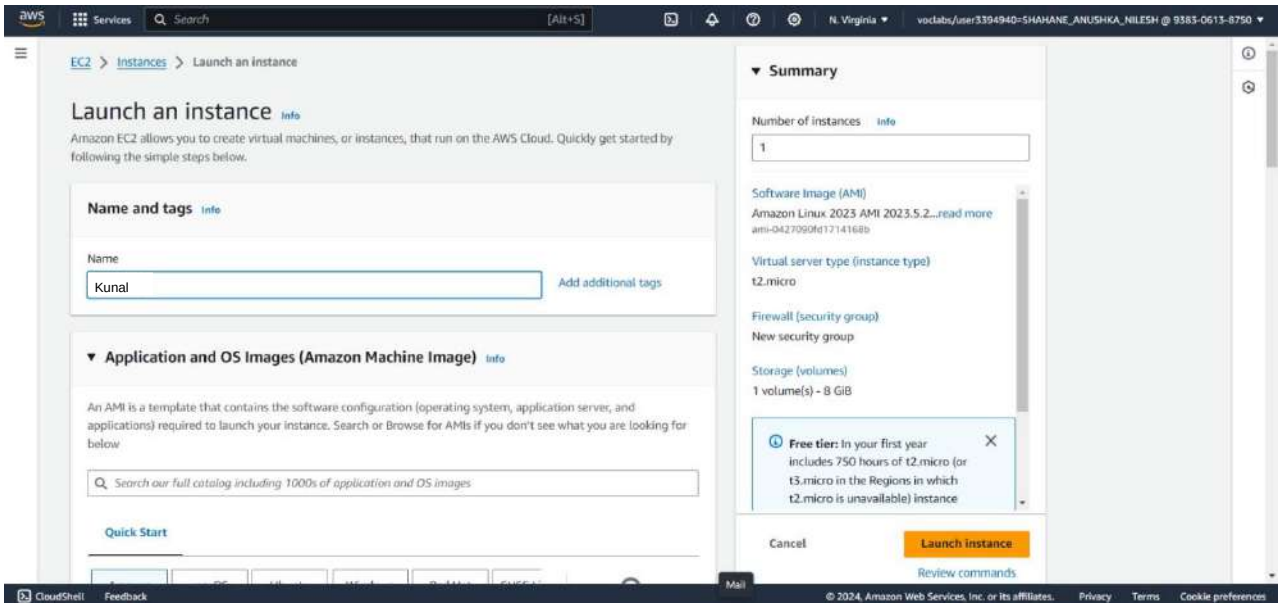
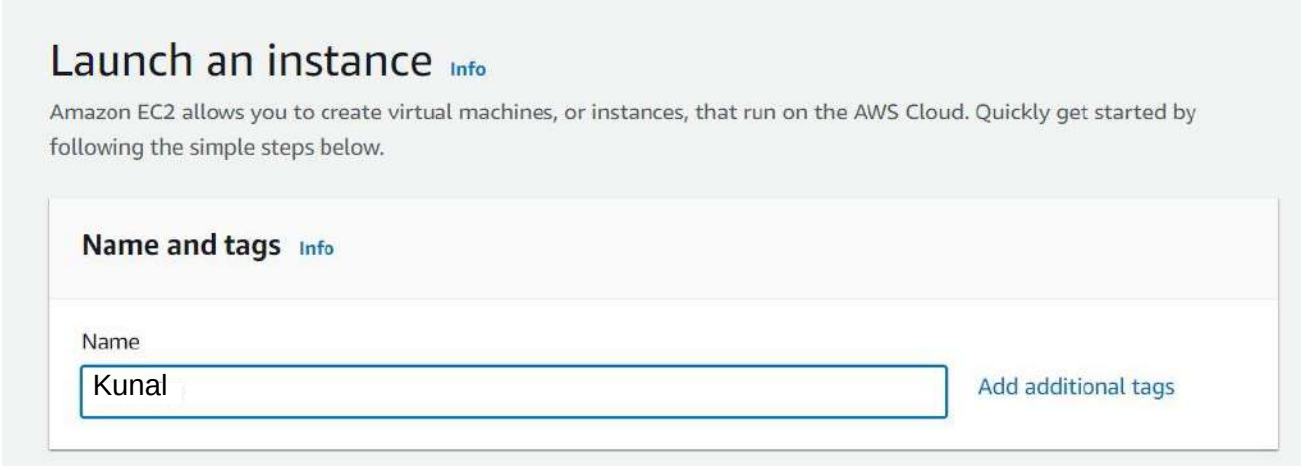
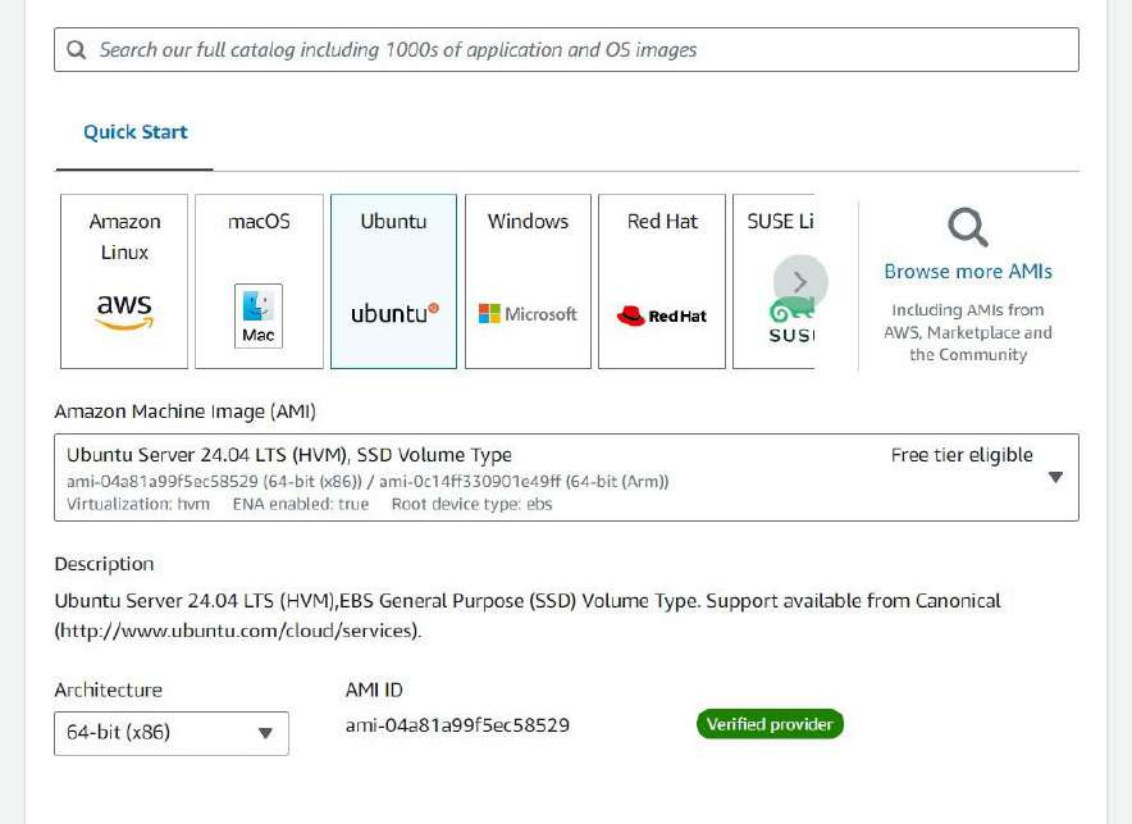
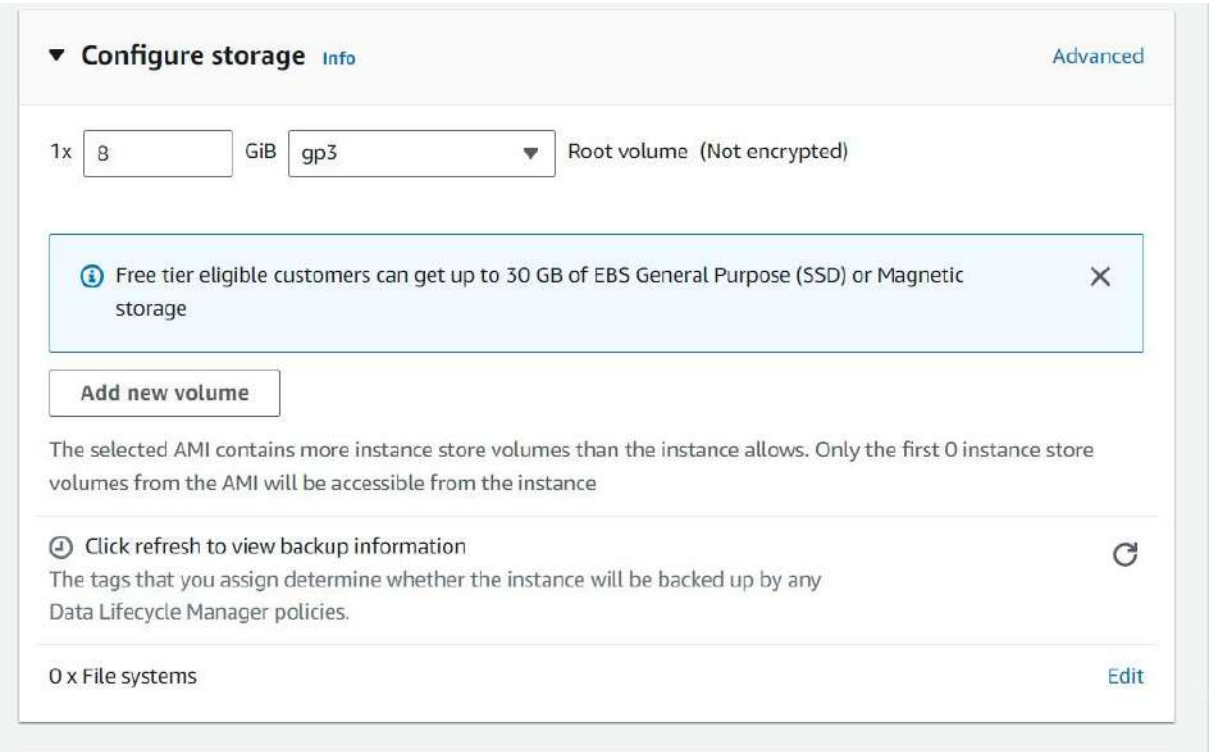
Kunal punjabi D15A 44

ADVANCE DEVOPS EXPERIMENT 1

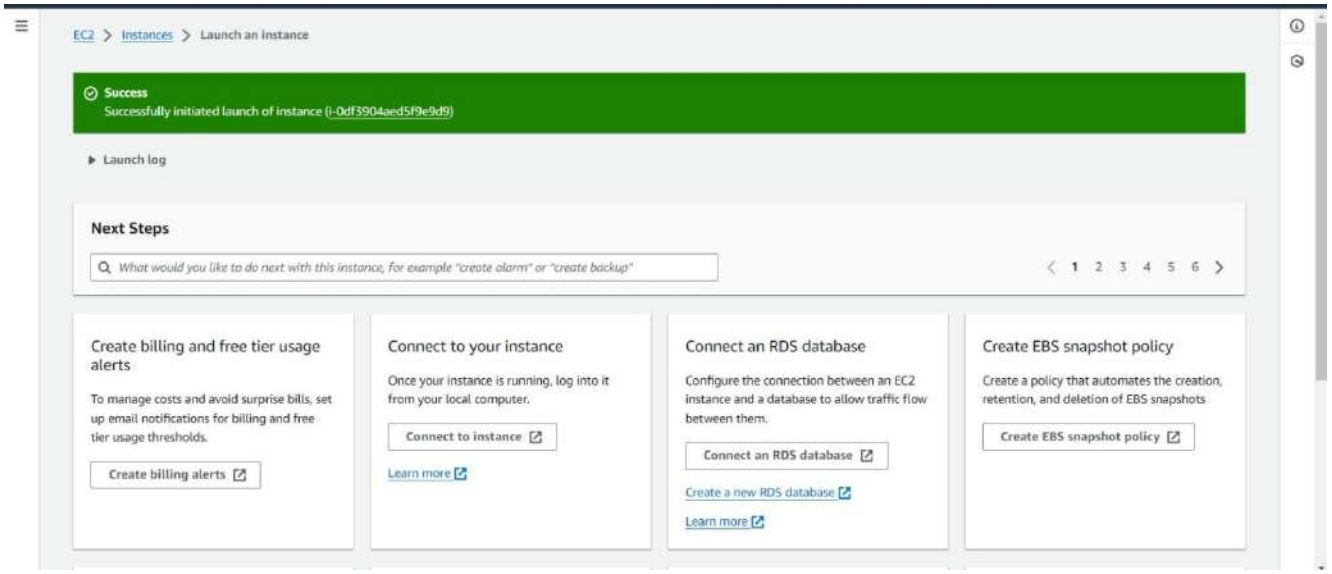


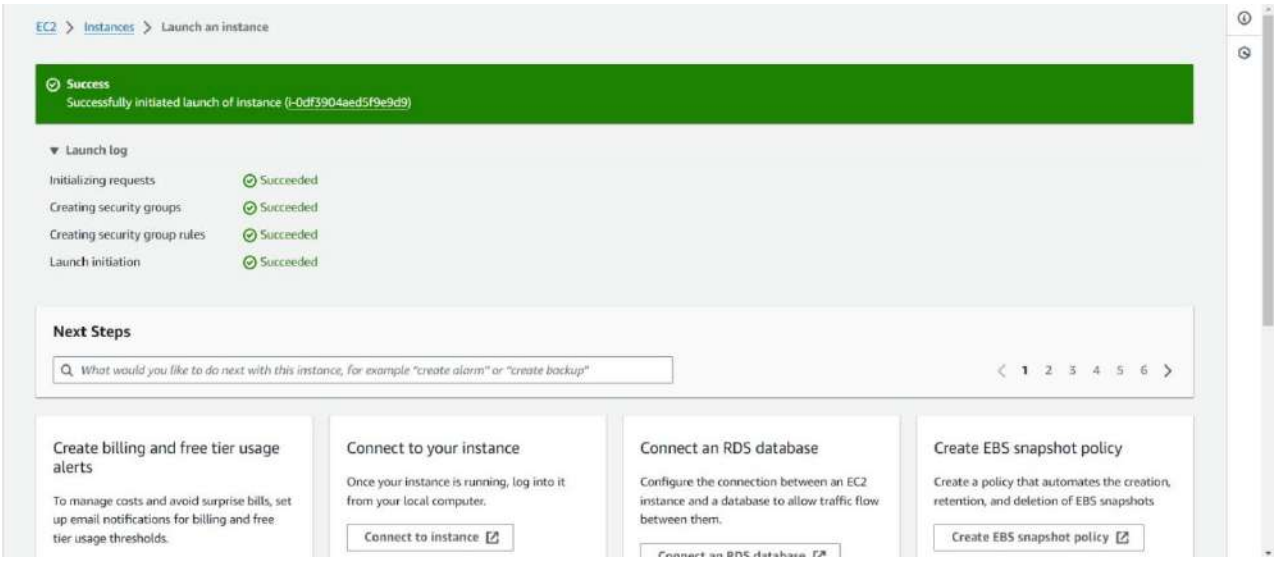



Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Architecture
64-bit (x86)
AMI ID
ami-04a81a99f5ec58529
Verified provider

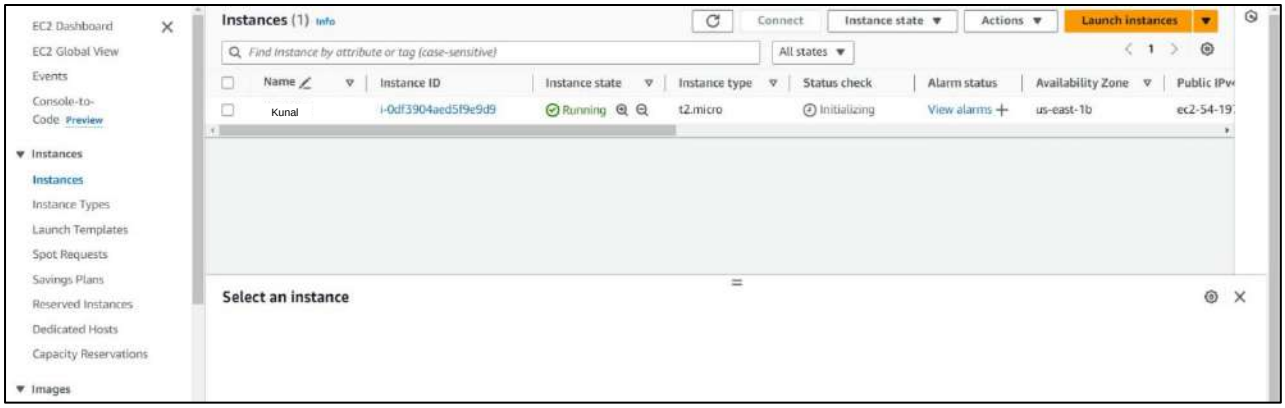

Configure storage Info
Advanced
1x 8 GiB gp3 Root volume (Not encrypted)
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
Add new volume
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance
Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
0 x File systems
Edit

Kunal punjabi D15A 44



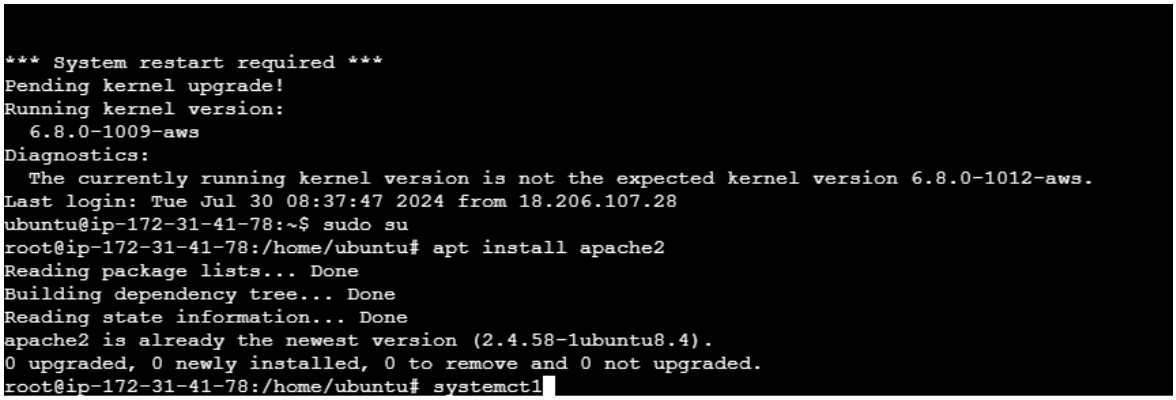


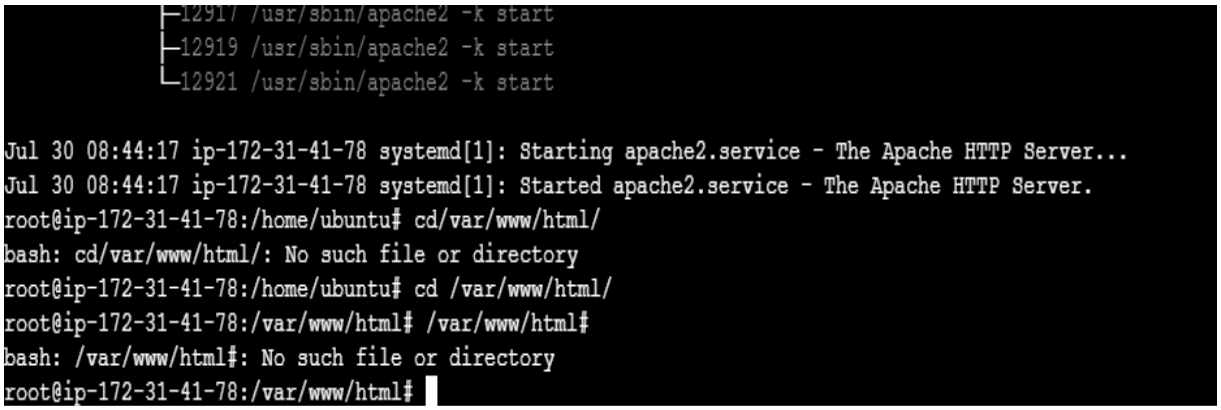
Kunal punjabi D15A 44



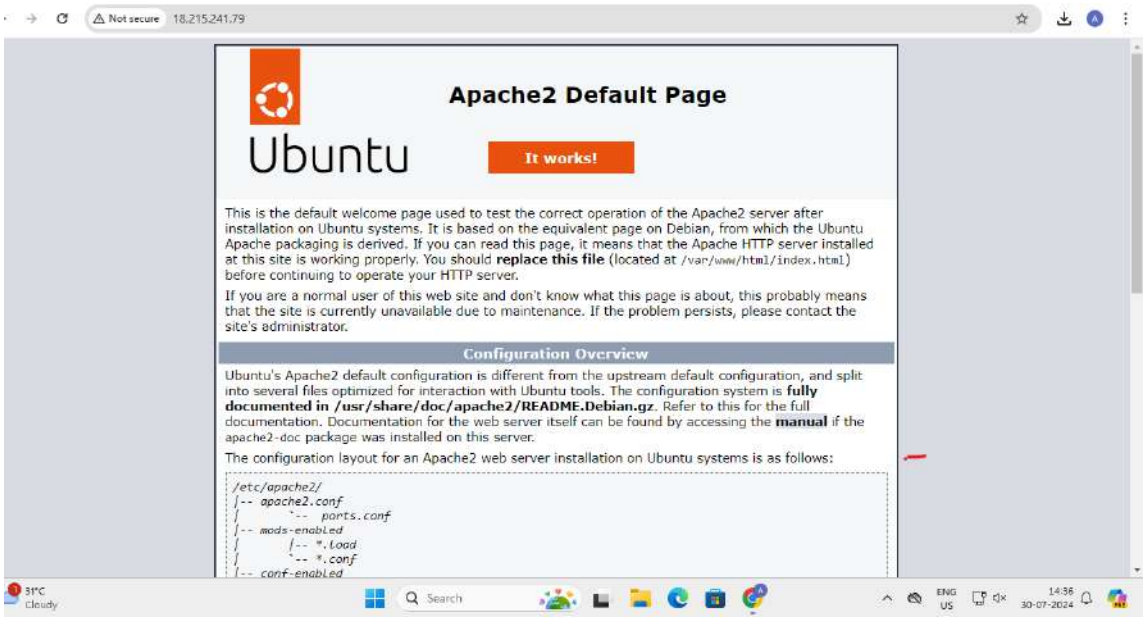

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-42-176:~$ ls
ubuntu@ip-172-31-42-176:~$ echo "hello"
hello
ubuntu@ip-172-31-42-176:~$ cat > myfile.txt
This is Advance devops lab
^C
ubuntu@ip-172-31-42-176:~$ cat myfile
cat: myfile: No such file or directory
ubuntu@ip-172-31-42-176:~$ cat myfile.txt
This is Advance devops lab
ubuntu@ip-172-31-42-176:~$
```

Hosting a static website using EC2 instance:

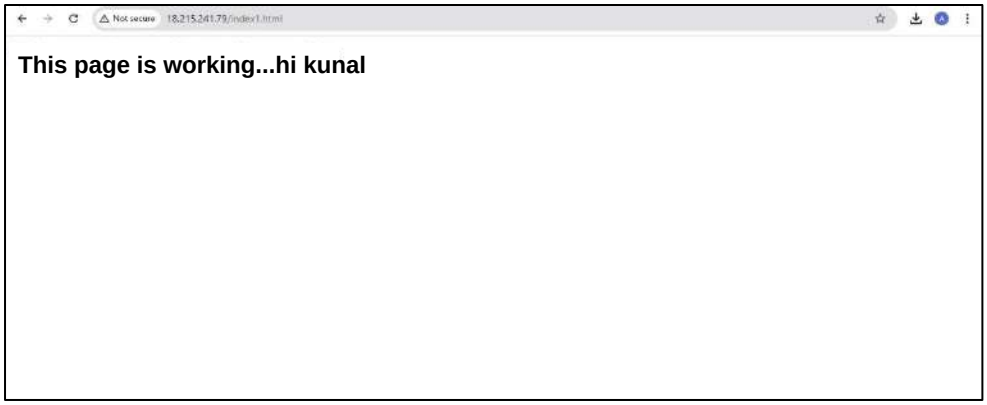




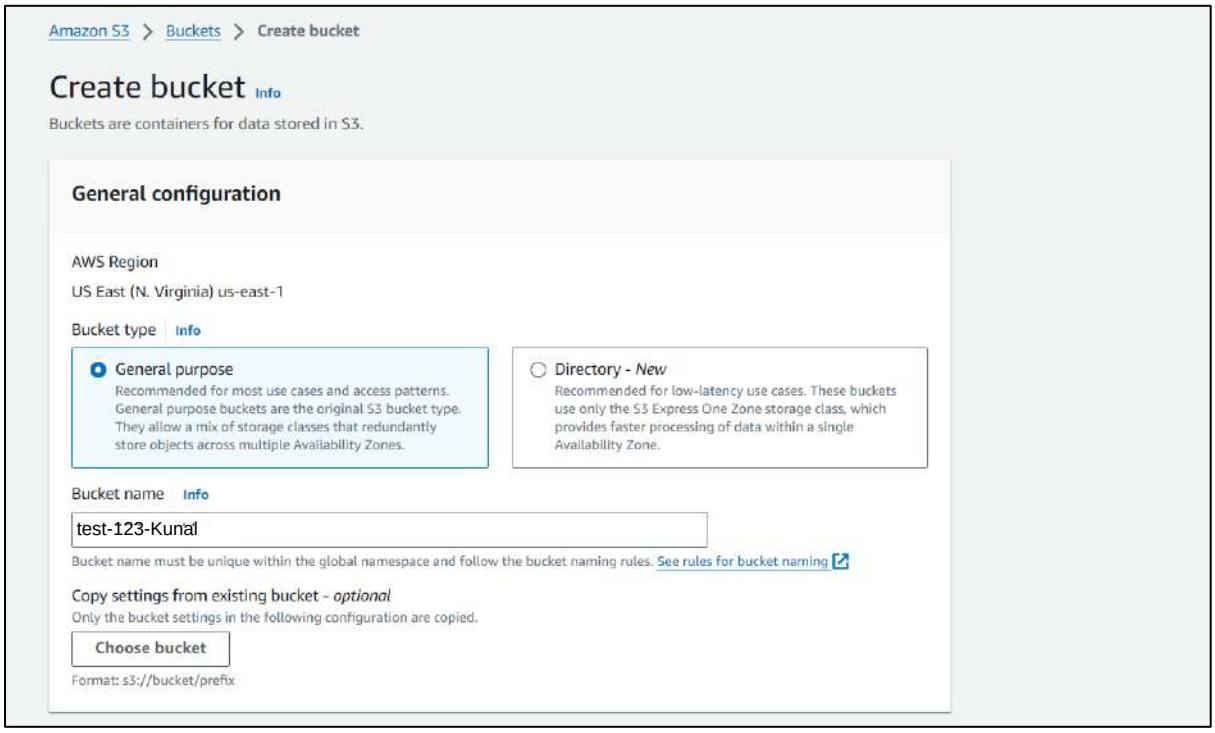
```
  command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: apt install <deb name>
root@ip-172-31-41-78:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-30 08:44:17 UTC; 12min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 12917 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.3M (peak: 5.4M)
        CPU: 74ms
     CGroup: /system.slice/apache2.service
             ├─12917 /usr/sbin/apache2 -k start
             ├─12919 /usr/sbin/apache2 -k start
             └─12921 /usr/sbin/apache2 -k start

Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-78:/home/ubuntu#
```

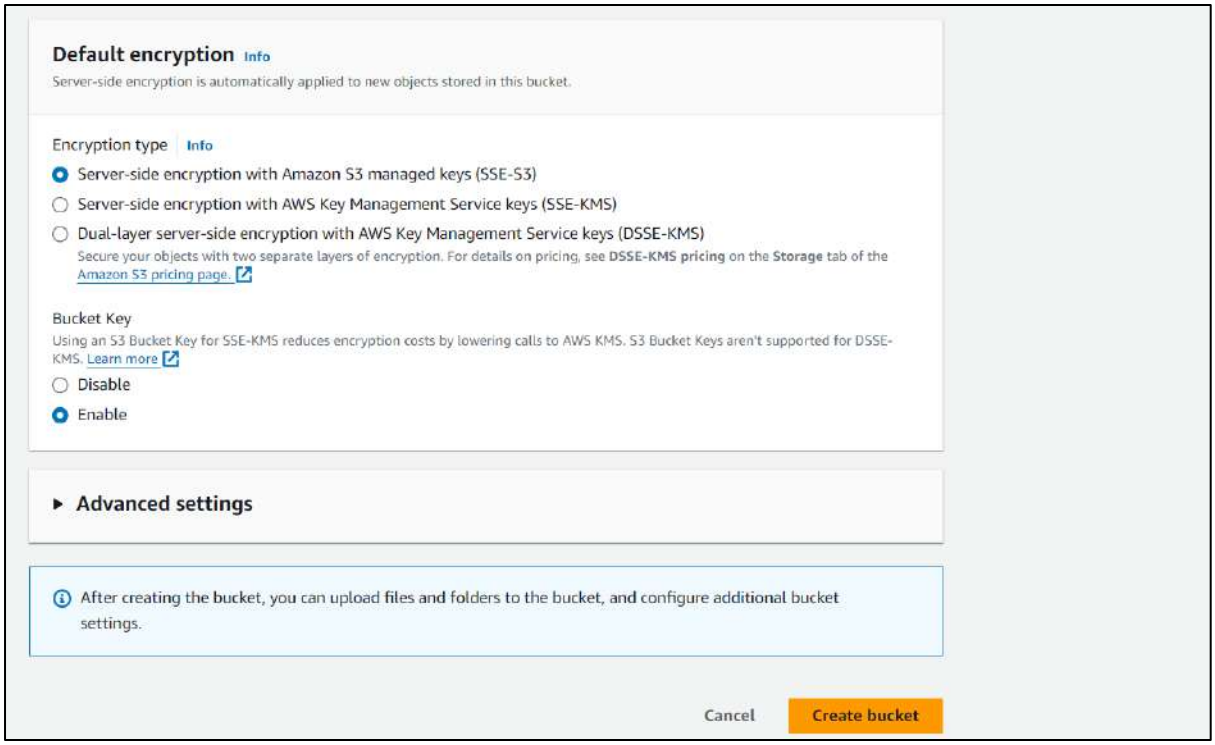


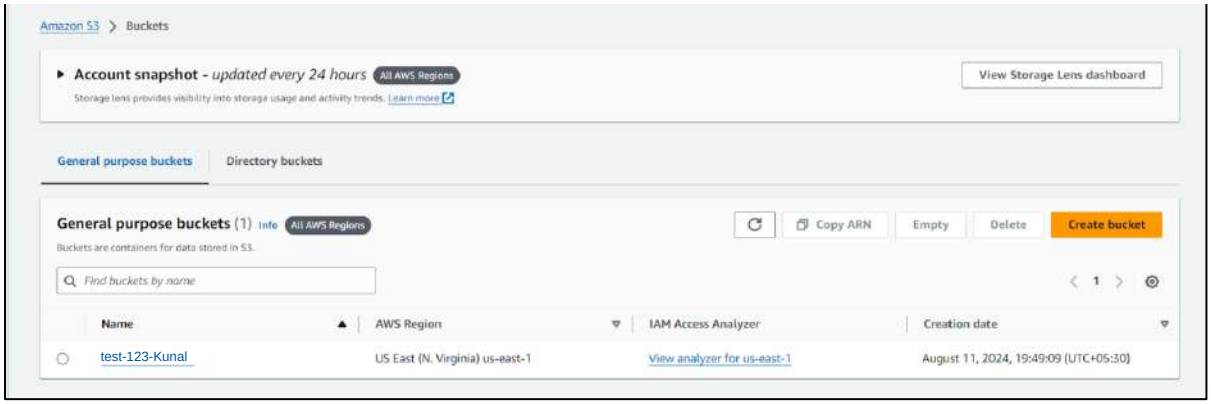
Kunal punjabi D15A 44



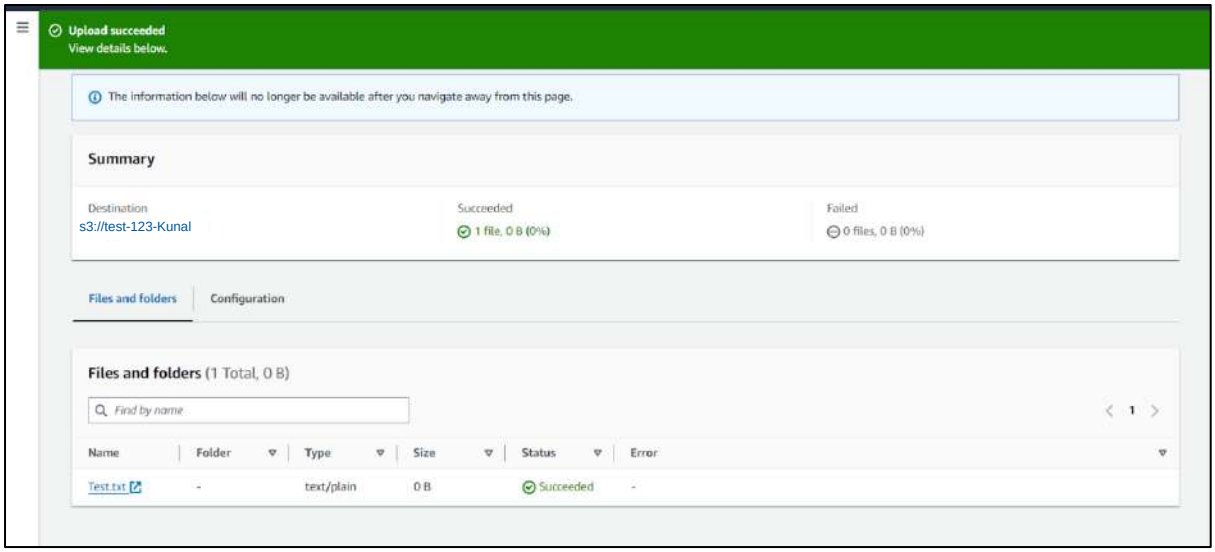
Hosting using S3 bucket :

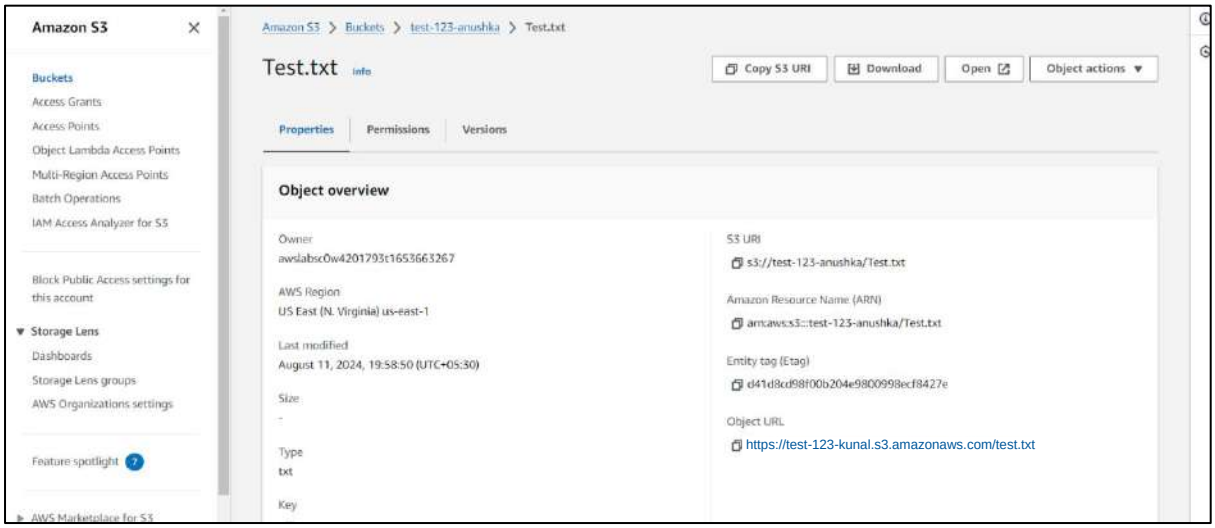



Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

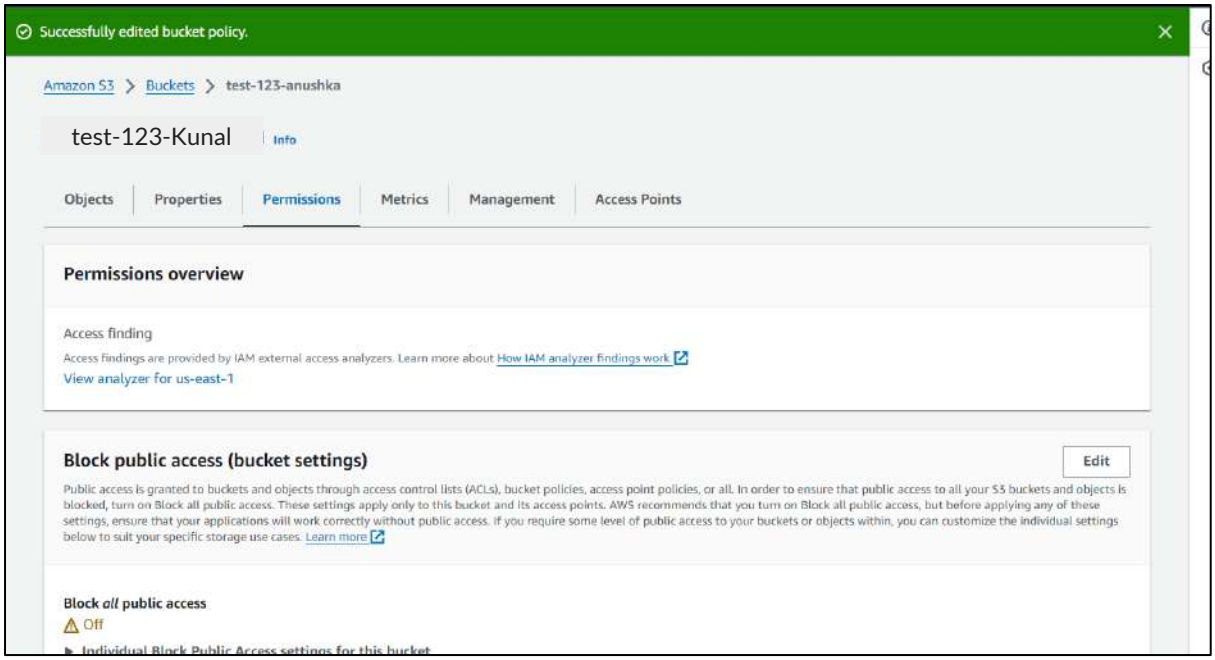

Amazon S3 > Buckets
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
test-123-Kunal
US East (N. Virginia) us-east-1
View analyzer for us-east-1
August 11, 2024, 19:49:09 (UTC+05:30)

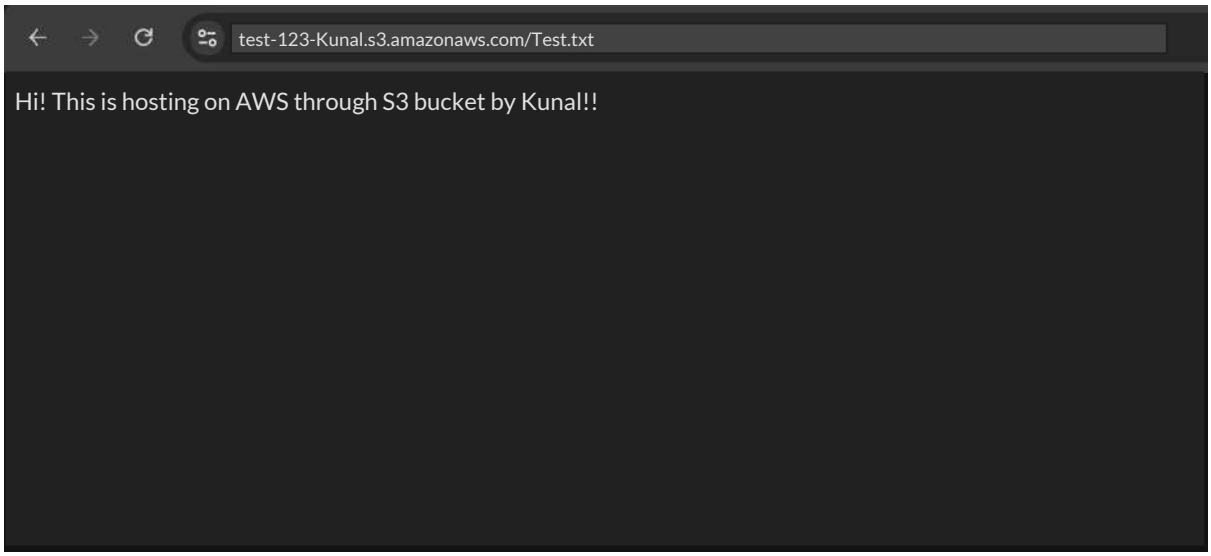
Kunal punjabi D15A 44



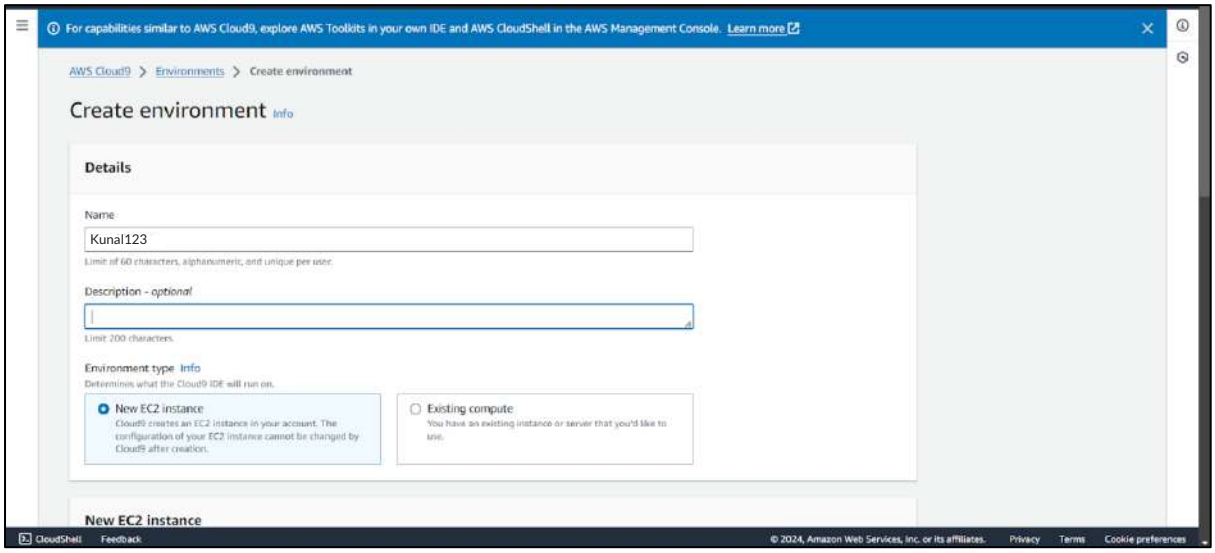


Kunal punjabi D15A 44

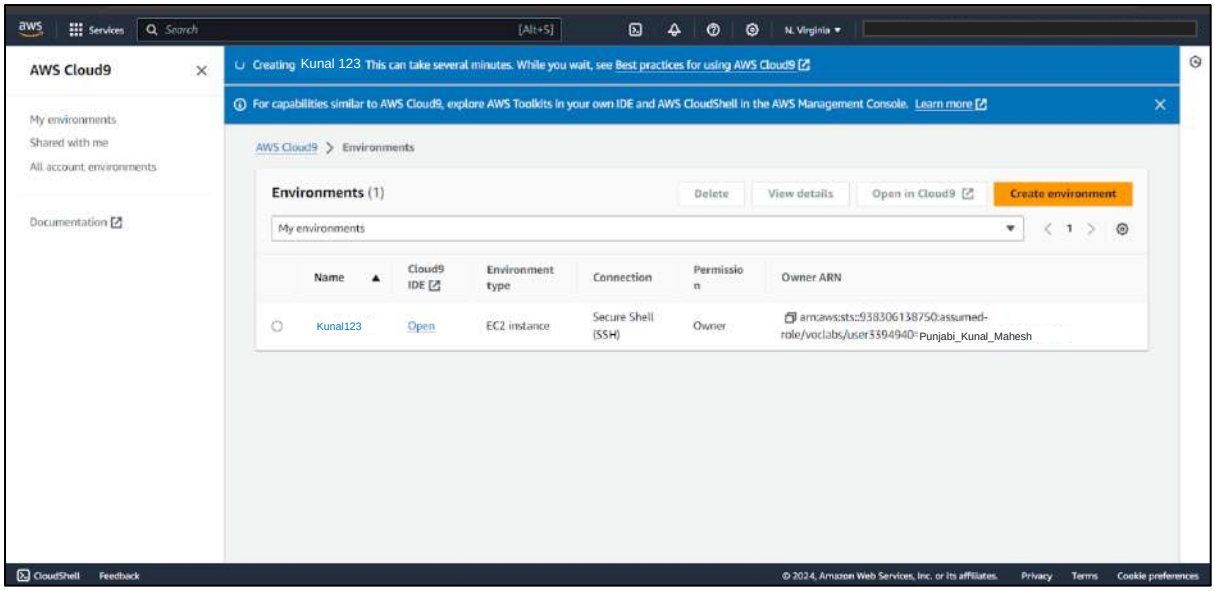


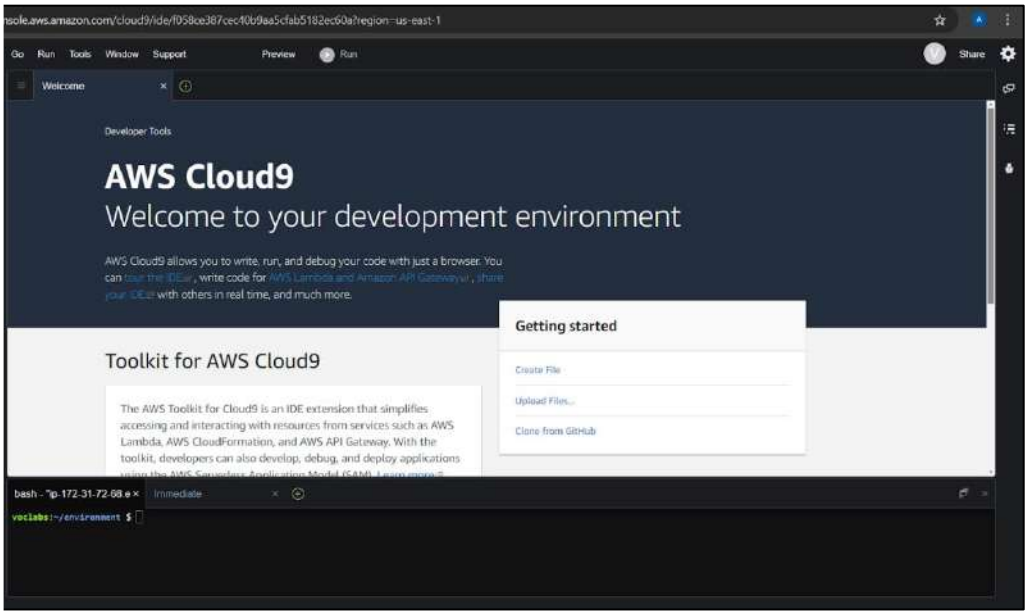


Hosting using Cloud 9 :

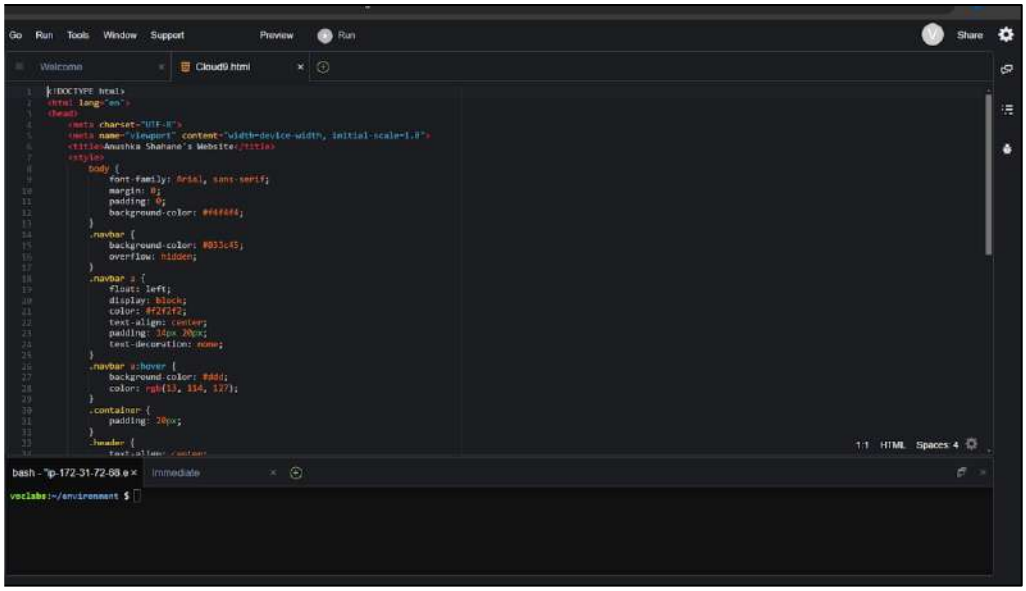


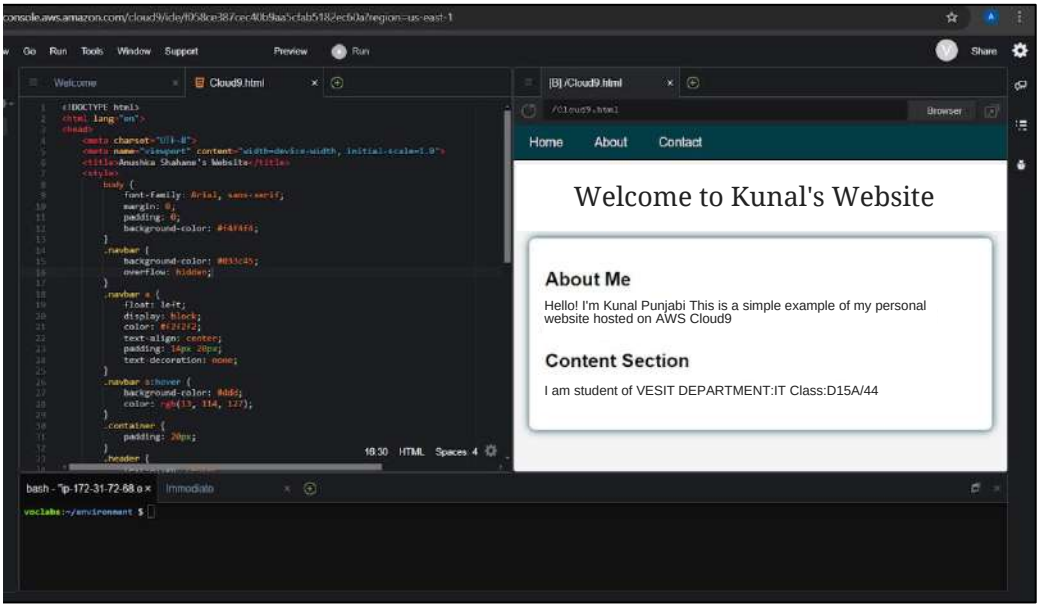
Kunal punjabi D15A 44





Kunal punjabi D15A 44





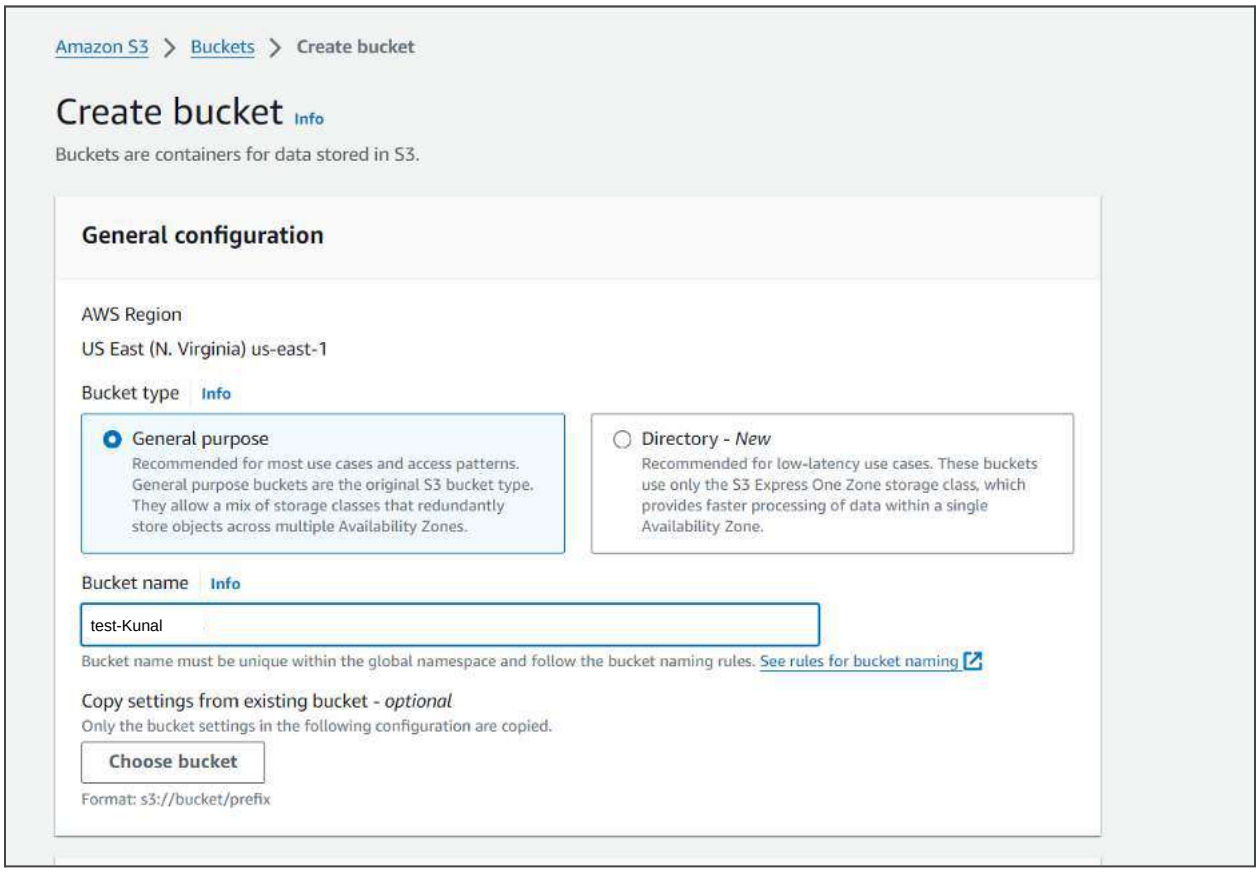
Experiment 2

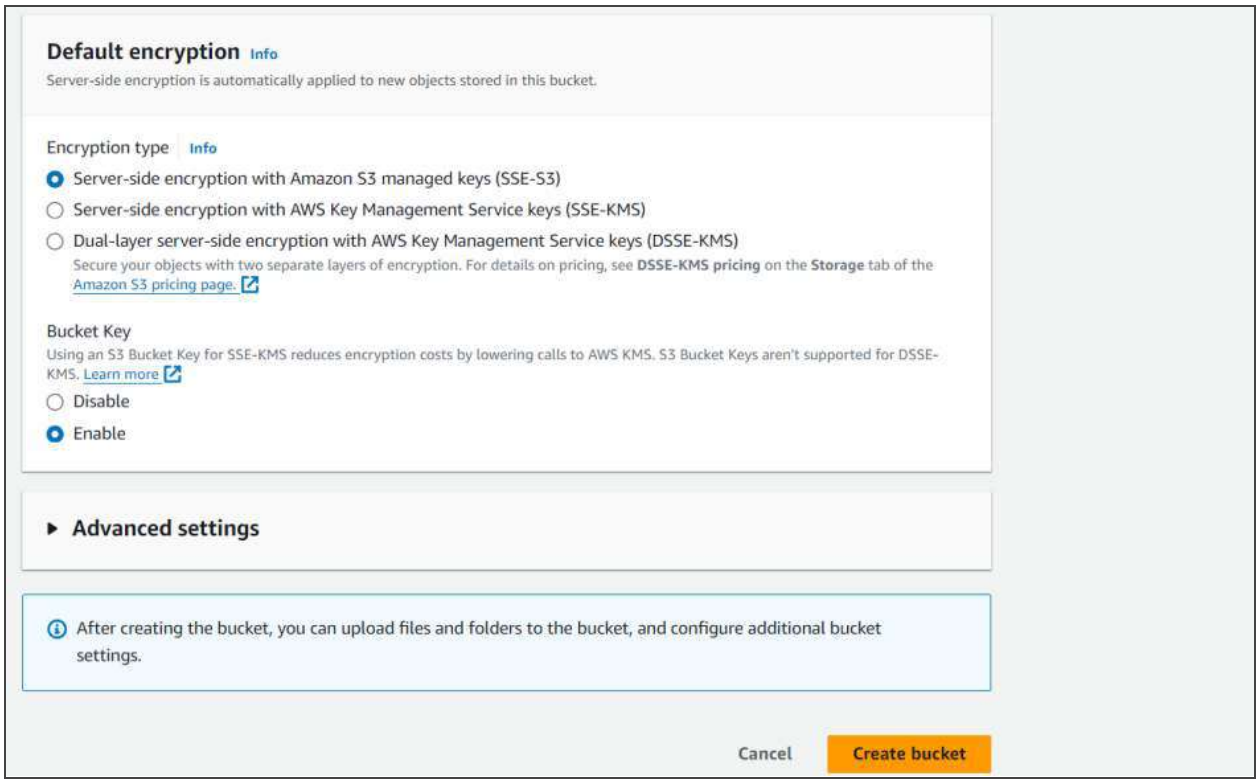
Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

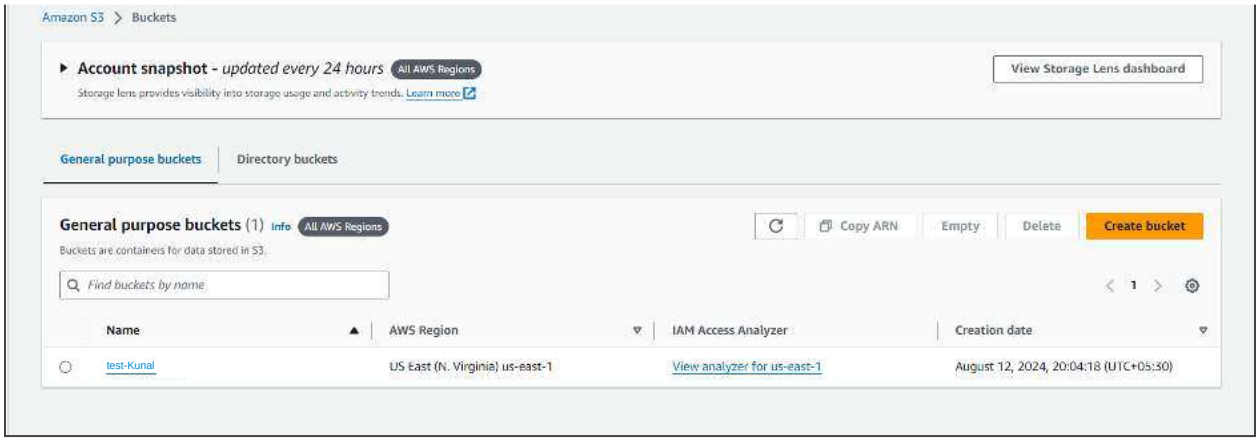
Contents:

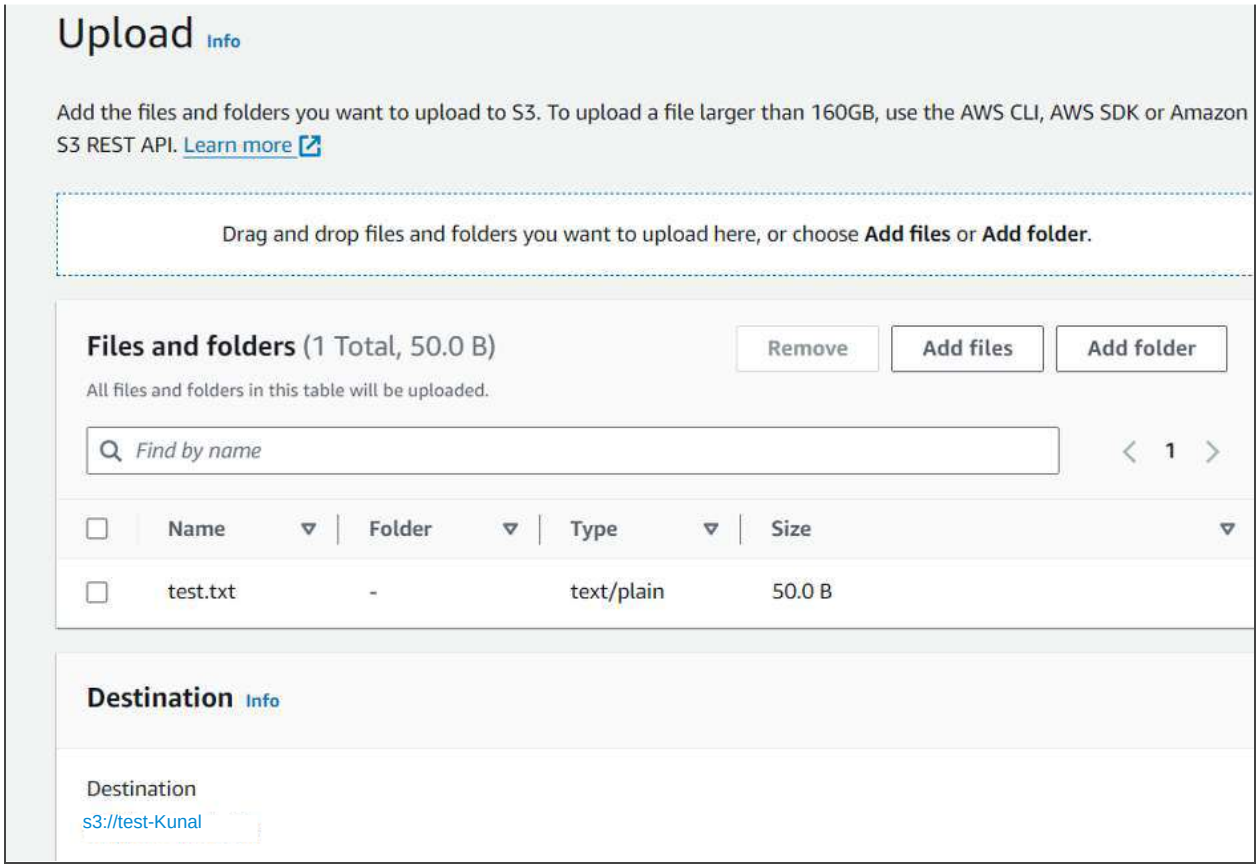
1. s3 bucket
2. ec2 instance
3. elastic beanstalk

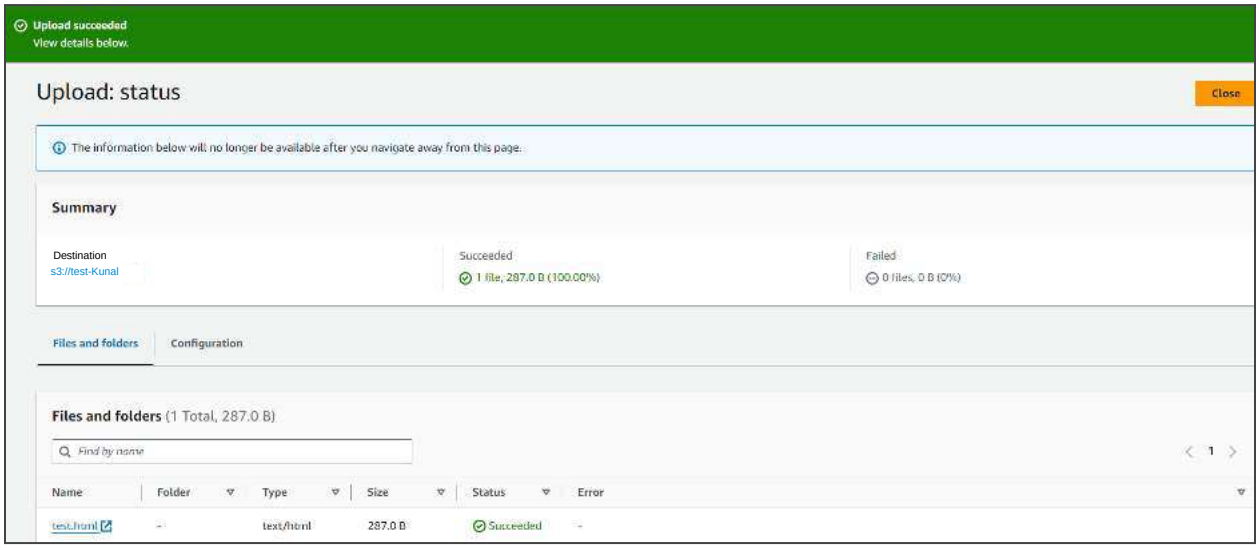
- s3 bucket

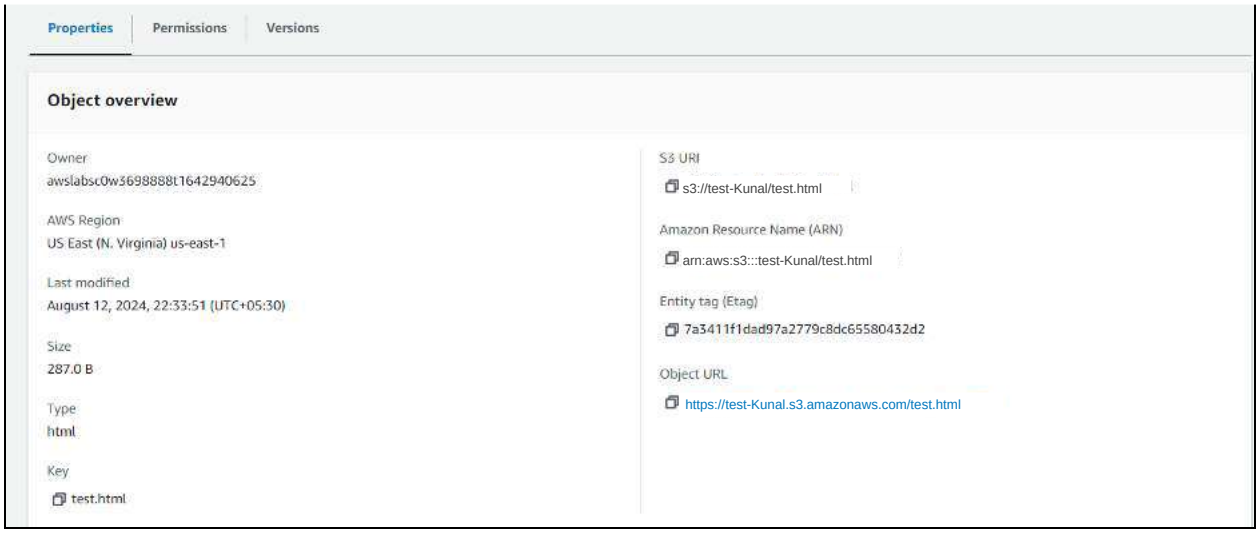


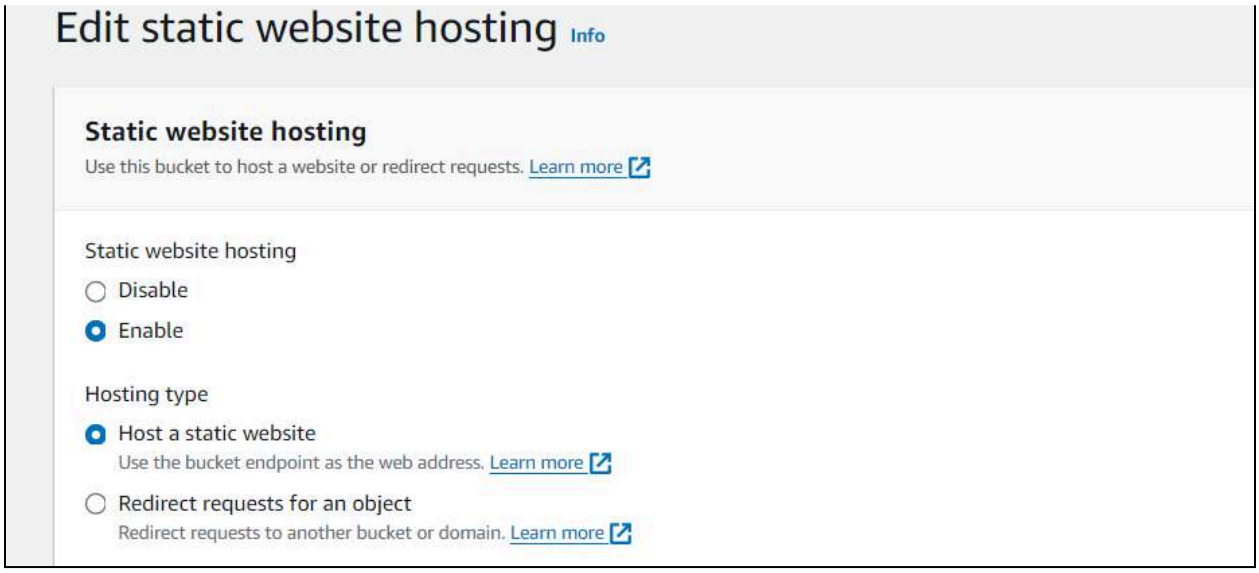

Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

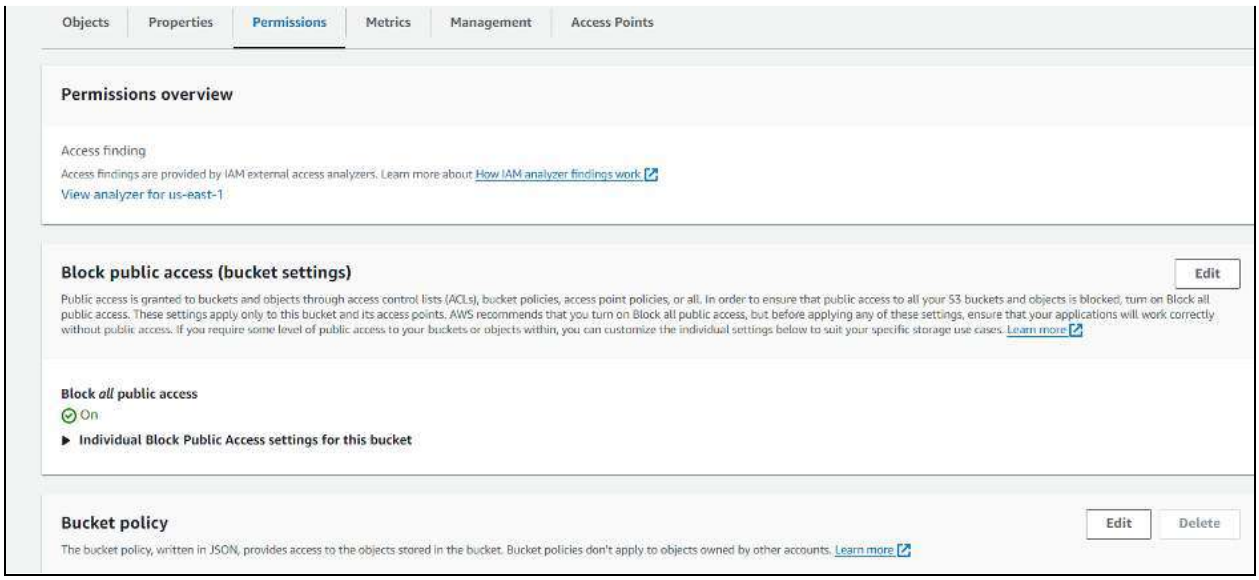

Amazon S3 > Buckets
Account snapshot - updated every 24 hours All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
test-Kunal
US East (N. Virginia) us-east-1
View analyzer for us-east-1
August 12, 2024, 20:04:18 (UTC+05:30)

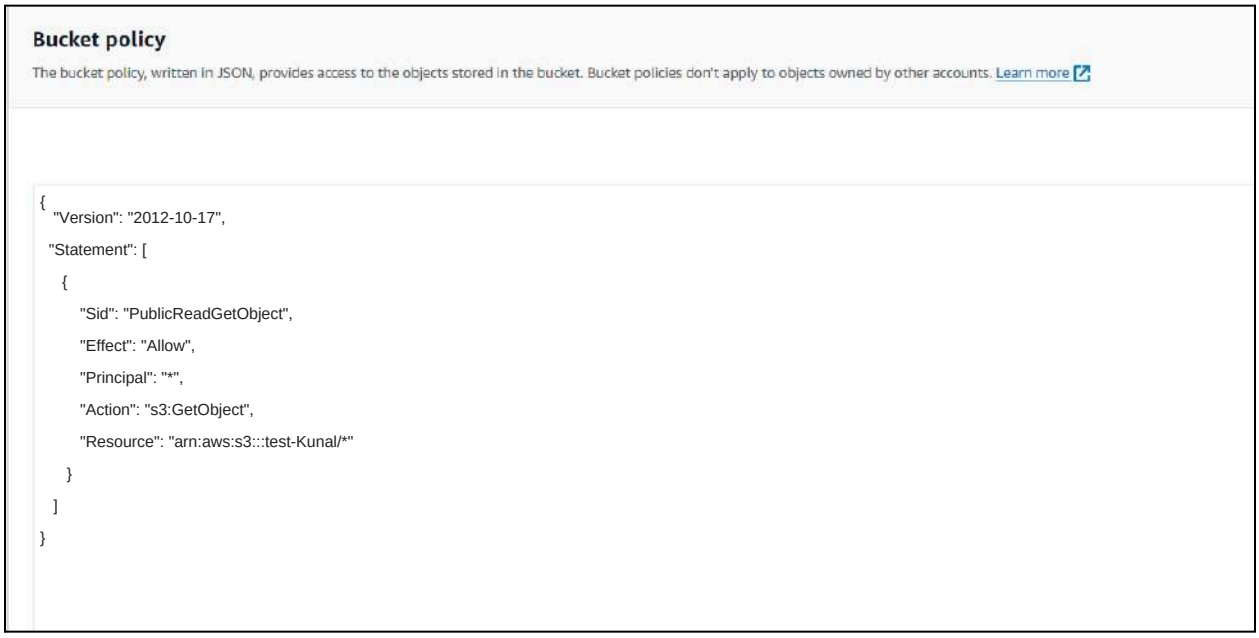

Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 50.0 B)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Type
Size
test.txt
-
text/plain
50.0 B
Destination Info
Destination
s3://test-Kunal

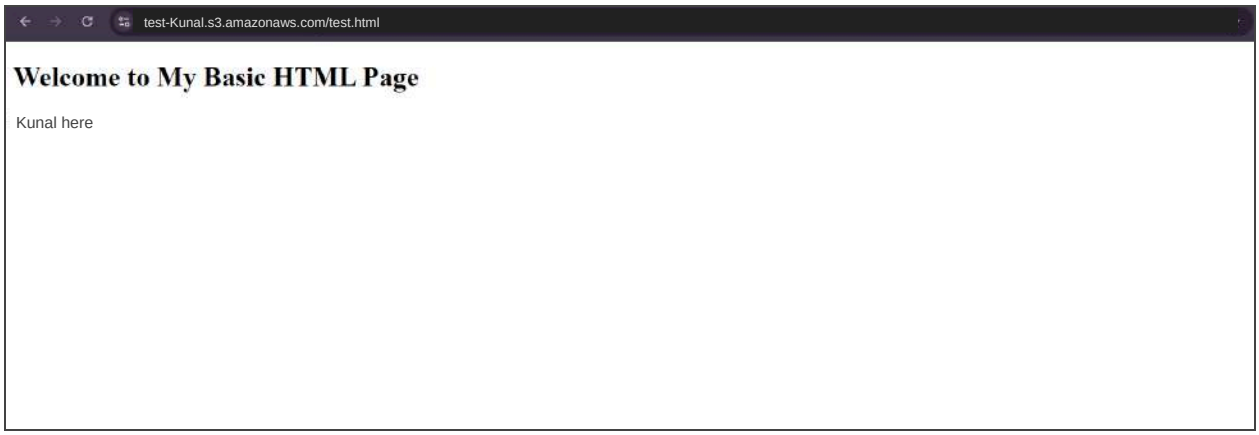

Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://test-Kunal
Succeeded
1 file, 287.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 287.0 B)
Find by name
1
Name
Folder
Type
Size
Status
Error
test.html
-
text/html
287.0 B
Succeeded
-


Properties
Permissions
Versions
Object overview
Owner
awslabsc0w3698888t1642940625
AWS Region
US East (N. Virginia) us-east-1
Last modified
August 12, 2024, 22:33:51 (UTC+05:30)
Size
287.0 B
Type
html
Key
test.html
S3 URI
s3://test-Kunal/test.html
Amazon Resource Name (ARN)
arn:aws:s3:::test-Kunal/test.html
Entity tag (Etag)
7a3411f1dad97a2779c8dc65580432d2
Object URL
https://test-Kunal.s3.amazonaws.com/test.html

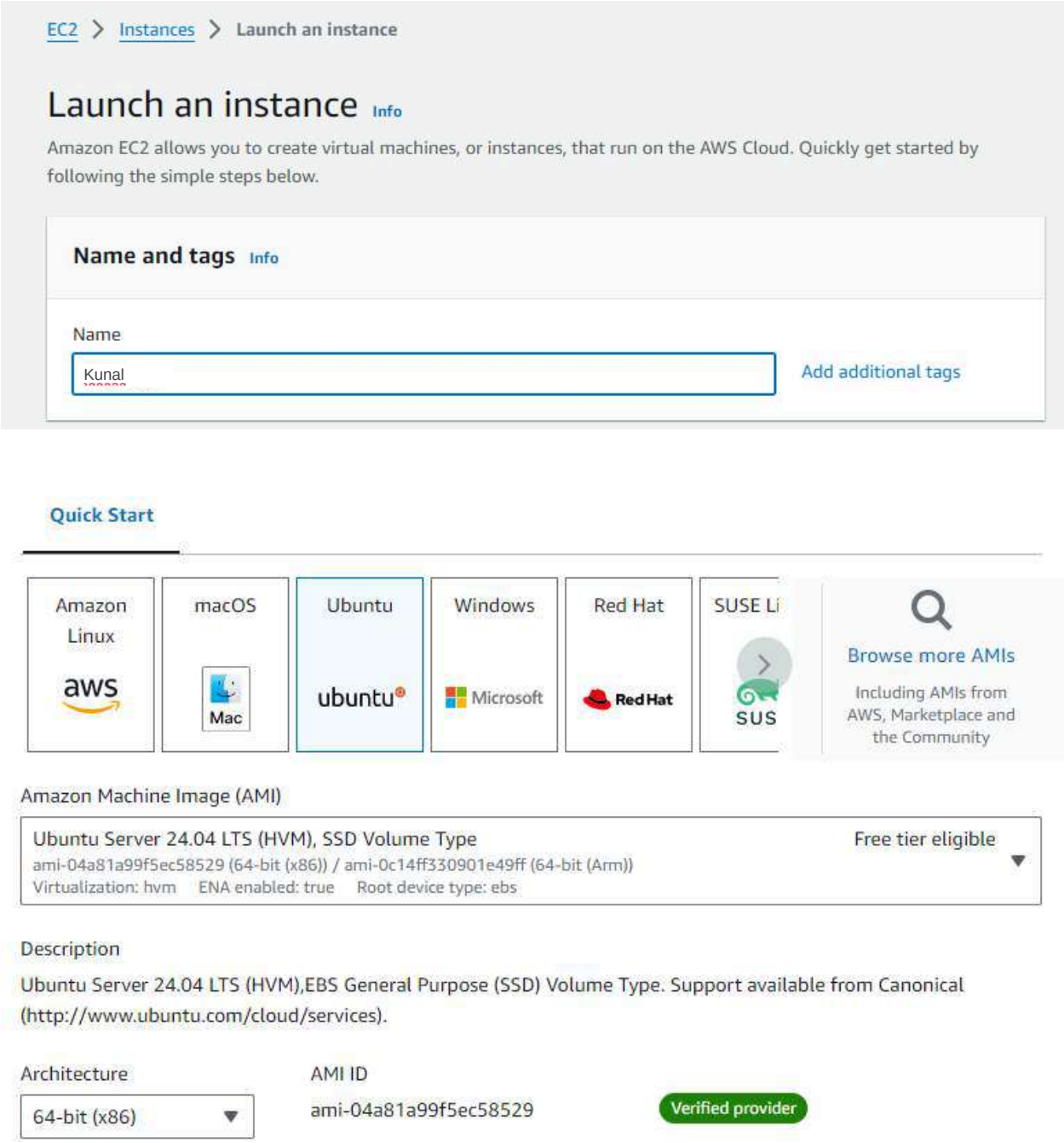

Edit static website hosting
Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more


Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for us-east-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
On
Individual Block Public Access settings for this bucket
Bucket policy
Edit
Delete
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more


Bucket policy
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "PublicReadGetObject",
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::test-Kunal/*"
}
]
}


test-Kunal.s3.amazonaws.com/test.html
Welcome to My Basic HTML Page
Kunal here

- Launching an EC2 instance



Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

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

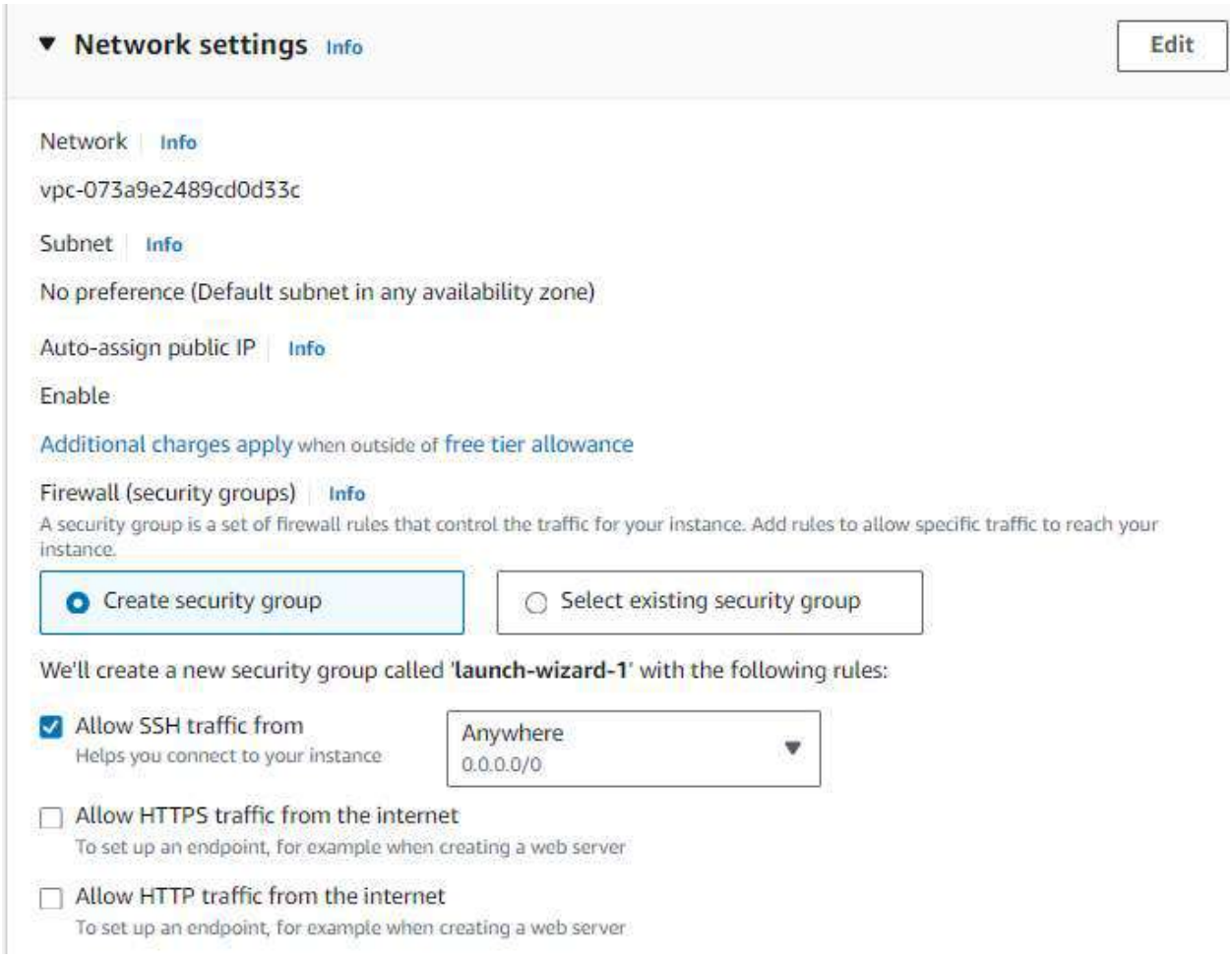
Add new volume

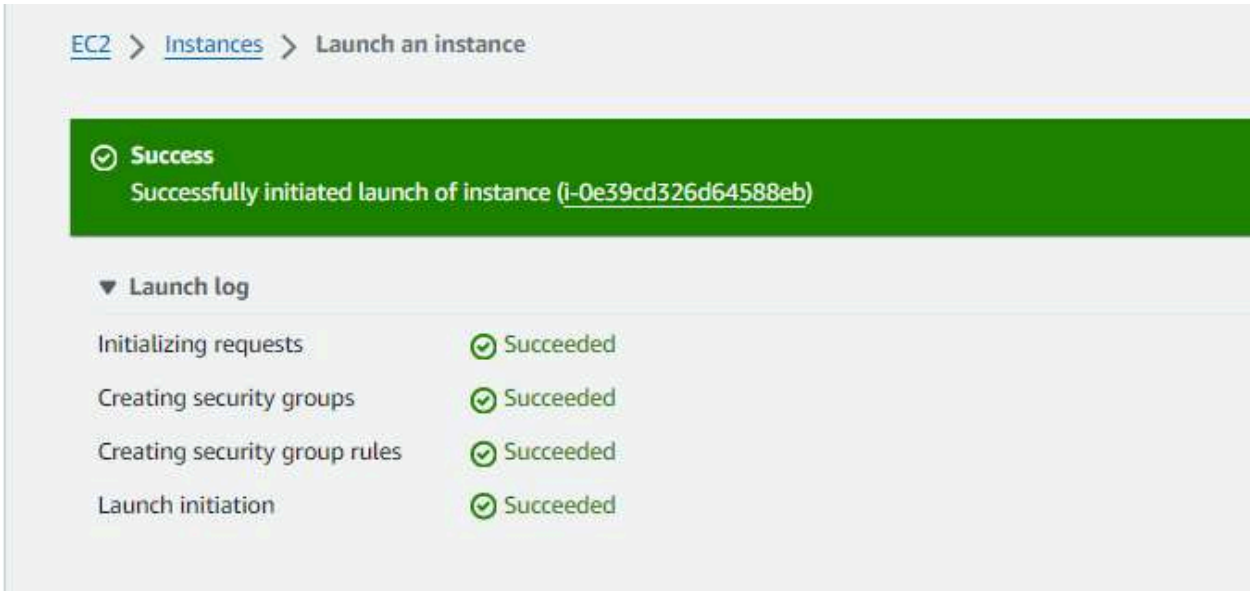
Click refresh to view backup information

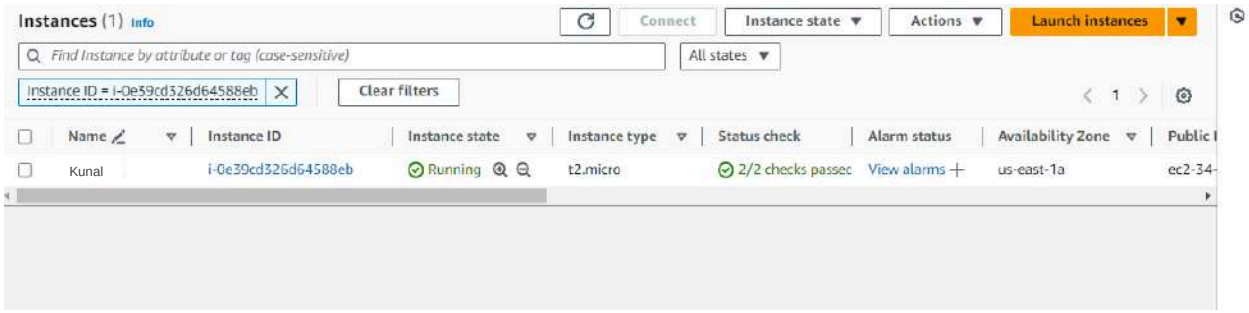
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

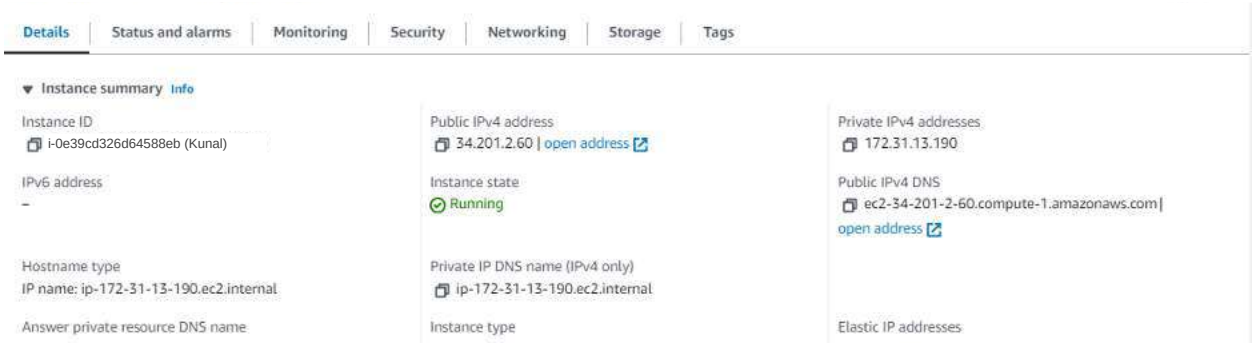
0 x File systems

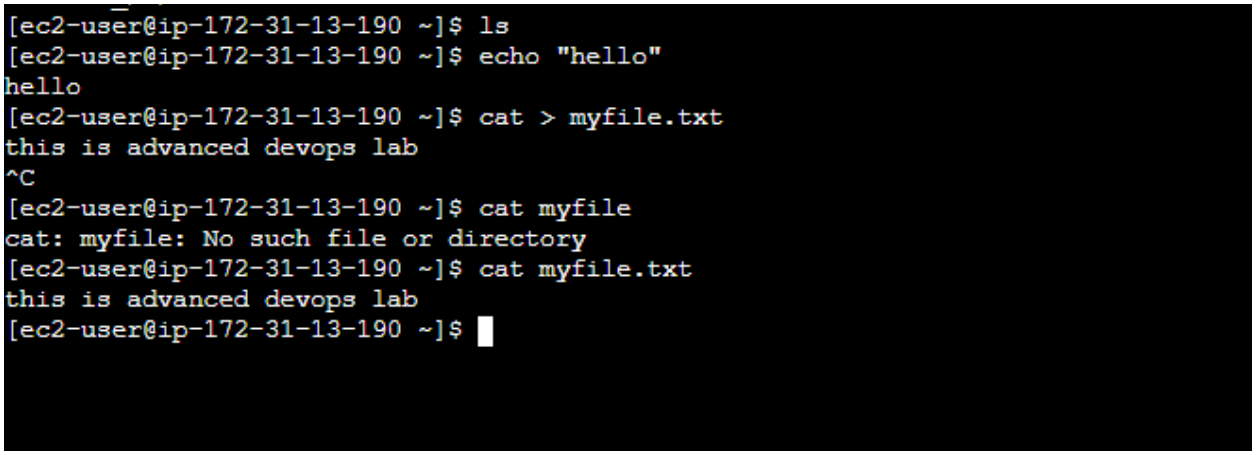
Edit


Network settings Info
Edit
Network Info
vpc-073a9e2489cd0d33c
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-1' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server


EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-0e39cd326d64588eb)
Launch log
Initializing requests
Succeeded
Creating security groups
Succeeded
Creating security group rules
Succeeded
Launch initiation
Succeeded


Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance ID = i-0e59cd326d64588eb
Clear filters
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public I
Kunal
i-0e39cd326d64588eb
Running
t2.micro
2/2 checks passed
View alarms
us-east-1a
ec2-34-

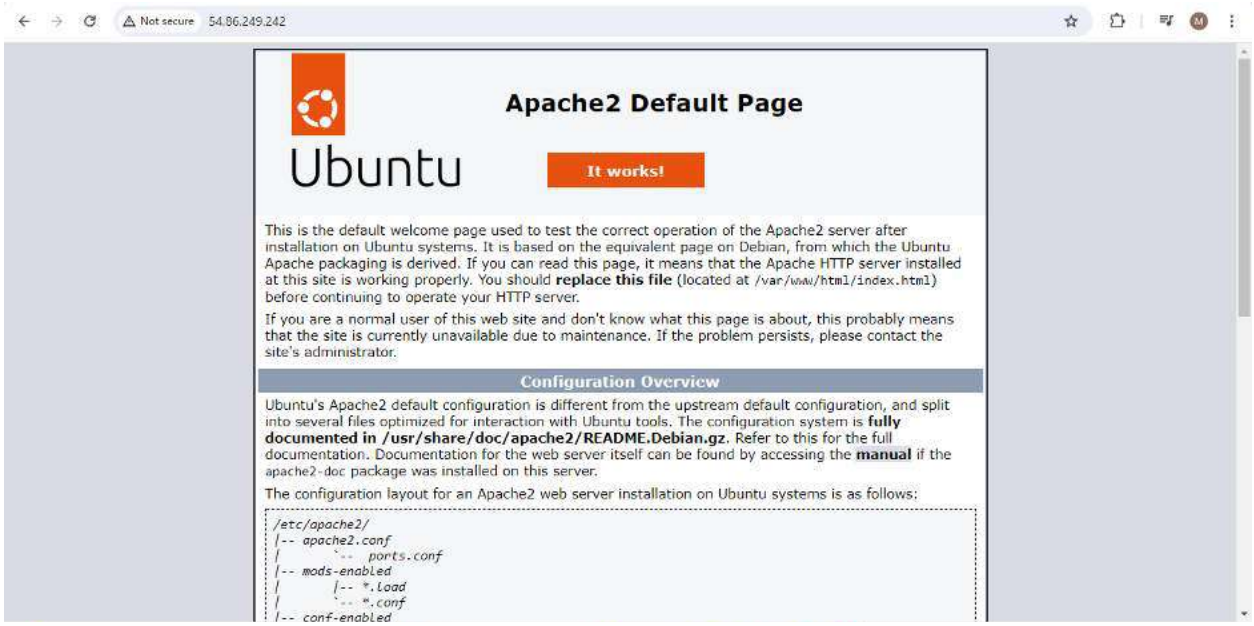

Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0e39cd326d64588eb (Kunal)
Public IPv4 address
34.201.2.60 | open address
Private IPv4 addresses
172.31.13.190
IPv6 address
-
Instance state
Running
Public IPv4 DNS
ec2-34-201-2-60.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-13-190.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-13-190.ec2.internal
Answer private resource DNS name
Instance type
Elastic IP addresses


[ec2-user@ip-172-31-13-190 ~]$ ls
[ec2-user@ip-172-31-13-190 ~]$ echo "hello"
hello
[ec2-user@ip-172-31-13-190 ~]$ cat > myfile.txt
this is advanced devops lab
^C
[ec2-user@ip-172-31-13-190 ~]$ cat myfile
cat: myfile: No such file or directory
[ec2-user@ip-172-31-13-190 ~]$ cat myfile.txt
this is advanced devops lab
[ec2-user@ip-172-31-13-190 ~]$

```
root@ip-172-31-32-173:~# sudo su
root@ip-172-31-32-173:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqli
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browse
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-
0 upgraded, 10 newly installed, 0 to remove and 26 not upgraded.
Need to get 1680 kB/2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
```

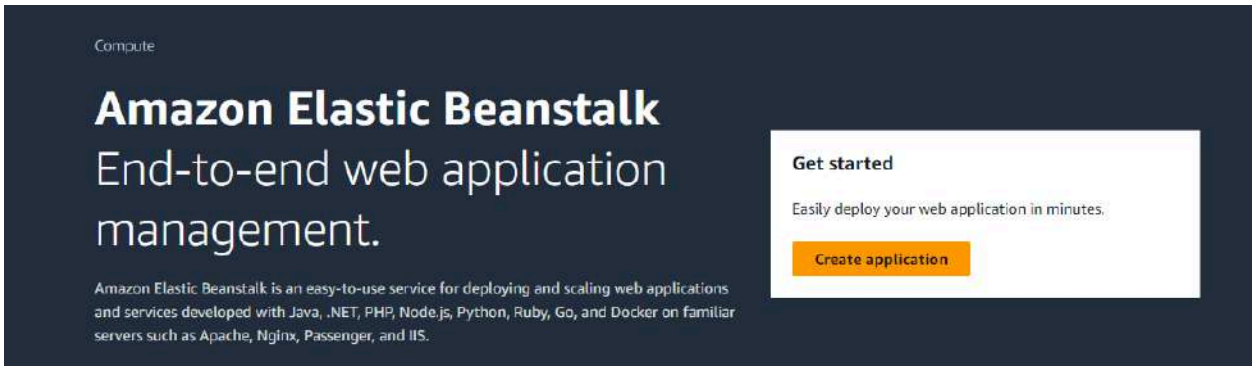
```
root@ip-172-31-32-173:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-30 08:58:11 UTC; 44s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2619 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.4M (peak: 5.5M)
        CPU: 40ms
     CGroup: /system.slice/apache2.service
             ├─2619 /usr/sbin/apache2 -k start
             ├─2621 /usr/sbin/apache2 -k start
             └─2623 /usr/sbin/apache2 -k start

Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

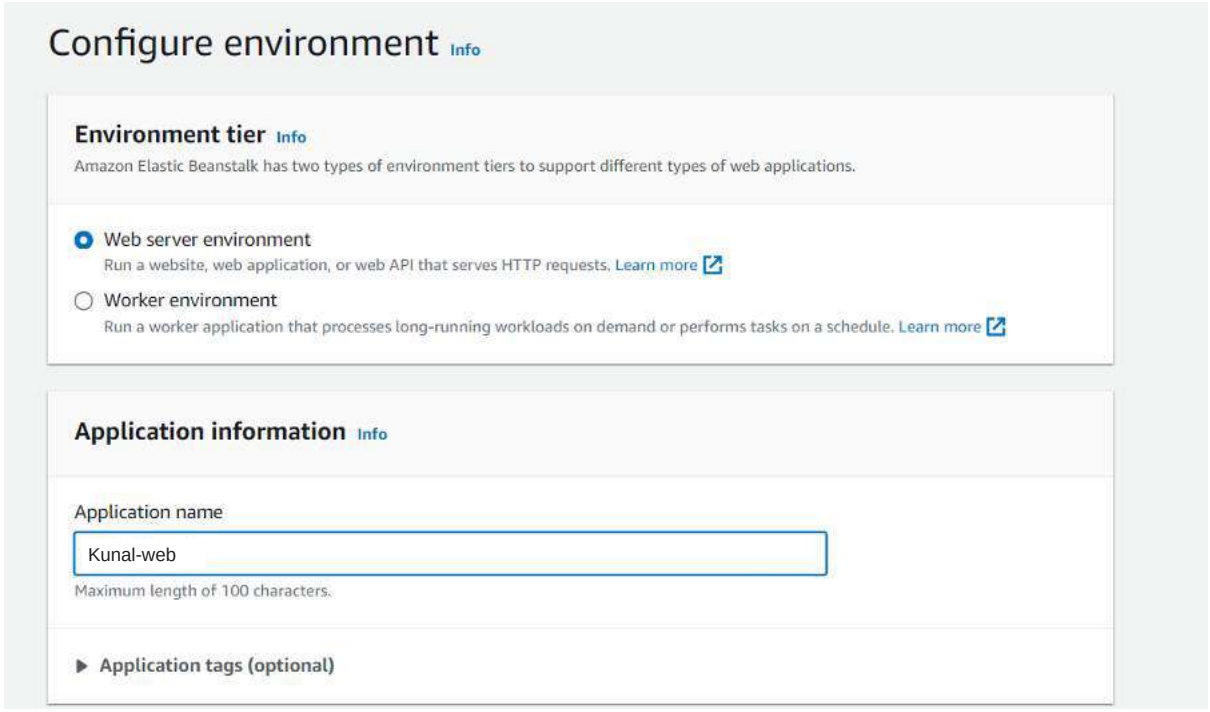
```
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-32-173:~# cd /var/www/html
```


Not secure
54.86.249.242
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled


Not secure
54.86.249.242/index2.html
Hi..Kunal here

Elastic beanstalk





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

Node.js

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023

Platform version

6.2.0 (Recommended)

Review Info

Step 1: Configure environment

Edit

Environment information

Environment tier Web server environment	Application name Kunal-web
Environment name Kunal-web-env	Application code Sample application
Platform arn:aws:elasticbeanstalk:eu-north-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

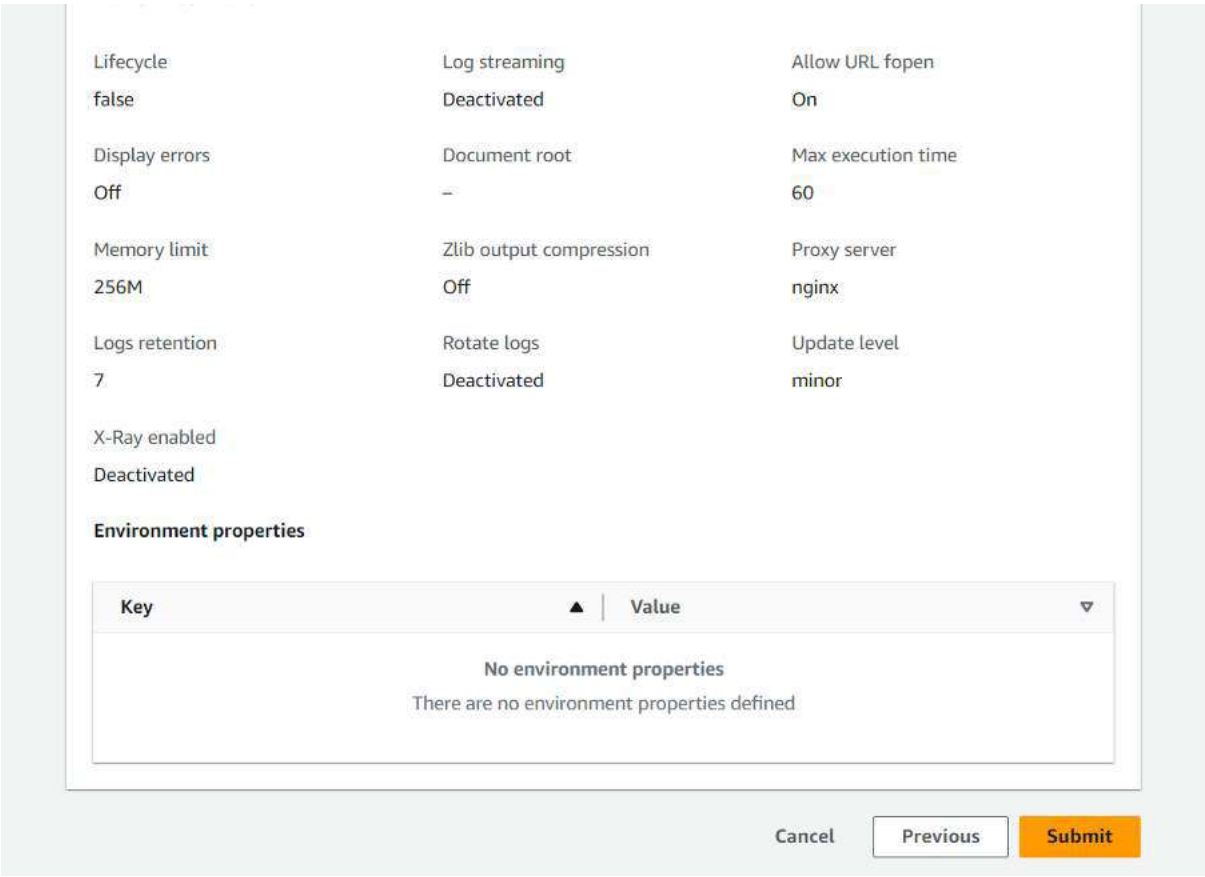
Step 2: Configure service access

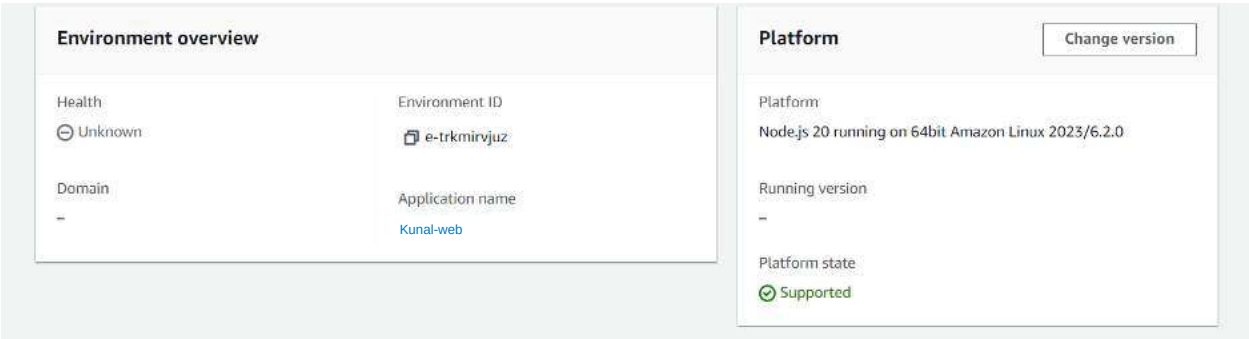
Edit

Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile


Lifecycle
false
Log streaming
Deactivated
Allow URL fopen
On
Display errors
Off
Document root
–
Max execution time
60
Memory limit
256M
Zlib output compression
Off
Proxy server
nginx
Logs retention
7
Rotate logs
Deactivated
Update level
minor
X-Ray enabled
Deactivated
Environment properties
Key
Value
No environment properties
There are no environment properties defined
Cancel
Previous
Submit


Environment overview
Health
Unknown
Environment ID
e-trkmirvjuz
Domain
–
Application name
Kunal-web
Platform
Change version
Platform
Node.js 20 running on 64bit Amazon Linux 2023/6.2.0
Running version
–
Platform state
Supported

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploying an Express Application to AWS Elastic Beanstalk
- Deploying an Express application with clustering to Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

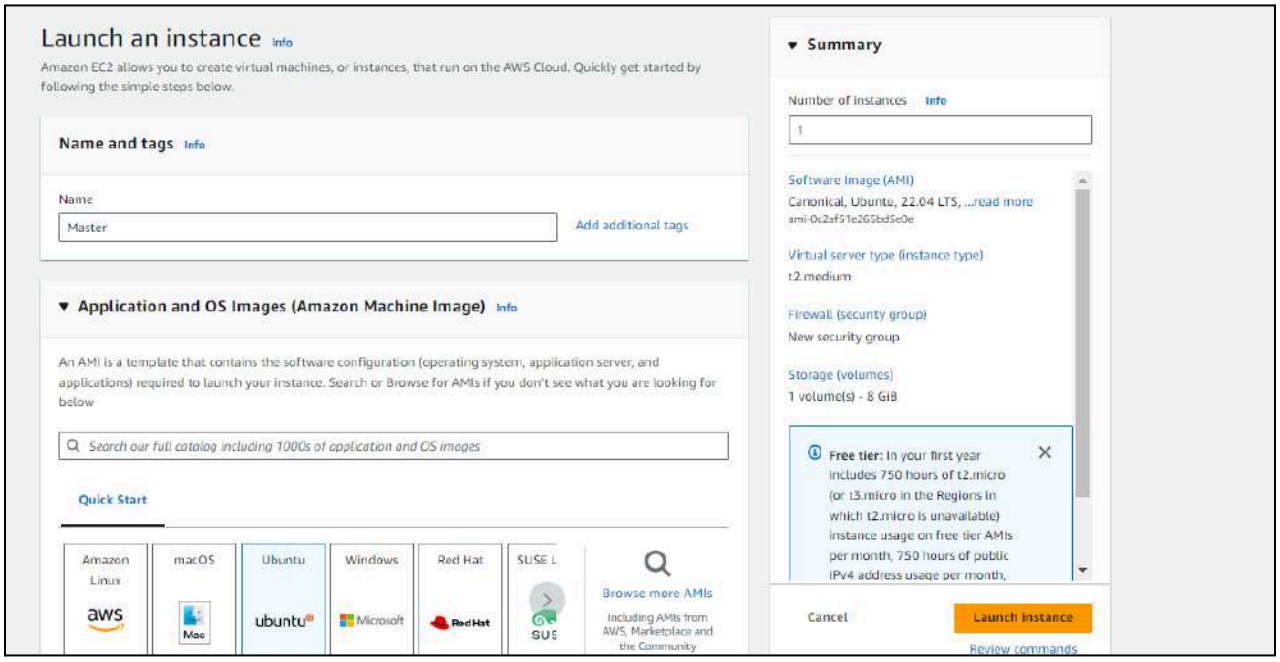
ADVANCE DEVOPS
EXPERIMENT - 3

Name:Kunal Punjabi
Class:D15A
Roll No:44

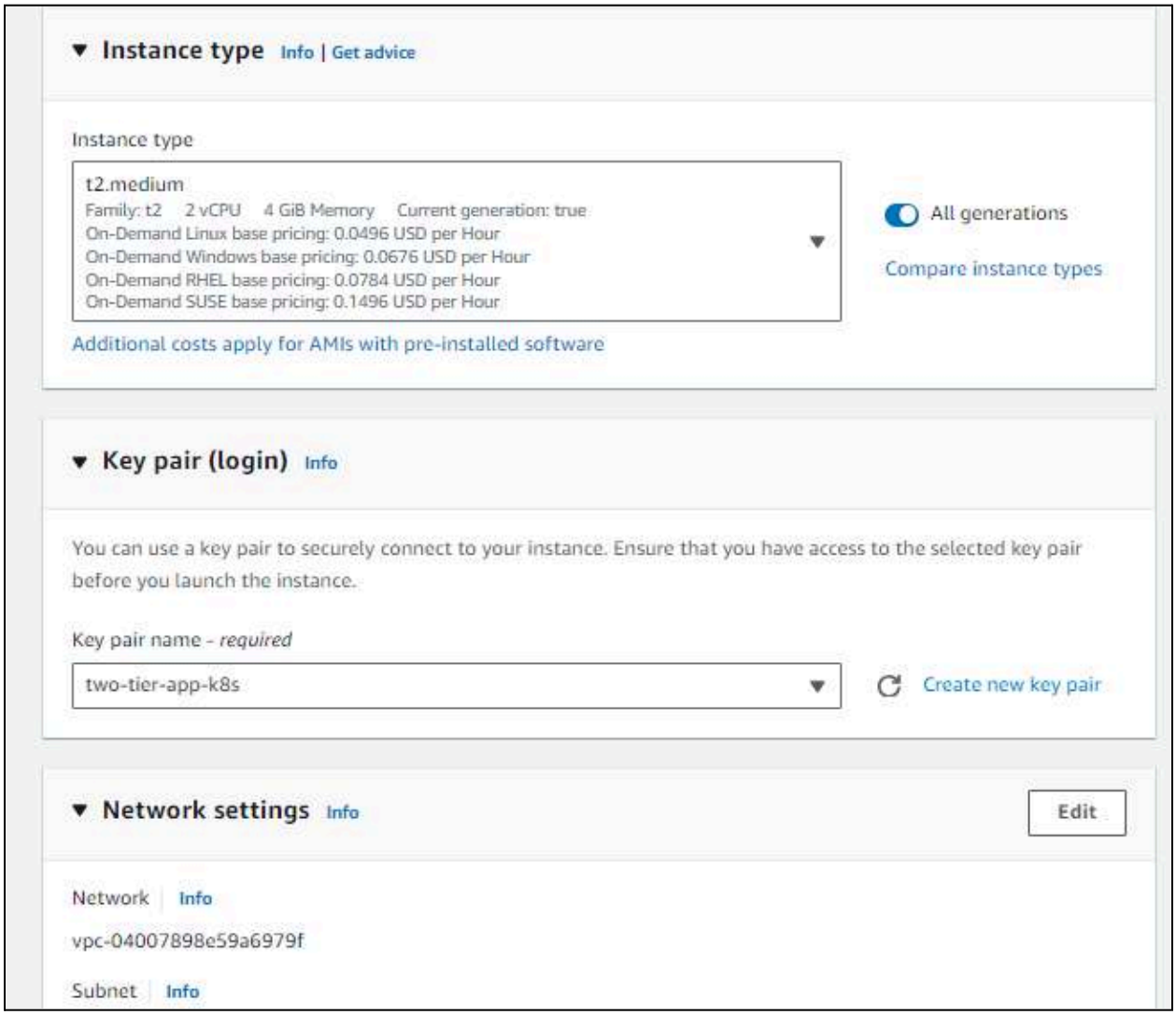
Aim:To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Step 1: Pre-requisites

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)



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

two-tier-app-k8s

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

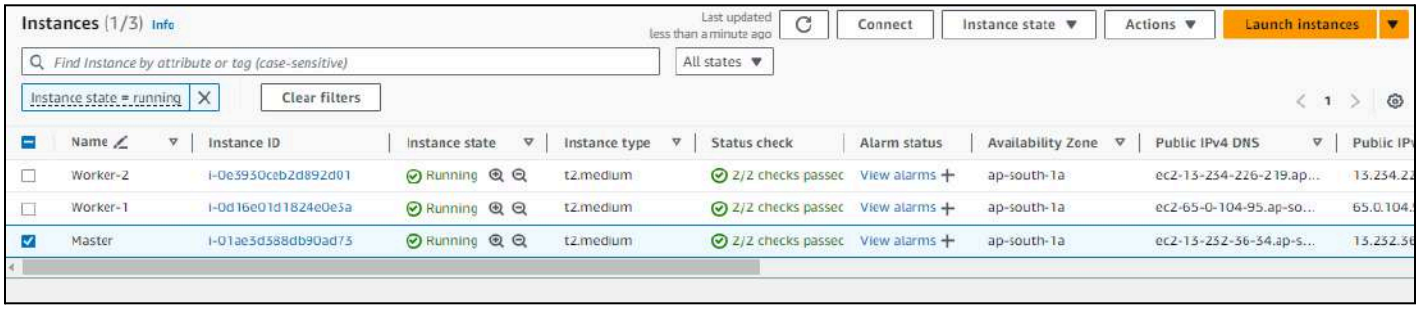
- RSA
  RSA encrypted private and public key pair
- ED25519
  ED25519 encrypted private and public key pair

Private key file format

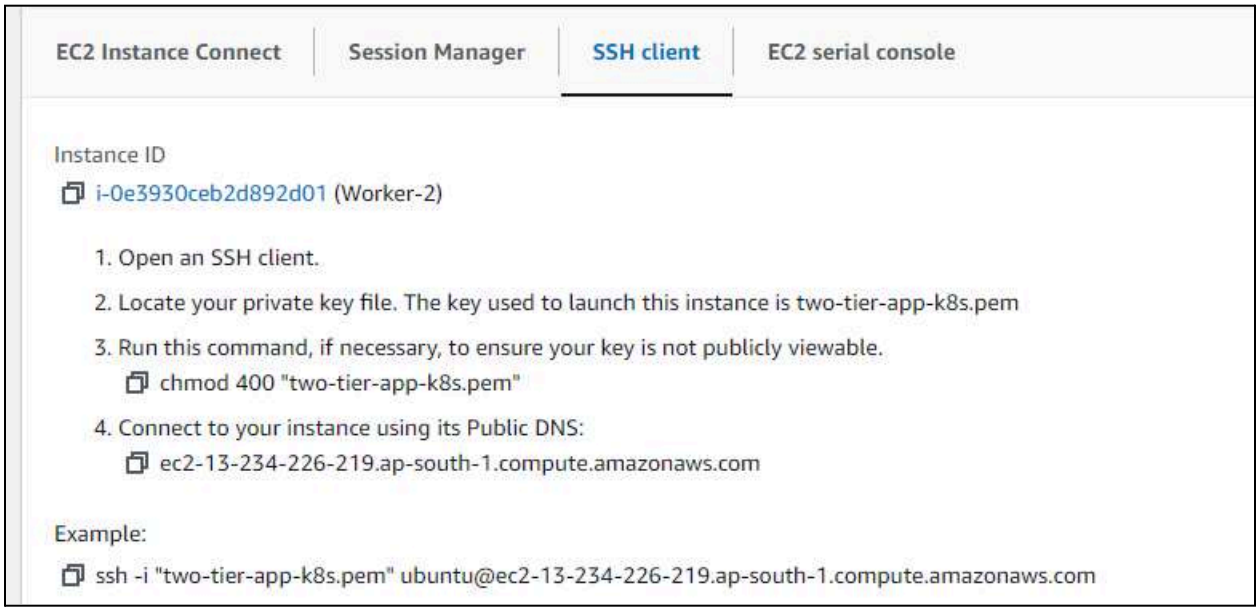
- .pem
  For use with OpenSSH
- .ppk
  For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on

Cancel | Create key pair



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

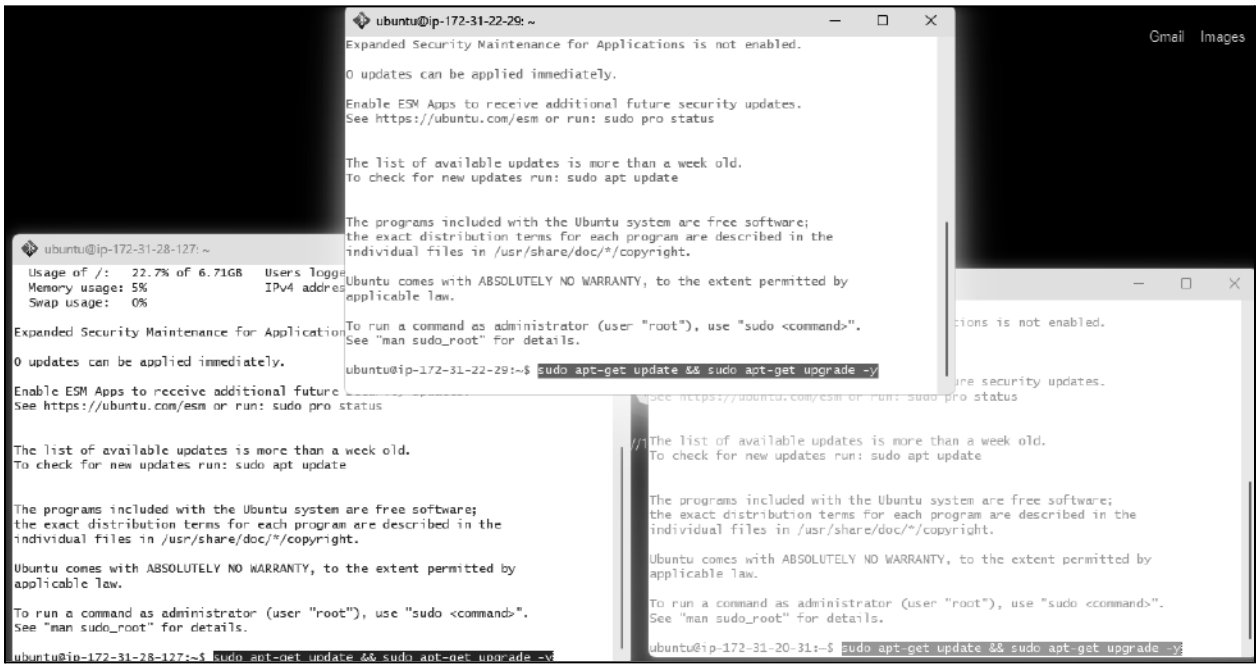



```
acer@TMP214-53 MINGW64 ~/Downloads
$ ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-232-36-34.ap-south-1.compute.amazo
naws.com
The authenticity of host 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com (13.
232.36.34)' can't be established.
ED25519 key fingerprint is SHA256:uVGEO+FWYefj60j0ft70Sralv8NrzEi/IwxAtBY+EPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com' (
ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

 System information as of Wed Sep 11 14:07:10 UTC 2024

  System load:  0.0               Processes:             106
  Usage of /:   20.7% of 7.57GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for eth0: 172.31.45.227
  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
```

Step 2:Prepare Nodes

2.1. Update the package manager on all nodes:

sudo apt-get update && sudo apt-get upgrade -y




```
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packag
es [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-
en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f
Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Pack
ages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translatio
n-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n
-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 P
ackages [2023 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Transla
tion-en [352 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c
-n-f Metadata [17.8 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted a
md64 Packages [2437 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted T
ranslation-en [419 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted a
```

2.2. Disable Swap (Kubernetes requires swap to be off):
sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab

```
ubuntu@ip-172-31-22-29:~$ sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

2.3. Load necessary kernel modules for networking and iptables:
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
```

2.4. Configure sysctl settings for Kubernetes networking:
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
```

Step 3: Install Docker

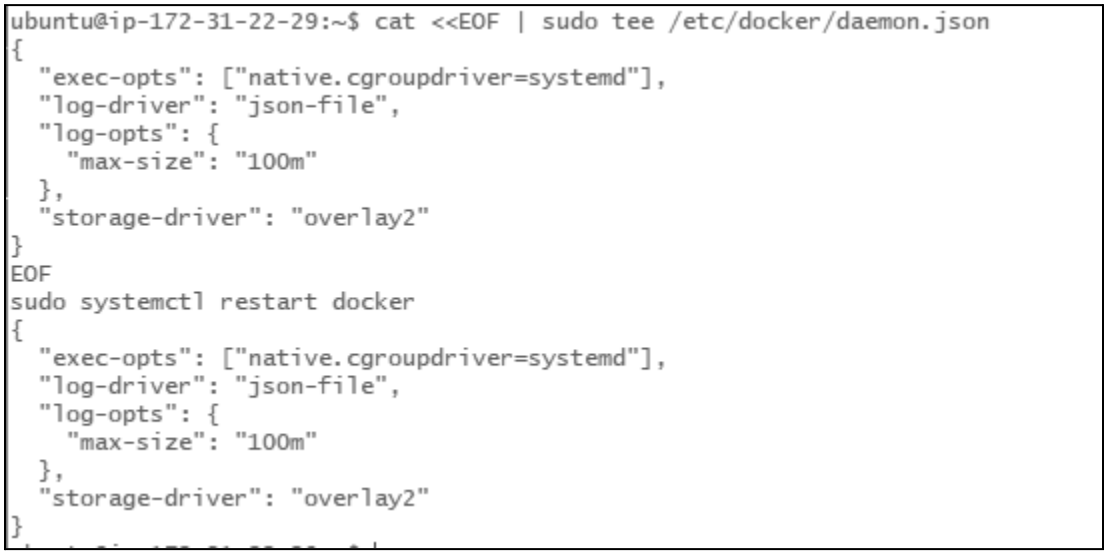
Kubernetes uses container runtimes like Docker. Install Docker on all nodes.

sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-proper
ties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubun
tu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Fetched 129 kB in 1s (241 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.17).
curl set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
```

Configure Docker for Kubernetes:

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
 "exec-opts": ["native.cgroupdriver=systemd"],
 "log-driver": "json-file",
 "log-opts": {
  "max-size": "100m"
 },
 "storage-driver": "overlay2"
}
EOF
sudo systemctl restart docker
```



Step 4: Install kubeadm, kubelet, kubectl

Install Kubernetes tools on all nodes.

4.1. Add Kubernetes APT repository:

```
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-22-29:~$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archiv
e-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https:/
/apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/k
ubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.k
ubernetes.io/ kubernetes-xenial main
```

4.2. Install kubeadm, kubelet, and kubectl:

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
```

Step 5: Initialize the Kubernetes Cluster on Master Node

On the master node:

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

```
ubuntu@ip-172-31-22-29:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --v
=5
Found multiple CRI endpoints on the host. Please define which one do you wish to
 use by setting the 'criSocket' field in the kubeadm configuration file: unix://
/var/run/containerd/containerd.sock, unix:///var/run/crio/crio.sock
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.detectCRISocketImpl
        cmd/kubeadm/app/util/runtime/runtime.go:167
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.DetectCRISocket
        cmd/kubeadm/app/util/runtime/runtime.go:175
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetNodeRegistrationDynamicDefaults
        cmd/kubeadm/app/util/config/initconfiguration.go:118
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetInitDynamicDefaults
        cmd/kubeadm/app/util/config/initconfiguration.go:64
k8s.io/kubernetes/cmd/kubeadm/app/util/config.DefaultedInitConfiguration
        cmd/kubeadm/app/util/config/initconfiguration.go:248
k8s.io/kubernetes/cmd/kubeadm/app/util/config.LoadOrDefaultInitConfiguration
        cmd/kubeadm/app/util/config/initconfiguration.go:282
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newInitData
        cmd/kubeadm/app/cmd/init.go:319
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func3
        cmd/kubeadm/app/cmd/init.go:170
k8s.io/kubernetes/cmd/kubeadm/app/cmd/phases/workflow.(*Runner).InitData
        cmd/kubeadm/app/cmd/phases/workflow/runner.go:183
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func1
```

5.1. Set up kubectl on the master node:
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
ubuntu@ip-172-31-22-29:~$ 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 --v=5
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in the kubeadm configuration file: unix:///var/run/containerd/containerd
ock, unix:///var/run/crio/crio.sock
To see the stack trace of this error execute with --v=5 or higher
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in the kubeadm configuration file: unix:///var/run/containerd/containerd
ock, unix:///var/run/crio/crio.sock
```

Step 6: Install a Pod Network Add-on

To enable communication between pods, install a pod network plugin like Flannel or Calico.

Install Flannel:

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

```
ubuntu@ip-172-31-22-29:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml --validate=false

E0913 15:35:04.261458   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
E0913 15:35:04.261902   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
E0913 15:35:04.263424   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
E0913 15:35:04.263795   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
E0913 15:35:04.265840   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
E0913 15:35:04.266524   19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:808
unable to recognize "https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml": Get "http://localhost:8080/api?timeout=32s": dia
```

Step 7: Join Worker Nodes to the Cluster

On the **worker nodes**, run the command provided by the master node during initialization . It looks something like this:

sudo kubeadm join <master-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>

```
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
kubeadm join 172.31.62.216:6443 --token br7fe5.hq28adbmn1mu17ky --discovery-token-ca-cert-hash sha256:2bc469a8d14fbeb0f879328d2b416fad
32b29a8505d3f448b98703fff3b014d9
```

Step 8: Verify the Cluster

Once the worker node joins, check the status on the **master node**

```
ubuntu@ip-172-31-45-227:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-43-211   Ready    <none>          50s     v1.29.0
ip-172-31-45-13    Ready    <none>          34s     v1.29.0
ip-172-31-45-227   Ready    control-plane   5m17s   v1.29.0
ubuntu@ip-172-31-45-227:~$
```

ADVANCE DEVOPS
EXPERIMENT - 4

Name: Kunal Punjabi
Class:D15A
Roll No:44

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

Step 1: Install Kubectl on Ubuntu

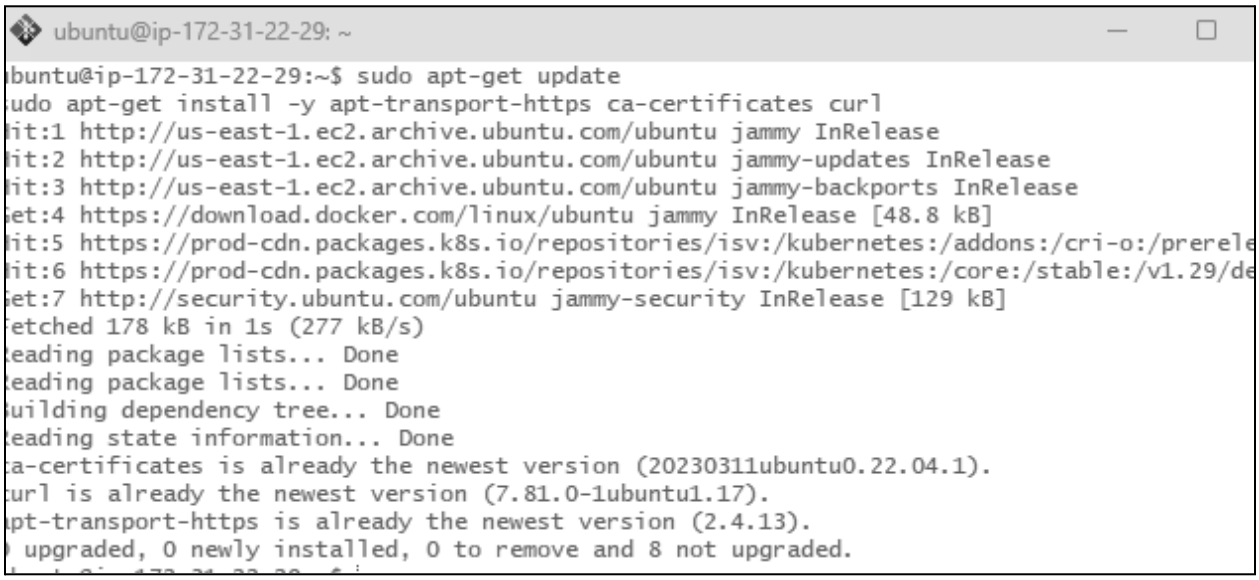
1.1 Add Kubernetes APT repository

First, add the Kubernetes repository to your system.

1. **Install prerequisites:**

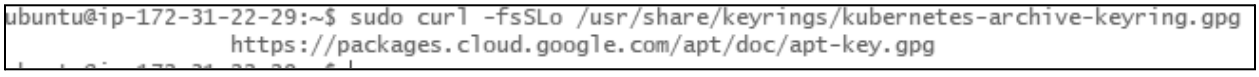
   sudo apt-get update

   sudo apt-get install -y apt-transport-https ca-certificates curl



2. **Add the GPG key for Kubernetes:**

   sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg



3. **Add the Kubernetes repository:**

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-22-29:~$  echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring
.gpg]  https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/ku
bernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]  https://apt.kubernetes.io/
 kubernetes-focal main
```

1.2 Install kubectl

Now install kubectl:

sudo apt-get update

sudo apt-get install -y kubectl

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb  InRelease
Ign:7 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.62.138 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

Verify the installation(extra):

kubectl version --client

```
ubuntu@ip-172-31-22-29:~$ kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 2: Deploying Your Application on Kubernetes

2.1 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

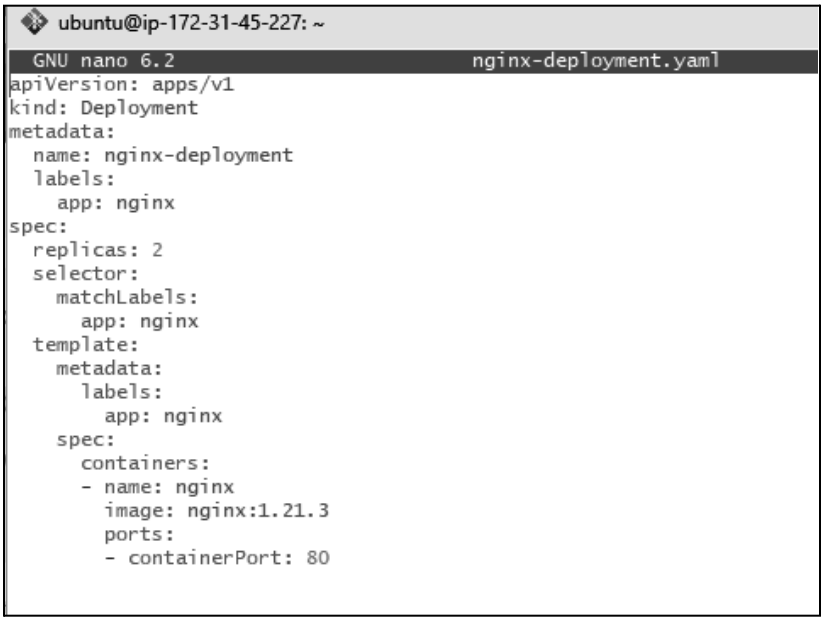
```
ubuntu@ip-172-31-45-227:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE     VERSION
ip-172-31-43-211    Ready    <none>          50s     v1.29.0
ip-172-31-45-13     Ready    <none>          34s     v1.29.0
ip-172-31-45-227    Ready    control-plane   5m17s   v1.29.0
ubuntu@ip-172-31-45-227:~$
```

Step 3:**Create the Deployment YAML file**

a)Create the YAML file: Use a text editor to create a file named nginx-deployment.yaml

```
ubuntu@ip-172-31-45-227:~$ nano nginx-deployment.yaml
```

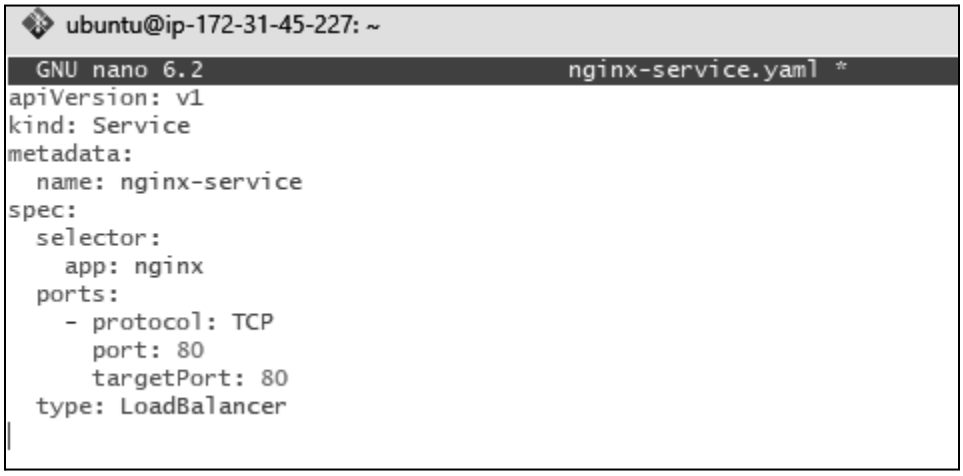
b)Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).



Step 4:Create the Service YAML File

a)Create the YAML File: Create another file named nginx-service.yaml

b)Add the Service Configuration: Copy and paste the following YAML content into the file given below.



Step 5:Apply the YAML Files

a)Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files.

```
ubuntu@ip-172-31-45-227:~$ kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-service.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
```

b)Verify the Deployment: Check the status of your Deployment,Pods and Services.

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
kubectl get pods
kubectl get services
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           40s
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-6k84m   1/1     Running   0          40s
nginx-deployment-6b4d6fdbf-9d8j6   1/1     Running   0          40s
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1        <none>        443/TCP        40m
nginx-service   LoadBalancer   10.106.182.152   <pending>     80:32317/TCP   40s
```

Describe the deployment(Extra)

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           14h
ubuntu@ip-172-31-45-227:~$ kubectl describe deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Wed, 11 Sep 2024 17:16:17 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:latest
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from website-volume (rw)
  Volumes:
   website-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      nginx-website
    Optional:  false
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  nginx-deployment-6b4d6fdbf (0/0 replicas created)
NewReplicaSet:   nginx-deployment-776b8fd845 (1/1 replicas created)
Events:          <none>
```

Step 6:Ensure Service is Running

6.1 **Verify Service**: Run the following command to check the services running in your cluster:

kubectl get service

```
ubuntu@ip-172-31-45-227:~$ kubectl get service
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP        16h
nginx           NodePort    10.106.0.176     <none>        80:32618/TCP   76m
nginx-service   NodePort    10.106.182.152   <none>        80:30007/TCP   15h
nginx2          NodePort    10.99.32.156     <none>        80:31421/TCP   8s
```

Step 7: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/<service-name> <local-port>:<service-port>

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

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-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Cubuntu@ip-172-31-45-227:~kubectl port-forward service/nginx-service 8081:8080
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-776b8fd845-k9cx4   1/1     Running   0          113m
ubuntu@ip-172-31-45-227:~$ kubectl logs nginx-deployment-776b8fd845-k9cx4
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/12 06:35:51 [notice] 1#1: using the "epoll" event method
2024/09/12 06:35:51 [notice] 1#1: nginx/1.27.1
2024/09/12 06:35:51 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/12 06:35:51 [notice] 1#1: OS: Linux 6.5.0-1022-aws
2024/09/12 06:35:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/12 06:35:51 [notice] 1#1: start worker processes
2024/09/12 06:35:51 [notice] 1#1: start worker process 24
2024/09/12 06:35:51 [notice] 1#1: start worker process 25
```

Step 8: Access the Application Locally

1. **Open a Web Browser**: Now open your web browser and go to the following URL:

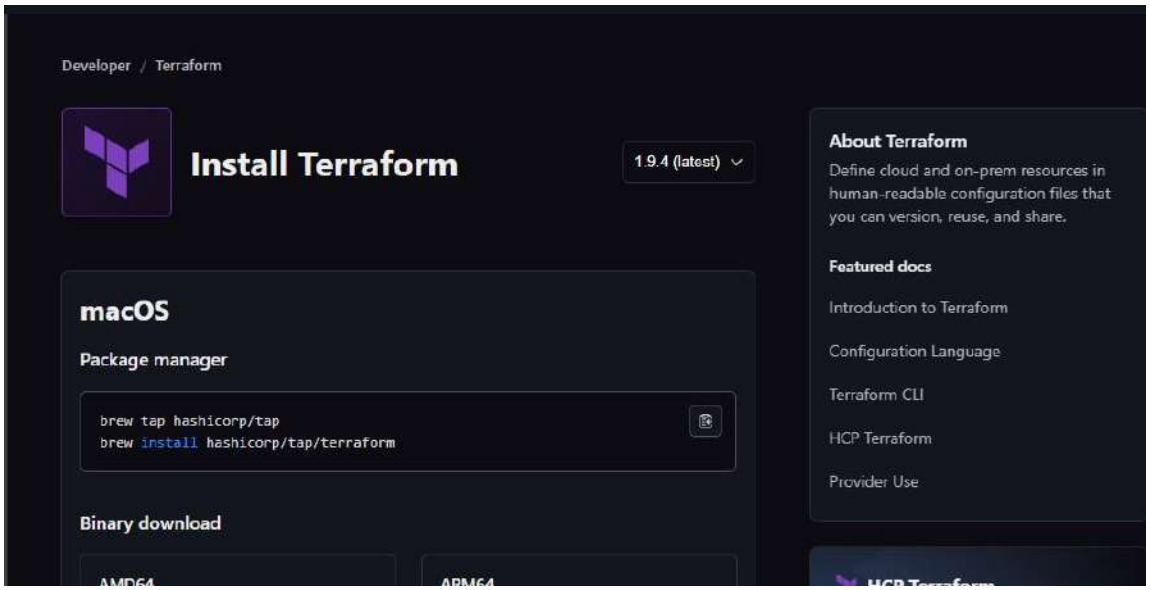
   http://localhost:8080

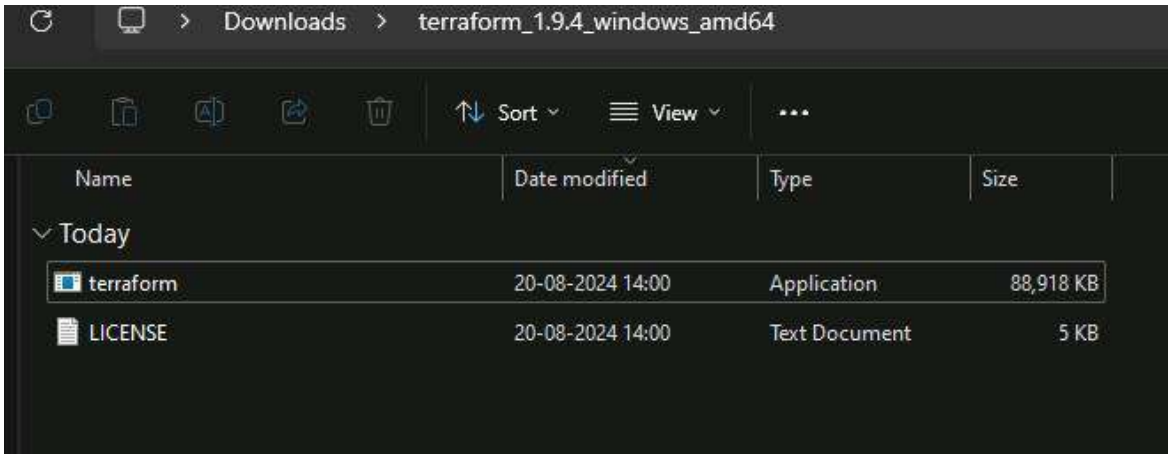
You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080.

In case the port 8080 is unavailable, try using a different port like 8081

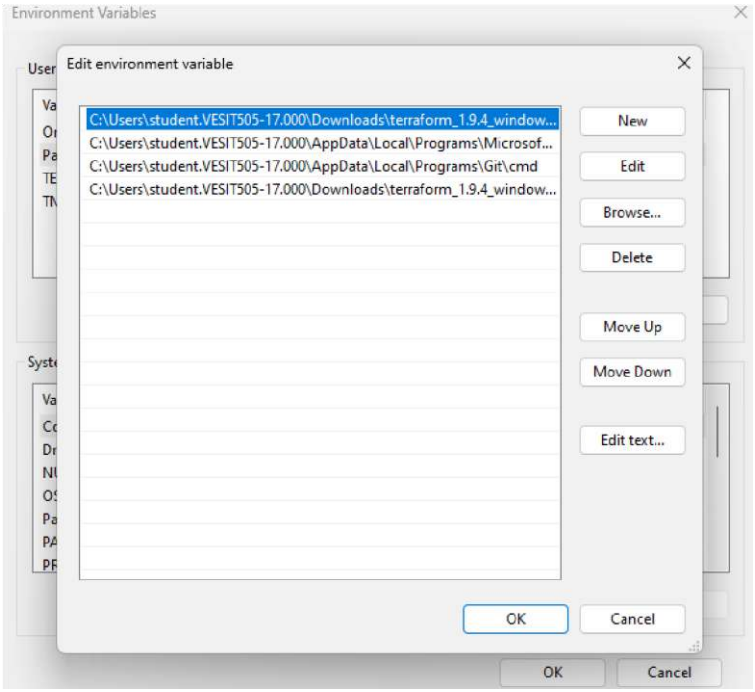
AdvDevops Experiment 5

AIM: Installation and configuration of terraform on Windows





Kunal Punjabi D15A ‘44’




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

PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

```
PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64> terraform --version
Terraform v1.9.4
on windows_amd64
PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64>
```

Kunal Punjabi D15A 44

ADVANCE DEVOPS EXP 6

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

Part A: Creating docker image using terraform

Step 1: Check Docker functionality

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

C:\Users\student>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

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

Check for the docker version with the following command.

```
C:\Users\student>docker --version
Docker version 27.1.1, build 6312585

C:\Users\student>
```

Create a folder named ‘Terraform Scripts’ in which we save our different typesof scripts which will be further used in this experiment.

Step 2:

Creating a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.
Creating a new docker.tf file using Atom editor and write the following contents into.

This will create a Ubuntu Linux container

docker.tf

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

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

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

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

Step 3: Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform plan

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "sleep",
          + "3600",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + 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.
```

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”**

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform apply

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "sleep",
          + "3600",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
```

```
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

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

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

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

  Enter a value: yes
```

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 9s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Before Executing Apply step:

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

After Executing Apply step:

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

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Terraform will perform the following actions:

  # docker_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach           = false -> null
      - command          = [
          - "sleep",
          - "3600",
        ] -> null
      - cpu_shares       = 0 -> null
      - dns              = [] -> null
      - dns_opts         = [] -> null
      - dns_search       = [] -> null
      - entrypoint       = [] -> null
      - env              = [] -> null
      - gateway          = "172.17.0.1" -> null
      - group_add        = [] -> null
      - hostname         = "01adf07e5918" -> null
      - id               = "01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24" -> null
      - image            = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - init             = false -> null
      - ip_address       = "172.17.0.2" -> null
      - ip_prefix_length = 16 -> null
      - ipc_mode         = "private" -> null
      - links            = [] -> null
      - log_driver       = "json-file" -> null
      - log_opts         = {} -> null
      - logs             = false -> null
      - max_retry_count  = 0 -> null
```

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

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

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

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

  Enter a value: yes

docker_container.foo: Destroying... [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

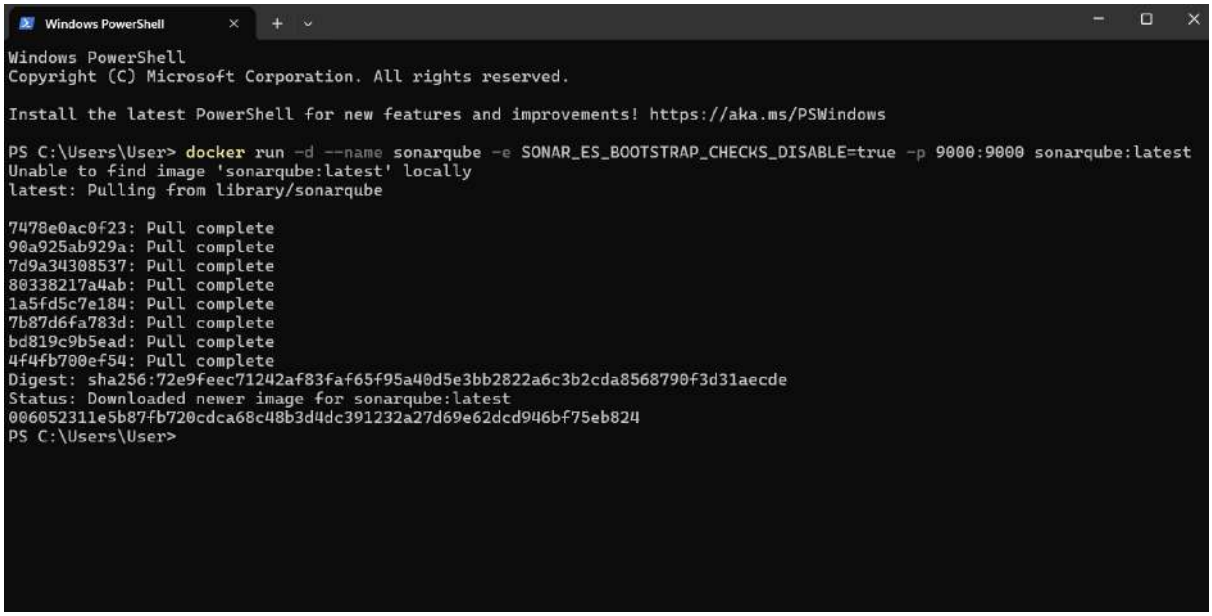
Docker images After Executing Destroy step

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

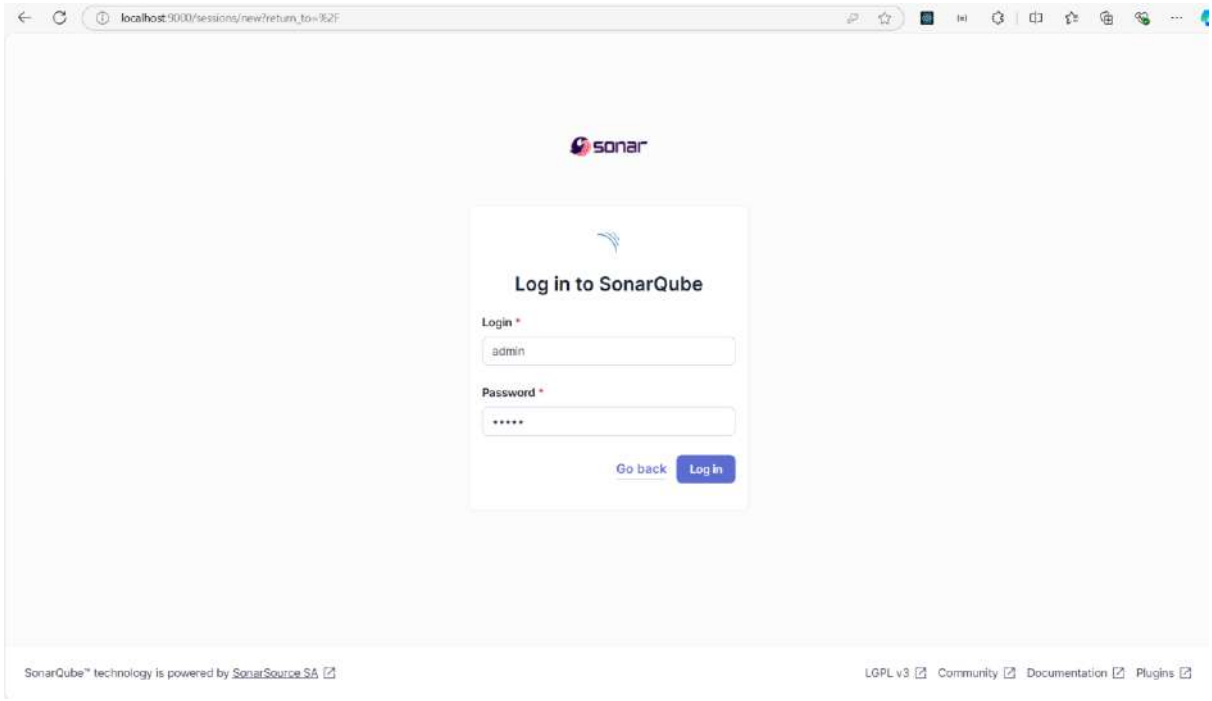
EXPERIMENT 7

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

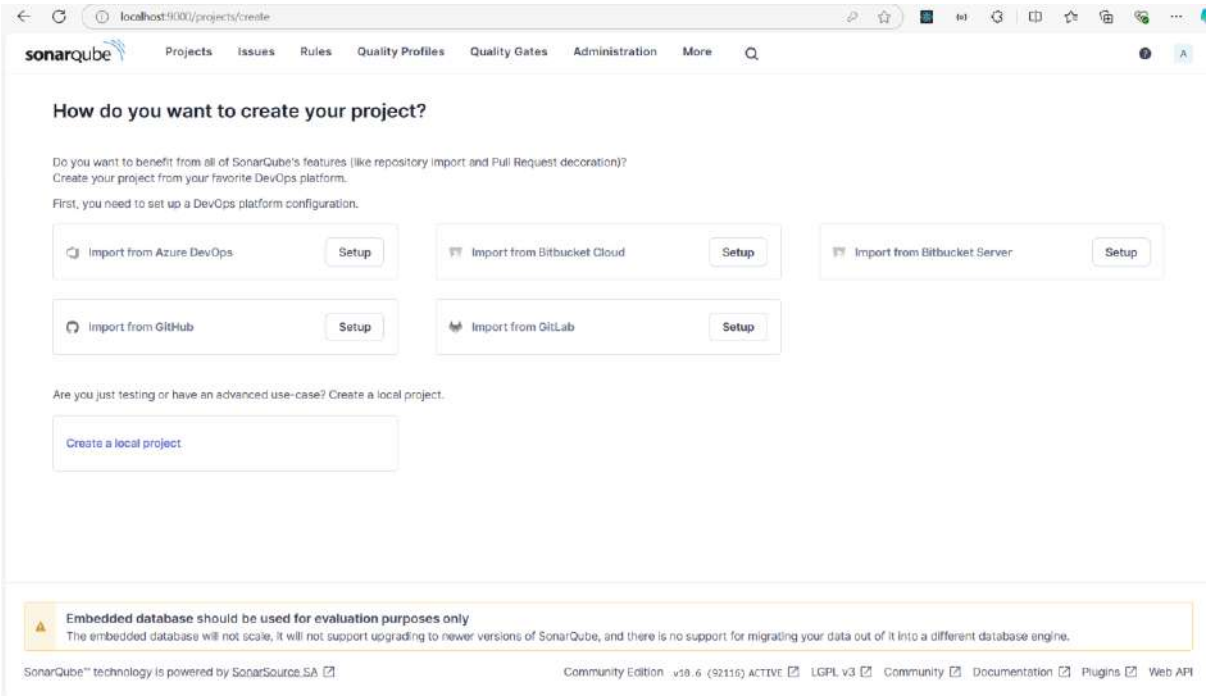
Step 1: Open Windows PowerShell and run the following command –
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
WARNING: Run the following command only once

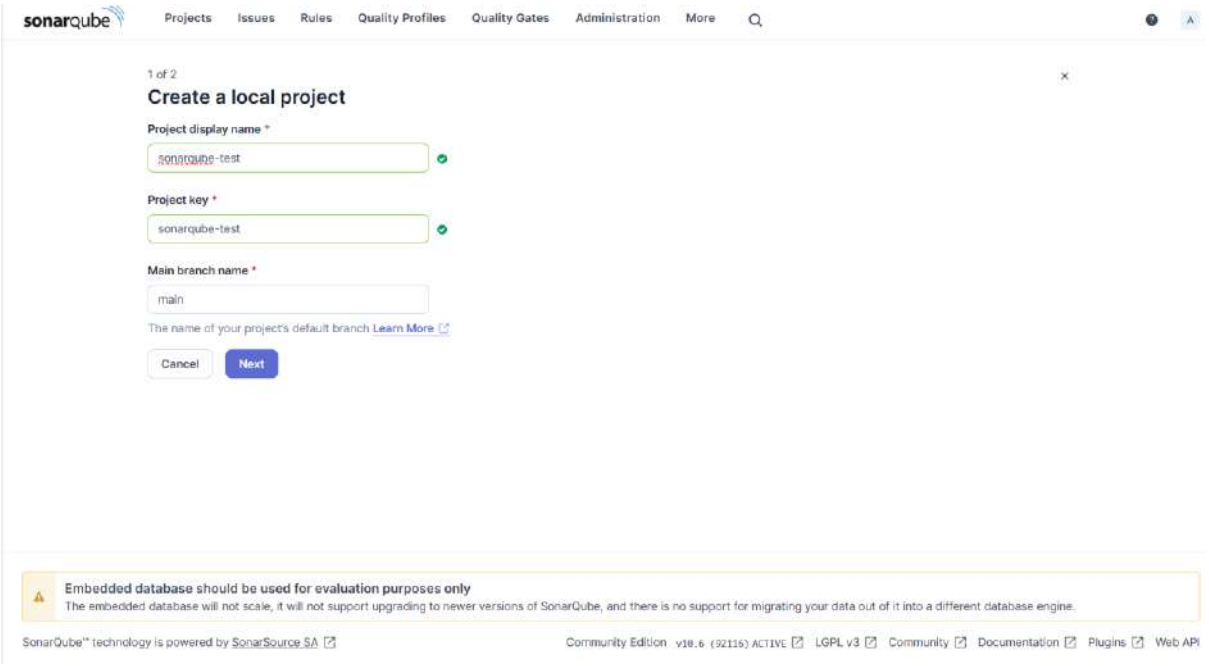


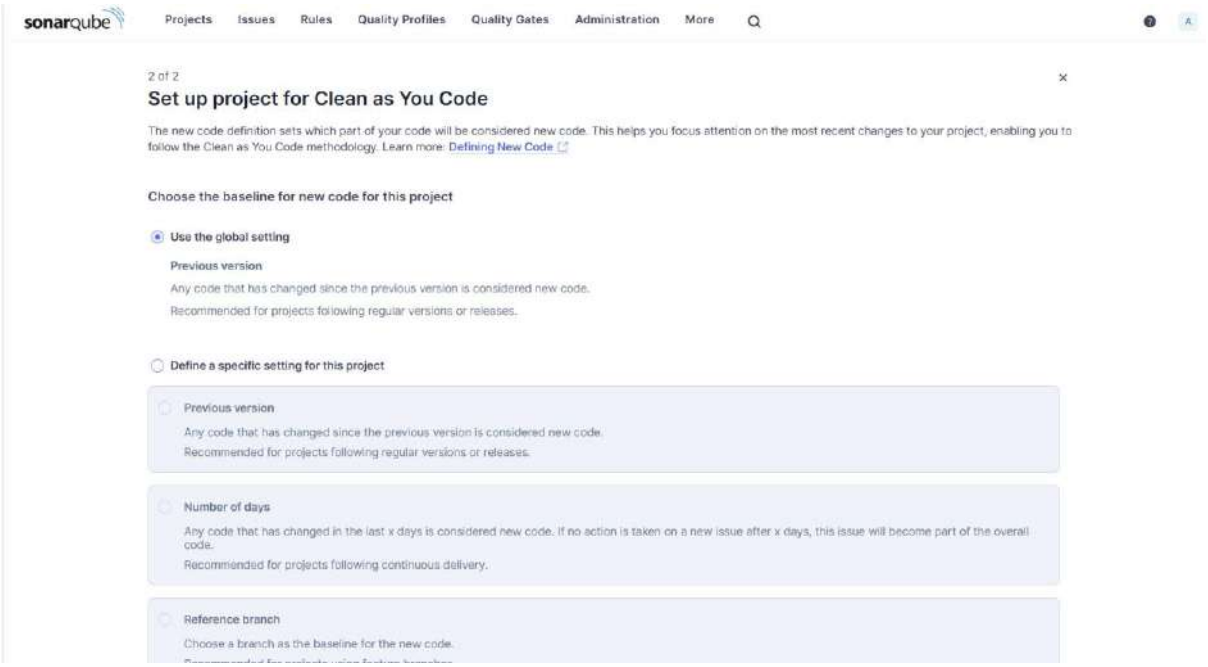
Step 2: Visit http://localhost:9000/ to open SonarQube. Login with username: admin and password: admin.



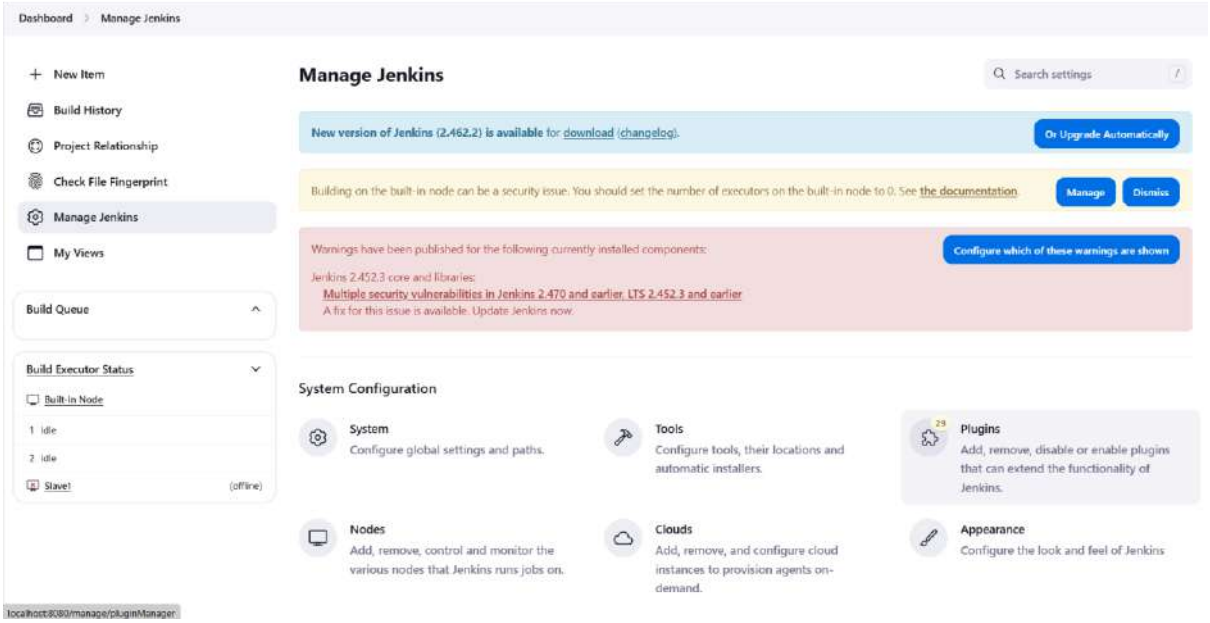
Step 3: Click on create a local project and name the project as sonarqube-test and key as sonarqube-test and click on the next button. In the next step select the “Use the global setting” option and click on create project.

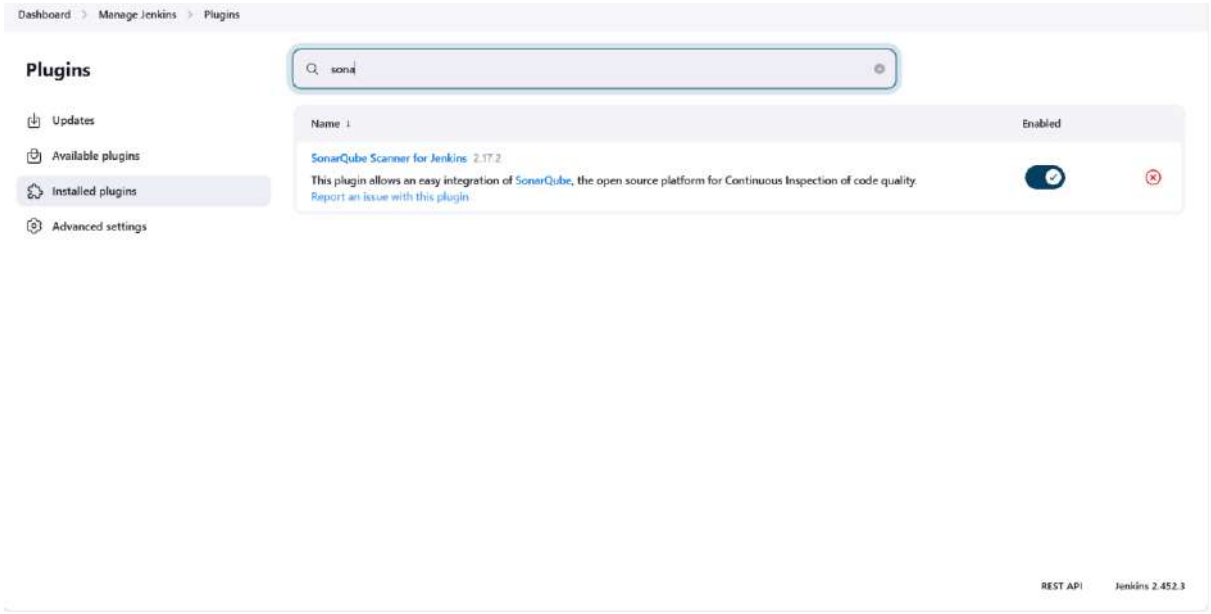

localhost:9000/projects/create
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
How do you want to create your project?
Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.
First, you need to set up a DevOps platform configuration.
Import from Azure DevOps
Setup
Import from Bitbucket Cloud
Setup
Import from Bitbucket Server
Setup
Import from GitHub
Setup
Import from GitLab
Setup
Are you just testing or have an advanced use-case? Create a local project.
Create a local project
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.
SonarQube™ technology is powered by SonarSource SA
Community Edition v10.6 (92116) ACTIVE
LGPL v3
Community
Documentation
Plugins
Web API


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
1 of 2
Create a local project
Project display name *
sonarqube-test
Project key *
sonarqube-test
Main branch name *
main
The name of your project's default branch Learn More
Cancel
Next
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.
SonarQube™ technology is powered by SonarSource SA
Community Edition v10.6 (92116) ACTIVE
LGPL v3
Community
Documentation
Plugins
Web API

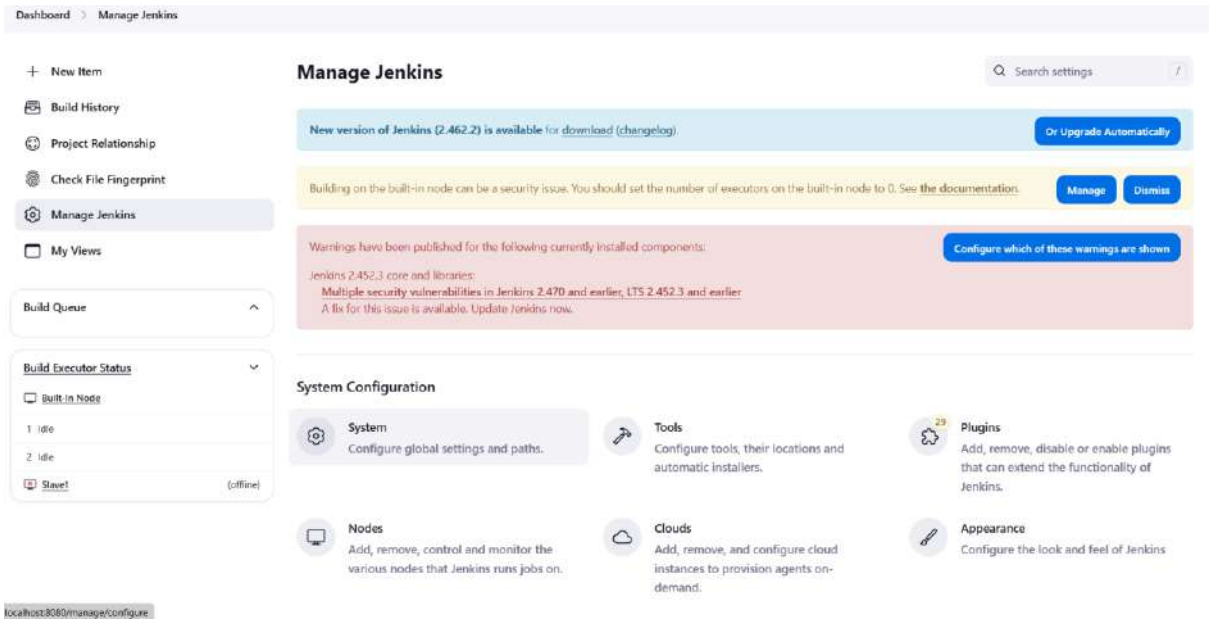


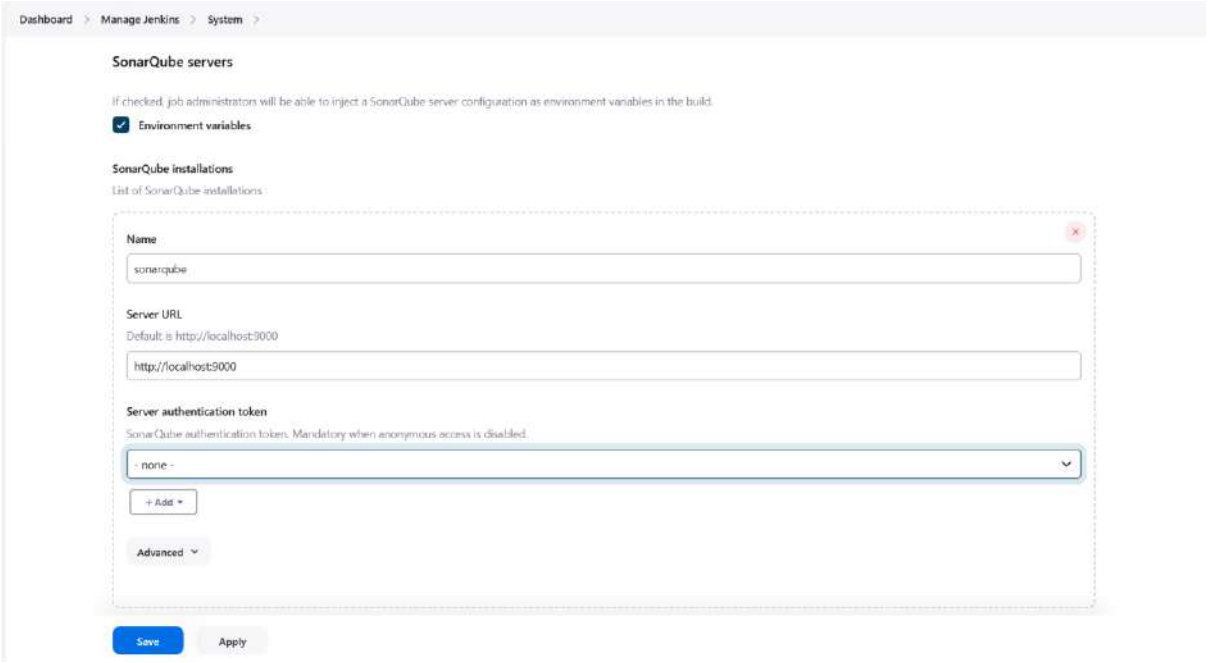
Step 4: Open Jenkins using http://localhost:8080/ and select Manage Jenkins, then select the Plugins and select available plugins from sidebar and search for SonarQube Scanner and install it. Once installed you can view the installed plugin in installed plugins section in sidebar.



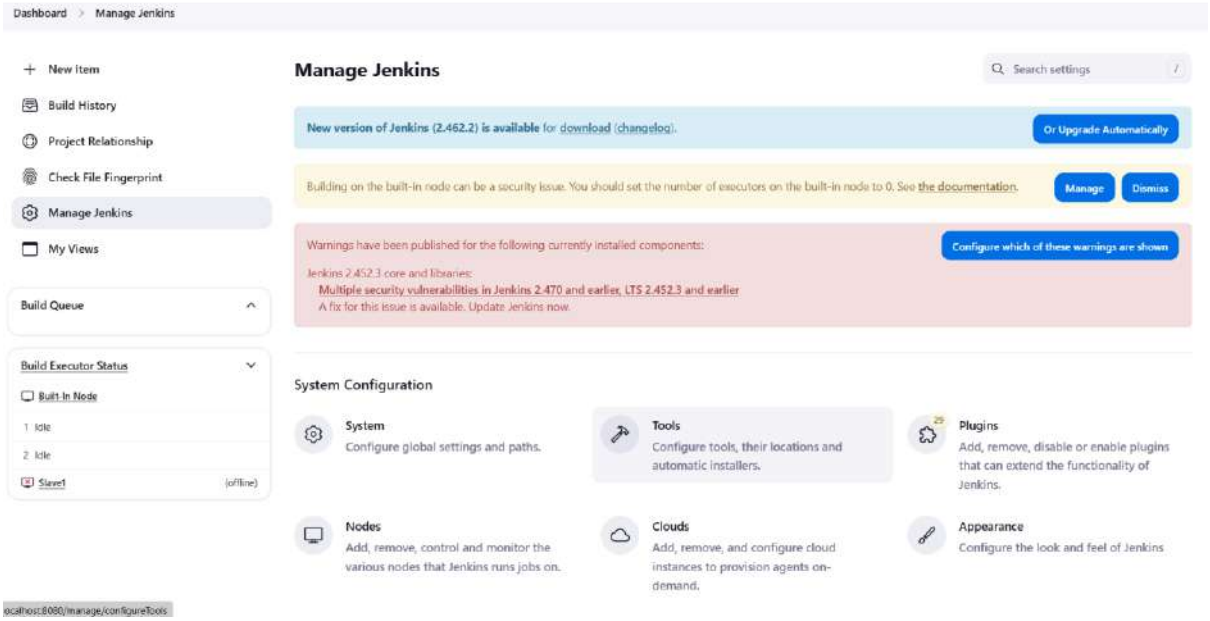


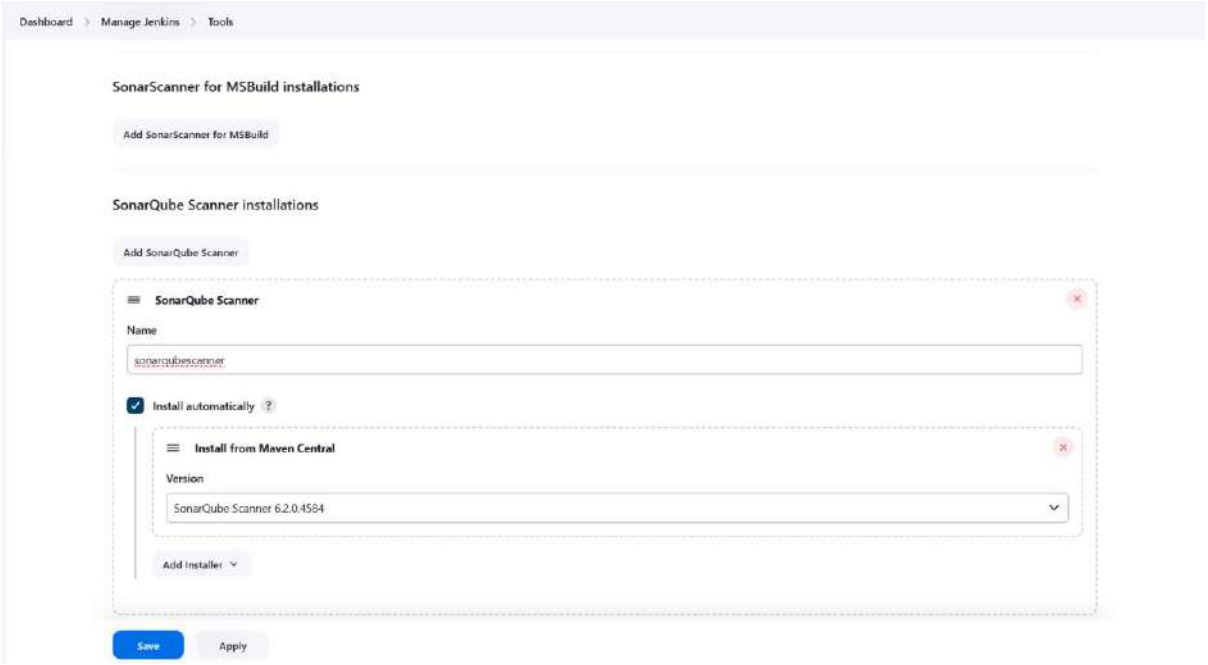
Step 5: Select Manage Jenkins, then select the System and then scroll down to SonarQube Server. Name the server as sonarqube and set the server url as http://localhost:9000/ then click on save.



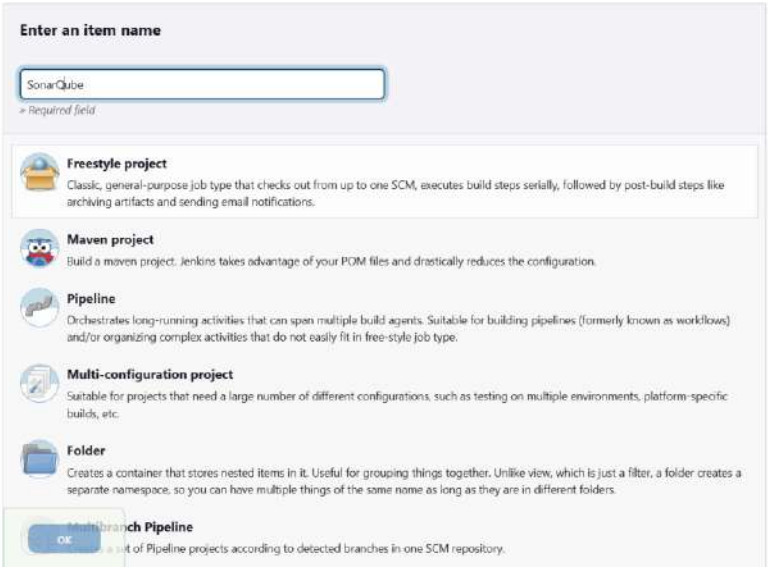


Step 6: Go to Jenkins Dashboard and select Manage Jenkins, then select the Tools and then scroll down to SonarQube Scanner installations. Name the sonarqube scanner as sonarqubescanner and select install automatically then click on save.

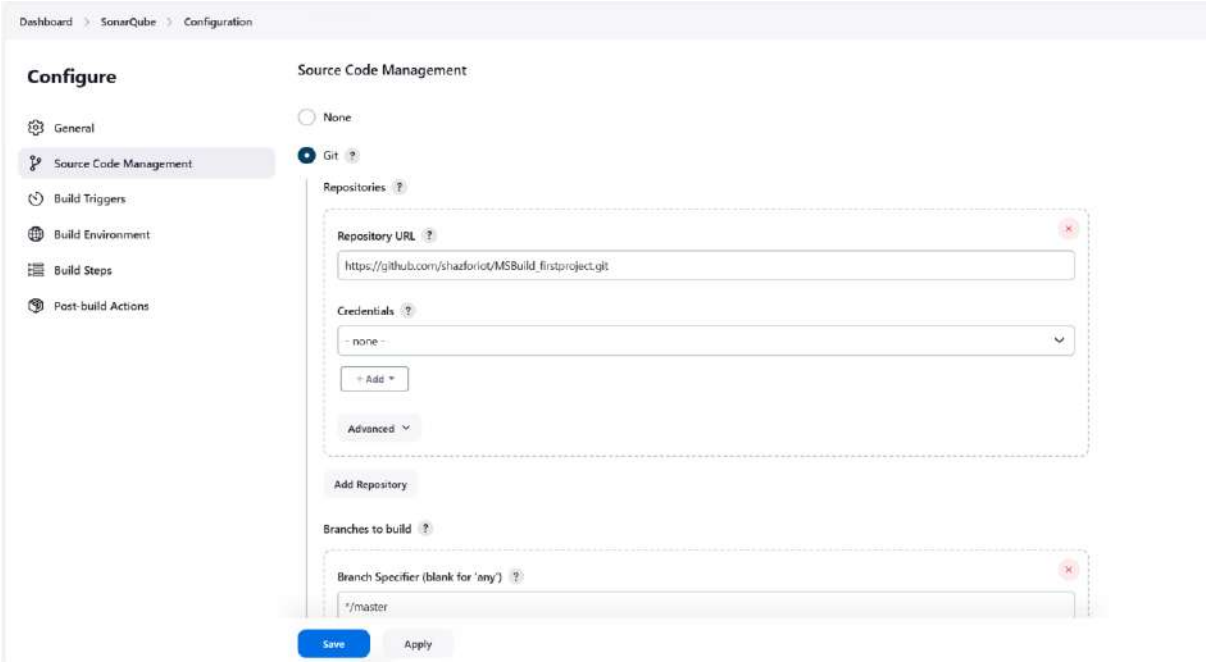




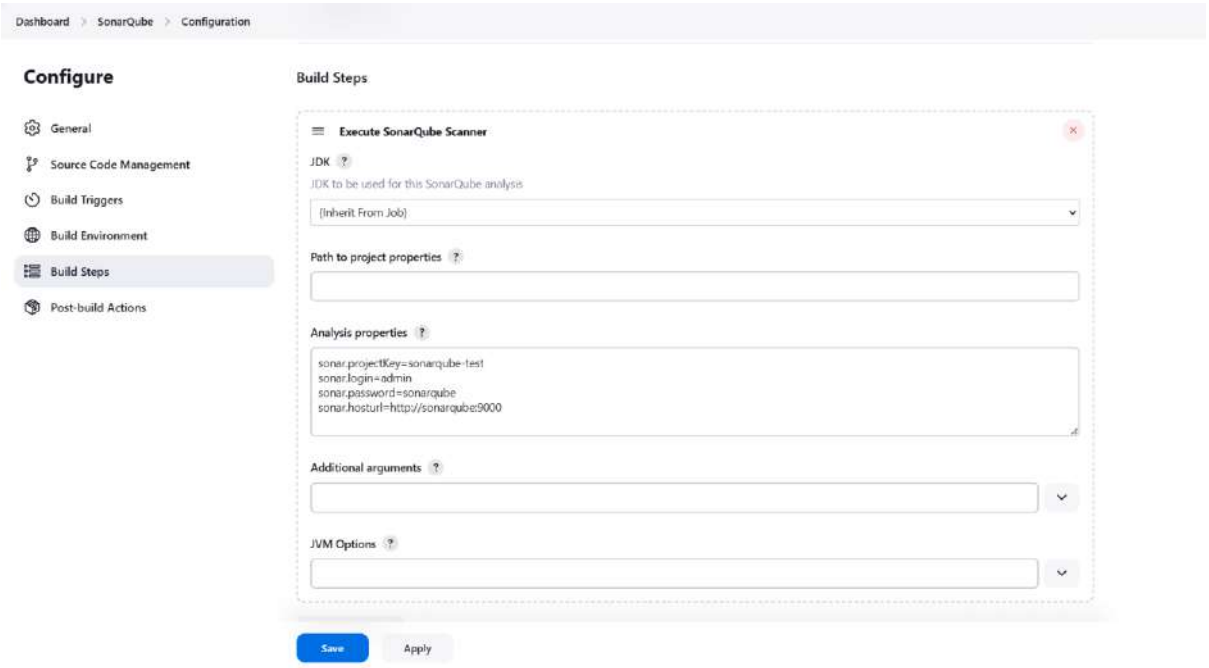
Step 7: Go to Jenkins dashboard and click on New Item and select Freestyle project and name it as SonarQube and then click on ok.

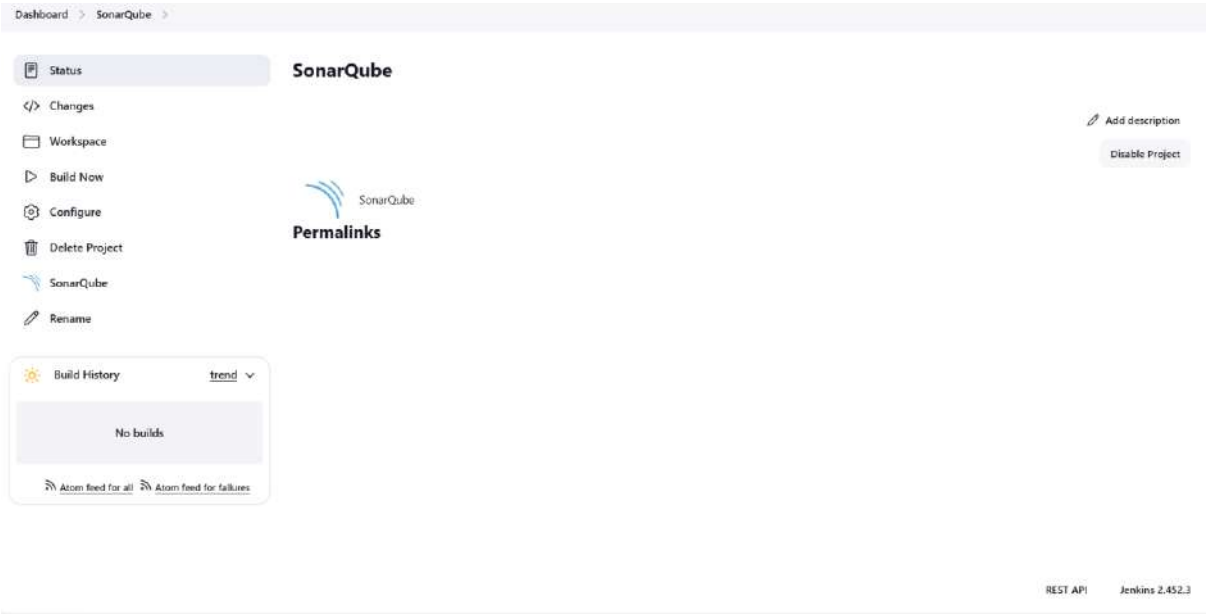


Step 8: For configuration, Select git and paste the following git repository in the repository url.
https://github.com/shazforiot/MSBuild_firstproject
This is a simple Hello world project

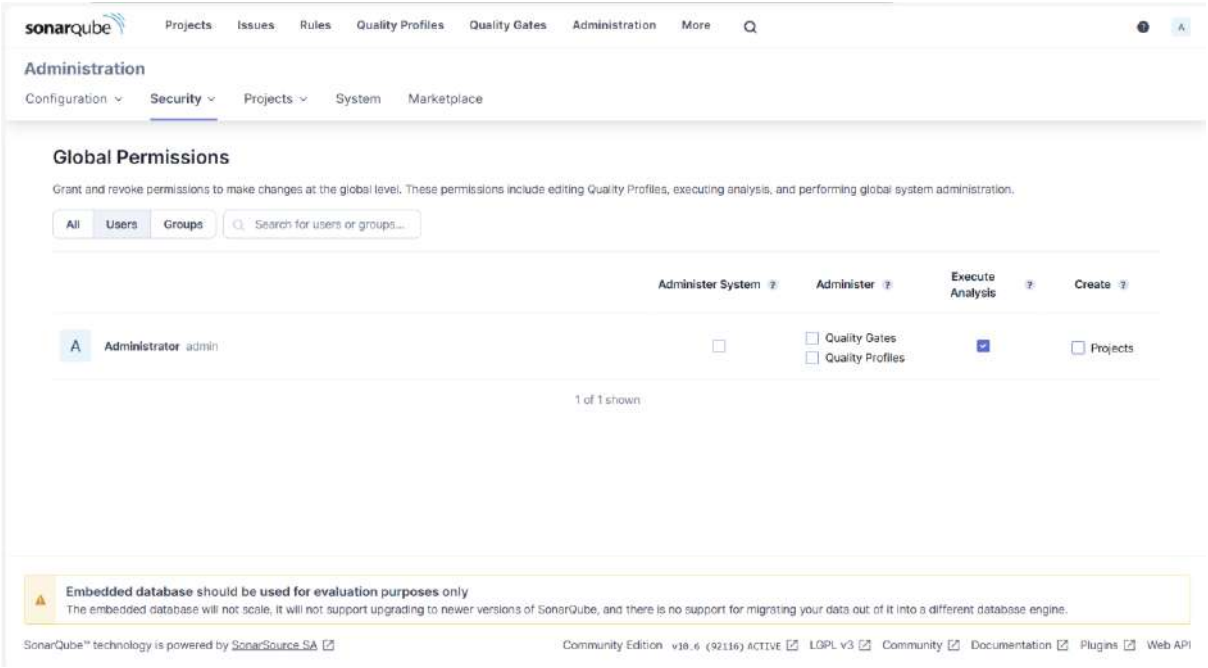


Step 9: Under the Build steps select "Execute SonarQube Scanner" option and under Analysis Properties write the following -
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=sonarqube
sonar.hosturl=http://sonarqube:9000
Then click on the save button.

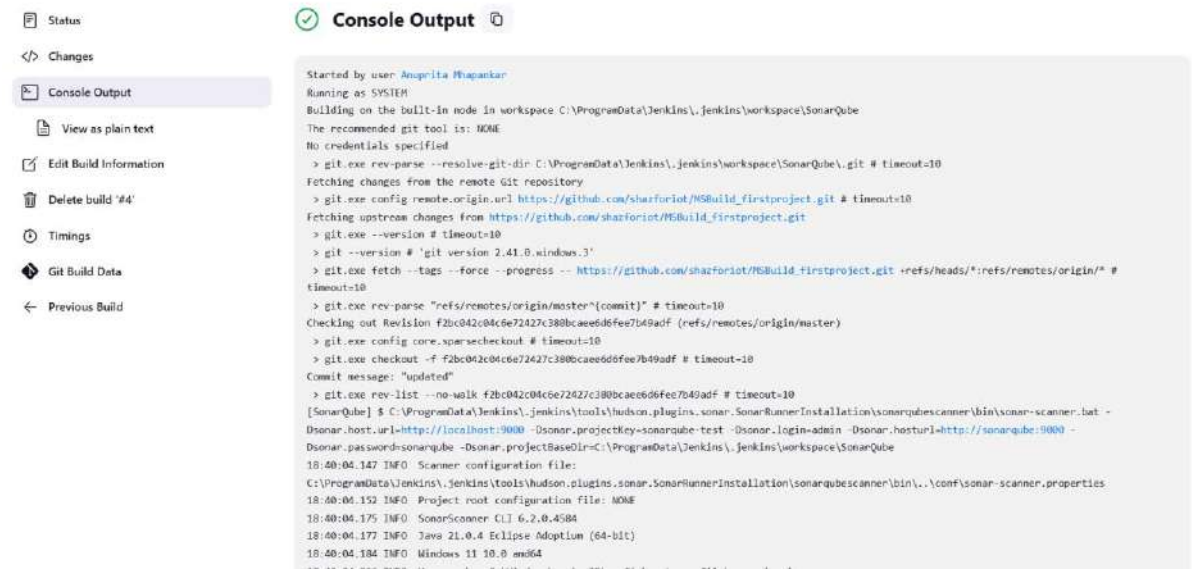


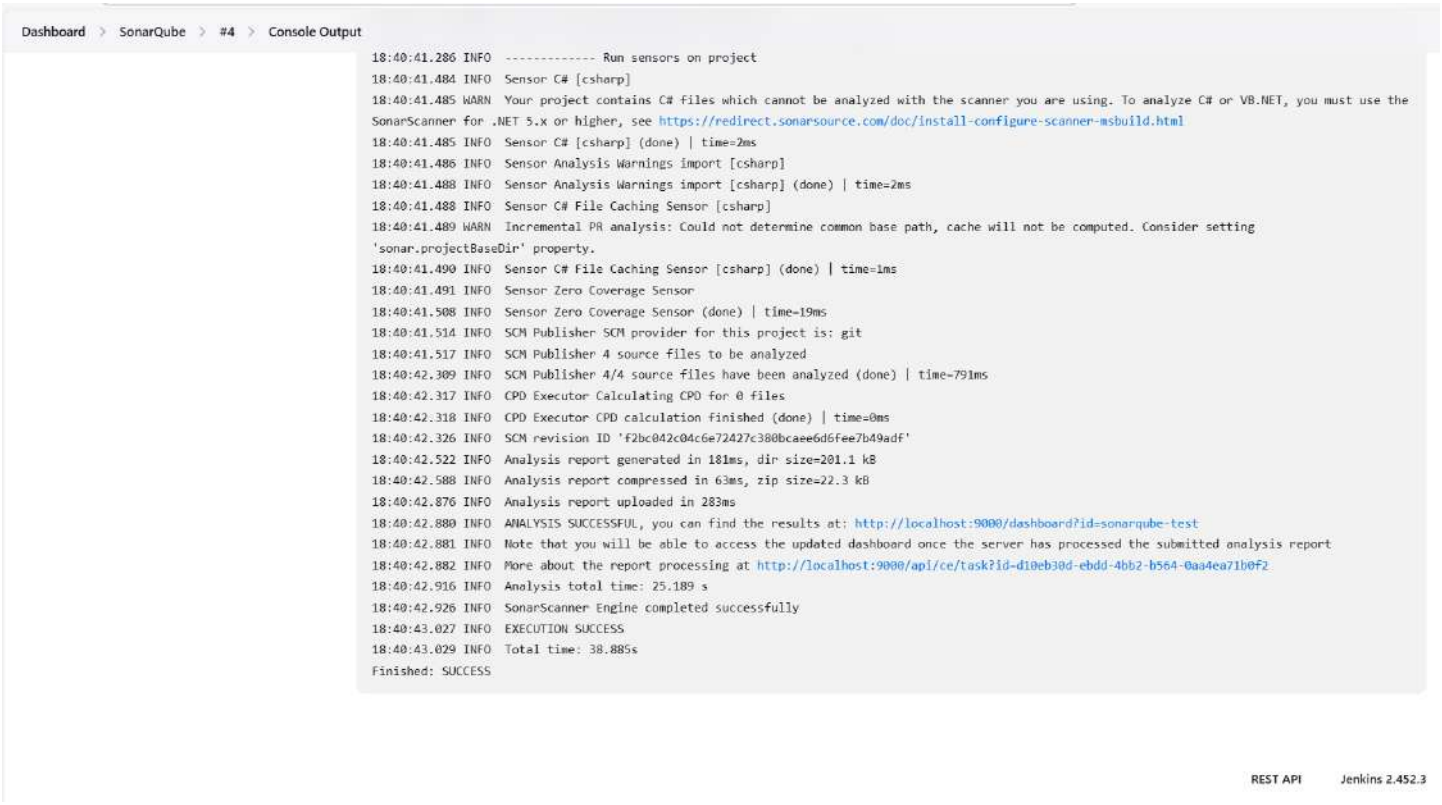


Step 10: Visit http://localhost:9000/admin/permissions and select the Users tab and for Administrator select the checkbox Execute Analysis.

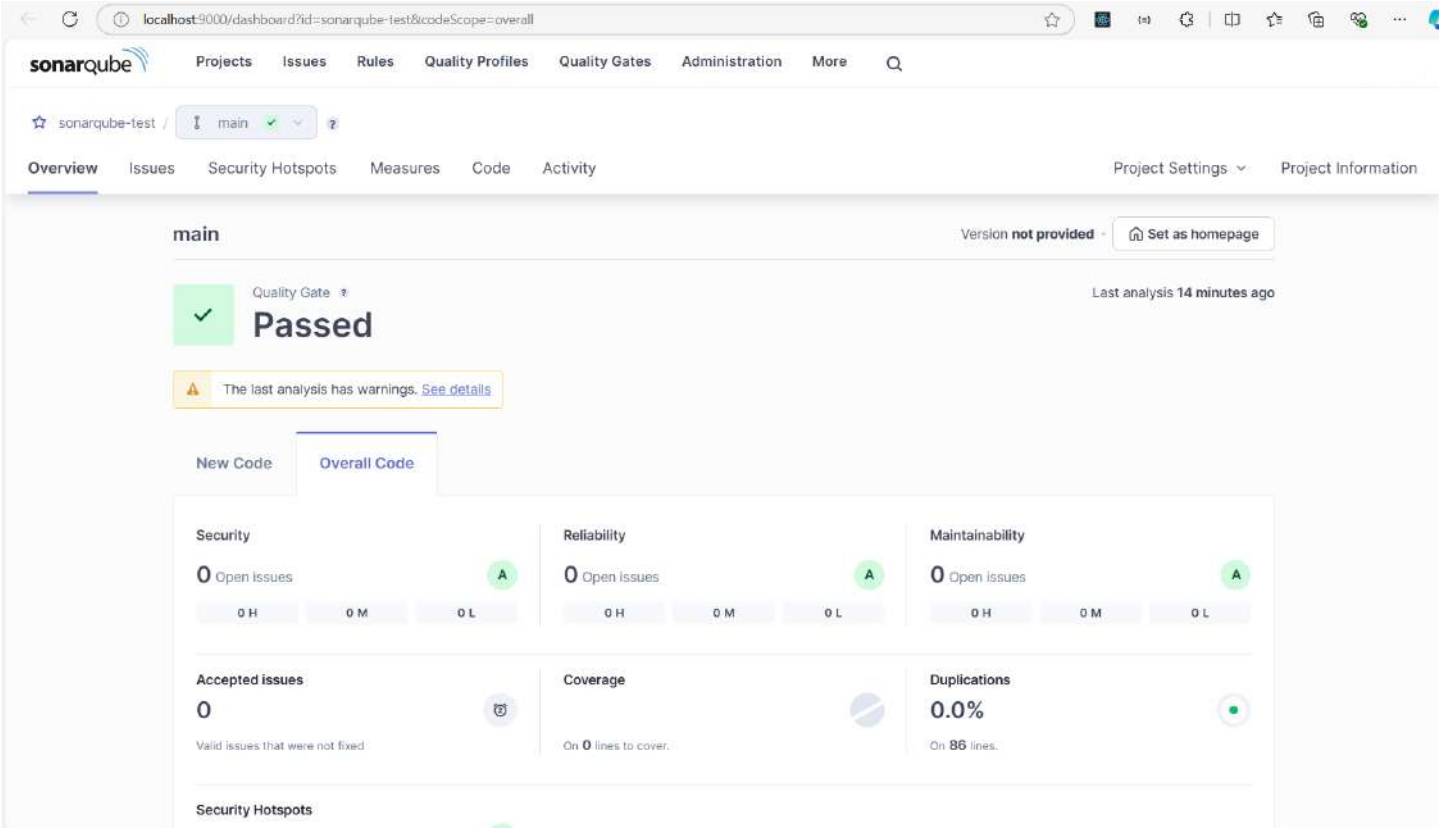


Step 11: Now, come back to Jenkins and click on Build Now. The build is success.





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


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

Advance DevOps

EXPERIMENT -8

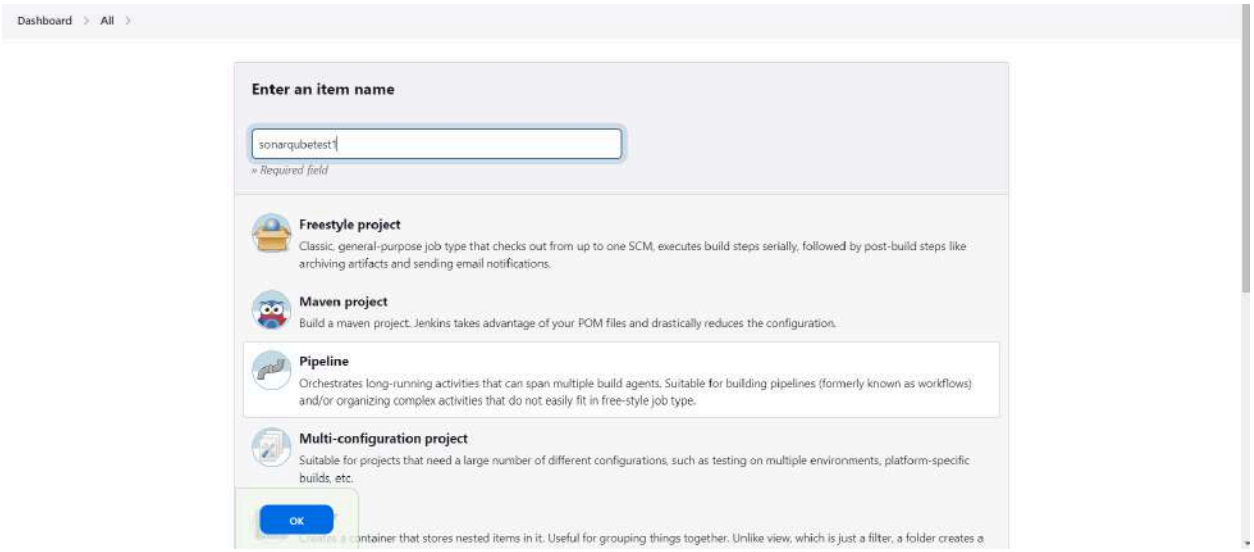
Kunal Punjabi
D15A
ROLL NO - 44

Step 1: Open Windows PowerShell and run the following command – docker run -d --name sonarqube-test1 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
erShell does not load commands from the current location by default. If you trust this command, instead type: ".\sonar-s
canner.bat". See "get-help about_Command_Precedence" for more details.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin> .\sonar-scanner.bat
11:02:02.120 INFO  Scanner configuration file: C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\..\conf\sonar-s
canner.properties
11:02:02.124 INFO  Project root configuration file: NONE
11:02:02.140 INFO  SonarScanner CLI 6.2.0.4584
11:02:02.142 INFO  Java 17.0.12 Eclipse Adoptium (64-bit)
11:02:02.142 INFO  Windows 11 10.0 amd64
11:02:02.160 INFO  User cache: C:\Users\navan\.sonar\cache
11:02:02.644 INFO  JRE provisioning: os[windows], arch[amd64]
11:02:06.241 INFO  EXECUTION FAILURE
11:02:06.243 INFO  Total time: 4.126s
11:02:06.244 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [https://api.sonarcloud.io/analysis/jres?os=windows&arch=a
md64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory
.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
11:02:06.246 ERROR
11:02:06.246 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>
```

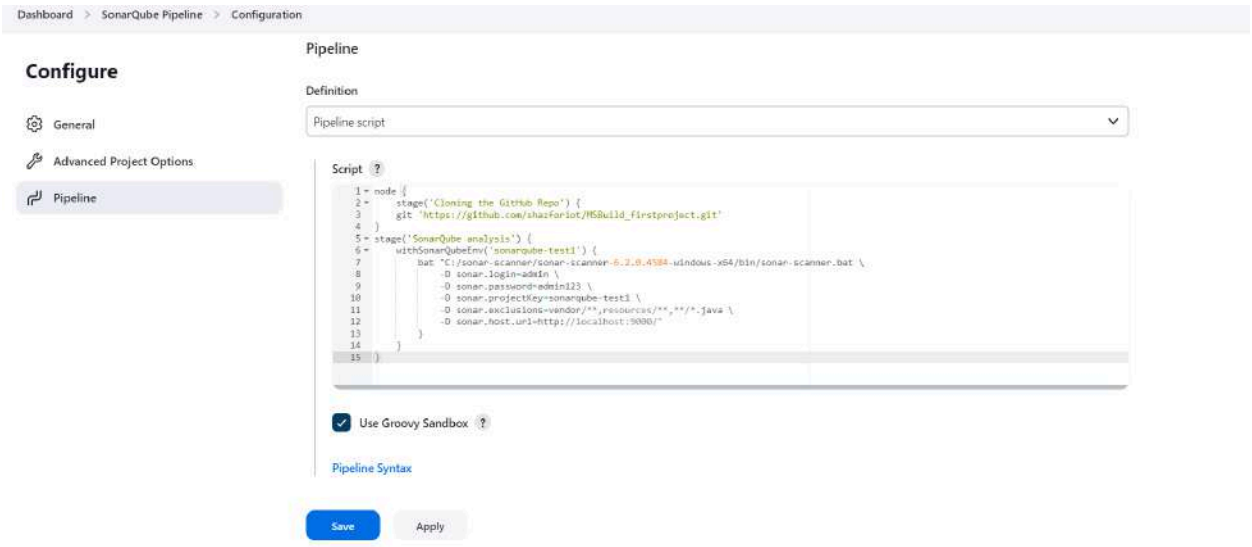
- Login to SonarQube using username admin and password admin.
- Create a manual project in SonarQube with the name sonarqube-test1

Step2: go to the jenkins and create new item select pipeline:

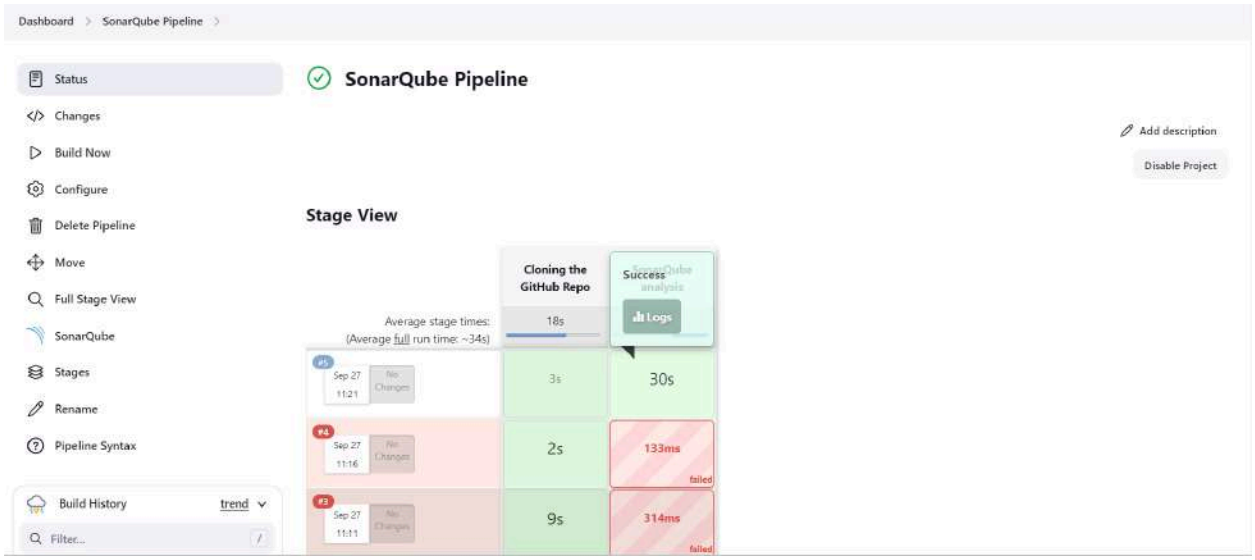


Under Pipeline Script, enter the following -

```
node {
        stage('Cloning the GitHub Repo') {
                git 'https://github.com/shazforiot/GOL.git'
        }
stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
        sh "<PATH_TO_SONARQUBE_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=http://127.0.0.1:9000/"
                }
        }
}
```



Save the changes and go to build now:



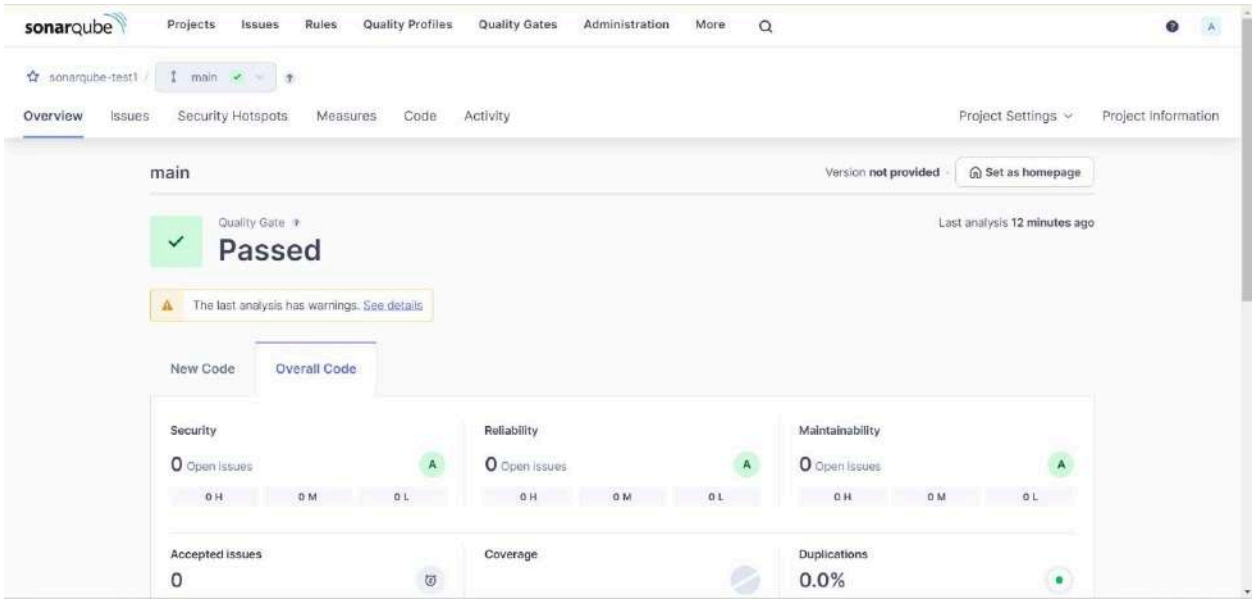
Console output:

Console Output

```
Started by user jai navani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
```

```
11:22:18.236 INFO  Sensor C# File Caching Sensor [csharp]
11:22:18.237 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir'
property.
11:22:18.237 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=1ms
11:22:18.237 INFO  Sensor Zero Coverage Sensor
11:22:18.251 INFO  Sensor Zero Coverage Sensor (done) | time=14ms
11:22:18.256 INFO  SCM Publisher SCM provider for this project is: git
11:22:18.257 INFO  SCM Publisher 4 source files to be analyzed
11:22:18.789 INFO  SCM Publisher 4/4 source files have been analyzed (done) | time=531ms
11:22:18.793 INFO  CPD Executor Calculating CPD for 0 files
11:22:18.795 INFO  CPD Executor CPD calculation finished (done) | time=0ms
11:22:18.810 INFO  SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
11:22:19.074 INFO  Analysis report generated in 134ms, dir size=201.0 kB
11:22:19.137 INFO  Analysis report compressed in 45ms, zip size=22.5 kB
11:22:19.351 INFO  Analysis report uploaded in 212ms
11:22:19.353 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test1
11:22:19.354 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
11:22:19.354 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=971ae2f2-4e0f-49a7-88c4-ad4a6cceddf8
11:22:19.366 INFO  Analysis total time: 24.819 s
11:22:19.368 INFO  SonarScanner Engine completed successfully
11:22:19.454 INFO  EXECUTION SUCCESS
11:22:19.455 INFO  Total time: 29.696s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

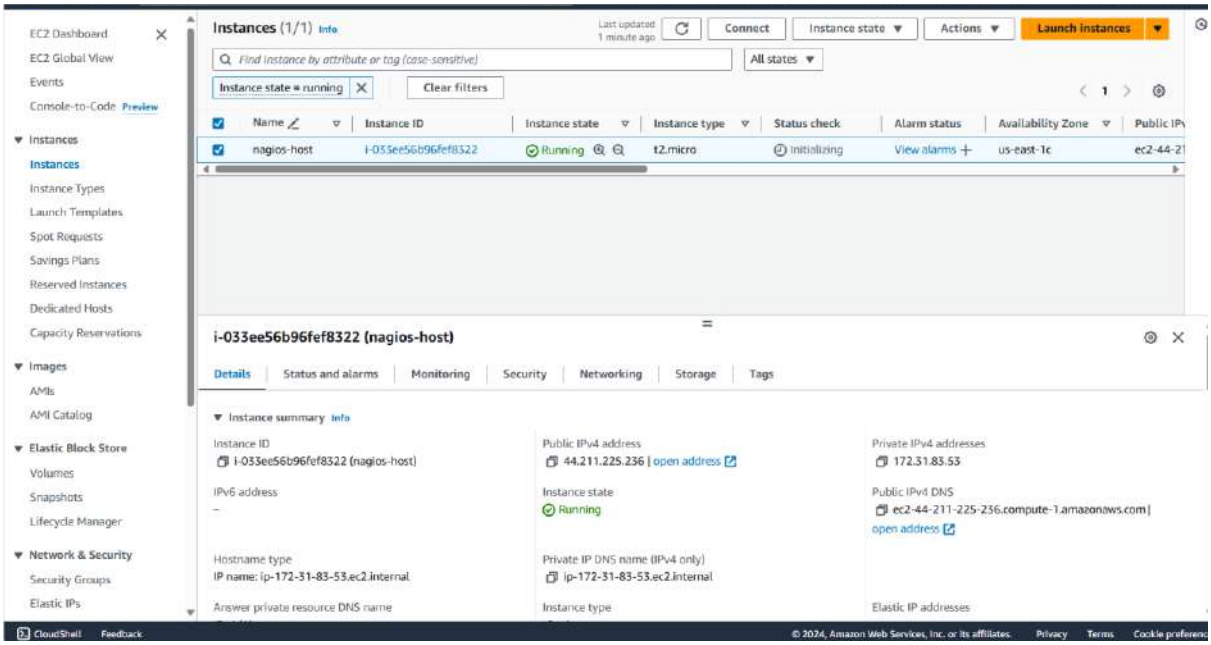
Output:



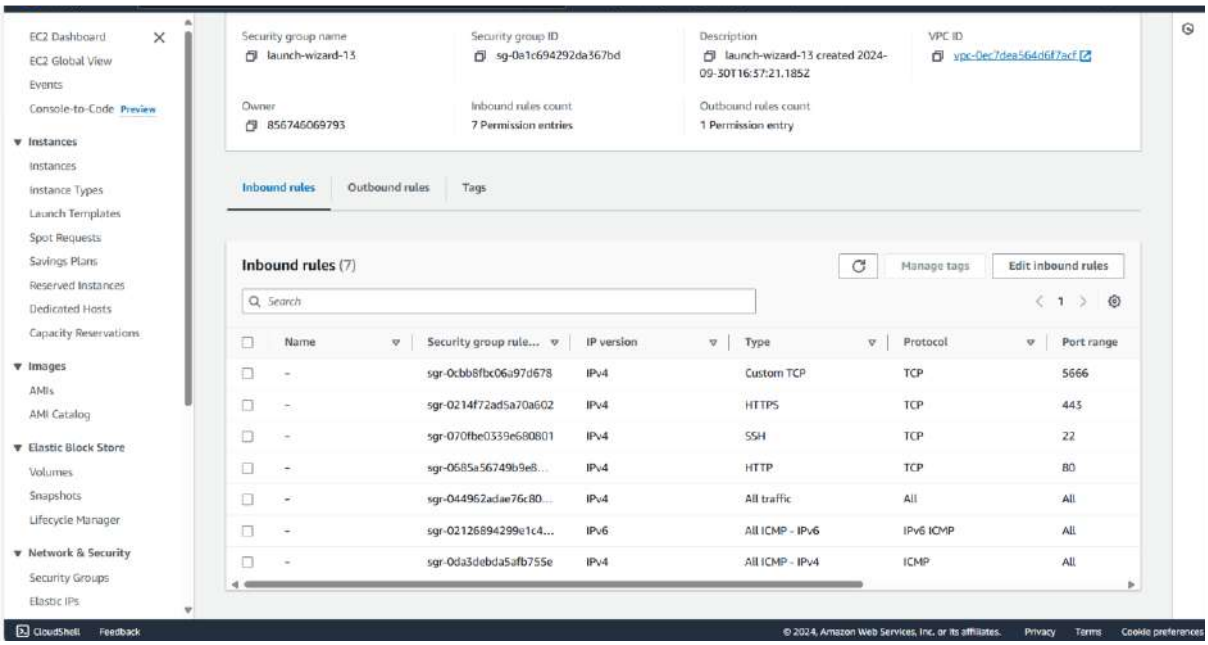
Experiment 9

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

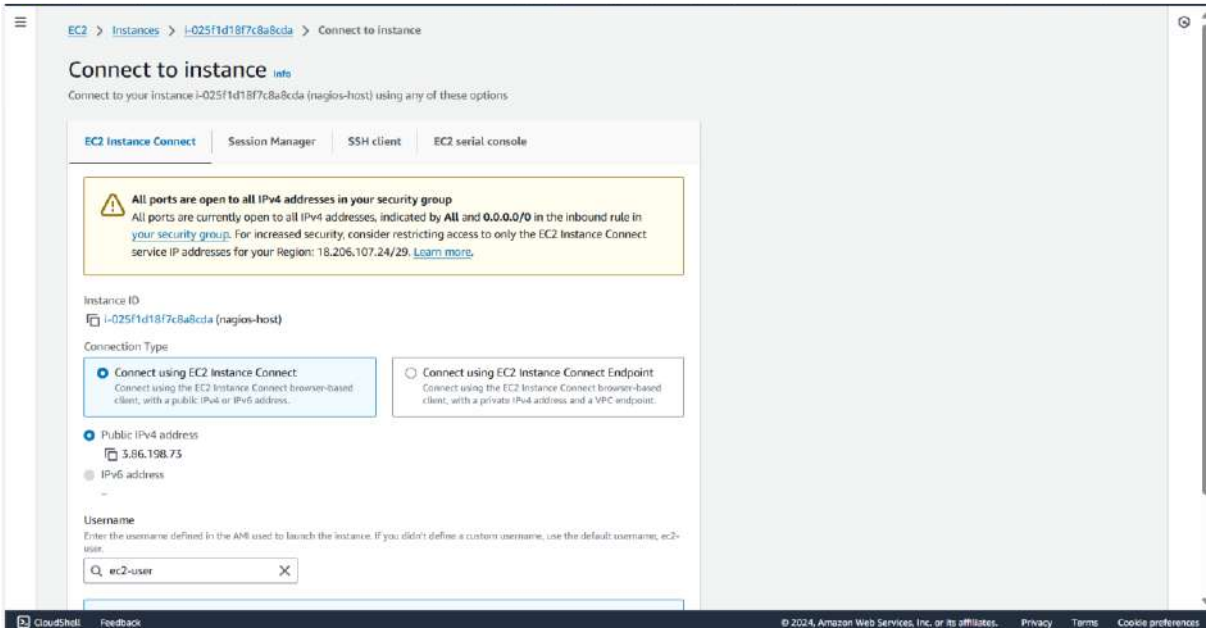
Step 1: Create an EC2 Instance and name it as nagios-host



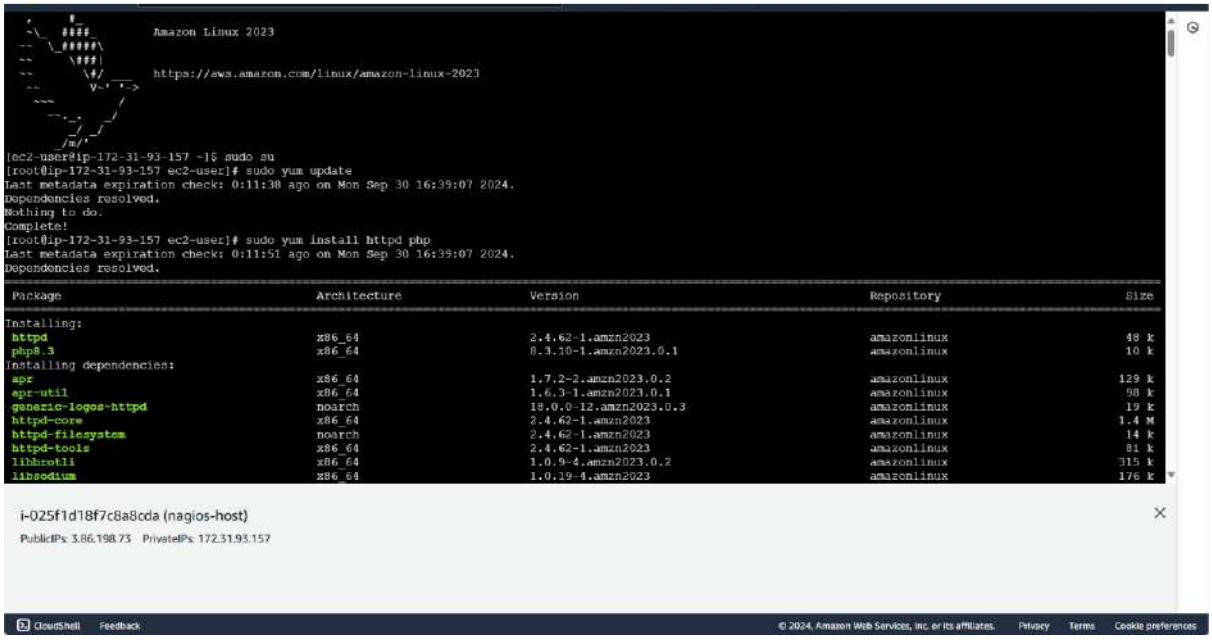
Step 2: Under the security groups, click on edit inbound rules and set as shown in the figure below

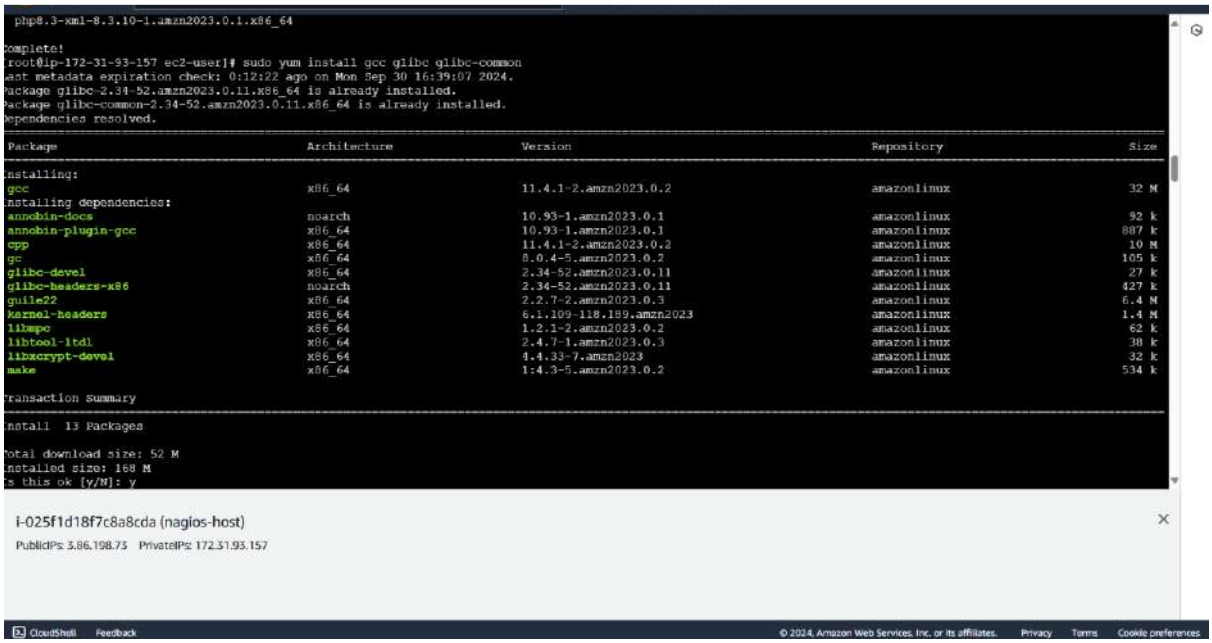


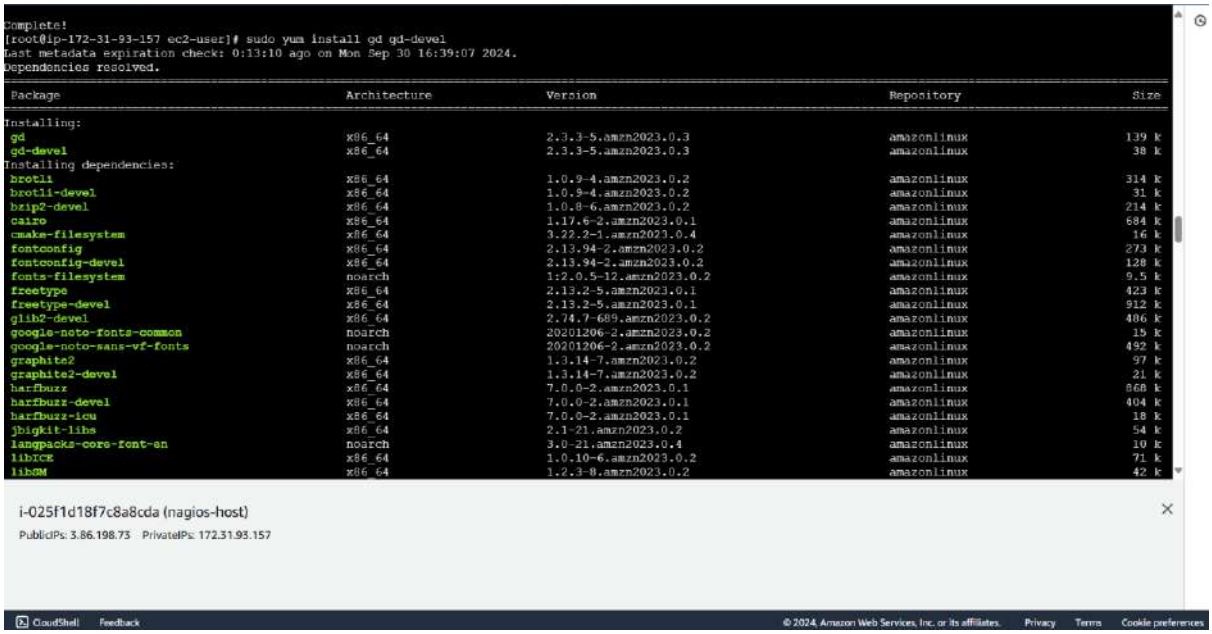
Step 3: Then select the instance nagios-host and then connect the instance.



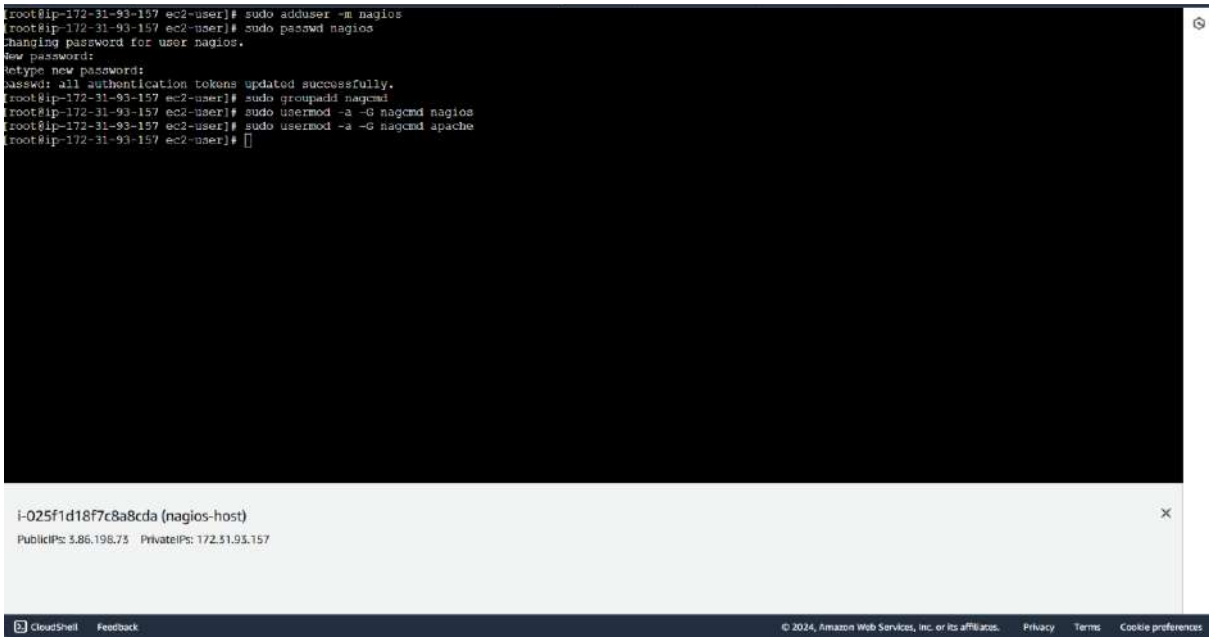
Step 3: Now, run the following commands -
sudo su
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel



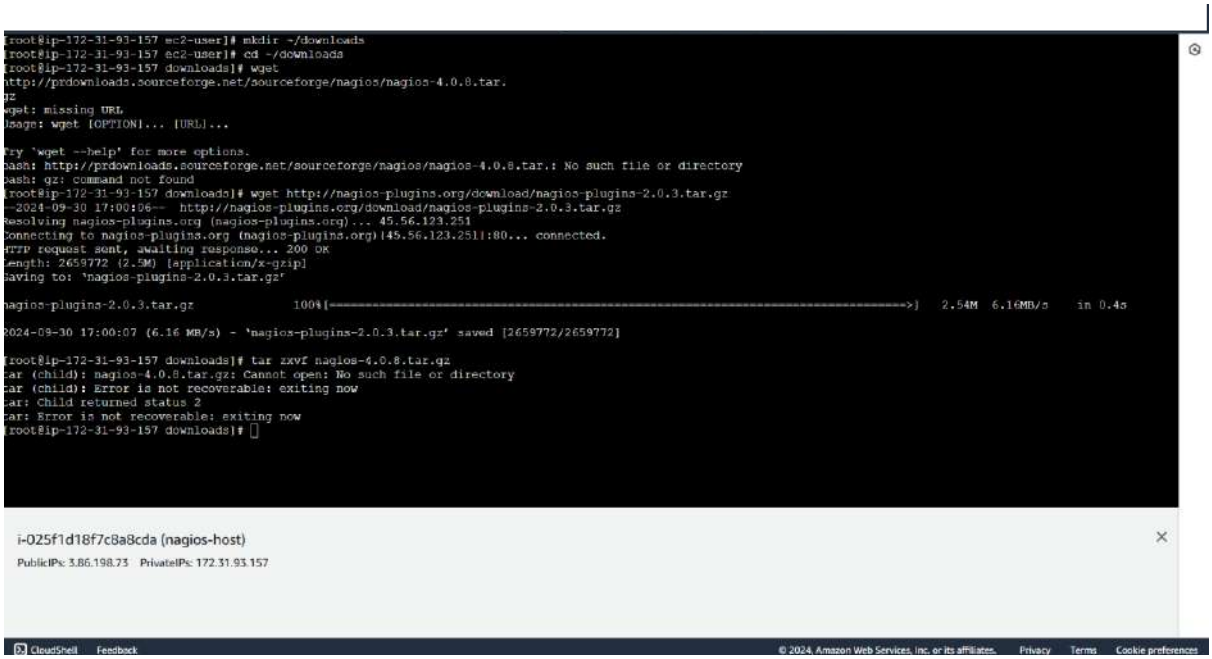




Step 4: Create a new nagios user with its password.
sudo adduser -m nagios
sudo passwd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache



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



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

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

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

2024-09-30 17:03:05 (2.2 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

```
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

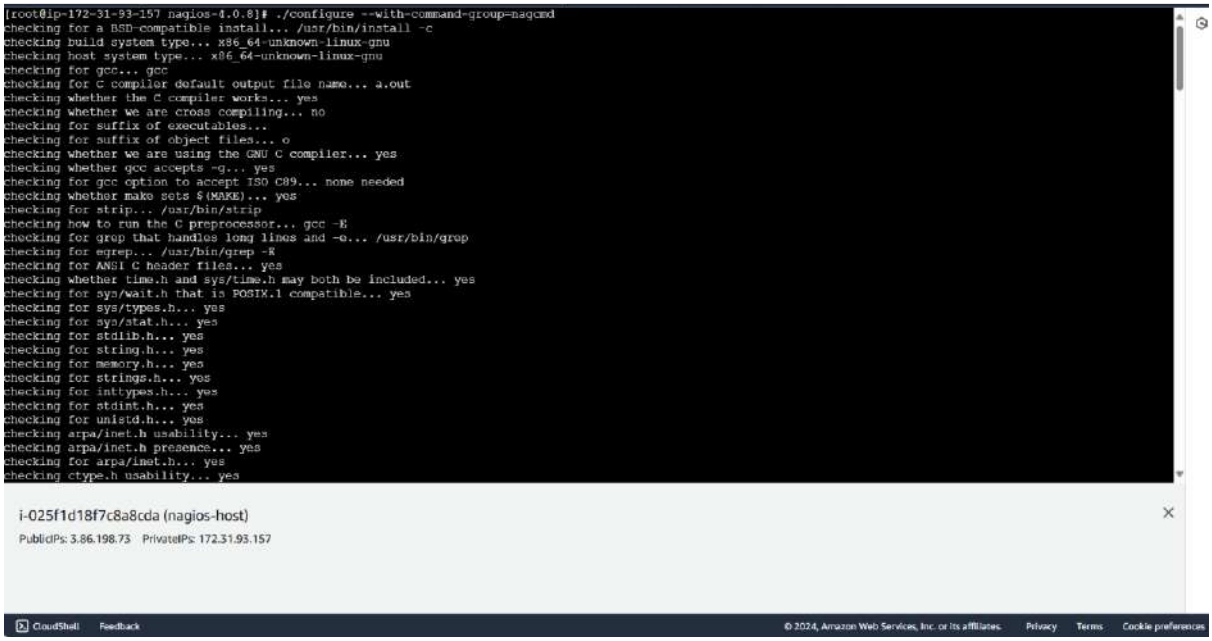
```
nagios-plugins-2.0.3/plugins-scripts/Makefile.am
nagios-plugins-2.0.3/plugins-scripts/subst.in
nagios-plugins-2.0.3/plugins-scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins-scripts/check_log.sh
nagios-plugins-2.0.3/plugins-scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
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)

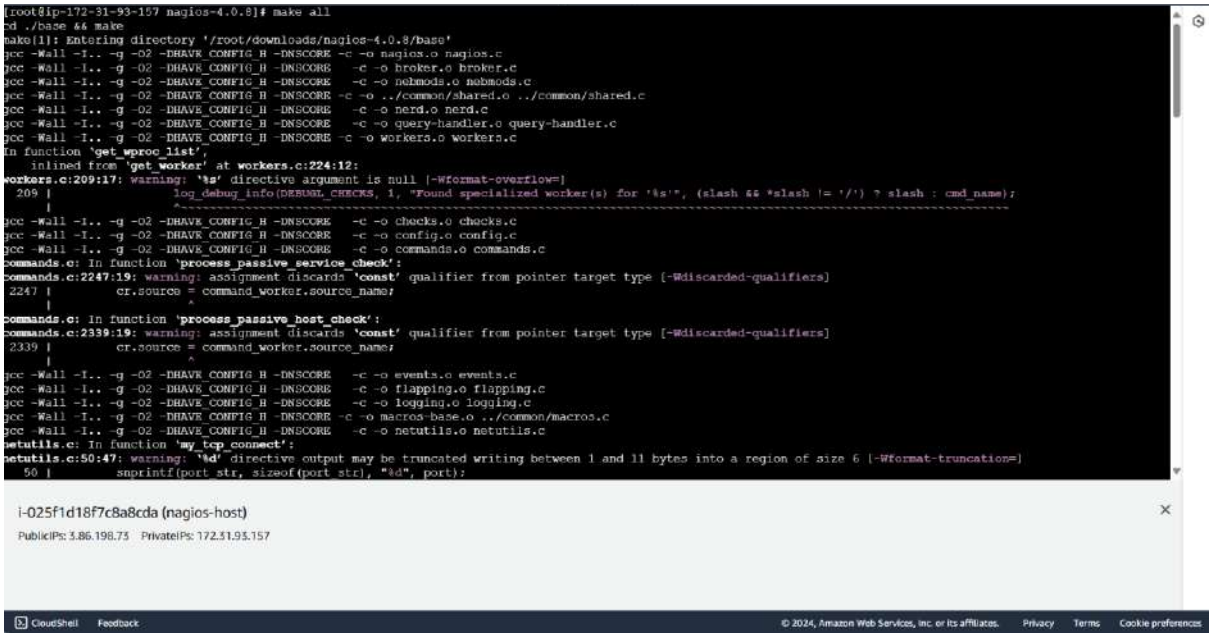
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

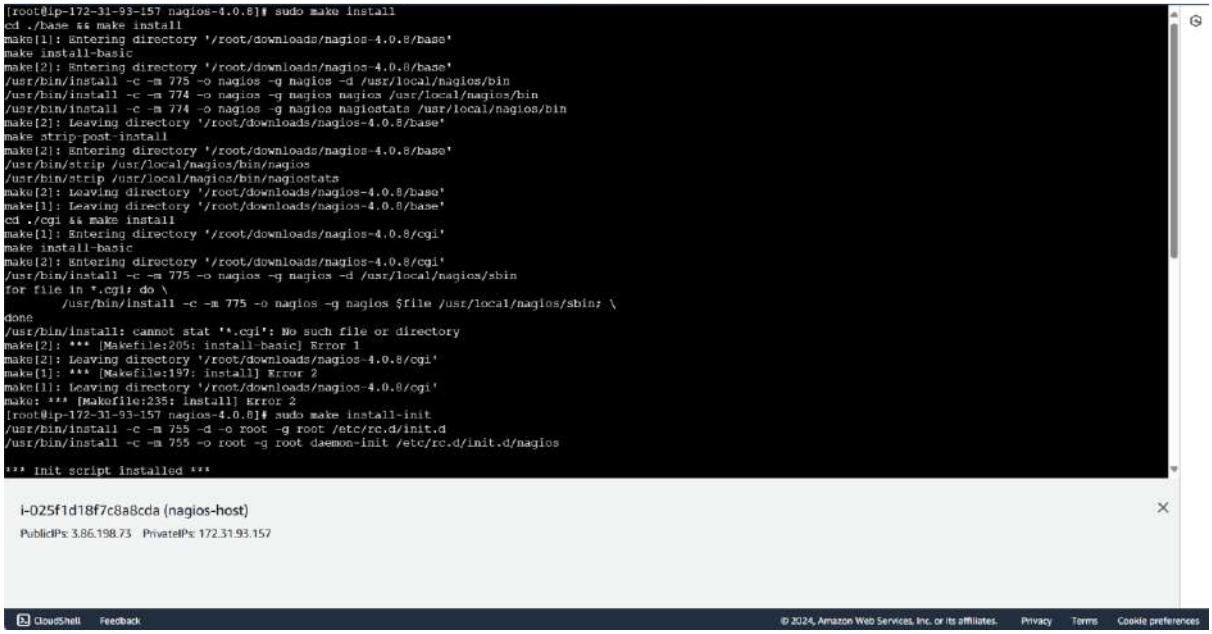
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

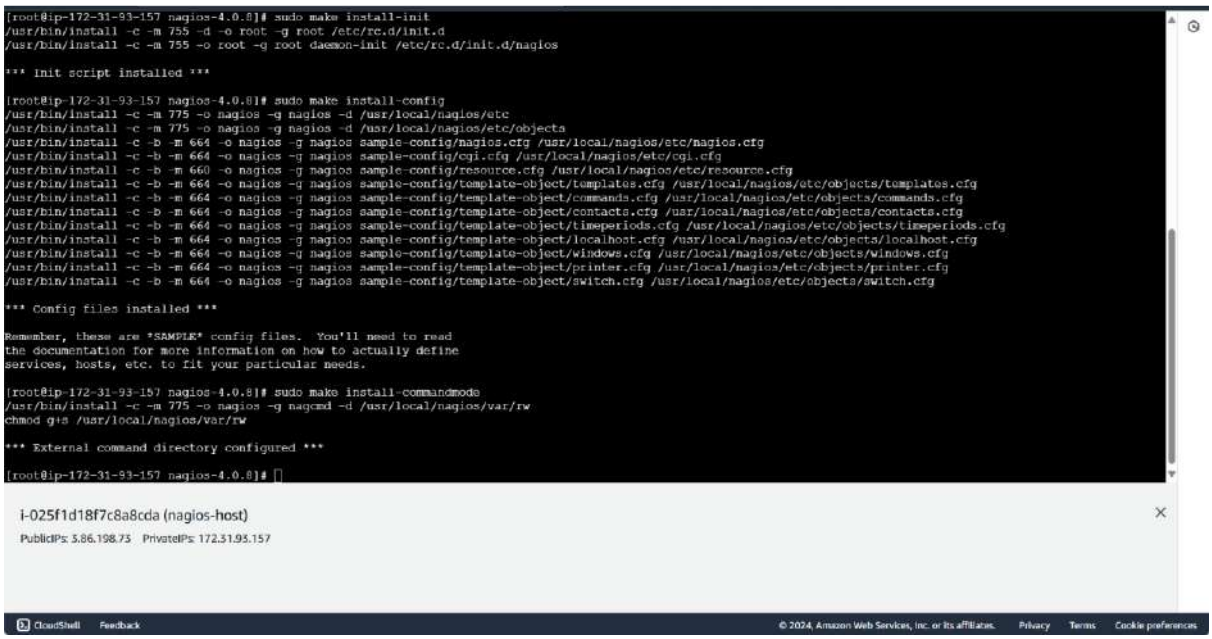
Step 6: Now to run the configuration script run the following command.
./configure --with-command-group=nagcmd



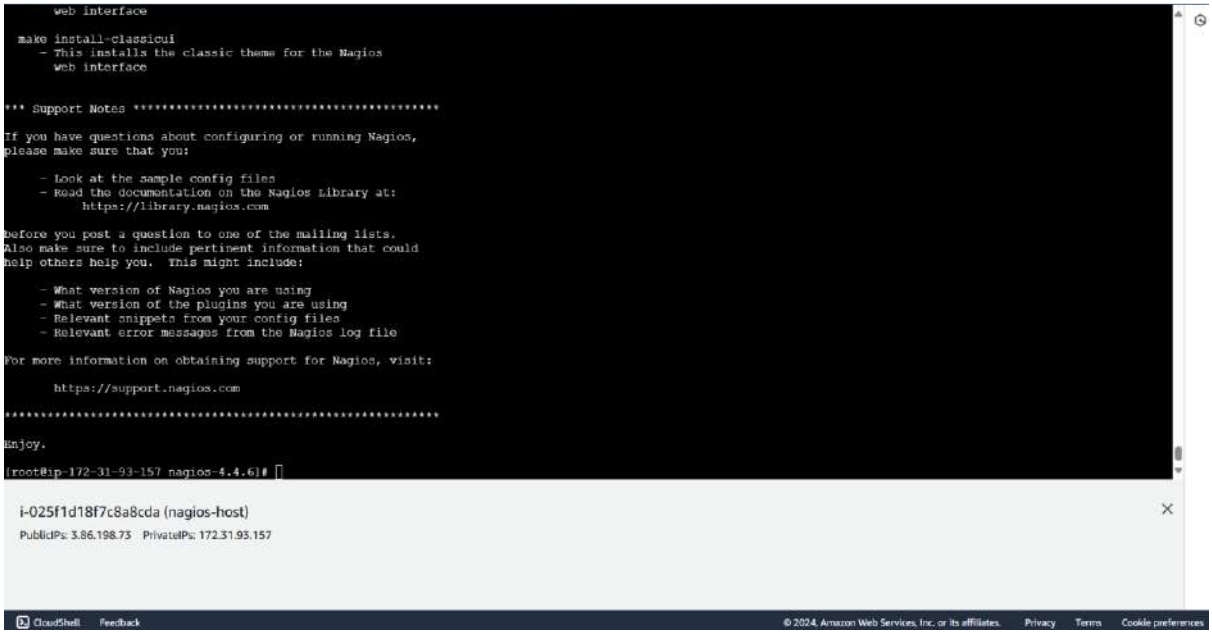
Step 7: Now, to compile the source code run the following command -
make all



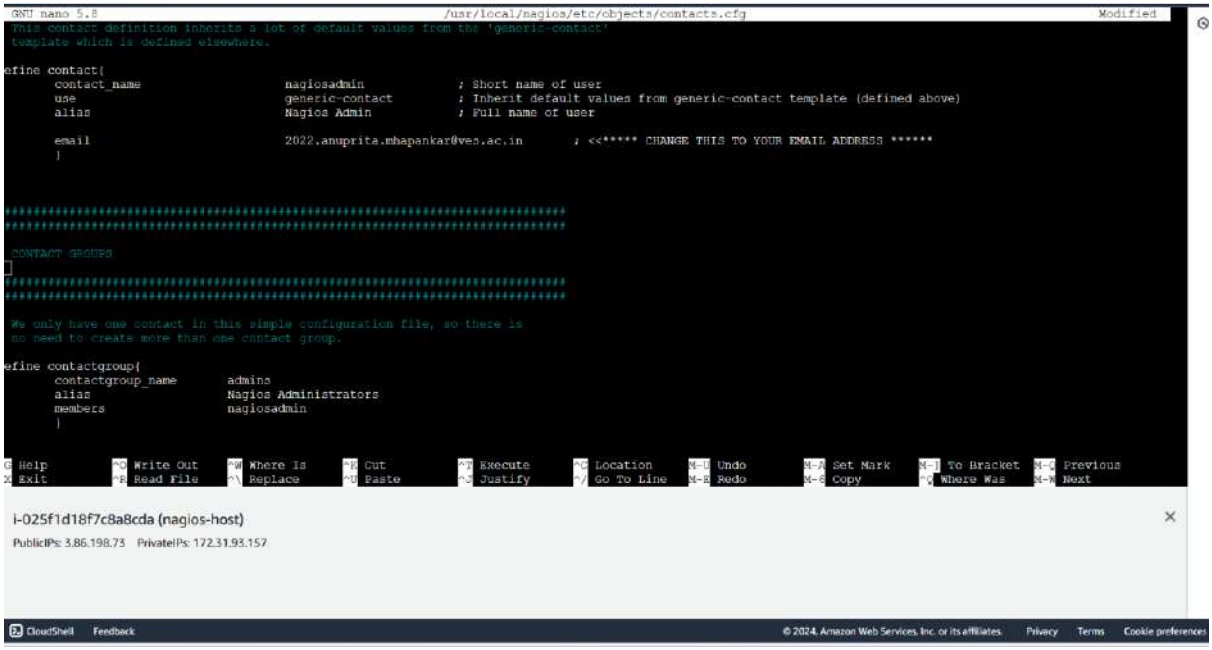




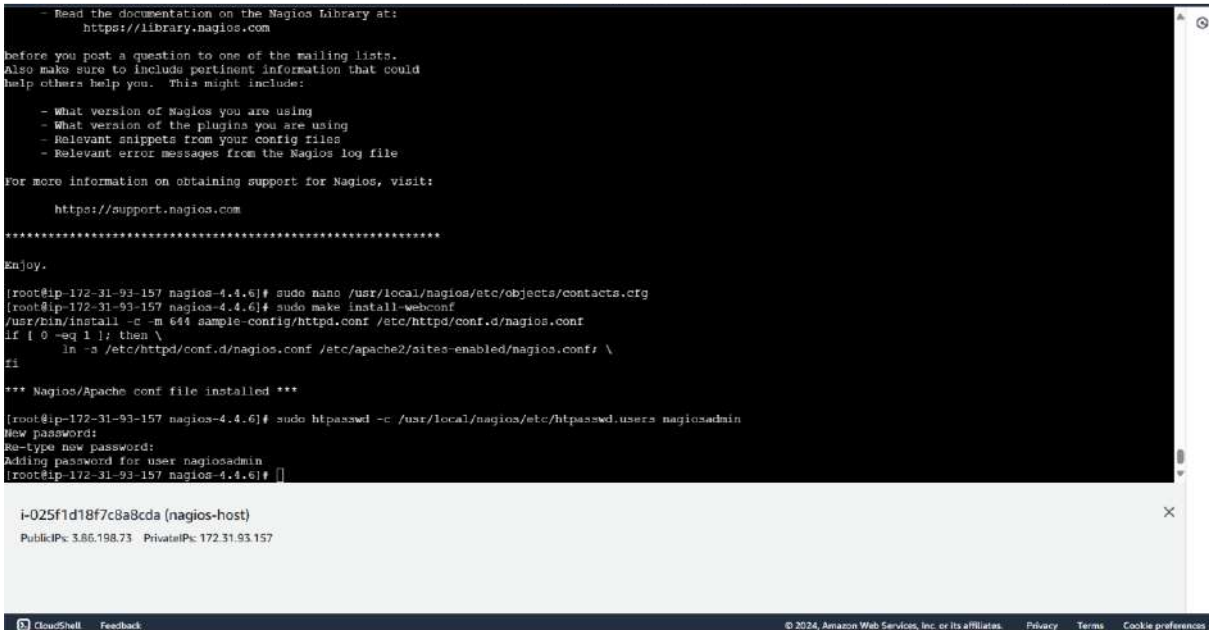
To resolve the errors run the following commands -
sudo yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel
rm -rf nagios-4.0.8
cd ~/downloads/nagios-4.4.6
./configure --with-command-group=nagcmd
make all
sudo make install

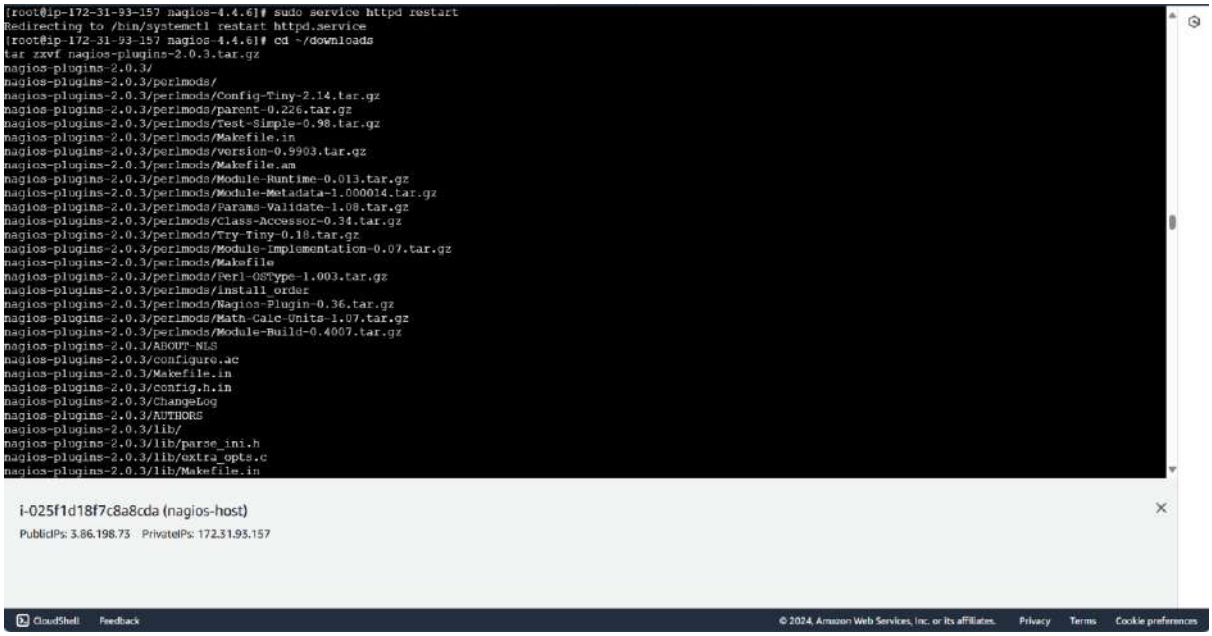


Step 8: Edit the config file and change the email address.
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

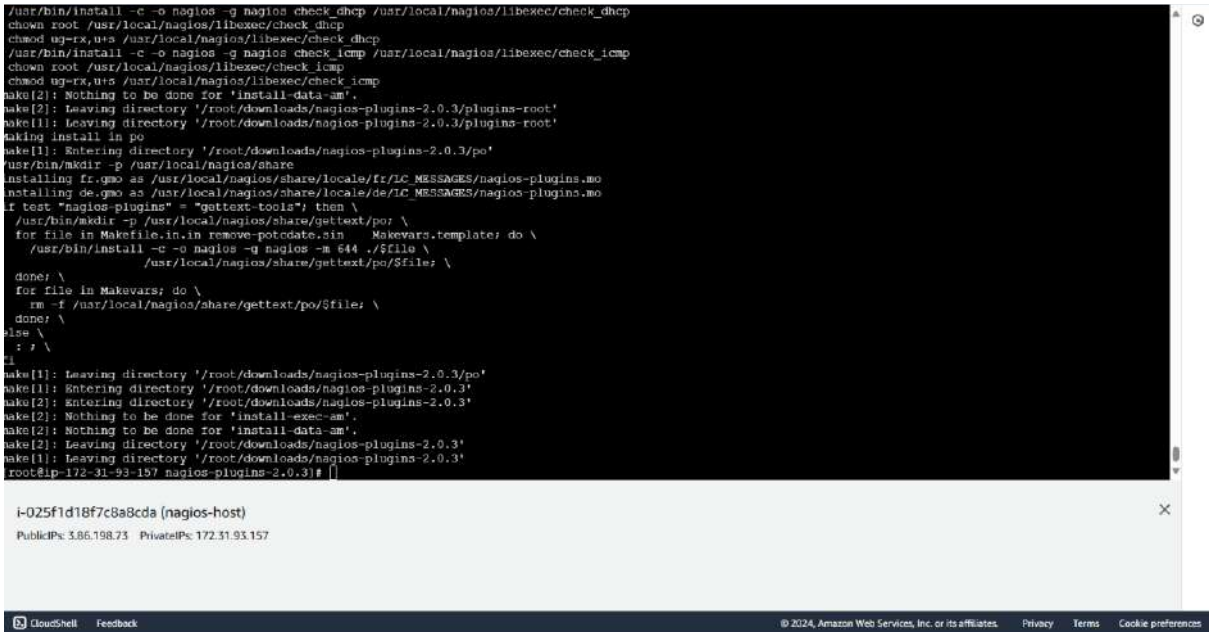


Step 9: Now run the following commands –
sudo make install-webconf
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
sudo service httpd restart
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz





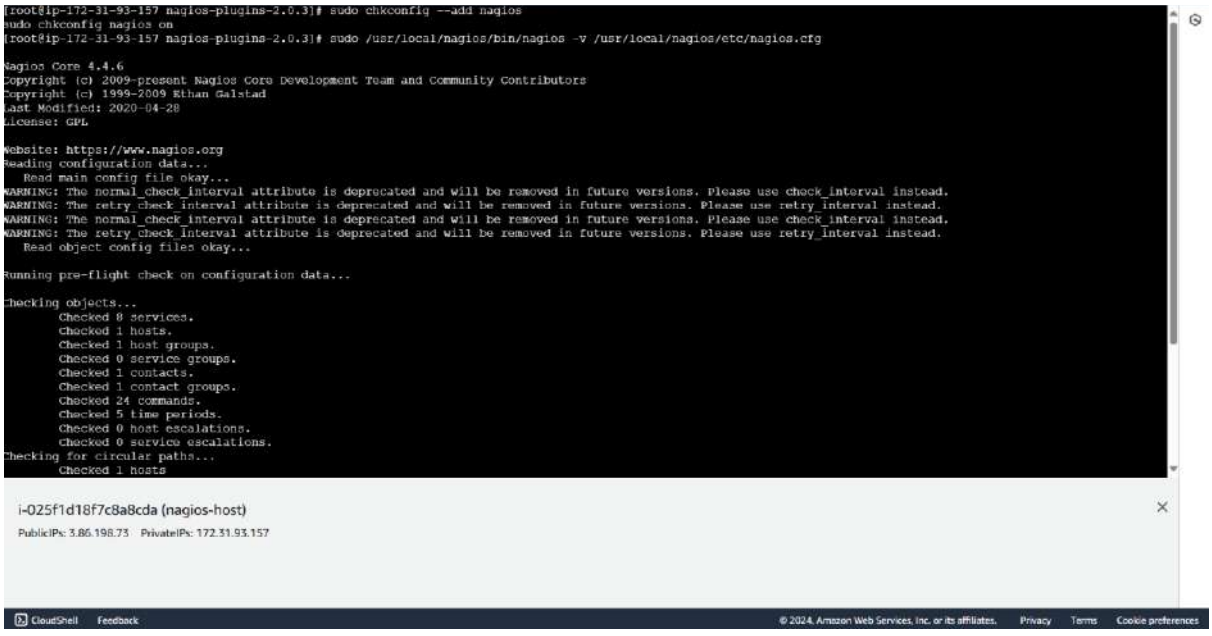
Step 10: Compile and install plugins
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install



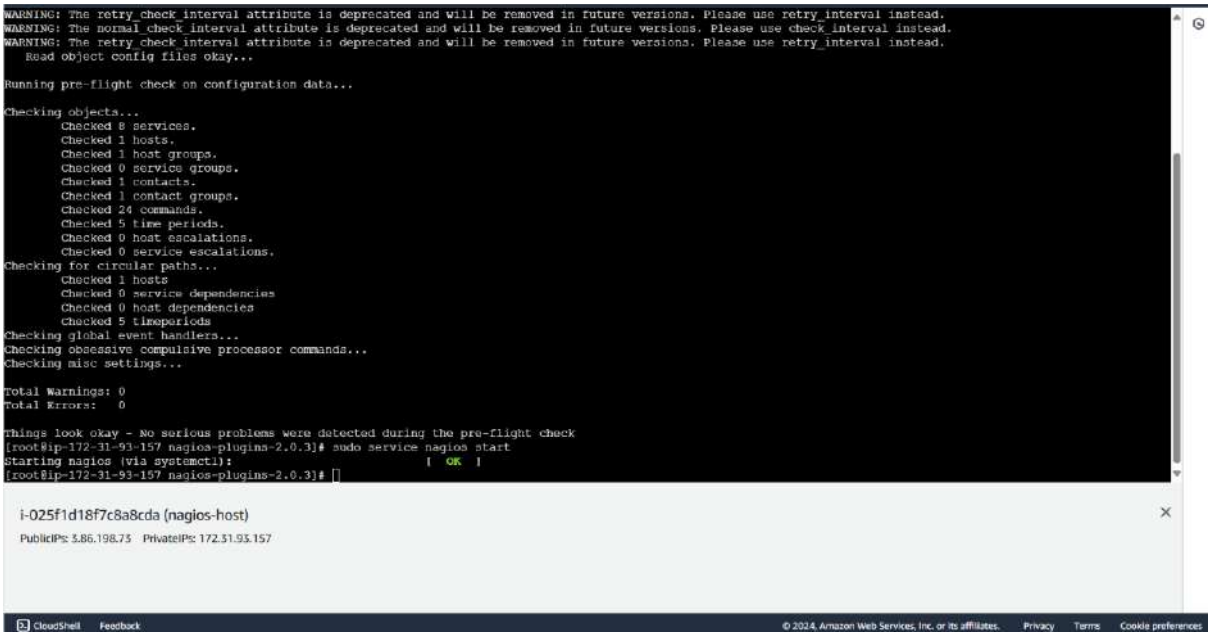
Step 11: To start nagios run the following commands –
sudo chkconfig --add nagios

sudo chkconfig nagios on

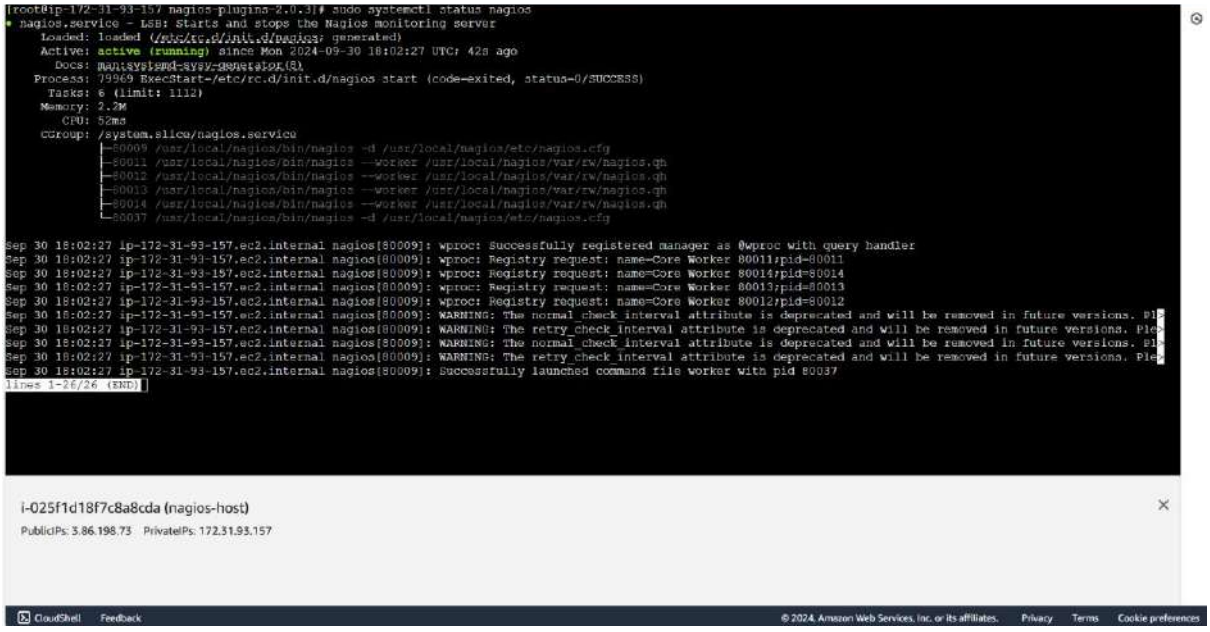
Verify using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



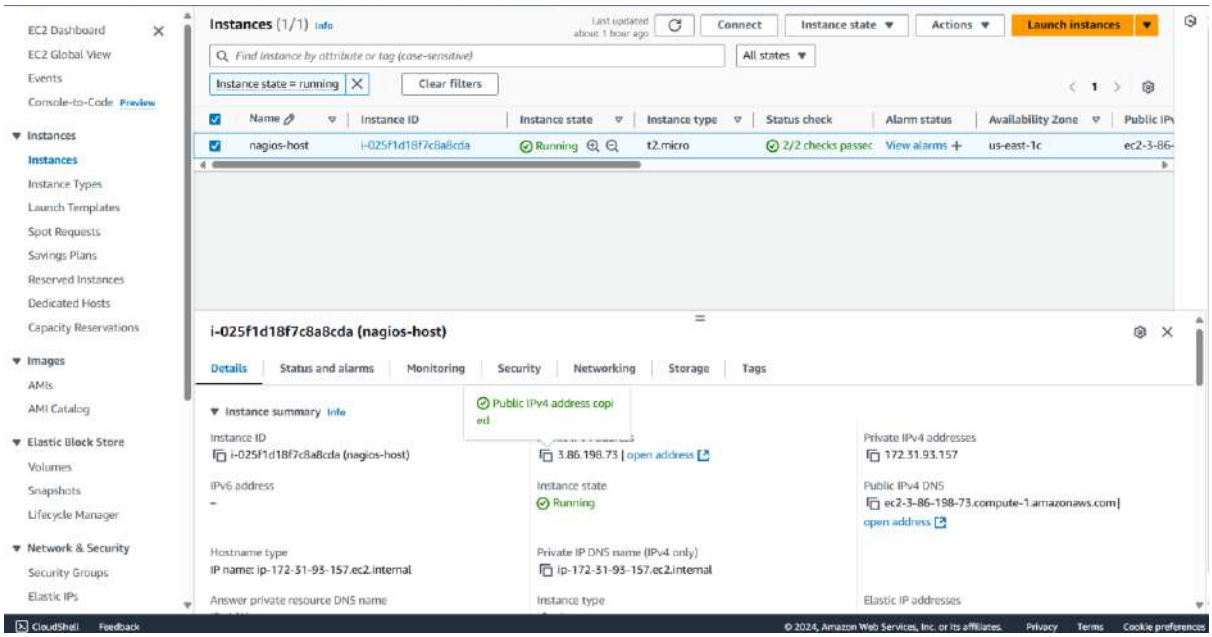
If there are no errors run the following command –
sudo service nagios start



Check status using the following command -
sudo systemctl status nagios



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

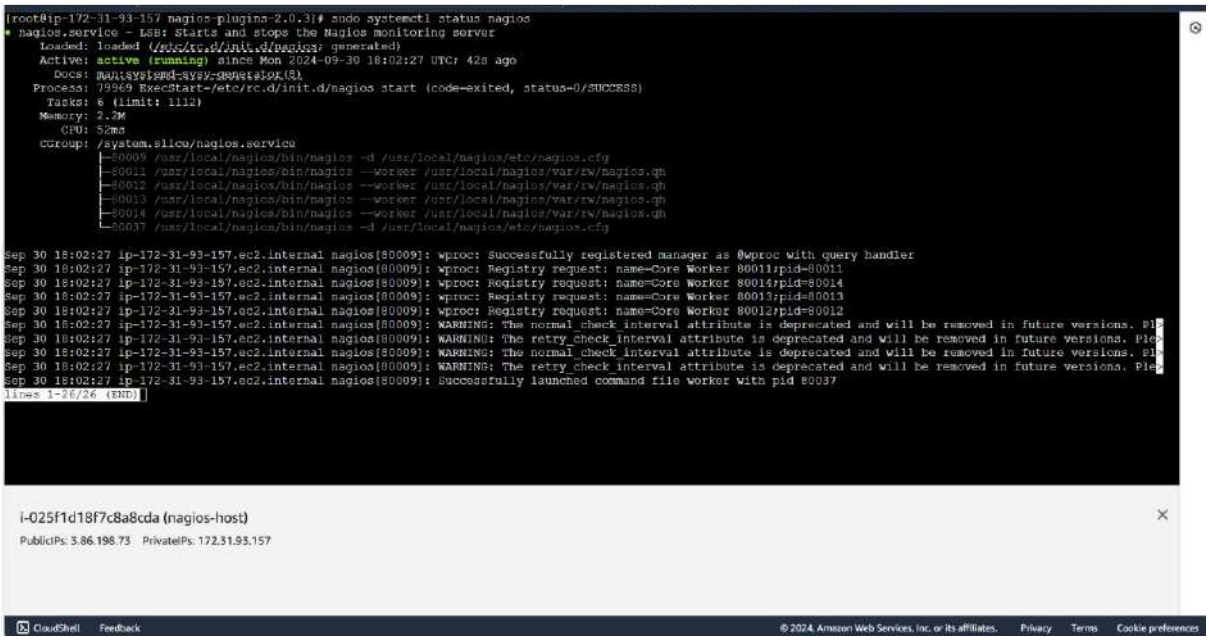


Step 13: Now visit http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.

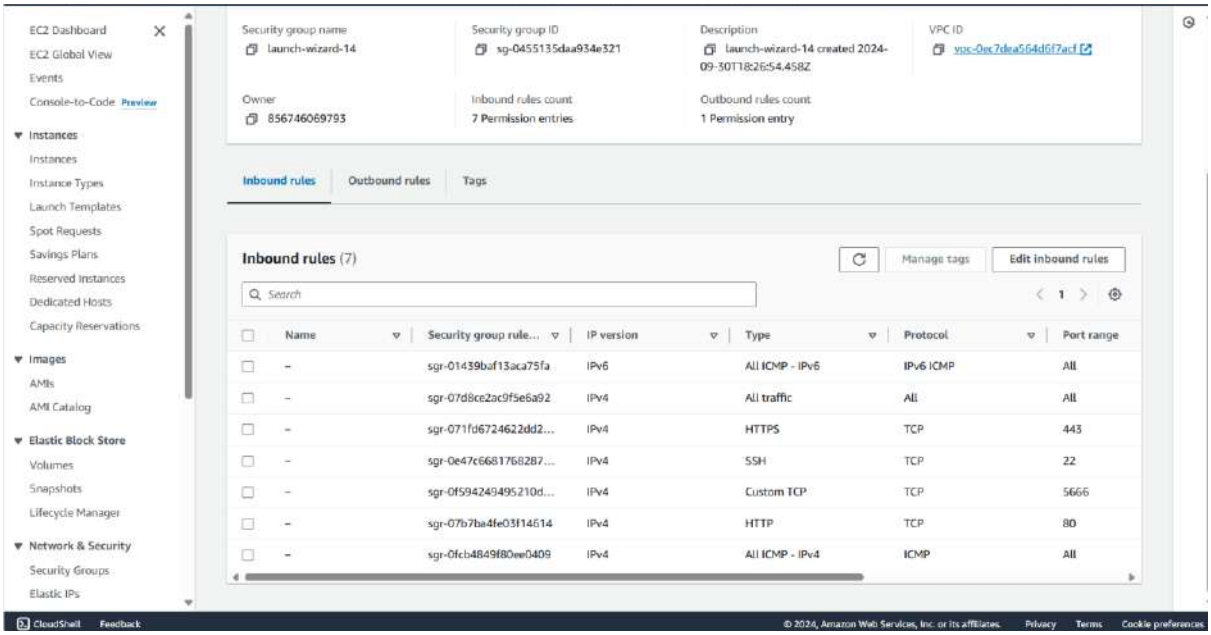
Experiment 10

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

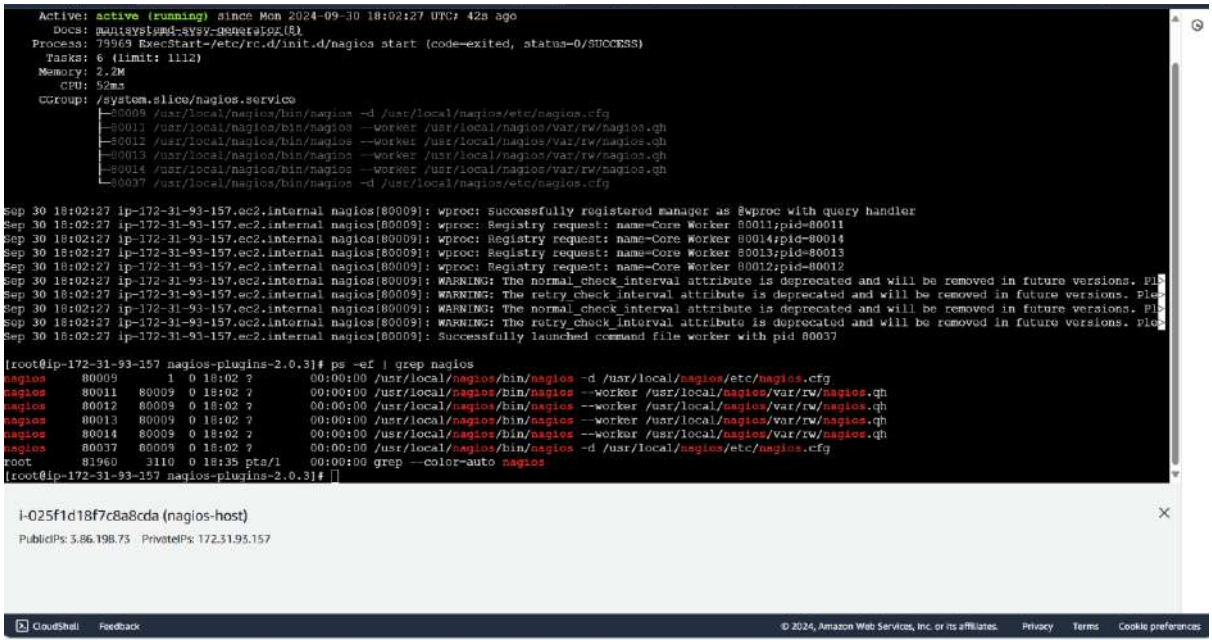
Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -
sudo systemctl status nagios
on the nagios-host instance.



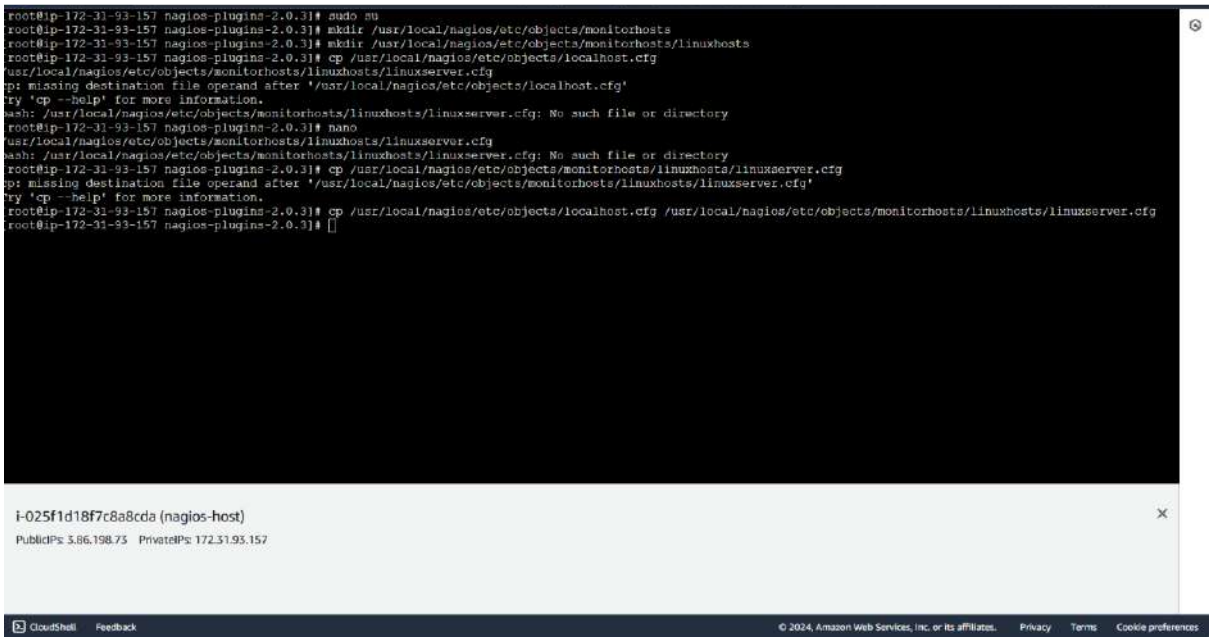
Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.
For now, leave this machine as it is, and go back to your nagios-host machine.



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

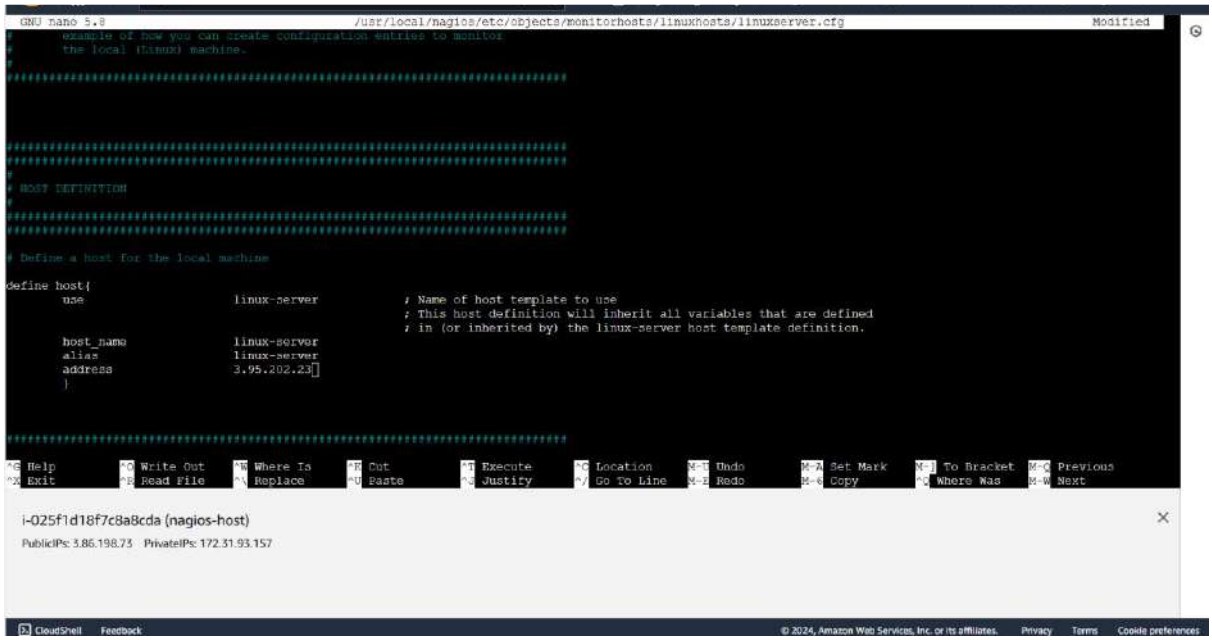


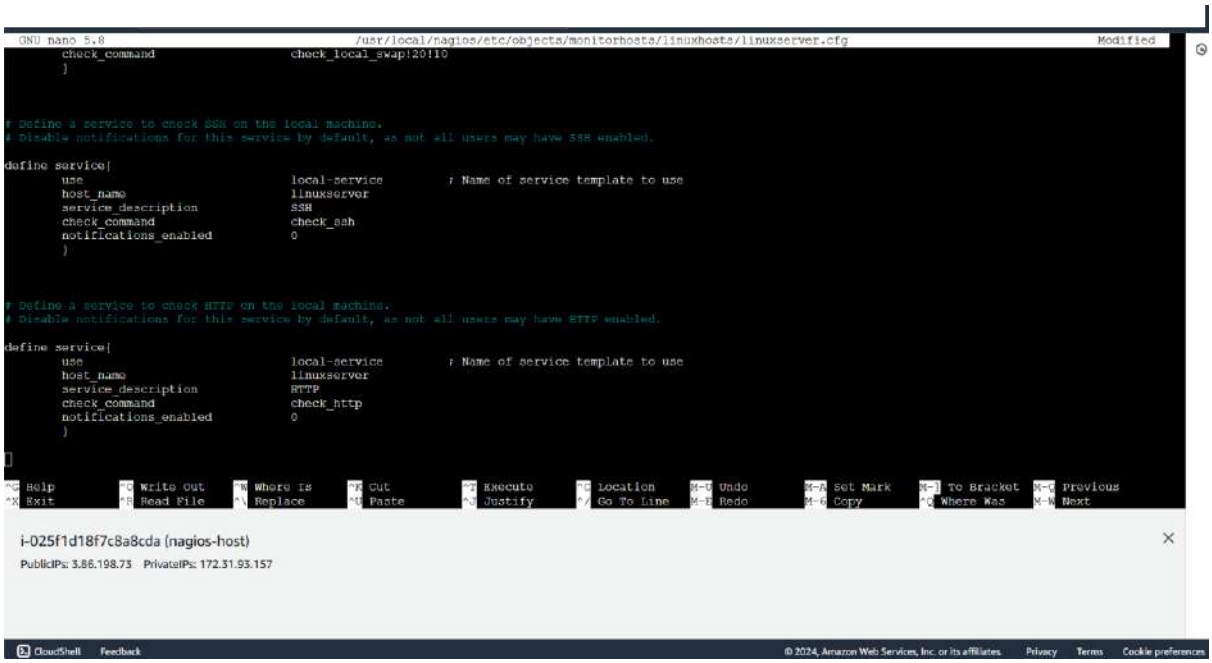
Step 4: Now, run the following commands -
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp/usr/local/nagios/etc/objects/localhost.cfg/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg



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

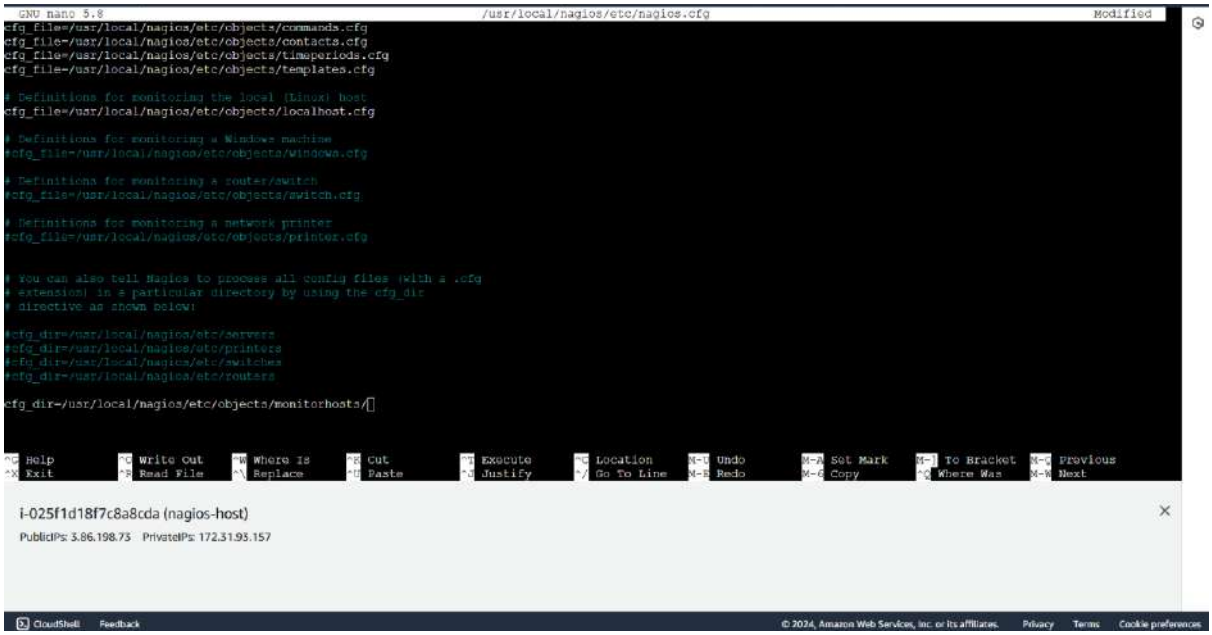
Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
Change address to the public IP address of your LINUX CLIENT.
Change hostgroup_name under hostgroup to linux-servers1





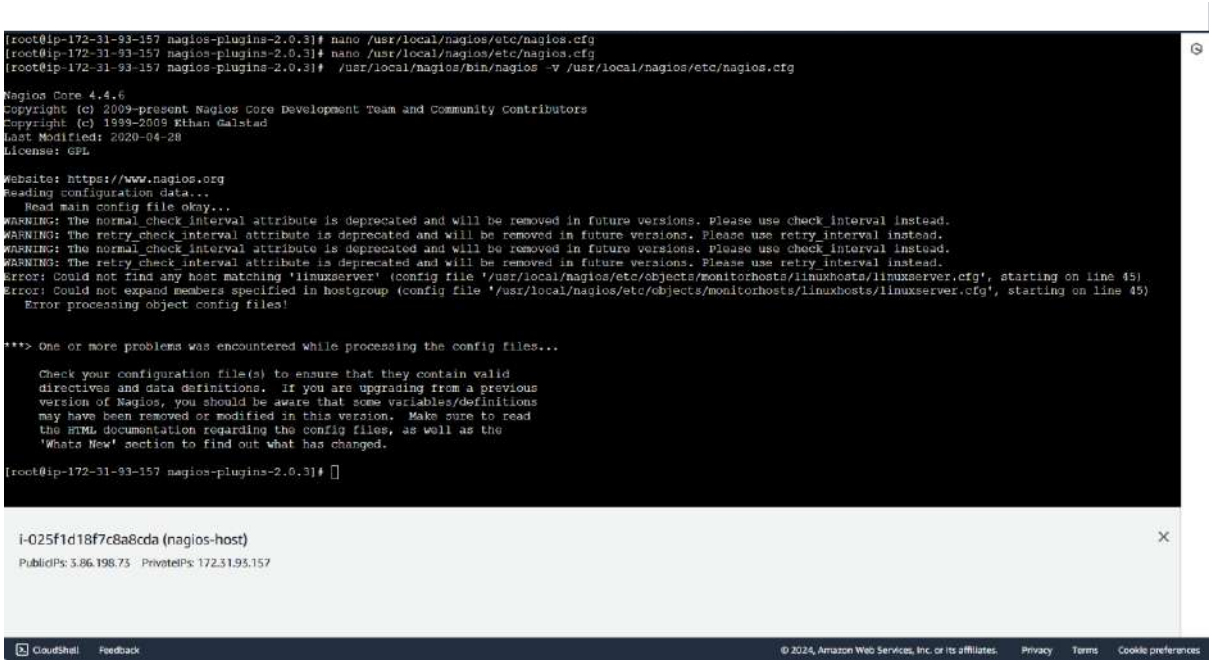
Step 6: Open Nagios config file and add the following line -
nano /usr/local/nagios/etc/nagios.cfg

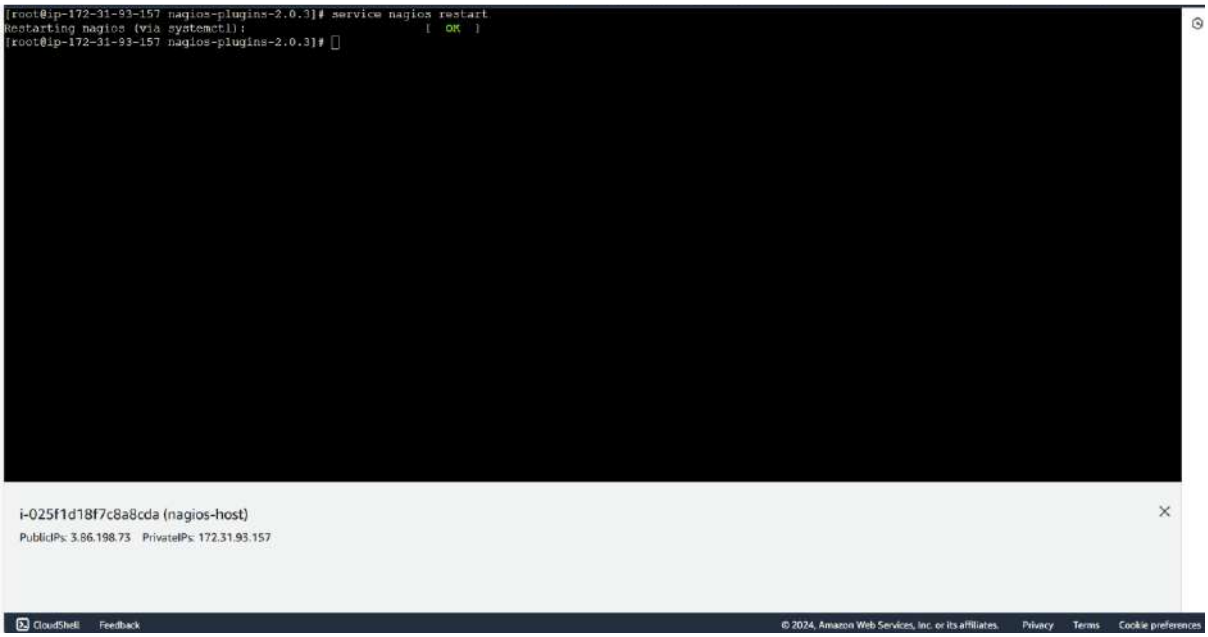
Then add this line -
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



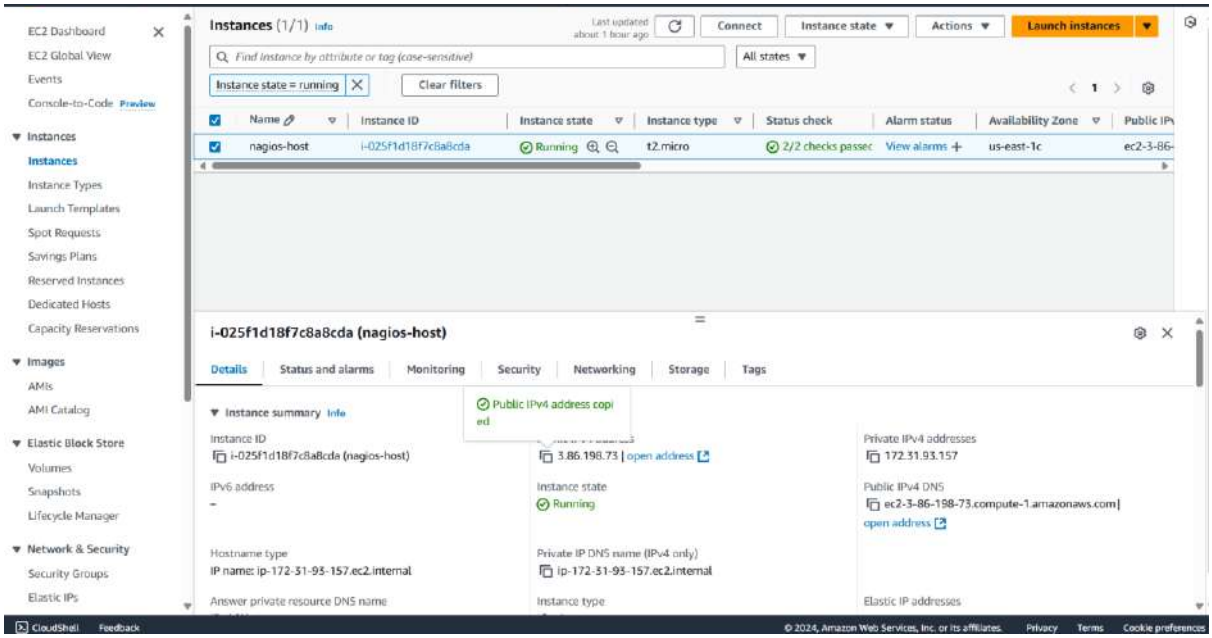
Step 8: Verify configuration files using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

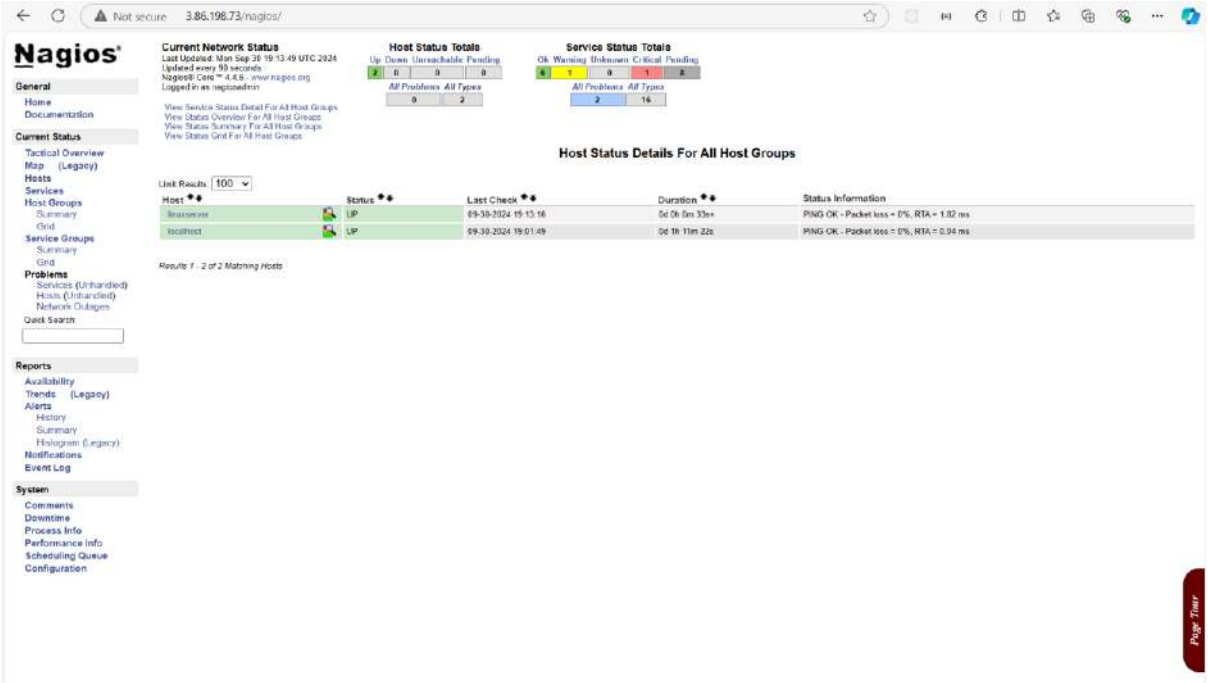
If there are no errors, run the following command -
sudo service nagios start



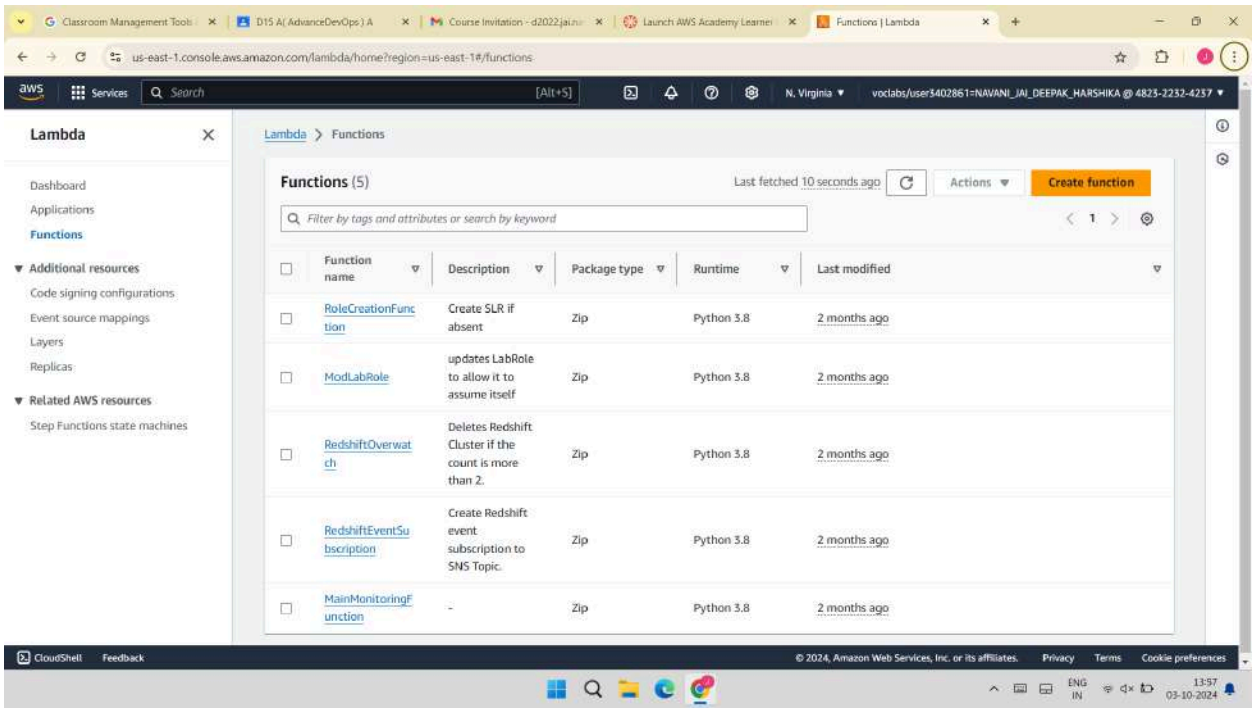


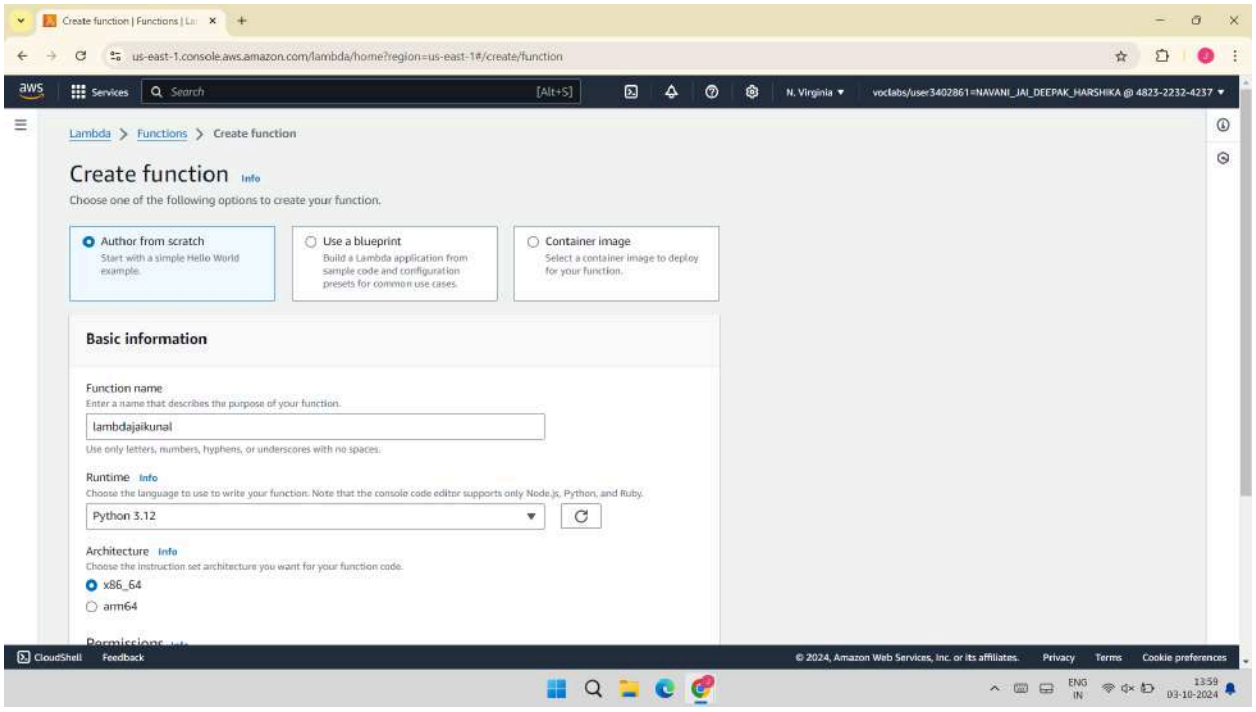
Step 9: After entering the correct credentials, you will see this page.

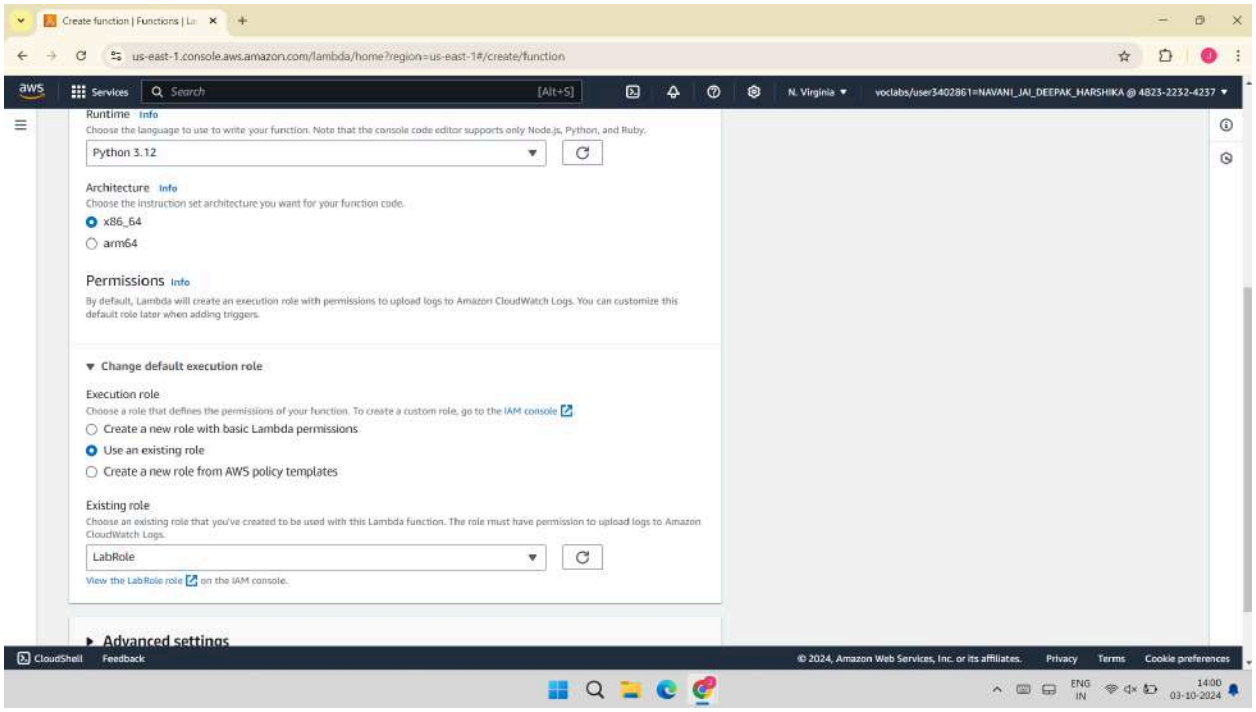


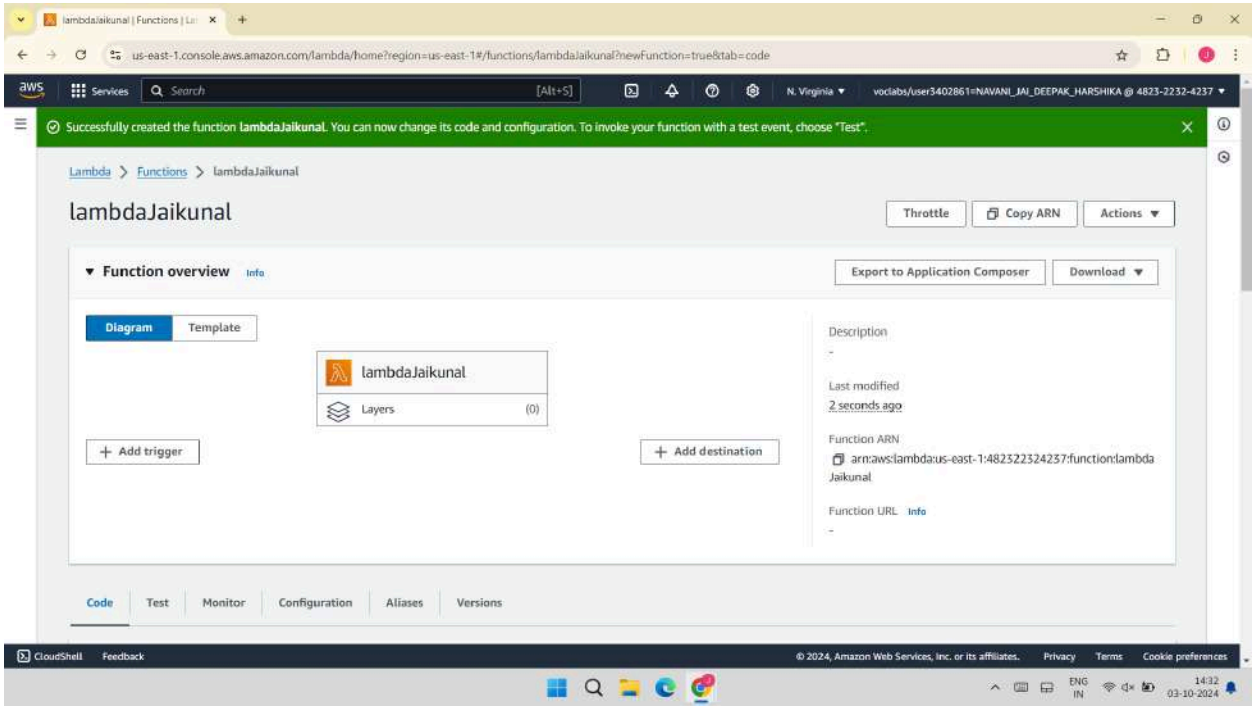

Not secure 3.86.198.73/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Current Network Status
Last Updated: Mon Sep 30 19:13:49 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
6 1 0 1 8
All Problems All Types
2 16
Host Status Details For All Host Groups
Limit Results: 100
Host
Status
Last Check
Duration
Status Information
Bruceserver
UP
09-30-2024 15:15:16
0d 0h 0m 33s+
PING OK - Packet loss = 0%, RTA = 1.82 ms
localhost
UP
09-30-2024 19:01:49
0d 1h 11m 22s
PING OK - Packet loss = 0%, RTA = 0.04 ms
Results 1 - 2 of 2 Matching Hosts
Page Tour

Advance devops- experiment 11
Kunal Punjabi
D15-A(44)






Create function | Functions | La
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12
Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console
Create a new role with basic Lambda permissions
Use an existing role
Create a new role from AWS policy templates
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role on the IAM console.
Advanced settings
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
14:00
03-10-2024

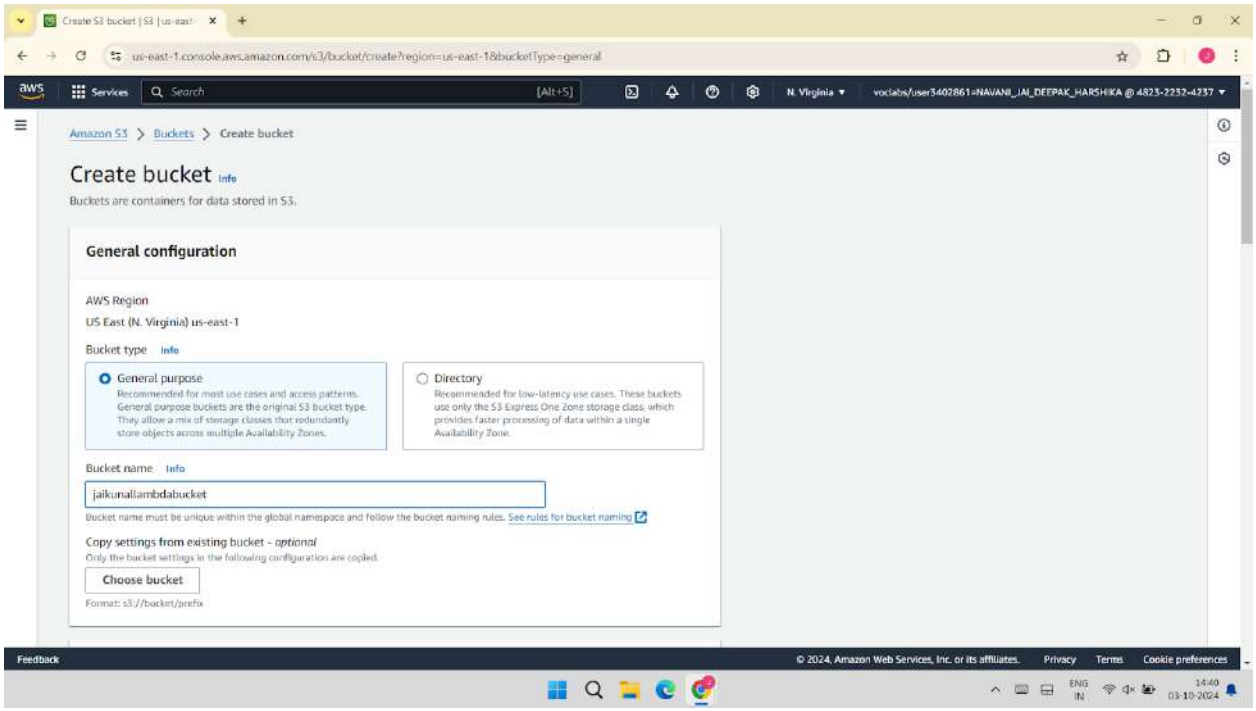

lambdaJaikunal | Functions | La
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/lambdaJaikunal?newFunction=true&tab=code
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237
Successfully created the function lambdaJaikunal. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > lambdaJaikunal
lambdaJaikunal
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
lambdaJaikunal
Layers
(0)
Add trigger
Add destination
Description
-
Last modified
2 seconds ago
Function ARN
arn:aws:lambda:us-east-1:482322324237:function:lambdaJaikunal
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
14:32
03-10-2024

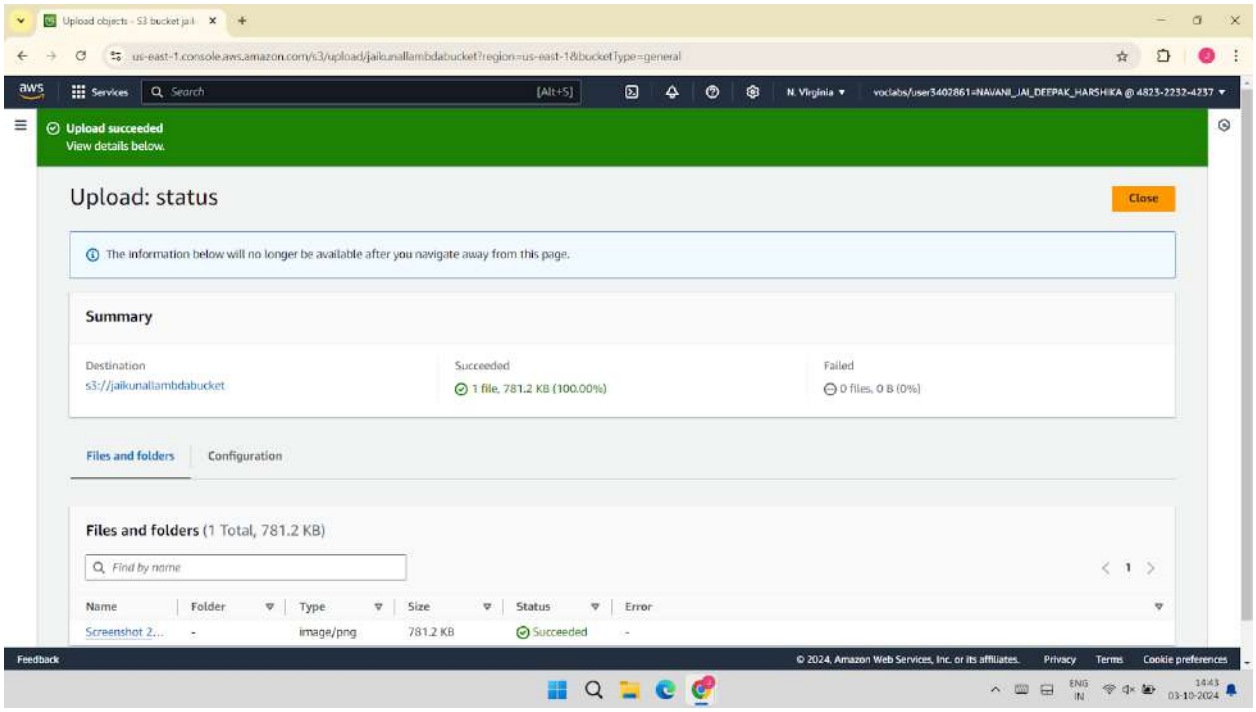
Advanced DevOps Exp-12

Kunal Punjabi
D15A 44

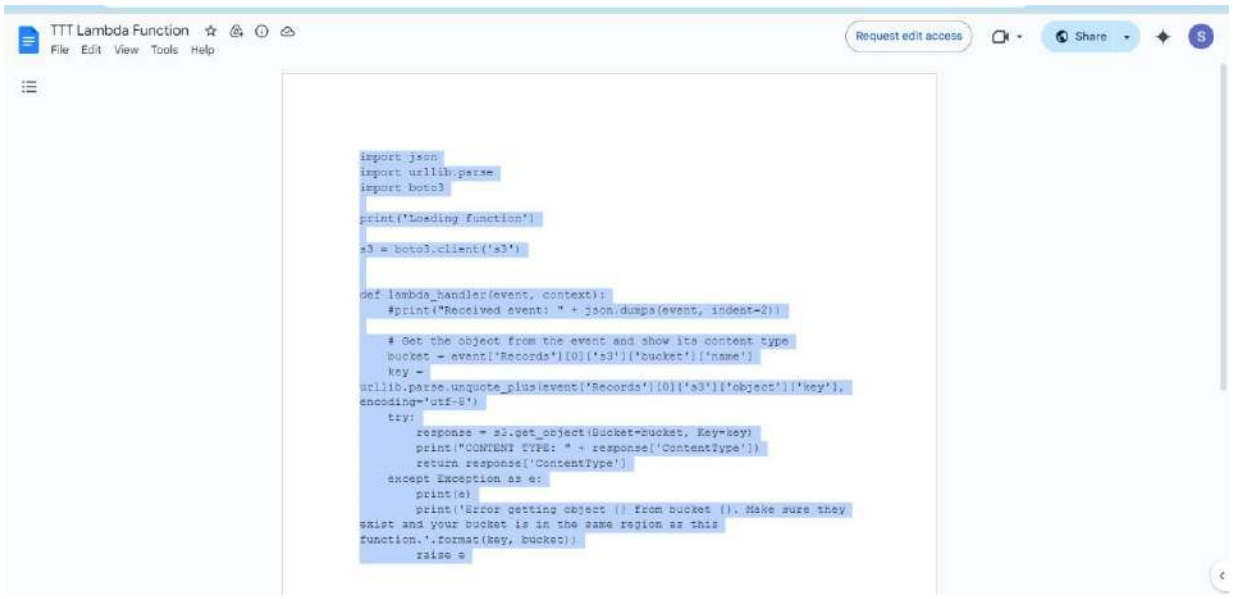
Procedure:-

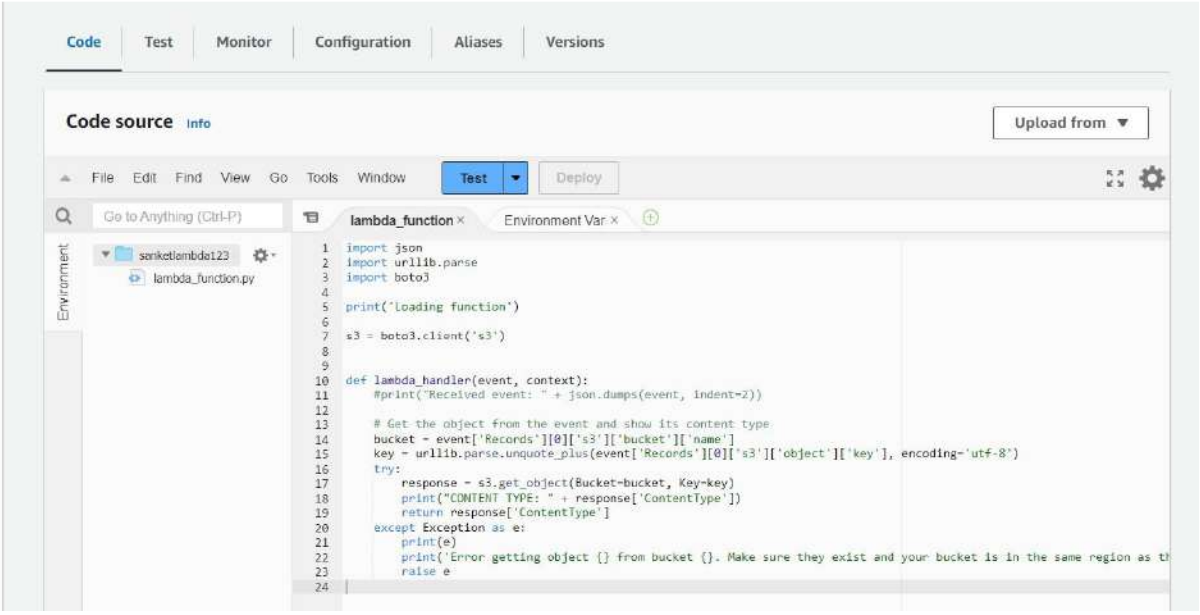
1. Create an S3 bucket of the same location as that of the Lambda function



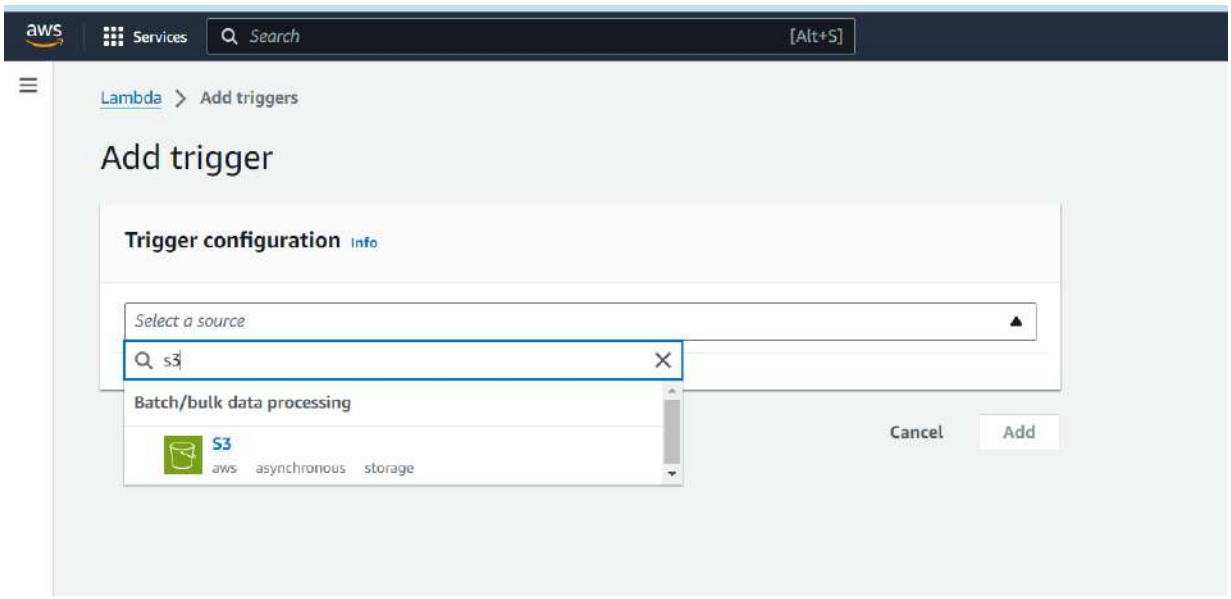

Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://jaikunallambdabucket
Succeeded
1 file, 781.2 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 781.2 KB)
Find by name
Name
Folder
Type
Size
Status
Error
Screenshot 2...
-
image/png
781.2 KB
Succeeded
-
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

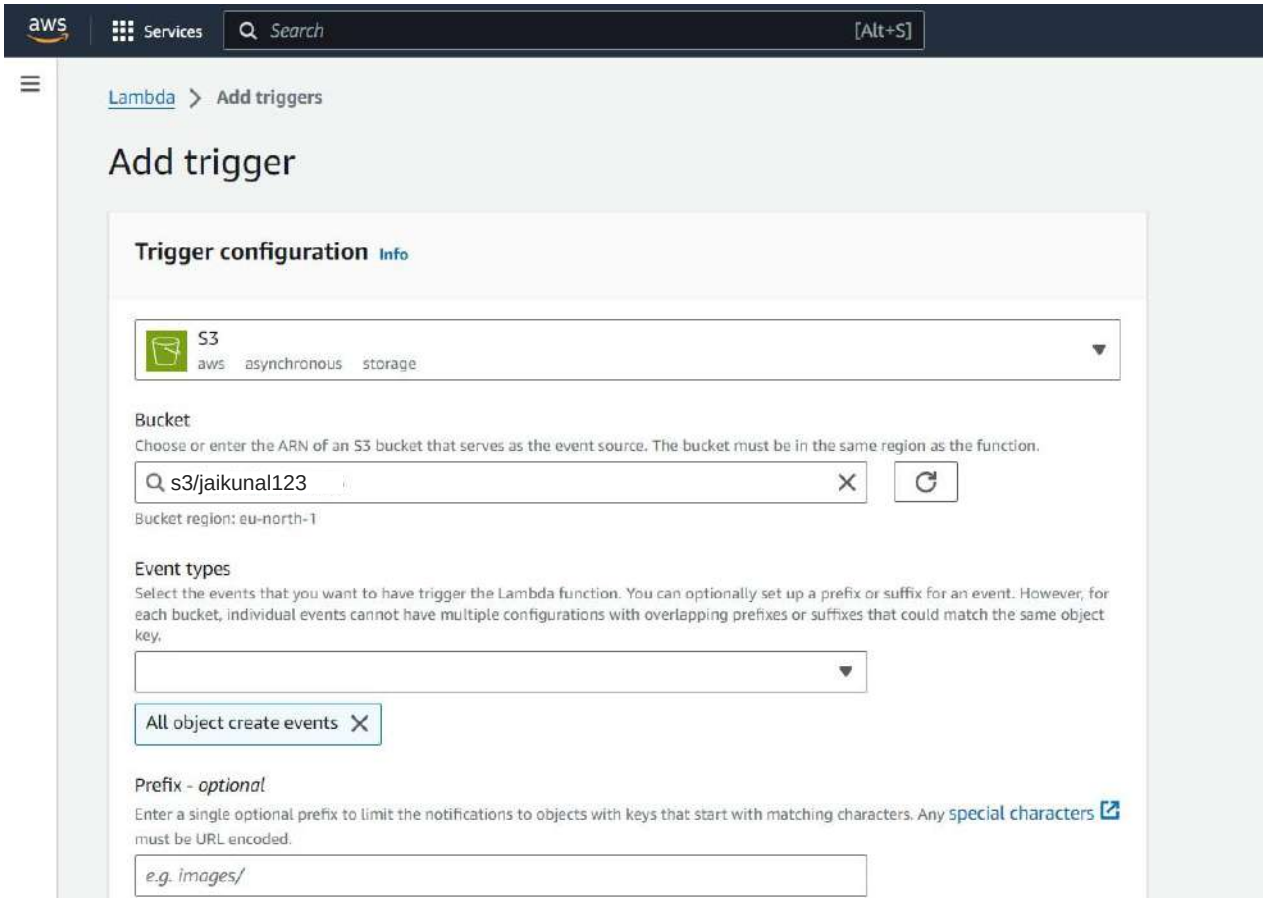
2. After creating the Lambda function copy a code available on the internet which allows the Lambda function to access the S3 bucket contents.

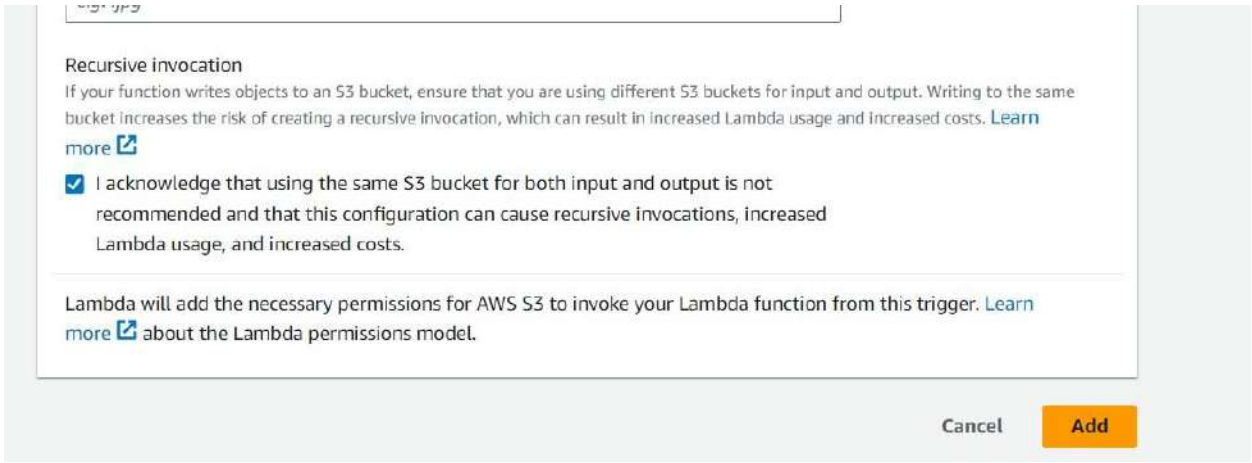




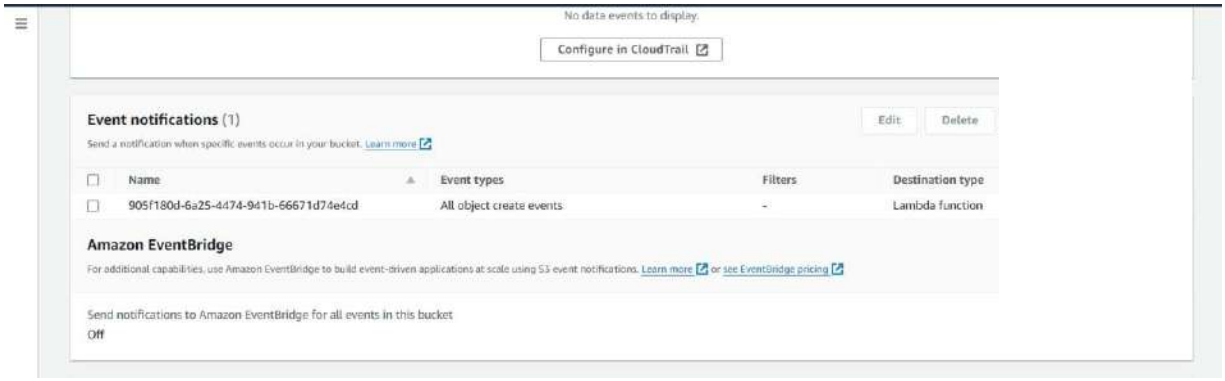
3. Add a trigger to the Lambda function so any changes in the S3 bucket will be first visible to the user.

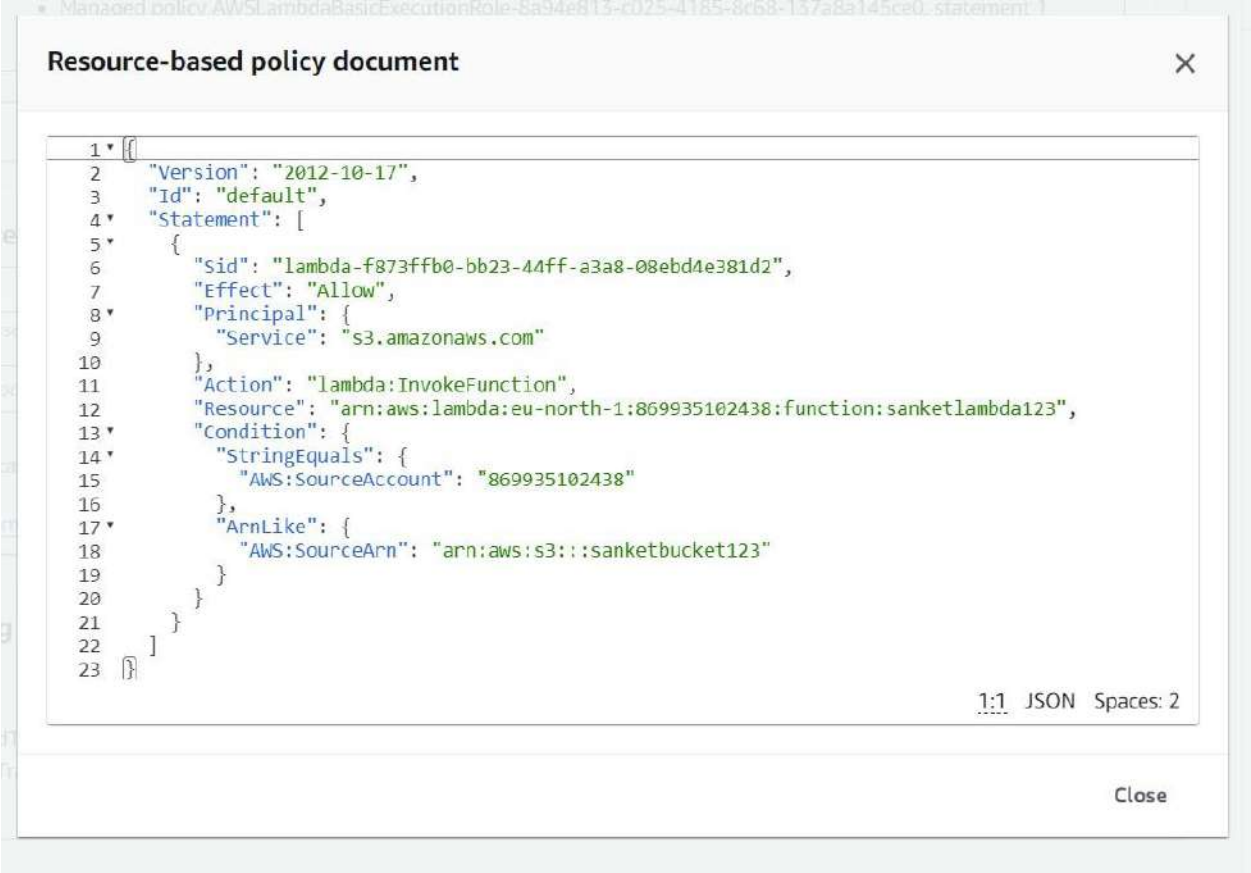



aws
Services
Search
[Alt+S]
Lambda > Add triggers
Add trigger
Trigger configuration Info
S3
aws asynchronous storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/jaikunal123
Bucket region: eu-north-1
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.
e.g. images/

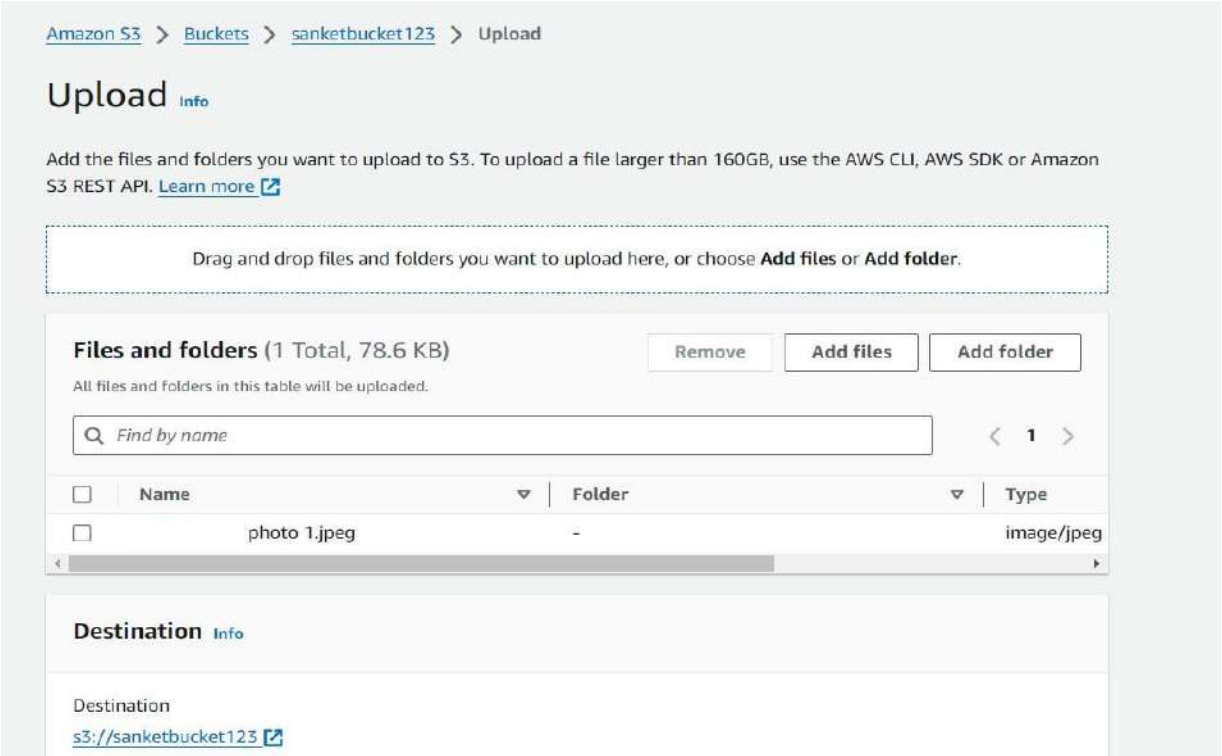

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more
I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.
Cancel
Add

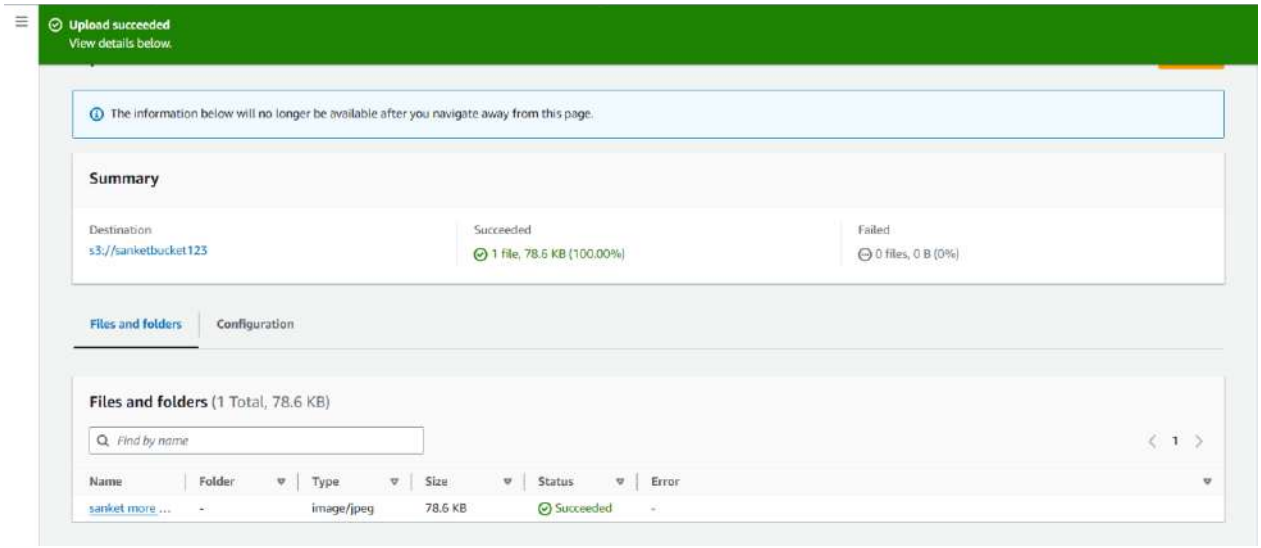
4. In the event notification of the S3 bucket we can see that it has been connected to the Lambda function .



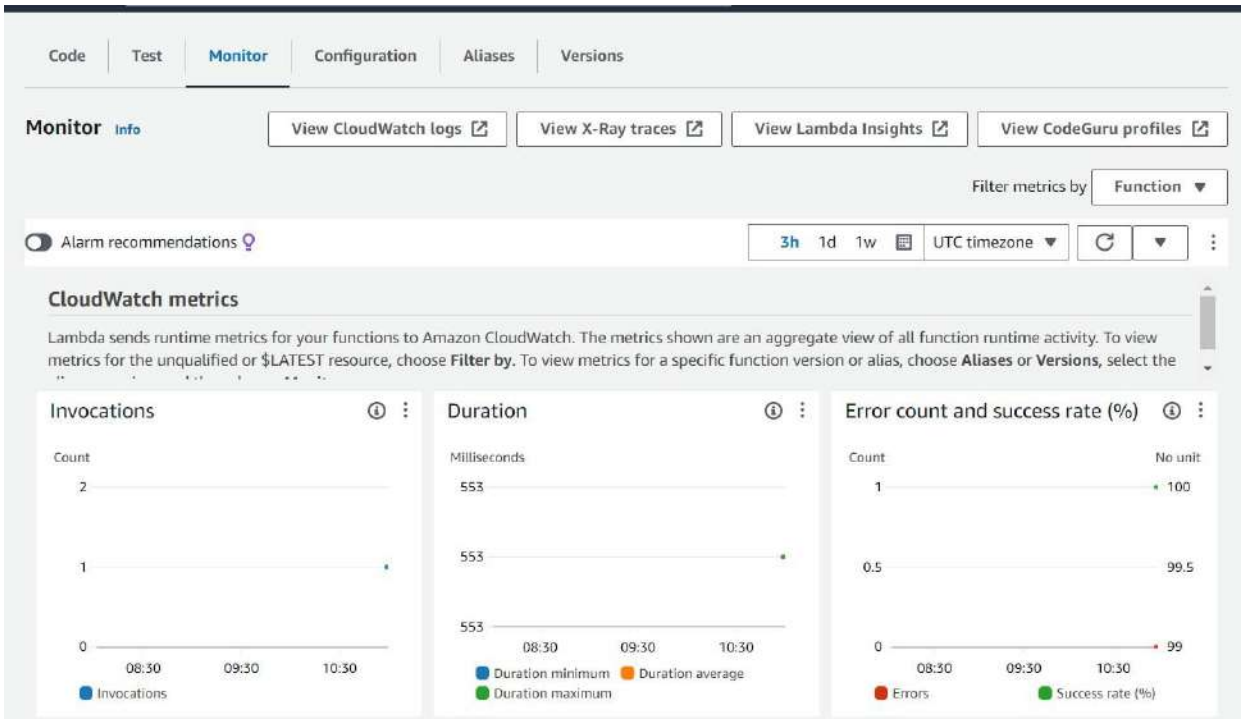


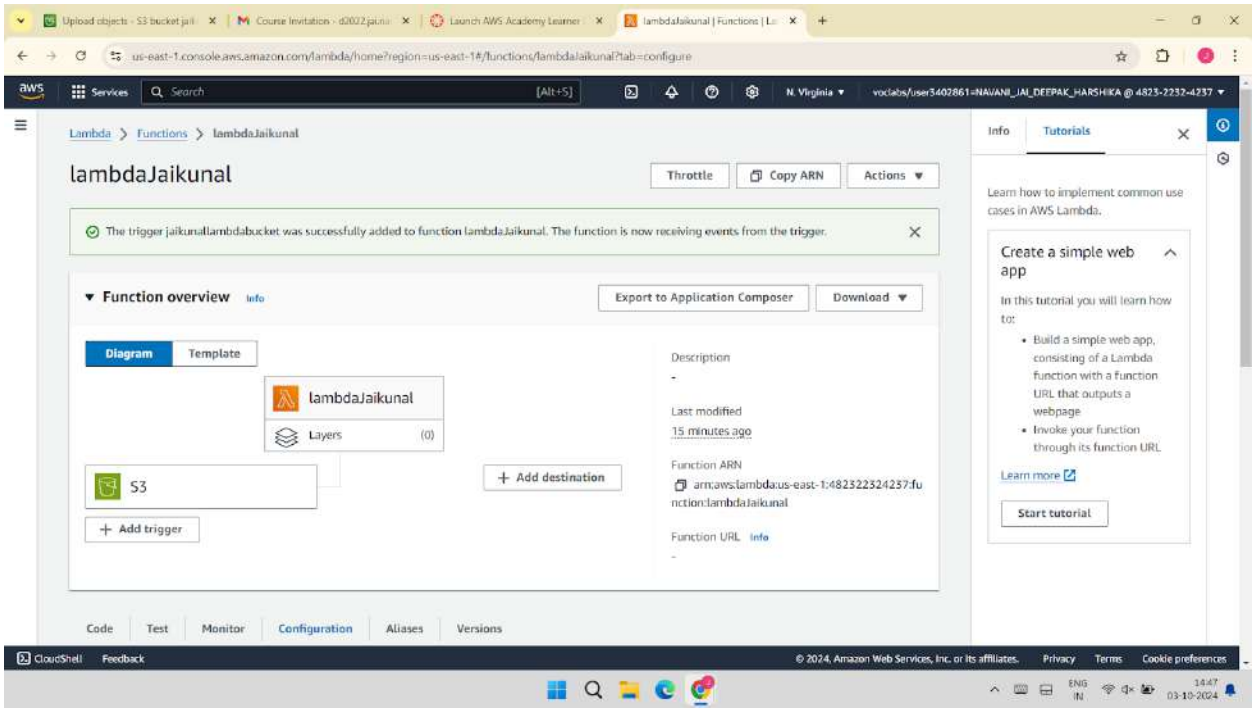
5. Upload a photo to the S3 bucket

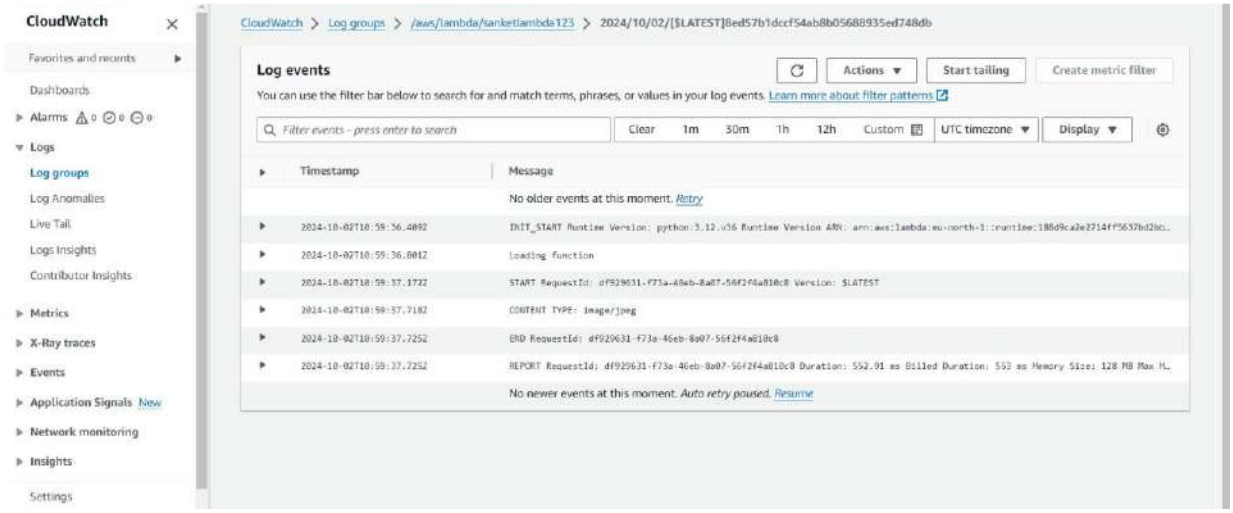




6. Now run the function and in the cloud watch logs of AWS you can see the message printed and all the other details of the working of the Lambda function.




Lambda > Functions > lambdaJaikunal
lambdaJaikunal
Throttle
Copy ARN
Actions
The trigger jaikunallambdabucket was successfully added to function lambdaJaikunal. The function is now receiving events from the trigger.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
lambdaJaikunal
Layers
(0)
S3
Add trigger
Add destination
Description
-
Last modified
15 minutes ago
Function ARN
arn:aws:lambda:us-east-1:482322324237:function:lambdaJaikunal
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


CloudWatch
Favorites and recents
Dashboards
Alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
X-Ray traces
Events
Application Signals New
Network monitoring
Insights
Settings
CloudWatch > Log groups > /aws/lambda/sanketlambda123 > 2024/10/02/[$LATEST]8ed57b1dccf54ab8b05688935ed748db
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
2024-10-02T10:59:36.489Z
INIT_START Runtime Version: python:3.12.v56 Runtime Version ARN: arn:aws:lambda:eu-north-1::runtime:186d9ca2e2714ff5637bd2bc...
2024-10-02T10:59:36.801Z
Loading function
2024-10-02T10:59:37.172Z
START RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Version: $LATEST
2024-10-02T10:59:37.718Z
CONTENT TYPE: image/jpeg
2024-10-02T10:59:37.725Z
END RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8
2024-10-02T10:59:37.725Z
REPORT RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Duration: 552.91 ms Billed Duration: 553 ms Memory Size: 128 MB Max M...
No newer events at this moment. Auto retry paused. Resume

Kunal Mahesh Punjabi
D15A
Roll No: 44

Adv DevOps: Assignment No1

04/05

Q:1 Use S3 bucket and host video Streaming.

-> To Host video streaming using S3 bucket on AWS, you can follow these steps.

Steps:

1) Login to AWS console
- Go to AWS management console
- Enter your login credentials.

2) Creat an S3 bucket.
- In the Console, Search for S3 in the search bar and select S3 from the reesult.
- Click creat bucket.
- Give your bucket a unique name (like "My-video-Streaming-~~video~~ bucket").
- Choose a region (closer to your audience).
- Scroll down and uncheck Block all public acces
- Confirm by checking the acknowledgement box.
- Click creat bucket.

3) Upload your video file to S3.
- Click on your newly created bucket.
- Click the upload button.
- Add your video file from your computer
- Click upload to Start the upload process.

4) Set Permissions for Public Access.

- once the video file is uploaded
- select your video file in the S3 bucket.
- click the Actions dropdown and choose make public
- confirm the action by clicking make public again.

5) Get the video URL

- After making the file Public, click on the video file.
- you will see a URL for the video under object URL. This is direct link to your video.
- Copy this URL

Conclusion: Above method for bucket policy does not work
I ~~For~~ Personally used this method:-

- Go to permission tab of your bucket.
- Click on the permissions tab for my bucket.
- Edit bucket Policy
- Add the following Policy.

```
json
{
"Version": "2012-10-17",
"Statement": [ {
"Sid": "Public ReadGetObject",
"Effect": "Allow",
"Principal": "*",
"Action": "S3:GetObject",
"Resources": "arn:aws:::my-video-streaming bucket/*" }
]
}
```

Testing Public Access → final Step: After making ur ~~video~~ video public.
Paste the URL into web Browser to see video plays.

- Personalized Recommendations: Hotstar uses AWS's machine learning tools to suggest content to users, making their viewing experience more enjoyable.

Key AWS Services used by BMW and Hotstar.

- Compute : Amazon EC2, Amazon Lambda
- Storage : Amazon S3, Amazon EFS.
- Database : Amazon RDS, Amazon DynamoDB.
- Networking : Amazon VPC, AWS Direct Connect.
- Analytics : Amazon Sagemaker, Amazon Rekognition.

Both BMW and Hotstar show how AWS can be a valuable resource for different types of business. By using AWS, they have been able to innovate, grow and provide great experiences for their customers.

Q.3 Why Kubernetes and advantages and disadvantages of Kubernetes. Explain How adidas uses Kubernetes.

Answer:

Kubernetes is popular because it simplifies the management of containerized application. It automates tasks such as deployment, scaling and monitoring making it easier for organizations to manage their applications in a cloud environment.

Advantages of Kubernetes.

1) Portability
2) Scalibility
3) Reliability
4) efficiency.

Disadvantage of Kubernetes.

1) Complexity
2) Steep learning curve
3) Resource intensive
4) management overhead.

Sp How adidas uses Kubernetes.

Adidas has adopted Kubernetes to enhance its IT infrastructure and improve its response to market needs.

1) faster Application Development.
2) operational Efficiency.

3) Scalabity for Demand.
4) Encouring Innovation.

Q.4 What are Nagios and Explain How Nagios are used in E-Services.

-) Nagios is an open-source monitoring tool that helps organizations keep track of their IT infrastructure including. It provides a way to ensure that systems are running smoothly and alerts users if any issues arise.

Key feature of nagios:
-) Monitoring
Alerts
Reporting.

How Nagios is used in E-Services.
Nagios plays a vital role in the operation of eservices by ensuring the online systems are reliable and efficient.

1) Infrastructure Monitoring
2) Service Availability
3) Performance management.
4) Incident management.
5) User Experience Monitoring.

Kunal Punjabi
DISA
44

Adv DevOps - Assignment 2

04/05

Create a Rest API with Serverless framework.

-> creating Rest APIs with Serverless framework is an efficient way to deploy Serverless application that can scale automatically without managing servers

(i) Serverless framework:
A powerful tool that deployment of Services and Serverless application across various cloud providers such as AWS, Azure and Google Cloud.

ii) Serverless architecture: This design model allows developers to build application without worrying about underlying infrastructure enabling focus on code and business logic.

iii) Rest API: Representation State Transfer is architecture style for designing application.

Steps for creating REST API for Serverless framework.

1) Install Serverless framework:
You start by installing Serverless framework globally using Node Package Manager (NPM) this allow you to manage Serverless application directly from your terminal.

Creating a Node.js Serverless project.
A directory is created for your project where you initialize a Serverless Service. using the command Serverless create you set up a template for AWS Node.js microservices that will automatically deploy to AWS Lambda.

Project Structure: The project creates essential files like handler.js (which contains) code for Lambda functions and Serverless.yml.

Create a Rest API Resource.

5) Deploy the Service:
with the sls deploy commands Serverless framework packages your application uploads, ~~at~~ necessary resources on AWS and Setup the infrastructure.

6) Testing the API:
Once deployed you can test REST API using tools like Cu or Postman by making post request to generated API

7) Storing data in Dynamo DB: To store Submitted Candidate data you integrate AWS DynamoDB as a data

8) Adding more functionalities like list all Candidates, get Candidate by ID

a) AWS IAM Permission
You need to ensure that Serverless framework is given rig Permission to interact with AWS resources like Dynamodb.

10) Monitoring and maintance.
After Deployment Serverless framework provides Service information like Deployed endpoints, API Key, log Structure.

case Study for SonaQube.

-> SonarQube is an open Sourc platform used for continuous inspection of quality to detect bugs, code smells and security vulnerabilities in Projects across various programming languages.

1) Profile creation in SonarQube.

Quality profiles in SonarQube are essential configuration that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'Sonar way' profile comes built in for all languages. Custom profiles can be created by copying or extending existing ones. Copying creates an independent profile, while extending existing ones. Copying: creates an independent profile, while extending inherit rules from parent profile and reflects future changes automatically You can activate or deactivate rules prioritize certain rules to Specific Projects. Permissions to manage quality profile are restricted. To ensure profiles include new rules its important to check against updated built in profiles or the Sonarqube rules page.

2) using Sonar Cloud to analyze GitHub code:

Sonar Cloud is Cloud-based counterpart of SonarQube that integrates directly with GitHub, Bitbucket, Azure and Gitlab repositories To get Started with Sonar-Cloud via GitHub Signup via Sonar Cloud product page and connect your GitHub Organization or personal account once connected, Sonar Cloud mirrors your GitHub setup with each project corresponding to GitHub repository. After setting up the organization Choose Subscription plan Next, import repositories into your SonarCloud Organization where each

repo becomes a Sonar Cloud proje. Defin 'new cod' o
focus on recent changes and choose between automat
analysis or CI-based analysis Automat analysis happ
directly in Sonar Cloud, while CI based analysis inte
with your build proces once the analysis is comple
result can be viewed in both Sonar-Cloud and
GitHub including Security impor issue.

3) Sonarlint in JAVA IDE:
Sonarlint is an IDE that peforms on-the fly code
analysis on your writ code. It helps developers dete
bugs, Security vulnerabilites and Code Smells direct
in the development environment Such as INTELLIJ idea
Eclipse. To Set it up install the Sonarlint Plugin,
Configure the connection with Sonar Qube or Sonar Cloud
and Select the Proje profile to analyze Java Code.
This approach ensures immediate feedback on code
quality, Promoting clean and maintainable code
from beginning.

4) Analyzing Python Projects with SonarQube:
SonarQube Supports python test Coverage, reporting but
requires third Party tool like Coverage.Py to generate
Coverage report in XML format. The build Process c
also be automated using GitHub Actions, which inst
dependencies, runs, tests and invoke SonarQube Scan.
Ensure report in Cobertura XML format and Place
where Scanner can access it.

Analyzing Node.js Projects with SonarQube

For Node.js projects SonarQube can analyze JavaScript and TypeScript code. Similar to the Python setup, you can configure SonarQube to analyze Node.js projects by installing the appropriate plugin and using Sonar Scanner to scan the project. SonarQube will check the code against industry standard rules and best practices, flagging issues related to security vulnerabilities, bugs and performance optimization.

At a large organization your centralized operation team may get many repetitive infrastructure requests. You can use Terraform to build a Self Service infrastructure model that lets product team manage their own infrastructure independently. You can create and use Terraform modules that code the standards for deploying Terraform Cloud can also integrate with ticketing System like Service now to automatically generate new infrastructure request.

Implementing a Self Service infrastructure model using Terraform can transform How large organizations manage their infrastructure independently. Organizations can enhance efficiency reduce bottlenecks and complying with established needs.

The need for Self-Service infrastructure: In large organizations operating teams often face an overwhelming number of repetitive requests this can lead to delay in Service delivery and move quickly.

A SaaS service model allows teams to provision and manage their infrastructure without or operations te... for every request.

Benefits of using Terraform.

1) modularity and Reusability.
Terraform modules encapsulate standard configuration for various infrastructure components networks databases, compute resources
Teams can reuse these modules across different projects reducing redundancy and minimizing the risk of errors.

2) Standardizations:
By defining best practices within modules organizations can ensure that all deployments comply with required policies and standards
This consistency helps maintain security and operational integrity across the organization.

4) Integration with existing System Terraform Cloud can integrate with existing systems like ServiceNow. to automate the generation their existing platform, reducing manual intervention.

Implementation Steps

1. Identify infrastructure components Begin by identifying which components of your infrastructure will be managed by Terraform e.g. VPCs, Security group, load balancers.

2. Develop Terraform modules.
Create reusable modules to define the desired configuration and resources.
Ensure each module includes input variables for customization and output for integration with other

Establish Guidelines and Best practices:
Define guidelines for module usage, versioning and documentation to ensure clarity and maintainability.

Testing and validation

Implement a testing framework to validate module functionality before deployment.

Best practices for module management

Utilize the Terraform registry.

Leverage existing community modules from the Terraform Registry to avoid reinventing solutions and Sure to best Practices.

Version control:
Implement versioning for your modules & track chan
Overtime this helps many dependency efficiency o
minimize disruption during updates.

Ensuring collaboration: fosters a culture of collaboration o
Sharing Modules across teams. This promotes consis
in deployments and By adopting a self-ser
infrastructure model with transform organisation
can empower product teams to efficiency no
approach Not only streamlines process but al enhan
agility in responding to changing busines eds.
ultimately it leads to a more responsiv It
environment that supports innovation and growth within th
organization.