ADVANCE DEVOPS EXP-1

Ansh Sarfare D15A/50

Aim: To understand the benefits of Cloud infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and and Perform Collaboration Demonstration.

Theory : EC2 Hosting Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers by providing virtual servers, known as instances, to run applications.

Key Concepts of EC2:

1. Instances: An instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications. Instances can be launched on demand and scaled according to the needs of the application.

2. AMI (Amazon Machine Image): AMIs are pre-configured templates for instances. They include the operating system, application server, and applications themselves, allowing for quick and consistent instance launches

3. Instance Types: EC2 offers a variety of instance types optimized for different use cases, such as compute-optimized, memory-optimized, and storage-optimized instances. Each instance type offers different combinations of CPU, memory, storage, and networking capacity.

AWS Cloud9 Infrastructure

AWS Cloud9 is a cloud-based integrated development environment (IDE) that allows you to write, run, and debug code with just a browser. It supports multiple programming languages, including Python, JavaScript, and more. AWS Cloud9 comes pre-packaged with essential tools and libraries, making it easier for developers to start coding without the need for complex setup processes.

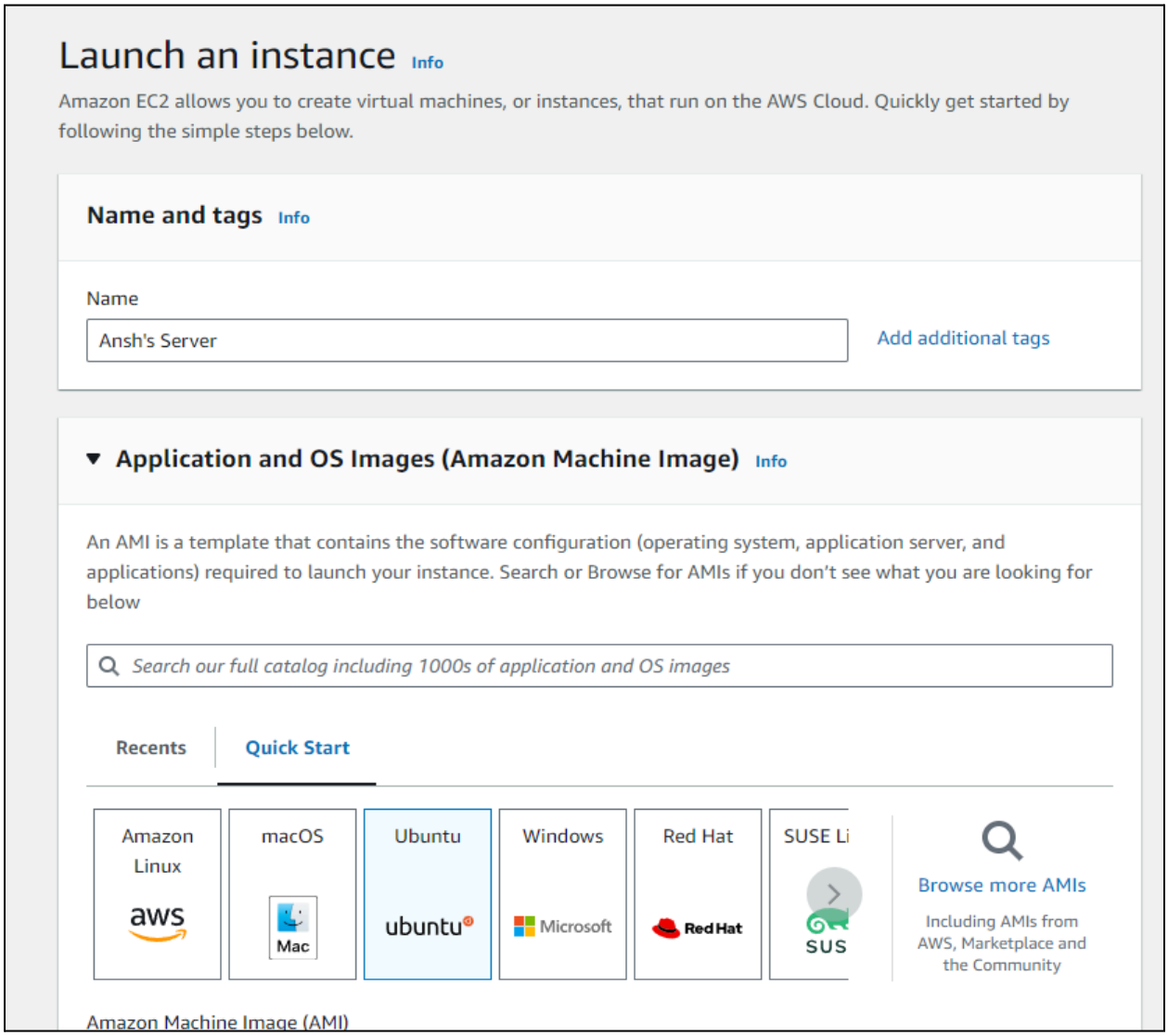
Key Features of AWS Cloud9:

1. Cloud-Based IDE: AWS Cloud9 provides a full-featured development environment accessible through a web browser. This eliminates the need for local IDE installations and configurations.

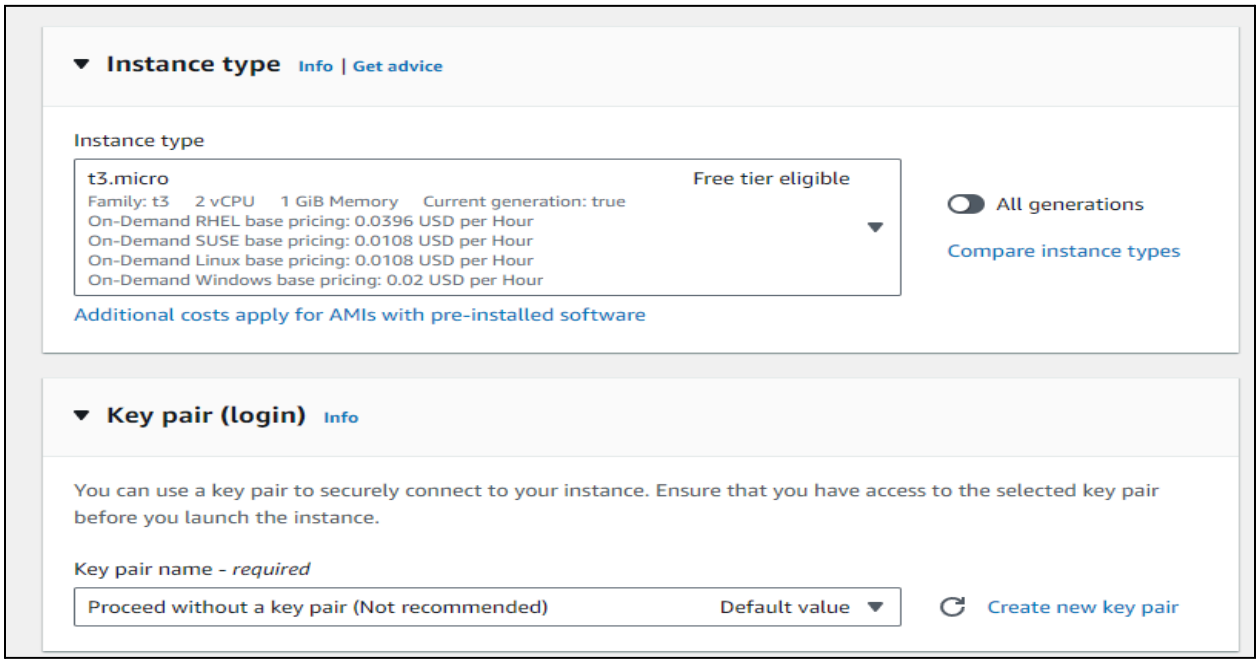
2. Collaborative Development: Developers can collaborate in real-time, with multiple users able to work on the same project simultaneously. It includes features like chat and simultaneous editing, making it ideal for pair programming and team collaborations.

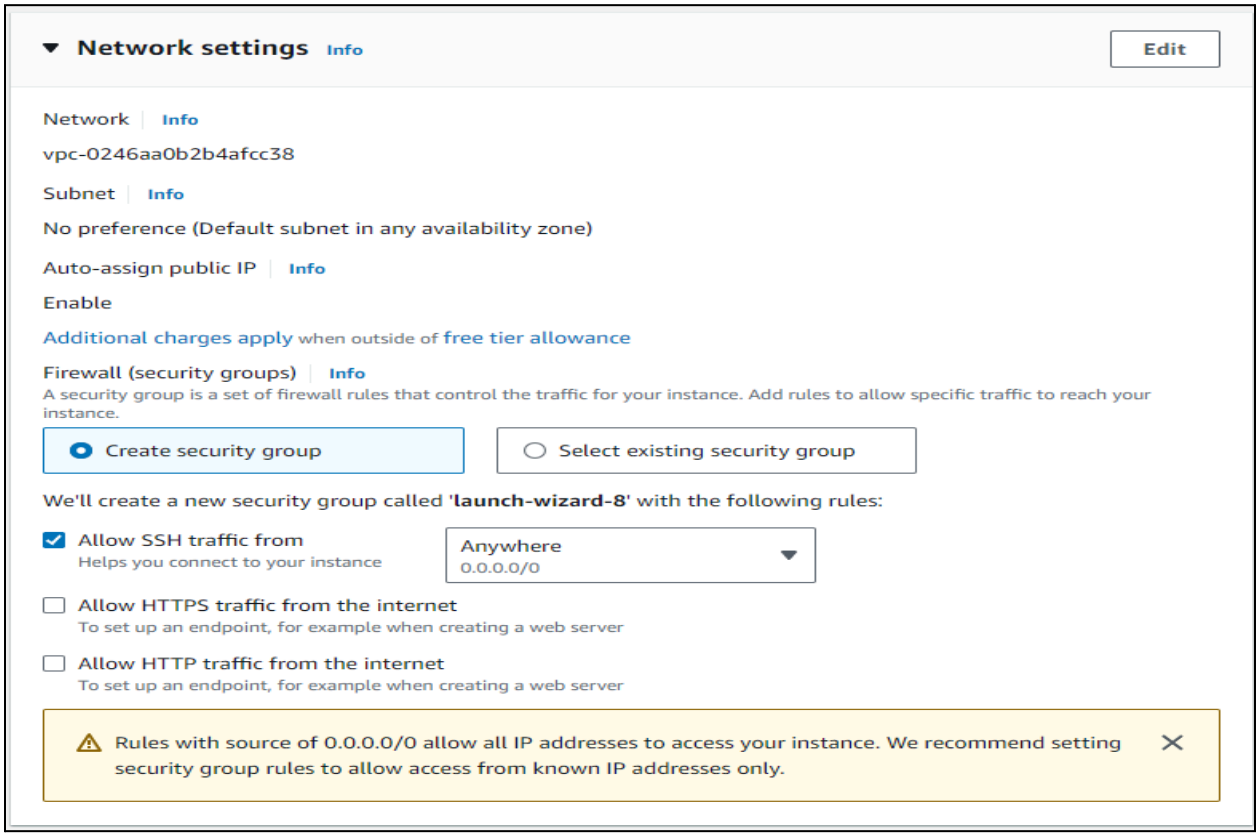
3. Seamless Integration with AWS Services: AWS Cloud9 integrates seamlessly with other AWS services like EC2, S3, and Lambda, allowing developers to easily deploy and manage their applications directly from the IDE.

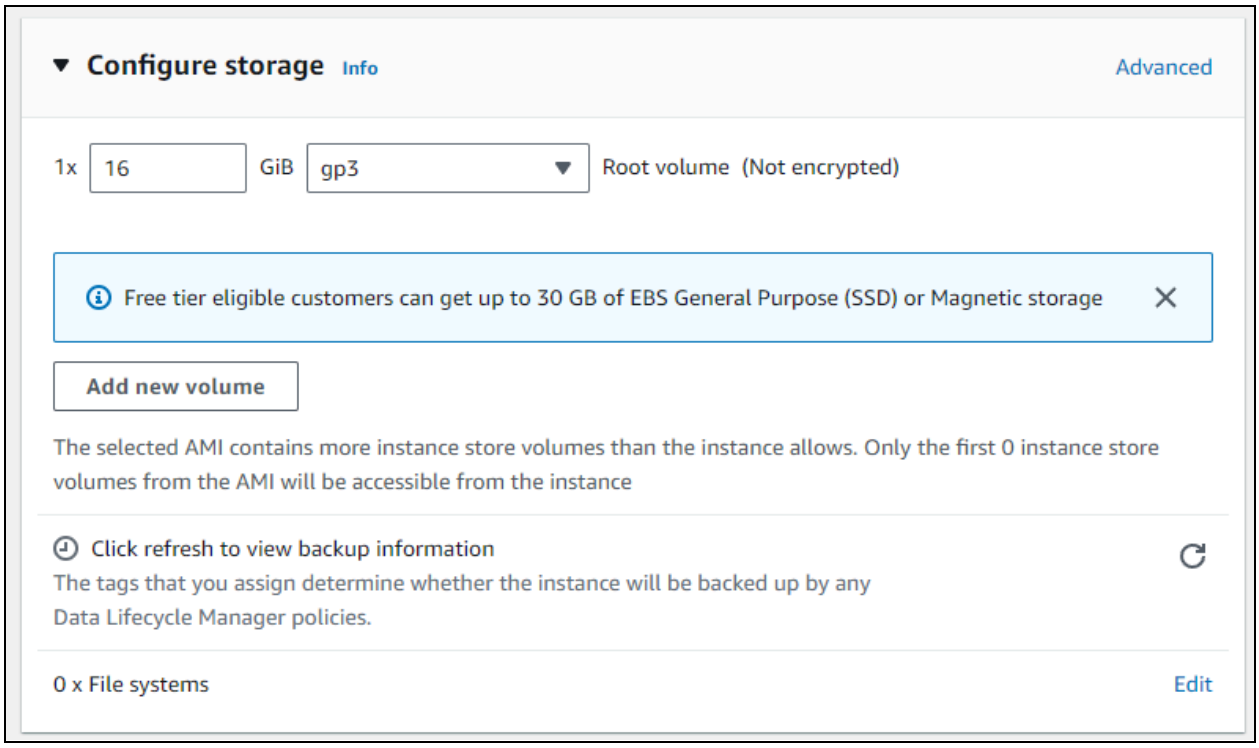
Step-1: Open AWS Academy , launch a instance and select Ubuntu as AMI



Step- 2: Select default configurations



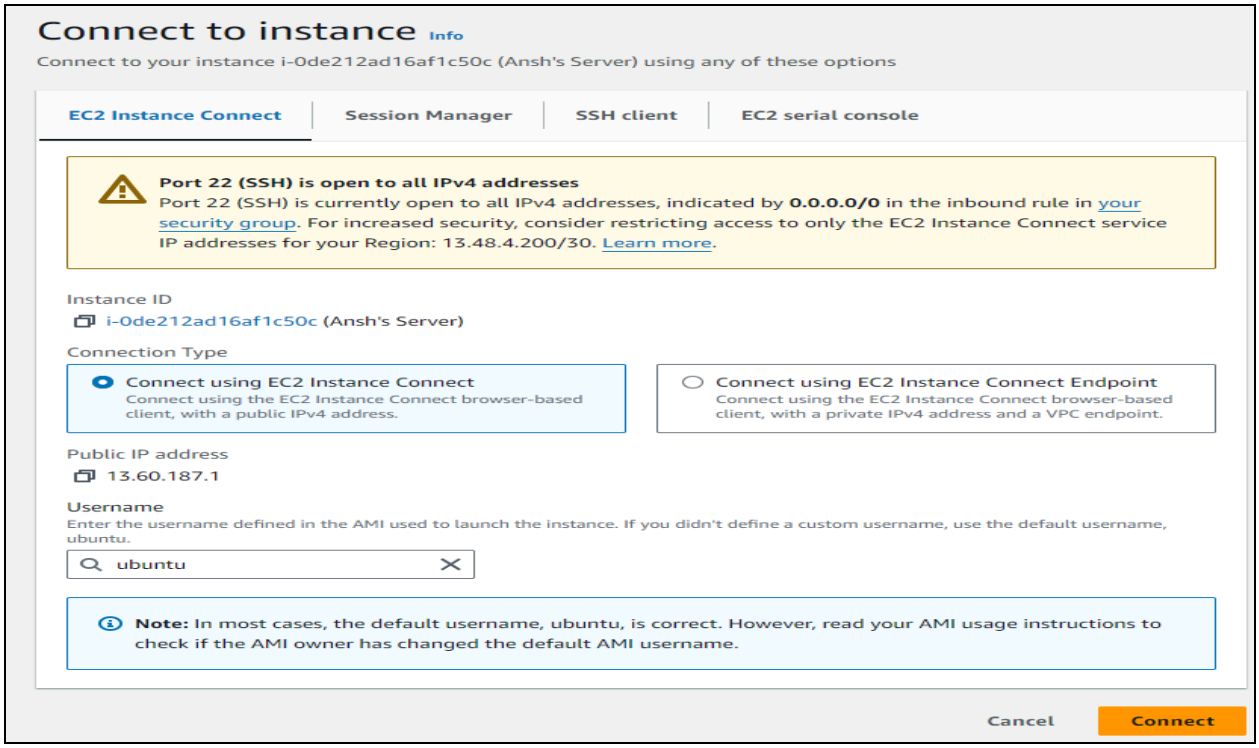


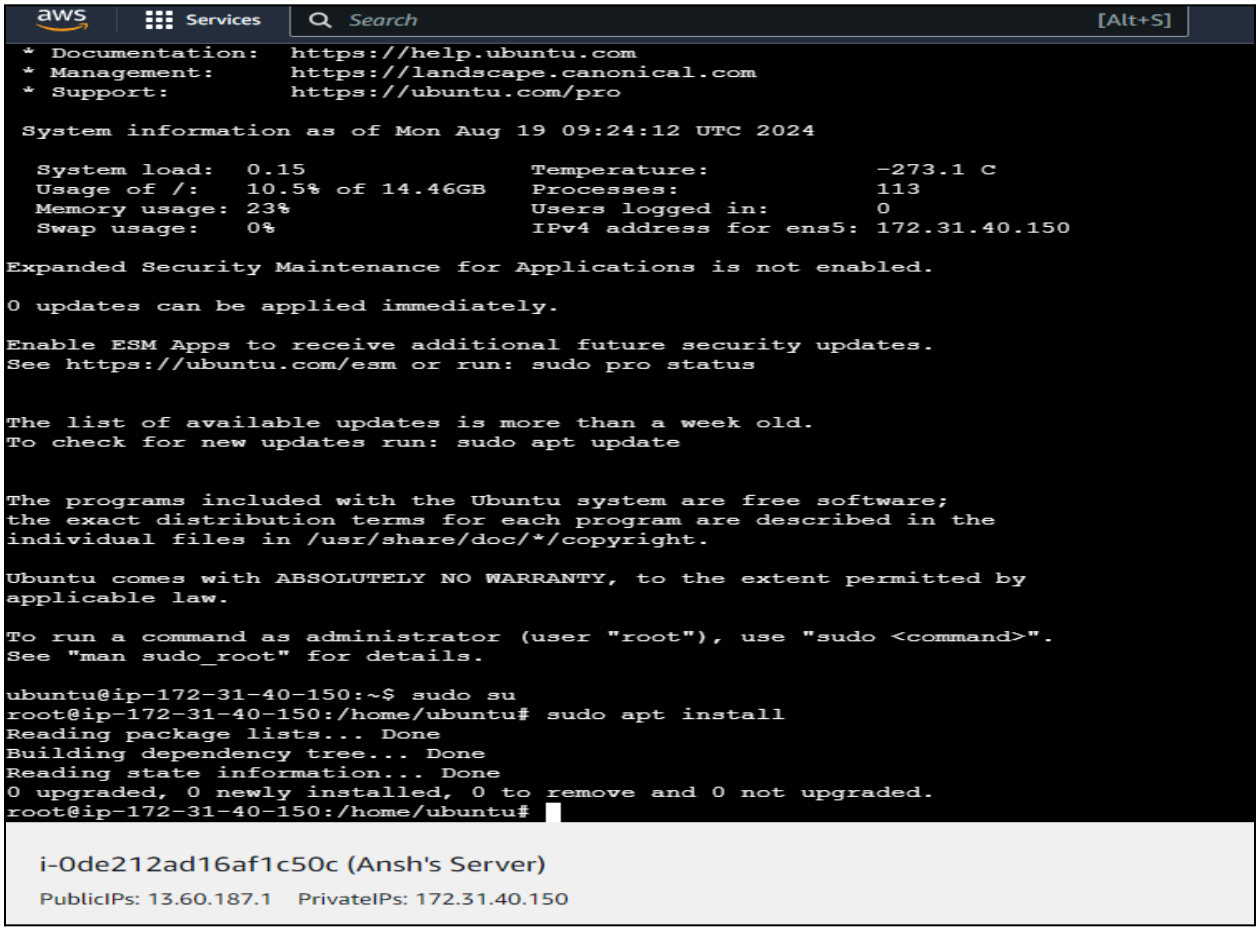


Step-3: Connect to instance and run following commands

Commands :

sudo su
sudo apt install
sudo apt-get update
apt install apache2
systemctl status apache2
cd /var/www/html/


Connect to instance Info
Connect to your instance i-0de212ad16af1c50c (Ansh's Server) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Port 22 (SSH) is open to all IPv4 addresses
Port 22 (SSH) is currently open to all IPv4 addresses, indicated by 0.0.0.0/0 in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.48.4.200/30. Learn more.
Instance ID
i-0de212ad16af1c50c (Ansh's Server)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
13.60.187.1
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect


aws
Services
Search
[Alt+S]
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Mon Aug 19 09:24:12 UTC 2024
System load: 0.15
Usage of /: 10.5% of 14.46GB
Memory usage: 23%
Swap usage: 0%
Temperature: -273.1 C
Processes: 113
Users logged in: 0
IPv4 address for ens5: 172.31.40.150
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-40-150:~$ sudo su
root@ip-172-31-40-150:/home/ubuntu# sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-40-150:/home/ubuntu#
i-0de212ad16af1c50c (Ansh's Server)
PublicIPs: 13.60.187.1 PrivateIPs: 172.31.40.150

```
root@ip-172-31-40-150:/home/ubuntu# sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [348 kB]
Get:14 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [87.9 kB]
Get:15 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5764 B]
Get:16 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [323 kB]
Get:17 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [136 kB]
Get:18 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.8 kB]
Get:20 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [245 kB]
Get:21 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [47.8 kB]
Get:22 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:23 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:24 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:25 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:26 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:27 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:28 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:30 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:31 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:32 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:33 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:34 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:35 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:36 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
```

i-0de212ad16af1c50c (Ansh's Server)

PublicIPs: 13.60.187.1 PrivateIPs: 172.31.40.150

```
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 28.4 MB in 5s (5454 kB/s)
Reading package lists... Done
root@ip-172-31-40-150:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 upgraded, 10 newly installed, 0 to remove and 54 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

i-0de212ad16af1c50c (Ansh's Server)

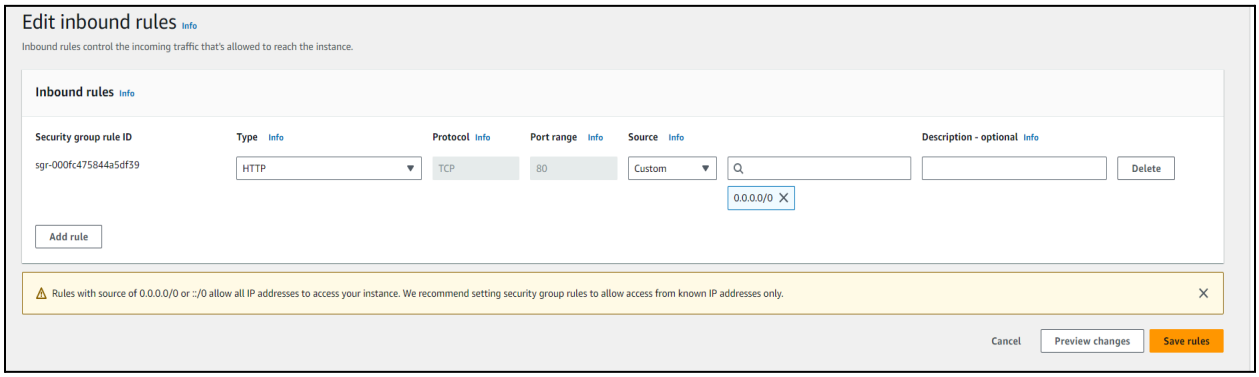
PublicIPs: 13.60.187.1 PrivateIPs: 172.31.40.150

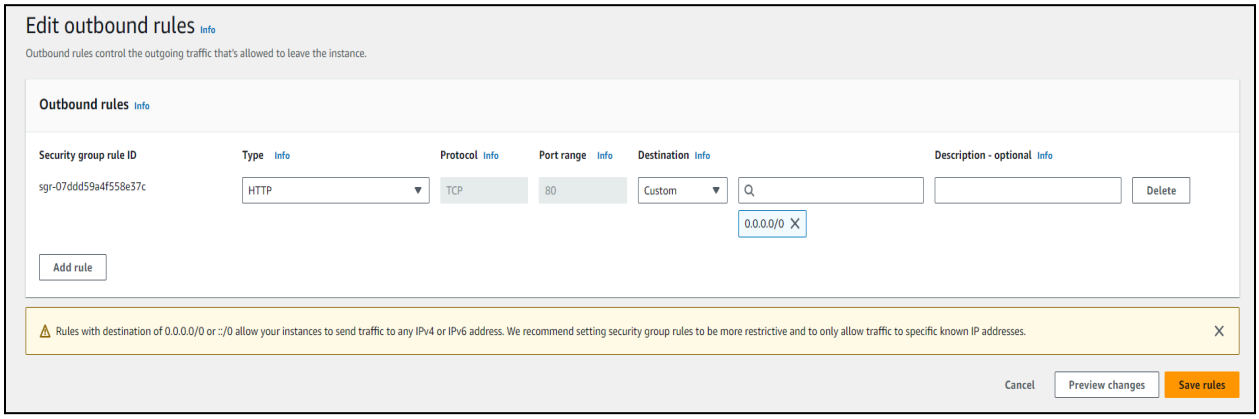
```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-40-150:/home/ubuntu# cd /var/www/html/
root@ip-172-31-40-150:/var/www/html#
```

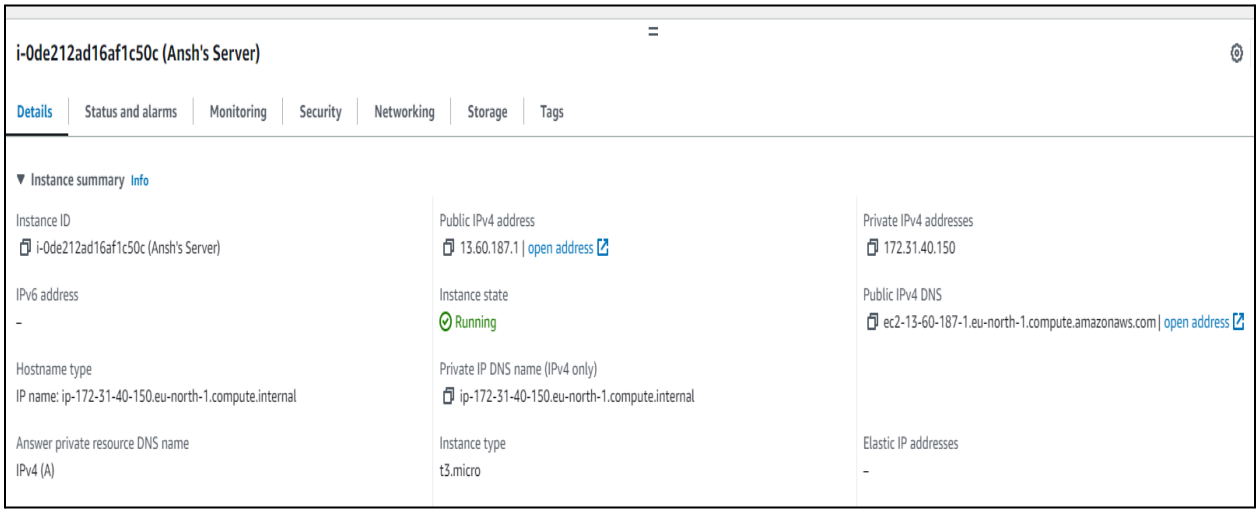
i-0de212ad16af1c50c (Ansh's Server)

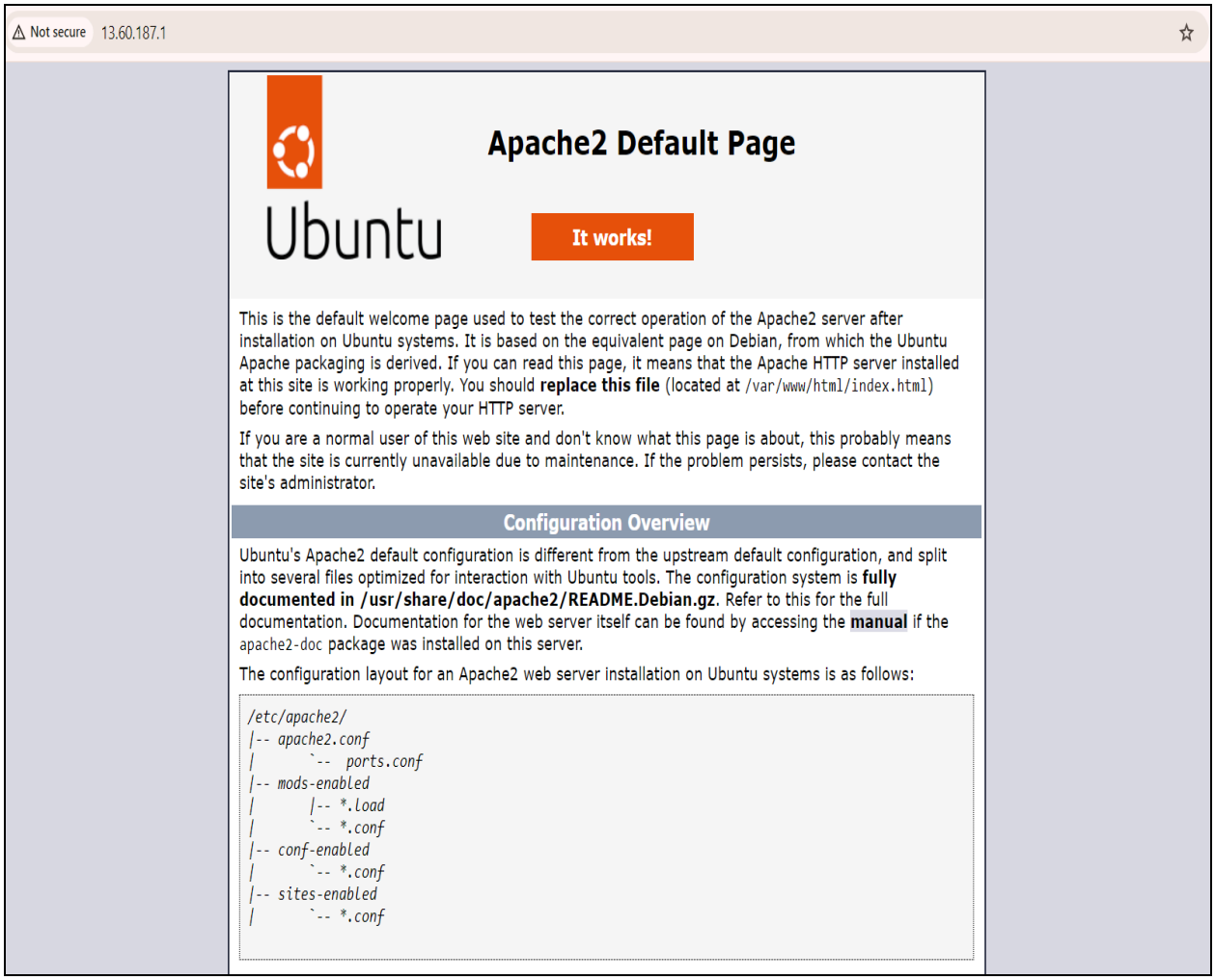
PublicIPs: 13.60.187.1 PrivateIPs: 172.31.40.150

Step-4: Go to Security Groups and edit inbound & outbound rules



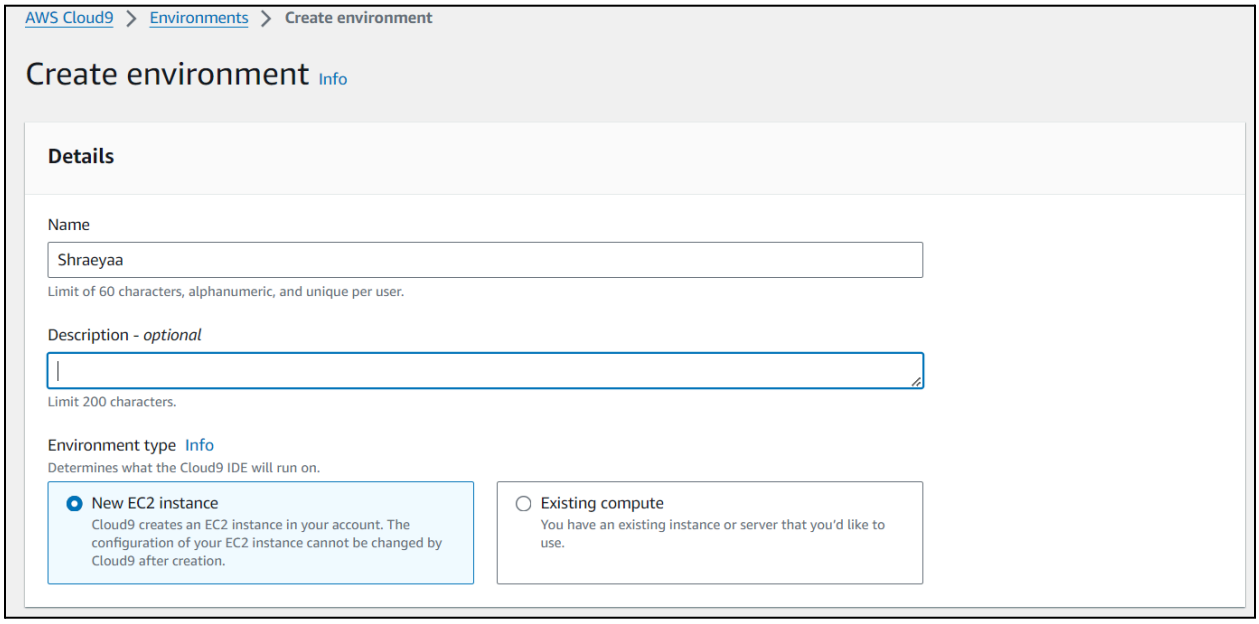




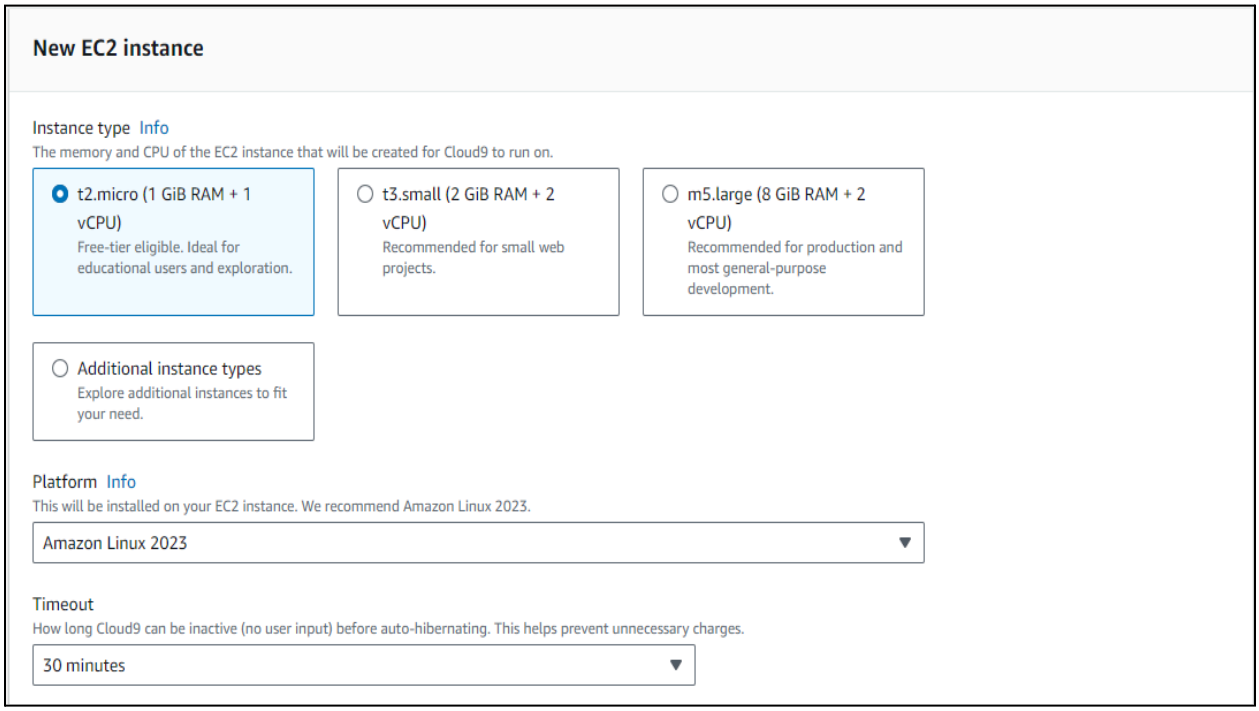

Not secure
13.60.187.1
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read 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
| `-- *.conf
|-- sites-enabled
| `-- *.conf

Cloud9:

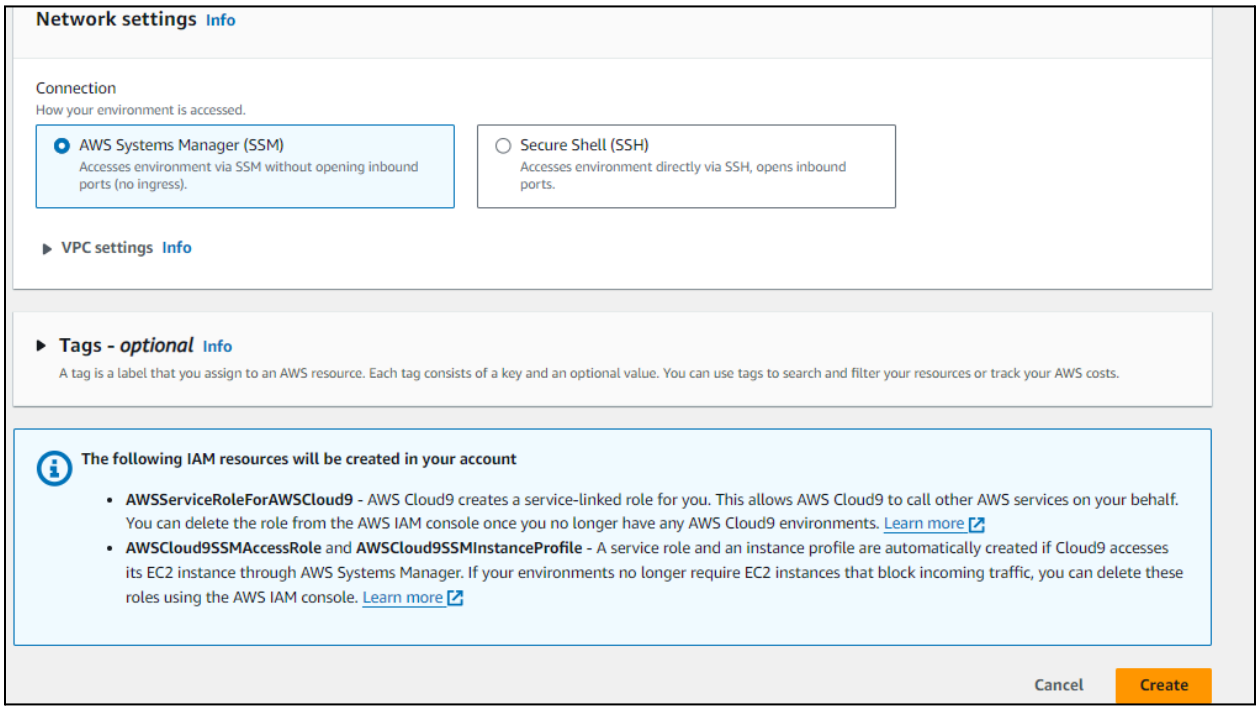
Step-1: Create a environment in Cloud9

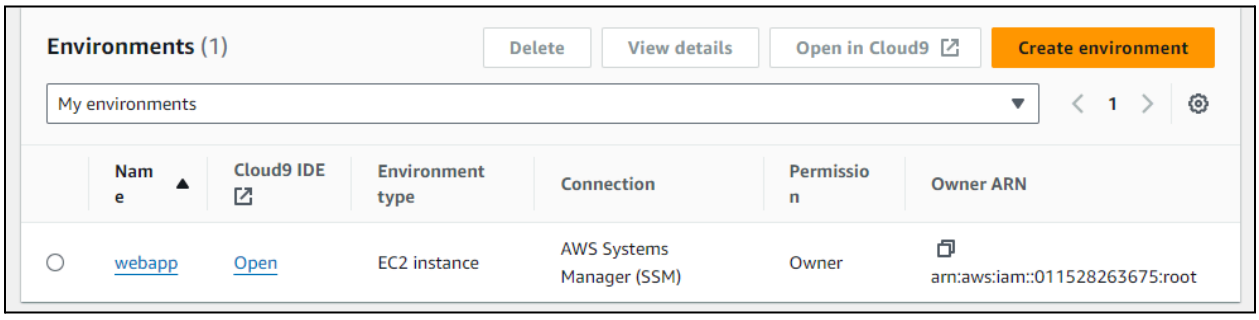


Step-2: Select new instance

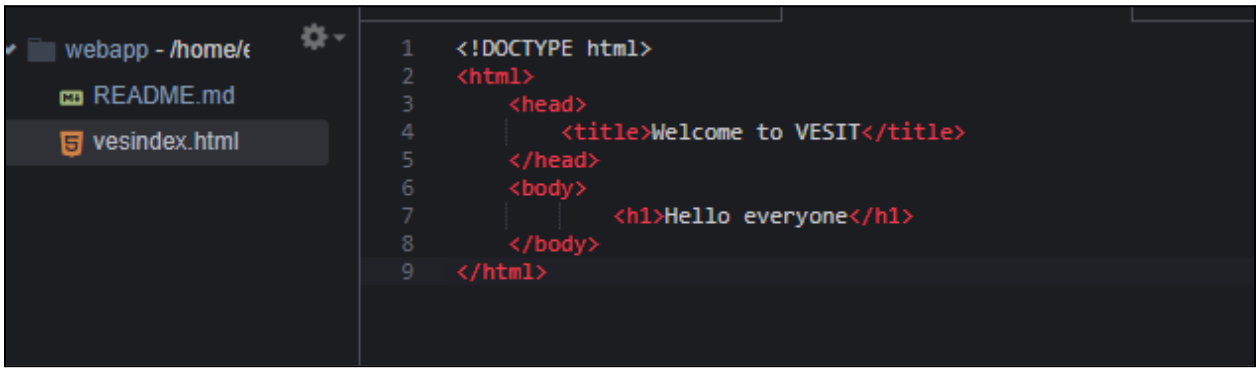


Step-3: Edit network settings and click on create environment

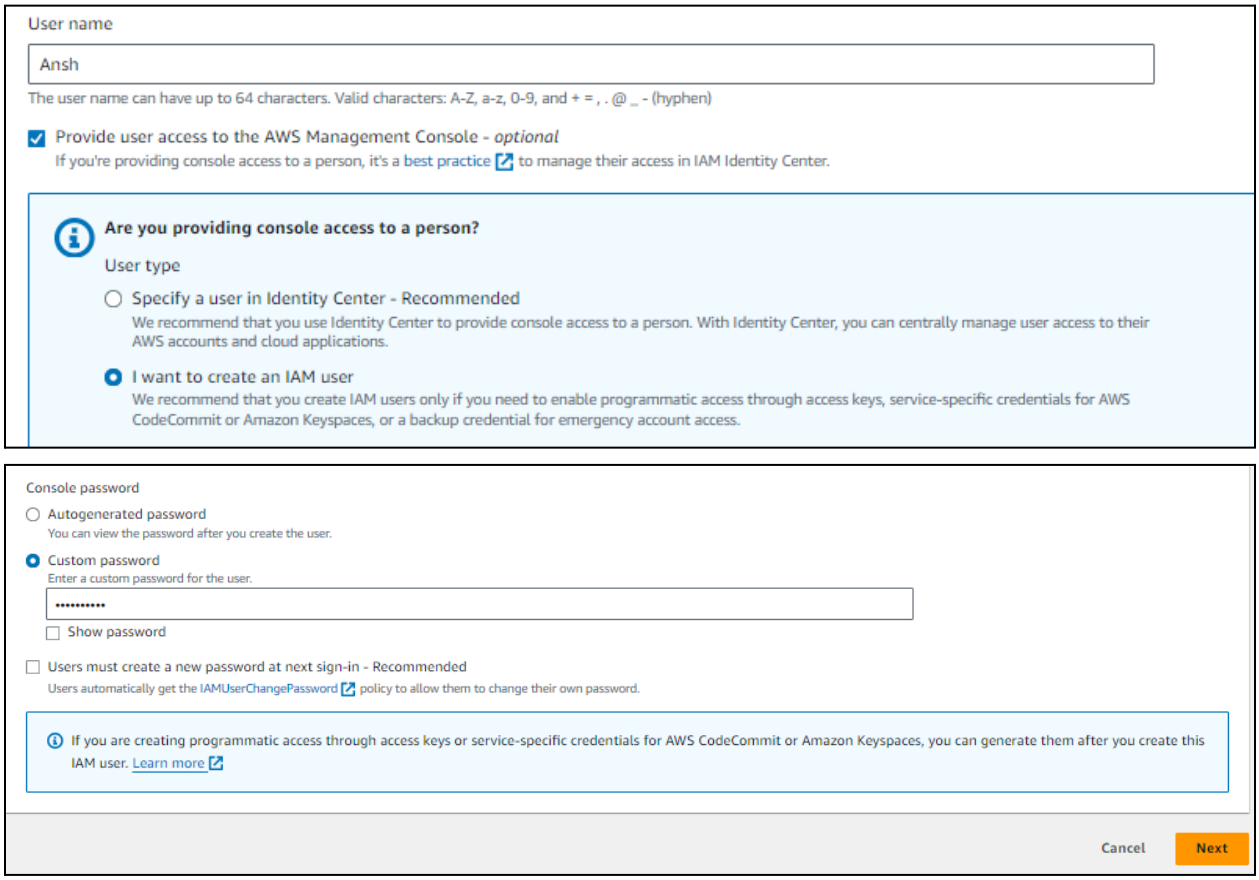




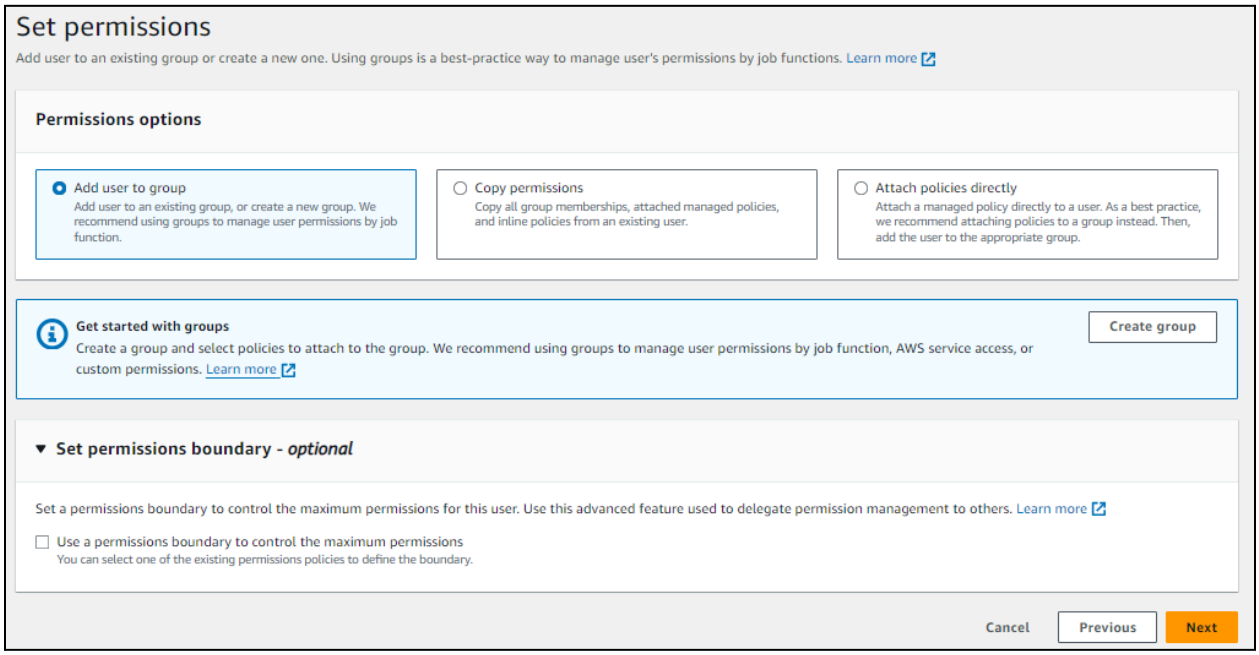
Step-4: Write a sample html code



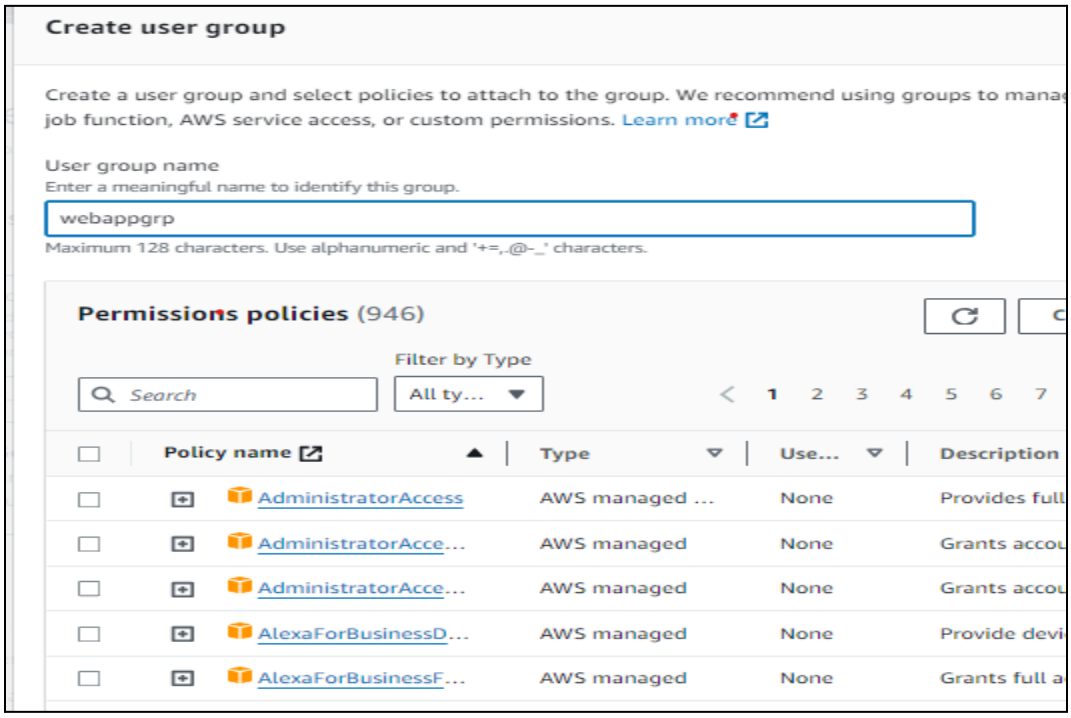
Step-5: Head to create a IAM User

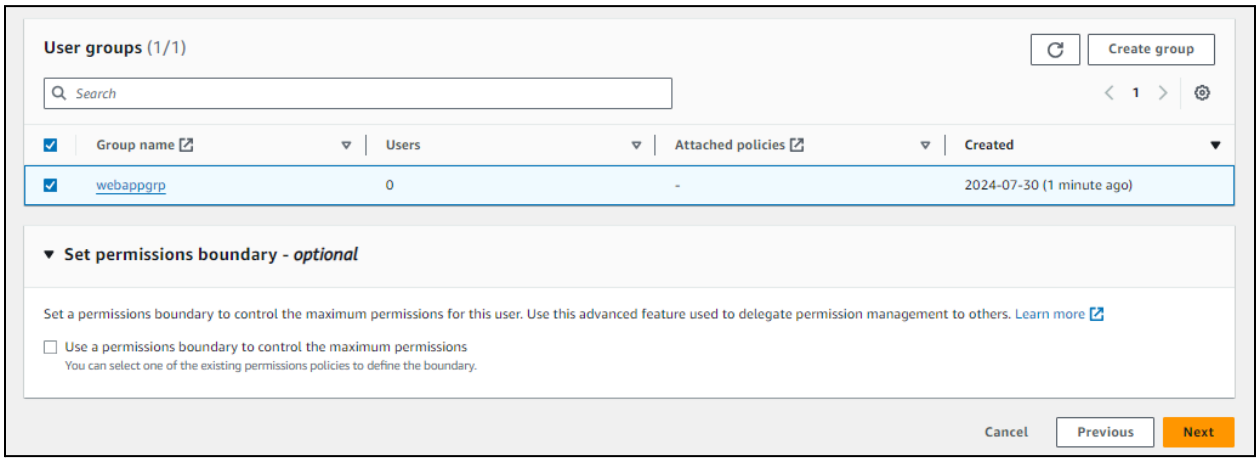


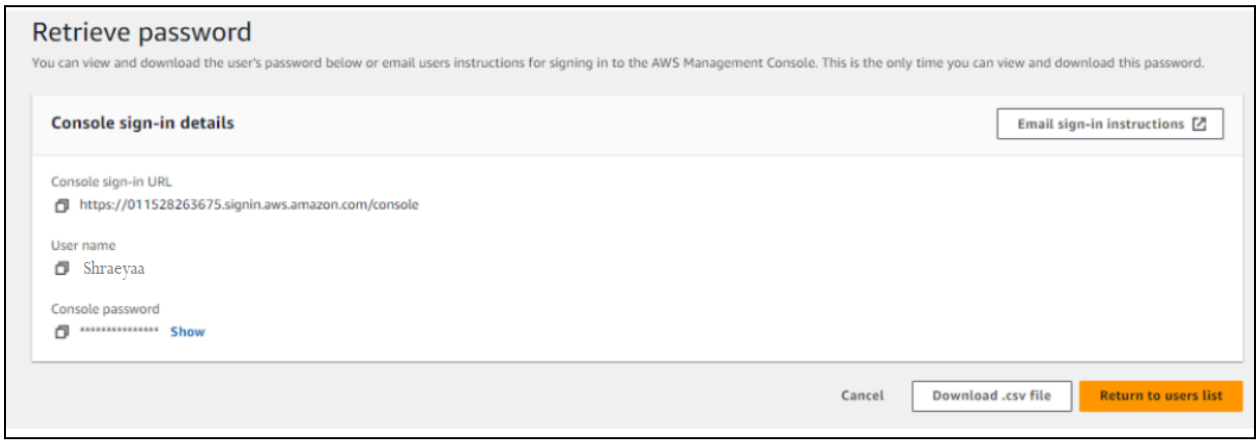
Step-6: Set the permissions



Step-7: Create a user group and set the password







ADVANCE DEVOPS EXP-2

Ansh Sarfare D15A/50

Aim: To build your application using AWS Codebuild and deploy on S3 using AWS CodePipeline deploy sample application on EC2 instance using AWS codedeploy.

Theory:

AWS Elastic Beanstalk is a Platform as a Service (PaaS) offering from Amazon Web Services (AWS) that allows developers to deploy and manage applications in the AWS Cloud without needing to manage the underlying infrastructure. It automates the deployment process, including provisioning resources like EC2 instances, load balancers, and databases, making it easier to manage web applications and services.

Key Features of Elastic Beanstalk:

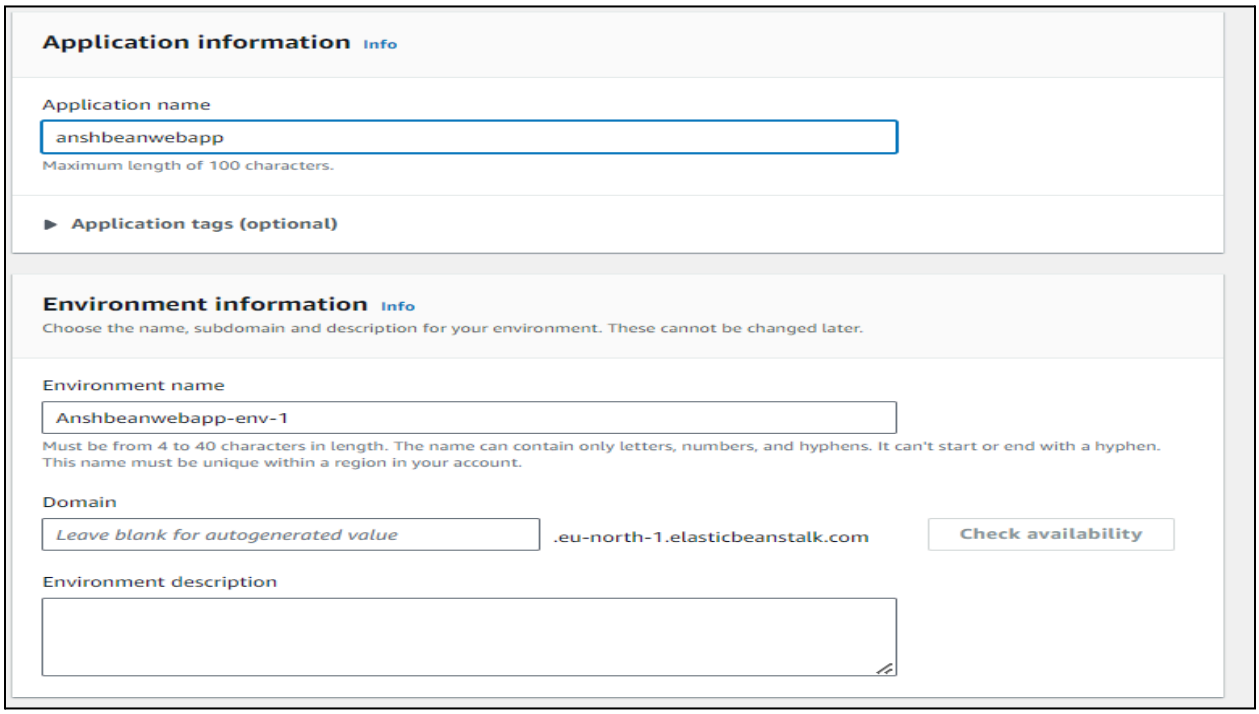
1. Ease of Use: Elastic Beanstalk simplifies the deployment process by automatically handling the infrastructure setup, deployment, monitoring, and scaling of your application. Developers can focus on writing code without worrying about the underlying infrastructure.

2. Support for Multiple Languages and Frameworks: Elastic Beanstalk supports a wide range of programming languages and frameworks, including Java, .NET, Node.js, PHP, Python, Ruby, Go, and Docker.

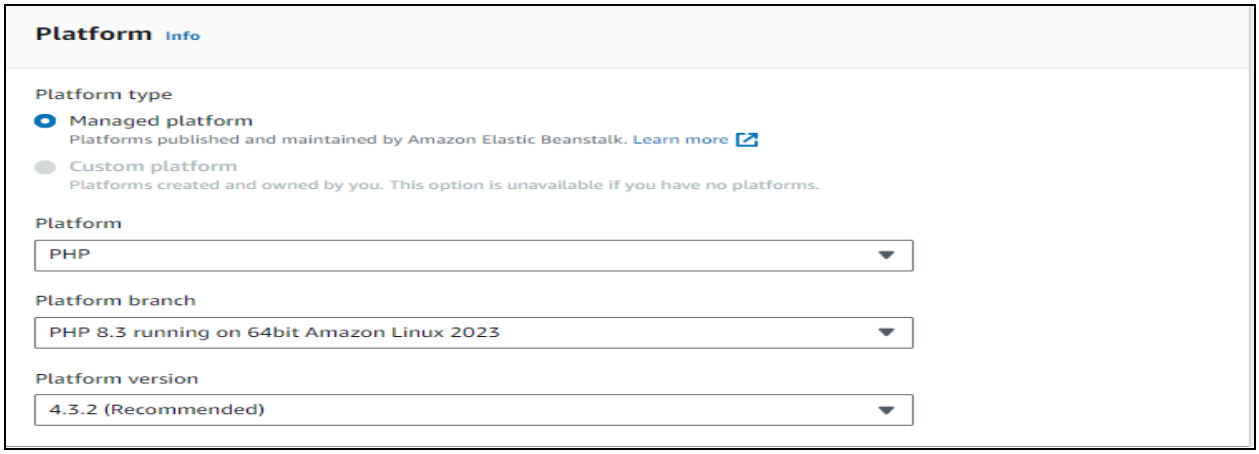
3. Automatic Scaling: Elastic Beanstalk automatically scales your application up or down based on the demand. It adjusts the number of instances running your application to meet traffic demands, ensuring optimal performance and cost-efficiency.

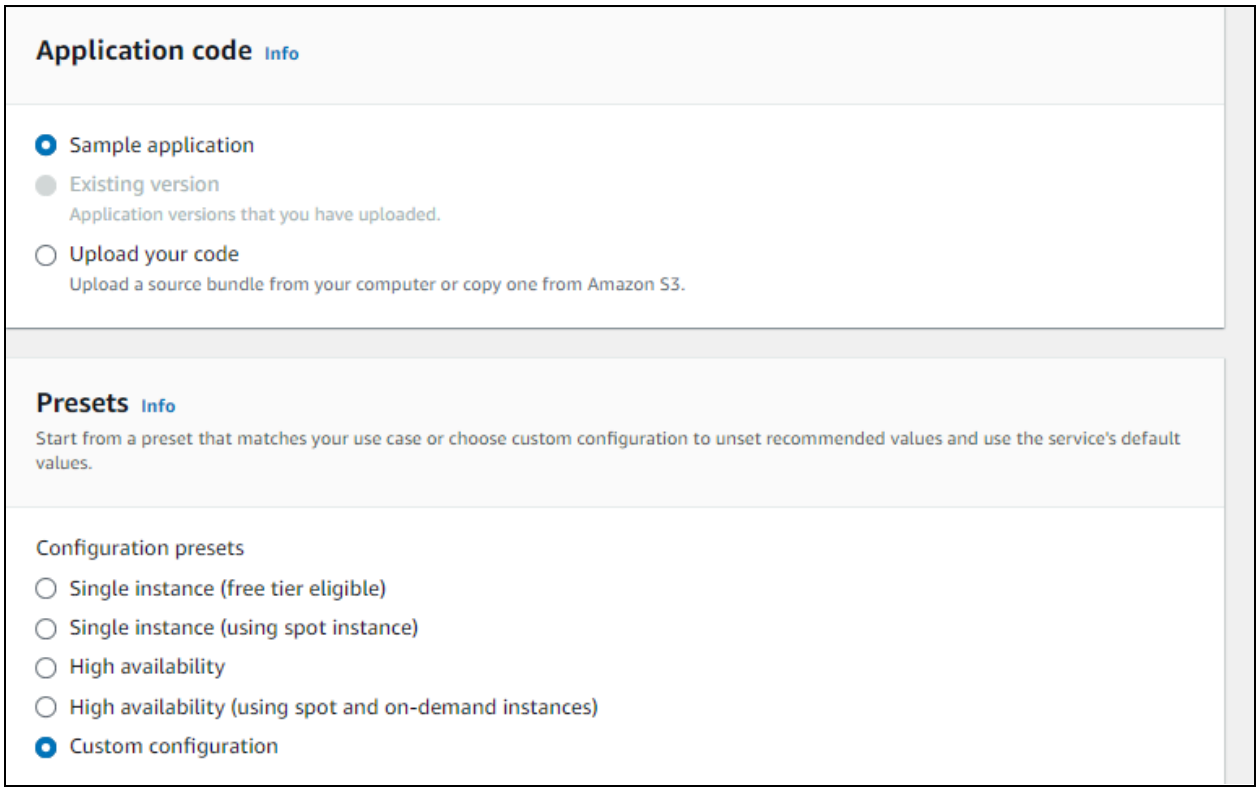
Using elastic beanstalk:

Step-1: Click on create application and configure the environment

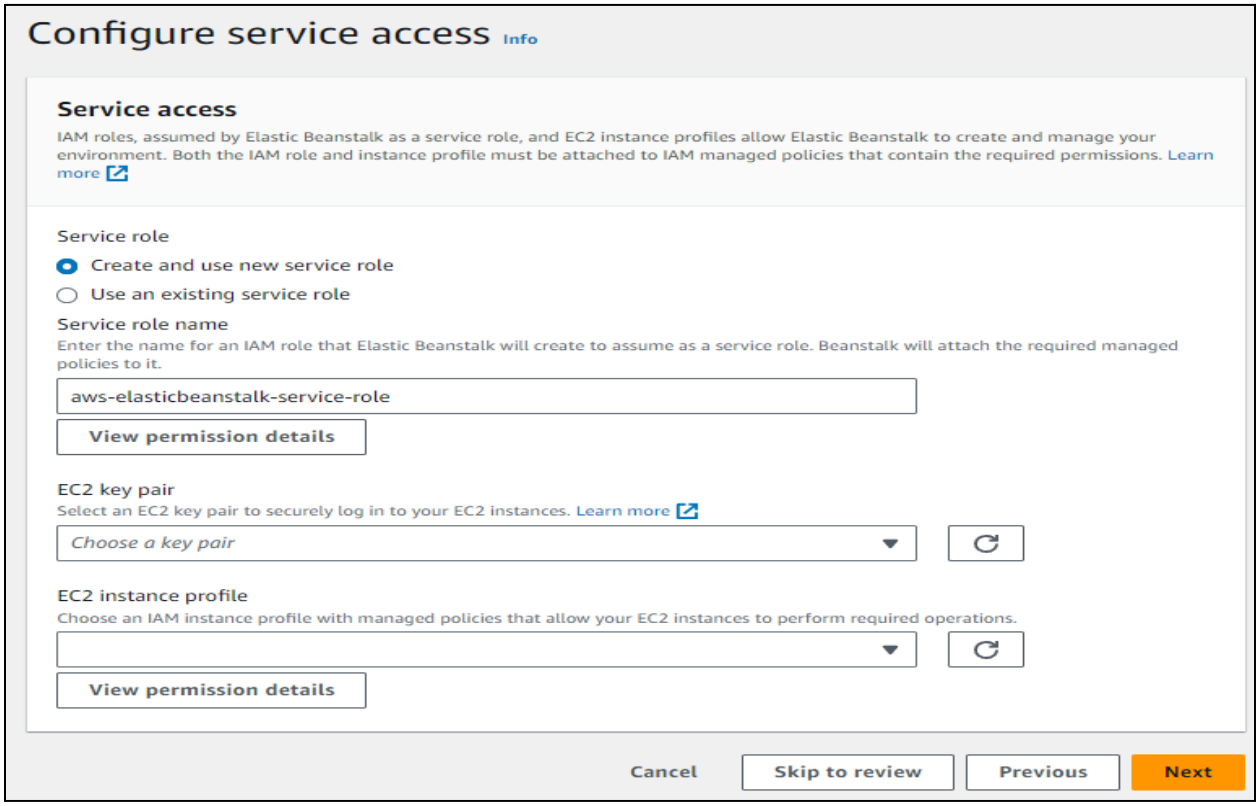


Step-2: Choose PHP from the dropdown menu and click next



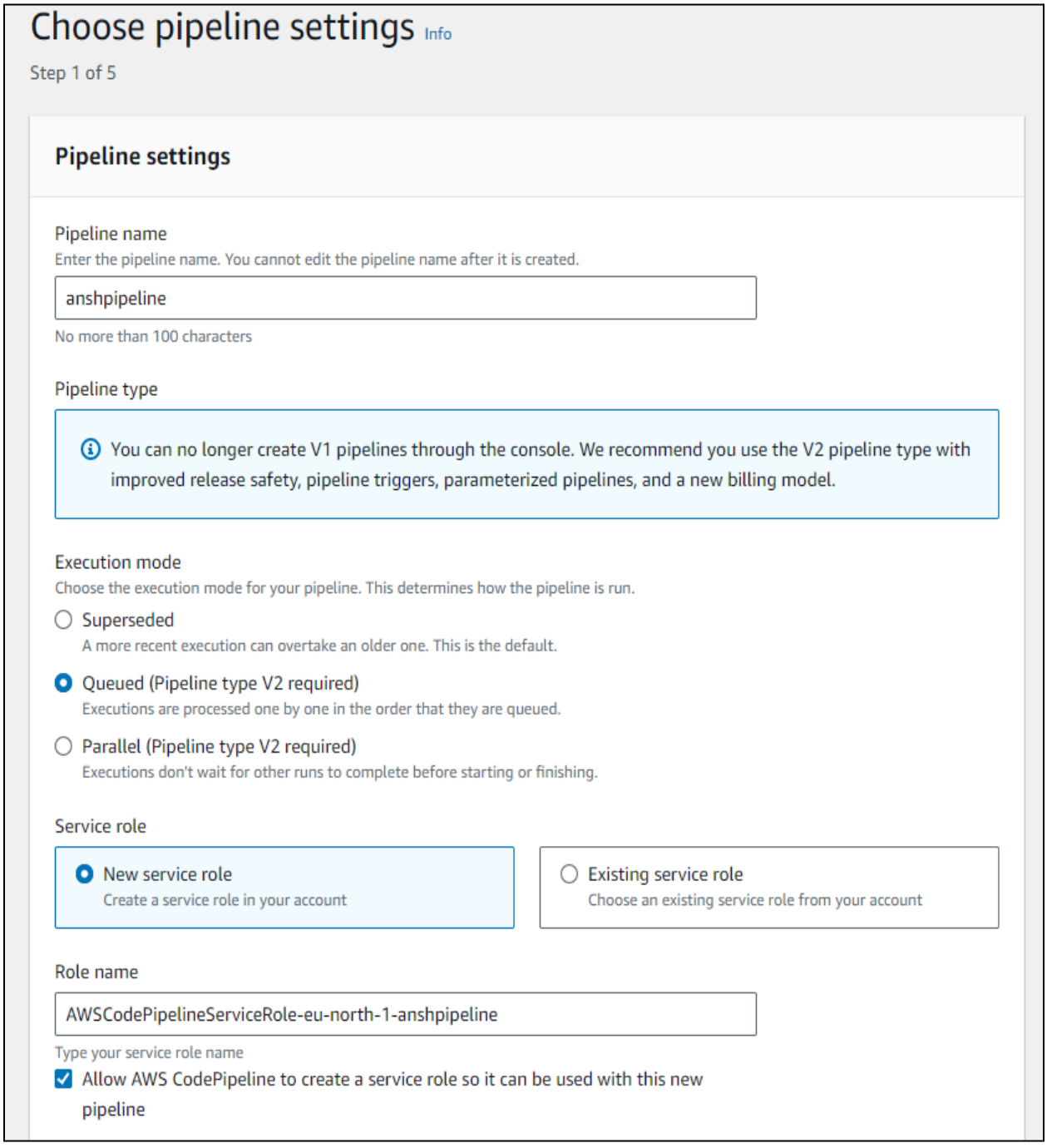


Step-3: Configure service access and then skip to review and submit

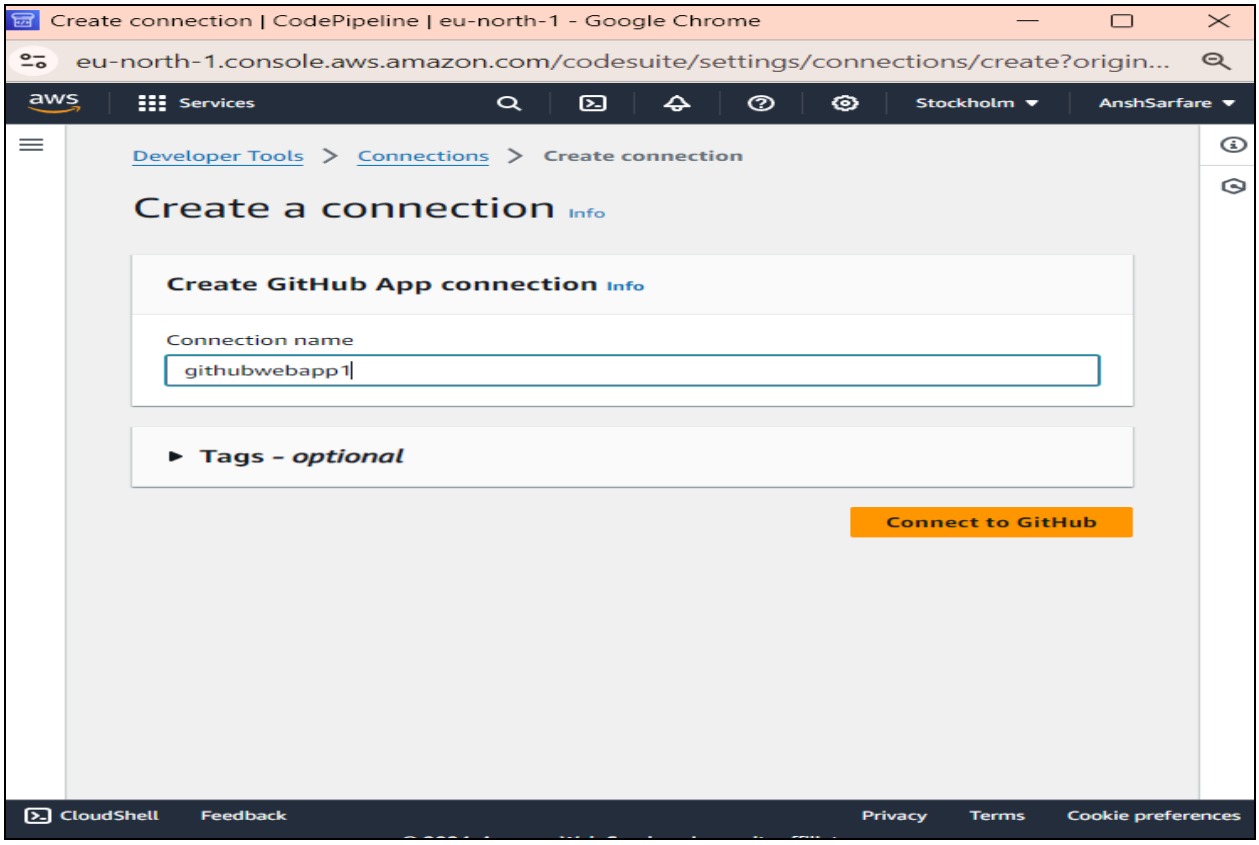


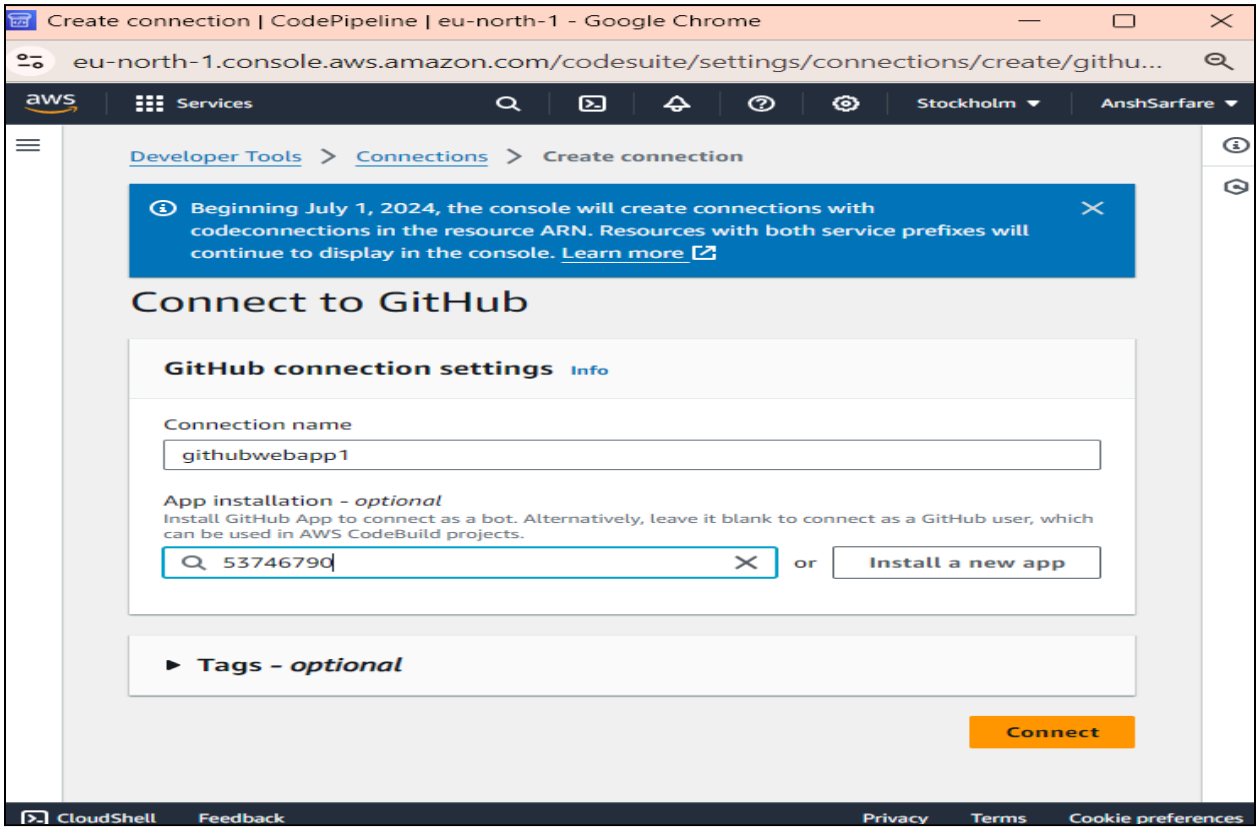
Codepipeline:

Step-1: Go to developer tools and select CodePipeline and create a new pipeline

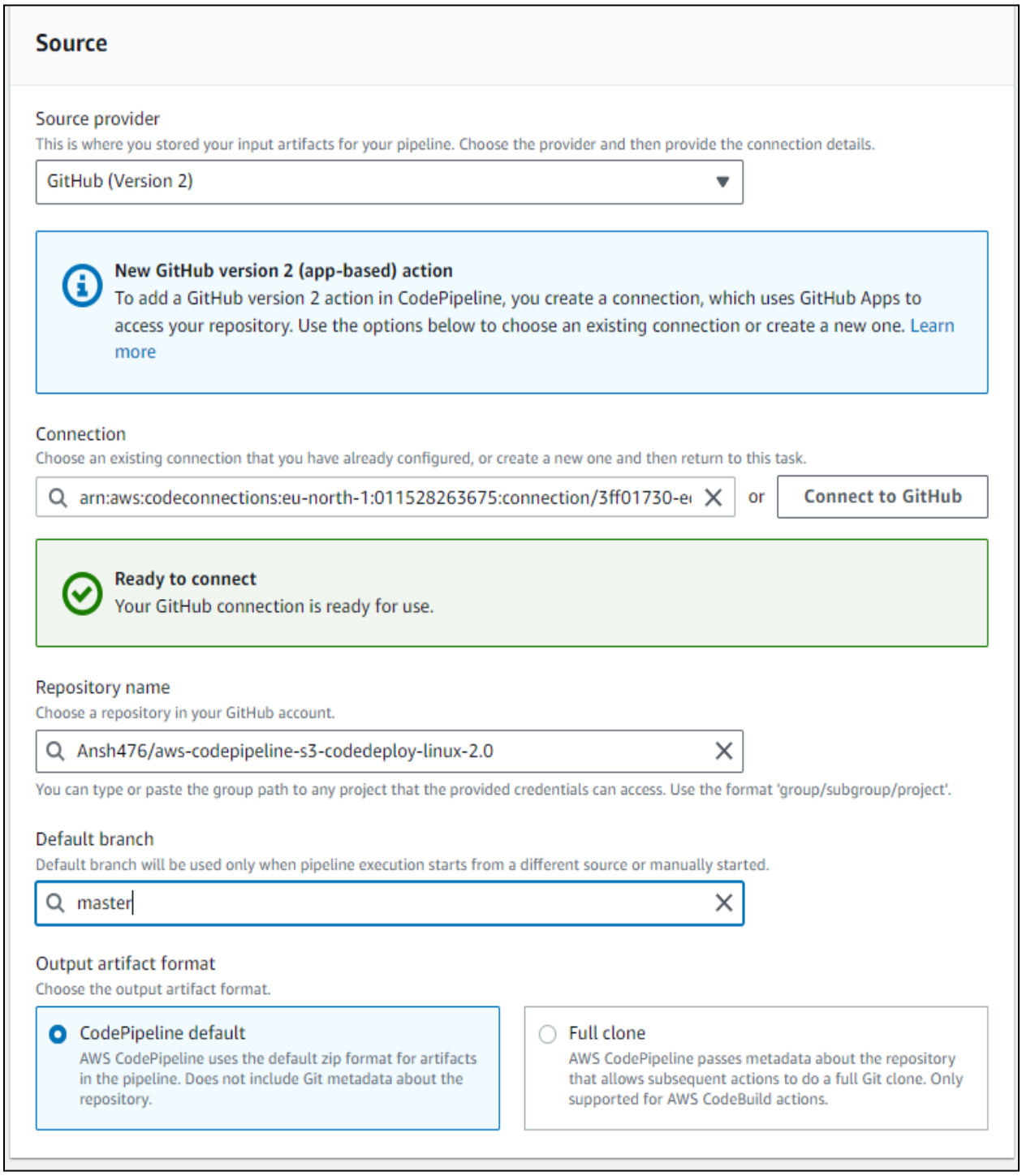


Step-2: In the source stage select Github v2 as the provider and then connect your github

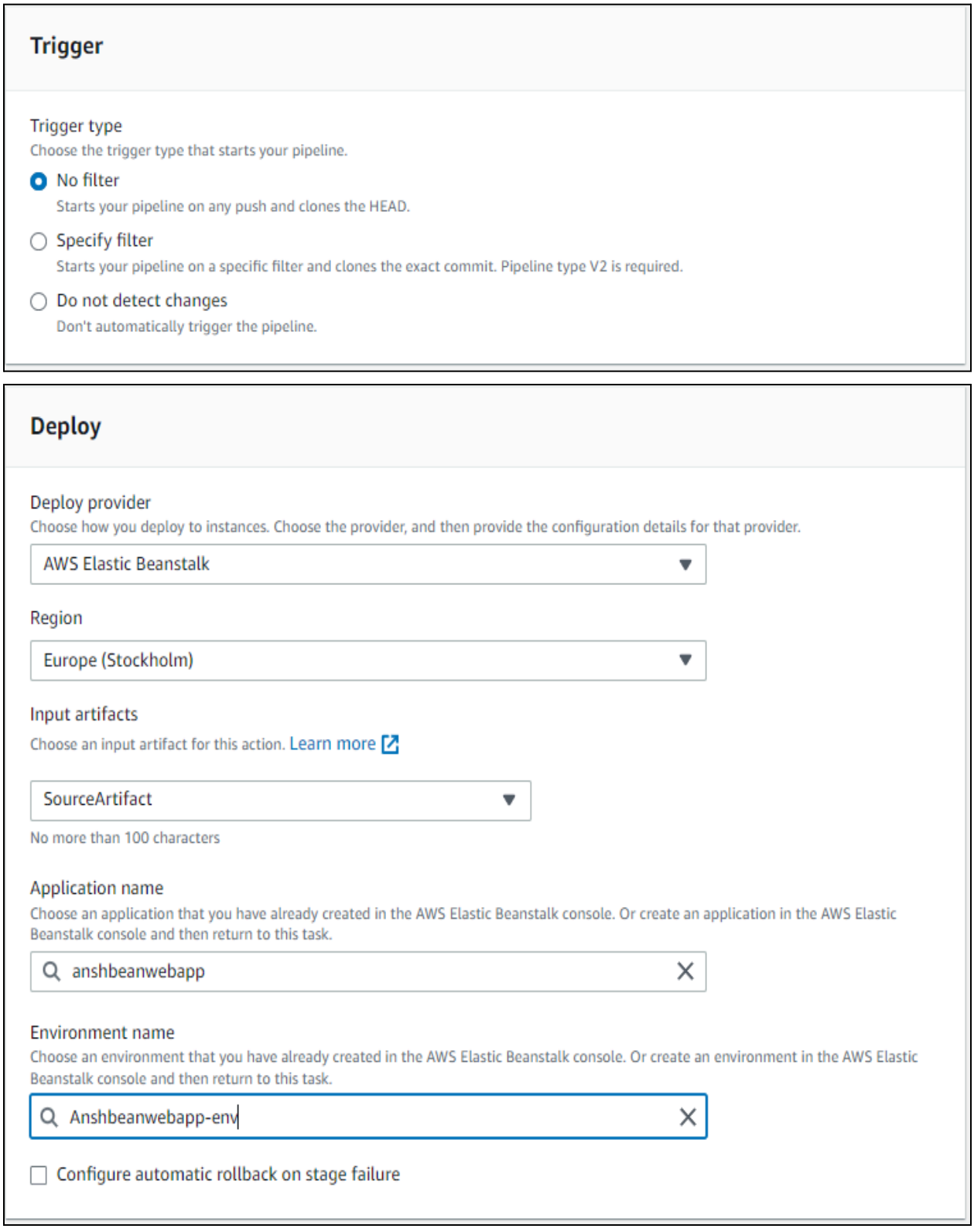

Create connection | CodePipeline | eu-north-1 - Google Chrome
eu-north-1.console.aws.amazon.com/codesuite/settings/connections/create?origin...
aws
Services
Stockholm
AnshSarfare
Developer Tools > Connections > Create connection
Create a connection Info
Create GitHub App connection Info
Connection name
githubwebapp1
Tags - optional
Connect to GitHub
CloudShell
Feedback
Privacy
Terms
Cookie preferences


Create connection | CodePipeline | eu-north-1 - Google Chrome
eu-north-1.console.aws.amazon.com/codesuite/settings/connections/create/githu...
aws
Services
Stockholm
AnshSarfare
Developer Tools > Connections > Create connection
Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. Learn more
Connect to GitHub
GitHub connection settings Info
Connection name
githubwebapp1
App installation - optional
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.
5374679O
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
Privacy
Terms
Cookie preferences

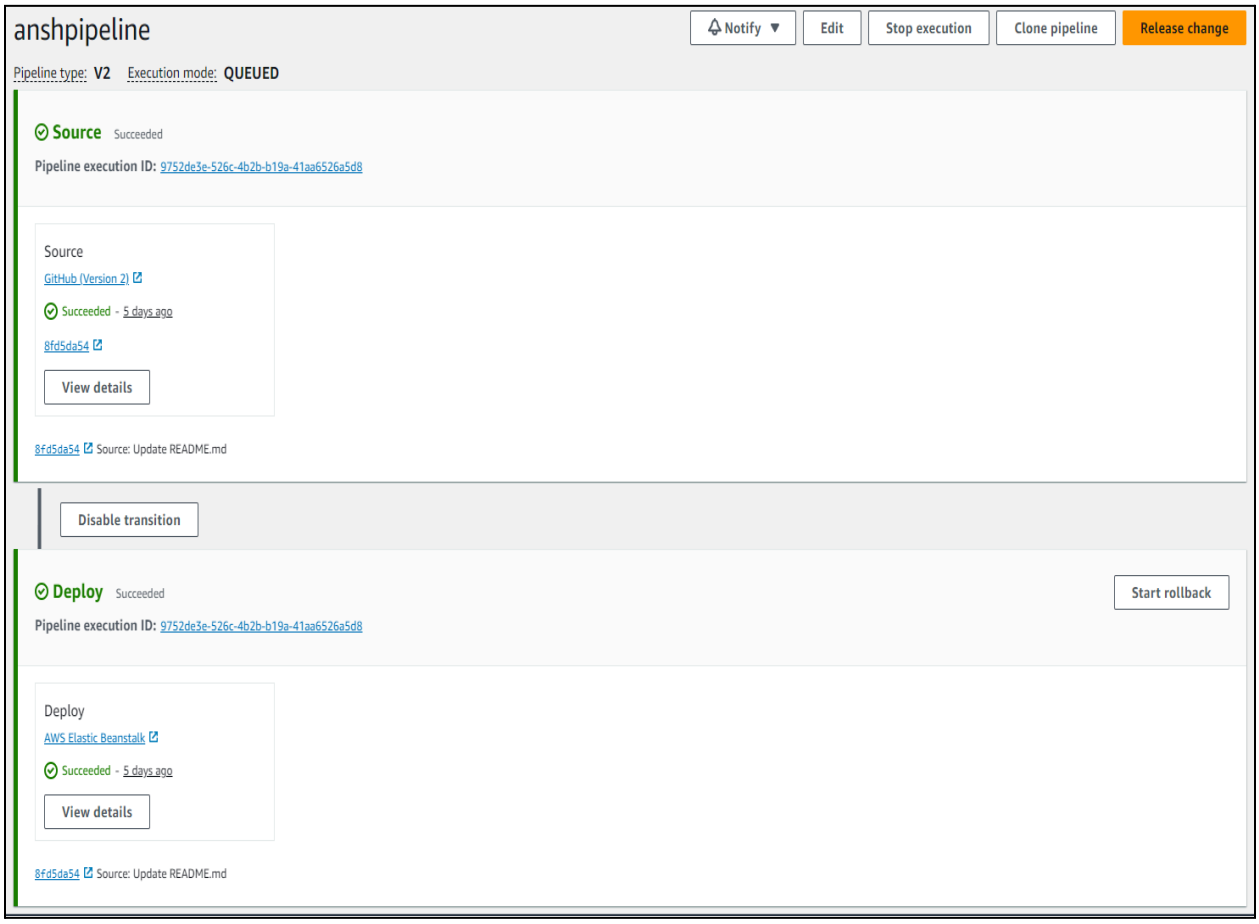
Step-3: Once the connection is established from the drop down menu select the repository and the branch



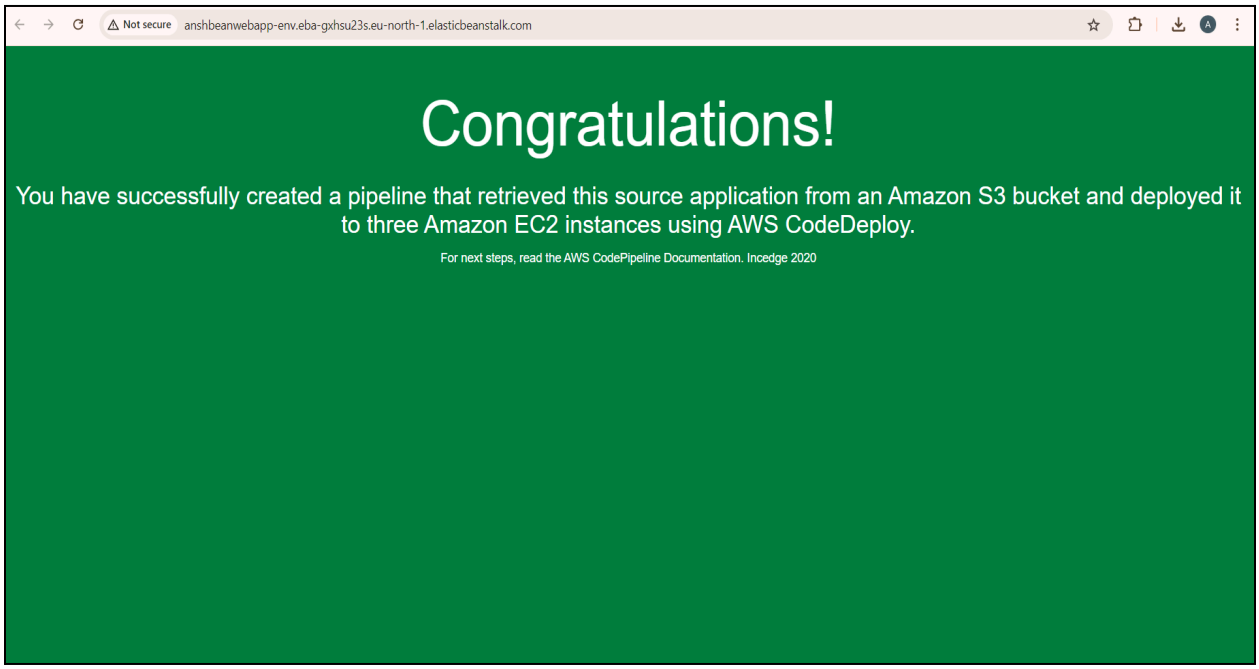
Step-4: Select No filter for triggers and your environment in Deploy field



Step-5: Check the URL provided in the EBS environment.

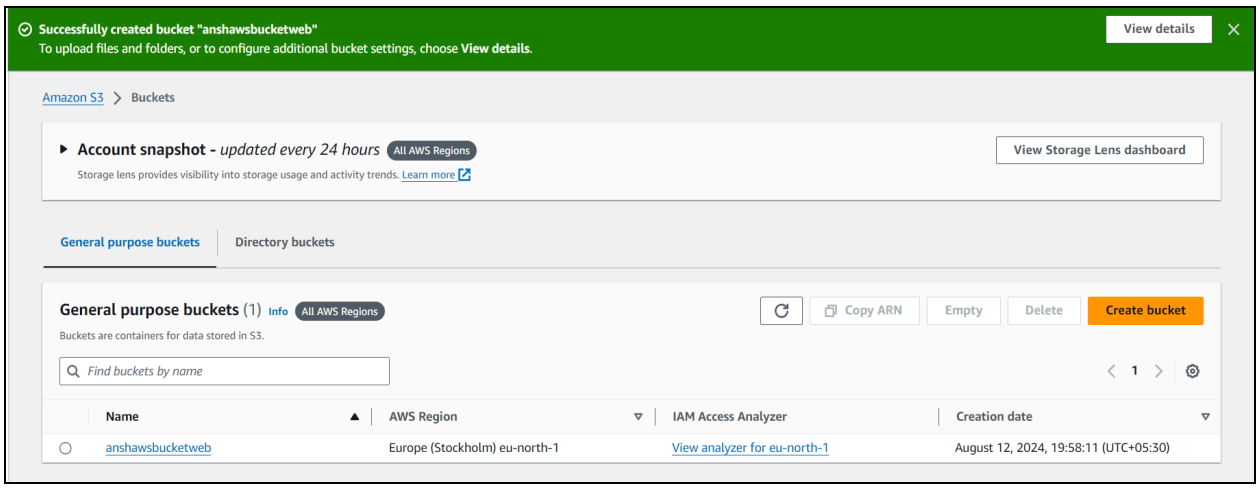


Step-6: The website is hosted from the forked repo in our beanstalk environment

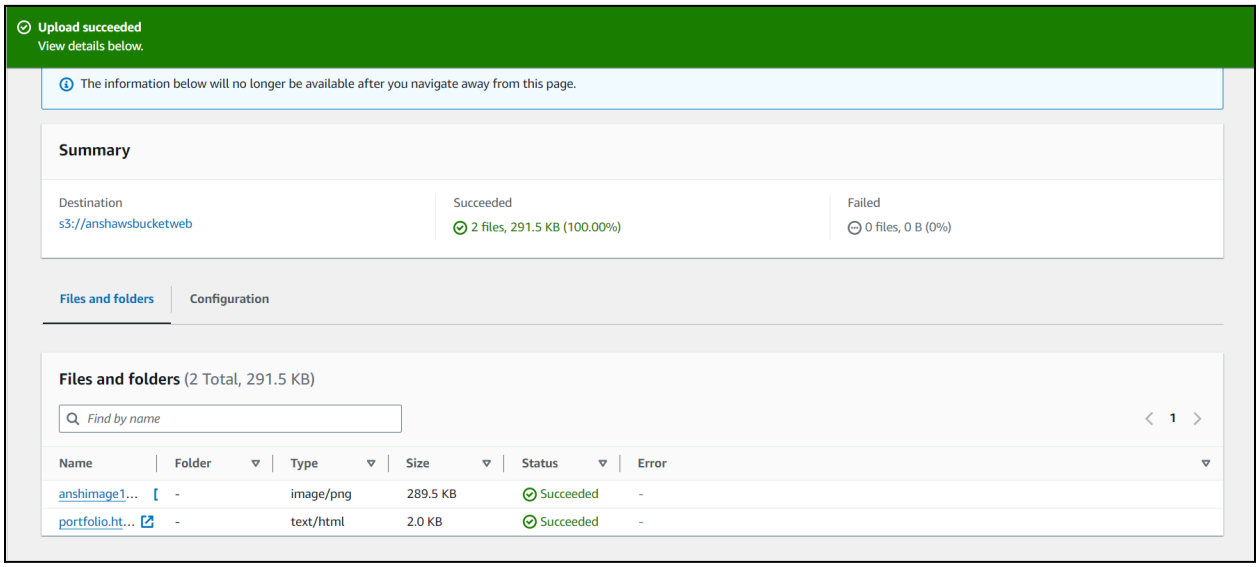


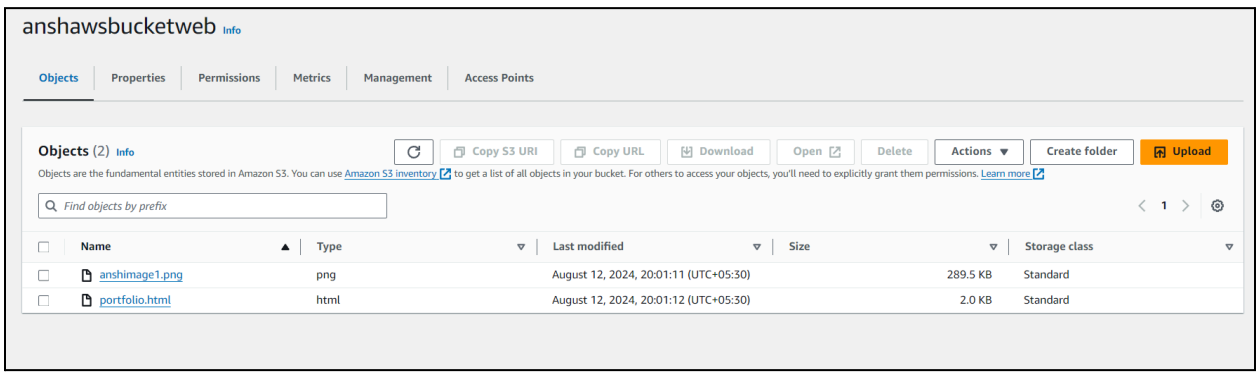
Using S3 Bucket:

Step-1: Create Bucket

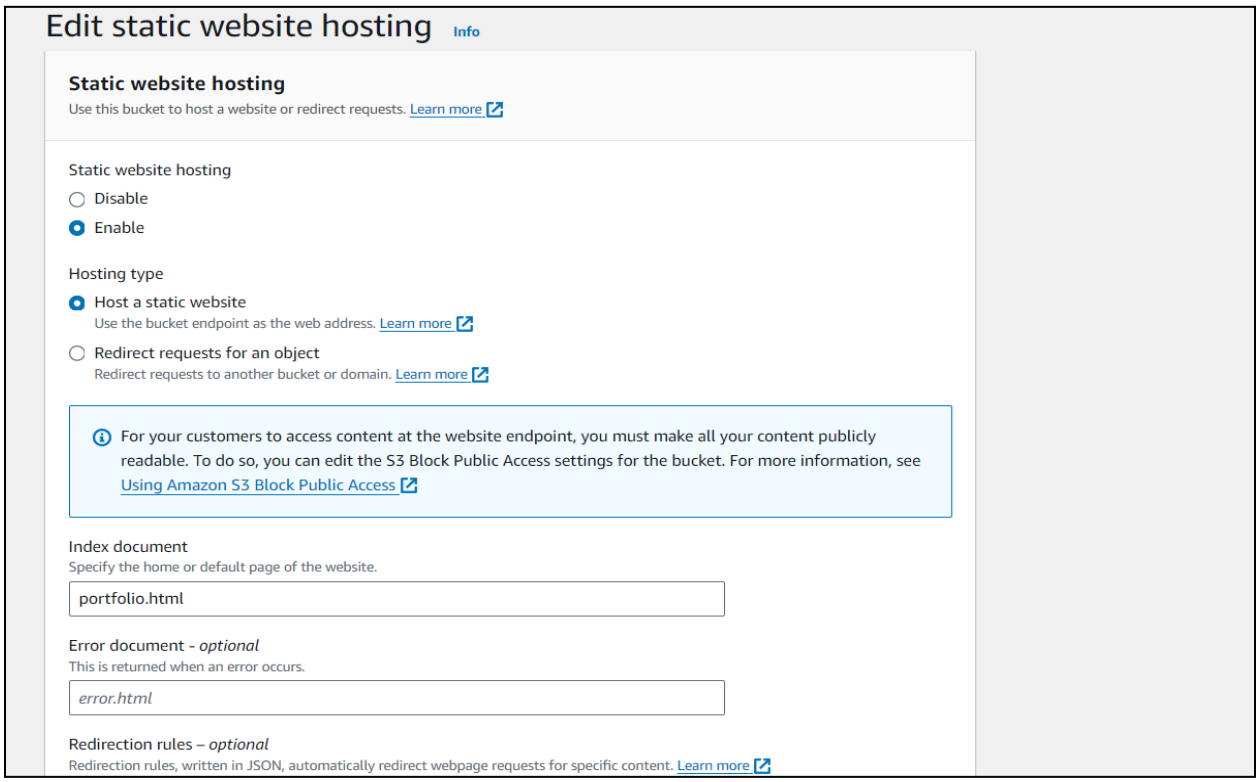


Step-2: Upload your Files

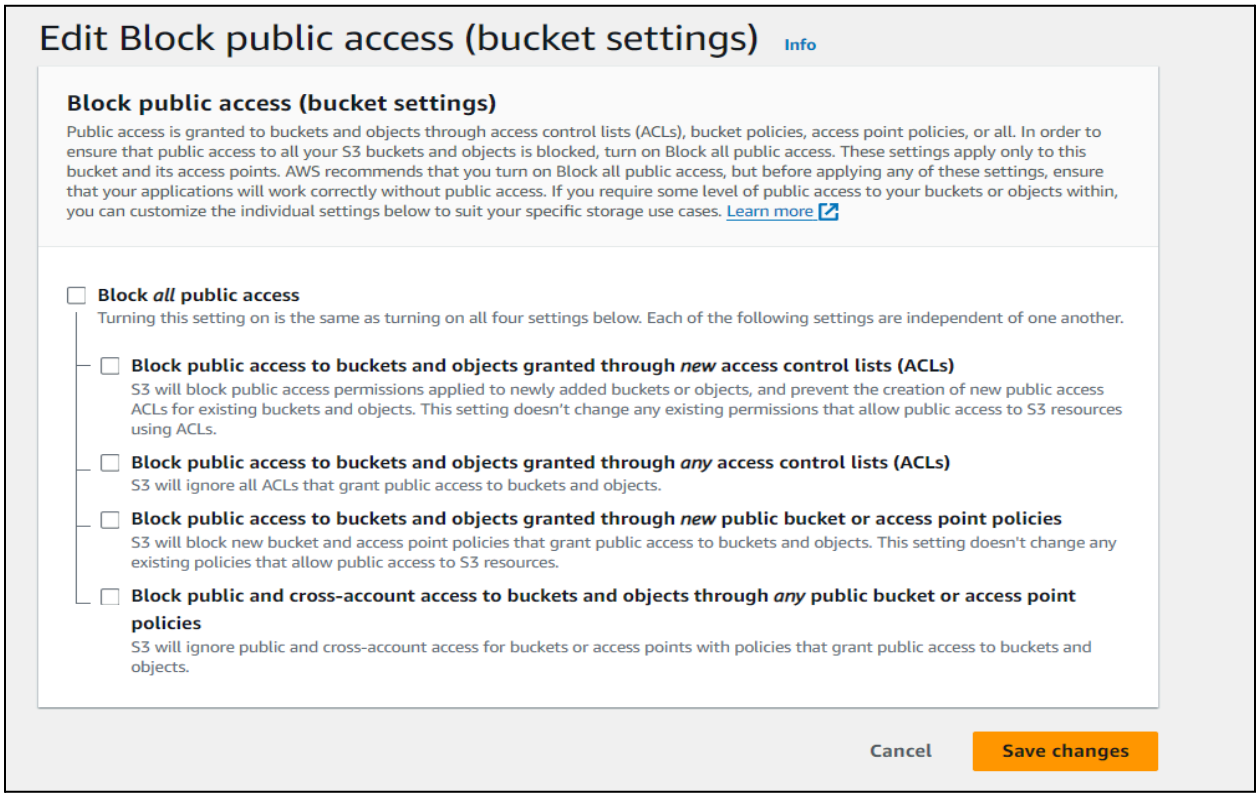




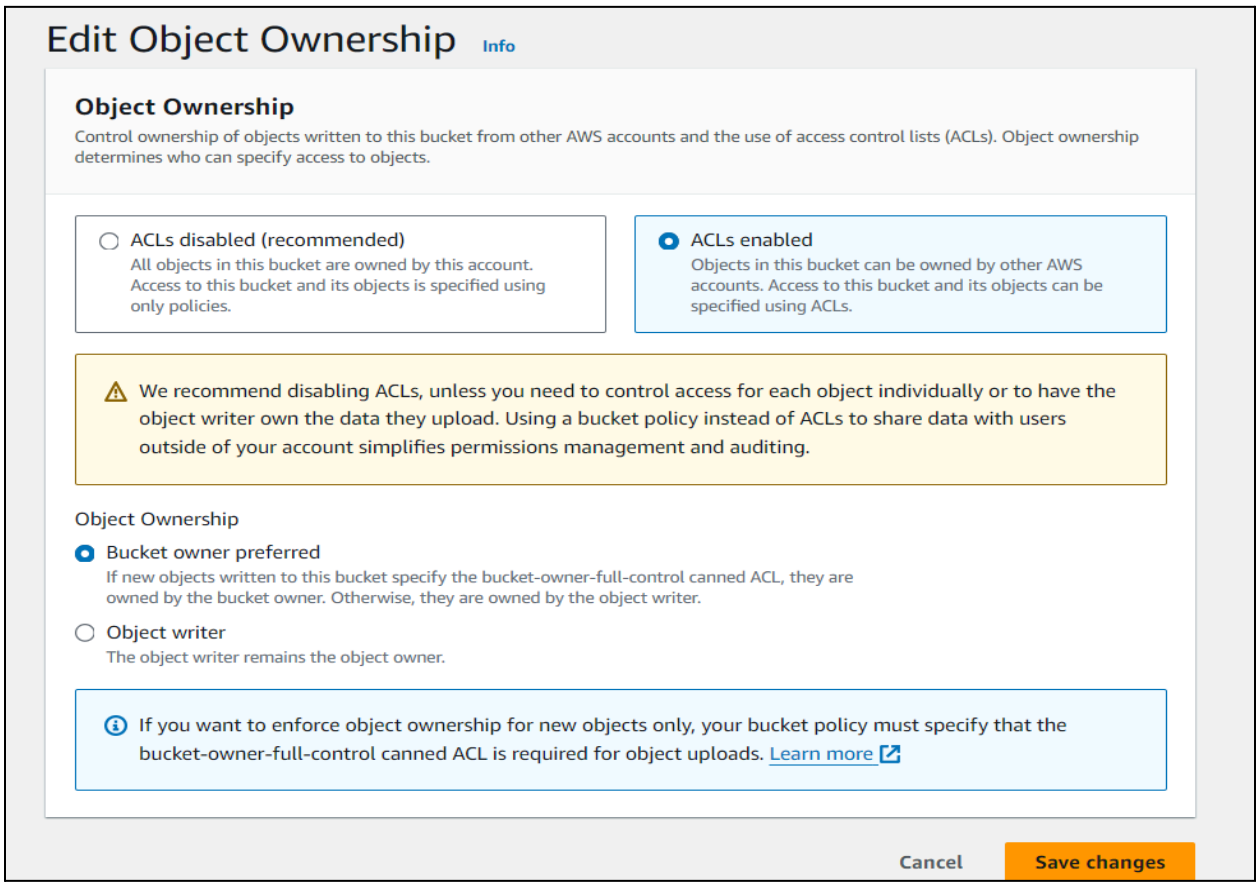
Step-3: Go to properties and edit static website hosting



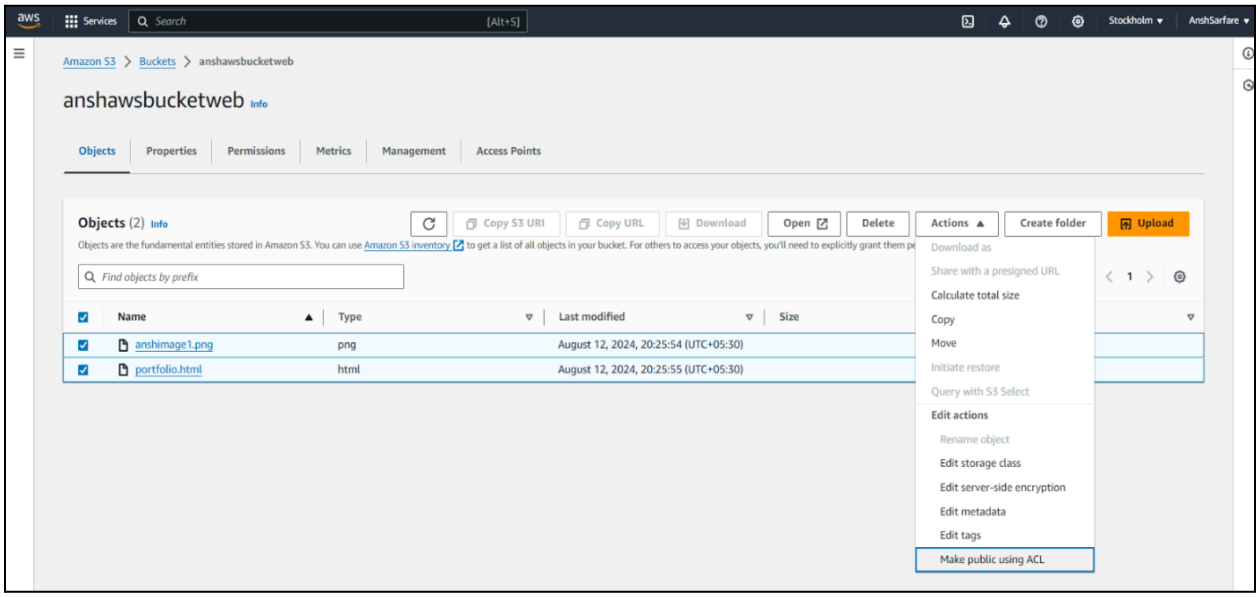
Step-4: Go to permissions and edit public block access



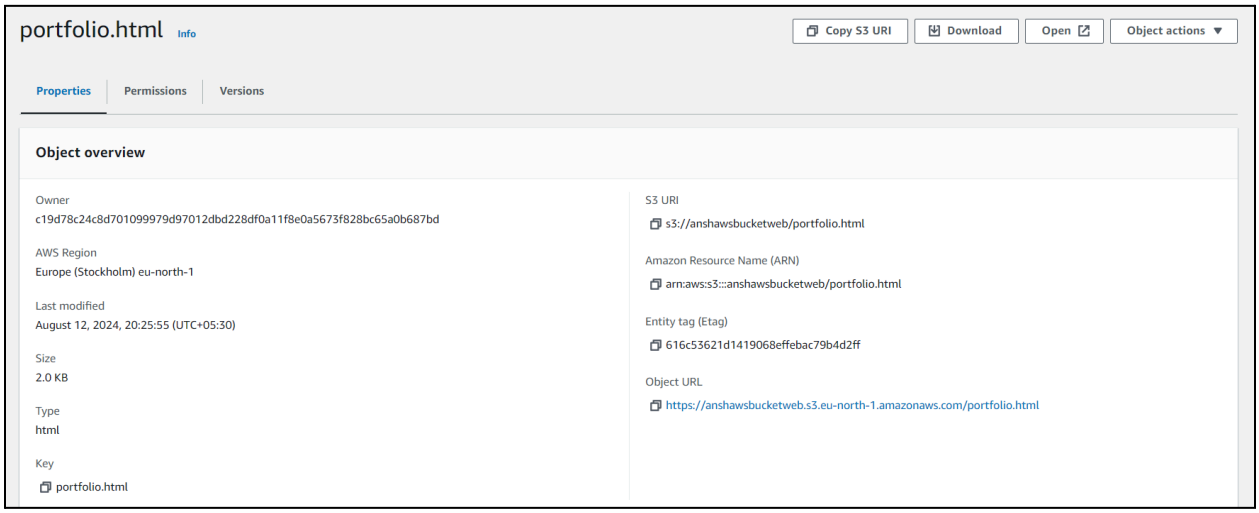
Step-5: Then in permissions enable ACL



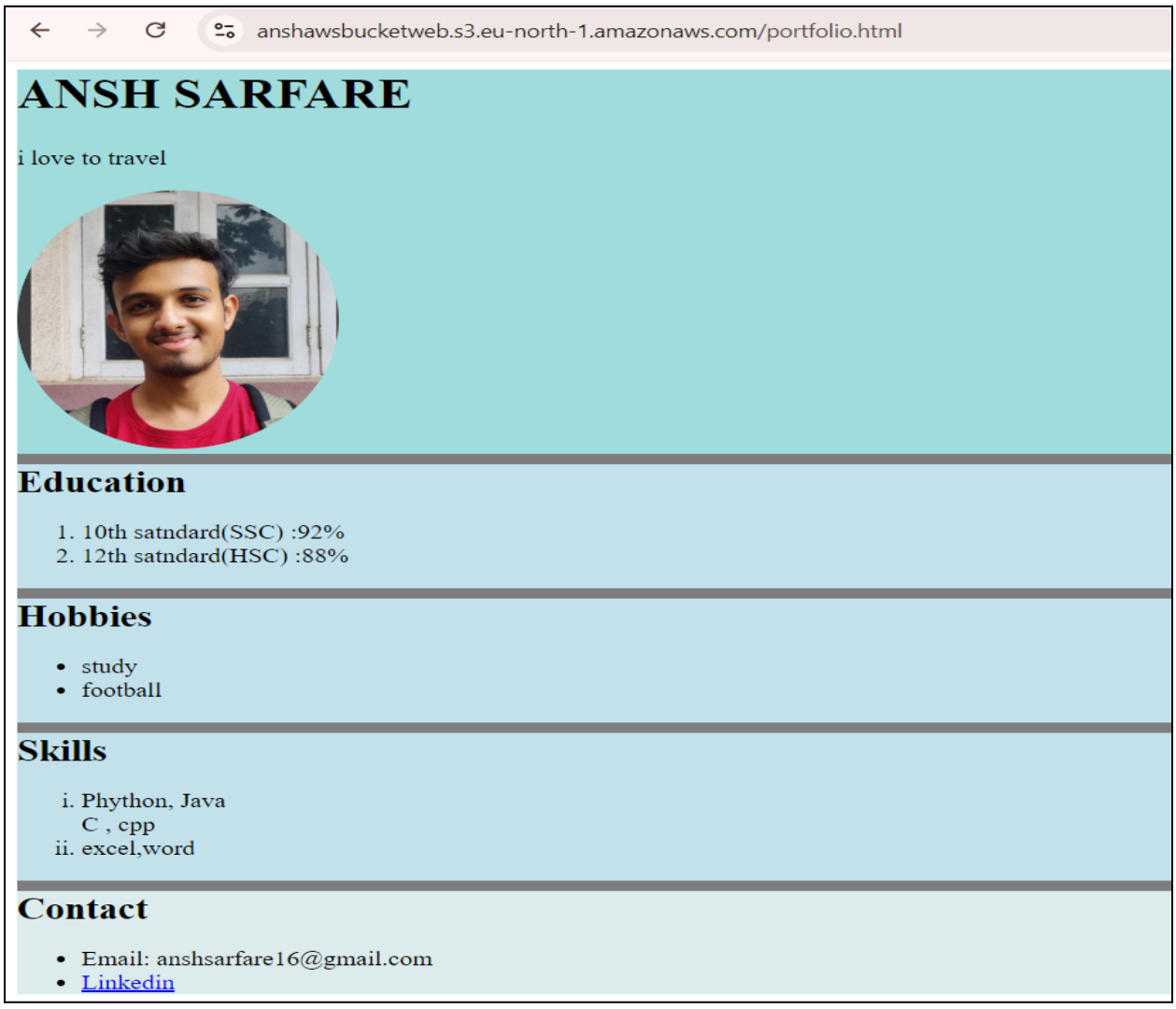
Step-6: Select your files and in actions click on make public using ACL



Step-7: Then in objects click on html file and copy the object URL

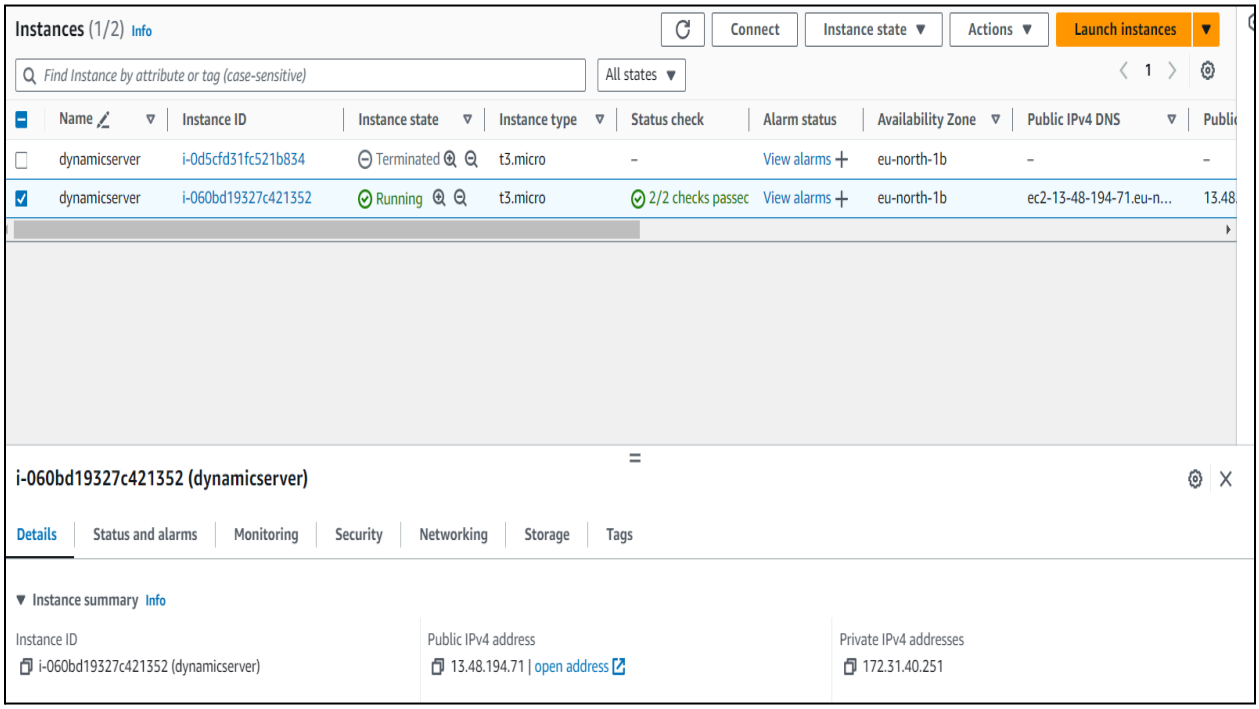


Hosted website:



Using EC2:

Step-1: Create new instance

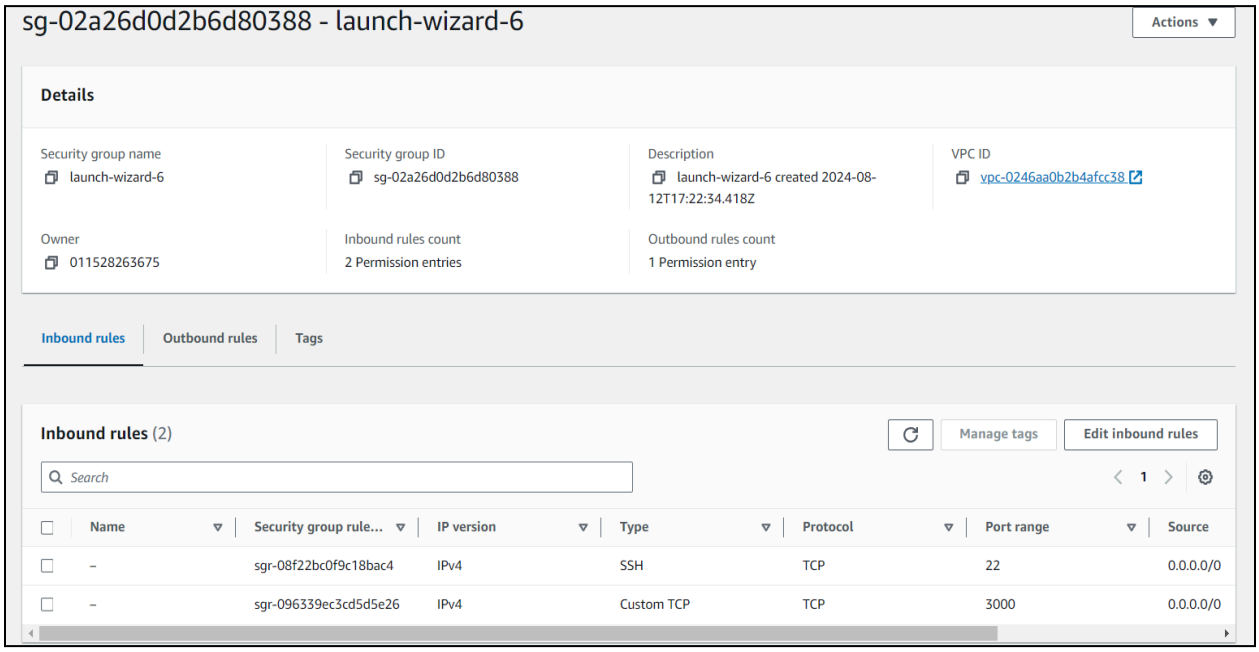


Step-2: Run the commands mentioned to clone your repo and run it

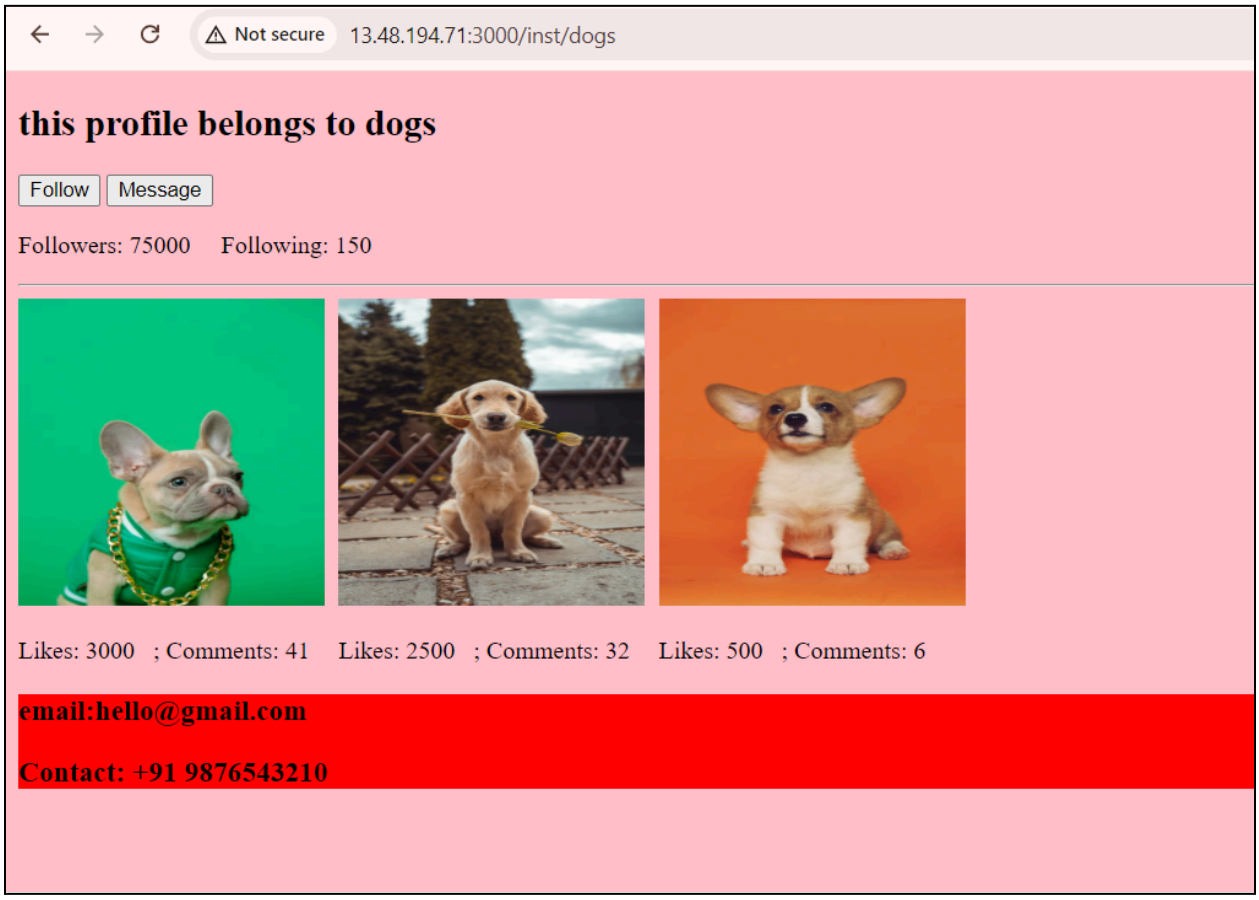
```
ubuntu@ip-172-31-33-138:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-33-138:~$ mkdir ansh
ubuntu@ip-172-31-33-138:~$ cd ansh
ubuntu@ip-172-31-33-138:~/ansh$ git clone https://github.com/Ansh476/instademo.git
Cloning into 'instademo'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 54 (delta 19), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (54/54), 23.02 KiB | 1.15 MiB/s, done.
Resolving deltas: 100% (19/19), done.
ubuntu@ip-172-31-33-138:~/ansh$ ls
instademo
ubuntu@ip-172-31-33-138:~/ansh$ cd instademo
ubuntu@ip-172-31-33-138:~/ansh/instademo$ ls
data.json  index.js  package-lock.json  package.json  public  views
ubuntu@ip-172-31-33-138:~/ansh/instademo$
```

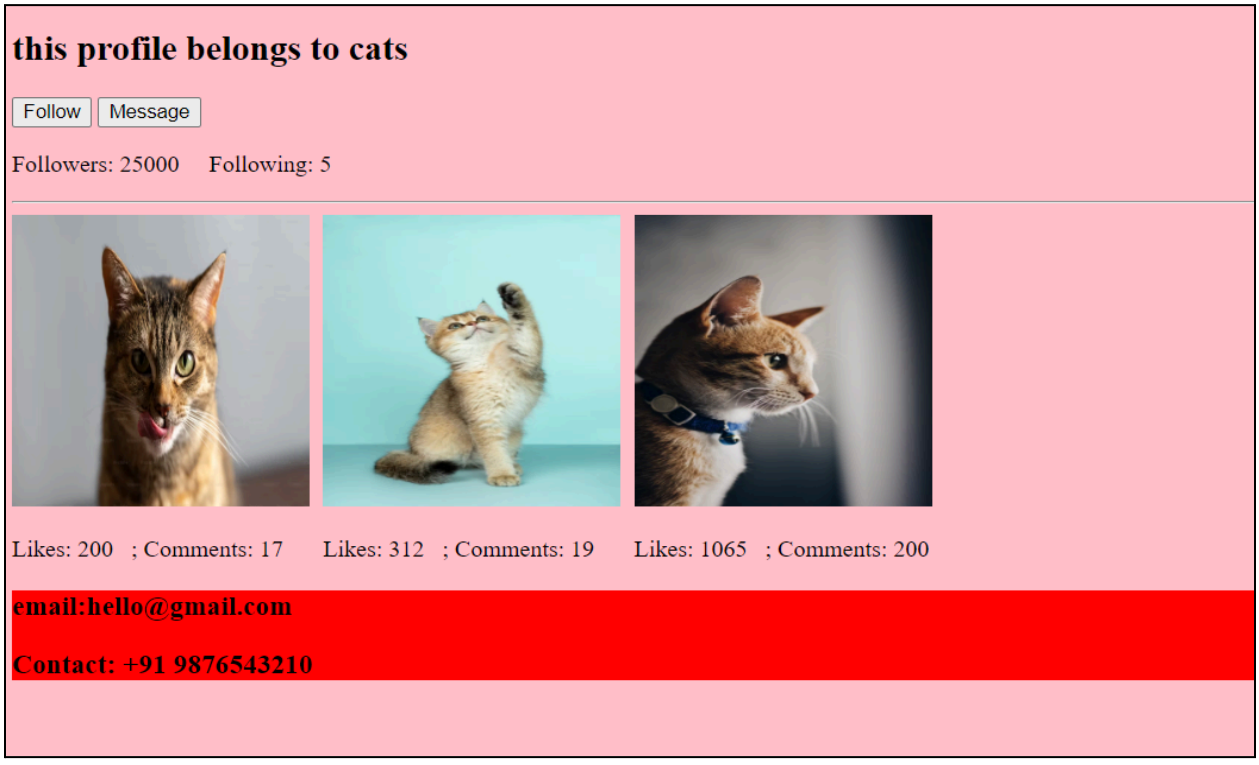
```
ubuntu@ip-172-31-40-251:~/ansh/instademo$ npm i

added 80 packages, and audited 81 packages in 2s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-40-251:~/ansh/instademo$ npm start

> ejs@1.0.0 start
> nodemon index.js

sh: 1: nodemon: not found
ubuntu@ip-172-31-40-251:~/ansh/instademo$ sudo npm install -g nodemon

added 29 packages in 2s

4 packages are looking for funding
  run `npm fund` for details
ubuntu@ip-172-31-40-251:~/ansh/instademo$ npm start

> ejs@1.0.0 start
> nodemon index.js

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
listening on port3000
```

Step-3: Edit the security group rule: ssh and custom tcp port:3000



Step-4: Hosted Website





ADVANCE DEVOPS EXP-3

ANSH SARFARE **D15A/50**

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

Theory:

Container-based microservices architectures have profoundly changed the way development and operations teams test and deploy modern software. Containers help companies modernize by making it easier to scale and deploy applications, but containers have also introduced new challenges and more complexity by creating an entirely new infrastructure ecosystem.

Large and small software companies alike are now deploying thousands of container instances daily, and that's a complexity of scale they have to manage.

Enter the age of Kubernetes.

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Infact, Kubernetes has established itself as the defacto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), backed by key players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

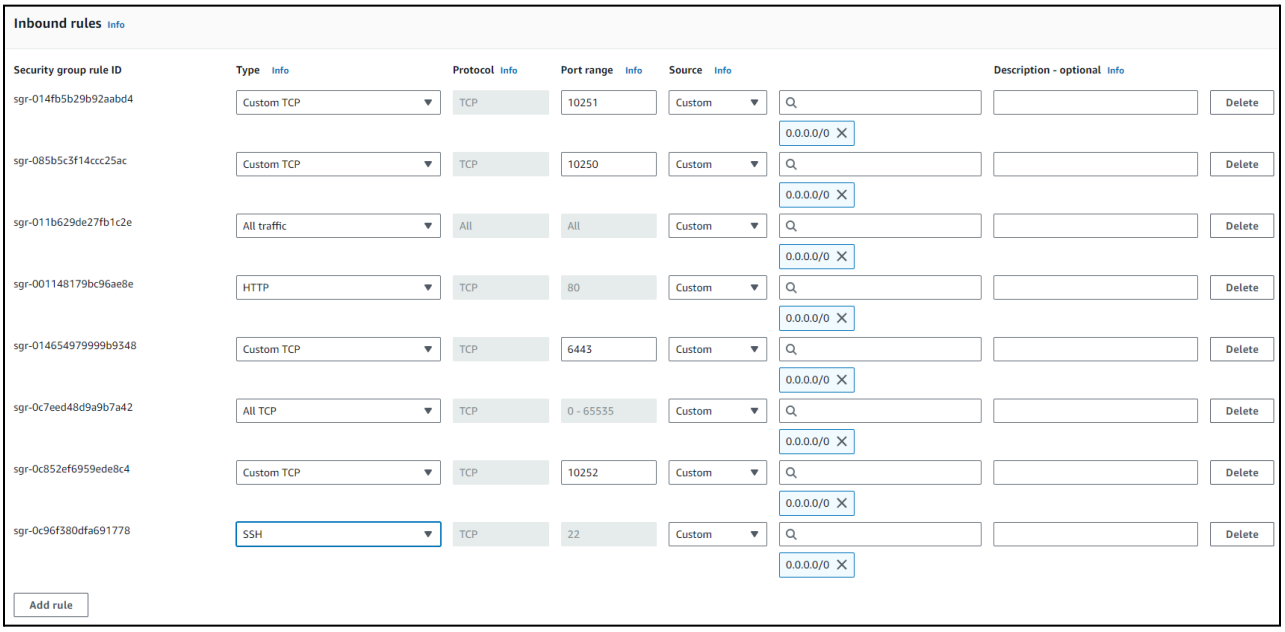
Kubernetes makes it easy to deploy and operate applications in a microservice architecture. It does so by creating an abstraction layer on top of a group of hosts so that development teams can deploy their applications and let Kubernetes manage the following activities:

- Controlling resource consumption by application or team
- Evenly spreading application load across a hosting infrastructure
- Automatically load balancing requests across different application instances
- Monitoring resource consumption and resource limits to automatically stop applications from consuming too many resources and restarting them again

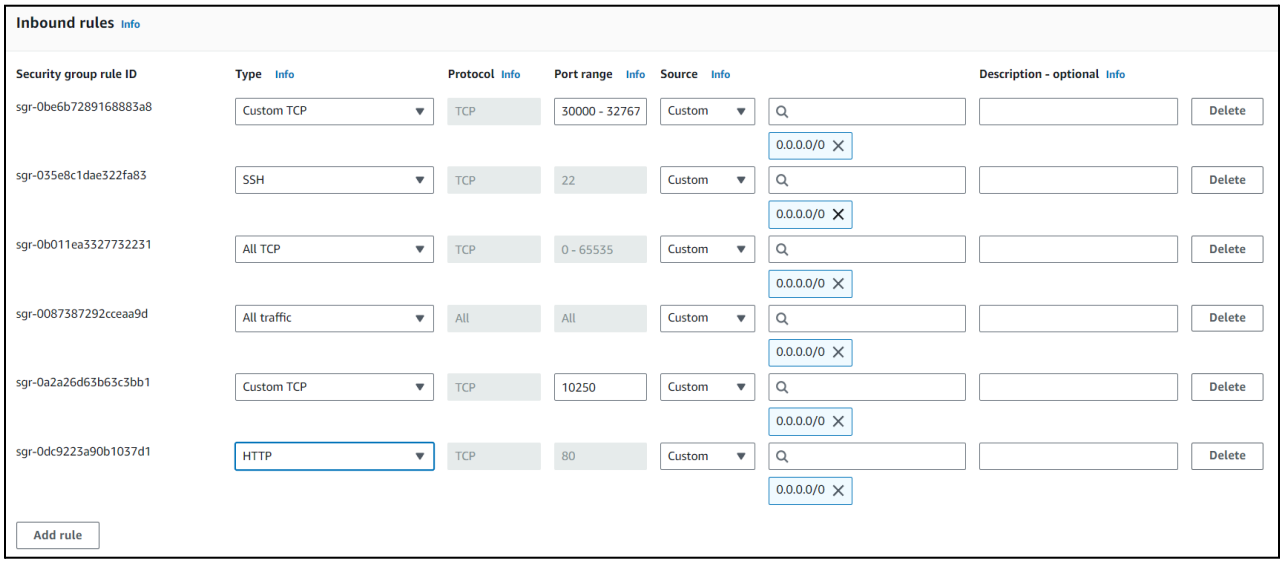
● Moving an application instance from one host to another if there is a shortage of resources in a host, or if the host dies
● Automatically leveraging additional resources made available when a new host is added
to the cluster

Step 1: Create 2 Security Groups for Master and Nodes and add the following inbound rules in those groups

Master:

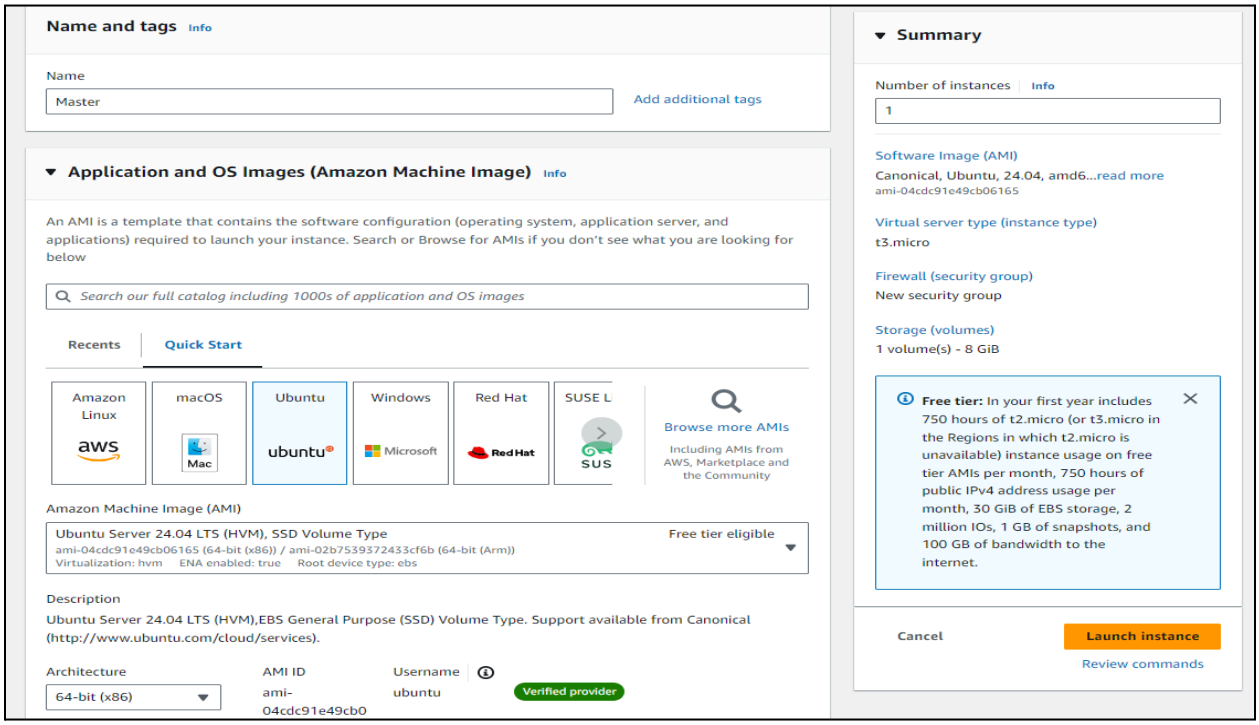


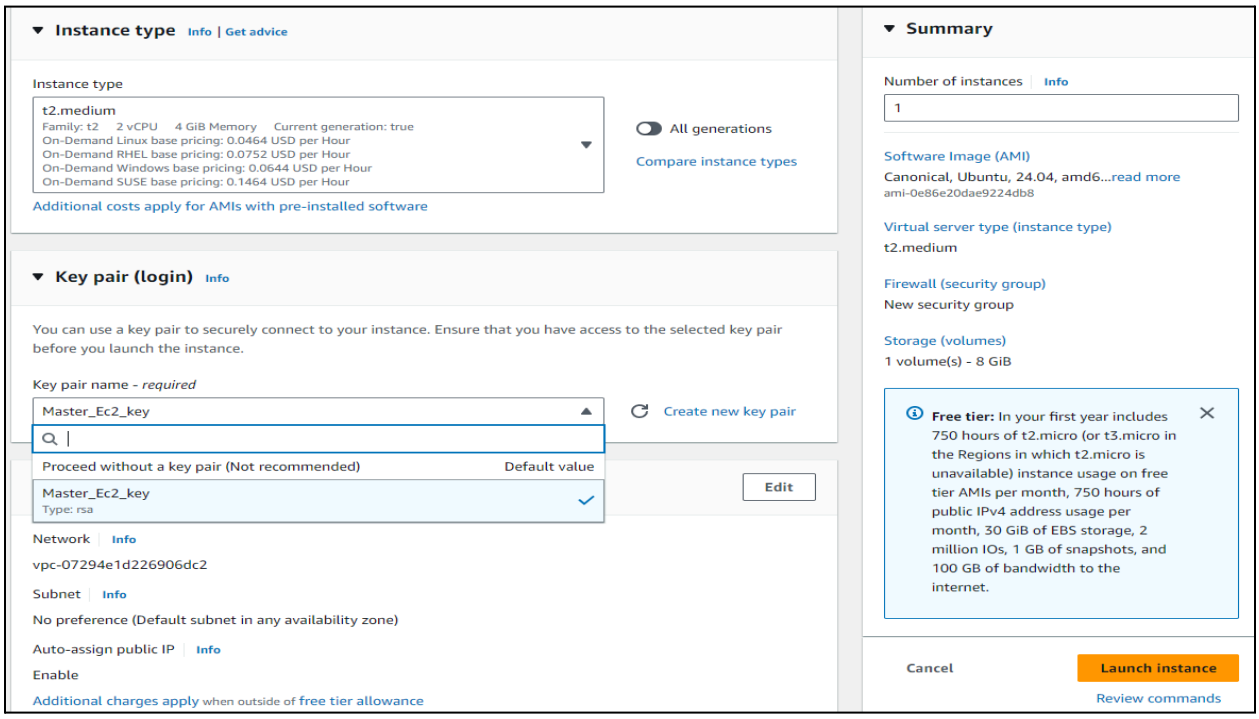
Node:

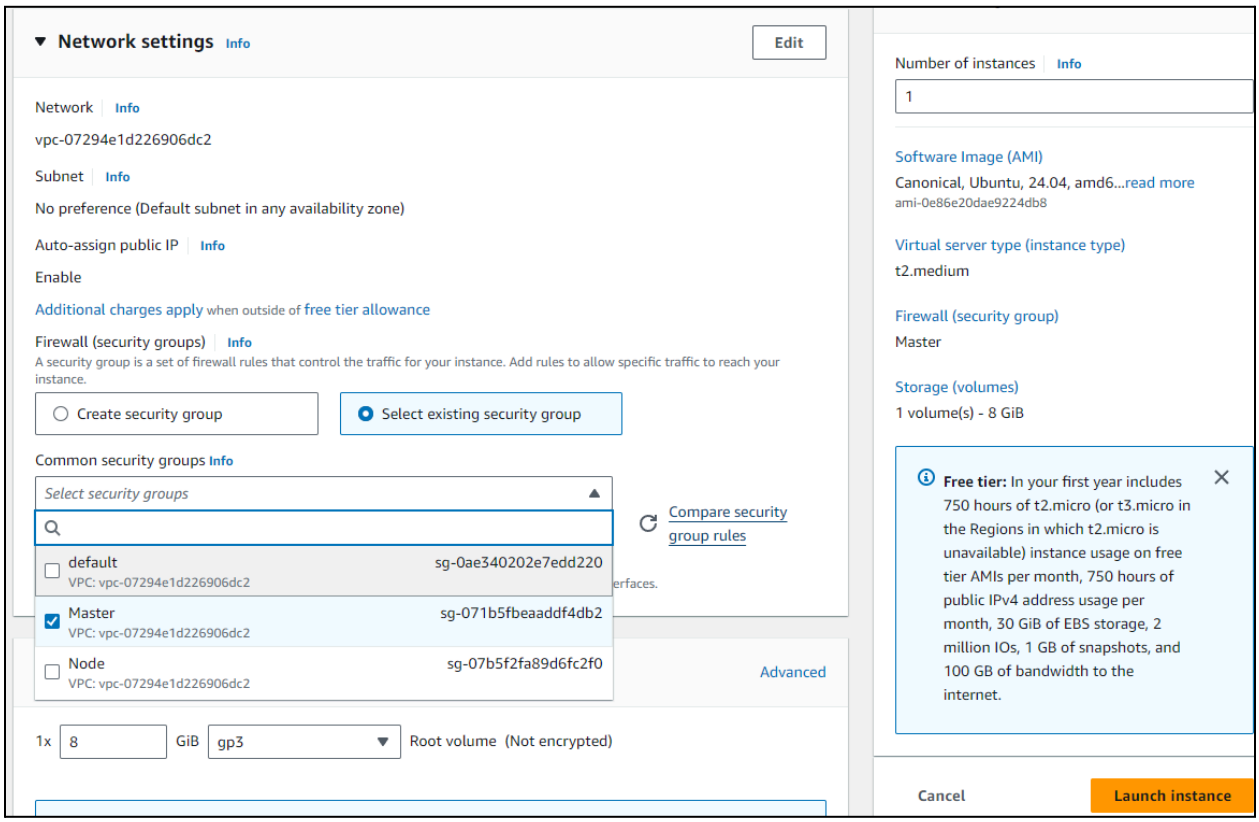


Step 2: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances(1 for Master and 2 for Node).Select Ubuntu as AMI and t2.medium as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.We can use 2 Different keys, 1 for Master and 1 for Node. Also Select Security Groups from the existing.

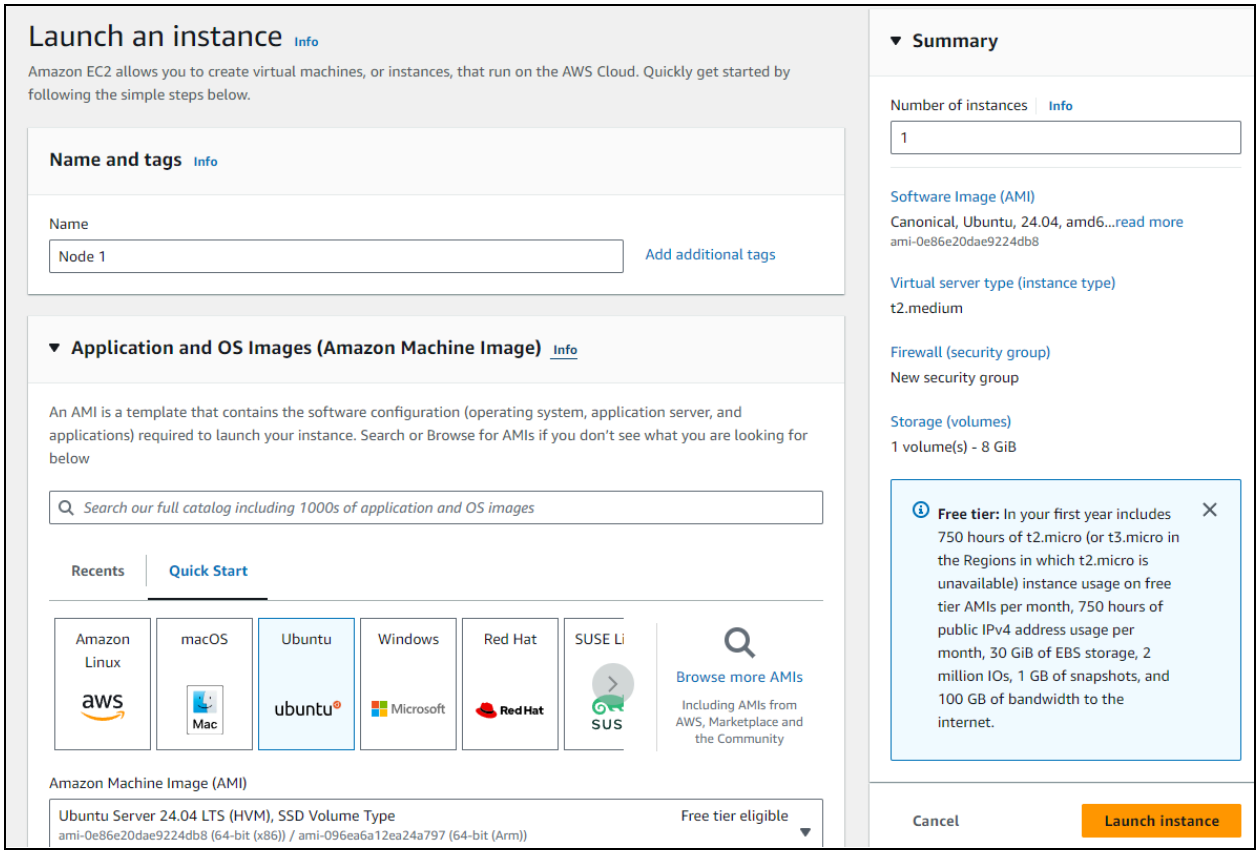
Master:

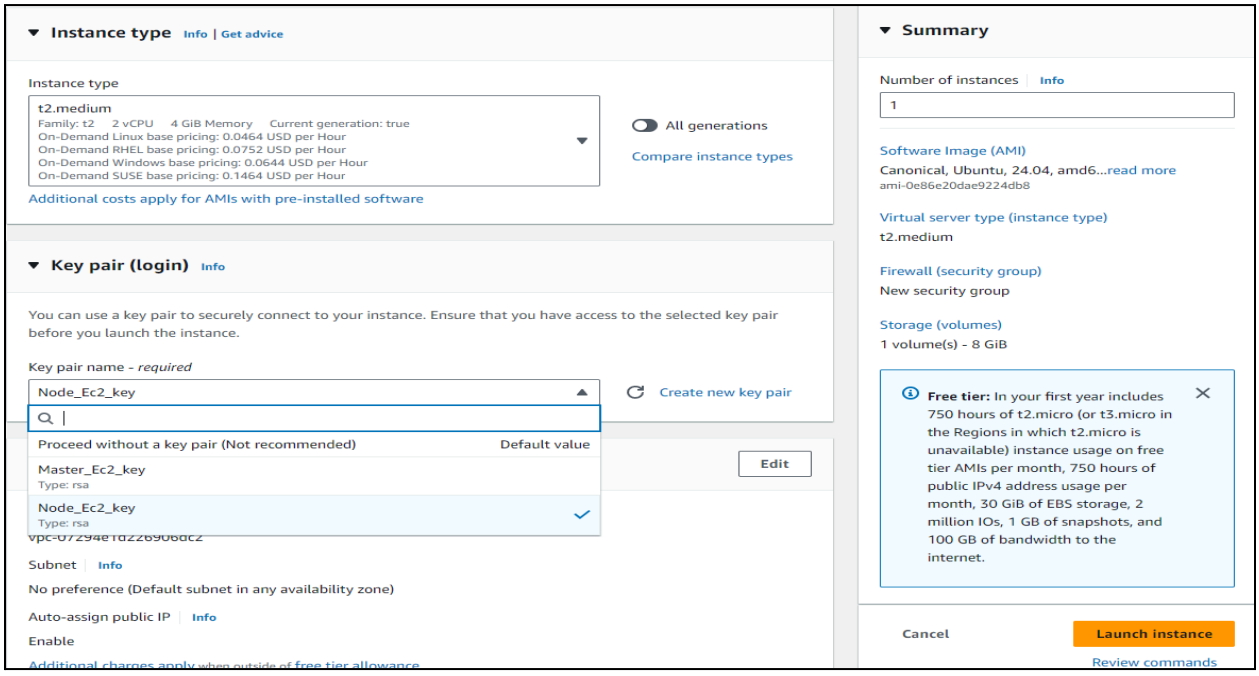


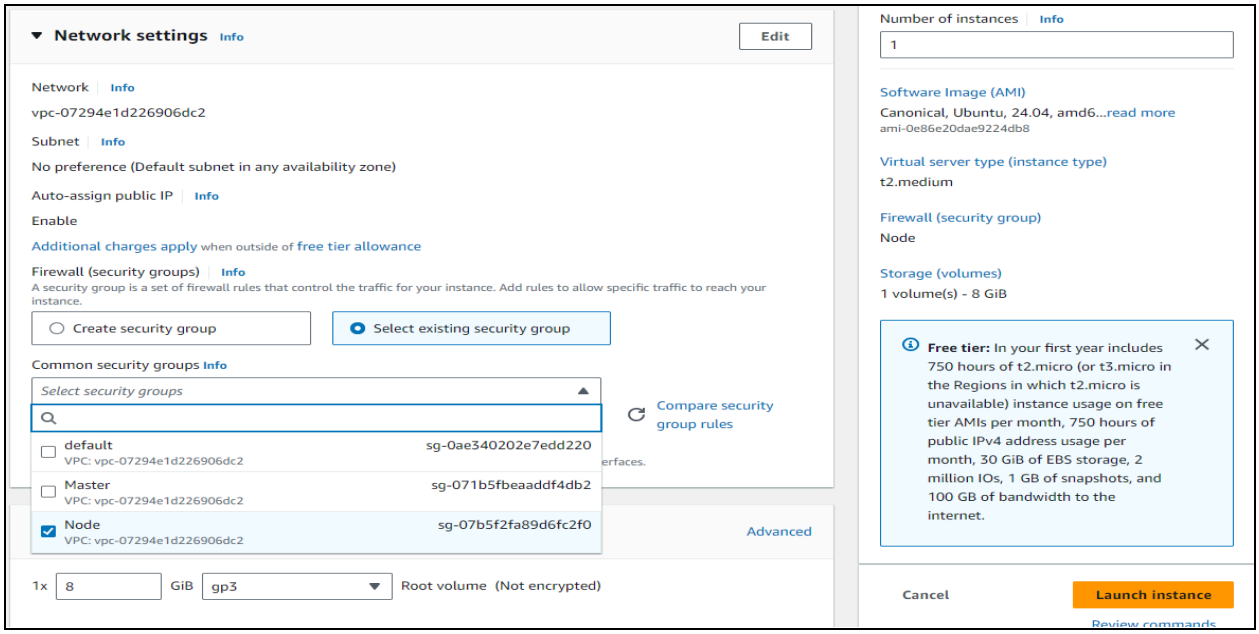


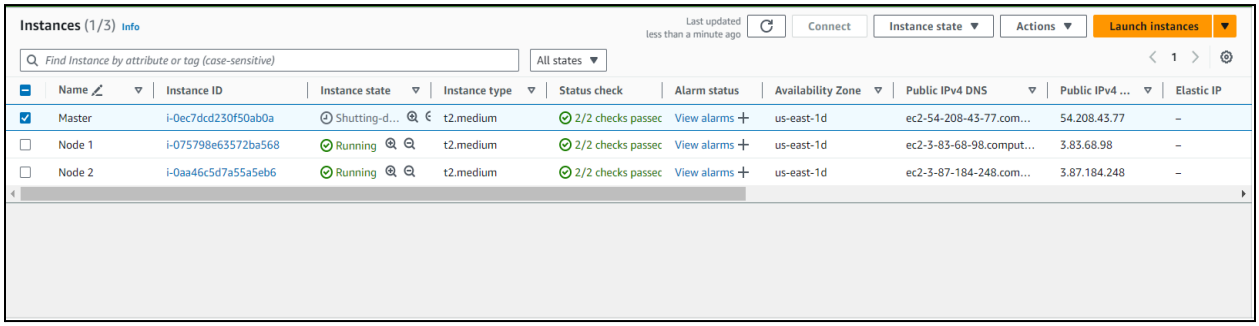

Network settings Info
Edit
Network Info
vpc-07294e1d226906dc2
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Common security groups Info
Select security groups
default
VPC: vpc-07294e1d226906dc2
sg-0ae340202e7edd220
Master
VPC: vpc-07294e1d226906dc2
sg-071b5fbeaaddf4db2
Node
VPC: vpc-07294e1d226906dc2
sg-07b5f2fa89d6fc2f0
Compare security group rules
Advanced
1x
8
GiB
gp3
Root volume (Not encrypted)
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8
Virtual server type (instance type)
t2.medium
Firewall (security group)
Master
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance

Node:


Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
Node 1
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
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
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8
Virtual server type (instance type)
t2.medium
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance

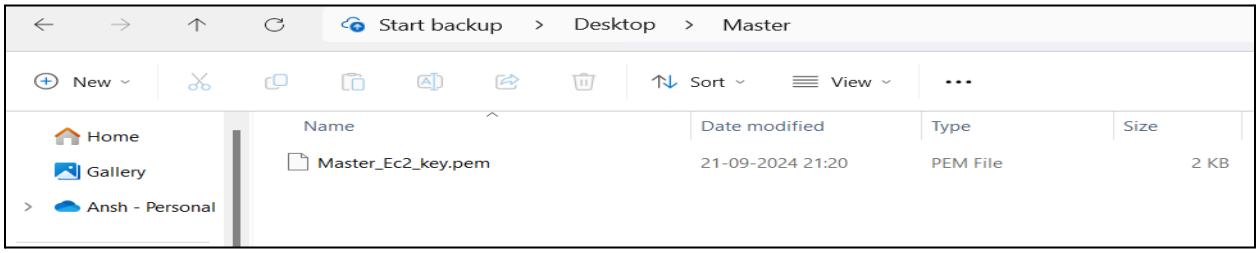

Instance type Info | Get advice
Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Node_Ec2_key
Create new key pair
Proceed without a key pair (Not recommended)
Default value
Master_Ec2_key
Type: rsa
Node_Ec2_key
Type: rsa
Edit
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
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8
Virtual server type (instance type)
t2.medium
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Review commands



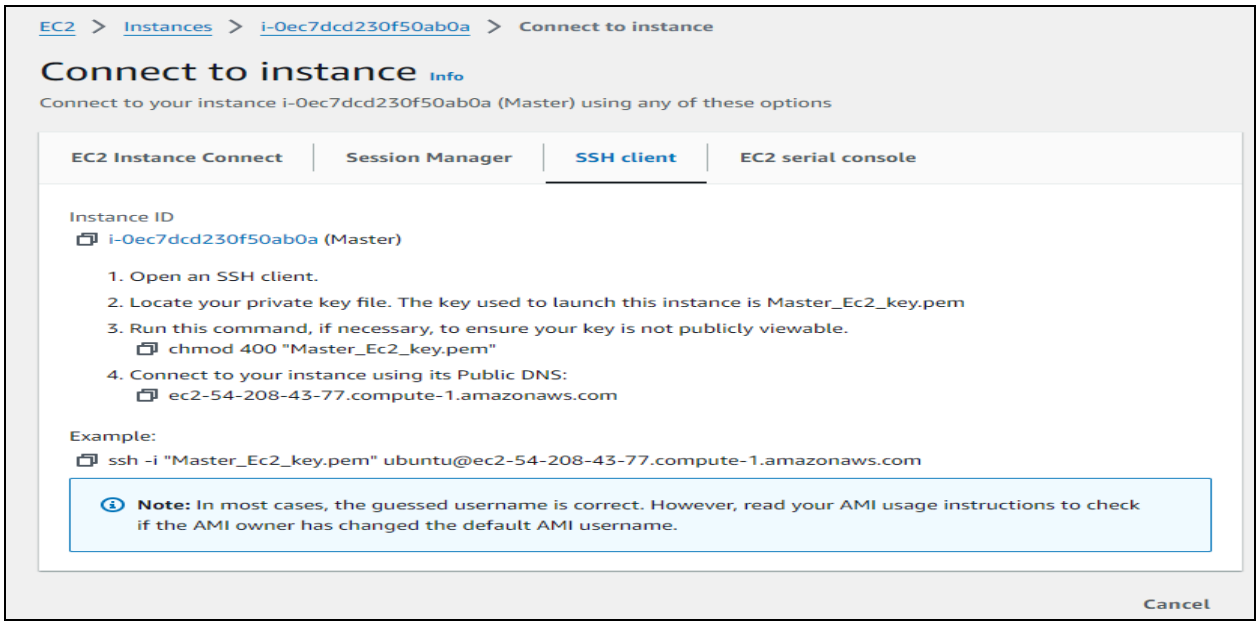


Step 3: Connect the instance and navigate to SSH client and copy the example command. Now open the folder in the terminal 3 times for Master, Node1 & Node 2 where our .pem key is stored and paste the Example command from ssh client (starting with ssh -i .....) in the terminal.

Downloaded Key:

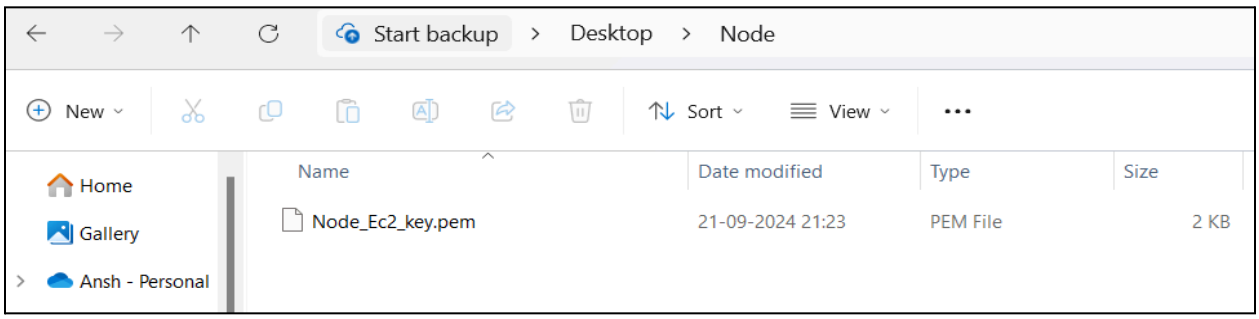


Master:

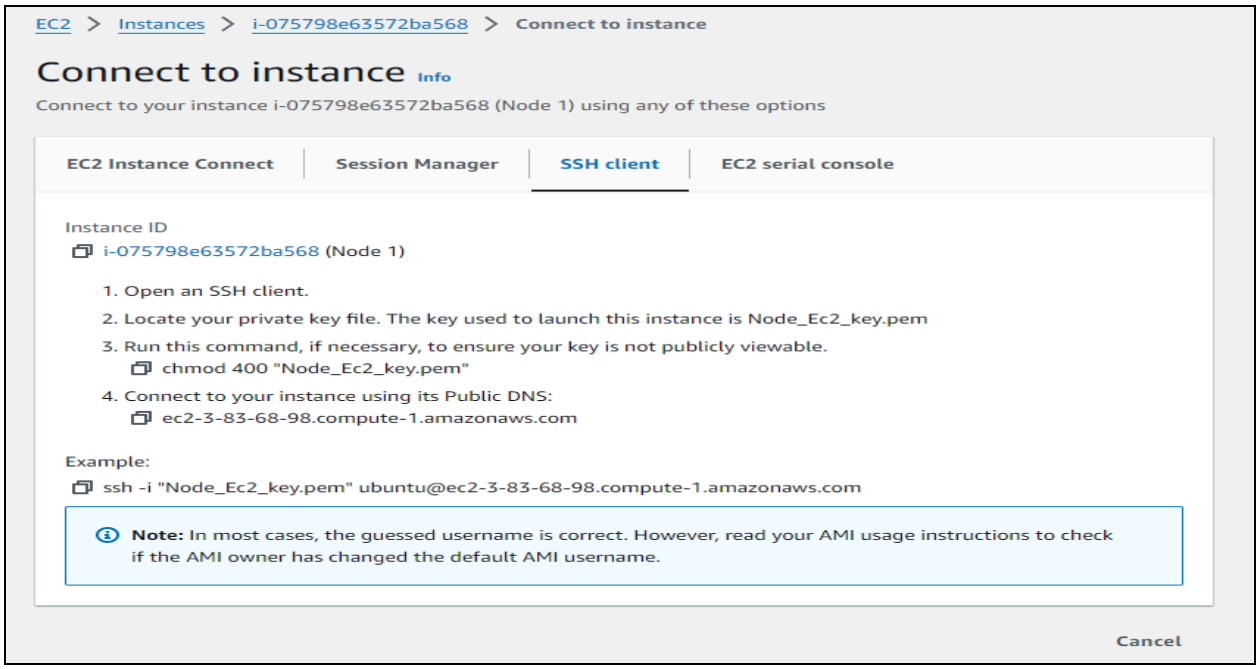



```
PS C:\Users\Ansh\Desktop\Master> ssh -i "Master_Ec2_key.pem" ubuntu@ec2-54-208-43-77.compute-1.amazonaws.com
The authenticity of host 'ec2-54-208-43-77.compute-1.amazonaws.com (54.208.43.77)' can't be established.
ED25519 key fingerprint is SHA256:GEXNsxYEo5wgzQPomUHsm+3oE1vRL4EAEjhm1lggTpU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-208-43-77.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sat Sep 21 16:47:07 UTC 2024

  System load:  0.0               Processes:             115
  Usage of /:   23.1% of 6.71GB   Users logged in:       0
  Memory usage: 7%                IPv4 address for enX0: 172.31.95.244
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


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

Last login: Sat Sep 21 16:28:45 2024 from 18.206.107.28
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-95-244:~$
```

Downloaded Key:

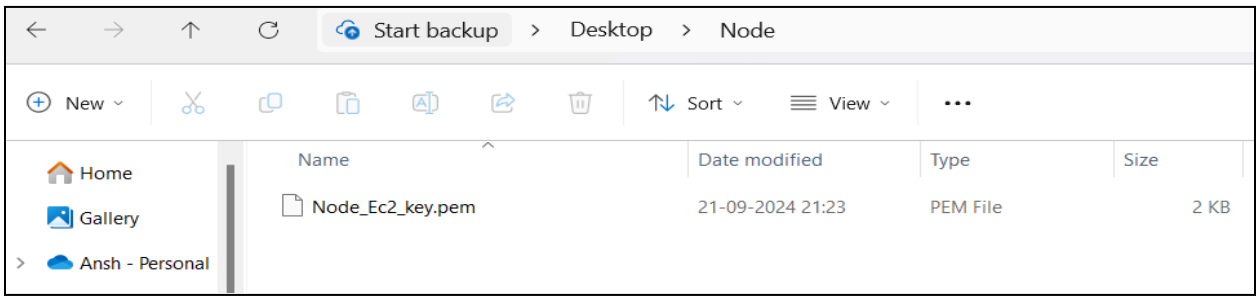


Node 1:

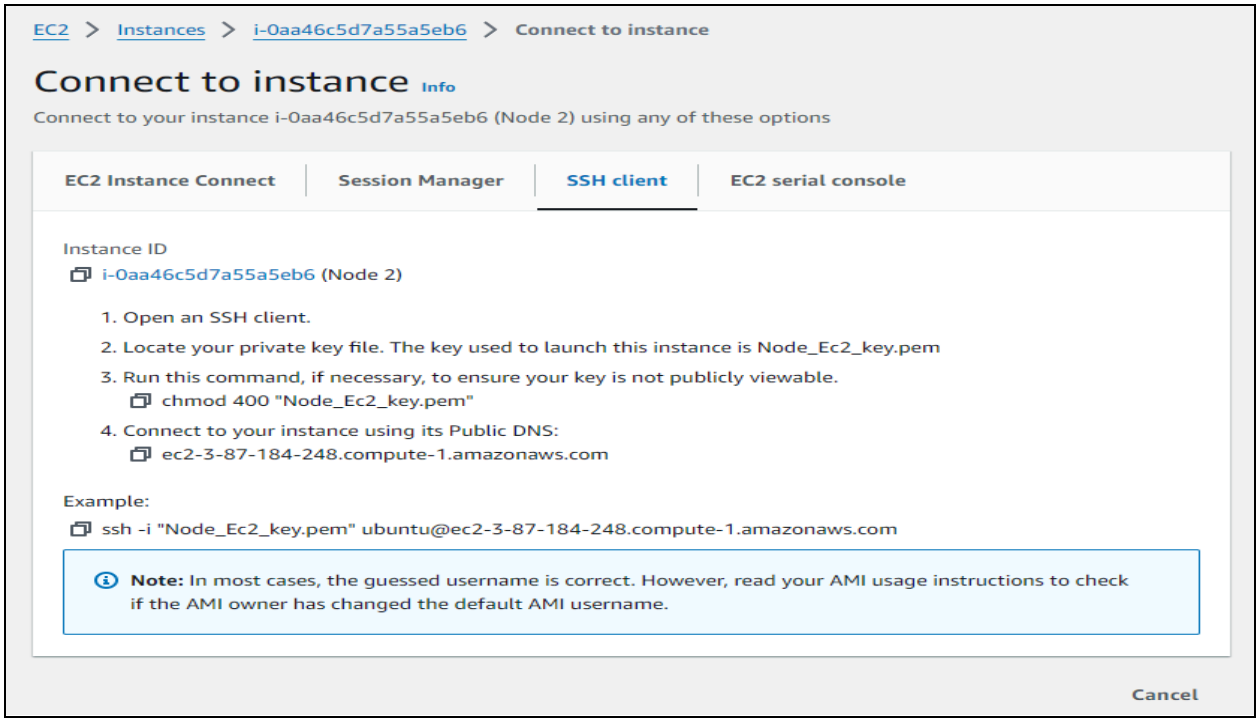



```
PS C:\Users\Ansh\Desktop\Node> ssh -i "Node_Ec2_key.pem" ubuntu@ec2-3-83-68-98.compute-1.amazonaws.com
The authenticity of host 'ec2-3-83-68-98.compute-1.amazonaws.com (3.83.68.98)' can't be established.
ED25519 key fingerprint is SHA256:21QMbe+vHlvpbqWK7g3/dY14ckA4LLH0mijbam4WIvQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-83-68-98.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sat Sep 21 17:31:03 UTC 2024

  System load:  0.0               Processes:             117
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 8%                IPv4 address for enX0: 172.31.84.209
  Swap usage:   0%
```

Downloaded Key:



Node 2:




```
PS C:\Users\Ansh\Desktop\Node> ssh -i "Node_Ec2_key.pem" ubuntu@ec2-3-87-184-248.compute-1.amazonaws.com
The authenticity of host 'ec2-3-87-184-248.compute-1.amazonaws.com (3.87.184.248)' can't be established.
ED25519 key fingerprint is SHA256:cwvHL/f3y1isWTr/hU72bPMq6+a03thKQtCkQfII2gg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-87-184-248.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sat Sep 21 17:32:30 UTC 2024

  System load:  0.08              Processes:             113
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for enX0: 172.31.87.15
  Swap usage:   0%
```

Step 4: Run on Master,Node 1,and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
- sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

```
ubuntu@ip-172-31-95-244:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOwQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVMtSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYfOS8QcixNzHUYnb6emjlANyEVlZzeqo7XKl7
UrwV5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N
cmhh4HgeQscQHYgOJjjDVfoY5MucvglbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzlgaRL5Z4POXDDZTlIQ/El58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtl/94VFYVJOleAv8W92KdgDkhTcTD
G7c0tIkVEKNUq48b3aQ64NOZQW7fVjfoKwEZdOqPE72Pa45jrZzvUFxSpdiNk2tZ
```

```
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:51 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:52 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 29.1 MB in 4s (7658 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring
ection in apt-key(8) for details.
ubuntu@ip-172-31-95-244:~$
```

- sudo apt-get update
- sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-95-244:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg
ection in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin lib
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-
```

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-95-244:~$
```

- sudo mkdir -p /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {
  "exec-opts": ["native.cgroupdriver=systemd"]
  }
  EOF

```
ubuntu@ip-172-31-95-244:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-95-244:~$
```

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

```
ubuntu@ip-172-31-95-244:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-95-244:~$
```

Step 5: Run the below command to install Kubernetes.

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

```
ubuntu@ip-172-31-95-244:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-95-244:~$
```

- sudo apt-get update
- sudo apt-get install -y kubelet kubeadm kubectl
- sudo apt-mark hold kubelet kubeadm kubectl

error:

```
ubuntu@ip-172-31-95-244:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
ubuntu@ip-172-31-95-244:~$
```

To solve the error:
Added **sudo mkdir -p /etc/apt/keyrings** in the previous command.

```
ubuntu@ip-172-31-95-244:~$ 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 noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 1s (10.1 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/
ection in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 139 not upgraded.
Need to get 87.4 MB of archives.
```

```
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-95-244:~$
```

- sudo systemctl enable --now kubelet
- sudo apt-get install -y containerd

```
ubuntu@ip-172-31-95-244:~$ sudo systemctl enable --now kubelet
sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-pl
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 139 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 ru
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 co
Fetched 47.2 MB in 1s (76.6 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
```

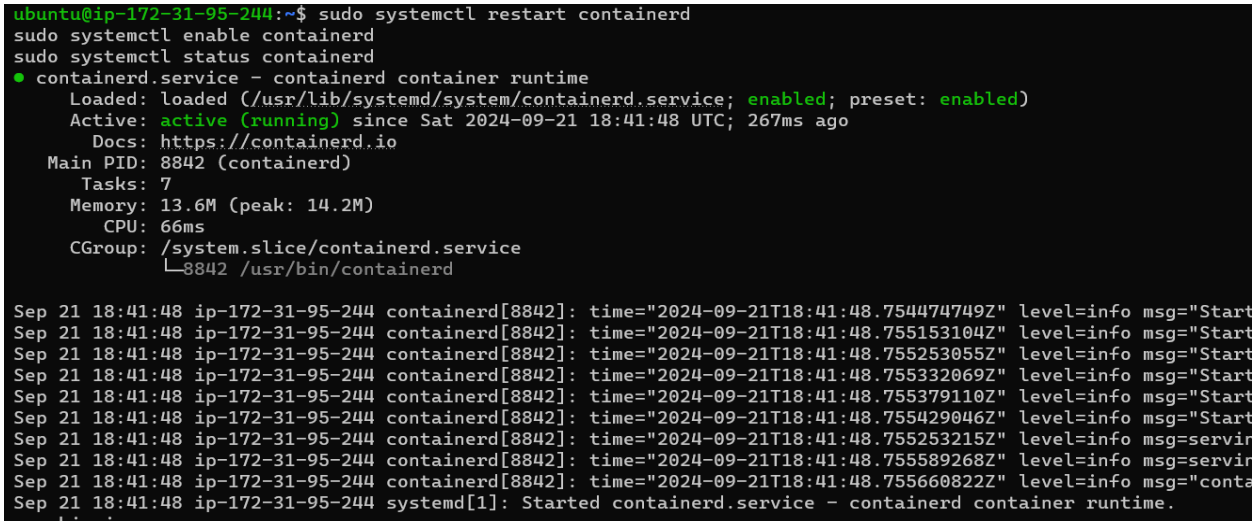
```
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-95-244:~$
```

- sudo mkdir -p /etc/containerd
- sudo containerd config default | sudo tee /etc/containerd/config.toml

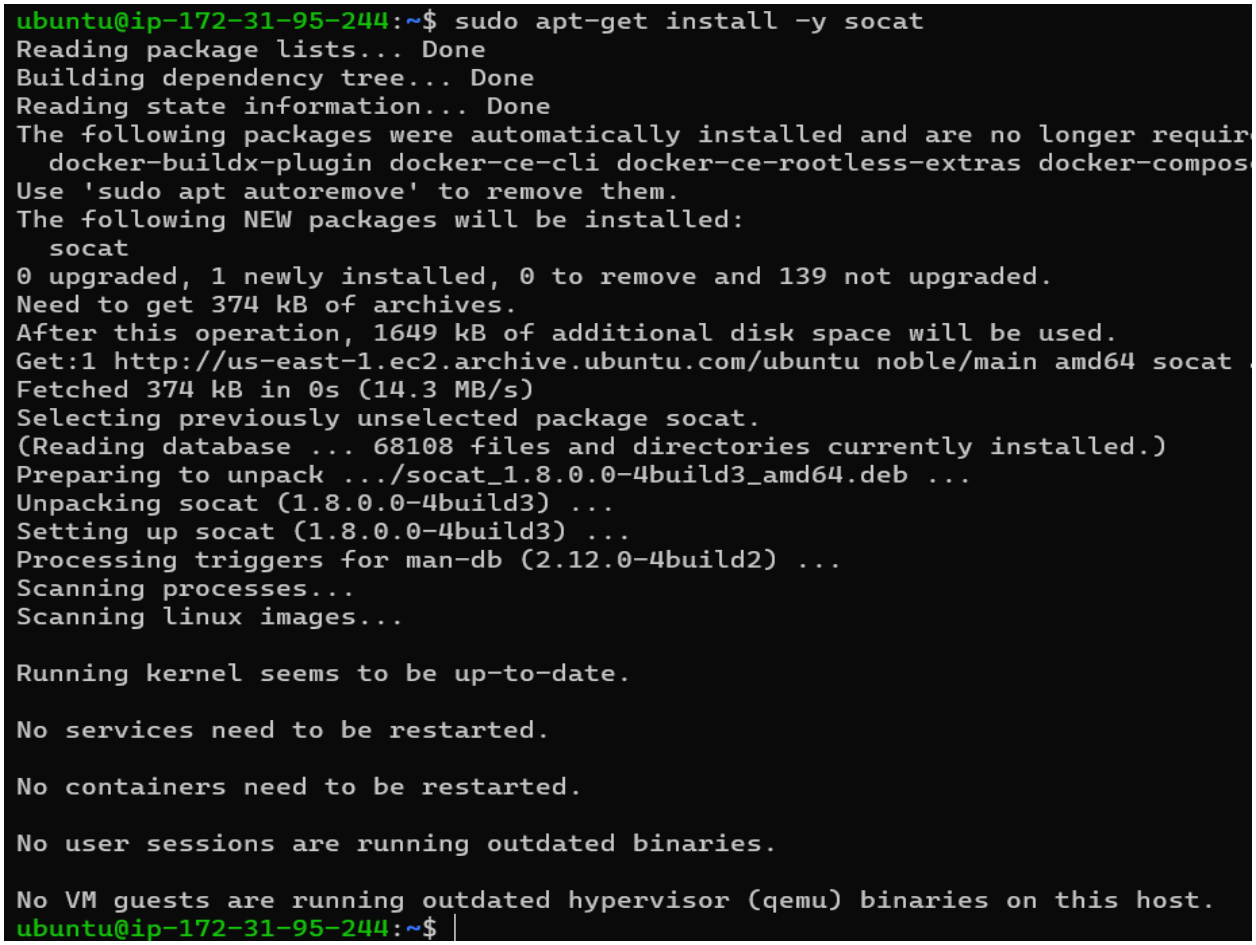
```
ubuntu@ip-172-31-95-244:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

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

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

```
  [stream_processors."io.containerd.ocicrypt.decoder.v1.tar.gzip"]
    accepts = ["application/vnd.oci.image.layer.v1.tar+gzip+encrypted"]
    args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
    env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt
    path = "ctd-decoder"
    returns = "application/vnd.oci.image.layer.v1.tar+gzip"

[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
ubuntu@ip-172-31-95-244:~$
```

- sudo systemctl restart containerd
- sudo systemctl enable containerd
- sudo systemctl status containerd



- sudo apt-get install -y socat



Step 6: Initialize the Kubecluster .Now Perform this Command only for Master.

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

```
ubuntu@ip-172-31-95-244:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0921 18:47:47.470667    9264 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container ru
used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-95-244 kubernetes kubernetes.default kubernetes.default.sv
.local] and IPs [10.96.0.1 172.31.95.244]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-95-244 localhost] and IPs [172.31.95.244 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-95-244 localhost] and IPs [172.31.95.244 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
```

```
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.95.244:6443 --token kzfth2.ug3970lp3qeeieb4 \
        --discovery-token-ca-cert-hash sha256:dec27d33f1bfd1dca7a50caa2c05d4cad1d0a18aa88ad75c7ea83f15c529f4ca
ubuntu@ip-172-31-95-244:~$ mkdir -p $HOME/.kube
```

Copy the kudeadm join any number of worker nodes command to use it later for joining Node 1 and Node 2 with master

- 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-172-31-95-244:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-95-244:~$
```

Step 7: Now Run the command **kubectl get nodes** to see the nodes before executing Join command on nodes.

```
ubuntu@ip-172-31-95-244:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE     VERSION
ip-172-31-95-244   NotReady   control-plane   2m52s   v1.31.1
ubuntu@ip-172-31-95-244:~$
```

Step 8: Now Run the following command on Node 1 and Node 2 to Join to master.

- sudo kubeadm join 172.31.95.244:6443 --token kzfth2.ug3970lp3qeeieb4\
  --discovery-token-ca-cert-hash
  sha256:dec27d33f1bfd1dca7a50caa2c05d4cad1d0a18aa88ad75c7ea83f15c529f4ca

Node 1:

```
ubuntu@ip-172-31-84-209:~$ sudo kubeadm join 172.31.95.244:6443 --token kzfth2.ug3970lp3qeeieb4 \
        --discovery-token-ca-cert-hash sha256:dec27d33f1bfd1dca7a50caa2c05d4cad1d0a18aa88ad75c7ea83f15c529f4ca
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001303324s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-84-209:~$
```

Node 2:

```
ubuntu@ip-172-31-87-15:~$ sudo kubeadm join 172.31.95.244:6443 --token kzfth2.ug3970lp3qeeieb4 \
        --discovery-token-ca-cert-hash sha256:dec27d33f1bfd1dca7a50caa2c05d4cad1d0a18aa88ad75c7ea83f15c529f4ca
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.501340478s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-87-15:~$
```

Step 9: Now Run the command **kubectl get nodes** to see the nodes after executing Join command on nodes.

```
ubuntu@ip-172-31-95-244:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE     VERSION
ip-172-31-84-209   NotReady   <none>          113s    v1.31.1
ip-172-31-87-15    NotReady   <none>          65s     v1.31.1
ip-172-31-95-244   NotReady   control-plane   8m30s   v1.31.1
ubuntu@ip-172-31-95-244:~$
```

Step 10: Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.

- kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

```
ubuntu@ip-172-31-95-244:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-95-244:~$
```

- sudo systemctl status kubelet

```
ubuntu@ip-172-31-95-244:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
    Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Sat 2024-09-21 18:48:07 UTC; 9min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 9932 (kubelet)
      Tasks: 10 (limit: 4676)
     Memory: 32.4M (peak: 32.9M)
        CPU: 9.201s
     CGroup: /system.slice/kubelet.service
             └─9932 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/k

Sep 21 18:57:49 ip-172-31-95-244 kubelet[9932]: I0921 18:57:49.455496    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:49 ip-172-31-95-244 kubelet[9932]: I0921 18:57:49.457800    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:49 ip-172-31-95-244 kubelet[9932]: I0921 18:57:49.468016    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:49 ip-172-31-95-244 kubelet[9932]: I0921 18:57:49.478177    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:52 ip-172-31-95-244 kubelet[9932]: I0921 18:57:52.466936    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:52 ip-172-31-95-244 kubelet[9932]: E0921 18:57:52.467071    9932 pod_workers.go:1301] "Error syncing pod,
Sep 21 18:57:58 ip-172-31-95-244 kubelet[9932]: I0921 18:57:58.981488    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:58 ip-172-31-95-244 kubelet[9932]: E0921 18:57:58.981598    9932 pod_workers.go:1301] "Error syncing pod,
Sep 21 18:57:59 ip-172-31-95-244 kubelet[9932]: I0921 18:57:59.349434    9932 scope.go:117] "RemoveContainer" containe
Sep 21 18:57:59 ip-172-31-95-244 kubelet[9932]: E0921 18:57:59.349579    9932 pod_workers.go:1301] "Error syncing pod,
lines 1-23/23 (END)
```

- Now Run command **kubectl get nodes** -o wide we can see Status is ready.

```
ubuntu@ip-172-31-95-244:~$ kubectl get nodes -o wide
NAME              STATUS   ROLES          AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE          KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-84-209  Ready    <none>         4m24s  v1.31.1   172.31.84.209   <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws   containerd://1.7.12
ip-172-31-87-15   Ready    <none>         3m36s  v1.31.1   172.31.87.15    <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws   containerd://1.7.12
ip-172-31-95-244  Ready    control-plane  11m    v1.31.1   172.31.95.244   <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws   containerd://1.7.12
ubuntu@ip-172-31-95-244:~$
```

The Roles are not yet assigned to the Nodes

```
ubuntu@ip-172-31-95-244:~$ kubectl get nodes
NAME              STATUS   ROLES          AGE    VERSION
ip-172-31-84-209  Ready    <none>         5m27s  v1.31.1
ip-172-31-87-15   Ready    <none>         4m39s  v1.31.1
ip-172-31-95-244  Ready    control-plane  12m    v1.31.1
ubuntu@ip-172-31-95-244:~$
```

- **Rename to Node 1:** kubectl label node ip-172-31-28-117 kubernetes.io/role=Node1
- **Rename to Node 2:** kubectl label node ip-172-31-18-135 kubernetes.io/role=Node2

```
ubuntu@ip-172-31-95-244:~$ kubectl label node ip-172-31-84-209 kubernetes.io/role=Node1
node/ip-172-31-84-209 labeled
ubuntu@ip-172-31-95-244:~$ kubectl label node ip-172-31-95-15 kubernetes.io/role=Node2
Error from server (NotFound): nodes "ip-172-31-95-15" not found
ubuntu@ip-172-31-95-244:~$ kubectl label node ip-172-31-87-15 kubernetes.io/role=Node2
node/ip-172-31-87-15 labeled
ubuntu@ip-172-31-95-244:~$
```

- Run **kubectl get nodes** to check if roles are assigned now to the nodes

```
ubuntu@ip-172-31-95-244:~$ kubectl get nodes
NAME              STATUS   ROLES          AGE    VERSION
ip-172-31-84-209  Ready    Node1          8m57s  v1.31.1
ip-172-31-87-15   Ready    Node2          8m9s   v1.31.1
ip-172-31-95-244  Ready    control-plane  15m    v1.31.1
ubuntu@ip-172-31-95-244:~$
```

ADVANCE DEVOPS EXP-4

ANSH SARFARE **D15A/50**

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

Theory:

Kubernetes, originally developed by Google, is an open-source container orchestration platform. It automates the deployment, scaling, and management of containerized applications, ensuring high availability and fault tolerance.

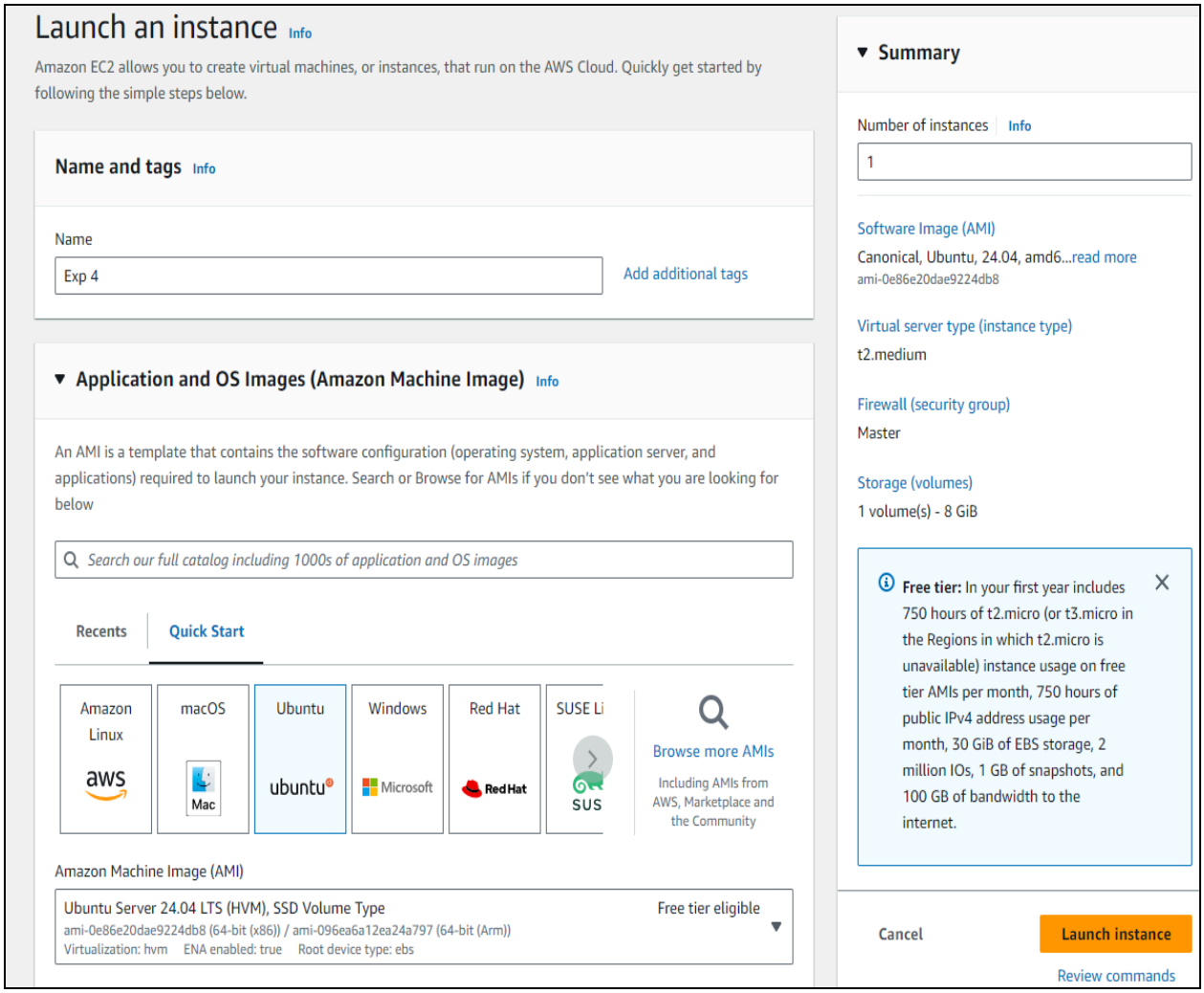
Kubernetes is now the industry standard for container orchestration and
is governed by the Cloud Native Computing Foundation (CNCF), with contributions from major cloud and software providers like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

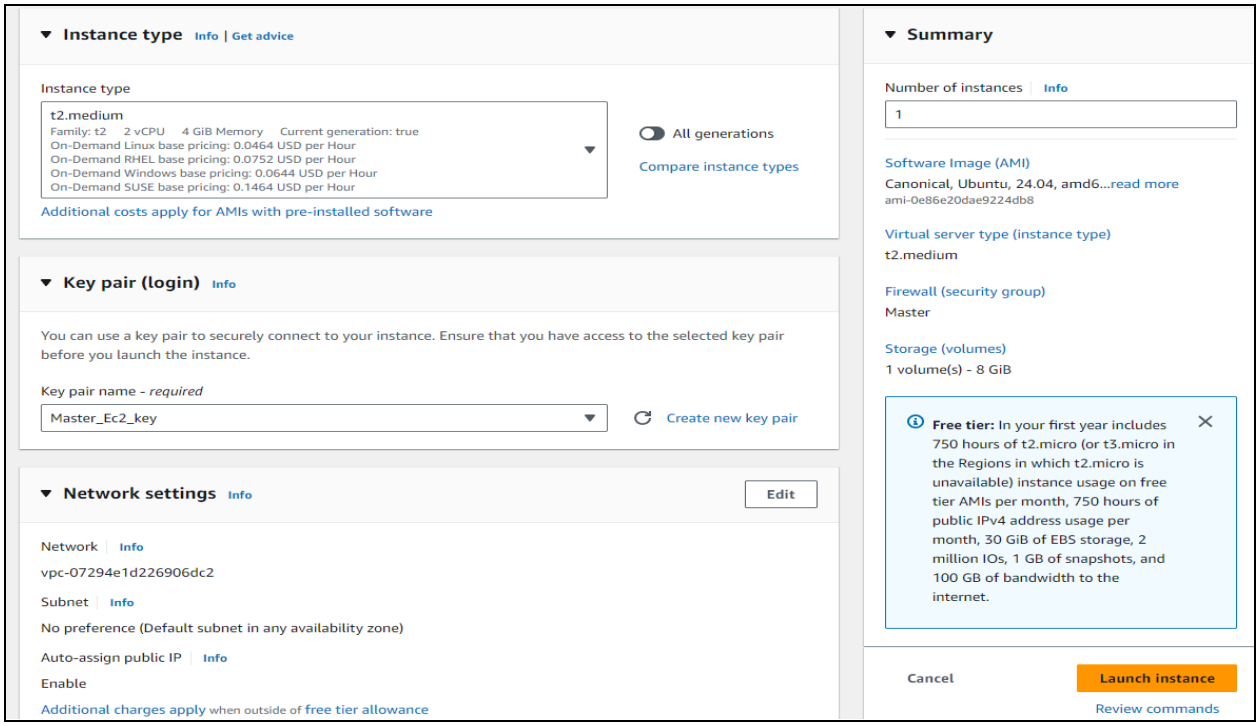
Kubernetes Deployment: Is a resource in Kubernetes that provides declarative updates for Pods and ReplicaSets. With a Deployment, you can define how many replicas of a pod should run, roll out new versions of an application, and roll back to previous versions if necessary. It ensures that the desired number of pod replicas are running at all times.

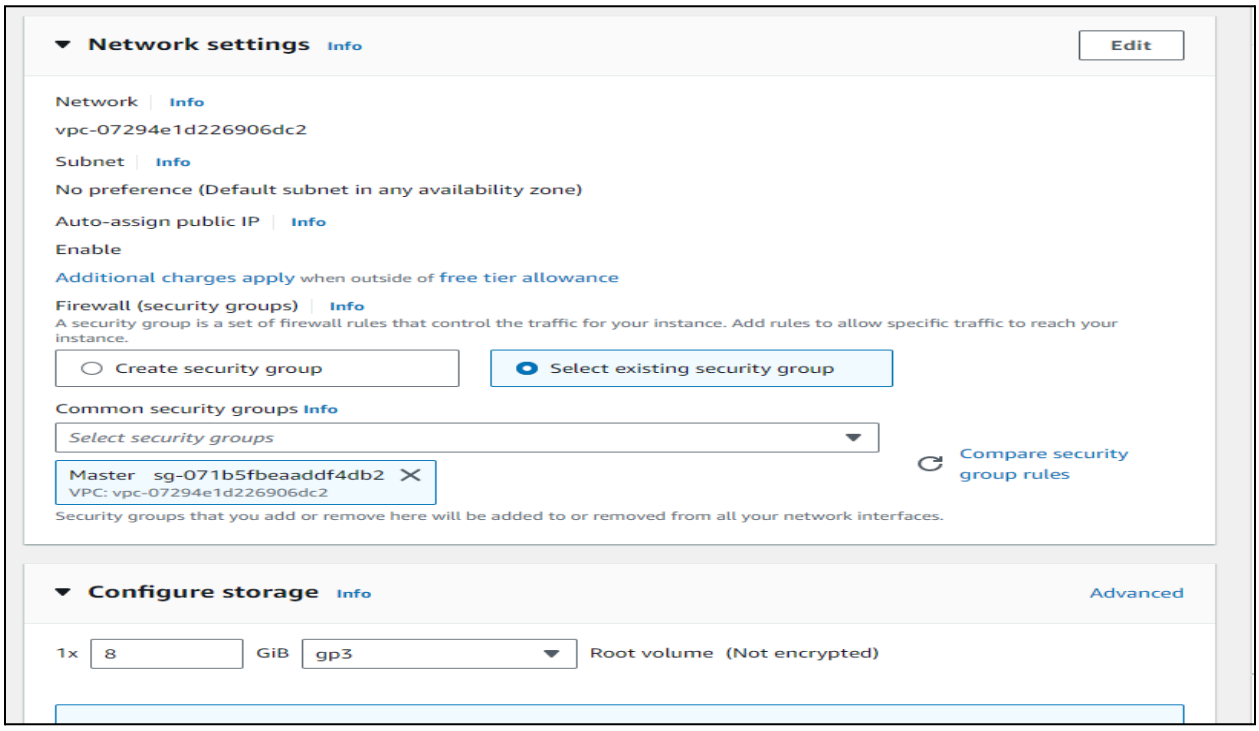
Necessary Requirements:
● EC2 Instance: The experiment required launching a t2.medium EC2 instance with 2 CPUs, as
Kubernetes demands sufficient resources for effective functioning.
● Minimum Requirements:
○ Instance Type: t2.medium
○ CPUs: 2
○ Memory: Adequate for container orchestration.

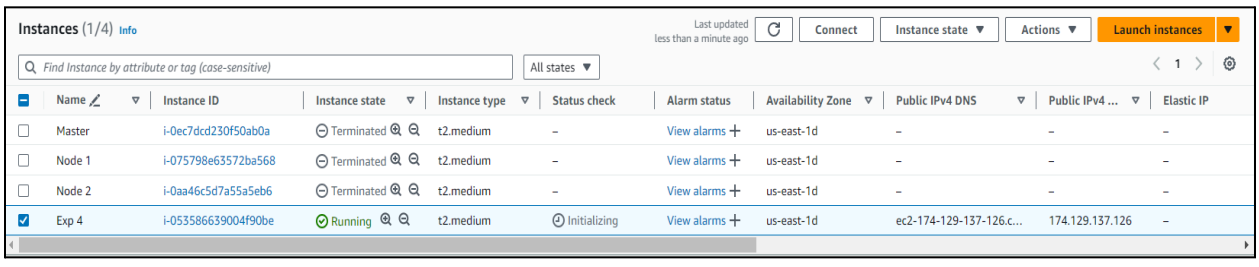
This ensured that the Kubernetes cluster had the necessary resources to function smoothly.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance.Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.




Instance type Info | Get advice
Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Master_Ec2_key
Create new key pair
Network settings Info
Edit
Network Info
vpc-07294e1d226906dc2
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
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8
Virtual server type (instance type)
t2.medium
Firewall (security group)
Master
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Review commands


Network settings Info
Edit
Network Info
vpc-07294e1d226906dc2
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
Common security groups Info
Select security groups
Master sg-071b5fbeaaddf4db2
VPC: vpc-07294e1d226906dc2
Compare security group rules
Security groups that you add or remove here will be added to or removed from all your network interfaces.
Configure storage Info
Advanced
1x 8 GiB gp3 Root volume (Not encrypted)


Instances (1/4) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
Master i-0ec7dcd230f50ab0a Terminated t2.medium – View alarms us-east-1d – – –
Node 1 i-075798e63572ba568 Terminated t2.medium – View alarms us-east-1d – – –
Node 2 i-0aa46c5d7a55a5eb6 Terminated t2.medium – View alarms us-east-1d – – –
Exp 4 i-053586639004f90be Running t2.medium Initializing View alarms us-east-1d ec2-174-129-137-126.c... 174.129.137.126 –

Step 2: After creating the instance click on Connect the instance and navigate to SSH Client. Copy the example command. Open your key Folder in terminal and paste the command there.

```
PS C:\Users\Ansh\Desktop\Master> ssh -i "Master_Ec2_key.pem" ubuntu@ec2-174-129-137-126.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sat Sep 21 20:05:43 UTC 2024

  System load:  0.0               Processes:             115
  Usage of /:   22.9% of 6.71GB   Users logged in:       1
  Memory usage: 5%                IPv4 address for enX0: 172.31.89.118
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


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

Last login: Sat Sep 21 19:59:54 2024 from 110.224.118.148
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 3: Run the below commands to install and setup Docker.

- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
- sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

```
ubuntu@ip-172-31-89-118:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-89-118:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOwQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVMtSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYfOS8QcixNzHUYnb6emjlANyEVlZzeqo7XKl7
UrwV5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N
cmhh4HgeQscQHYgOJjjDVfoY5MucvglbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzlgaRL5Z4POXDDZTlIQ/El58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtl/94VFYVJOleAv8W92KdgDkhTcTD
G7c0tIkVEKNUq48b3aQ64NOZQW7fVjfoKwEZdOqPE72Pa45jrZzvUFxSpdiNk2tZ
XYukHjlxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTeOnmDuDYwr9/obA8t016Yljj
q5rdkywPf4JF8mXUW5eCN1vAFHxeg9ZWemhBtQmGxXnw9M+z6hWwc6ahmwARAQAB
tCtEb2NrZXIgUmVsZWFzZSAoQ0UgZGViKSA8ZG9ja2VyQGRvY2tlci5jb20+iQI3
BBMBCgAhBQJYrefAAhsvBQsJCAcDBRUKCQgLBRYCAwEAAh4BAheAAAoJEI2BgDwO
v82IsskP/iQZo68flDQmNvn8X5XTd6RRaUH33kXYXquT6NkHJciS7E2gTJmqvMqd
tI4mNYHCSEYxI5qpcYV5YqX9P6+Ko+vozo4nseUQLPH/ATQ4qL0Zok+1jkag3Lgk
```

```
ubuntu@ip-172-31-89-118:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
```

```
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.1 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 29.1 MB in 4s (7215 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.g
EPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-89-118:~$
```

- sudo apt-get update
- sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-89-118:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trus
EPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plu
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-c
  slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 139 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /us
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/li
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-89-118:~$
```

- sudo mkdir -p /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {
  "exec-opts": ["native.cgroupdriver=systemd"]
  }
  EOF

```
ubuntu@ip-172-31-89-118:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-89-118:~$
```

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

```
ubuntu@ip-172-31-89-118:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-89-118:~$
```

Step 4: Run the below command to install Kubernetes.

- sudo mkdir -p /etc/apt/keyrings
- curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
- echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-89-118:~$ sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.l
ist.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-89-118:~$
```

- sudo apt-get update
- sudo apt-get install -y kubelet kubeadm kubectl
- sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-89-118:~$ 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 noble InRelease
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.3
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.3
Fetched 6051 B in 1s (10.2 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in leg
EPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
```

```
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-89-118:~$
```

- sudo systemctl enable --now kubelet
- sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
ubuntu@ip-172-31-89-118:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0921 20:17:57.322813    4728 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container
ed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/conta
rd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/co
ainerd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are
n make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-89-118:~$
```

Now We have got an error.
So we have to perform some additional commands as follows.

- sudo apt-get install -y containerd

```
ubuntu@ip-172-31-89-118:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libs
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 139 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0u
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7
Fetched 47.2 MB in 1s (83.0 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
```

```
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-89-118:~$
```

- sudo mkdir -p /etc/containerd
- sudo containerd config default | sudo tee /etc/containerd/config.toml

```
ubuntu@ip-172-31-89-118:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

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

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

```
[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
ubuntu@ip-172-31-89-118:~$
```

- sudo systemctl restart containerd
- sudo systemctl enable containerd
- sudo systemctl status containerd

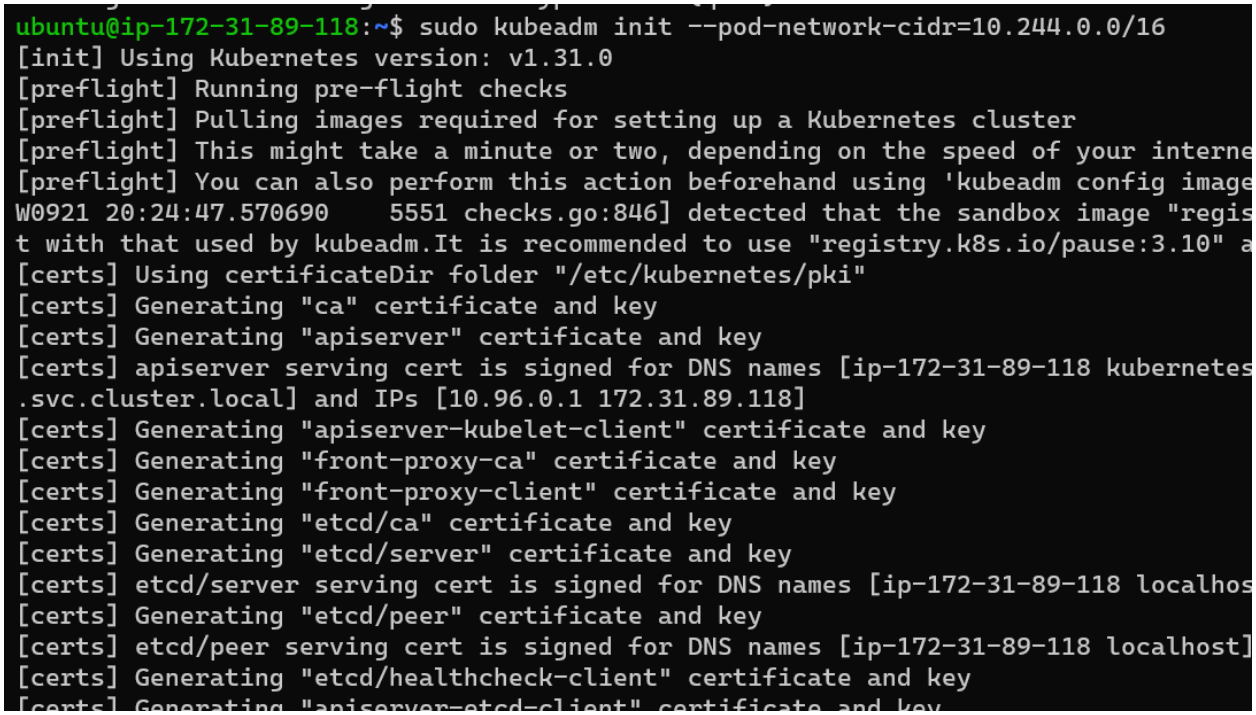
```
ubuntu@ip-172-31-89-118:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-21 20:23:08 UTC; 288ms ago
       Docs: https://containerd.io
   Main PID: 5321 (containerd)
      Tasks: 7
     Memory: 13.5M (peak: 13.8M)
        CPU: 64ms
     CGroup: /system.slice/containerd.service
             └─5321 /usr/bin/containerd

Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302792464Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302839912Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302890921Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302916611Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302925128Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.302931755Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.303011125Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.303086007Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 containerd[5321]: time="2024-09-21T20:23:08.303196108Z" level=info msg
Sep 21 20:23:08 ip-172-31-89-118 systemd[1]: Started containerd.service - containerd container runtime.
lines 1-21/21 (END)
```

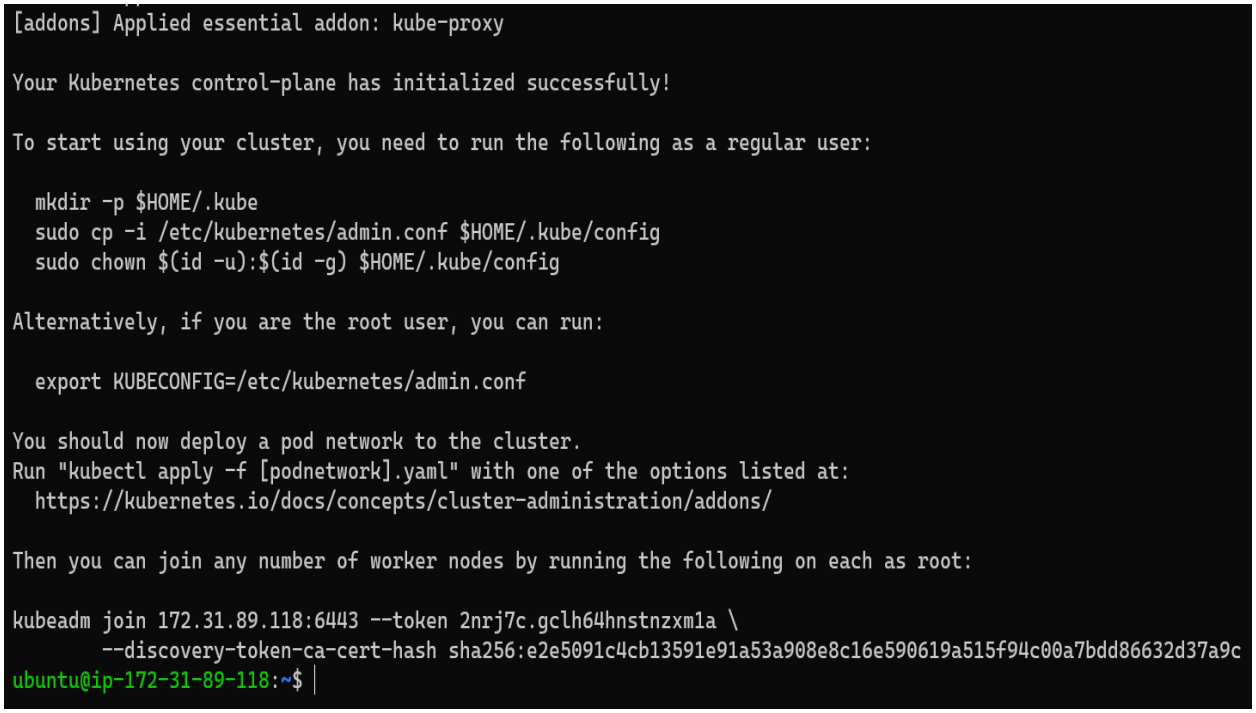
- sudo apt-get install -y socat

```
ubuntu@ip-172-31-89-118:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer requ
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-comp
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 139 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 soca
Fetched 374 kB in 0s (10.6 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-89-118:~$
```

Step 5: Initialize the Kubecluster .

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





Copy the mkdir and chown commands from the top and execute them.

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

```
ubuntu@ip-172-31-89-118:~$    mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-89-118:~$
```

Add a common networking plugin called flannel as mentioned in the code.

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

```
ubuntu@ip-172-31-89-118:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-89-118:~$
```

Step 6: Now that the cluster is up and running, we can deploy our nginx server on this cluster.Apply this deployment file using this command to create a deployment

- kubectl apply -f https://k8s.io/examples/application/deployment.yaml

```
ubuntu@ip-172-31-89-118:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-89-118:~$
```

- kubectl get pods

```
ubuntu@ip-172-31-89-118:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-2nqdp    0/1     Pending   0          56s
nginx-deployment-d556bf558-lfxg4    0/1     Pending   0          56s
ubuntu@ip-172-31-89-118:~$
```

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

```
ubuntu@ip-172-31-89-118:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-89-118:~$
```

Note : We have faced an error as pod status is pending so make it running, run the below commands then again run above 2 commands.

Error:

```
ubuntu@ip-172-31-89-118:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
ubuntu@ip-172-31-89-118:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE     VERSION
ip-172-31-89-118   Ready    control-plane   8m42s   v1.31.1
ubuntu@ip-172-31-89-118:~$
```

Solving error:

- kubectl get nodes -o=custom-columns=NAME:.metadata.name,TAINTS:.spec.taints
- kubectl taint nodes --all node-role.kubernetes.io/control-plane:NoSchedule-

```
ubuntu@ip-172-31-89-118:~$ kubectl get nodes -o=custom-columns=NAME:.metadata.name,TAINTS:.spec.taints
NAME              TAINTS
ip-172-31-89-118   [map[effect:NoSchedule key:node-role.kubernetes.io/control-plane]]
ubuntu@ip-172-31-89-118:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane:NoSchedule-
node/ip-172-31-89-118 untainted
```

- kubectl get pods

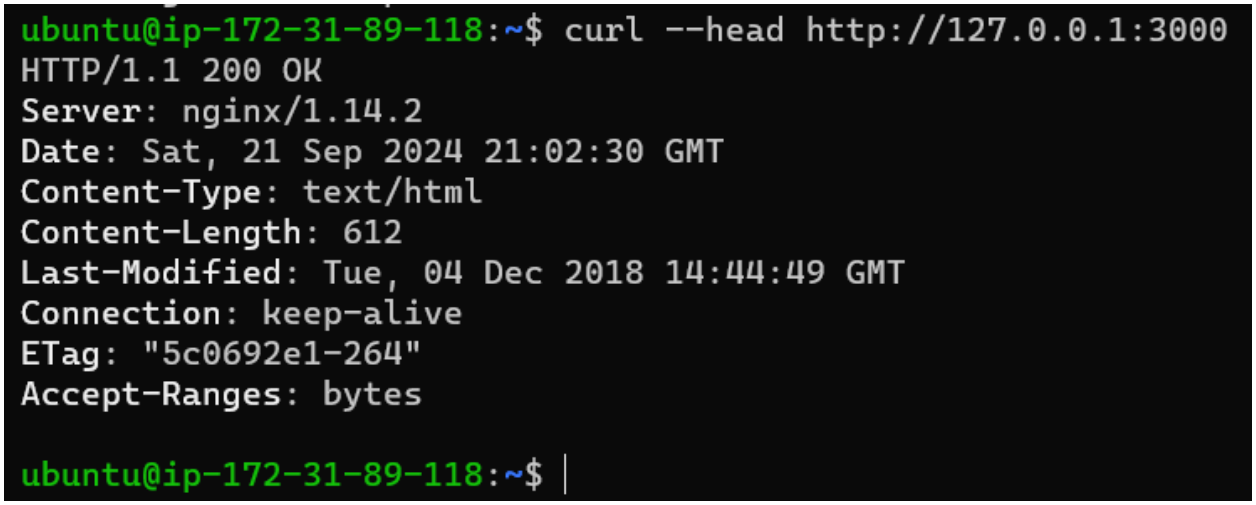
```
ubuntu@ip-172-31-89-118:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-2nqdp   1/1     Running   0          13m
nginx-deployment-d556bf558-lfxg4   1/1     Running   0          13m
ubuntu@ip-172-31-89-118:~$
```

- POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
- kubectl port-forward $POD_NAME 3000:80

```
ubuntu@ip-172-31-89-118:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 3000:80
Forwarding from 127.0.0.1:3000 -> 80
Forwarding from [::1]:3000 -> 80
```

Step 7: Verify your deployment
Open up a new terminal and ssh to your EC2 instance.
Then, use this curl command to check if the Nginx server is running.

- curl --head http://127.0.0.1:8080



ADVANCE DEVOPS EXP 5

ANSH SARFARE **D15A/50**

Aim: To understand Terraform lifecycle ,core concepts/terminologies and install it on a Linux Machine and Windows

Theory :

Terraform is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It allows users to define and provision infrastructure using a high-level configuration language known as HashiCorp Configuration Language (HCL) or JSON. Terraform supports a wide range of cloud providers, such as AWS, Azure, Google Cloud, and on-premises solutions, enabling users to manage infrastructure across multiple environments consistently.

Core Concepts and Terminologies

1. Providers:

Providers are plugins that allow Terraform to interact with various APIs of cloud providers, SaaS providers, and other services. Each provider requires configuration and manages resources for that specific service.

2. Resources:

Resources are the most fundamental elements in Terraform. They represent components of your infrastructure, such as virtual machines, databases, networks, and more.

3. Modules:

Modules are containers for multiple resources that are used together. A module can call other modules, creating a hierarchical structure. This makes it easier to organize and reuse code.

4. State:

Terraform maintains a state file that keeps track of the infrastructure managed by Terraform. The state file is crucial as it provides a mapping between the real-world resources and the configuration defined in Terraform.

5. Variables:

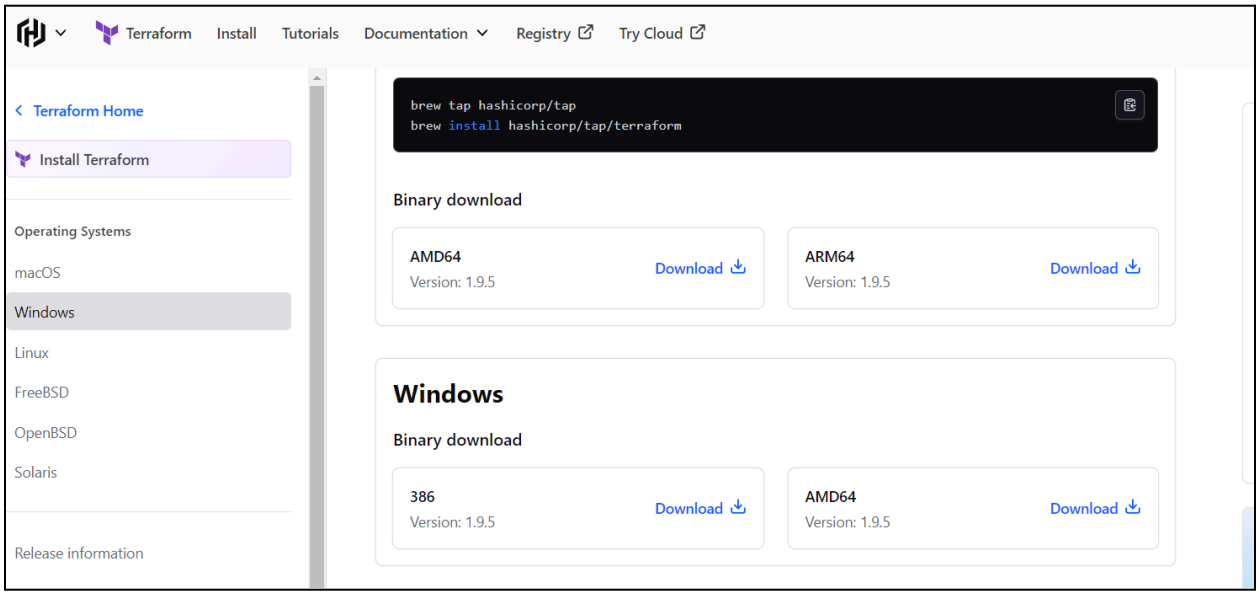
Variables in Terraform are used to make configurations dynamic and reusable. They can be defined in the configuration files and assigned values at runtime.

6. Outputs: Outputs are used to extract information from the Terraform-managed infrastructure and display it after the execution of a Terraform plan or apply.

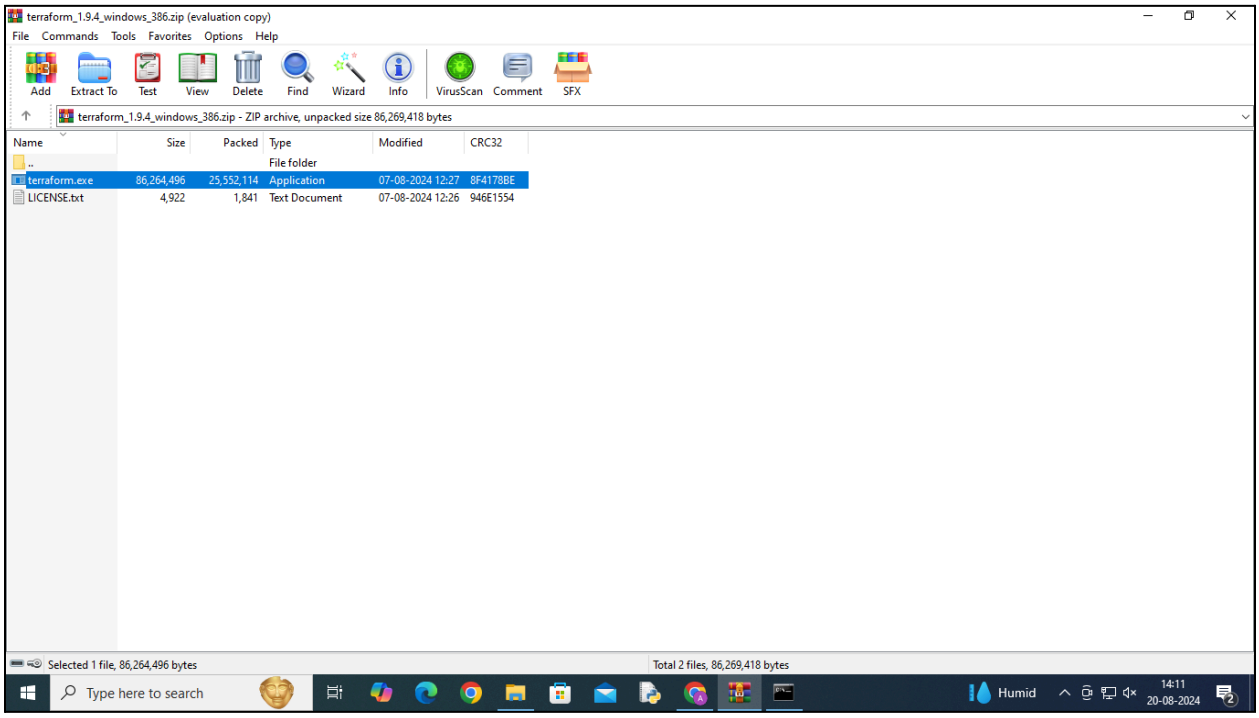
Terraform Lifecycle

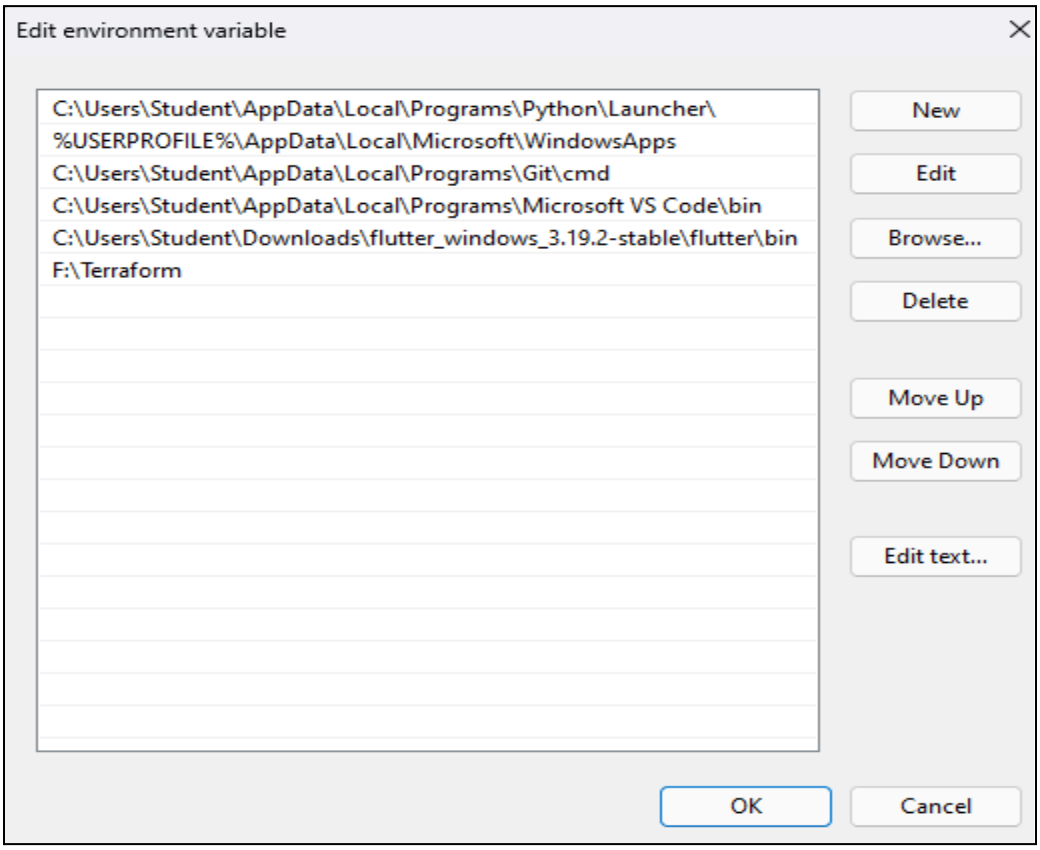
1. Write: Write the configuration file (typically with .tf extension) using HCL to describe the desired infrastructure.
2. Initialize (terraform init): Initialize the working directory containing the configuration files. This command downloads the necessary provider plugins and sets up the environment.
3. Plan (terraform plan): Terraform creates an execution plan based on the configuration files. It compares the current state with the desired state and shows the changes that will be made.
4. Apply (terraform apply): Apply the changes required to reach the desired state of the configuration. Terraform will prompt for confirmation before making any changes.
5. Destroy (terraform destroy): Destroy the infrastructure managed by Terraform. This command is used to remove all resources defined in the configuration files

Step-1: Install Terraform from the official Website



Step-2: Unzip the folder and Copy the path in environmental variables





Step-3: Run Terraform

```
PS F:\> 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
  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          Experimental support for module integration testing
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management
```

ADVANCE DEVOPS EXP 6

ANSH SARFARE **D15A/50**

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

Theory :

Terraform is an open-source tool that enables developers and operations teams to define, provision, and manage cloud infrastructure through code. It uses a declarative language to specify the desired state of infrastructure, which can include servers, storage, networking components, and more. With Terraform, infrastructure changes can be automated, versioned, and tracked efficiently.

Building Infrastructure

When you build infrastructure using Terraform, you define the desired state of your infrastructure in configuration files. For example, you may want to create an S3 bucket or deploy a Docker container on an EC2 instance. Terraform reads these configuration files and, using the specified cloud provider (such as AWS), it provisions the necessary resources to match the desired state.

● **S3 Buckets:** Terraform can create and manage S3 buckets, which are used to store and retrieve data objects in the cloud. You can define the properties of the bucket, such as its name, region, access permissions, and versioning.
● **Docker on AWS**:Terraform can deploy Docker containers on AWS infrastructure. This often involves setting up an EC2 instance and configuring it to run Docker containers, which encapsulate applications and their dependencies.

Changing Infrastructure

As your needs evolve, you may need to modify the existing infrastructure. Terraform makes it easy to implement changes by updating the configuration files to reflect the new desired state. For instance, you might want to change the storage settings of an S3 bucket, add new security policies, or modify the Docker container's configuration

Terraform's "plan" command helps you preview the changes that will be made to your infrastructure before applying them. This step ensures that you understand the impact of your changes and can avoid unintended consequences.

Destroying Infrastructure

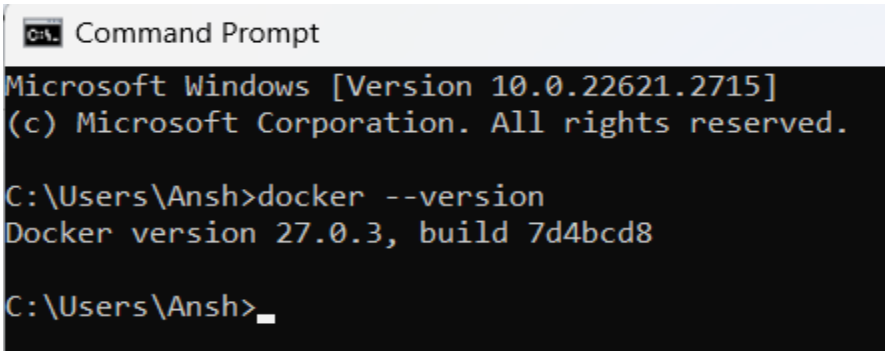
When certain resources are no longer needed, Terraform allows you to destroy them in a controlled manner. This might involve deleting an S3 bucket or terminating an EC2 instance running Docker containers. By running the "destroy" command, Terraform ensures that all associated resources are properly de-provisioned and removed.

Destroying infrastructure with Terraform is beneficial because it helps avoid unnecessary costs associated with unused resources and ensures that the environment remains clean and free of clutter.

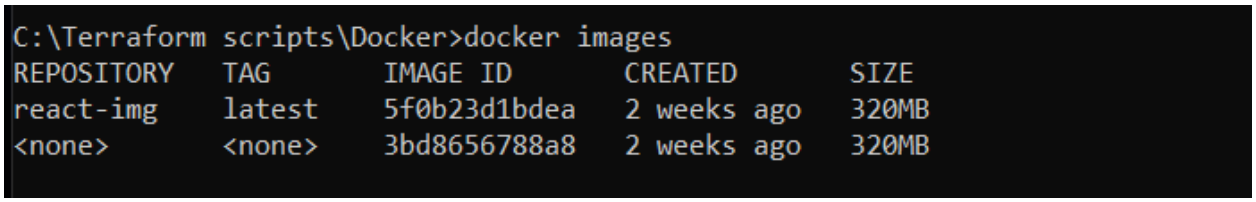
Benefits of Using Terraform for AWS Infrastructure

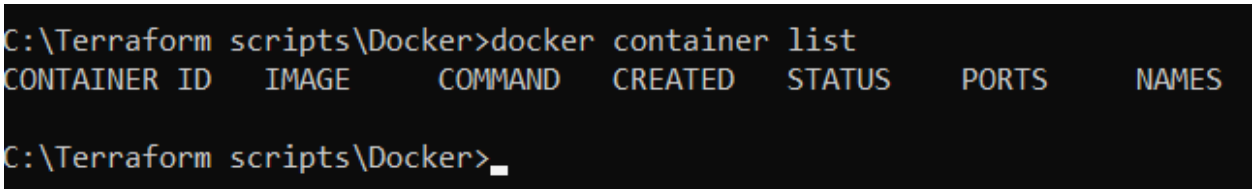
1. Consistency: Terraform ensures that infrastructure is consistent across environments by applying the same configuration files.
2. Automation: Manual processes are reduced, and infrastructure is provisioned, updated, and destroyed automatically based on code.
3. Version Control: Infrastructure configurations can be stored in version control systems (like Git), allowing teams to track changes, collaborate, and roll back if necessary.
4. Scalability: Terraform can manage complex infrastructures, scaling them up or down as needed, whether for small projects or large-scale applications.
5. Modularity: Terraform configurations can be broken down into reusable modules, making it easier to manage and scale infrastructure
Docker Installation:

Step-1: Docker installation



Step-2: Testing Docker images and container list before using terraform commands:





Step-3: Writing the docker.tf code and applying Terraform commands

Commands executed:

- terraform init
- terraform plan
- terraform apply

Code:

docker.tf

```
terraform {
        required_providers {
        docker = {
        source = "kreuzwerker/docker"
        version = "~> 3.0.1"
        }
    }
}
provider "docker" {
    host = "npipe:////.//pipe//docker_engine"
}
resource "docker_image" "nginx" {
    name = "nginx:latest"
    keep_locally = false
}
resource "docker_container" "nginx" {
    image = docker_image.nginx.image_id
    name = "tutorial"
    ports {
    internal = 80
    external = 8000
    }
}
```

Terraform Commands:

```
PS C:\Terraform scripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 3.0.1"...
- Installing kreuzwerker/docker v3.0.2...
- Installed kreuzwerker/docker v3.0.2 (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!
```

```
PS C:\Terraform scripts\Docker> terraform plan

Terraform used the selected providers to generate the following execution plan. Resou
  + create

Terraform will perform the following actions:

  # docker_container.nginx will be created
  + resource "docker_container" "nginx" {
      + attach                                      = false
      + bridge                                      = (known after apply)
      + command                                     = (known after apply)
      + container_logs                              = (known after apply)
      + container_read_refresh_timeout_milliseconds = 15000
      + entrypoint                                  = (known after apply)
      + env                                         = (known after apply)
      + exit_code                                   = (known after apply)
      + hostname                                    = (known after apply)
      + id                                          = (known after apply)
```

```
PS C:\Terraform scripts\Docker> terraform apply

Terraform used the selected providers to generate the following execution plan.
  + create

Terraform will perform the following actions:

  # docker_container.nginx will be created
  + resource "docker_container" "nginx" {
      + attach                                      = false
      + bridge                                      = (known after apply)
      + command                                     = (known after apply)
      + container_logs                              = (known after apply)
      + container_read_refresh_timeout_milliseconds = 15000
```

```
  Enter a value: yes

docker_image.nginx: Creating...
docker_image.nginx: Still creating... [10s elapsed]
docker_image.nginx: Still creating... [20s elapsed]
docker_image.nginx: Still creating... [30s elapsed]
docker_image.nginx: Creation complete after 38s [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda0
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=19c3b26e694e3b26a5daa18288d68c790f0168d547f94171a49e3491bd173ae9]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Terraform scripts\Docker>
```

Step-4: Checking Docker images and container list after using Terraform commands:

```
C:\Terraform scripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    5ef79149e0ec   12 days ago   188MB
react-img    latest    5f0b23d1bdea   2 weeks ago   320MB
<none>       <none>    3bd8656788a8   2 weeks ago   320MB
```

```
C:\Terraform scripts\Docker>docker container list
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS             PORTS                   NAMES
19c3b26e694e   5ef79149e0ec   "/docker-entrypoint.…"   2 minutes ago   Up About a minute   0.0.0.0:8000->80/tcp   tutorial

C:\Terraform scripts\Docker>
```

Step-5: Delete the Containers created: use terraform destroy

```
PS C:\Terraform scripts\Docker> terraform destroy
docker_image.nginx: Refreshing state... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03
docker_container.nginx: Refreshing state... [id=19c3b26e694e3b26a5daa18288d68c790f0168d547f94171a49e3491bd173ae9]

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

Terraform will perform the following actions:

  # docker_container.nginx will be destroyed
  - resource "docker_container" "nginx" {
      - attach                                      = false -> null
      - command                                     = [
```

ADVANCE DEVOPS EXP-7

ANSH SARFARE **D15A/50**

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

Theory:

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

What are the key steps to run SAST effectively?

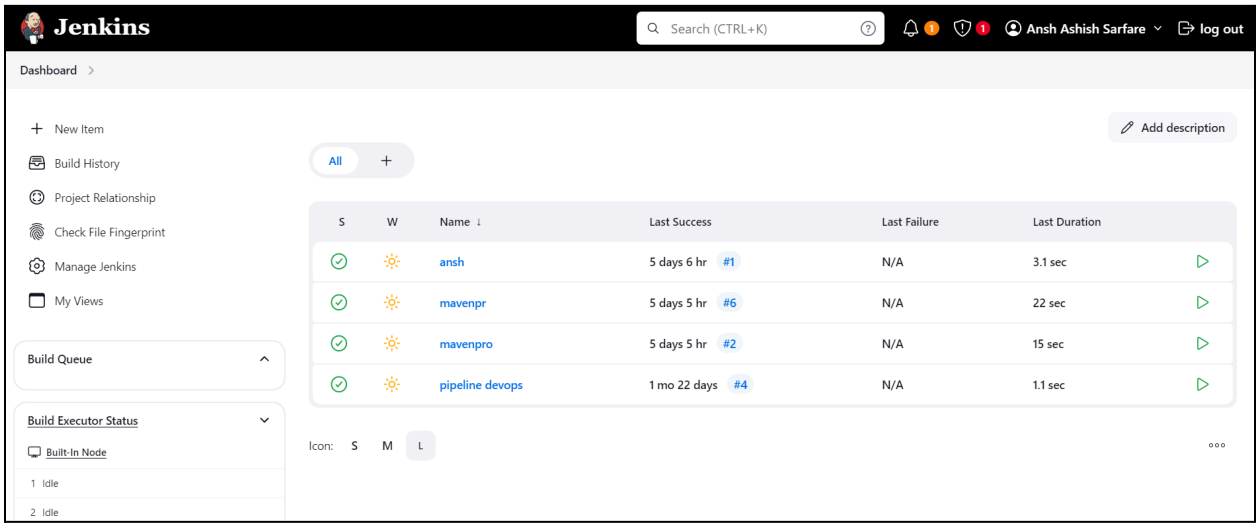
There are six simple steps needed to perform SAST efficiently in organizations that have a very large number of applications built with different languages, frameworks, and platforms.

1. Finalize the tool. Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.
2. Create the scanning infrastructure, and deploy the tool. This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.
3. Customize the tool. Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.
4. Prioritize and onboard applications. Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.
5. Analyze scan results. This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation

6. Provide governance and training Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

Steps to integrate Jenkins with SonarQube

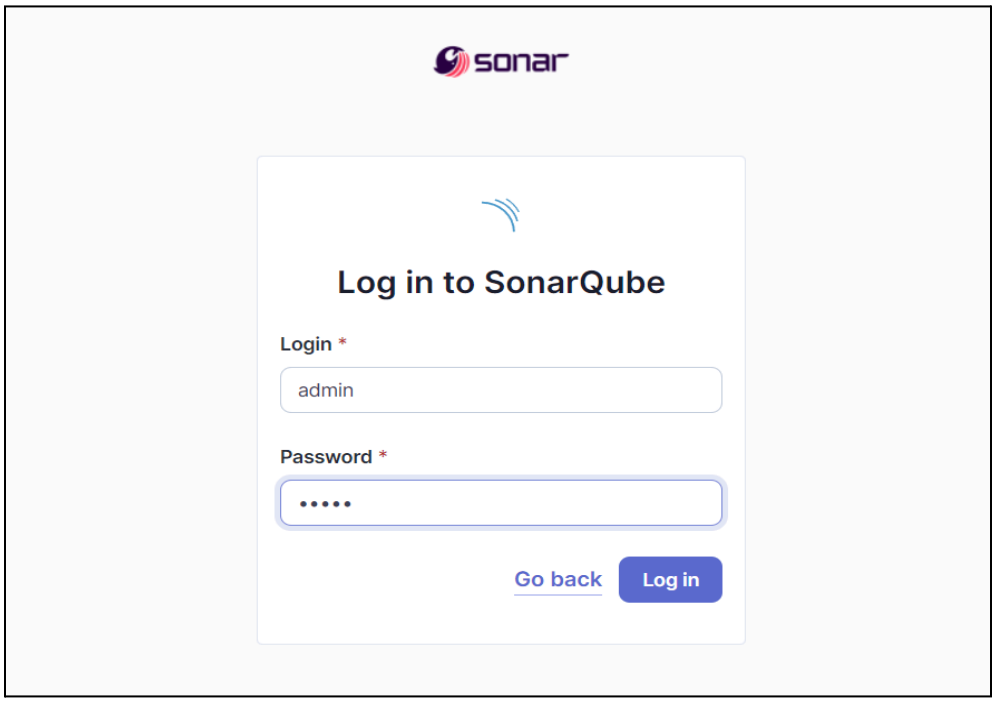
Step-1: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



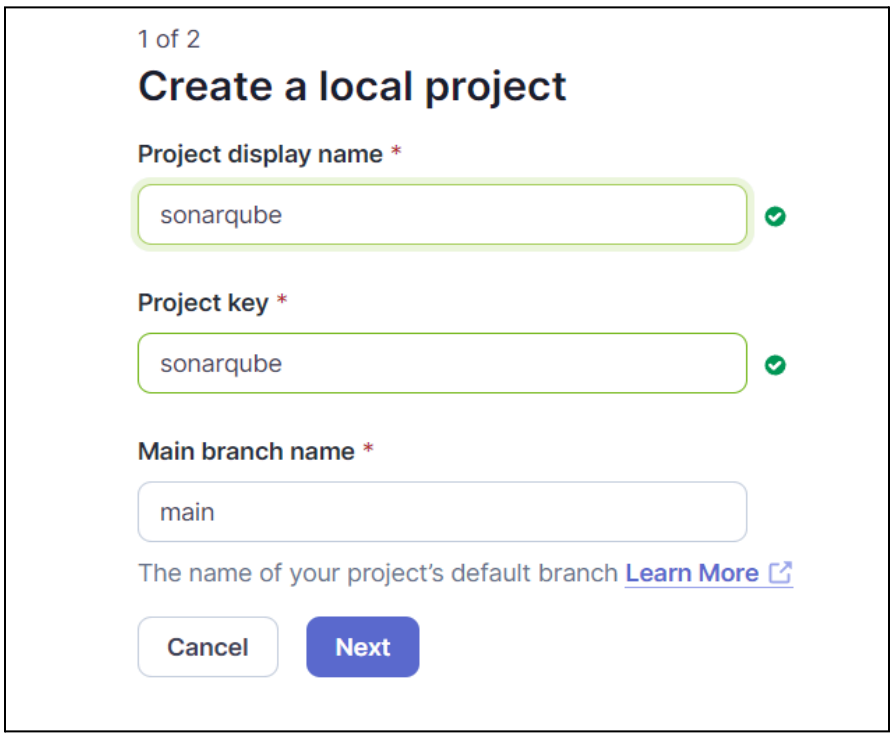
Step-2: Run SonarQube in a Docker container using this command :- a]docker -v
b] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

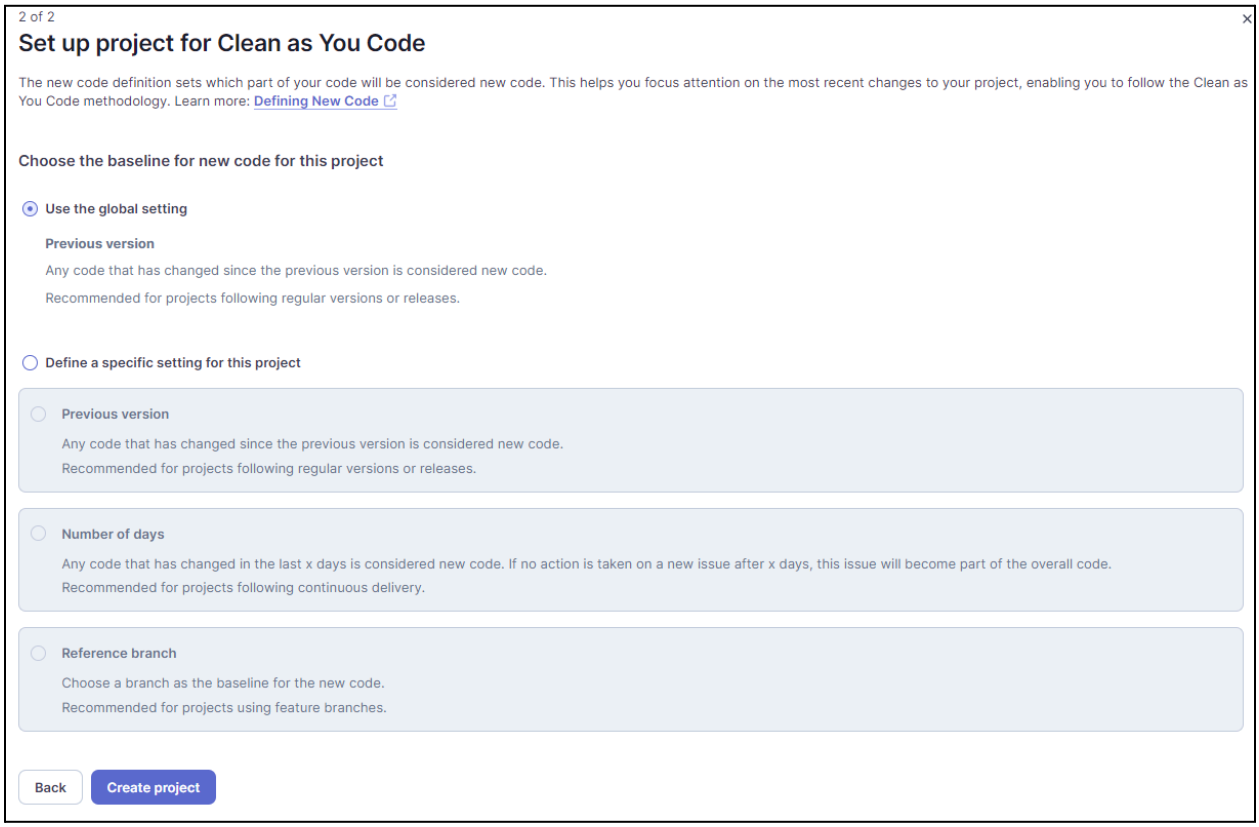
```
PS C:\Users\Ansh> docker -v
Docker version 27.0.3, build 7d4bcd8
PS C:\Users\Ansh> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube

7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
e4a886abe0db4f8a8c19e8f125ce97244d50eea97f9e98f2d829b724bc95c973
PS C:\Users\Ansh>
```

Step-3: Once the container is up and running, you can check the status of SonarQube at localhost port 9000. The login id is “admin” and the password is also “admin”.

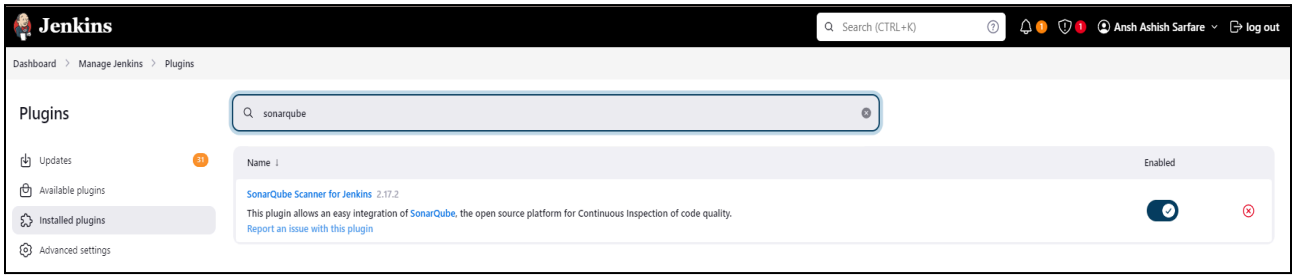


Step-4: Create a local project in SonarQube with the name sonarqube

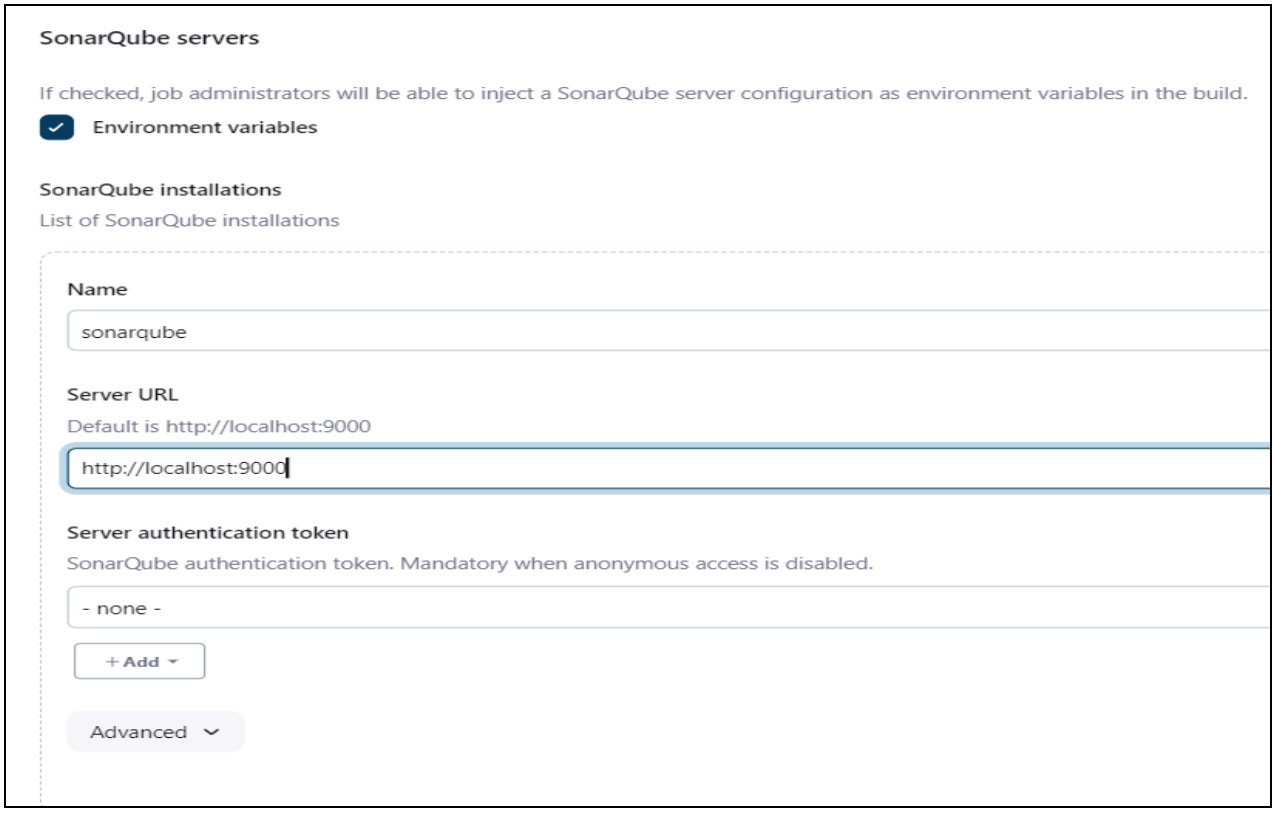




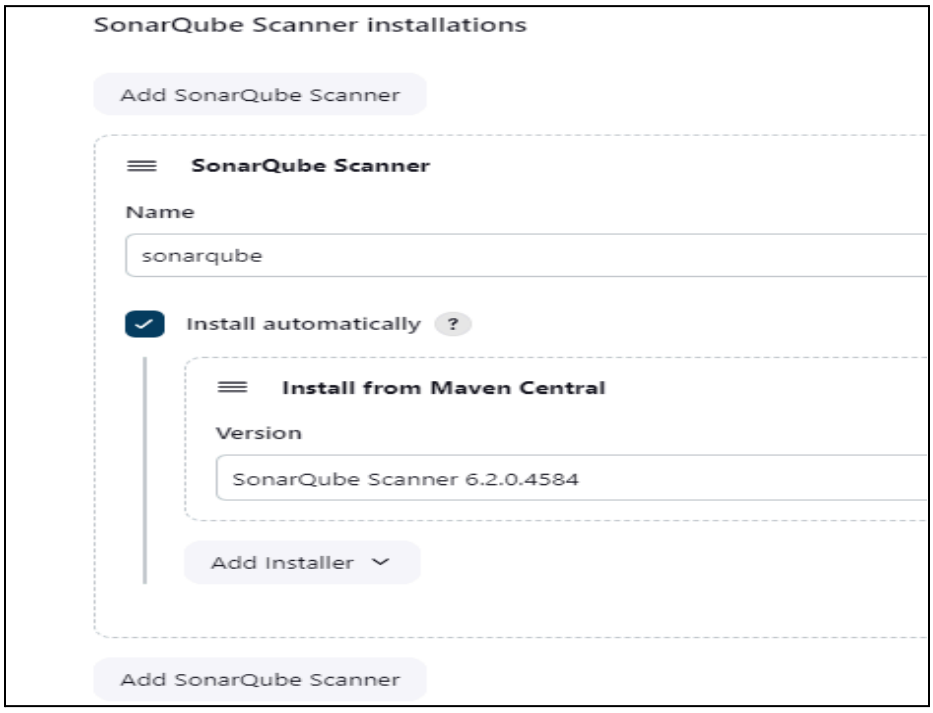
Step-5: Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins → Plugins and search for SonarQube Scanner in Available Plugins and install it.



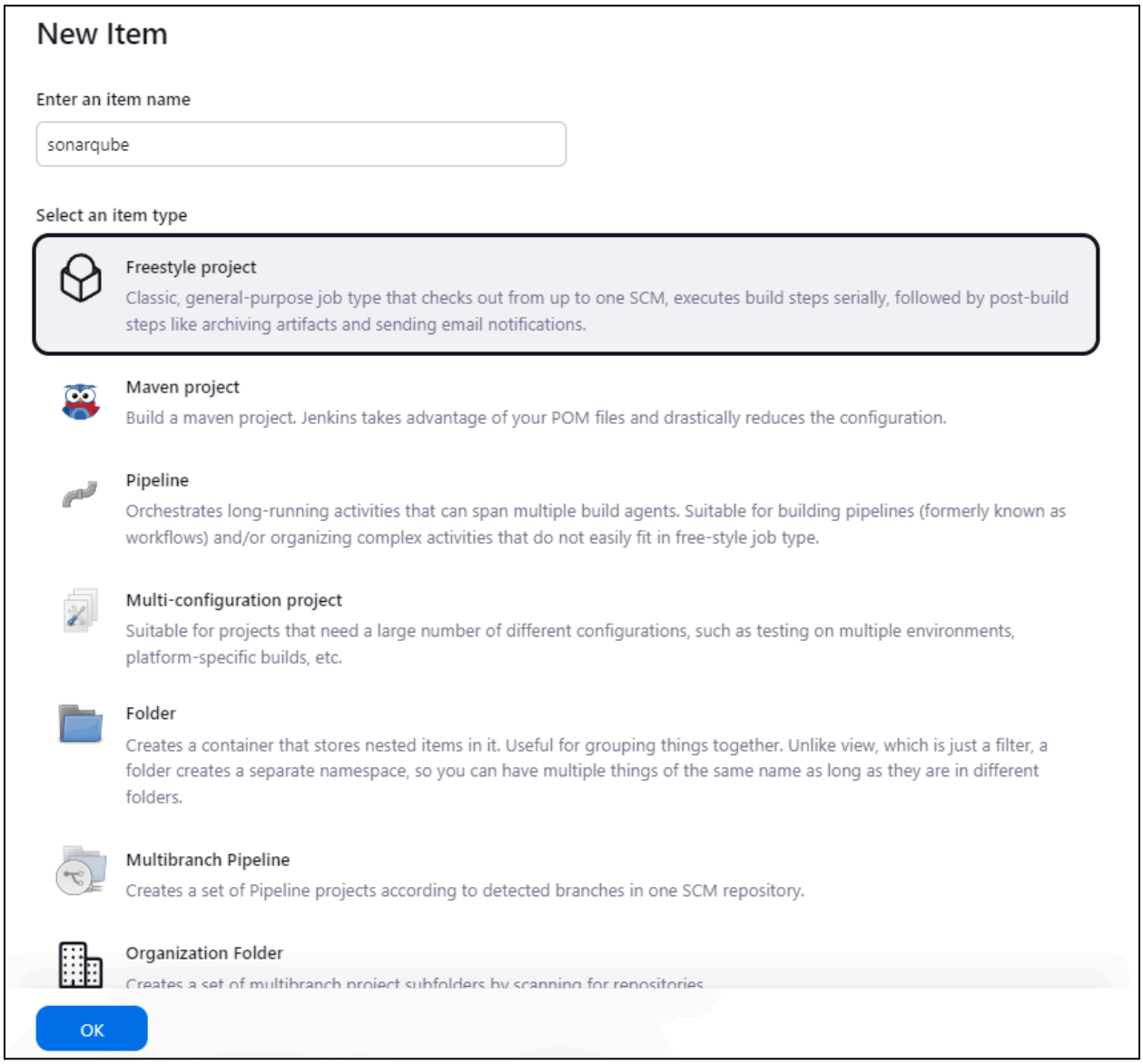
Step-6: Under ‘Manage Jenkins → System’, look for SonarQube Servers and enter these details. Name : sonarqube, Server URL : http://localhost:9000



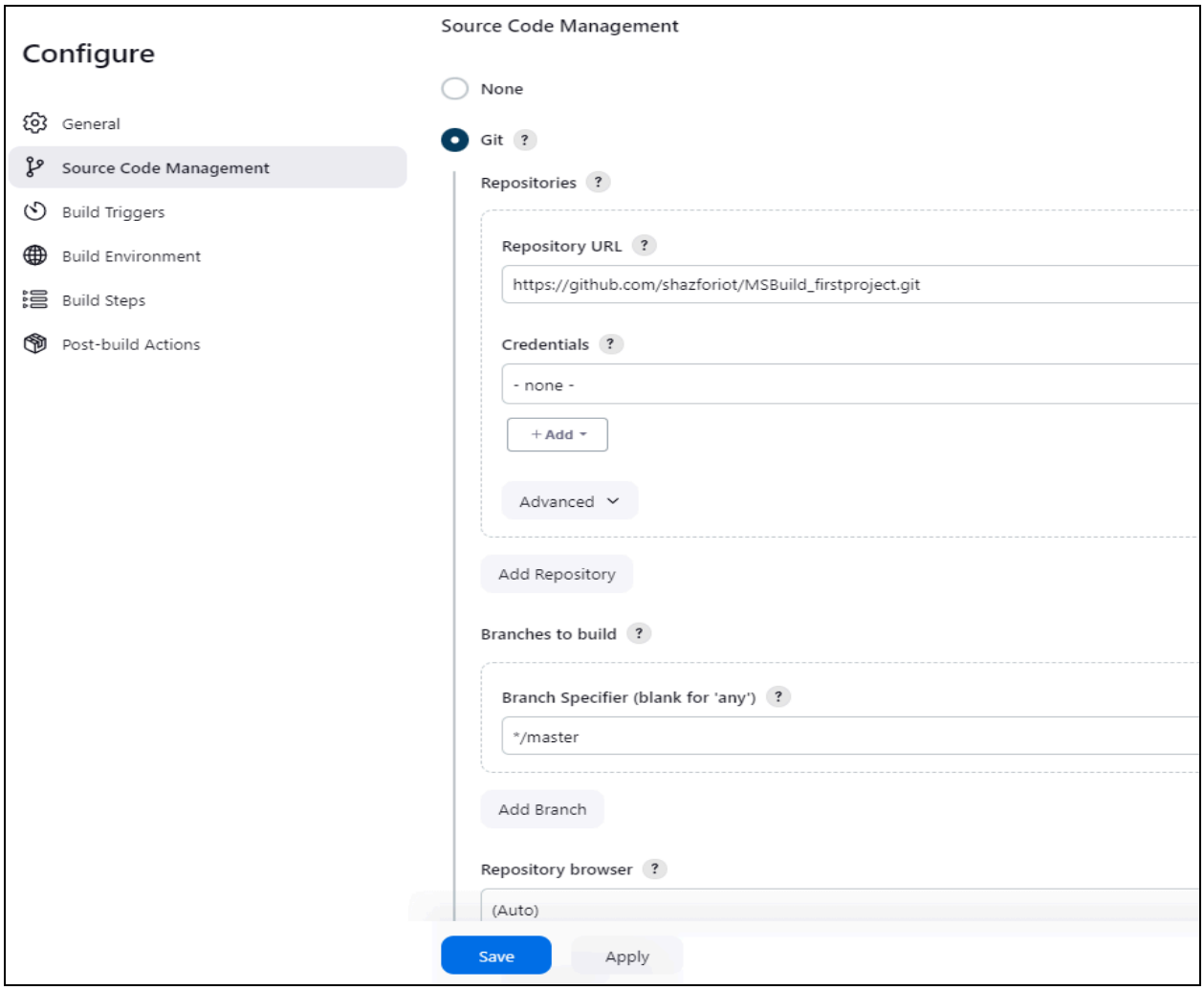
Step-7: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically. Manage Jeknins → Tools → SonarQube Scanner Installation.



Step-8: After the configuration, create a New Item in Jenkins, choose a freestyle project named sonarqube.

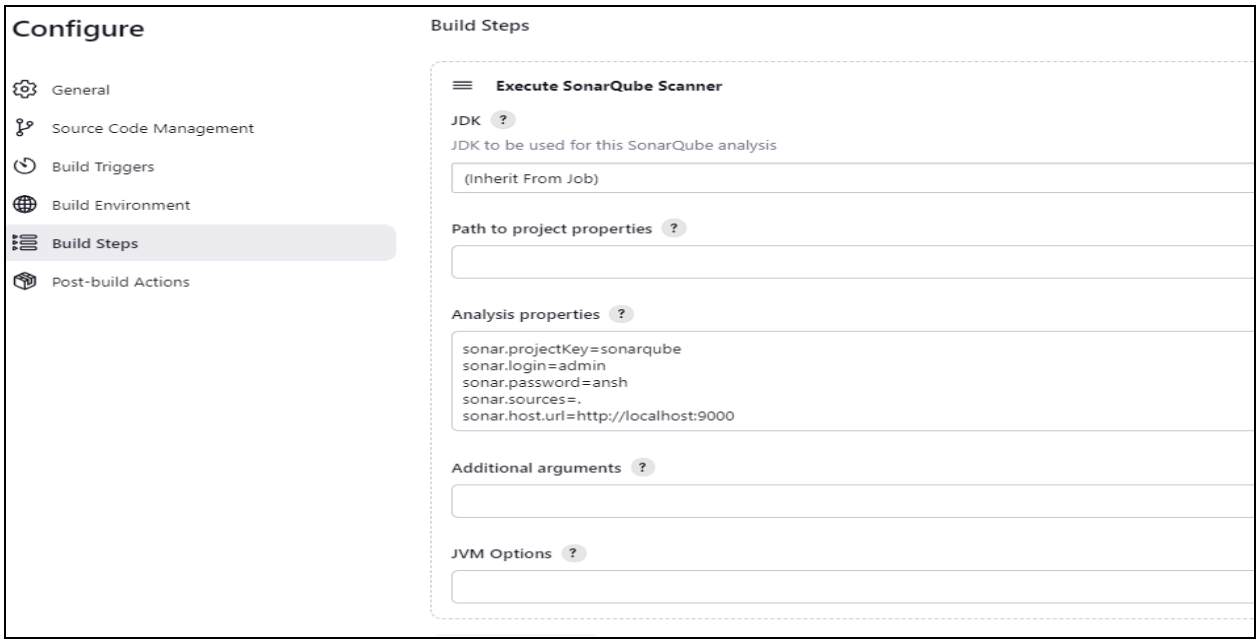


Step-9: Choose this GitHub repository in Source Code Management. https://github.com/shazforiot/MSBuild_firstproject.git . It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

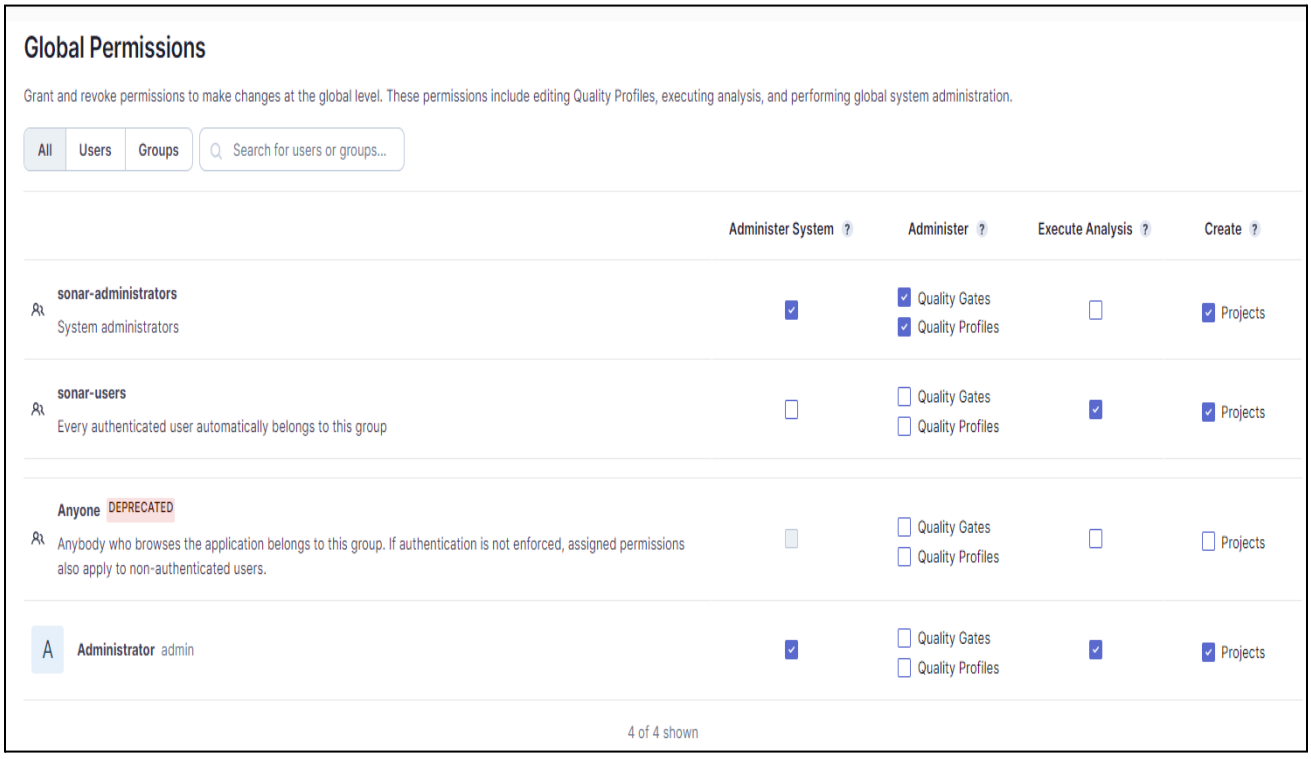


Step-10: Under Build-> Execute SonarQube Scanner, enter these Analysis Properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

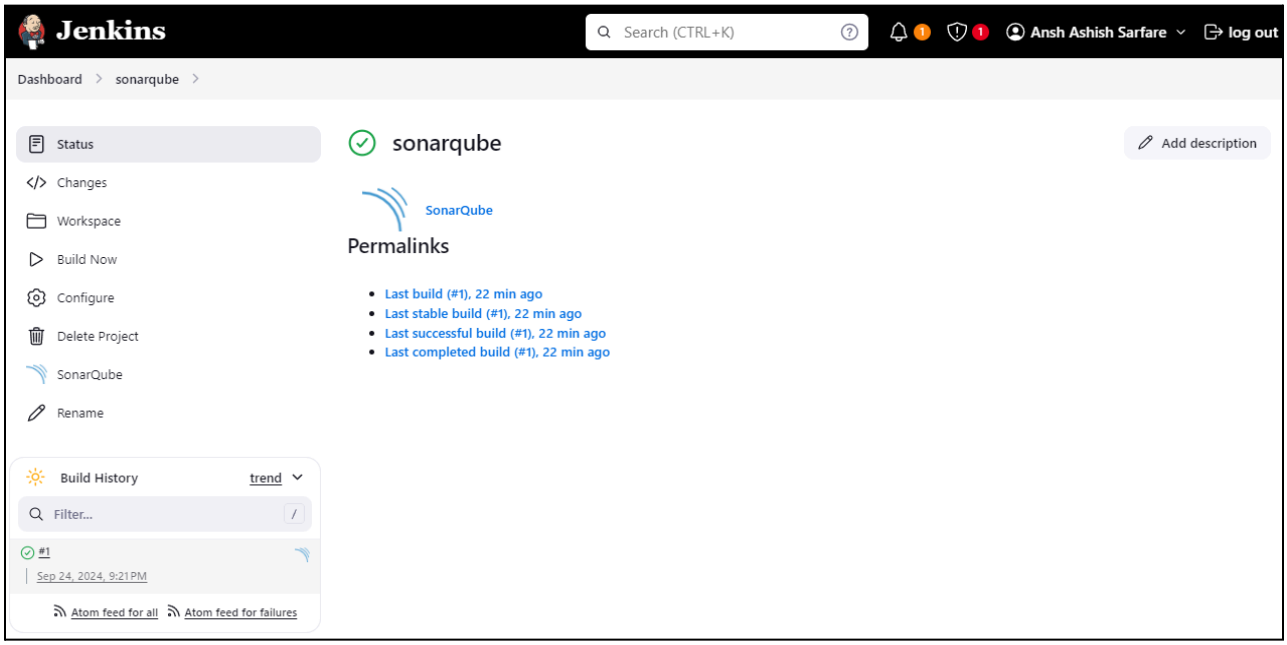
```
sonar.projectKey=sonarqube

sonar.login=admin

sonar.password=ansh

sonar.sources=.

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

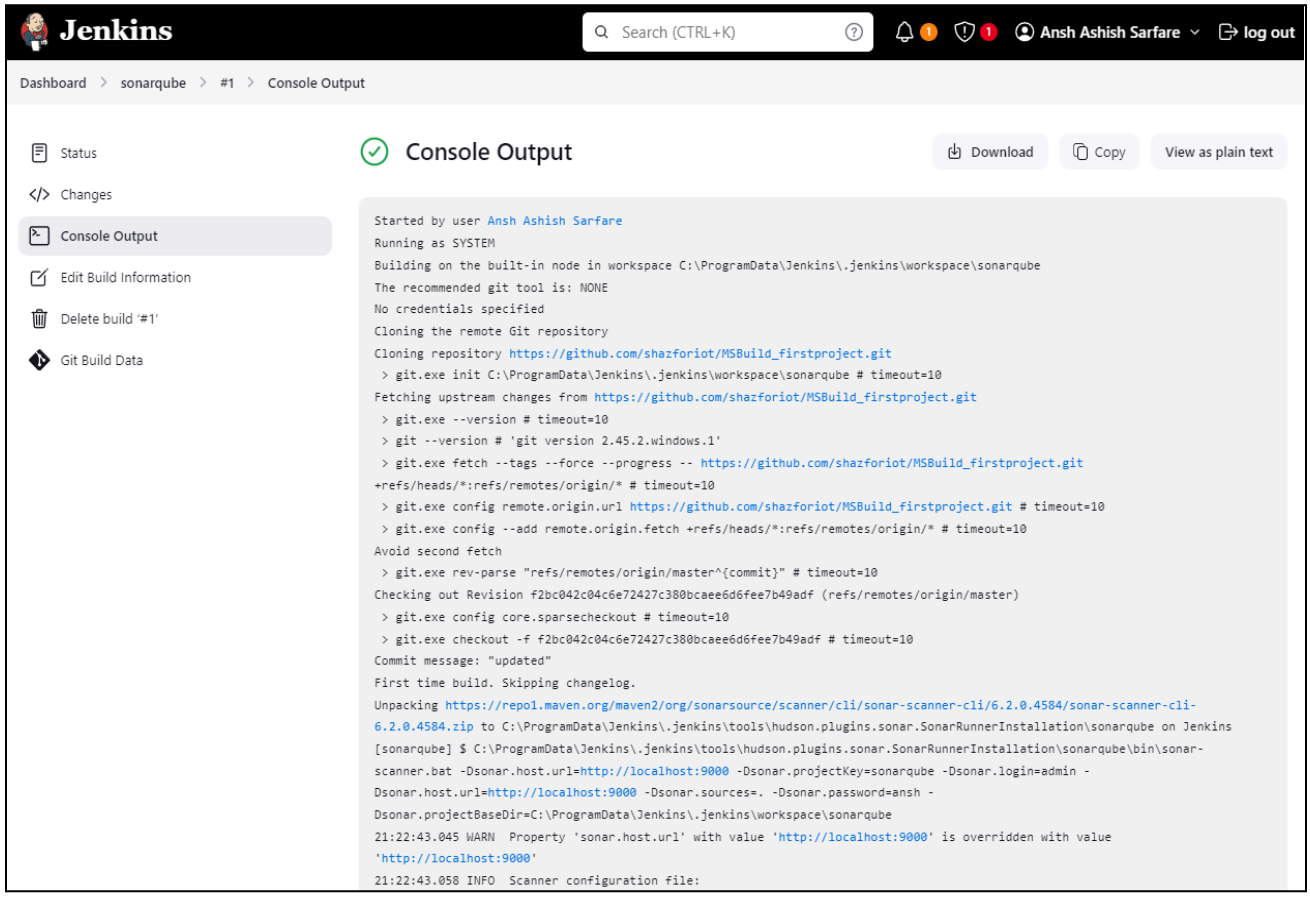


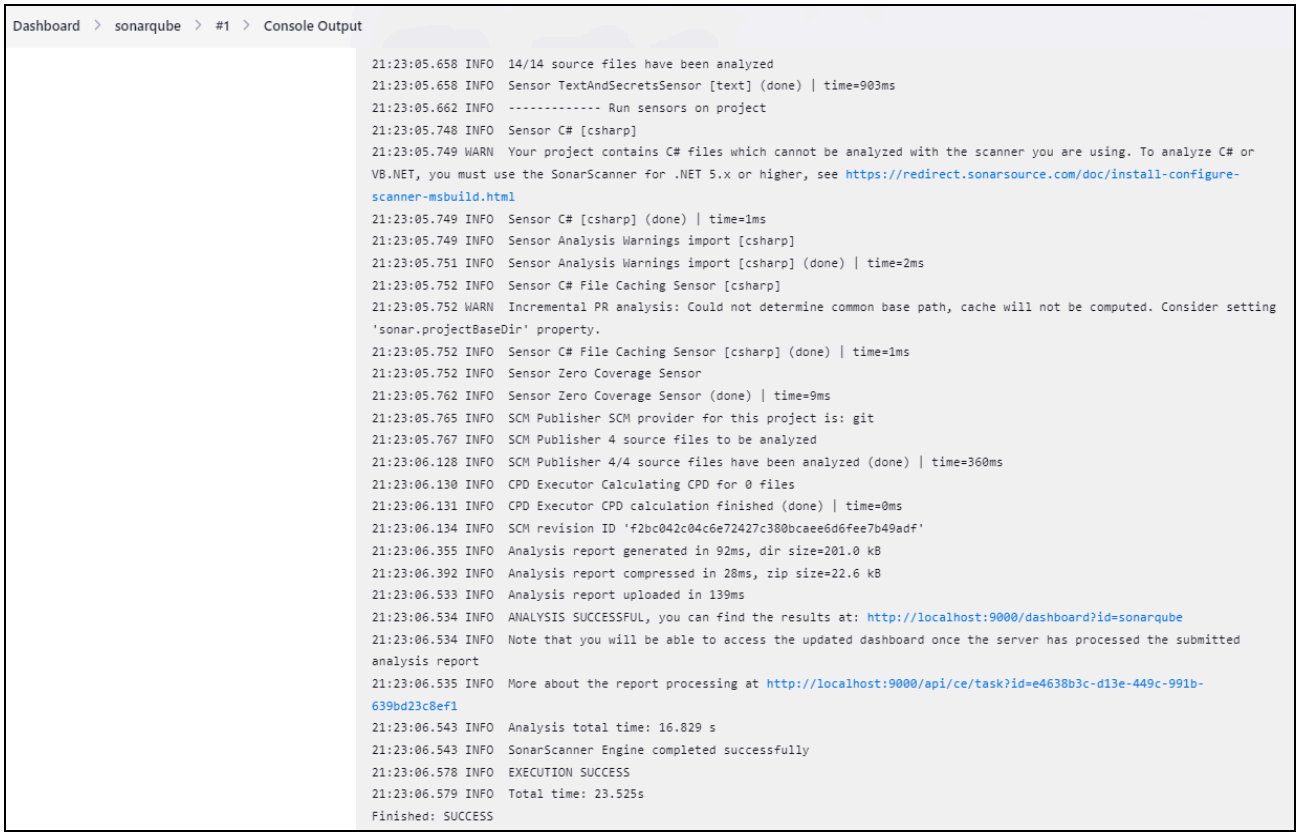
Step-11: Go to http://localhost:9000/admin/permissions and allow Execute Permissions to the Admin user.



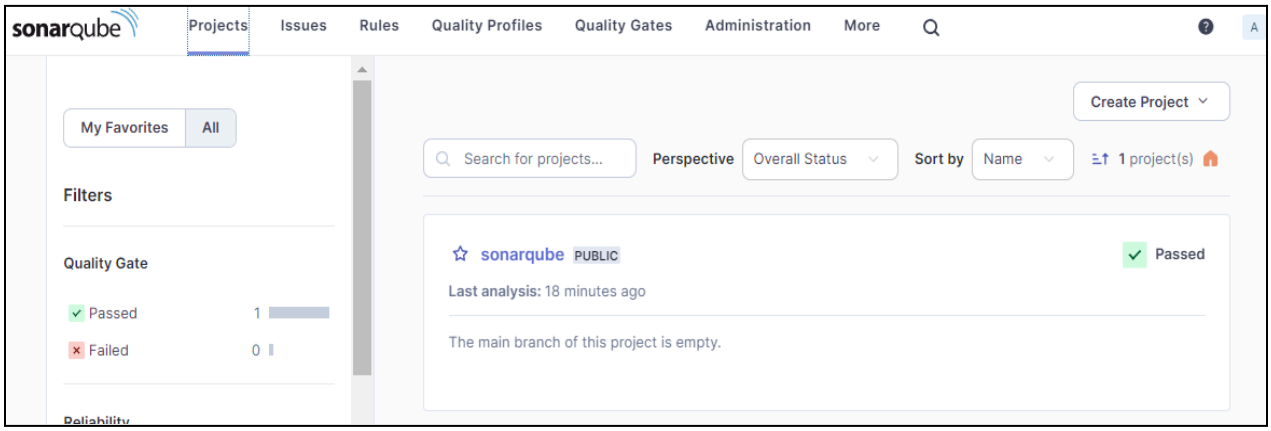
Step-12: Run The Build and check the console output.

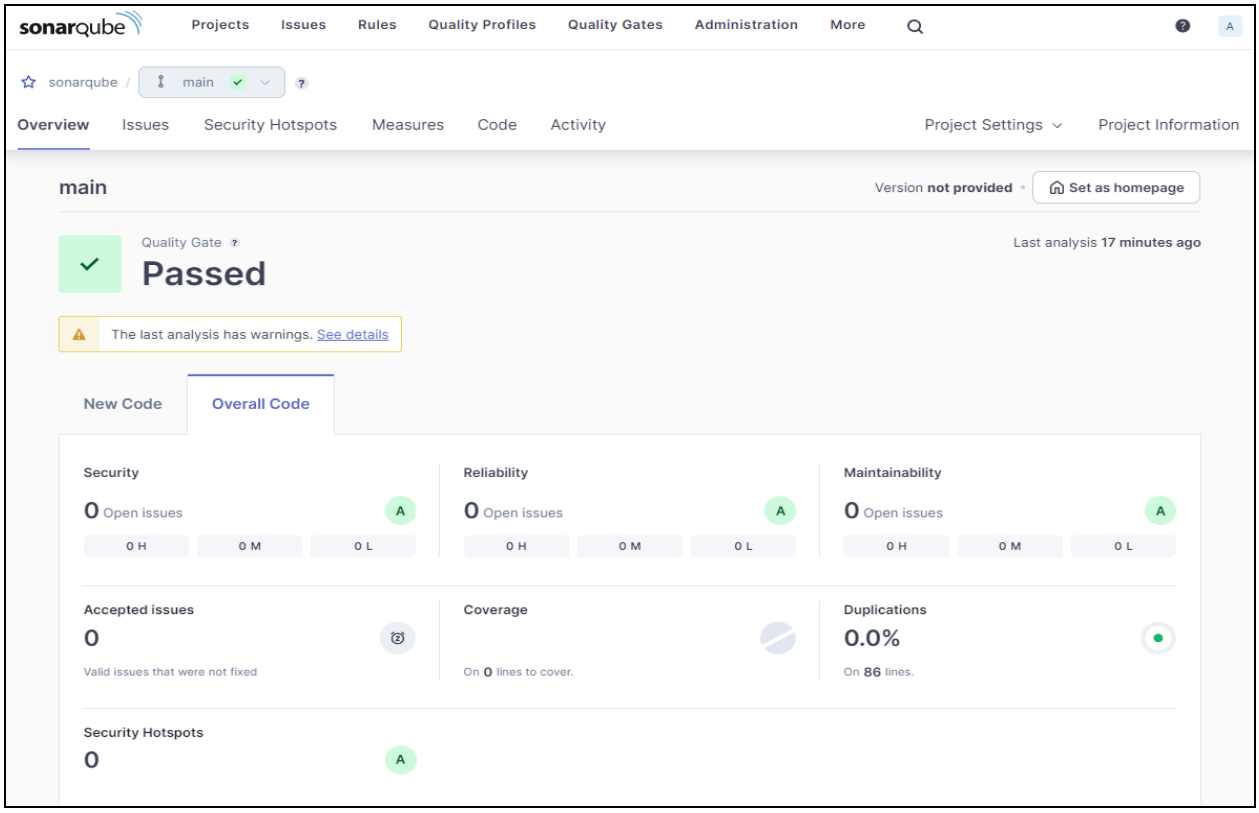






Step-13: Once the build is complete, check the project in SonarQube.




sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
sonarqube / main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
main
Version not provided
Set as homepage
Quality Gate
Passed
Last analysis 17 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
0
A

ADVANCE DEVOPS EXP-8

ANSH SARFARE **D15A/50**

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

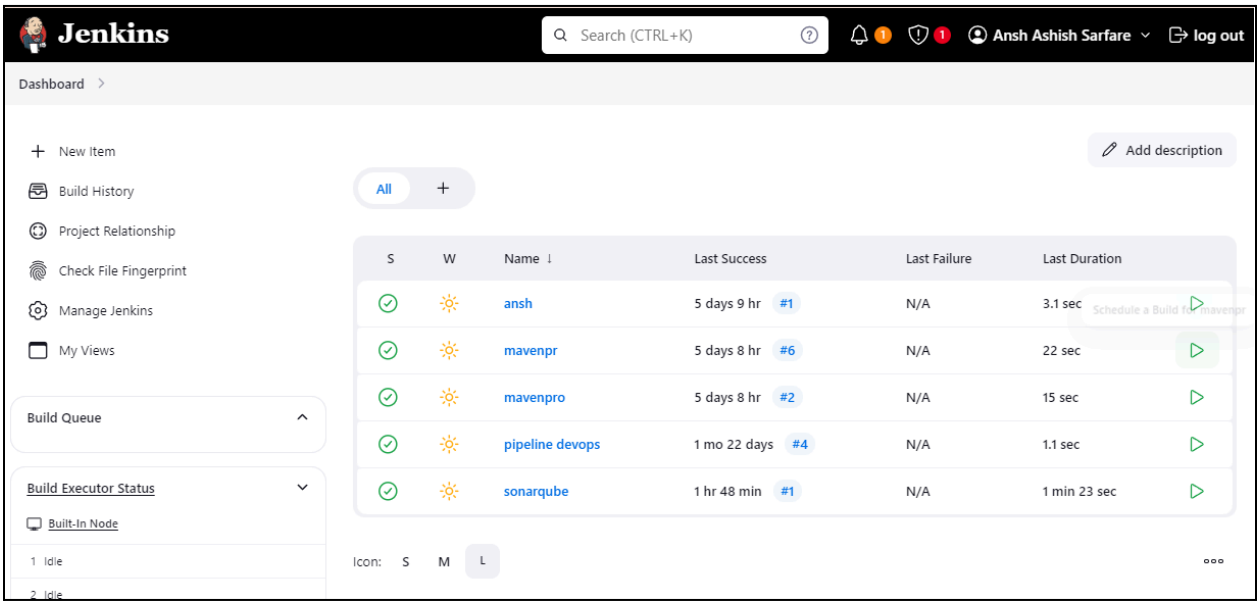
Theory:

- SAST: Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing. What problems does SAST solve?
- SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.
- SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought.
- SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code.
- Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.
- It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

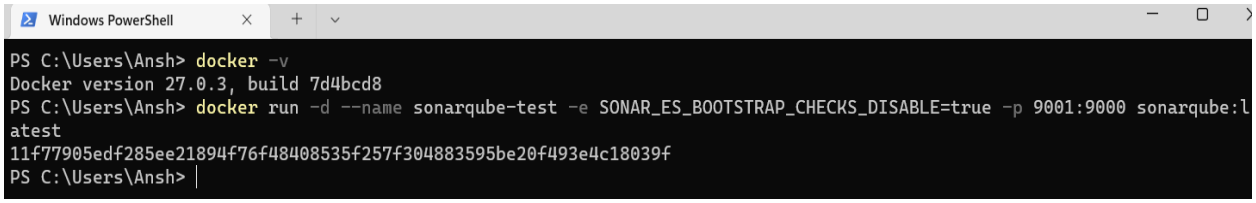
What is SonarQube

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications. It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

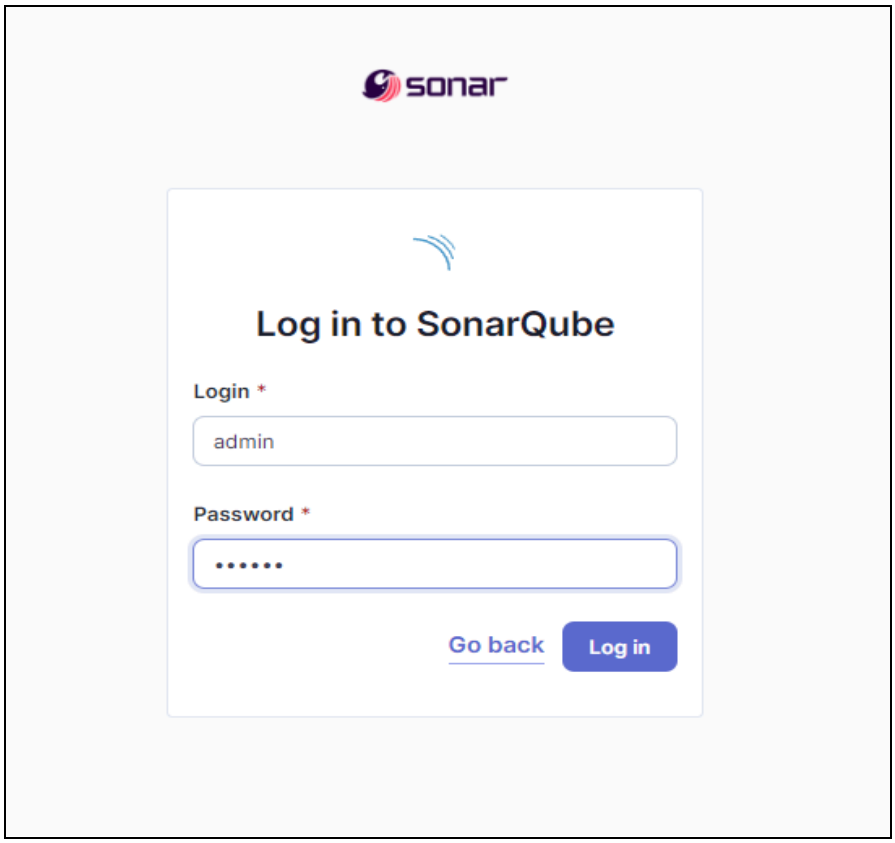
Step-1: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



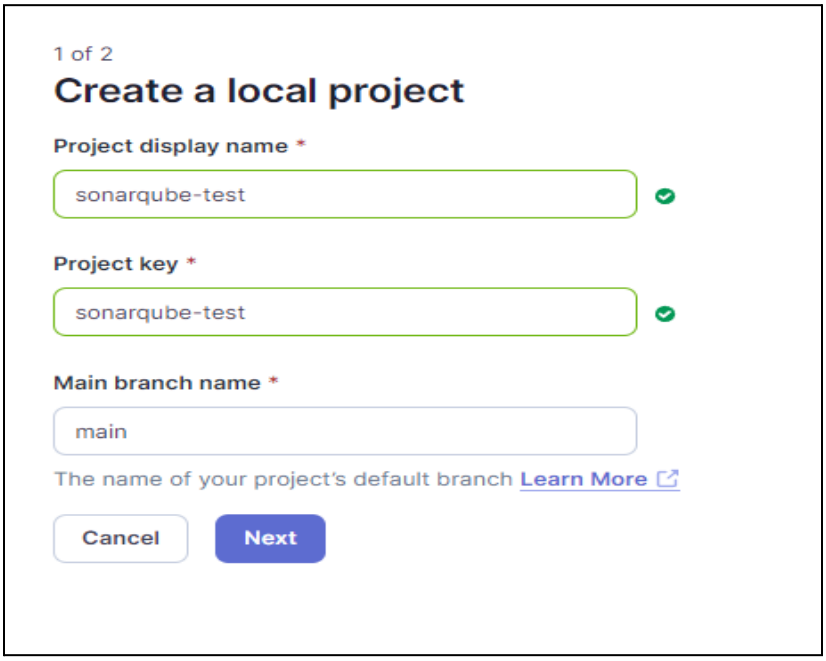
Step-2: Run SonarQube in a Docker container using this command :- a]docker -v
b] docker run -d --name sonarqube-test -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest



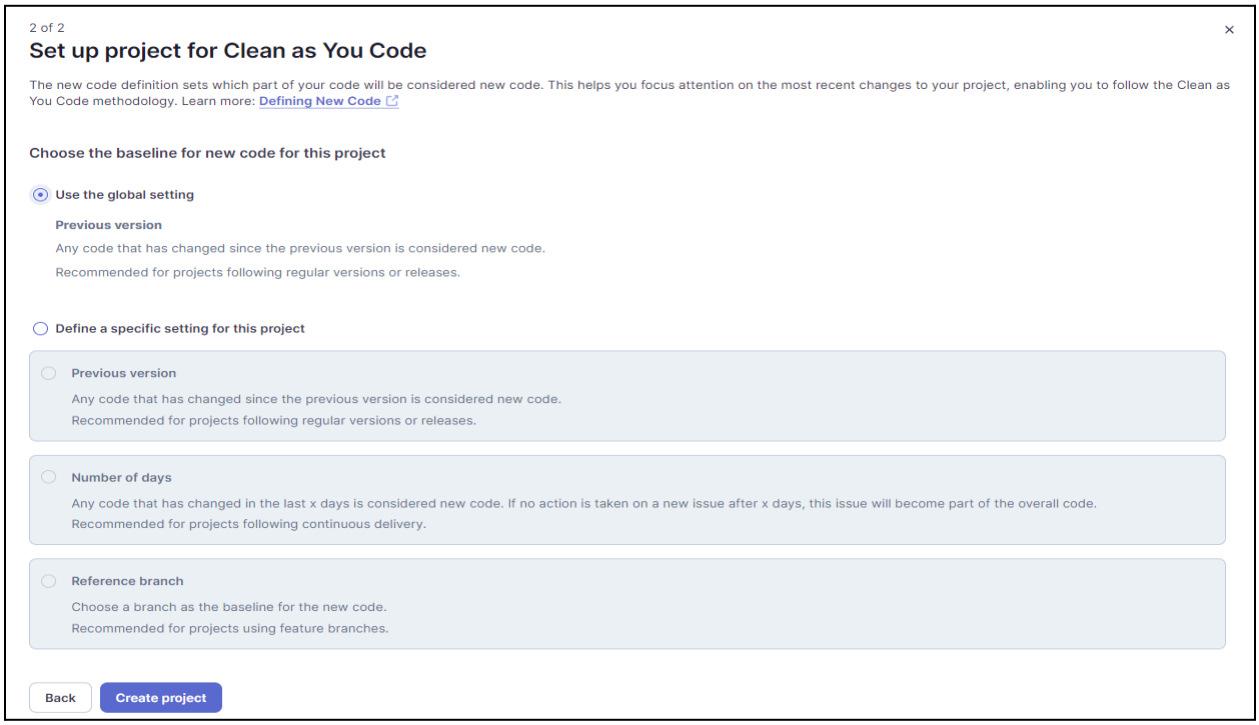
Step-3: Once the container is up and running, you can check the status of SonarQube at localhost port 9001. The login id is “admin” and the password is also “ansh16”.



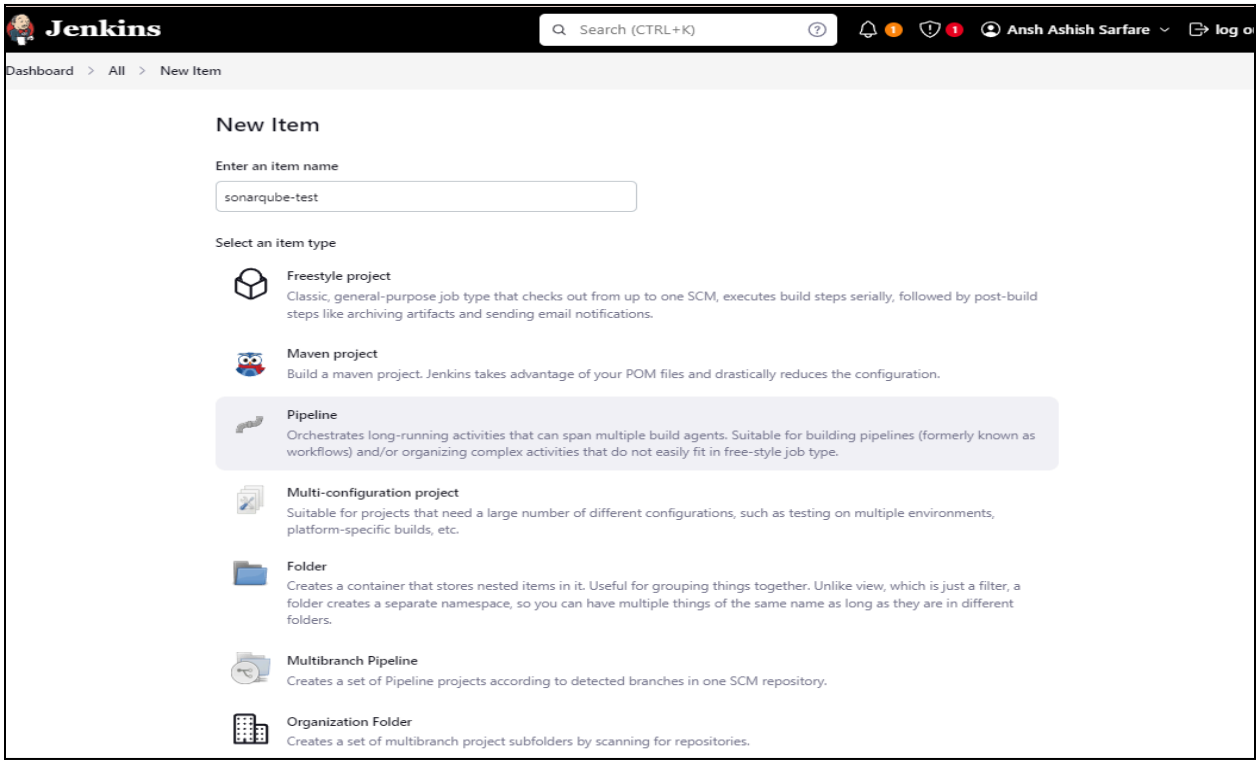
Step-4: Create a local project in SonarQube with the name sonarqube-test.



Step-5: Setup the project and come back to Jenkins Dashboard.



Step-6: Create a New Item in Jenkins, choose Pipeline.

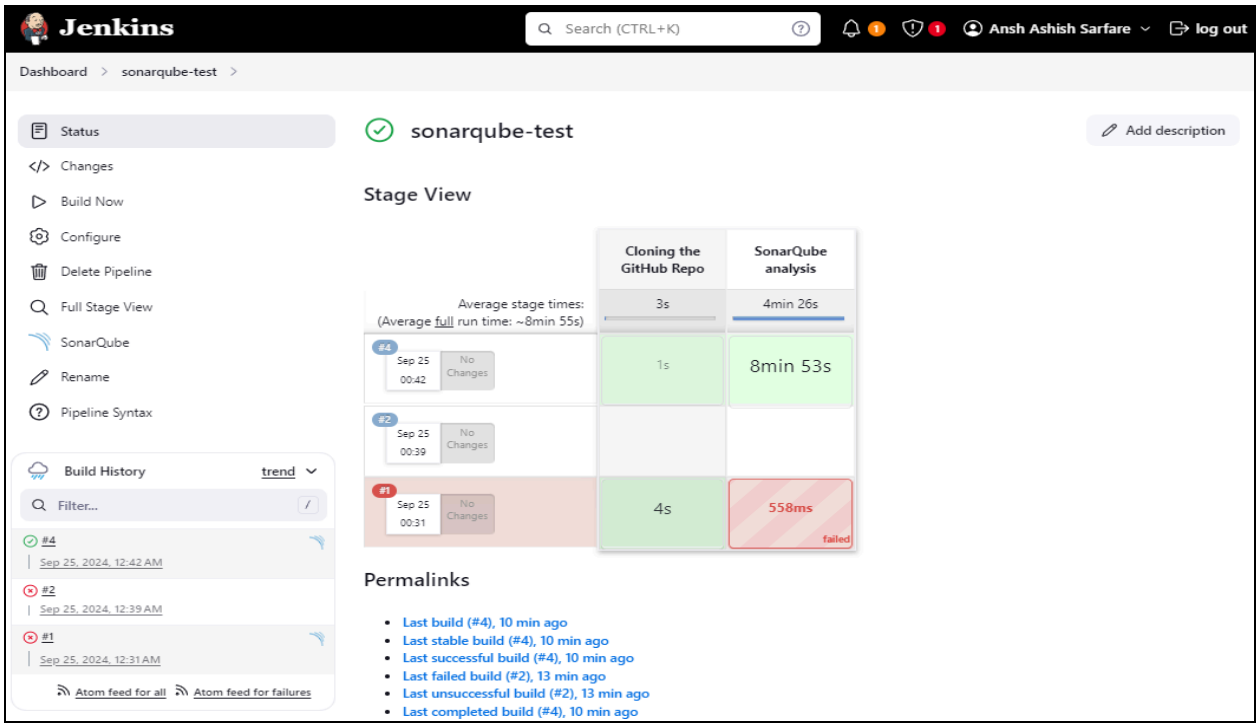


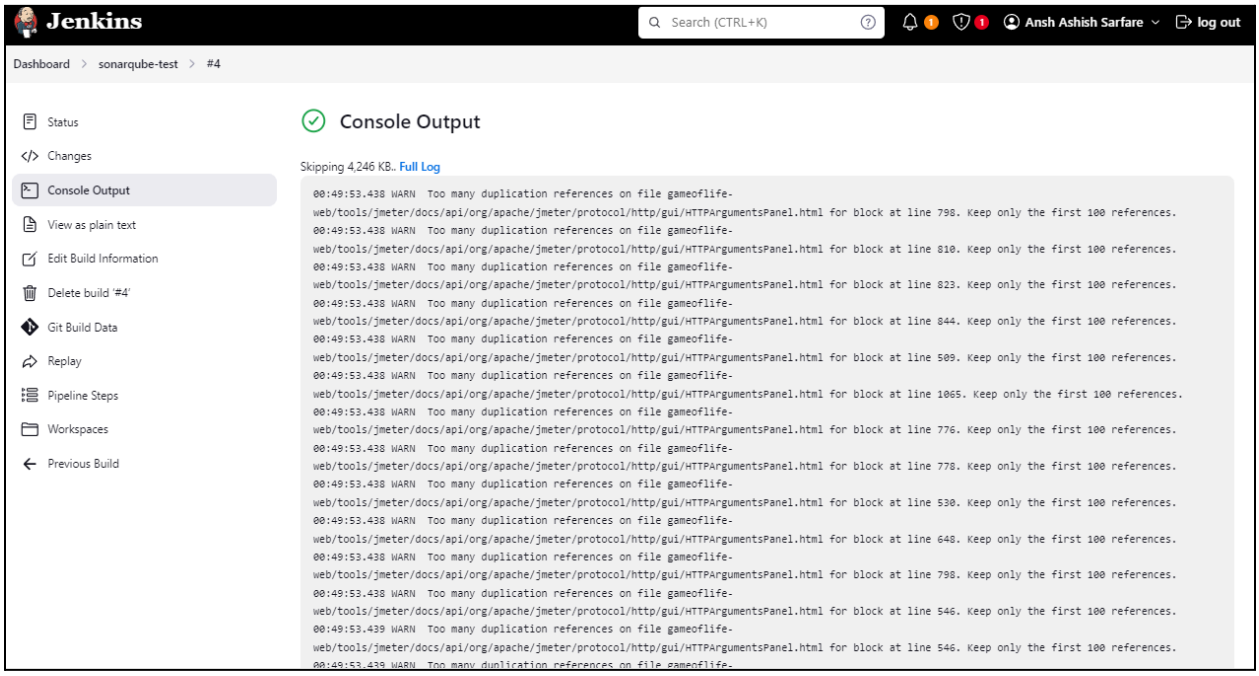
Step-7: Under Pipeline Script, enter the following -

```
node {
      stage('Cloning the GitHub Repo')
     {
      git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
bat
"C:\\Users\\Ansh\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.
4477-windows-x64\\bin\\sonar-scanner.bat \
-D sonar.login=admin \
-D sonar.password=ansh16 \
-D sonar.projectKey=sonarqube-test \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=http://localhost:9001/"
}
}
}
```

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

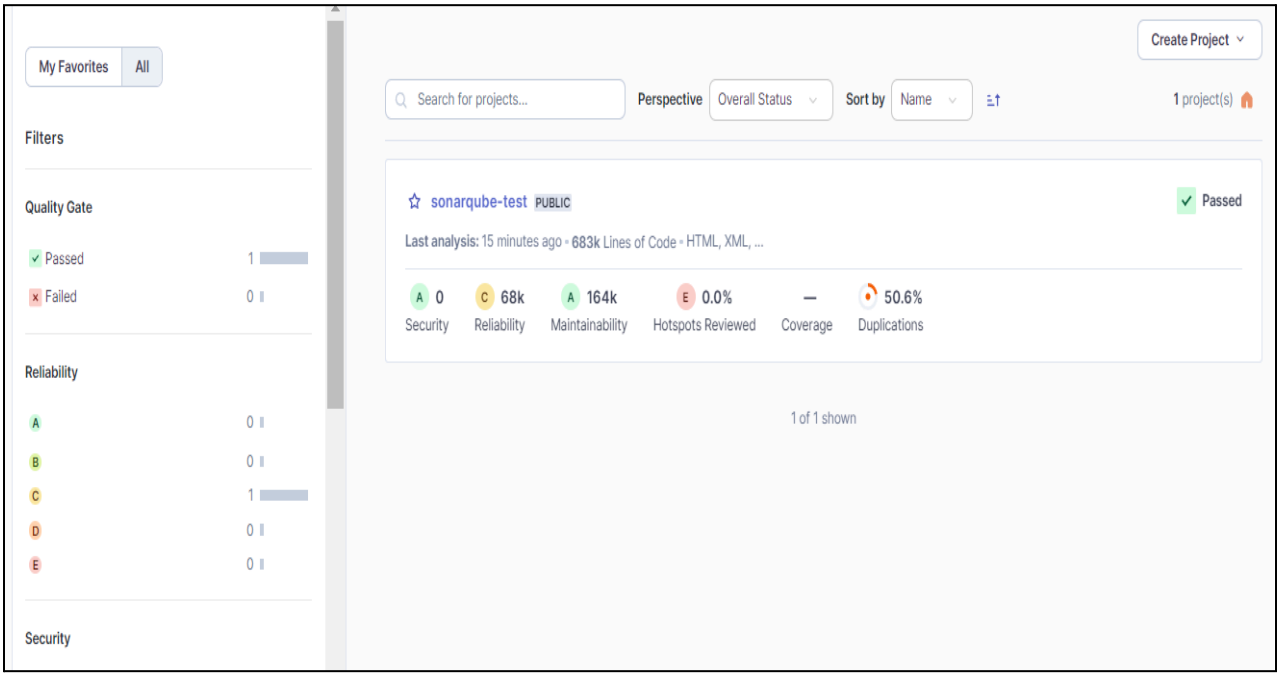
Step-8: Run The Build and check the console output:

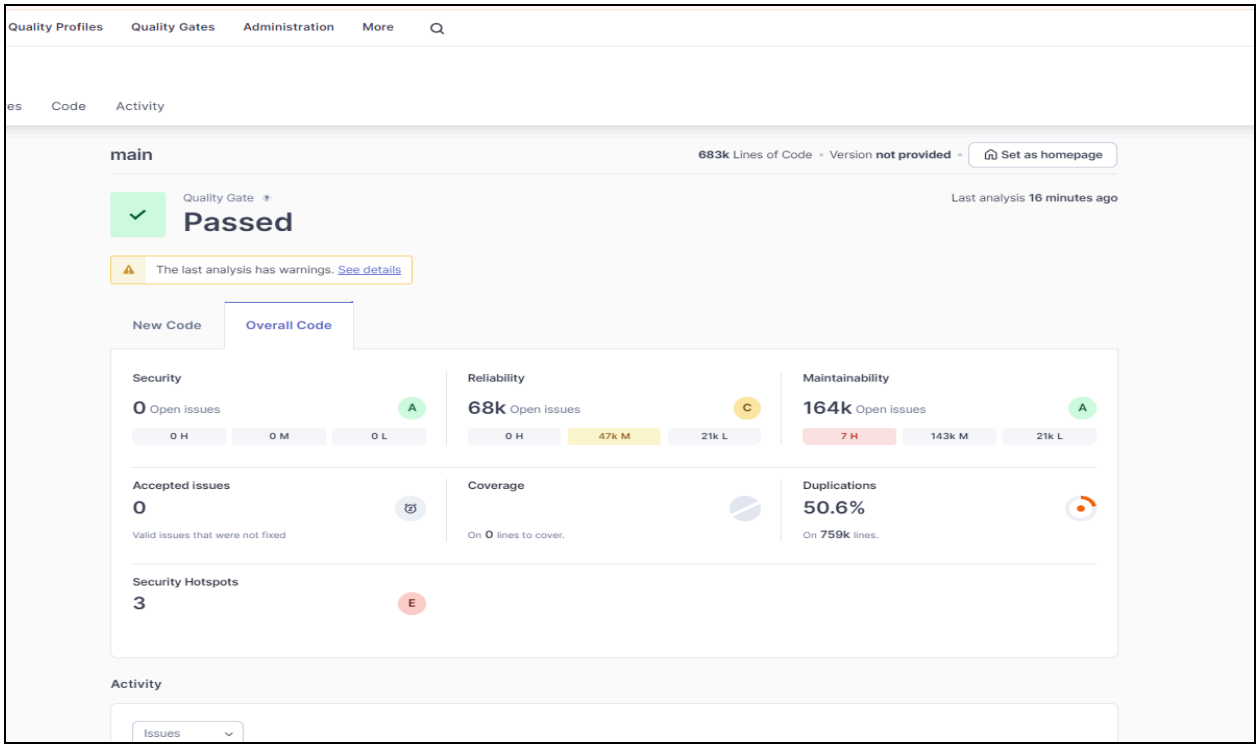

Jenkins
Search (CTRL+K)
Ansh Ashish Sarfare
log out
Dashboard > sonarqube-test >
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
SonarQube
Rename
Pipeline Syntax
Build History
trend
Filter...
#4
Sep 25, 2024, 12:42 AM
#2
Sep 25, 2024, 12:39 AM
#1
Sep 25, 2024, 12:31 AM
Atom feed for all
Atom feed for failures
sonarqube-test
Add description
Stage View
Cloning the GitHub Repo
SonarQube analysis
Average stage times:
(Average full run time: ~8min 55s)
3s
4min 26s
#4
Sep 25
00:42
No Changes
1s
8min 53s
#2
Sep 25
00:39
No Changes
#1
Sep 25
00:31
No Changes
4s
558ms
failed
Permalinks
Last build (#4), 10 min ago
Last stable build (#4), 10 min ago
Last successful build (#4), 10 min ago
Last failed build (#2), 13 min ago
Last unsuccessful build (#2), 13 min ago
Last completed build (#4), 10 min ago


Jenkins
Search (CTRL+K)
Ansh Ashish Sarfare
log out
Dashboard > sonarqube-test > #4
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#4'
Git Build Data
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Skipping 4,246 KB.. Full Log
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
00:49:53.438 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
00:49:53.439 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
00:49:53.439 WARN Too many duplication references on file gameoflife-

```
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 177. Keep only the first 100 referen
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 180. Keep only the first 100 referen
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 65. Keep only the first 100 referenc
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 349. Keep only the first 100 referen
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 40. Keep only the first 100 referenc
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 referenc
00:49:56.323 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 41. Keep only the first 100 referenc
00:49:56.324 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 17. Keep only the first 100 referenc
00:49:56.324 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 296. Keep only the first 100 referen
00:49:56.324 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 referenc
00:49:56.324 INFO  CPD Executor CPD calculation finished (done) | time=94621ms
00:49:56.350 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
00:51:30.402 INFO  Analysis report generated in 2893ms, dir size=127.2 MB
00:51:40.652 INFO  Analysis report compressed in 10210ms, zip size=29.6 MB
00:51:44.098 INFO  Analysis report uploaded in 3444ms
00:51:44.101 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9001/dashboard?id=sonarqube-test
00:51:44.101 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
00:51:44.101 INFO  More about the report processing at http://127.0.0.1:9001/api/ce/task?id=22b0b5c1-635d-4c1b-8d62-99d4ce4567b9
00:51:53.341 INFO  Analysis total time: 8:44.093 s
00:51:53.349 INFO  SonarScanner Engine completed successfully
00:51:54.059 INFO  EXECUTION SUCCESS
00:51:54.071 INFO  Total time: 8:51.363s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

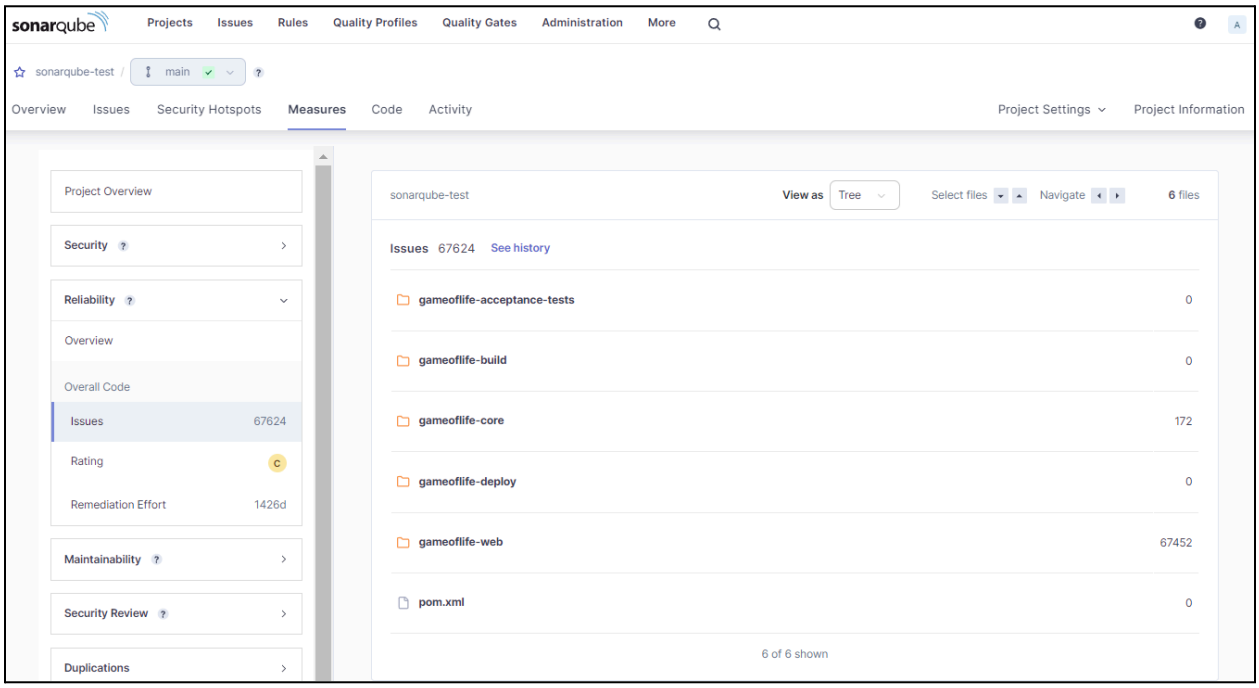
Step-9: After that, check the project in SonarQube.



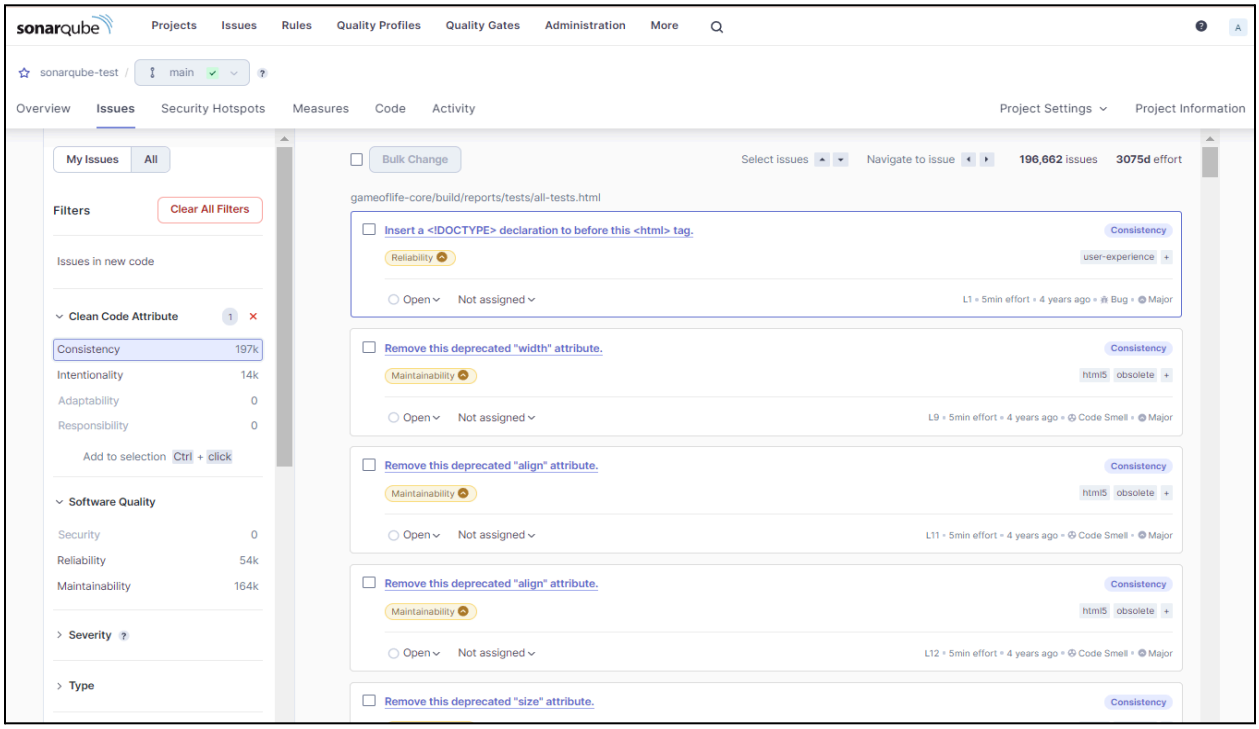


Step-10: Under different tabs, check all different issues with the code.
Code Problems

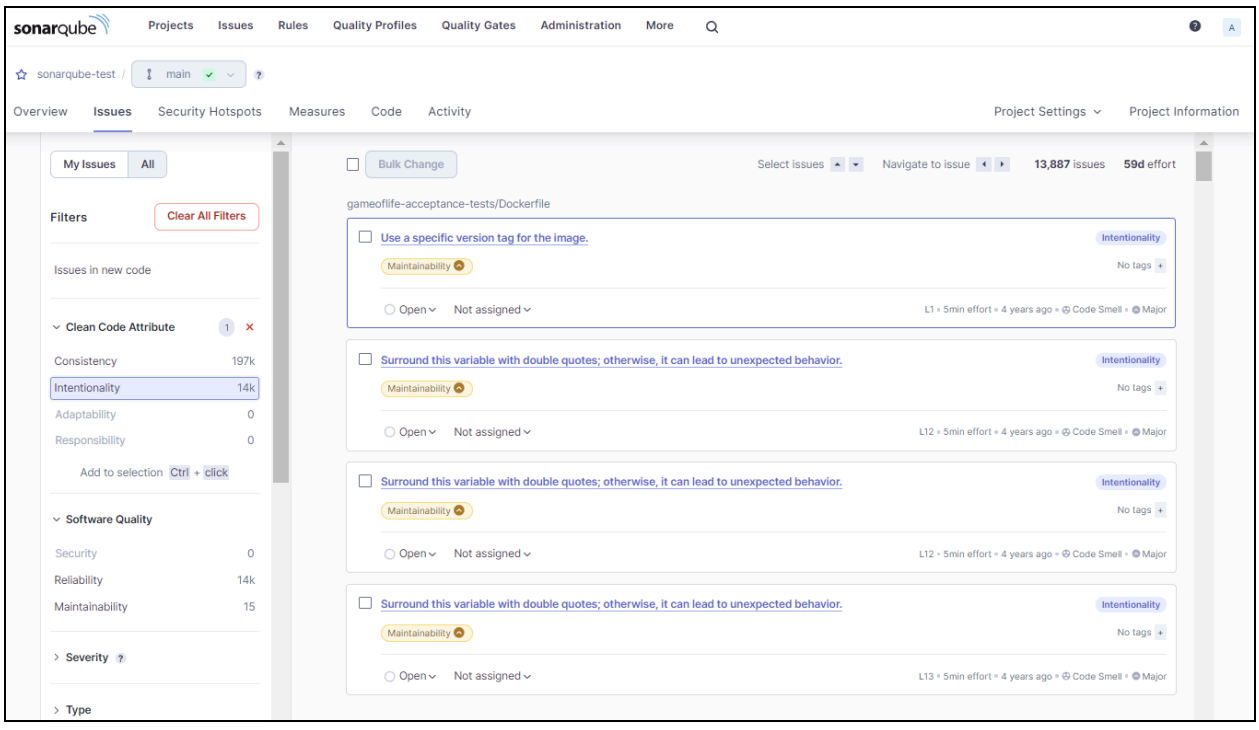
Code issues:



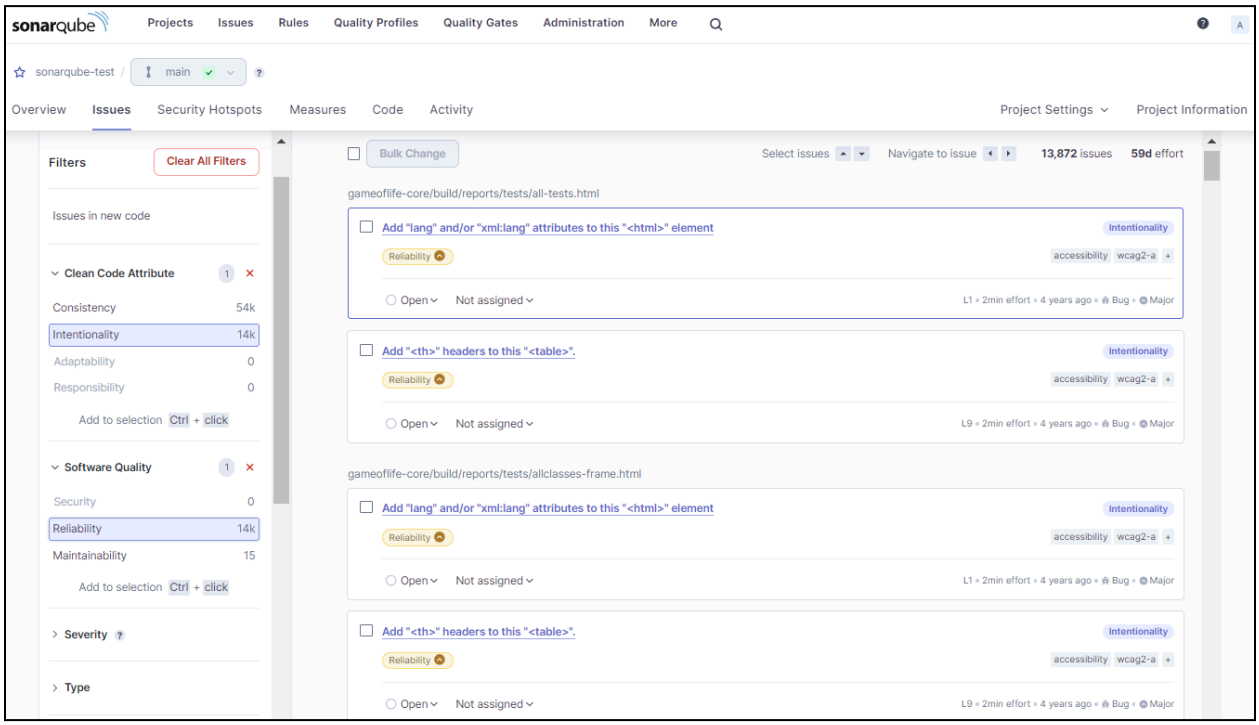
Consistency:



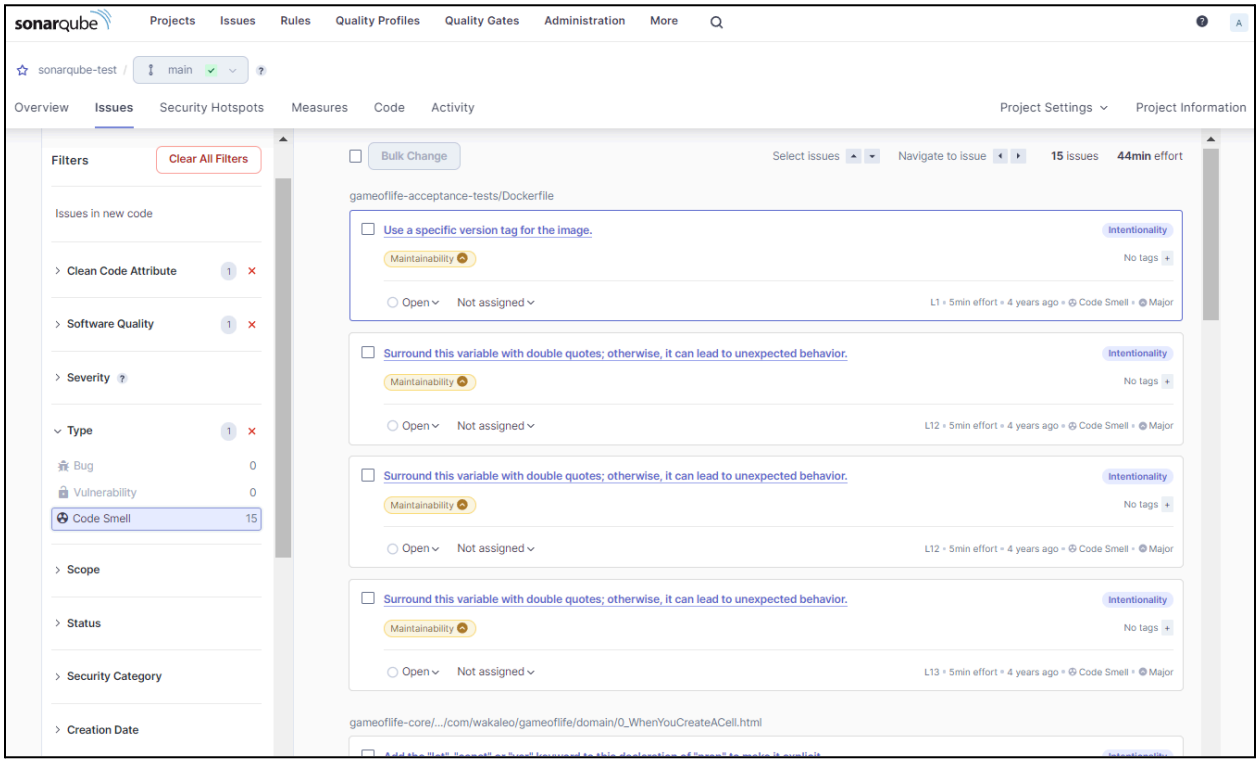
Intentionally:



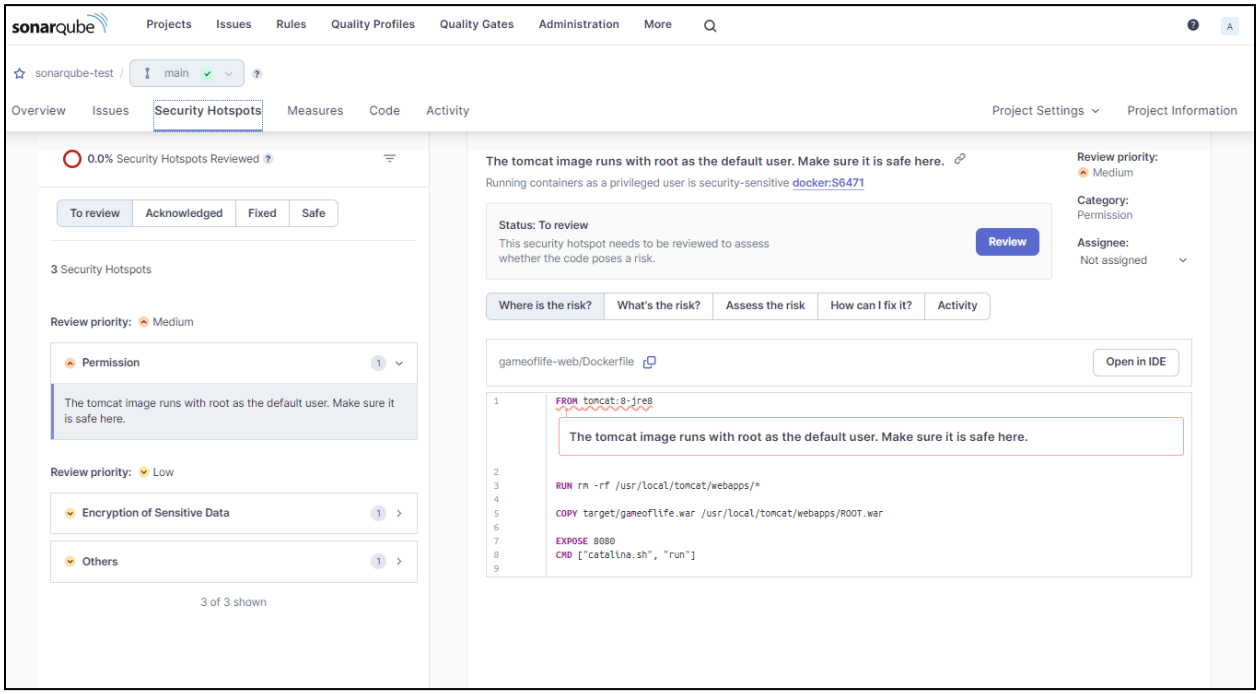
Reliability:



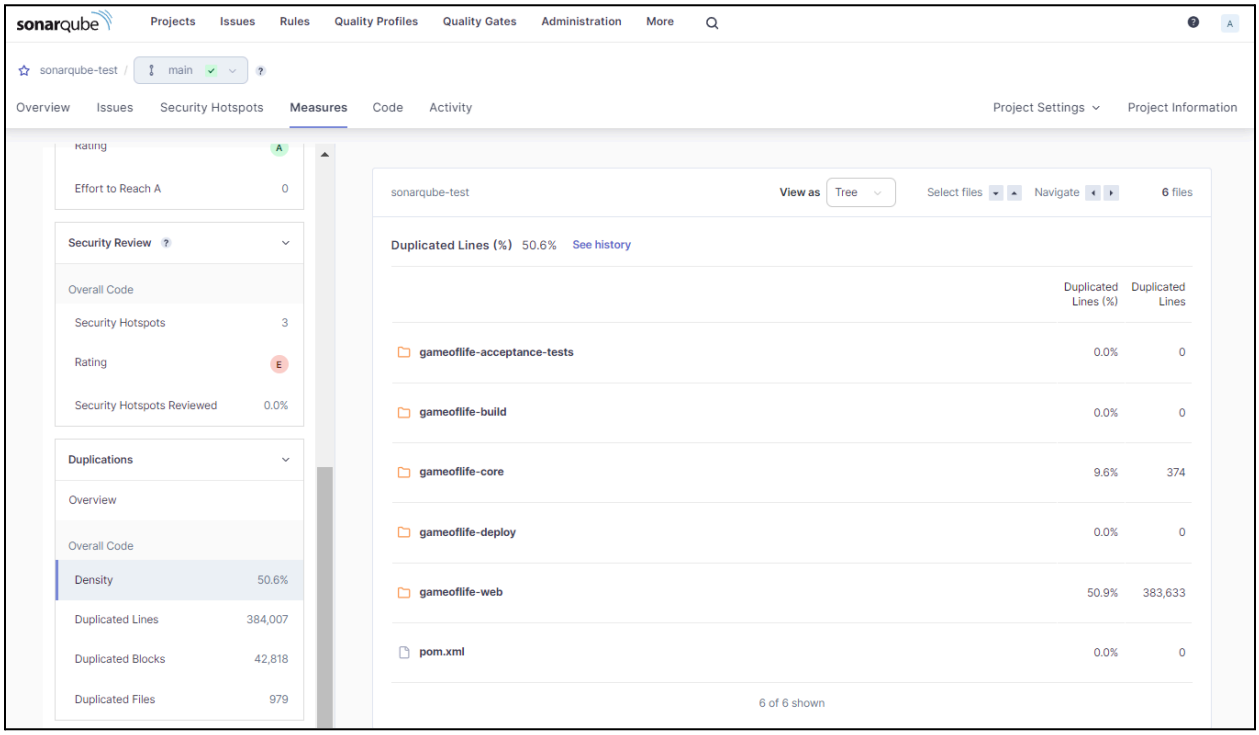
Code smells:



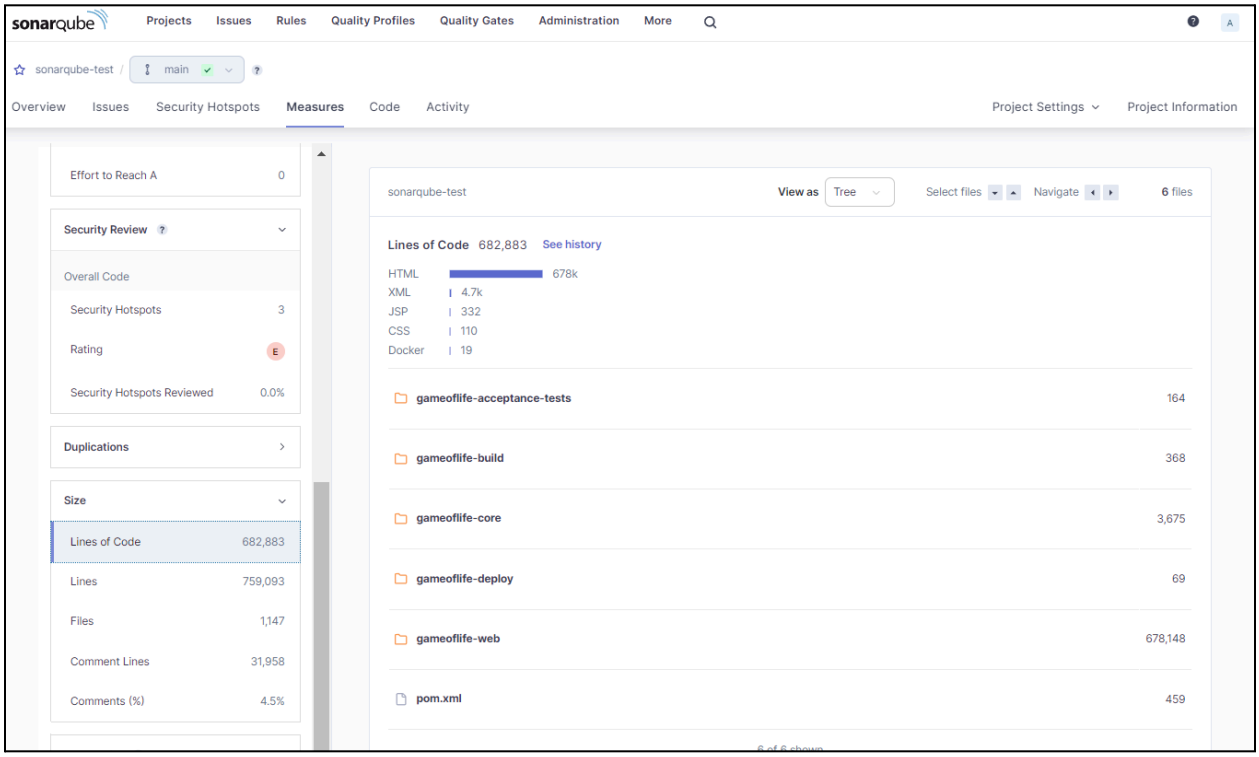
Security hotspot:



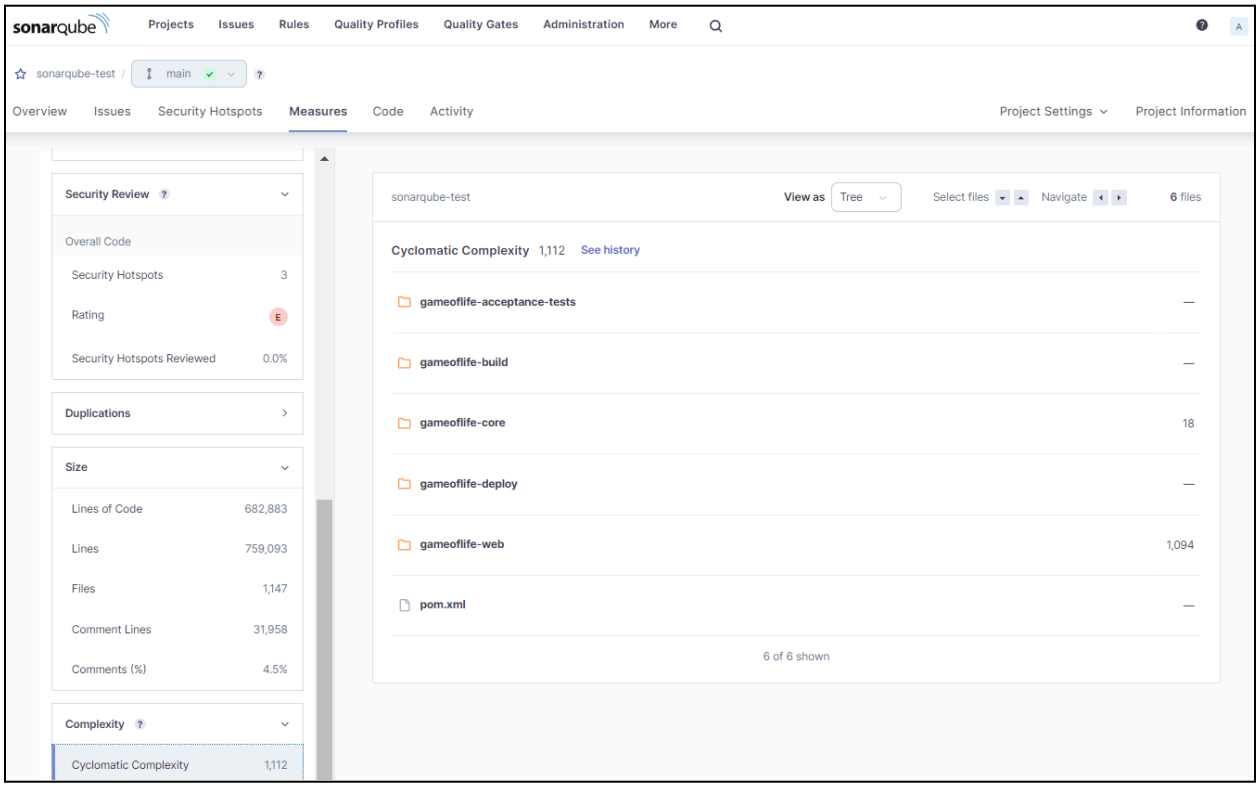
Duplicates:



Size:



Complexity:



ADVANCE DEVOPS EXP-9

ANSH SARFARE **D15A/50**

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

Theory:

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately. Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

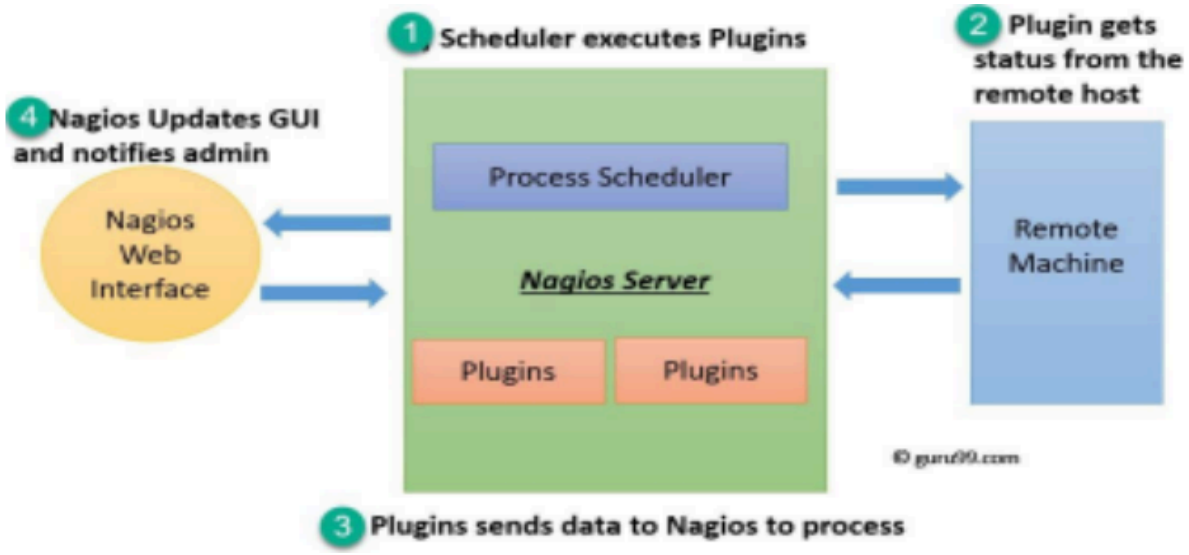
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.

● The issues, if any, can be fixed automatically as they are identified during the monitoring process
● You can monitor the entire business process and IT infrastructure with a single pass
● The product's architecture is easy to write new plugins in the language of your choice
● Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
● Utilizes topology to determine dependencies
● Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
● Helps you to define network host hierarchy using parent hosts

Nagios Architecture

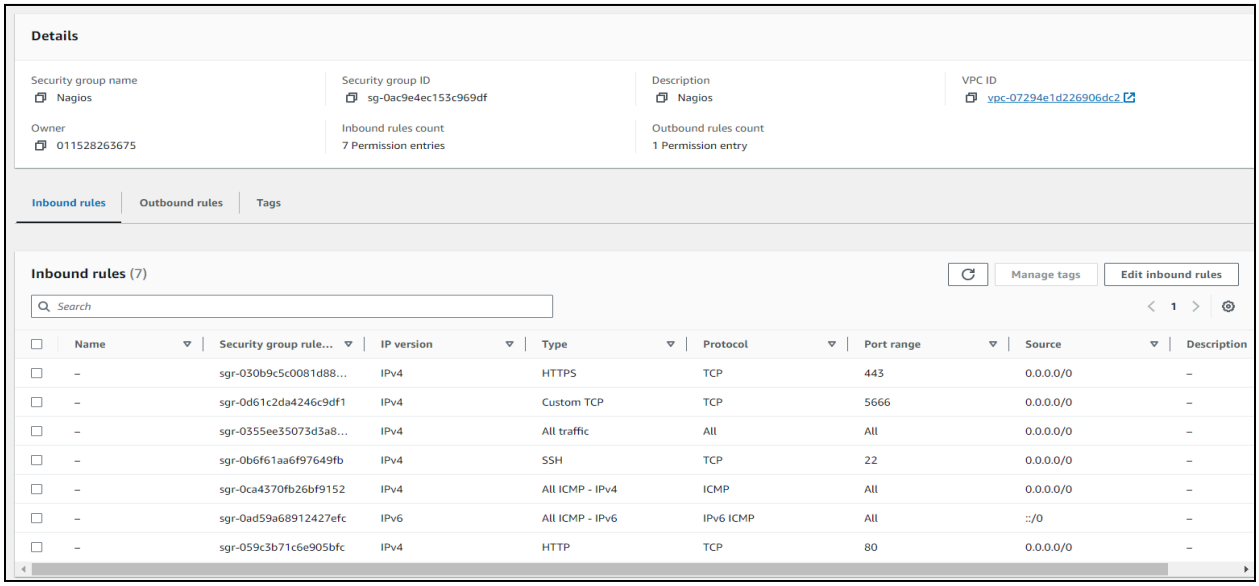
Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.



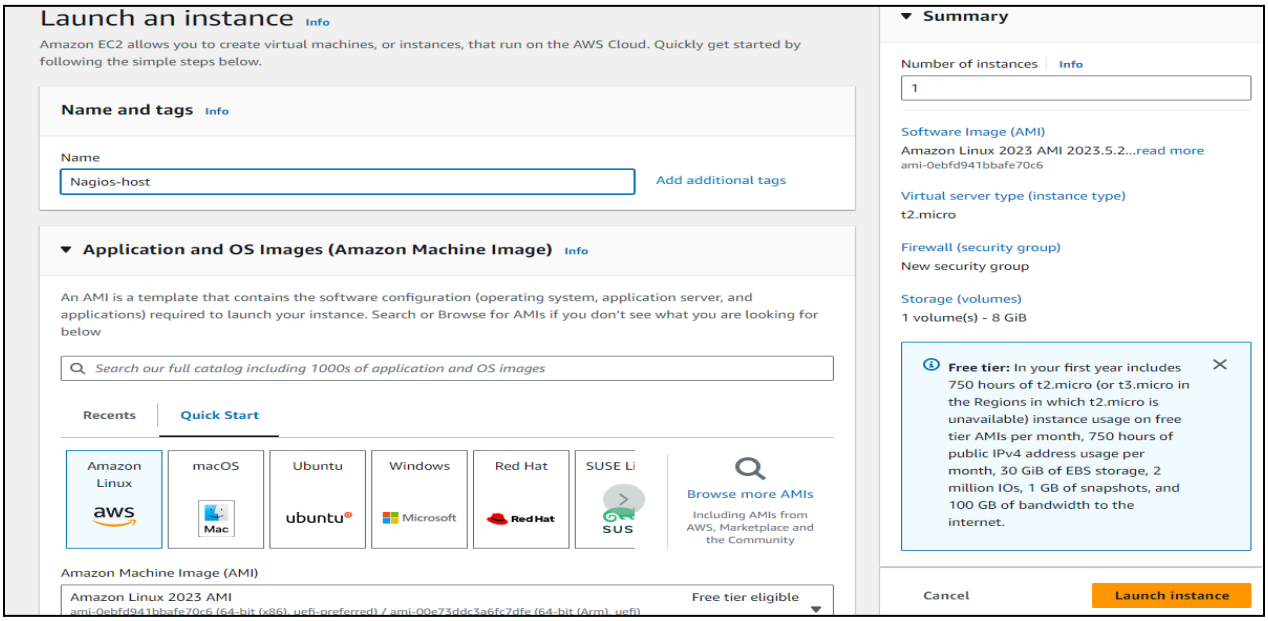
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Prerequisites: AWS Academy / AWS Personal Account

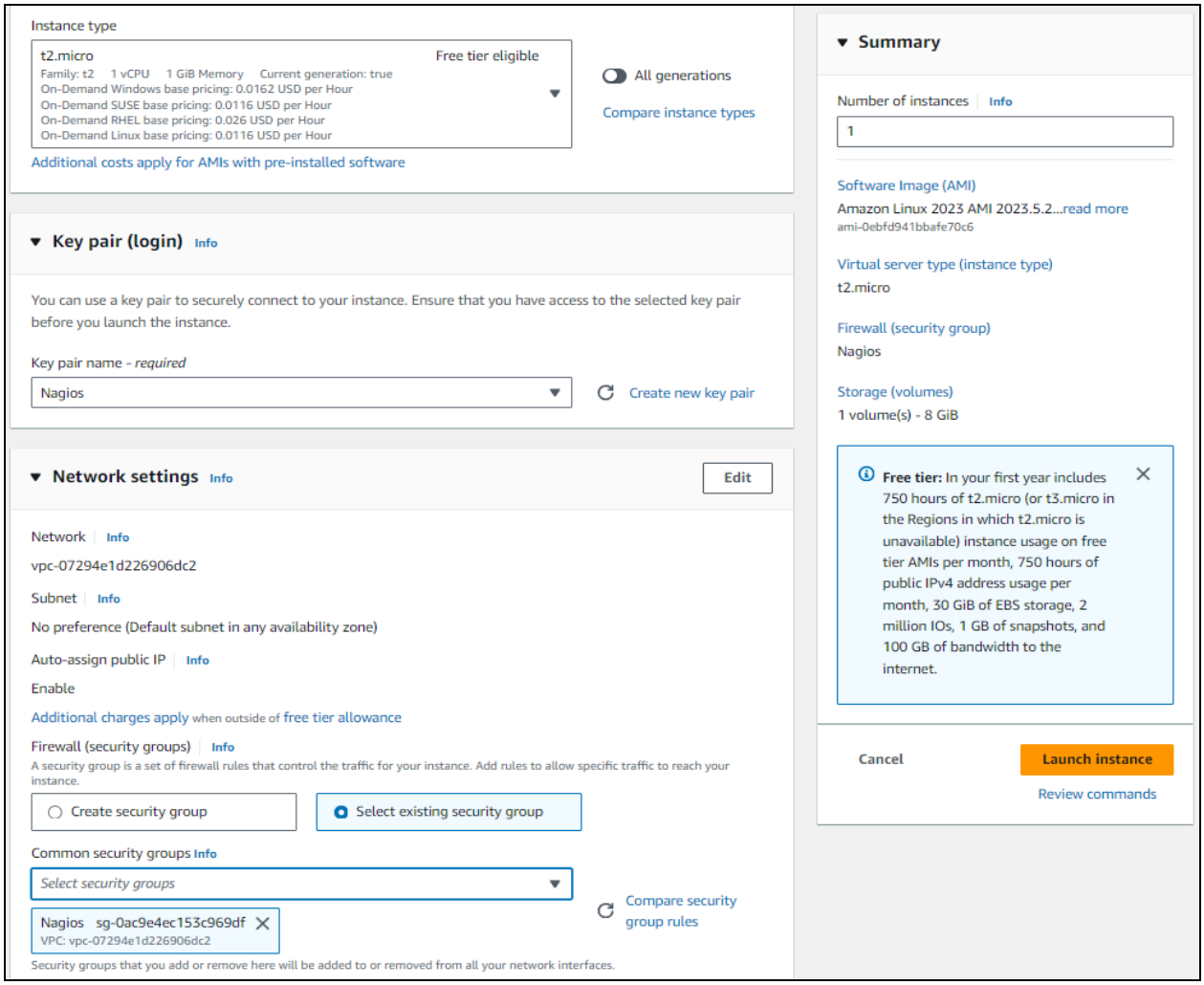
Step-1: Login to your AWS account Personal / Academy. Click on EC2 instance then click on Create Security Group. Give the name as Nagios and any description and add the following inbound rules.

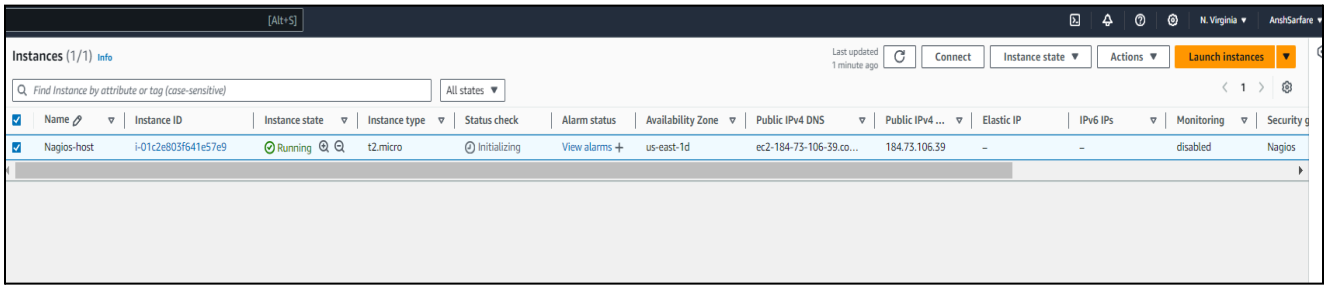


Step 2: Now Create a new EC2 instance. Name: Nagios-host ,AMI: Amazon Linux, Instance Type: t2.micro.

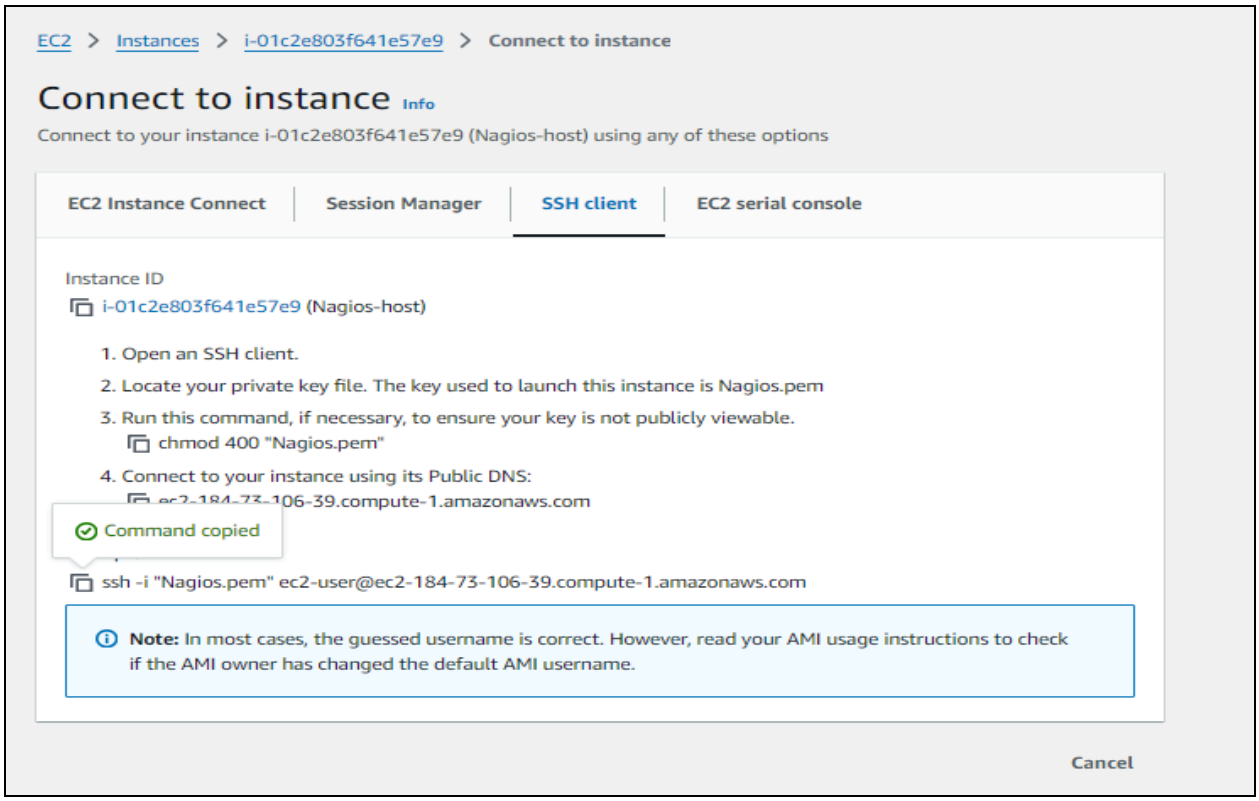


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine and select the **Existing Security Group** and select the Security Group we have created in Step 1.





Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .



Successfully connected to the instance.

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

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

PS C:\Users\Ansh> cd Desktop
PS C:\Users\Ansh\Desktop> cd Nagios
PS C:\Users\Ansh\Desktop\Nagios> ssh -i "Nagios.pem" ec2-user@ec2-184-73-106-39.compute-1.amazonaws.com
The authenticity of host 'ec2-184-73-106-39.compute-1.amazonaws.com (184.73.106.39)' can't be established.
ED25519 key fingerprint is SHA256:i7NMc6WI0yzJAd4YHlwTc+dDHM9igux+4/N/7+Lq4xU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-184-73-106-39.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-90-152 ~]$
```

Step 4: Now Run the following command to make a new user.- sudo adduser -m nagios
sudo passwd nagios

```
[ec2-user@ip-172-31-90-152 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-90-152 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-90-152 ~]$
```

Step 5: Now Run the following command to make a new user group.
sudo groupadd nagcmd ,
sudo usermod -a -G nagcmd nagios,
sudo usermod -a -G nagcmd apache

If apache is not installed on your system:

```
[ec2-user@ip-172-31-90-152 ~]$ sudo yum update -y
Last metadata expiration check: 0:13:57 ago on Mon Sep 30 17:57:12 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-152 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:15:05 ago on Mon Sep 30 17:57:12 2024.
Dependencies resolved.
================================================================================
 Package                         Architecture       Version
================================================================================
Installing:
 httpd                           x86_64             2.4.62-1.amzn2023
Installing dependencies:
 apr                             x86_64             1.7.2-2.amzn2023.0.2
 apr-util                        x86_64             1.6.3-1.amzn2023.0.1
```

```
[ec2-user@ip-172-31-90-152 ~]$ sudo groupadd nagcmd
groupadd: group 'nagcmd' already exists
[ec2-user@ip-172-31-90-152 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-90-152 ~]$ sudo usermod -a -G nagcmd httpd
usermod: user 'httpd' does not exist
[ec2-user@ip-172-31-90-152 ~]$ ps aux | grep apache
apache     22421  0.0  0.4  17020  4580 ?        S    18:12   0:00 /usr/sbin/httpd -DFOREGROUND
apache     22426  0.0  0.7 1084984 7560 ?        Sl   18:12   0:00 /usr/sbin/httpd -DFOREGROUND
apache     22427  0.0  0.7 1084984 7560 ?        Sl   18:12   0:00 /usr/sbin/httpd -DFOREGROUND
apache     22428  0.0  0.7 1248888 7560 ?        Sl   18:12   0:00 /usr/sbin/httpd -DFOREGROUND
ec2-user   26483  0.0  0.2 222312  2064 pts/0    S+   18:14   0:00 grep --color=auto apache
[ec2-user@ip-172-31-90-152 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-90-152 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-90-152 ~]$
```

Step 6: Now make a new directory and go to that directory. mkdir ~/downloads cd ~/downloads

```
[ec2-user@ip-172-31-90-152 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-90-152 downloads]$
```

Step 7: Now to download the Nagios 4.5.5 and Nagios-plugins 2.4.11 run the following commands respectively.

wget https://go.nagios.org/l/975333/2024-09-17/6kqcx

```
[ec2-user@ip-172-31-90-152 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-09-30 18:18:59--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 52.54.96.194, 3.92.120.28, 3.215.172.219, ...
Connecting to go.nagios.org (go.nagios.org)|52.54.96.194|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_conte
+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-30 18:18:59--  http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagio
aign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_cont
5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-30 18:18:59--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagi
paign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx                       100%[=================================================>]   1.97M  8.62MB/s    in 0.2s

2024-09-30 18:18:59 (8.62 MB/s) - '6kqcx' saved [2065473/2065473]
```

wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-90-152 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-30 18:20:08--  https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz          100%[=========================================================================

2024-09-30 18:20:09 (10.2 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

Step 8: Now to extract the files from the downloaded Nagios 4.5.5 run the following command. tar zxvf 6kqcx

```
[ec2-user@ip-172-31-90-152 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
```

Step 9: Now change the directory to nagios-4.5.5

```
[ec2-user@ip-172-31-90-152 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

Step 10: Now run the following command to configure../configure --with-command-group=nagcmd.
In this step you can get the error , it might be due to gcc is not installed in your system so to install it run the following command:**sudo yum groupinstall "Development Tools" -y**

```
[ec2-user@ip-172-31-90-152 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
```

At the end we have found the error of cannot find ssl header .

```
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

So run following command to install ssl. sudo yum install openssl-devel

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:39:47 ago on Mon Sep 30 17:57:12 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture            Version
================================================================================
Installing:
 openssl-devel           x86_64                  1:3.0.8-1.amzn2023.0.14

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

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

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

Complete!
```

Now rerun the command ./configure--with-command-group=nagcmd

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
```

```
          IOBroker Method:  epoll

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


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

[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ make all
cd ./base && make
```

```
*** Support Notes *******************************************

If you have questions about configuring or running Nagios,
please make sure that you:

      - Look at the sample config files
      - Read the documentation on the Nagios Library at:
            https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

      - What version of Nagios you are using
      - What version of the plugins you are using
      - Relevant snippets from your config files
      - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

        https://support.nagios.com

**************************************************************

Enjoy.
```

Step 11: Now run the following commands to steup the Nagios. sudo make install

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
       /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/s
done
```

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

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

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

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

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

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
```

sudo make install-init

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

sudo make install-config

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

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

sudo make install-webconf

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

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

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

Now to restart the httpd service run the following command. sudo service httpd restart

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$
```

Step 12: Now to extract the files from the downloaded Nagios plugin 2.4.11 run the following command first change the directory.
cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-90-152 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-90-152 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
```

Step 13: Now change the directory to nagios-plugins-2.4.11 and run the config command to configure.
cd nagios-plugins-2.4.11
./configure--with-nagios-user=nagios--with-nagios-group=nagios

```
[ec2-user@ip-172-31-90-152 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-90-152 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
```

Step 14: Run the following commands to check nagios and start it.
sudo chkconfig--add nagios

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-90-152 nagios-plugins-2.4.11]$
```

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

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0

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

cd
sudo service nagios start

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-90-152 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-90-152 ~]$
```

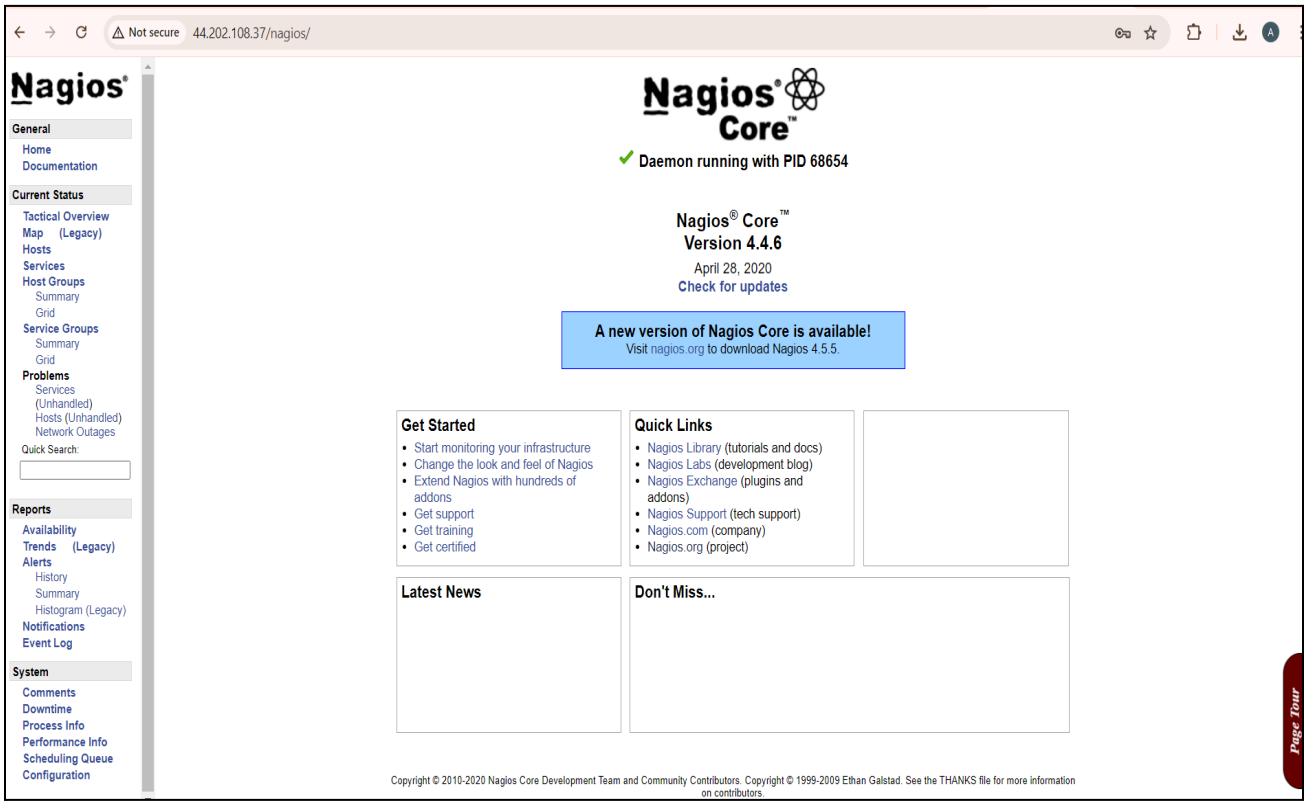
sudo /usr/local/nagios/bin/nagios-v /usr/local/nagios/etc/nagios.cfg

```
[ec2-user@ip-172-31-90-152 ~]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0

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

sudo systemctl restart nagios
sudo systemctl status nagios

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.3.3]$ cd
[ec2-user@ip-172-31-90-152 ~]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-90-152 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-09-30 19:41:36 UTC; 7s ago
       Docs: https://www.nagios.org/documentation
    Process: 80238 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 80239 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 80240 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.0M
        CPU: 15ms
     CGroup: /system.slice/nagios.service
             ├─80240 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─80241 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80242 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80243 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80244 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─80245 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

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

Step 15: We can see we have successfully launched the Nagios now . Open http:///nagios/ here it is **http://44.202.108.37/nagios** we can see the running web page of nagios.



ADVANCE DEVOPS EXP-10

ANSH SARFARE **D15A/50**

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

Theory:

Port and Service Monitoring

- Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports.
- This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

Windows/Linux Server Monitoring

- Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems.
- It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

Prerequisites:

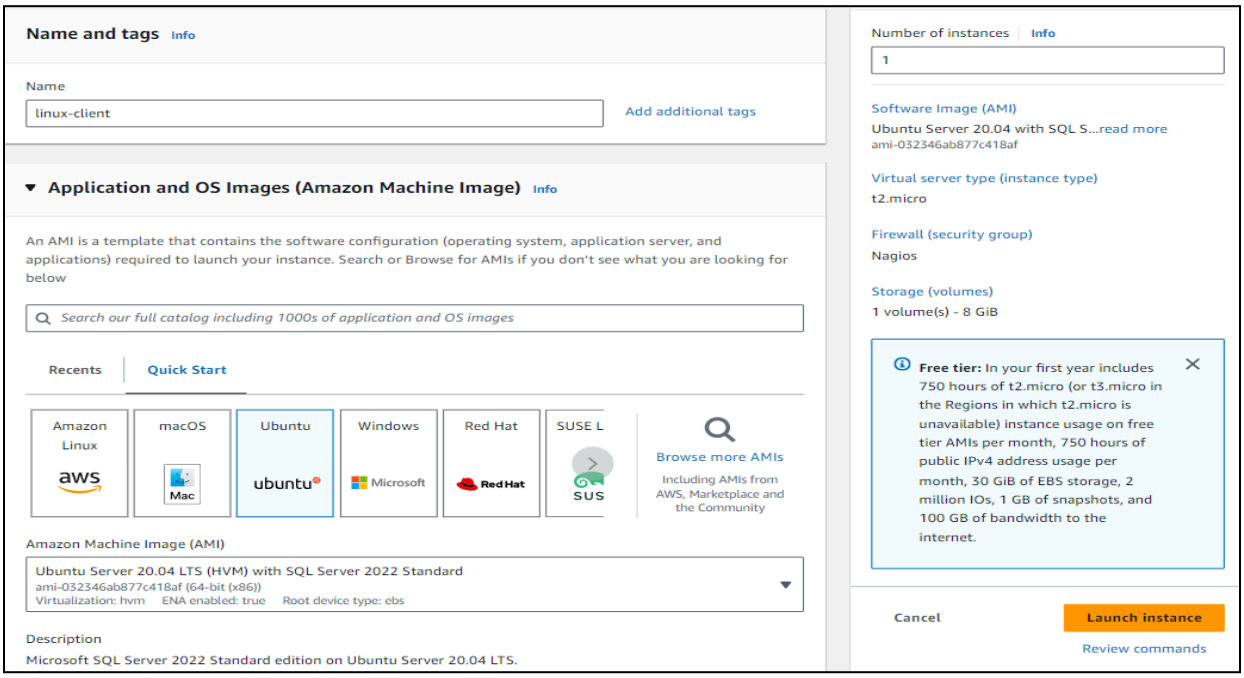
AWS Academy / AWS Personal account.
Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring using Nagios:

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

```
[ec2-user@ip-172-31-90-152 nagios-plugins-2.3.3]$ cd
[ec2-user@ip-172-31-90-152 ~]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-90-152 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-09-30 19:41:36 UTC; 7s ago
       Docs: https://www.nagios.org/documentation
    Process: 80238 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 80239 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 80240 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.0M
        CPU: 15ms
     CGroup: /system.slice/nagios.service
             ├─80240 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─80241 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80242 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80243 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─80244 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─80245 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

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

Step-2. Create an Ubuntu 20.04 Server EC2 Instance



Step-3: Verify Nagios Process on the Server

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

Step-4: Become Root User and Create Directories-

sudo su , mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
and to copy the same config file- cp /usr/local/nagios/etc/objects/localhost.cfg, /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

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

i-0ae1aae975bae3b7a (nagios-host)

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

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

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

# Define a host for the local machine

define host {

    use                     linux-server            ; Name of host template to use
                                                    ; This host definition will inherit all variables that are defined
                                                    ; in (or inherited by) the linux-server host template definition.
    host_name               linuxserver
    alias                   linuxserver
    address                 35.174.139.220
}

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

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name          linux-servers1          ; The name of the hostgroup
    alias                   Linux Servers           ; Long name of the group
    members                 localhost               ; Comma separated list of hosts that belong to this group
}
                                                                [ Read 157 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where
```

Step-6: Update Nagios Configuration
sudo nano /usr/local/nagios/etc/nagios.cfg
Add the command - cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

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


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

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

Step-7: Verify Configuration Files

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

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

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

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

Running pre-flight check on configuration data...

Checking objects...
       Checked 8 services.
       Checked 2 hosts.
       Checked 2 host groups.
       Checked 0 service groups.
       Checked 1 contacts.
       Checked 1 contact groups.
       Checked 24 commands.
       Checked 5 time periods.
       Checked 0 host escalations.
       Checked 0 service escalations.
Checking for circular paths...
       Checked 2 hosts
       Checked 0 service dependencies
       Checked 0 host dependencies
       Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0
```

Step-8: Restart Nagios Service

sudo systemctl restart nagios

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

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

```
ubuntu@ip-172-31-86-24:~$ sudo apt update -y
sudo apt install gcc -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [274 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8652 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [379 kB]
```

Step-11: Edit NRPE Configuration File

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

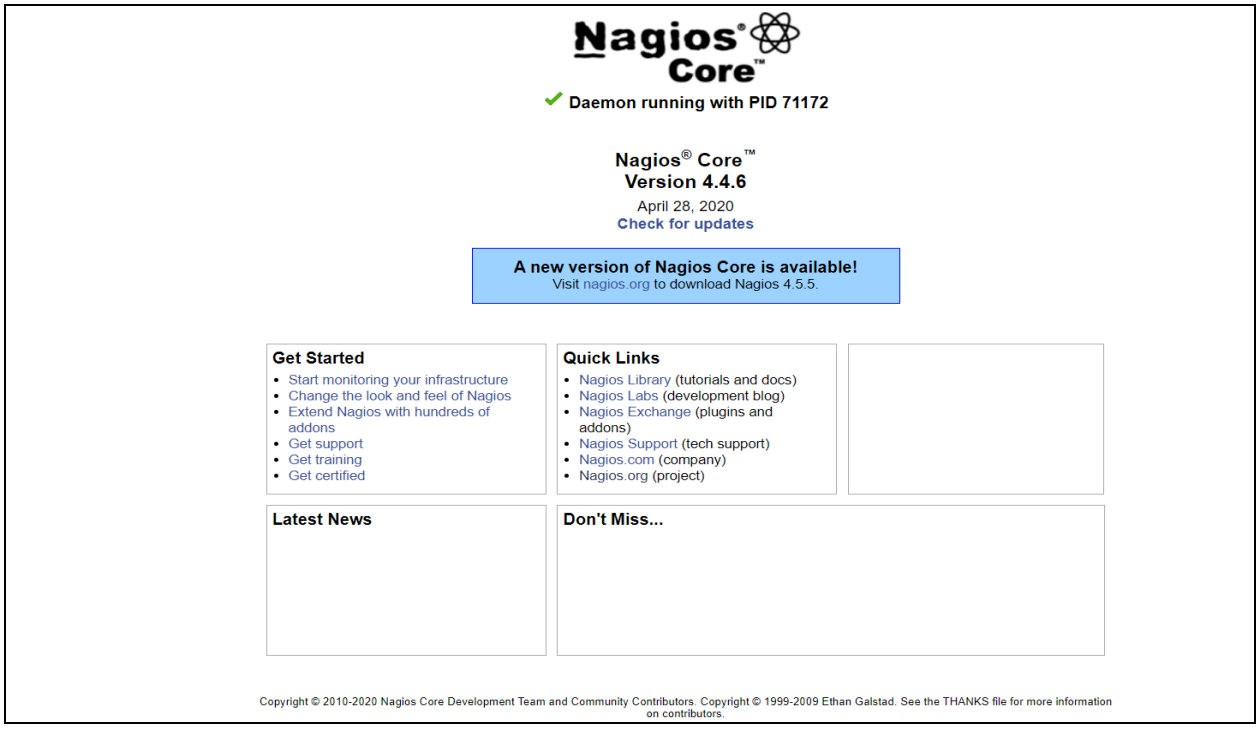
```
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running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

allowed_hosts=127.0.0.1,35.174.139.220



# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed.  This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments

dont_blame_nrpe=0
```

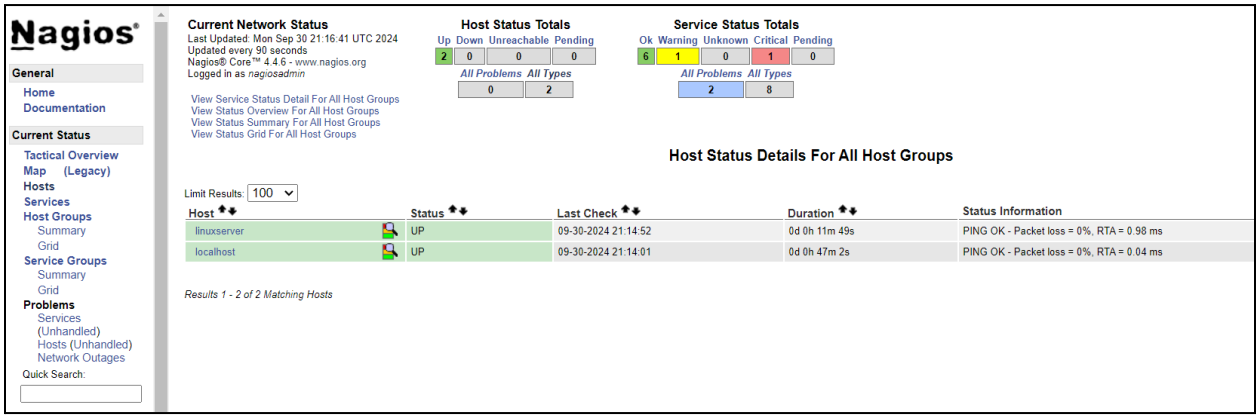
Step-12: Restart NRPE Server

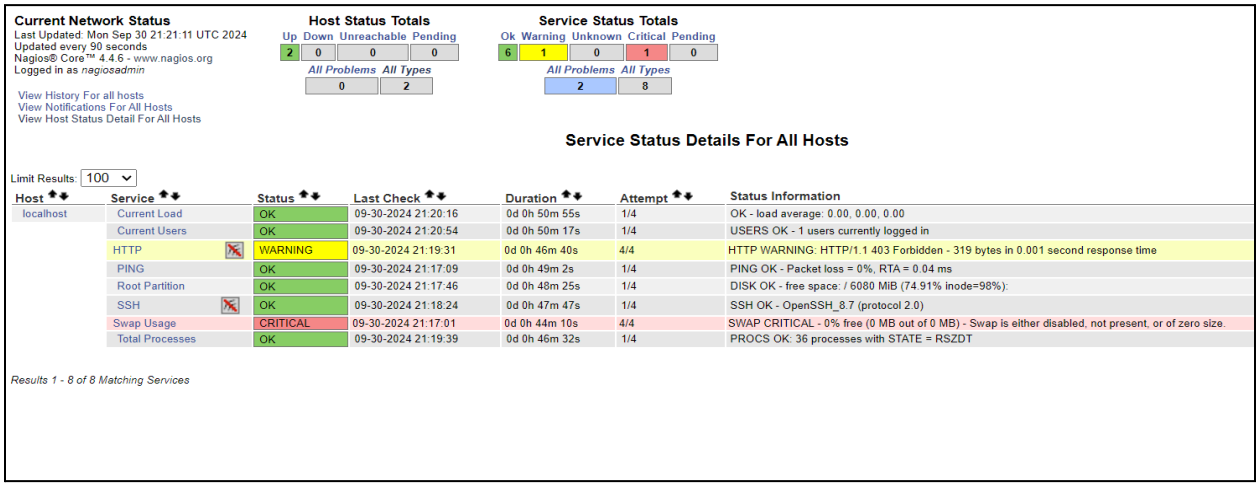
Commands -
sudo systemctl restart nagios-nrpe-server

Step-13: Check Nagios Dashboard

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


Nagios Core
Daemon running with PID 71172
Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates
A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.
Get Started
Start monitoring your infrastructure
Change the look and feel of Nagios
Extend Nagios with hundreds of addons
Get support
Get training
Get certified
Quick Links
Nagios Library (tutorials and docs)
Nagios Labs (development blog)
Nagios Exchange (plugins and addons)
Nagios Support (tech support)
Nagios.com (company)
Nagios.org (project)
Latest News
Don't Miss...
Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.


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


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

ADVANCE DEVOPS EXP-11

ANSH SARFARE **D15A/50**

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

Theory:

AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

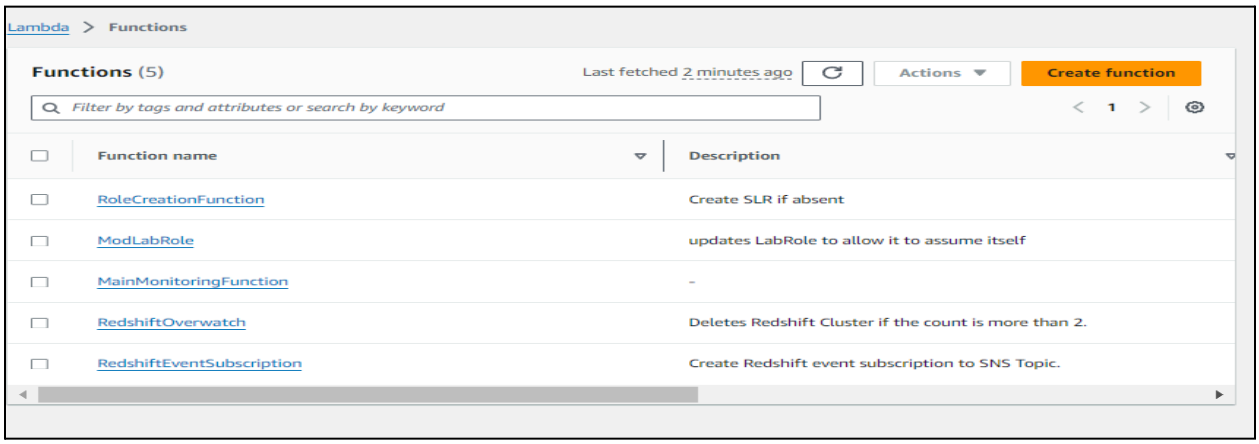
Lambda Workflow

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- Set Event Sources: Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring reserved concurrency to manage traffic.
- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

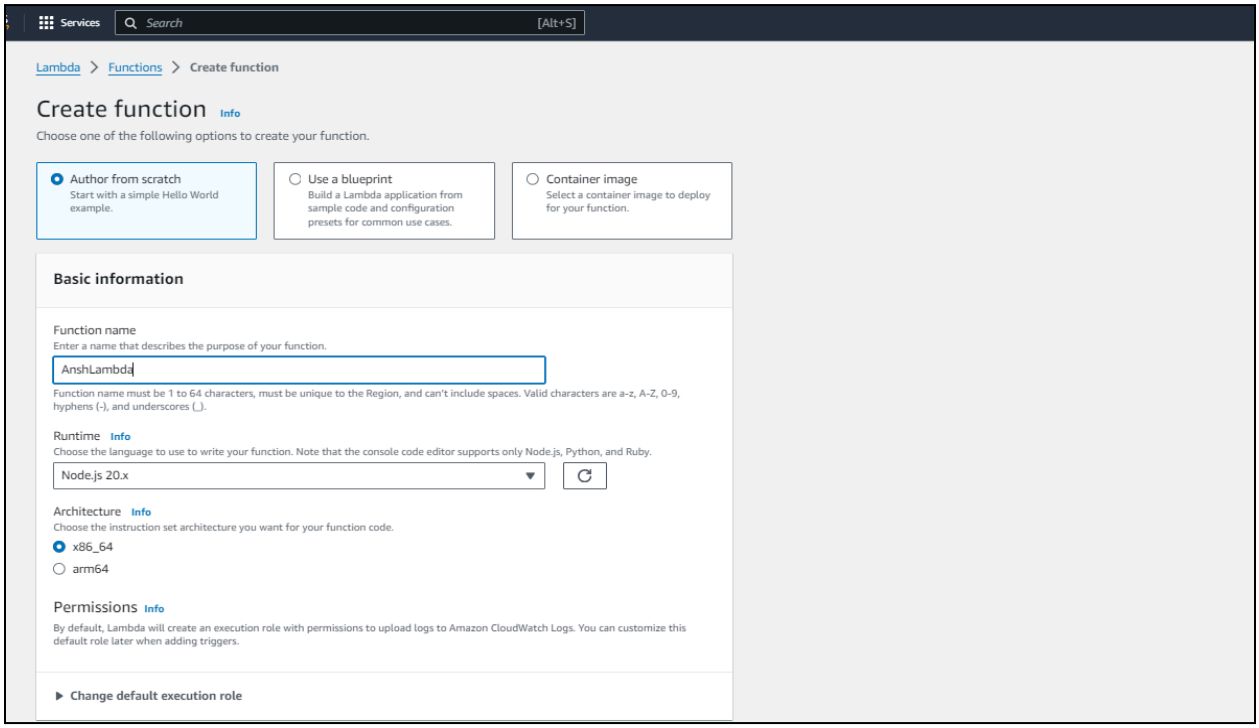
AWS Lambda Functions

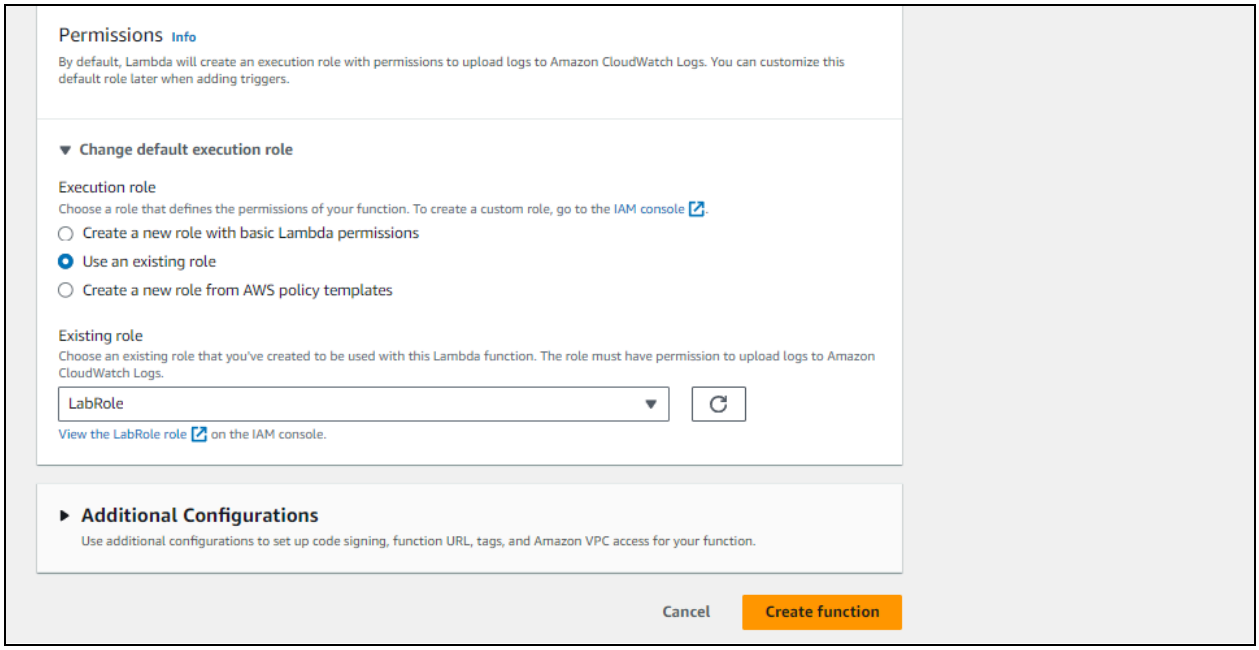
- **Python:** Great for quick development with its rich standard library and support for lightweight tasks.
- **Java:** Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

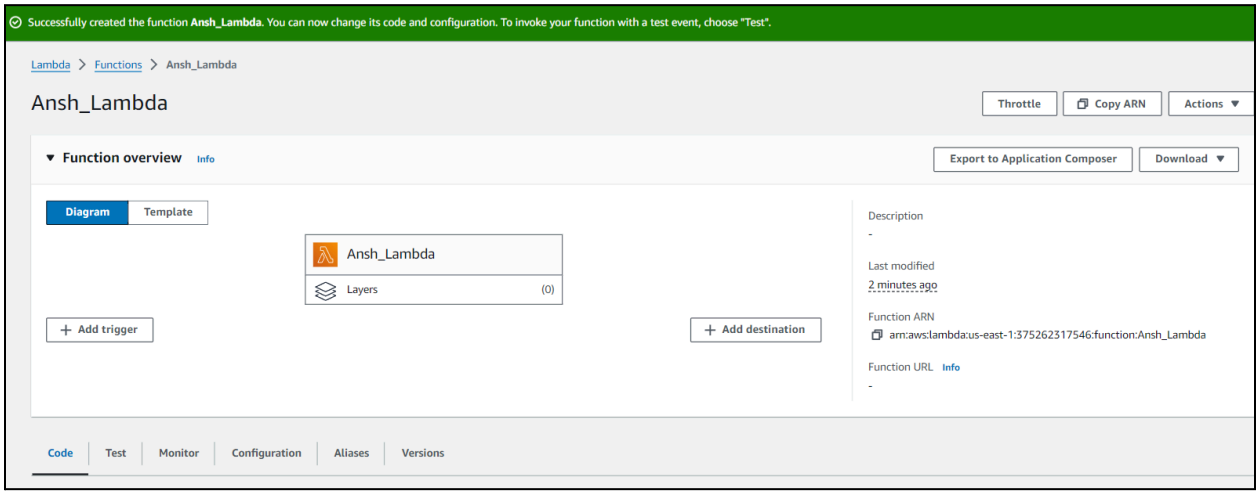
Step 1: Login to your AWS Personal/Academy Accout.Open lambda andclick on create function button.



Step 2: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Exceution role to Create a new role with basic Lambda permissions.

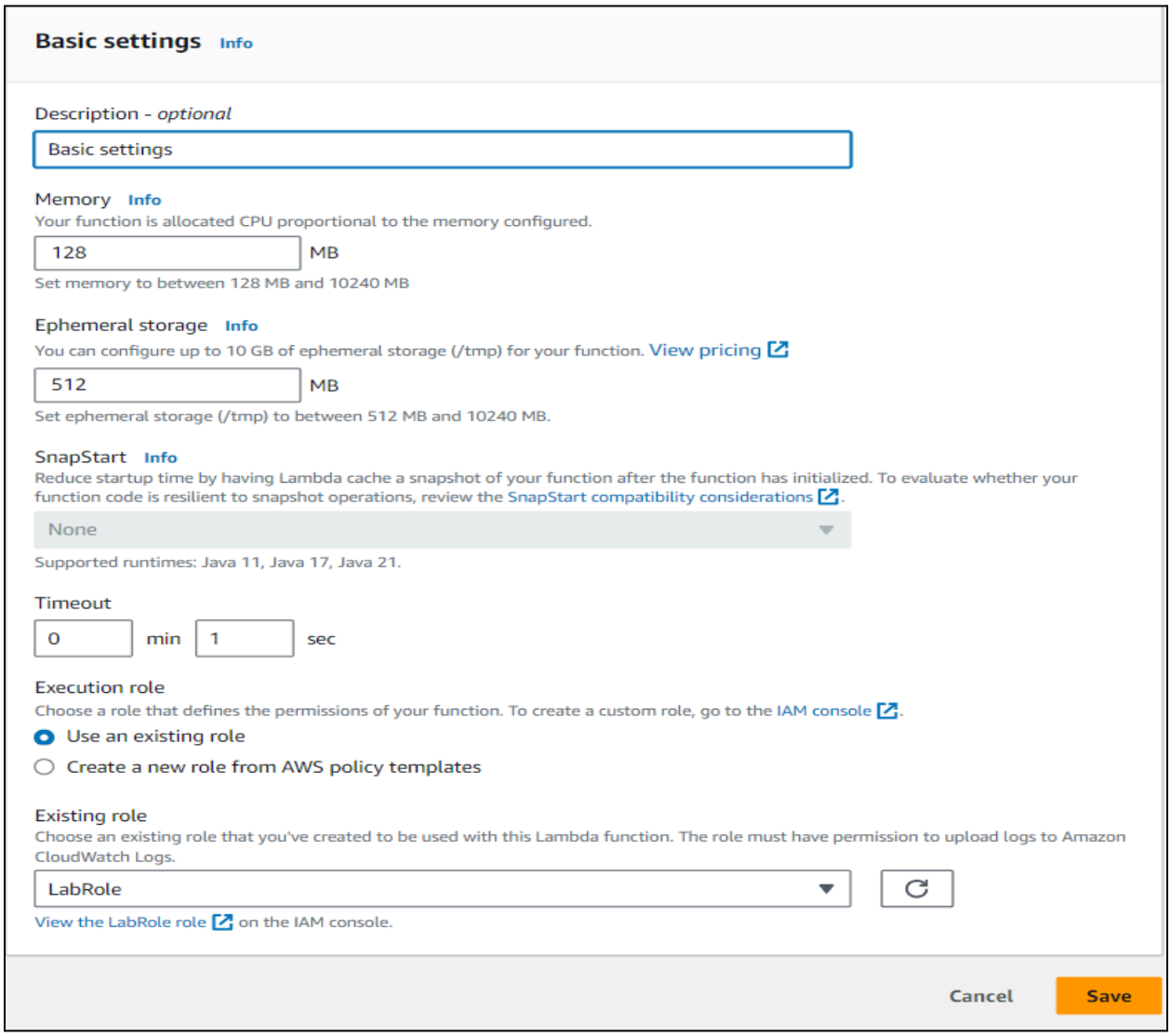




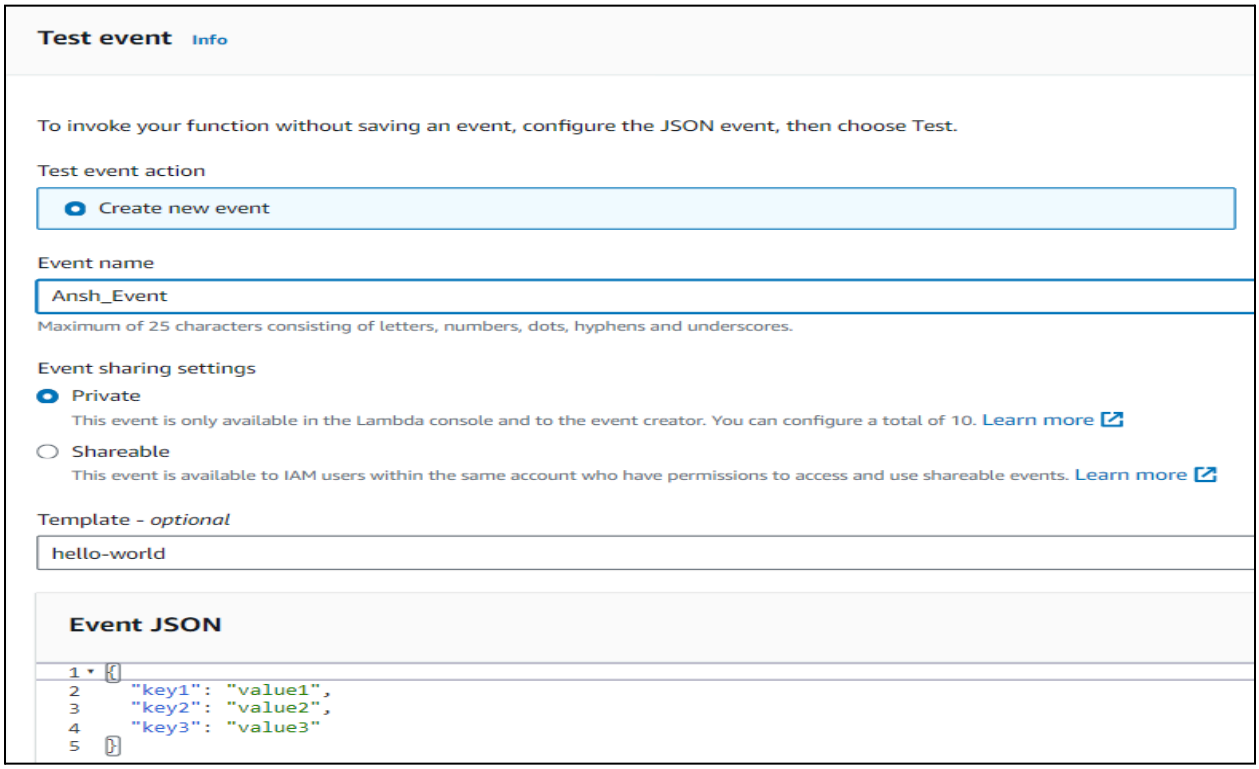


So See or Edit the basic settings go to configuration then click on edit general setting.

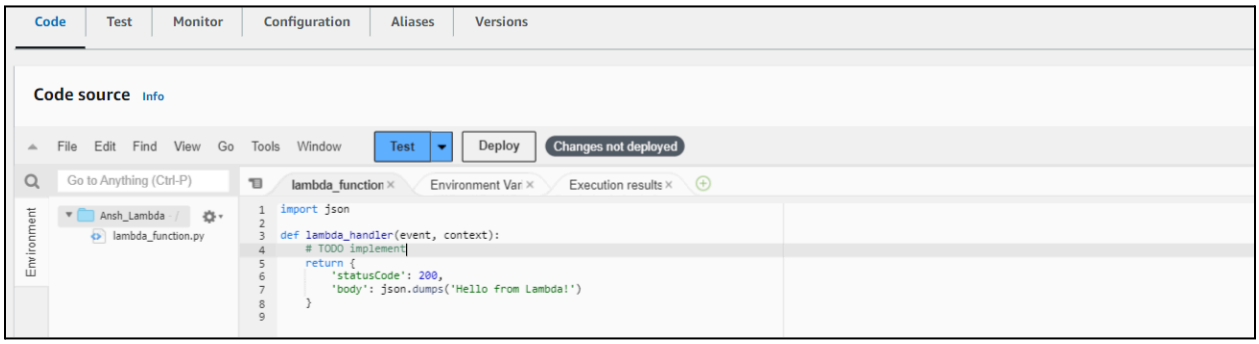
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

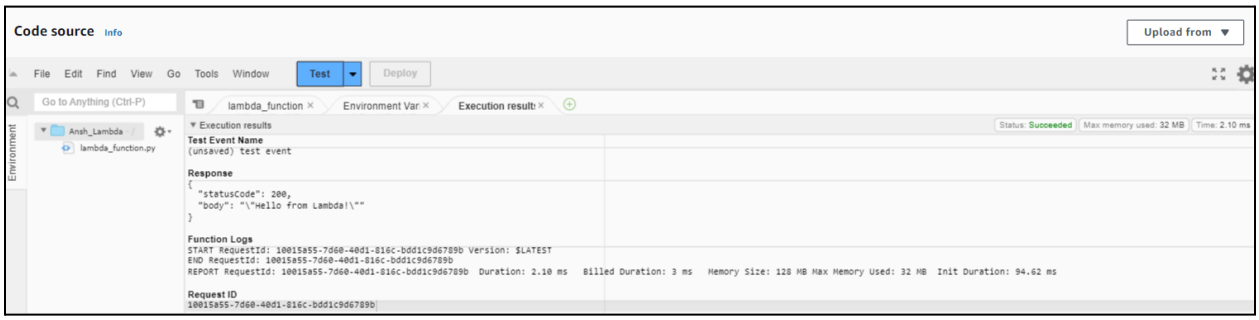


Step 3: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.



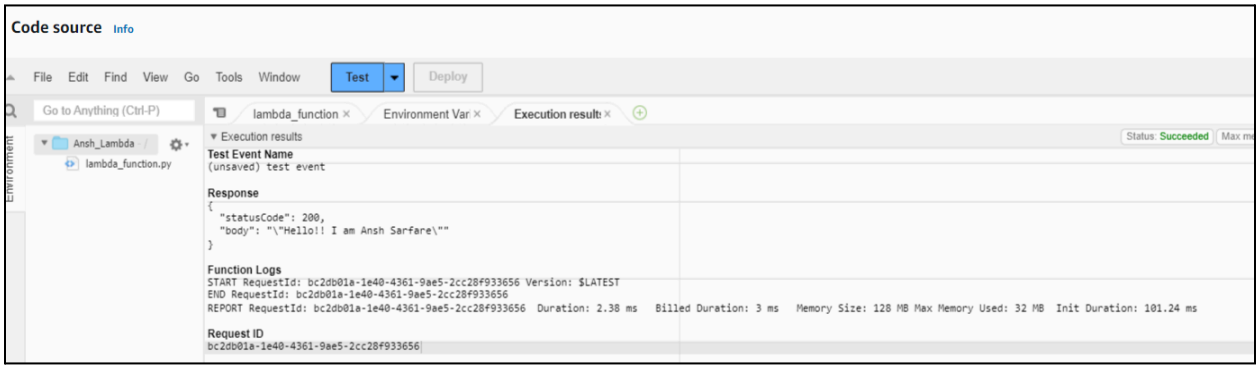
Step 4: Now In Code section select the created event from the dropdown of test then click on test . You will see the below output.





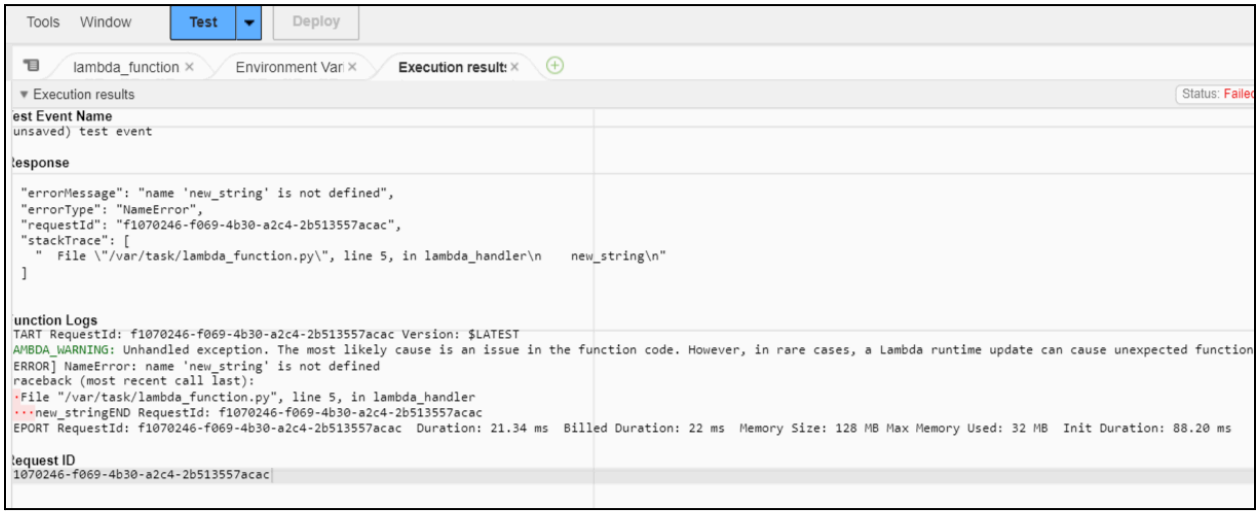
Step 5: You can edit your lambda function code. I have changed the code to display the new String.

Step 6: Now click on the test and observe the output. We can see the status code 200 and your string output and function logs. On successful deployment.



This error had occurred because the new string was not passed properly in the return statement

error:



ADVANCE DEVOPS EXP-12

ANSH SARFARE **D15A/50**

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

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

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

2. Create the Lambda Function:
○ Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java. ○ Write code that logs a message like “An Image has been added” when triggered.

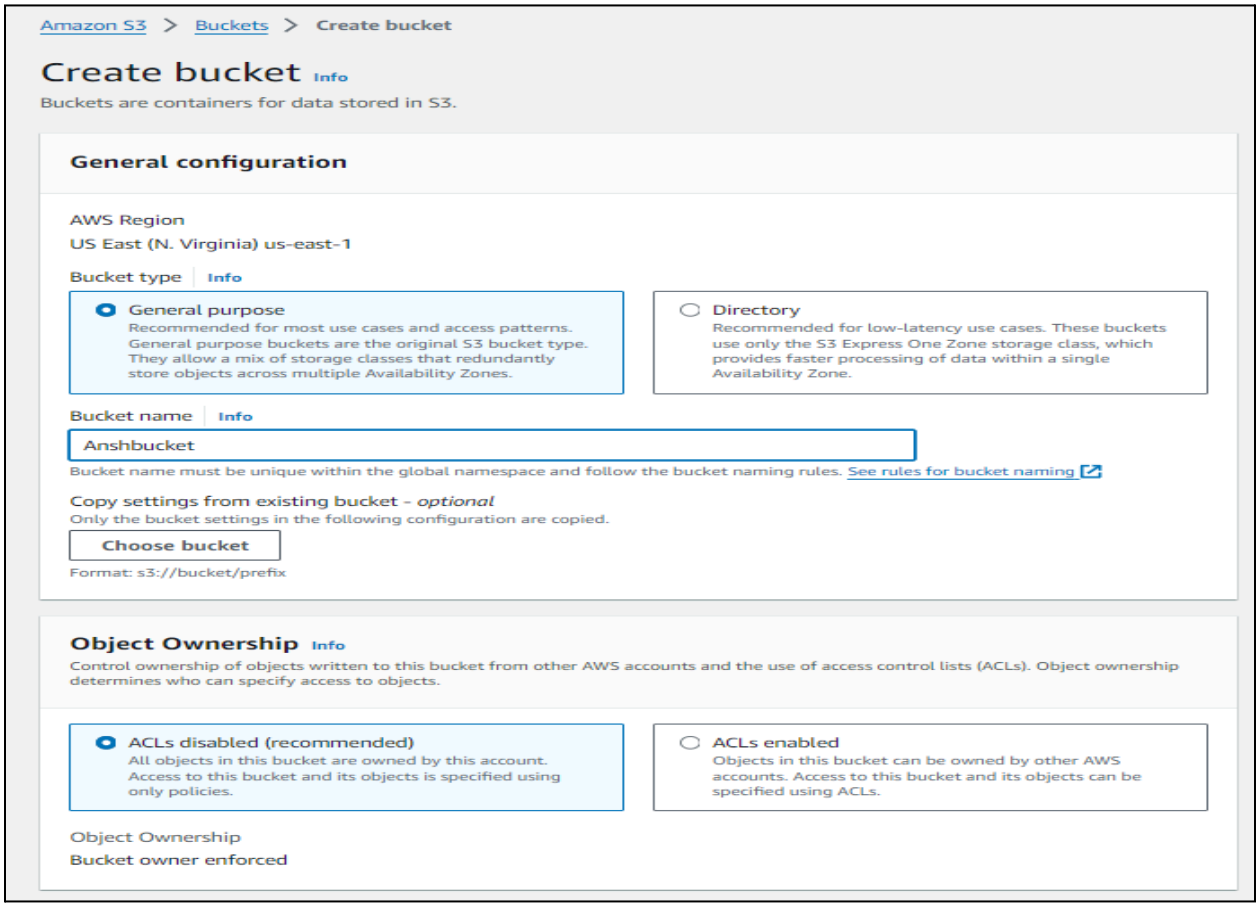
3. Set Up Permissions:
○ Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

4. Configure S3 Trigger:
○ Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

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

Step 1: Login to your AWS Personal account. Now open S3 from services and click on create S3 bucket and create a bucket.

Step 2: Now Give a name to the Bucket, select general purpose project and deselect the Block public access and keep other this to default.



Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- ☐ **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ **Turning off block all public access might result in this bucket and the objects within becoming public**

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☐ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

✓ **Successfully created bucket "anshbucketaws"**
To upload files and folders, or to configure additional bucket settings, choose **View details**.

View details

Amazon S3 > Buckets

▸ **Account snapshot** - *updated every 24 hours*
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

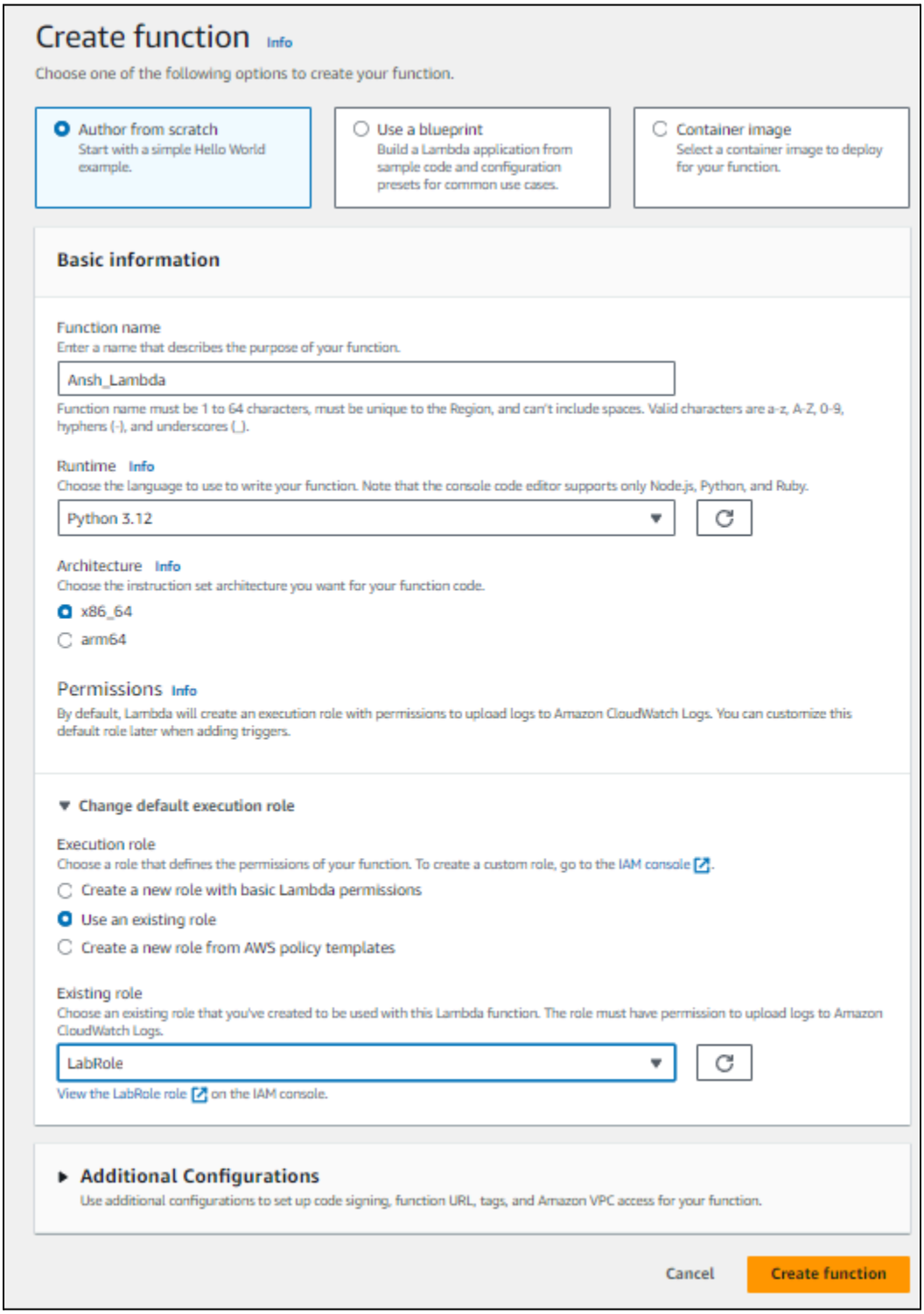
Copy ARN | Empty | Delete | Create bucket

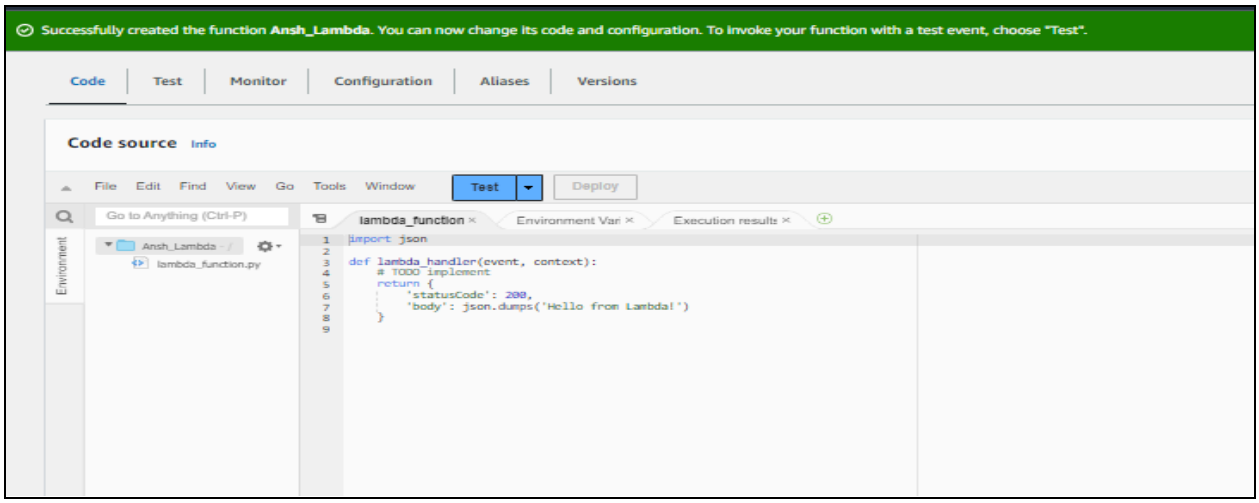
Find buckets by name

1

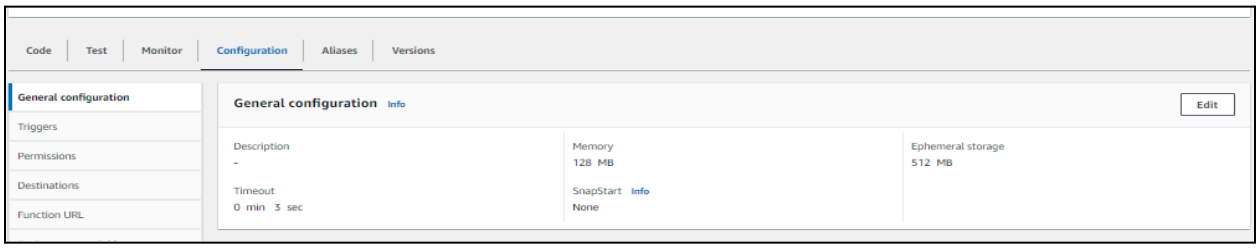
Name	AWS Region	IAM Access Analyzer	Creation date
anshbucketaws	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 4, 2024, 12:35:42 (UTC+05:30)

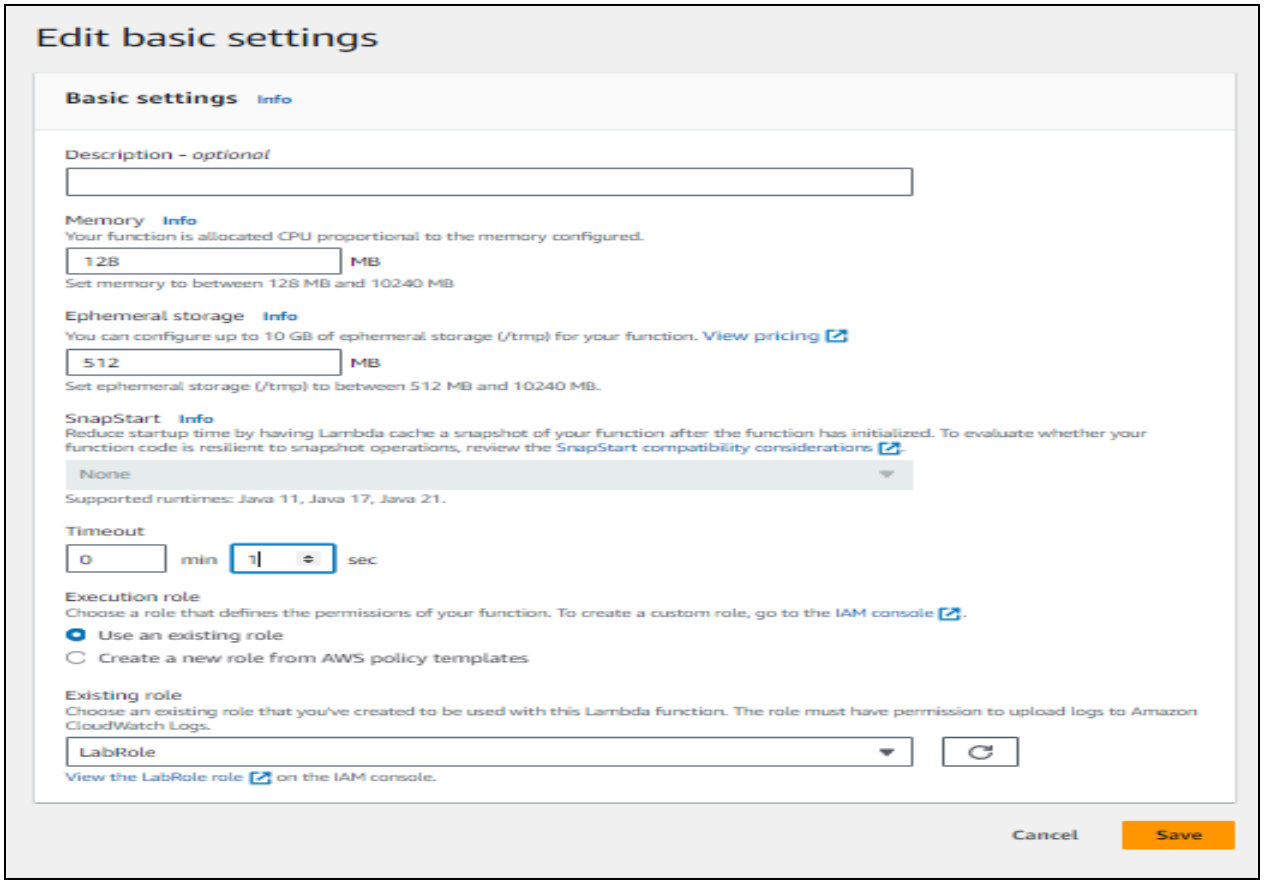
Step 3: Open lambda console and click on create function button.Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and existing Execution role



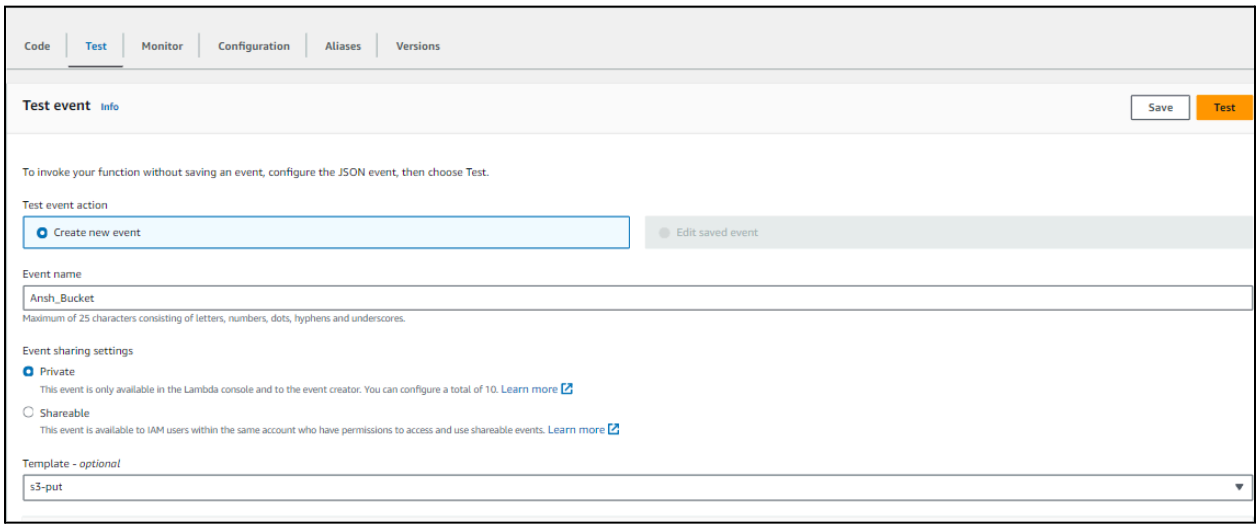


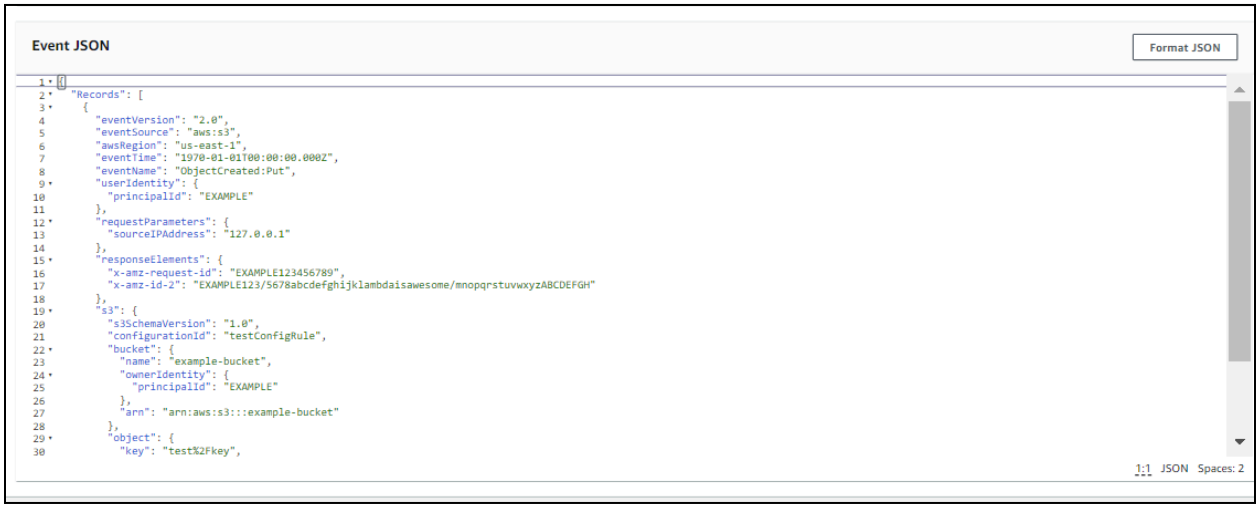
So See or Edit the basic settings go to configuration then click on edit general setting.



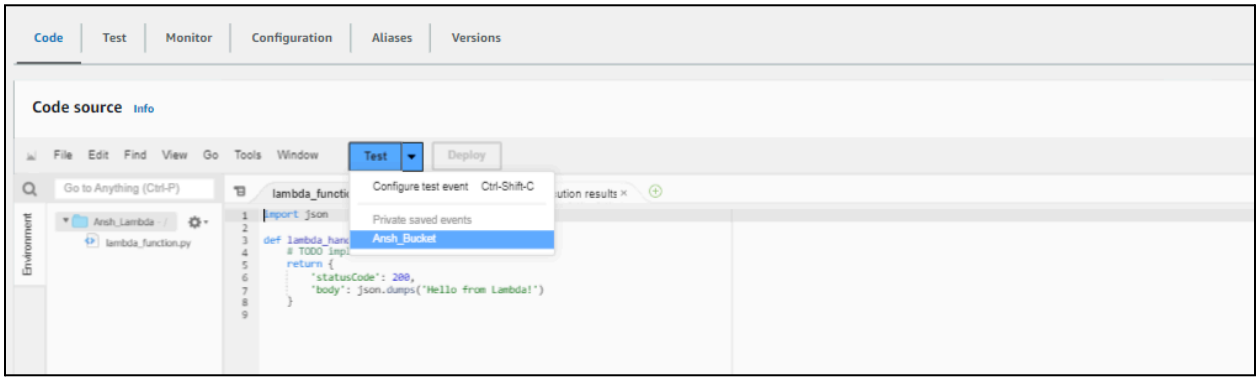


Step 4: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

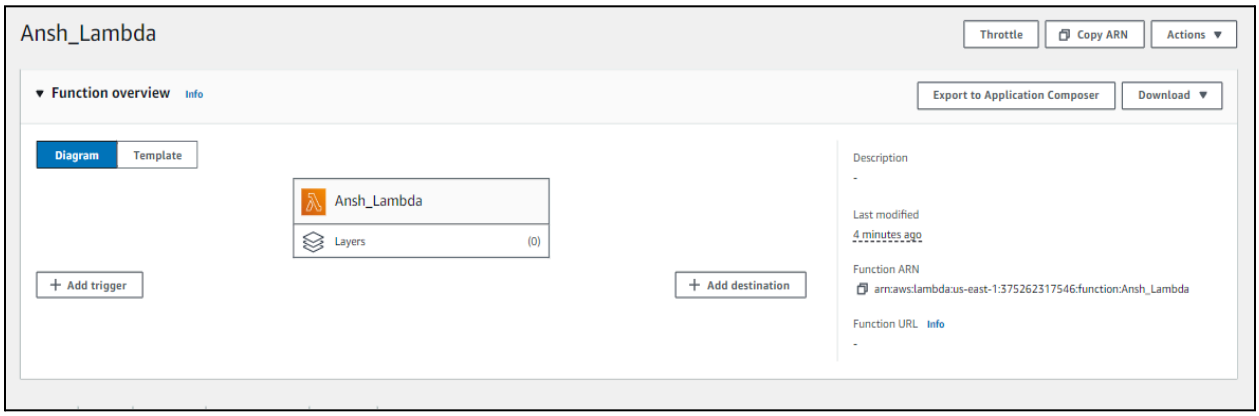




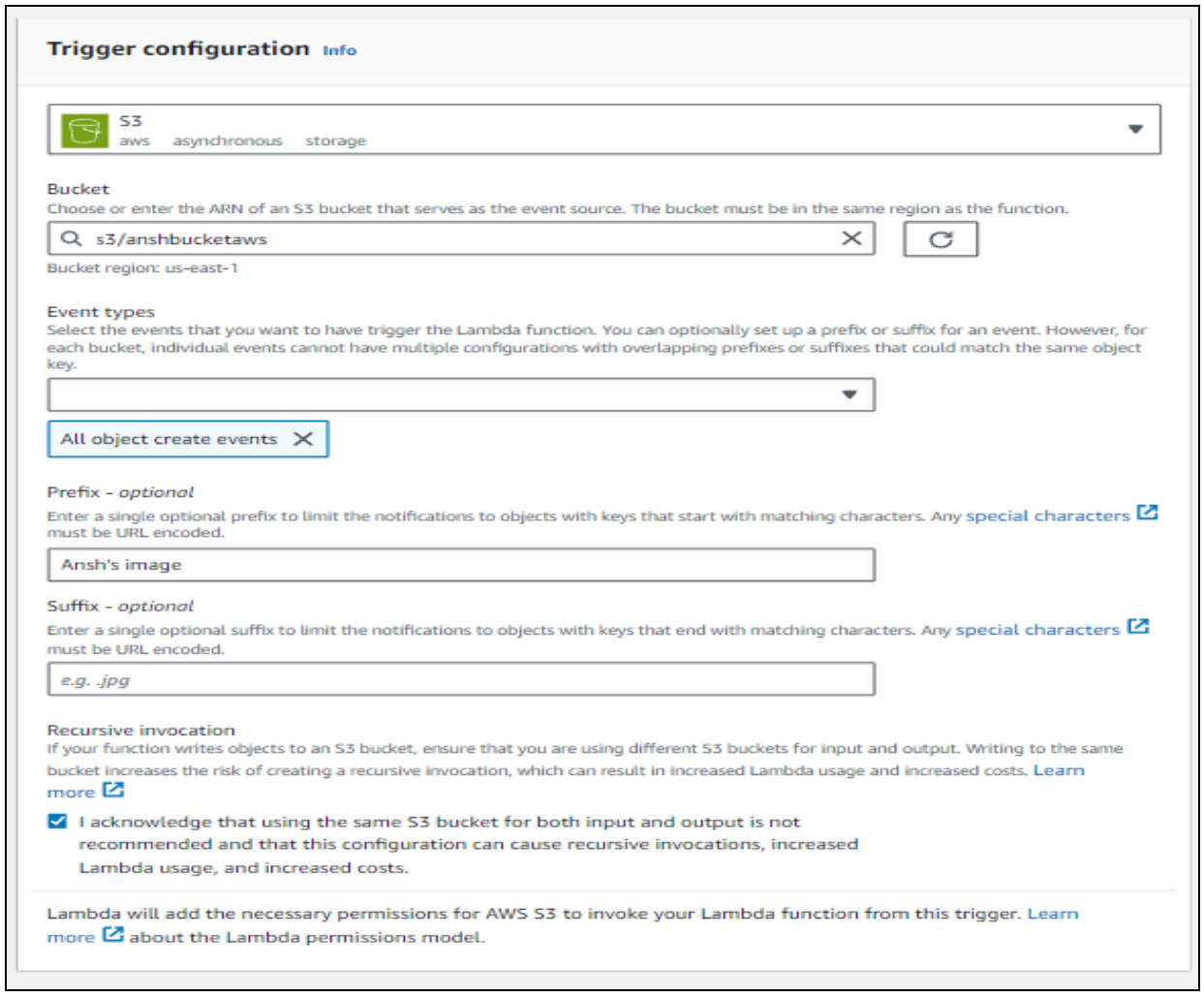
Step 5: Now In Code section select the created event from the dropdown .



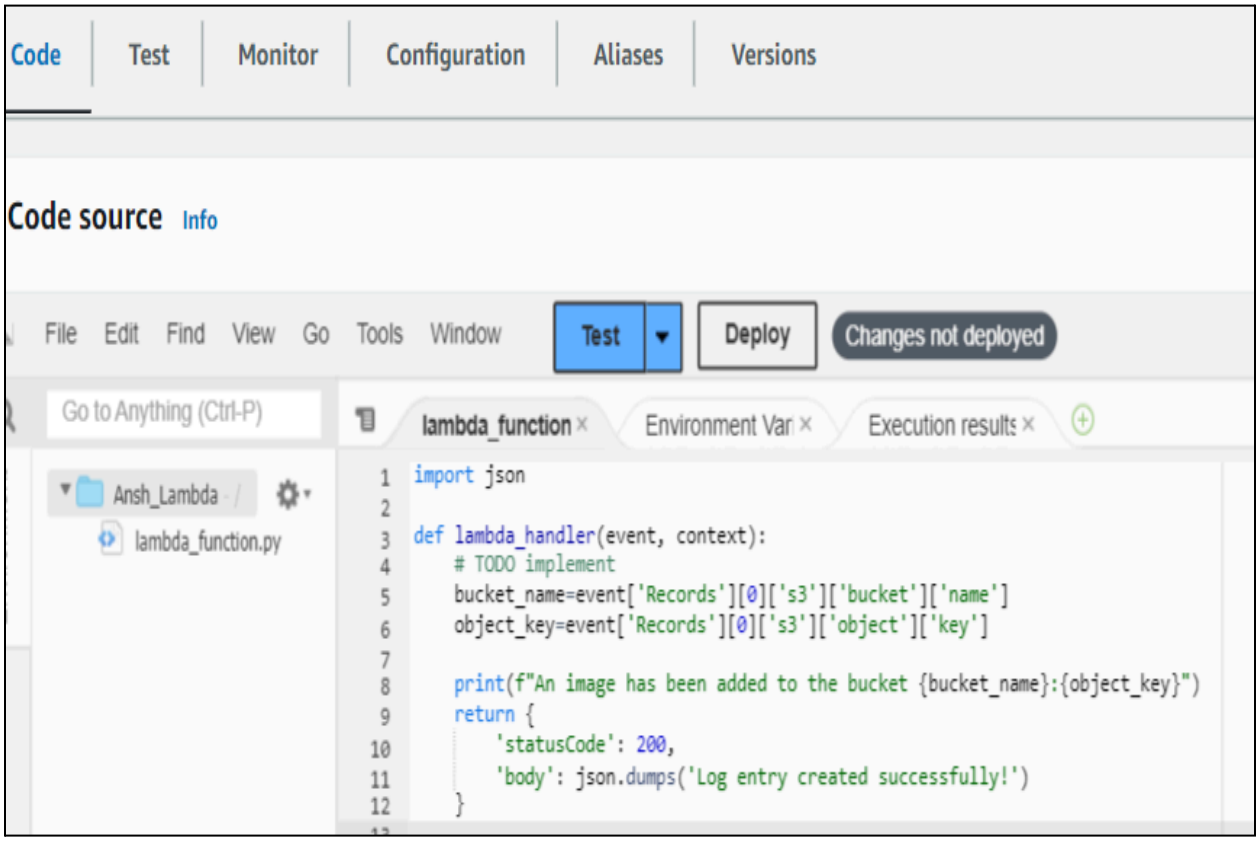
Step 6: Now In the Lambda function click on add tigger.



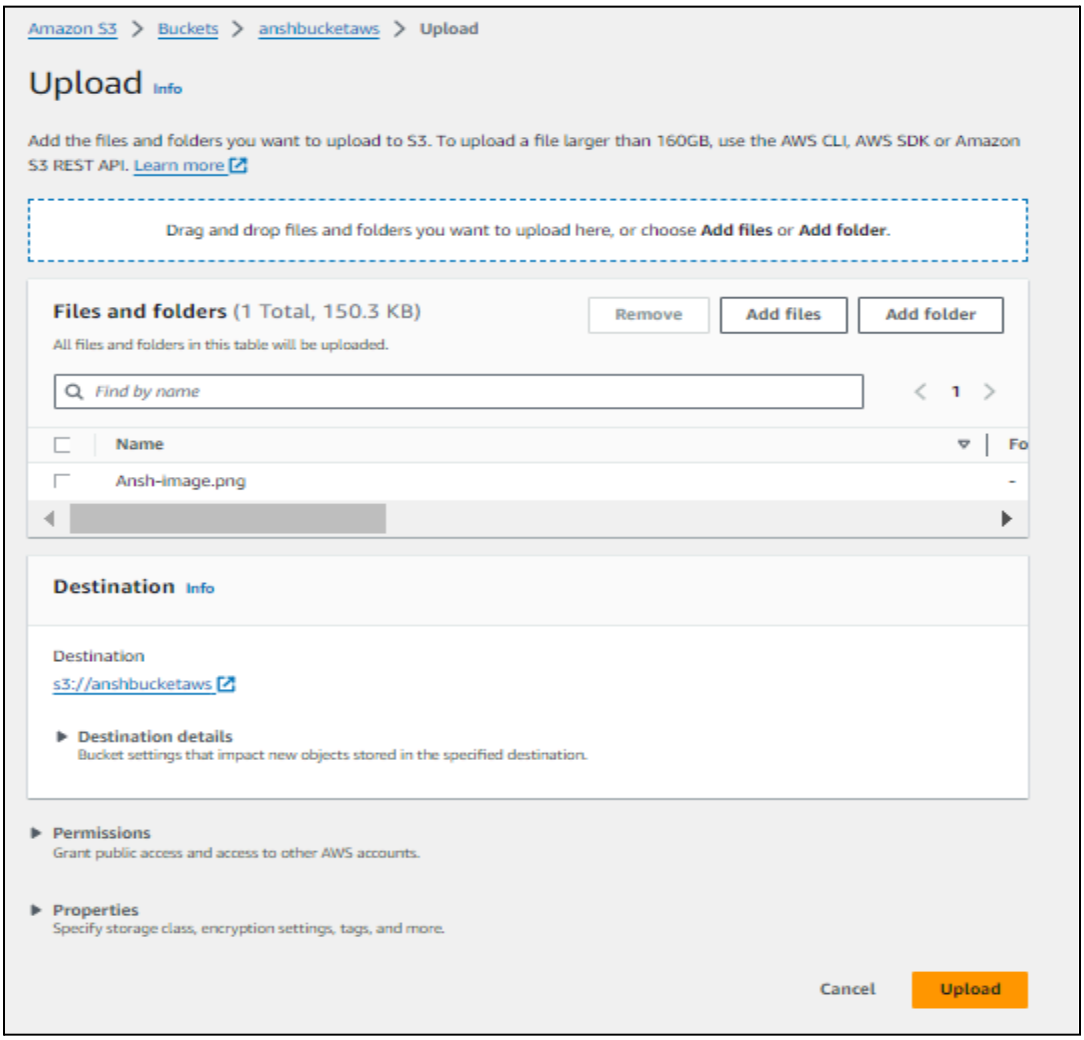
Now select the source as S3 then select the bucket name from the dropdown, keep other things to default and also you can add prefix to image.

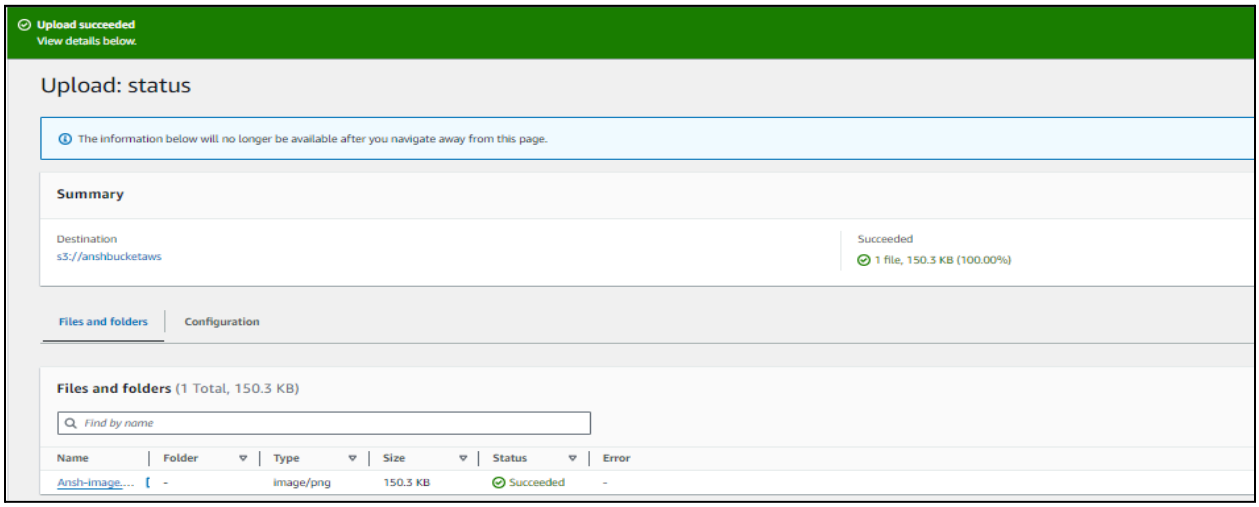


Step 7: Now Write code that logs a message like “An Image has been added” when triggered. Save the file and click on deploy



Step 8: Now upload any image to the bucket





Step 10: Now to click on test in lambda to check whether it is giving log when image is added to S3.

Step 11: Now Lets see the log on Cloud watch.To see it go to monitor section and then click on view cloudwatch logs.