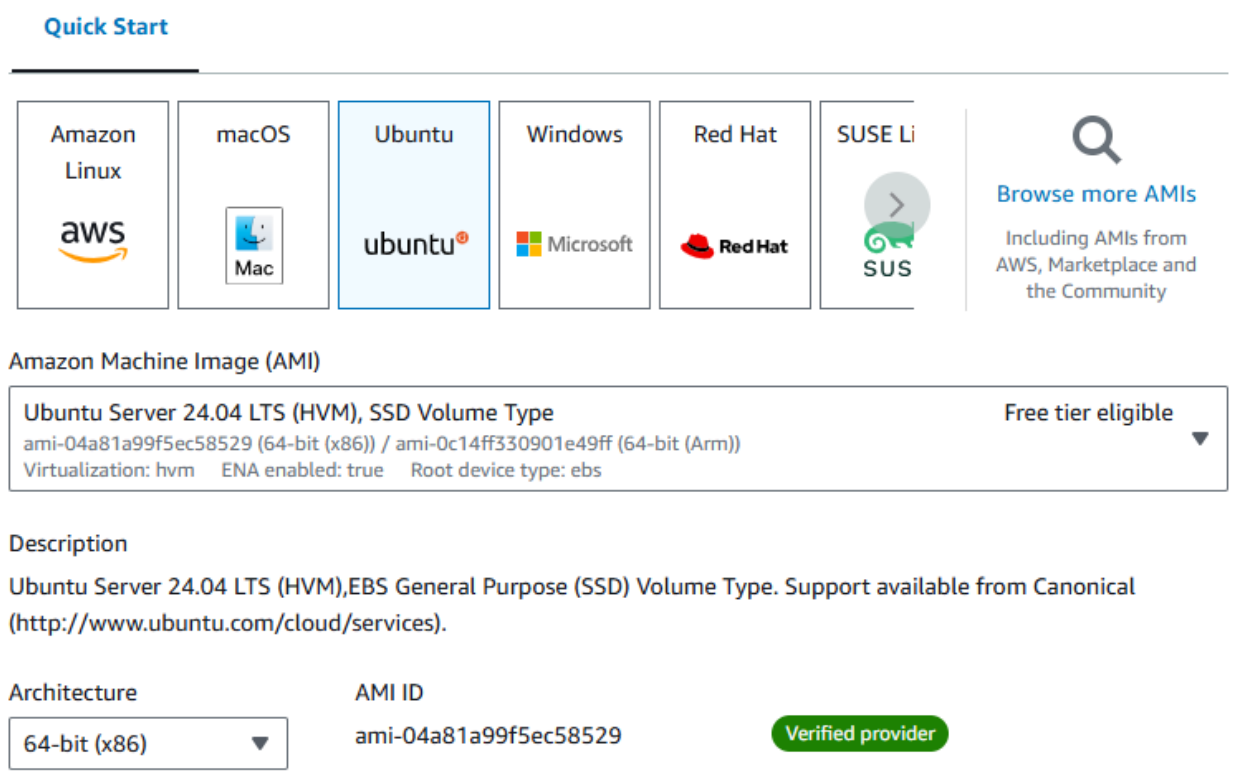
Siddhant Sathe **D15A/51**

Aim: Exp 1 To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Code and Output :



Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

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

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Network settings Info

Edit

Network Info

vpc-06c996635e2aef808

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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

(•) Create security group ( ) Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

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

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

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

Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

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

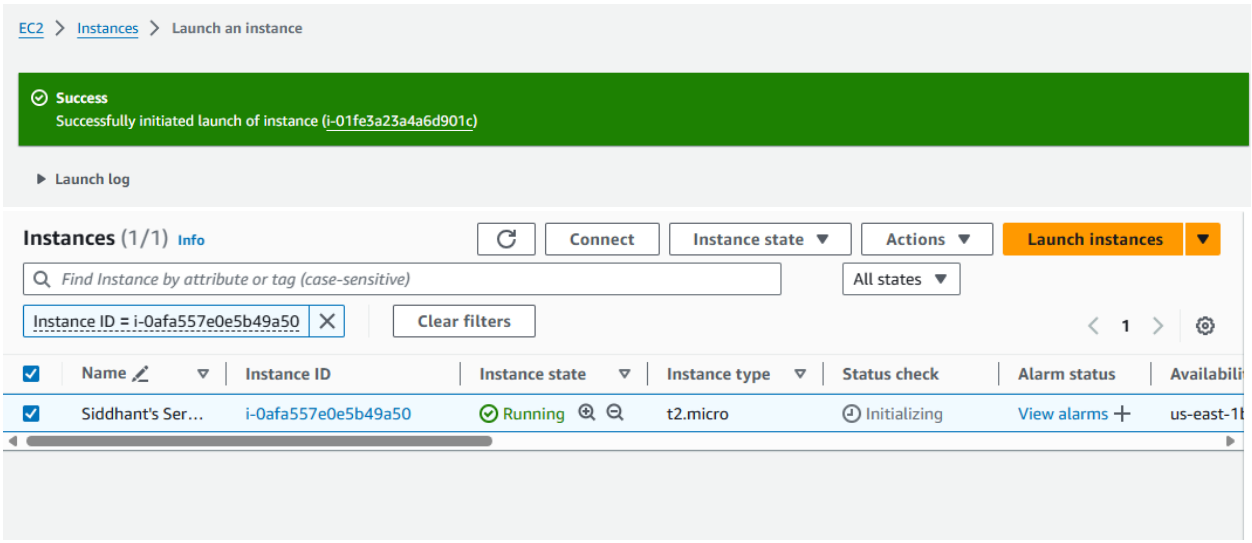
Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

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

0 x File systems Edit



Instance summary for i-0afa557e0e5b49a50 (Siddhant's Server) Info

Connect | Instance state | Actions

Updated less than a minute ago

Instance ID i-0afa557e0e5b49a50 (Siddhant's Server)	Public IPv4 address 34.200.231.88 | open address	Private IPv4 addresses 172.31.7.177
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-34-200-231-88.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-7-177.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-7-177.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 34.200.231.88 [Public IP]	VPC ID vpc-06c996635e2aef808	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more

aws | Services | Search [Alt+S] | N. Virginia | voclabs/user3387498=SATHE_SIDDHANT_SANJAY @ 1929-0520-1551

```
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


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


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

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

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

ubuntu@ip-172-31-7-177:~$
```

i-0afa557e0e5b49a50 (Siddhant's Server)

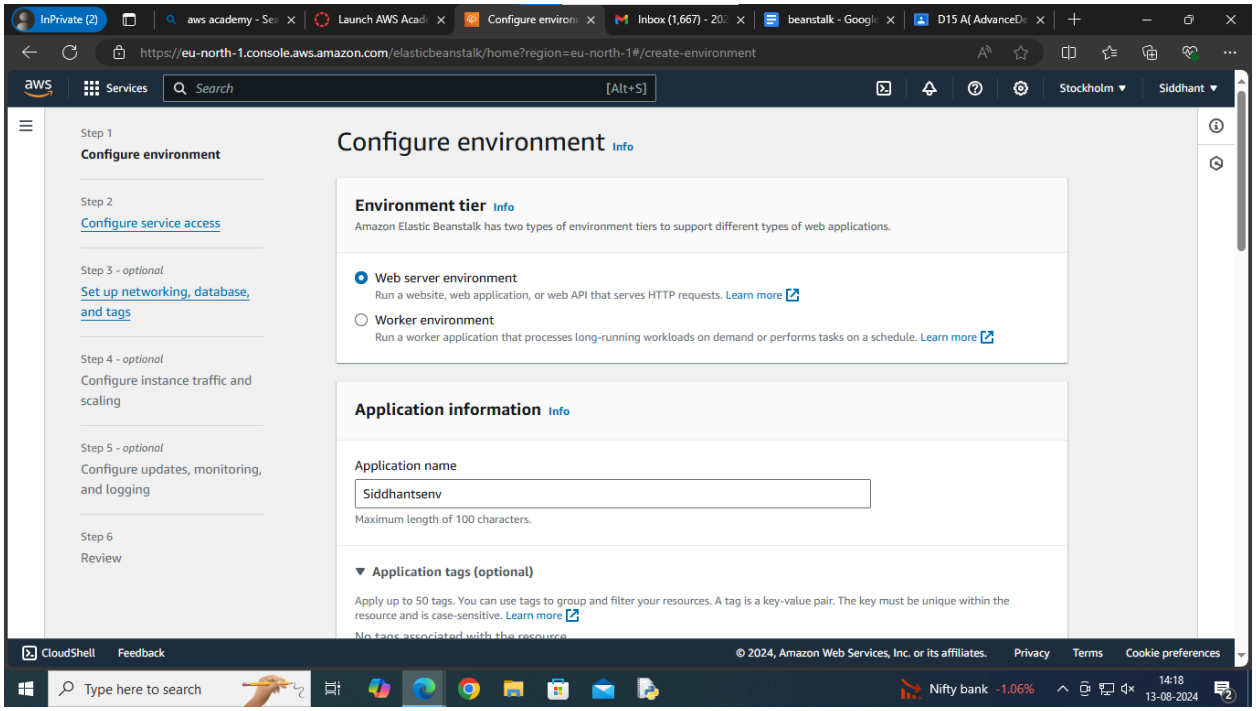
PublicIPs: 34.200.231.88 PrivateIPs: 172.31.7.177

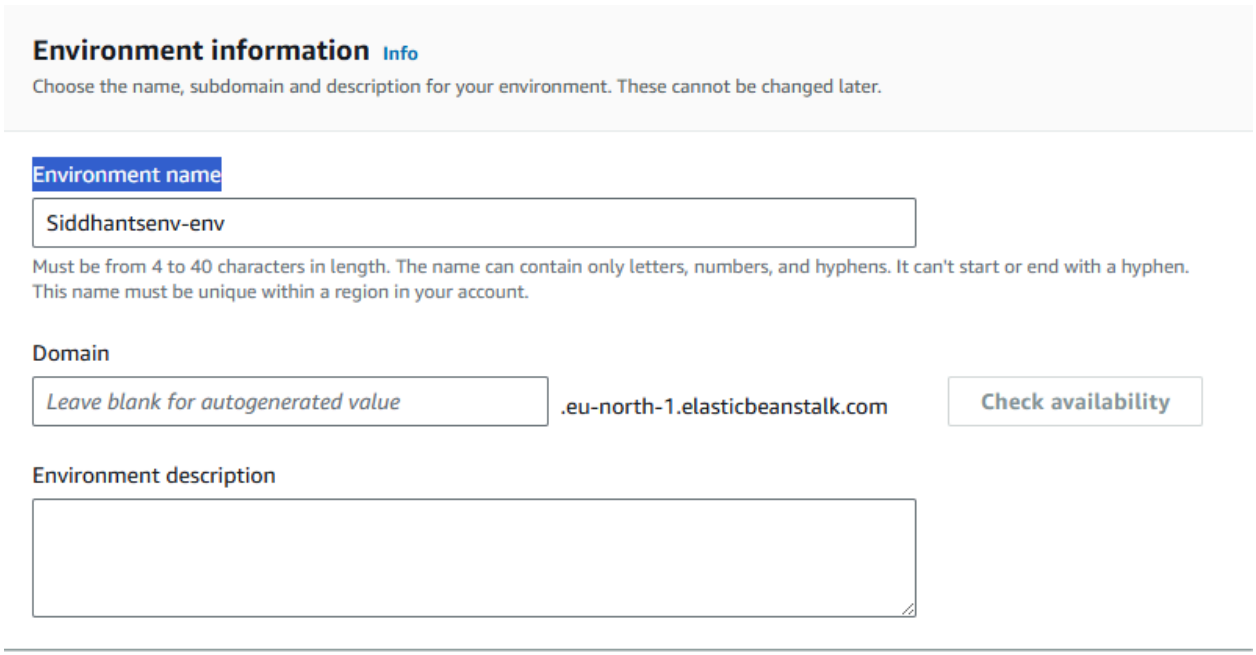
Siddhant's first Deployment

Siddhant Sathe **D15A/51**

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

Code and Output :
Using elastic beanstalk:





Platform Info

Platform type

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

Platform

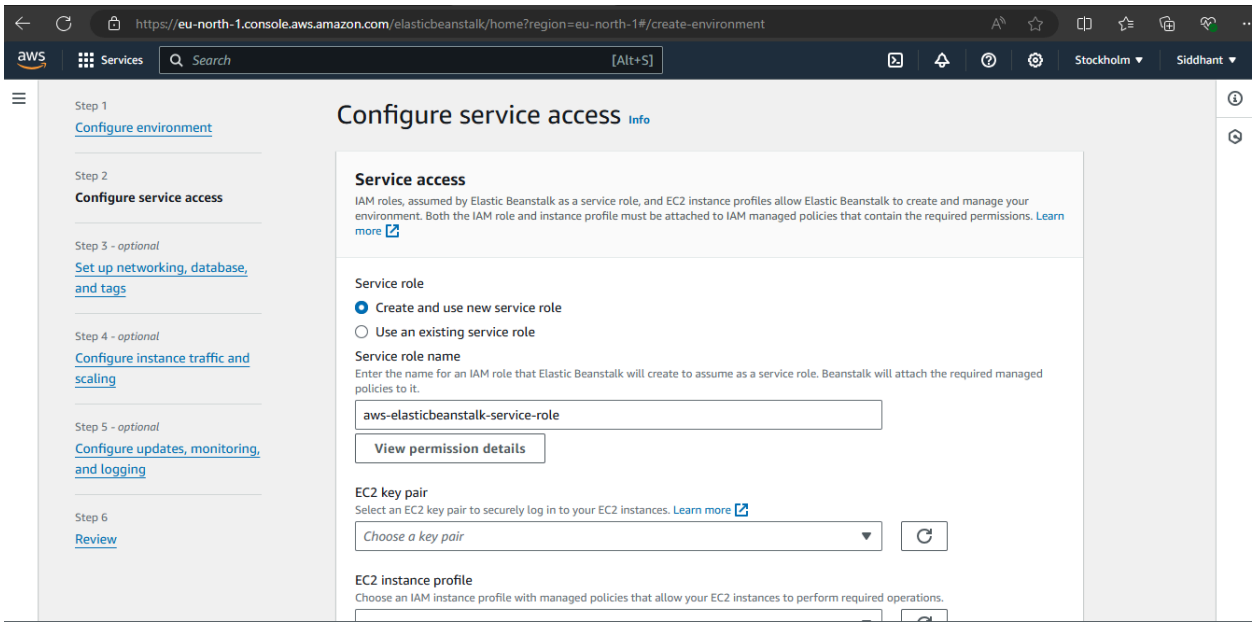
Python

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

Platform version

4.1.3 (Recommended)



Application code Info

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

Presets Info

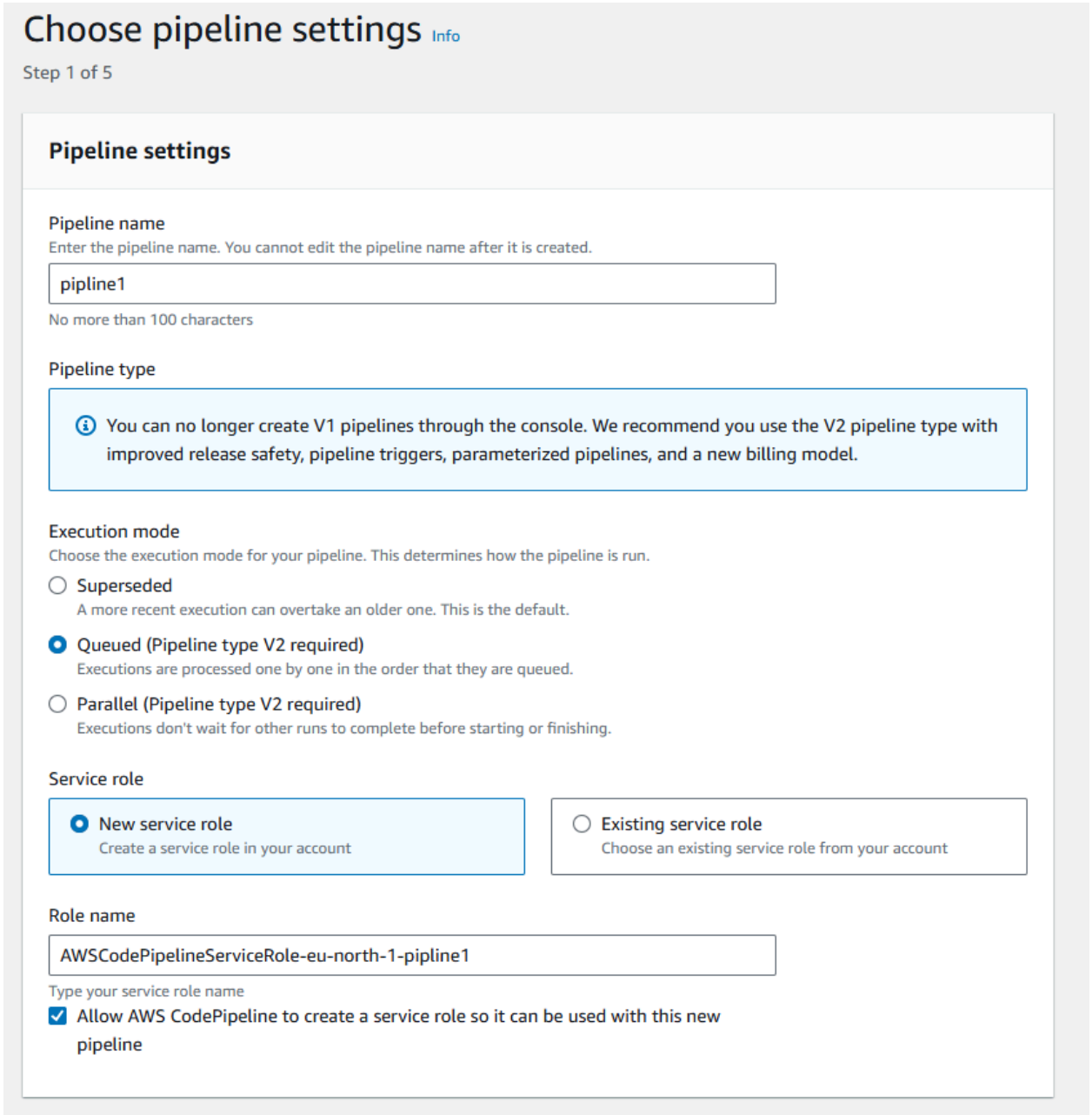
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

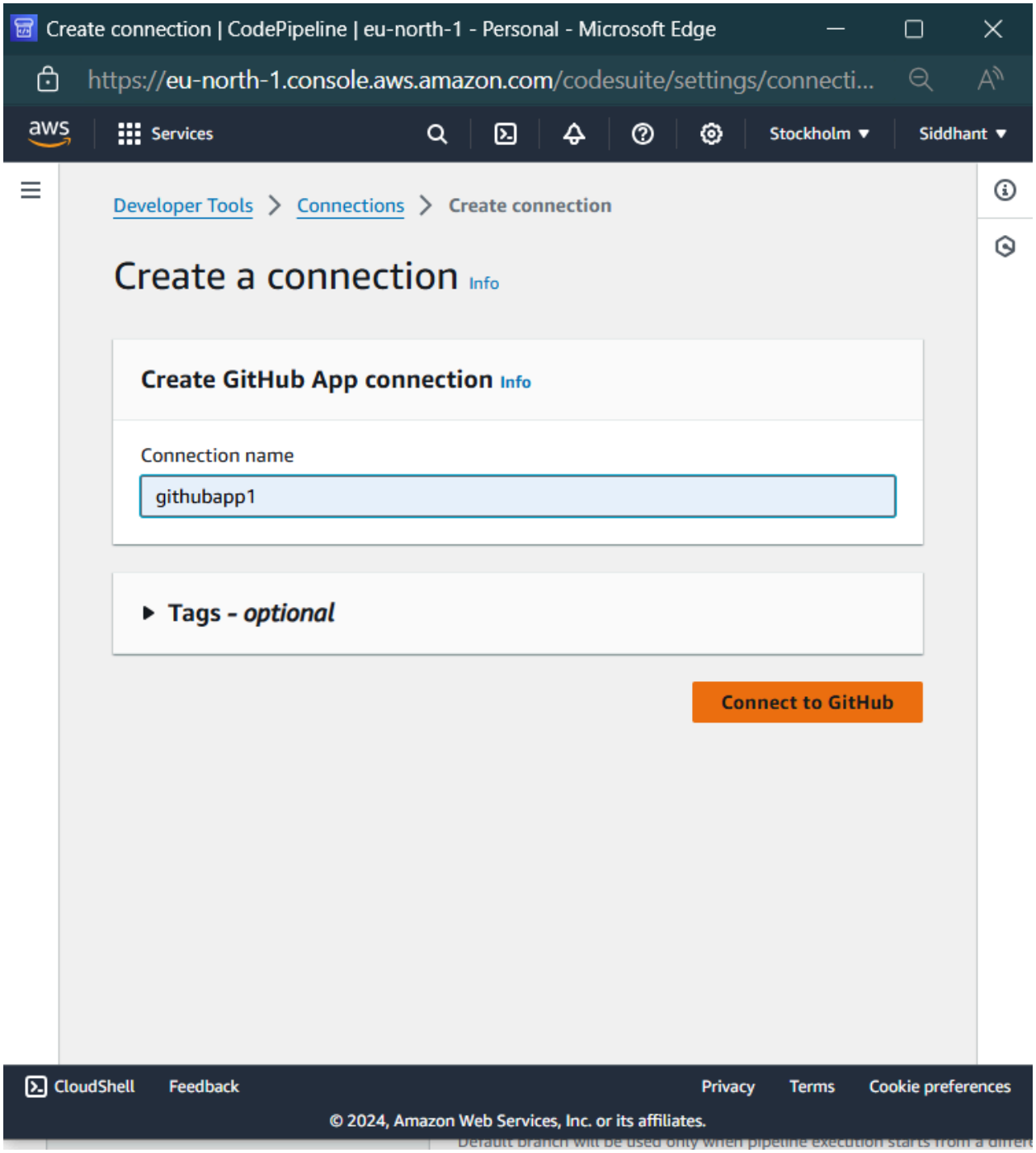
Configuration presets

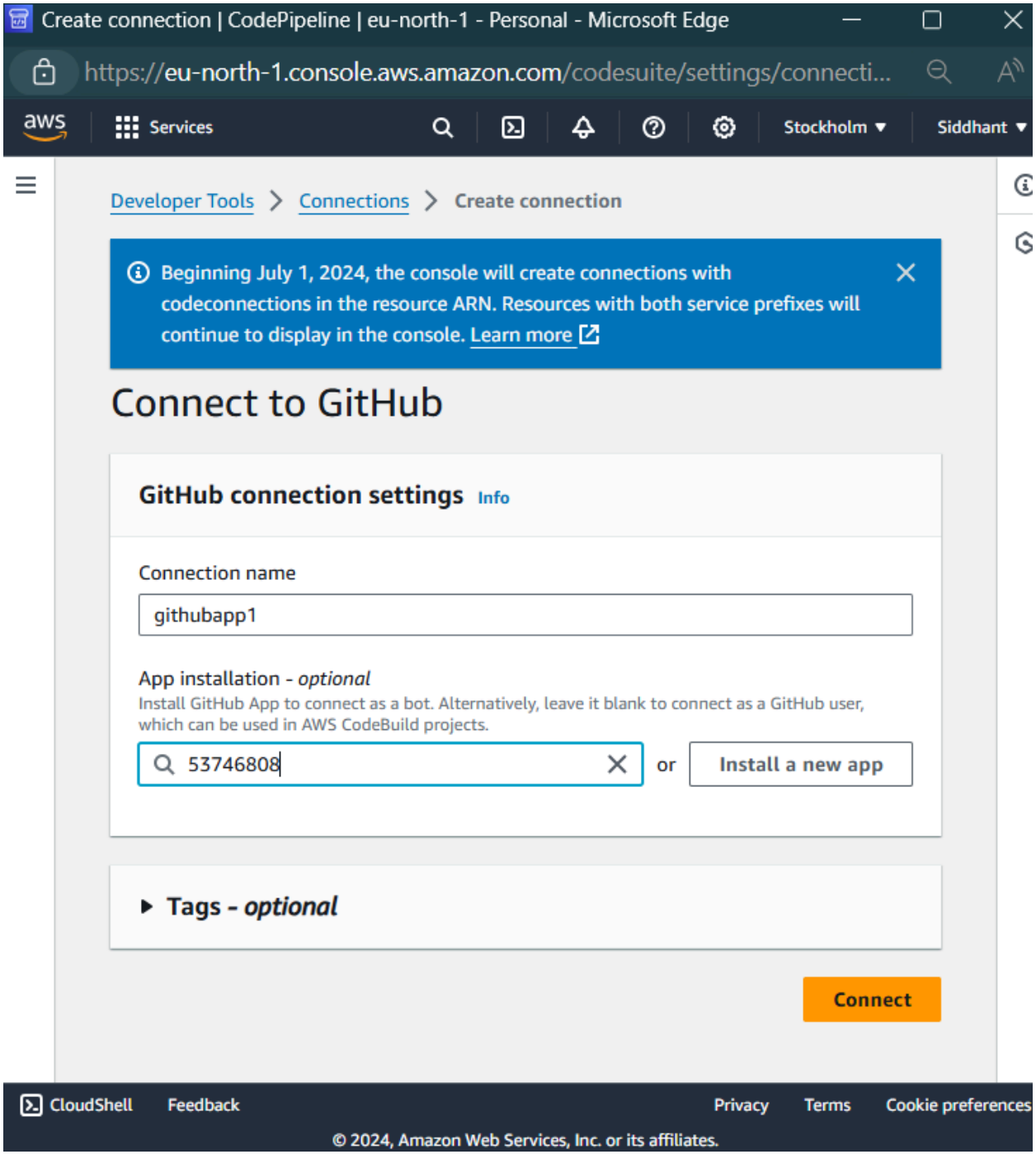
- ( ) Single instance (free tier eligible)
- ( ) Single instance (using spot instance)
- ( ) High availability
- ( ) High availability (using spot and on-demand instances)
- (•) Custom configuration

Cancel Next

Codepipeline




Create connection | CodePipeline | eu-north-1 - Personal - Microsoft Edge
https://eu-north-1.console.aws.amazon.com/codesuite/settings/connecti...
aws
Services
Stockholm
Siddhant
Developer Tools
Connections
Create connection
Create a connection
Info
Create GitHub App connection
Info
Connection name
githubapp1
Tags - optional
Connect to GitHub
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.


Create connection | CodePipeline | eu-north-1 - Personal - Microsoft Edge
https://eu-north-1.console.aws.amazon.com/codesuite/settings/connecti...
aws
Services
Stockholm
Siddhant
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
githubapp1
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.
53746808
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2)

New GitHub version 2 (app-based) action

To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:eu-north-1:010928222259:connection/0b5d0b9a-7 or Connect to GitHub

Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

SiddhantSathe/aws-codepipeline-s3-codedeploy-linux-2.0

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch

Default branch will be used only when pipeline execution starts from a different source or manually started.

master

Output artifact format

Choose the output artifact format.

- (selected) **CodePipeline default**
  AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
- ( ) **Full clone**
  AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Trigger

Trigger type
Choose the trigger type that starts your pipeline.

- (●) No filter
  Starts your pipeline on any push and clones the HEAD.
- ( ) Specify filter
  Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
- ( ) Do not detect changes
  Don't automatically trigger the pipeline.

ⓘ You can add additional sources and triggers by editing the pipeline after it is created.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk ▼

Region

Europe (Stockholm) ▼

Input artifacts
Choose an input artifact for this action. Learn more

SourceArtifact ▼

No more than 100 characters

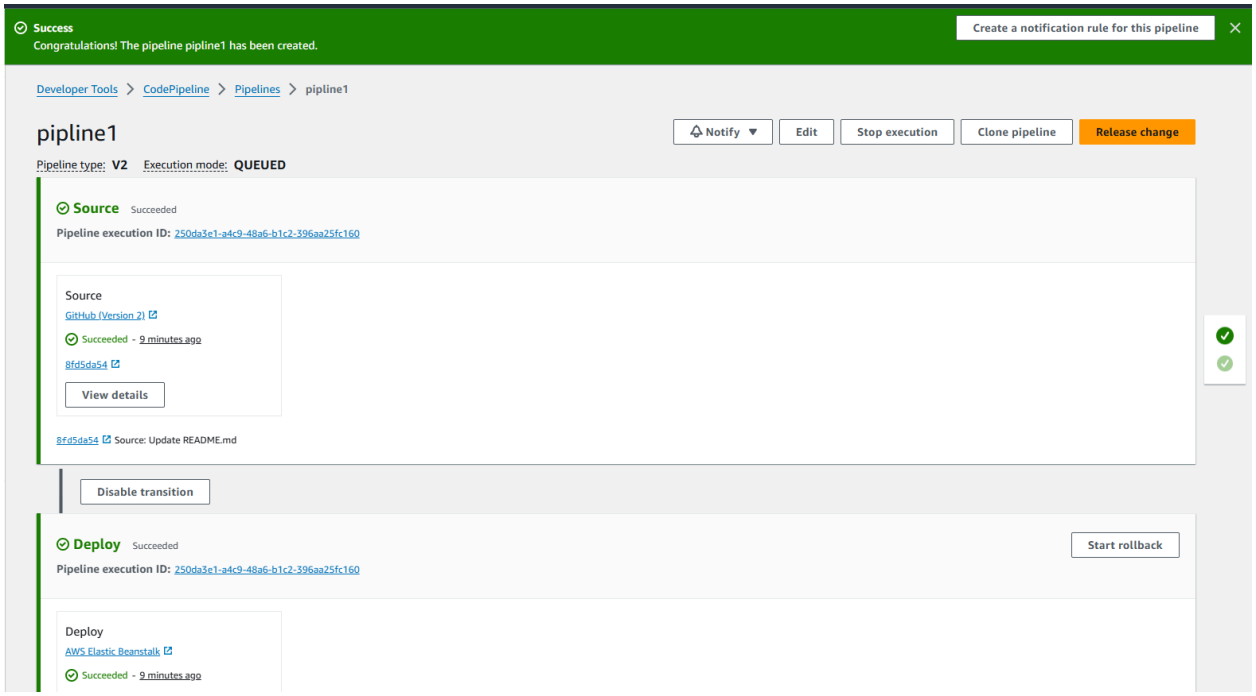
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

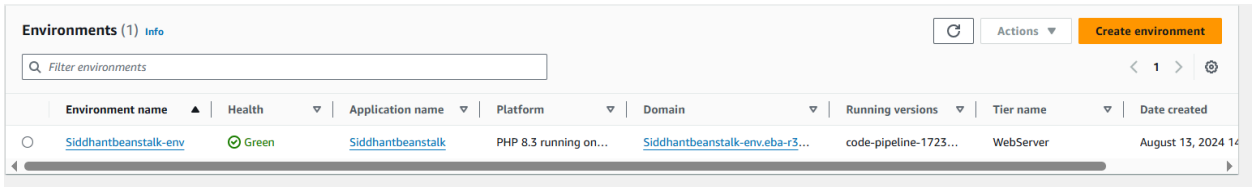
Siddhantbeanstalk ×

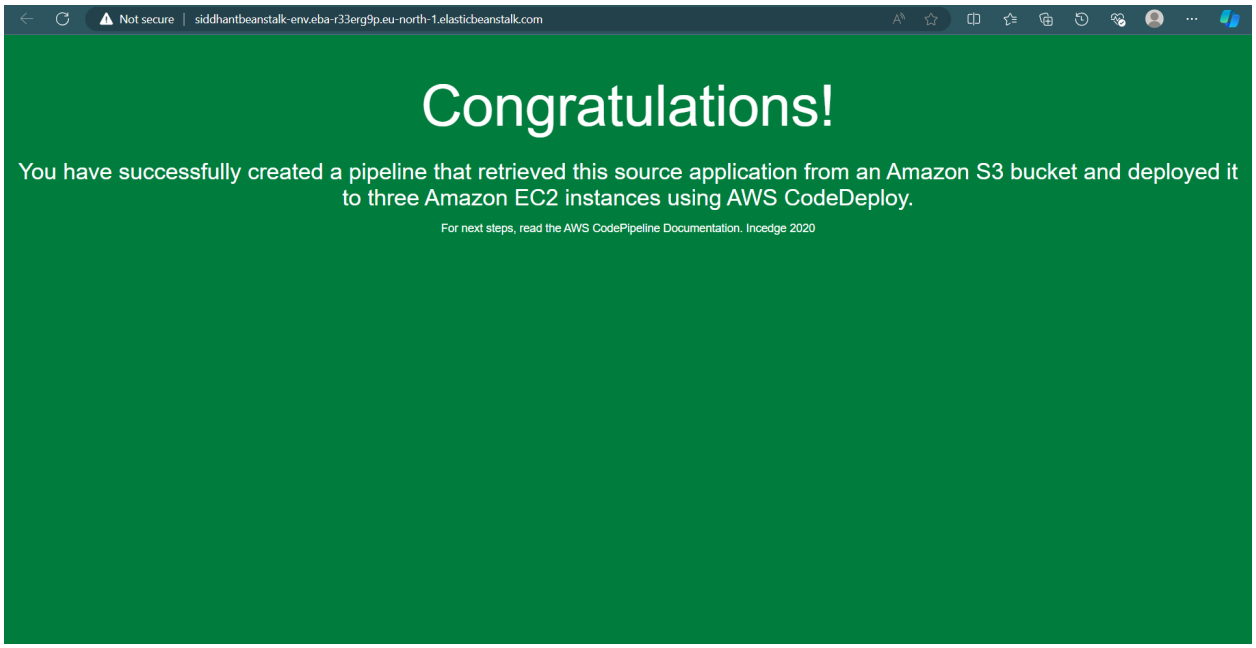
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

Siddhantbeanstalk-env ×

Siddhantbeanstalk-env

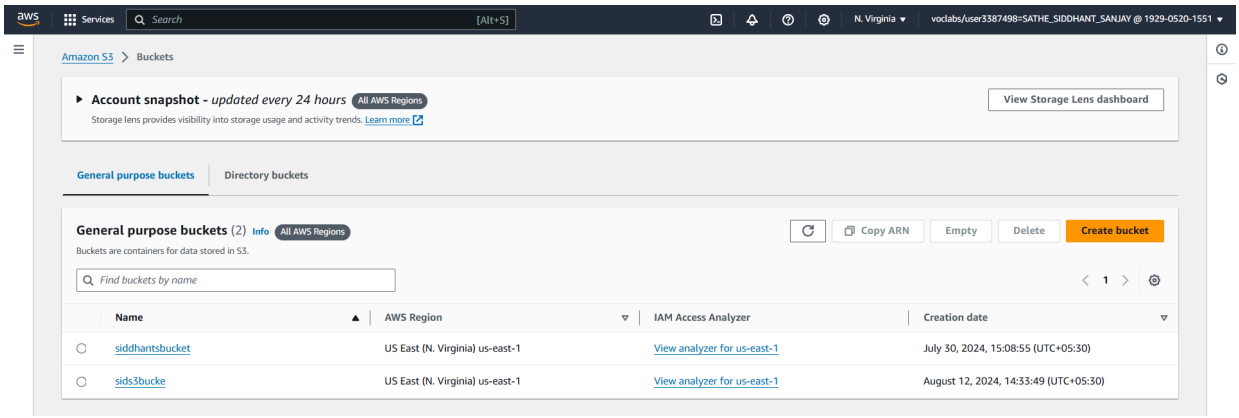

Success
Congratulations! The pipeline pipline1 has been created.
Create a notification rule for this pipeline
Developer Tools > CodePipeline > Pipelines > pipline1
pipline1
Notify
Edit
Stop execution
Clone pipeline
Release change
Pipeline type: V2 Execution mode: QUEUED
Source Succeeded
Pipeline execution ID: 250da3e1-a4c9-48a6-b1c2-396aa25fc160
Source
GitHub (Version 2)
Succeeded - 9 minutes ago
8fd5da54
View details
8fd5da54 Source: Update README.md
Disable transition
Deploy Succeeded
Start rollback
Pipeline execution ID: 250da3e1-a4c9-48a6-b1c2-396aa25fc160
Deploy
AWS Elastic Beanstalk
Succeeded - 9 minutes ago


Environments (1) Info
Actions
Create environment
Filter environments
Environment name
Health
Application name
Platform
Domain
Running versions
Tier name
Date created
Siddhantbeanstalk-env
Green
Siddhantbeanstalk
PHP 8.3 running on...
Siddhantbeanstalk-env.eba-r3...
code-pipeline-1723...
WebServer
August 13, 2024

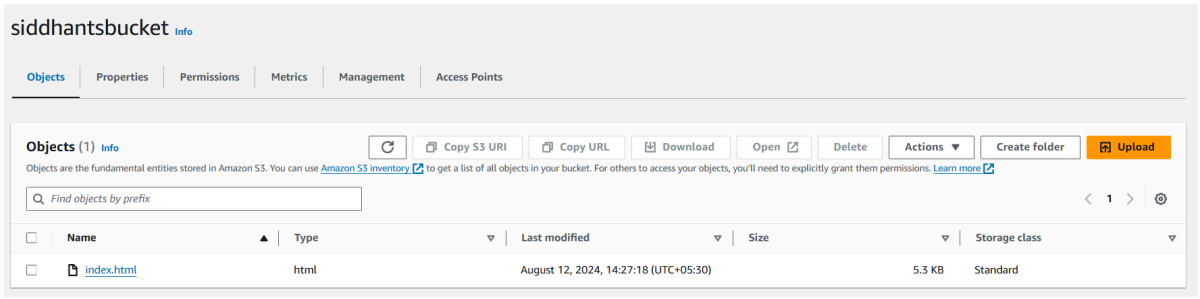

Not secure | siddhantbeanstalk-env.eba-r33erg9p.eu-north-1.elasticbeanstalk.com
Congratulations!
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.
For next steps, read the AWS CodePipeline Documentation. Incedge 2020

Using S3 bucket:

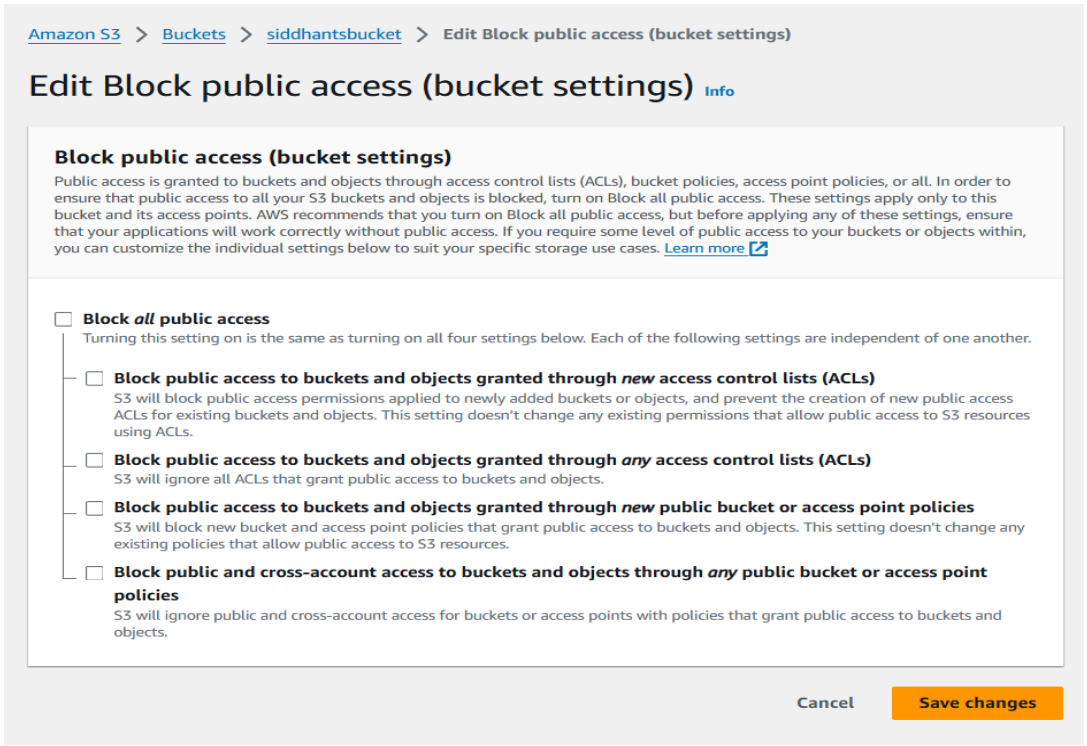
Create S3 bucket



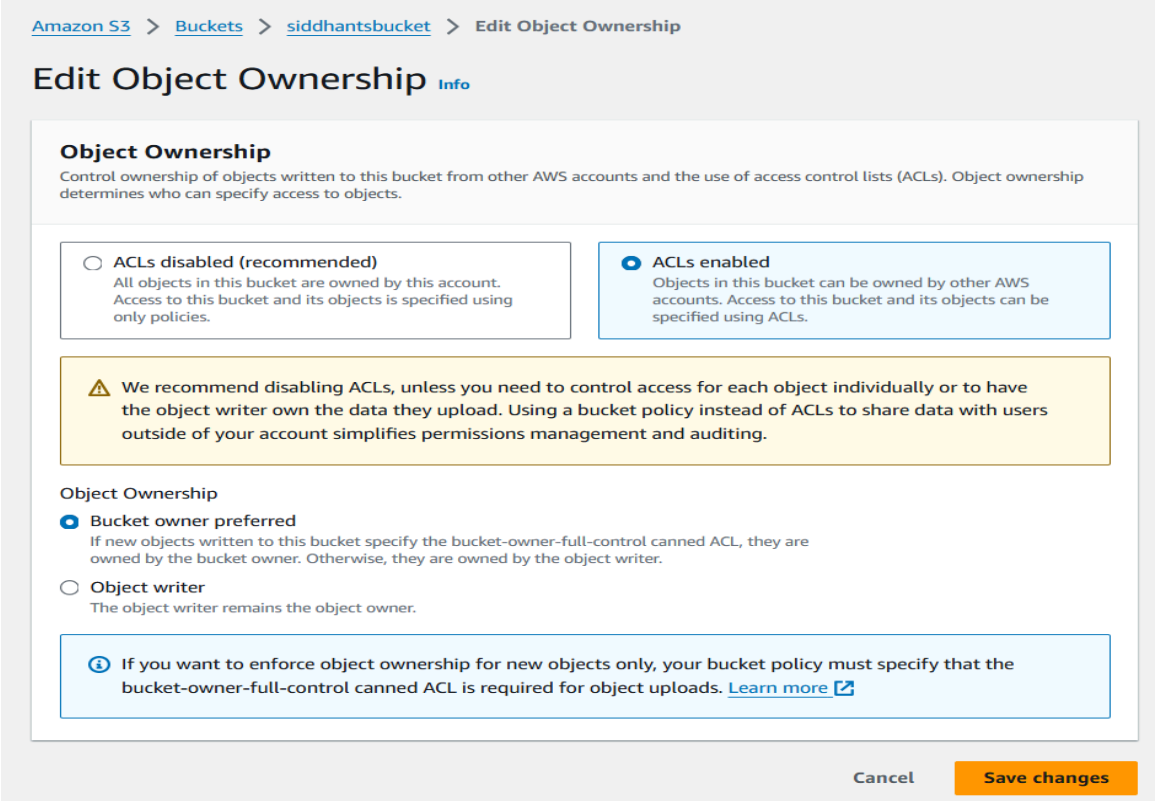
Upload File



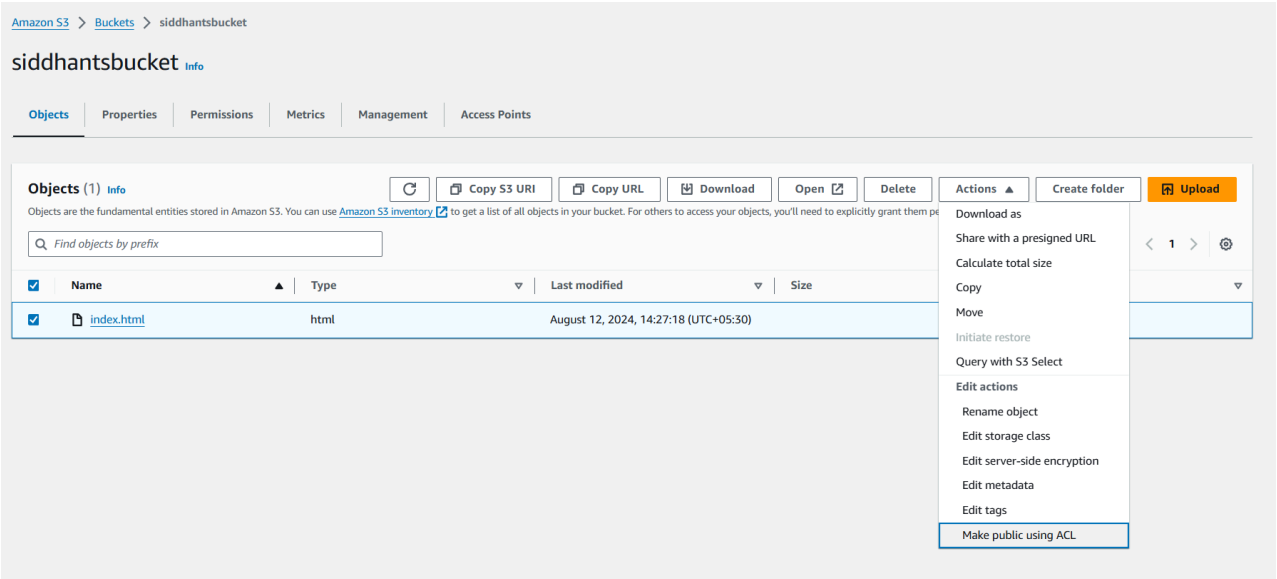
Edit block public access

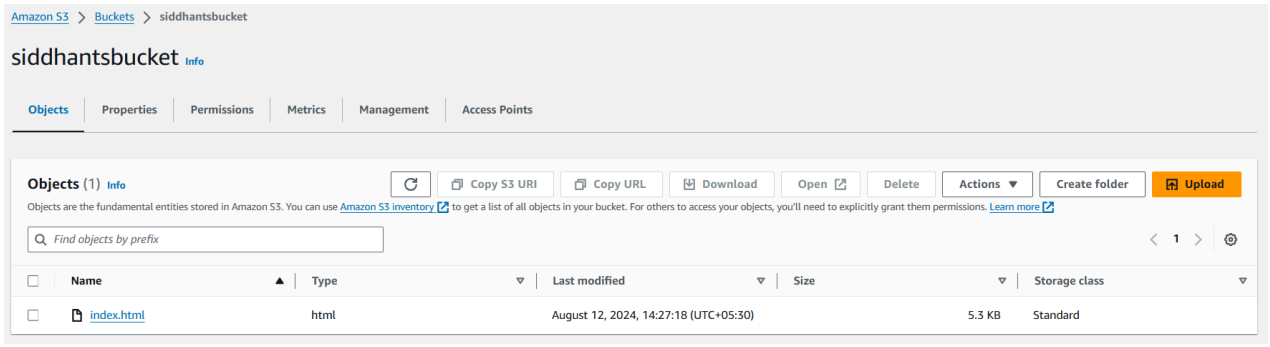


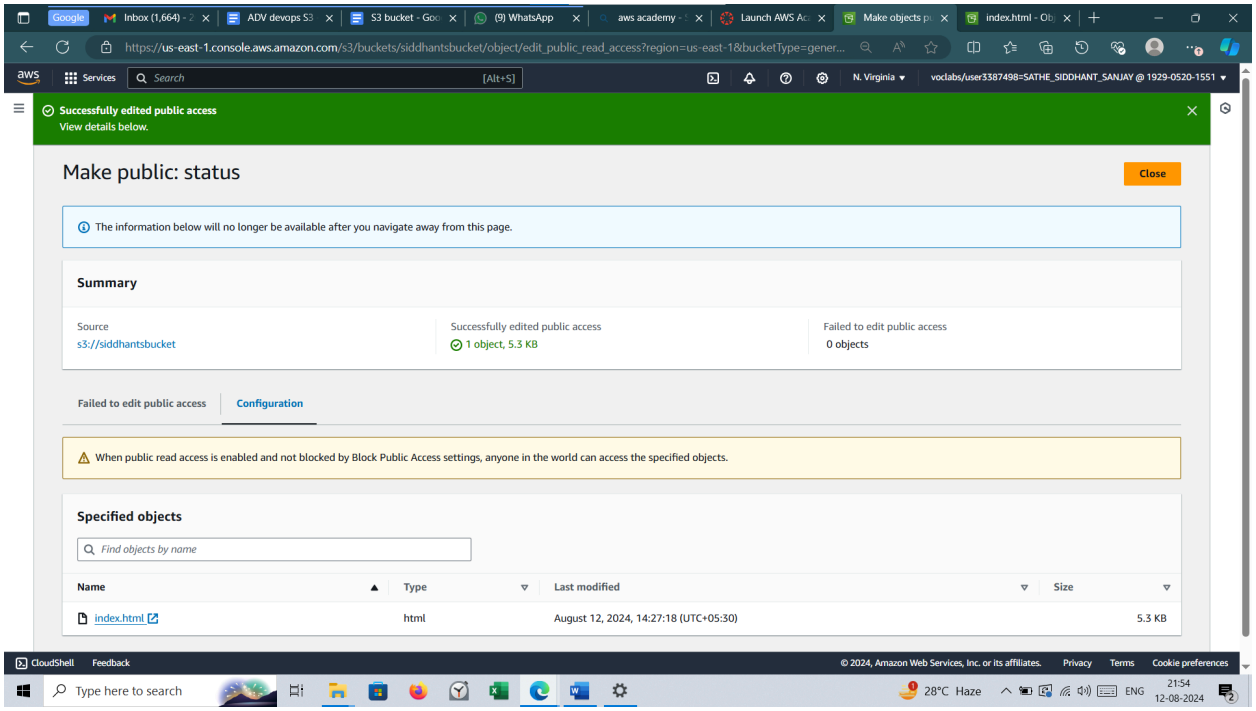
Edit object ownership

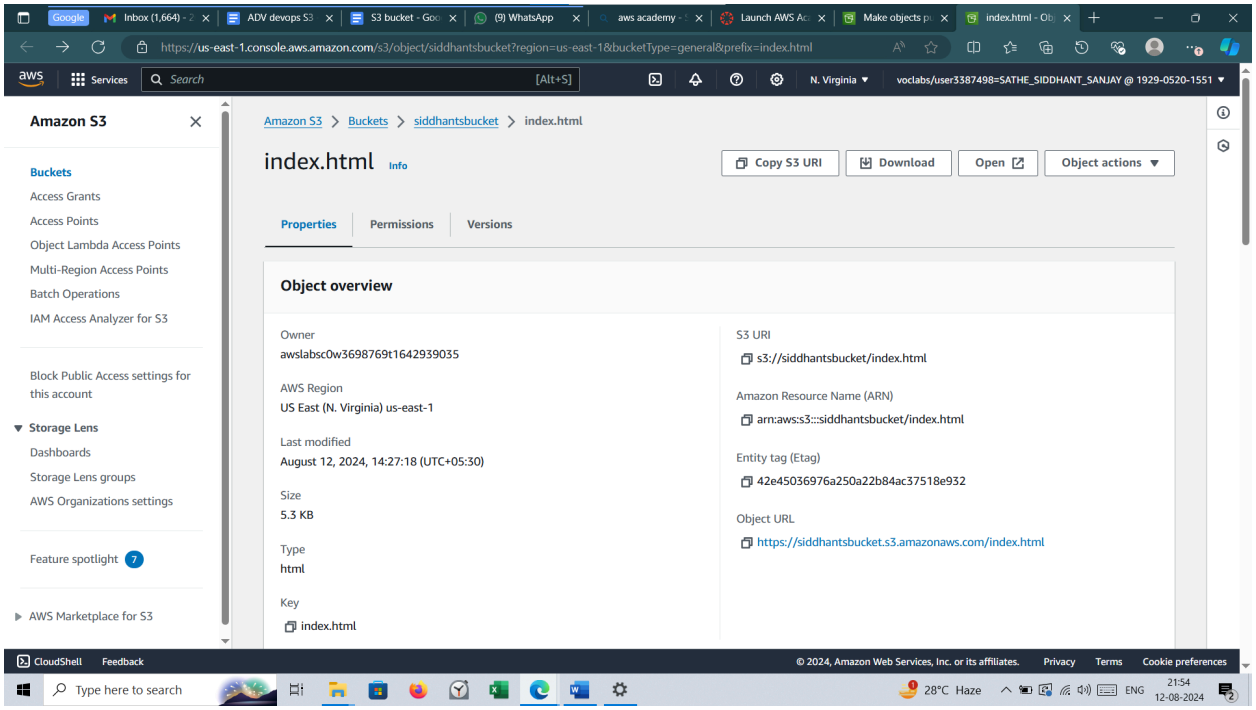


Make file public using ACL

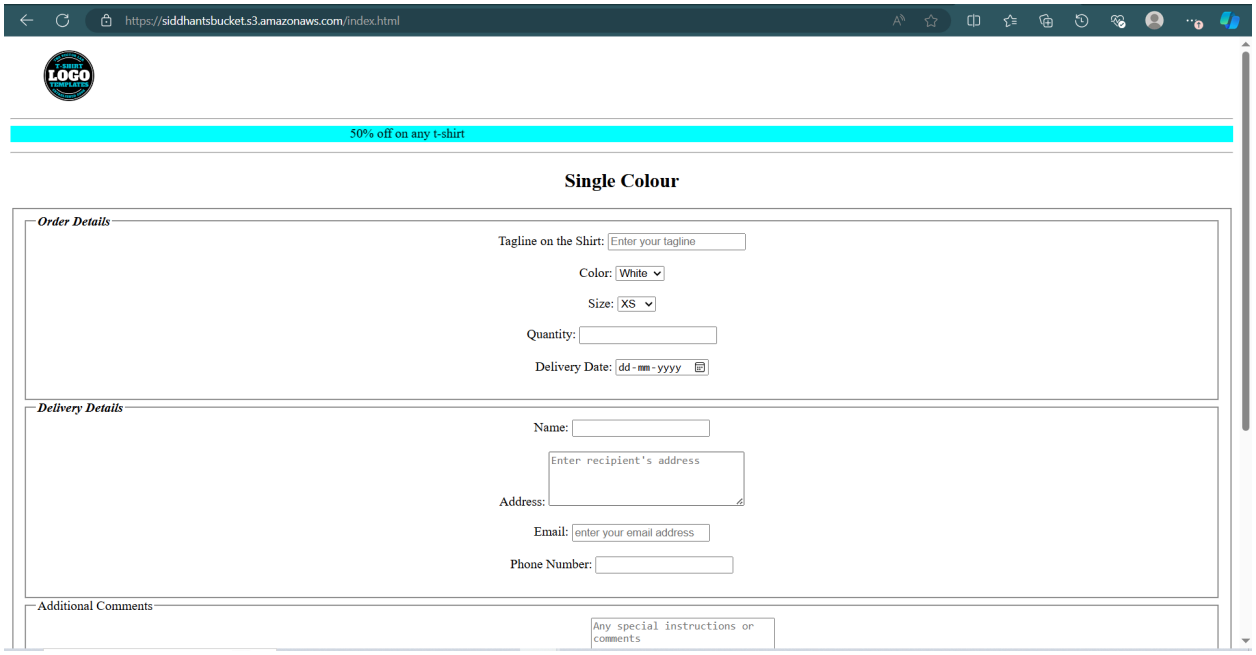



Amazon S3 > Buckets > siddhantsbucket
siddhantsbucket Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
index.html
html
August 12, 2024, 14:27:18 (UTC+05:30)
5.3 KB
Standard


https://us-east-1.console.aws.amazon.com/s3/buckets/siddhantsbucket/object/edit_public_read_access?region=us-east-1&bucketType=gener...
Services
Search
[Alt+S]
N. Virginia
Successfully edited public access
View details below.
Make public: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Source
s3://siddhantsbucket
Successfully edited public access
1 object, 5.3 KB
Failed to edit public access
0 objects
Failed to edit public access
Configuration
When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.
Specified objects
Find objects by name
Name
Type
Last modified
Size
index.html
html
August 12, 2024, 14:27:18 (UTC+05:30)
5.3 KB
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
28°C Haze
ENG
21:54
12-08-2024



Hosted website



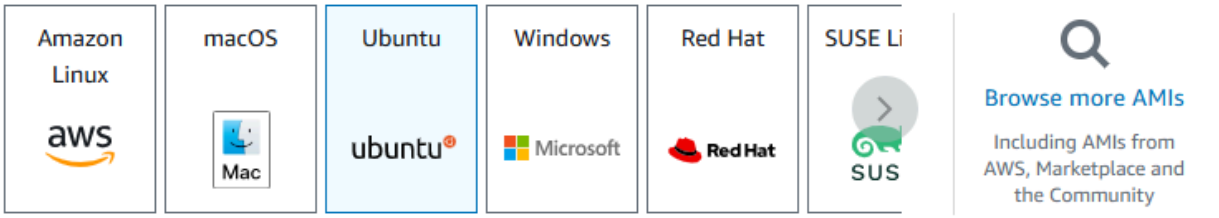
Using EC2:

Siddhant Sathe **D15A/51**

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

Code and Output :

Quick Start



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Architecture

64-bit (x86)

AMI ID

ami-04a81a99f5ec58529 Verified provider

Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

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

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Network settings Info

Edit

Network Info

vpc-06c996635e2aef808

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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

(•) Create security group ( ) Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

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

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

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

Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

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

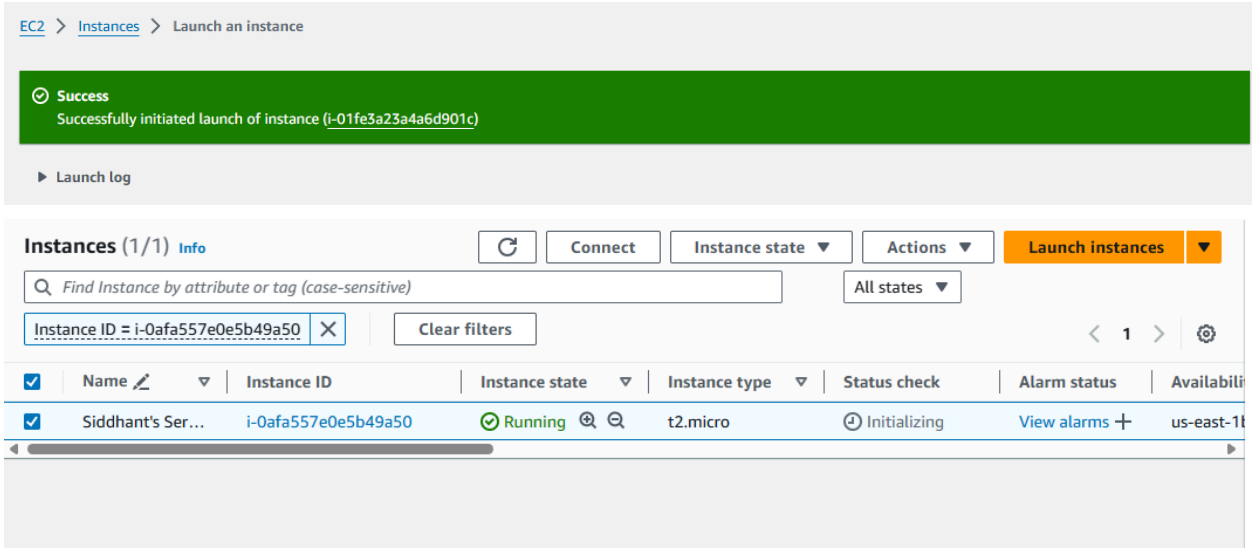
Add new volume

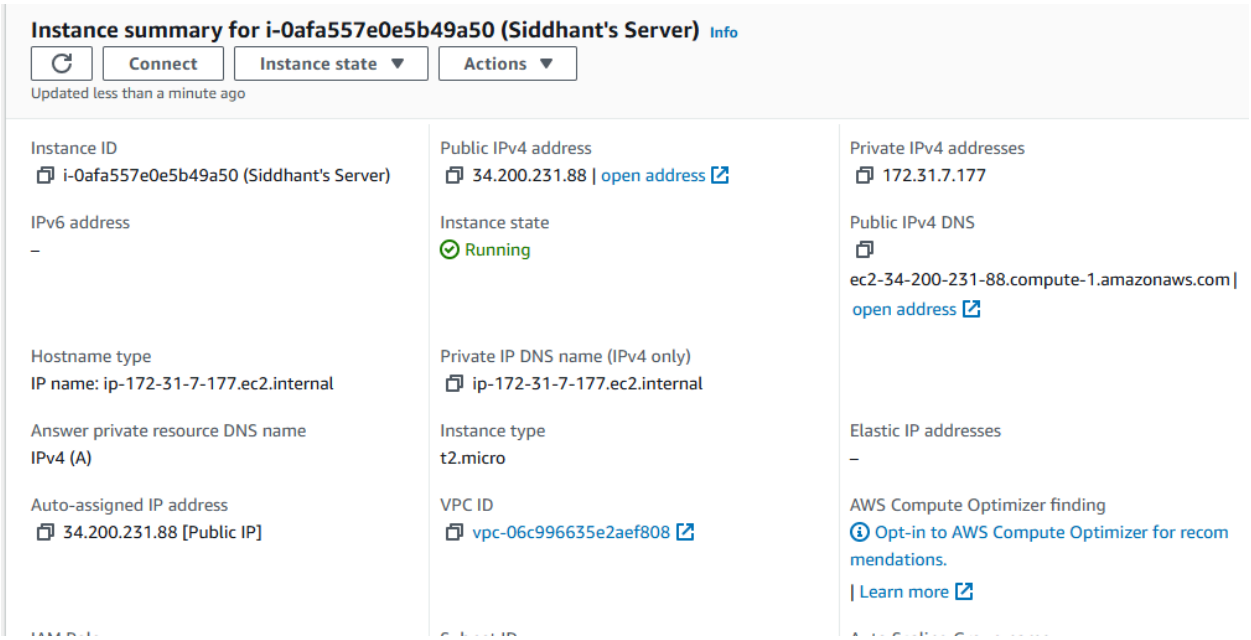
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

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

0 x File systems Edit

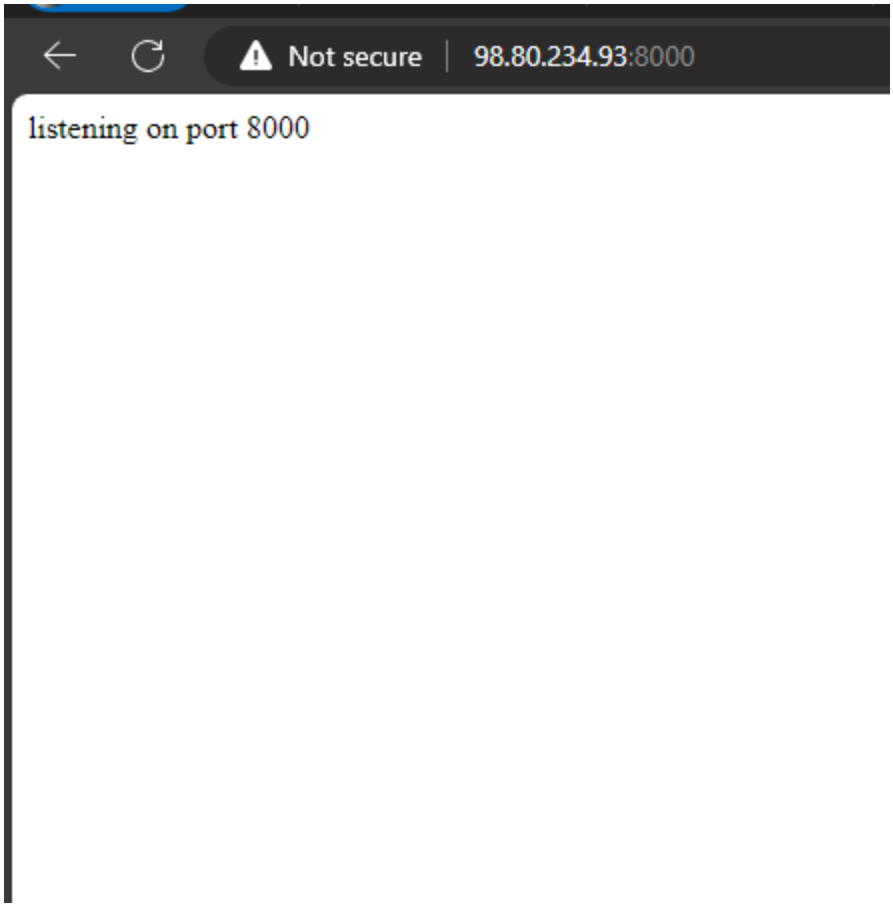


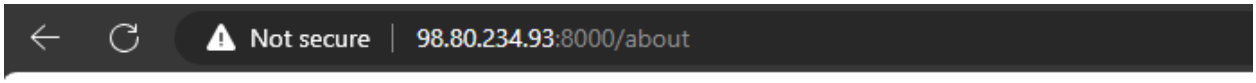



```
root@ip-172-31-90-124:/home/ubuntu/siddhant# git clone https://github.com/SiddhantSathe/nodejs.git
Cloning into 'nodejs'...
remote: Enumerating objects: 647, done.
remote: Counting objects: 100% (647/647), done.
remote: Compressing objects: 100% (488/488), done.
remote: Total 647 (delta 125), reused 635 (delta 121), pack-reused 0
Receiving objects: 100% (647/647), 713.52 KiB | 12.09 MiB/s, done.
Resolving deltas: 100% (125/125), done.
root@ip-172-31-90-124:/home/ubuntu/siddhant#
```

```
root@ip-172-31-90-124:/home/ubuntu/siddhant/nodejs# npm i

added 36 packages, and audited 102 packages in 5s

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

2 moderate severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.
```

```
root@ip-172-31-90-124:/home/ubuntu/siddhant/nodejs# npm start

> nodejs@1.0.0 start
> node index.js

listening on port 8000
```





about page

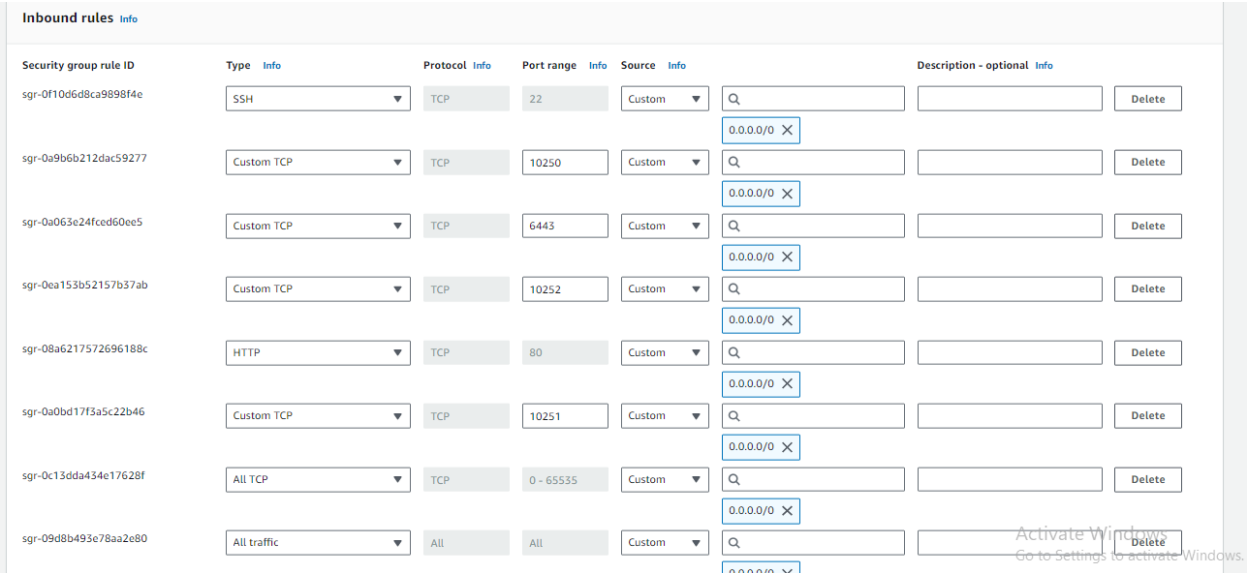
Siddhant Sathe **D15A/51**

ADVANCE DEVOPS EXP-3

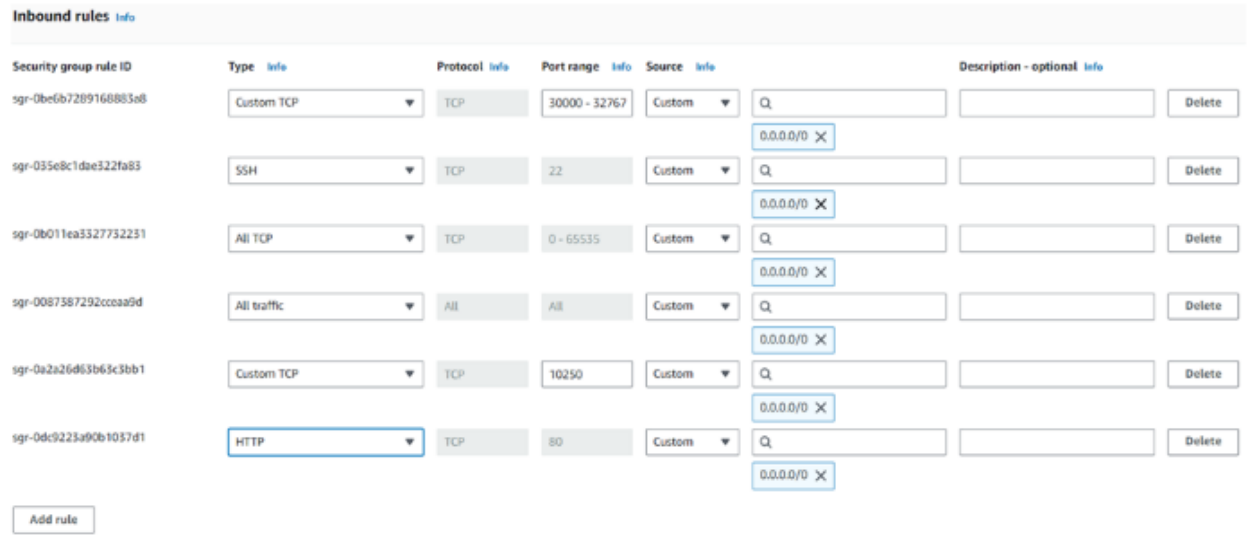
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

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

Master:



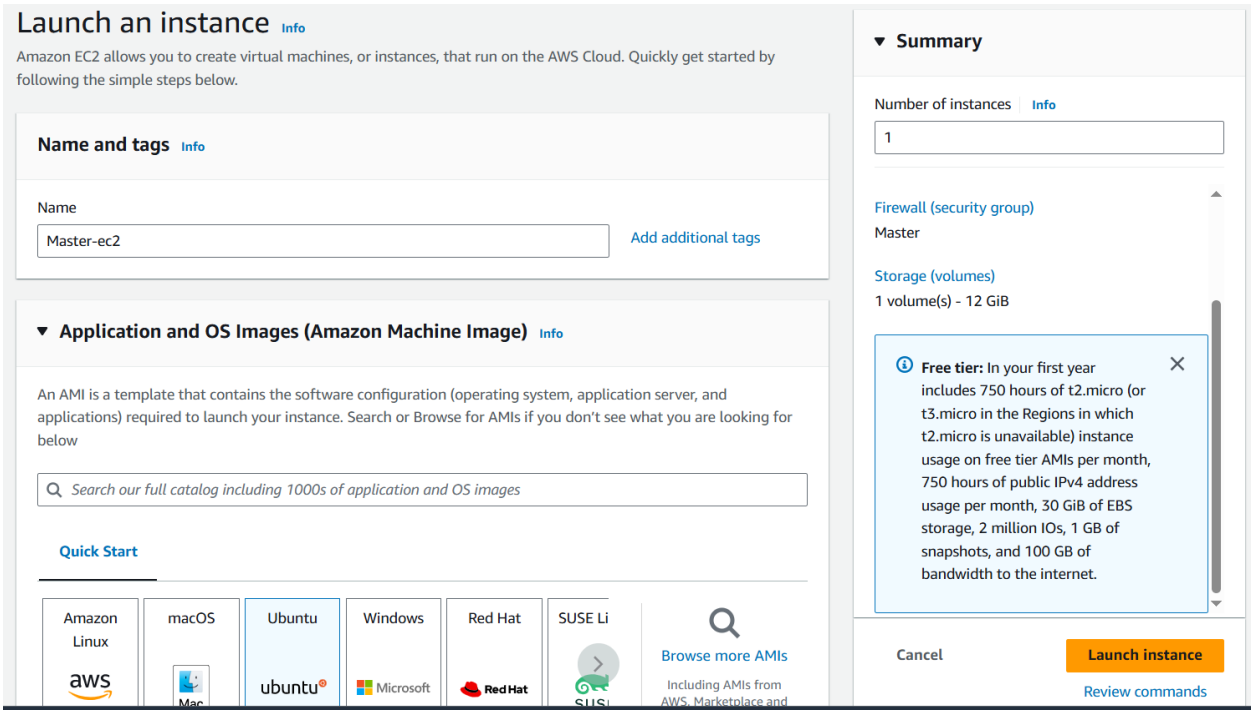
Node:

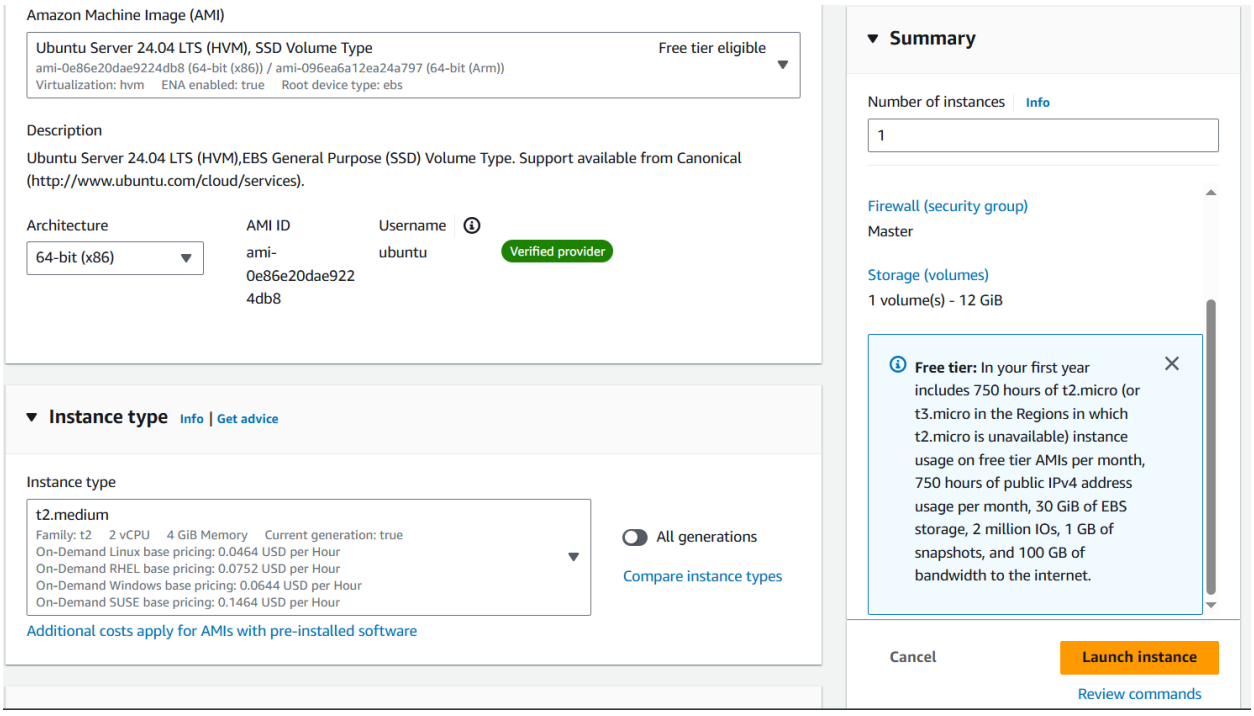


Step 2: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances(1 for Master and 2 for Node).Select Ubuntu as AMI and t2.medium as Instance Type and create a key

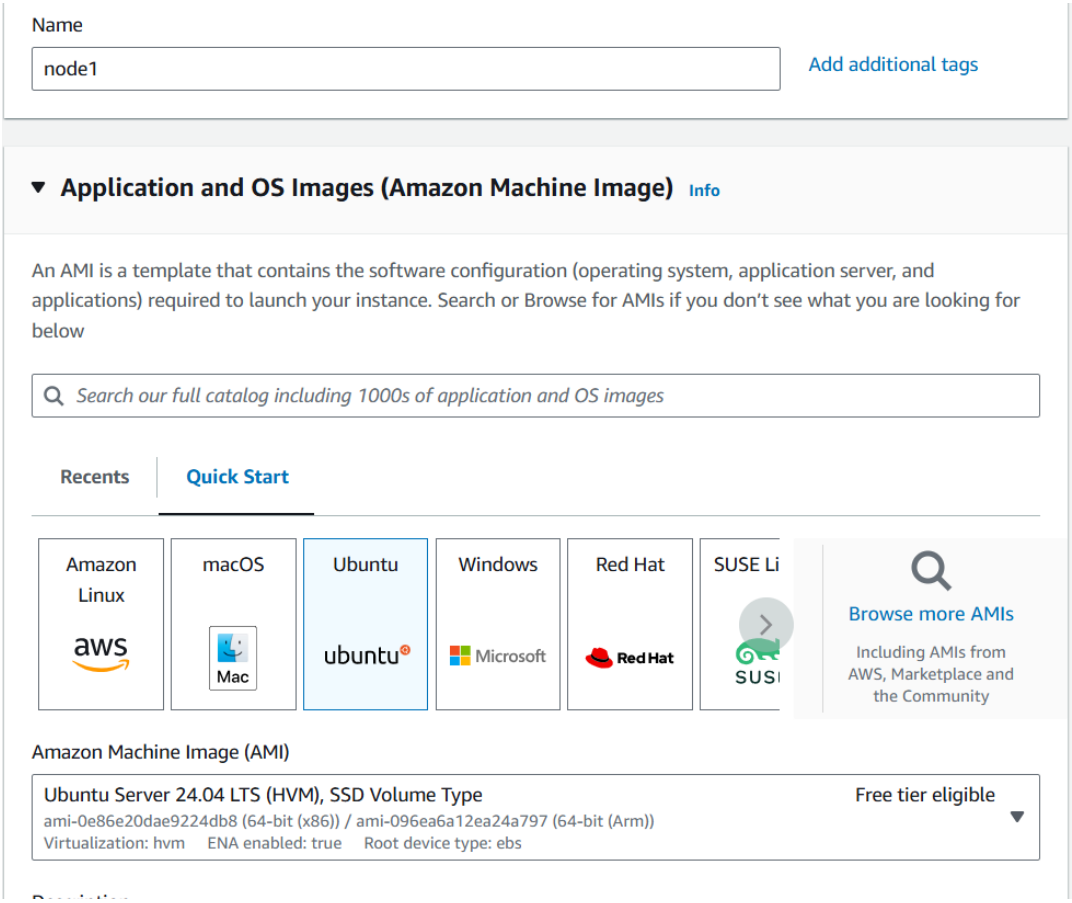
of type RSA with .pem extension and move the downloaded key to the new folder.We can use 2 Different keys, 1 for Master and 1 for Node. Also Select Security Groups from the existing.

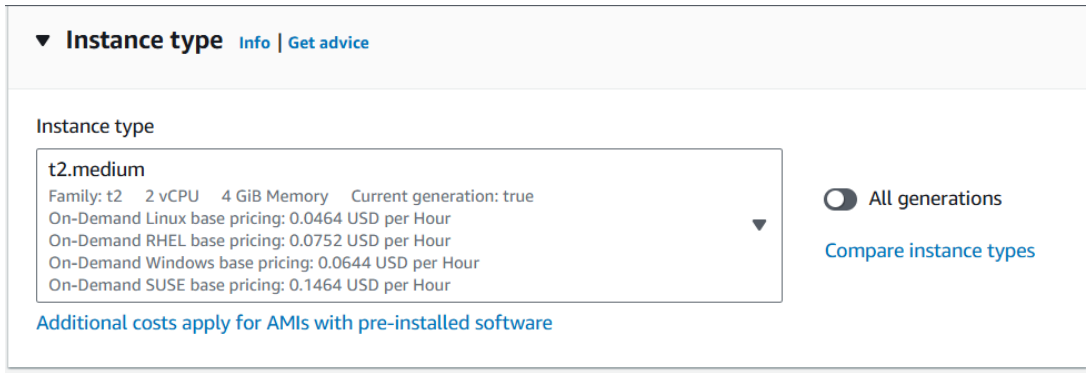
Master:

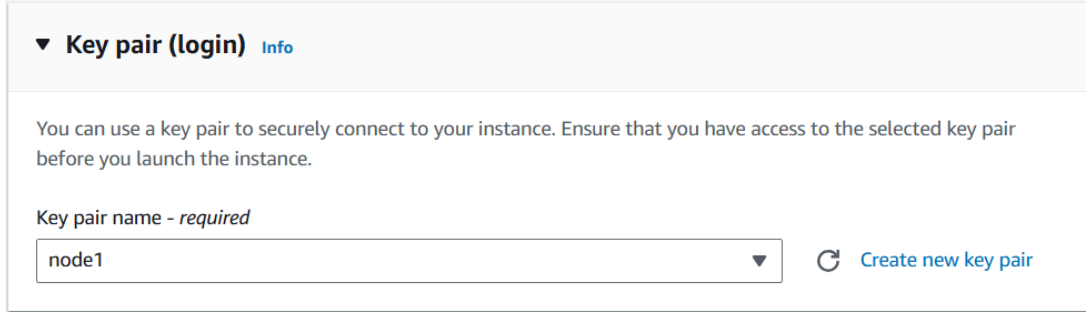


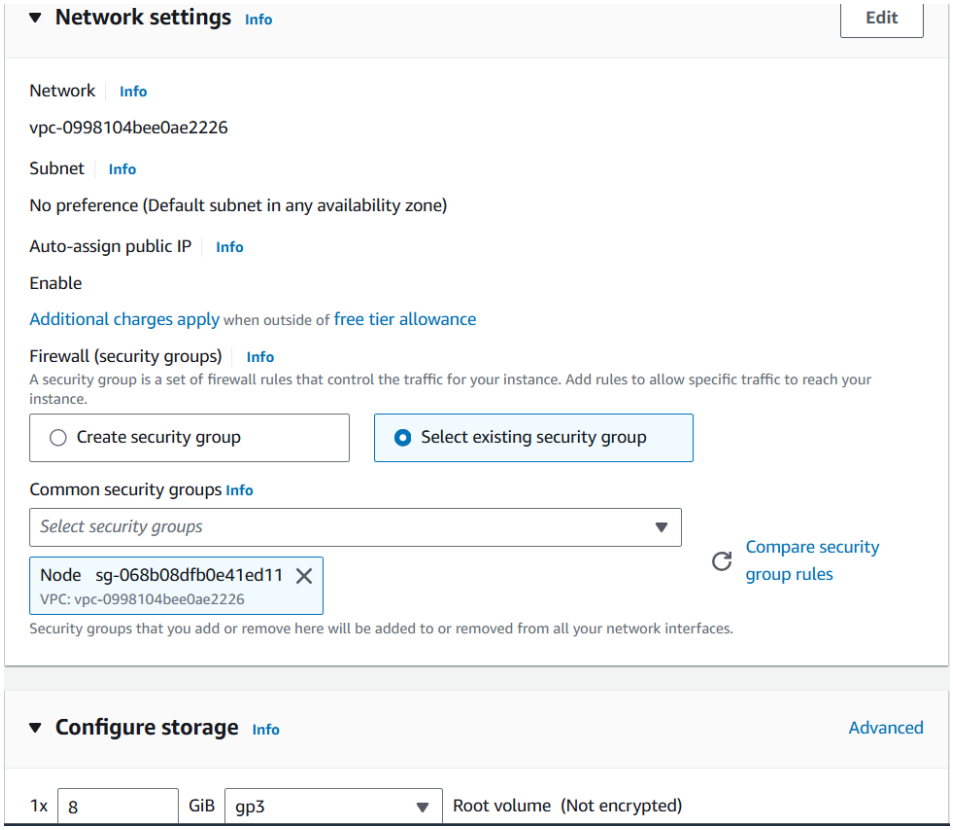


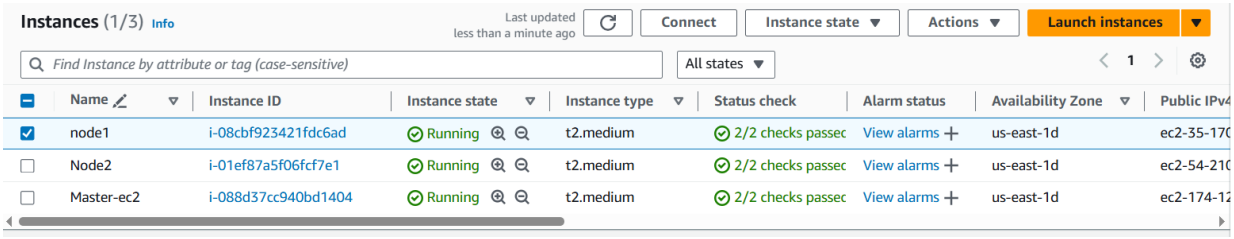
Node:


Name
node1
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
Mac
Ubuntu
Windows
Microsoft
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs


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

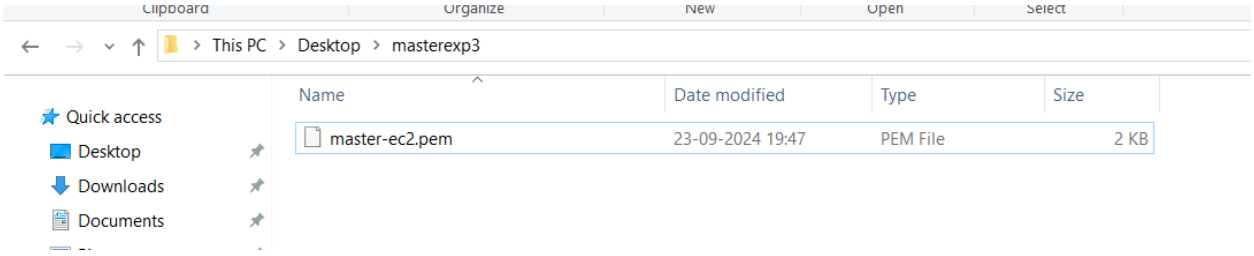

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
node1
Create new key pair



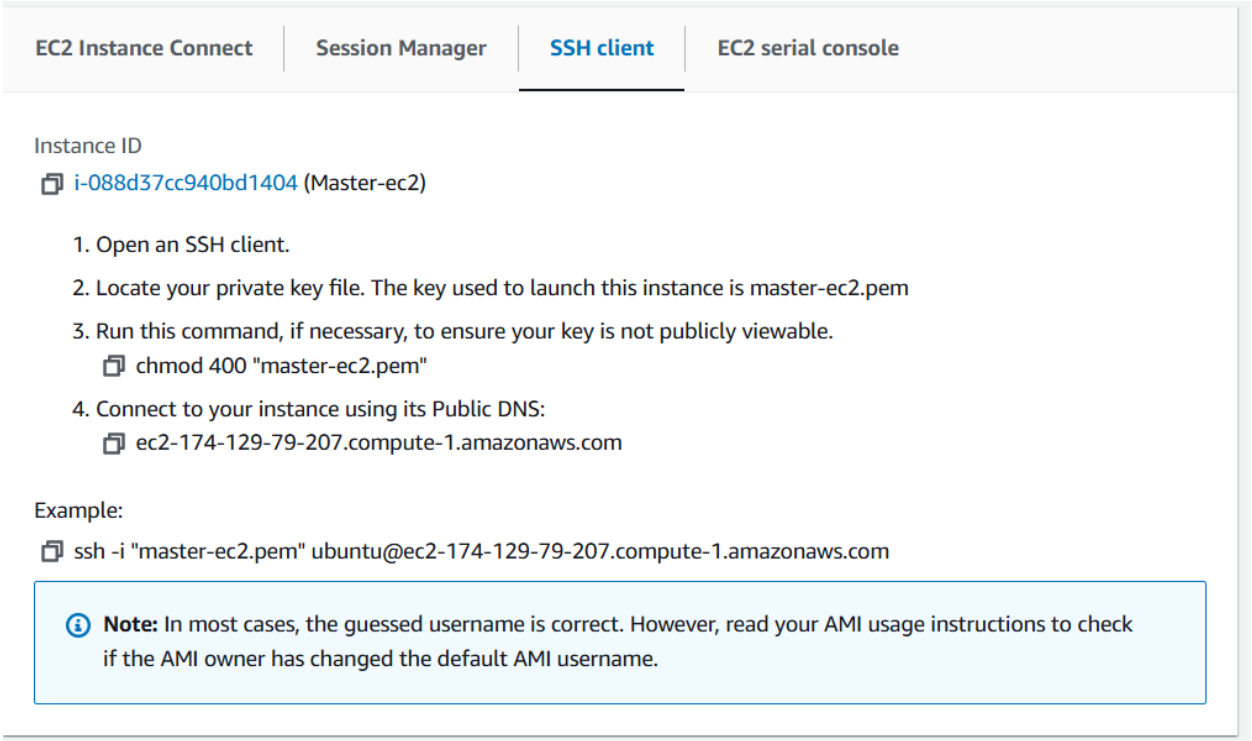


Step 3: Connect the instance and navigate to SSH client and copy the example command. Now open the folder in the terminal 3 times for Master, Node1 & Node 2 where our .pem key is stored and paste the Example command from ssh client (starting with ssh -i .....) in the terminal.

Downloaded Key:

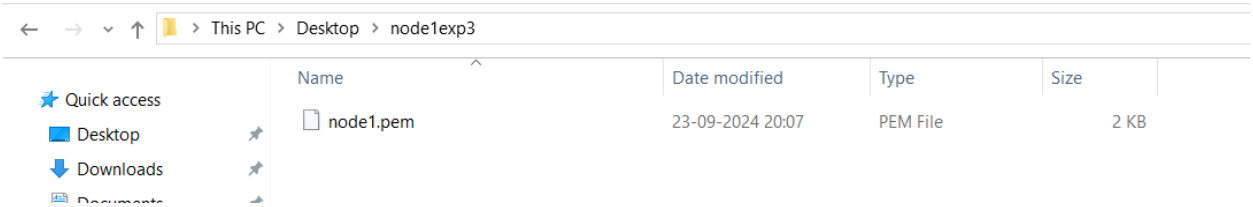


Master:

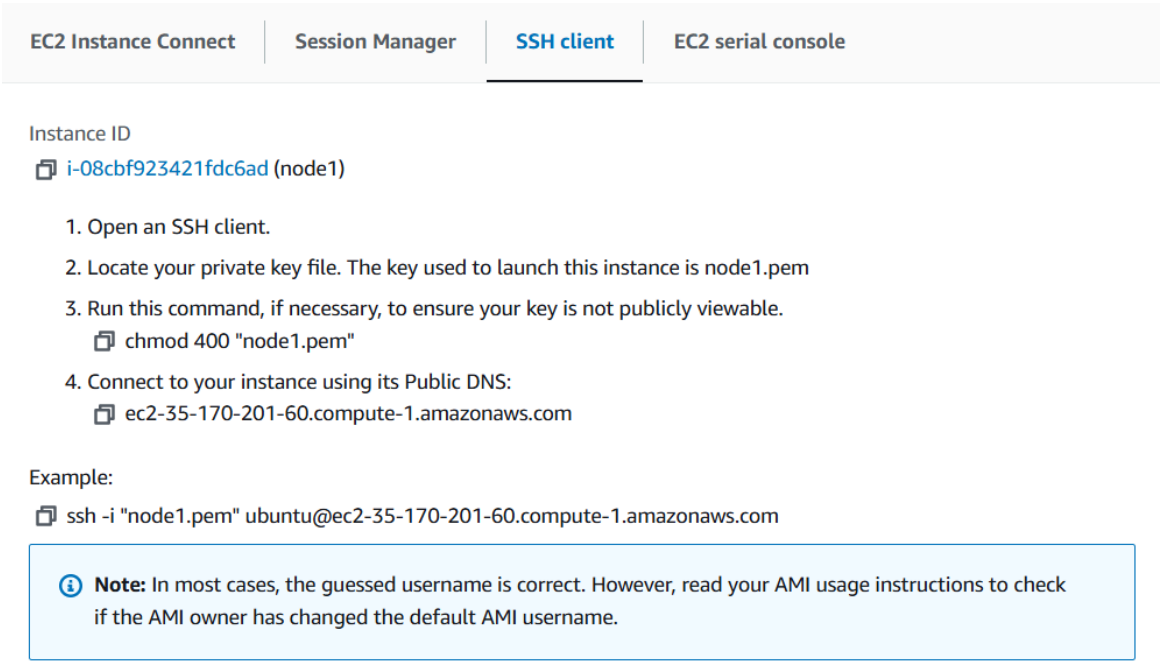



```
PS C:\Users\sathe> cd Desktop
PS C:\Users\sathe\Desktop> cd .\masterexp3\
PS C:\Users\sathe\Desktop\masterexp3> ssh -i "master-ec2.pem" ubuntu@ec2-174-129-79-207.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Mon Sep 23 15:00:49 UTC 2024

  System load:  0.0               Processes:             116
  Usage of /:   14.5% of 10.58GB  Users logged in:       1
  Memory usage: 5%                IPv4 address for enX0: 172.31.90.179
  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: Mon Sep 23 14:58:49 2024 from 58.146.120.240
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Downloaded Key Node:

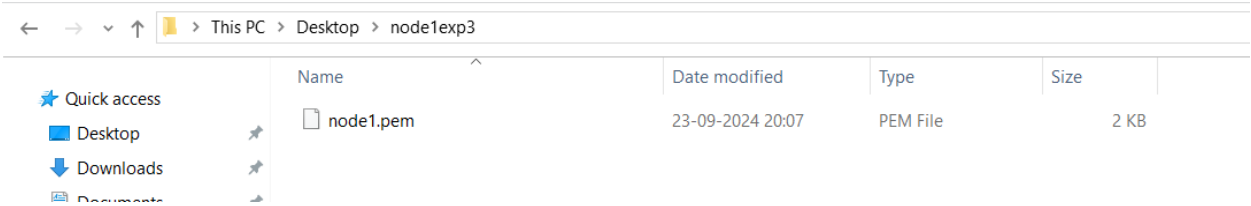


Node 1:

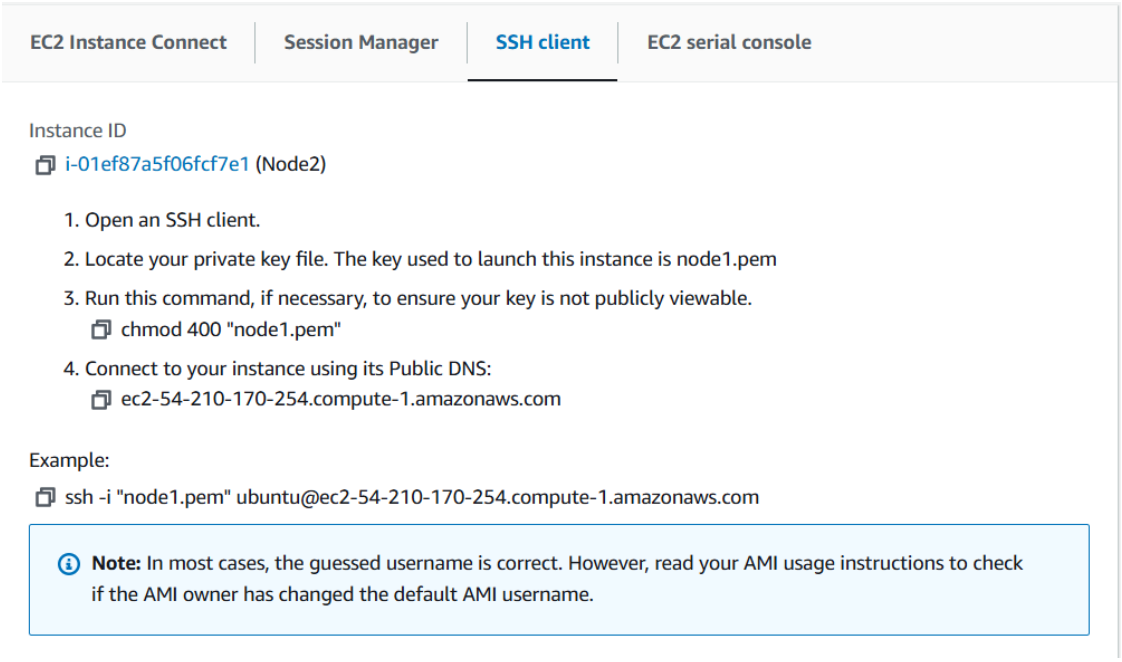


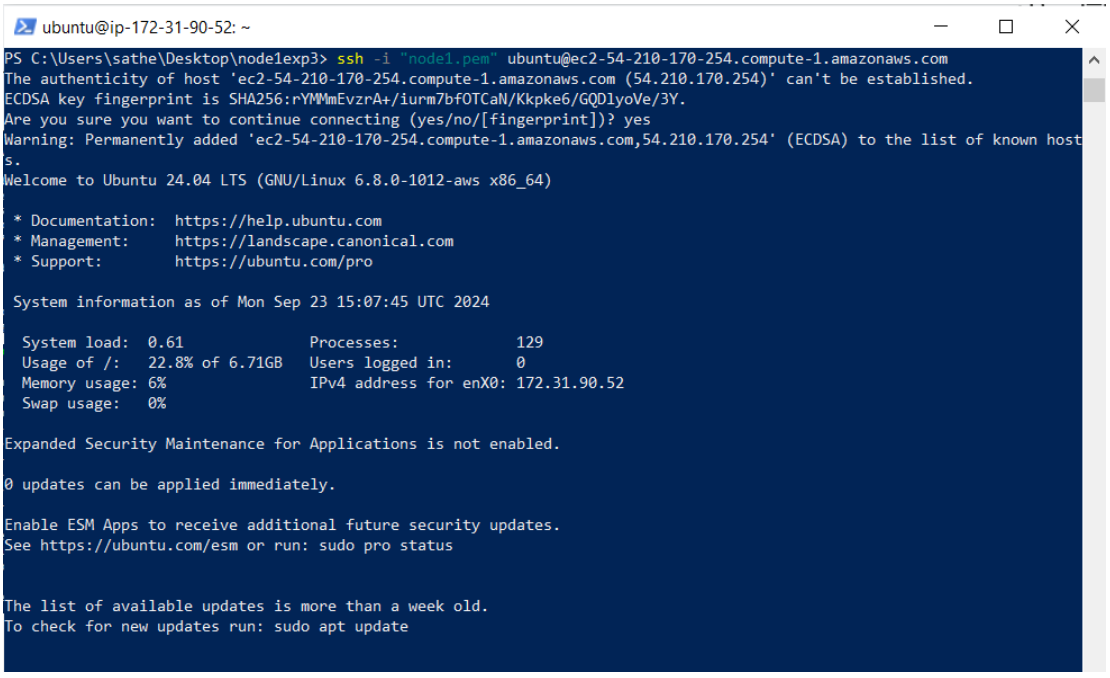

```
PS C:\Users\sathe\Desktop\node1exp3> ssh -i "node1.pem" ubuntu@ec2-35-170-201-60.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Mon Sep 23 15:04:19 UTC 2024

  System load:  0.0               Processes:             113
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for enX0: 172.31.90.171
  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: Mon Sep 23 14:41:42 2024 from 18.206.107.27
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Downloaded Key:



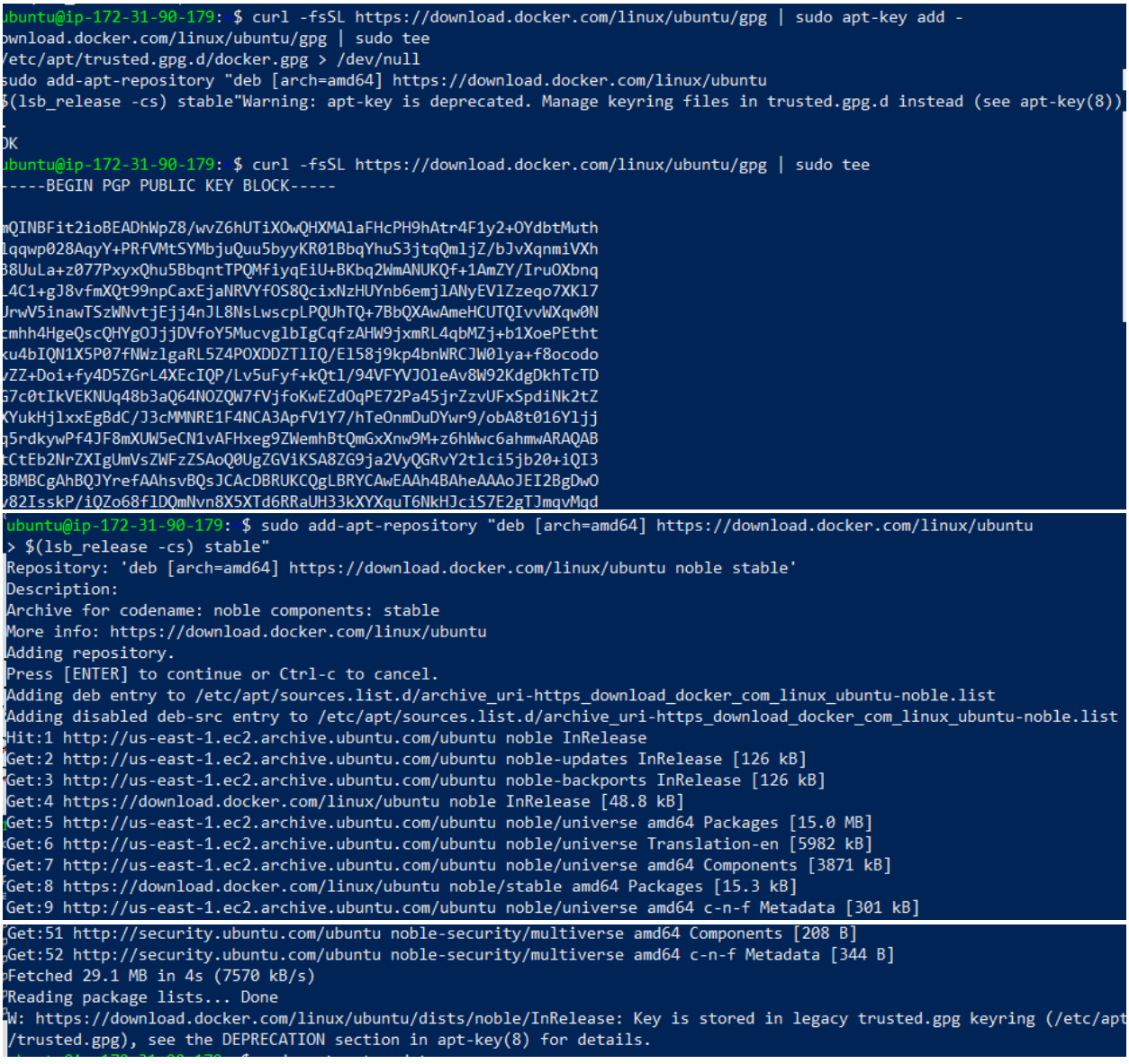
Node 2:





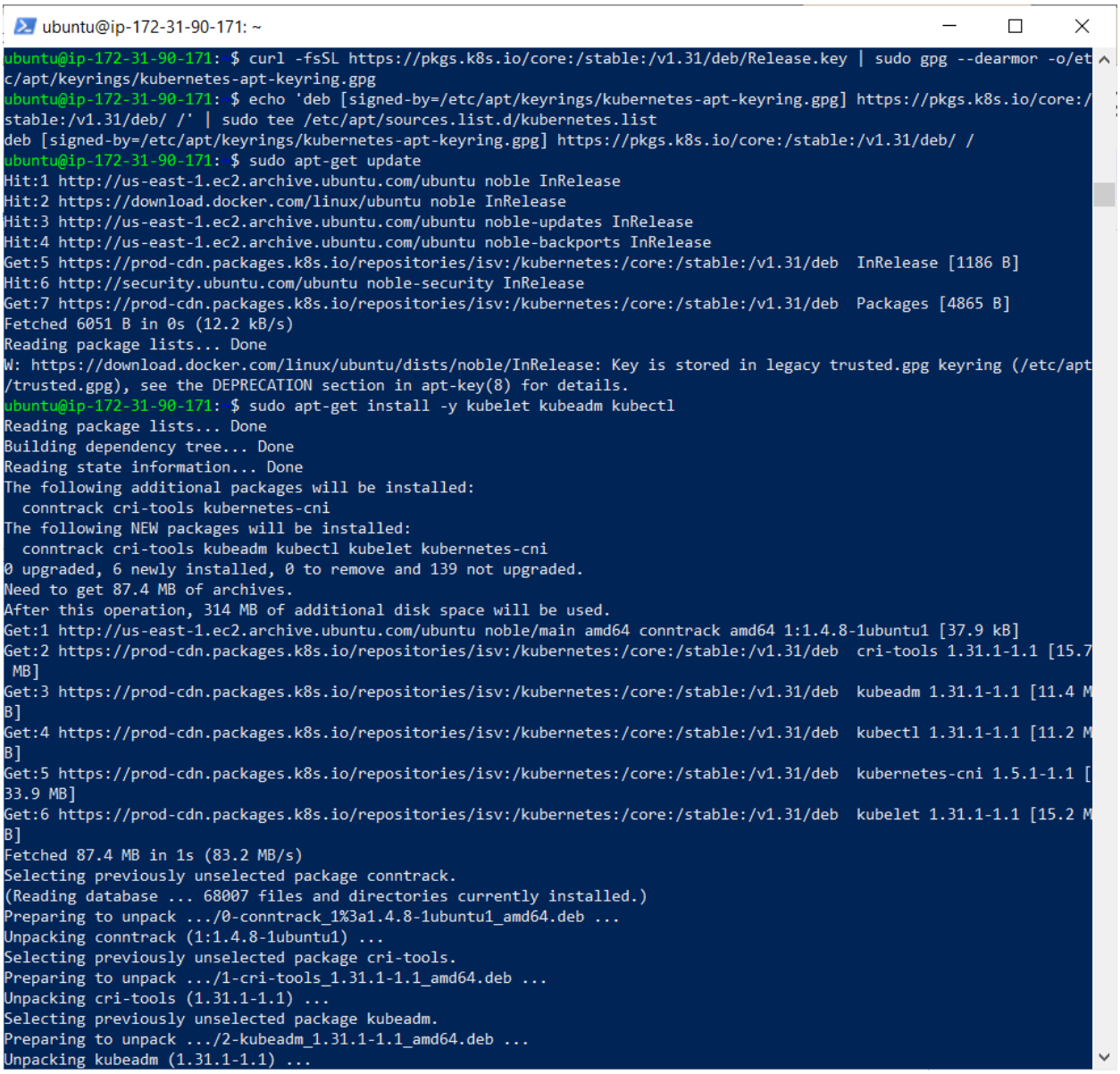
Step 4: Run on Master,Node 1,and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

- 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"



- sudo apt-get update
- sudo apt-get install -y docker-ce
- sudo mkdir -p /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {

"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF



```
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-172-31-90-179:~$ sudo mkdir -p /etc/docker
 | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOFcat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOFubuntu@ip-172-31-90-179:~$ sudo systemctl enable docker
```

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

```
EOFubuntu@ip-172-31-90-179:~$ sudo systemctl enable docker
o systemctl daemon-reload
sudo systemctl restart dockersudo systemctl daemon-reload
sudo systemctl restart dockerSynchronizing state of docker.service with SysV service script with /usr/lib/systemd/system
d-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

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-172-31-90-179: $ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o/et
c/apt/keyrings/kubernetes-apt-keyring.gpg
File '/etc/apt/keyrings/kubernetes-apt-keyring.gpg' exists. Overwrite? (y/N) y
ubuntu@ip-172-31-90-179: $ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/
stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

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

```
ubuntu@ip-172-31-90-179:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/ap
/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-90-179:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 139 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  cri-tools 1.31.1-1.1 [15.
 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubeadm 1.31.1-1.1 [11.4
B]
```

```
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-172-31-90-179:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

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

```
ubuntu@ip-172-31-90-179:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-90-179:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 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 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 M
B]
```

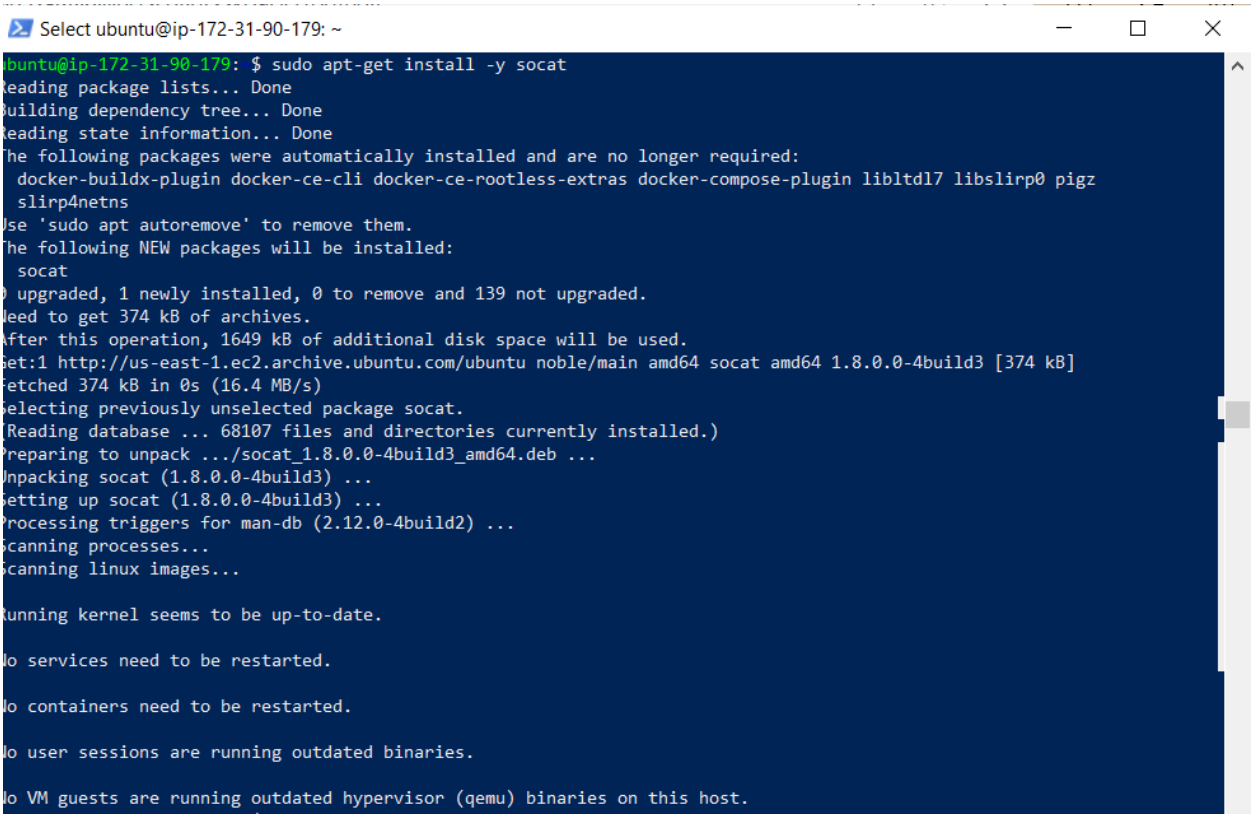
```
Running kernel seems to be up-to-date.

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-172-31-90-179:~$ sudo mkdir -p /etc/containerd
ubuntu@ip-172-31-90-179:~$ 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]
```

```
[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
```

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

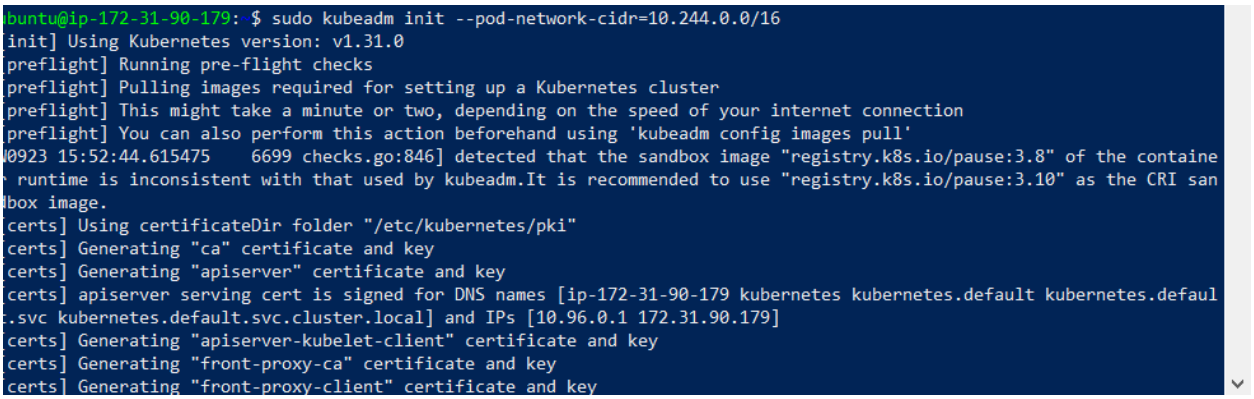
```
ubuntu@ip-172-31-90-179:~$ sudo systemctl restart containerd
ubuntu@ip-172-31-90-179:~$ sudo systemctl enable containerd
ubuntu@ip-172-31-90-179:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-09-23 15:48:22 UTC; 30s ago
       Docs: https://containerd.io
   Main PID: 6350 (containerd)
      Tasks: 8
     Memory: 13.1M (peak: 13.8M)
        CPU: 228ms
     CGroup: /system.slice/containerd.service
             └─6350 /usr/bin/containerd

Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390407424Z" level=info msg="Start subscrib>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390457482Z" level=info msg="Start recoveri>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390457512Z" level=info msg=serving... addr>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390506097Z" level=info msg=serving... addr>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390528794Z" level=info msg="Start event mo>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390542029Z" level=info msg="Start snapshot>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390550998Z" level=info msg="Start cni netw>
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.390557855Z" level=info msg="Start streamin>
Sep 23 15:48:22 ip-172-31-90-179 systemd[1]: Started containerd.service - containerd container runtime.
Sep 23 15:48:22 ip-172-31-90-179 containerd[6350]: time="2024-09-23T15:48:22.394691700Z" level=info msg="containerd suc>
```

- 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



```
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 172.31.90.179:6443 --token aw1zqe.njkmrtdi6k1zpz5k \
        --discovery-token-ca-cert-hash sha256:2c1c7752cb31b471bfb5281cc2426e96c19b9f06d0df9749f5aca61691986791
```

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-172-31-90-179:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-90-179:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
cp: overwrite '/home/ubuntu/.kube/config'? yes
ubuntu@ip-172-31-90-179:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

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

```
ubuntu@ip-172-31-90-179:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE     VERSION
ip-172-31-90-179   NotReady   control-plane   3m45s   v1.31.1
```

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-172-31-90-171:~$ sudo kubeadm join 172.31.90.179:6443 --token aw1zqe.njkmrtdi6k1zpz5k \
>         --discovery-token-ca-cert-hash sha256:2c1c7752cb31b471bfb5281cc2426e96c19b9f06d0df9749f5aca61691986791
[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 501.662134ms
[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.
```

Node 2:

```
ubuntu@ip-172-31-90-52:~$ sudo kubeadm join 172.31.90.179:6443 --token aw1zqe.njkmrtdi6k1zpz5k \
>         --discovery-token-ca-cert-hash sha256:2c1c7752cb31b471bfb5281cc2426e96c19b9f06d0df9749f5aca61691986791
[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 501.462877ms
[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.
```

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

```
ubuntu@ip-172-31-90-179:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE   VERSION
ip-172-31-90-171   NotReady   <none>          31s   v1.31.1
ip-172-31-90-179   NotReady   control-plane   17m   v1.31.1
ip-172-31-90-52    NotReady   <none>          22s   v1.31.1
```

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-172-31-90-179:~$ 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
```

- sudo systemctl status kubelet

```
ubuntu@ip-172-31-90-179:~$ 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 Mon 2024-09-23 15:53:03 UTC; 19min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 7369 (kubelet)
      Tasks: 10 (limit: 4676)
     Memory: 32.5M (peak: 33.0M)
        CPU: 16.010s
     CGroup: /system.slice/kubelet.service
             └─7369 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/ku>

Sep 23 16:12:08 ip-172-31-90-179 kubelet[7369]: I0923 16:12:08.364641    7369 pod_container_deletor.go:80] "Container n>
Sep 23 16:12:08 ip-172-31-90-179 kubelet[7369]: I0923 16:12:08.365702    7369 scope.go:117] "RemoveContainer" container>
Sep 23 16:12:08 ip-172-31-90-179 kubelet[7369]: I0923 16:12:08.370899    7369 scope.go:117] "RemoveContainer" container>
Sep 23 16:12:08 ip-172-31-90-179 kubelet[7369]: I0923 16:12:08.377678    7369 scope.go:117] "RemoveContainer" container>
Sep 23 16:12:09 ip-172-31-90-179 kubelet[7369]: I0923 16:12:09.368160    7369 pod_container_deletor.go:80] "Container n>
Sep 23 16:12:09 ip-172-31-90-179 kubelet[7369]: I0923 16:12:09.368187    7369 scope.go:117] "RemoveContainer" container>
Sep 23 16:12:09 ip-172-31-90-179 kubelet[7369]: E0923 16:12:09.456550    7369 pod_workers.go:1301] "Error syncing pod, >
Sep 23 16:12:10 ip-172-31-90-179 kubelet[7369]: I0923 16:12:10.385576    7369 scope.go:117] "RemoveContainer" container>
Sep 23 16:12:10 ip-172-31-90-179 kubelet[7369]: E0923 16:12:10.385687    7369 pod_workers.go:1301] "Error syncing pod, >
Sep 23 16:12:12 ip-172-31-90-179 kubelet[7369]: I0923 16:12:12.385087    7369 scope.go:117] "RemoveContainer" container>
```

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

```
ubuntu@ip-172-31-90-179:~$ kubectl get nodes -o wide
NAME              STATUS   ROLES           AGE     VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE           KERNEL-VE
RSION   CONTAINER-RUNTIME
ip-172-31-90-171   Ready    <none>          2m15s   v1.31.1   172.31.90.171   <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
ip-172-31-90-179   Ready    control-plane   19m     v1.31.1   172.31.90.179   <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
ip-172-31-90-52    Ready    <none>          2m6s    v1.31.1   172.31.90.52    <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
ubuntu@ip-172-31-90-179:~$ kubectl label node ip-172-31-90-171 kubernetes.io/role=Node1
node/ip-172-31-90-171 labeled
```

The Roles are not yet assigned to the Nodes

- **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-172-31-90-179:~$ kubectl label node ip-172-31-90-171 kubernetes.io/role=Node1
node/ip-172-31-90-171 labeled
ubuntu@ip-172-31-90-179:~$ kubectl label node ip-172-31-90-52 kubernetes.io/role=Node2
node/ip-172-31-90-52 labeled
```

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

```
ubuntu@ip-172-31-90-179:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-90-171   Ready    Node1           62m   v1.31.1
ip-172-31-90-179   Ready    control-plane   80m   v1.31.1
ip-172-31-90-52    Ready    Node2           62m   v1.31.1
```

Conclusion:

Learnt about kubernetes cluster architecture. Joined multiple worker nodes to master node.

Error:

The error I faced was that my private key was not accessible by others since it had too many open permissions.

```
PS C:\Users\sathe\Desktop> cd .\node1exp3\
PS C:\Users\sathe\Desktop\node1exp3> ssh -i "node1.pem" ubuntu@ec2-35-170-201-60.compute-1.amazonaws.com
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions for 'node1.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "node1.pem": bad permissions
ubuntu@ec2-35-170-201-60.compute-1.amazonaws.com: Permission denied (publickey).
PS C:\Users\sathe\Desktop\node1exp3>
```

To overcome it following command was used which by passes this error.

```
C:\Windows\system32>cd C:\Users\sathe\Desktop\masterexp3

C:\Users\sathe\Desktop\masterexp3>icacls master-ec2.pem /inheritance:r /grant:r %username%:F
processed file: master-ec2.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\sathe\Desktop\masterexp3>cd ..

C:\Users\sathe\Desktop>cd node1exp3

C:\Users\sathe\Desktop\node1exp3>icacls node1.pem /inheritance:r /grant:r %username%:F
processed file: node1.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\sathe\Desktop\node1exp3>
```

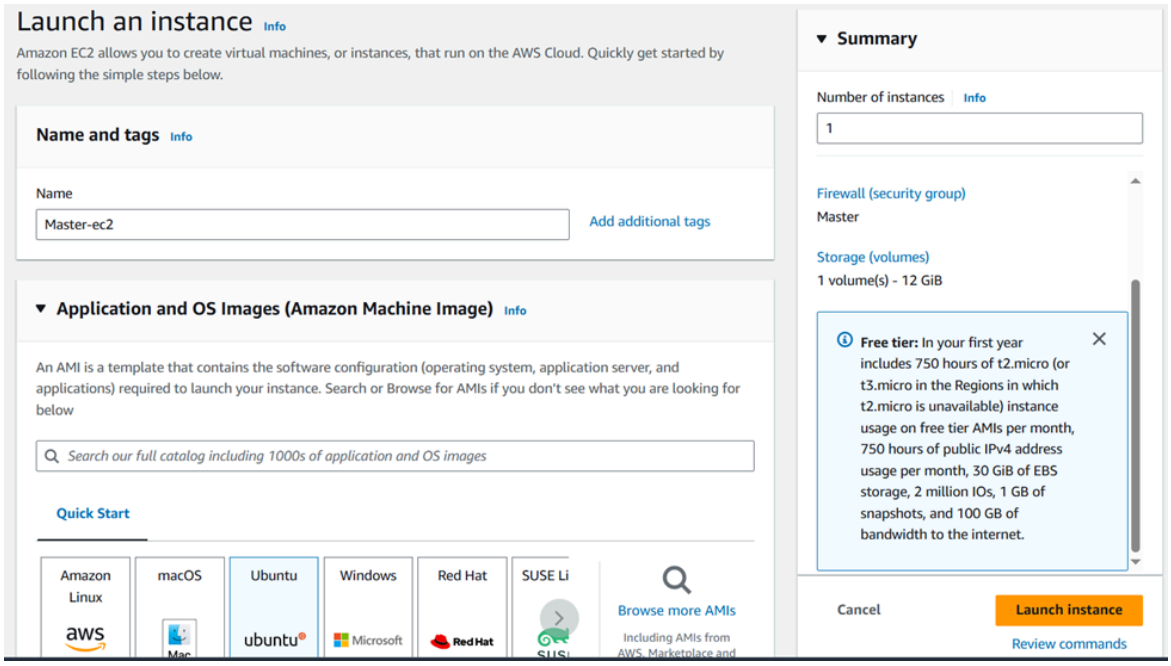
NOTE: This command needs to be executed in administrator mode

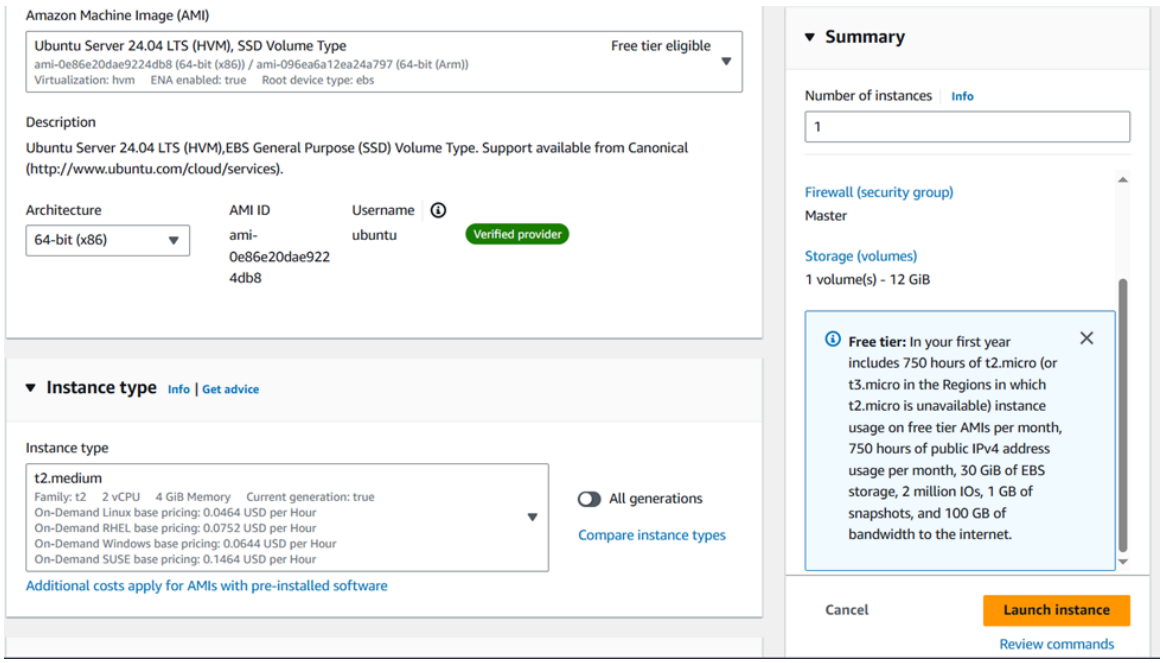
Siddhant Sathe **D15A/51**

Experiment 4

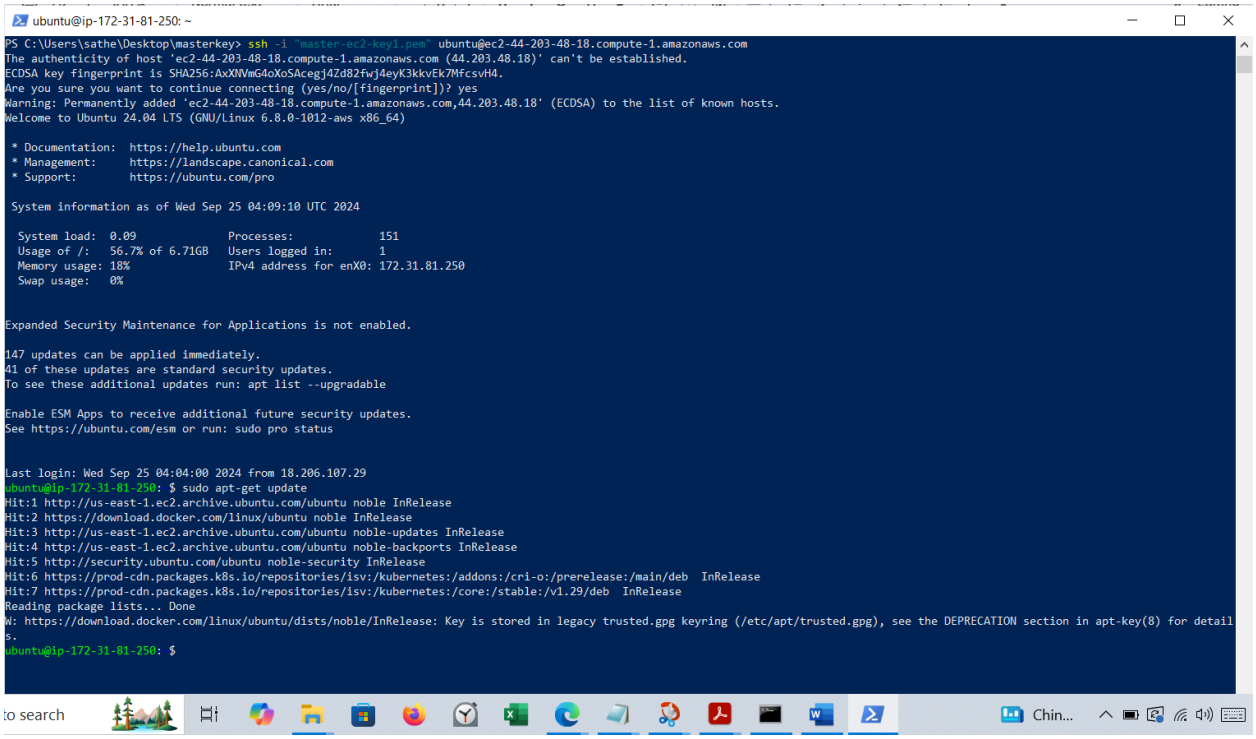
AIm: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

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.





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.



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-172-31-90-179:~$ 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 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [15.3 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [530 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [128 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8548 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [154 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.6 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
```

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

```
ubuntu@ip-172-31-81-250:~$ sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 140 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
```

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

Restarting services...
 systemctl restart packagekit.service

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.
```

Step 4:Download and add the GPG key:

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

```
ubuntu@ip-172-31-81-250: $ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
ubuntu@ip-172-31-81-250: $ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/ku
bernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
```

Step 5:Update package list:

sudo apt-get update

```
ubuntu@ip-172-31-81-250: $ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:6 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Hit:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb  InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.63.100 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for detail
s.
```

sudo apt-get install -y kubectl

```
ubuntu@ip-172-31-81-250:~$ sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 137 not upgraded.
ubuntu@ip-172-31-81-250:~$ kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 6: Verify the nodes

kubectl get nodes

```
ubuntu@ip-172-31-81-250:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-81-250   Ready    control-plane   9h    v1.29.0
ip-172-31-81-86    Ready    Node1           9h    v1.29.0
ip-172-31-83-216   Ready    Node2           9h    v1.29.0
ubuntu@ip-172-31-81-250:~$ nano nginx-deployment.yaml
ubuntu@ip-172-31-81-250:~$ ubuntu@ip-172-31-81-250:~$ nano nginx-service.yaml
```

Step 7: Create the Deployment YAML File

1. Create a file named nginx-service.yaml:*

yaml

apiVersion: v1

kind: Service

metadata:

```
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

And deploy it

```
ubuntu@ip-172-31-81-250:~$ nano nginx-service.yaml
ubuntu@ip-172-31-81-250:~$ ubuntu@ip-172-31-81-250:~$ kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Step 8: Verify the deployment:

kubectl get deployments

kubectl get pods

kubectl get services

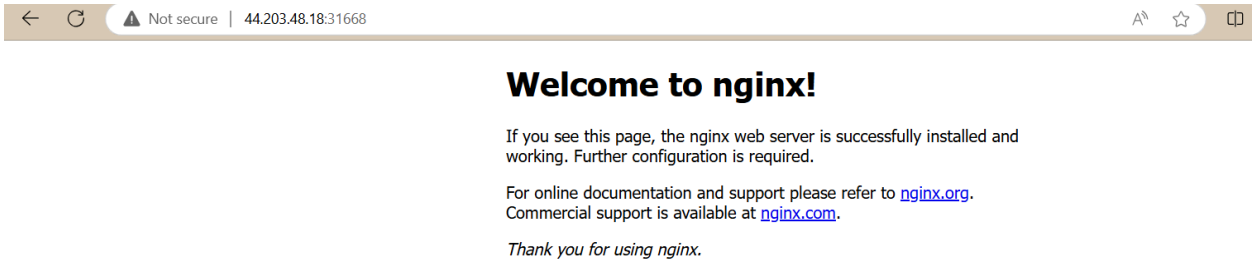
```
ubuntu@ip-172-31-81-250:~$ kubectl get deployments
   kubectl get pods
   kubectl get servicesNAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           4m55s
ubuntu@ip-172-31-81-250:~$    kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-2l9tr   1/1     Running   0          4m55s
nginx-deployment-86dcfdf4c6-ksdx9   1/1     Running   0          4m55s
nginx-deployment-86dcfdf4c6-qdqgx   1/1     Running   0          4m55s
ubuntu@ip-172-31-81-250:~$    kubectl get services
NAME            TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1      <none>        443/TCP        9h
nginx-service   LoadBalancer   10.106.1.134   <pending>     80:31668/TCP   60s
```

Step 9: Forward the service port:

kubectl port-forward service/nginx-service 8080:80

```
ubuntu@ip-172-31-81-250:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

Step 10: Open your web browser and go to http://44.203.48.18:31668/



Error : the connection string changes if the ip address is changed

```
PS C:\Users\sathe\Desktop\masterkey> ssh -i "master-ec2-key1.pem" ubuntu@ec2-3-89-117-235.compute-1.amazonaws.com
ssh: connect to host ec2-3-89-117-235.compute-1.amazonaws.com port 22: Connection timed out
PS C:\Users\sathe\Desktop\masterkey> ssh -i "master-ec2-key1.pem" ubuntu@ec2-44-203-48-18.compute-1.amazonaws.com
The authenticity of host 'ec2-44-203-48-18.compute-1.amazonaws.com (44.203.48.18)' can't be established.
ECDSA key fingerprint is SHA256:AxXNVmG4oXoSAcegj4Zd82fwj4eyK3kkvEk7MfcsvH4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-203-48-18.compute-1.amazonaws.com,44.203.48.18' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)
```

To correct it copy the connection string again

Error due to incorrect indentation

```
ubuntu@ip-172-31-81-250: $ kubectl apply -f nginx-deployment.yaml
error: error parsing nginx-deployment.yaml: error converting YAML to JSON: yaml: line 2: mapping values are not allowed in this context
```

Siddhant Sathe **D15A/51**

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.

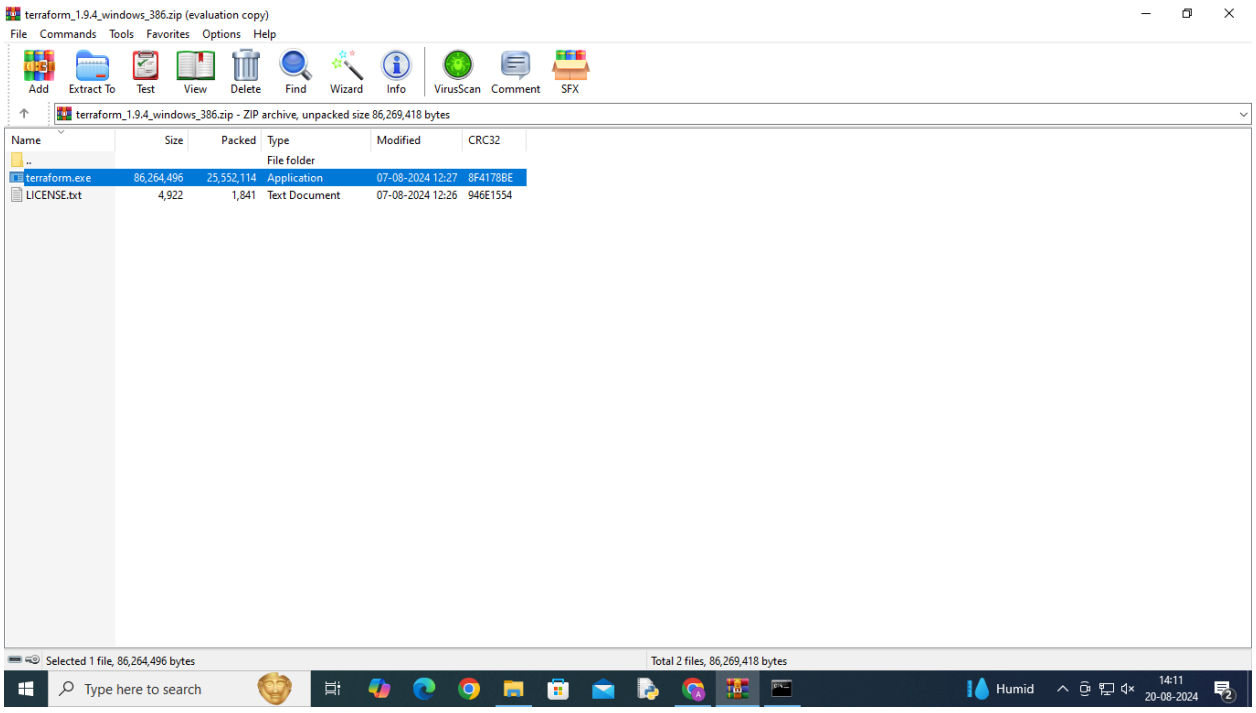
6.Outputs:
Outputs are used to extract information from the Terraform-managed infrastructure and display it after the execution of a Terraform plan or apply.

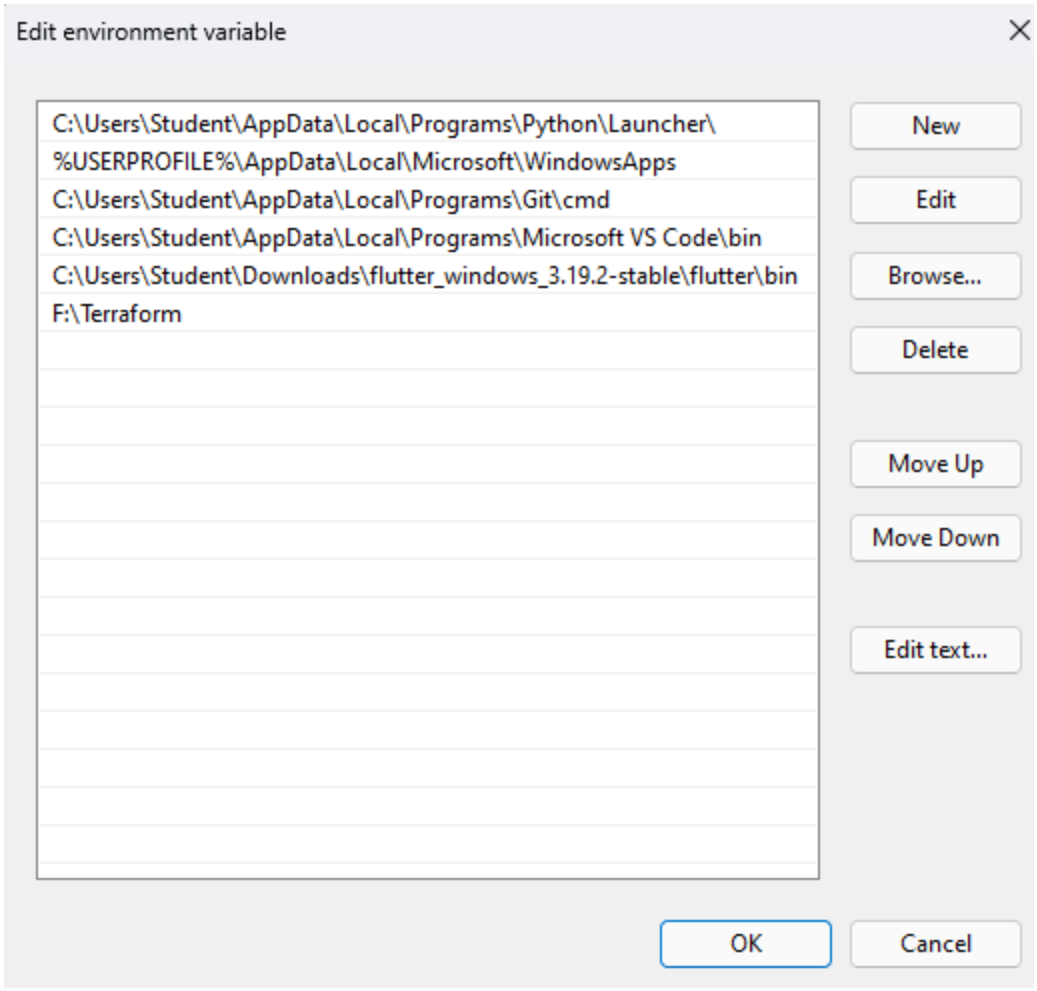
Terraform Lifecycle

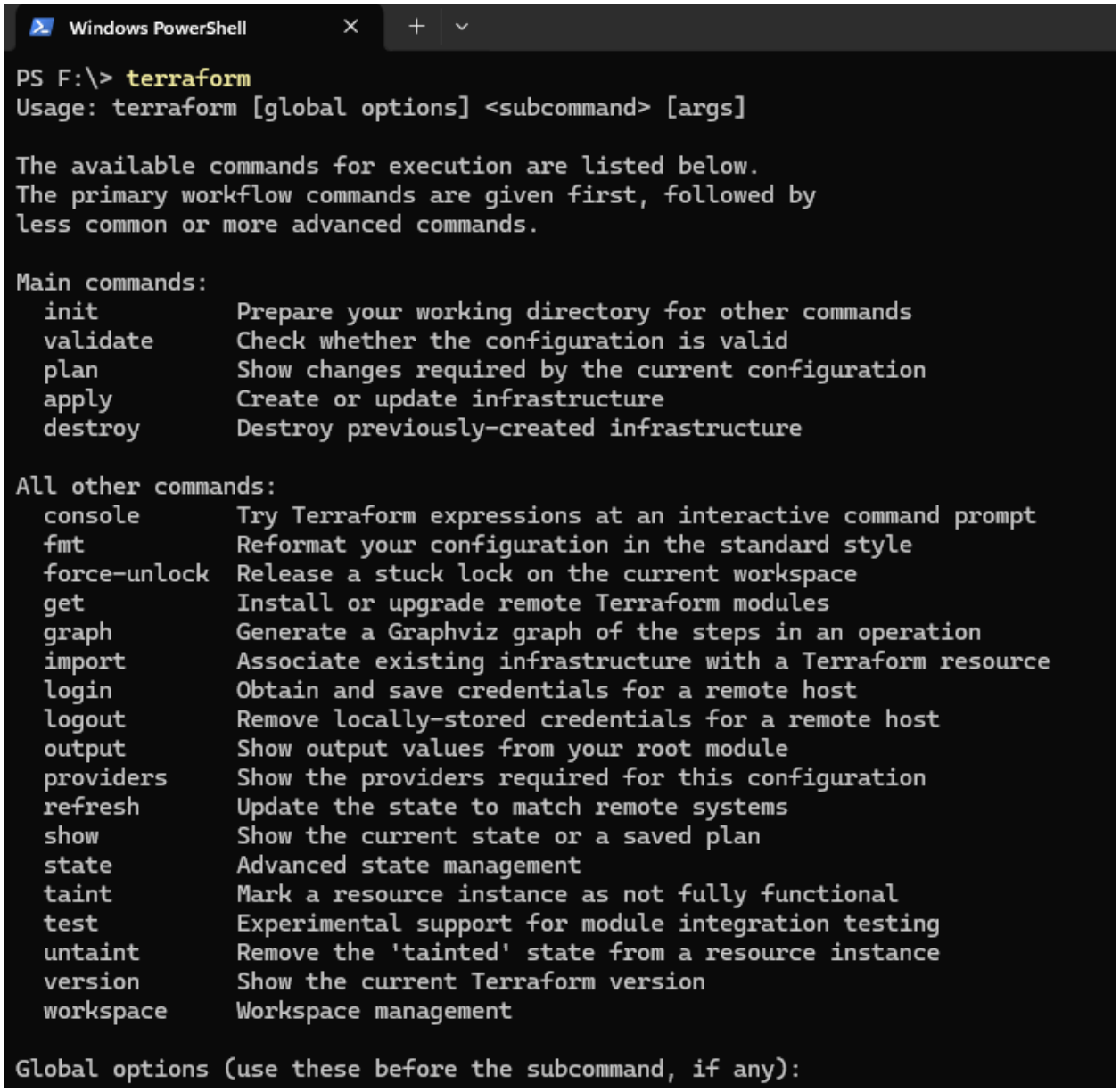
1.Write:
Write the configuration file (typically with .tf extension) using HCL to describe the desired infrastructure.

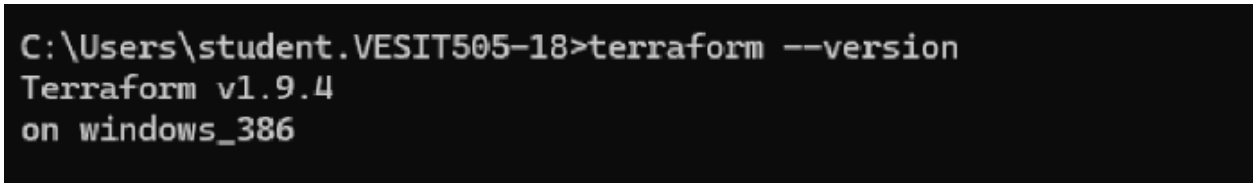
2.Initialize (terraform init):
Initialize the working directory containing the configuration files. This command downloads the necessary provider plugins and sets up the environment.

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.




Edit environment variable
C:\Users\Student\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\Student\AppData\Local\Programs\Git\cmd
C:\Users\Student\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\Student\Downloads\flutter_windows_3.19.2-stable\flutter\bin
F:\Terraform
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel


Windows PowerShell
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
Global options (use these before the subcommand, if any):

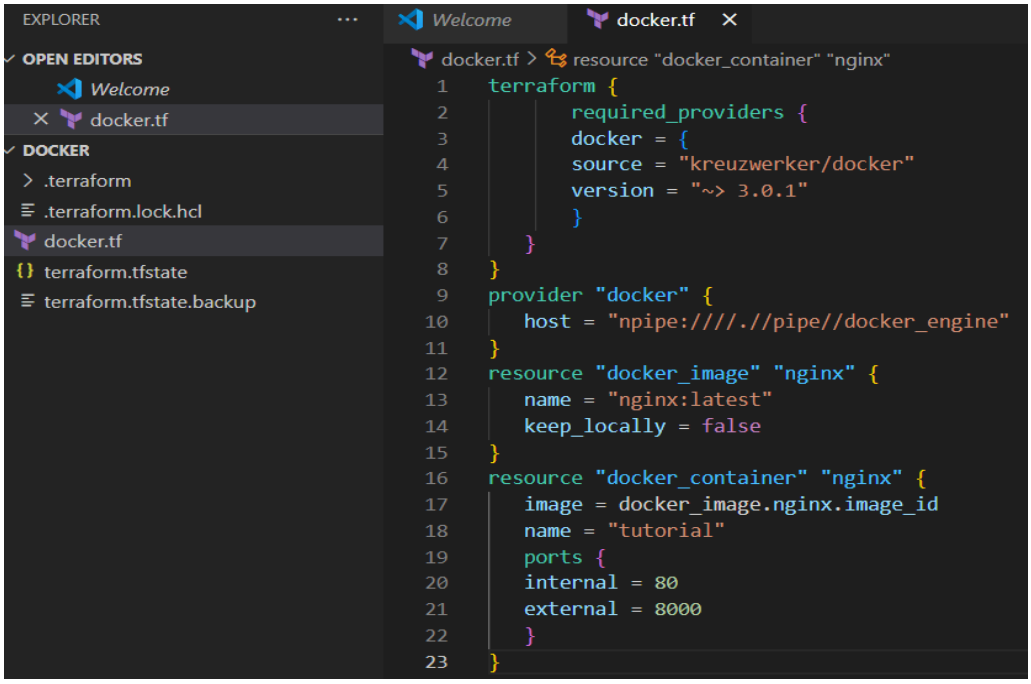

C:\Users\student.VESIT505-18>terraform --version
Terraform v1.9.4
on windows_386

Siddhant Sathe **D15A/51**

Aim: Exp 6 To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker) fdp

Code:

```
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
  }
}
```



Output:

```
C:\Users\sathe>docker --version
Docker version 26.1.4, build 5650f9b
```

Docker Images before using terraform commands:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                            TAG         IMAGE ID       CREATED         SIZE
mindsdb/mindsdb-docker-extension      1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                       latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                       v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker              latest      c1f619b6477e   9 months ago    18.6MB
hello-world                           latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                           latest      2932f7e14783   54 years ago    144MB
```

Terraform init:

```
PS D:\Siddhant\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!

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

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

Terraform plan:

```
PS D:\Siddhant\Terraform Scripts\Docker> terraform plan

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

Terraform will perform the following actions:

  # docker_container.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)
      + image                                       = (known after apply)
      + init                                        = (known after apply)
      + ipc_mode                                    = (known after apply)
      + log_driver                                  = (known after apply)
      + logs                                        = false
      + must_run                                    = true
      + name                                        = "tutorial"
      + network_data                                = (known after apply)
      + read_only                                   = false
      + remove_volumes                              = true
      + restart                                     = "no"
      + rm                                          = false
      + runtime                                     = (known after apply)
      + security_opts                               = (known after apply)
      + shm_size                                    = (known after apply)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
      + wait_timeout                                = 60
      + ports {
          + external = 8000
          + internal = 80
          + ip       = "0.0.0.0"
          + protocol = "tcp"
        }
    }

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

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

Terraform apply:

```
PS D:\Siddhant\Terraform Scripts\Docker> terraform apply

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

Terraform will perform the following actions:

  # docker_container.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)
```

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

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

  Enter a value: yes

docker_image.nginx: Creating...
docker_image.nginx: Still creating... [10s elapsed]
docker_image.nginx: Still creating... [20s elapsed]
docker_image.nginx: Creation complete after 29s [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 5s [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]

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

Docker Images after using terraform apply:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                          TAG         IMAGE ID       CREATED         SIZE
nginx                               latest      5ef79149e0ec   10 days ago     188MB
mindsdb/mindsdb-docker-extension    1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                     latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                     v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker            latest      c1f619b6477e   9 months ago    18.6MB
hello-world                         latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                         latest      2932f7e14783   54 years ago    144MB
```

Terraform destroy:

```
PS D:\Siddhant\Terraform Scripts\Docker> terraform destroy
docker_image.nginx: Refreshing state... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_container.nginx: Refreshing state... [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]

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

Terraform will perform the following actions:

  # docker_container.nginx will be destroyed
  - resource "docker_container" "nginx" {
      - attach                                      = false -> null
      - command                                     = [
          - "nginx",
          - "-g",
          - "daemon off;",
        ] -> null
      - container_read_refresh_timeout_milliseconds = 15000 -> null
      - cpu_shares                                  = 0 -> null
      - dns                                         = [] -> null
      - dns_opts                                    = [] -> null
      - dns_search                                  = [] -> null
      - entrypoint                                  = [
          - "/docker-entrypoint.sh",
        ] -> null
      - env                                         = [] -> null
      - group_add                                   = [] -> null
      - hostname                                    = "21a0a0573bc9" -> null
      - id                                          = "21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c" -> null
```

```
  # docker_image.nginx will be destroyed
  - resource "docker_image" "nginx" {
      - id           = "sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest" -> null
      - image_id     = "sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03c" -> null
      - keep_locally = false -> null
      - name         = "nginx:latest" -> null
      - repo_digest  = "nginx@sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add" -> null
    }

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

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

  Enter a value: yes

docker_container.nginx: Destroying... [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]
docker_container.nginx: Destruction complete after 1s
docker_image.nginx: Destroying... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_image.nginx: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

Docker Images after using terraform destroy:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                          TAG         IMAGE ID       CREATED         SIZE
mindsdb/mindsdb-docker-extension    1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                     latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                     v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker            latest      c1f619b6477e   9 months ago    18.6MB
hello-world                         latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                         latest      2932f7e14783   54 years ago    144MB
```

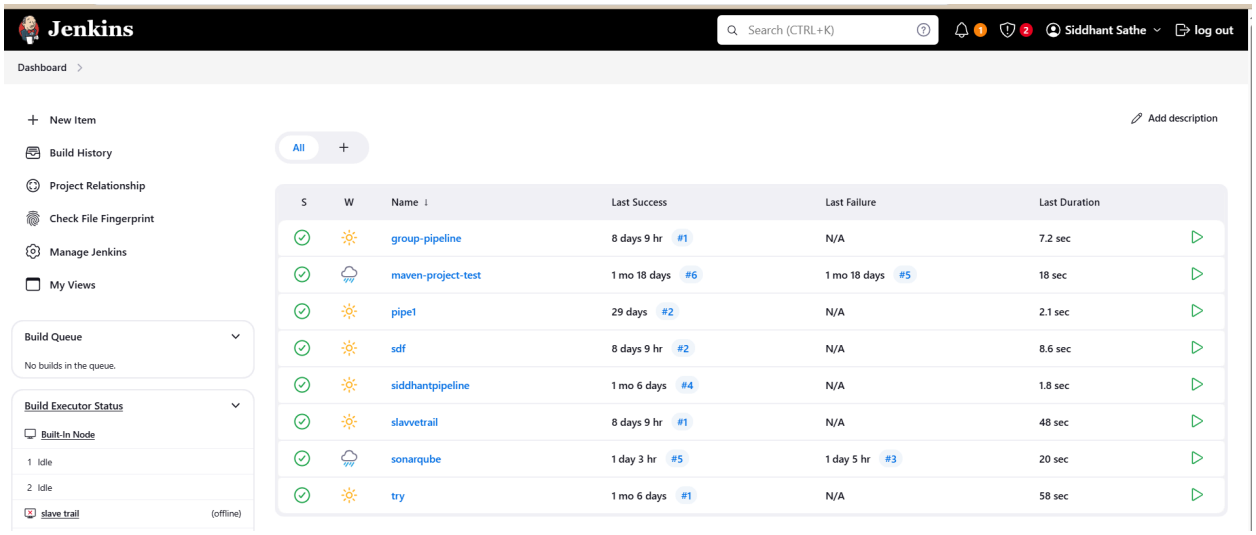
Siddhant Sathe **D15A/51**

Experiment 7

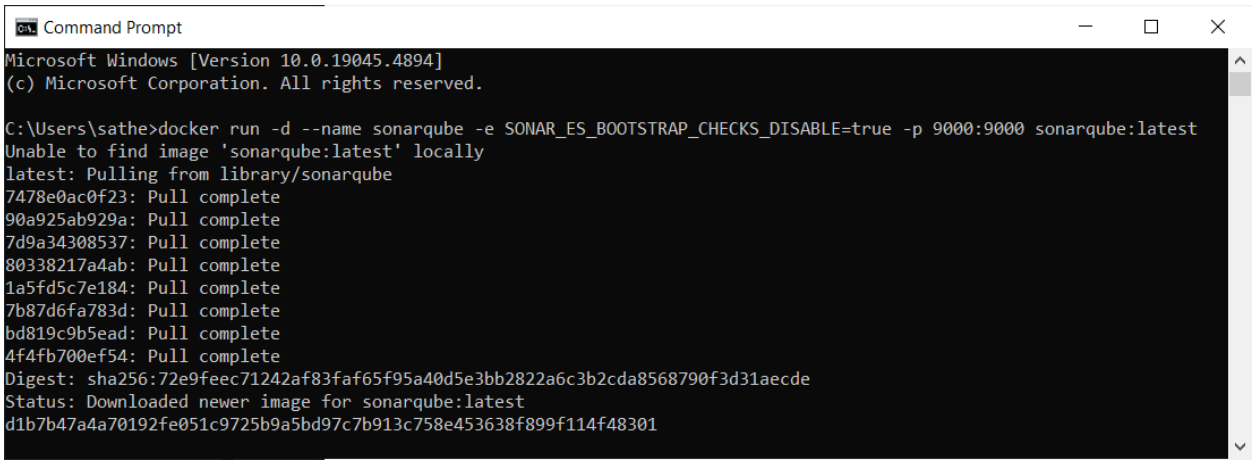
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Theory:

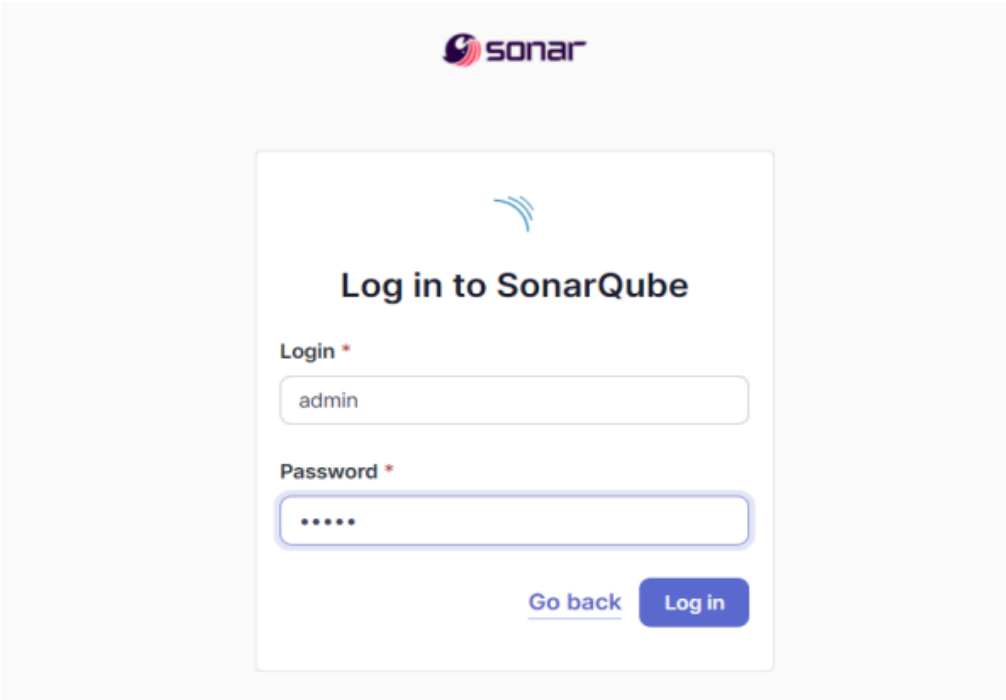
Step-1: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



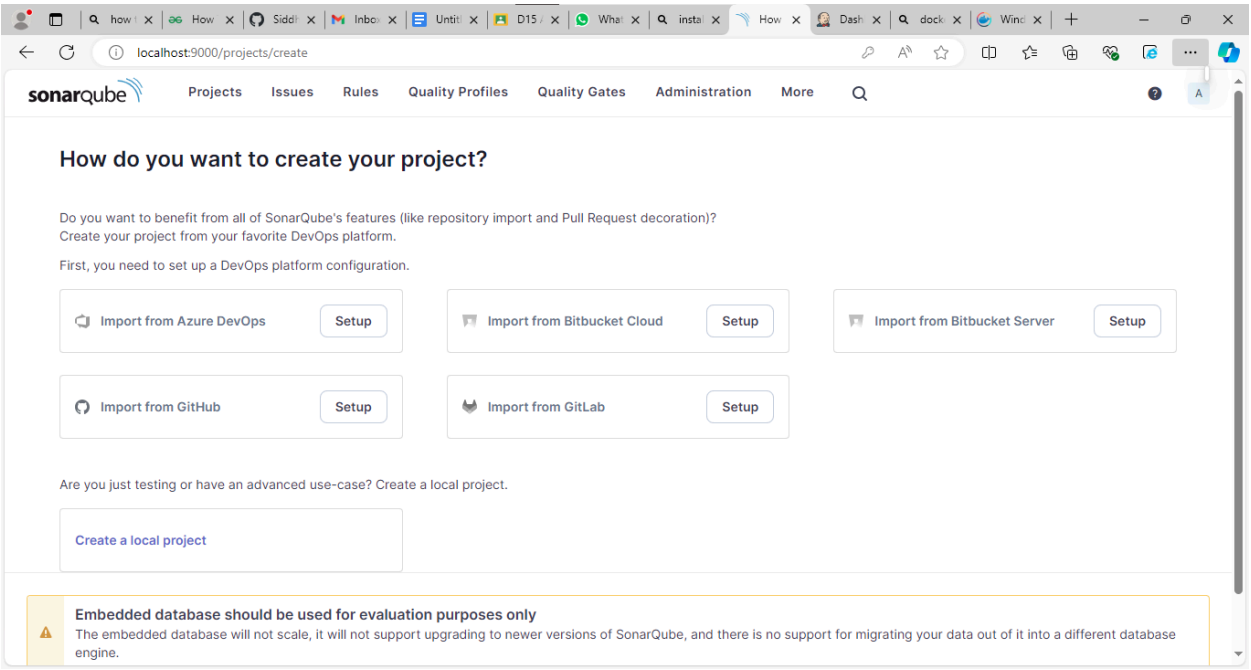
Step-2: Run SonarQube in a Docker container using this command :- docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest



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



1 of 2

Create a local project

Project display name *

sonarqube

Project key *

sonarqube

Main branch name *

main

The name of your project's default branch Learn More

Cancel Next

Set up project for Clean as You Code

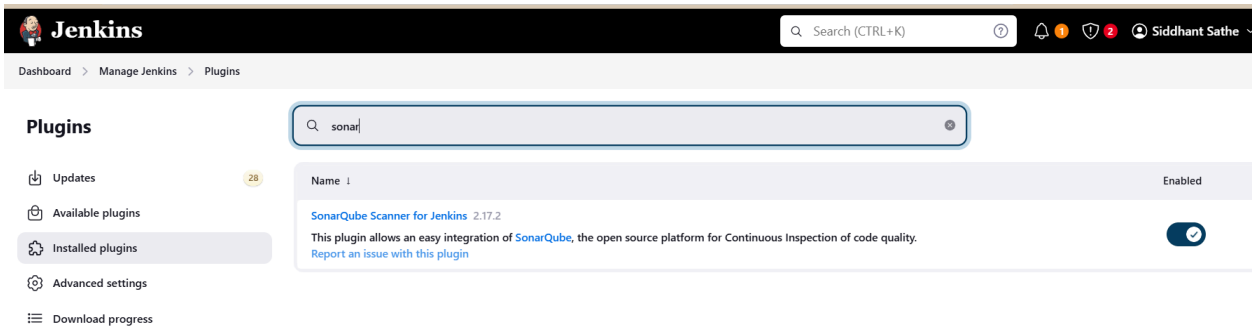
The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: Defining New Code

Choose the baseline for new code for this project

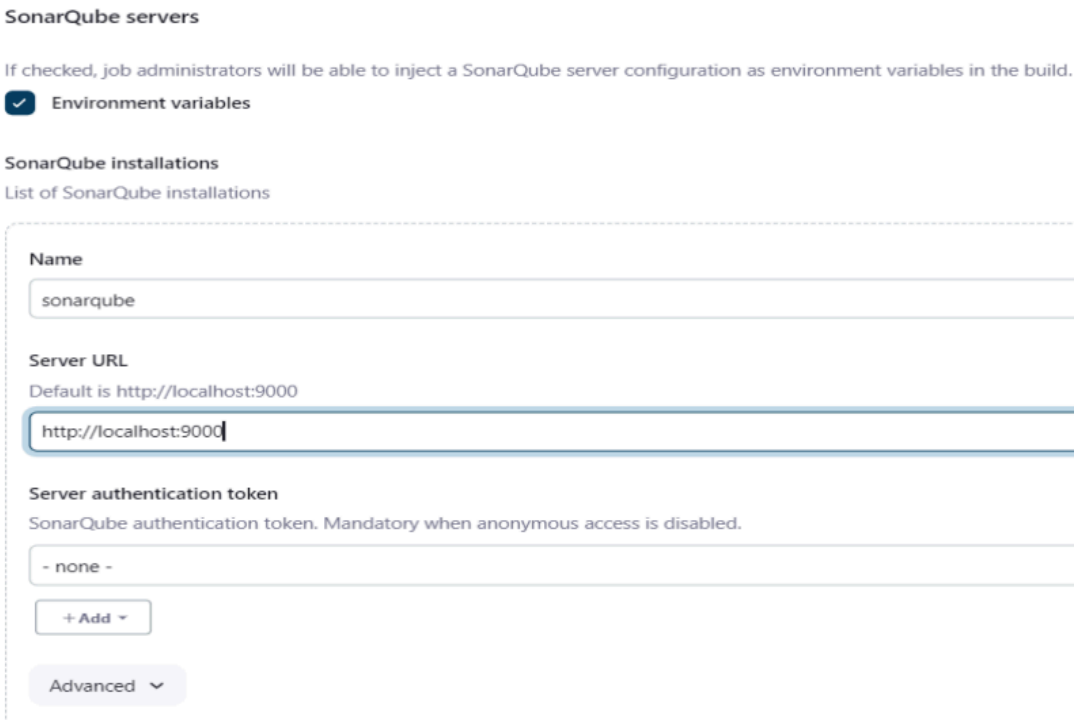
- (•) **Use the global setting**
  - **Previous version**
    - Any code that has changed since the previous version is considered new code.
    - Recommended for projects following regular versions or releases.
- ( ) **Define a specific setting for this project**
  - ( ) **Previous version**
    - Any code that has changed since the previous version is considered new code.
    - Recommended for projects following regular versions or releases.
  - ( ) **Number of days**
    - Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
    - Recommended for projects following continuous delivery.
  - ( ) **Reference branch**
    - Choose a branch as the baseline for the new code.
    - Recommended for projects using feature branches.

Back Create project

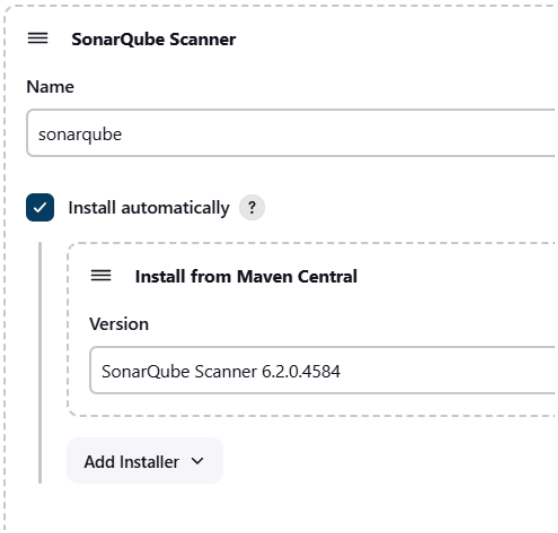
Step-5: Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins → Plugins and search for SonarQube Scanner in Available Plugins and install it.



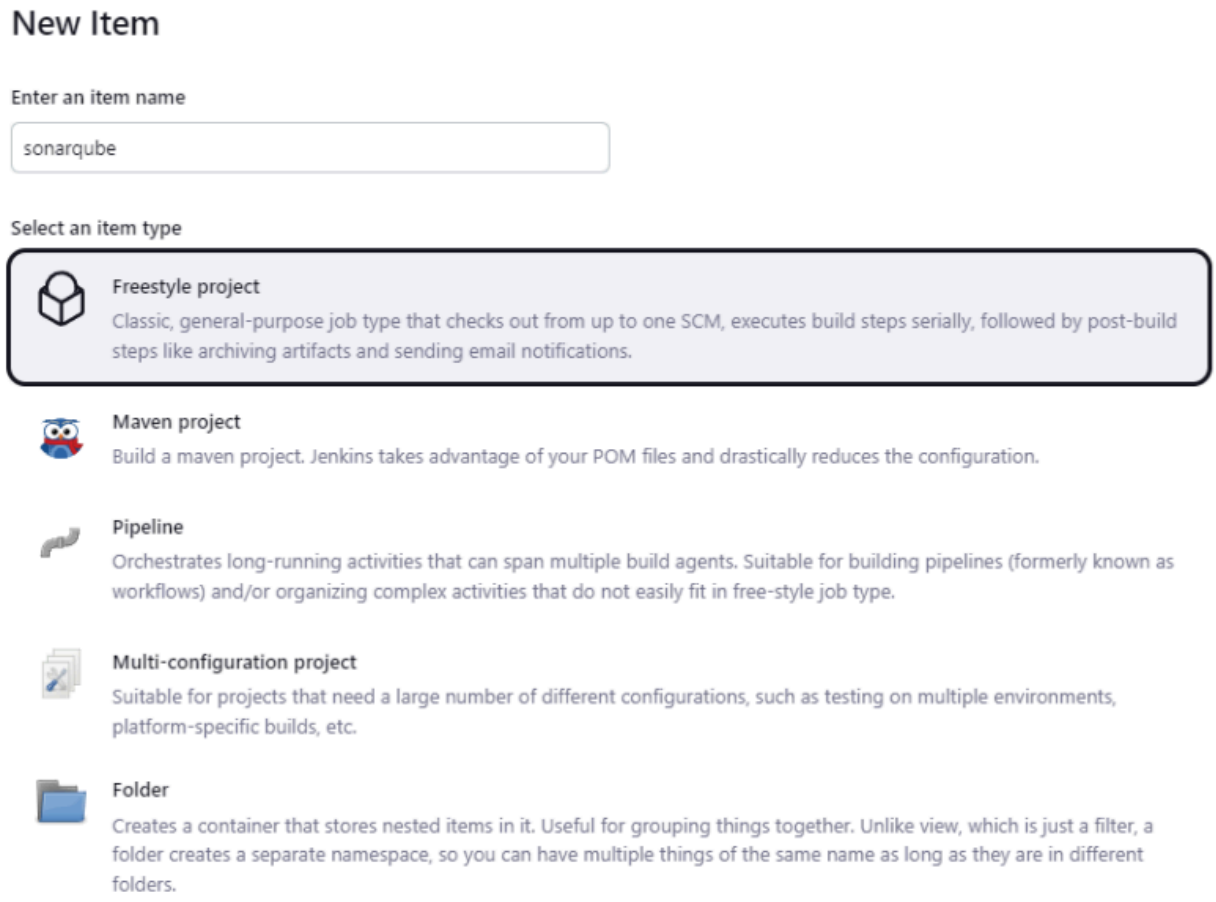
Step-6: Under 'Manage Jenkins → System', look for SonarQube Servers and enter these details. Name : sonarqube, Server URL : http://localhost:9000



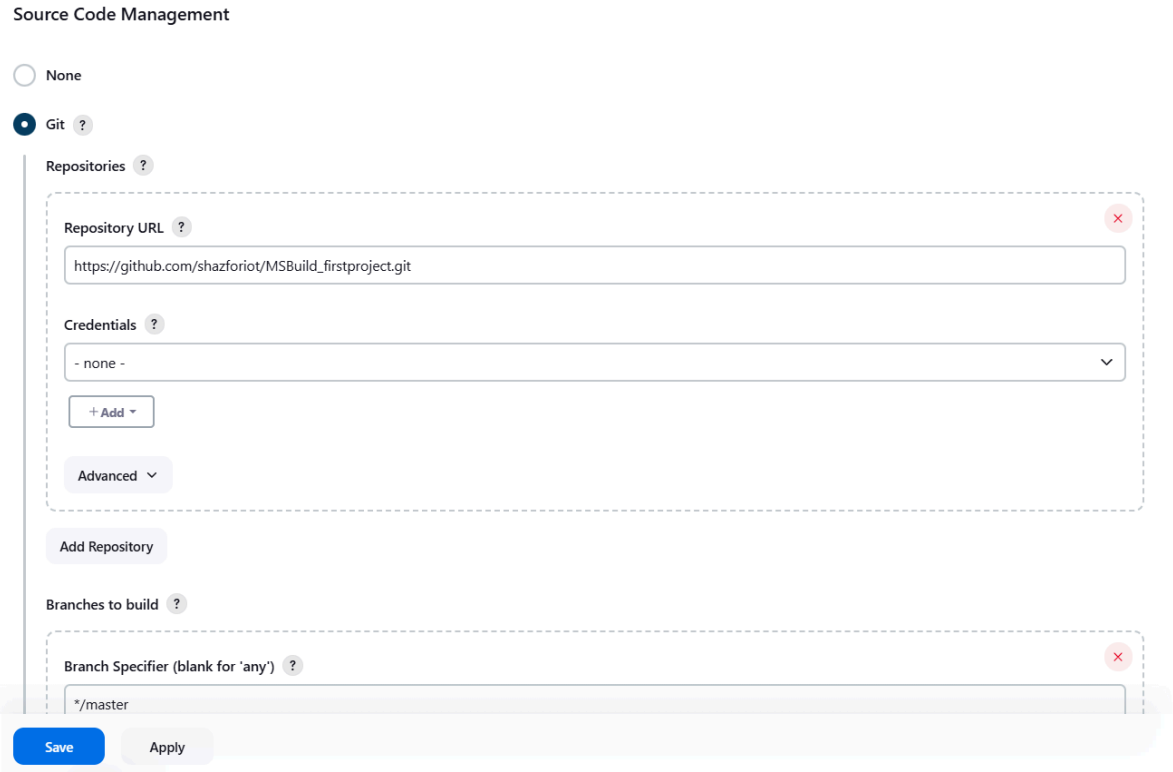
Step-7: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically. Manage Jeknins → Tools → SonarQube Scanner Installation



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

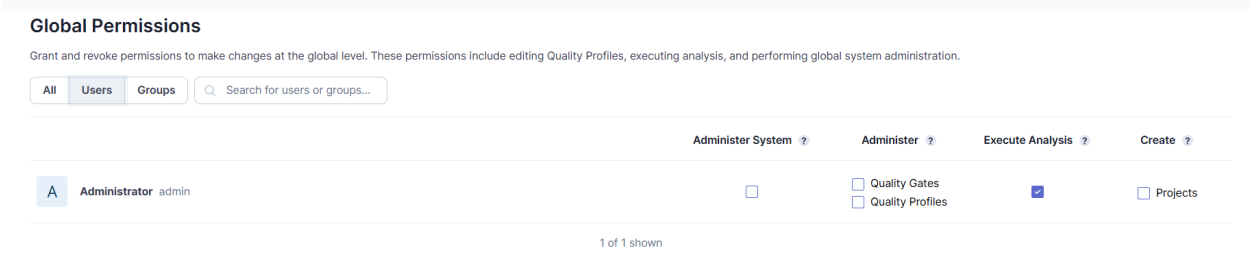


Step-9: Choose this GitHub repository in Source Code Management. https://github.com/shazforiot/MSBuild_firstproject.git . It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

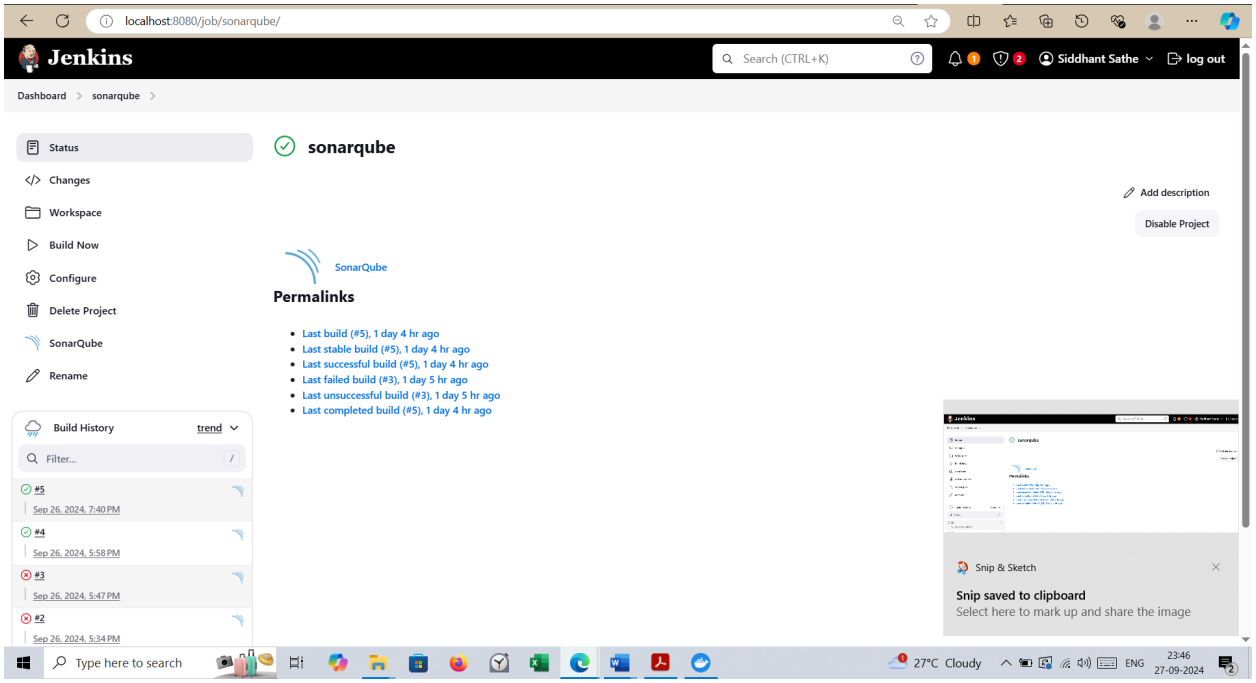


Step-10: Under Build-> Execute SonarQube Scanner, enter these Analysis Properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

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.

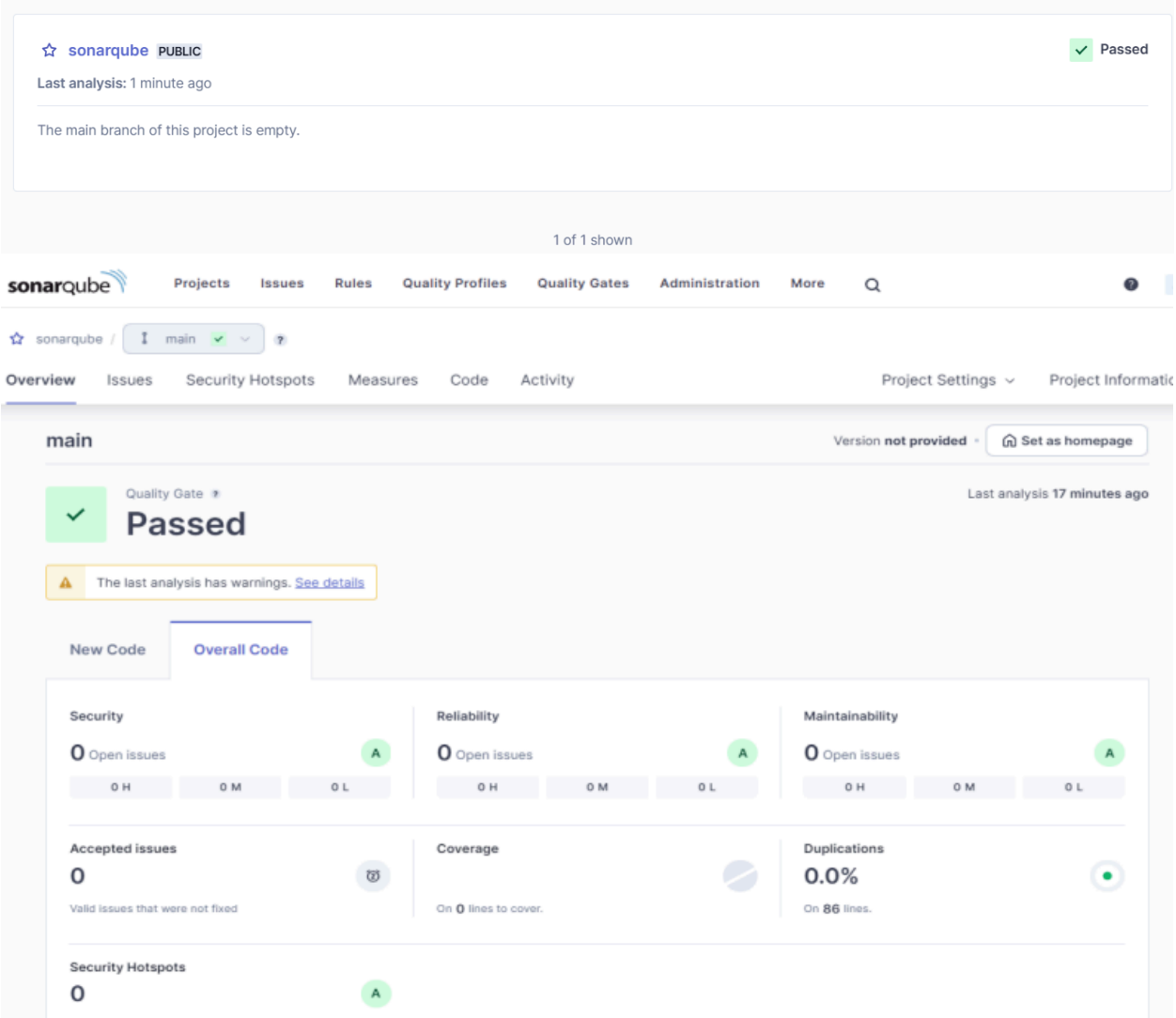


Console Output

```
Started by user Siddhant Sathe
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
[sonarqube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -
Dsonar.projectKey=sonarqube -Dsonar.login=sqp_9b85237263881f919ee5cb245bab294c2f3ce329 -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=HelloWorldCore -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
17:58:01.518 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
17:58:01.527 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-
scanner.properties
17:58:01.528 INFO  Project root configuration file: NONE
17:58:01.551 INFO  SonarScanner CLI 6.2.0.4584
17:58:01.553 INFO  Java 21.0.4 Eclipse Adoptium (64-bit)
17:58:01.558 INFO  Windows 10 10.0 amd64
17:58:01.586 INFO  User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
17:58:02.288 INFO  JRE provisioning: os[windows], arch[amd64]
17:58:02.600 INFO  Communicating with SonarQube Server 10.6.0.92116
17:58:03.043 INFO  Starting SonarScanner Engine...
17:58:03.044 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
17:58:04.214 INFO  Load global settings
17:58:04.535 INFO  Load global settings (done) | time=318ms
17:58:04.540 INFO  Server id: 147B411E-AZIk70wCd_37MaEi8bMy
```

```
17:58:37.850 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
17:58:37.851 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
17:58:37.851 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=30158bd7-ca2f-47d2-94c3-09b6c41bf32b
17:58:37.865 INFO  Analysis total time: 33.094 s
17:58:37.867 INFO  SonarScanner Engine completed successfully
17:58:37.917 INFO  EXECUTION SUCCESS
17:58:37.919 INFO  Total time: 36.394s
Finished: SUCCESS
```

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



Error:
Build failed due to missing sonar source while configurations
Overcame by adding correct path to sonar.source

Console Output

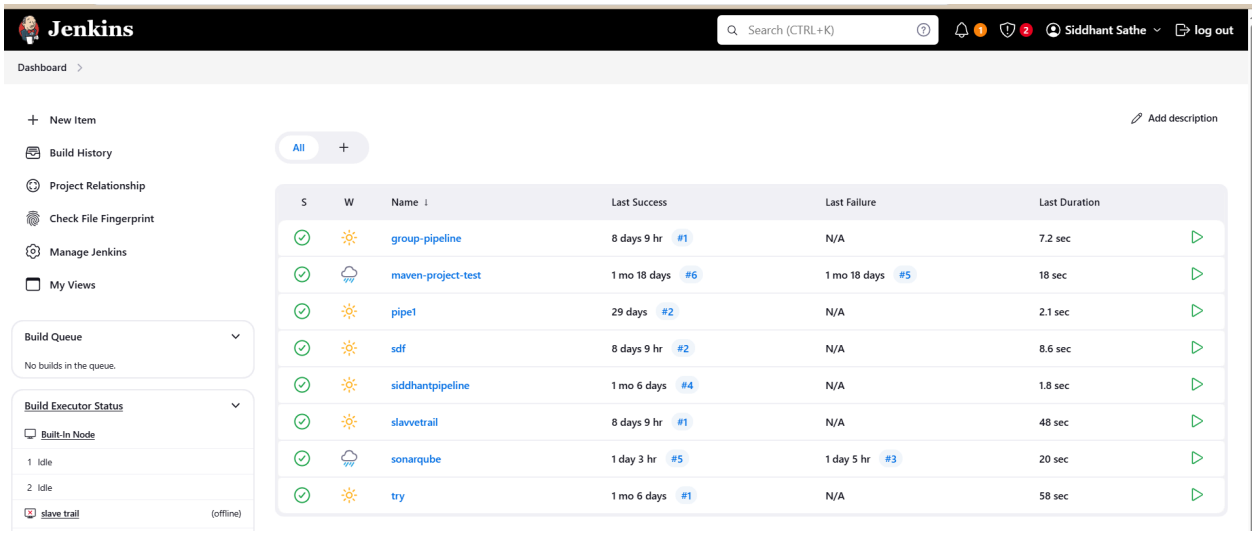
```
Started by user Siddhant Sathe
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
[sonarqube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -
Dsonar.login=sqp_9b85237263881f919ee5cb245bab294c2f3ce329 -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=sonarqube -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
17:47:09.958 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
17:47:09.970 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-scanner.properties
17:47:09.972 INFO  Project root configuration file: NONE
17:47:09.996 INFO  SonarScanner CLI 6.2.0.4584
17:47:09.998 INFO  Java 21.0.4 Eclipse Adoptium (64-bit)
17:47:10.003 INFO  Windows 10 10.0 amd64
17:47:10.039 INFO  User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
17:47:10.718 INFO  JRE provisioning: os[windows], arch[amd64]
17:47:11.098 INFO  Communicating with SonarQube Server 10.6.0.92116
17:47:11.709 INFO  Starting SonarScanner Engine...
17:47:11.710 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
17:47:13.465 INFO  Load global settings
17:47:13.665 INFO  Load global settings (done) | time=199ms
17:47:13.670 INFO  Server id: 147B411E-AZIk70wCd_37MaEi8bMy
17:47:13.688 INFO  Loading required plugins
17:47:13.688 INFO  Load plugins index
17:47:13.757 INFO  Load plugins index (done) | time=69ms
17:47:13.758 INFO  Load/download plugins
17:47:13.855 INFO  Load/download plugins (done) | time=96ms
17:47:14.525 INFO  Process project properties
17:47:14.542 ERROR Invalid value of sonar.sources for sonarqube
17:47:14.559 ERROR The folder 'sonarqube' does not exist for 'sonarqube' (base directory = C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube)
17:47:14.597 INFO  EXECUTION FAILURE
17:47:14.599 INFO  Total time: 4.632s
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
ERROR: SonarQube scanner exited with non-zero code: 1
Finished: FAILURE
```

Siddhant Sathe **D15A/51**

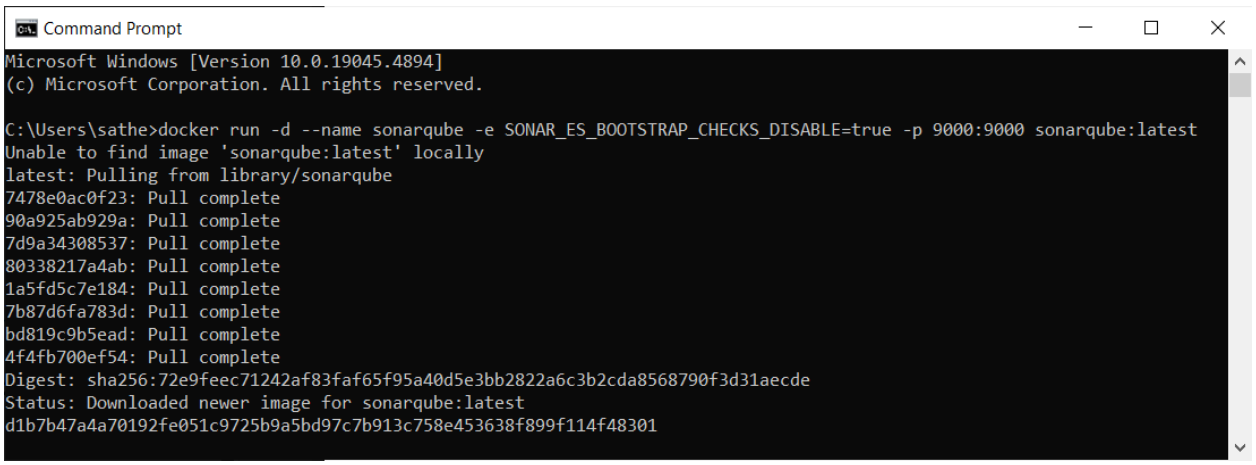
Experiment 8

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

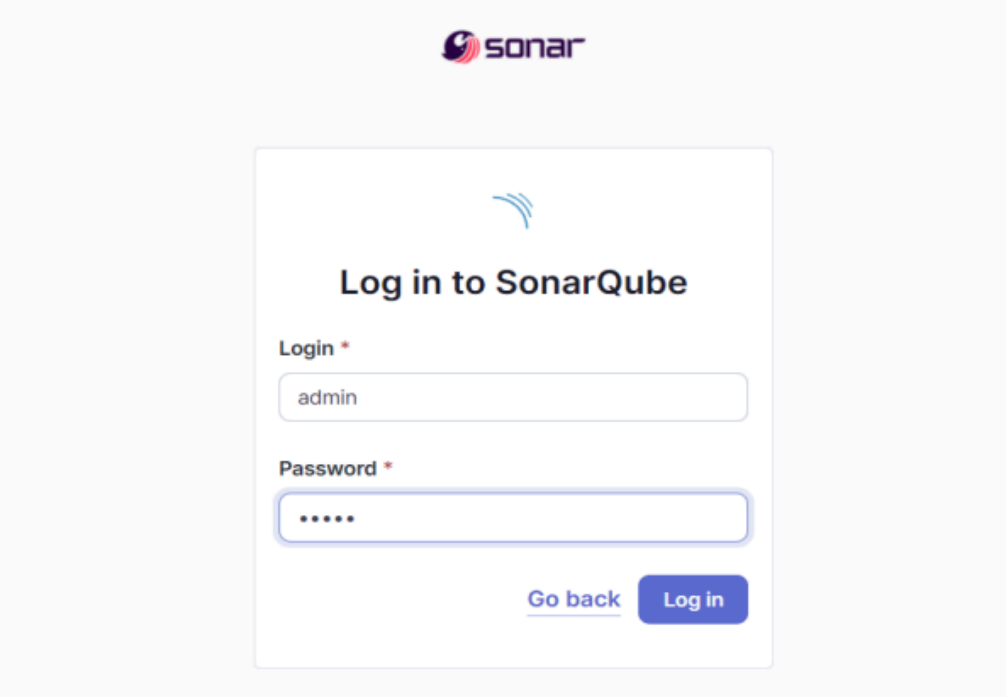
Step-1: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



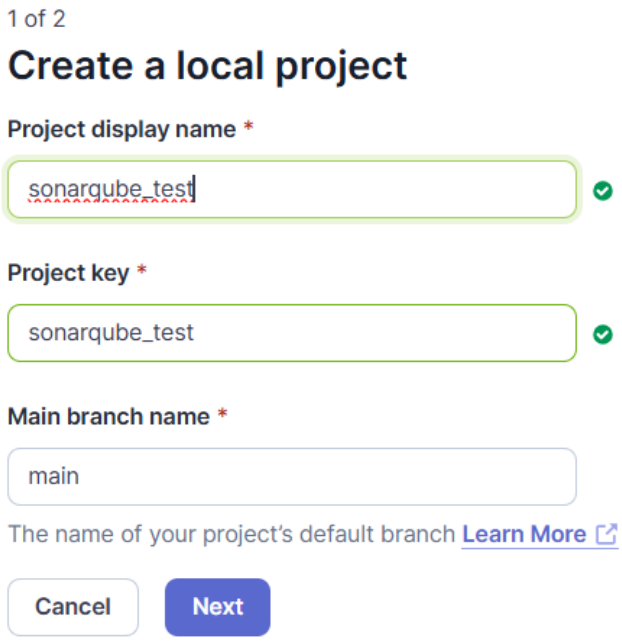
Step-2: Run SonarQube in a Docker container using this command :- a]docker -v
b] docker run -d --name sonarqube-test -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest



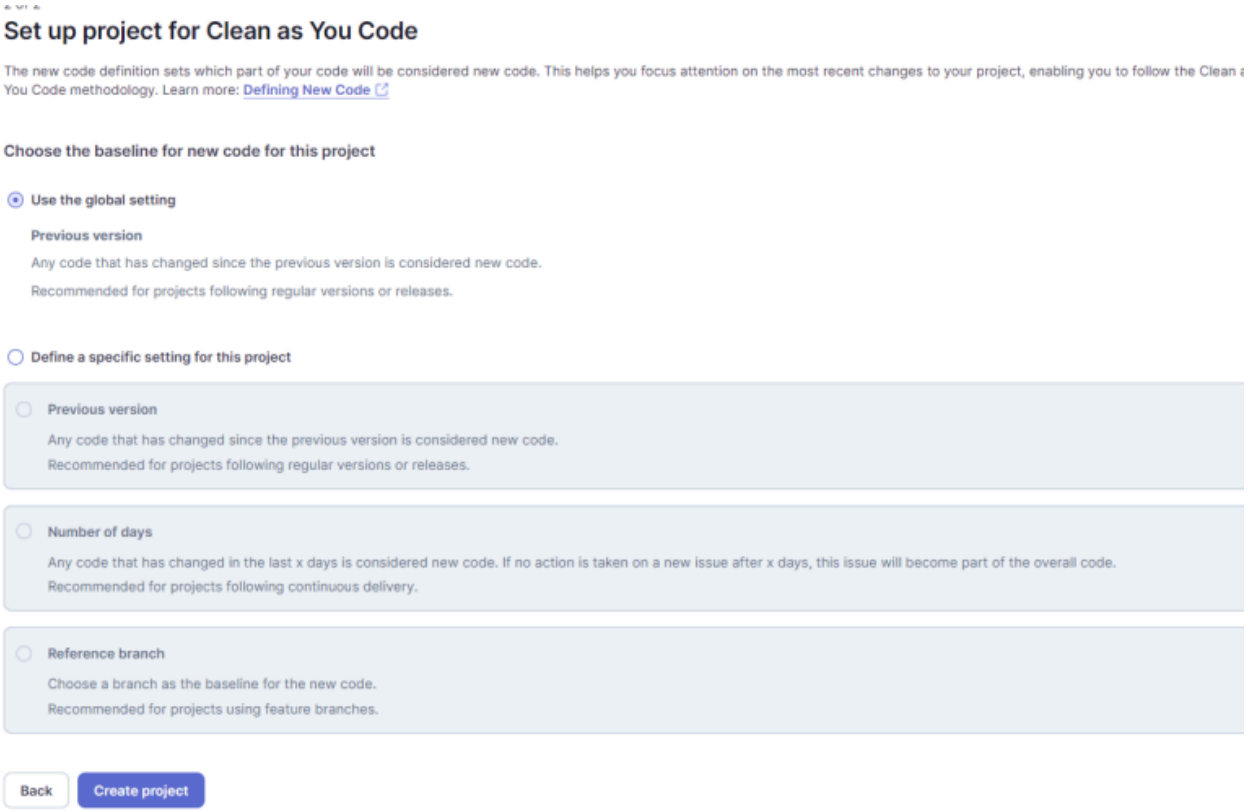
Step-3: Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



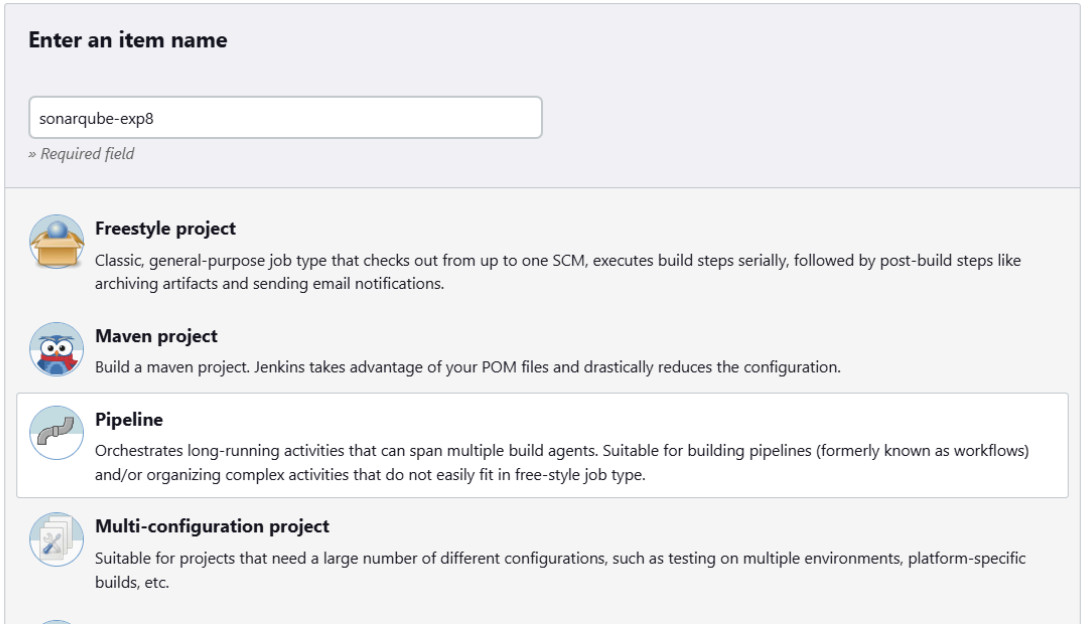
Step-4: Create a local project in SonarQube with the name sonarqube-pipeline.



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

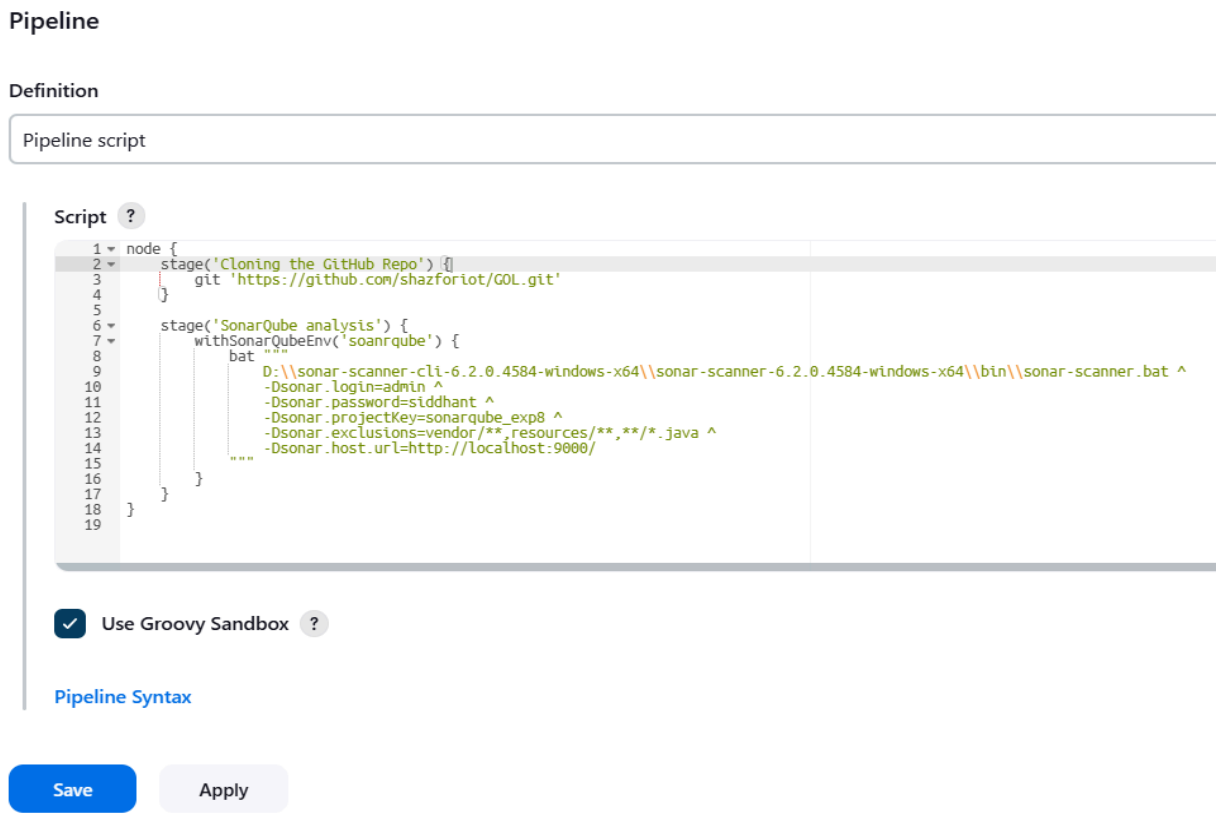


Step-6: Create a New Item in Jenkins, choose Pipeline.



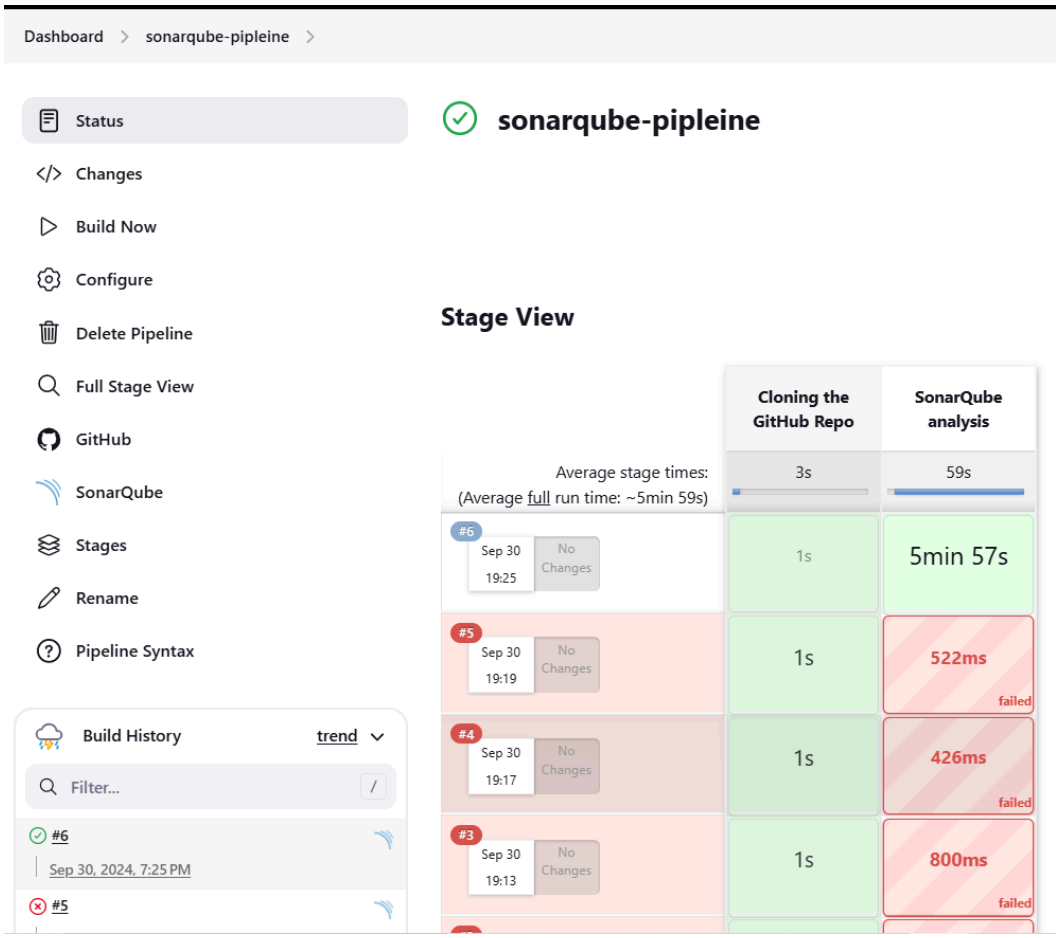
Step-7: Under Pipeline Script, enter the following -

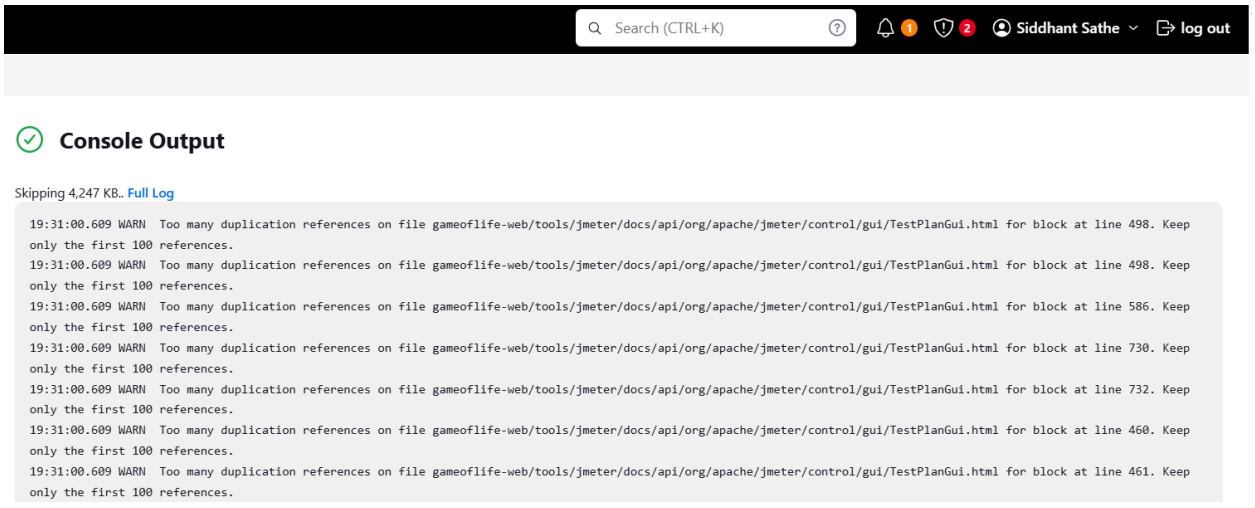
```
node {
   stage('Cloning the GitHub Repo') {
      git 'https://github.com/shazforiot/GOL.git'
   }

   stage('SonarQube analysis') {
      withSonarQubeEnv('soanrqube') {
         bat """

D:\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\so
nar-scanner.bat ^
               -Dsonar.login=admin ^
               -Dsonar.password=siddhant ^
               -Dsonar.projectKey=sonarqube_exp8 ^
               -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^
               -Dsonar.host.url=http://localhost:9000/
         """
      }
   }
}
```



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

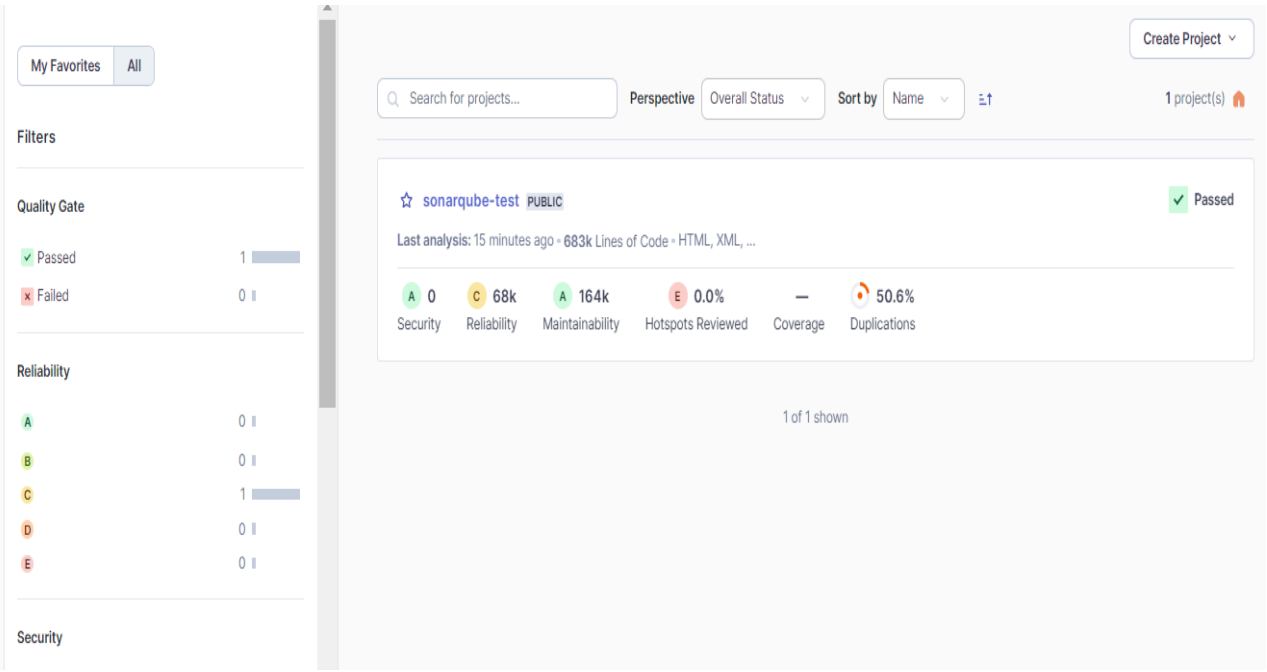
Step-8: Run The Build and check the console output:

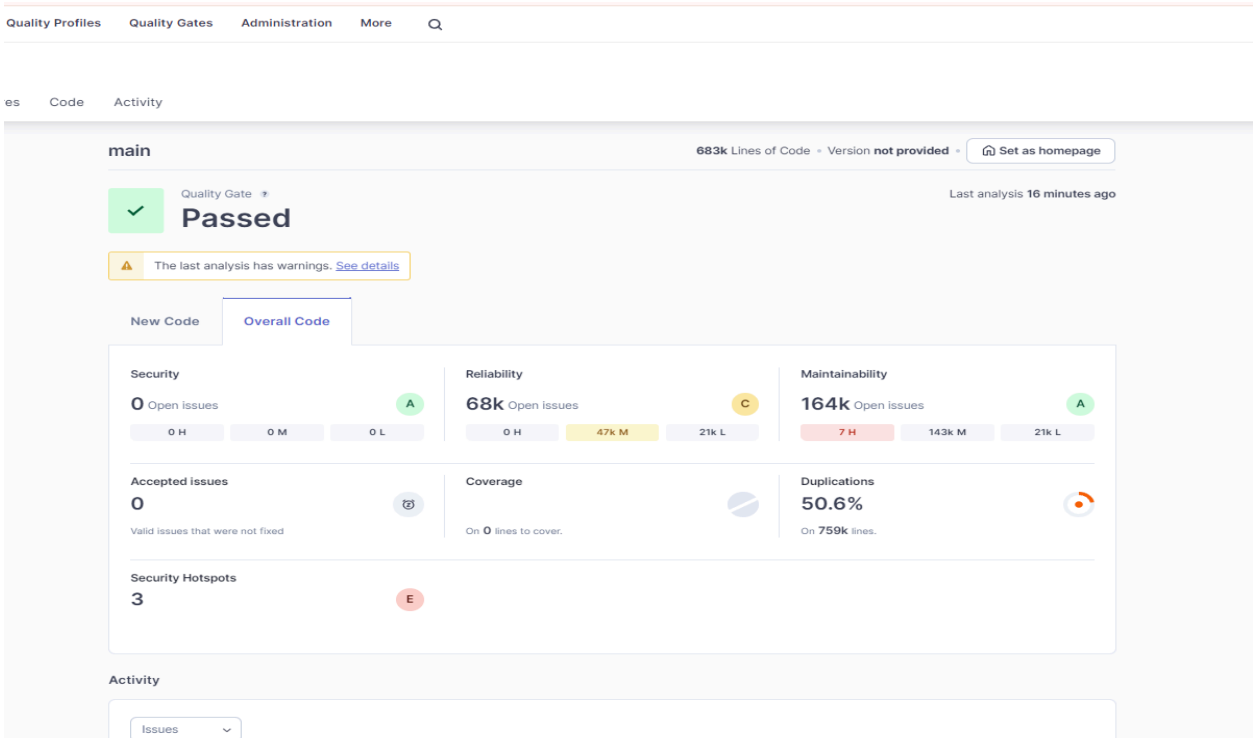




```
19:31:04.014 INFO  CPD Executor CPD calculation finished (done) | time=102876ms
19:31:04.076 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
19:31:08.393 INFO  Analysis report generated in 3362ms, dir size=127.2 MB
19:31:21.102 INFO  Analysis report compressed in 12708ms, zip size=29.6 MB
19:31:27.398 INFO  Analysis report uploaded in 6293ms
19:31:27.404 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube_exp8
19:31:27.404 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
19:31:27.404 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=c5c55adc-d9e9-41ac-89fe-132bbaf790e1
19:31:38.725 INFO  Analysis total time: 5:45.452 s
19:31:38.739 INFO  SonarScanner Engine completed successfully
19:31:39.402 INFO  EXECUTION SUCCESS
19:31:39.803 INFO  Total time: 5:52.770s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

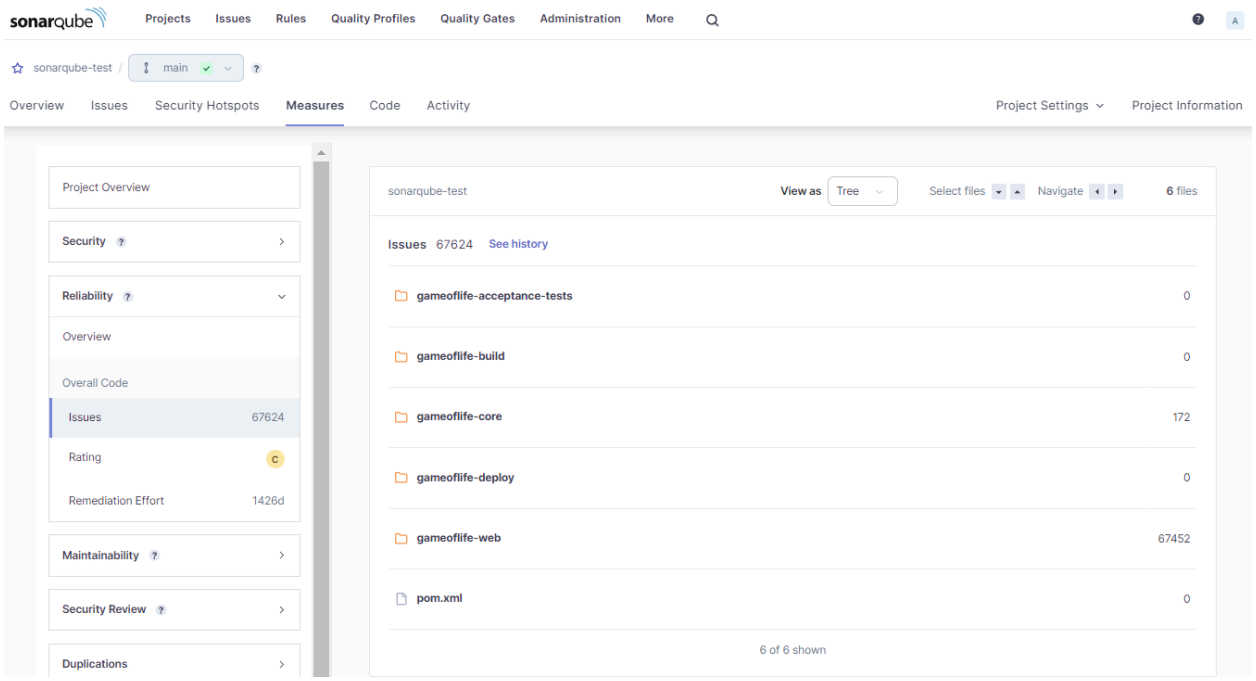
Step-9: After that, check the project in SonarQube.



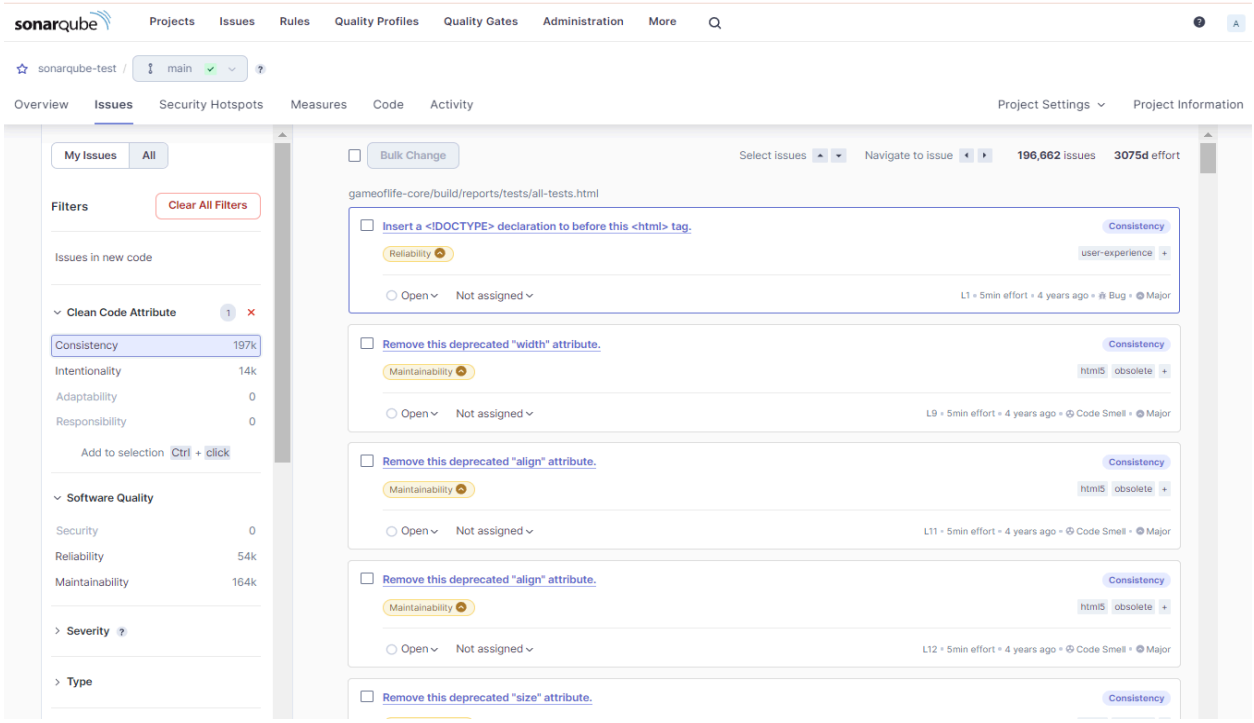


Step-10: Under different tabs, check all different issues with the code.
Code Problems

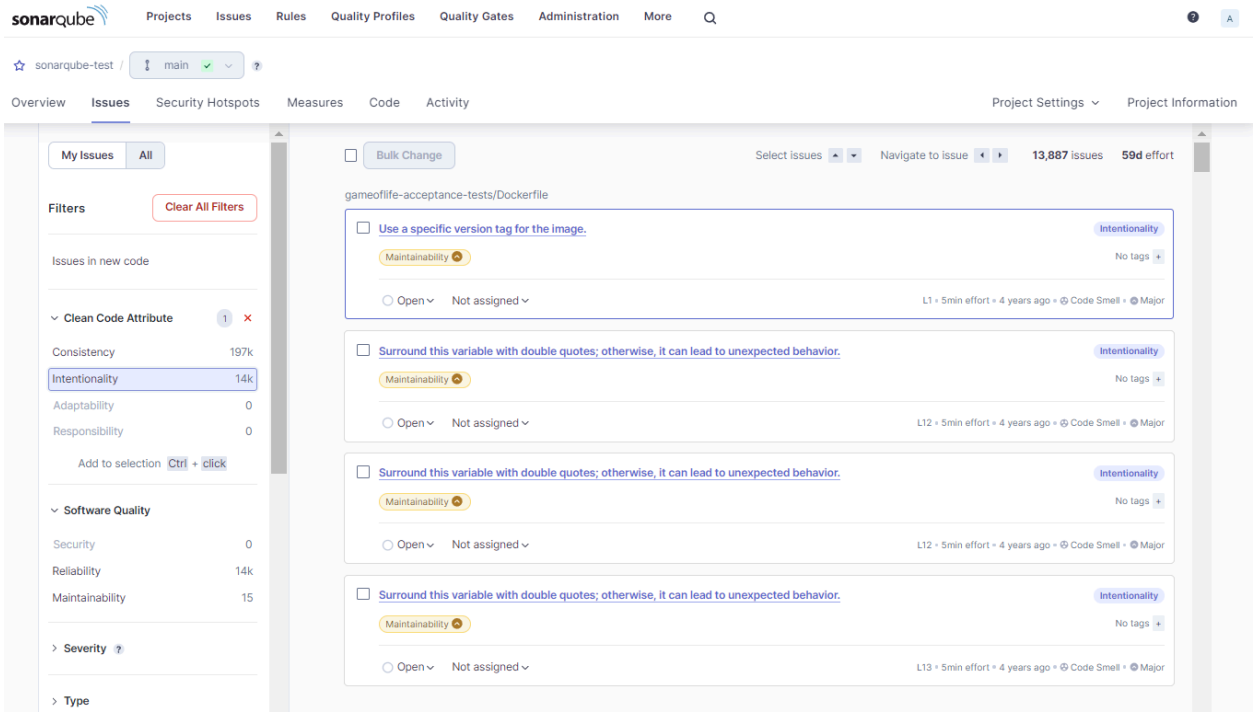
Code issues:



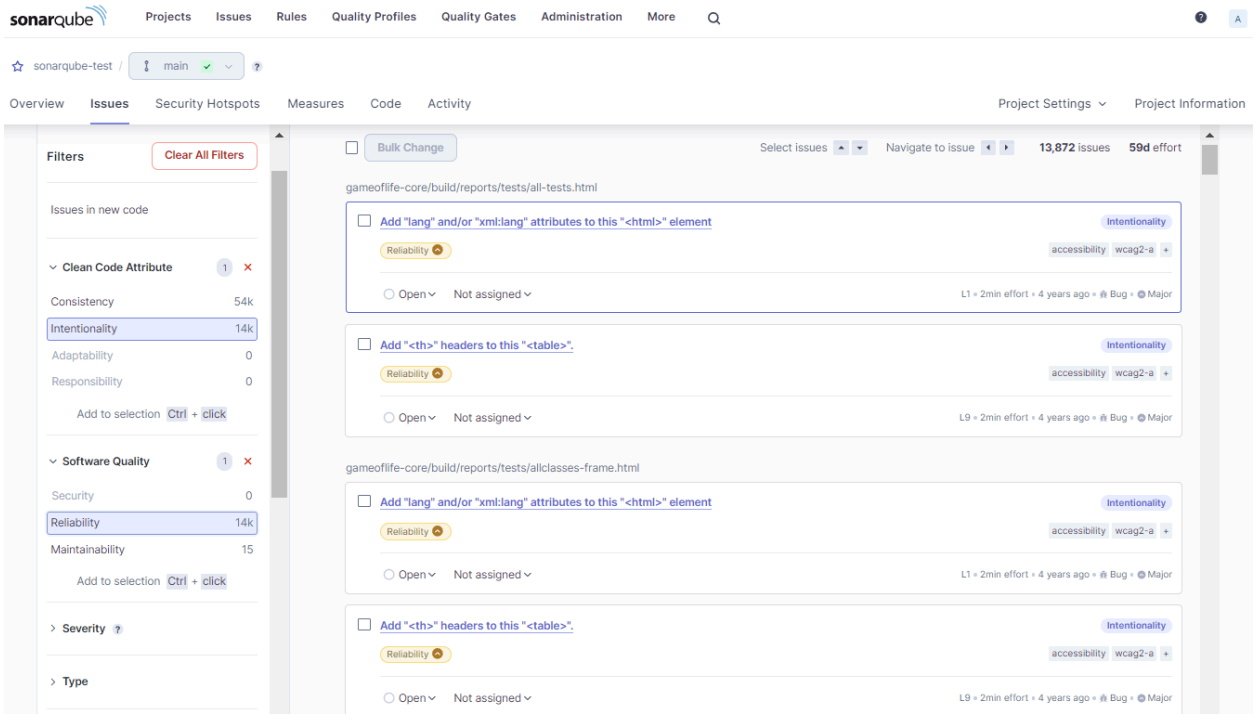
Consistency:



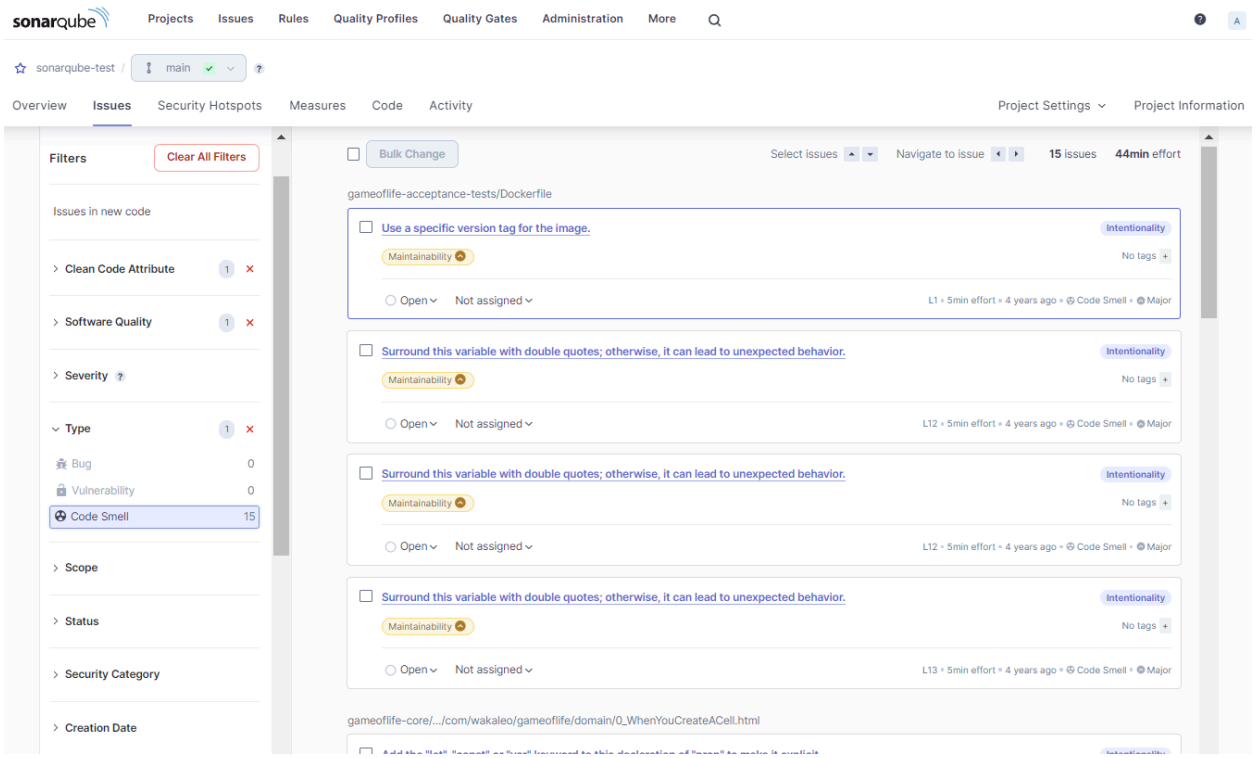
Intentionally:



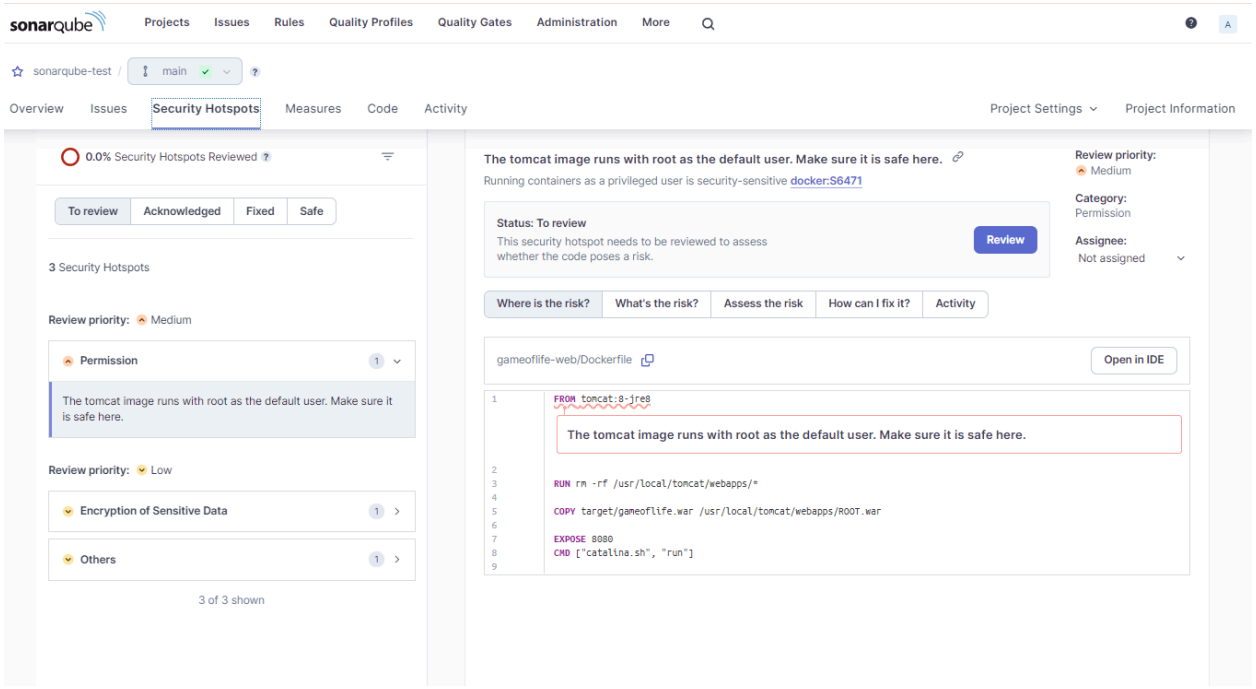
Reliability:



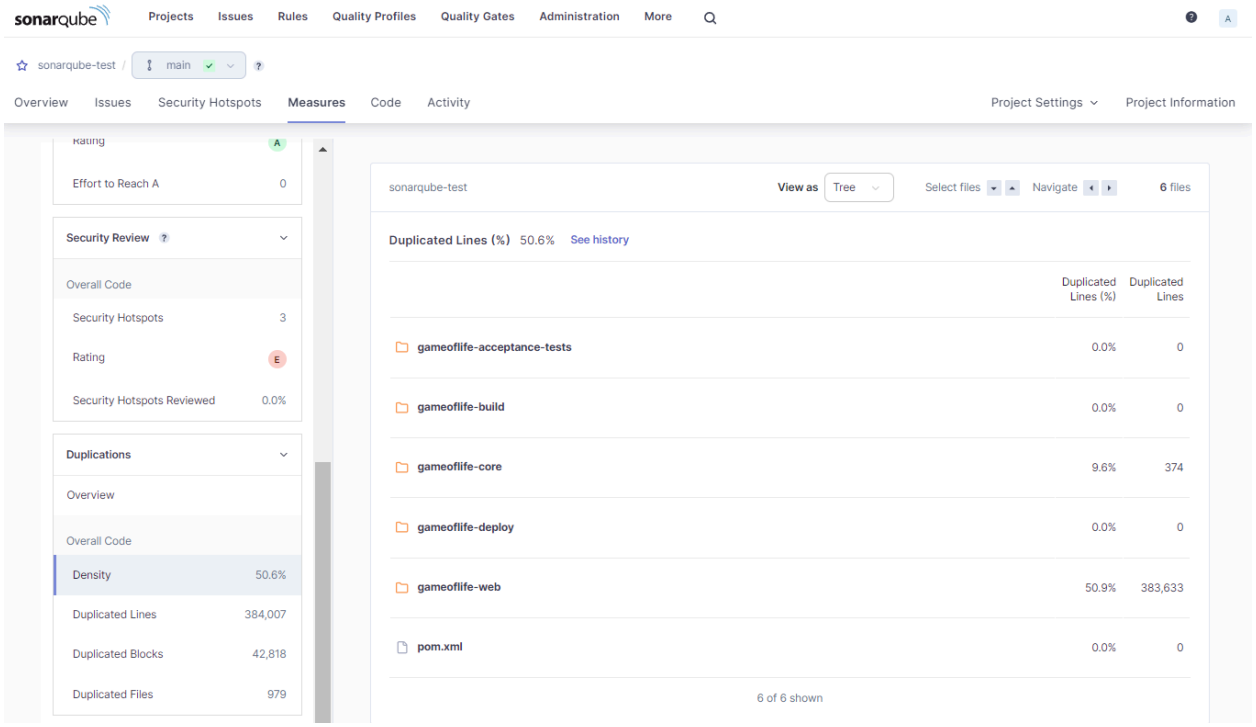
Code smells:



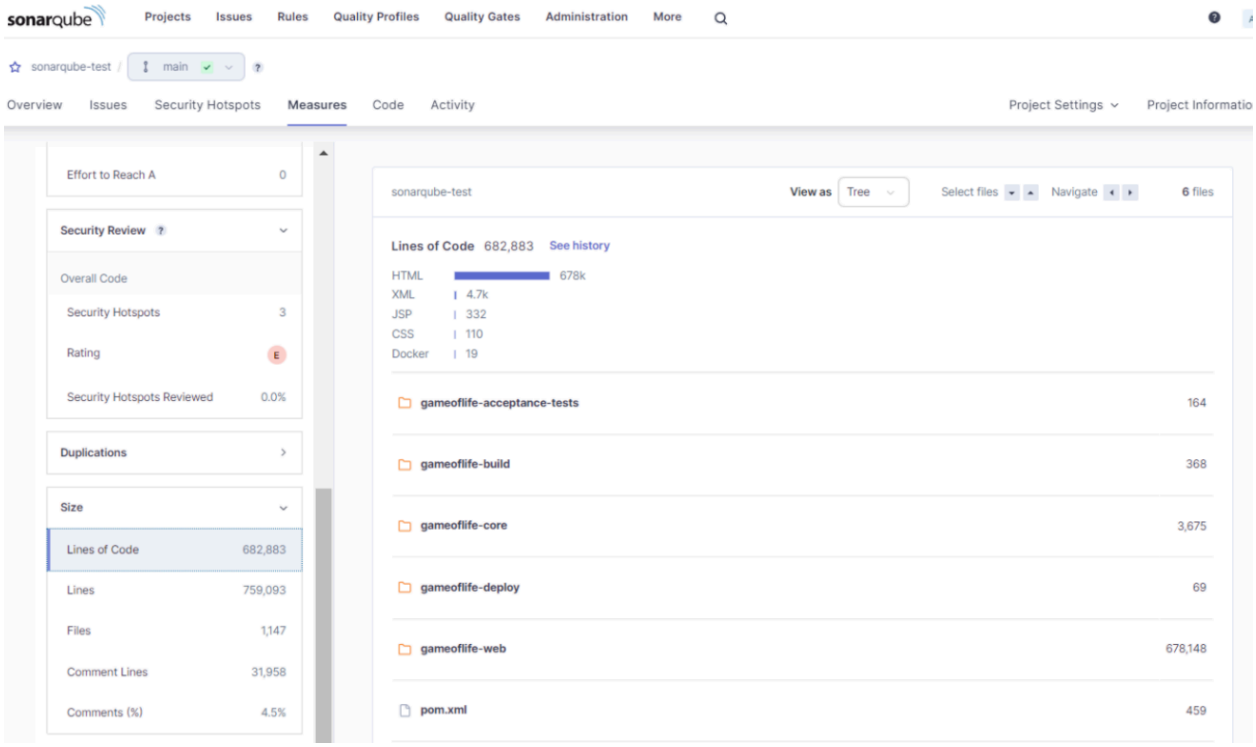
Security hotspot:



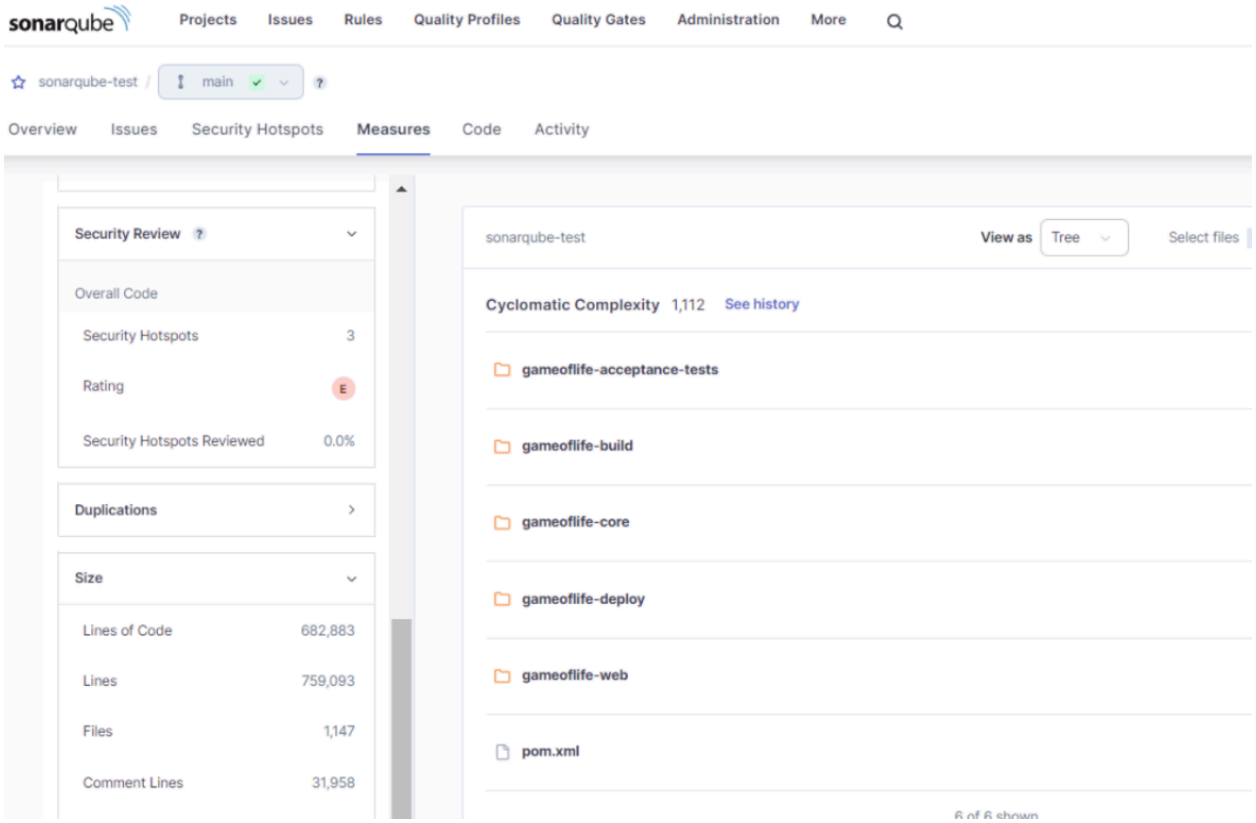
Duplicates:



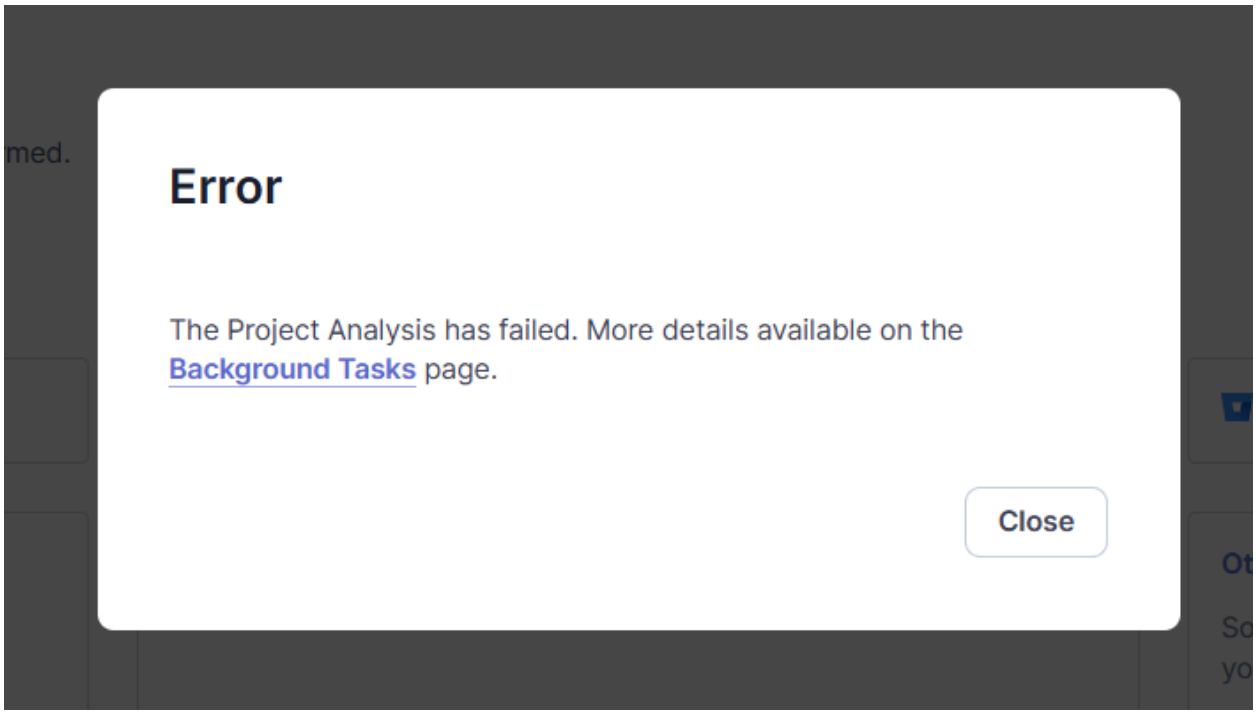
Size:



Complexity:



Errors:
Build was not successful because my sonarqube env name was wrong in scripts



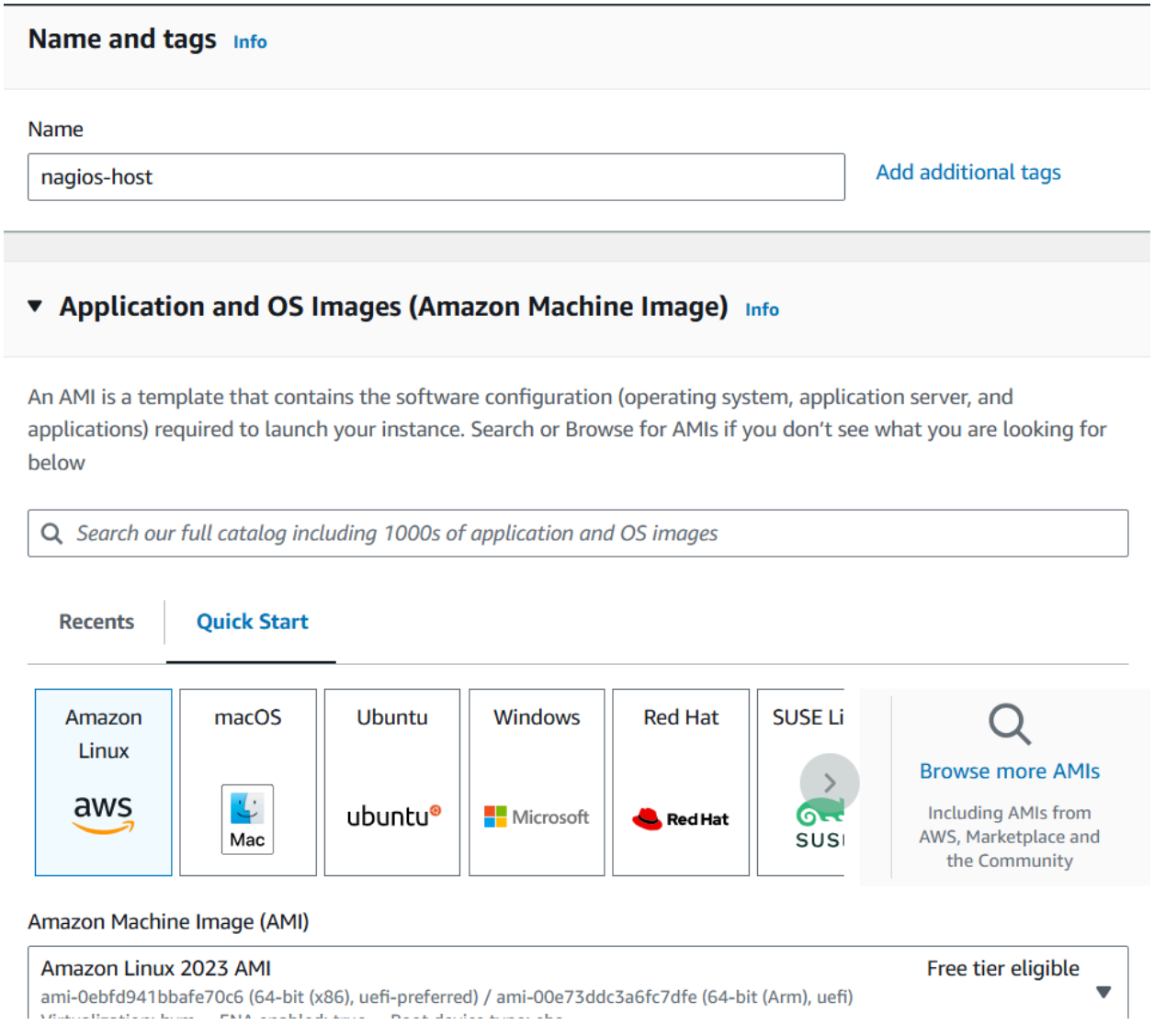
Siddhant Sathe **D15A/51**

EXPERIMENT 9

User added: nagios
New passwd: nagios

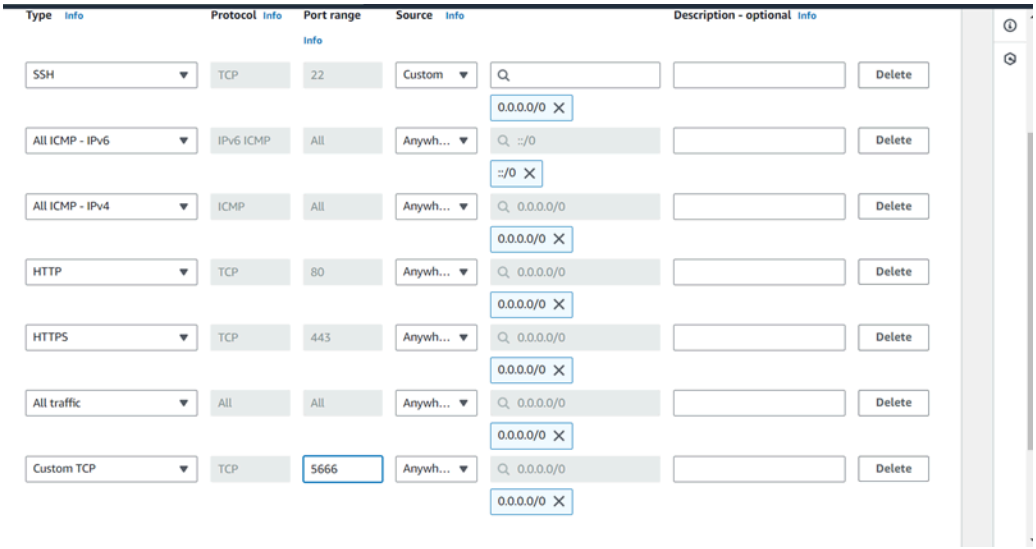
Step 1. Create an Amazon Linux EC2 Instance

- Name it `nagios-host`.



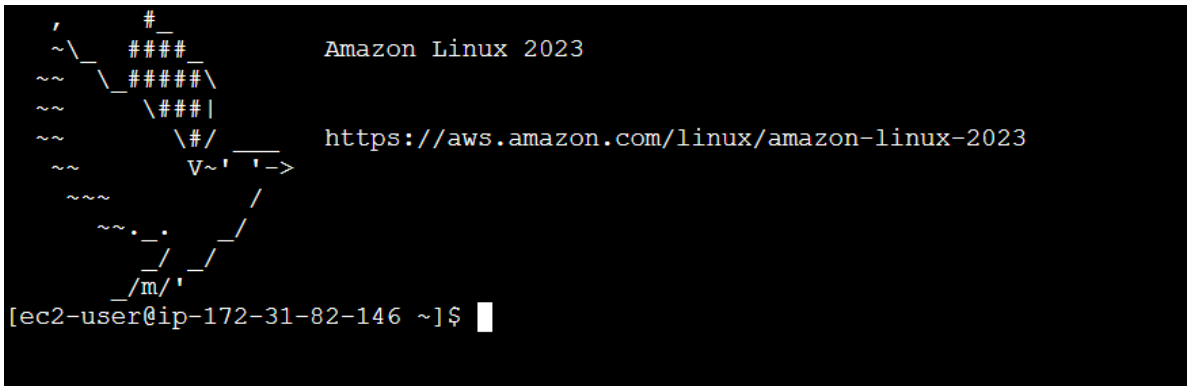
Step 2. Configure Security Group

- Ensure HTTP, HTTPS, SSH, and ICMP are open from everywhere.
- Edit the inbound rules of the specified Security Group



Step 3. Connect to Your EC2 Instance

- SSH into your EC2 instance or use EC2 Instance Connect from the browser



Step 4. Update Package Indices and Install Required Packages

Commands -

sudo yum update

sudo yum install httpd php

sudo yum install gcc glibc glibc-common

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-82-146 ~]$ ^[[200~sudo yum update
-bash: $'\E[200~sudo': command not found
[ec2-user@ip-172-31-82-146 ~]$ sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
^[[201~Last metadata expiration check: 0:06:52 ago on Tue Oct  1 08:39:57 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository
Installing:			
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux
php8.3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux
Installing dependencies:			
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux
libxslt	x86_64	1.1.34-5.amzn2023.0.2	amazonlinux
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.4	amazonlinux
php8.3-cli	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux
php8.3-common	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux
php8.3-process	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux
php8.3-xml	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux
Installing weak dependencies:			
pcre2-devel	x86_64	10.40-1.amzn2023.0.3	amazonlinux
pcre2-utf16	x86_64	10.40-1.amzn2023.0.3	amazonlinux
pcre2-utf32	x86_64	10.40-1.amzn2023.0.3	amazonlinux
pixman	x86_64	0.40.0-3.amzn2023.0.3	amazonlinux
sysprof-capture-devel	x86_64	3.40.1-2.amzn2023.0.2	amazonlinux
xml-common	noarch	0.6.3-56.amzn2023.0.2	amazonlinux
xorg-x11-proto-devel	noarch	2021.4-1.amzn2023.0.2	amazonlinux
xz-devel	x86_64	5.2.5-9.amzn2023.0.2	amazonlinux
zlib-devel	x86_64	1.2.11-33.amzn2023.0.5	amazonlinux

```
Transaction Summary
Install  62 Packages

Total download size: 23 M
Installed size: 87 M
Is this ok [y/N]: y
Downloading Packages:
(1/62): brotli-devel-1.0.9-4.amzn2023.0.2.x86_64.rpm                561 kB/s |  31 kB     00:00
(2/62): brotli-1.0.9-4.amzn2023.0.2.x86_64.rpm                      4.2 MB/s | 314 kB     00:00
(3/62): bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64.rpm                 2.2 MB/s | 214 kB     00:00
(4/62): cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64.rpm           654 kB/s |  16 kB     00:00
(5/62): fontconfig-2.13.94-2.amzn2023.0.2.x86_64.rpm                9.8 MB/s | 273 kB     00:00
(6/62): cairo-1.17.6-2.amzn2023.0.1.x86_64.rpm                      7.7 MB/s | 684 kB     00:00
(7/62): fonts-filesystem-2.0.5-12.amzn2023.0.2.noarch.rpm           408 kB/s | 9.5 kB     00:00
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64                pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
  pixman-0.40.0-3.amzn2023.0.3.x86_64                    sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch                xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                   zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
```

Step 5. Create a New Nagios User

Commands -

sudo adduser -m nagios

sudo passwd nagios

```
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

Step 6. Create a New User Group

Commands -

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-82-146 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-82-146 ~]$
```

Step 7. Add Users to the Group

Commands -

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-82-146 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

Step 8. Create a Directory for Nagios Downloads

Commands -

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-80-22 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-80-22 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
```

Step 9. Download Nagios and Plugins Source Files

Commands -

wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz

wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz

```
[ec2-user@ip-172-31-82-146 downloads]$ Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
-bash: Wget: command not found
--2024-10-01 09:03:42--  https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'

nagios-plugins-2.3.3.tar.gz             100%[=========================================================================>]   2.65M  6.61MB/s    in 0.4s

2024-10-01 09:03:42 (6.61 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]
```

```
[ec2-user@ip-172-31-82-146 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
--2024-10-01 09:18:08--  https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz.1'

nagios-plugins-2.3.3.tar.gz.1           100%[=========================================================================>]   2.65M  6.81MB/s    in 0.4s
```

Step 10. Extract the Nagios Source File

Commands -

tar zxvf nagios-4.4.6.tar.gz

cd nagios-4.4.6

```
[ec2-user@ip-172-31-82-146 downloads]$ tar zxvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
```

```
nagios-4.4.6/xdata/xrddefault.c
nagios-4.4.6/xdata/xrddefault.h
nagios-4.4.6/xdata/xsddefault.c
nagios-4.4.6/xdata/xsddefault.h
```

```
[ec2-user@ip-172-31-82-146 downloads]$ cd nagios-4.4.6
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ ./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... no
checking for cc... no
checking for cl.exe... no
configure: error: in `/home/ec2-user/downloads/nagios-4.4.6':
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details
```

Step 11. Run the Configuration Script

Commands -

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

```
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ ./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 we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
```

Step 12. Compile Source Code

Commands -

make all

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
    inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |                 log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o commands.o commands.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
```

```
*** 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 13. Install Binaries, Init Script, and Sample Config Files

Commands -

./sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ ./sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
-bash: ./sudo: No such file or directory
/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
/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 ***
```

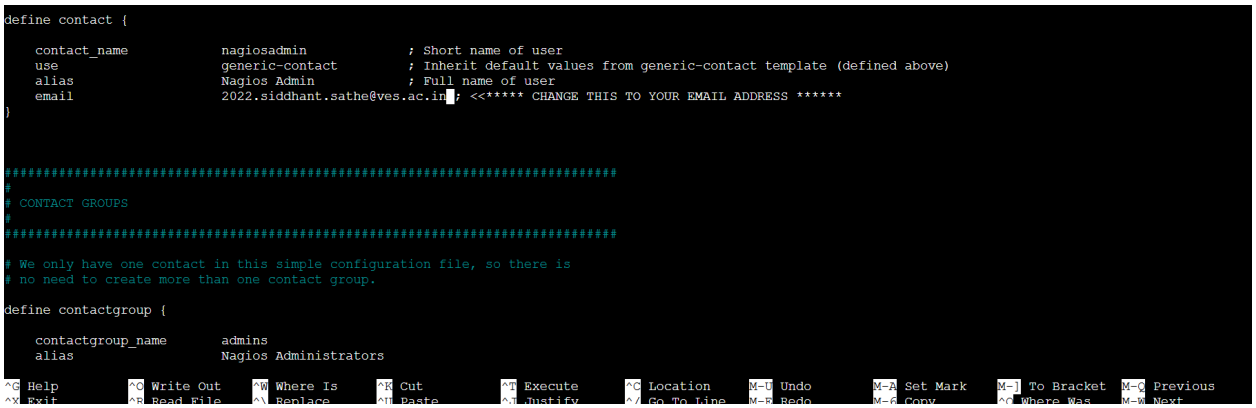
```
Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

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

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

Step 14. Edit the Config File to Change the Email Address

Commands -

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

Change the email address in the `contacts.cfg` file to your preferred email.



Step 15. Configure the Web Interface

Commands -

sudo make install-webconf

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

*** External command directory configured ***

[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

Step 16. Create a Nagios Admin Account

Commands -

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

- You will be prompted to enter and confirm the password for the `nagiosadmin` user.(password is nagios)

```
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Step 17. Restart Apache

Commands - sudo systemctl restart httpd

```
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ sudo systemctl restart httpd
```

Step 18. Extract the Plugins Source File

Commands -

cd ~/downloads

tar zxvf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

```
[ec2-user@ip-172-31-82-146 nagios-4.4.6]$ cd ~/downloads
tar zxvf nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/Class-Accessor-0.34.tar.gz
```

Step 19. Compile and Install Plugins

Commands -

./configure --with-nagios-user=nagios --with-nagios-group=nagios

make

sudo make install

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
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 to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
```

Step 20. Start Nagios

Commands -

sudo chkconfig --add nagios

sudo chkconfig nagios on

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

sudo systemctl start nagios

```
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
```

Step 21. Check the Status of Nagios

Commands -

sudo systemctl status nagios

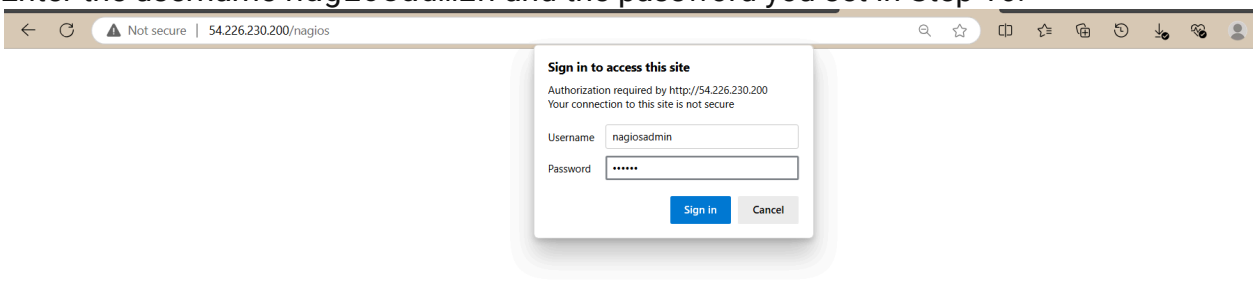
```
[ec2-user@ip-172-31-40-11 nagios-plugins-2.3.3]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Thu 2024-10-03 04:01:15 UTC; 1min 18s ago
       Docs: https://www.nagios.org/documentation
    Process: 68368 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 68369 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 68370 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.3M
        CPU: 29ms
     CGroup: /system.slice/nagios.service
             ├─68370 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─68371 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68372 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68373 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68374 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─68375 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: qh: core query handler registered
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: qh: echo service query handler registered
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: qh: help for the query handler registered
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: wproc: Successfully registered manager as @wproc with query handler
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: wproc: Registry request: name=Core Worker 68374;pid=68374
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: wproc: Registry request: name=Core Worker 68373;pid=68373
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: wproc: Registry request: name=Core Worker 68371;pid=68371
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: wproc: Registry request: name=Core Worker 68372;pid=68372
Oct 03 04:01:15 ip-172-31-40-11.ec2.internal nagios[68370]: Successfully launched command file worker with pid 68375
```

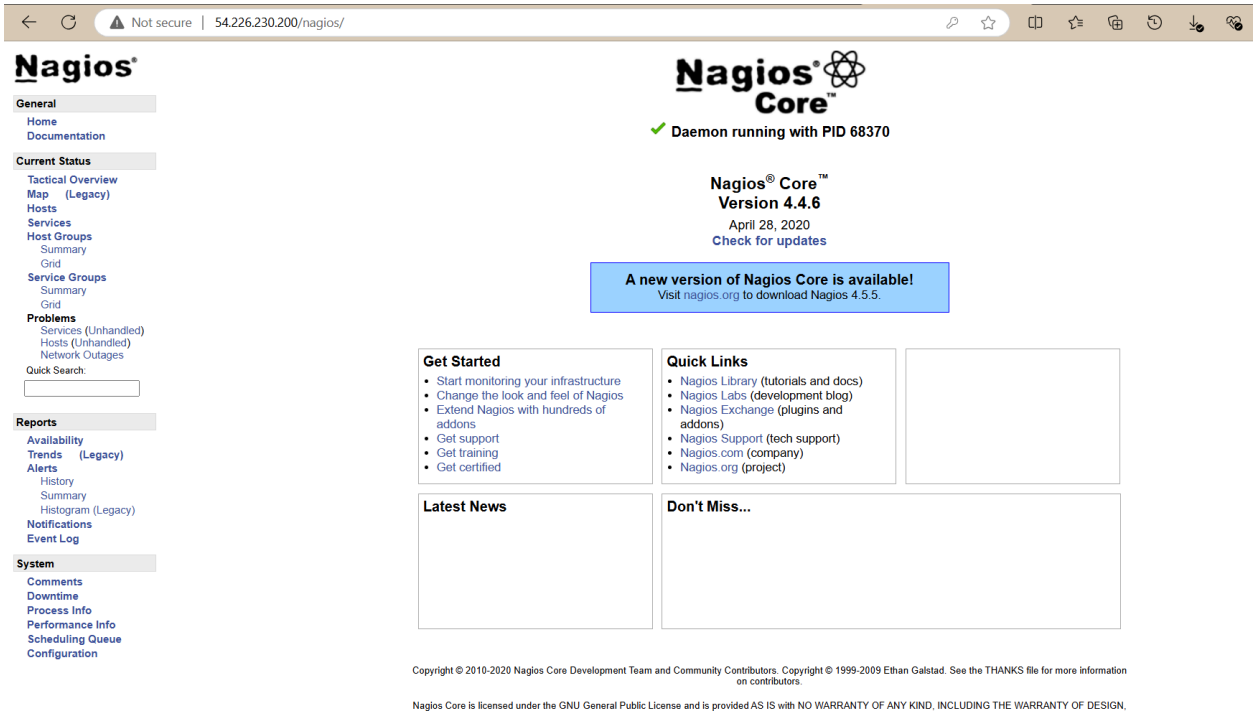
Step 22. Access Nagios Web Interface

- Copy the Public IP address of your EC2 instance.

- Open your browser and navigate to `http://<your_public_ip_address>/nagios`.

Enter the username `nagiosadmin` and the password you set in Step 16.





Error:
This error arises because nagios 4.4.6 was not found. It was arised due to sudo yum install gcc glibc glibc-common was not executed successfully.

```
[ec2-user@ip-172-31-82-146 downloads]$ tar zxvf nagios-4.4.6.tar.gz
cd nagios-4.4.6
tar (child): nagios-4.4.6.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
-bash: cd: nagios-4.4.6: No such file or directory
[ec2-user@ip-172-31-82-146 downloads]$
```

Siddhant Sathe **D15A/51**

EXPERIMENT 10

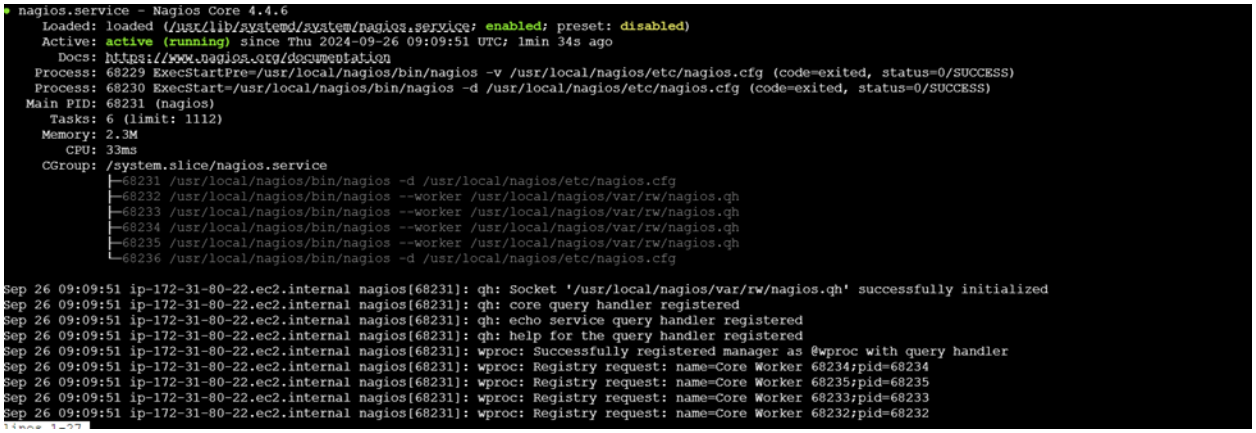
Aim: Exp 10 To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

1. Confirm Nagios is Running on the Server

Commands -

sudo systemctl status nagios

- Proceed if you see that Nagios is active and running.

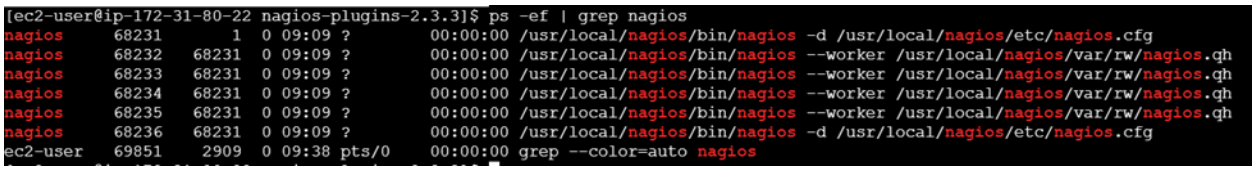


2. Create an Ubuntu 20.04 Server EC2 Instance

- Name it `linux-client`.
- Use the same security group as the Nagios Host.

3. Verify Nagios Process on the Server

Commands - ps -ef | grep nagios



4. Become Root User and Create Directories

Commands -

sudo su

mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

5. Copy Sample Configuration File

Commands -

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

```
[root@ip-172-31-80-22 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-80-22 ec2-user]# sudo nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

6. Edit the Configuration File

Commands -
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`.

```
# 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                 3.85.25.81
}
```

- Change `hostgroup_name` under hostgroup to `linux-servers1`

```
###############################################################################
#
# 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                 linuxserver             ; Comma separated list of hosts that belong to this group
}
```

7. Update Nagios Configuration

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

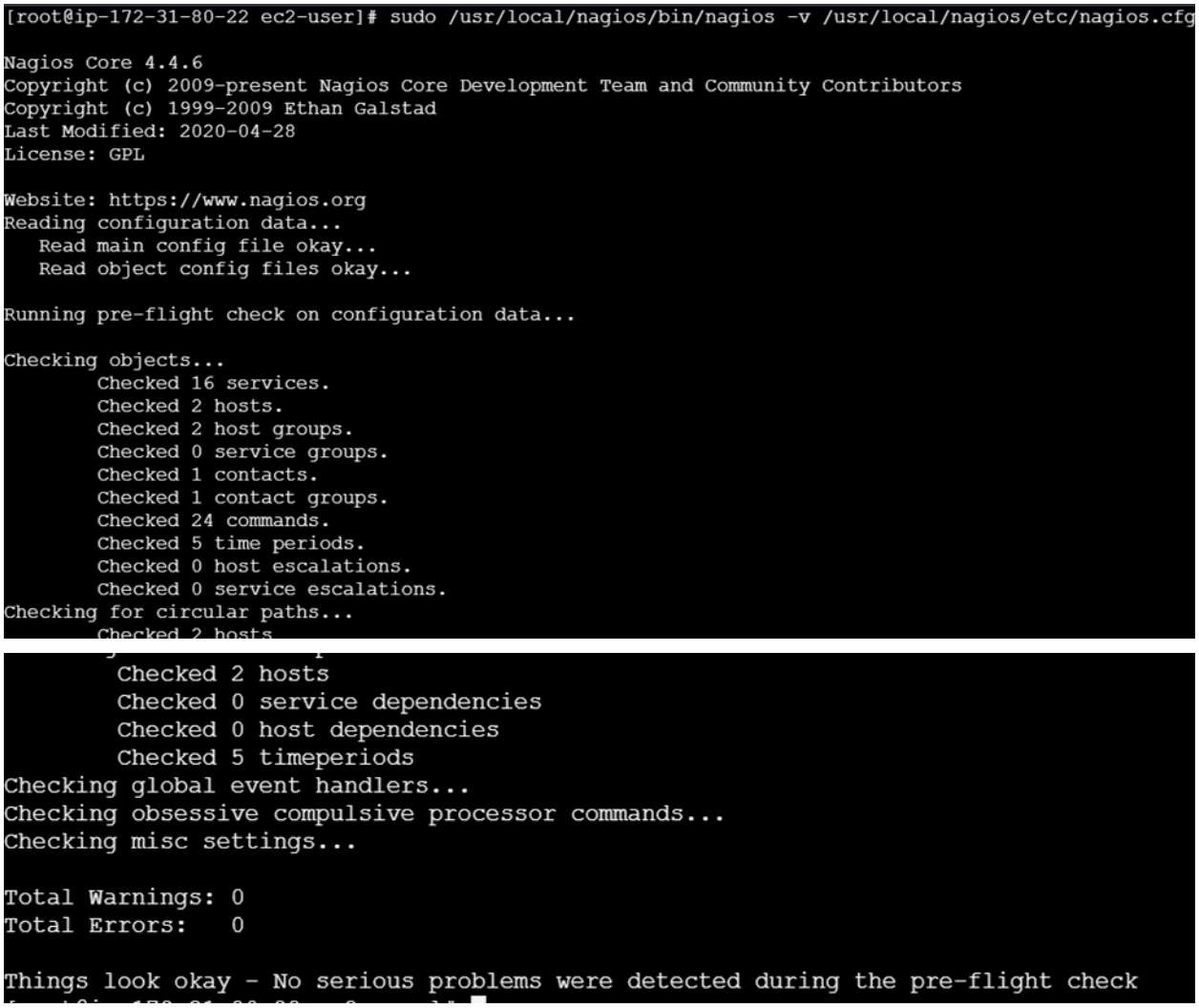
- Add the following line:
  cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
# 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/
```

8. Verify Configuration Files

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

- Ensure there are no errors.



9. Restart Nagios Service

Commands - sudo systemctl restart nagios

10. SSH into the Client Machine

- Use SSH or EC2 Instance Connect to access the `linux-client`.

11. Update Package Index and Install Required Packages

Commands -

sudo apt update -y

sudo apt install gcc -y

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

```
ubuntu@ip-172-31-88-112:~$ 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://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [378 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.0 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4548 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [271 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
```

12. 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>`

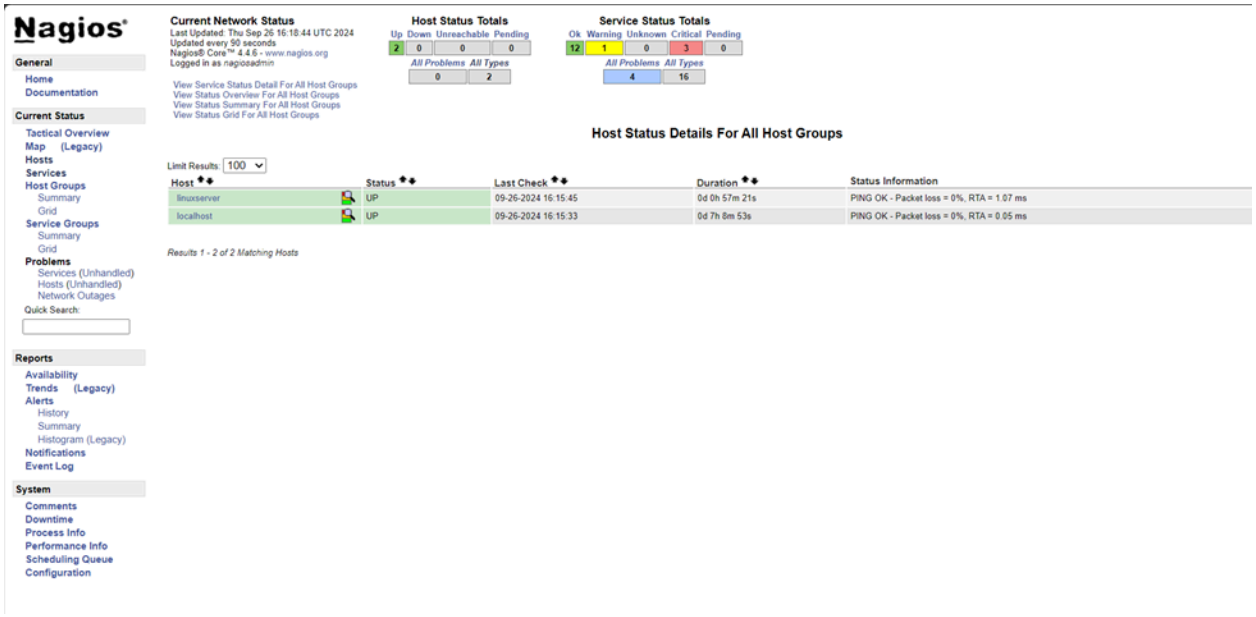
```
  GNU nano 7.2                                                    /etc/nagios/nr
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,18.208.138.41



# 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.
#
```

13. Restart NRPE Server

Commands - sudo systemctl restart nagios-nrpe-server

14. 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
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Current Network Status
Last Updated: Thu Sep 26 16:18:44 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
12 1 0 3 0
All Problems All Types
4 16
Host Status Details For All Host Groups
Limit Results: 100
Host
Status
Last Check
Duration
Status Information
linuxserver
UP
09-26-2024 16:15:45
0d 0h 57m 21s
PING OK - Packet loss = 0%, RTA = 1.07 ms
localhost
UP
09-26-2024 16:15:33
0d 7h 8m 53s
PING OK - Packet loss = 0%, RTA = 0.05 ms
Results 1 - 2 of 2 Matching Hosts

Siddhant Sathe **D15A/51**

EXPERIMENT 11

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

Theory:

AWS Lambda

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

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

Features of AWS Lambda

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

Packaging Functions

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

Execution Model

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

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

Stateless Functions

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

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

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

Common Use Cases for Lambda

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

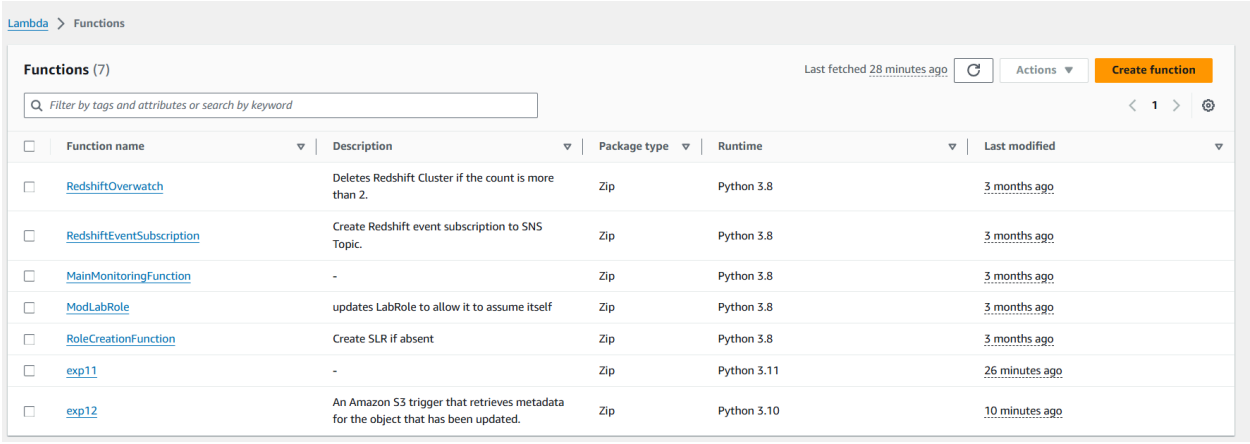
1. Individual tasks run for a short time;
2. Each task is generally self-contained;
3. There is a large difference between the lowest and highest levels in the workload of the application.

Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

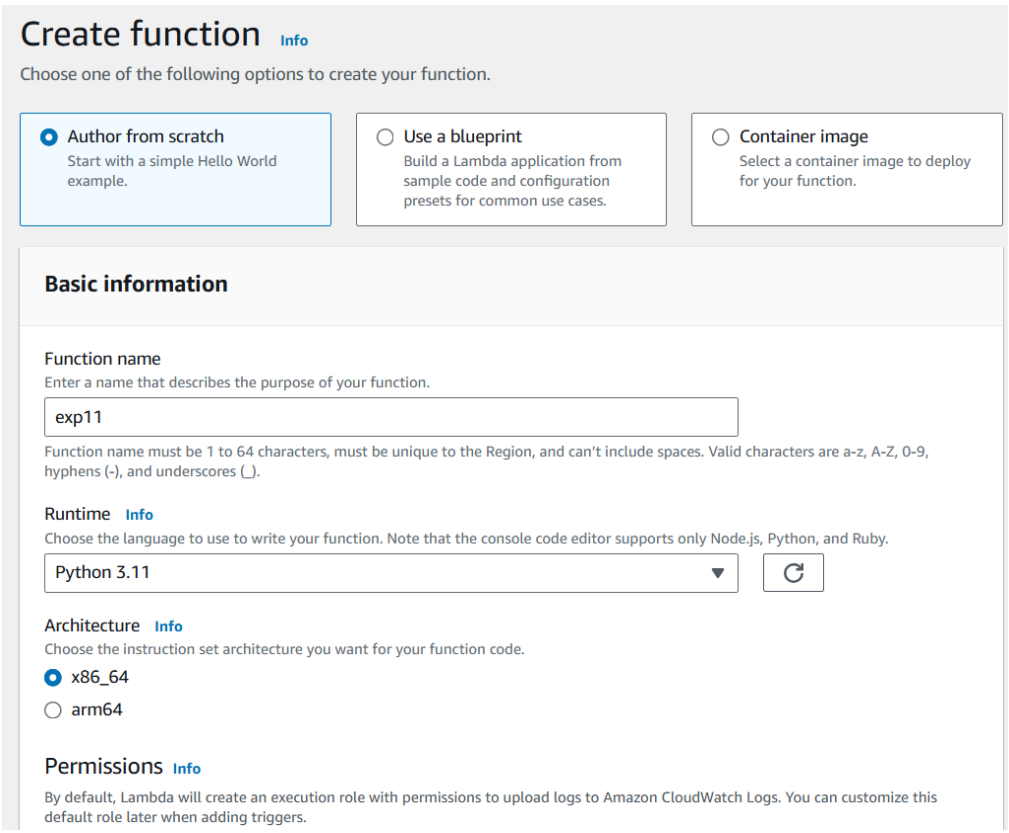
Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events. For example, you could employ Lambda to do some work every time an item in DynamoDB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

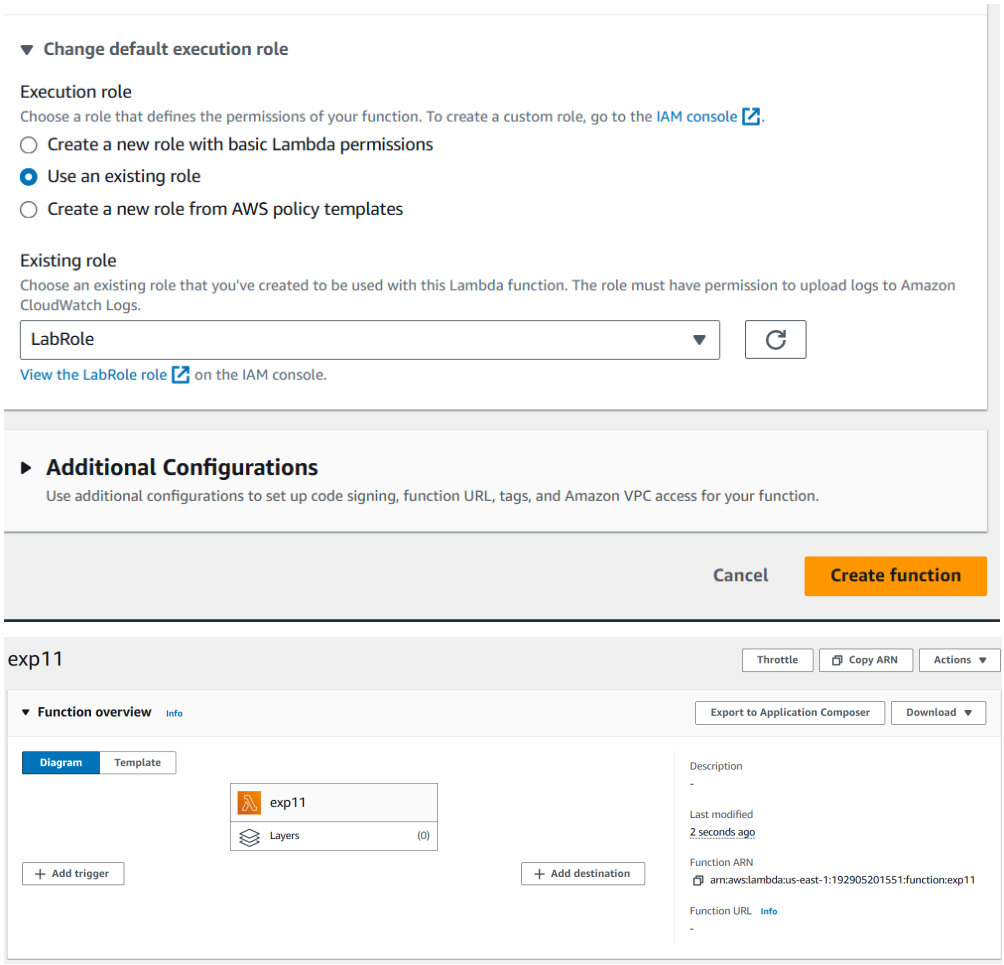
Steps to create an AWS Lambda function

Step 1: Open up the Lambda Console and click on the Create button. Be mindful of where you create your functions since Lambda is region-dependent.

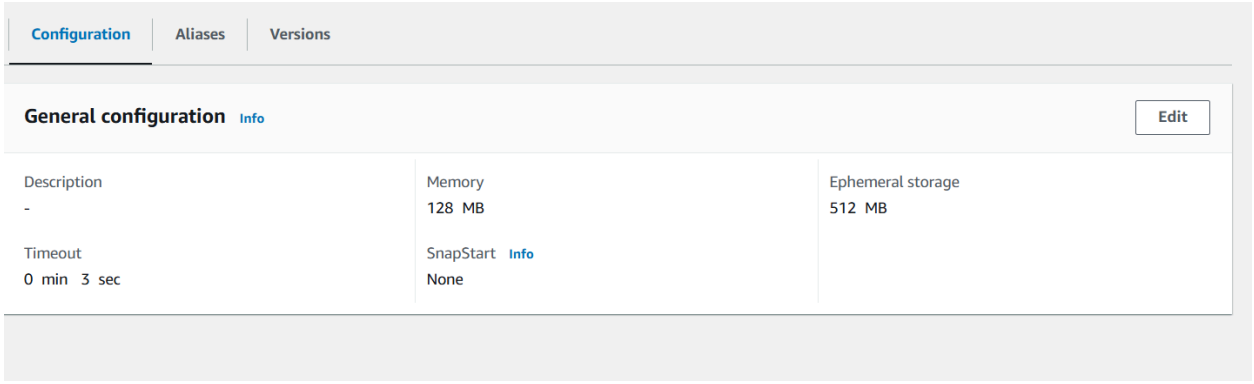


Step 2: Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones. After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

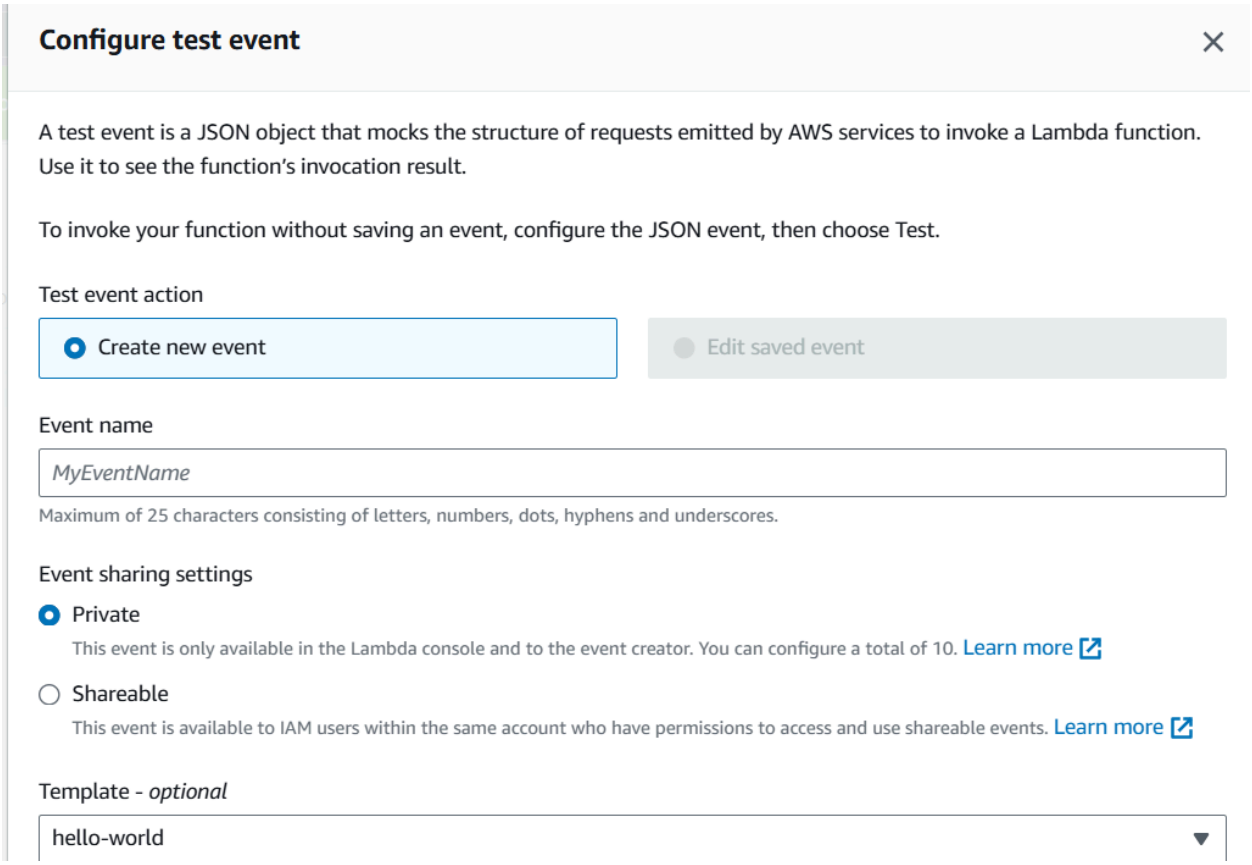




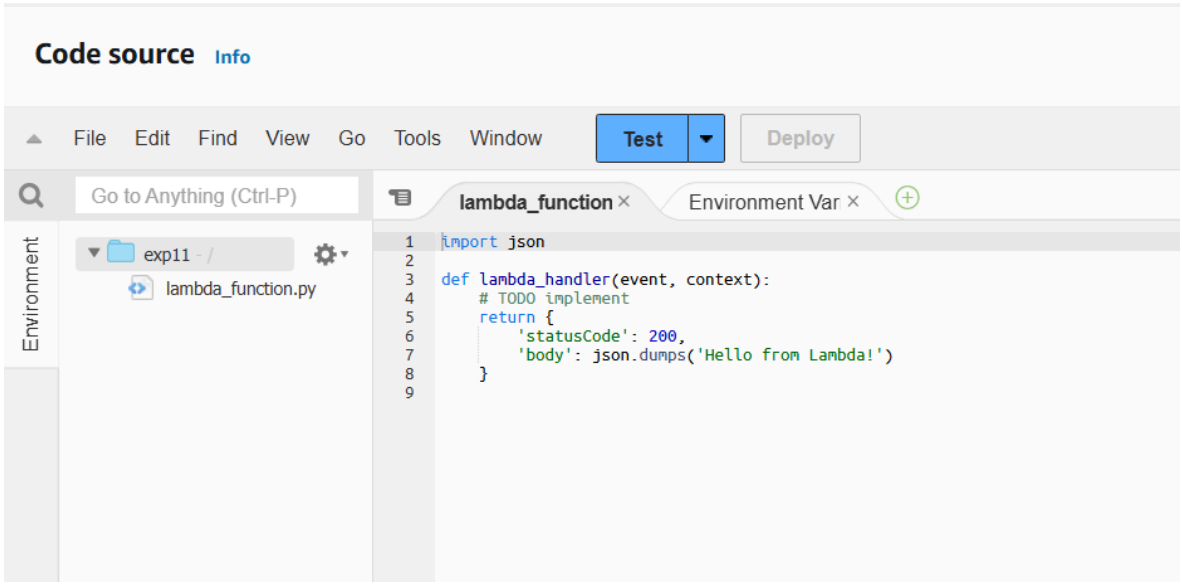
Step 3:. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit. Here, you can enter a description and change Memory and Timeout.



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 hello-world template.

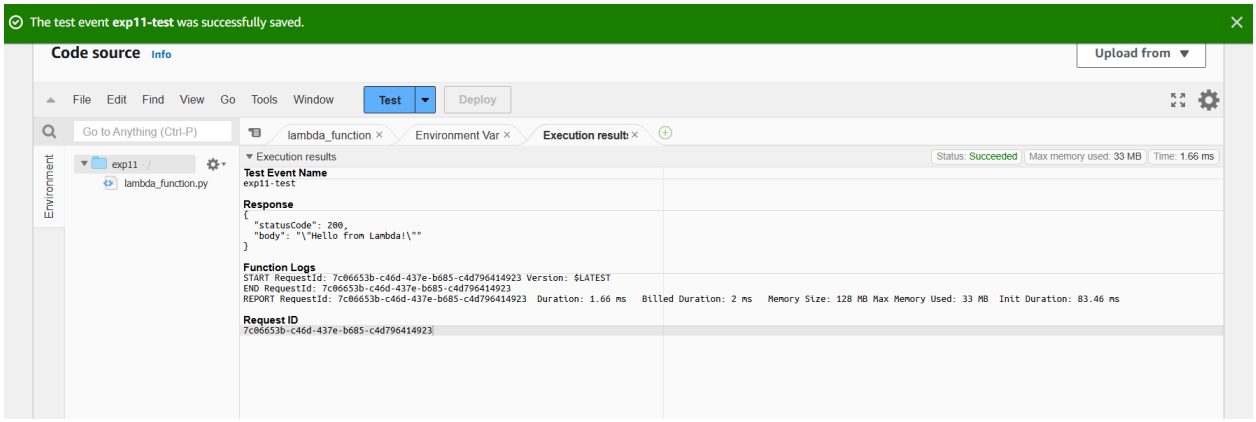


Step 5: Now In the Code section select the created event from the dropdown of test then click on test . You will see the below output.




Event JSON
Format JSON
1 {
2 "key1": "value1",
3 "key2": "value2",
4 "key3": "value3"
5 }
1:1 JSON Spaces: 2


Cancel
Invoke
Save


The test event exp11-test was successfully saved.
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Execution result
Environment
exp11
lambda_function.py
Execution results
Status: Succeeded
Max memory used: 33 MB
Time: 1.66 ms
Test Event Name
exp11-test
Response
{
"statusCode": 200,
"body": "\"Hello from Lambda!\""
}
Function Logs
START RequestId: 7c06653b-c46d-437e-b685-c4d796414923 Version: $LATEST
END RequestId: 7c06653b-c46d-437e-b685-c4d796414923
REPORT RequestId: 7c06653b-c46d-437e-b685-c4d796414923 Duration: 1.66 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 33 MB Init Duration: 83.46 ms
Request ID
7c06653b-c46d-437e-b685-c4d796414923

Siddhant Sathe **D15A/51**

EXPERIMENT 12

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

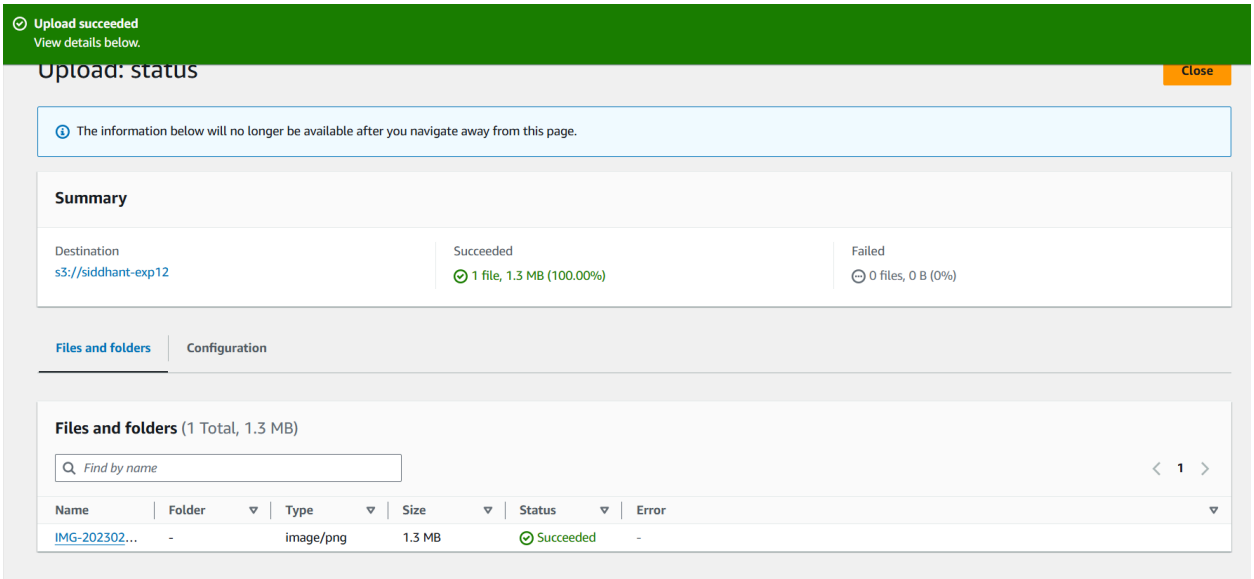
Theory:

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

Workflow:

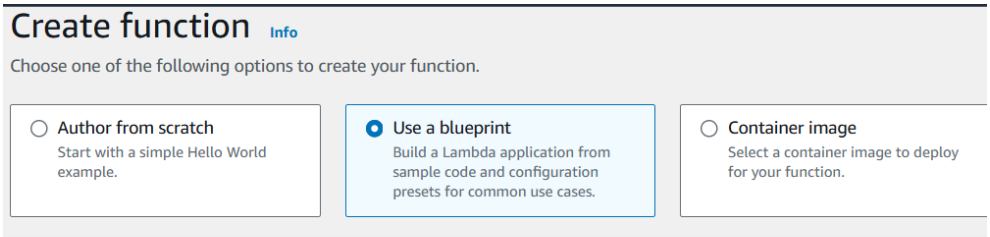
Step 1: Create an S3 Bucket:

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



Step 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.



Basic information Info

Blueprint name

Get S3 object — python3.10
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Function name
Enter a name that describes the purpose of your function.

exp12

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime
python3.10

Architecture
x86_64

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

- ○ Create a new role with basic Lambda permissions
- ◉ Use an existing role
- ○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

View the LabRole role on the IAM console.

○ Write code that gets S3 object and logs a message that returns content type when triggered.

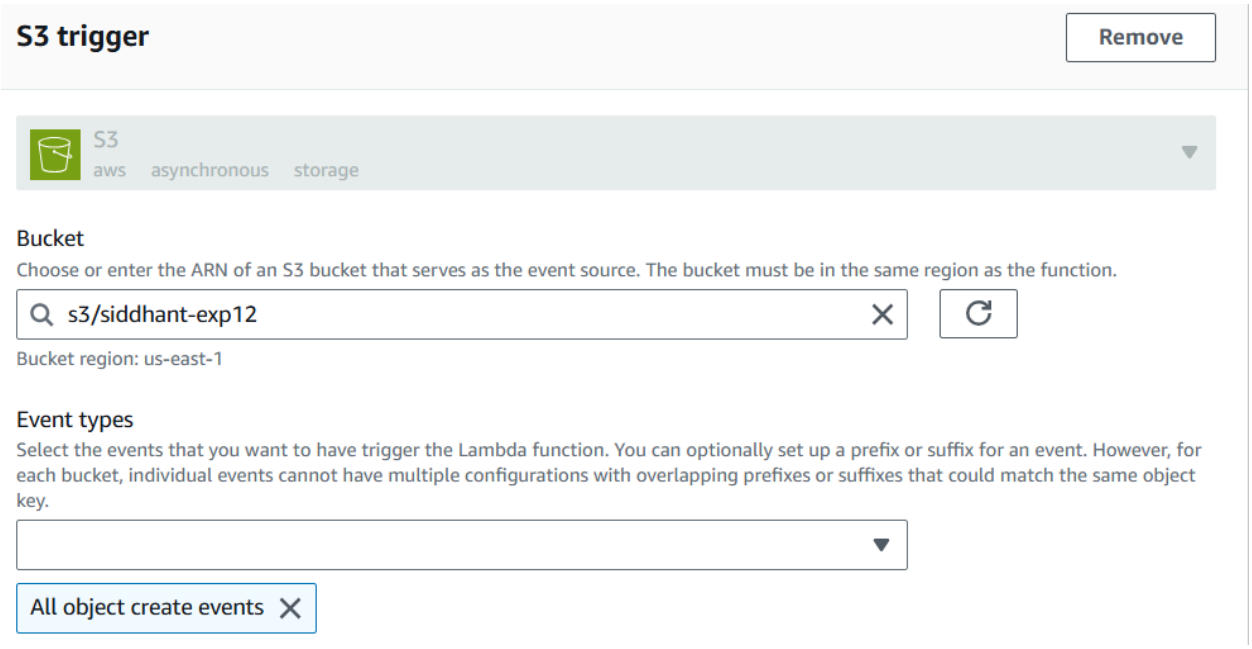
Lambda function code

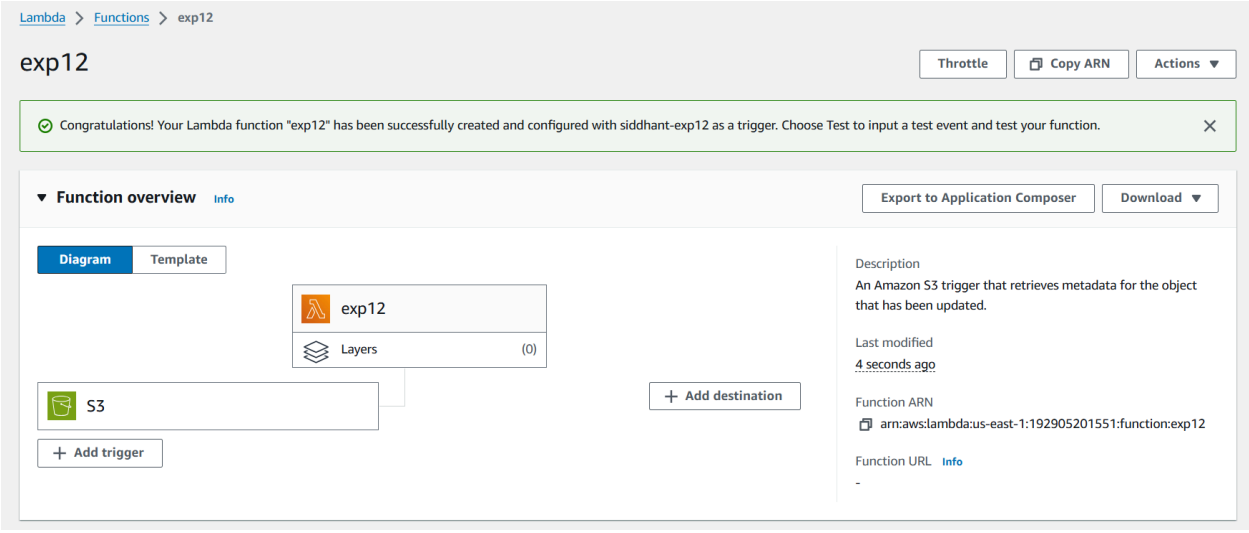
Code is preconfigured by the chosen blueprint. You can configure it after you create the function. Learn more about deploying Lambda functions.

```python
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the
        raise e

```

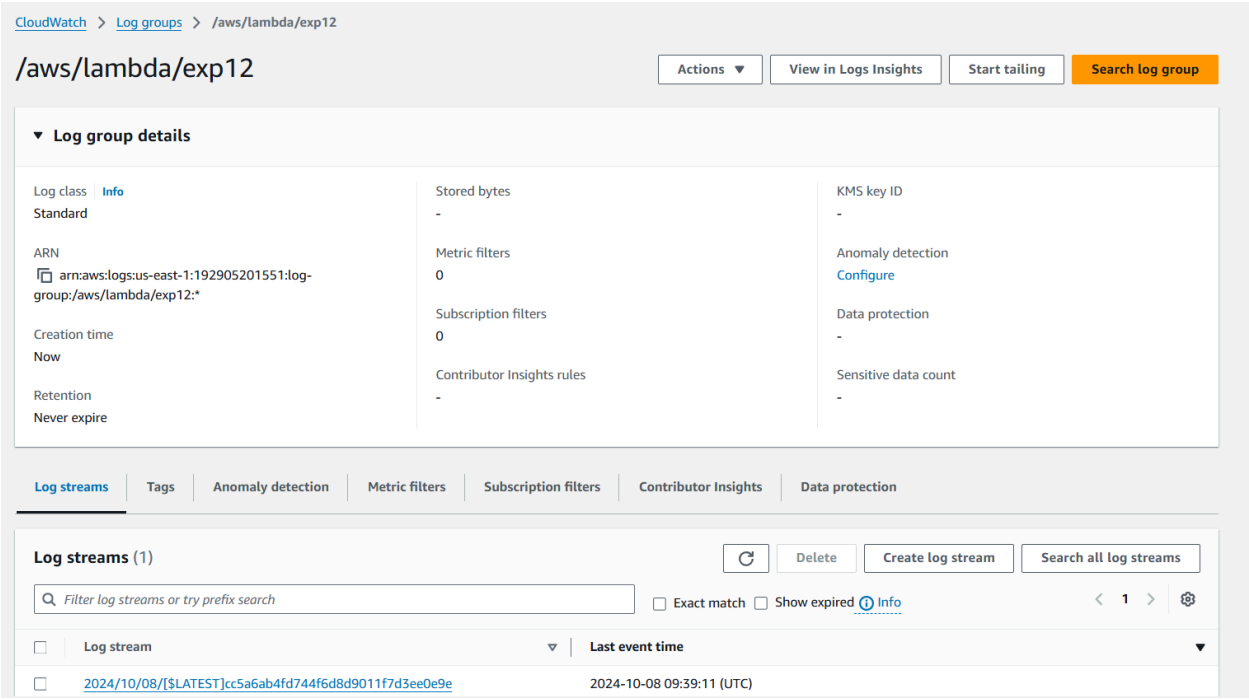
1:1 Python Spaces: 4

Step 3:. 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).





Step 4 : Search for CloudWatch in services. After Selecting CloudWatch select log groups and then select your respecting lambda function.



Step 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.