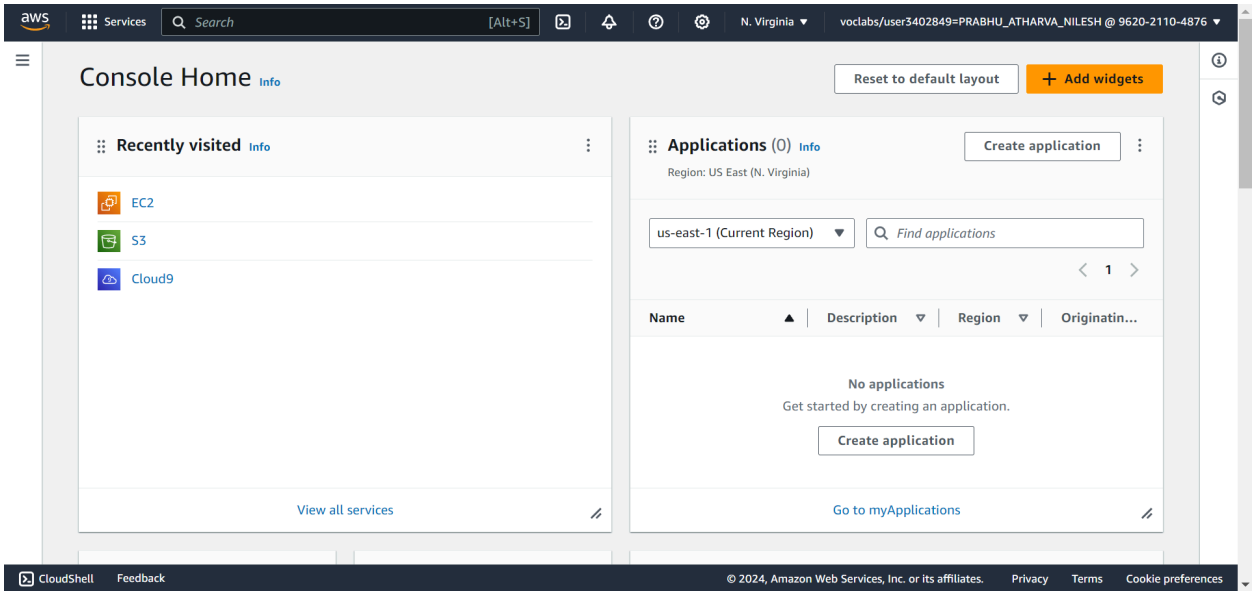
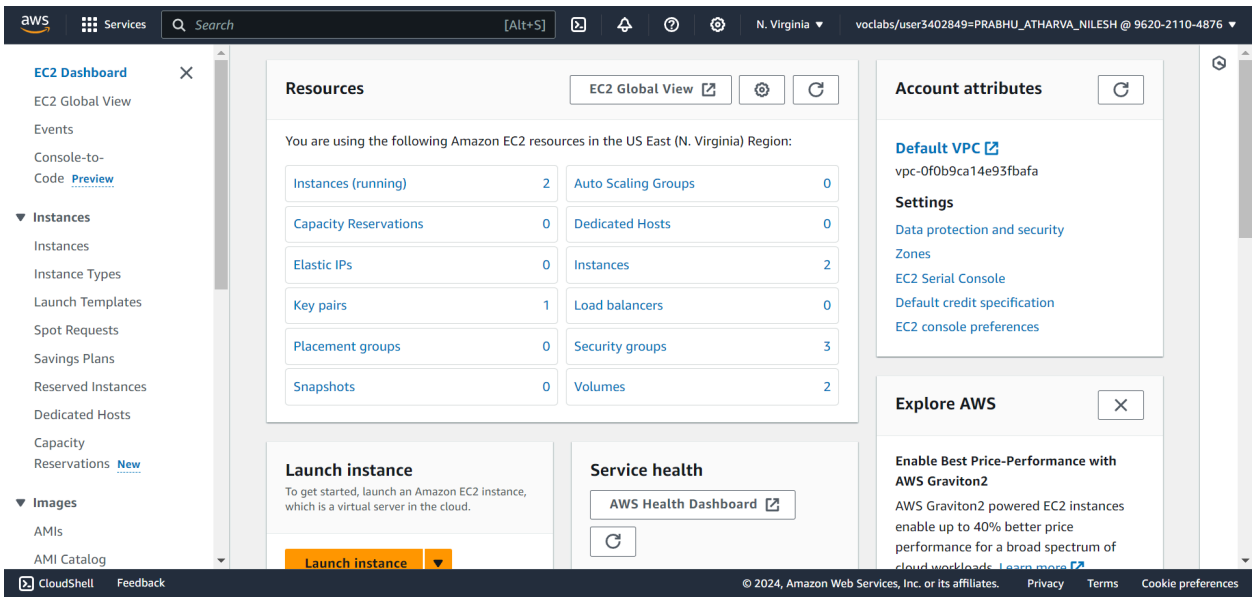
Experiment 1

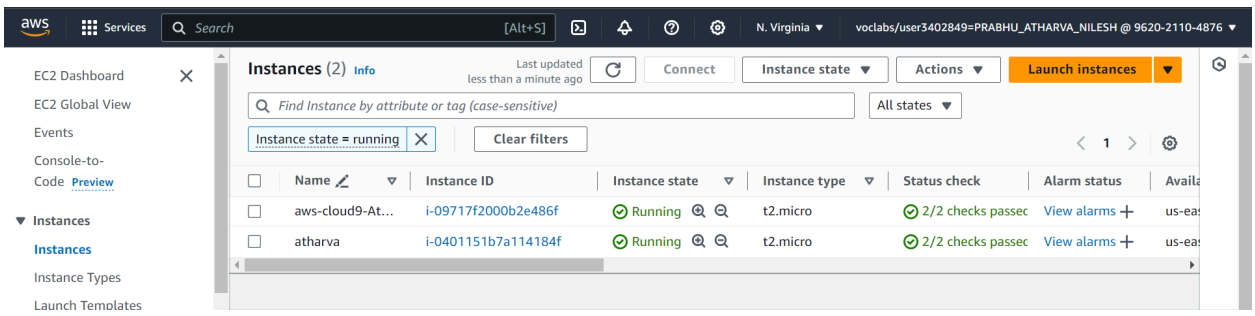
Atharva Prabhu

D15A 43

1. Open AWS Academia and select launch instance







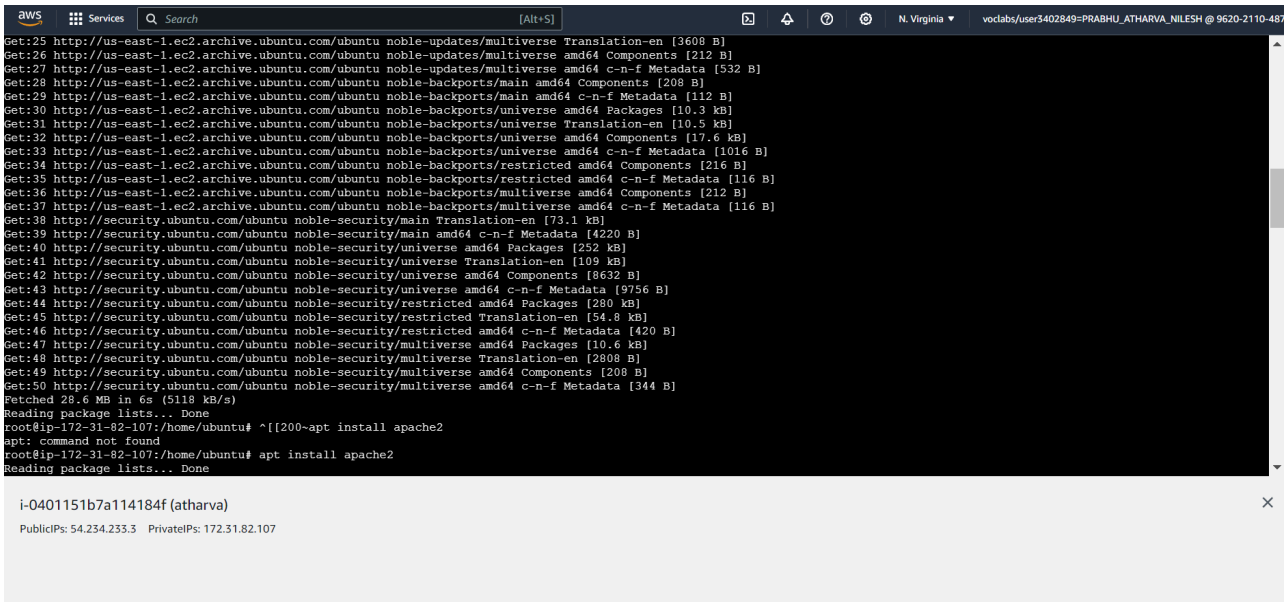
2. Execute the following commands in the aws console.
Commands :
sudo su
sudo apt install
sudo apt-get update
apt install apache2
systemctl status apache2
cd /var/www/html/

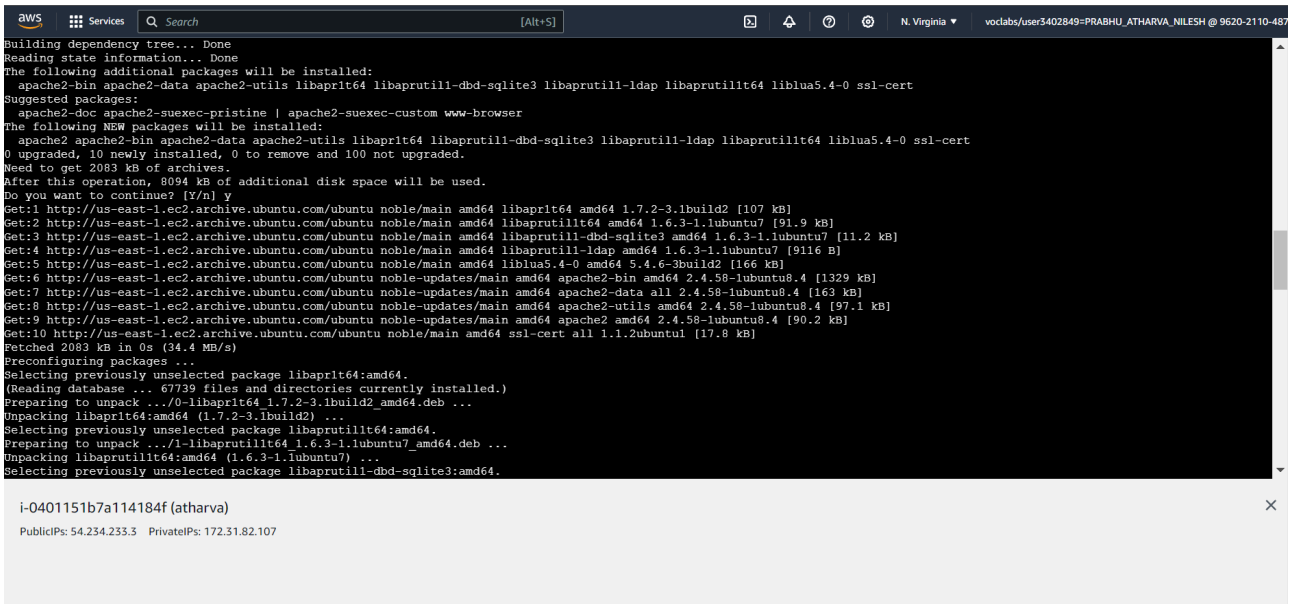
```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

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

 System information as of Sun Aug 25 03:11:28 UTC 2024

  System load:  0.09              Processes:             105
  Usage of /:   23.4% of 6.71GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for enX0: 172.31.82.107
  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

Last login: Thu Aug 22 05:23:58 2024 from 18.206.107.29
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-82-107:~$ sudo su
root@ip-172-31-82-107:/home/ubuntu# sudo apt install
Reading package lists... Done
```

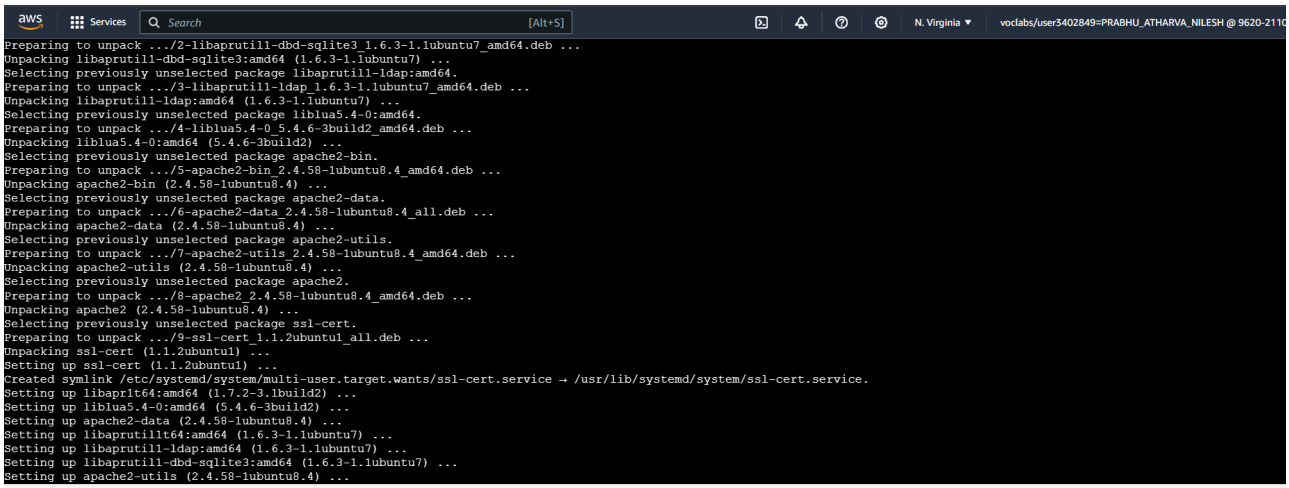
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107

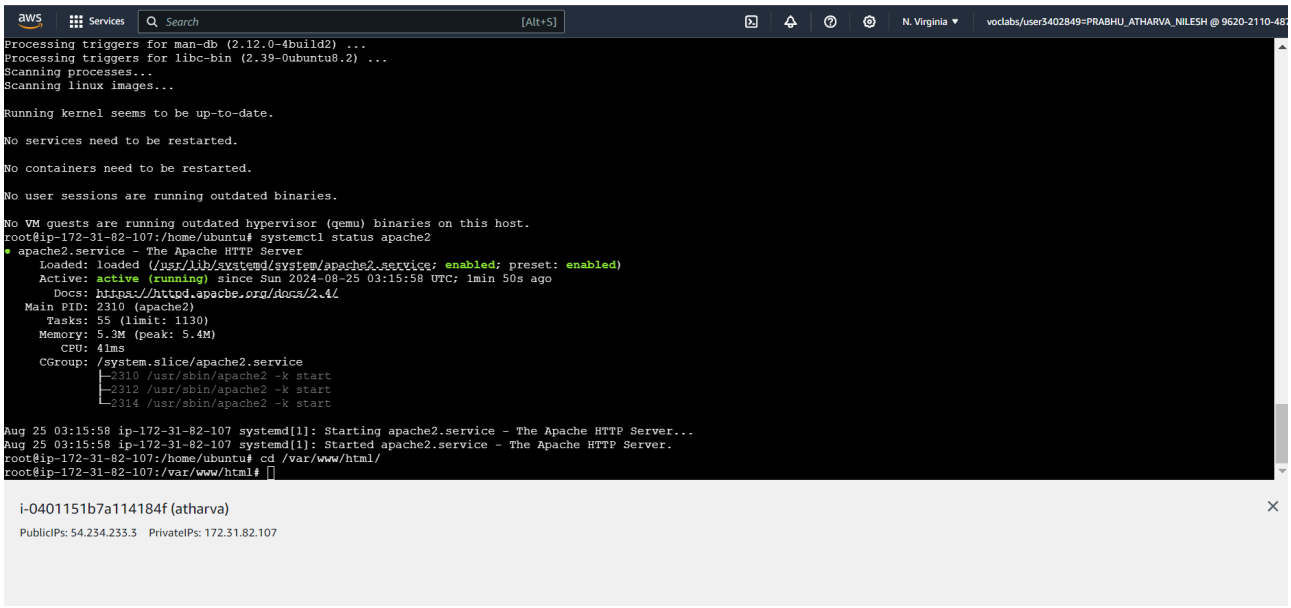
```
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-82-107:/home/ubuntu# ^[[200~sudo apt-get update
sudo: command not found
root@ip-172-31-82-107:/home/ubuntu# ^[[200~sudo apt-get update
sudo: command not found
root@ip-172-31-82-107:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [323 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [114 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7192 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [142 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [13.6 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [280 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [54.8 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
```

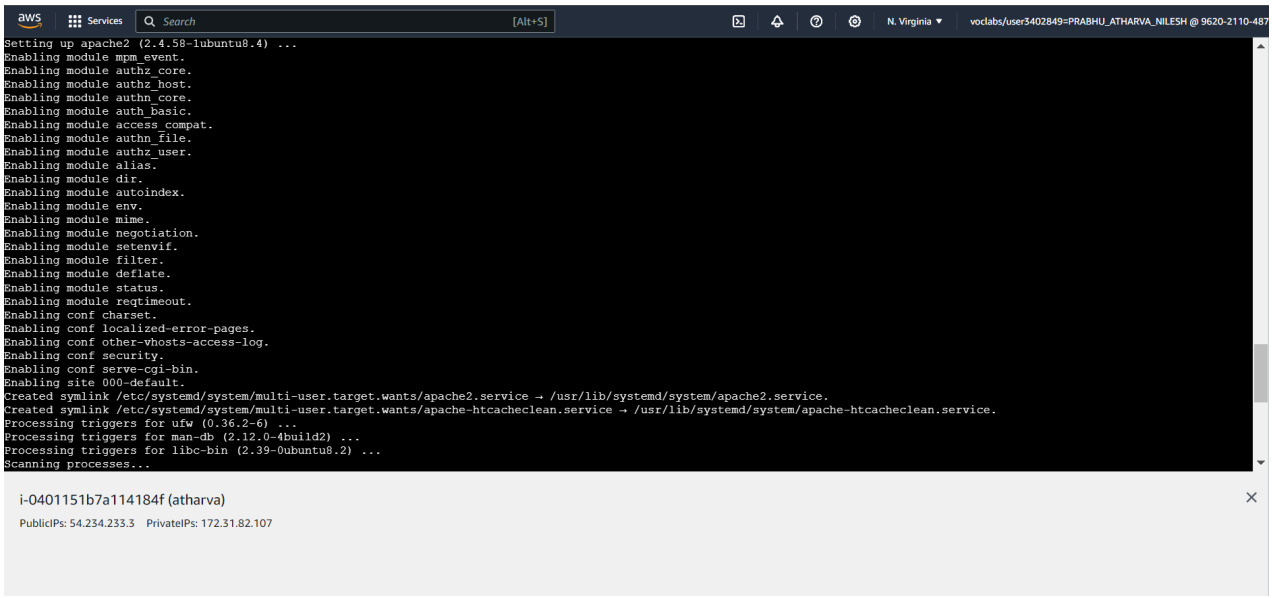
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402849=PRABHU_ATHARVA_NILESH @ 9620-2110-487
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [73.1 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4220 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [252 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [109 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9756 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [280 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [54.8 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [420 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 28.6 MB in 6s (5118 kB/s)
Reading package lists... Done
root@ip-172-31-82-107:/home/ubuntu# ^[[200~apt install apache2
apt: command not found
root@ip-172-31-82-107:/home/ubuntu# apt install apache2
Reading package lists... Done
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107

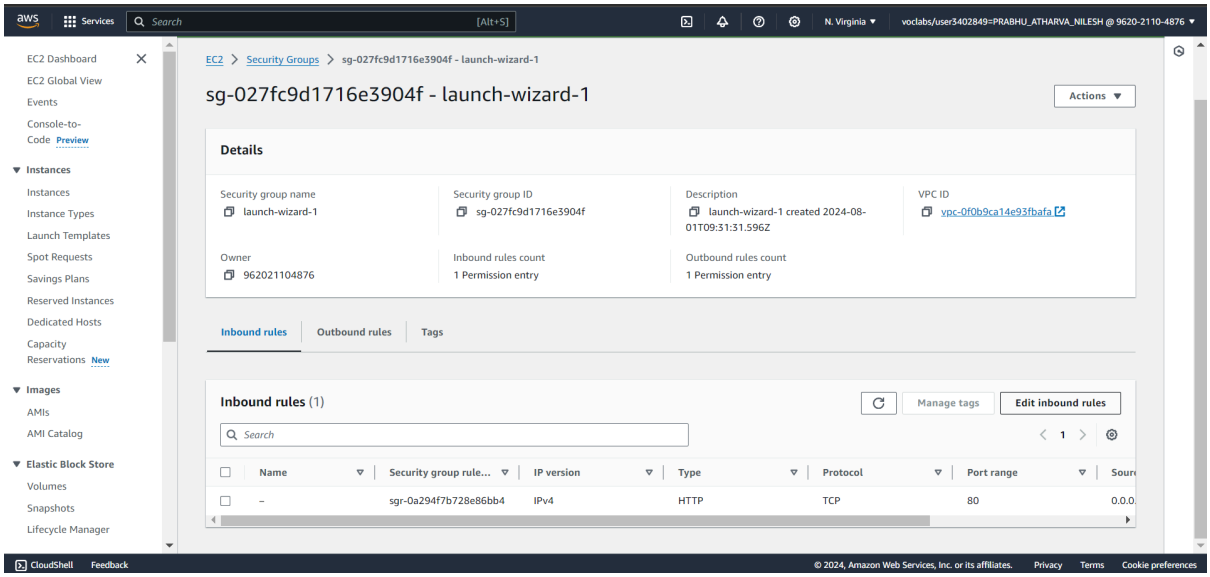

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402849=PRABHU_ATHARVA_NILESH @ 9620-2110-487
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 100 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
Fetched 2083 kB in 0s (34.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1t64:amd64.
(Reading database ... 67739 files and directories currently installed.)
Preparing to unpack .../0-libapr1t64_1.7.2-3.1build2_amd64.deb ...
Unpacking libapr1t64:amd64 (1.7.2-3.1build2) ...
Selecting previously unselected package libaprutil1t64:amd64.
Preparing to unpack .../1-libaprutil1t64_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107

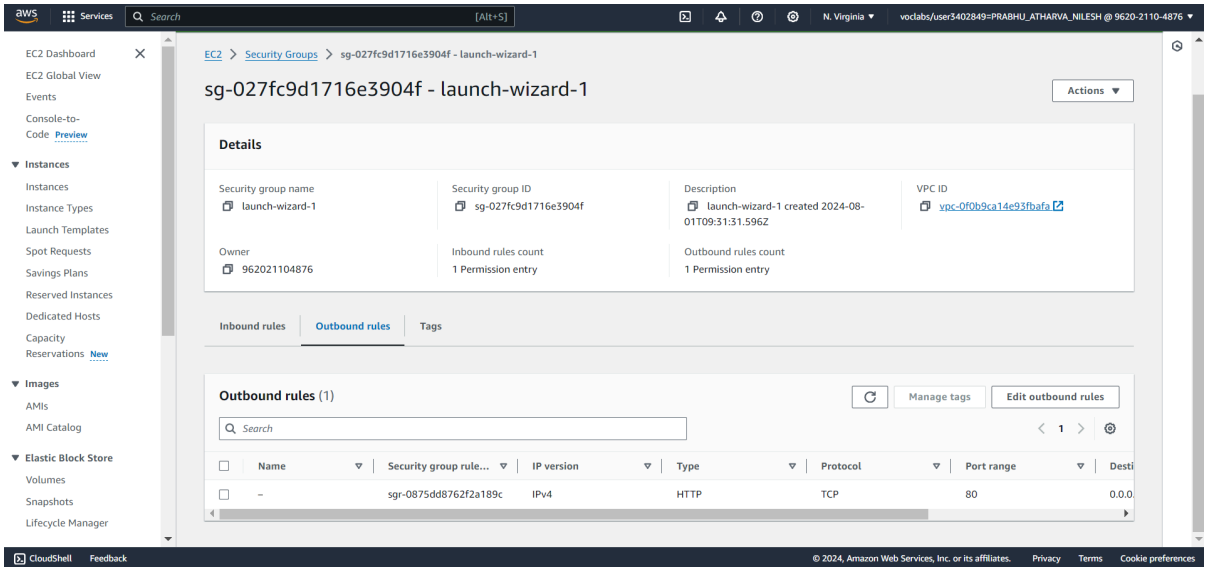

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402849=PRABHU_ATHARVA_NILESH @ 9620-2110
Preparing to unpack .../2-libaprutil1-dbd-sqlite3_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../3-libaprutil1-ldap_1.6.3-1.1ubuntu7_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package liblua5.4-0:amd64.
Preparing to unpack .../4-liblua5.4-0_5.4.6-3build2_amd64.deb ...
Unpacking liblua5.4-0:amd64 (5.4.6-3build2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../5-apache2-bin_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2-bin (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../6-apache2-data_2.4.58-1ubuntu8.4_all.deb ...
Unpacking apache2-data (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../7-apache2-utils_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2-utils (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2.
Preparing to unpack .../8-apache2_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2 (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../9-ssl-cert_1.1.2ubuntu1_all.deb ...
Unpacking ssl-cert (1.1.2ubuntu1) ...
Setting up ssl-cert (1.1.2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ssl-cert.service → /usr/lib/systemd/system/ssl-cert.service.
Setting up libapr1t64:amd64 (1.7.2-3.1build2) ...
Setting up liblua5.4-0:amd64 (5.4.6-3build2) ...
Setting up apache2-data (2.4.58-1ubuntu8.4) ...
Setting up libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
Setting up libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Setting up apache2-utils (2.4.58-1ubuntu8.4) ...
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie pr


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402849=PRABHU_ATHARVA_NILESH @ 9620-2110-482
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-82-107:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
Active: active (running) since Sun 2024-08-25 03:15:58 UTC; 1min 50s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 2310 (apache2)
Tasks: 55 (limit: 1130)
Memory: 5.3M (peak: 5.4M)
CPU: 41ms
CGroup: /system.slice/apache2.service
├─2310 /usr/sbin/apache2 -k start
├─2312 /usr/sbin/apache2 -k start
└─2314 /usr/sbin/apache2 -k start
Aug 25 03:15:58 ip-172-31-82-107 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 25 03:15:58 ip-172-31-82-107 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-82-107:/home/ubuntu# cd /var/www/html/
root@ip-172-31-82-107:/var/www/html#
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107

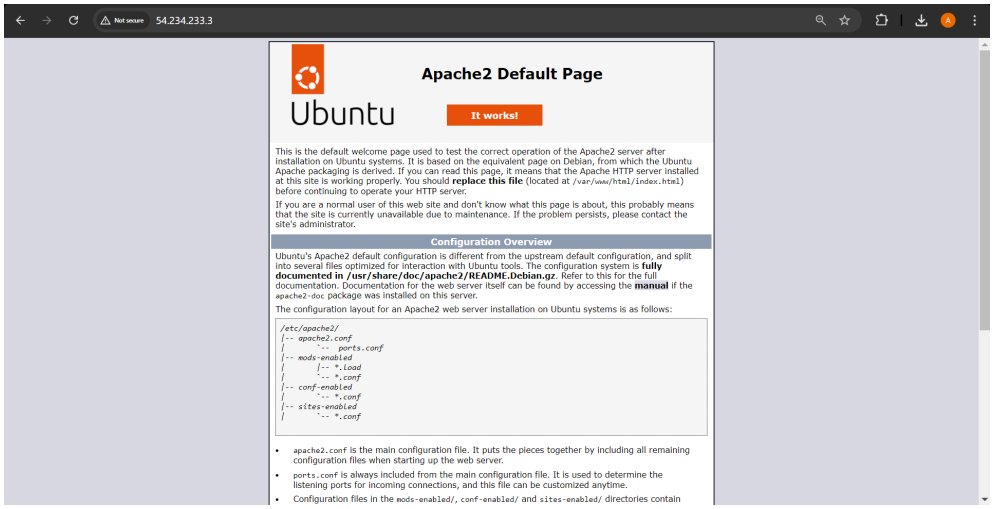

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402849=PRABHU_ATHARVA_NILESH @ 9620-2110-487
Setting up apache2 (2.4.58-1ubuntu8.4) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
i-0401151b7a114184f (atharva)
PublicIPs: 54.234.233.3 PrivateIPs: 172.31.82.107

3.Edit the inbound and outbound rules.



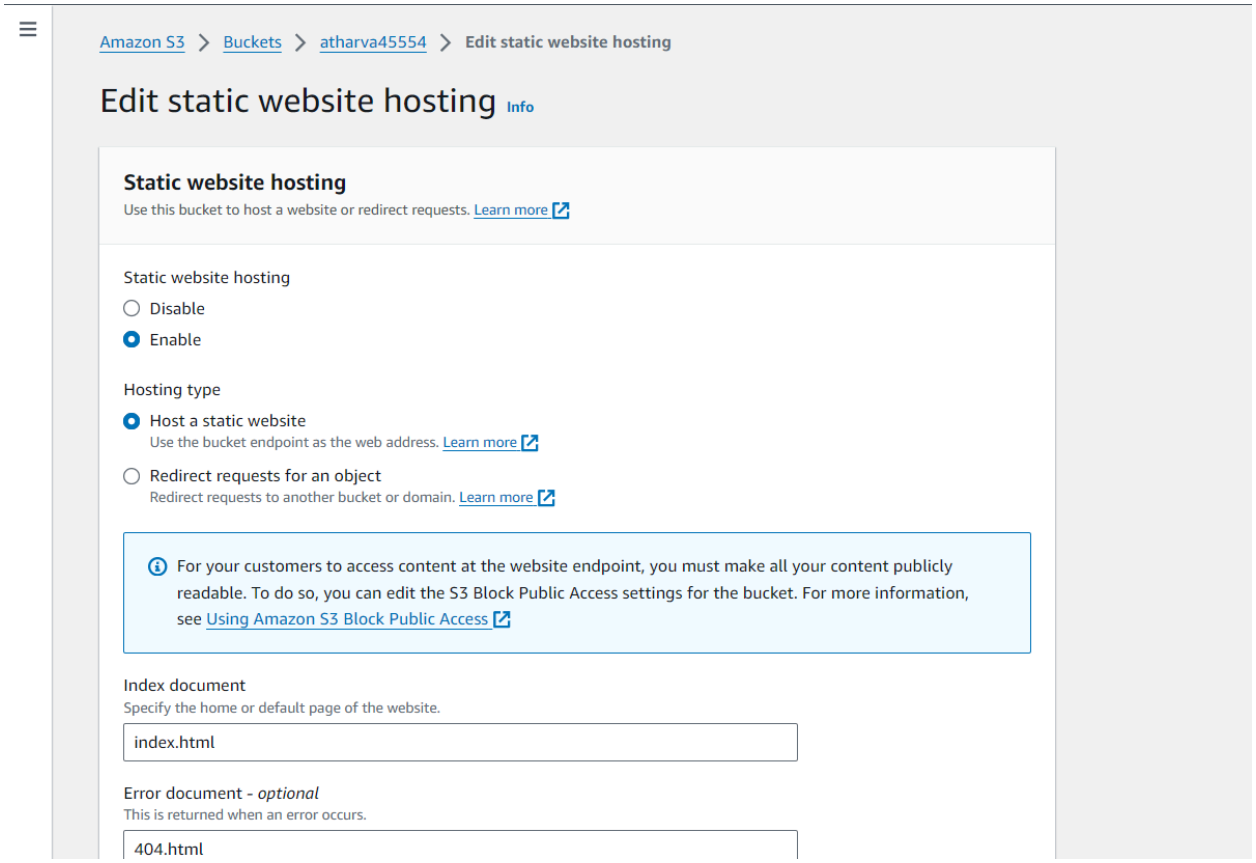


4. This is the hosted Static Website.

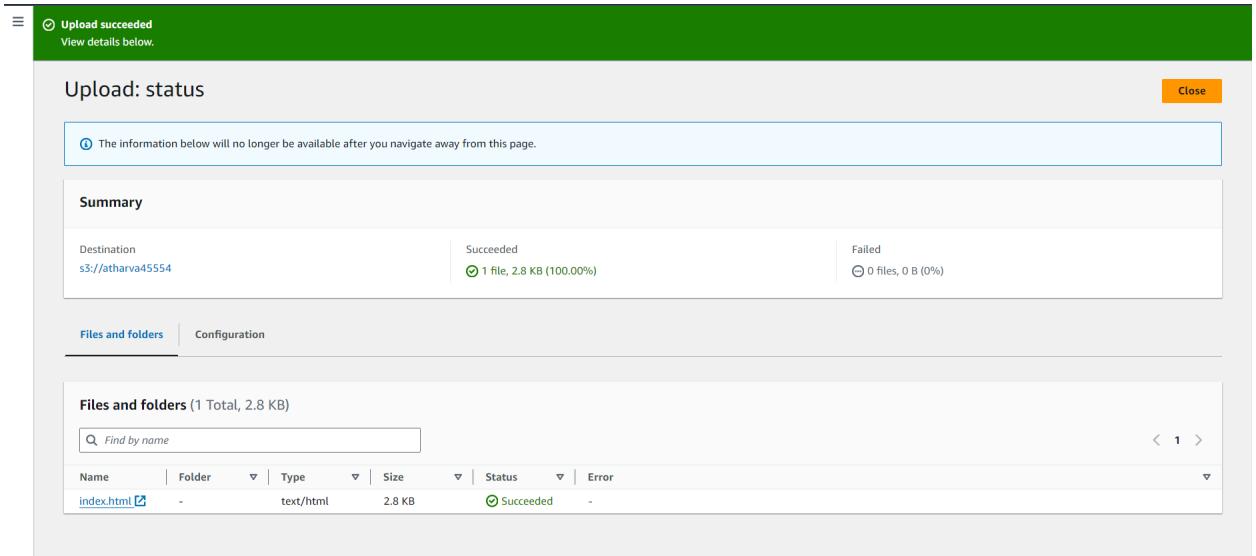


Static Site Hosting using S3 bucket

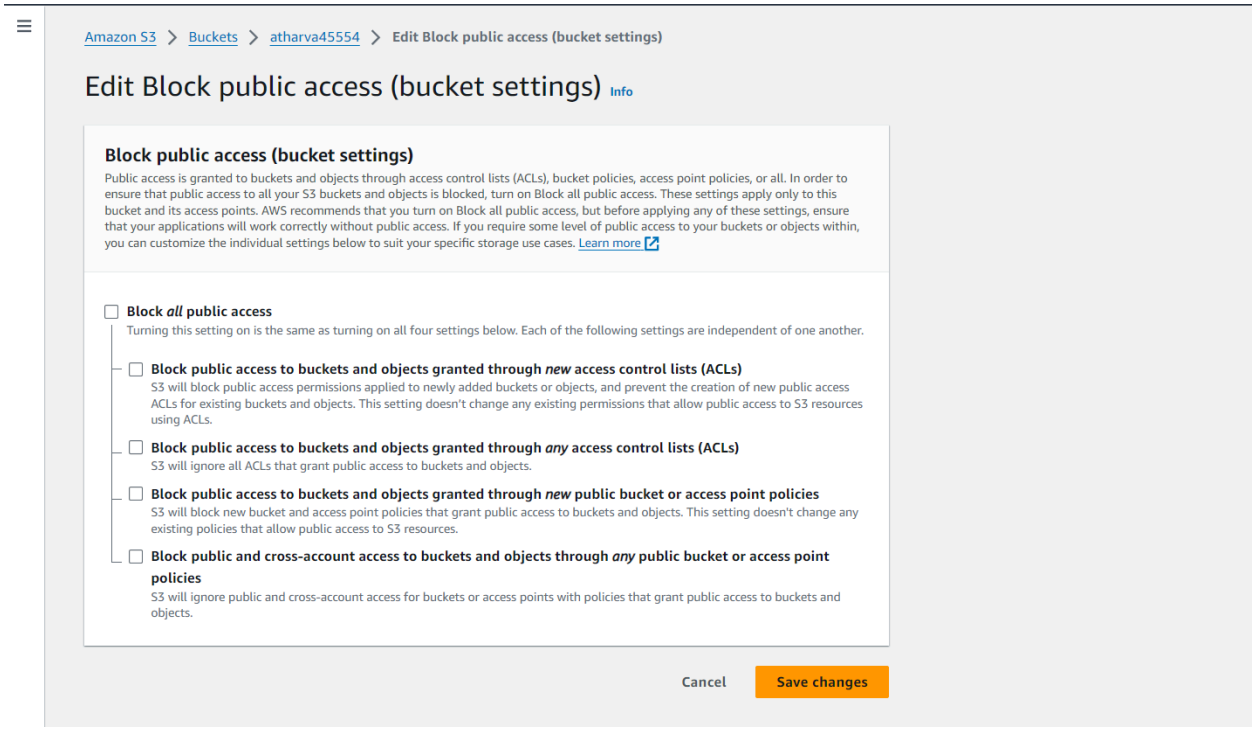
1. Visit S3 under the developer tools and create a Bucket.Click on the Edit Static Website Hosting under the properties tab



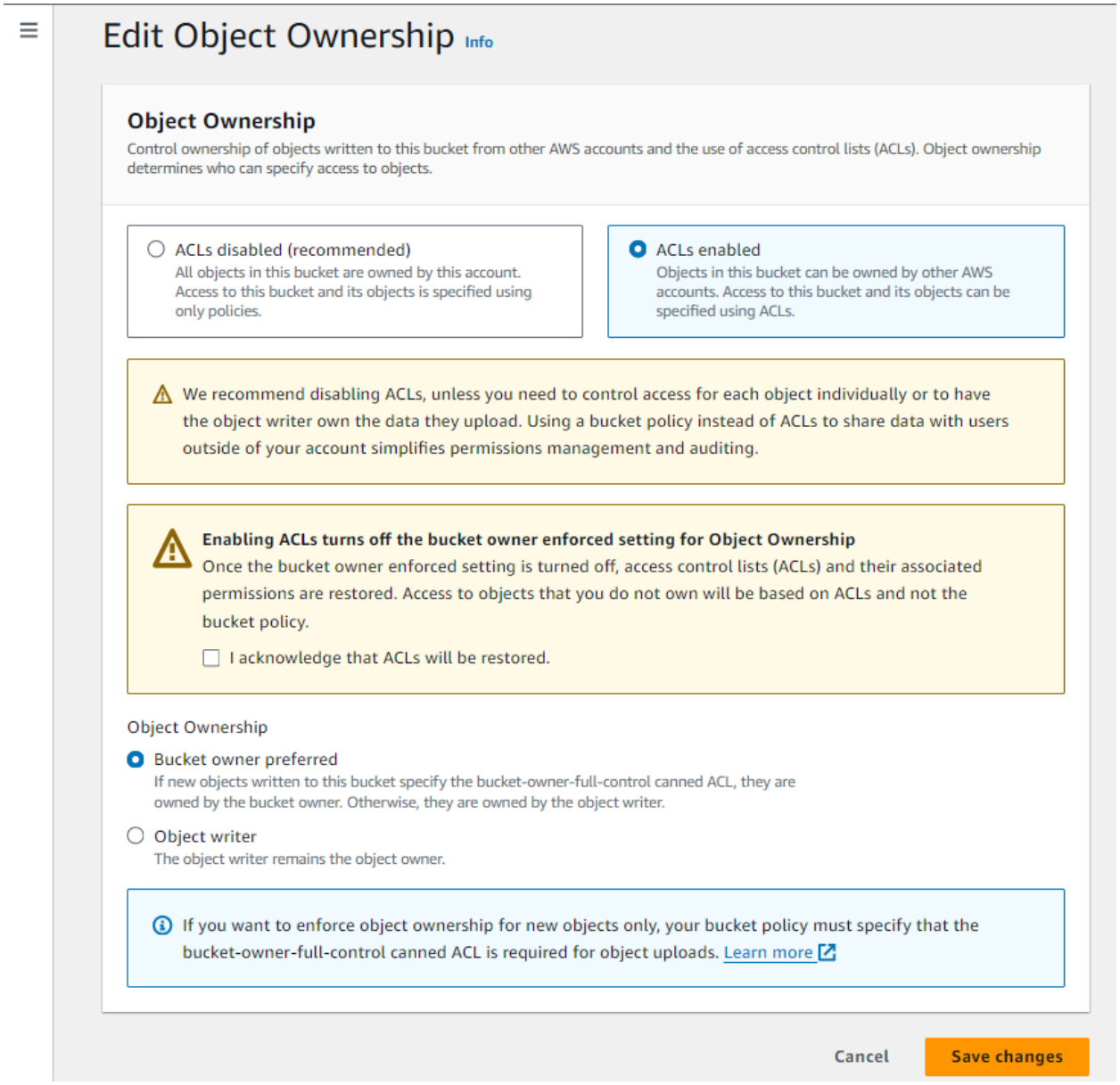
2. Upload a file



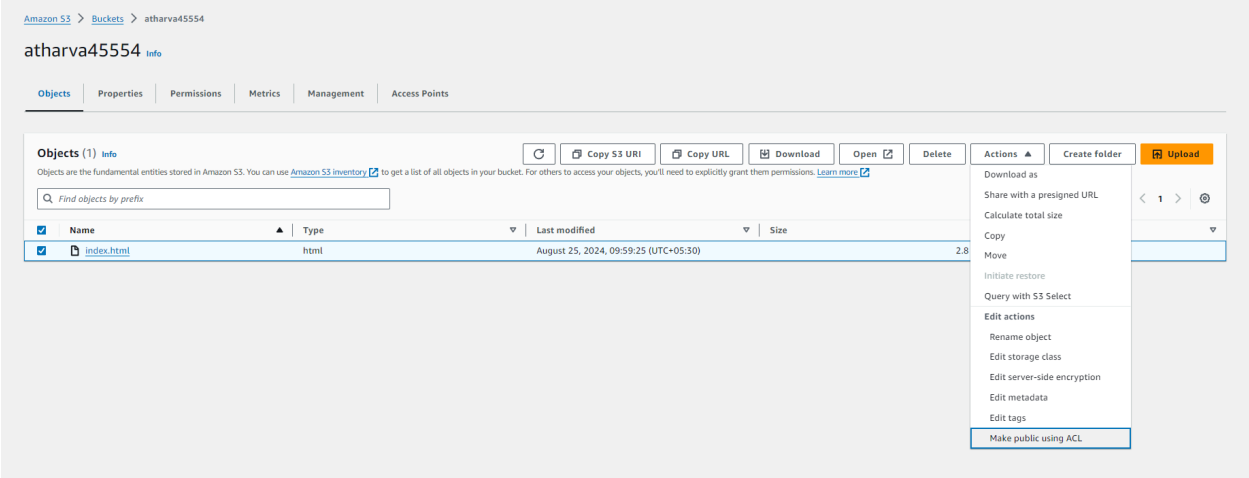
3. Click on the Edit block public access under the Permissions tab



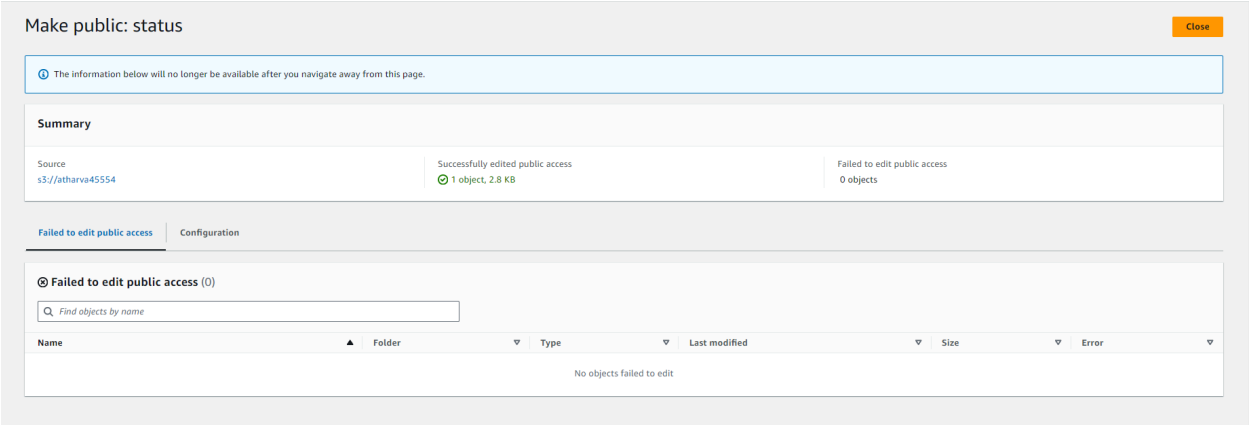
4. Click on Object Ownership under Permission Tab



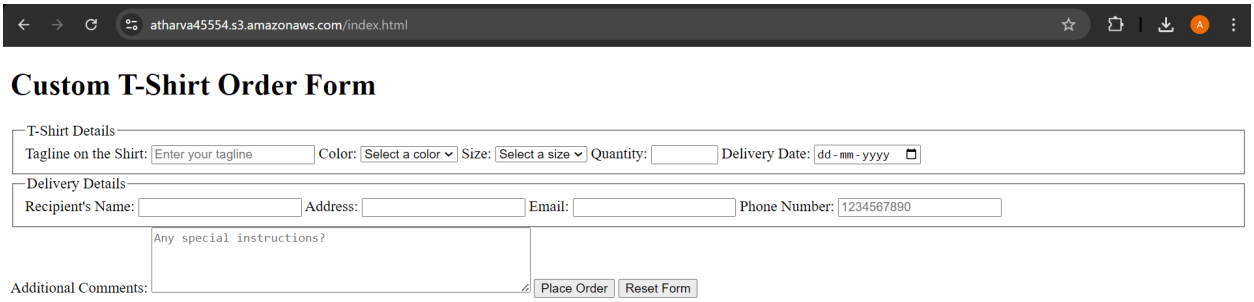
5. Select the file and click on Actions and select the option Make Public using ACL from the dropdown



6. Select on Make Public

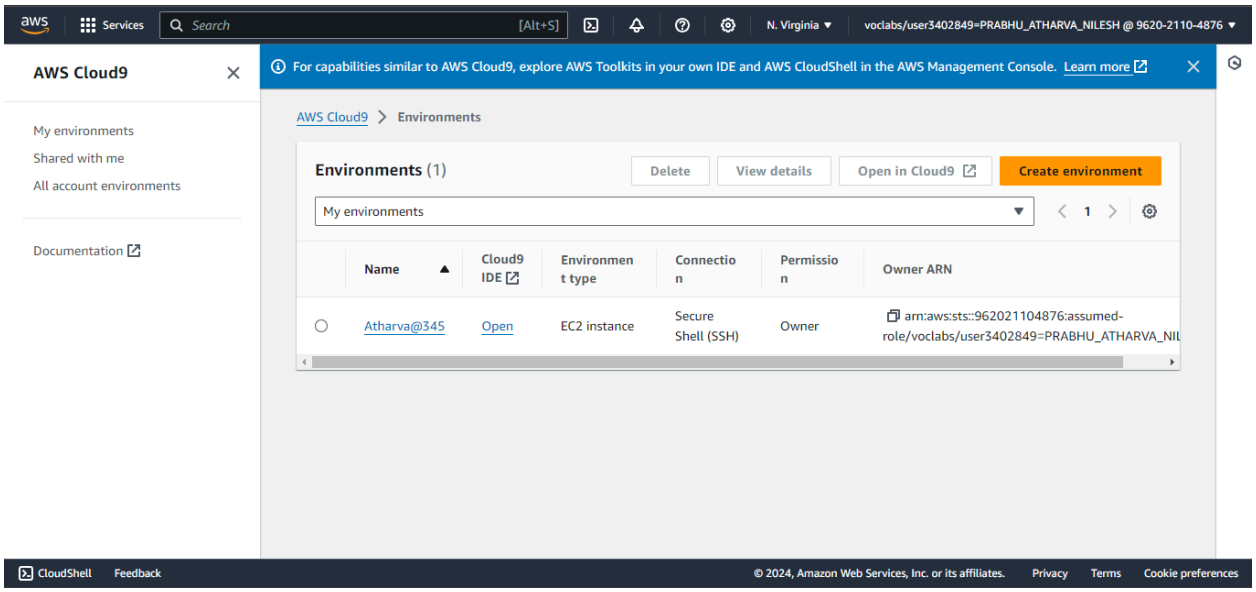


7. Visit the domain and the website hosted.

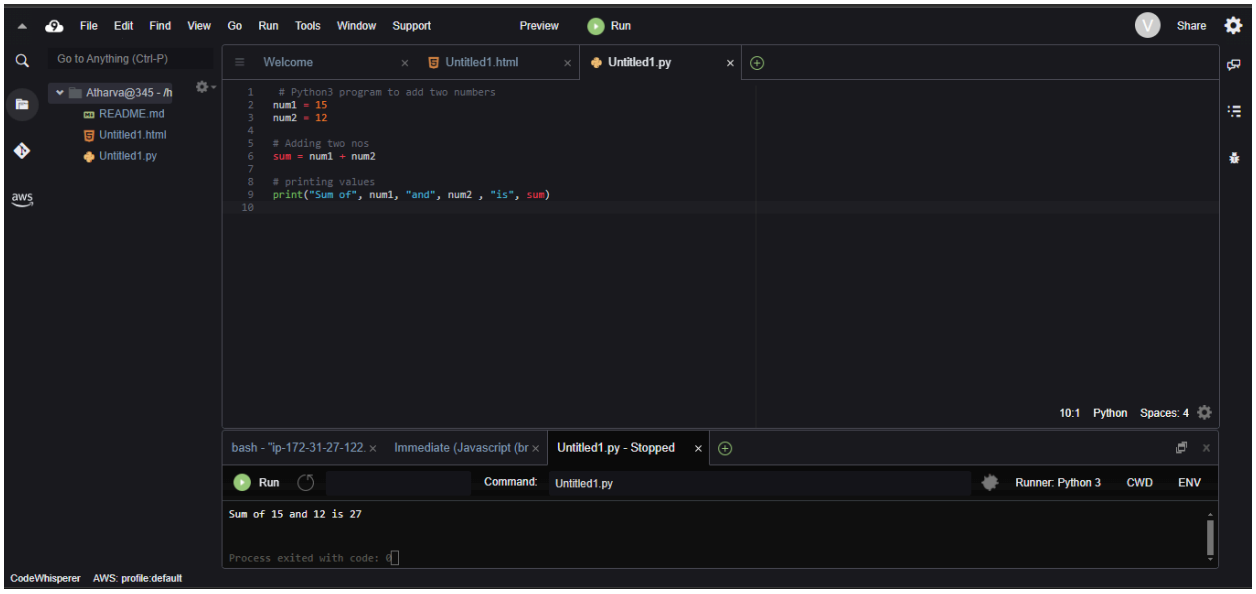


Cloud 9 IDE Site Hosting

Step 1: Create Environment



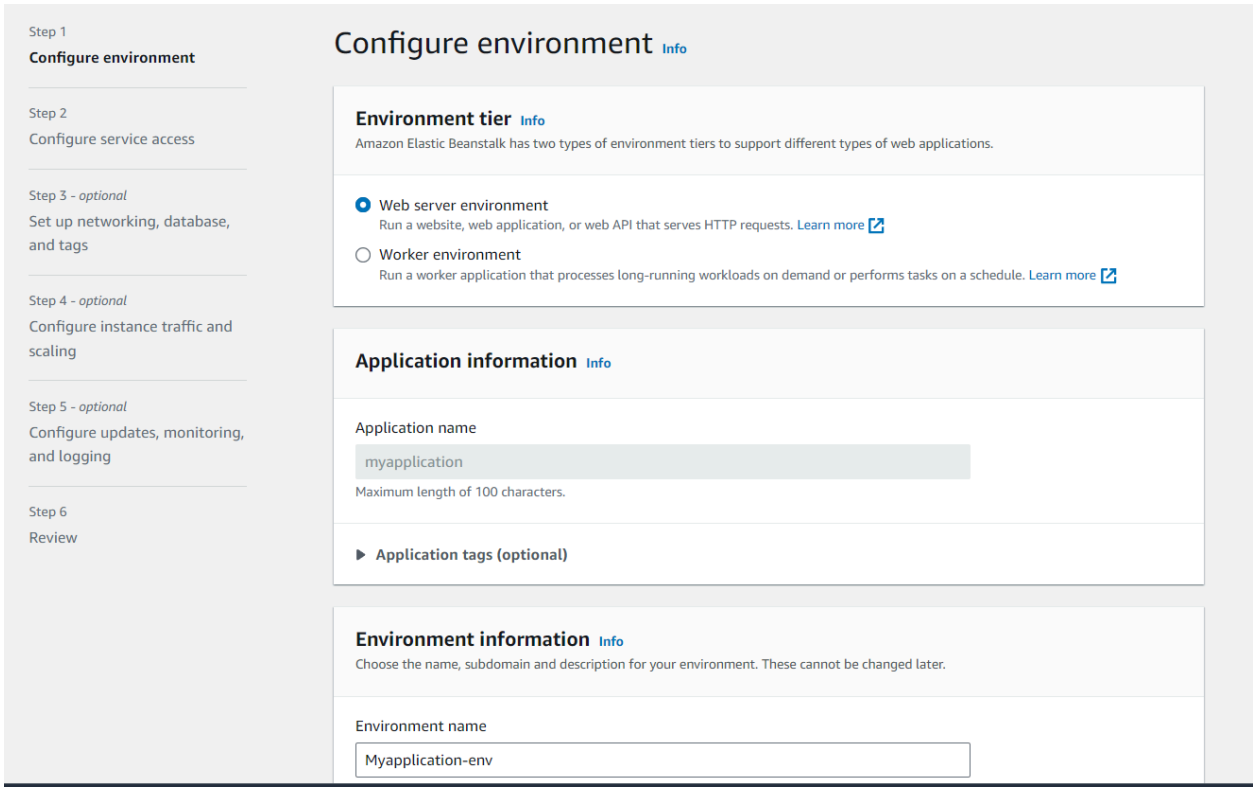
Step 2 :Open the Environment IDE and add the code and run the code



EXPERIMENT NO. 2

ATHARVA PRABHU
D15A 43

Step 1: create environment



Platform Info

Platform type

- Managed platform
  Platforms published and maintained by Amazon Elastic Beanstalk. Learn more
- Custom platform
  Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Python

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

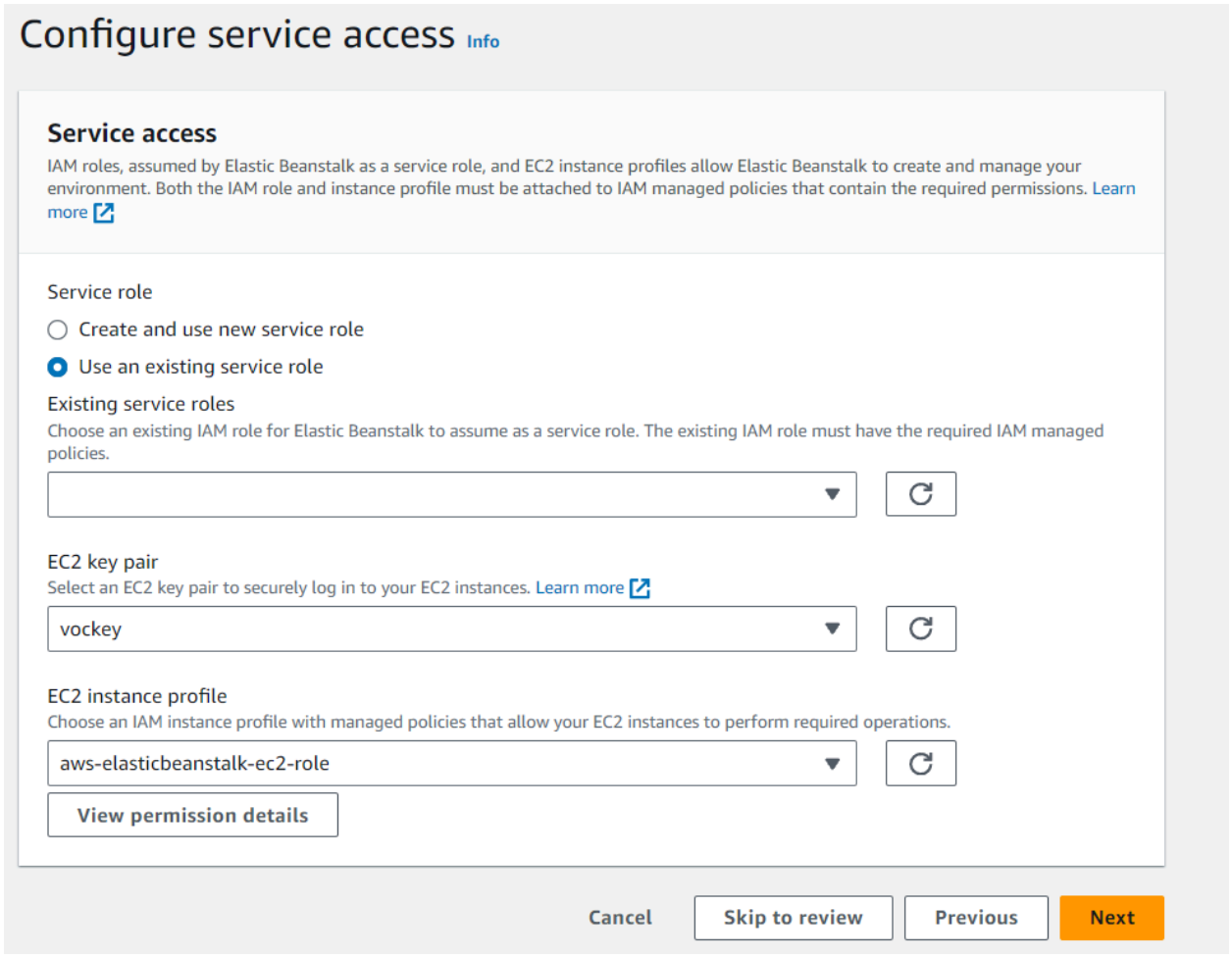
Platform version

4.1.3 (Recommended)

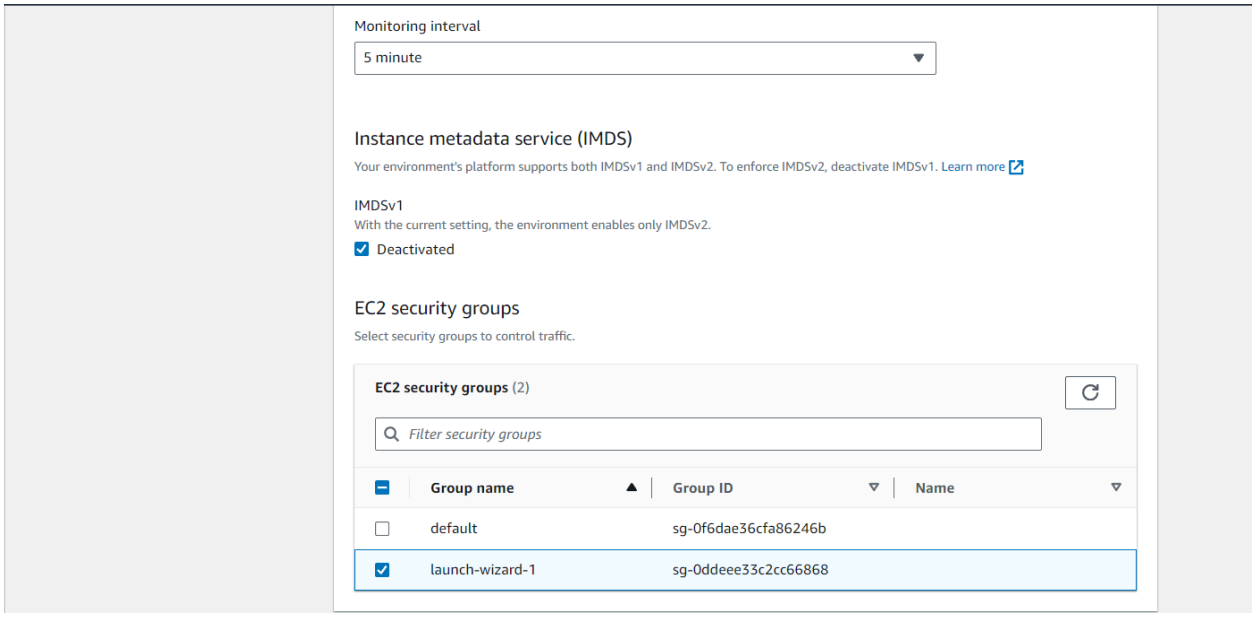
Application code Info

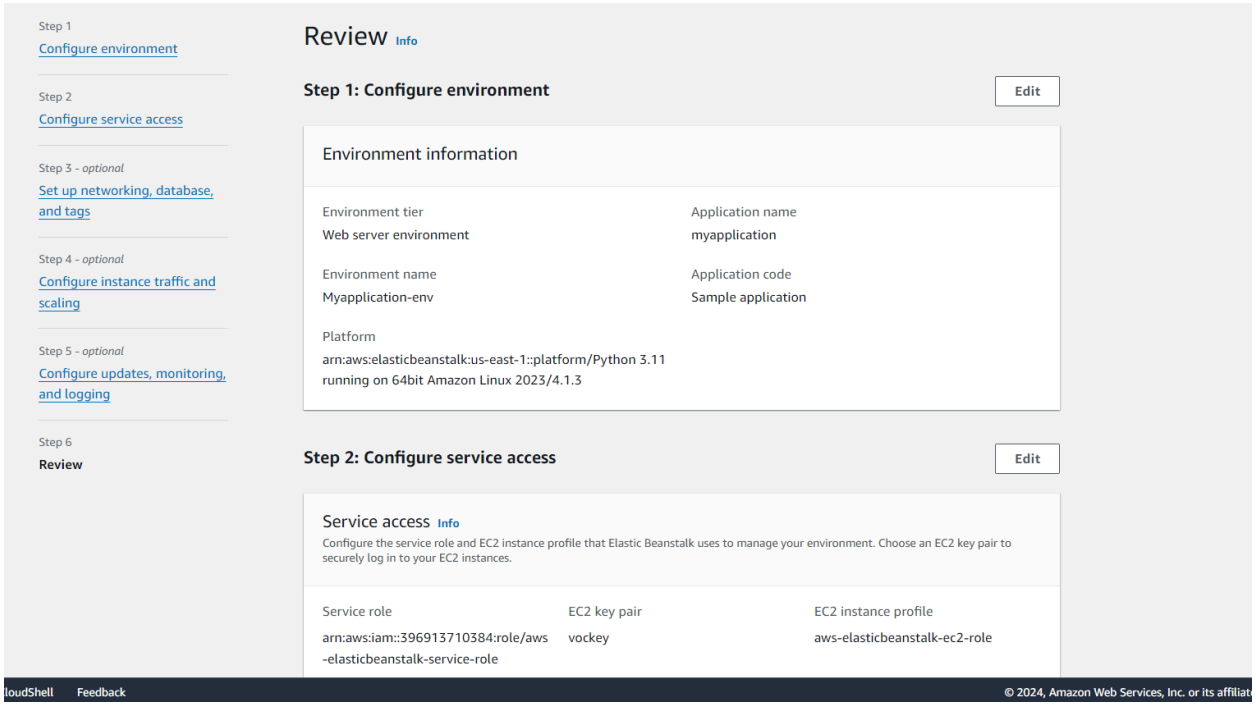
- Sample application
- Existing version
  Application versions that you have uploaded.
- Upload your code
  Upload a source bundle from your computer or copy one from Amazon S3.

Step 2 : add your Ec2 key pair and instance profile

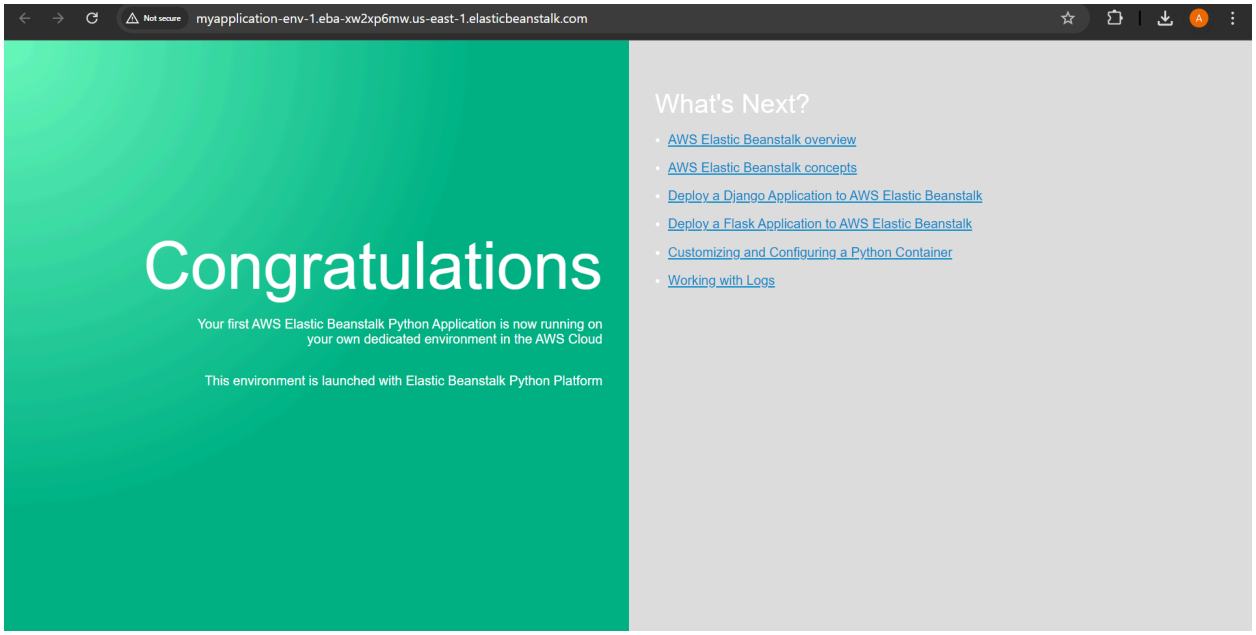


Step 3 : add security config and review all settings



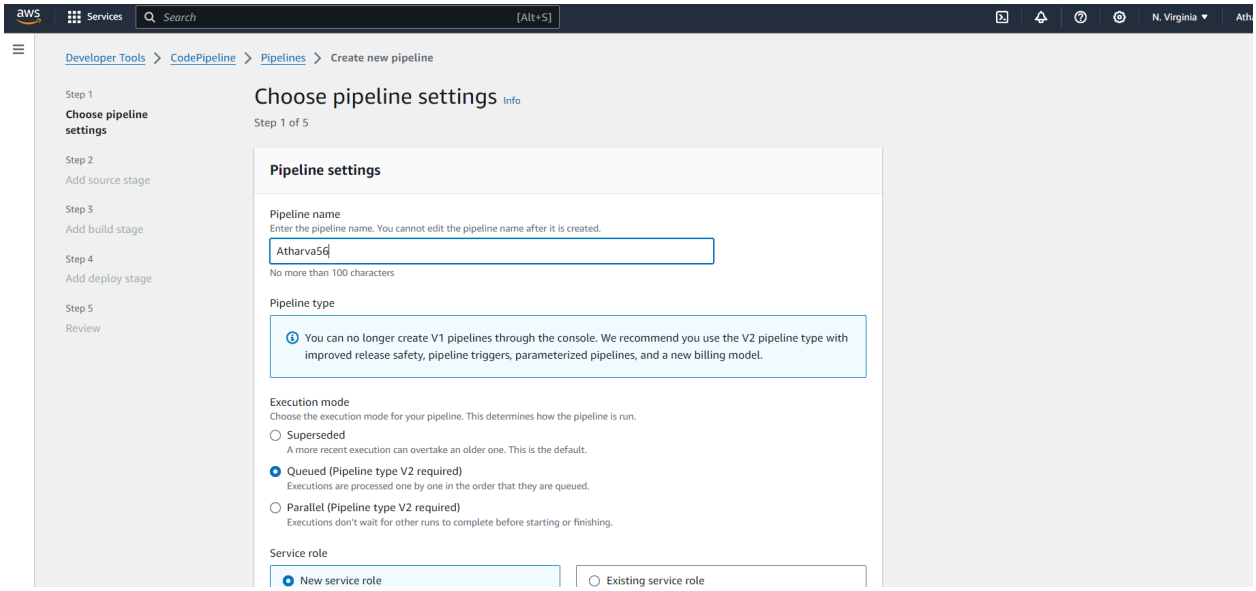


Step 4 : Beanstalk environment is created

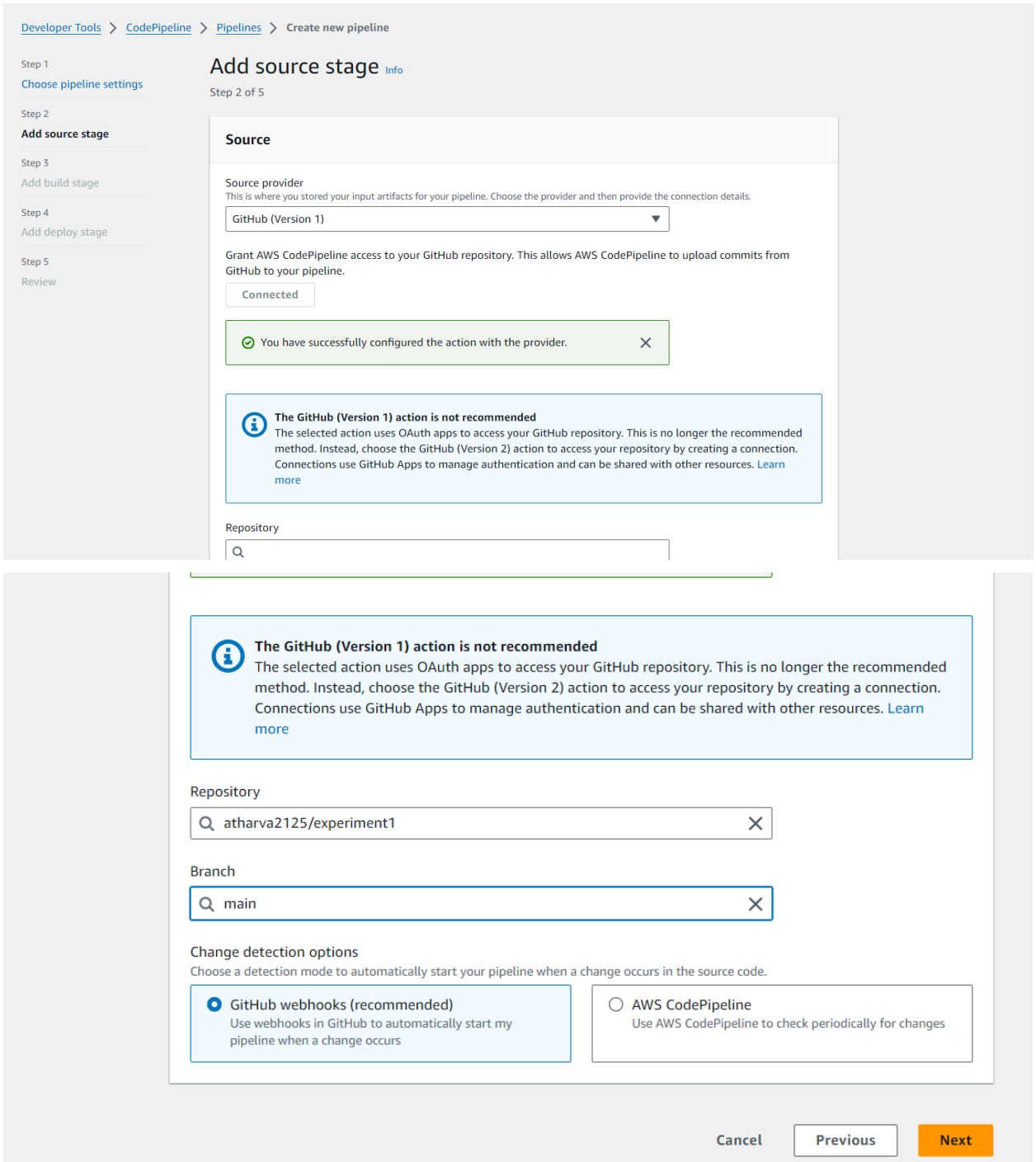


Pipeline Creation

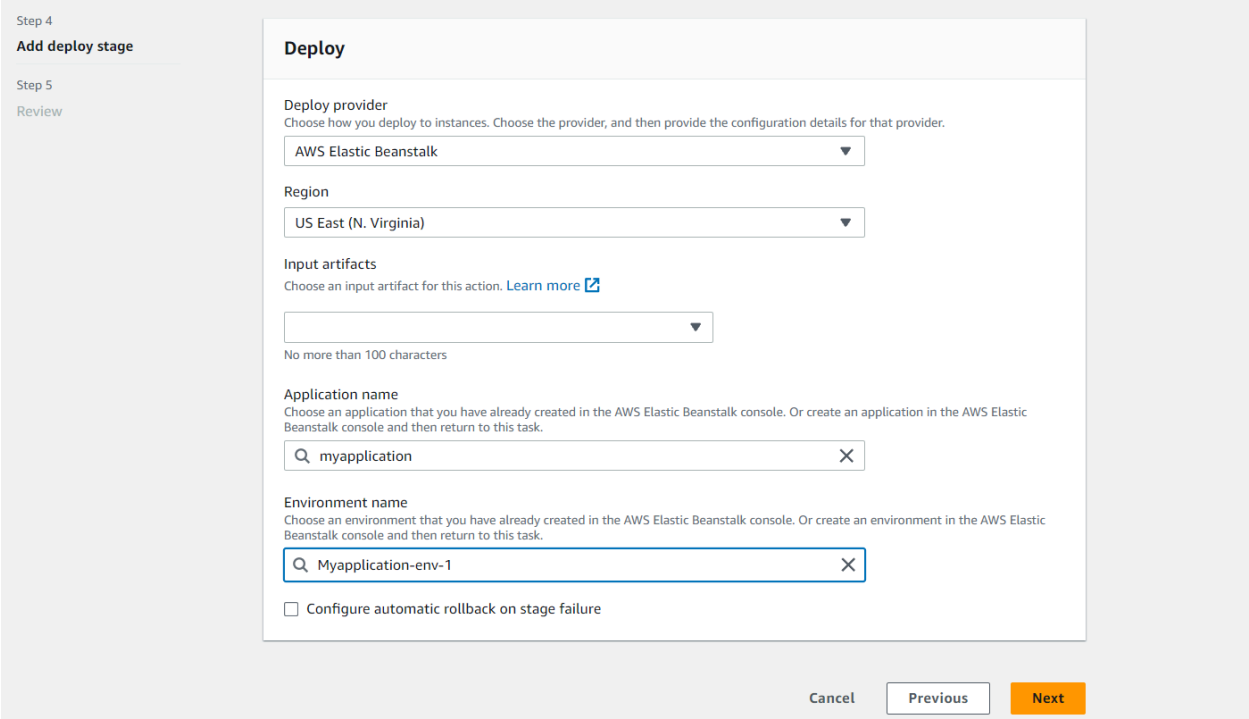
Step 1 : click on create pipeline and give name



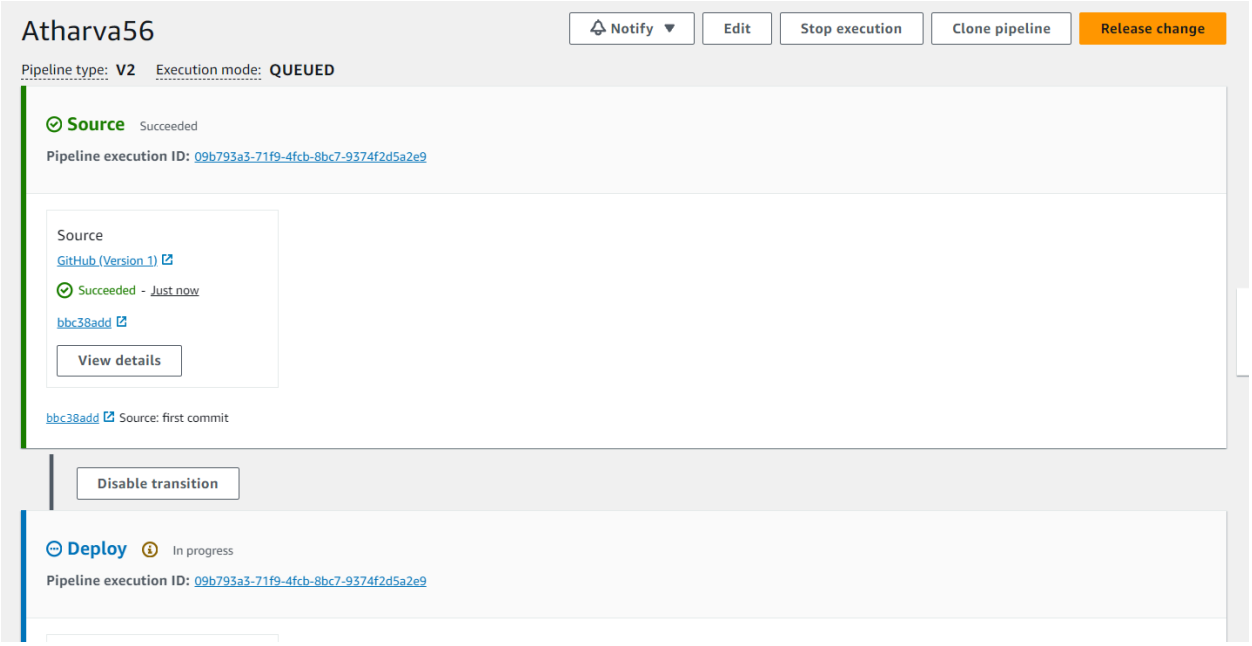
Step 2 : Add Your github account and add the file to add to pipeline deployment



Step 3 : Add deploy config choosing the elastic beanstalk



Step 5 : view the pipeline build and deployment



Step 6 : Check the deployed website at beanstalk link

PixelPerfect

Pixel Perfect

About
Services
Contact

Pixel Perfect is company which transform your digital vision into reality.

creative by profession

graphic design | web development | mobile app development | ux/ui design

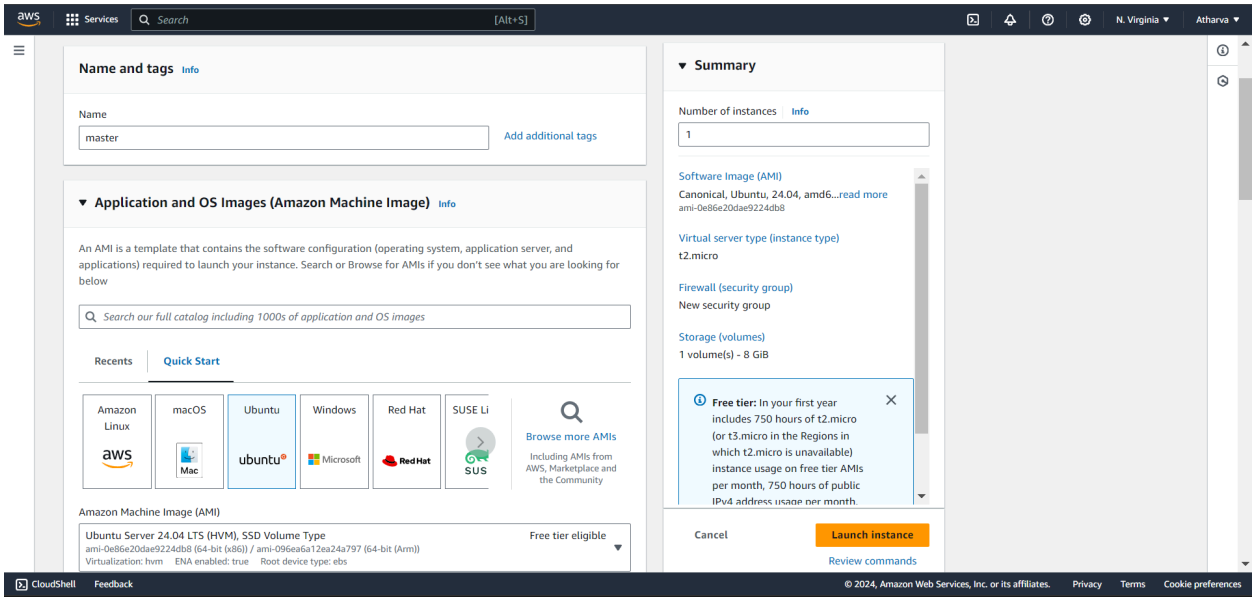
Watch Our Promotional Video

ATHARVA PRABHU **Experiment 3**
D15A 43

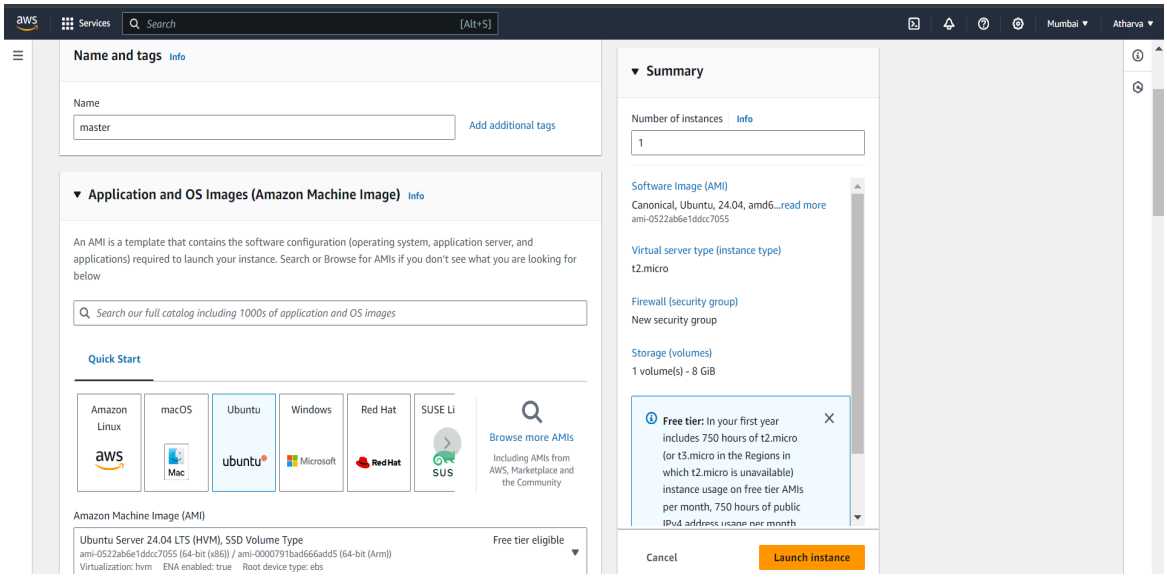
AIM: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

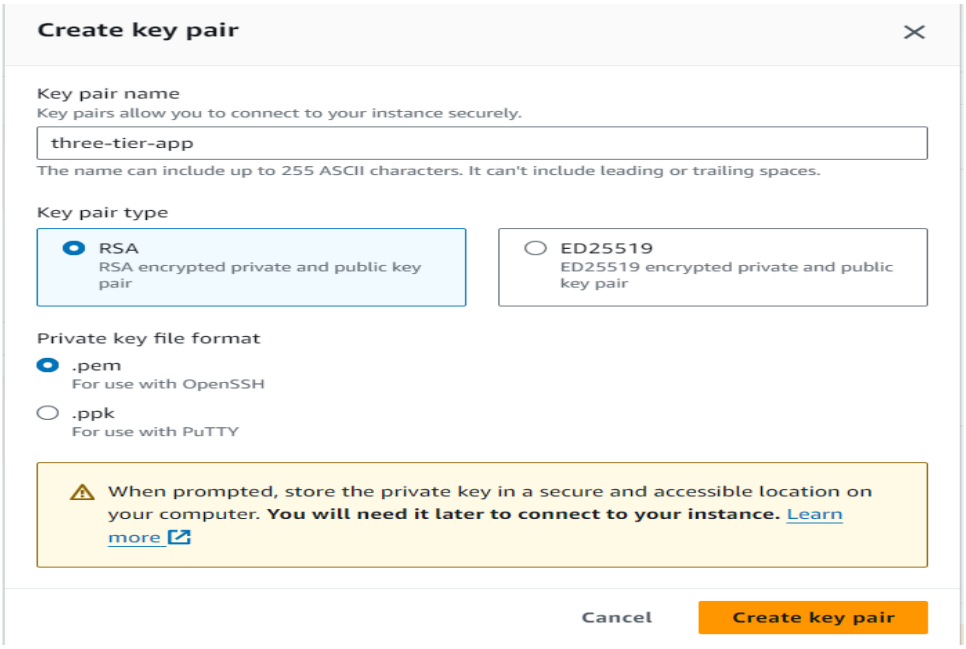
Step 1: Prerequisites

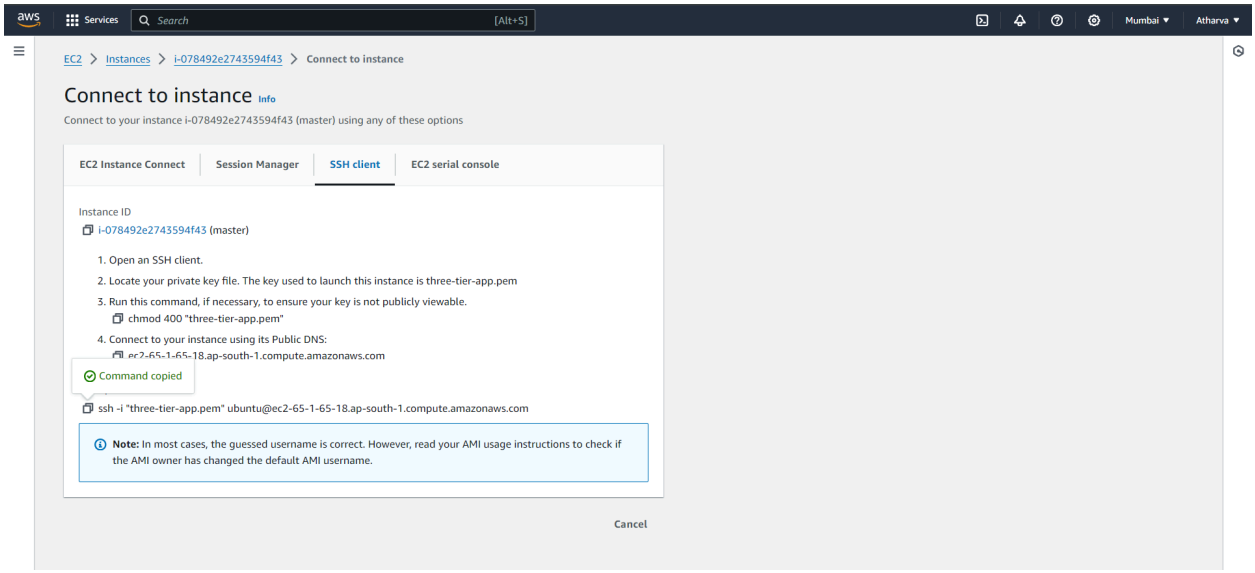
1.1 Create 3 EC2 instances,one for the master node and two for the worker nodes.



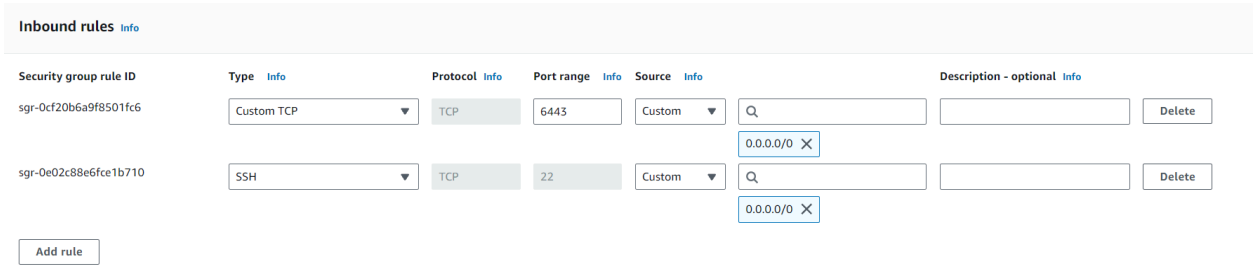
1.2 Proceed with the following settings and create a new key pair as follows(use the same key pair for all the three nodes)


aws
Services
Search
[Alt+S]
Mumbai
Atharva
Name and tags Info
Name
master
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0522ab6e1ddcc7055
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public
Cancel
Launch instance


Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
three-tier-app
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Cancel
Create key pair



1.3 Add port 6443 in each security group



1.4 After the instances have been created,copy the text given in the example part of each of the three instances into git bash.

```
C:\Users\Atharva\Downloads>ssh -i "three-tier-app.pem" ubuntu@ec2-65-1-65-18.ap-south-1.compute.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 Sat Sep 21 10:50:19 UTC 2024

  System load:  0.11              Processes:             115
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for enX0: 172.31.46.220
  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

Last login: Sat Sep 21 10:44:19 2024 from 49.36.97.186
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 2: Run the following commands on both the master and worker nodes to prepare them for kubeadm.

```
# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system
```

```
## Install CRIO Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https
ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg]
https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee
/etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/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:/v1.29/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet
```

```
ubuntu@ip-172-31-46-220:~$ # disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRIO Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /sudo systemctl start kubeletkubelet29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"k8s.io/co
re:/stable:/v1.29/deb/ /' | sud
overlay
br_netfilter
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
* Applying /usr/lib/sysctl.d/10-apparmor.conf ...
* Applying /etc/sysctl.d/10-console-messages.conf ...
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
* Applying /etc/sysctl.d/10-map-count.conf ...
* Applying /etc/sysctl.d/10-network-security.conf ...
* Applying /etc/sysctl.d/10-ptrace.conf ...
* Applying /etc/sysctl.d/10-zeropage.conf ...
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
* Applying /etc/sysctl.d/99-sysctl.conf ...
```

Step3: Run the above command only on master node

sudo kubeadm config images pull

sudo kubeadm init

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

Network Plugin = calico
kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml

kubeadm token create --print-join-command

```
ubuntu@ip-172-31-46-220:~$ sudo kubeadm config images pull

sudo kubeadm init

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


# Network Plugin = calico
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml

kubeadm token create --print-join-command
I0921 11:12:21.776309    3963 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.29
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.29.9
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.29.9
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.29.9
[config/images] Pulled registry.k8s.io/kube-proxy:v1.29.9
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.11.1
[config/images] Pulled registry.k8s.io/pause:3.9
[config/images] Pulled registry.k8s.io/etcd:3.5.10-0
I0921 11:12:40.995686    4304 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.9
[preflight] Running pre-flight checks
[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'
W0921 11:12:41.763411    4304 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.10" of the container runtime is inconsistent with that used by kubeadm. It is recommended that using "regis
try.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-172-31-46-220 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.46.220]
[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-172-31-46-220 localhost] and IPs [172.31.46.220 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-46-220 localhost] and IPs [172.31.46.220 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
```

You will get kubeadm token, Copy it.

Step 4: Run the above command only on worker nodes

sudo kubeadm reset pre-flight checks
sudo your-token --v=5

```
ubuntu@ip-172-31-36-212:~$ sudo kubeadm reset pre-flight checks
W0921 11:14:17.713660    3933 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: yes
[preflight] Running pre-flight checks
W0921 11:14:20.535200    3933 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/
scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
```

```
ubuntu@ip-172-31-36-212:~$ sudo kubeadm join 172.31.46.220:6443 --token k4psyh.ns1g1yet9he59kd4 --discovery-token-ca-cert-hash sha256:80e7e9abf8f31f0333a9d2f7a680d6bdf961267ae45cf71bada8de069d4a292e --v=5
I0921 11:28:31.063878    4097 join.go:413] [preflight] found NodeName empty; using OS hostname as NodeName
I0921 11:28:31.064085    4097 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/crio/crio.sock
[preflight] Running pre-flight checks
I0921 11:28:31.064143    4097 preflight.go:93] [preflight] Running general checks
I0921 11:28:31.064183    4097 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I0921 11:28:31.064207    4097 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0921 11:28:31.064219    4097 checks.go:104] validating the container runtime
I0921 11:28:31.089669    4097 checks.go:639] validating whether swap is enabled or not
I0921 11:28:31.089763    4097 checks.go:370] validating the presence of executable crictl
I0921 11:28:31.089799    4097 checks.go:370] validating the presence of executable conntrack
I0921 11:28:31.089819    4097 checks.go:370] validating the presence of executable ip
I0921 11:28:31.089843    4097 checks.go:370] validating the presence of executable iptables
I0921 11:28:31.089870    4097 checks.go:370] validating the presence of executable mount
I0921 11:28:31.089897    4097 checks.go:370] validating the presence of executable nsenter
I0921 11:28:31.089919    4097 checks.go:370] validating the presence of executable ebtables
I0921 11:28:31.089954    4097 checks.go:370] validating the presence of executable ethtool
I0921 11:28:31.089977    4097 checks.go:370] validating the presence of executable socat
I0921 11:28:31.089996    4097 checks.go:370] validating the presence of executable tc
I0921 11:28:31.090011    4097 checks.go:370] validating the presence of executable touch
I0921 11:28:31.090035    4097 checks.go:516] running all checks
I0921 11:28:31.103935    4097 checks.go:401] checking whether the given node name is valid and reachable using net.LookupHost
I0921 11:28:31.105638    4097 checks.go:605] validating kubelet version
I0921 11:28:31.162593    4097 checks.go:130] validating if the "kubelet" service is enabled and active
I0921 11:28:31.176512    4097 checks.go:203] validating availability of port 10250
I0921 11:28:31.176737    4097 checks.go:280] validating the existence of file /etc/kubernetes/pki/ca.crt
I0921 11:28:31.176765    4097 checks.go:430] validating if the connectivity type is via proxy or direct
I0921 11:28:31.176803    4097 checks.go:329] validating the contents of file /proc/sys/net/bridge/bridge-nf-call-iptables
I0921 11:28:31.176849    4097 checks.go:329] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0921 11:28:31.176883    4097 join.go:532] [preflight] Discovering cluster-info
I0921 11:28:31.176917    4097 token.go:80] [discovery] Created cluster-info discovery client, requesting info from "172.31.46.220:6443"
I0921 11:28:31.187676    4097 token.go:118] [discovery] Requesting info from "172.31.46.220:6443" again to validate TLS against the pinned public key
I0921 11:28:31.194531    4097 token.go:135] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.46.220:6443"
I0921 11:28:31.194600    4097 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0921 11:28:31.194622    4097 join.go:546] [preflight] Fetching init configuration
I0921 11:28:31.194629    4097 join.go:592] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0921 11:28:31.201909    4097 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0921 11:28:31.205146    4097 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0921 11:28:31.208370    4097 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0921 11:28:31.208595    4097 interface.go:432] Looking for default routes with IPv4 addresses
I0921 11:28:31.208617    4097 interface.go:437] Default route transits interface "enX0"
I0921 11:28:31.208751    4097 interface.go:209] Interface enX0 is up
I0921 11:28:31.208803    4097 interface.go:257] Interface "enX0" has 2 addresses :[172.31.36.212/20 fe80::75:41ff:fea5:afb1/64].
I0921 11:28:31.208819    4097 interface.go:224] Checking addr  172.31.36.212/20.
I0921 11:28:31.208829    4097 interface.go:231] IP found 172.31.36.212
I0921 11:28:31.208840    4097 interface.go:263] Found valid IPv4 address 172.31.36.212 for interface "enX0".
I0921 11:28:31.208849    4097 interface.go:443] Found active IP 172.31.36.212
I0921 11:28:31.215002    4097 preflight.go:104] [preflight] Running configuration dependant checks
I0921 11:28:31.215028    4097 controlplaneprepare.go:225] [download-certs] Skipping certs download
```

Step5: Run the given command to verify cluster creation

kubectl get nodes

```
ubuntu@ip-172-31-46-220:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-36-212   Ready    <none>          47s   v1.29.0
ip-172-31-46-220   Ready    control-plane   16m   v1.29.0
ip-172-31-47-26    Ready    <none>          29s   v1.29.0
```

ATHARVA PRABHU **EXPERIMENT 4**
D15A 43

Step 1: Deploying Your Application on Kubernetes
1.1 Set up Kubernetes Cluster

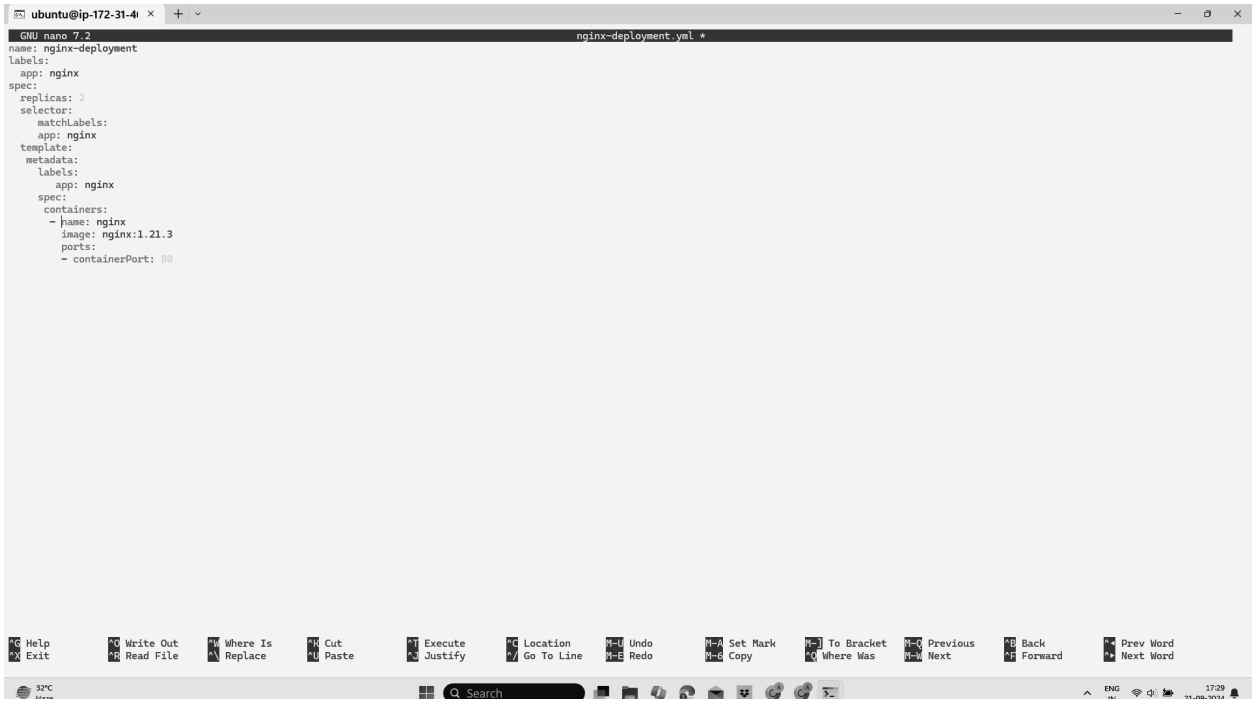
1. If you haven't already set up a Kubernetes cluster (e.g., with kubeadm), use minikube or any managed Kubernetes service (like EKS, GKE, etc.) to get a cluster running.

2. Once your cluster is ready, verify the nodes:

kubectl get nodes

```
ubuntu@ip-172-31-46-220:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-36-212   Ready    <none>          47s   v1.29.0
ip-172-31-46-220   Ready    control-plane   16m   v1.29.0
ip-172-31-47-26    Ready    <none>          29s   v1.29.0
```

Step 2: Create the Deployment YAML file
a) Create the YAML file: Use a text editor to create a file named nginx-deployment.yaml Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).



Step 3: Create the Service YAML File
a) Create the YAML File: Create another file named nginx-service.yaml Add the Service Configuration: Copy and paste the following YAML content into the file given below

Step 4: Apply the YAML Files a) Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files. Verify the Deployment: Check the status of your Deployment,Pods and Services. Describe the deployment(Extra)

```
ubuntu@ip-172-31-46-220:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
```

```
ubuntu@ip-172-31-46-220:~$ kubectl apply -f nginx-service.yaml
service/nginx-server created
```

Step 5: Ensure Service is Running 6.1 Verify Service: Run the following command to check the services running in your cluster: Kubectl get deployment Kubectl get pods kubectl get service

```
ubuntu@ip-172-31-46-220:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           7m27s
```

```
ubuntu@ip-172-31-46-220:~$ kubectl get services
NAME           TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP      10.96.0.1        <none>        443/TCP        85m
nginx-server   LoadBalancer   10.111.218.213   <pending>     80:30798/TCP   110s
```

Step 6: Forward the Service Port to Your Local Machine kubectl port-forward allows you to forward a port from your local machine to a port on a service running in the Kubernetes cluster.

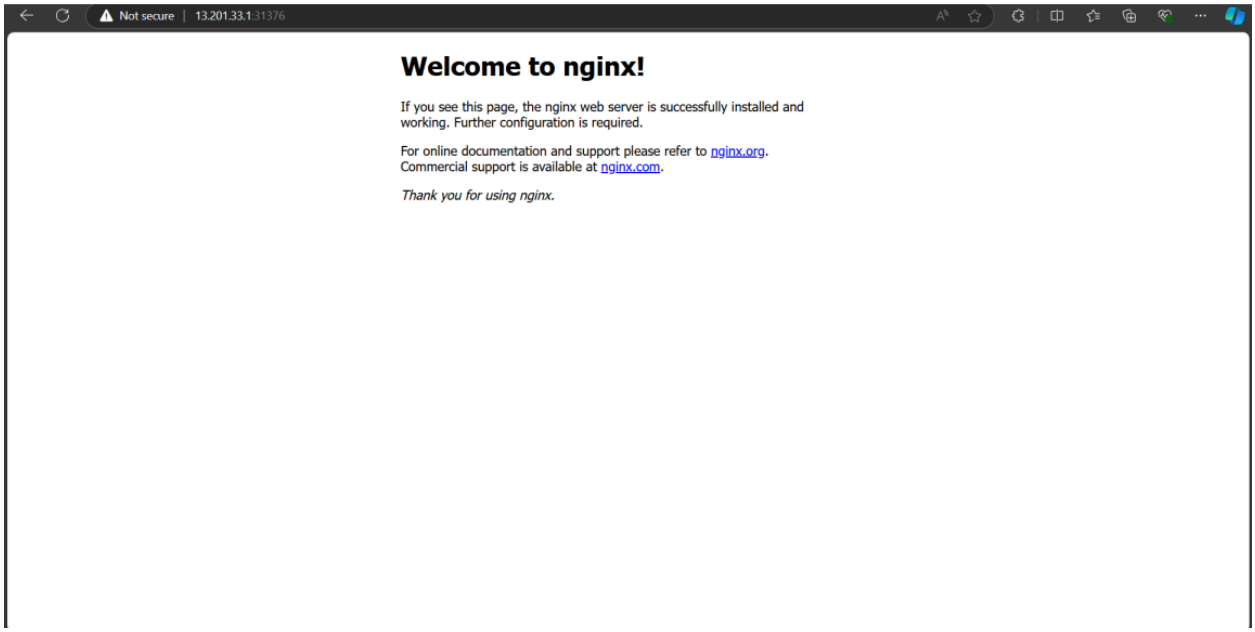
1. Forward the Service Port: Use the following command to forward a local port to the service's target port. kubectl port-forward service/ :
This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

```
ubuntu@ip-172-31-46-220:~$ kubectl describe deployments
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Sat, 21 Sep 2024 12:30:54 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.16
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-854bc88786 (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  11m   deployment-controller  Scaled up replica set nginx-deployment-854bc88786 to 3
```

2. This means port forwarding is now active, and any traffic to localhost:8080 will be routed to the nginx-service on port 80.

```
ubuntu@ip-172-31-46-220:~$ kubectl port-forward service/nginx-server 8080:80
```

Step 7:
Access the Application Locally
1. Open a Web Browser: Now open your web browser and go to the following URL: http://localhost:8080 You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080. In case the port 8080 is unavailable, try using a different port like 8081

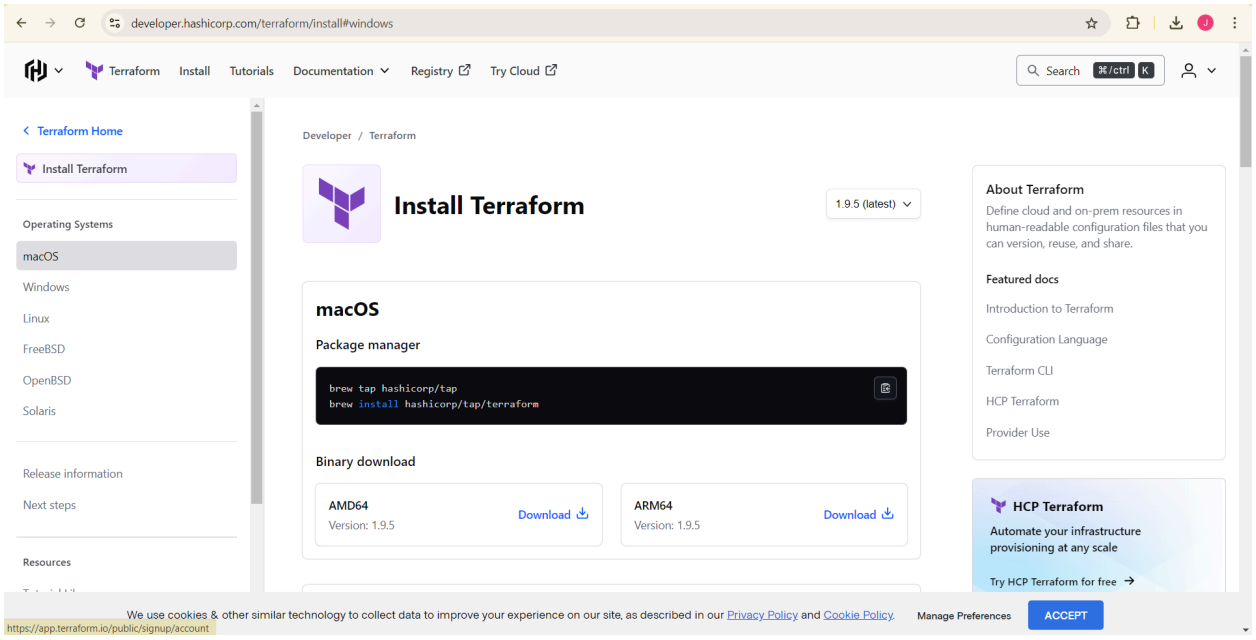

Not secure | 13.201.33.1:31376
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.

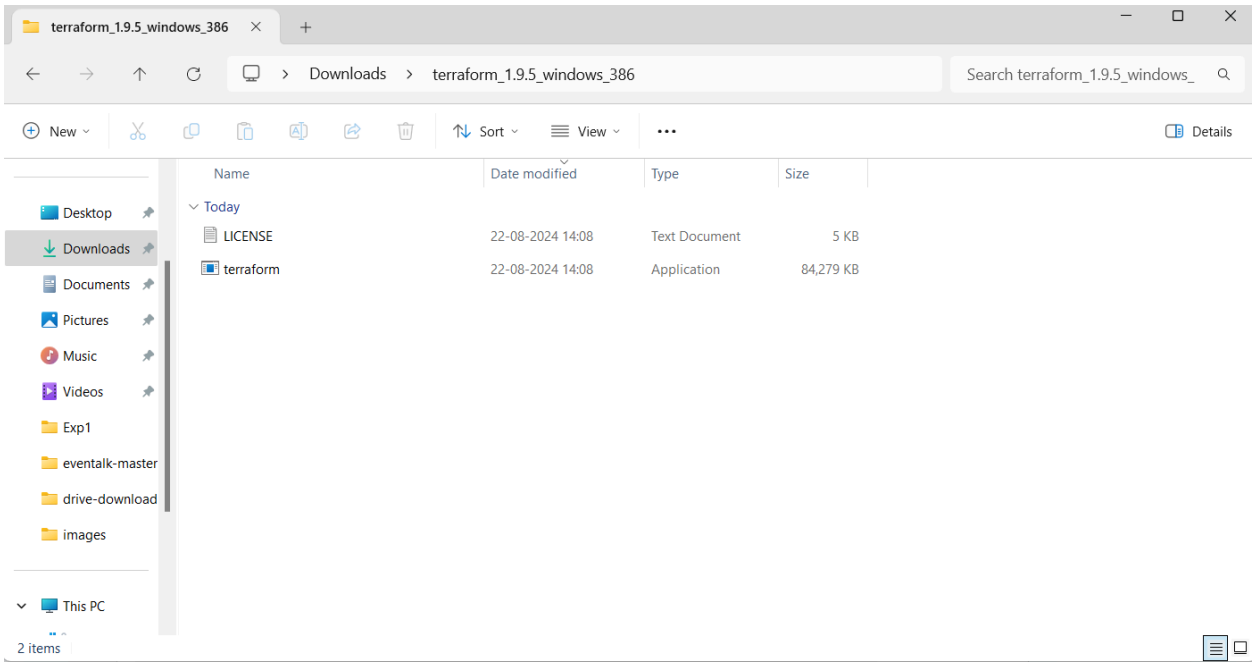
Advance devops Exp:5

Atharva Prabhu
D15 A-43

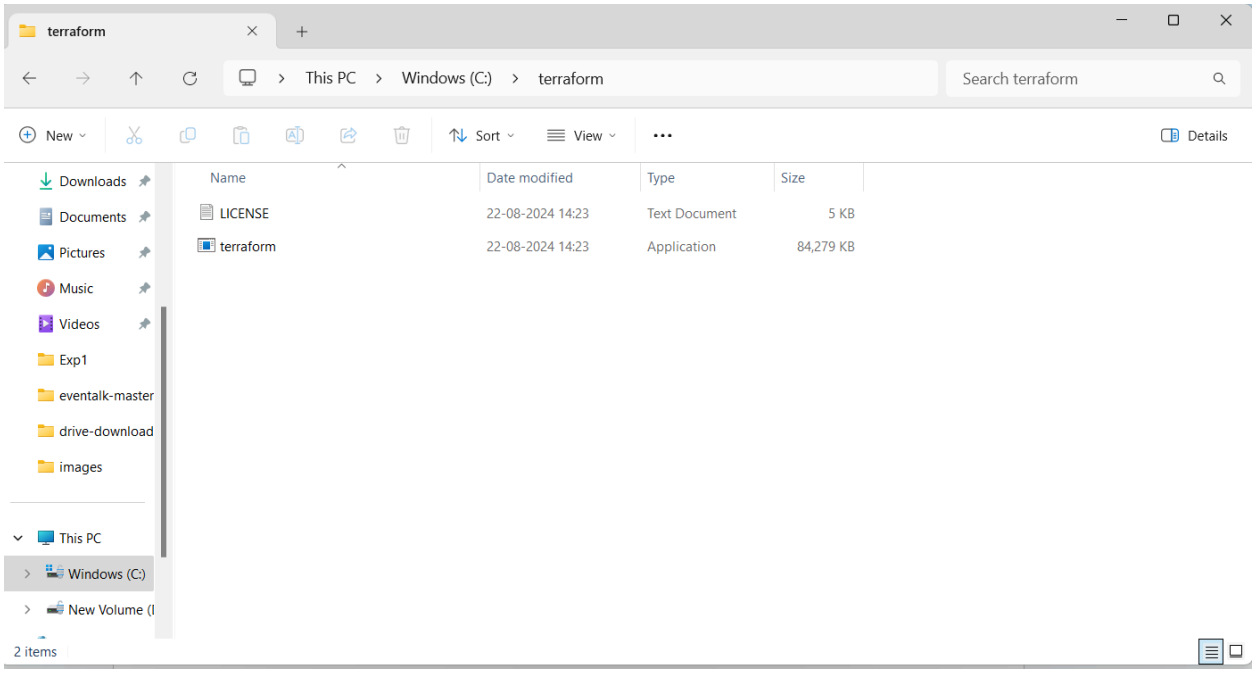
Aim: To understand terraform lifecycle, core concepts/terminologies and install it on a linux machine and windows

Step1: Download Terraform from the official website

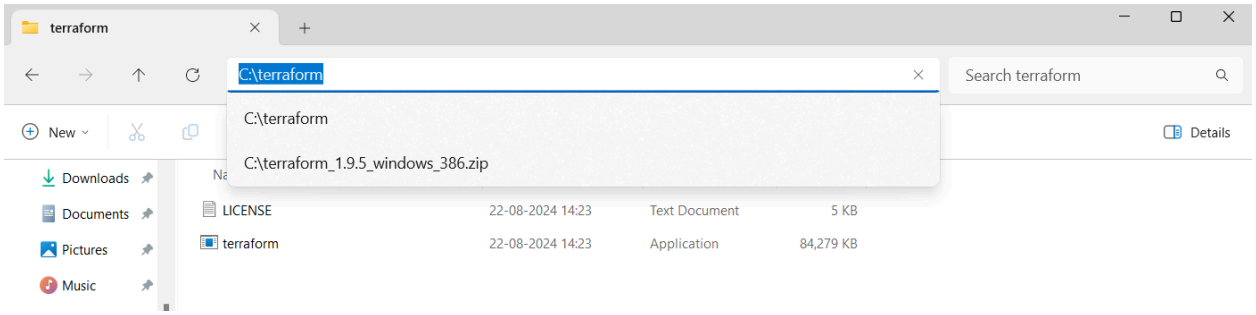




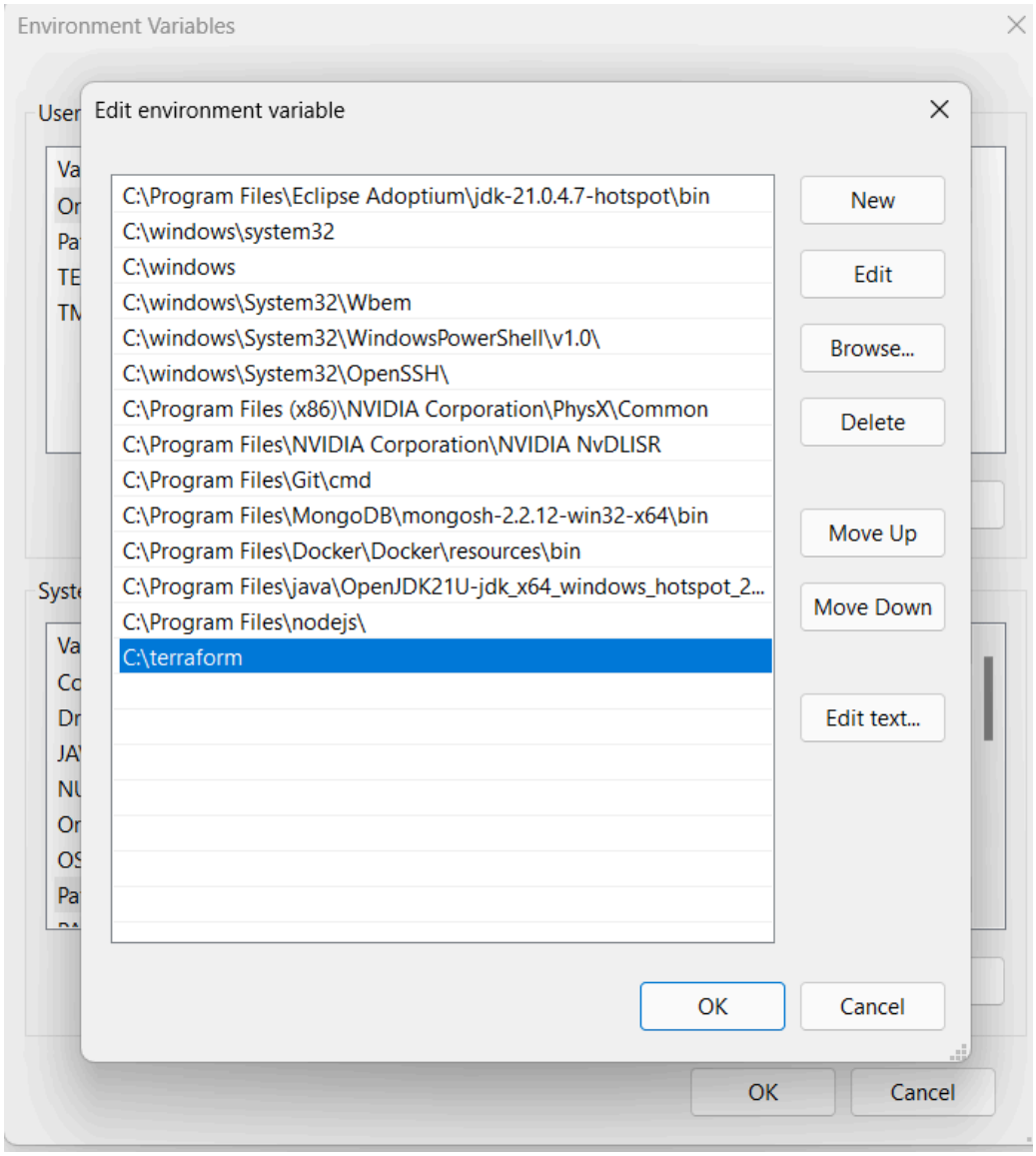
Step 2: Copy and extract Terraform from the downloads and paste it in the C drive



Step 3: Copy the file path to paste in the environment variables



Step 4: Set the environment variables for terraform



Step 5: Check weather the terraform is installed

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

C:\Users\navan>terraform --version
Terraform v1.9.5
on windows_386
```

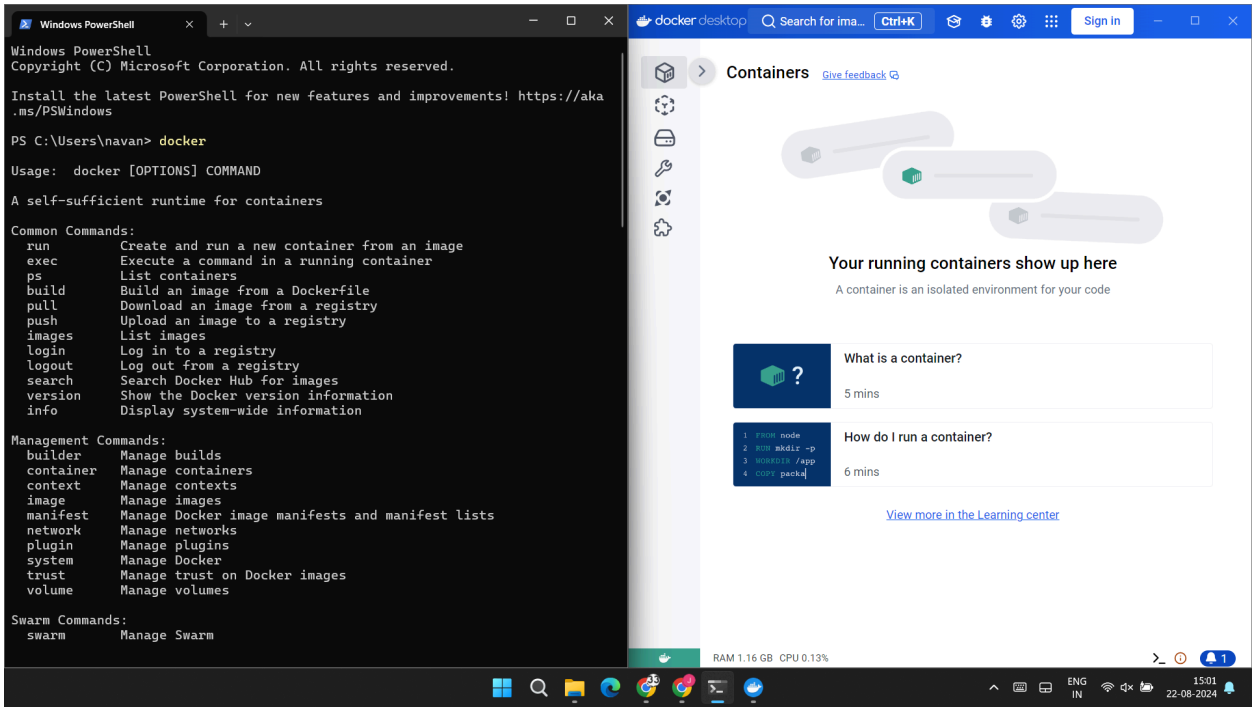
Advance Devops Experiment 6

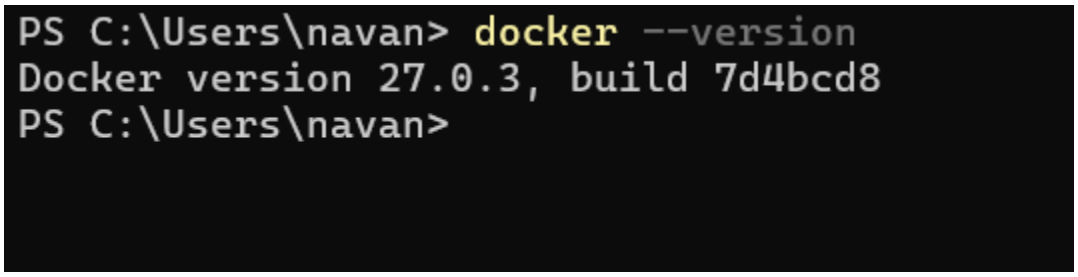
Atharva Prabhu

D15 A-43

Aim: Creating docker image using Terraform

Step 1: Install docker Desktop after installation check the functionality



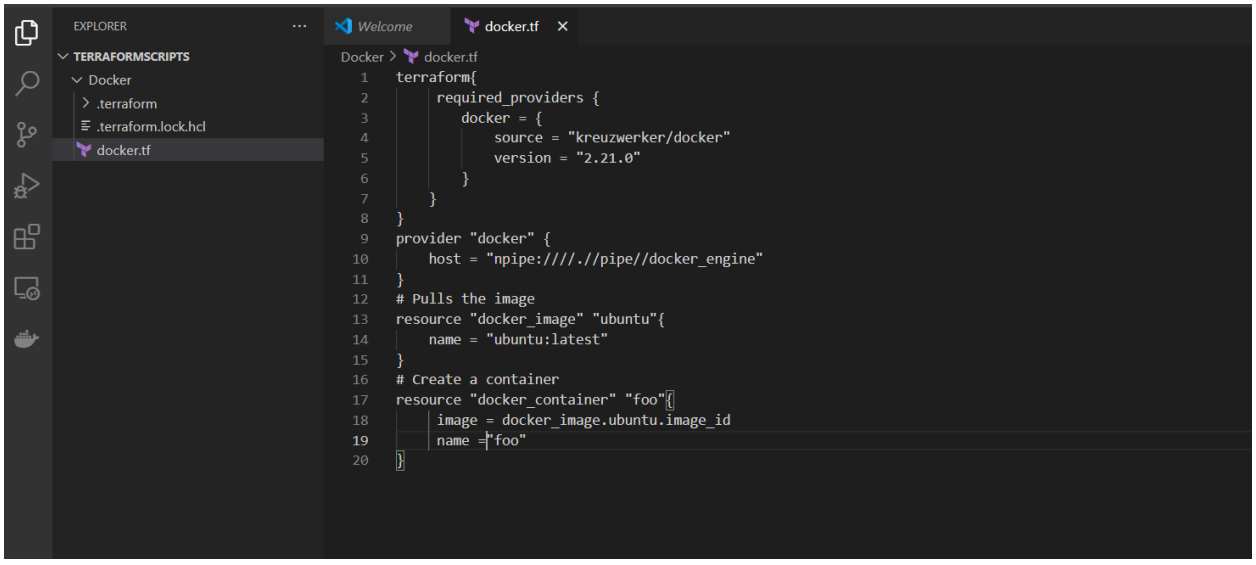


Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

Step 2: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using Atom editor and write the followingcontents into it to create a Ubuntu Linux container.

Script:

```
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_id
    name ="foo"
}
```



Step 3: Execute terraform init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

C:\Users\navan\Desktop\TerraformScripts\Docker>
```

Step 4: Execute Terraform plan to see the available resources

```
C:\Users\navan\Desktop\TerraformScripts\Docker>terraform plan

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
```

```
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

  # docker_image.ubuntu will be created
  + resource "docker_image" "ubuntu" {
      + id          = (known after apply)
      + image_id    = (known after apply)
      + latest      = (known after apply)
      + name        = "ubuntu:latest"
      + output      = (known after apply)
      + repo_digest = (known after apply)
    }

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

─────────────────────────────────────────────────────────────────────────────

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

C:\Users\navan\Desktop\TerraformScripts\Docker>
```

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

```
    }

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

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

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

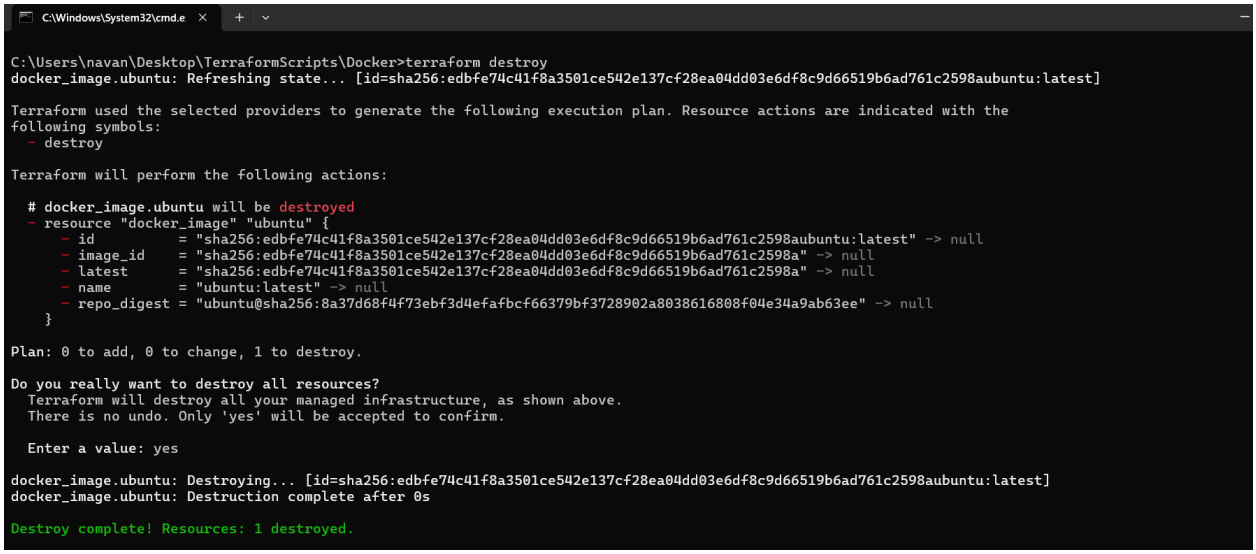
Docker images,before Executing Apply step:

```
C:\Users\navan\Desktop\TerraformScripts\Docker>docker images
REPOSITORY   TAG          IMAGE ID       CREATED       SIZE
ubuntu       latest       edbfe74c41f8   2 weeks ago   78.1MB
node         20-alpine    e2997a3fdff8   4 weeks ago   133MB
```

```
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28e
3e6df8c9d66519b6ad761c2598aubuntu:latest]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last
"terraform apply" which may have affected this plan:

  # docker_image.ubuntu has been deleted
  - resource "docker_image" "ubuntu" {
        id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6
2598aubuntu:latest"
      - image_id    = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6
2598a" -> null
        name        = "ubuntu:latest"
        # (2 unchanged attributes hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relev
attributes using ignore_changes, the following plan may include actions to undo or
respond to these changes.
```

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the ubuntu container.

```
C:\Users\navan\Desktop\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c259
tu:latest]

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

Terraform will perform the following actions:

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

Plan: 0 to add, 0 to change, 1 to destroy.

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

  Enter a value:
```



Docker images after executing destroy step

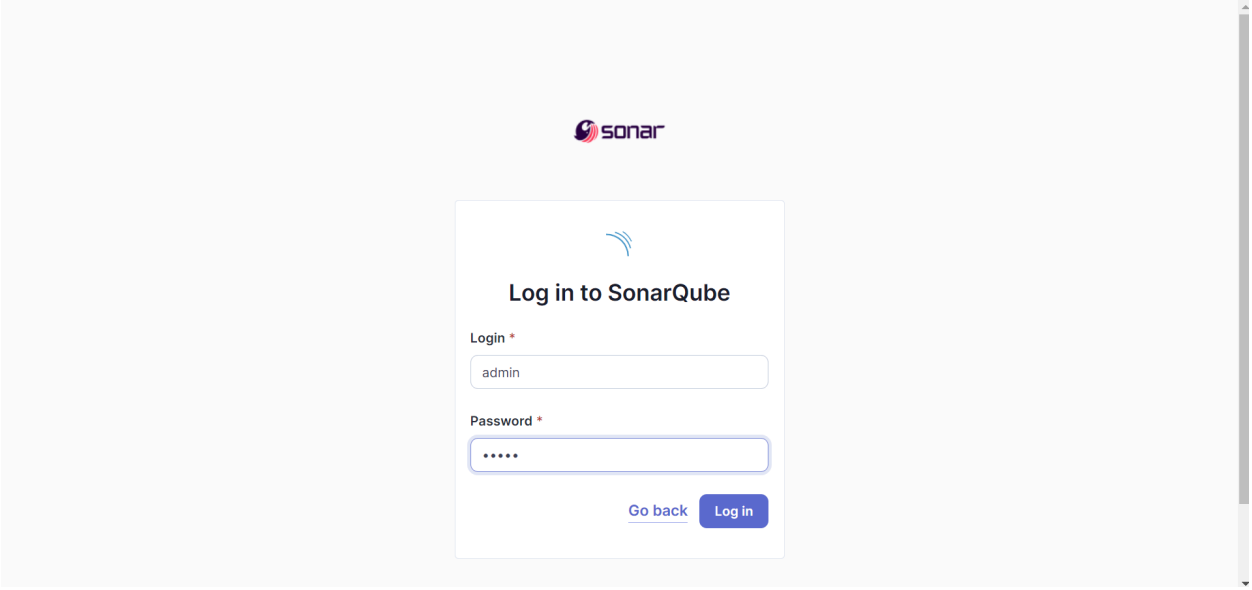
```
C:\Users\navan\Desktop\TerraformScripts\Docker>docker images
REPOSITORY   TAG          IMAGE ID       CREATED       SIZE
node         20-alpine    e2997a3fdff8   5 weeks ago   133MB
```

Atharva Prabhu EXPERIMENT 7
D15A 43
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

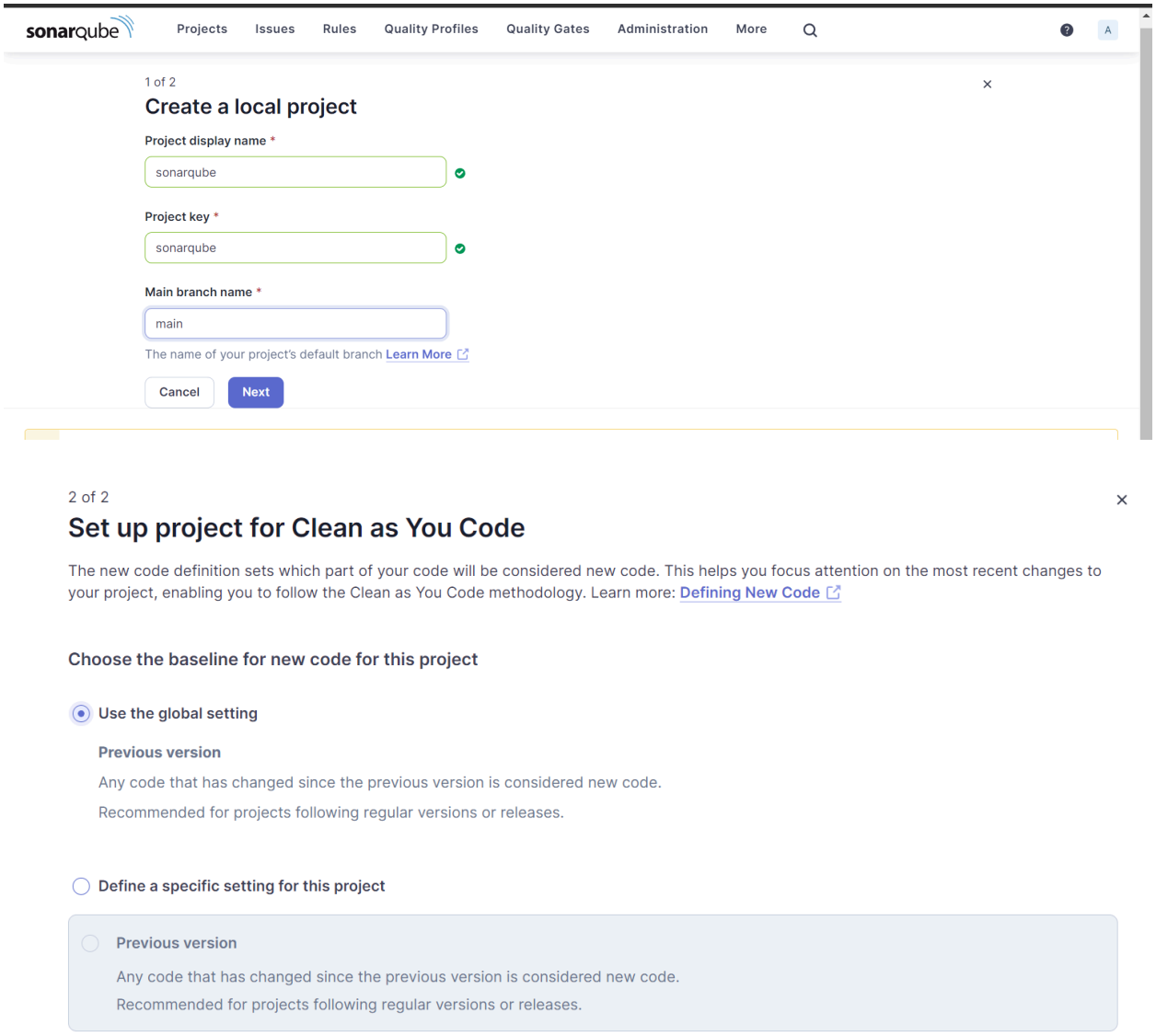
Step 1: Open Windows PowerShell and run the following command – docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest WARNING: Run the following command only once

```
C:\Users\Atharva\OneDrive\Desktop\docker>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
bc8a095e5c3155db49536b93661f49be9a4d3eff42237d7f162d1d67a01cd78f
```

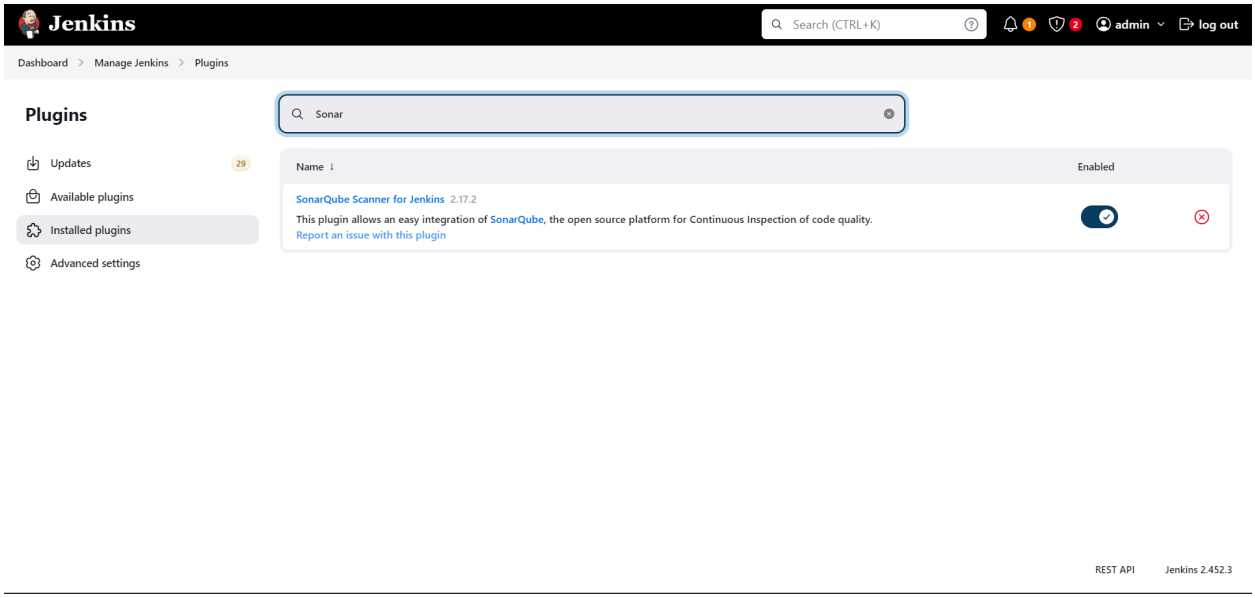
Step 2: Visit http://localhost:9000/ to open SonarQube. Login with username: admin and password: admin.



Step 3: Click on create a local project and name the project as sonarqube-test and key as sonarqube-test and click on the next button. In the next step select the “Use the global setting” option and click on create project.

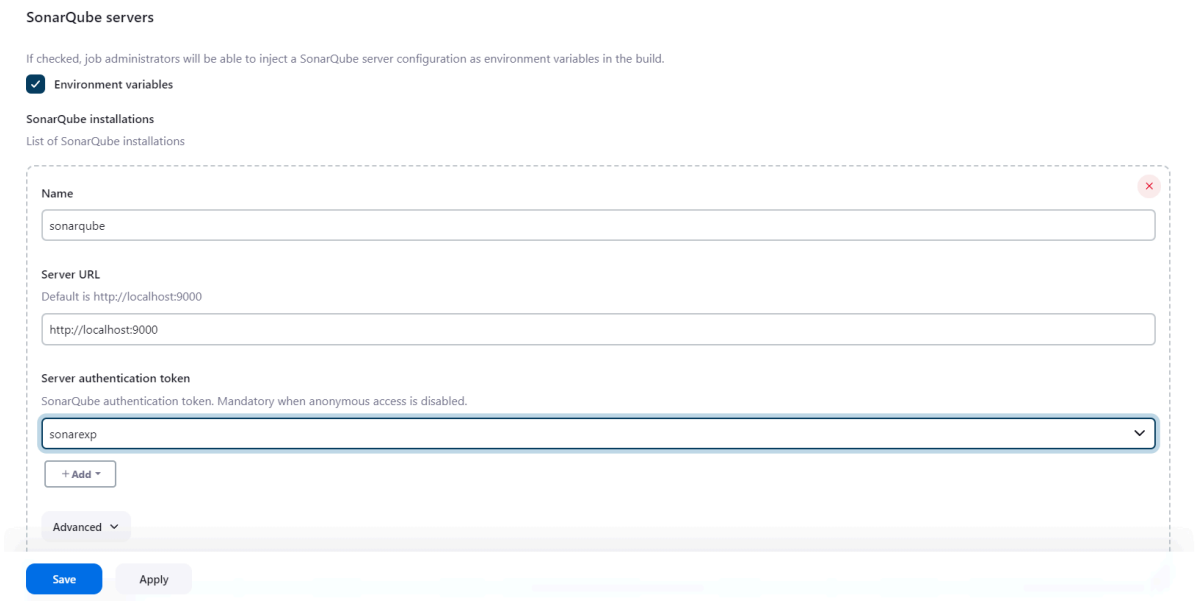


Step 4: Open Jenkins using http://localhost:8080/ and select Manage Jenkins, then select the Plugins and select available plugins from sidebar and search for SonarQube Scanner and install it. Once installed you can view the installed plugin in installed plugins section in sidebar.

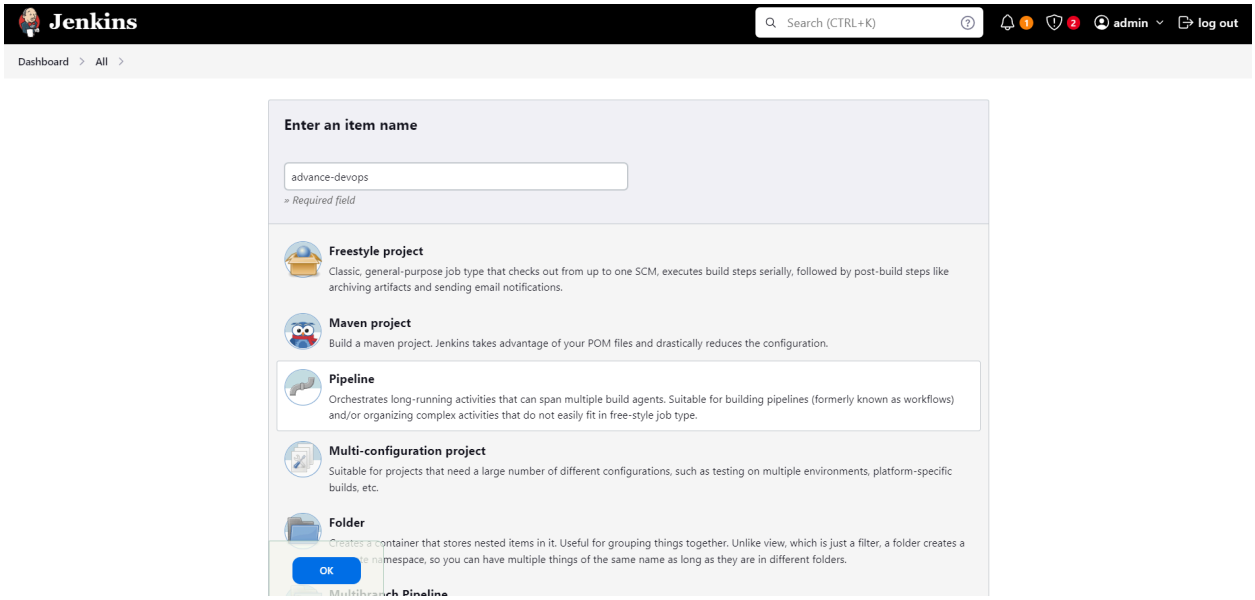


Step 5: Select Manage Jenkins, then select the System and then scroll down to SonarQube Server. Name the server as sonarqube and set the server url as http://localhost:9000/ then click on save.

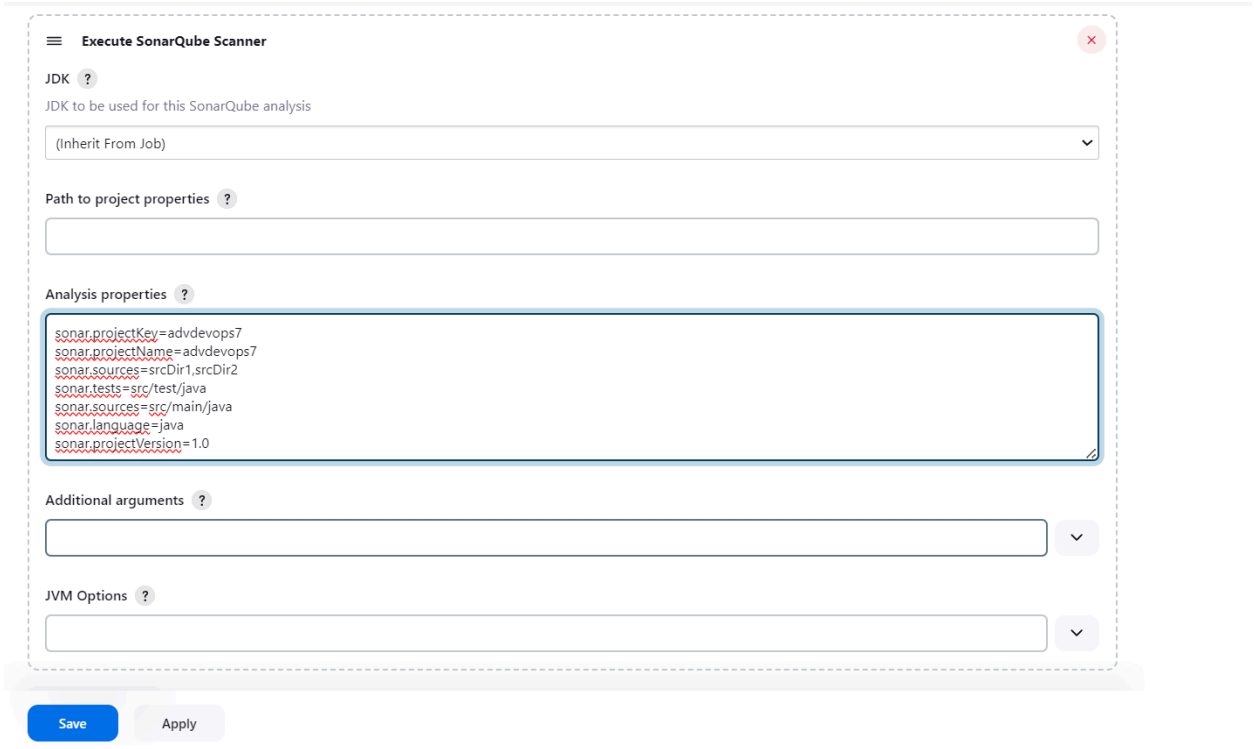
Step 6: Go to Jenkins Dashboard and select Manage Jenkins, then select the Tools and then scroll down to SonarQube Scanner installations. Name the sonarqube scanner as sonarqubescanner and select install automatically then click on save.



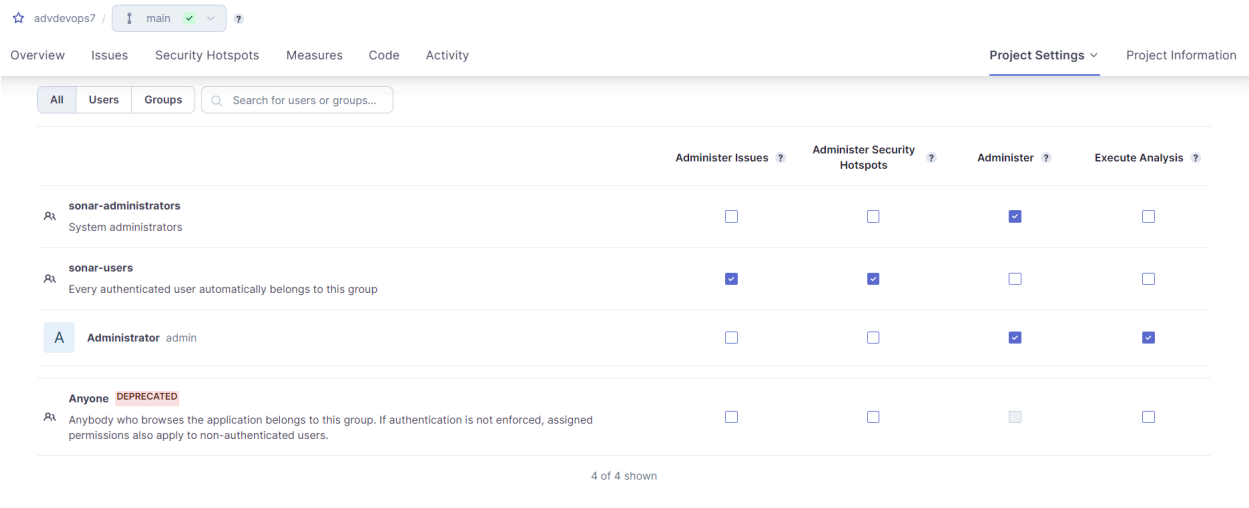
Step 7: Go to Jenkins dashboard and click on New Item and select Freestyle project and name it as SonarQube and then click on ok.



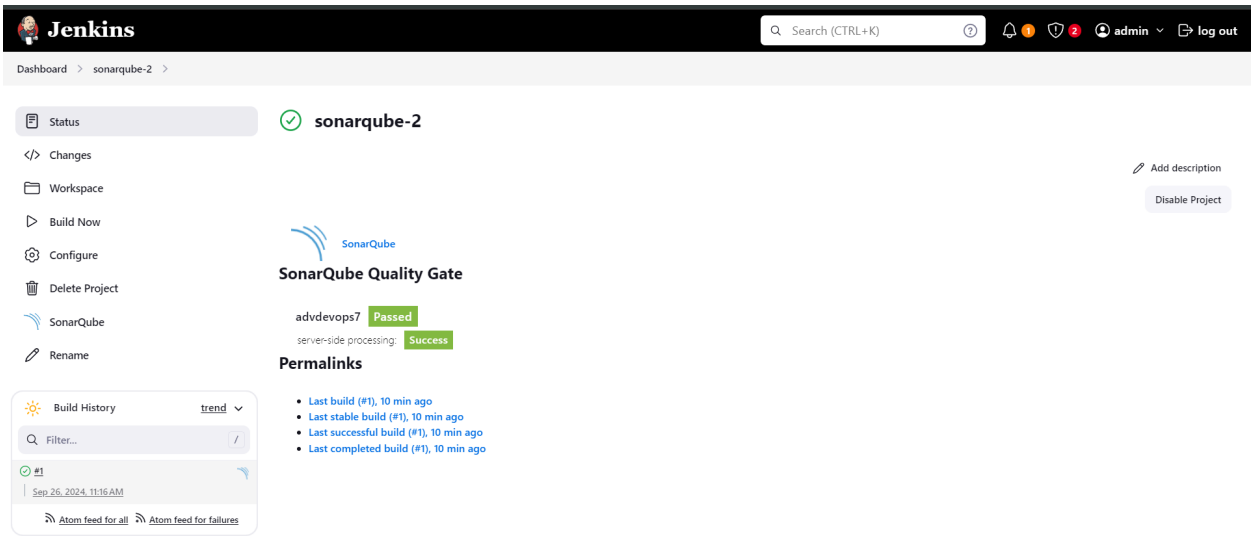
Step 8: For configuration, Select git and paste the following git repository in the repository url. https://github.com/shazforiot/MSBuild_firstproject This is a simple Hello world project Step 9: Under the Build steps select “Execute SonarQube Scanner” option and under Analysis Properties write the following - sonar.projectKey, sonar.login, sonar.password, sonar.hosturl. Then click on the save button.



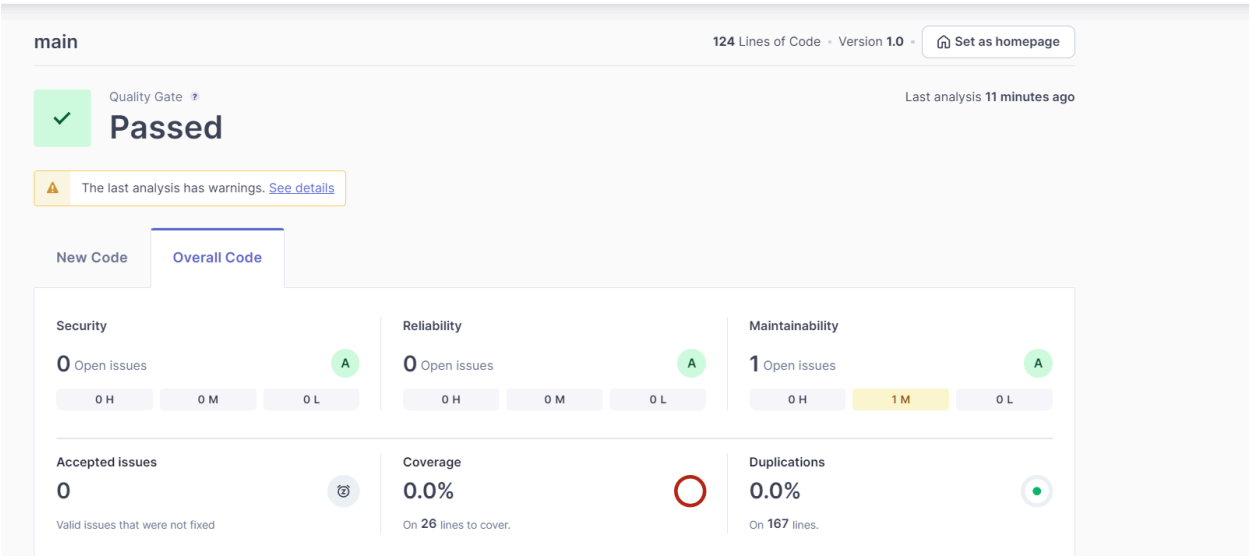
Step 10: Visit http://localhost:9000/admin/permissions and select the Users tab and for Administrator select the checkbox Execute Analysis.



Step 11: Now, come back to Jenkins and click on Build Now. The build is success.



Step 12: Visit the following URL to see the result - http://localhost:9000/dashboard?id=sonarqubetest&codeScope=overall


main
124 Lines of Code • Version 1.0 • Set as homepage
Quality Gate
Passed
Last analysis 11 minutes ago
The last analysis has warnings. See details
New Code
Overall Code
Security
0 Open issues
A
0 H
0 M
0 L
Reliability
0 Open issues
A
0 H
0 M
0 L
Maintainability
1 Open issues
A
0 H
1 M
0 L
Accepted issues
0
Valid issues that were not fixed
Coverage
0.0%
On 26 lines to cover.
Duplications
0.0%
On 167 lines.

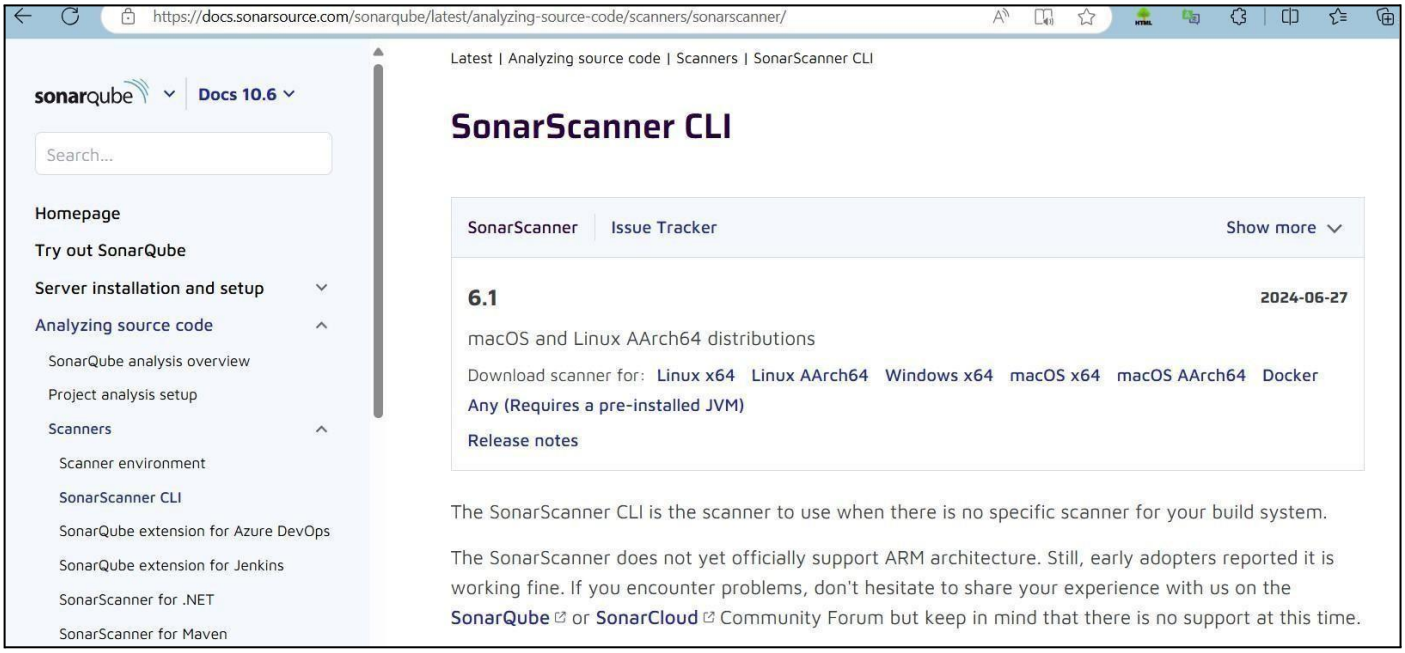
ADVANCE DEVOPS EXP 8

Name:Atharva Prabhu
Class:D15A

Roll No:43

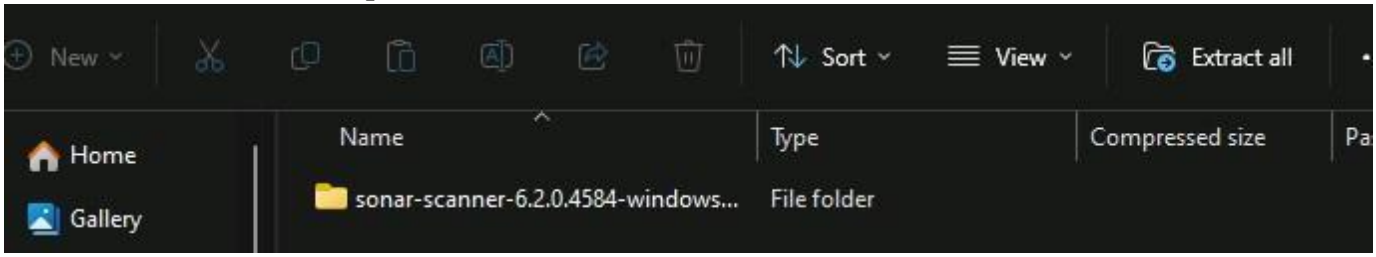
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan

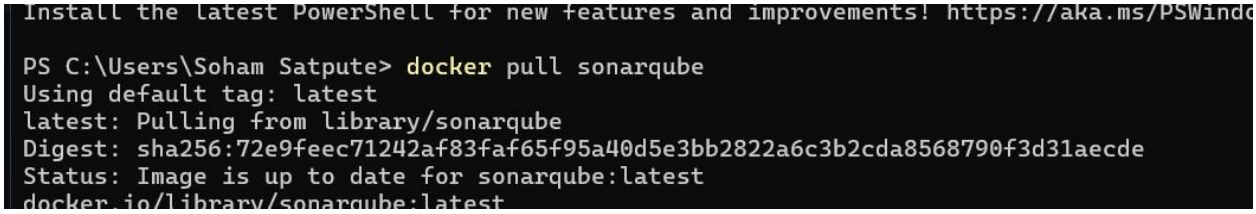


ner/ Visit this link and download the sonarqube scanner CLI.

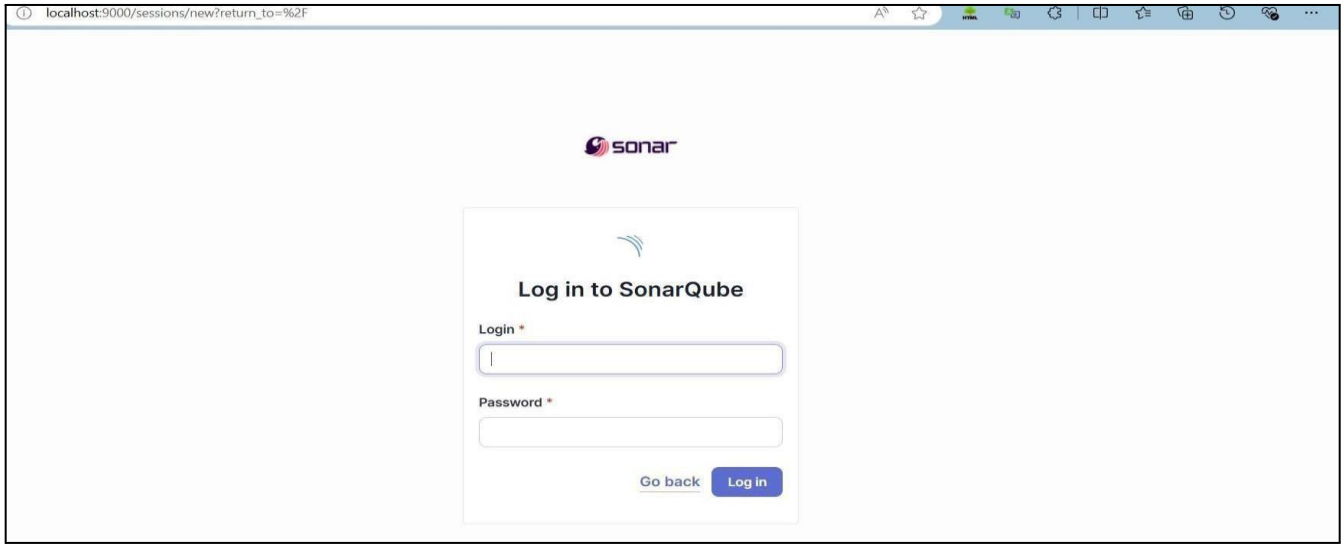
Extract the downloaded zip file in a folder.



1. Install sonarqube image Command: **docker pull sonarqube**

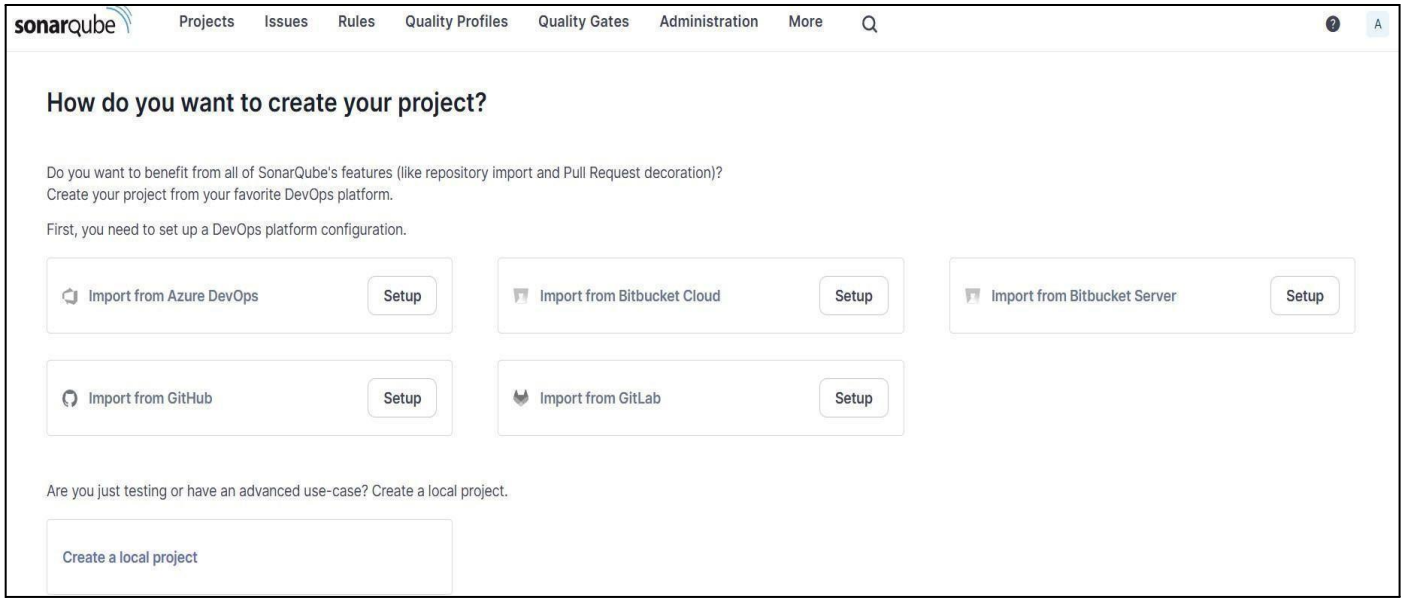

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindo
PS C:\Users\Soham Satpute> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

2. Once the container is up and running, you can check the status of

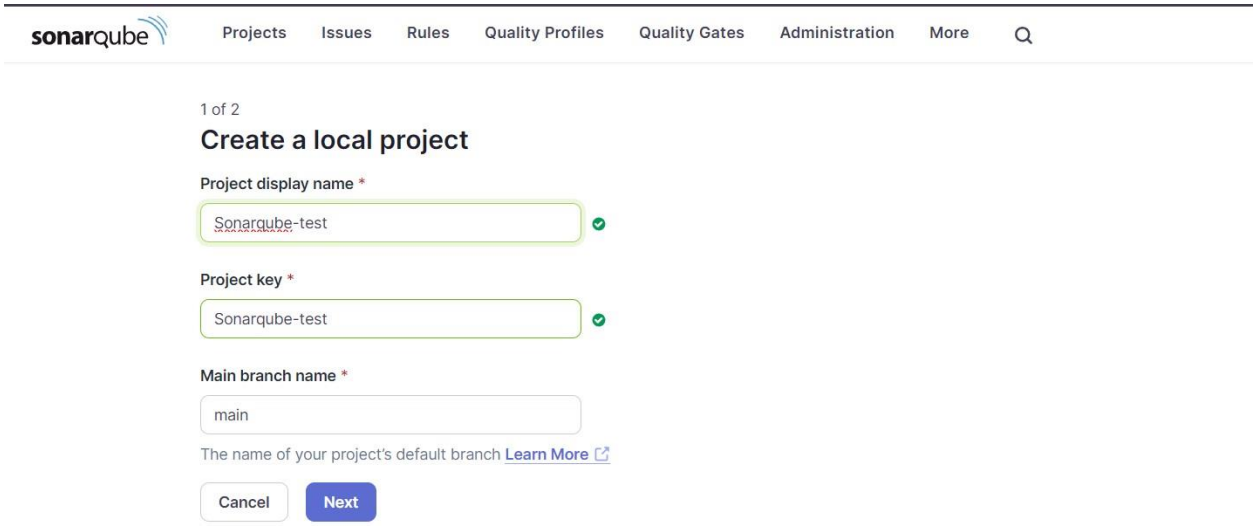


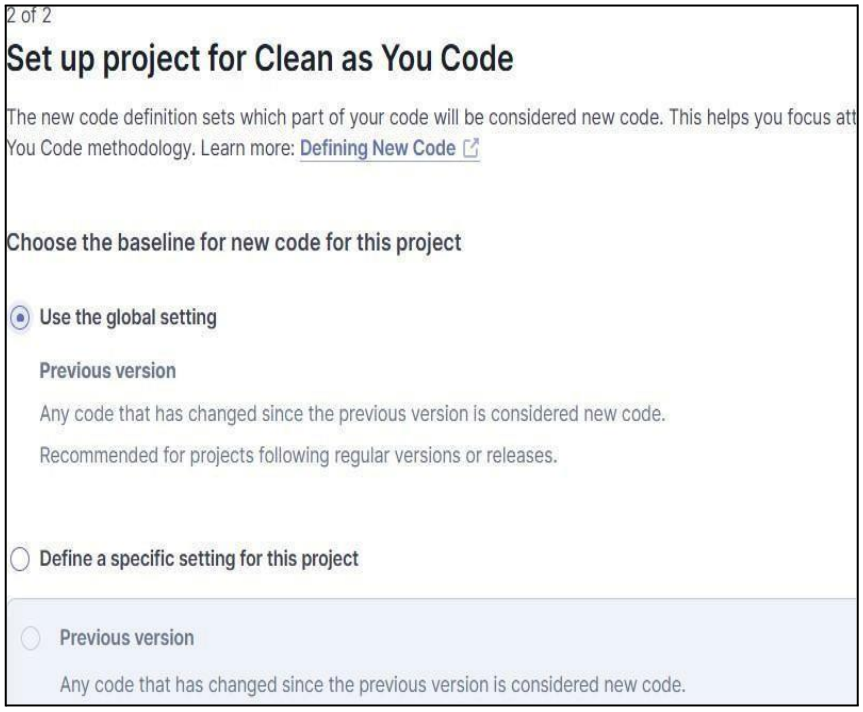
SonarQube at localhost port 9000.

3. Login to SonarQube using username admin and password admin.

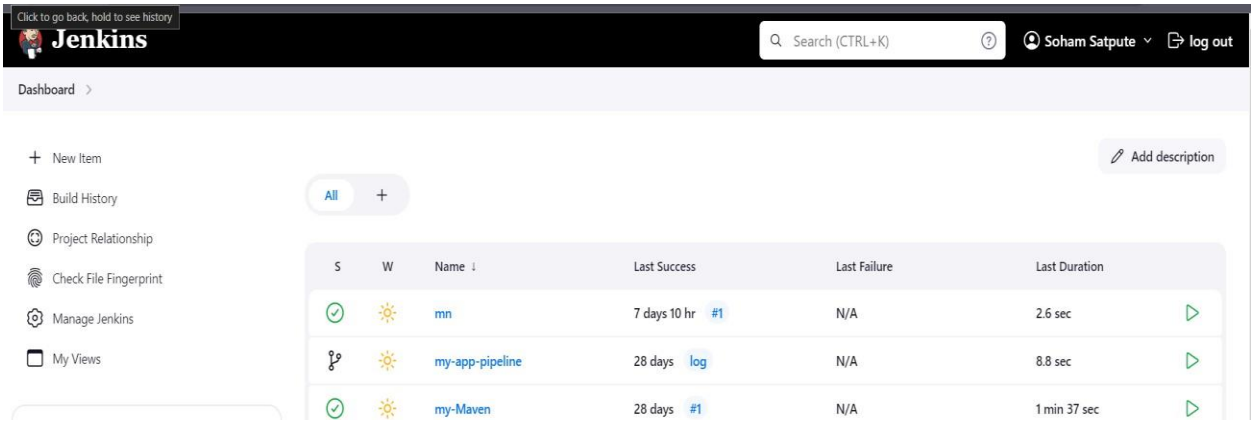


4. Create a manual project in SonarQube with the name sonarqube

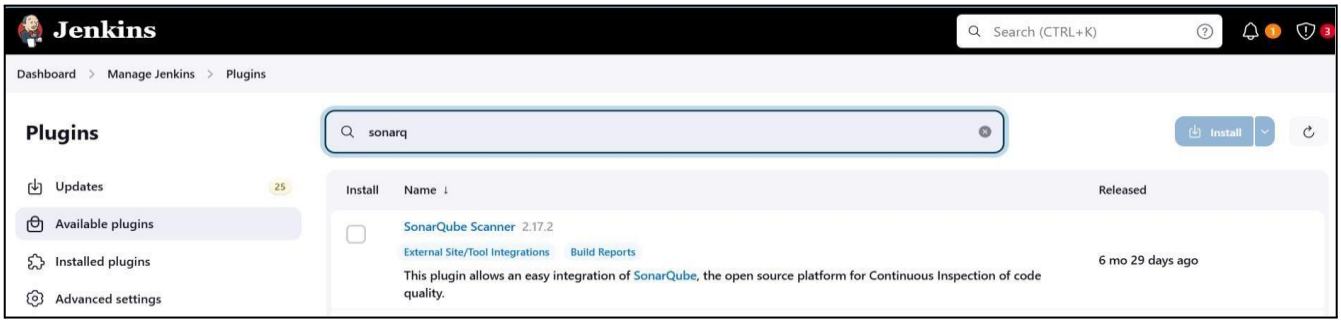


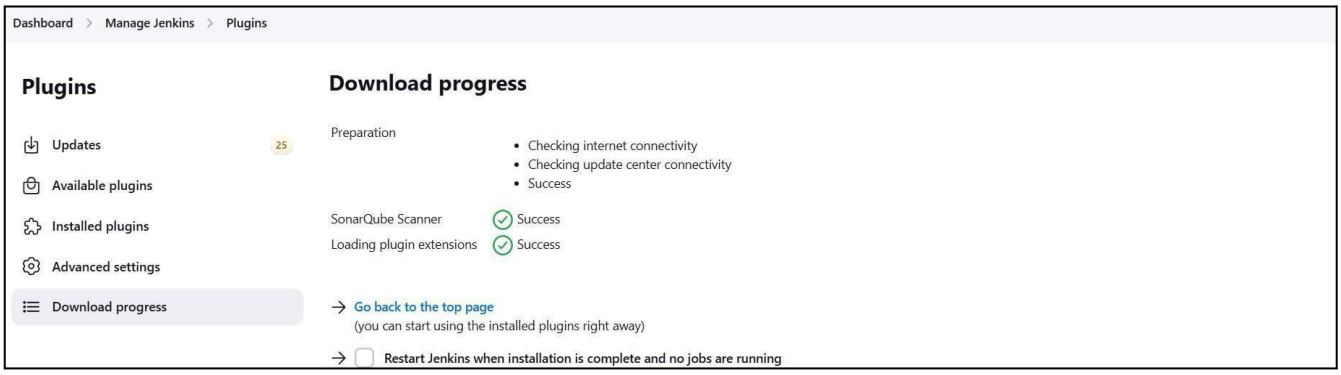


5. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



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





7. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter
the details.

Enter the Server Authentication token if needed.

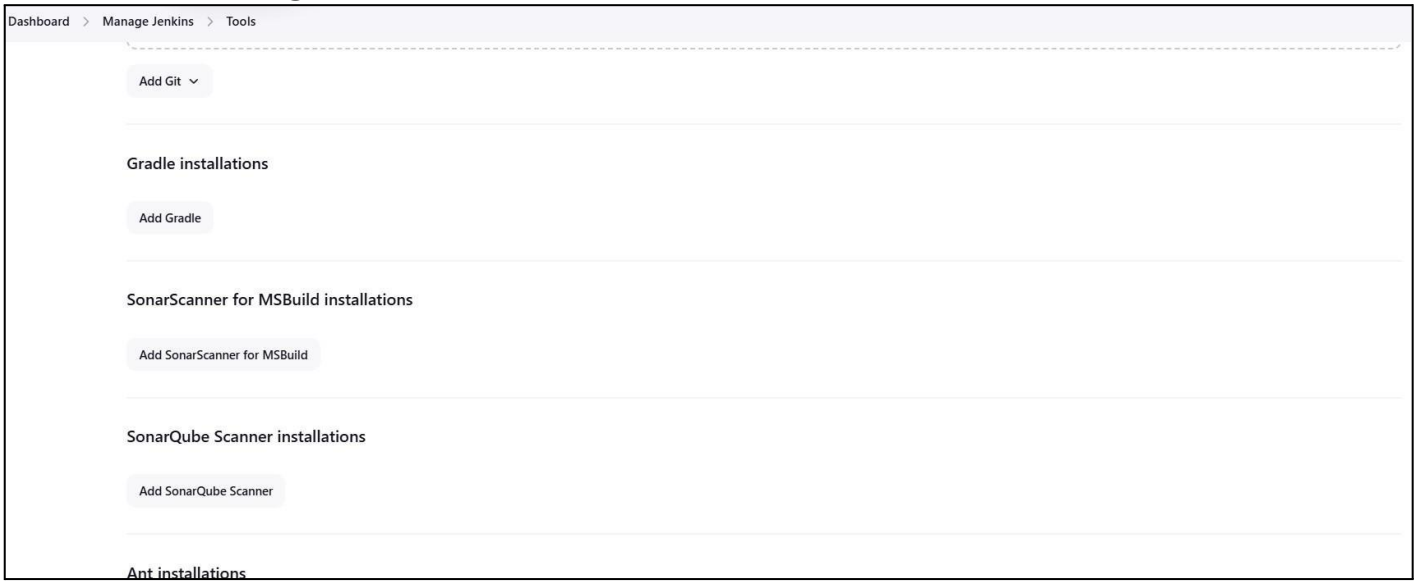
In SonarQube installations: Under **Name** add <project name of sonarqube> for me **adv_devops_7_sonarqube**

In **Server URL** Default is **http://localhost:9000**

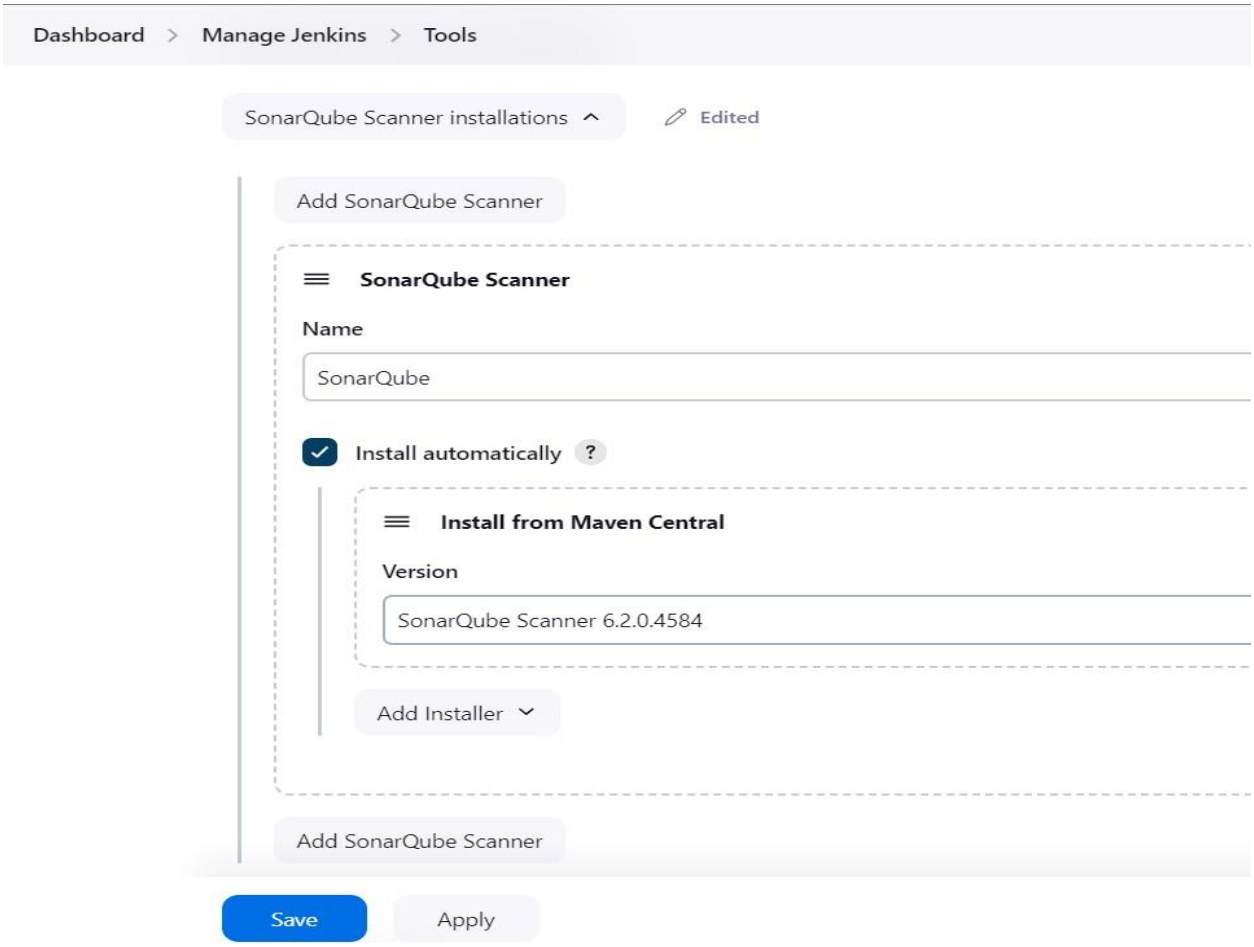
8. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

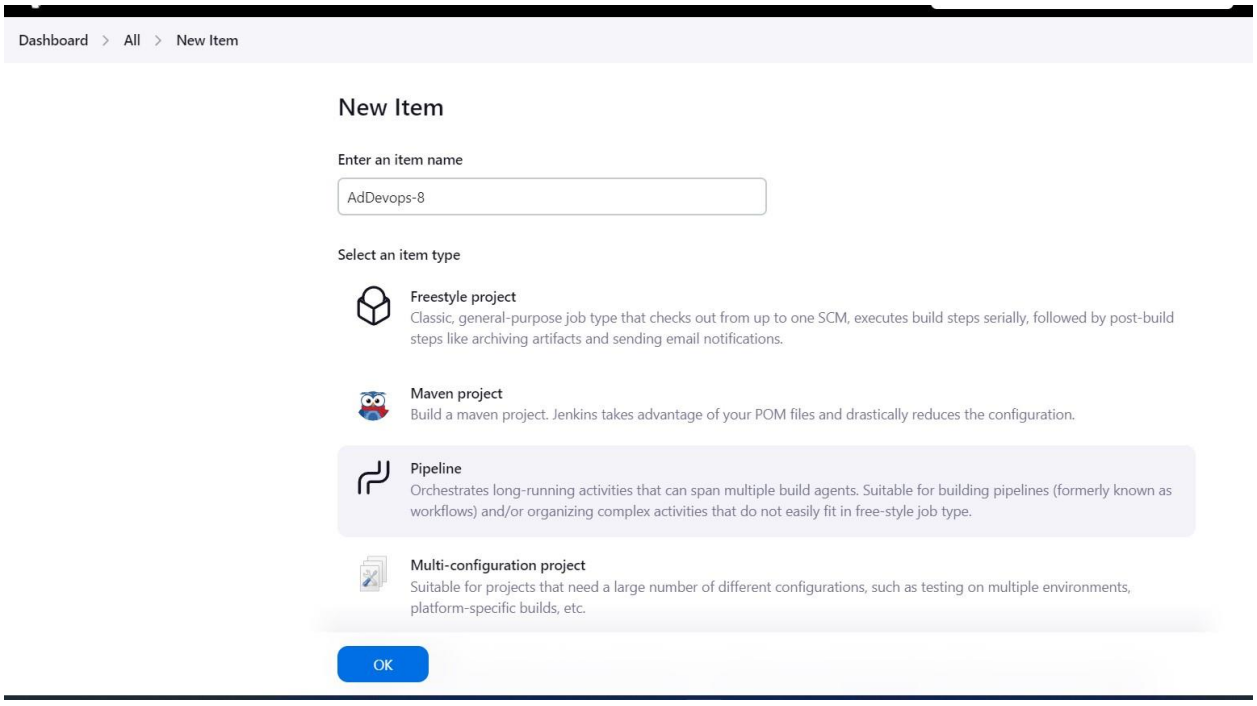
Dashboard > Manage Jenkins > Tools



Check the “Install automatically” option. → Under name any name as identifier → Check



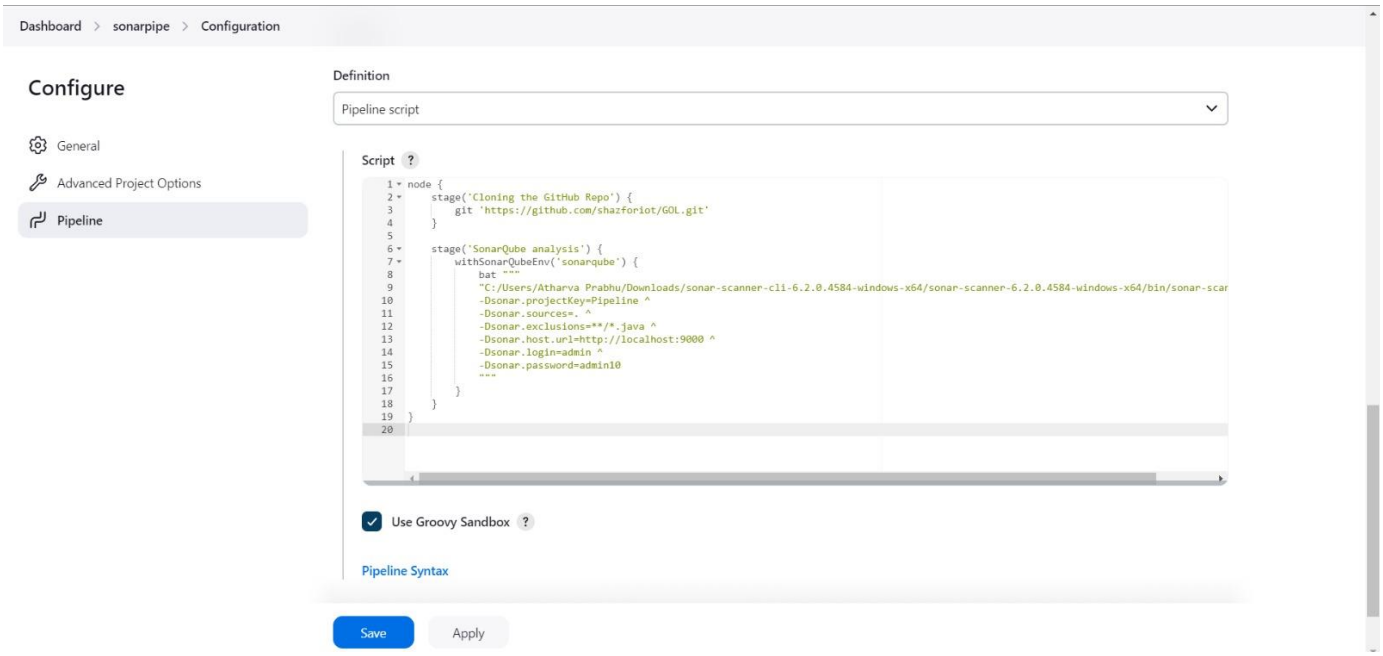
9. After configuration, create a New Item → choose a pipeline project.

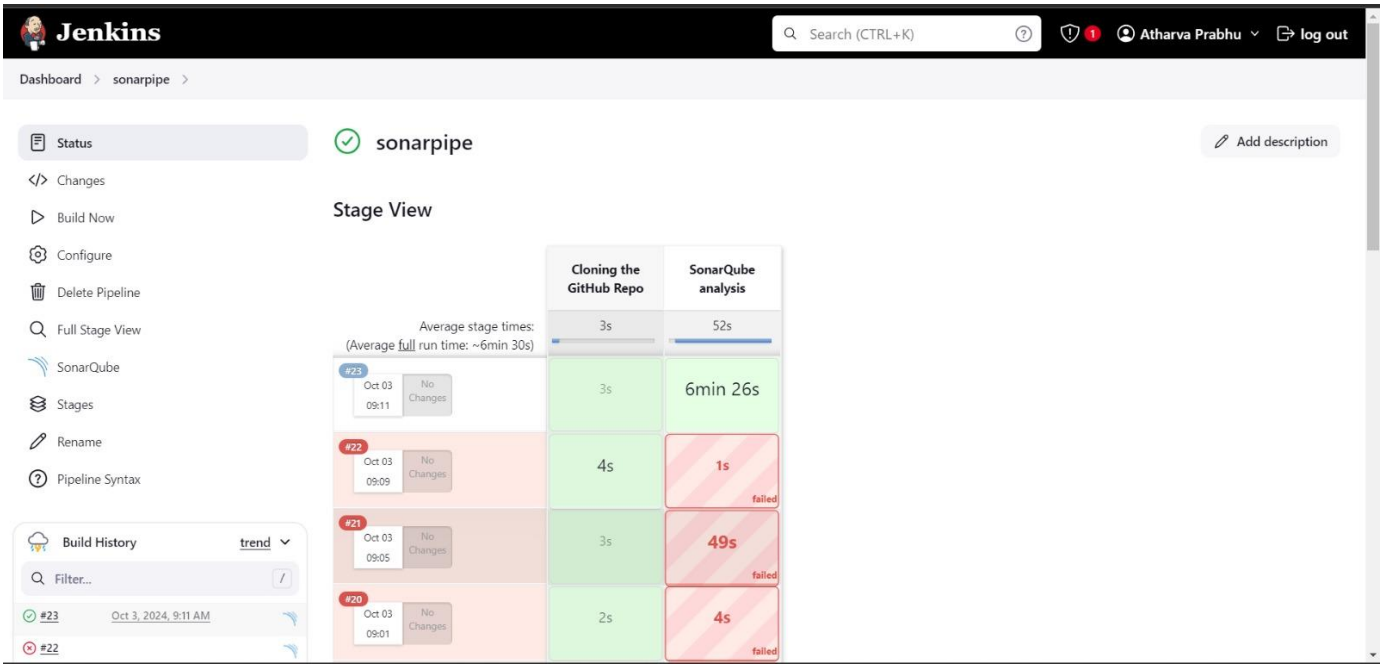


10. Under Pipeline script, enter the following:

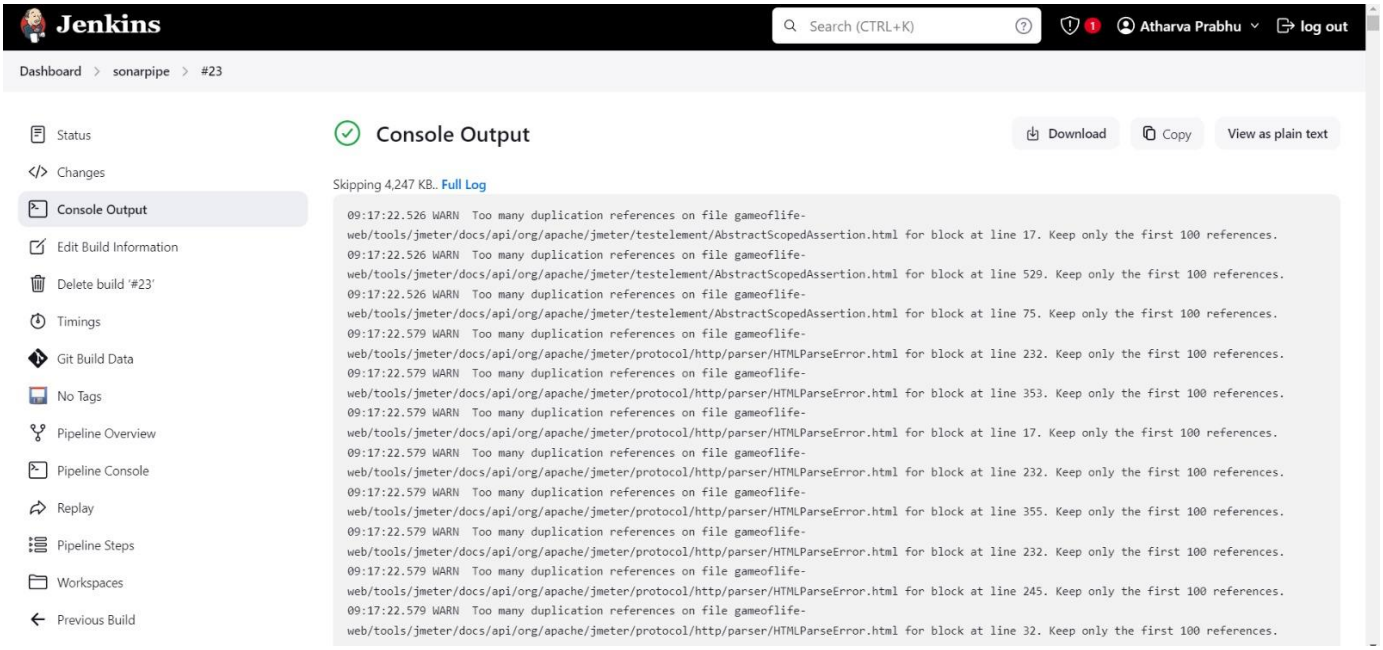
```
        node {
    stage('Cloning the GitHub Repo') { git
      'https://github.com/shazforiot/GOL.git'
    } stage('SonarQube

    analysis') {

      withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenk
      i ns>') { sh """
            <PATH_TO_SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
            -D sonar.login=<SonarQube_USERNAME> \
            -D sonar.password=<SonarQube_PASSWORD> \
            -D sonar.projectKey=<Project_KEY> \
            -D sonar.exclusions=vendor/**,resources/**,**/*.java \
            -D sonar.host.url=<SonarQube_URL>(default: http://localhost:9000/)
          """
      }
    }
}
```

It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

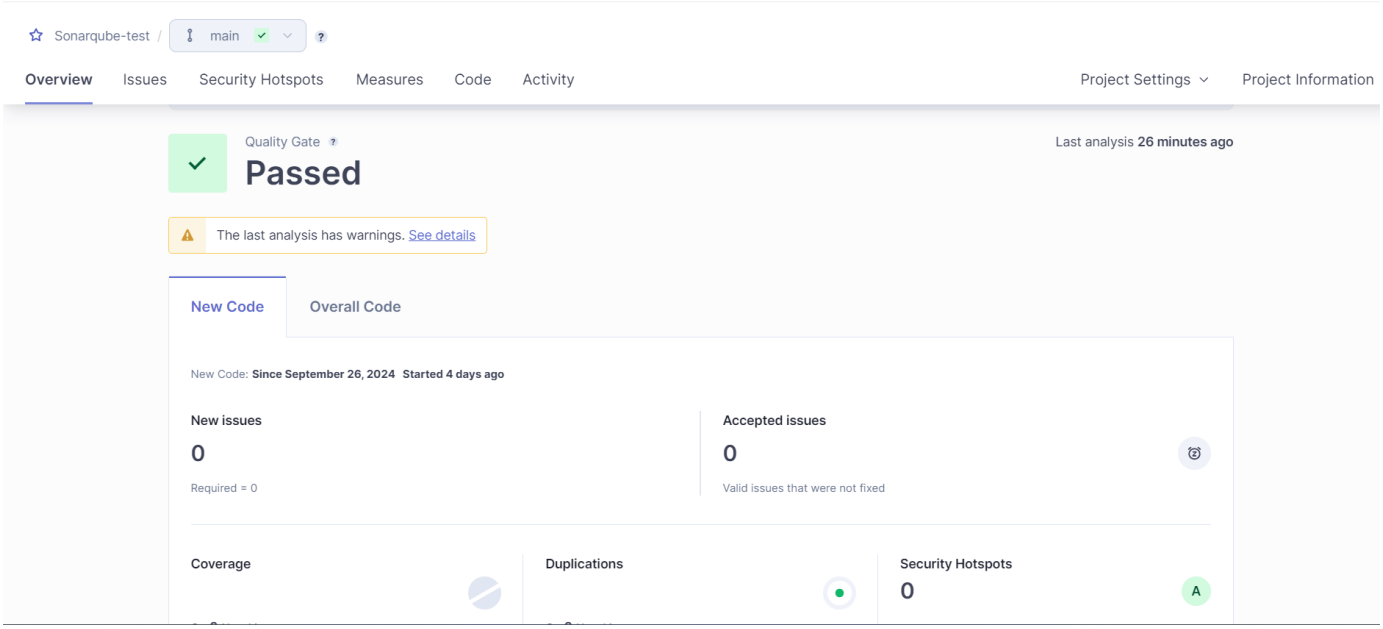




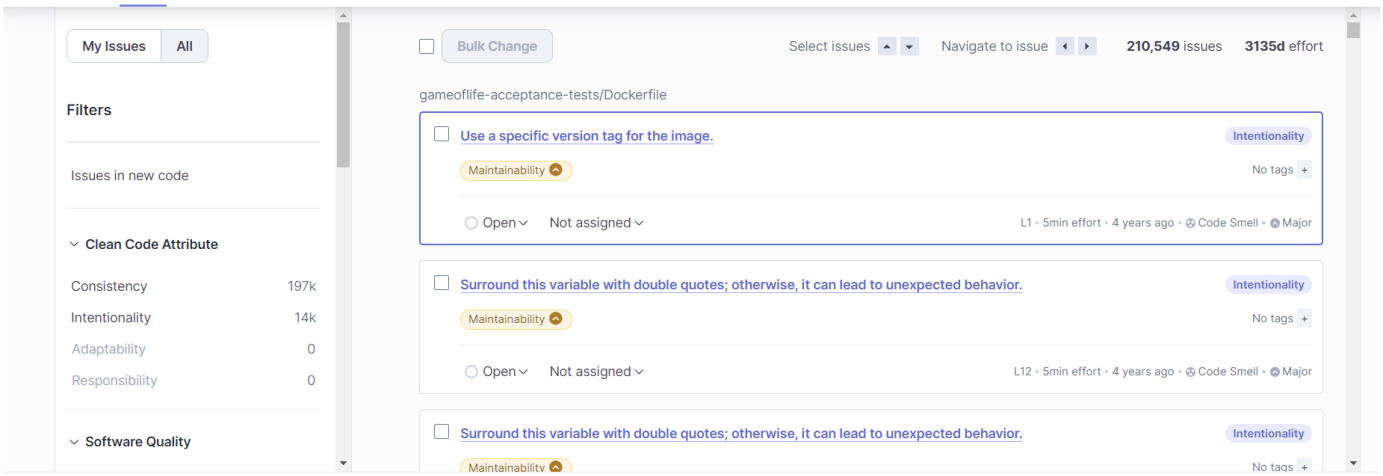
11.Check console



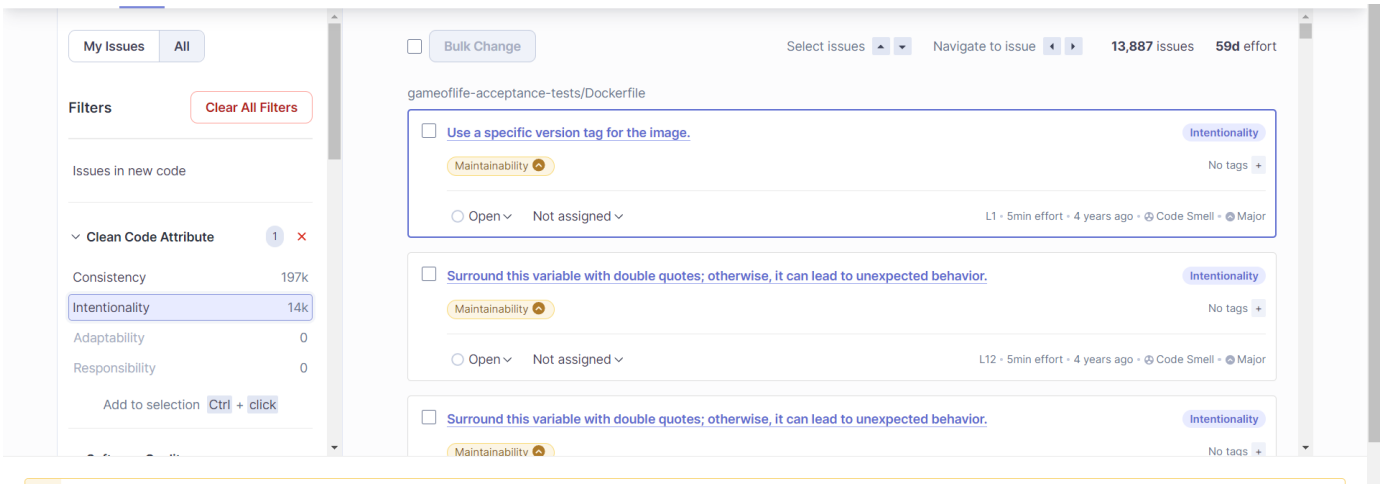
12.Now, check the project in SonarQube:



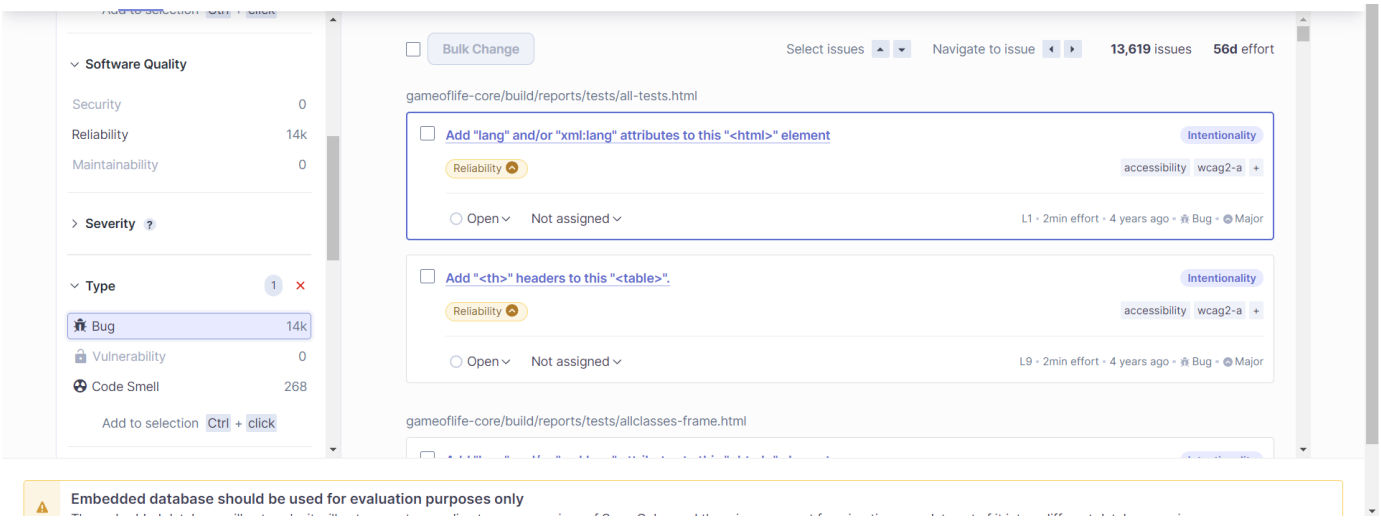
13.code problems consistency:



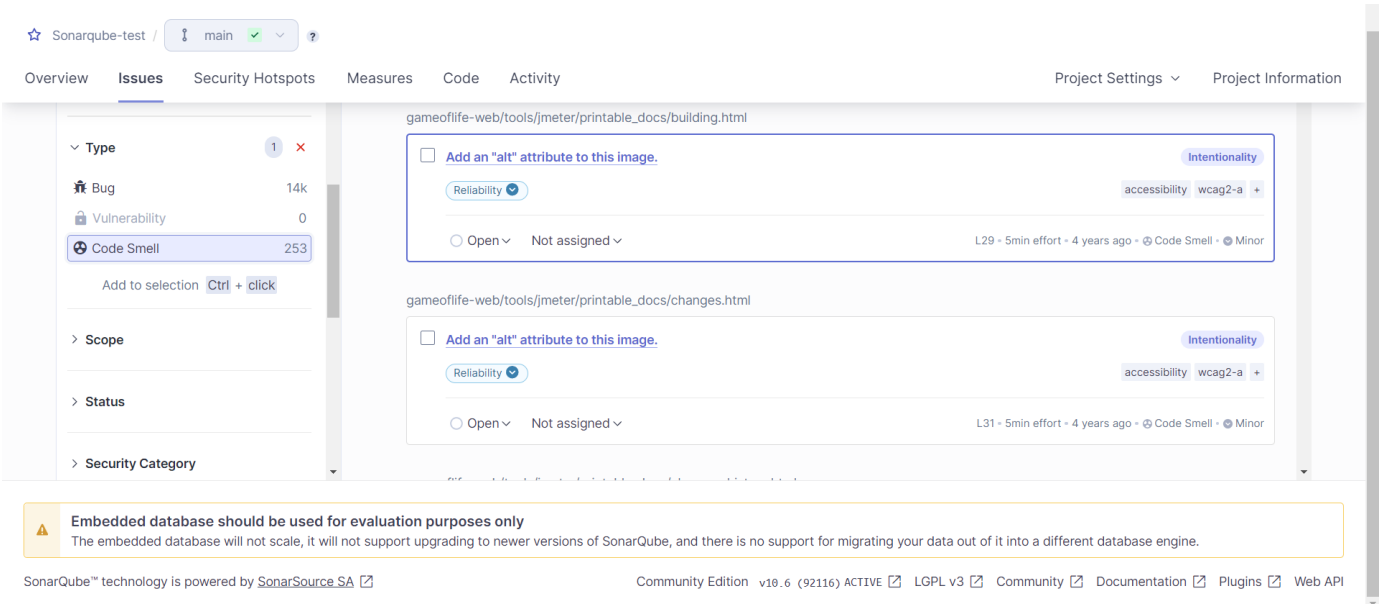
14.Intentionality:



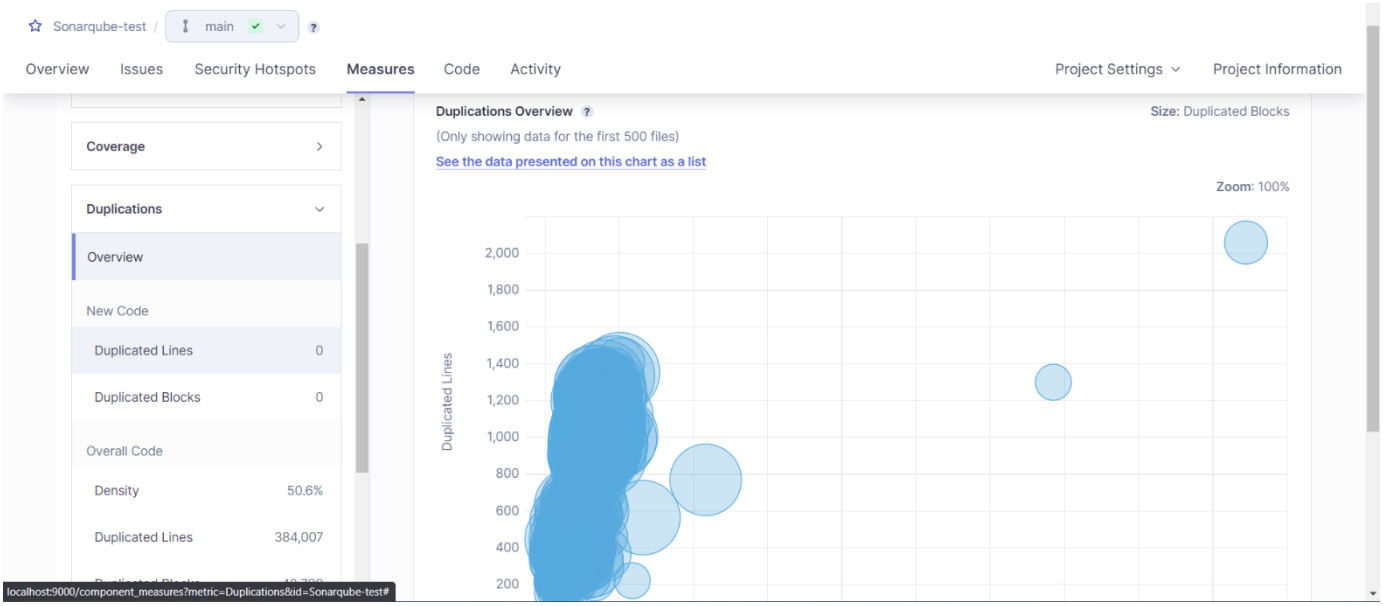
15.Bugs



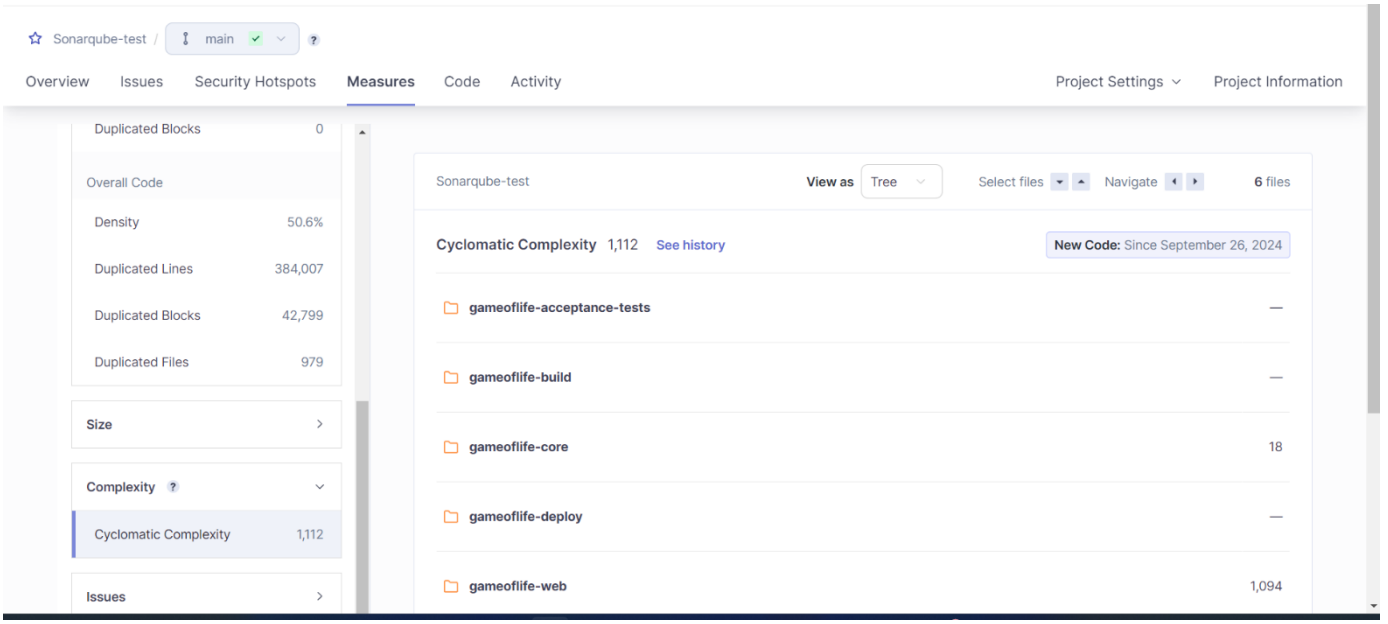
Code smells:



Duplications:



Cyclomatic Complexities:



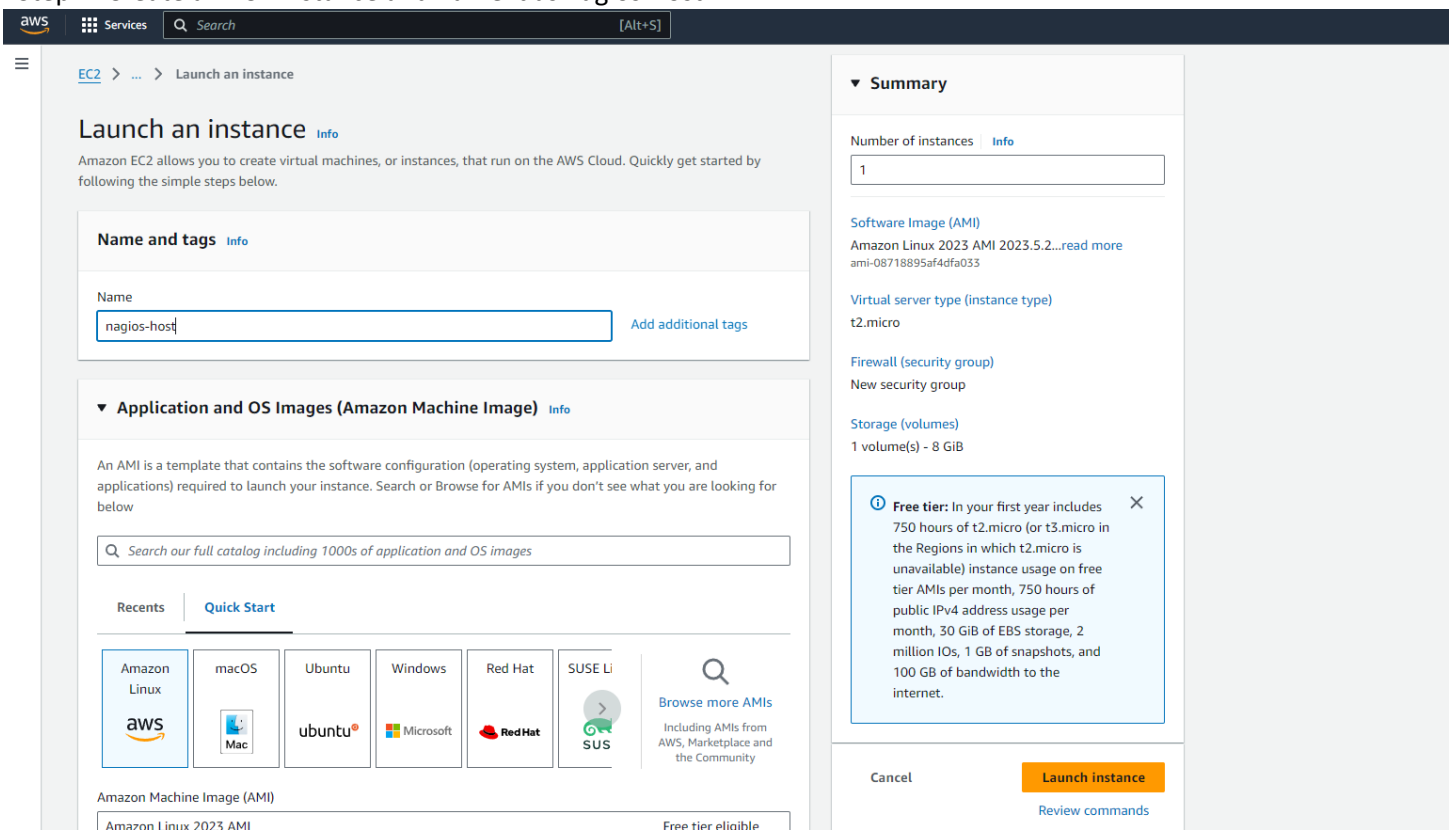
In this way, we have integrated Jenkins with SonarQube for SAST.

Experiment 9

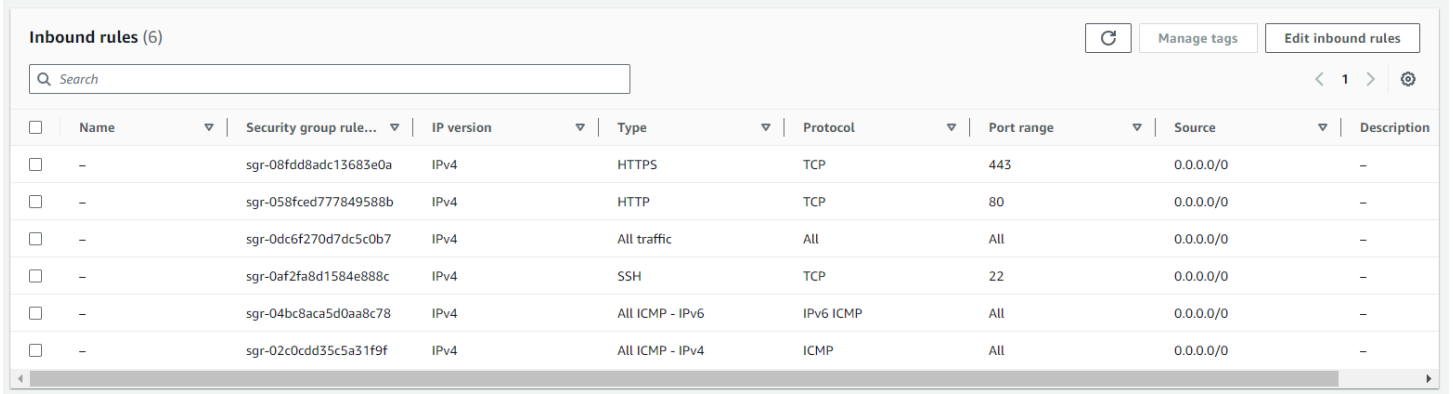
Atharva Prabhu
D15A 43

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

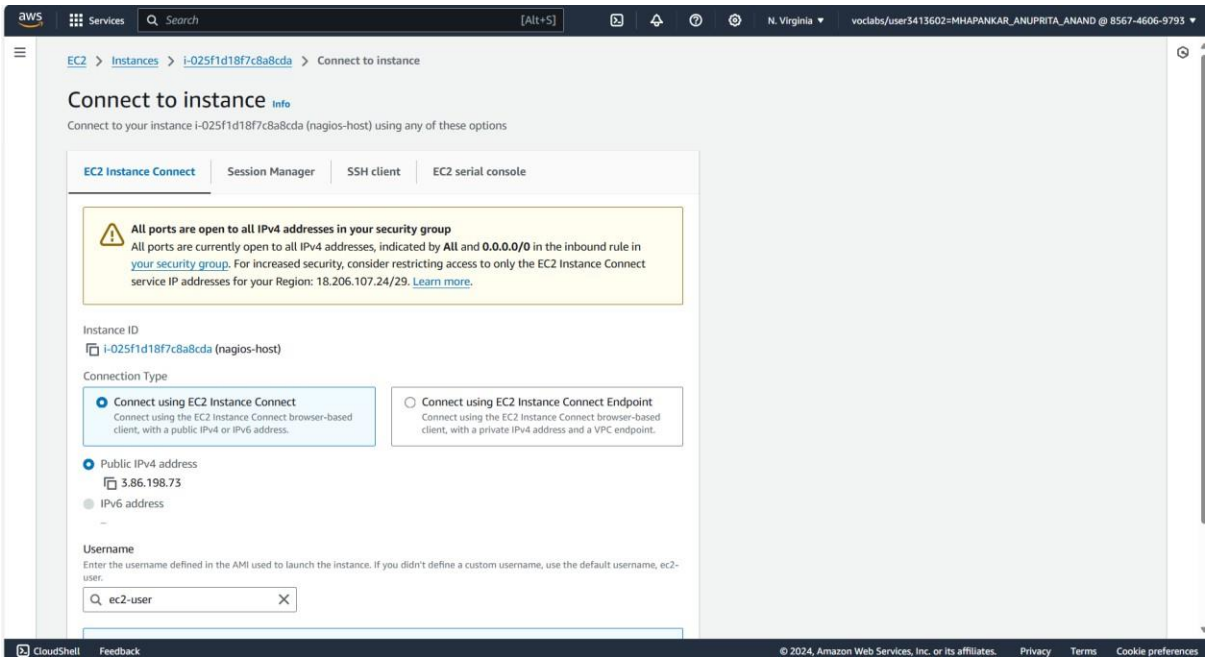
Step 1: Create an EC2 Instance and name it as nagios-host



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



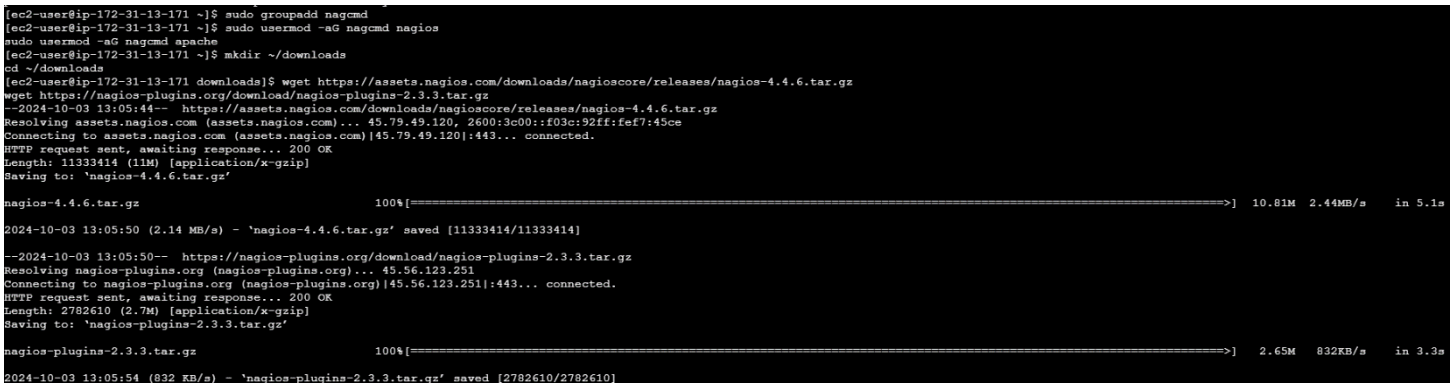
Step 3: Then select the instance nagios-host and then connect the instance.



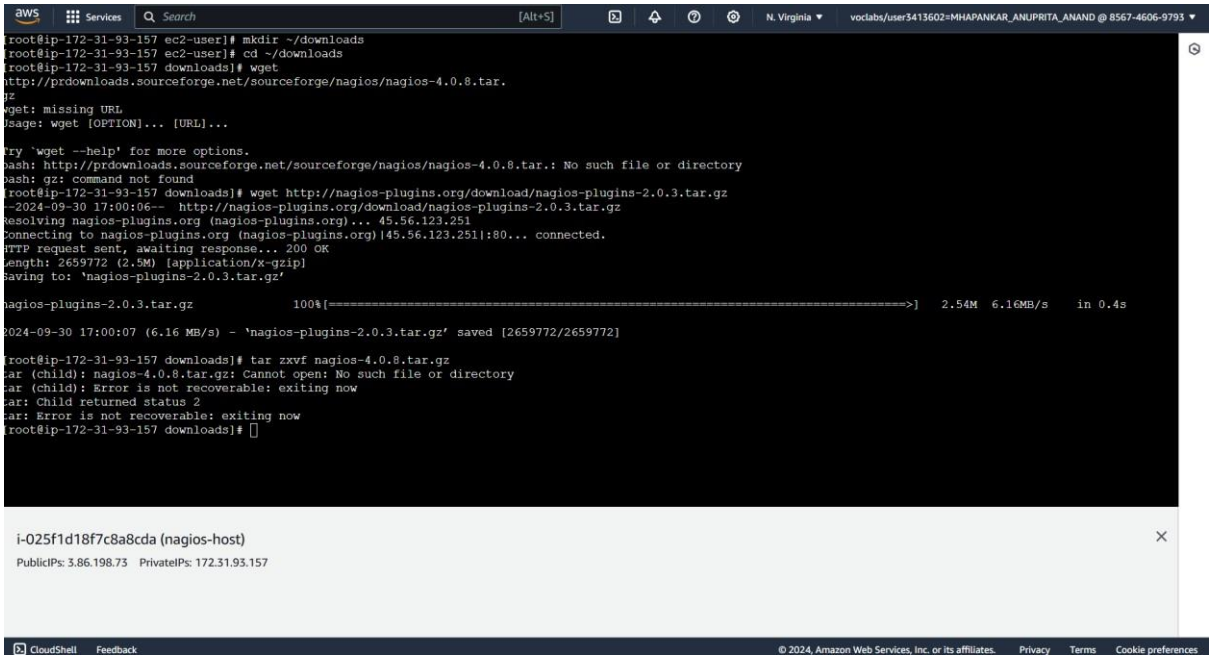
Step 3: Now, run the following commands -
sudo su
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-13-171 ~]$ sudo yum update -y
sudo yum install -y httpd php
sudo yum install -y gcc glibc glibc-common
sudo yum install -y gd gd-devel
Last metadata expiration check: 0:04:14 ago on Thu Oct  3 12:58:05 2024.
================================================================================
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.5.20241001:
    Run the following command to upgrade to 2023.5.20241001:

      dnf upgrade --releasever=2023.5.20241001

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.5.20241001.html

================================================================================
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:04:14 ago on Thu Oct  3 12:58:05 2024.
Dependencies resolved.
================================================================================
 Package                    Architecture        Version                      Repository                 Size
================================================================================
Installing:
 httpd                      x86_64              2.4.62-1.amzn2023            amazonlinux                48 k
 php8.3                     x86_64              8.3.10-1.amzn2023.0.1        amazonlinux                10 k
Installing dependencies:
 apr                        x86_64              1.7.2-2.amzn2023.0.2         amazonlinux               129 k
 apr-util                   x86_64              1.6.3-1.amzn2023.0.1         amazonlinux                98 k
 generic-logos-httpd        noarch              18.0.0-12.amzn2023.0.3       amazonlinux                19 k
 httpd-core                 x86_64              2.4.62-1.amzn2023            amazonlinux               1.4 M
 httpd-filesystem           noarch              2.4.62-1.amzn2023            amazonlinux                14 k
 httpd-tools                x86_64              2.4.62-1.amzn2023            amazonlinux                81 k
 libbrotli                  x86_64              1.0.9-4.amzn2023.0.2         amazonlinux               315 k
```

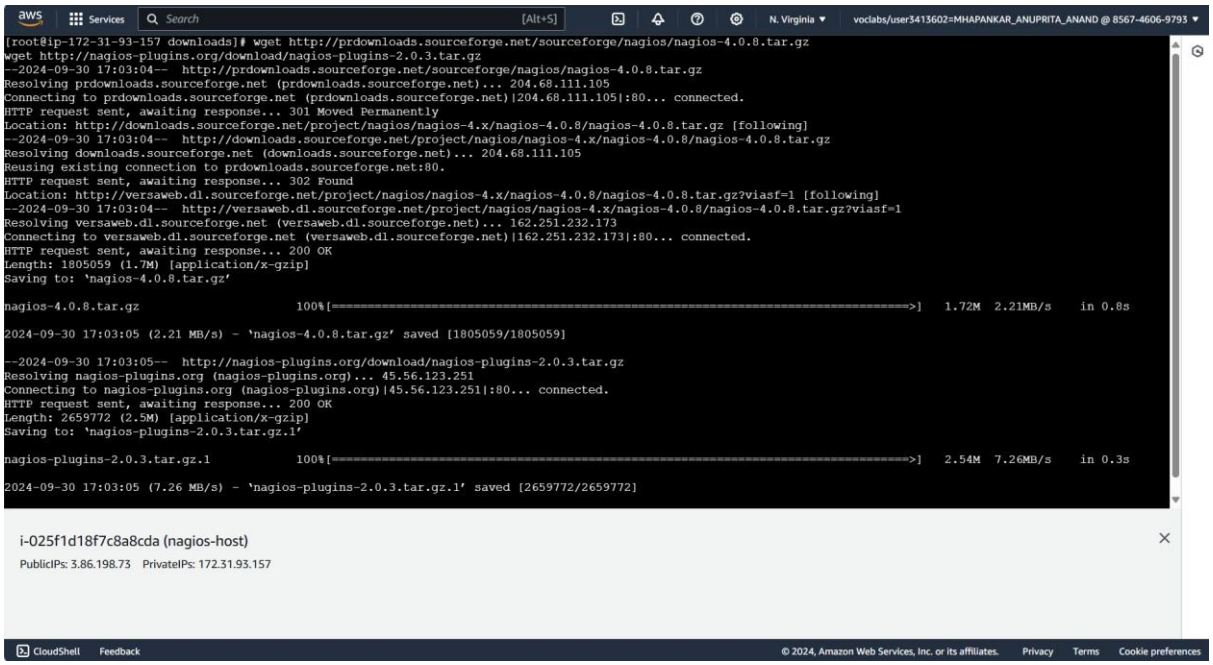
Step 4: Create a new nagios user with its password.
sudo adduser -m nagios
sudo passwd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache

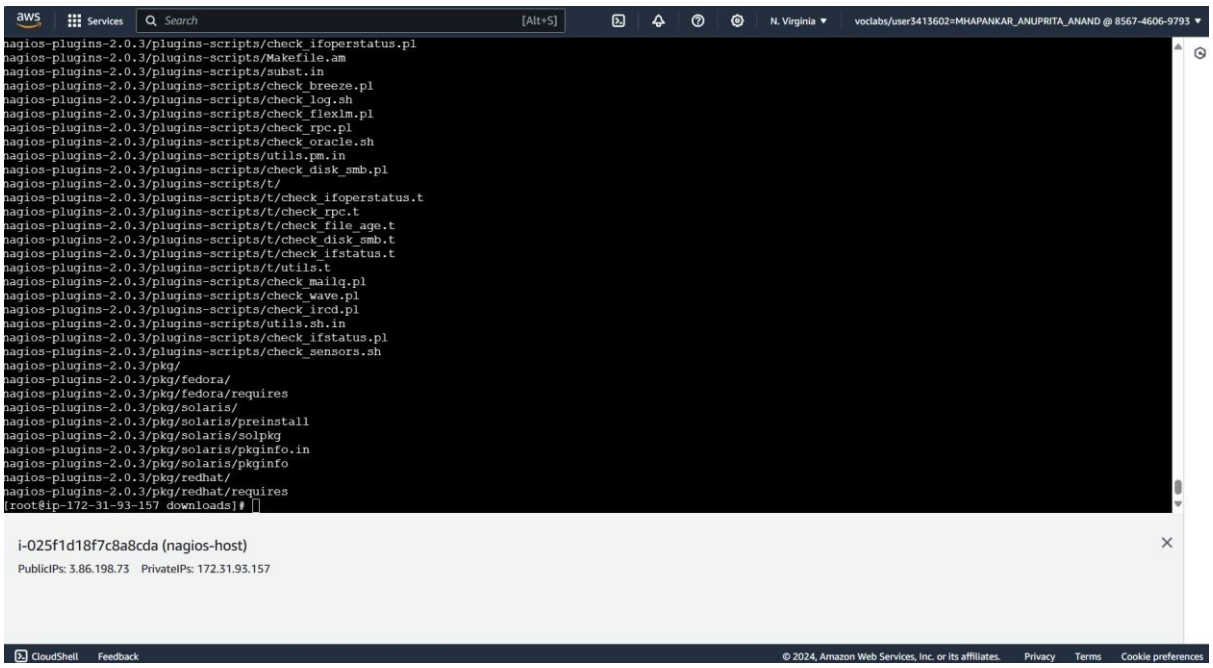


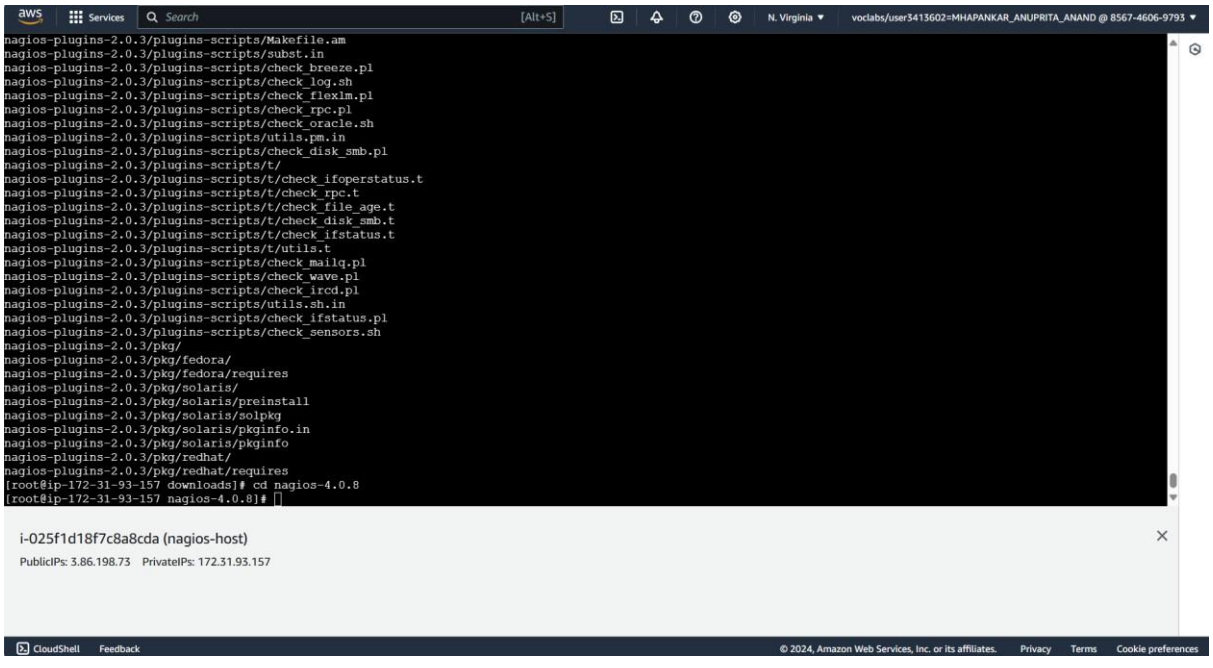
Step 5: Now, run the following commands -
mkdir ~/downloads
cd ~/downloads
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz



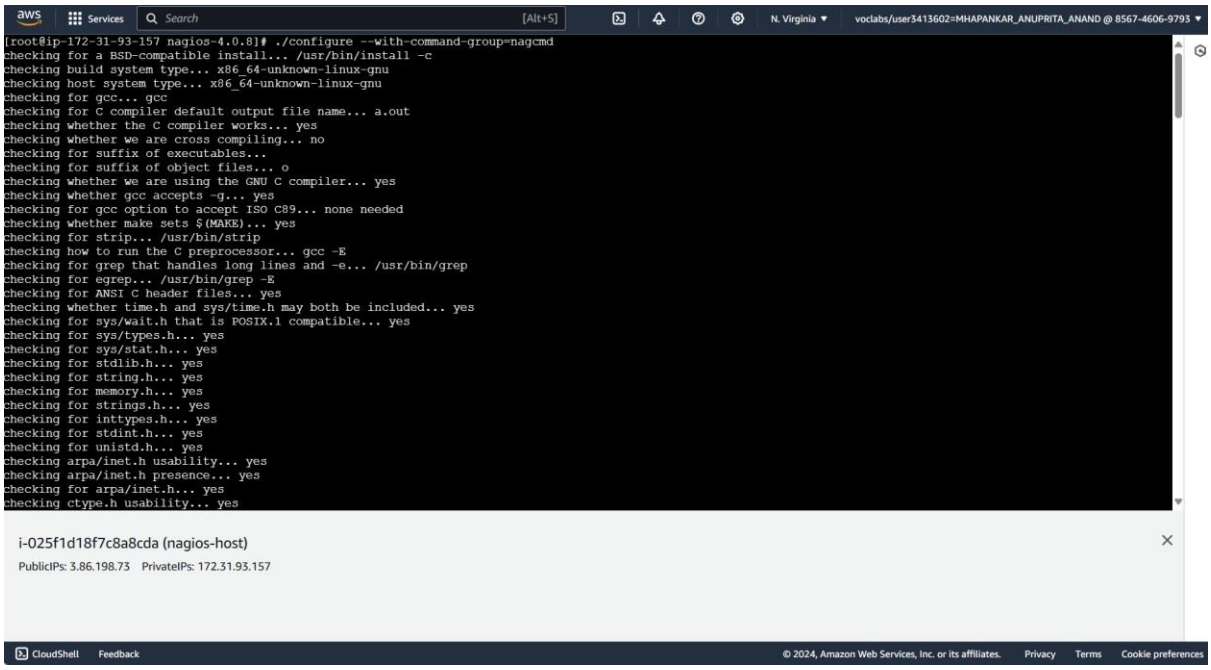
To resolve the error run the following commands -
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz
tar zxvf nagios-plugins-2.0.3.tar.gz
cd nagios-4.0.8


Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793
[root@ip-172-31-93-157 downloads]# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-09-30 17:03:04-- http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net)... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-30 17:03:04-- http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Reusing existing connection to prdownloads.sourceforge.net:80.
HTTP request sent, awaiting response... 302 Found
Location: http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1 [following]
--2024-09-30 17:03:04-- http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1
Resolving versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)... 162.251.232.173
Connecting to versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)|162.251.232.173|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'
nagios-4.0.8.tar.gz 100%[=>] 1.72M 2.21MB/s in 0.8s
2024-09-30 17:03:05 (2.21 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]
--2024-09-30 17:03:05-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz.1'
nagios-plugins-2.0.3.tar.gz.1 100%[=>] 2.54M 7.26MB/s in 0.3s
2024-09-30 17:03:05 (7.26 MB/s) - 'nagios-plugins-2.0.3.tar.gz.1' saved [2659772/2659772]
i-025f1d18f7c8a8cda (nagios-host)
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

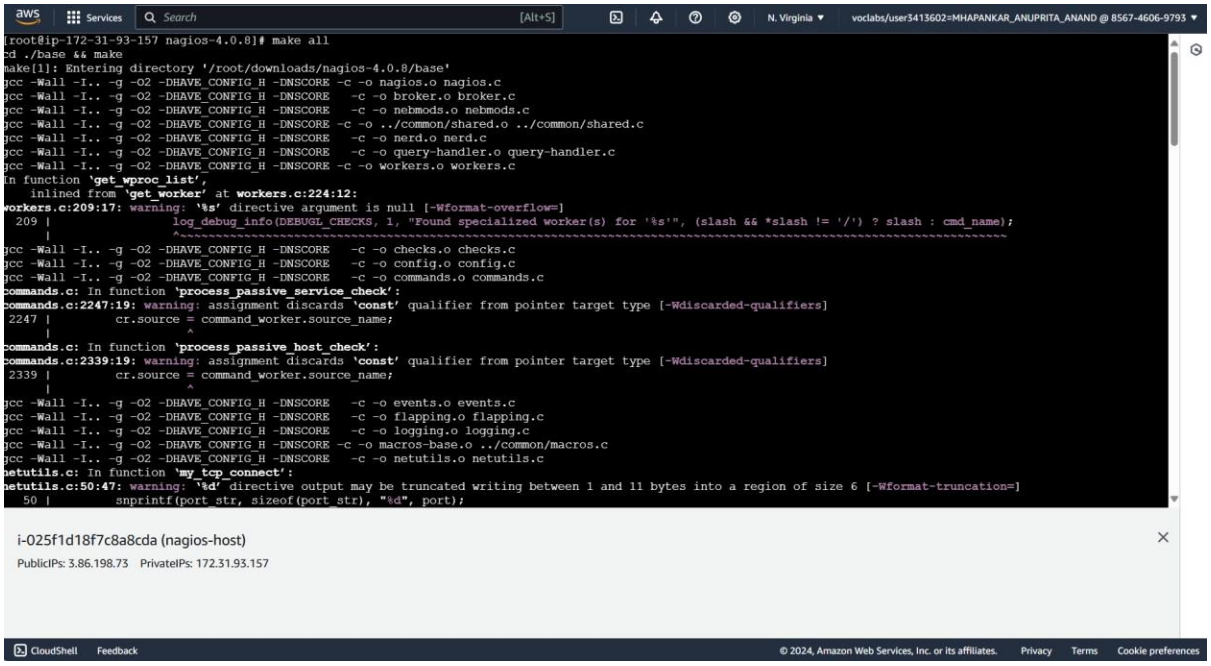

Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793
nagios-plugins-2.0.3/plugins-scripts/check_ifoperstatus.pl
nagios-plugins-2.0.3/plugins-scripts/Makefile.am
nagios-plugins-2.0.3/plugins-scripts/subst.in
nagios-plugins-2.0.3/plugins-scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins-scripts/check_log.sh
nagios-plugins-2.0.3/plugins-scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins-scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins-scripts/utils.pm.in
nagios-plugins-2.0.3/plugins-scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins-scripts/t/
nagios-plugins-2.0.3/plugins-scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins-scripts/t/check_file_age.t
nagios-plugins-2.0.3/plugins-scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins-scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/utils.t
nagios-plugins-2.0.3/plugins-scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins-scripts/check_wave.pl
nagios-plugins-2.0.3/plugins-scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins-scripts/utils.sh.in
nagios-plugins-2.0.3/plugins-scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requires
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requires
[root@ip-172-31-93-157 downloads]#
i-025f1d18f7c8a8cda (nagios-host)
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

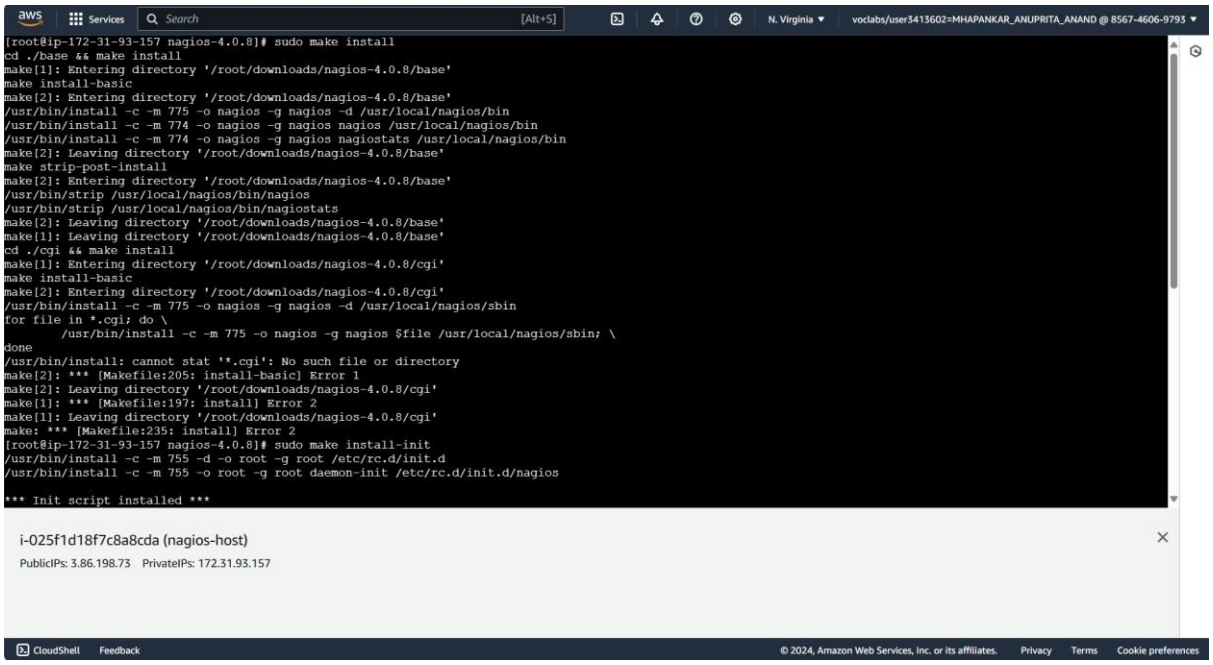

Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793
nagios-plugins-2.0.3/plugins-scripts/Makefile.am
nagios-plugins-2.0.3/plugins-scripts/subst.in
nagios-plugins-2.0.3/plugins-scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins-scripts/check_log.sh
nagios-plugins-2.0.3/plugins-scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins-scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins-scripts/utils.pm.in
nagios-plugins-2.0.3/plugins-scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins-scripts/t/
nagios-plugins-2.0.3/plugins-scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins-scripts/t/check_file_age.t
nagios-plugins-2.0.3/plugins-scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins-scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/utils.t
nagios-plugins-2.0.3/plugins-scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins-scripts/check_wave.pl
nagios-plugins-2.0.3/plugins-scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins-scripts/utils.sh.in
nagios-plugins-2.0.3/plugins-scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requires
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requires
[root@ip-172-31-93-157 downloads]# cd nagios-4.0.8
[root@ip-172-31-93-157 nagios-4.0.8]#
i-025f1d18f7c8a8cda (nagios-host)
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

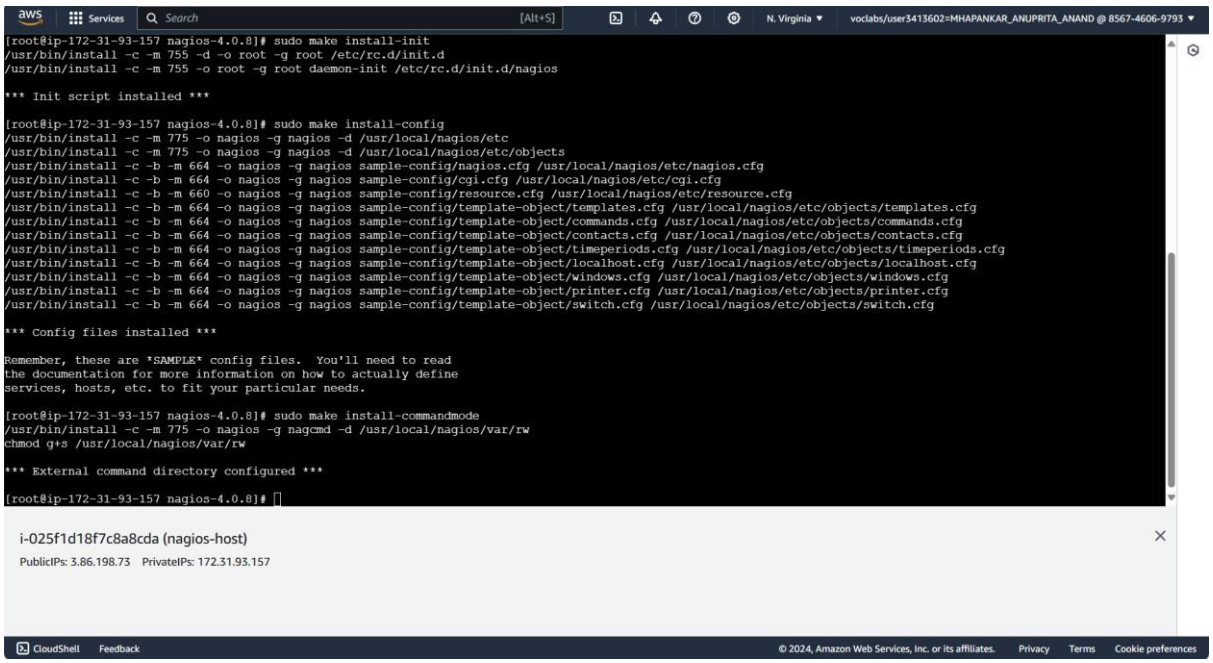
Step 6: Now to run the configuration script run the following command.
./configure --with-command-group=nagcmd



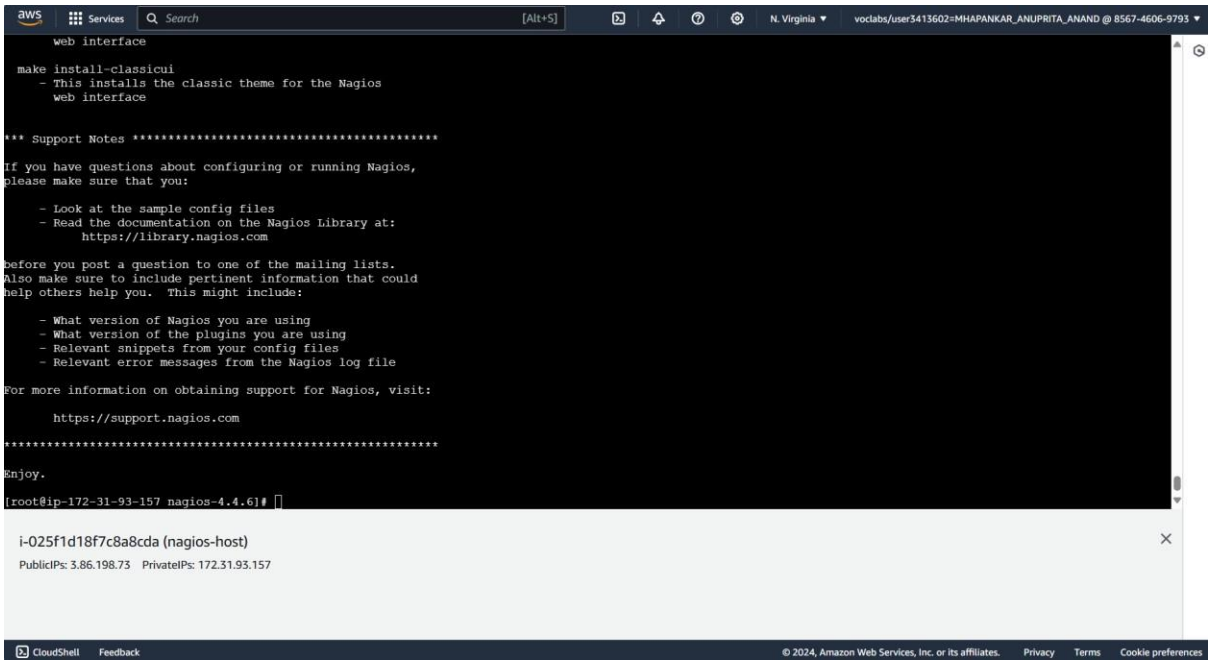
Step 7: Now, to compile the source code run the following command -
make all



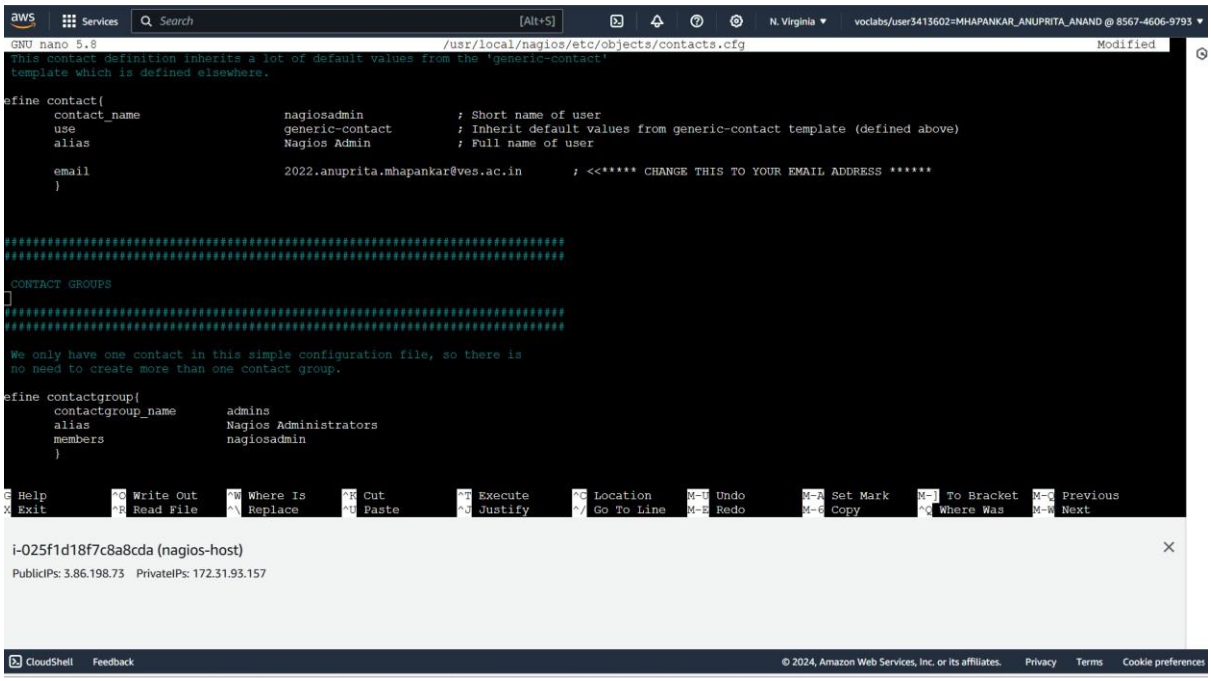




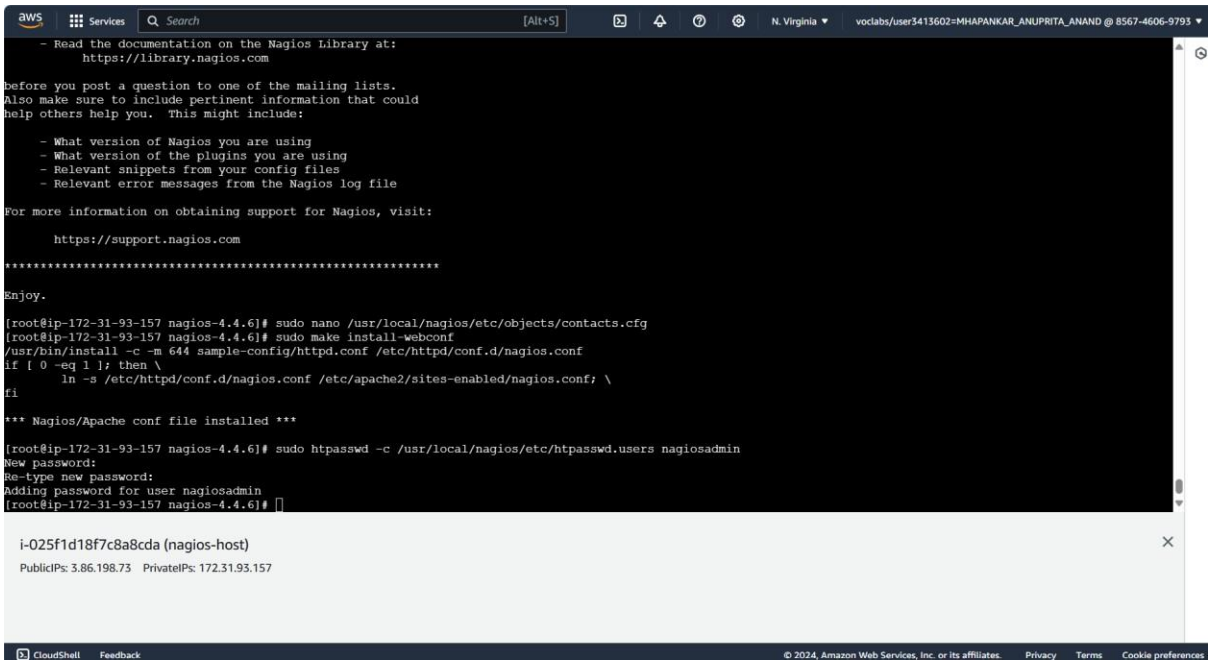
To resolve the errors run the following commands -
sudo yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel
rm -rf nagios-4.0.8
cd ~/downloads/nagios-4.4.6
./configure --with-command-group=nagcmd
make all
sudo make install

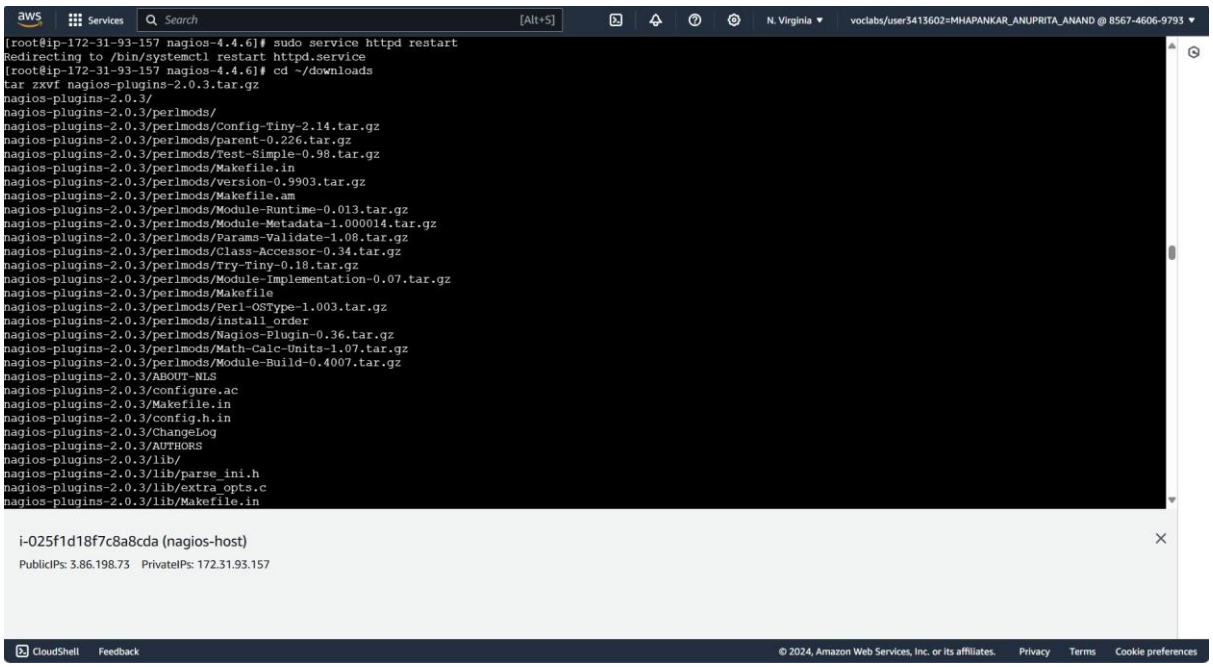


Step 8: Edit the config file and change the email address.
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

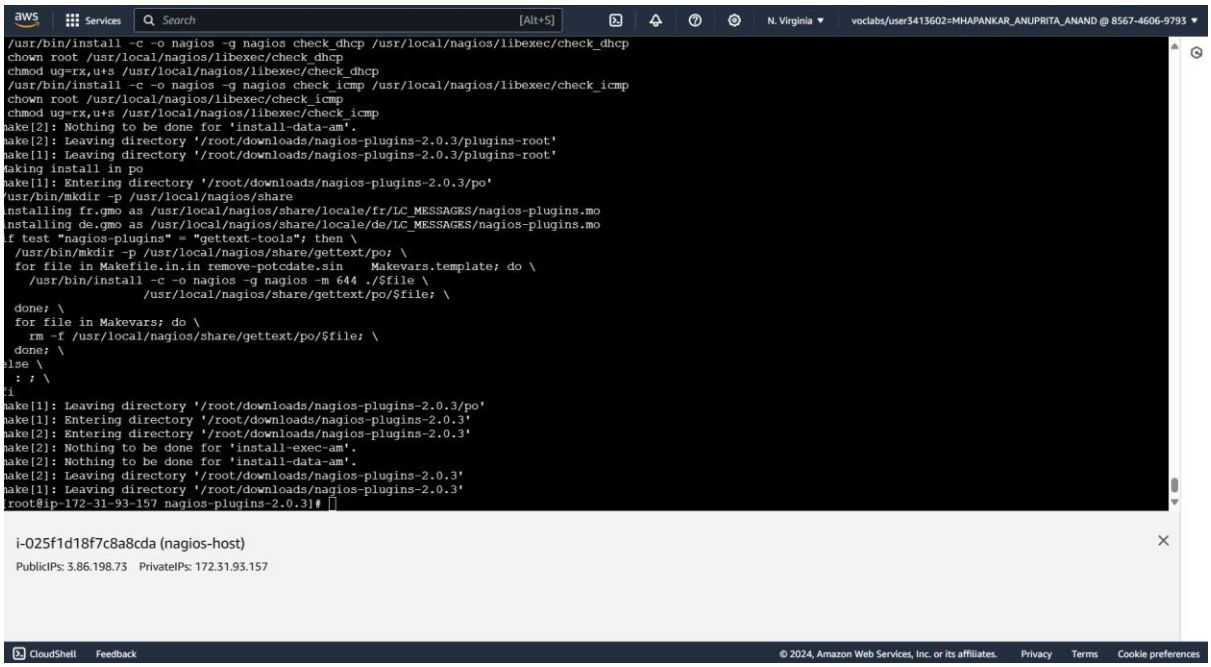


Step 9: Now run the following commands –
sudo make install-webconf
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
sudo service httpd restart
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz





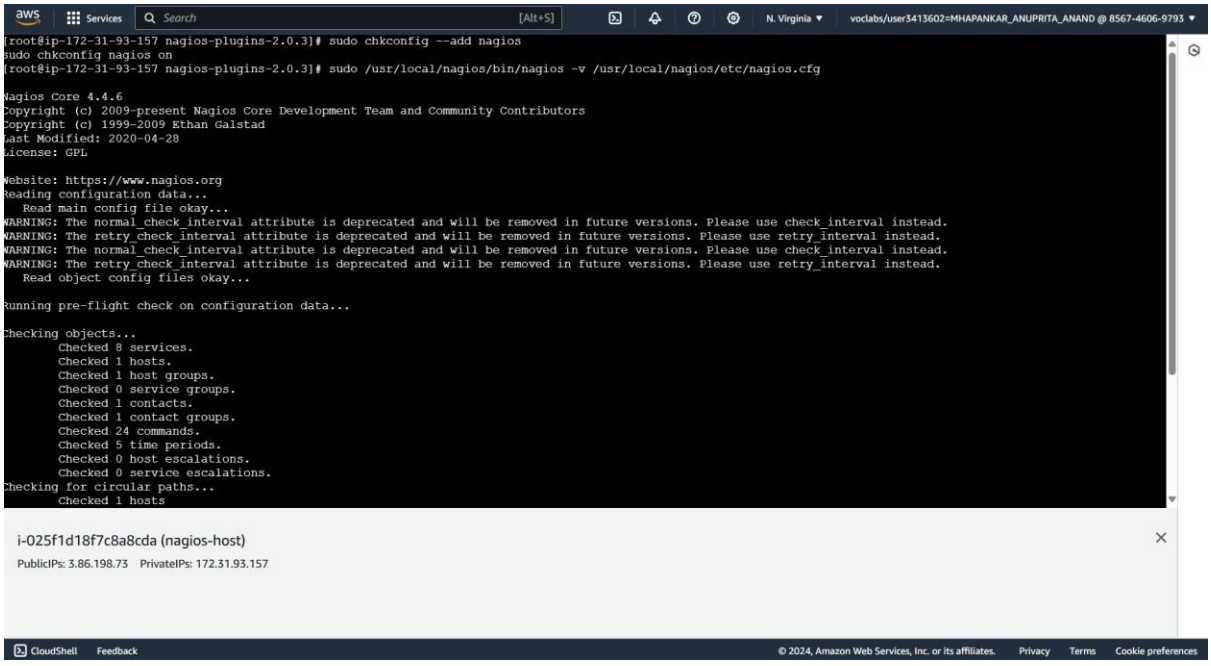
Step 10: Compile and install plugins
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install



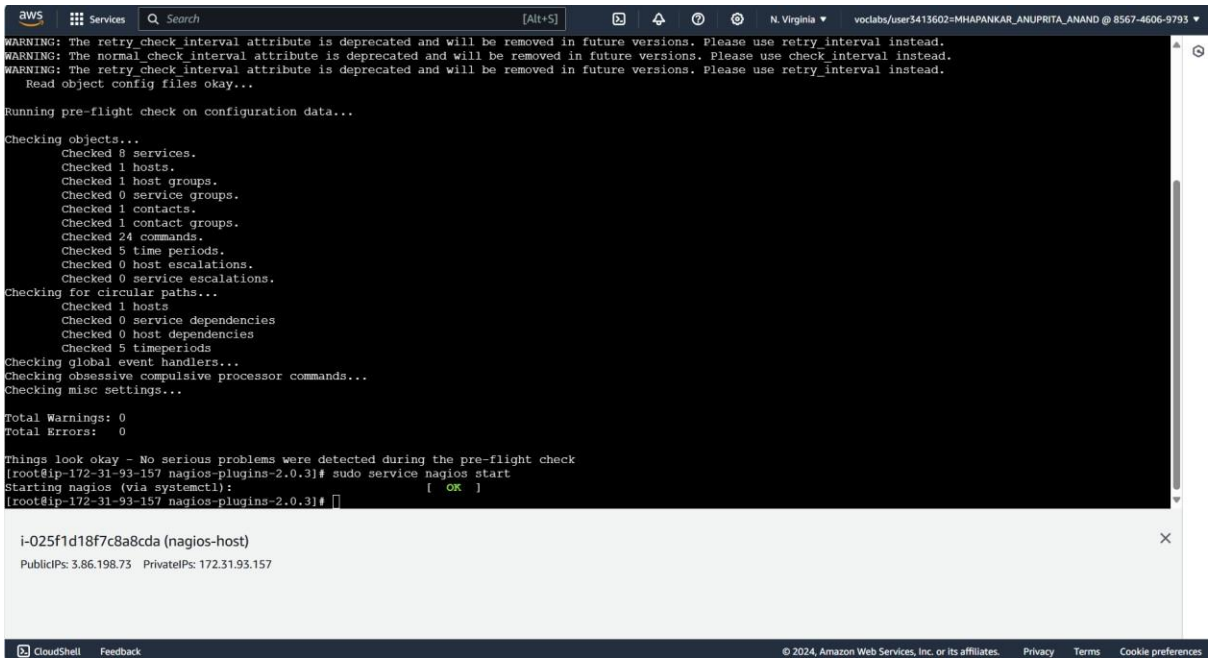
Step 11: To start nagios run the following commands –
sudo chkconfig --add nagios

sudo chkconfig nagios on

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



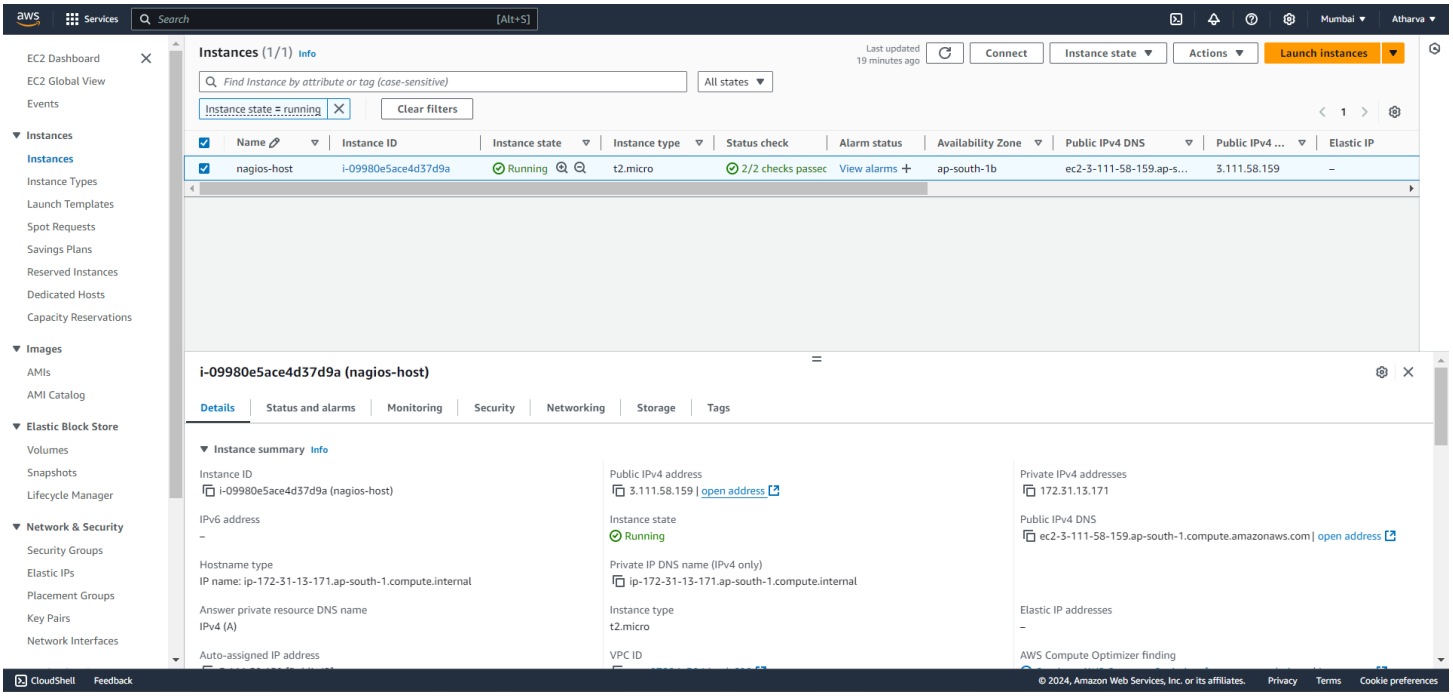
If there are no errors run the following command –
sudo service nagios start



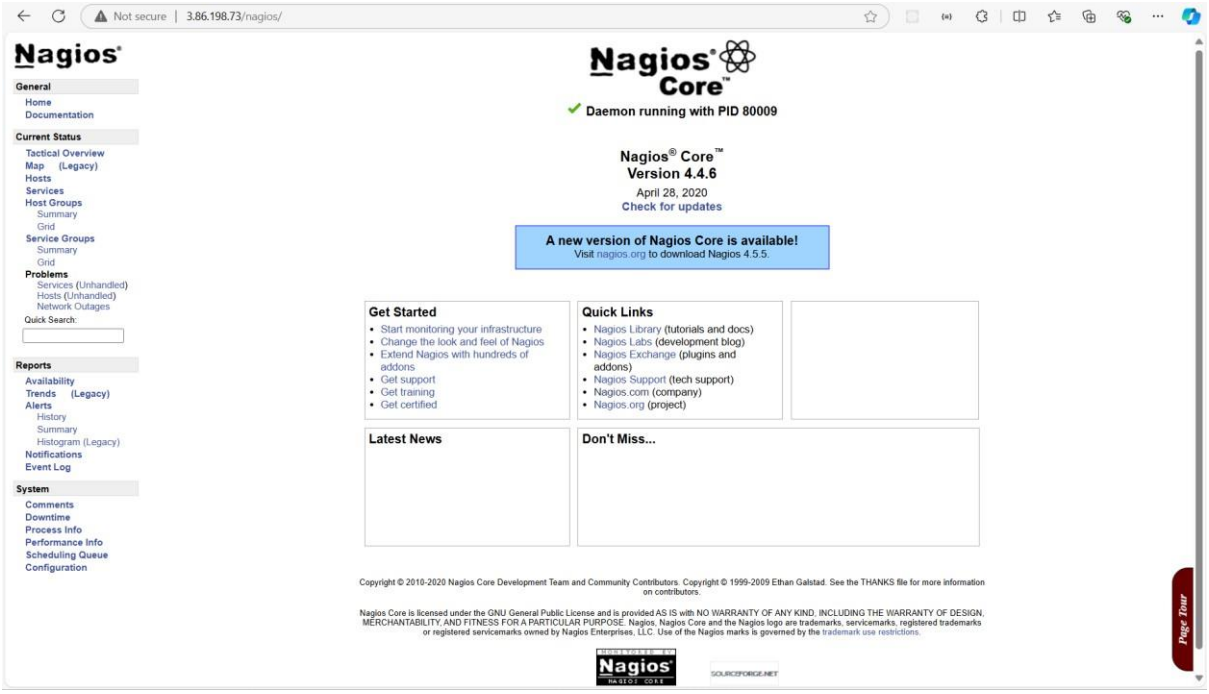
Check status using the following command -
sudo systemctl status nagios

```
[ec2-user@ip-172-31-13-171 nagios-plugins-2.3.3]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Thu 2024-10-03 13:18:35 UTC; 13s ago
       Docs: https://www.nagios.org/documentation
    Process: 67878 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 67879 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 67880 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.0M
        CPU: 18ms
     CGroup: /system.slice/nagios.service
             ├─67880 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67881 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67882 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67883 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67884 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67885 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: qh: core query handler registered
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: qh: echo service query handler registered
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: qh: help for the query handler registered
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: wproc: Successfully registered manager as @wproc with query handler
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: wproc: Registry request: name=Core Worker 67884;pid=67884
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: wproc: Registry request: name=Core Worker 67883;pid=67883
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: wproc: Registry request: name=Core Worker 67882;pid=67882
Oct 03 13:18:35 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: wproc: Registry request: name=Core Worker 67881;pid=67881
Oct 03 13:18:36 ip-172-31-13-171.ap-south-1.compute.internal nagios[67880]: Successfully launched command file worker with pid 67885
[ec2-user@ip-172-31-13-171 nagios-plugins-2.3.3]$
```

Step 12: Go to EC2 instance and copy the public IP address of the instance



Step 13: Now visit http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.



ADVANCE DEVOPS EXP-10

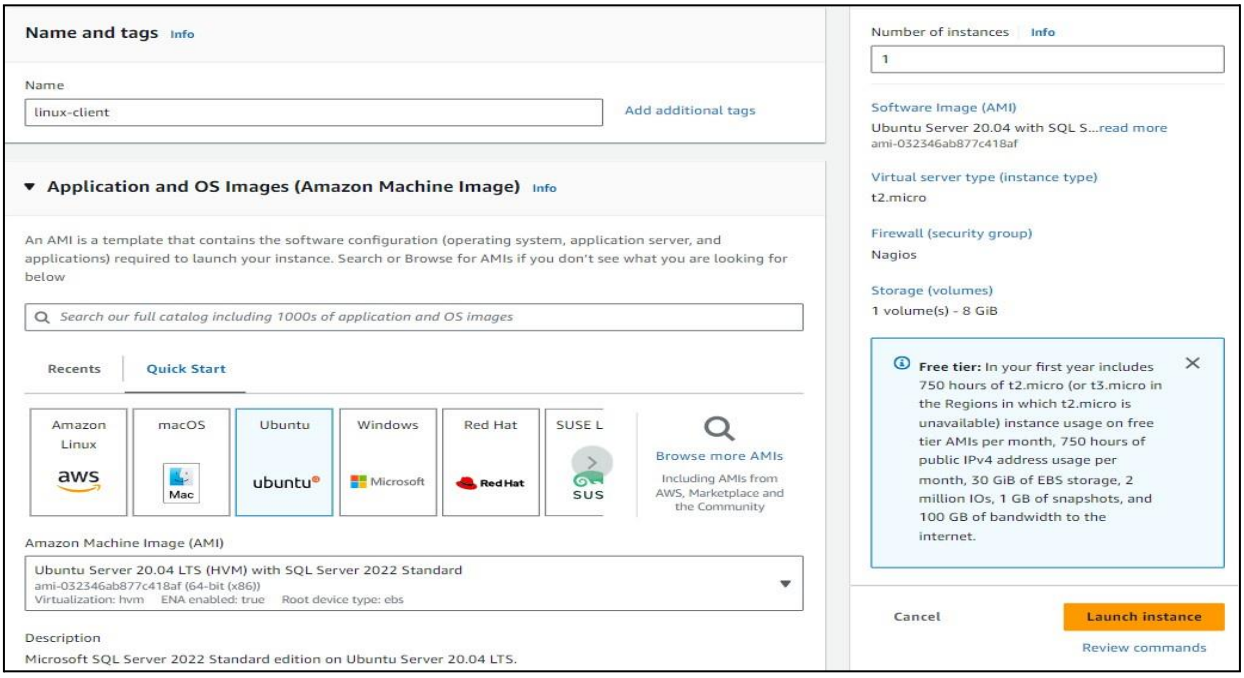
ATHARVA PRABHU
D15A 43

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

Step-1. Confirm Nagios is Running on the Server. sudo systemctl status nagios Proceed if you see that Nagios is active and running.

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.3.3]$ cd
[ec2-user@ip-172-31-90-152 ~]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-90-152 ~]$ 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 Mon 2024-09-30 19:41:36 UTC; 7s ago
       Docs: https://www.nagios.org/documentation
    Process: 80238 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 80239 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 80240 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.0M
        CPU: 15ms
     CGroup: /system.slice/nagios.service
             ├─80240 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─80241 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80242 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80243 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80244 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─80245 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: qh: core query handler registered
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: qh: echo service query handler registered
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: qh: help for the query handler registered
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: wproc: Registry request: name=Core Worker 80244;pid=80244
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: wproc: Registry request: name=Core Worker 80243;pid=80243
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: wproc: Registry request: name=Core Worker 80242;pid=80242
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: wproc: Registry request: name=Core Worker 80241;pid=80241
Sep 30 19:41:36 ip-172-31-90-152.ec2.internal nagios[80240]: Successfully launched command file worker with pid 80245
```

Step-2. Create an Ubuntu 20.04 Server EC2 Instance



Step-3: Verify Nagios Process on the Server

```
[ec2-user@ip-172-31-80-215 nagios-plugins-2.3.3]$ ps -ef | grep nagios
nagios     68654       1  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     68655   68654  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68656   68654  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68657   68654  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68658   68654  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68659   68654  0 20:29 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   69588   26447  0 20:44 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-215 nagios-plugins-2.3.3]$
```

Step-4: Become Root User and Create Directories-
sudo su , mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
and to copy the same config file- cp /usr/local/nagios/etc/objects/localhost.cfg, /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
[ec2-user@ip-172-31-80-215 nagios-plugins-2.3.3]$ sudo su
[root@ip-172-31-80-215 nagios-plugins-2.3.3]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-215 nagios-plugins-2.3.3]# cp /usr/local/nagios/etc/objects/localhost.cfg
cp: missing destination file operand after '/usr/local/nagios/etc/objects/localhost.cfg'
Try 'cp --help' for more information.
[root@ip-172-31-80-215 nagios-plugins-2.3.3]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-80-215 nagios-plugins-2.3.3]#
```

i-0ae1aae975bae3b7a (nagios-host)

Step-5: Edit the Configuration File
sudo nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

- Change hostname to linuxserver everywhere in the file
- Change address to the public IP address of your linux-client.
- Change host_group name under hostgroup to linux_server

```
##############################################################################
#
# HOST DEFINITION
#
##############################################################################

# Define a host for the local machine

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                   linuxserver
    address                 35.174.139.220
}

##############################################################################
#
# 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
}
                                                                                [ Read 157 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where
```

Step-6: Update Nagios Configuration

sudo nano /usr/local/nagios/etc/nagios.cfg

Add the command - cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg


# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

Step-7: Verify Configuration Files

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

```
[ec2-user@ip-172-31-80-215 ~]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg'
Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg',
Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhos
Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localh
Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/loc
Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objects/local
Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/loca
Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg'
   Read object config files okay...

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
```

Step-8: Restart Nagios Service
sudo systemctl restart nagios

Step-9:. SSH into the Client Machine
Use SSH or EC2 Instance Connect to access the `linux-client`.

Step-10: Update Package Index and Install Required Packages sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-86-24:~$ sudo apt update -y
sudo apt install gcc -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 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 [83.1 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [274 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8652 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [379 kB]
```

Step-11: Edit NRPE Configuration File

Commands -
sudo nano /etc/nagios/nrpe.cfg
Add your Nagios host IP address under `allowed_hosts:`
`allowed_hosts=<Nagios_Host_IP>`

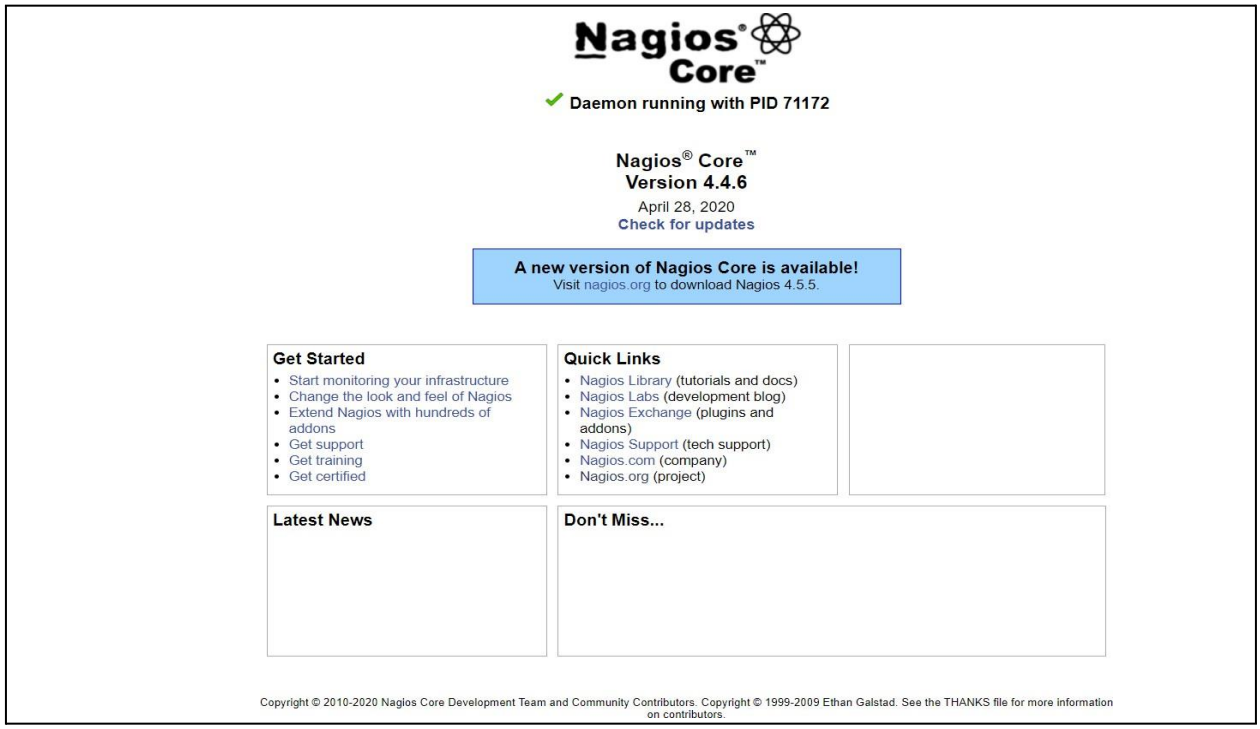
```
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,35.174.139.220



# 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.
#
# Values: 0=do not allow arguments, 1=allow command arguments

dont_blame_nrpe=0
```

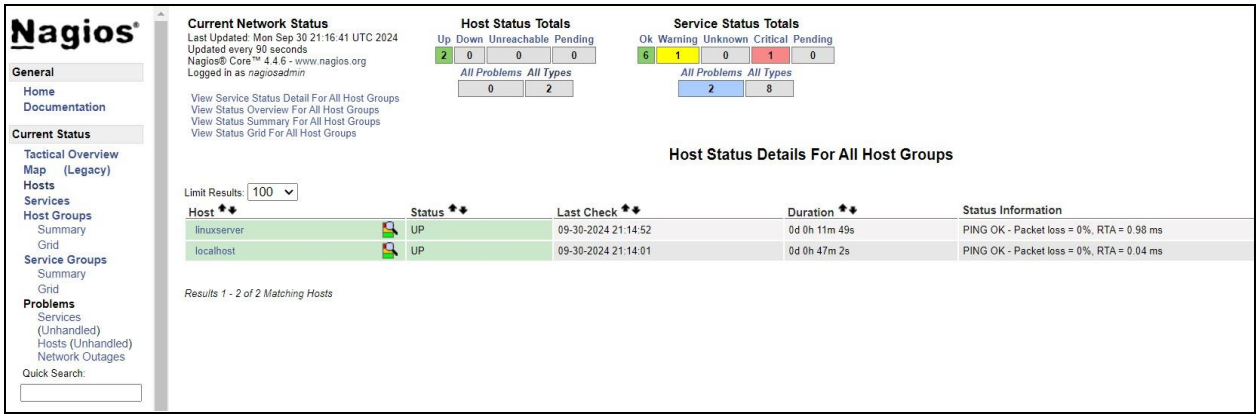
Step-12: Restart NRPE Server

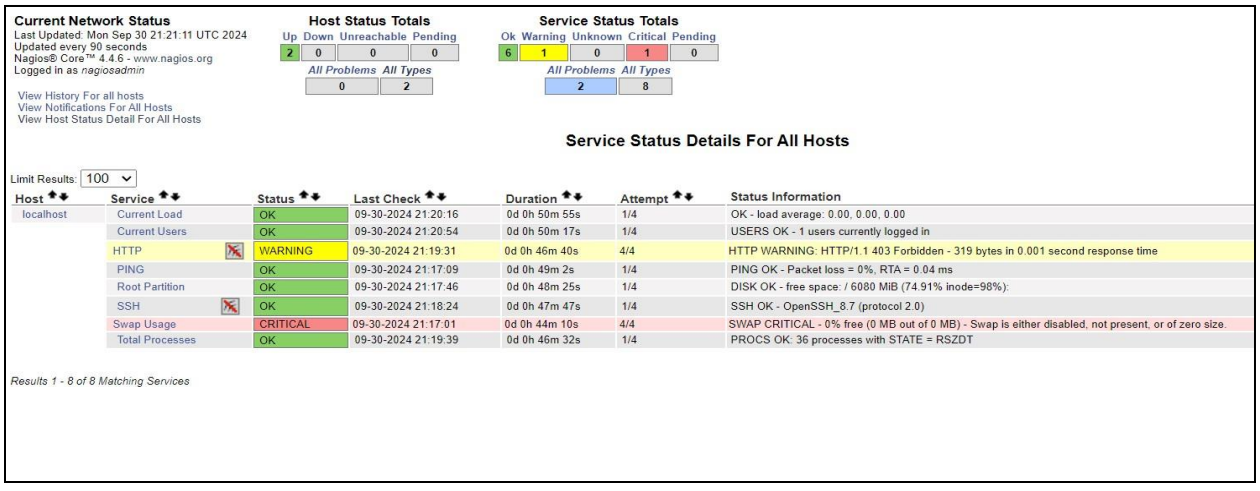
Commands -
sudo systemctl restart nagios-nrpe-server

Step-13: Check Nagios Dashboard

Open your browser and navigate to http://<Nagios_Host_IP>/nagios.
Log in with nagiosadmin and the password you set earlier.
You should see the new host linuxserver added.
Click on Hosts to see the host details.
Click on Services to see all services and ports being monitored


Nagios Core
Daemon running with PID 71172
Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates
A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.
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...
Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.


Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Current Network Status
Last Updated: Mon Sep 30 21:16:41 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
6 1 0 1 0
All Problems All Types
2 8
Host Status Details For All Host Groups
Limit Results: 100
Host Status Last Check Duration Status Information
linuxserver UP 09-30-2024 21:14:52 0d 0h 11m 49s PING OK - Packet loss = 0%, RTA = 0.98 ms
localhost UP 09-30-2024 21:14:01 0d 0h 47m 2s PING OK - Packet loss = 0%, RTA = 0.04 ms
Results 1 - 2 of 2 Matching Hosts


Current Network Status
Last Updated: Mon Sep 30 21:21:11 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin
View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
6 1 0 1 0
All Problems All Types
2 8
Service Status Details For All Hosts
Limit Results: 100
Host Service Status Last Check Duration Attempt Status Information
localhost Current Load OK 09-30-2024 21:20:16 0d 0h 50m 55s 1/4 OK - load average: 0.00, 0.00, 0.00
Current Users OK 09-30-2024 21:20:54 0d 0h 50m 17s 1/4 USERS OK - 1 users currently logged in
HTTP WARNING 09-30-2024 21:19:31 0d 0h 46m 40s 4/4 HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
PING OK 09-30-2024 21:17:09 0d 0h 49m 2s 1/4 PING OK - Packet loss = 0%, RTA = 0.04 ms
Root Partition OK 09-30-2024 21:17:46 0d 0h 48m 25s 1/4 DISK OK - free space: / 6080 MiB (74.91% inode=98%):
SSH OK 09-30-2024 21:18:24 0d 0h 47m 47s 1/4 SSH OK - OpenSSH_8.7 (protocol 2.0)
Swap Usage CRITICAL 09-30-2024 21:17:01 0d 0h 44m 10s 4/4 SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
Total Processes OK 09-30-2024 21:19:39 0d 0h 46m 32s 1/4 PROCS OK: 36 processes with STATE = RSZDT
Results 1 - 8 of 8 Matching Services

Experiment 11

Atharva Prabhu
D15A 43

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

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). Users of AWS Lambda create functions, self-contained applications written in one of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner.The Lambda functions can perform any kind of computing task, from serving web pages and processing streams of data to calling APIs and integrating with other AWS services.

The concept of “serverless” computing refers to not needing to maintain your own servers to run these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure for you. And so “serverless” doesn’t mean that there are no servers involved: it just means that the servers, the operating systems, the network layer and the rest of the infrastructure have already been taken care of so that you can focus on writing application code.

Features of AWS Lambda

- AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
- It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
- You don’t need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
- AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.
- It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket. And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served, a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container with our function to serve those requests.

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code. AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

The above execution model makes Lambda functions effectively stateless. This means that every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event.

However, due to the optimization noted above, the actual Lambda function is invoked only once per container instantiation. Recall that our functions are run inside containers. So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.

For example, the createNewDbConnection method below is called once per container instantiation and not every time the Lambda function is invoked. The myHandler function on the other hand is called on every invocation.

Common Use Cases for Lambda

Due to Lambda's architecture, it can deliver great benefits over traditional cloud computing setups for applications where:

1. Individual tasks run for a short time;
2. Each task is generally self-contained;

3. There is a large difference between the lowest and highest levels in the workload of the application.

Some of the most common use cases for AWS Lambda that fit these criteria are:
Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to

the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events. For example, you could employ Lambda to do some work every time an item in DynamoDB is created or updated, thus making it a good fit for things like notifications, counters and analytics.

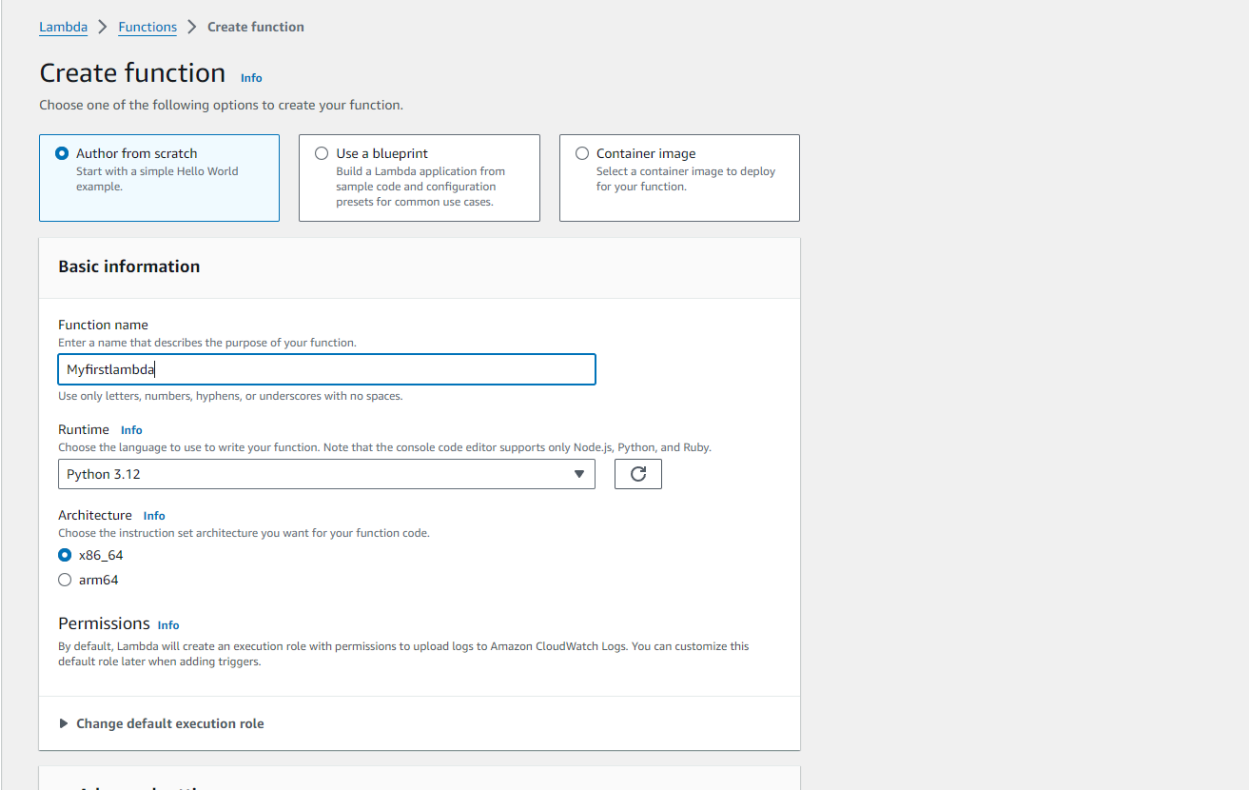
Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1.Choose a Function Creation Method:
Select Author from scratch.

2.Configure the Function:
Function name: Enter a name for your function (e.g., MyFirstLambda).
Runtime: Choose Python 3.x (the latest available version).
Permissions:Choose Create a new role with basic Lambda permissions (this creates a role with the necessary permissions).

3.Click on Create function.



Step 2: Write Your Lambda Function Code

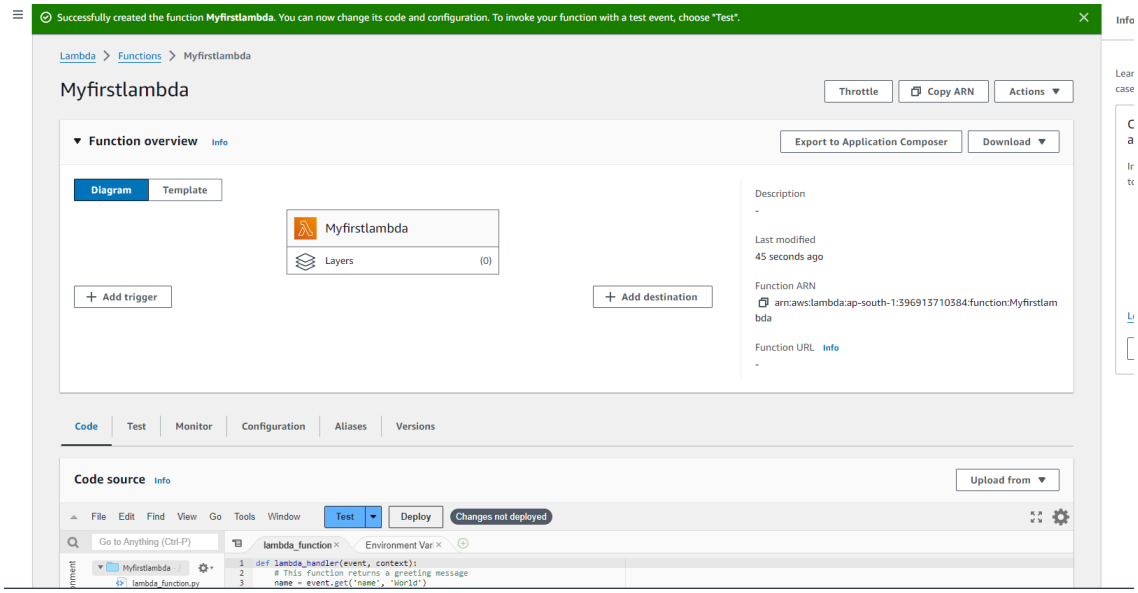
In the Function code section, you will see a code editor. Replace the default code with the following Python code:

python

Copy code

```
def lambda_handler(event, context):
    # This function returns a greeting message
    name = event.get('name', 'World')
    return {
        'statusCode': 200,
        'body': f'Hello, {name}!'
    }
```

This function reads a name from the event and returns a greeting message. If no name is provided, it defaults to "World".



Step3:

1.Configure a Test Event:

Click on the Test button.
In the Configure test event dialog, give your event a name (e.g., TestEvent).
Replace the default JSON with the following:

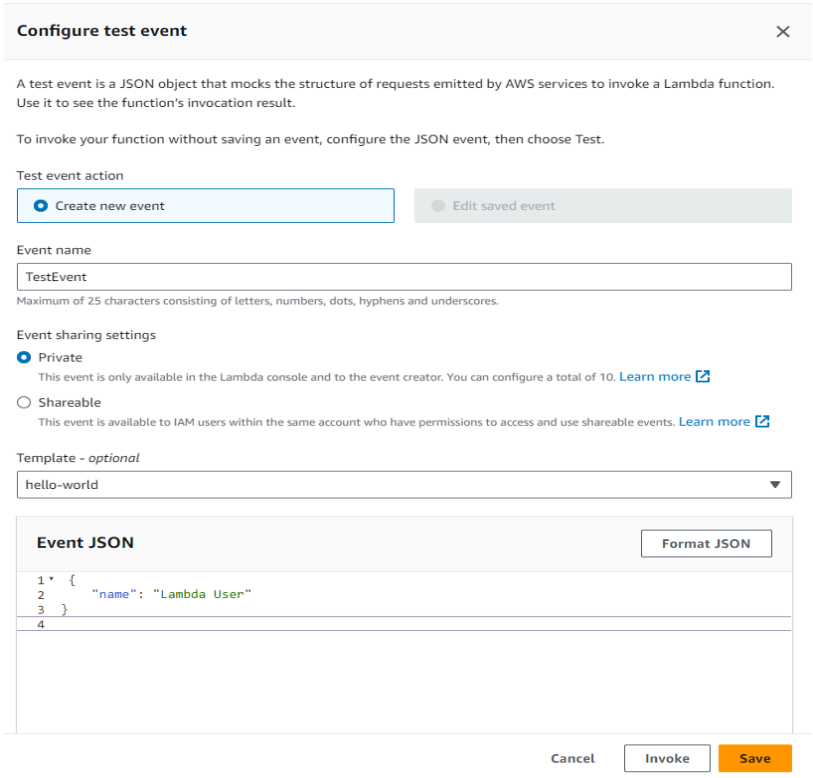
```
{
   "name": "Lambda User"
}
```

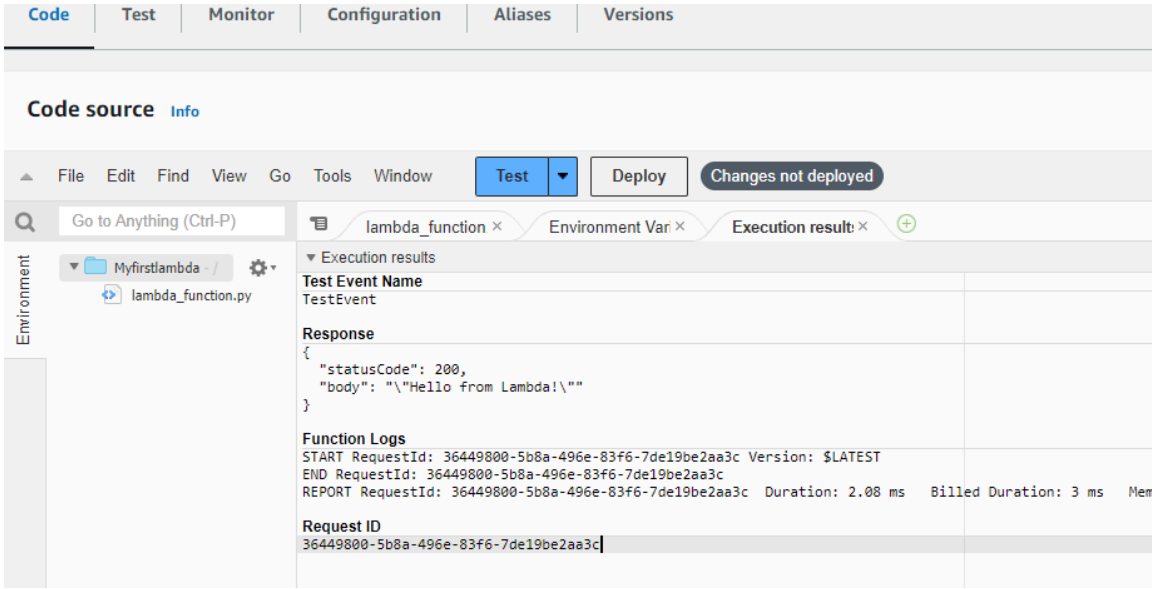
2.Run the Test:

Click on the Test button again to execute your Lambda function.

You should see the execution results below the code editor, including the response:
json
Copy code

```
{
    "statusCode": 200,
    "body": "Hello, Lambda User!"
}
```





Conclusion:

AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database updates, and scales based on the demand. The workflow of AWS Lambda involves defining a function with specific logic, configuring triggers that will invoke the function, and setting permissions to control access. Lambda supports multiple programming languages, including Python, Java, and Node.js, enabling developers to choose the best fit for their applications. Creating your first Lambda function is straightforward: you write the code, define triggers, and deploy, allowing you to quickly build and run applications without the overhead of managing infrastructure. This simplicity and flexibility make AWS Lambda an excellent tool for building modern, event-driven applications.

EXPERIMENT NO - 12

ATHARVA PRABHU D15A 43

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

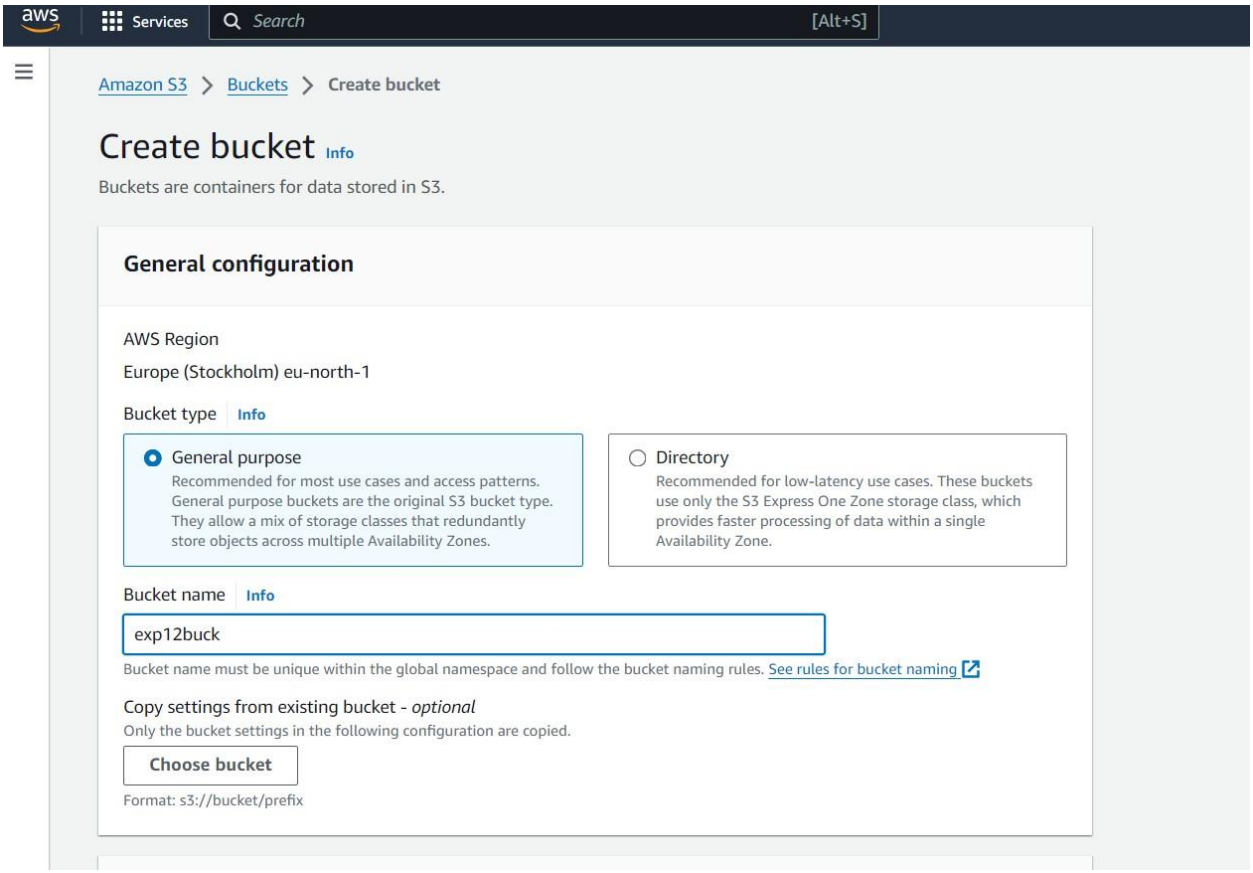
Theory:

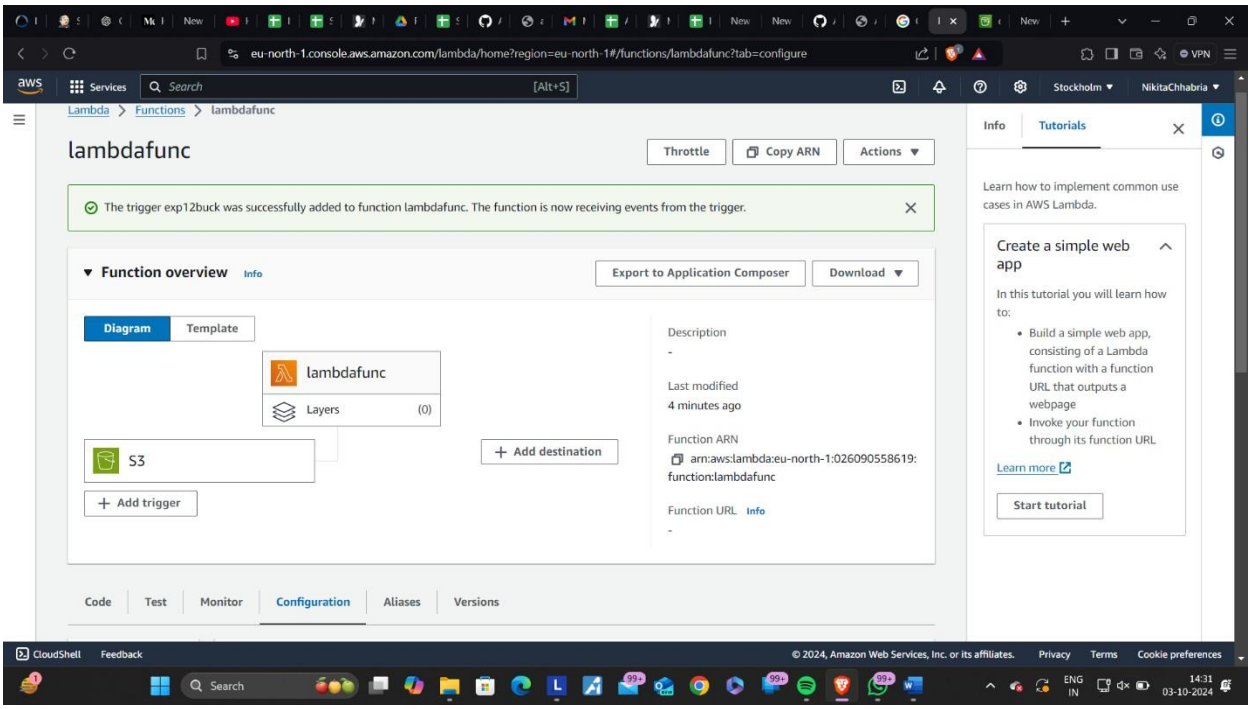
AWS Lambda and S3 Integration: AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3
bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:

1. **Create an S3 Bucket:**
    - First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.
2. **Create the Lambda Function:**
    - Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java.
    - Write code that logs a message like "An Image has been added" when triggered.
3. **Set Up Permissions:**
    - Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.
4. **Configure S3 Trigger:**
    - Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).
5. **Test the Setup:**
    - Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message "An Image has been added" in AWS CloudWatch Logs.

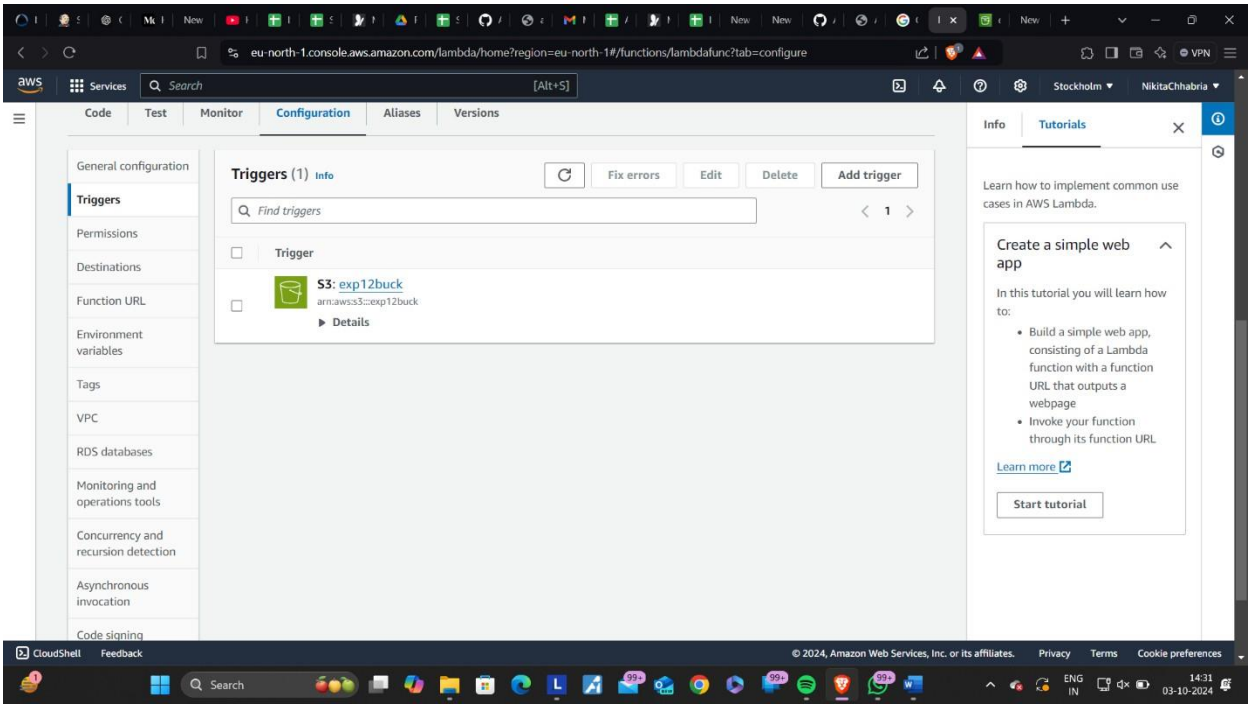
Outcomes:

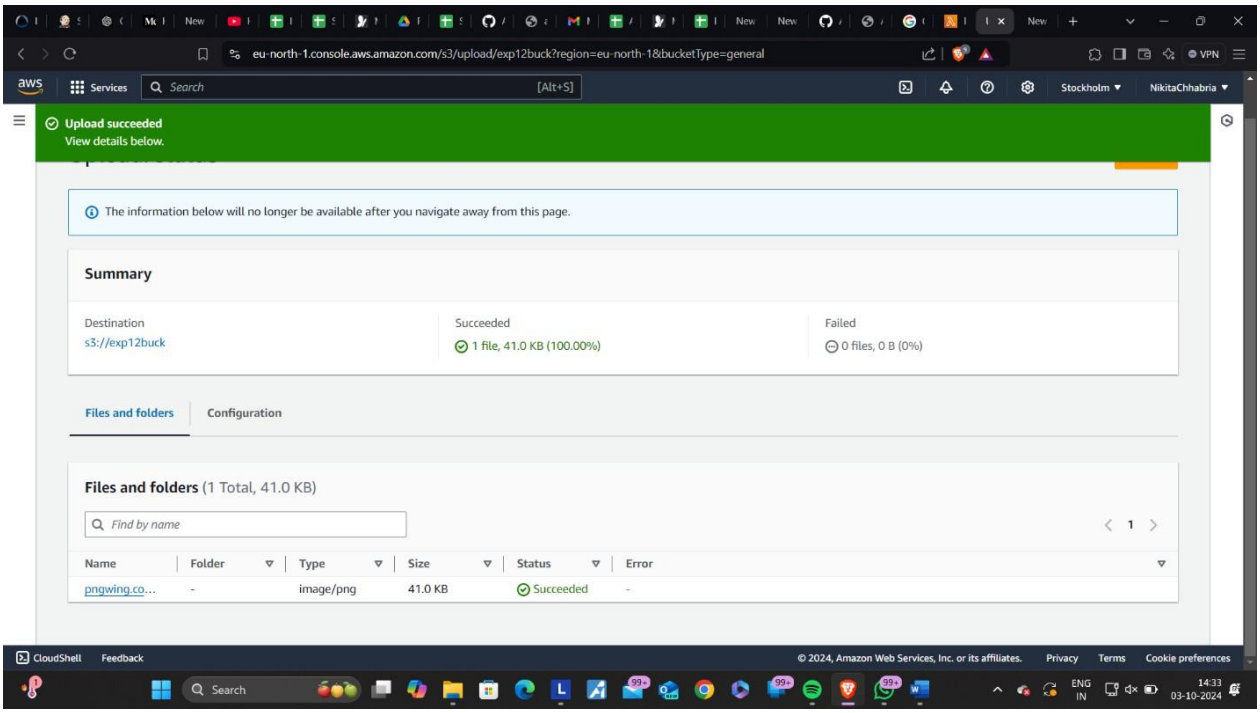

aws
Services
Search
[Alt+S]
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
Europe (Stockholm) eu-north-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
exp12buck
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix

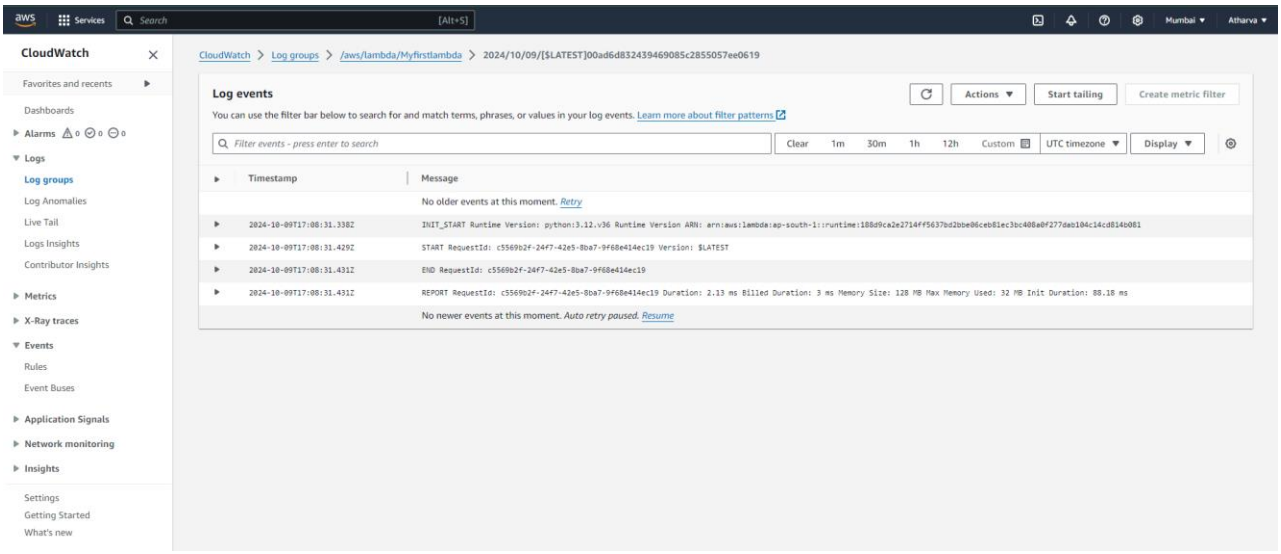

eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc?tab=configure
Lambda > Functions > lambdafunc
lambdafunc
Throttle
Copy ARN
Actions
The trigger exp12buck was successfully added to function lambdafunc. The function is now receiving events from the trigger.
Function overview Info
Export to Application Composer
Download
Diagram
Template
lambdafunc
Layers (0)
S3
+ Add destination
+ Add trigger
Description
-
Last modified
4 minutes ago
Function ARN
arn:aws:lambda:eu-north-1:026090558619:function:lambdafunc
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

lambda_function × Environment Var ×

```python
import json

def lambda_handler(event, context):
    # Extract bucket name and object key from the event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = event['Records'][0]['s3']['object']['key']

    # Log a message
    print(f"An Image has been added to the bucket {bucket_name}: {object_key}")

    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }
```







Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket.