-172-31-86-175 ec2-user]# sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-86-175 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-86-175 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-86-175 ec2-user]#
```

If HTTPD is not active, start and enable it:

`sudo systemctl start httpd`

`sudo systemctl enable httpd`.

Step 15: Accessing Nagios Dashboard

To view the Nagios dashboard, open your browser and go to:

`http://<Nagios-host-ip>/nagios`.

Click on **Hosts** from the left panel to view the status of your Linux server.

Name:Atharv Nikam Div:D15C Roll:36

Nagios Core

✔ Daemon running with PID 3969

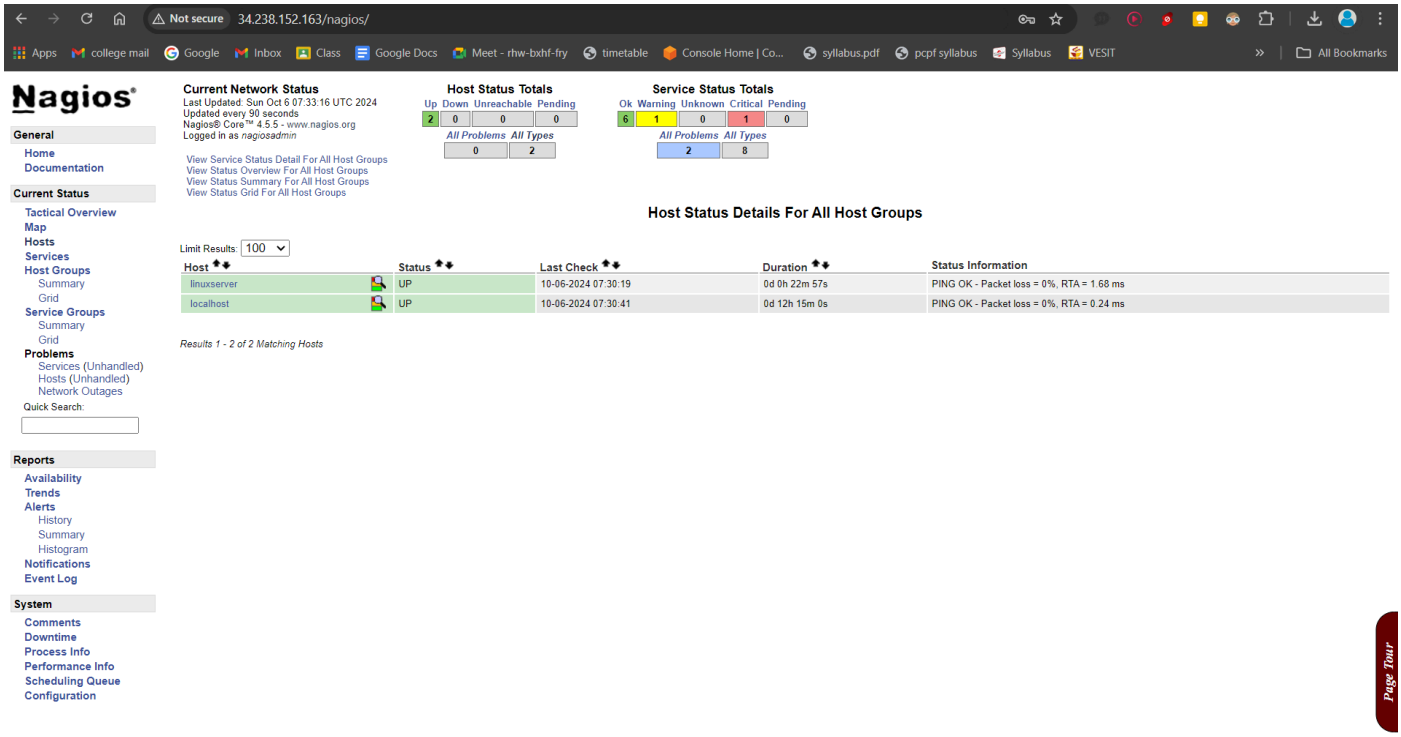
Nagios® Core™
Version 4.5.5
September 17, 2024
Check for updates

Get Started
- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links
- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...



Page Tour

Name:Atharv Nikam Div:D15C Roll:36

Host Information
Last Updated: Sun Oct 6 07:34:10 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as *nagiosadmin*

View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications For This Host

Host
localhost
(linuxserver)

Member of
No hostgroups

172.31.46.49

Host State Information

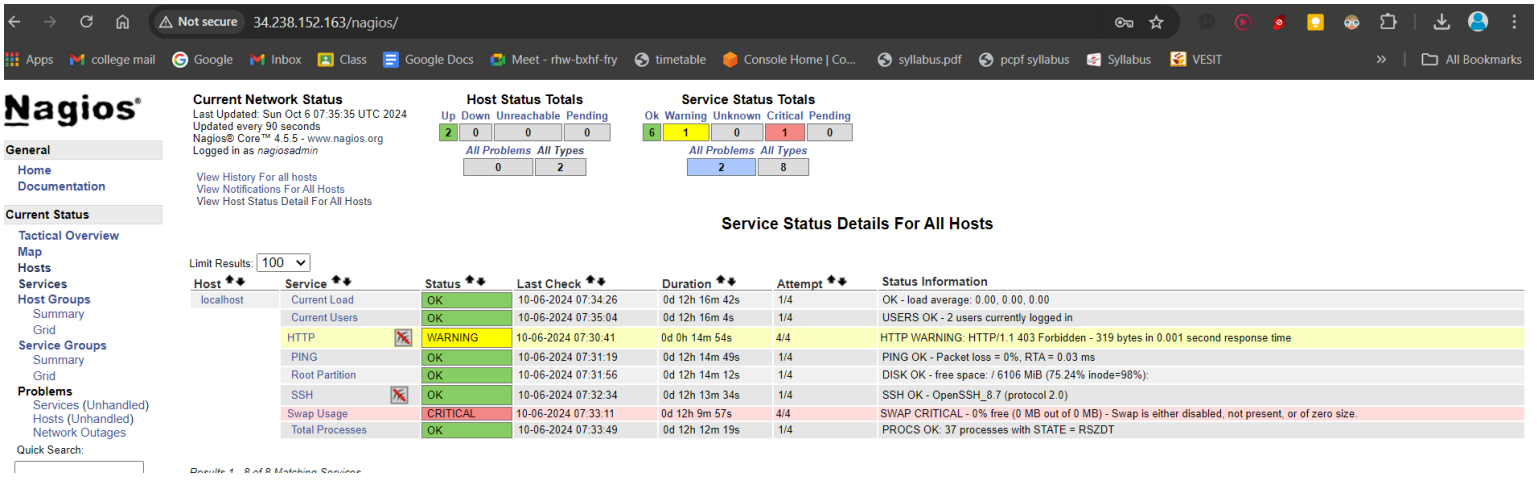
Host Status:	UP (for 0d 0h 23m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.68 ms
Performance Data:	rta=1.679000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-06-2024 07:30:19
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.011 seconds
Next Scheduled Active Check:	10-06-2024 07:35:19
Last State Change:	10-06-2024 07:10:19
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-06-2024 07:34:09 (0d 0h 0m 1s ago)

Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

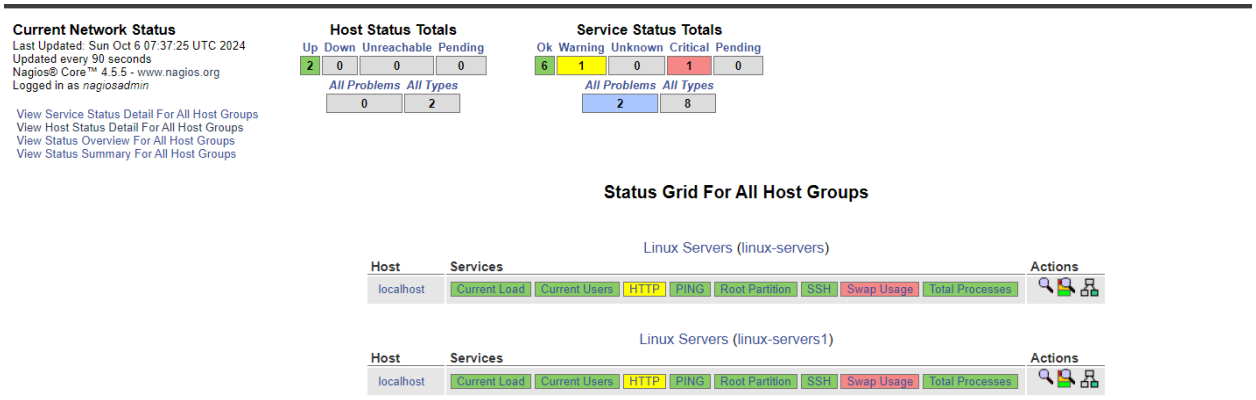
Host Commands
- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

Host Comments
Add a new comment Delete all comments

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it							



Name:Atharv Nikam Div:D15C Roll:36



Conclusion:

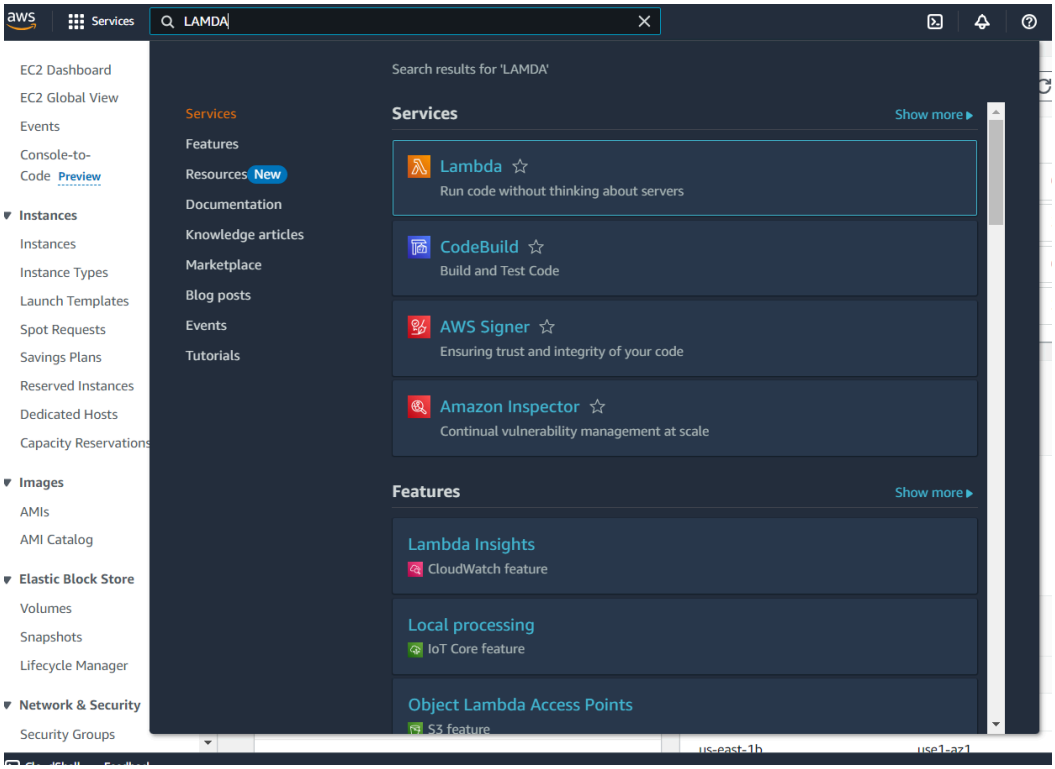
This experiment was designed to set up monitoring for ports, services, and a Linux server using Nagios. By carefully configuring both the Nagios host and client, we were able to monitor essential network services and assess server performance. The experiment demonstrated how Nagios can be used effectively to track system metrics such as CPU usage and memory consumption. This hands-on experience highlights the importance of proactive monitoring in maintaining server health and ensuring the availability of critical services across Linux and Windows platforms.

Name:Atharv Nikam **Std:D15C** **ROLL:36**

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

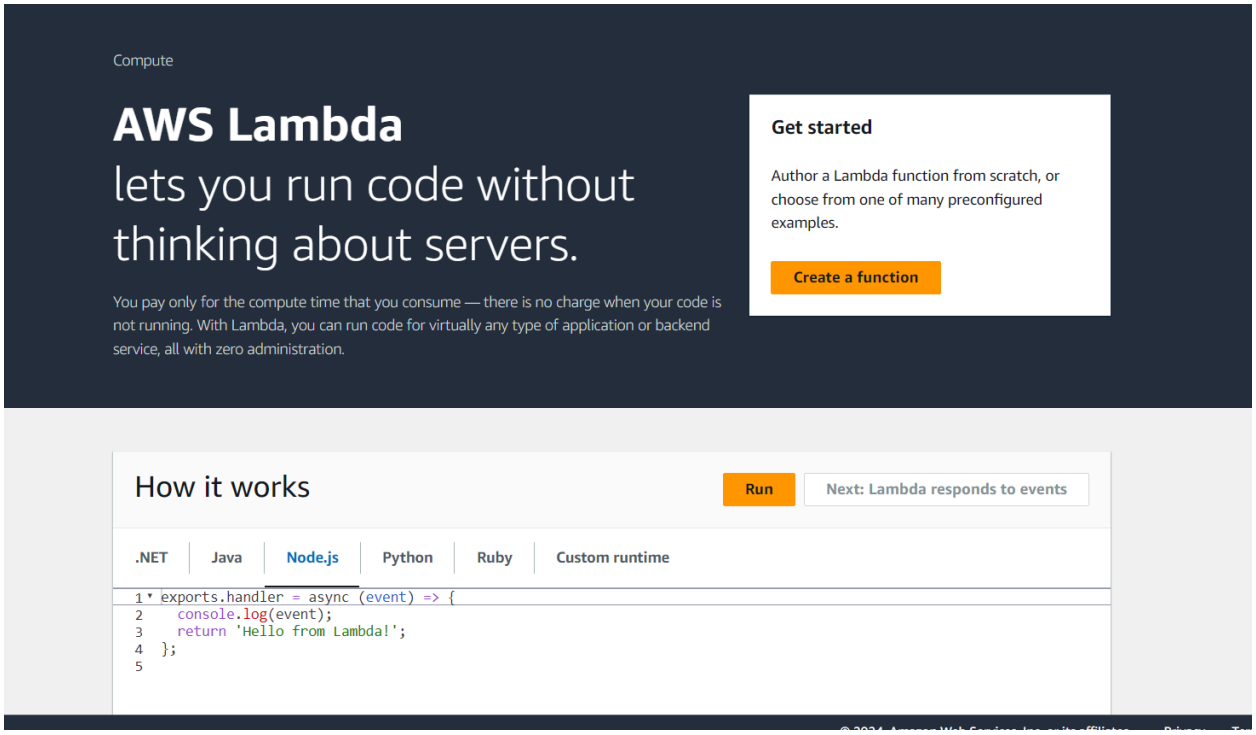
Step 1: Accessing AWS

Log in to your AWS Personal/Academy account. Navigate to the Lambda service by searching for "Lambda" in the AWS Management Console.



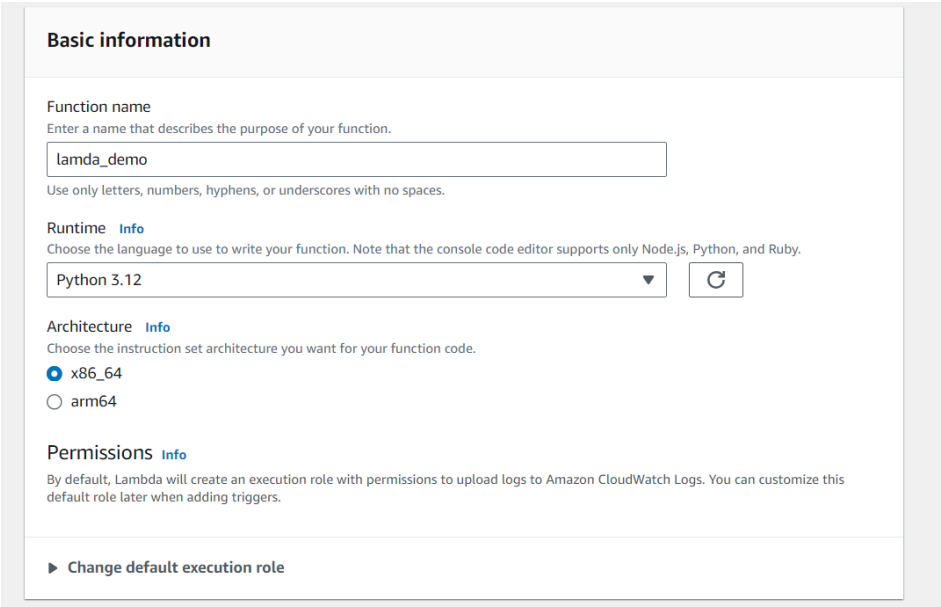
Step 2: Creating a New Lambda Function

Click on the "Create function" button. Provide a name for your Lambda function and select the language you wish to use, such as Python 3.12. For architecture, choose x86, and for execution role, opt to create a new role with basic Lambda g permissions.



Step 3: Configuring Basic Settings

To modify the basic settings, navigate to the "Configuration" tab and click on "Edit" under General Settings. Here, you can add a description and adjust the memory and timeout settings. For this experiment, I set the timeout to 1 second, which is sufficient for testing.



Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

- (●) Create a new role with basic Lambda permissions
- ( ) Use an existing role
- ( ) Create a new role from AWS policy templates

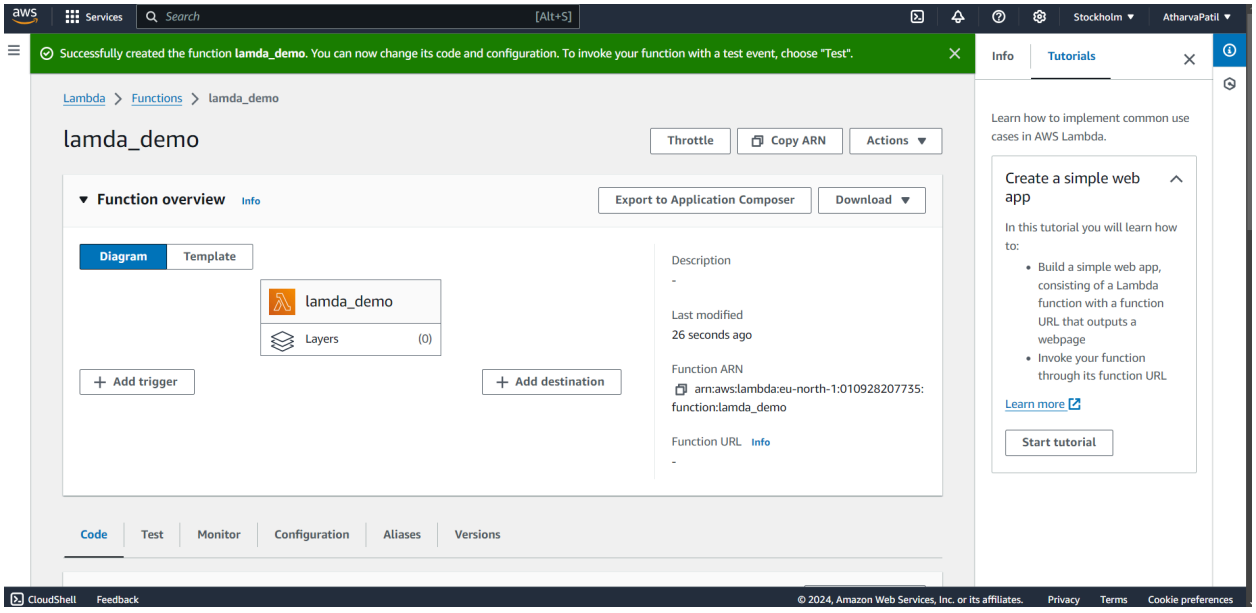
ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

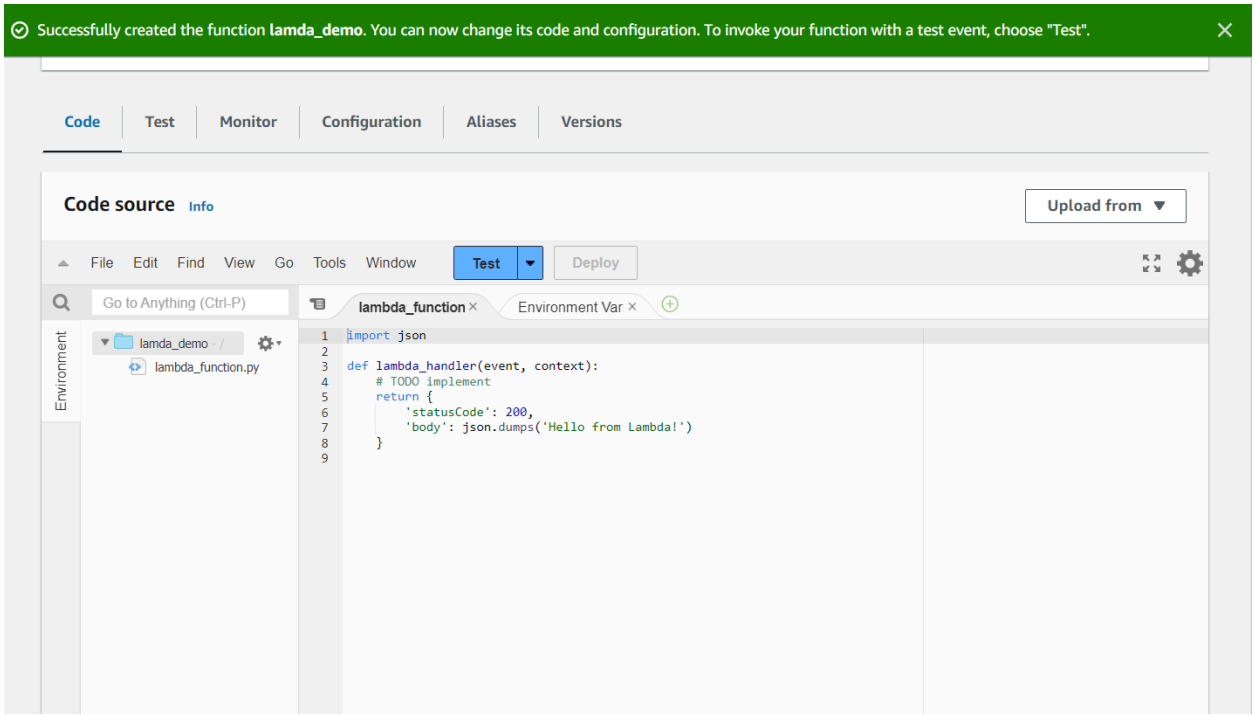
Lambda will create an execution role named ATHARV_LAMDA-role-0u7c9ooi, with permission to upload logs to Amazon CloudWatch Logs.

▶ **Additional Configurations**

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

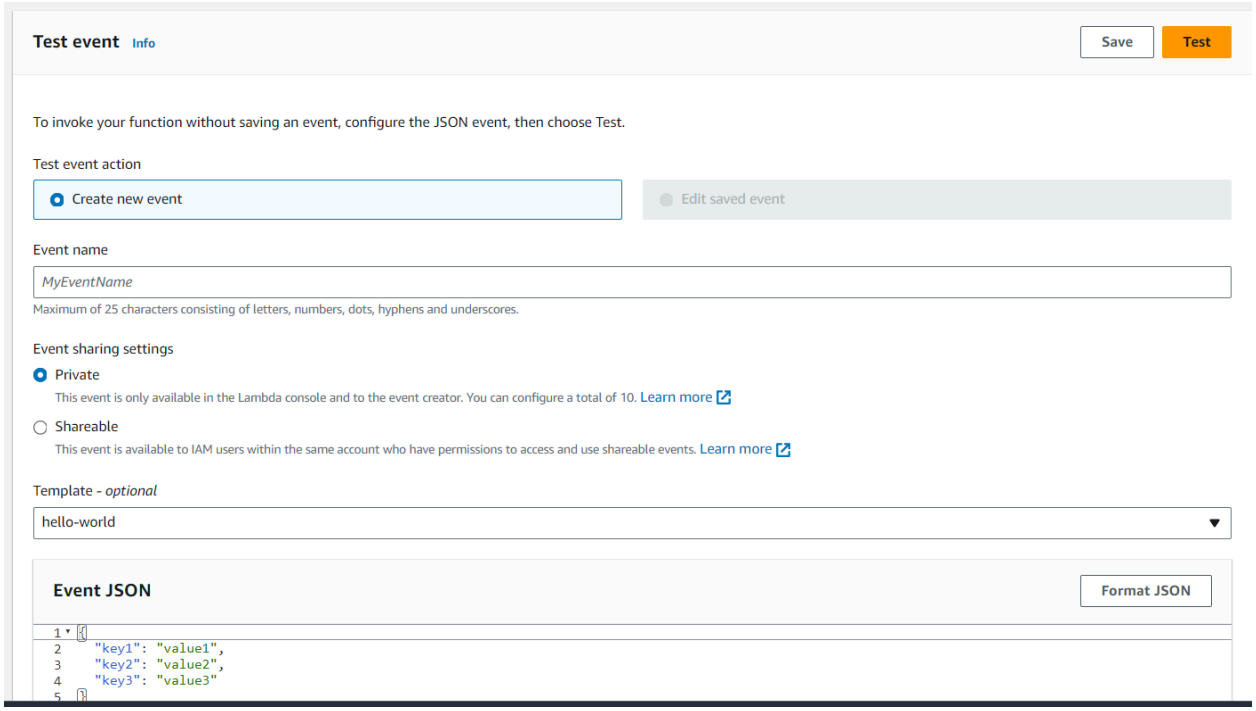
Cancel | **Create function**

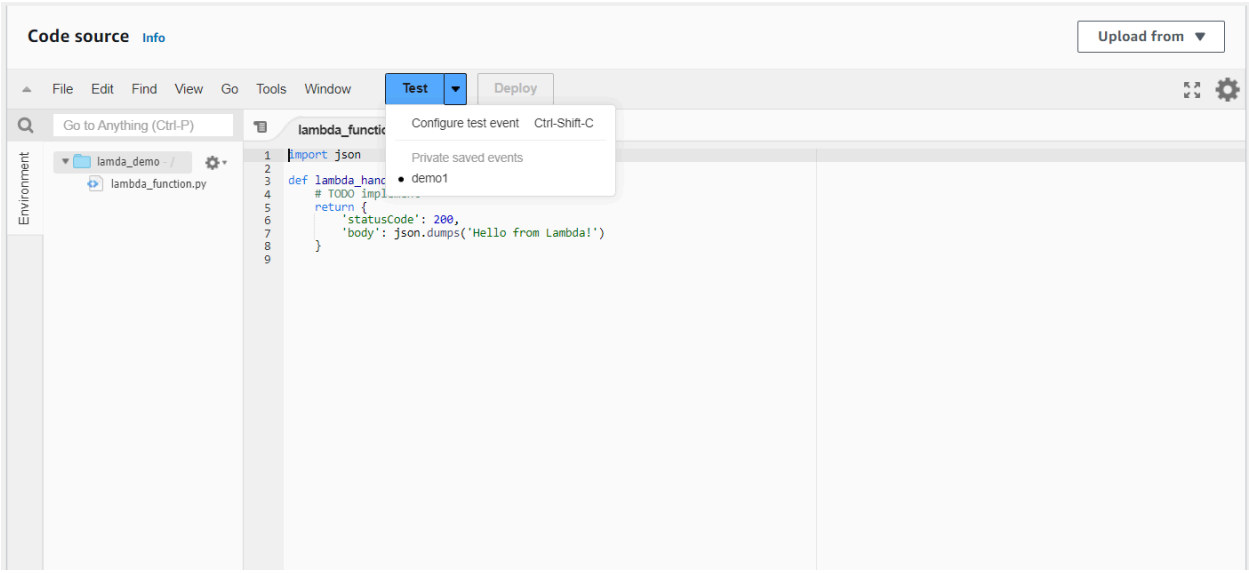


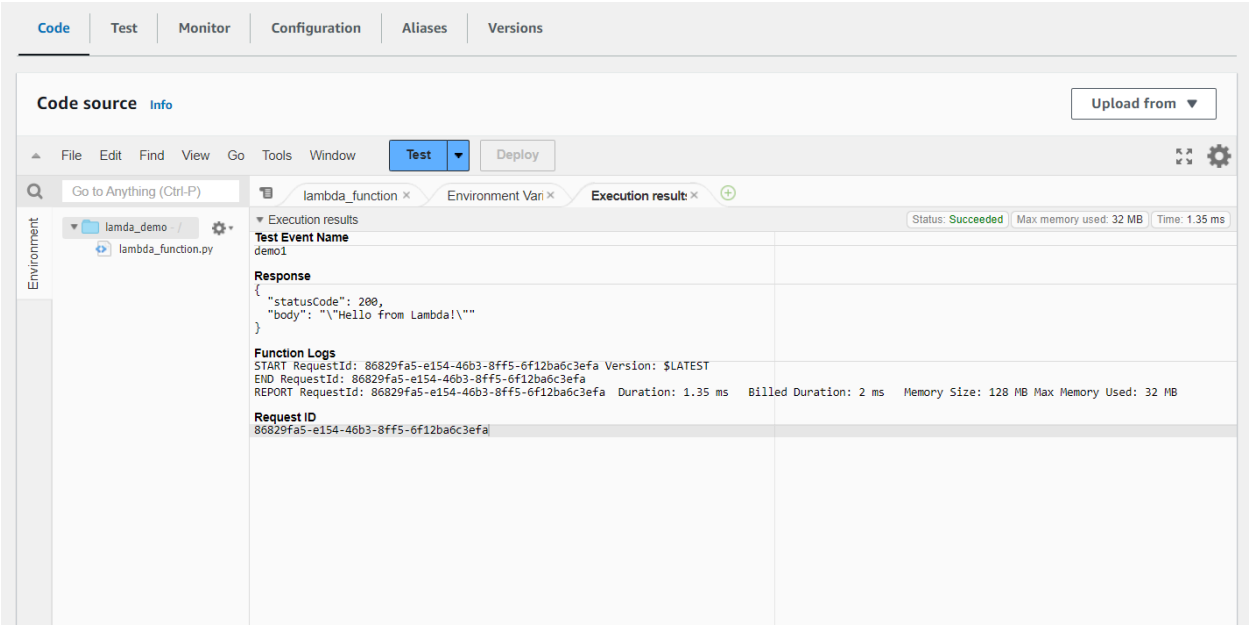


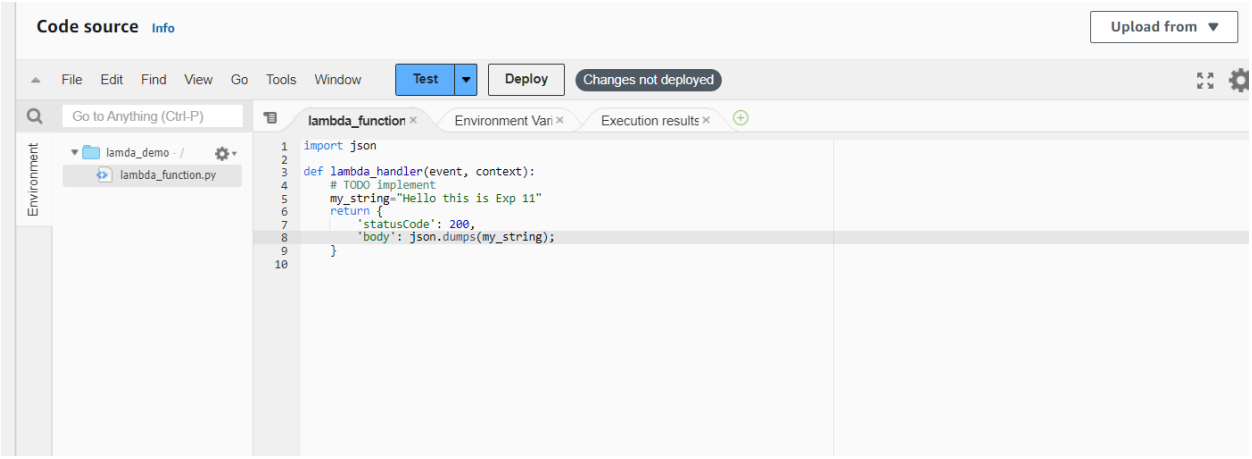
Step 4: Testing the Function

Click on the "Test" tab and select "Create a new event." Name your event, set the event sharing to private, and choose the "hello-world" template.



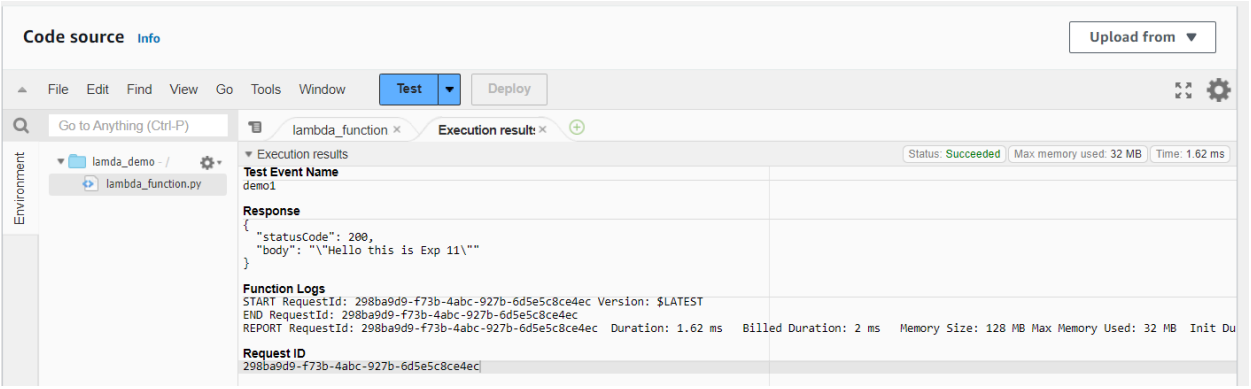

Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Configure test event Ctrl-Shift-C
Private saved events
demo1
Go to Anything (Ctrl-P)
lamda_demo
lambda_function.py
Environment
import json
def lambda_hand
# TODO impl
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')
}


Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lamda_demo
lambda_function.py
Environment
lambda_function
Environment Vari
Execution result:
Execution results
Status: Succeeded
Max memory used: 32 MB
Time: 1.35 ms
Test Event Name
demo1
Response
{
"statusCode": 200,
"body": "\"Hello from Lambda!\""
}
Function Logs
START RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Version: $LATEST
END RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa
REPORT RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Duration: 1.35 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
Request ID
86829fa5-e154-46b3-8ff5-6f12ba6c3efa



Step 5: Running the Test

In the Code section, select the newly created event from the dropdown menu and click on "Test." You should see the output displayed below.

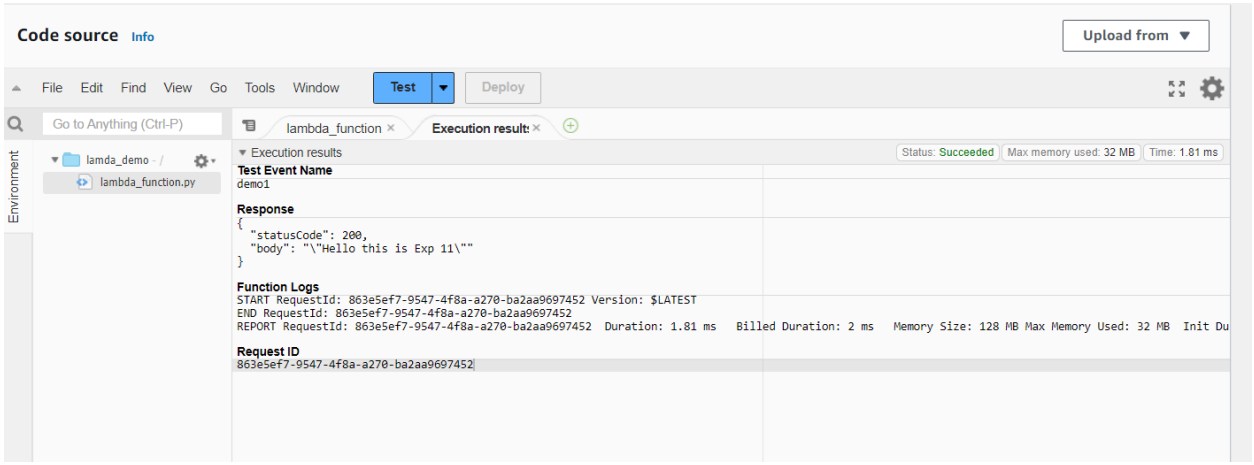


Step 6: Editing and Deploying the Code

You can modify your Lambda function's code as needed. I updated the code to display a new string. After making changes, press `Ctrl + S` to save and then click on "Deploy" to apply the updates.

Step 7: Final Testing

Return to the "Test" tab and execute the test again to observe the output. You should see a status code of 200 along with your string output and function logs confirming a successful deployment.



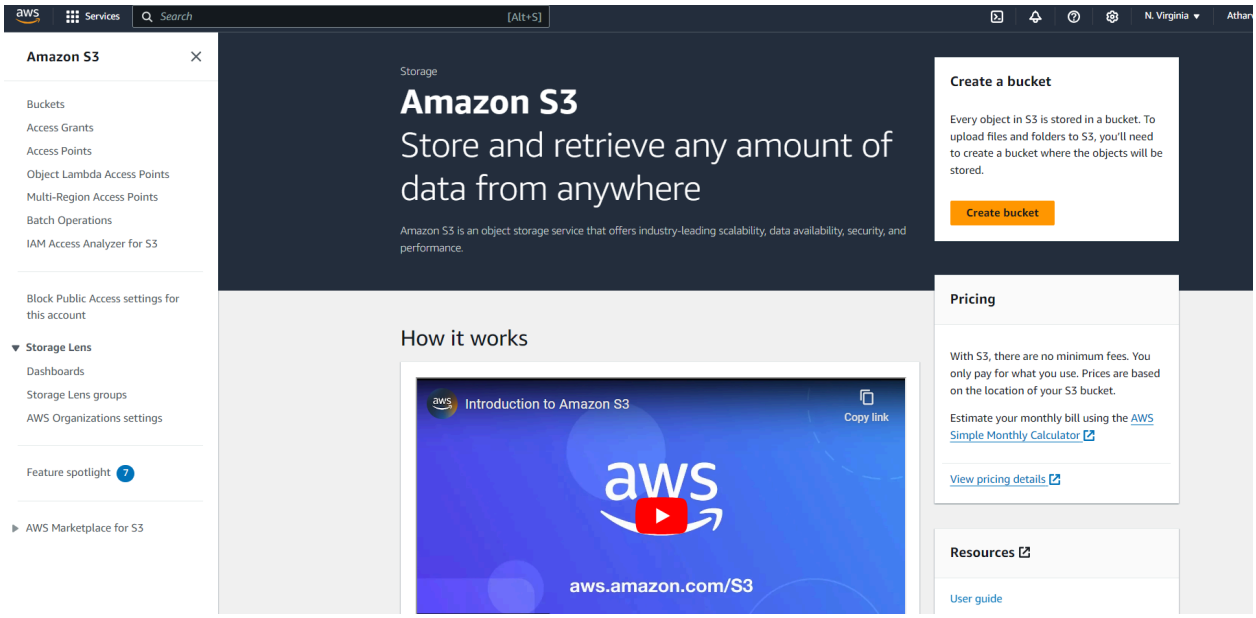
Conclusion

In this experiment, I successfully navigated the process of creating an AWS Lambda function. After configuring the function with Python, I adjusted the settings to optimize its performance. I created a test event, deployed the function, and verified the output, which confirmed the expected behavior. This hands-on experience highlighted the user-friendly nature of AWS Lambda, illustrating how it enables developers to focus on coding while AWS efficiently handles the underlying infrastructure and scaling. This project not only deepened my understanding of serverless computing but also reinforced the practical application of cloud services in modern software development.

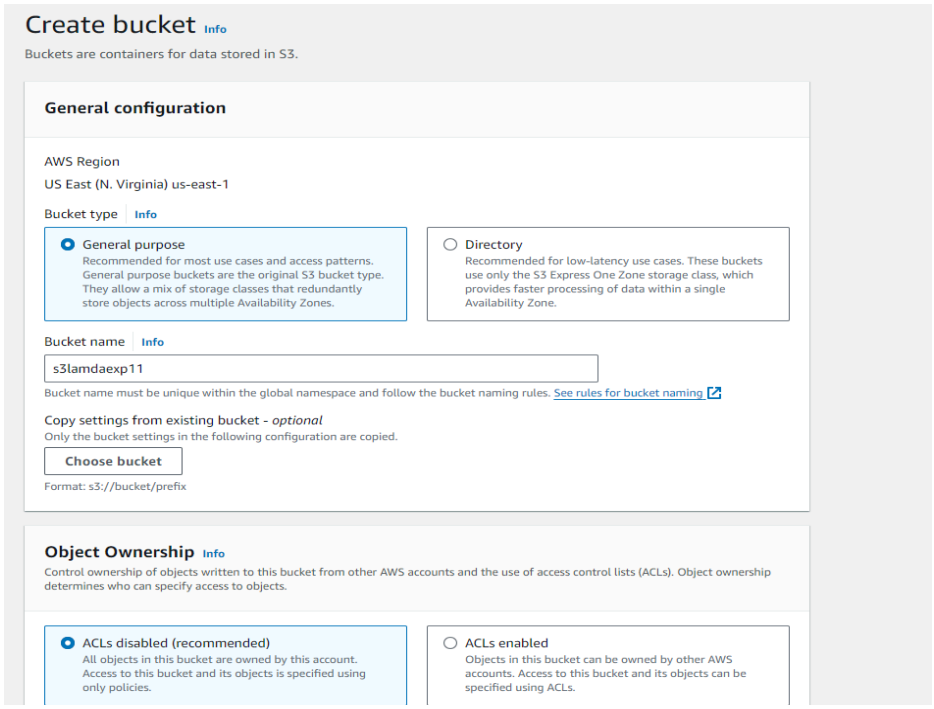
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Step 1: Create a s3 bucket.

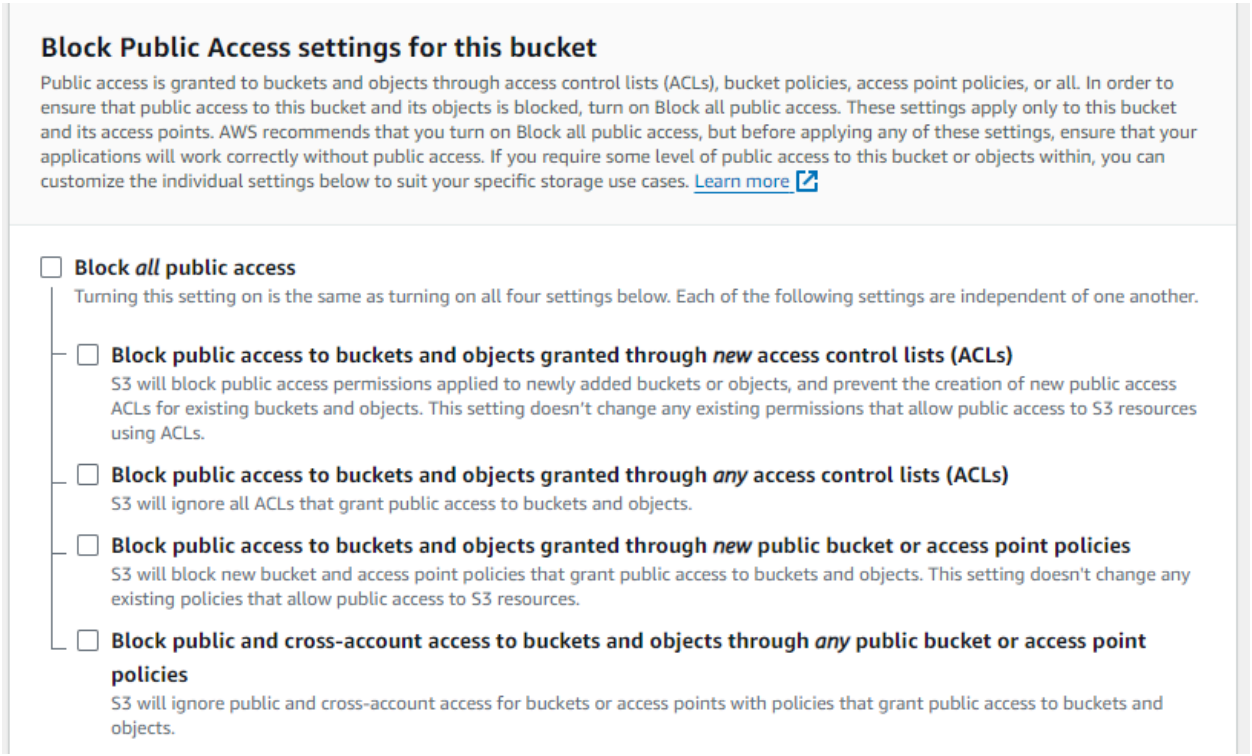
1) Search for S3 bucket in the services search. Then click on create bucket.



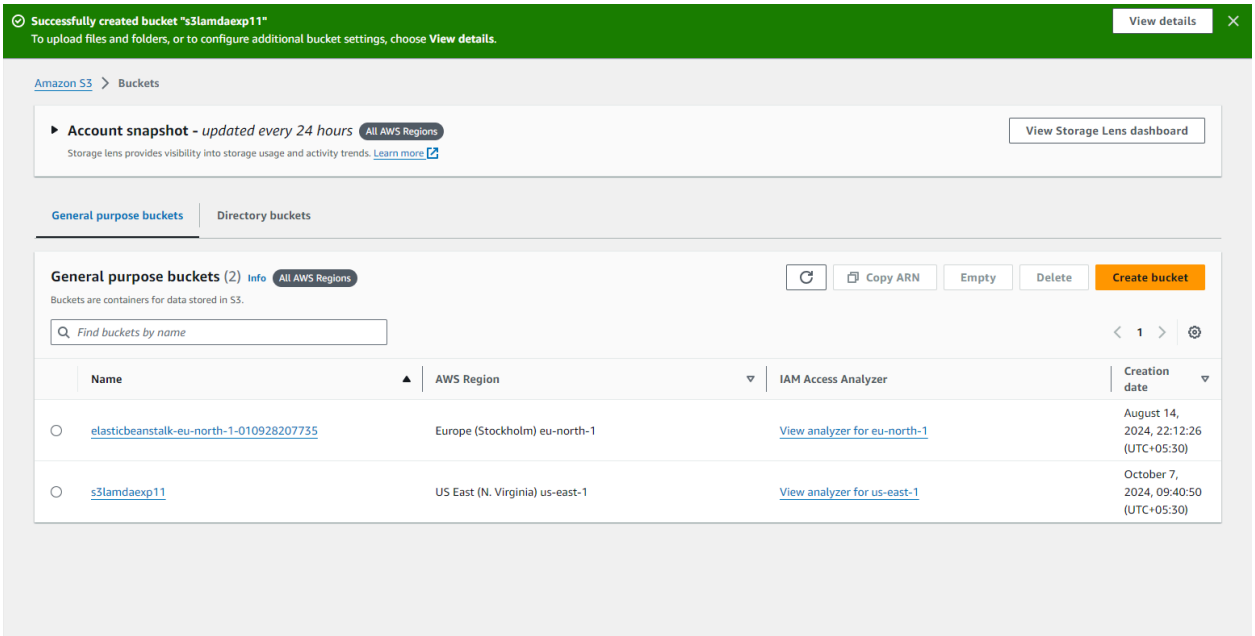
2) Keep the bucket as a general purpose bucket. Give a name to your bucket.



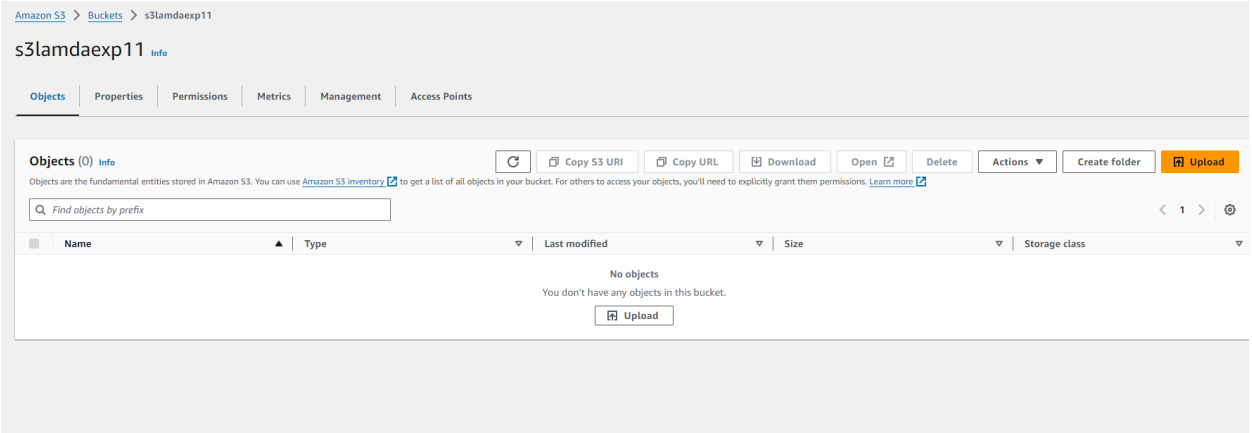
3) Uncheck block all public access

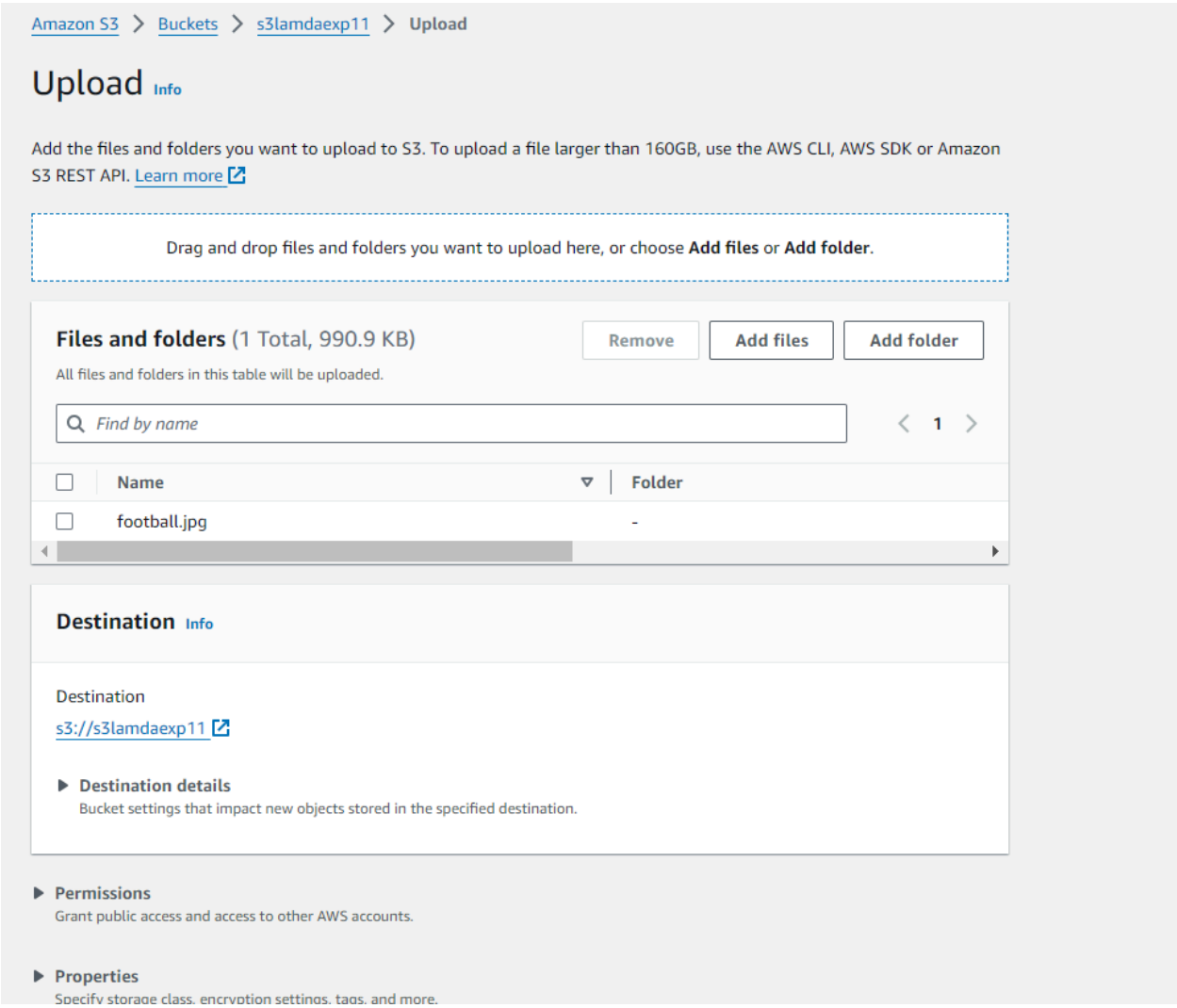


4) Keeping all other options same, click on create. This would create your bucket. Now click on the name of the bucket

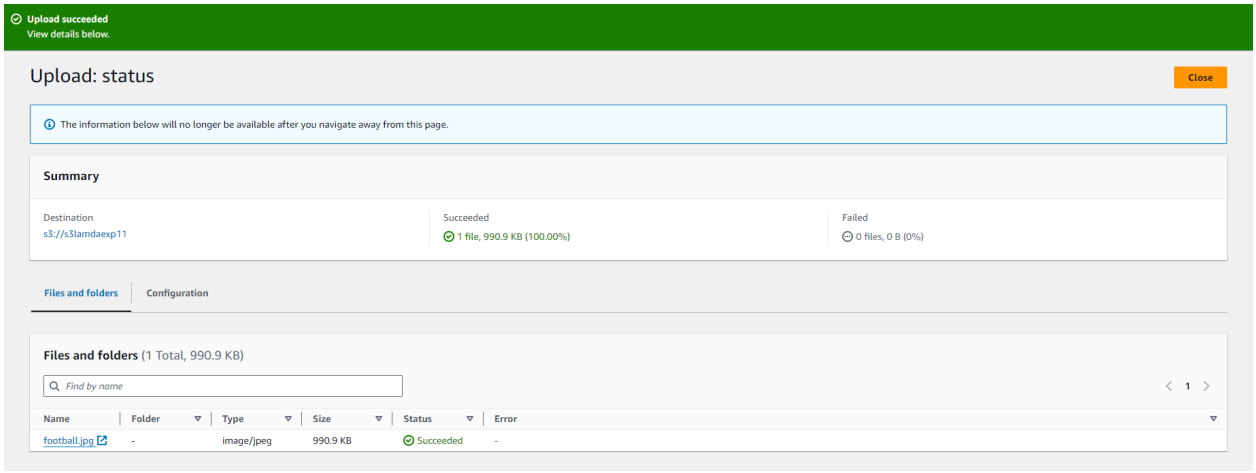


5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload



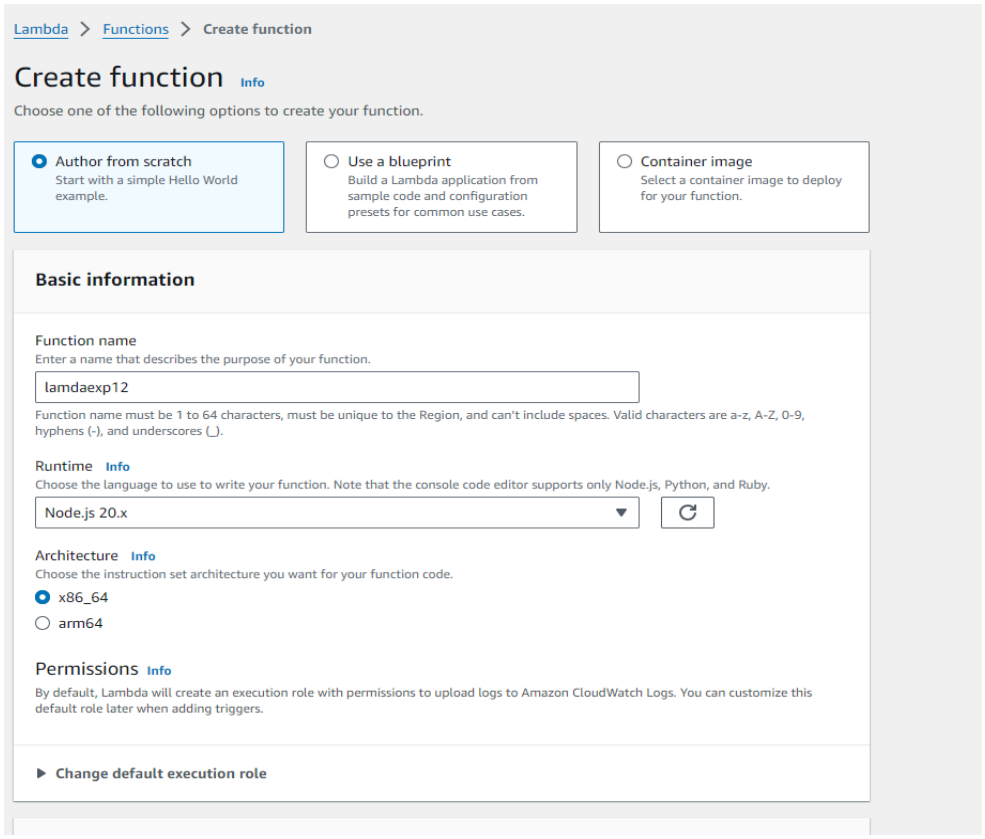


6) The image has been uploaded to the bucket

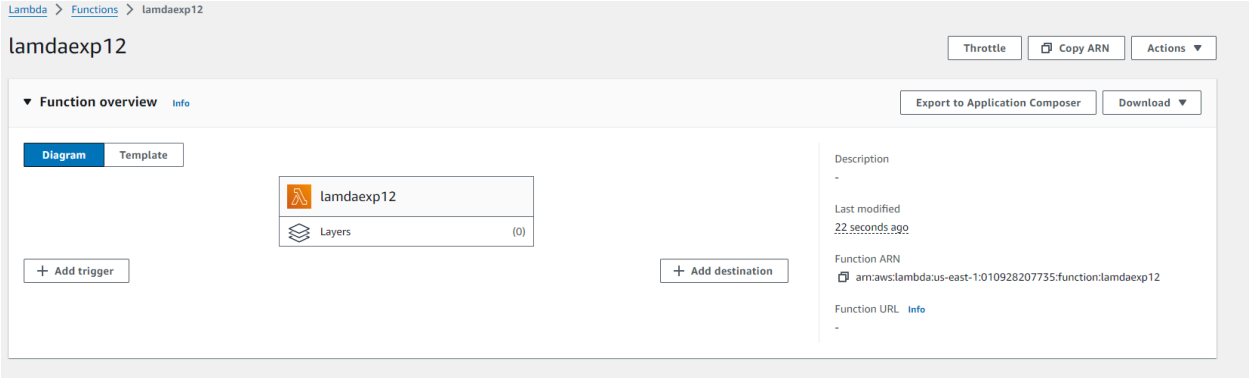


Step 2: Configure Lambda function

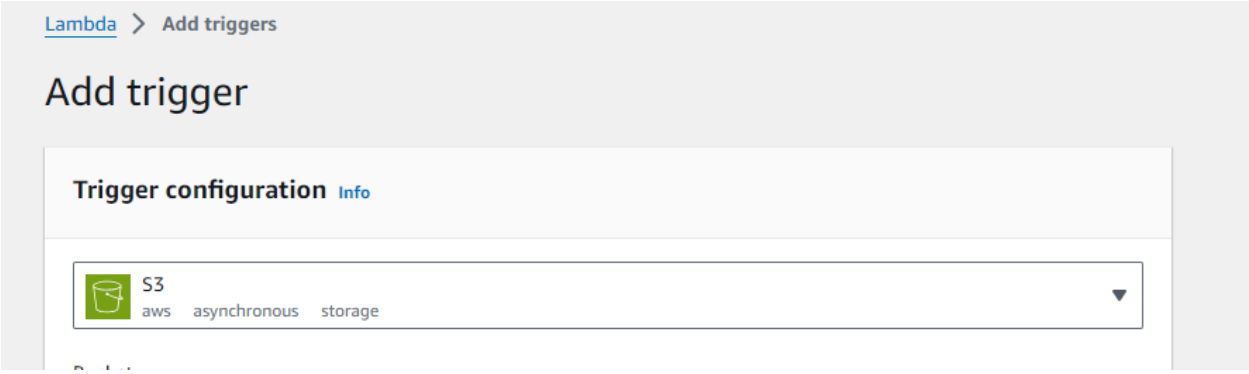
1) Go to the lambda function you had created berfor. (Services → Lambda → Click on name of function). Here, click on add trigger

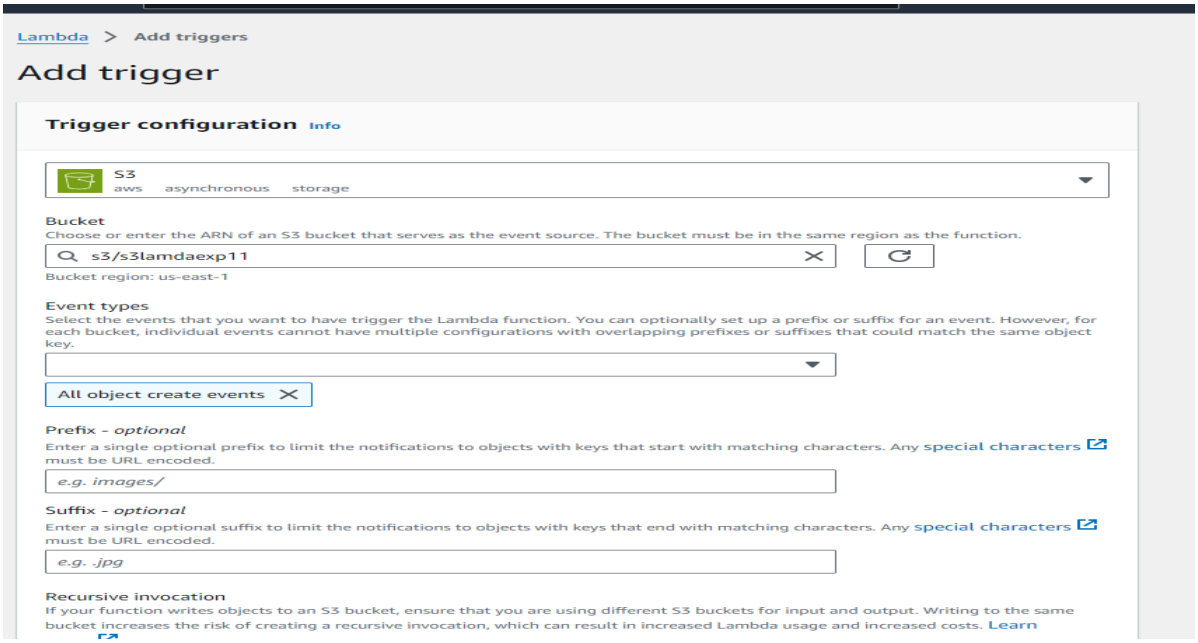


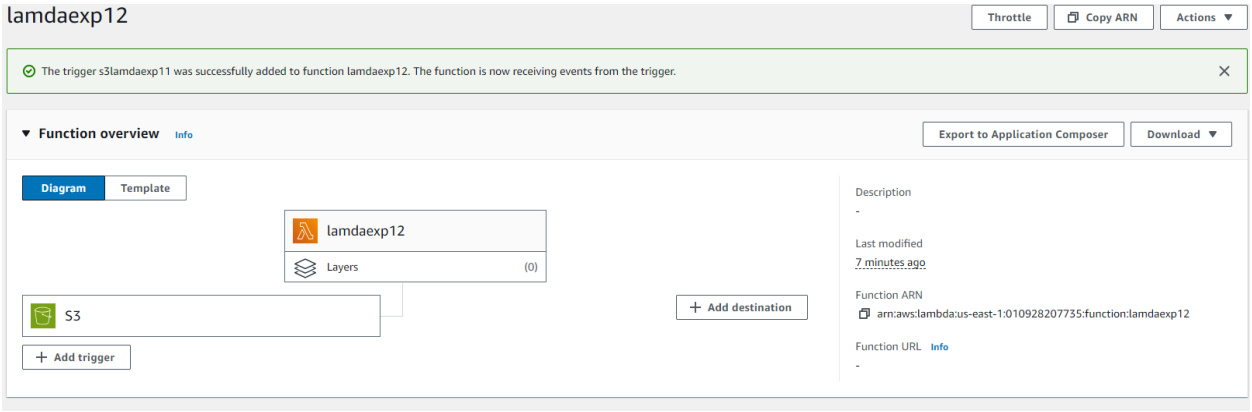
2) Under trigger configuration, search for S3 and select it.



3) Here, select teh S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function



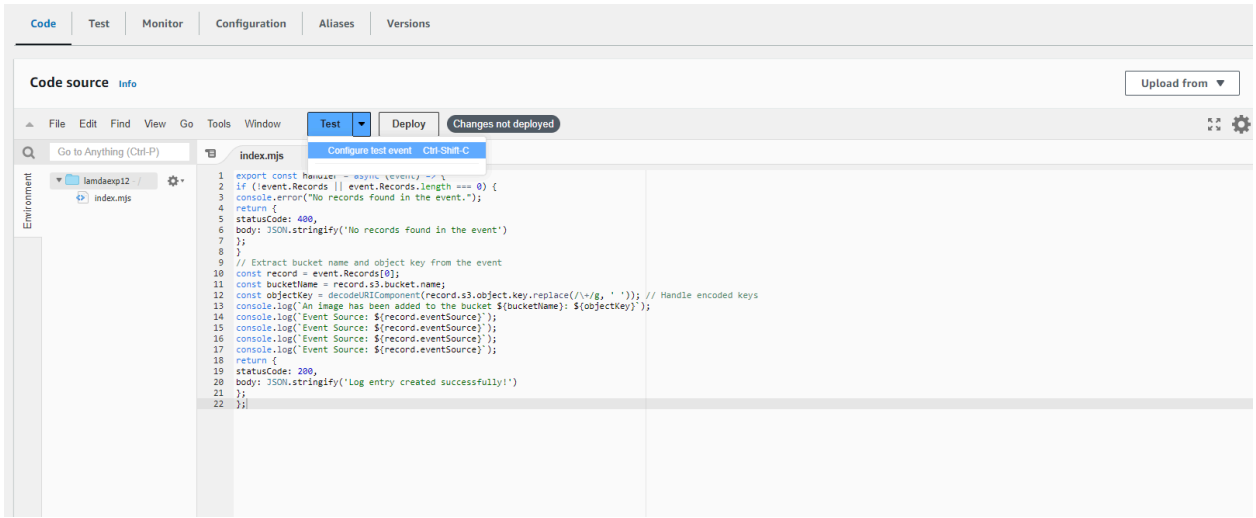




4) Scroll down to the code section of the function. Add the following javascript code to the code area by replacing the existing code

```
export const handler = async (event) => {
if (!event.Records || event.Records.length === 0) {
console.error("No records found in the event.");
return {
statusCode: 400,
body: JSON.stringify('No records found in the event')
};
}
// Extract bucket name and object key from the event
const record = event.Records[0];
const bucketName = record.s3.bucket.name;
const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
return {
statusCode: 200,
body: JSON.stringify('Log entry created successfully!')
};
};
```

This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.



Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world

Event JSON Format JSON

```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-bucket"
        },
        "object": {
          "key": "test%2Fkey",
```

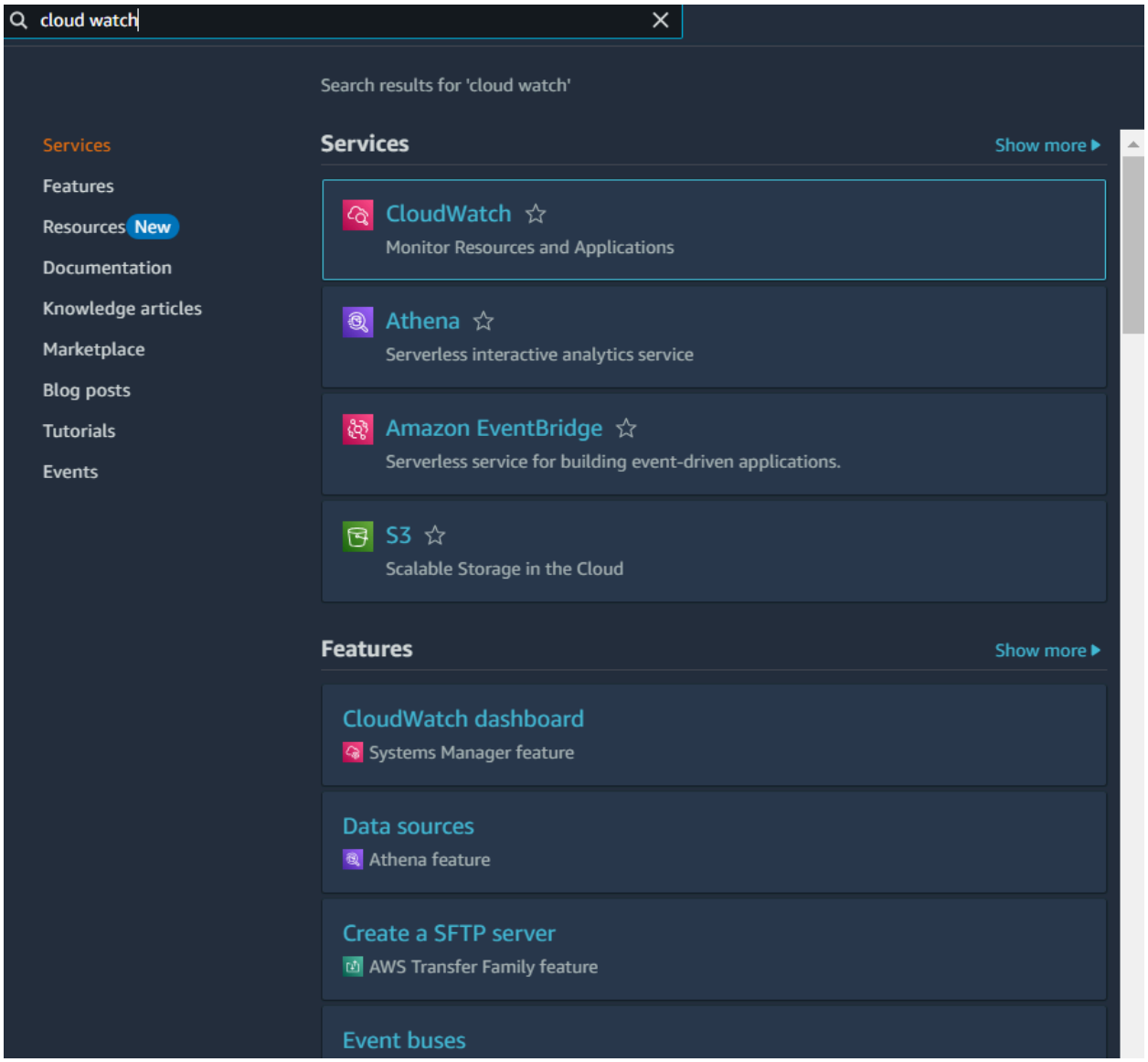
1:1 JSON Spaces: 2

Cancel Invoke Save

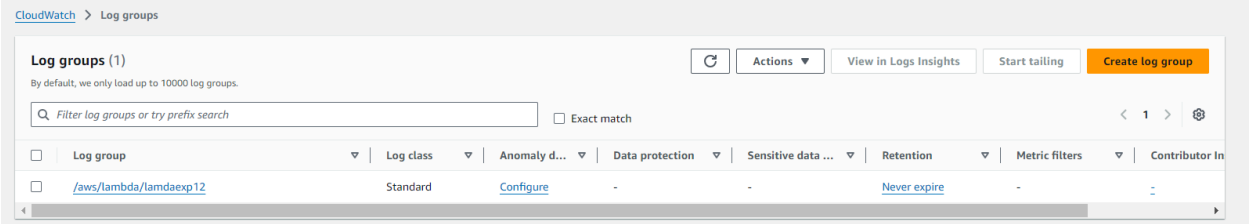
The test event **myevent1** was successfully saved.

Step 3: Check the logs

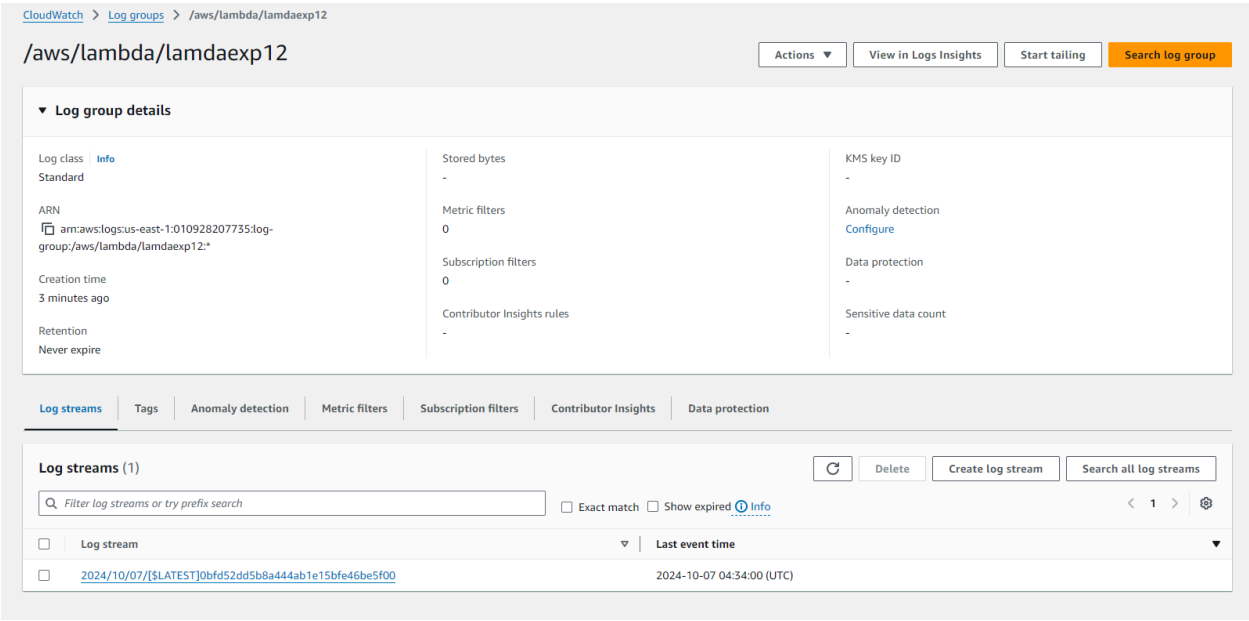
1) To check the logs explicitly, search foe CloudWatch on services and open it in a new tab



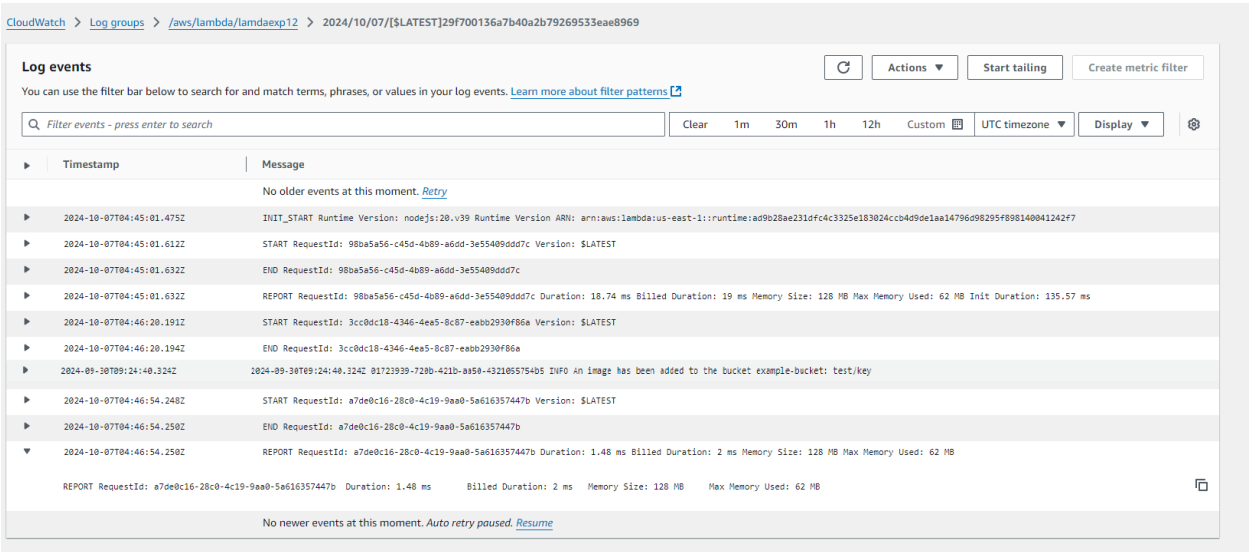
2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.



3) Here, under Log streams, select the log stream you want to check.



4) Here again, we can see that ‘An image has been added to the bucket’.



Conclusion: In this experiment, we successfully implemented a Lambda function that automatically logs a message whenever an image is uploaded to a designated S3 bucket. By setting up an S3 bucket trigger, we showcased how AWS services can efficiently interact to automate workflows. The Lambda function captured essential event details, such as the bucket name and object key. After deploying the function, we tested it with a sample image upload and confirmed through CloudWatch logs that the function correctly identified and logged the event. This experiment demonstrated the smooth integration of AWS Lambda with S3 for handling event-driven automation.