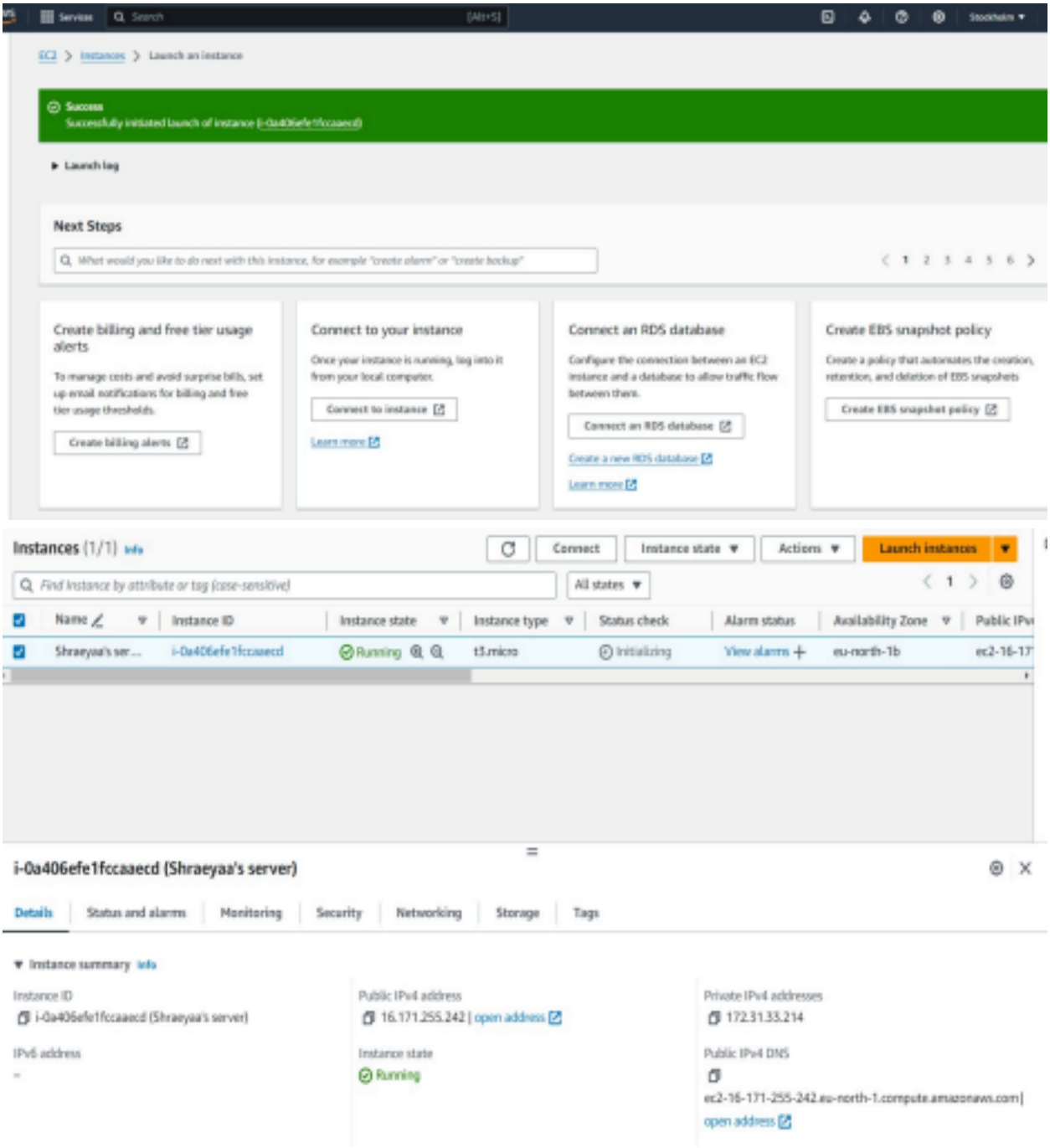
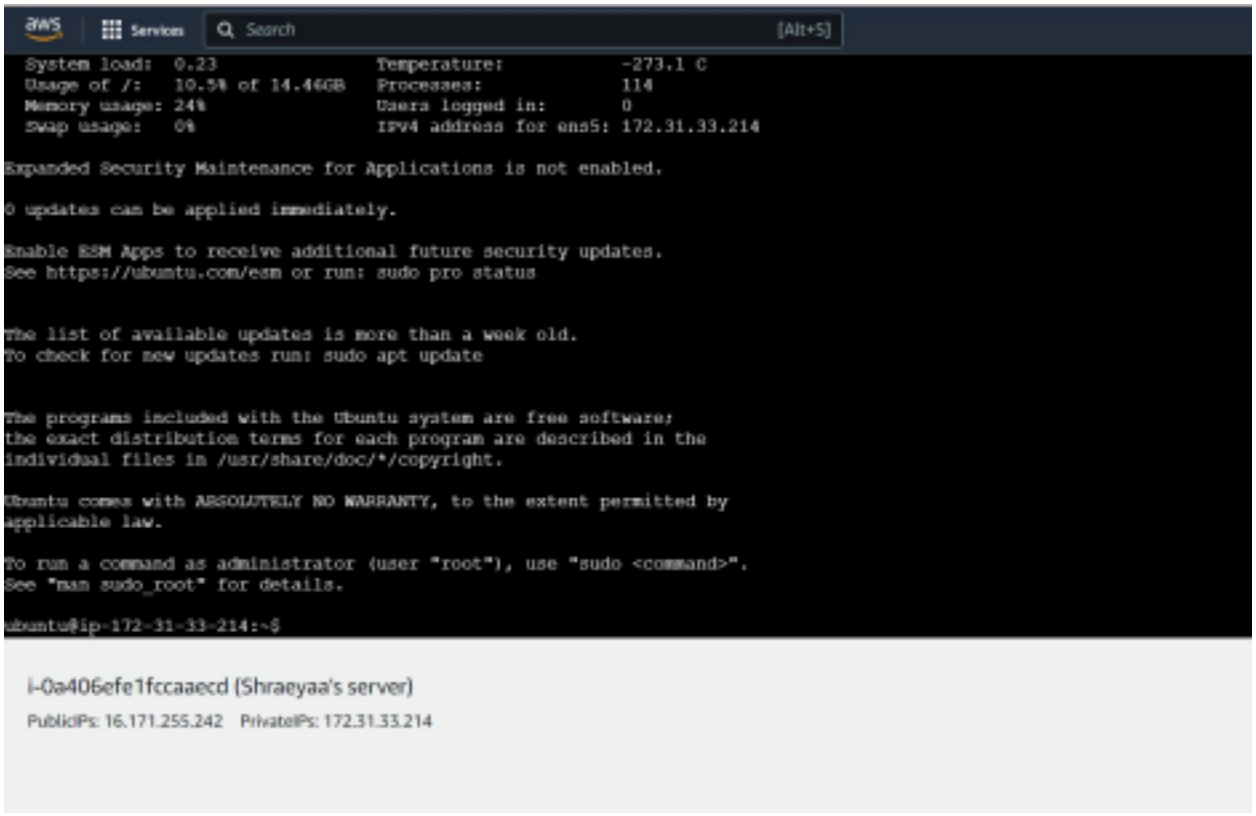
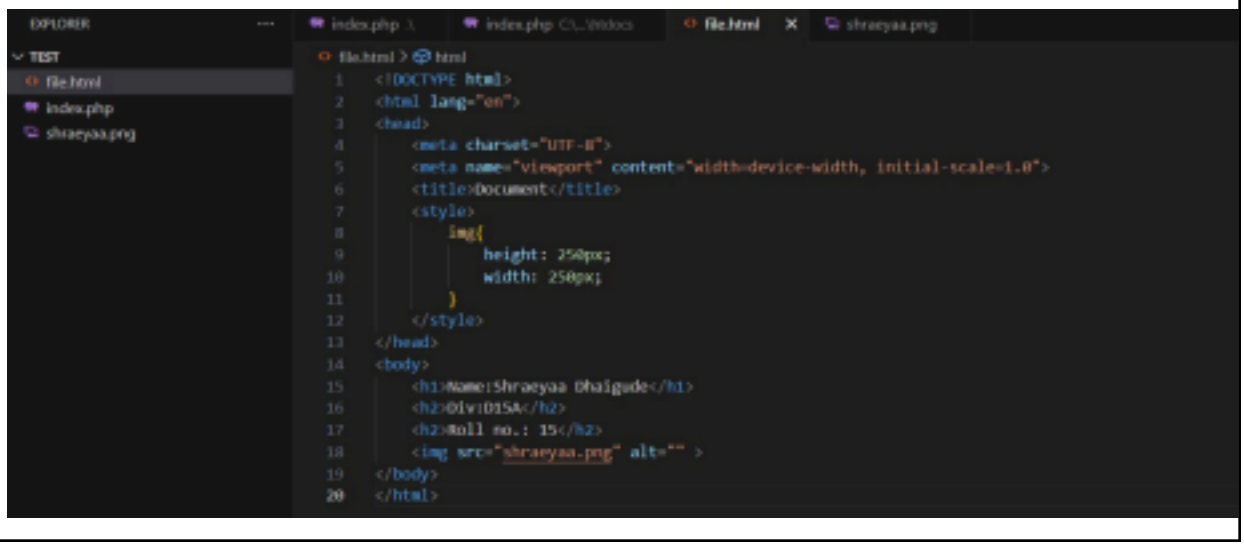
EC2 Instance:

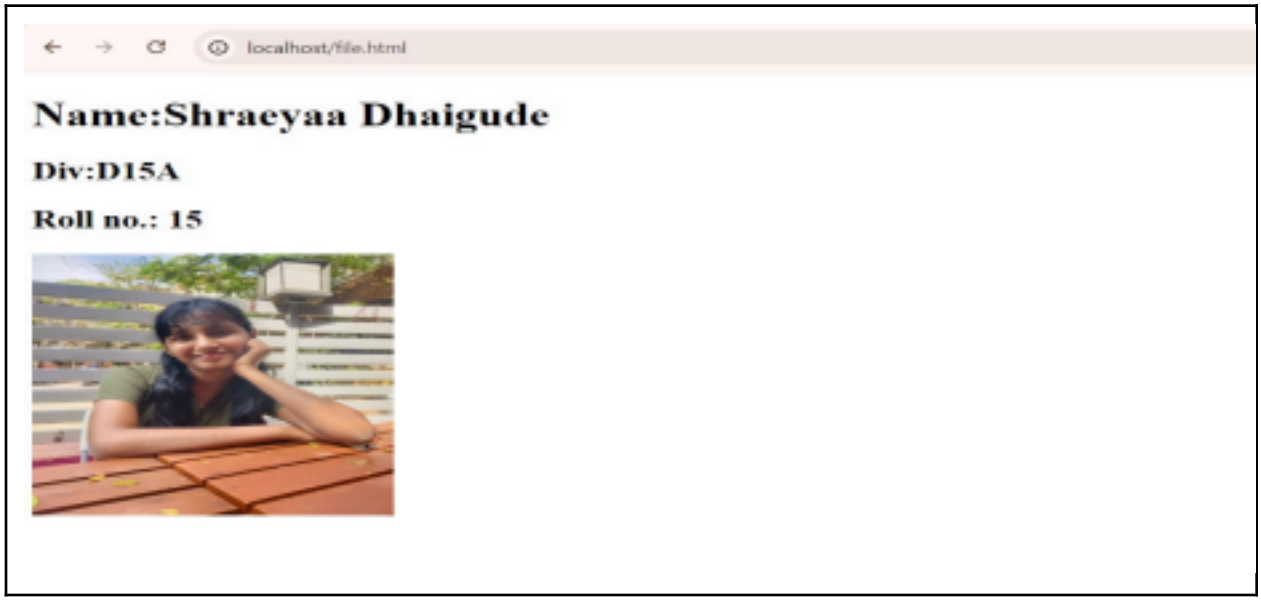
Instance Launching

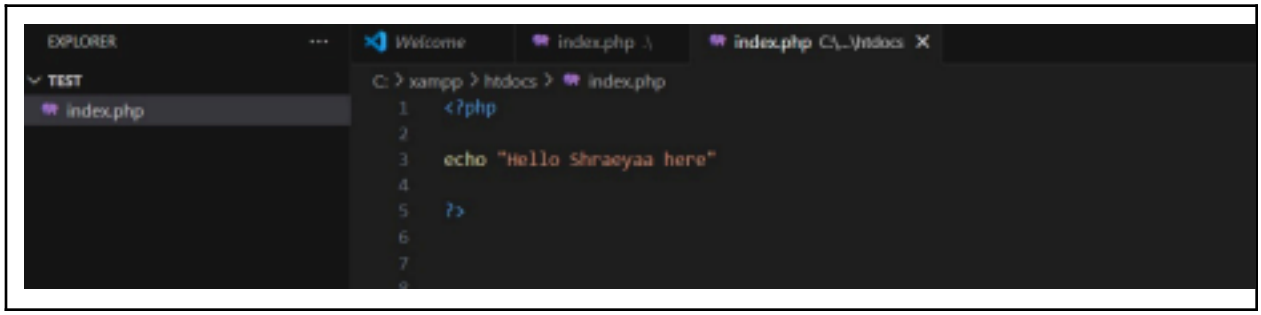


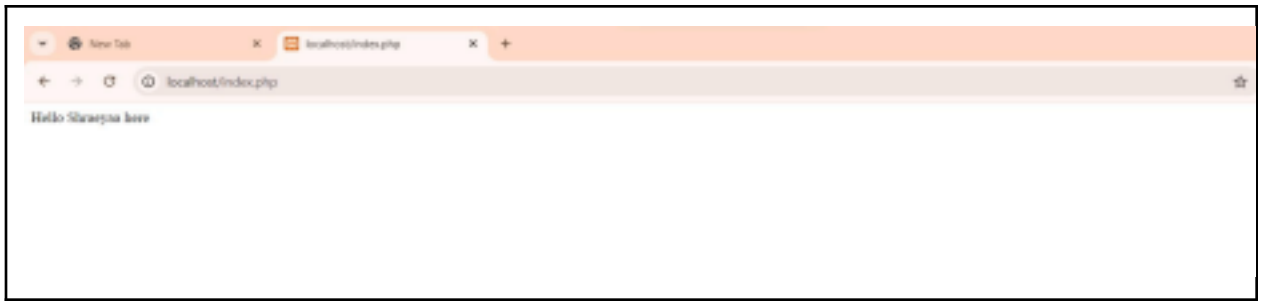


A) To develop a website and host it on local machine on a VM

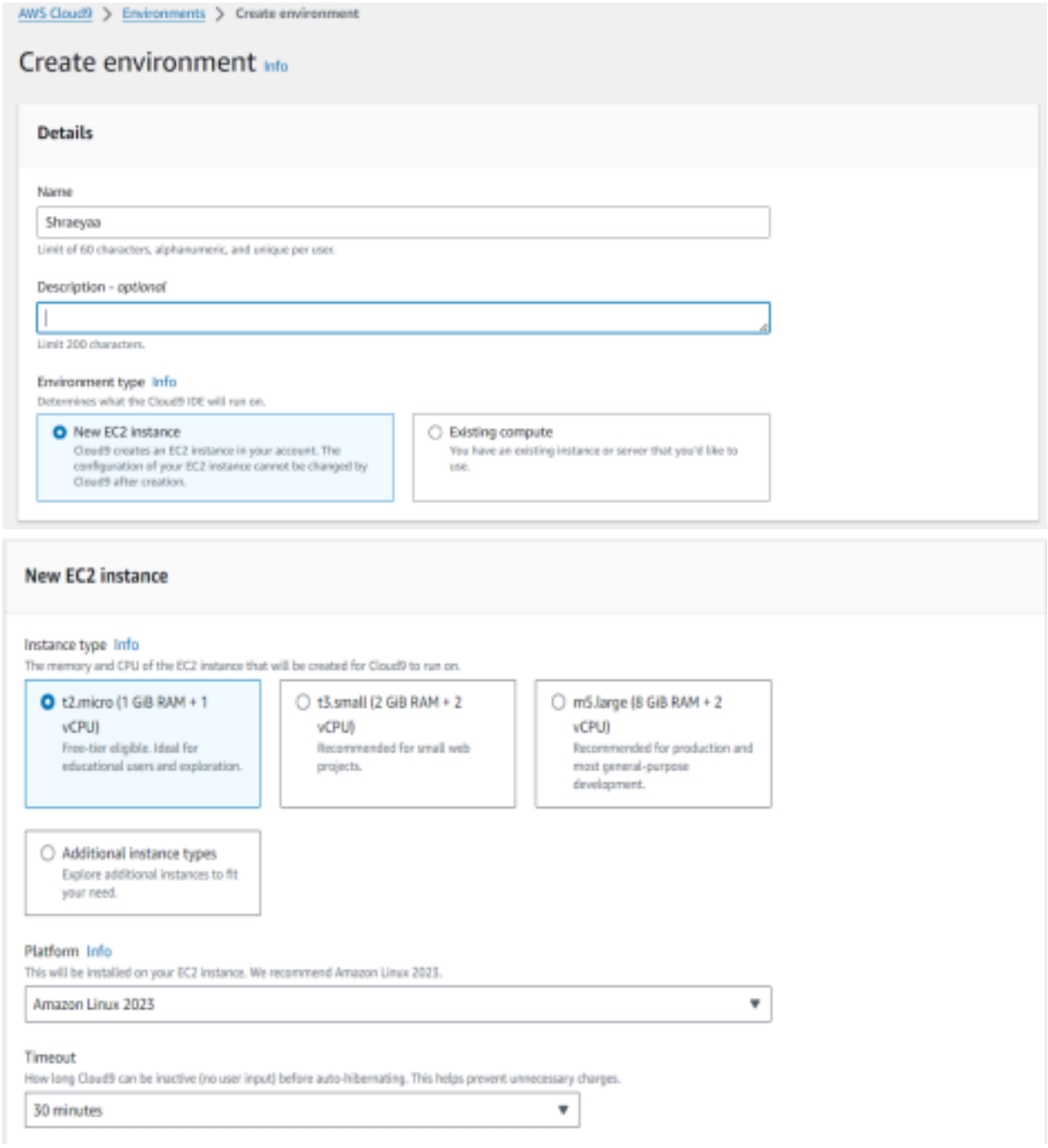


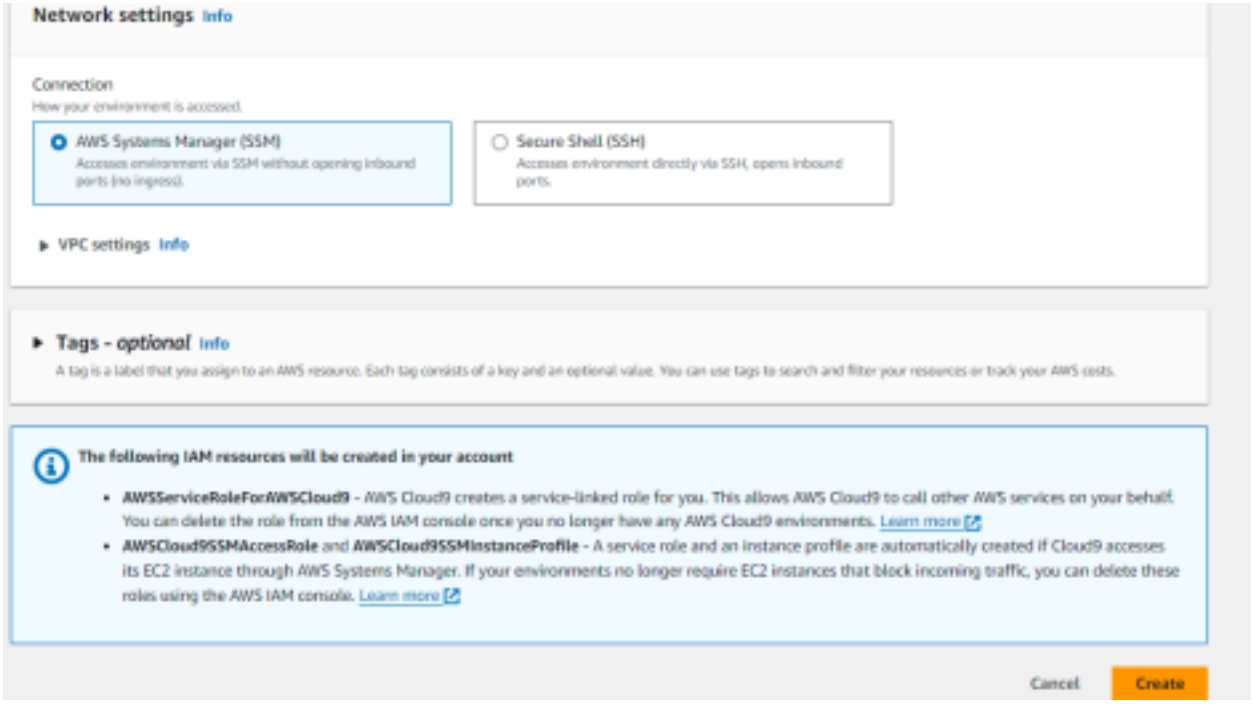


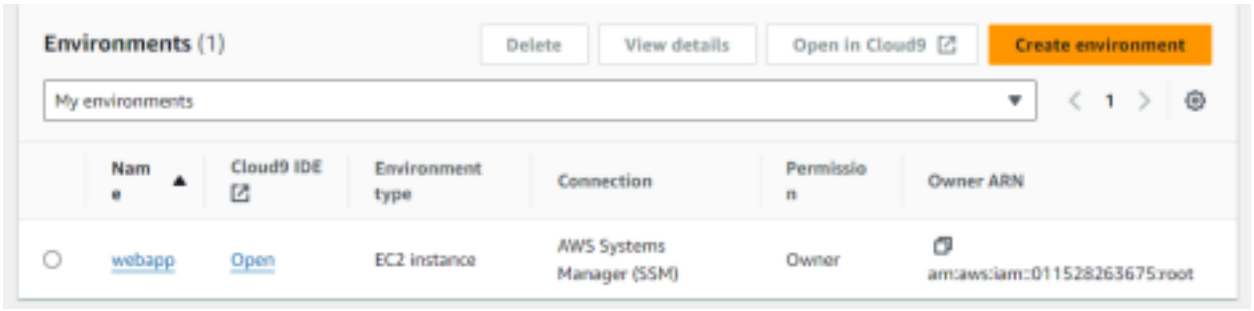


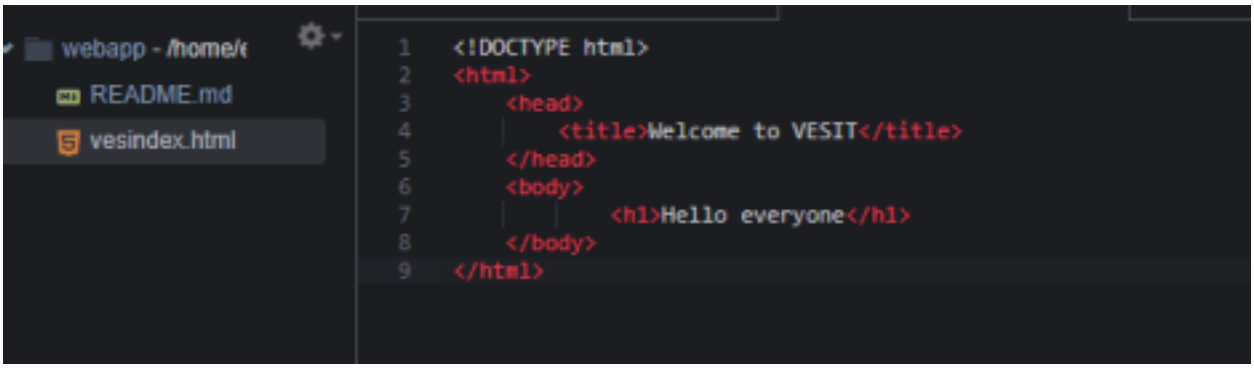


B)Setup Devops infrastructure on CloudWork and setup IDE on Cloud9


AWS Cloud9 > Environments > Create environment
Create environment Info
Details
Name
Shraeyaa
Limit of 60 characters, alphanumeric, and unique per user.
Description - optional
Limit 200 characters.
Environment type Info
Determines what the Cloud9 IDE will run on.
New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.
Existing compute
You have an existing instance or server that you'd like to use.
New EC2 instance
Instance type Info
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.
t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.
m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.
Additional instance types
Explore additional instances to fit your need.
Platform Info
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.
Amazon Linux 2023
Timeout
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.
30 minutes


Network settings Info
Connection
How your environment is accessed.
AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).
Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.
VPC settings Info
Tags - optional Info
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
The following IAM resources will be created in your account
AWSServiceRoleForAWSCloud9 - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
AWSCloud9SSMAccessRole and AWSCloud9SSMInstanceProfile - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. Learn more
Cancel
Create


Environments (1)
Delete
View details
Open in Cloud9
Create environment
My environments
1
Name
Cloud9 IDE
Environment type
Connection
Permission
Owner ARN
webapp
Open
EC2 instance
AWS Systems Manager (SSM)
Owner
arn:aws:iam::011528263675:root


webapp - /home/
README.md
vesindex.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to VESIT</title>
</head>
<body>
<h1>Hello everyone</h1>
</body>
</html>

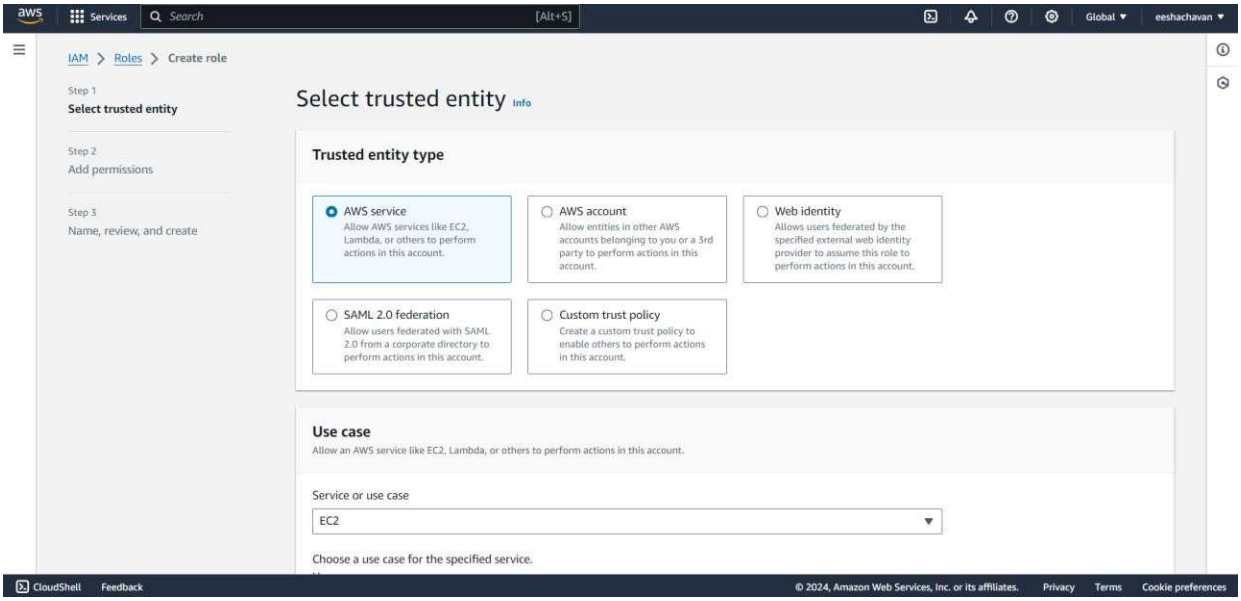
SHRAEYAA DHAIGUDE **D15A15**

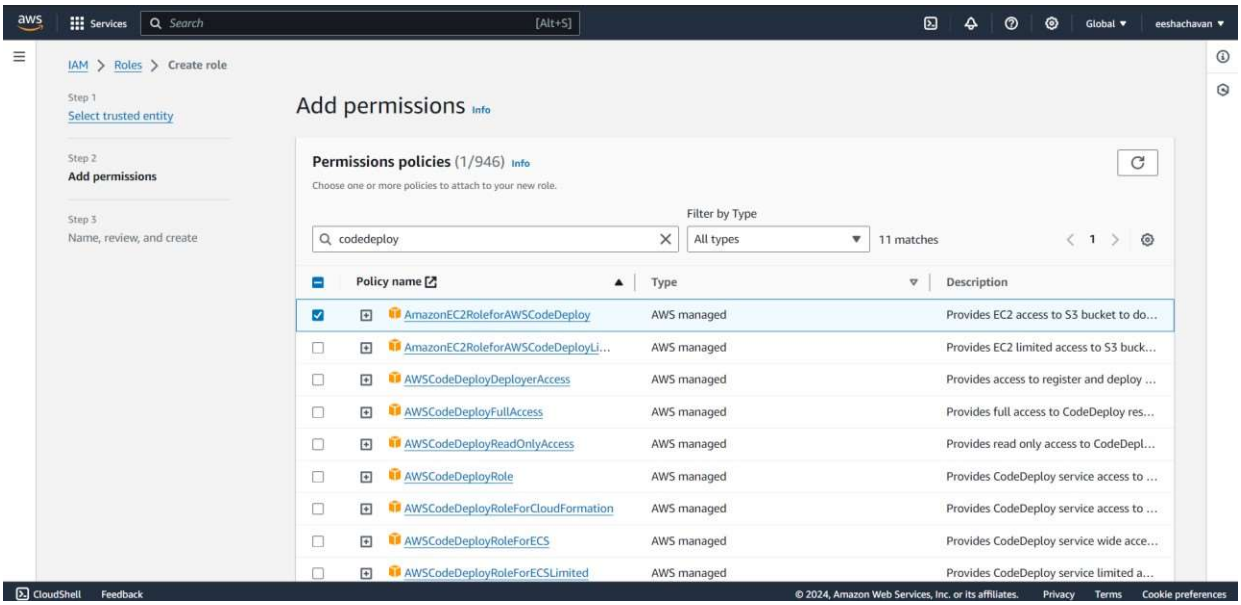
ADV DEVOPS EXP2

AIM:

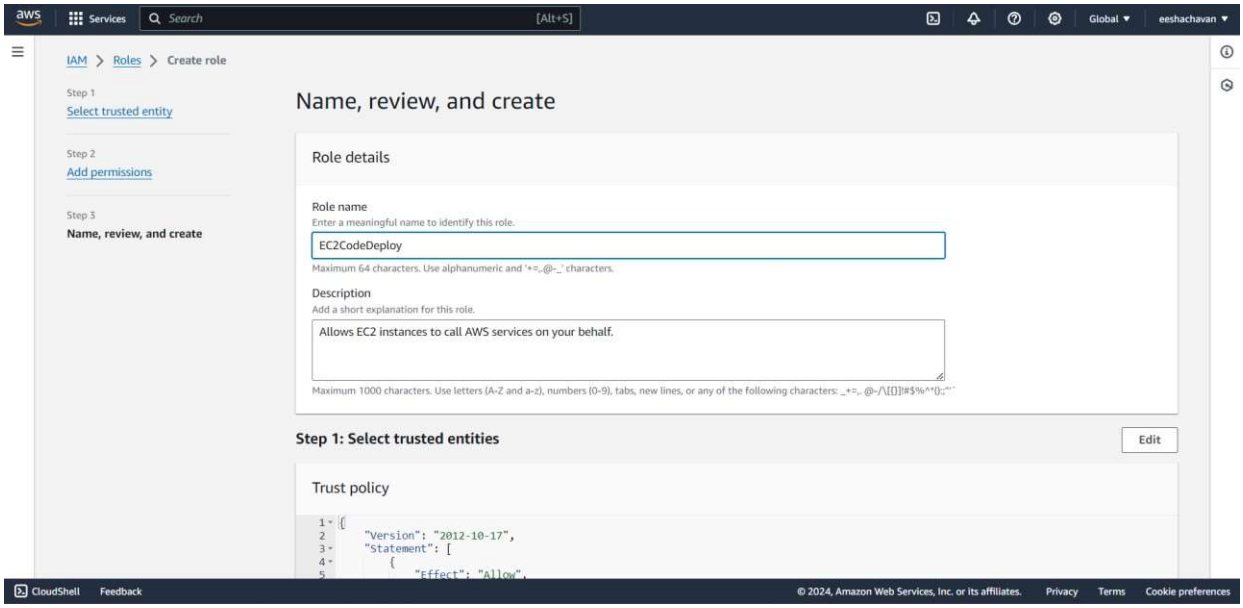
To Build Your Application using AWS Code Build and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS Code Deploy.

Create role

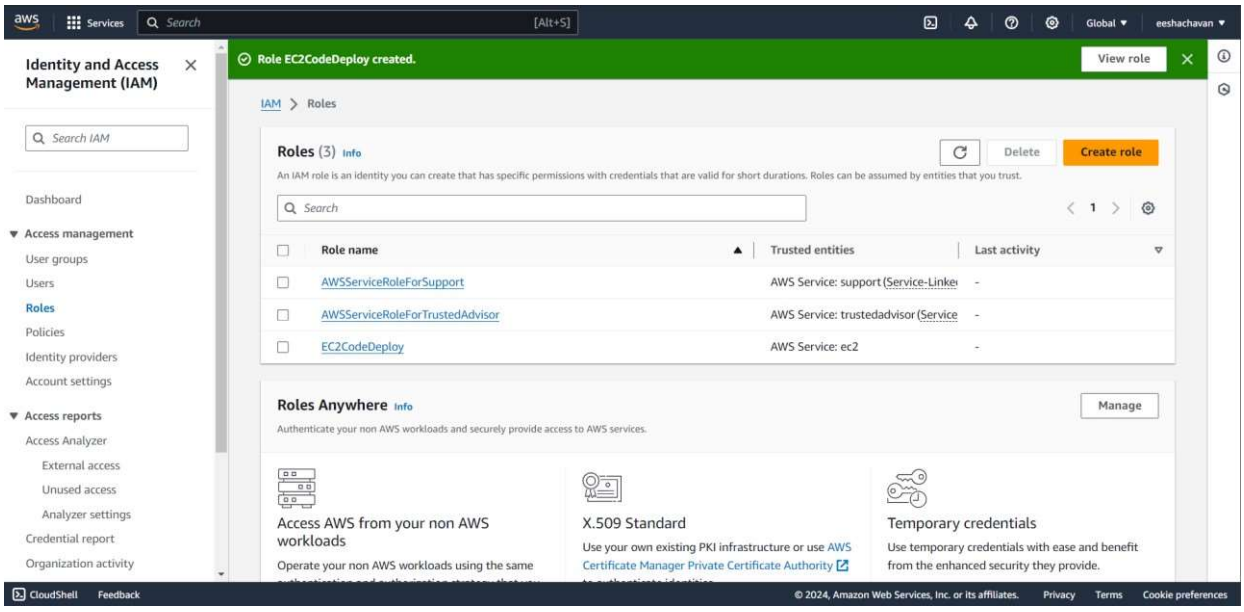




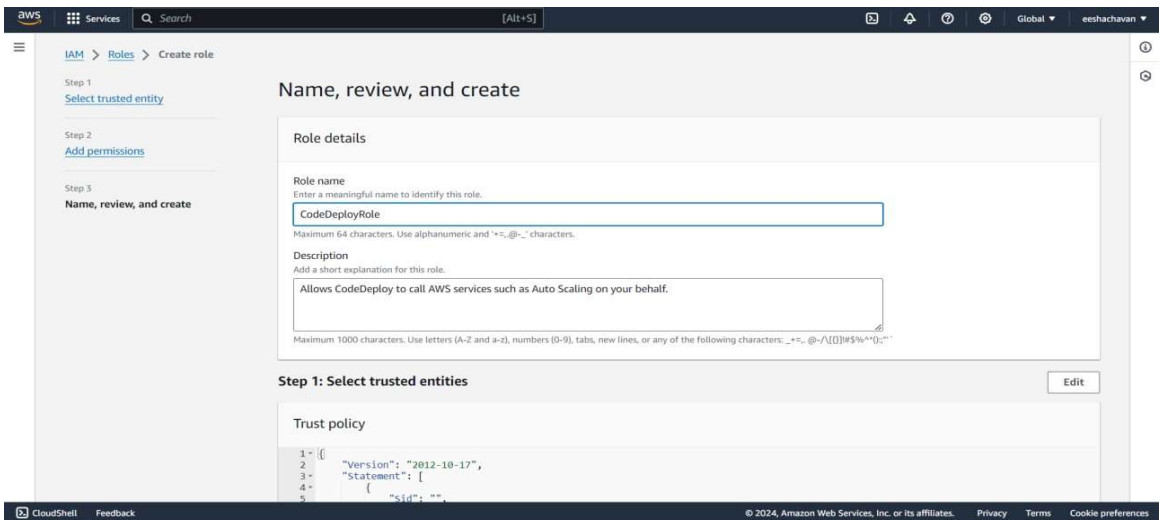
Role Details

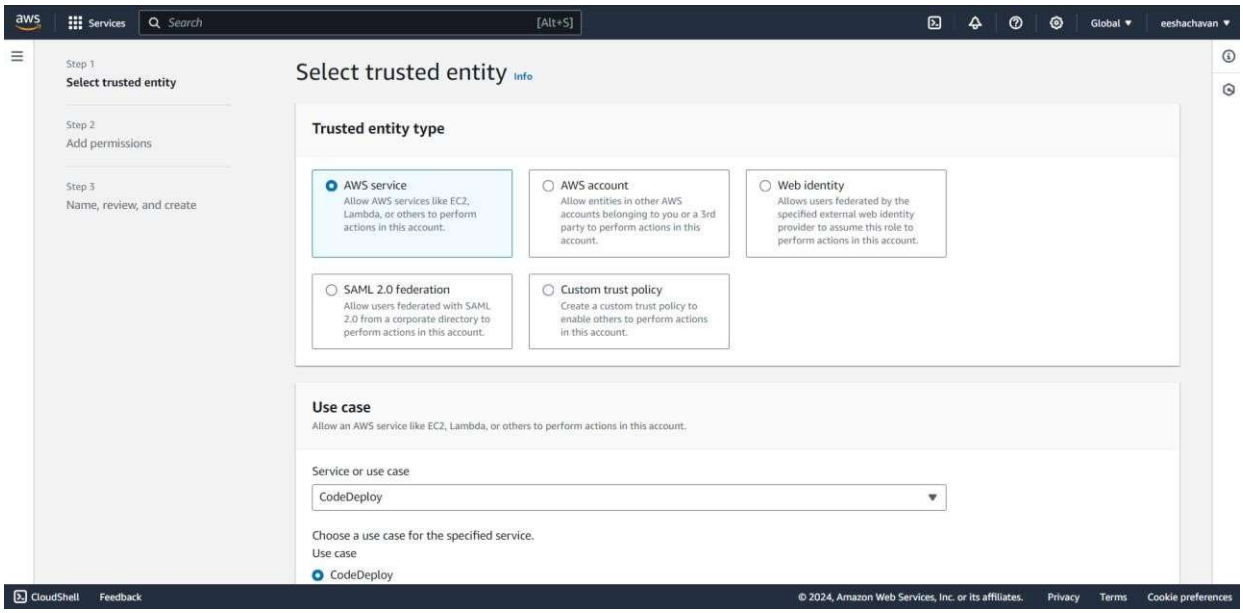


Role Created

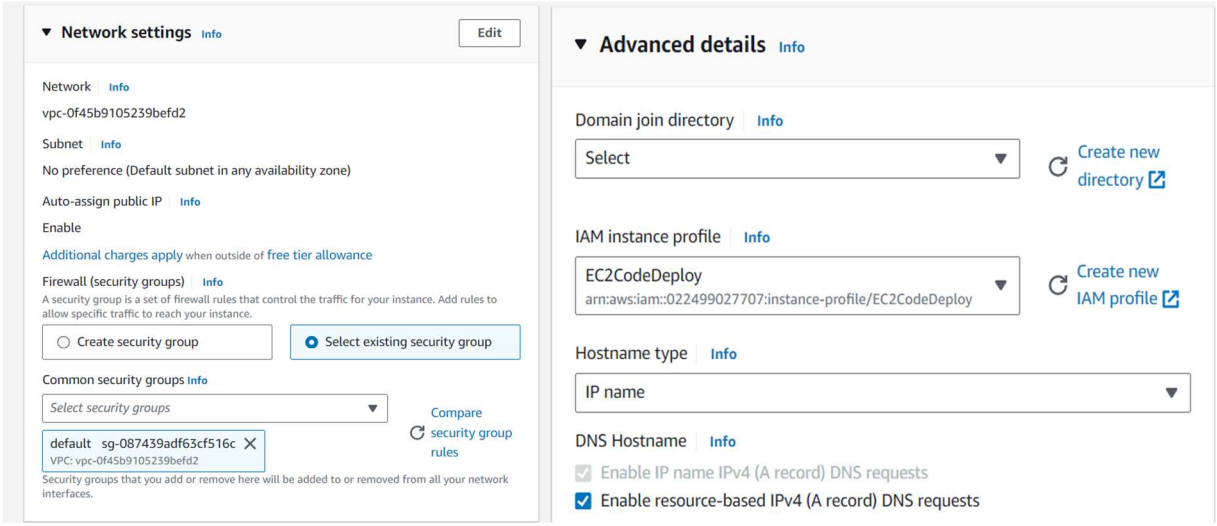


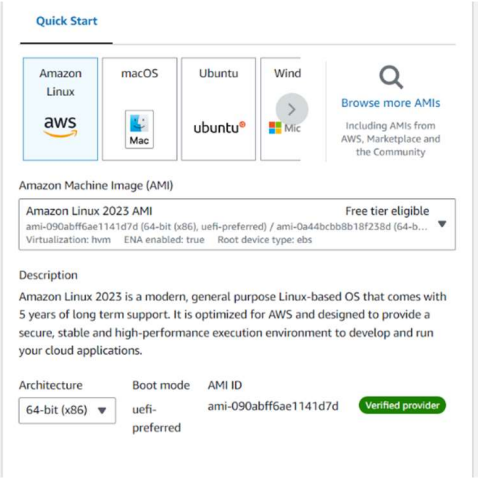
Create Role 2

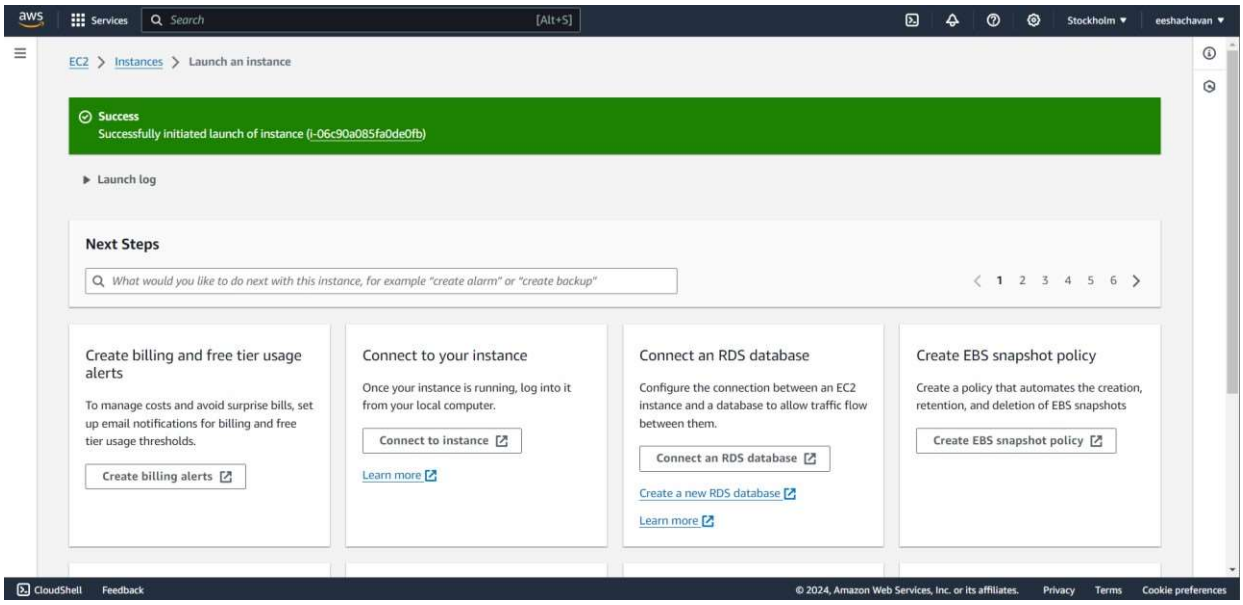


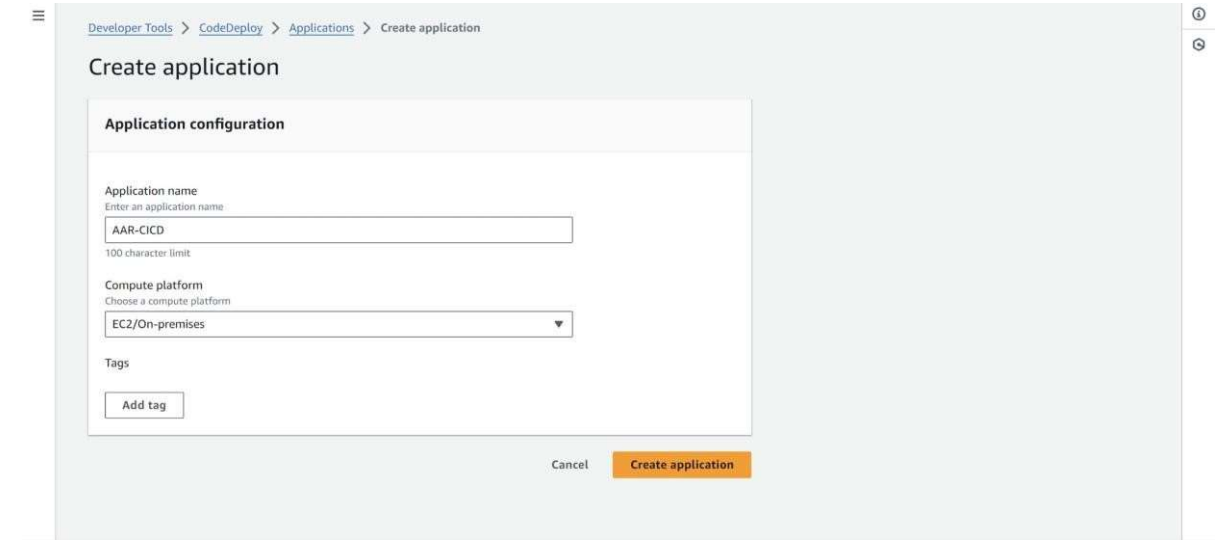


Launch instance after creating role









Enter a deployment group name

AAR-CICD-DP

100 character limit

Service role

Enter a service role

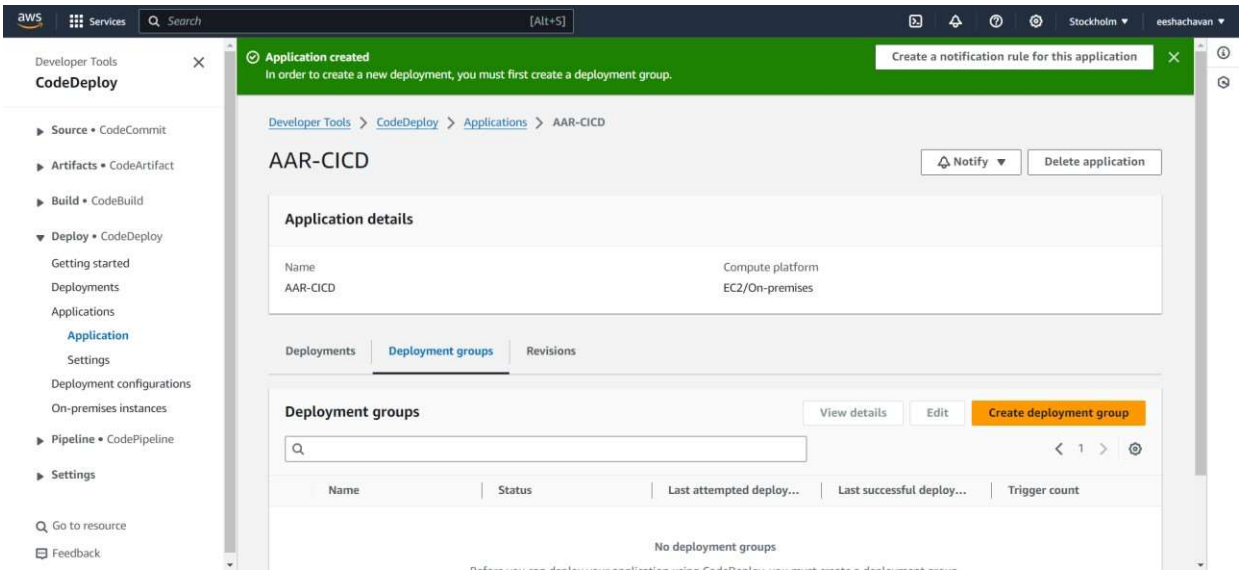
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

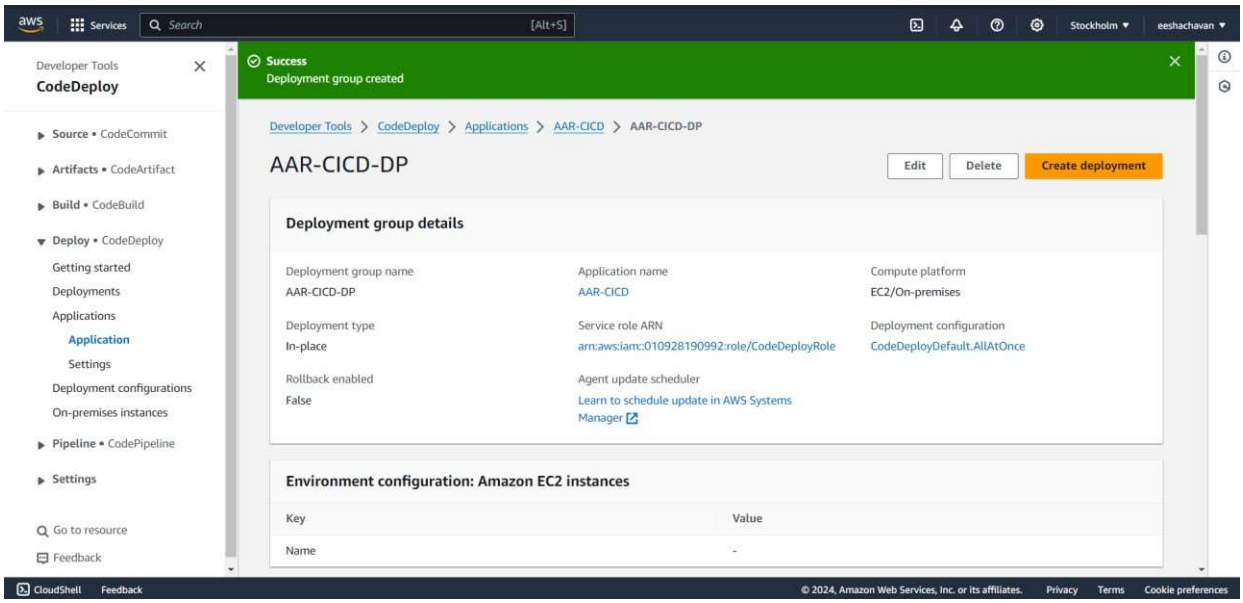
arn:aws:iam::022499027707:role/CodeDeployRole

Deployment type

Choose how to deploy your application

- In-place
  Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update
- Blue/green
  Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.


aws
Services
Search
[Alt+S]
Stockholm
eeshachavan
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Application
Settings
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Go to resource
Feedback
Application created
In order to create a new deployment, you must first create a deployment group.
Create a notification rule for this application
Developer Tools > CodeDeploy > Applications > AAR-CICD
AAR-CICD
Notify
Delete application
Application details
Name
AAR-CICD
Compute platform
EC2/On-premises
Deployments
Deployment groups
Revisions
Deployment groups
View details
Edit
Create deployment group
Name
Status
Last attempted deploy...
Last successful deploy...
Trigger count
No deployment groups


aws
Services
Search
[Alt+S]
Stockholm
eeshachavan
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Application
Settings
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Go to resource
Feedback
Success
Deployment group created
Developer Tools > CodeDeploy > Applications > AAR-CICD > AAR-CICD-DP
AAR-CICD-DP
Edit
Delete
Create deployment
Deployment group details
Deployment group name
AAR-CICD-DP
Application name
AAR-CICD
Compute platform
EC2/On-premises
Deployment type
In-place
Service role ARN
arn:aws:iam::010928190992:role/CodeDeployRole
Deployment configuration
CodeDeployDefault.AllAtOnce
Rollback enabled
False
Agent update scheduler
Learn to schedule update in AWS Systems Manager
Environment configuration: Amazon EC2 instances
Key
Value
Name
-
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Advanced details Info

Domain join directory Info

Select

Create new directory

IAM instance profile Info

EC2CodeDeploy
arn:aws:iam::022499027707:instance-profile/EC2CodeDeploy

Create new IAM profile

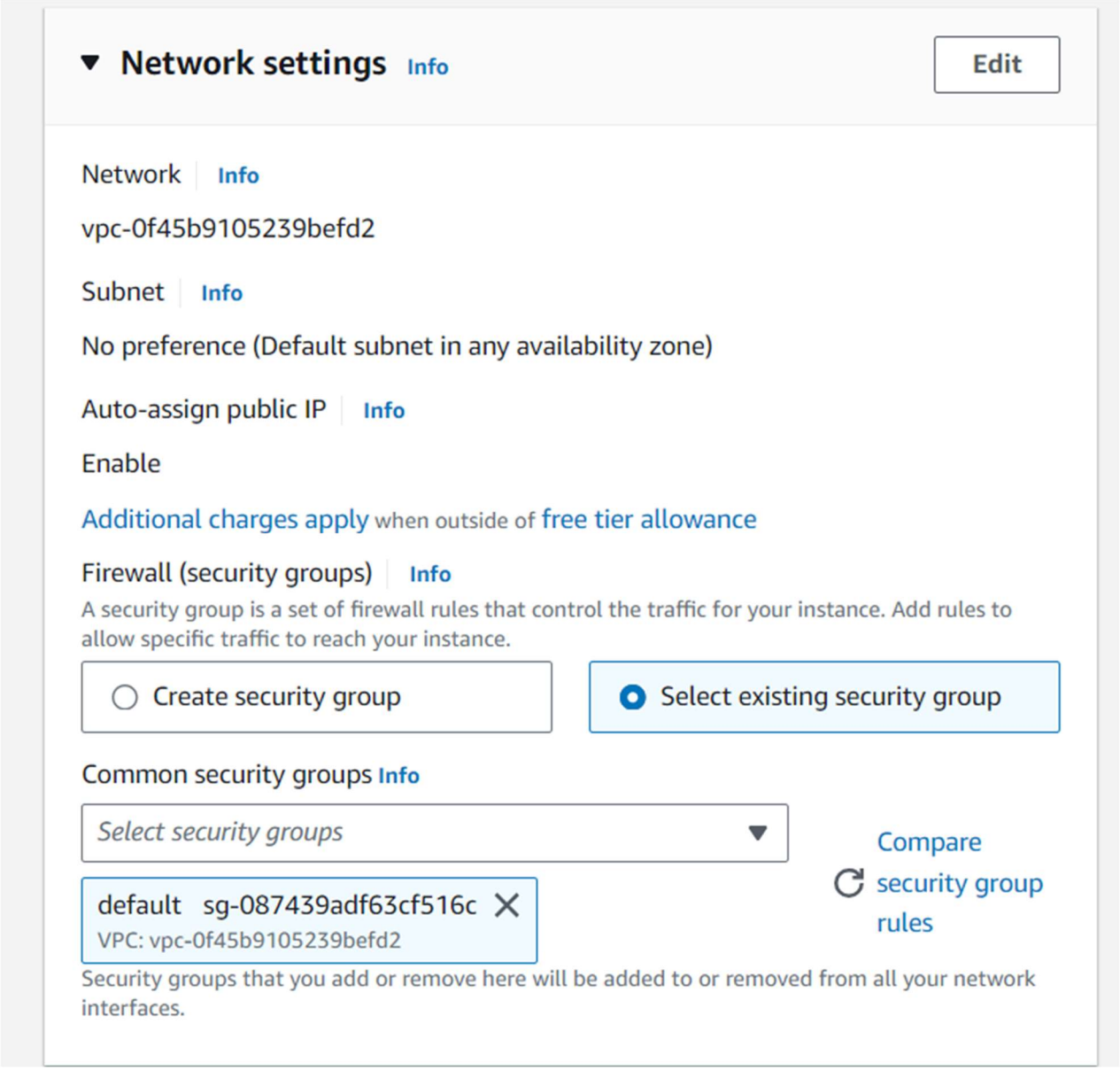
Hostname type Info

IP name

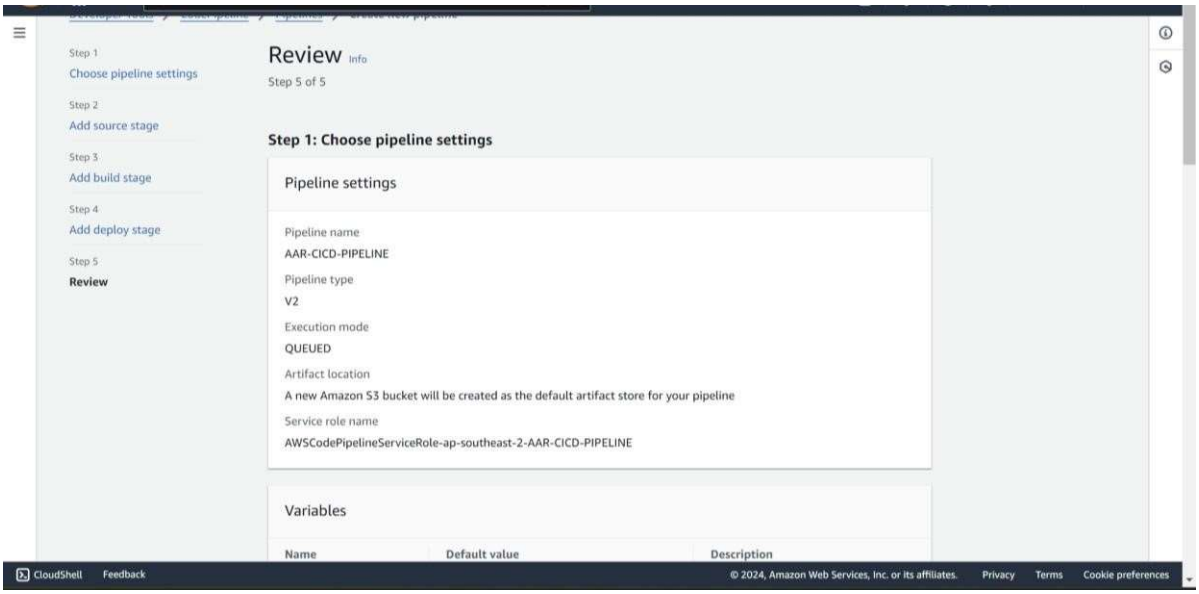
DNS Hostname Info

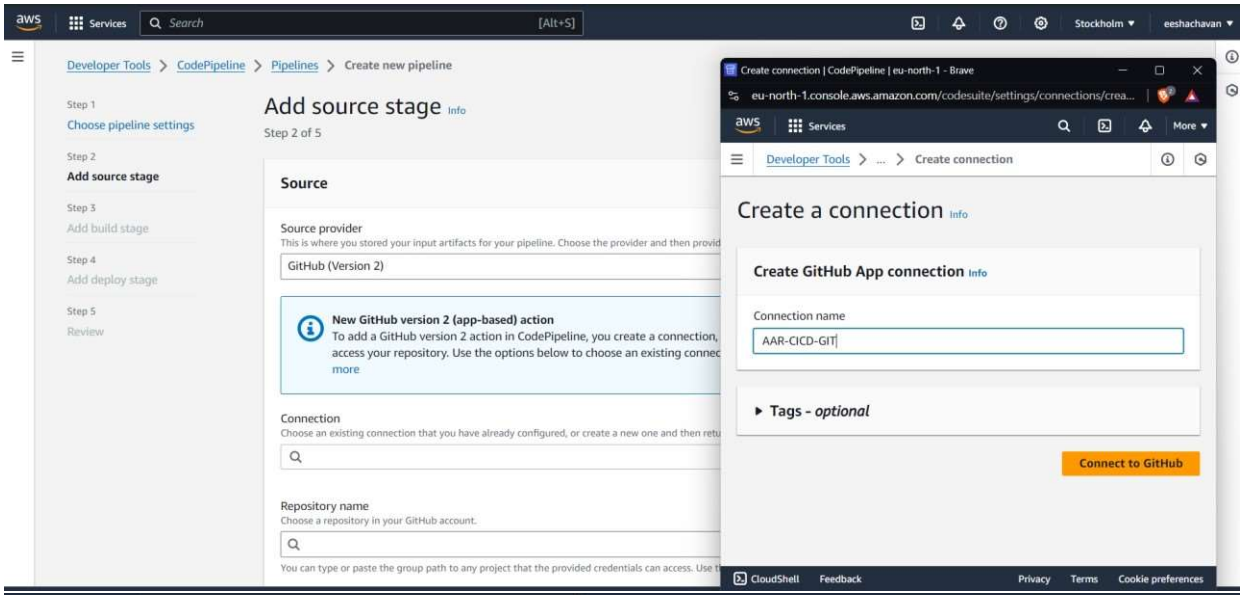
Enable IP name IPv4 (A record) DNS requests

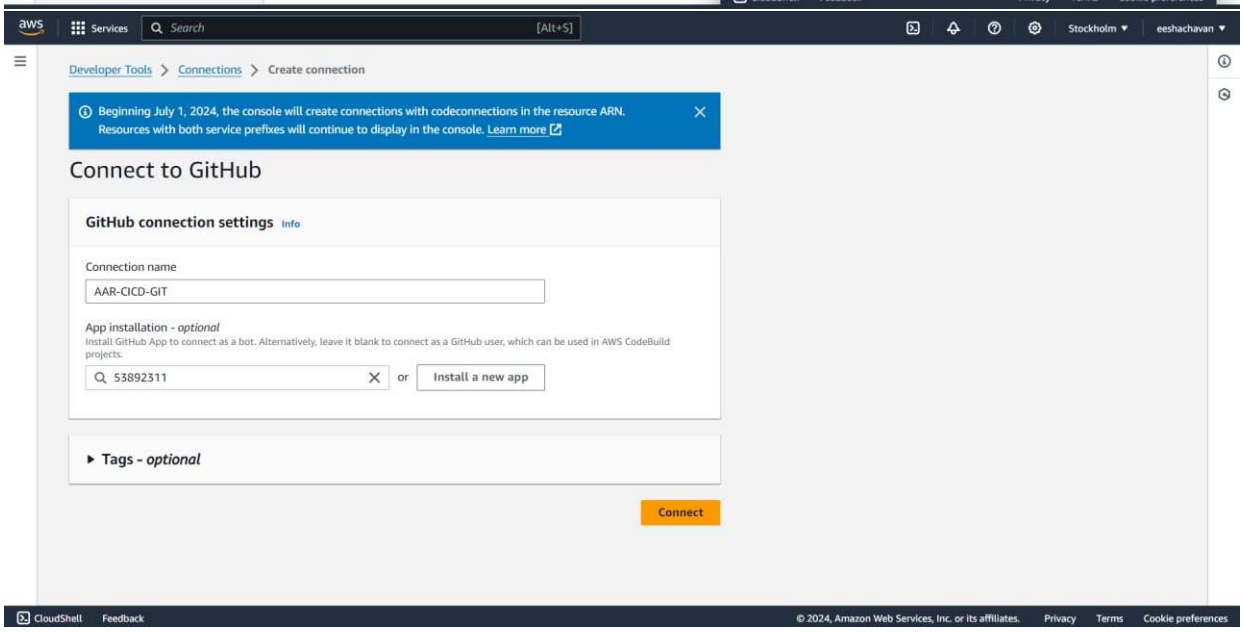
Enable resource-based IPv4 (A record) DNS requests

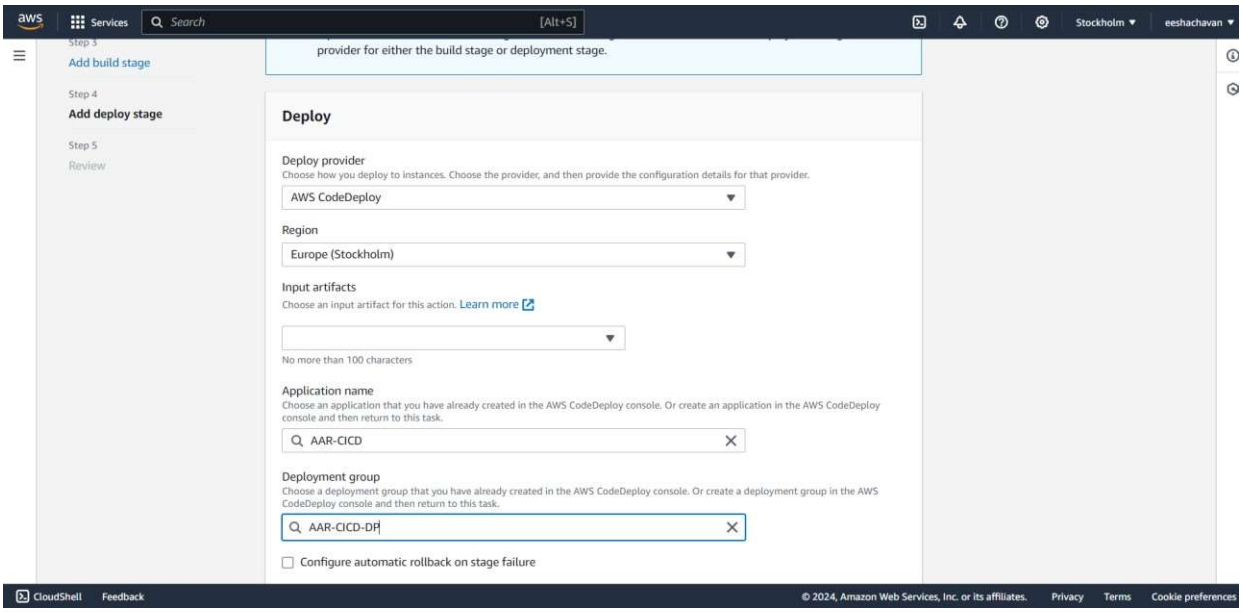


Create pipeline step by step


Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Review Info
Step 5 of 5
Step 1: Choose pipeline settings
Pipeline settings
Pipeline name
AAR-CICD-PIPELINE
Pipeline type
V2
Execution mode
QUEUED
Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline
Service role name
AWSCodePipelineServiceRole-ap-southeast-2-AAR-CICD-PIPELINE
Variables
Name
Default value
Description
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Stockholm
eeshachavan
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage Info
Step 2 of 5
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provid
GitHub (Version 2)
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection,
access your repository. Use the options below to choose an existing connec
more
Connection
Choose an existing connection that you have already configured, or create a new one and then retu
Repository name
Choose a repository in your GitHub account.
You can type or paste the group path to any project that the provided credentials can access. Use t
Create connection | CodePipeline | eu-north-1 - Brave
eu-north-1.console.aws.amazon.com/codesuite/settings/connections/crea...
Services
More
Developer Tools > ... > Create connection
Create a connection Info
Create GitHub App connection Info
Connection name
AAR-CICD-GIT
Tags - optional
Connect to GitHub
CloudShell
Feedback
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Stockholm
eeshachavan
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
AAR-CICD-GIT
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.
53892311
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

AAR-CICD-PIPELINE

Pipeline type

V2

Execution mode

QUEUED

Artifact location

A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name

AWSCodePipelineServiceRole-eu-north-1-AAR-CICD-PIPELINE

Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 2)

OutputArtifactFormat
CODE_ZIP

DetectChanges
false

ConnectionArn
arn:aws:codeconnections:eu-north-1:022499027707:connection/11f0aa91-d0a7-4358-9471-ba18c8472ba9

FullRepositoryId
shrayasic/aws_cicd_pipline_codedeploy

Default branch
main

Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type

Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS CodeDeploy

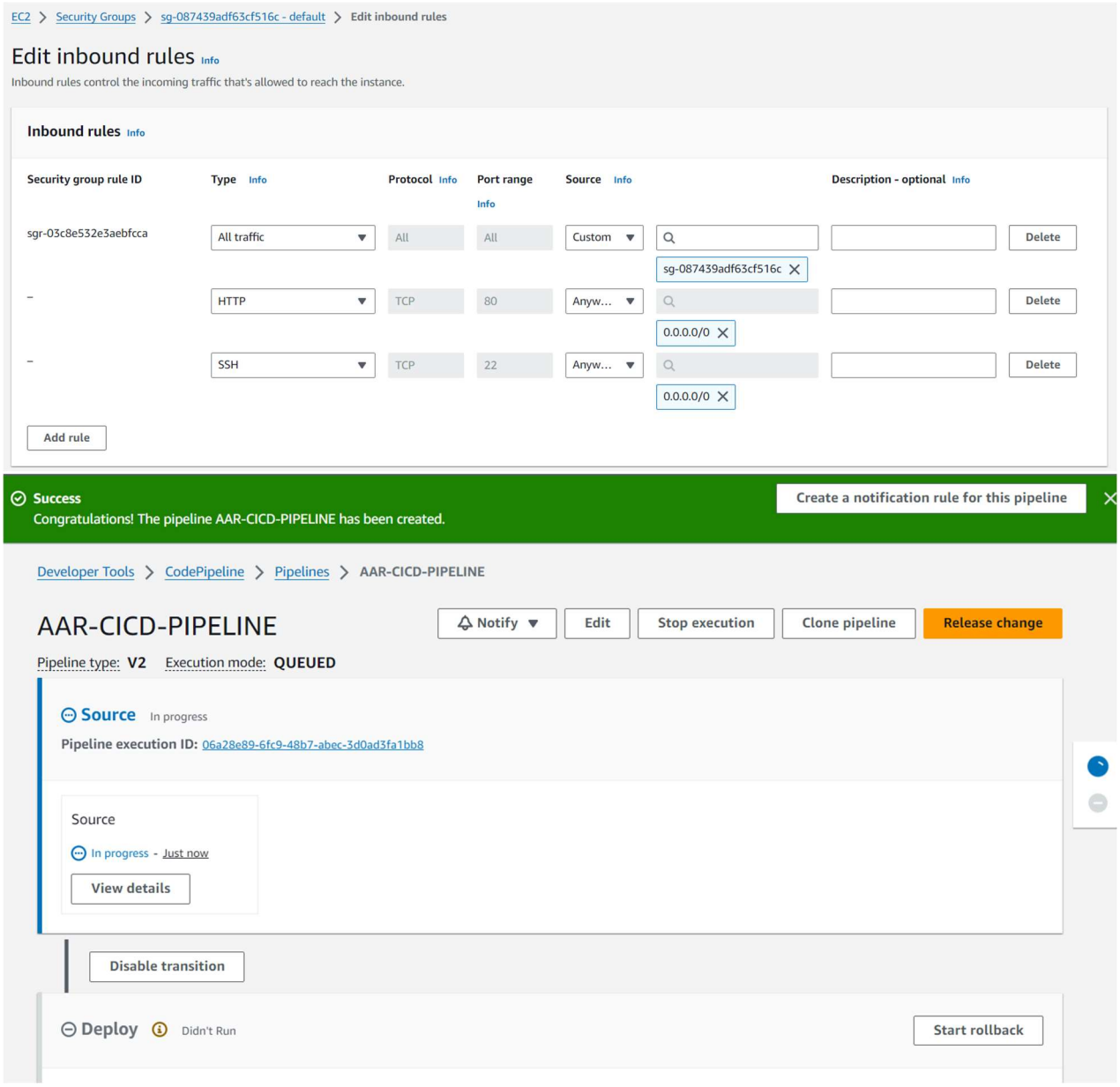
ApplicationName
AAR-CICD

DeploymentGroupName
AAR-CICD-DP

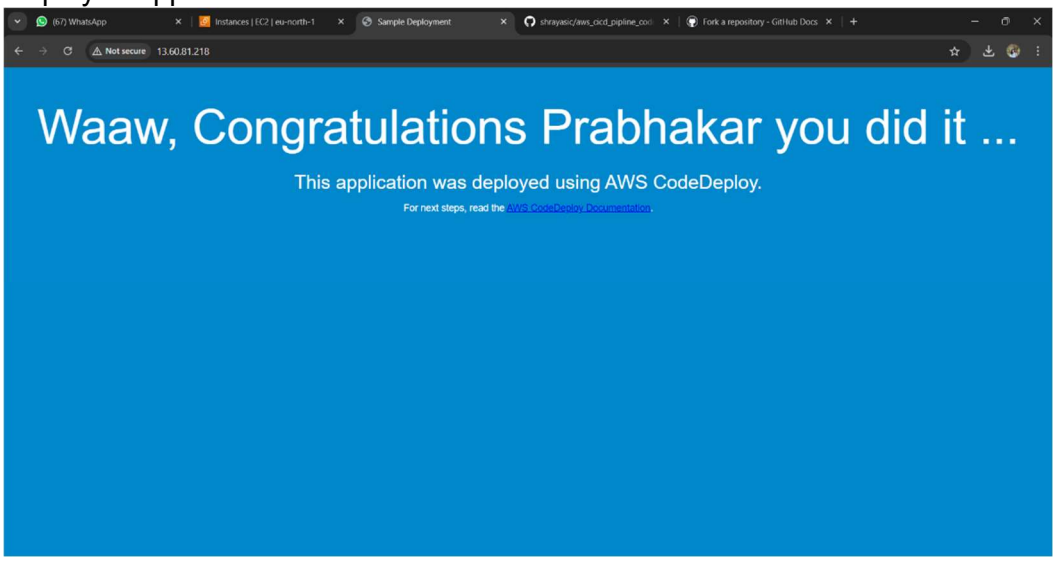
Configure automatic rollback on stage failure
Disabled

Cancel | Previous | Create pipeline

Pipeline Created



Deployed application



EXPERIMENT 3

SHRAEYAA DHAIGUDE **D15A15**

Experiment 3.

Aim: To understand the Kubernets Cluster Architechture, install and spin up a kubernets cluster on linux Machine / Cloud Platforms.

Theory:

Container-based microservices architechtures have transformed how software is developed and deployed by simplifying scaling and deployment. However, they also introduces complexity due to the new infrastructure ecosystem they create. Companies are now managing thousands of containerinstances daily, a challenge addressed by kubernets.

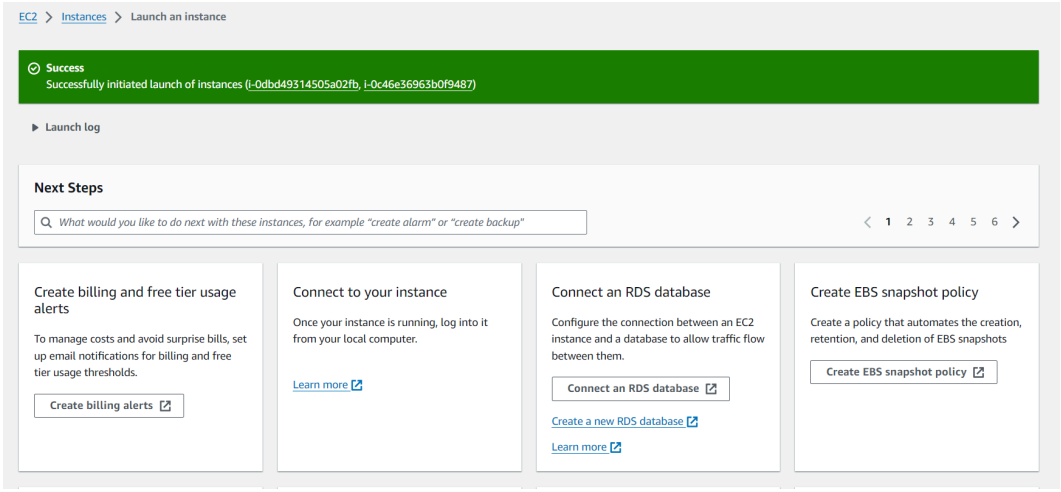
Kubernets automates the deployment, scaling and management of containerized applications. It abstracts the underlying infrastructure, allowing developers to focus on the app while kubernets handles key tasks:

- Managing resource consumption to prevent overuse
- Balancing application load across hosts
- Automatically restarting apps if they consume too many resources
- Shifting applications between hosts
- Seamlessly adding new nodes
- Enabling easy roll backs

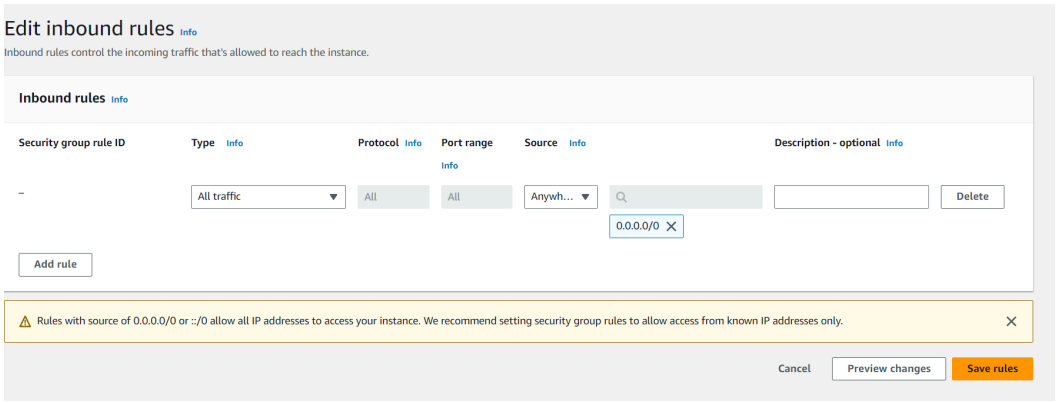
Steps:

1. Create 2 EC2 Ubuntu Instances on AWS.

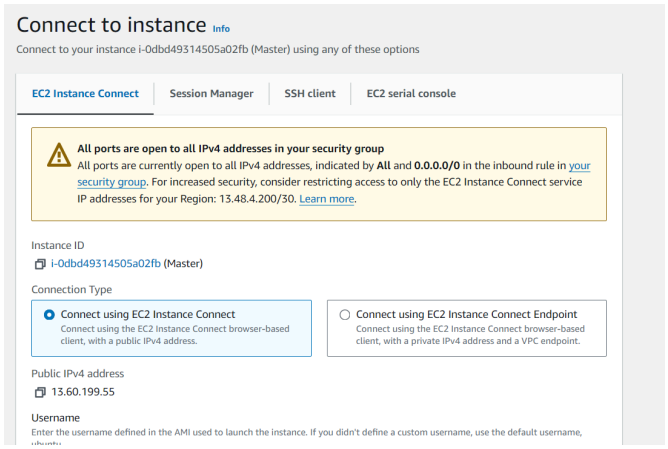
   (Name 1 as Master, the 2 as worker-node)



2. Edit the Security Group Inbound Rules to allow SSH



3. Connect Instances



4. From now on, until mentioned, perform these steps on all 3 machines.

Install Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
```

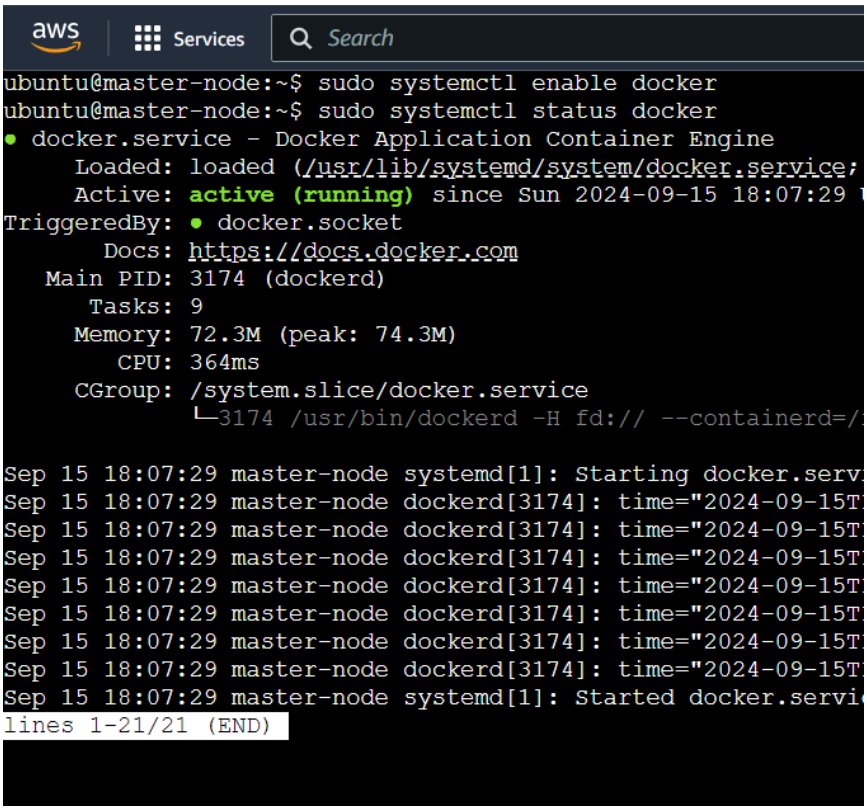
Then, configure cgroup in a daemon.json file.

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

```
ubuntu@ip-172-31-36-117:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-36-117:~$ exit
logout
```

```
aws | Services | Search
ubuntu@worker1:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InR
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports I
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 k
```

```
aws | Services | Search
ubuntu@master-node:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubun
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap dock
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz
0 upgraded, 8 newly installed, 0 to remove and 133 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```




```
ubuntu@master-node:~$ sudo apt-get install 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 packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
Need to get 900 kB of archives.
After this operation, 3072 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ sudo install -m 0755 -d /etc/apt/keyrings
ubuntu@master-node:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
ubuntu@master-node:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
>  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 1s (107 kB/s)
Reading package lists... Done
ubuntu@master-node:~$
```

```
ubuntu@master-node:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils dns-root-data dnsmasq-base ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras libltdl7 libslirp0 slirp4netns
```

Install Kubernetes on all 3 machines

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg |
sudo apt-key add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
```

```
ubuntu@master-node:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt
ubuntu@master-node:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable
es.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-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
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 1s (8955 B/s)
Reading package lists... Done
ubuntu@master-node:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@master-node:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"31", GitVersion:"v1.31.1"
"2024-09-11T21:26:49Z", GoVersion:"go1.22.6", Compiler:"gc", Platform:"lin
ubuntu@master-node:~$
```

After installing Kubernetes, we need to configure internet options to allow bridging.

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

5. Perform this **ONLY on the Master machine**

Initialize the Kubecluster

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=all
```

```
aws  Services  Search for services, features, blogs, docs, and more  [Alt+S]
ubuntu@master-node:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-erro
init] Using Kubernetes version: v1.25.2
preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the system RAM (966 MB) is less than the minimum 1700 MB
        [WARNING SystemVerification]: missing optional cgroups: blkio
preflight] Pulling images required for setting up a Kubernetes cluster
preflight] This might take a minute or two, depending on the speed of your internet connection
preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

Copy the join command and keep it in a notepad, we'll need it later.

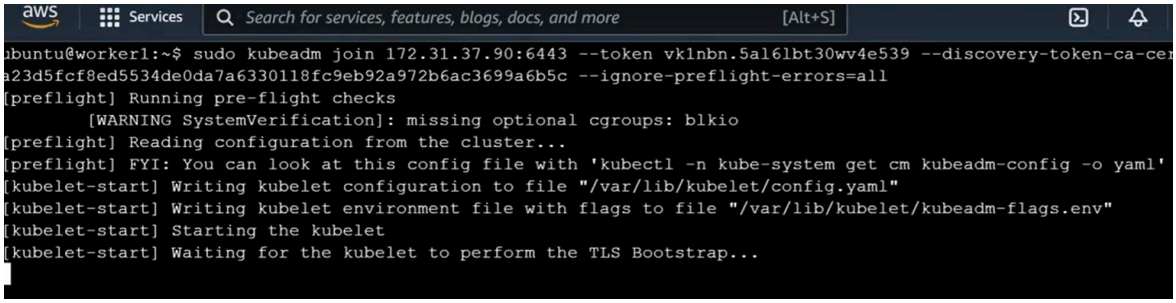
Copy the mkdir and chown commands from the top and execute them

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

kubeadm join 172.31.37.90:6443 --token vk1nbn.5al6lbt30wv4e539 \
        --discovery-token-ca-cert-hash sha256:f140f2ea454a23d5fcf8ed5534de0da7a6
ubuntu@master-node:~$ mkdir -p $HOME/.kube
ubuntu@master-node:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master-node:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master-node:~$
```

Then, add a common networking plugin called flammel file as mentioned in the code.

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



Check the created pod using this command

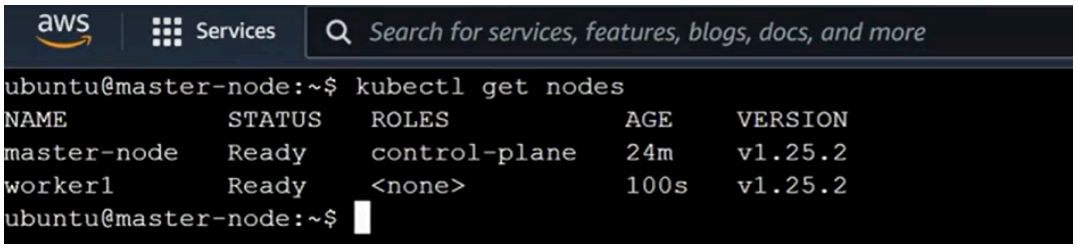
Now, keep a watch on all nodes using the following command

```
watch kubectl get nodes
```

6. Perform this **ONLY on the worker machines**

```
sudo kubeadm join <ip> --token <token> \
        --discovery-token-ca-cert-hash <hash>
```

Now, notice the changes on the master terminal



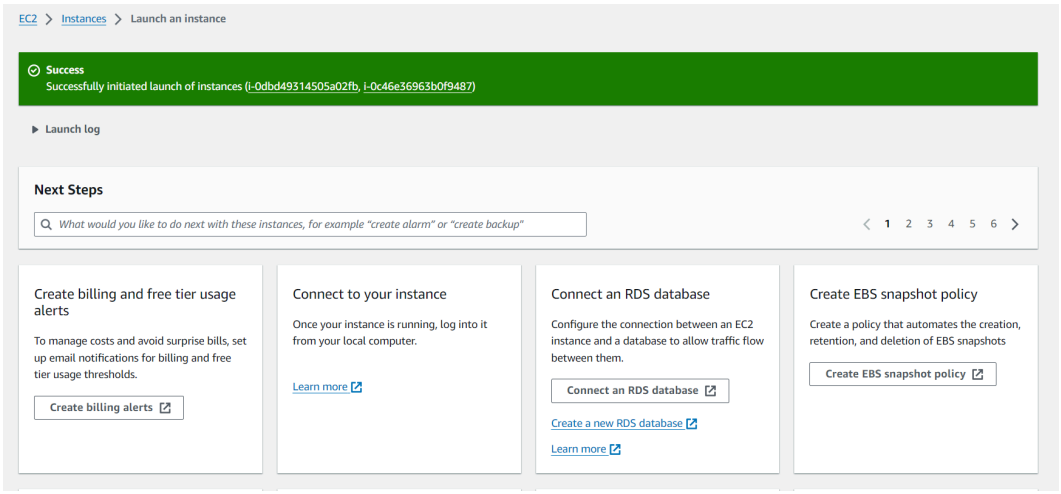
That's it, we now have a Kubernetes cluster running across 3 AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed across these machines.

Conclusion:

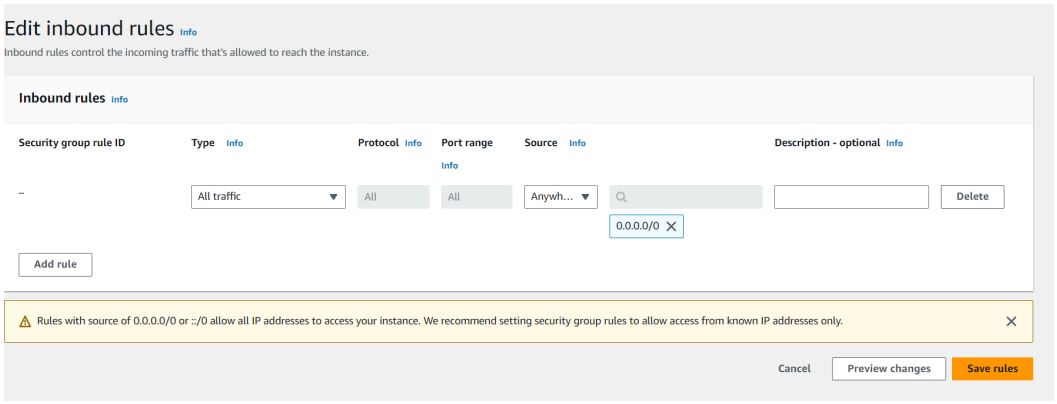
Steps:

1. Create 2 EC2 Ubuntu Instances on AWS.

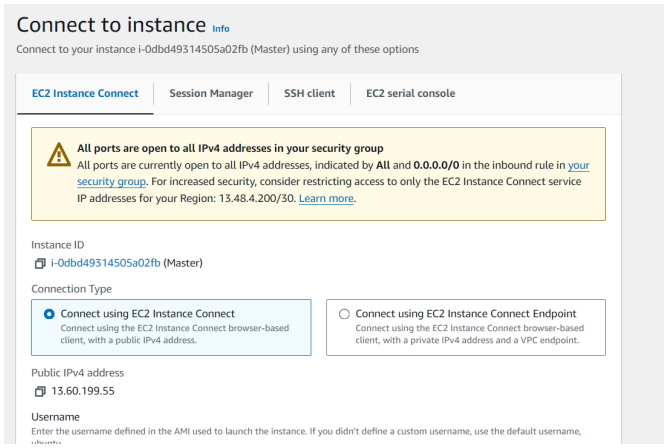
   (Name 1 as Master, the 2 as worker-node)



2. Edit the Security Group Inbound Rules to allow SSH



3. Connect Instances



4. From now on, until mentioned, perform these steps on all 3 machines.

Install Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable" sudo apt-get update
sudo apt-get install -y docker-ce
```

Then, configure cgroup in a daemon.json file.

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

```
ubuntu@ip-172-31-36-117:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-36-117:~$ exit
logout
```

aws | Services | Search

```
ubuntu@worker1:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InR
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports I
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 k
```

aws | Services | Search

```
ubuntu@master-node:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubun
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap dock
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz
0 upgraded, 8 newly installed, 0 to remove and 133 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

aws | Services | Search

```
ubuntu@master-node:~$ sudo systemctl enable docker
ubuntu@master-node:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service;
     Active: active (running) since Sun 2024-09-15 18:07:29
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3174 (dockerd)
      Tasks: 9
     Memory: 72.3M (peak: 74.3M)
        CPU: 364ms
     CGroup: /system.slice/docker.service
             └─3174 /usr/bin/dockerd -H fd:// --containerd=/

Sep 15 18:07:29 master-node systemd[1]: Starting docker.servi
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node dockerd[3174]: time="2024-09-15T
Sep 15 18:07:29 master-node systemd[1]: Started docker.servi
lines 1-21/21 (END)
```

```
ubuntu@master-node:~$ sudo apt-get install 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 packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
Need to get 900 kB of archives.
After this operation, 3072 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ sudo install -m 0755 -d /etc/apt/keyrings
ubuntu@master-node:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
ubuntu@master-node:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
>  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 1s (107 kB/s)
Reading package lists... Done
ubuntu@master-node:~$
```

```
ubuntu@master-node:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils dns-root-data dnsmasq-base ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras libltdl7 libslirp0 slirp4netns
```

Install Kubernetes on all 3 machines

```
curl -s
https://packages.cloud.google.com/apt/doc/apt-key.gpg |
sudo apt-key add -
cat << EOF | sudo tee
/etc/apt/sources.list.d/kubernetes.list deb
https://apt.kubernetes.io/ kubernetes-xenial main EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
```

```
ubuntu@master-node:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt
ubuntu@master-node:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable
es.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-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
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 1s (8955 B/s)
Reading package lists... Done
ubuntu@master-node:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@master-node:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"31", GitVersion:"v1.31.1"
"2024-09-11T21:26:49Z", GoVersion:"go1.22.6", Compiler:"gc", Platform:"lin
ubuntu@master-node:~$
```

After installing Kubernetes, we need to configure internet options to allow bridging.

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

5. Perform this **ONLY on the Master machine**

Initialize the Kubecluster

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=all
```

```
aws  Services  Search for services, features, blogs, docs, and more  [Alt+S]
ubuntu@master-node:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-erro
init] Using Kubernetes version: v1.25.2
preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the system RAM (966 MB) is less than the minimum 1700 MB
        [WARNING SystemVerification]: missing optional cgroups: blkio
preflight] Pulling images required for setting up a Kubernetes cluster
preflight] This might take a minute or two, depending on the speed of your internet connection
preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

Copy the join command and keep it in a notepad, we'll need it later.

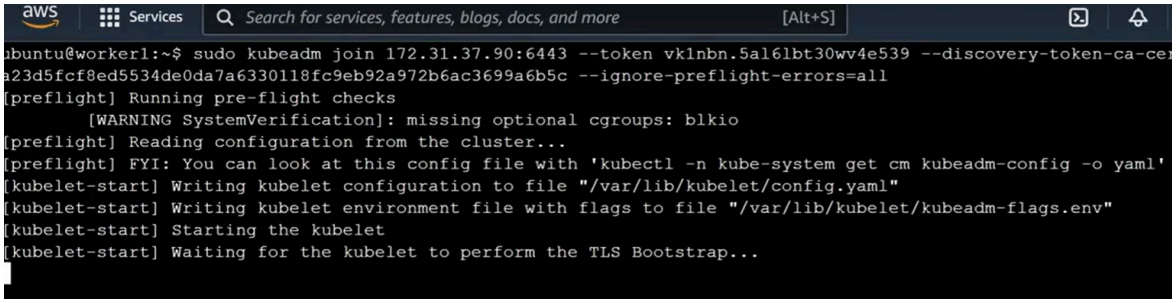
Copy the mkdir and chown commands from the top and execute them

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

kubeadm join 172.31.37.90:6443 --token vk1nbn.5al6lbt30wv4e539 \
        --discovery-token-ca-cert-hash sha256:f140f2ea454a23d5fcf8ed5534de0da7a6
ubuntu@master-node:~$ mkdir -p $HOME/.kube
ubuntu@master-node:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master-node:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master-node:~$
```

Then, add a common networking plugin called flammel file as mentioned in the code.

```
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documen
tation/ k ube-flannel.yml
```



Check the created pod using this command

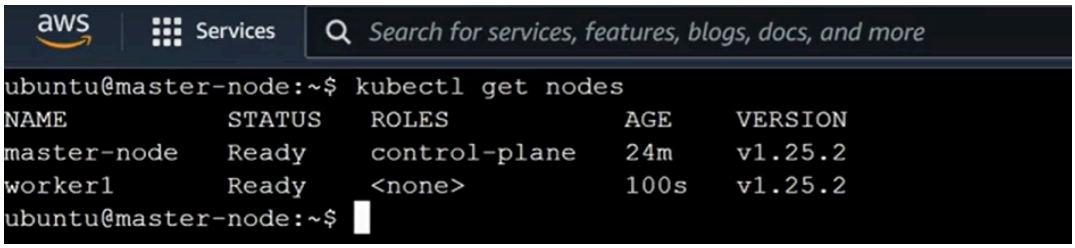
Now, keep a watch on all nodes using the following command

```
watch kubectl get nodes
```

6. Perform this **ONLY on the worker machines**

```
sudo kubeadm join <ip> --token <token> \
        --discovery-token-ca-cert-hash <hash>
```

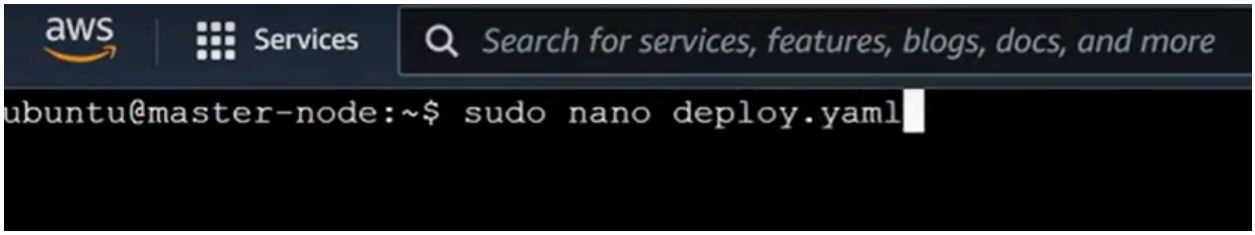
Now, notice the changes on the master terminal

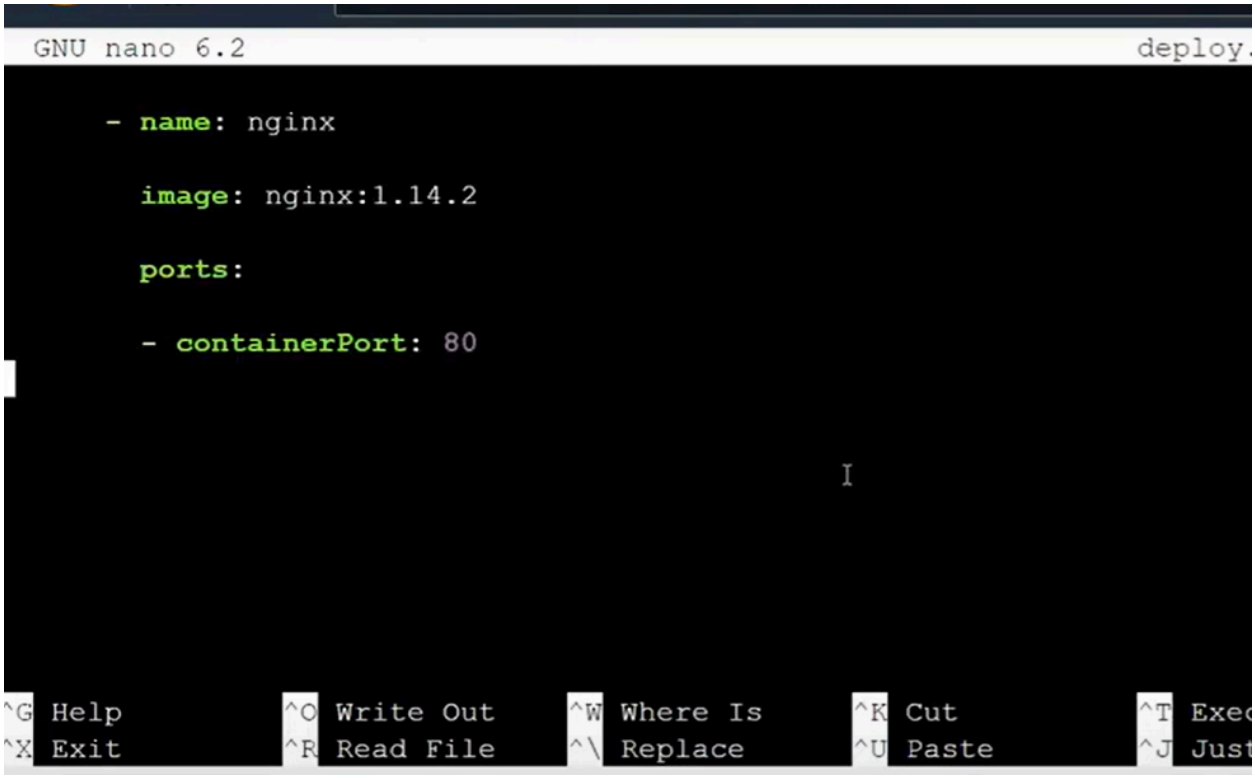


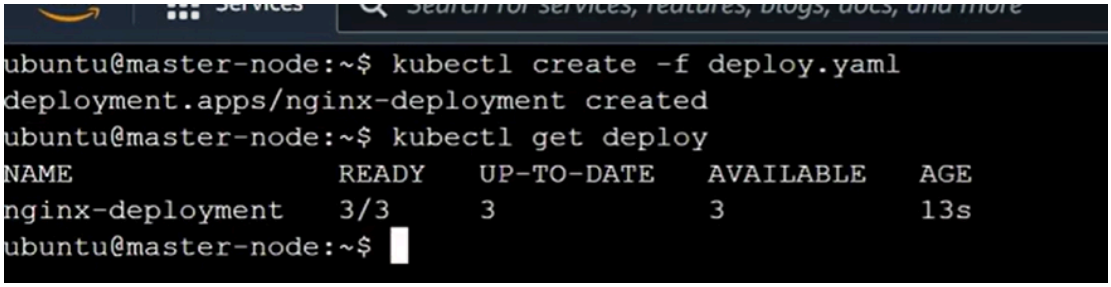
That's it, we now have a Kubernetes cluster running across 3 AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed across these machines.

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

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







Use ‘kubectl get pods’ to verify if the deployment was properly created and the pod is working correctly.

Next up, create a name alias for this pod.

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

8. Lastly, port forward the deployment to your localhost so that you can

view it. `kubectl port-forward $POD_NAME 8080:80`

```
ubuntu@master-node:~$ kubectl expose deployment.apps/nginx-deployment --type="LoadBalancer"
service/nginx-deployment exposed
ubuntu@master-node:~$ kubectl get svc
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes         ClusterIP      10.96.0.1      <none>        443/TCP        50m
nginx-deployment   LoadBalancer   10.111.72.92   <pending>     80:31642/TCP   13s
ubuntu@master-node:~$
```

9. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.


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

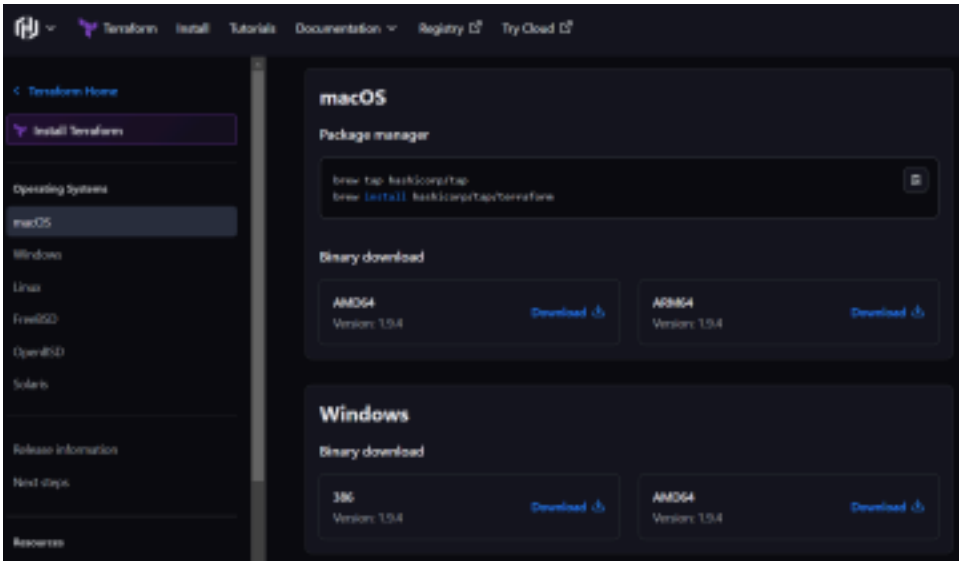
If the response is 200 OK and you can see the Nginx server name, your deployment was successful.

We have successfully deployed our Nginx server on our EC2 instance.

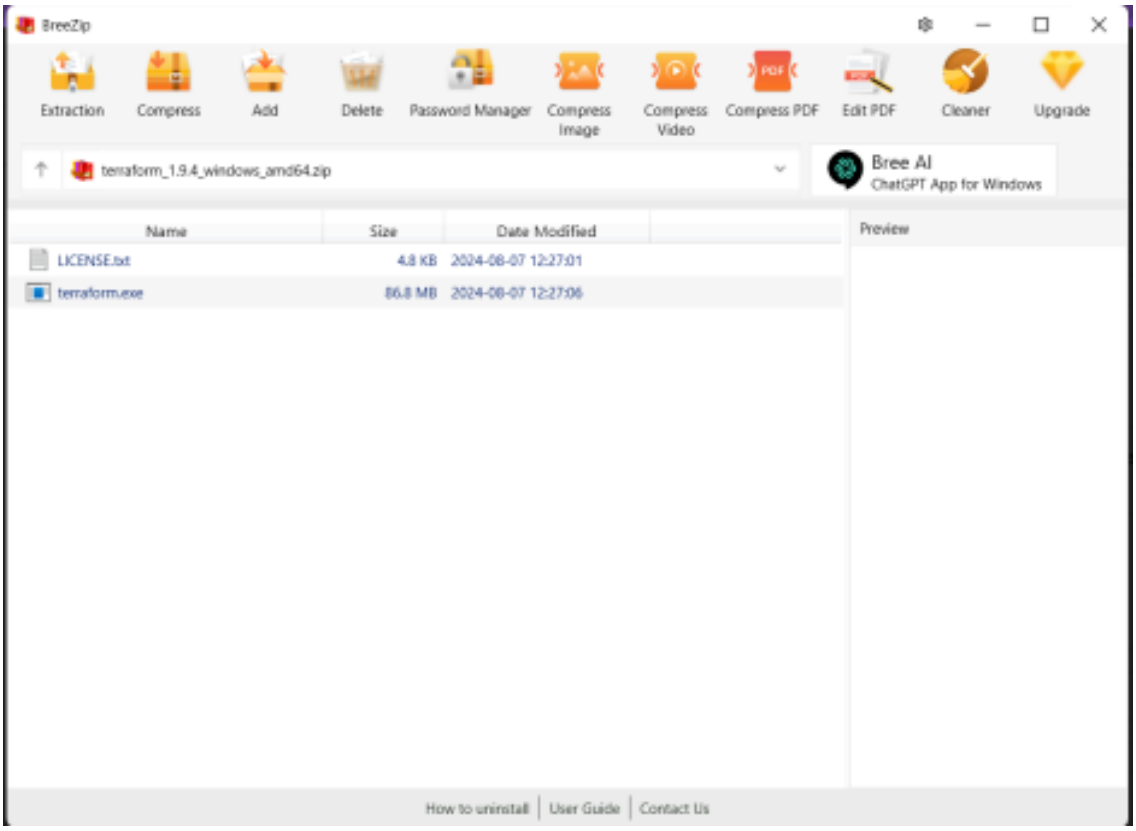
Conclusion:

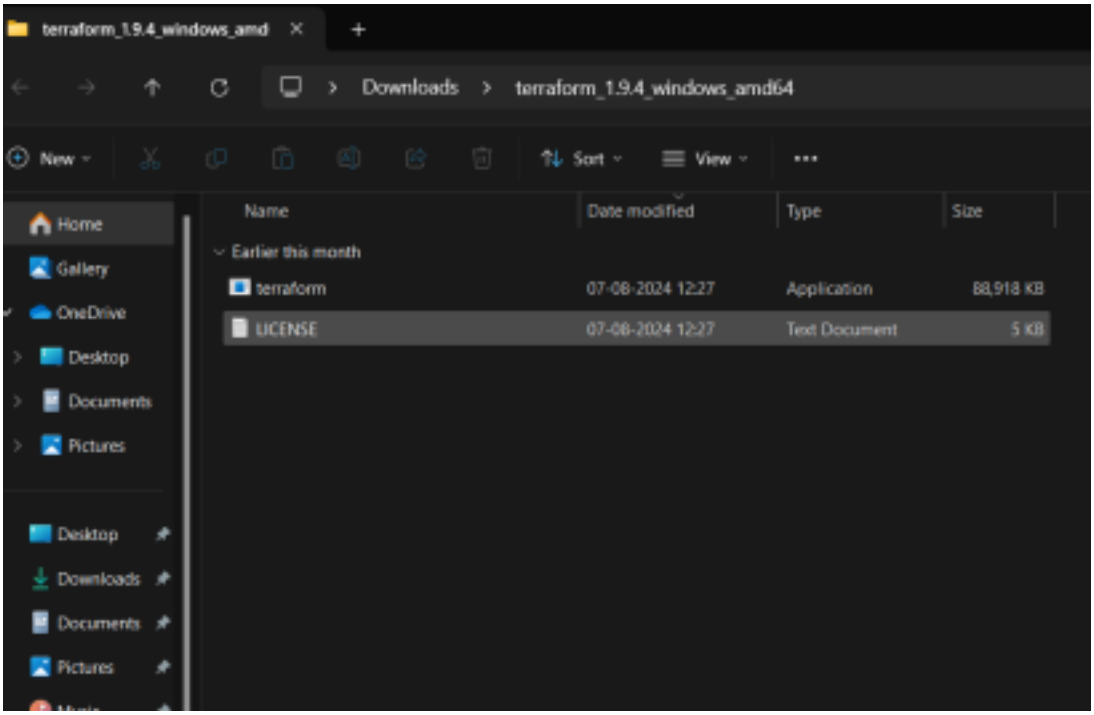
ADVDEVOPS EXP 5

A) Installation and Configuration of Terraform in Windows
Step 1: Download terraform
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website
website:https://www.terraform.io/downloads.html
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.

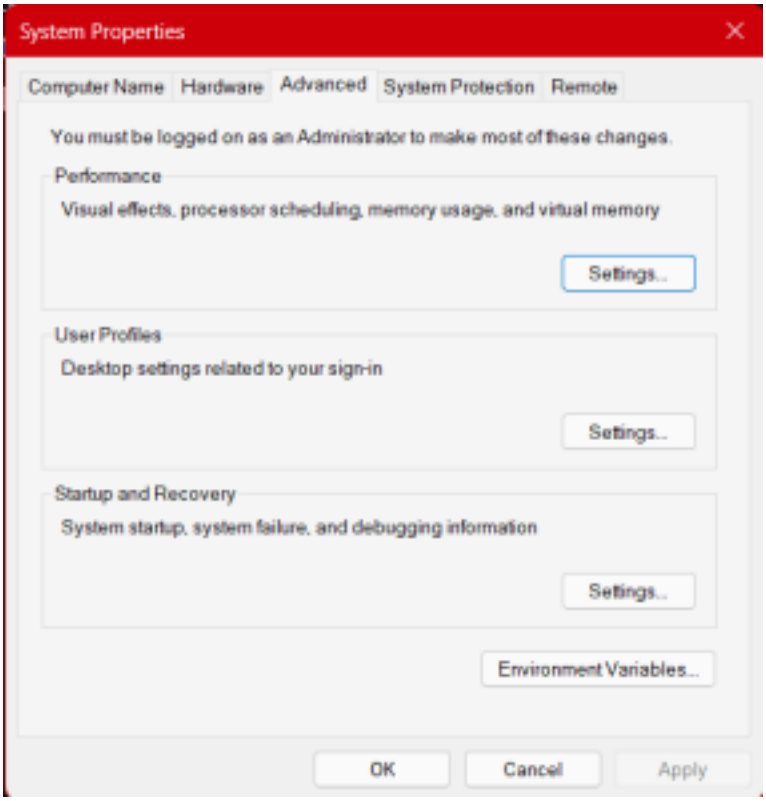


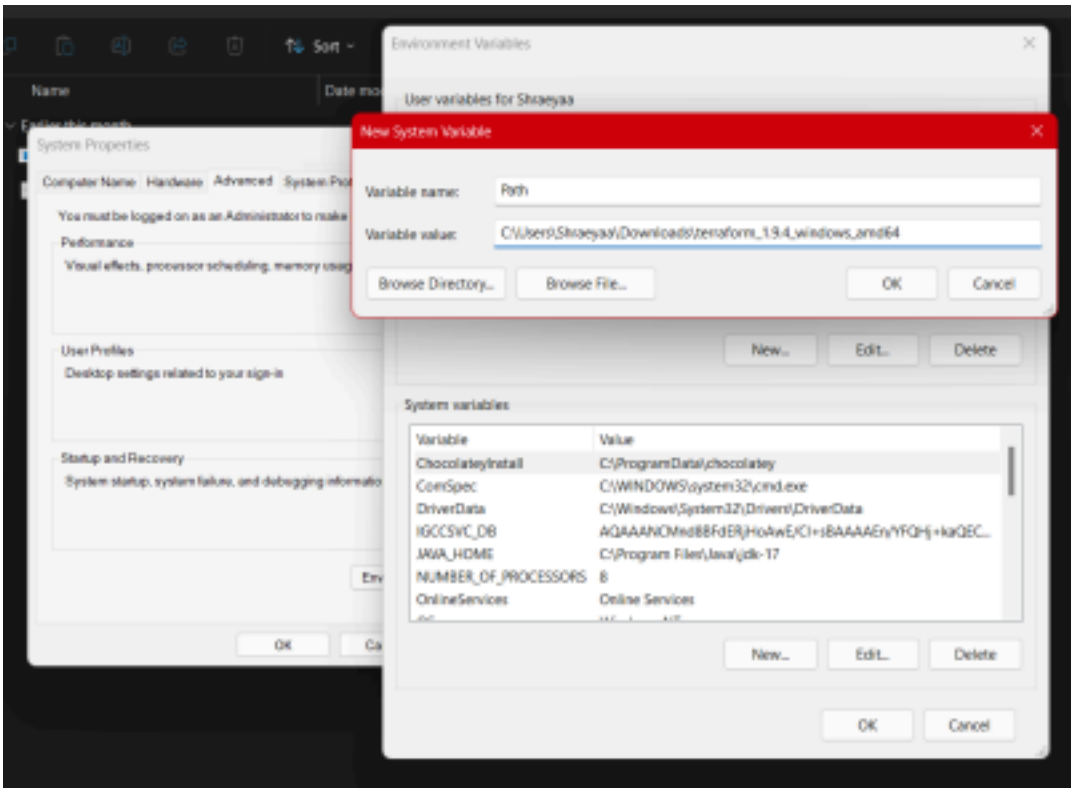
Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory

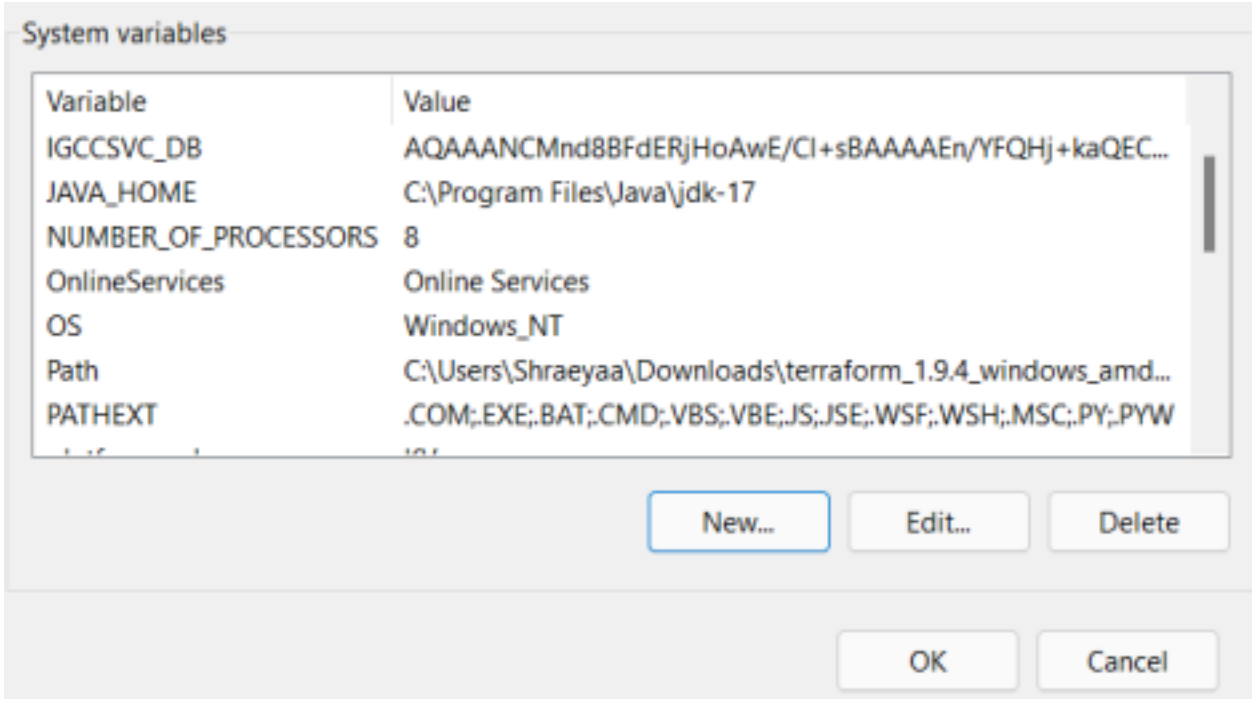




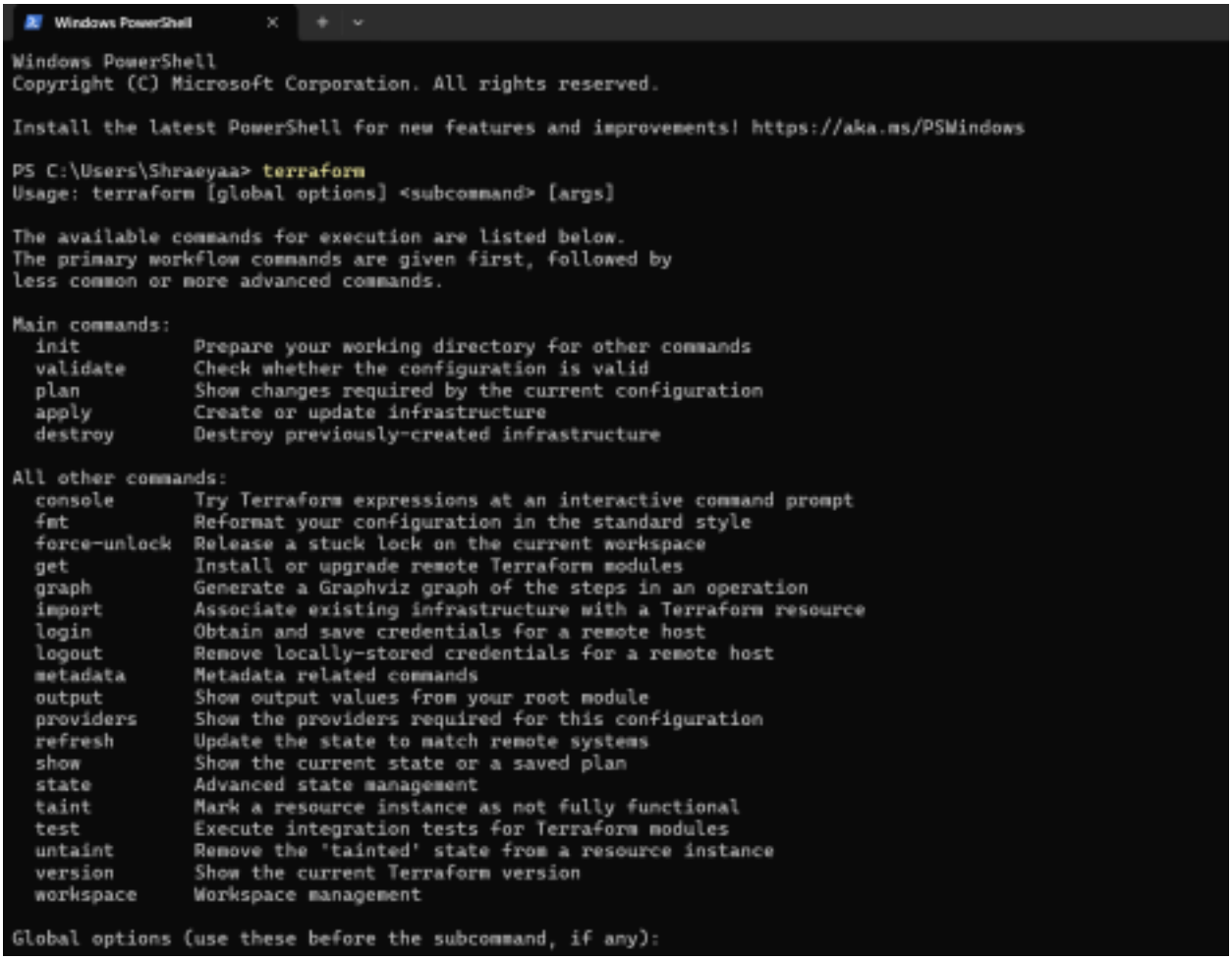
Step 3: Set the System path for Terraform in Environment Variables




Environment Variables
User variables for Shraeyaa
New System Variable
Variable name: Path
Variable value: C:\Users\Shraeyaa\Downloads\terraform_1.9.4_windows_amd64
Browse Directory...
Browse File...
OK
Cancel
New...
Edit...
Delete
System variables
Variable
Value
ChocolateyInstall
C:\ProgramData\chocolatey
ComSpec
C:\WINDOWS\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
IGCCSVC_DB
AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAEn/YFQHj+kaQEC...
JAVA_HOME
C:\Program Files\Java\jdk-17
NUMBER_OF_PROCESSORS
8
OnlineServices
Online Services
New...
Edit...
Delete
OK
Cancel


System variables
Variable
Value
IGCCSVC_DB
AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAEn/YFQHj+kaQEC...
JAVA_HOME
C:\Program Files\Java\jdk-17
NUMBER_OF_PROCESSORS
8
OnlineServices
Online Services
OS
Windows_NT
Path
C:\Users\Shraeyaa\Downloads\terraform_1.9.4_windows_amd...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
New...
Edit...
Delete
OK
Cancel

Step 4:: Open Terraform in PowerShell and check its functionality



ADVANCE DEVOPS EXP 6

Docker Installation:

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

Before using terraform commands:

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

```
C:\Terraform scripts\Docker>docker container list
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Terraform scripts\Docker>
```

Code:

docker.tf

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

Terraform Commands:

```
PS C:\Terraform scripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 3.0.1"...
- Installing kreuzwerker/docker v3.0.2...
- Installed kreuzwerker/docker v3.0.2 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!
```

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

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

Terraform will perform the following actions:

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

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

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

Terraform will perform the following actions:

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

```
  Enter a value: yes

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

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

After using Terraform commands:

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

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

C:\Terraform scripts\Docker>
```

To Delete the Containers created:

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

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

Terraform will perform the following actions:

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

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

Prerequisites:

1) Docker

Run docker -v command.

Use this command to check if docker is installed and running on your system.

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

2) Install SonarQube image Command: docker pull sonarqube

This command helps you to install an image of SonarQube that can be used on the local system without actually installing the SonarQube installer.

```
C:\Users\Shraeyaa>docker pull sonarqube
Using default tag: latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
error during connect: Post "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.46/images/create?fromImage=sonarqube&tag=latest": open //./pipe/dockerDeskto
pLinuxEngine: The system cannot find the file specified.

C:\Users\Shraeyaa>
```

3) Keep jenkins installed on your system.

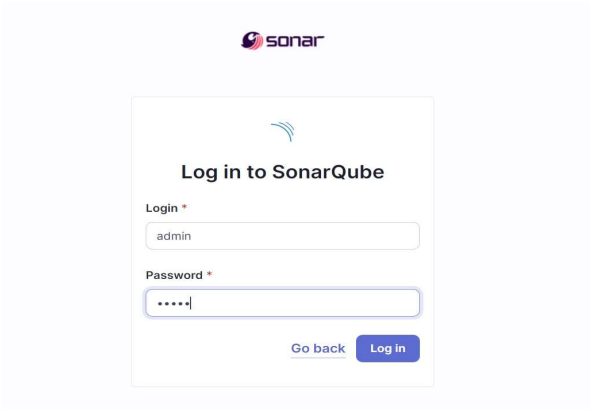
Experiment Steps:

1. 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

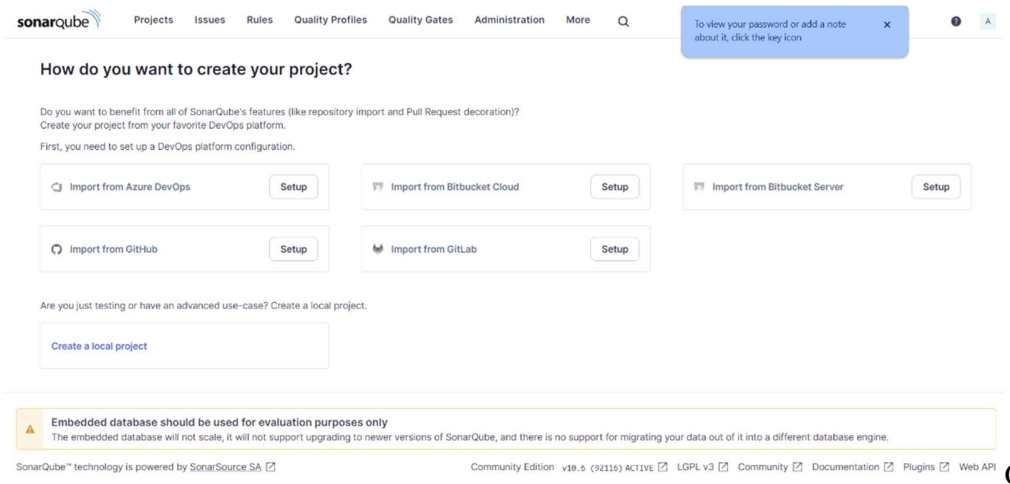
```
C:\Users\Shraeyaa>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
4be9b2af1602f07d33c1e6f4fbcc35a8f70386b231a43669a03e4e134b355058

C:\Users\Shraeyaa>
```

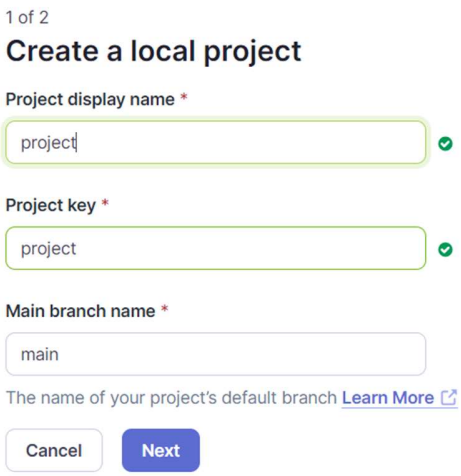
2. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
3. Login to SonarQube using username admin and password admin.



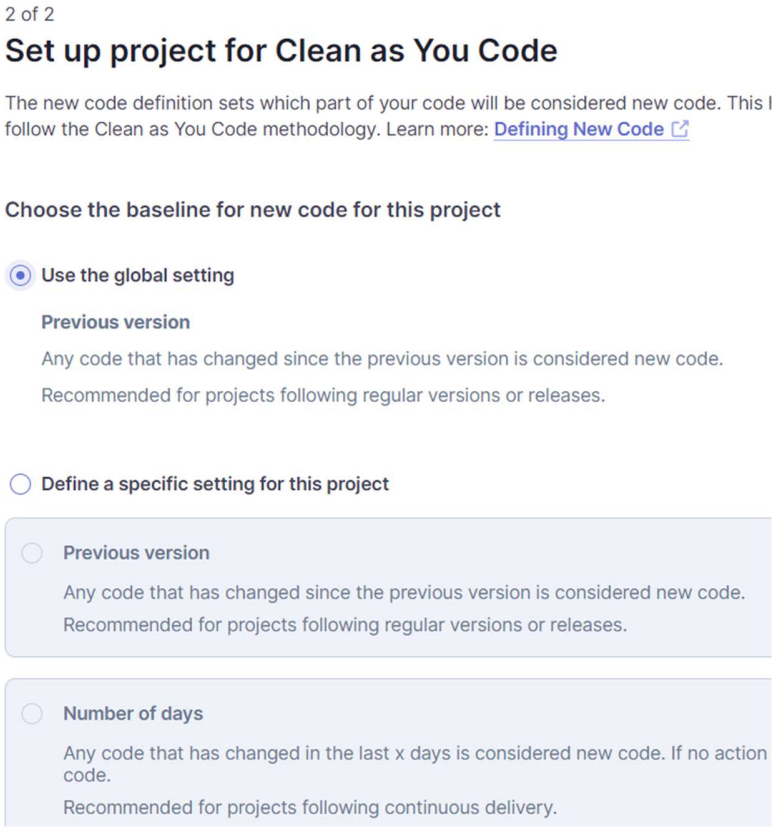
4. Create a manual project in SonarQube with the name sonarqube.



Click on Create a Local Project.

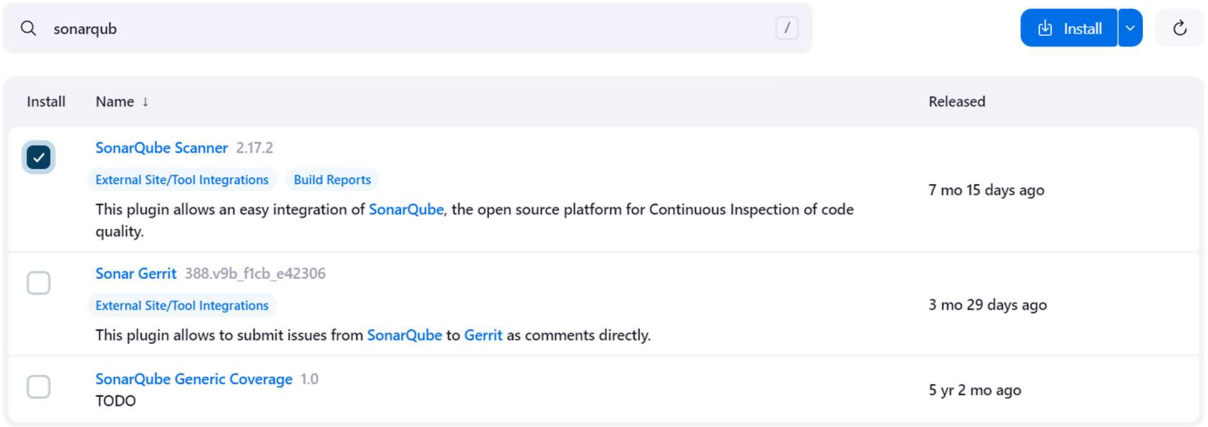


Set up the project as required and click on create.



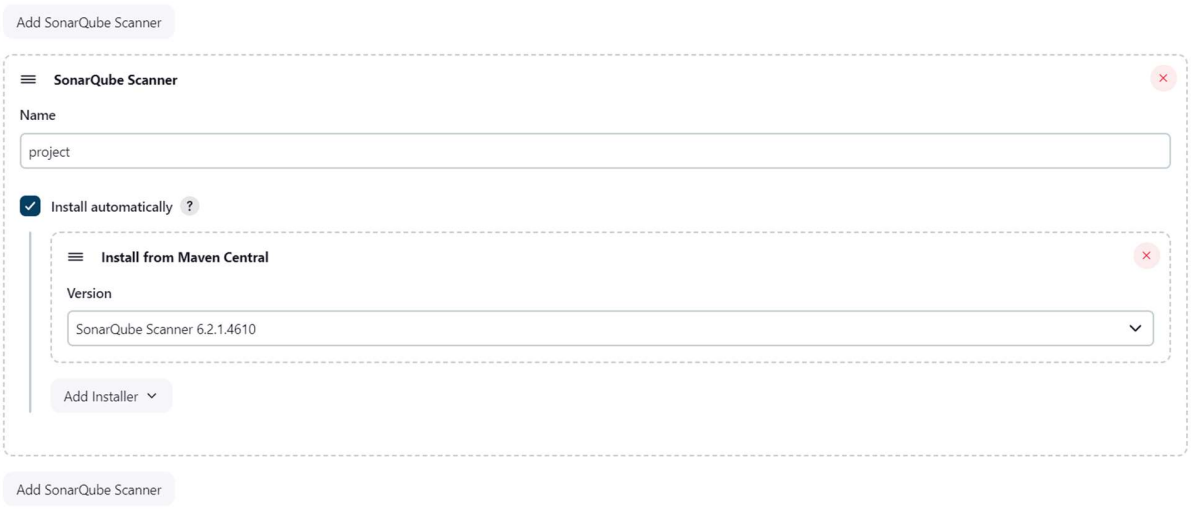
5. Open Jenkins on whichever port it is installed. (http://loaclhost:<port number>).

6. Go to Manage jenkins → Plugins → Available Plugins
   Search for Sonarqube Scanner for Jenkins and install it.

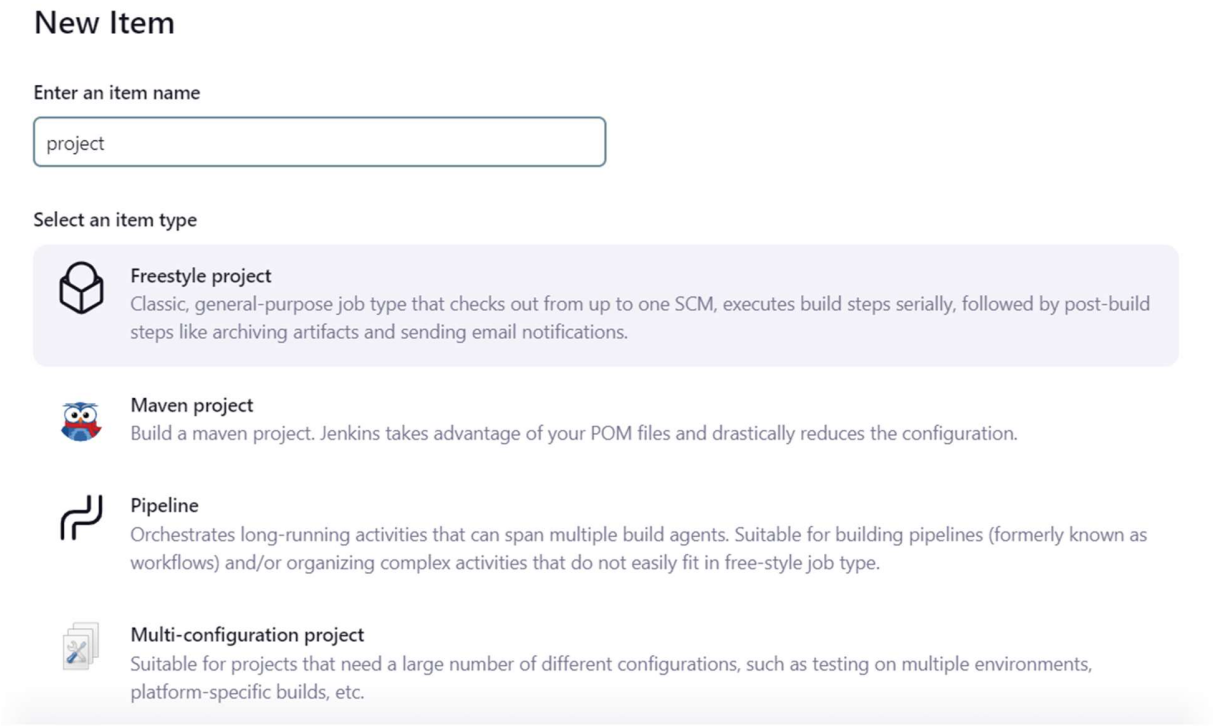


7. Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.
   I have named the server ‘sonarqube’ and added the server url for jenkins.
   Enter the Server Authentication token if needed.

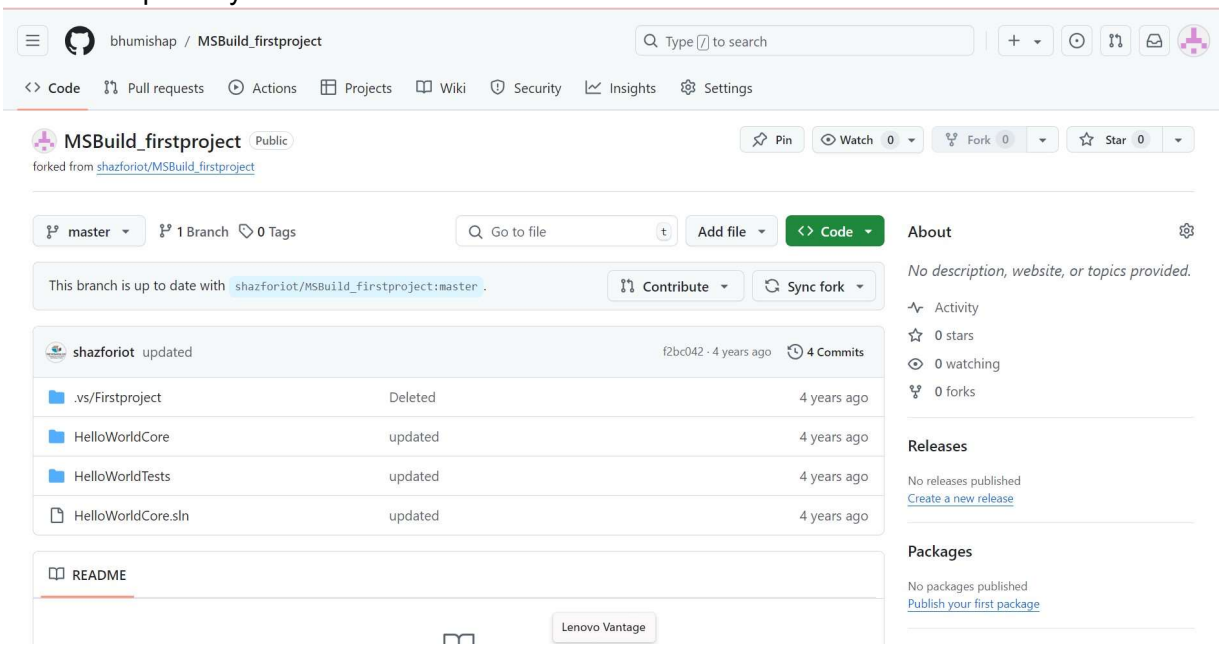
8. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.



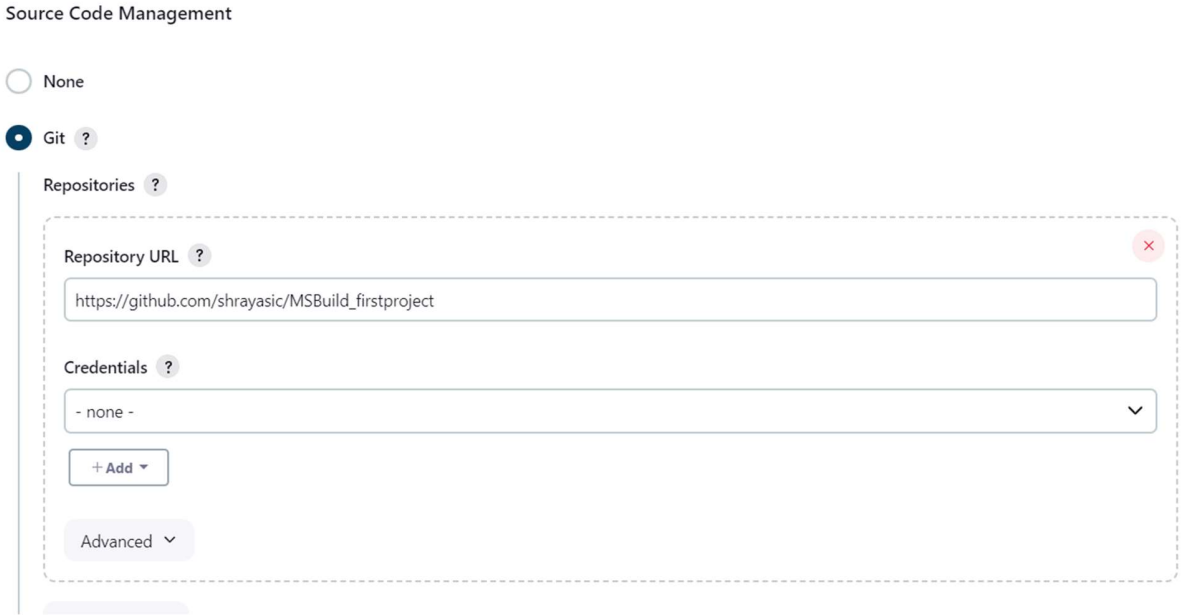
9. After configuration, create a New Item → choose a freestyle project and name your project. Here, I have given the ‘sonarqube’, then click on OK.



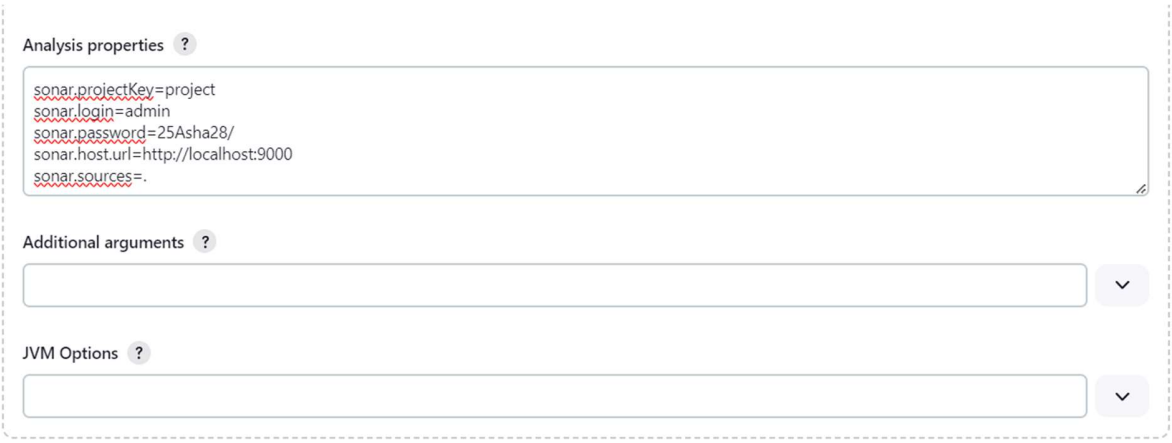
10. Choose the github repository https://github.com/shazforiot/MSBuild_firstproject. It is a sample hello-world project with no vulnerabilities and issues, just to test the integration. Fork this repository.



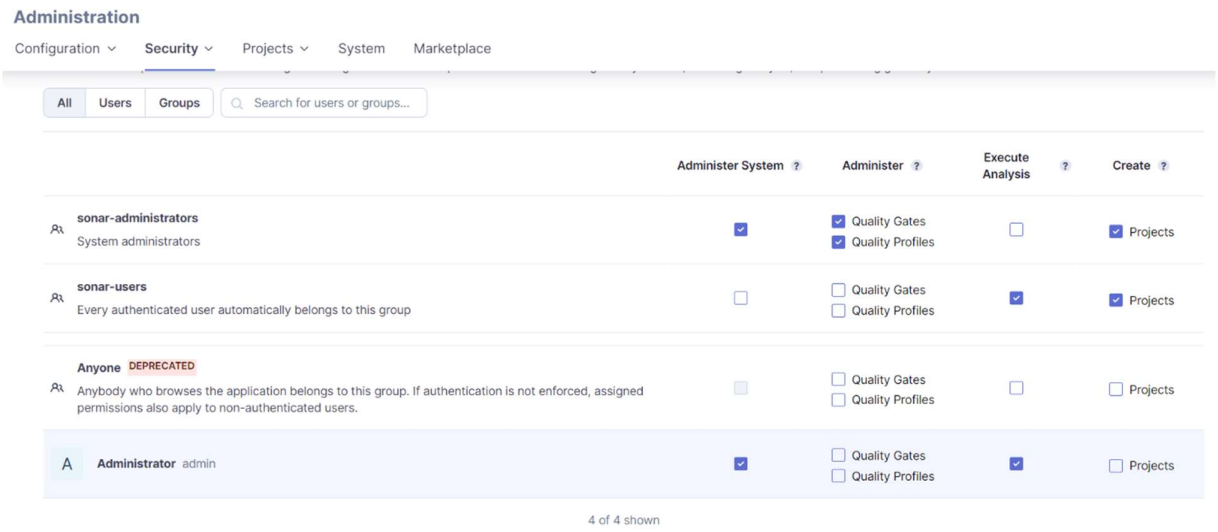
11. Enable git and add the repository you forked.



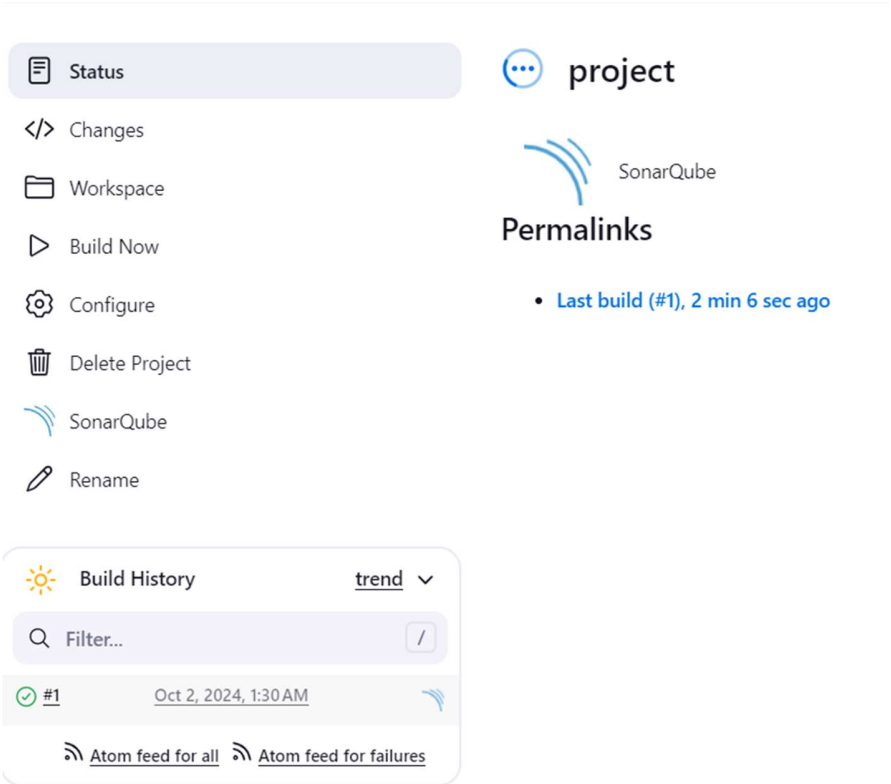
12. Under Build Steps, select Execute Sonarqube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.



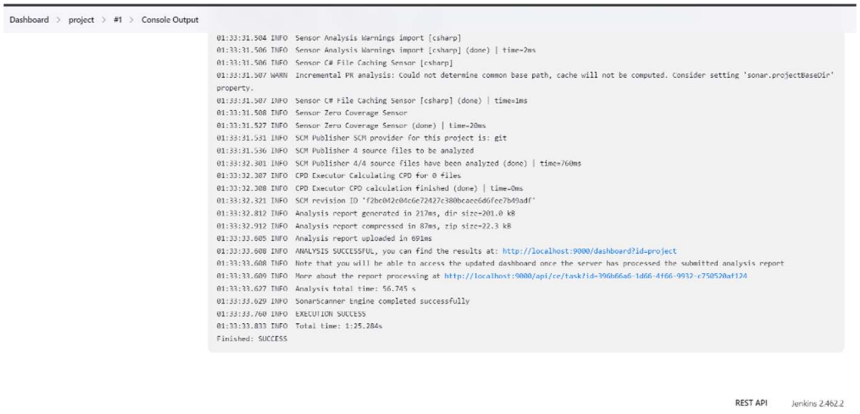
13. Then click on save.Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.



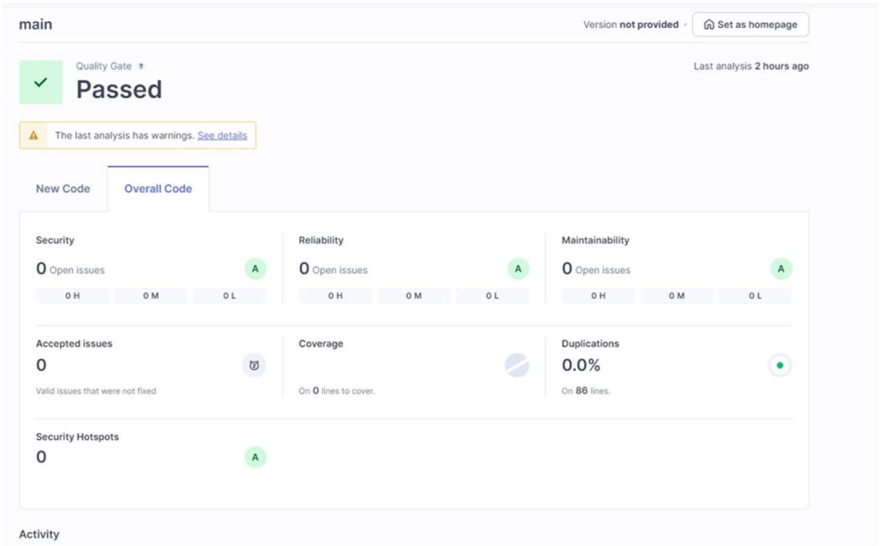
14. Go back to jenkins. Go to the job you had just built and click on Build Now.



15. Check the console Output.



16. Once the build is complete, go back to SonarQube and check the project linked.



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

Prerequisites:

1) Docker

Run docker -v command.

Use this command to check if docker is installed and running on your system.

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

2) Install SonarQube image Command: docker pull sonarqube

This command helps you to install an image of SonarQube that can be used on the local system without actually installing the SonarQube installer.

```
C:\Users\Shraeyaa>docker pull sonarqube
Using default tag: latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
error during connect: Post "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.46/images/create?fromImage=sonarqube&tag=latest": open //./pipe/dockerDeskto
pLinuxEngine: The system cannot find the file specified.

C:\Users\Shraeyaa>
```

3) Keep jenkins installed on your system.

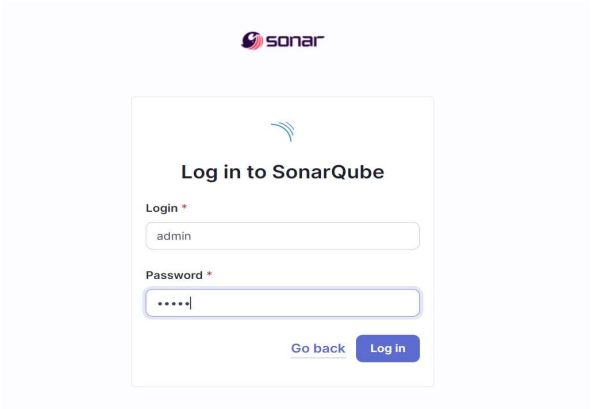
Experiment Steps:

1. 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

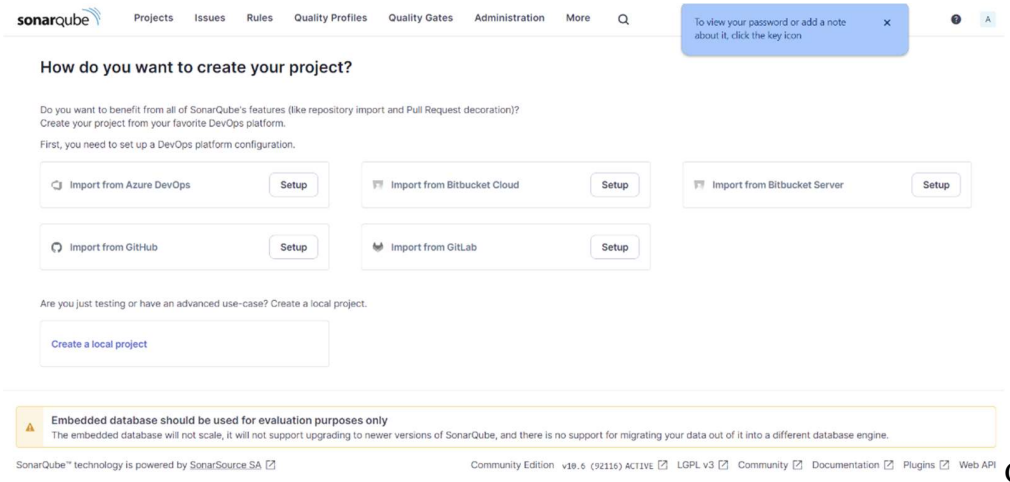
```
C:\Users\Shraeyaa>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
4be9b2af1602f07d33c1e6f4fbcc35a8f70386b231a43669a03e4e134b355058

C:\Users\Shraeyaa>
```

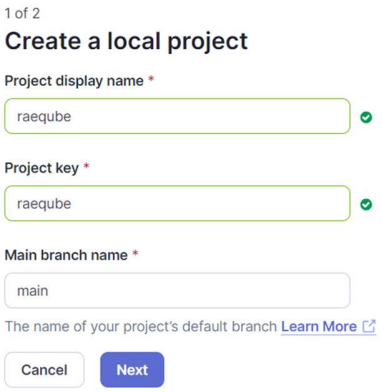
2. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
3. Login to SonarQube using username admin and password admin.



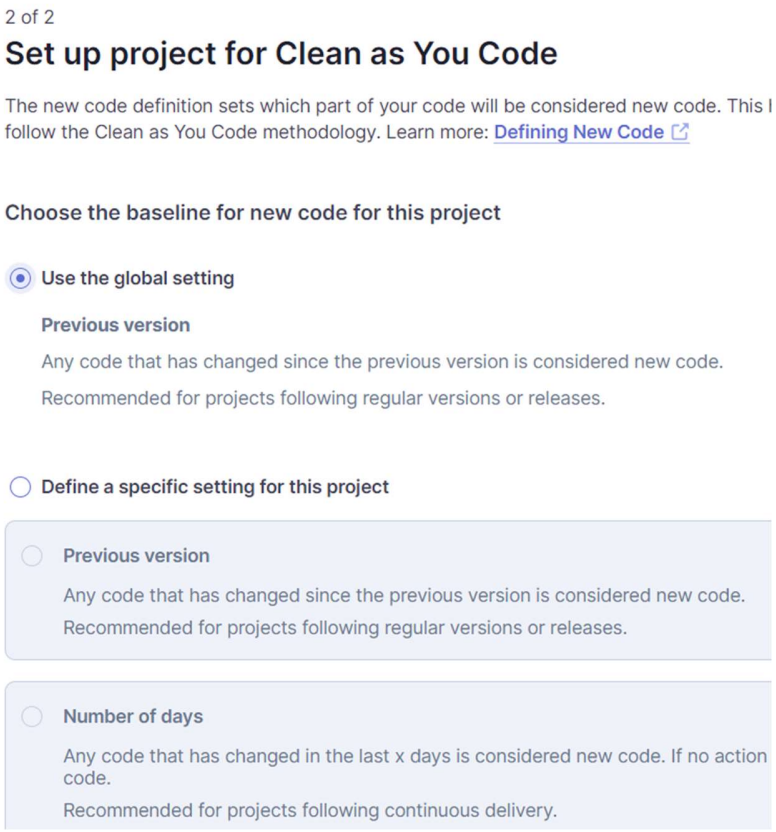
4. Create a manual project in SonarQube with the name sonarqube.



Click on Create a Local Project.

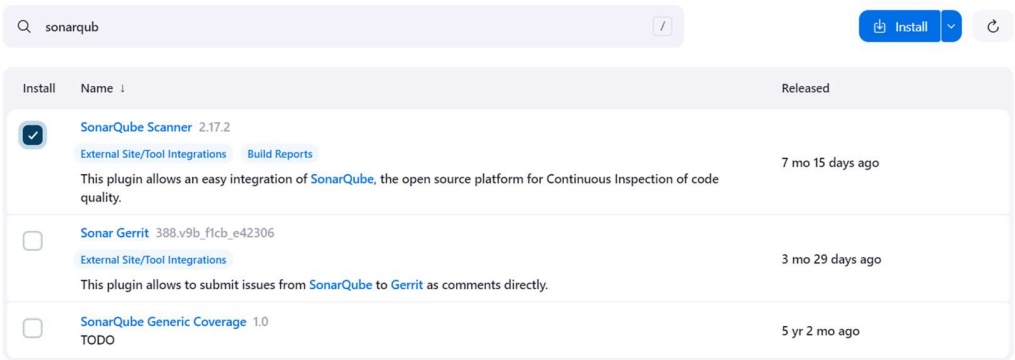


Set up the project as required and click on create.

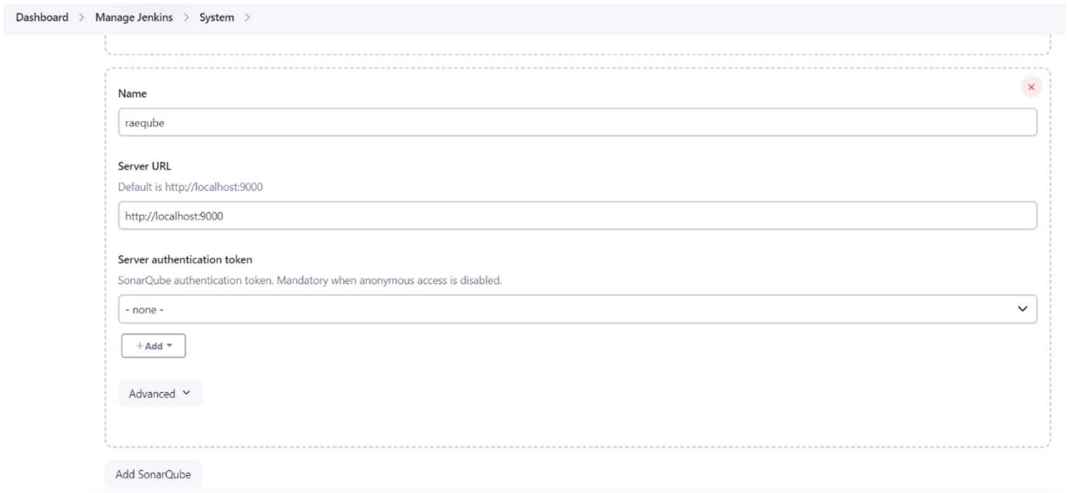


5. Open Jenkins on whichever port it is installed. (http://loaclhost:<port number>).

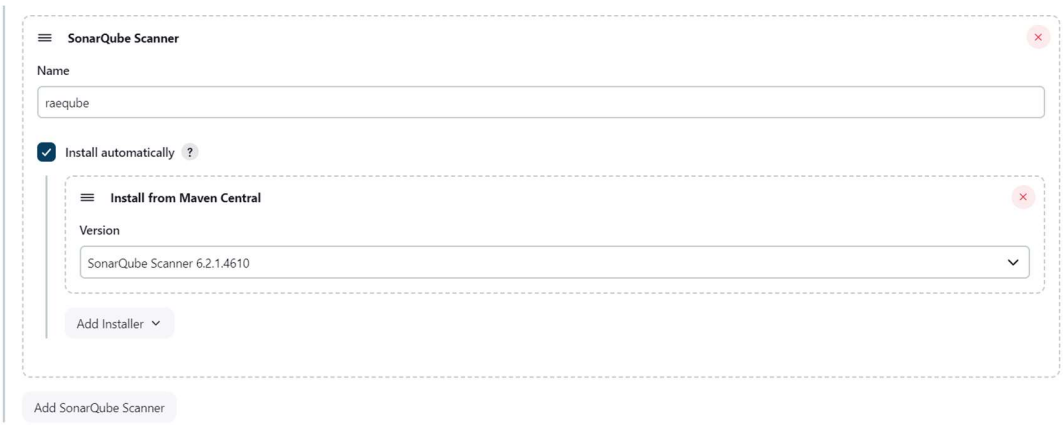
6. Go to Manage jenkins → Plugins → Available Plugins
   Search for Sonarqube Scanner for Jenkins and install it.



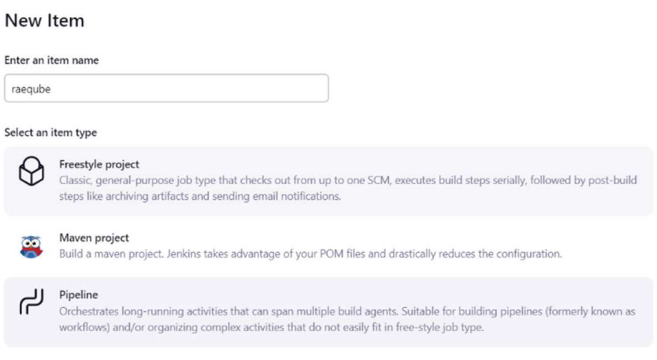
7. Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.
   I have named the server ‘sonarqube’ and added the server url for jenkins.
   Enter the Server Authentication token if needed.



8. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

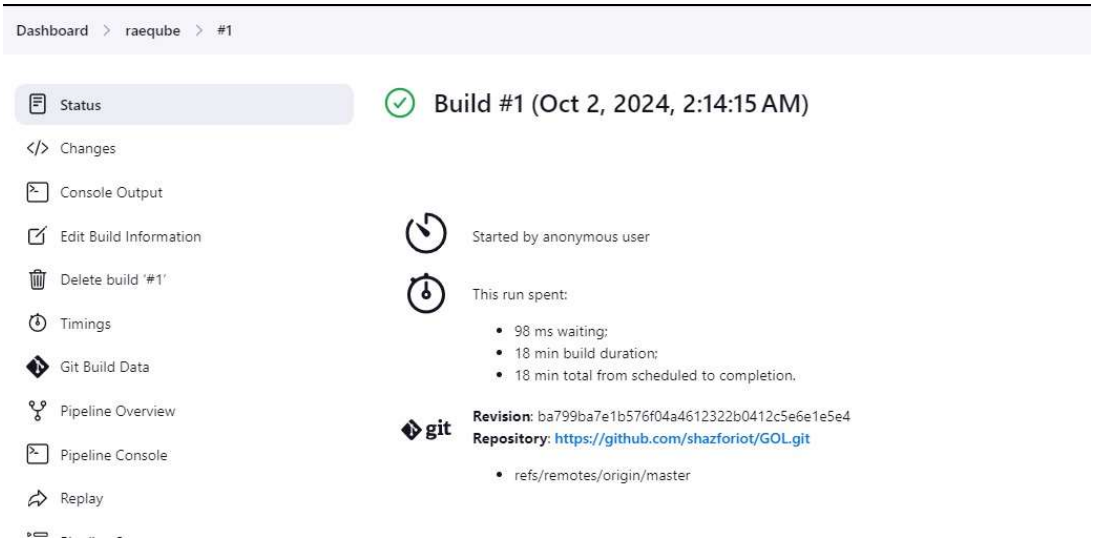


9. After configuration, create a New Item → choose a pipeline project.

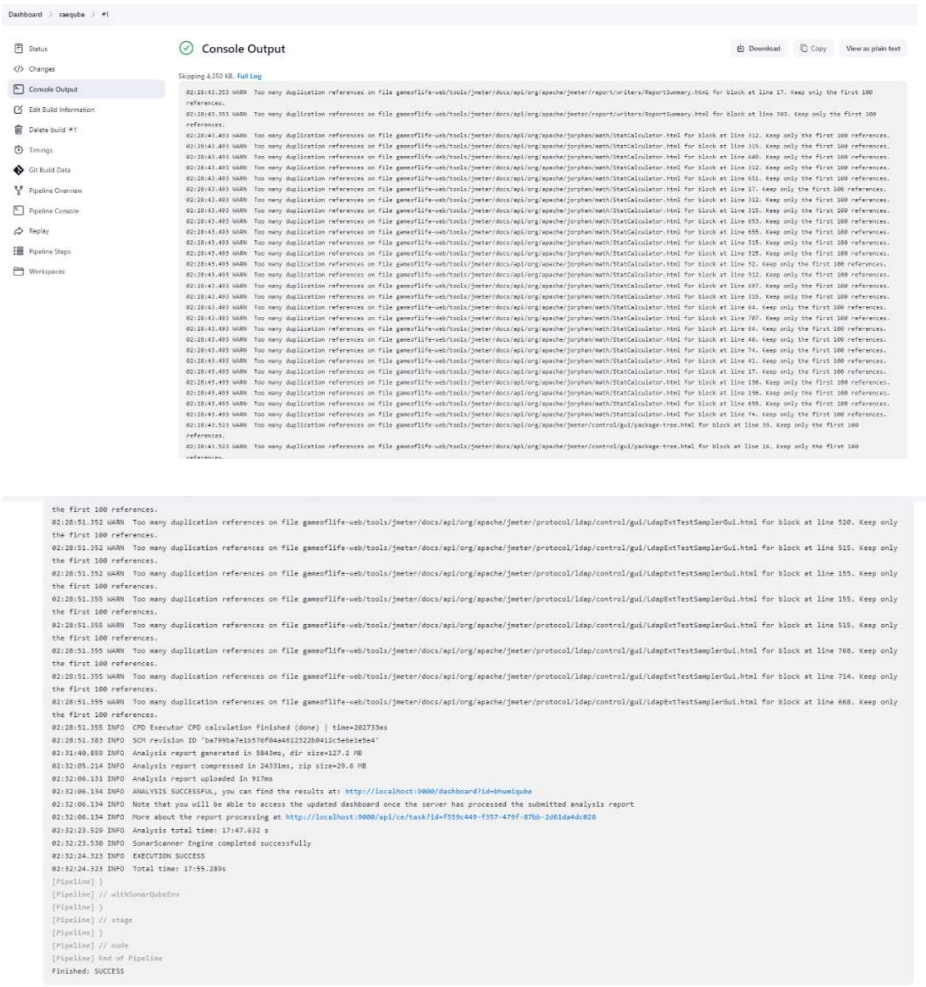


10.Under Pipeline script, enter the following:

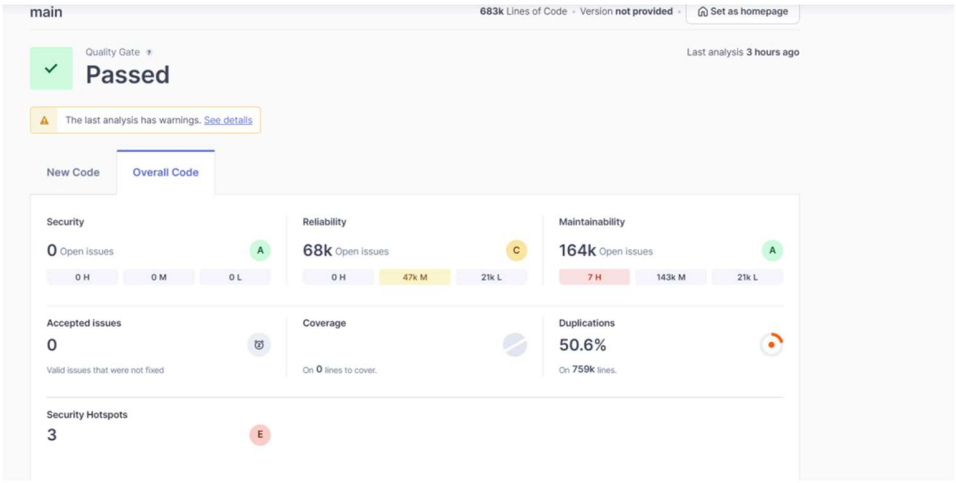
11. Go to the job you had just built and click on Build Now.



12.Once it is built, check the console output.



13.Once the build is complete, go back to SonarQube and check the project linked.

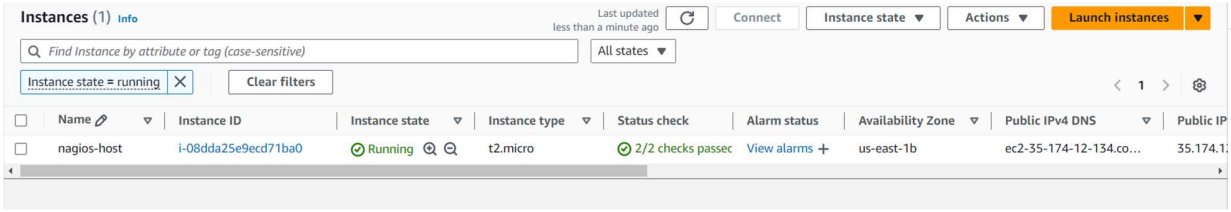


Installation of Nagios

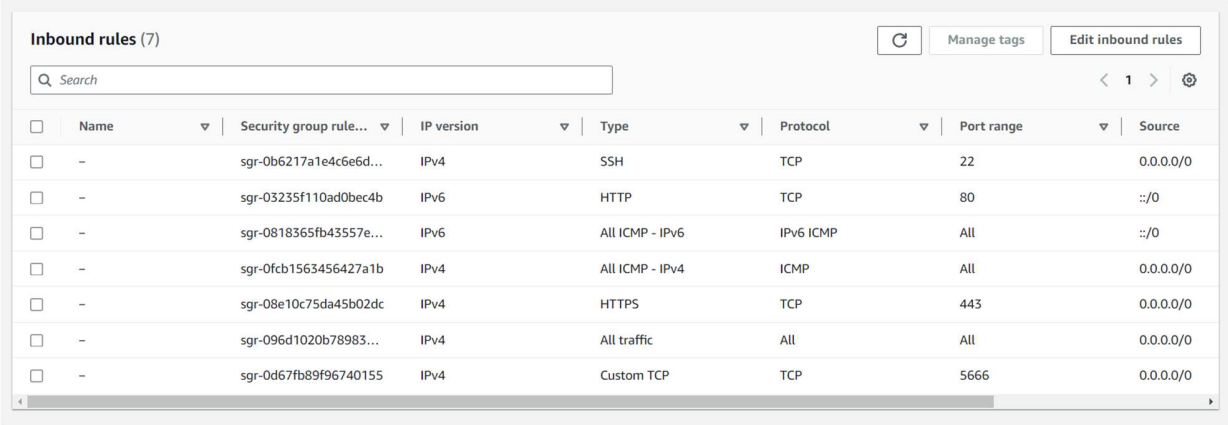
Prerequisites: AWS Free Tier

Steps:

1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host

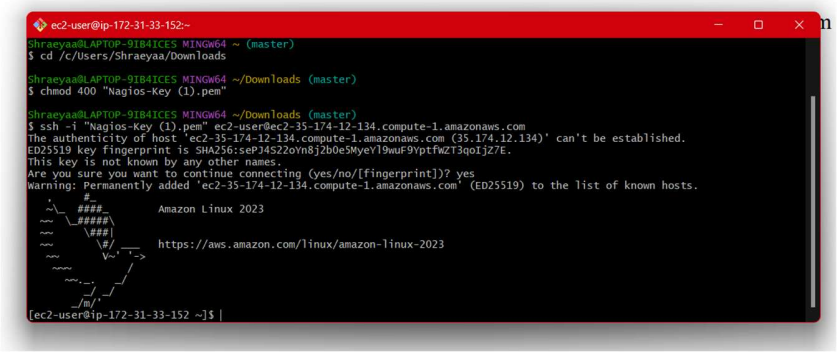


2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.



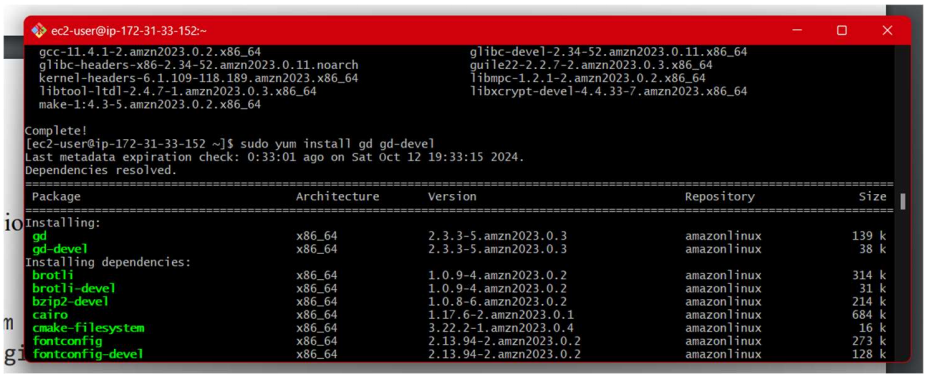
You have to edit the inbound rules of the specified Security Group for this.

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.



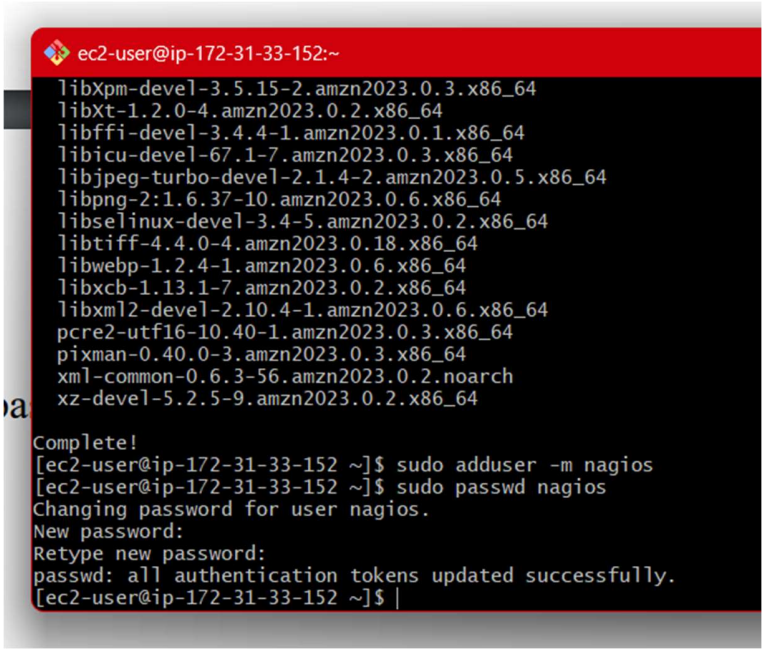
4. Update the package indices and install the following packages using yum

sudo yum update `sudo yum install httpd php`
`sudo yum install gcc glibc glibc-common`
`sudo yum install gd gd-devel`



5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

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

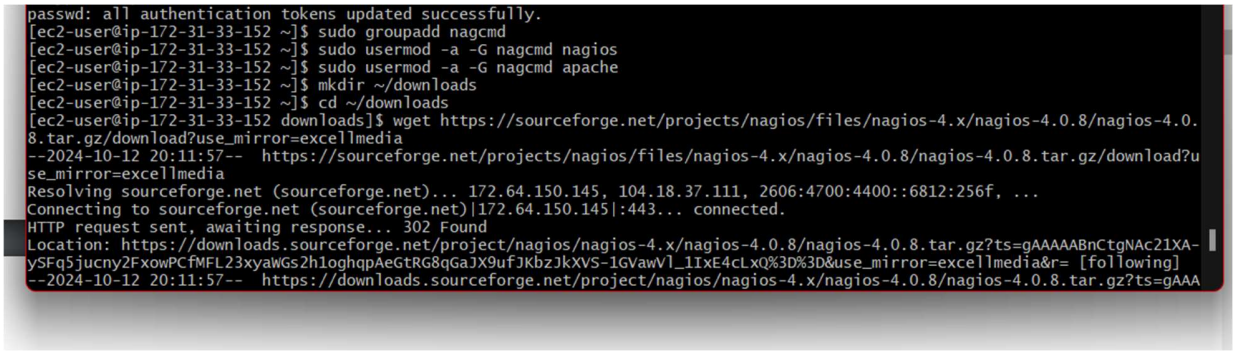


6. Create a new user group

```
sudo groupadd nagcmd
```

7. Use these commands so that you don't have to use sudo for Apache and Nagios

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

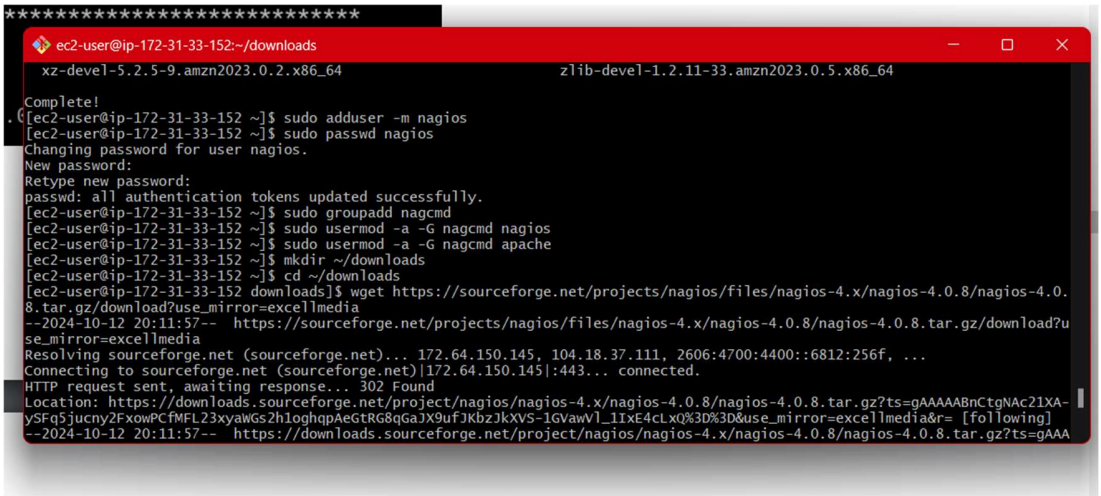


8. Create a new directory for Nagios downloads

```
mkdir ~/downloads cd
~/downloads
```

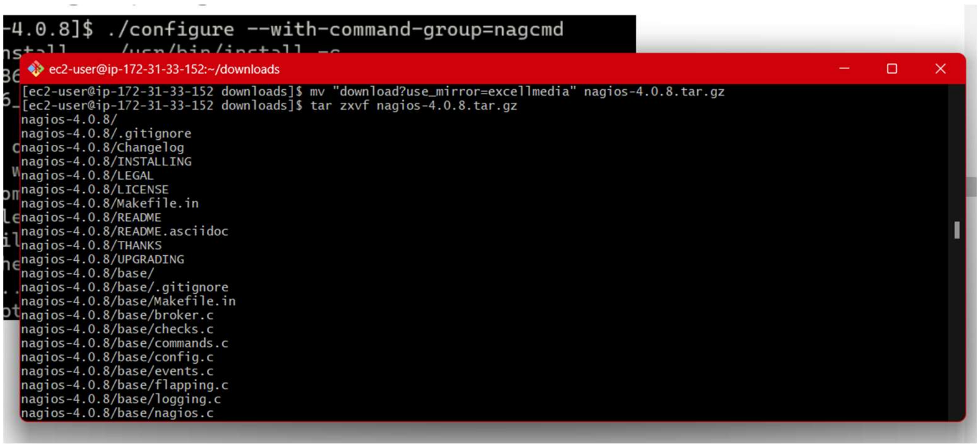
9. Use wget to download the source zip files.

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.
gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
```



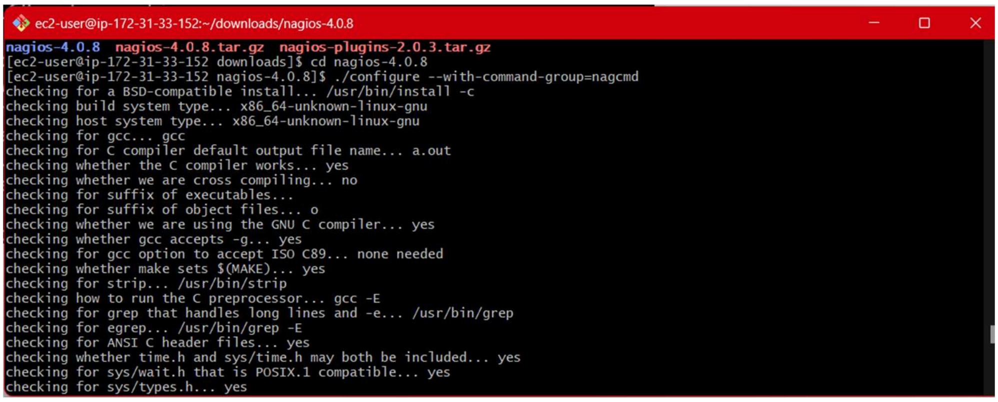
10. Use tar to unzip and change to that directory.

```
tar zxvf nagios-4.0.8.tar.gz
```



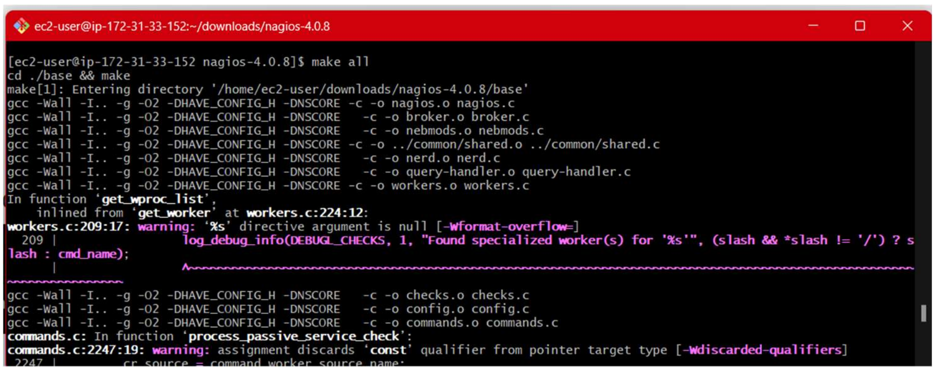
11. Run the configuration script with the same group name you previously created.

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



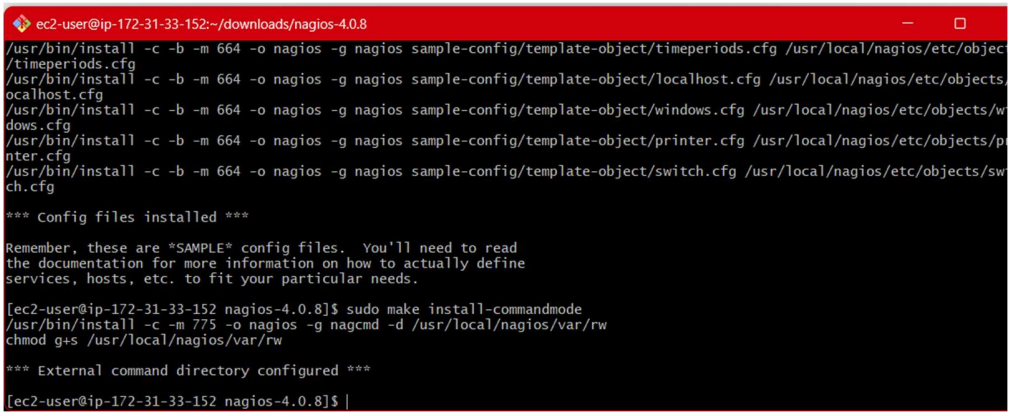
12. Compile the source code.

```
make all
```



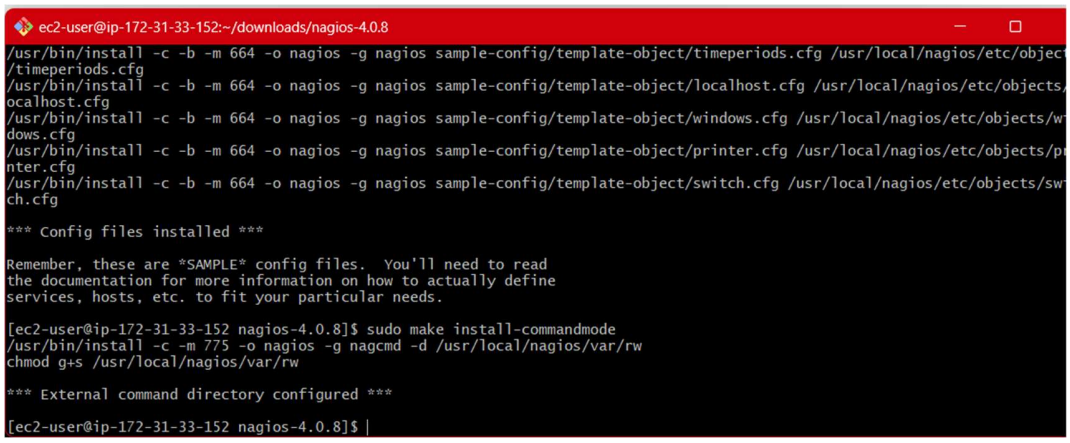
13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

```
sudo make install sudo make
install-init    sudo    make
install-config   sudo   make
install-commandmode
```



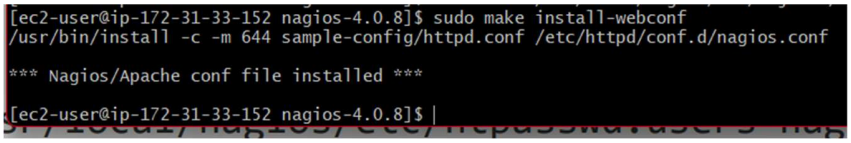
14. Edit the config file and change the email address. `sudo nano`

    `/usr/local/nagios/etc/objects/contacts.cfg`



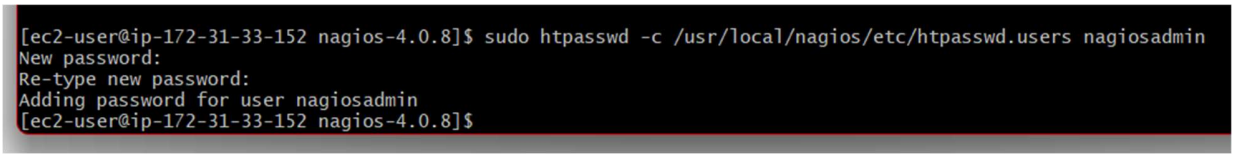
15. Configure the web interface.

    `sudo make install-webconf`



16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

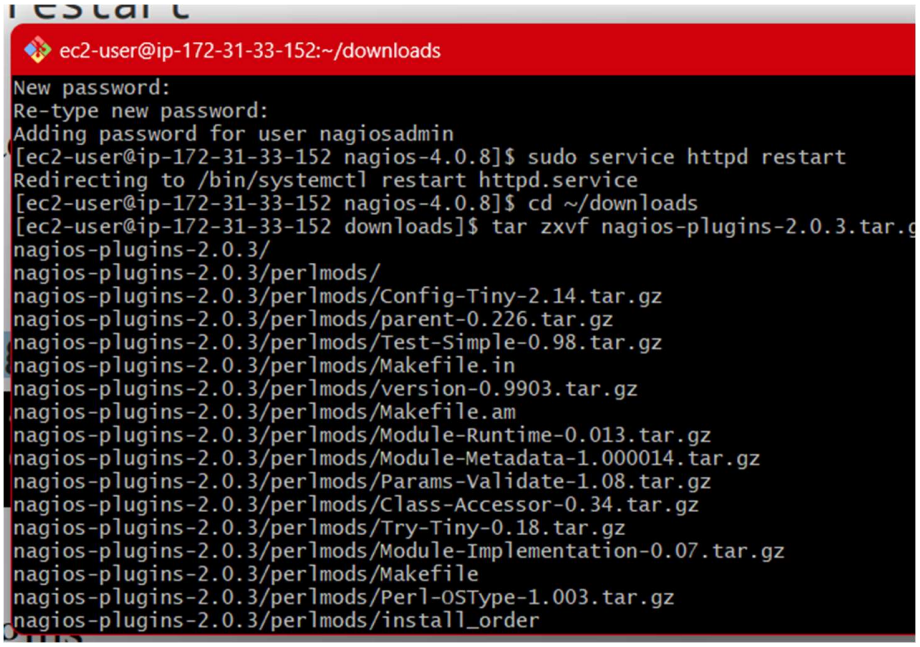


17. Restart Apache

```
sudo service httpd restart
```

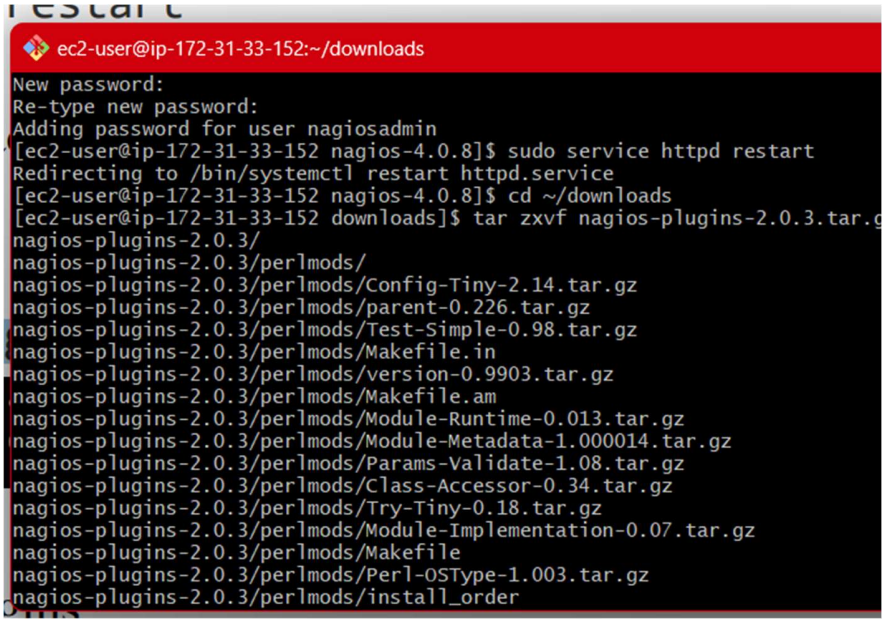
18. Go back to the downloads folder and unzip the plugins zip file.

```
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
```



19. Compile and install plugins

```
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make sudo make install
```



20. Start Nagios

    Add Nagios to the list of system services

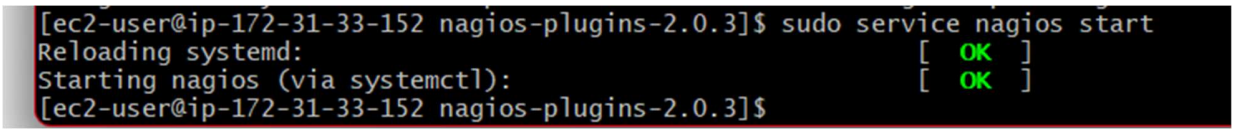
    ```
    sudo chkconfig --add nagios sudo
    chkconfig nagios on
    ```

    Verify the sample configuration files

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

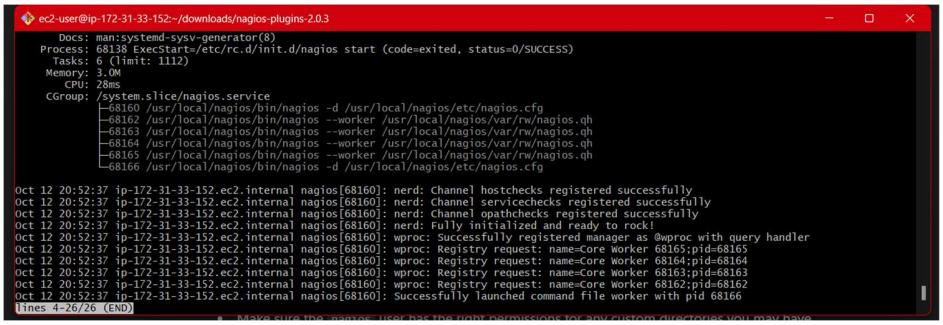
    If there are no errors, you can go ahead and start Nagios.

    ```
    sudo service nagios start
    ```

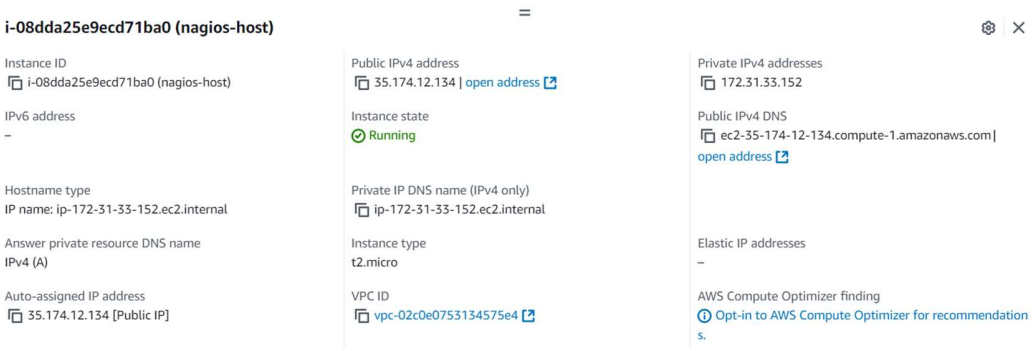


21. Check the status of Nagios

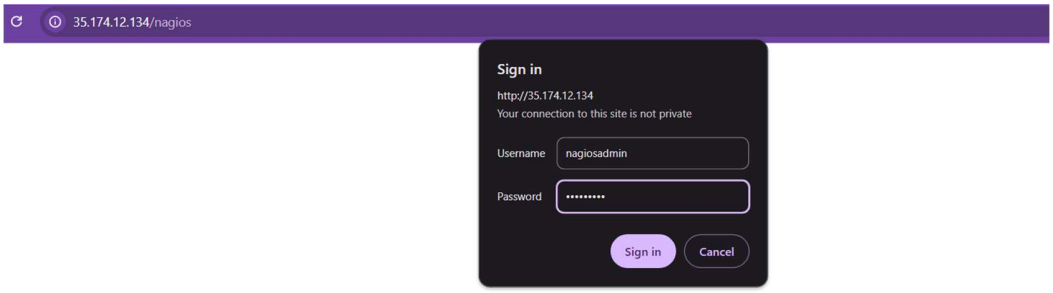
    ```
    sudo systemctl status nagios
    ```



22. Go back to EC2 Console and copy the Public IP address of this instance

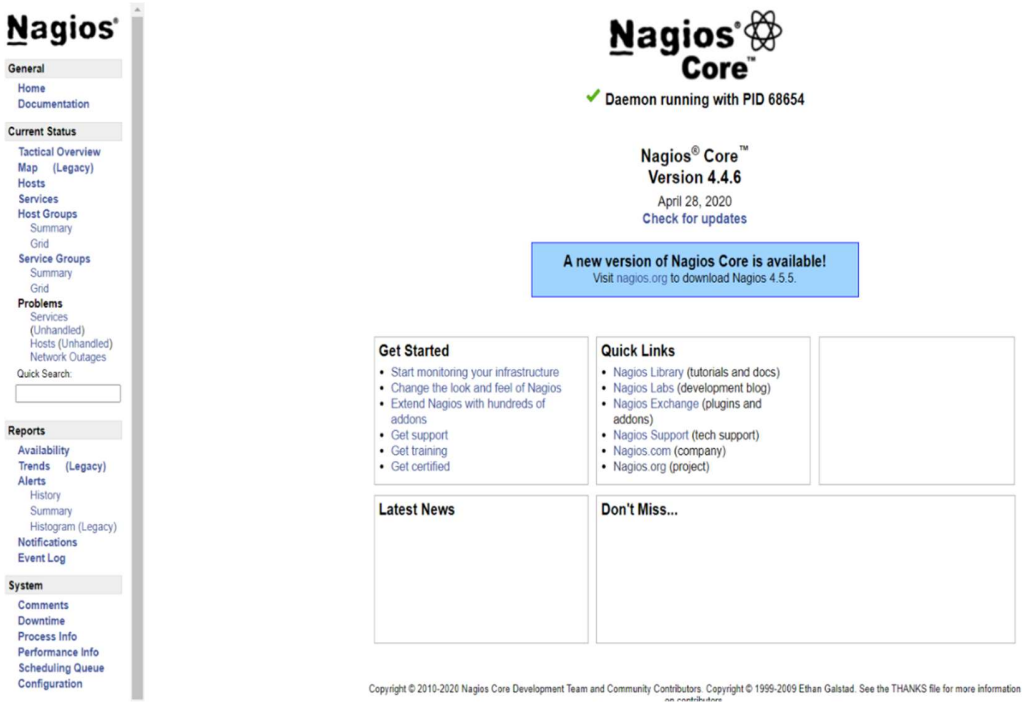


23. Open up your browser and look for **http://<your_public_ip_address>/nagios**



Enter username as nagiosadmin and password which you set in Step 16.

24. After entering the correct credentials, you will see this page.



This means that Nagios was correctly installed and configured with its plugins so far.

ADVANCE DEVOPS EXP-10

SHRAEYAA DHAIGUDE **D15A/15**

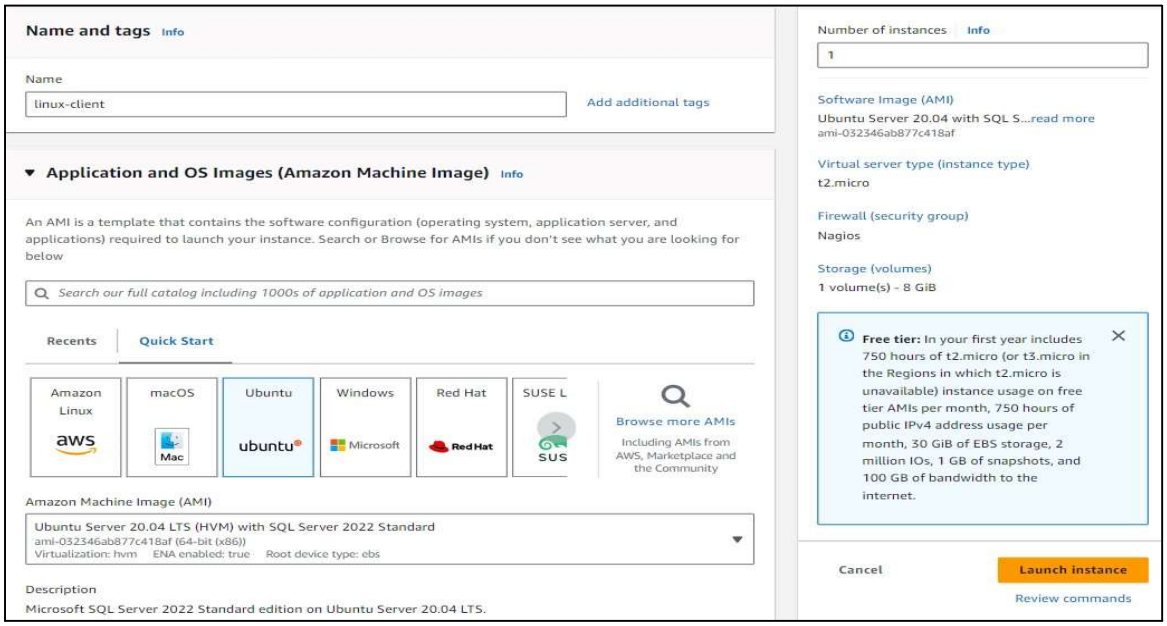
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

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

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

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

Step-2. Create an Ubuntu 20.04 Server EC2 Instance



Step-3: Verify Nagios Process on the Server

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

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

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

i-0ae1aae975bae3b7a (nagios-host)

Step-5: Edit the Configuration File

sudo nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

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

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

# Define a host for the local machine

define host {

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

    host_name               linuxserver
    alias                   linuxserver
    address                 35.174.139.220
}

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

# Define an optional hostgroup for Linux machines

define hostgroup {

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

Step-6: Update Nagios Configuration sudo
nano /usr/local/nagios/etc/nagios.cfg

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

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


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

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

Step-7: Verify Configuration Files sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0
```

Step-8: Restart Nagios Service sudo
systemctl restart nagios

Step-9:. SSH into the Client Machine

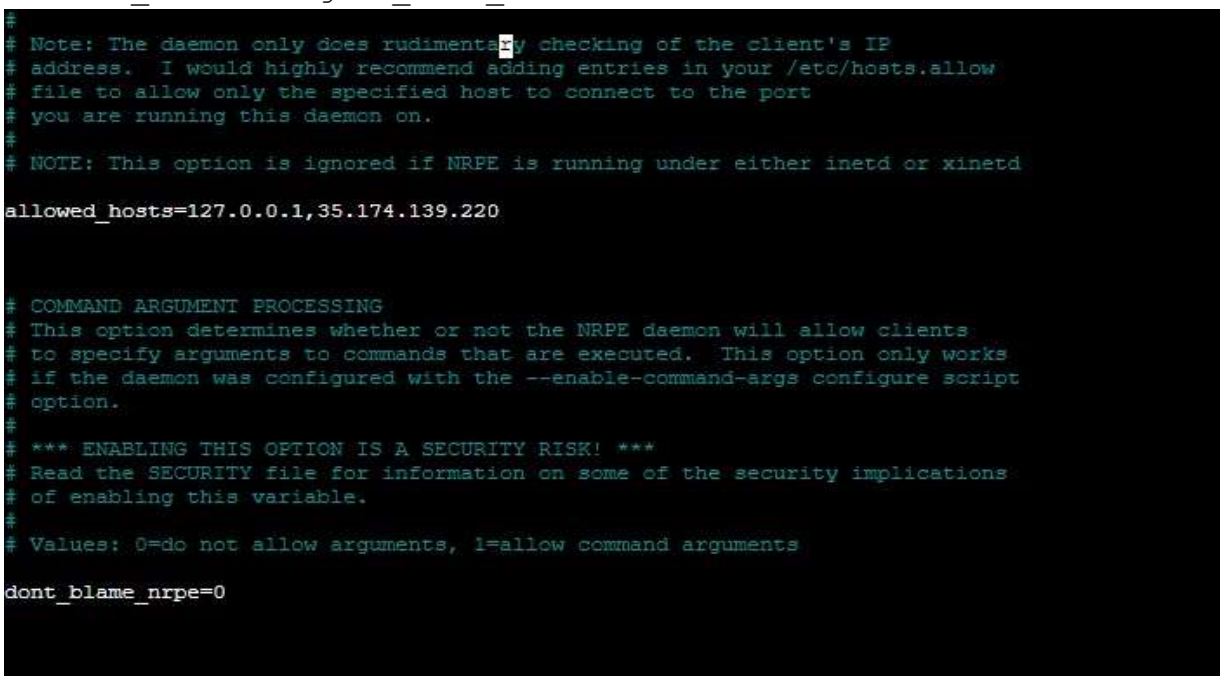
Use SSH or EC2 Instance Connect to access the `linux-client`.

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

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

Step-11: Edit NRPE Configuration File

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

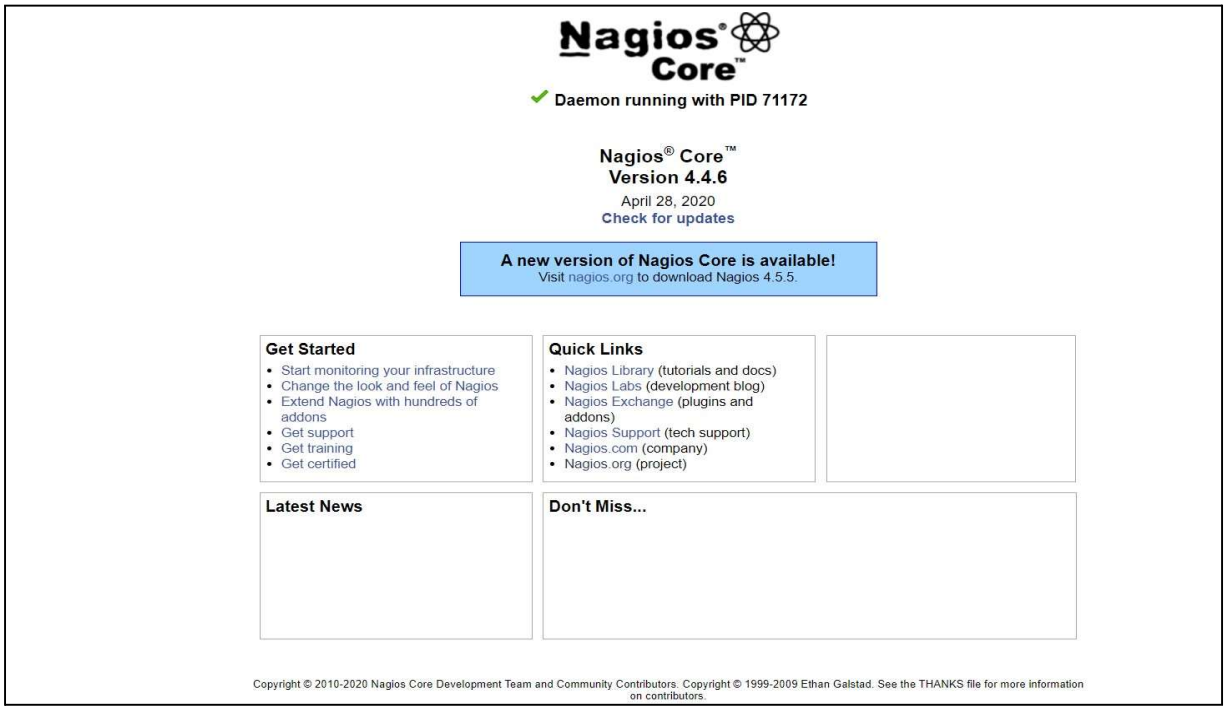


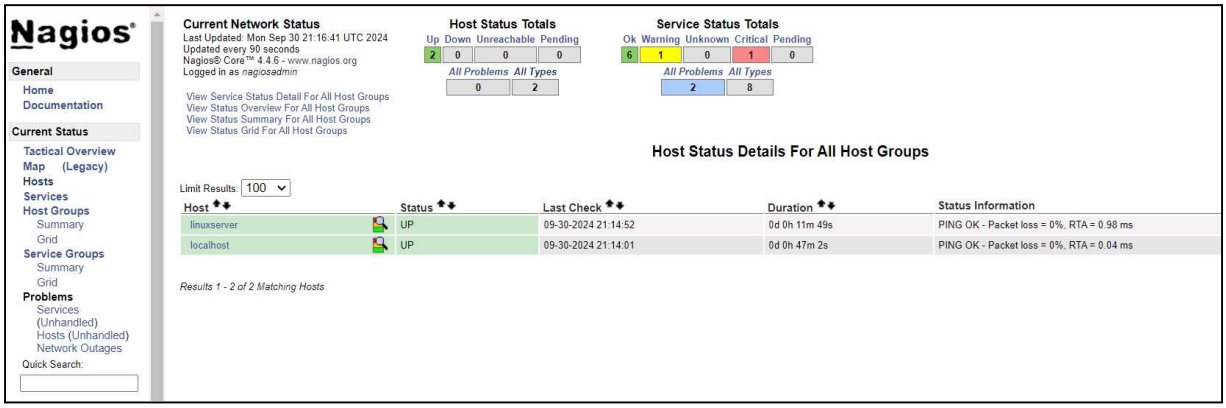
Step-12: Restart NRPE Server

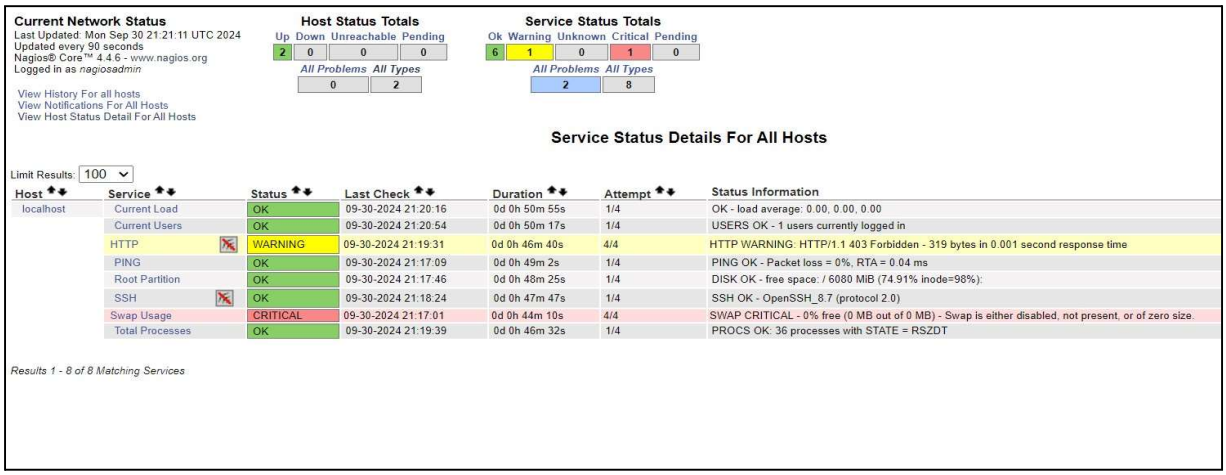
Commands -
sudo systemctl restart nagios-nrpe-server

Step-13: Check Nagios Dashboard

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


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


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


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

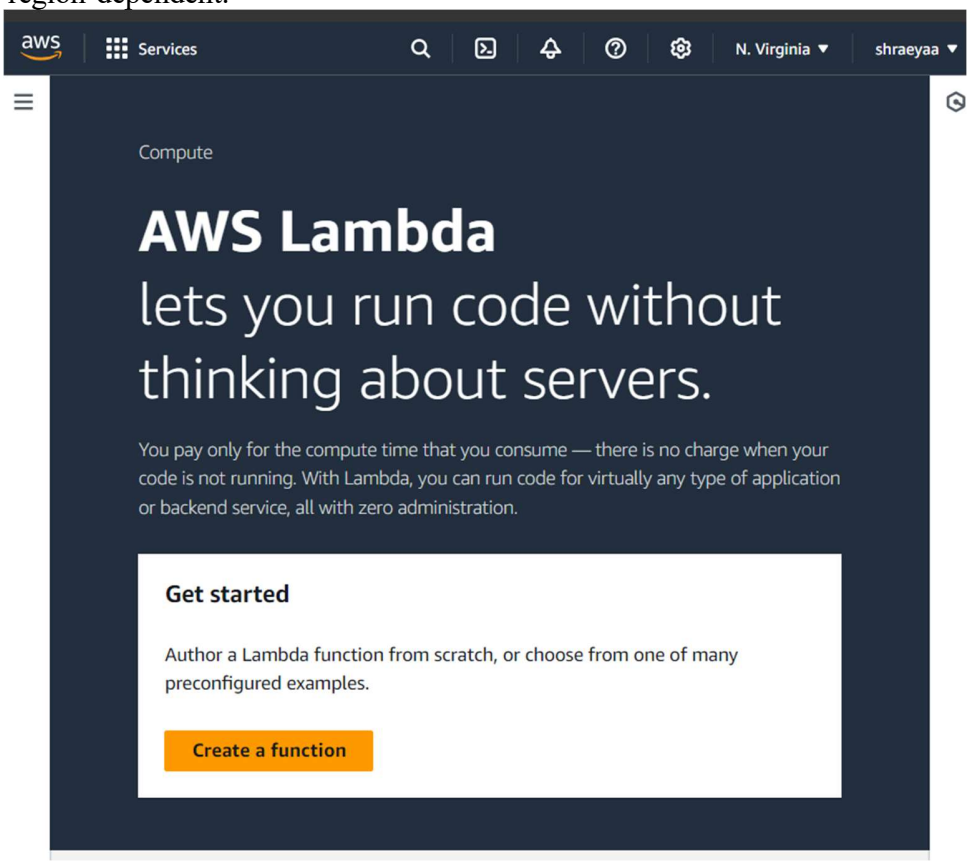
Advance Devops-11

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

Steps to create an AWS Lambda function

1. Open up the Lambda Console and click on the Create button.

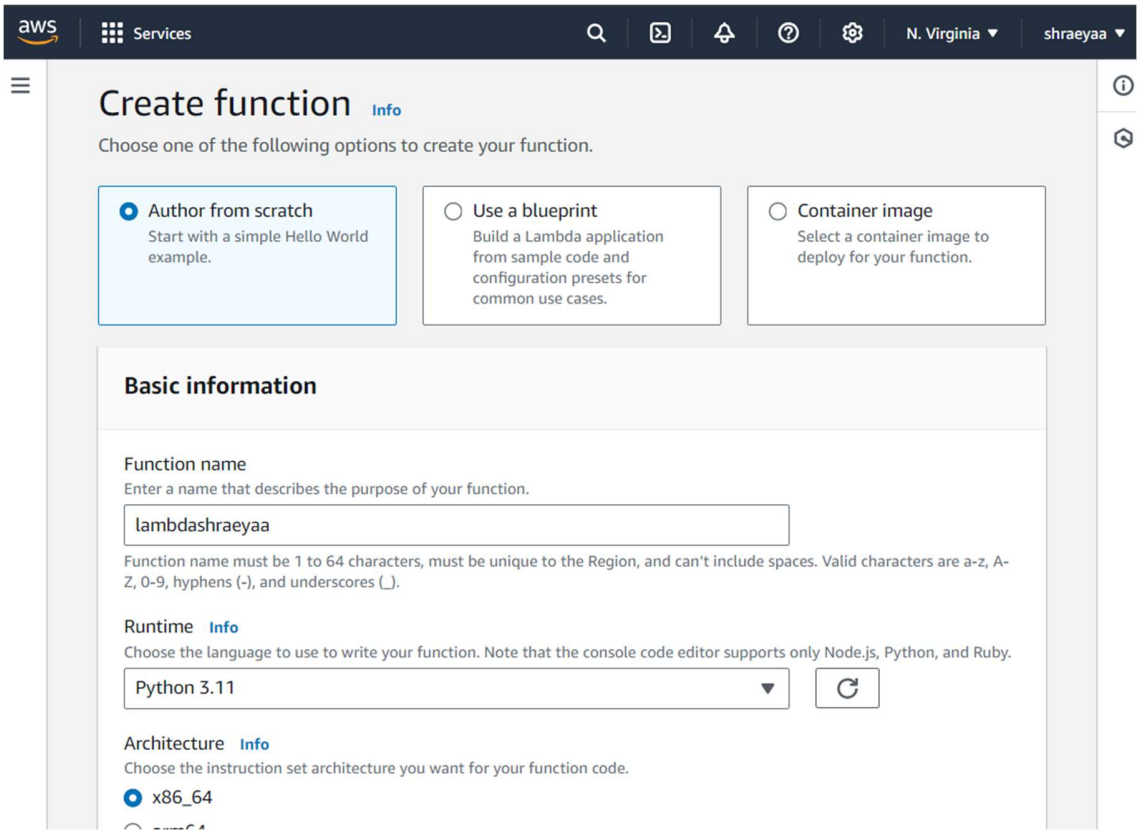
   Be mindful of where you create your functions since Lambda is region-dependent.



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.



arm64

Permissions Info

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

▼ **Change default execution role**

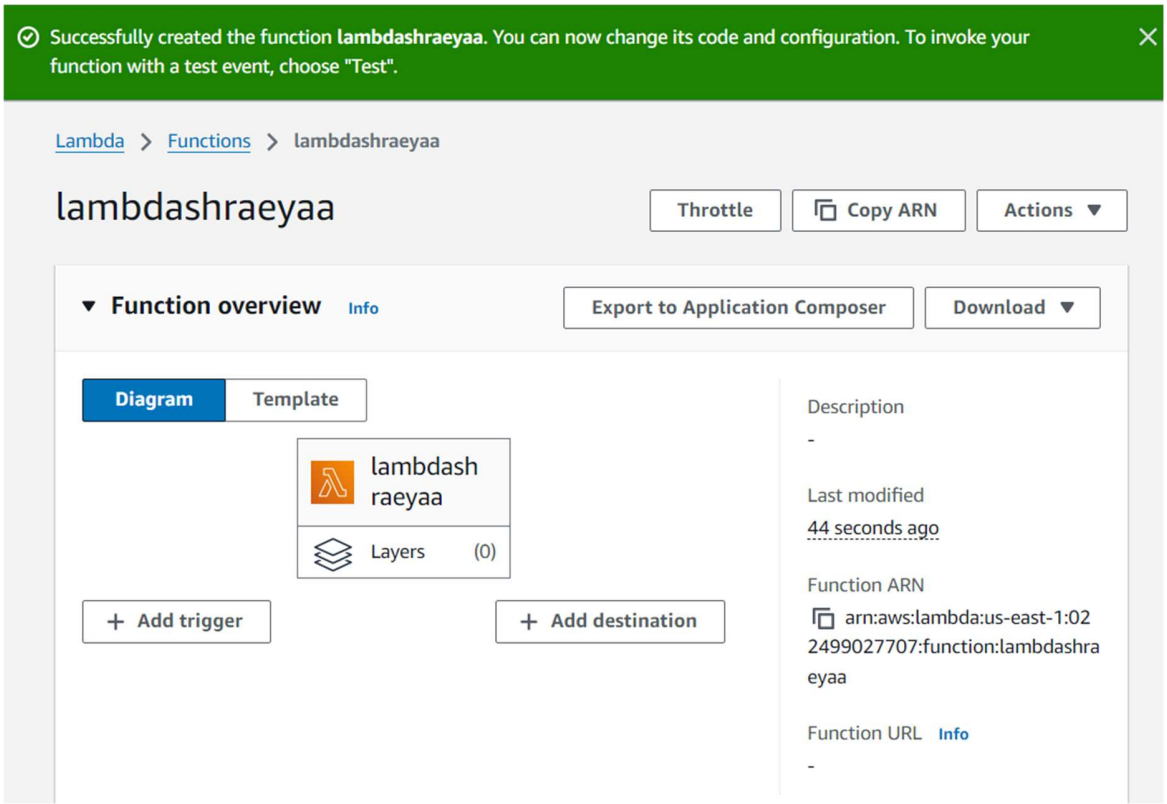
Execution role

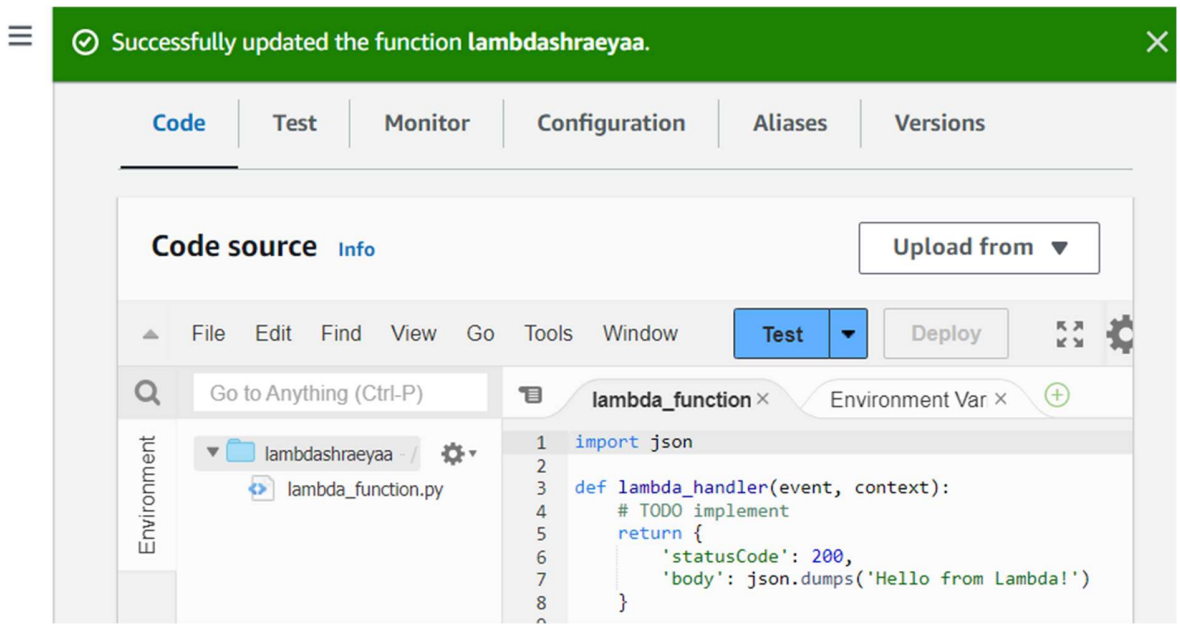
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

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

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

Lambda will create an execution role named lambdashraeyaa-role-l4l1w09a, with permission to upload logs to Amazon CloudWatch Logs.


Successfully created the function lambdashraeyaa. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > lambdashraeyaa
lambdashraeyaa
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
lambdash raeyaa
Layers (0)
+ Add trigger
+ Add destination
Description
-
Last modified
44 seconds ago
Function ARN
arn:aws:lambda:us-east-1:02 2499027707:function:lambdashra eyaa
Function URL Info
-


Successfully updated the function lambdashraeyaa.
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Environment
lambdashraeyaa
lambda_function.py
1 import json
2
3 def lambda_handler(event, context):
4 # TODO implement
5 return {
6 'statusCode': 200,
7 'body': json.dumps('Hello from Lambda!')
8 }

Basic settings Info

Description - *optional*

Memory Info

Your function is allocated CPU proportional to the memory configured.

128 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage Info

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations.

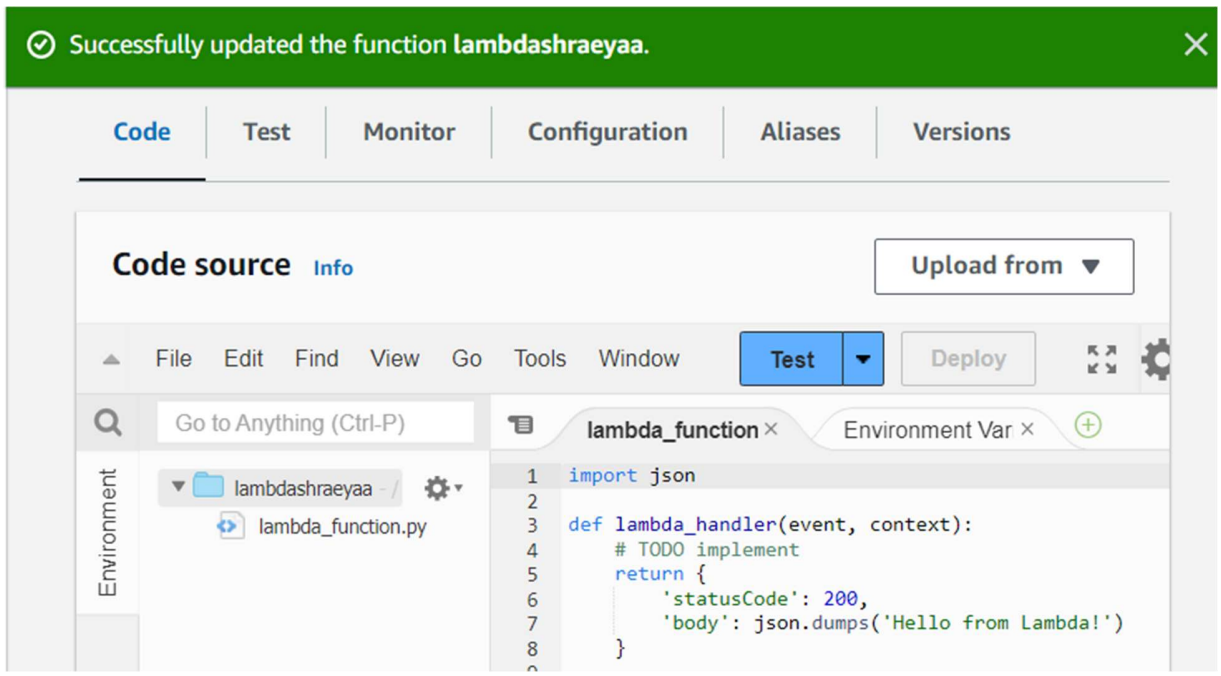
None

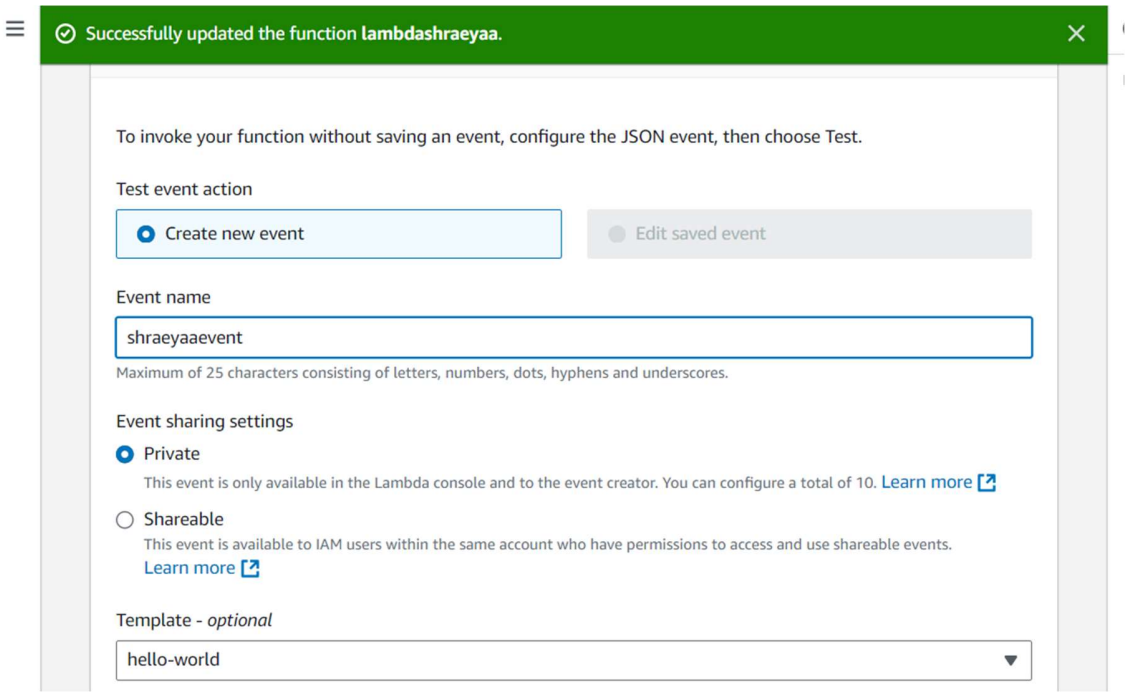
Supported runtimes: Java 11, Java 17, Java 21.

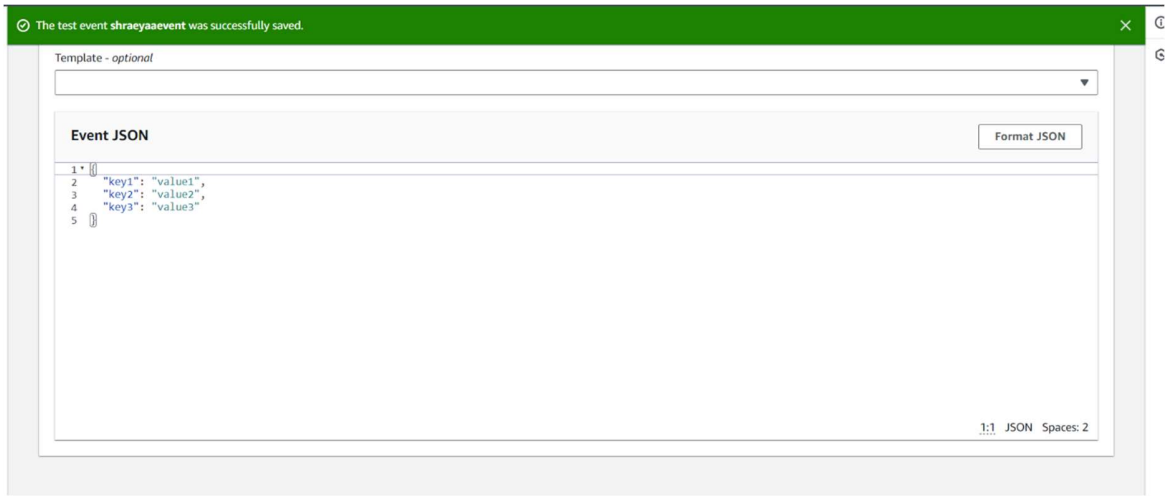
Timeout

0 min 3 sec

Execution role





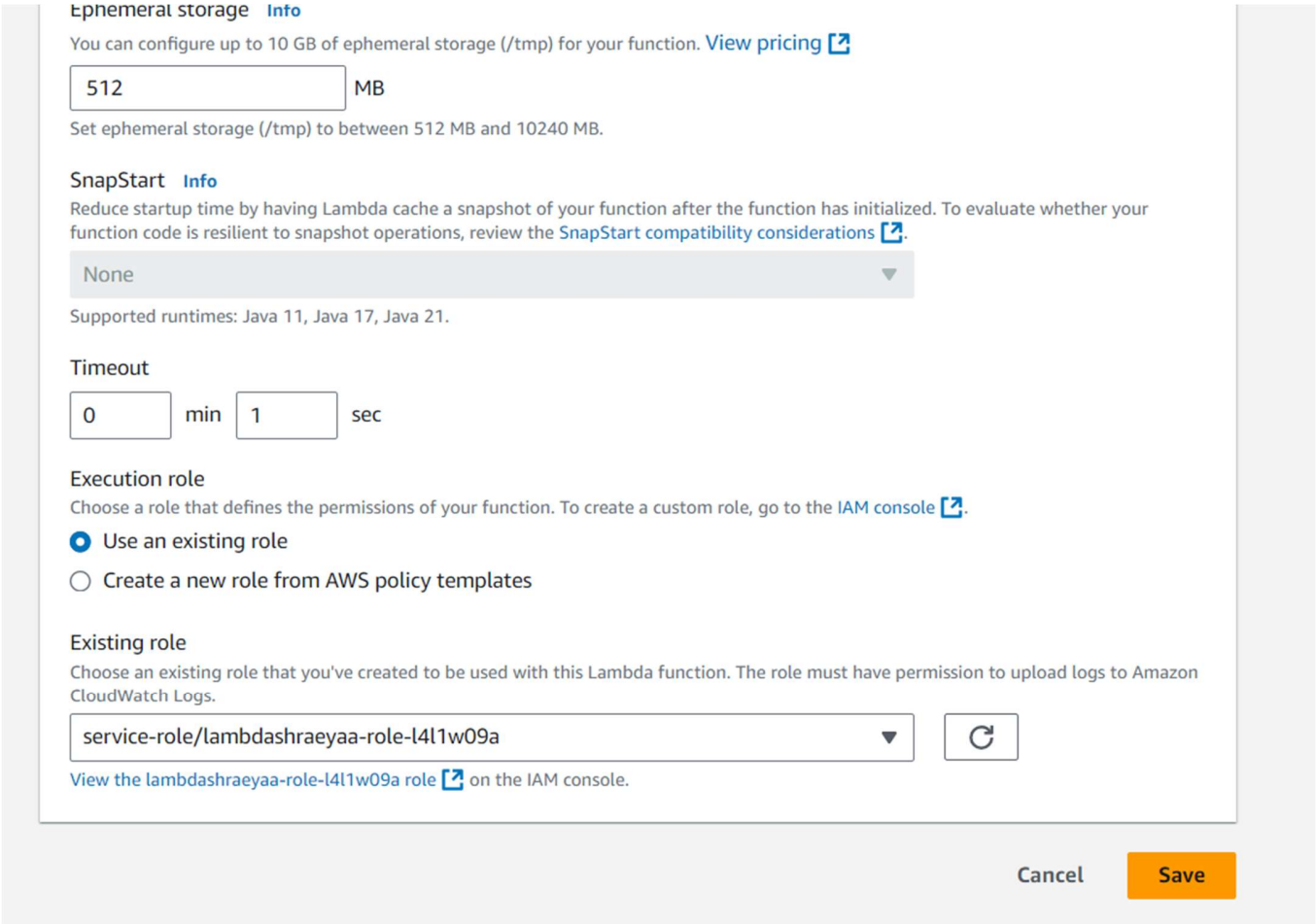


Click on the *Create* button.

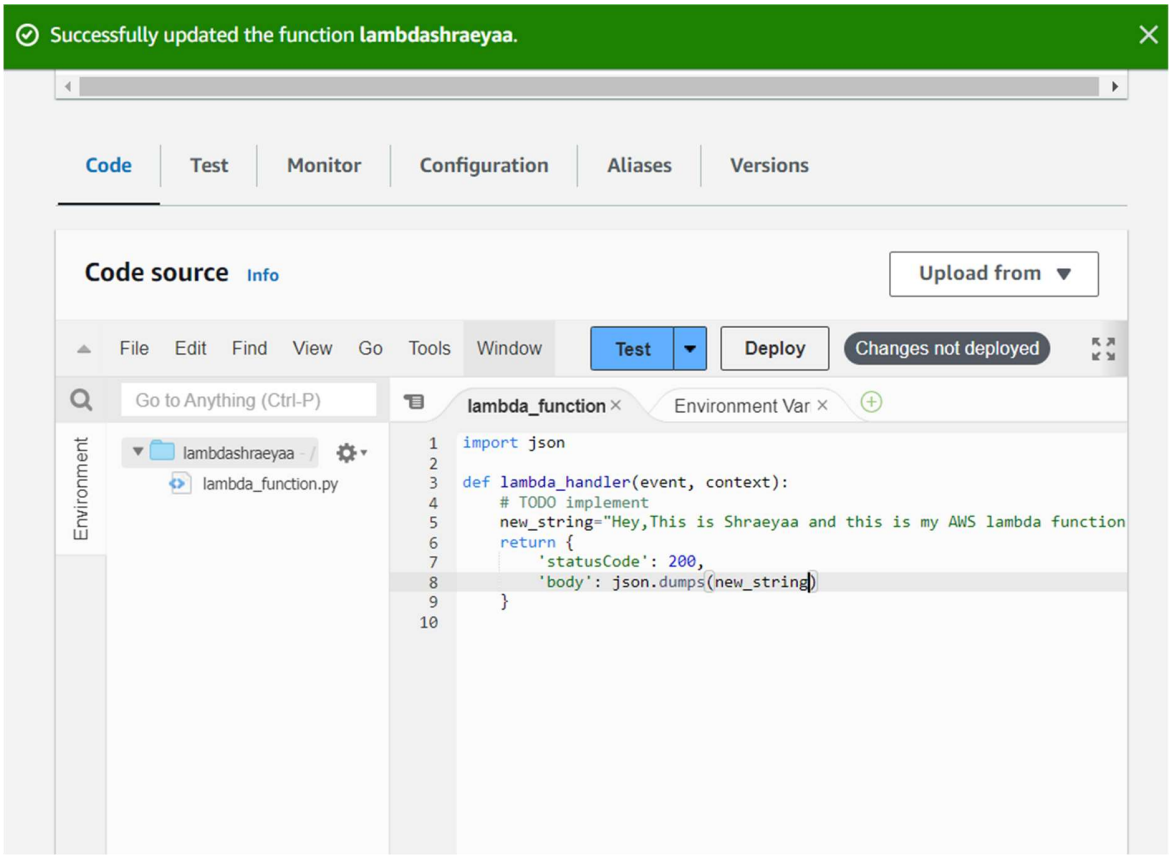
3. This process will take a while to finish and after that, you'll get a message that your function was successfully create

4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.

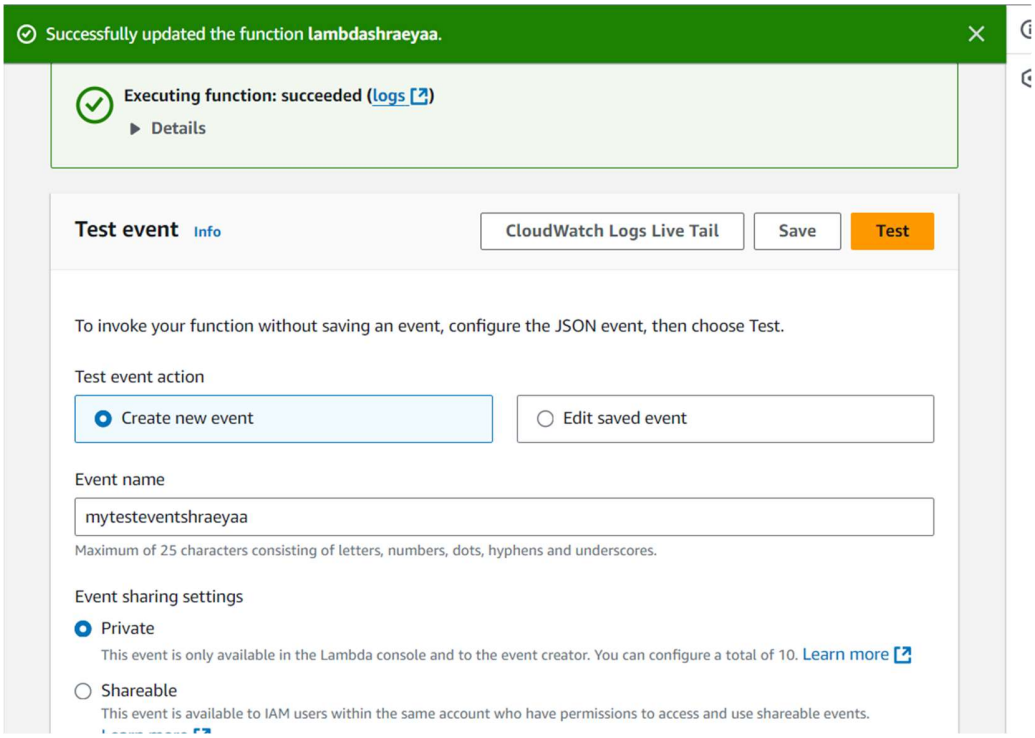
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



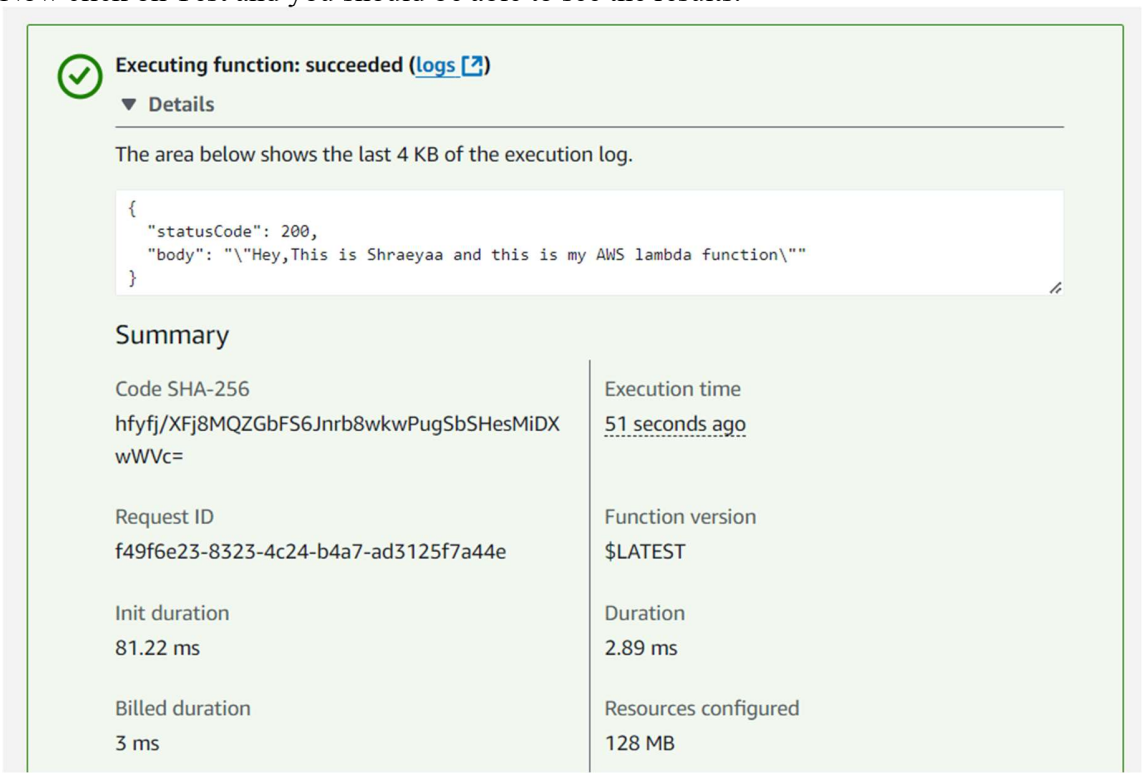
5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed.
   Press Ctrl + S to save the file and click Deploy to deploy the changes.

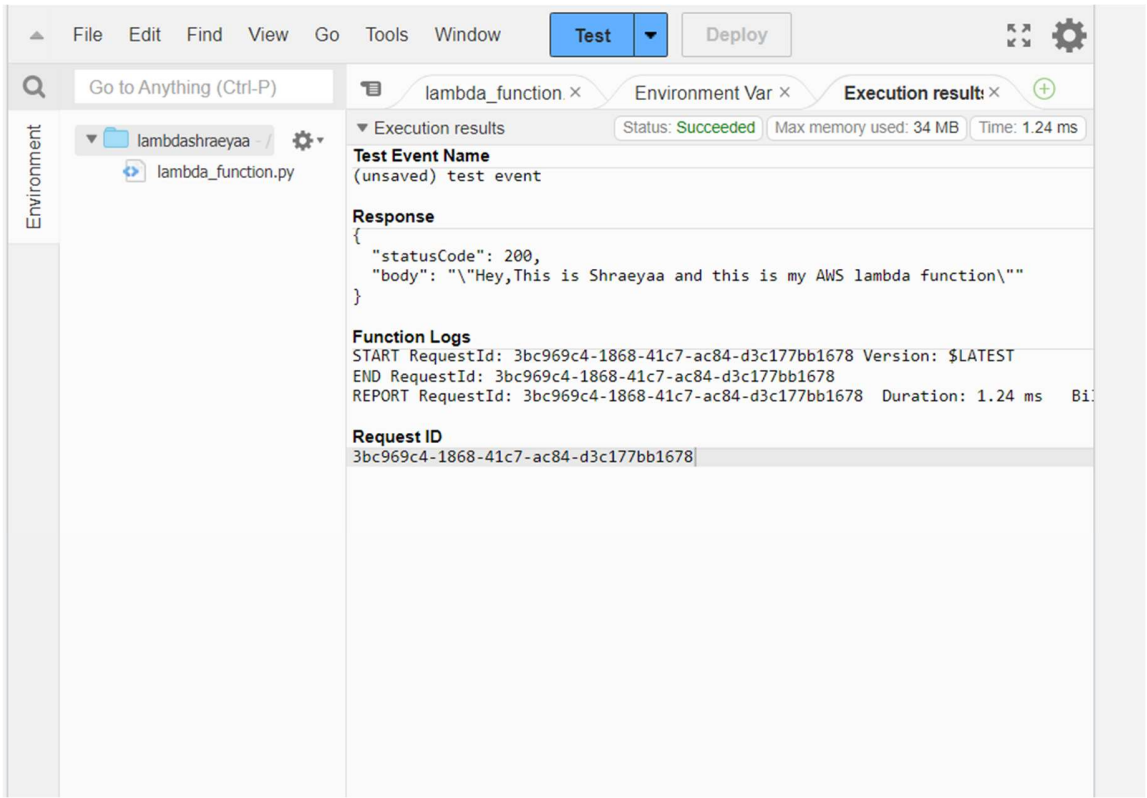


6. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



7. Now click on Test and you should be able to see the results.



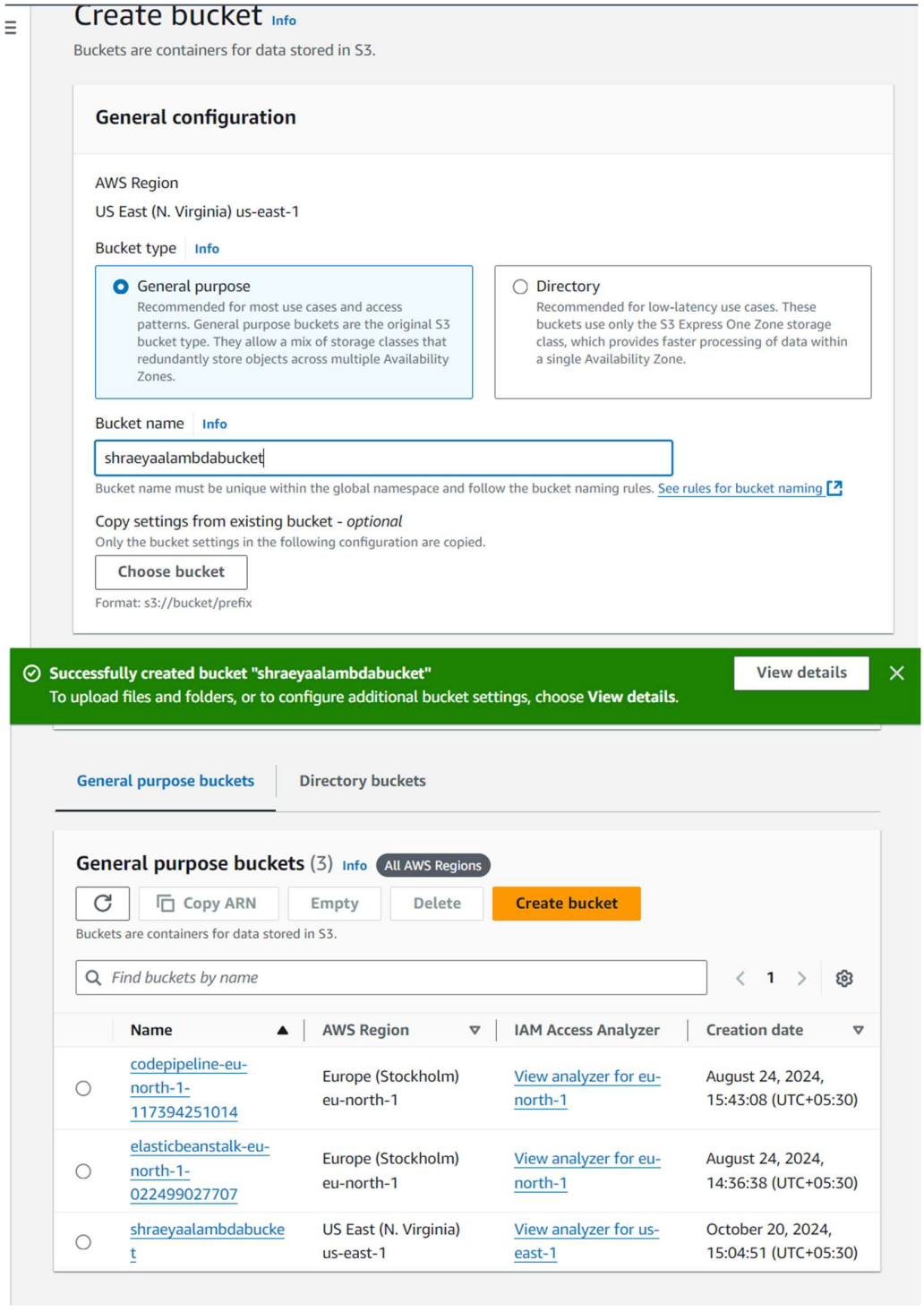


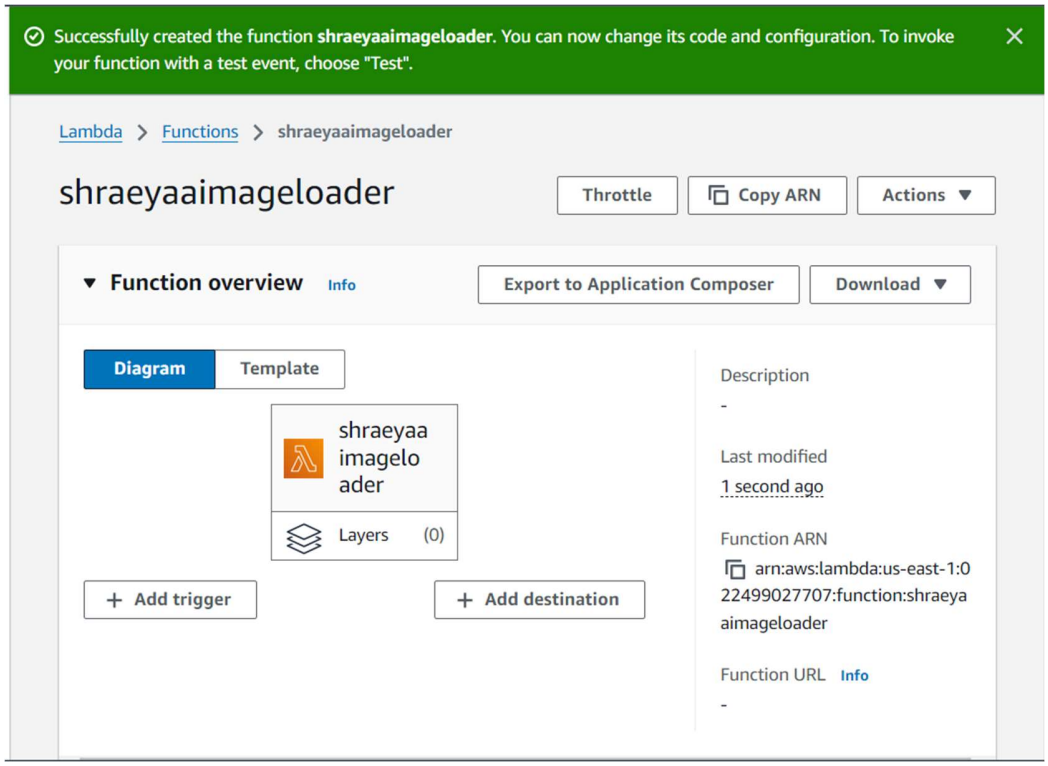
Conclusion:

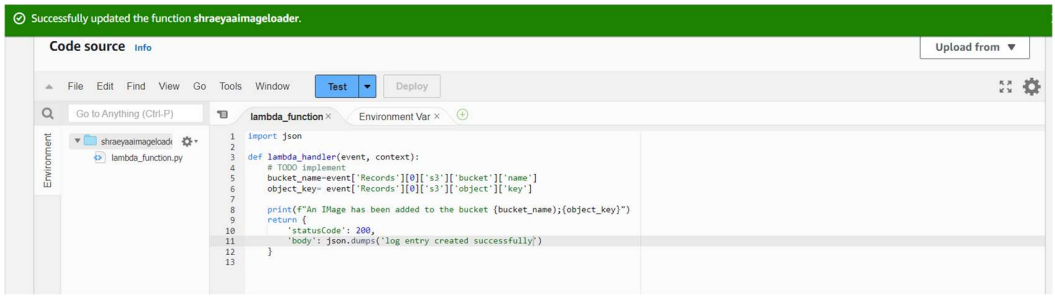
AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database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

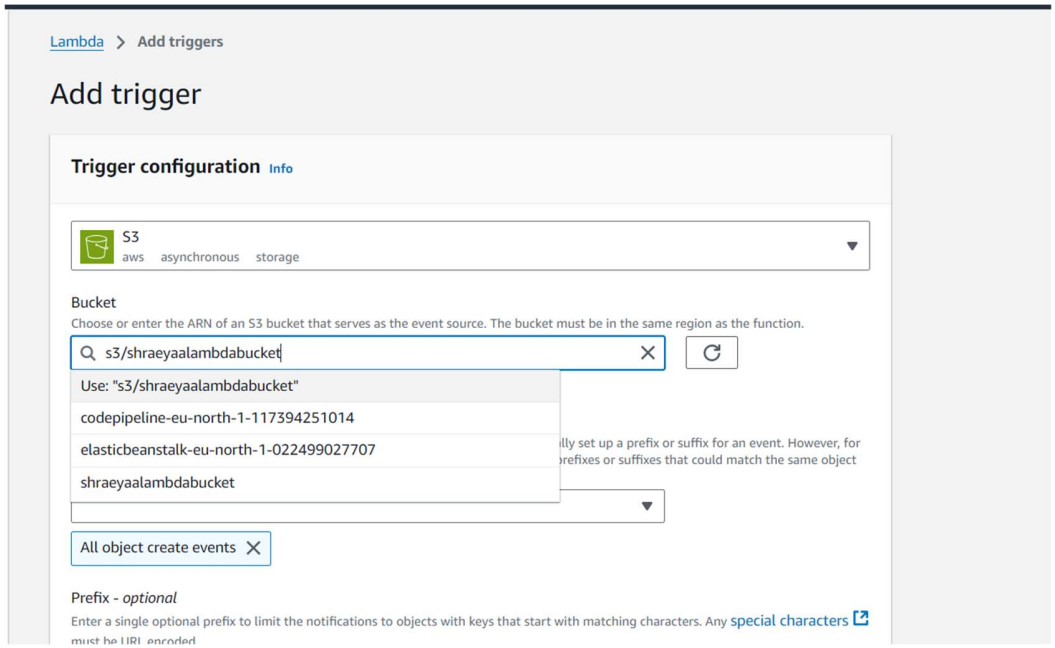
Advance Devops-12

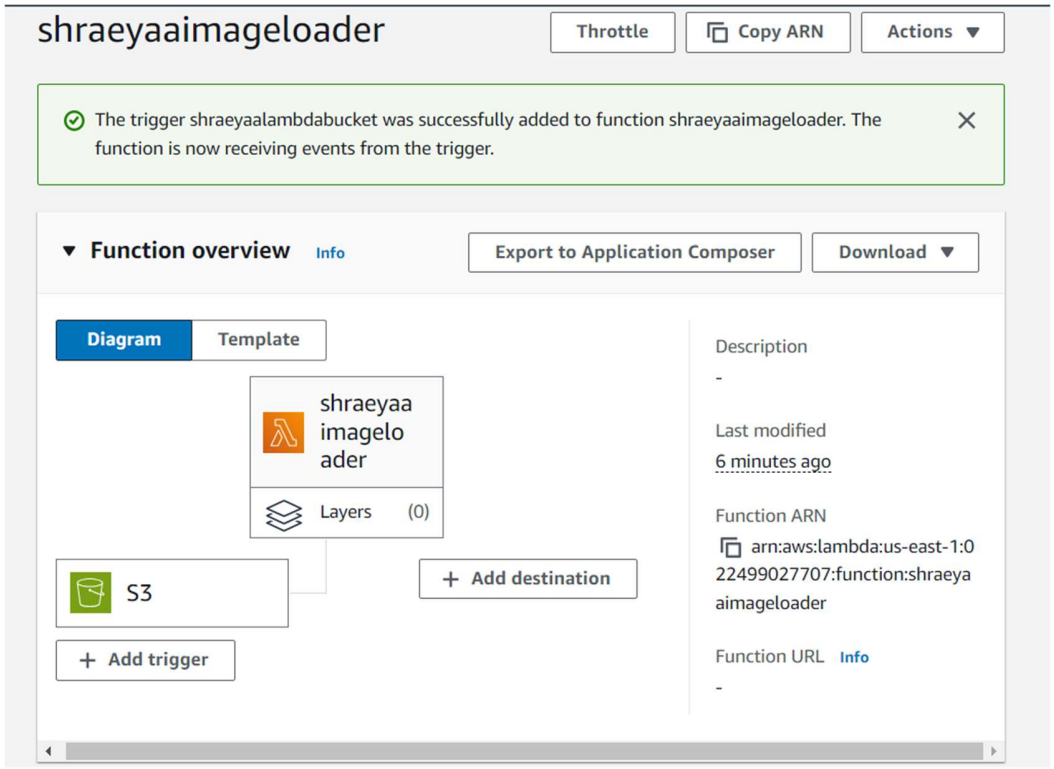
Aim:To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

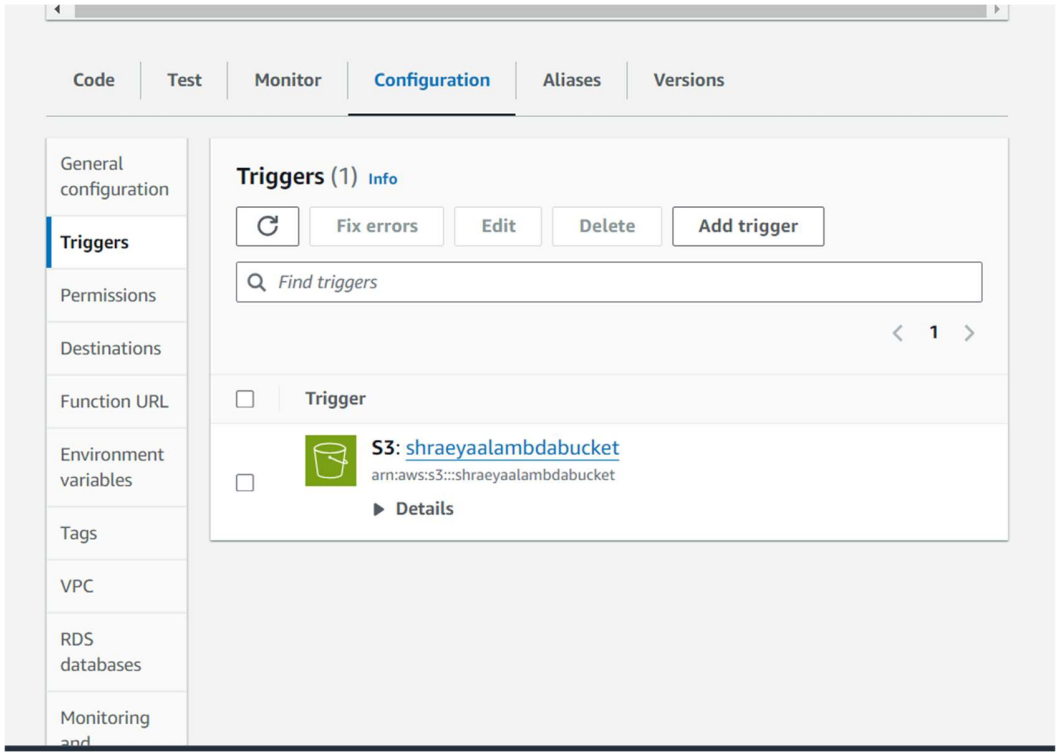


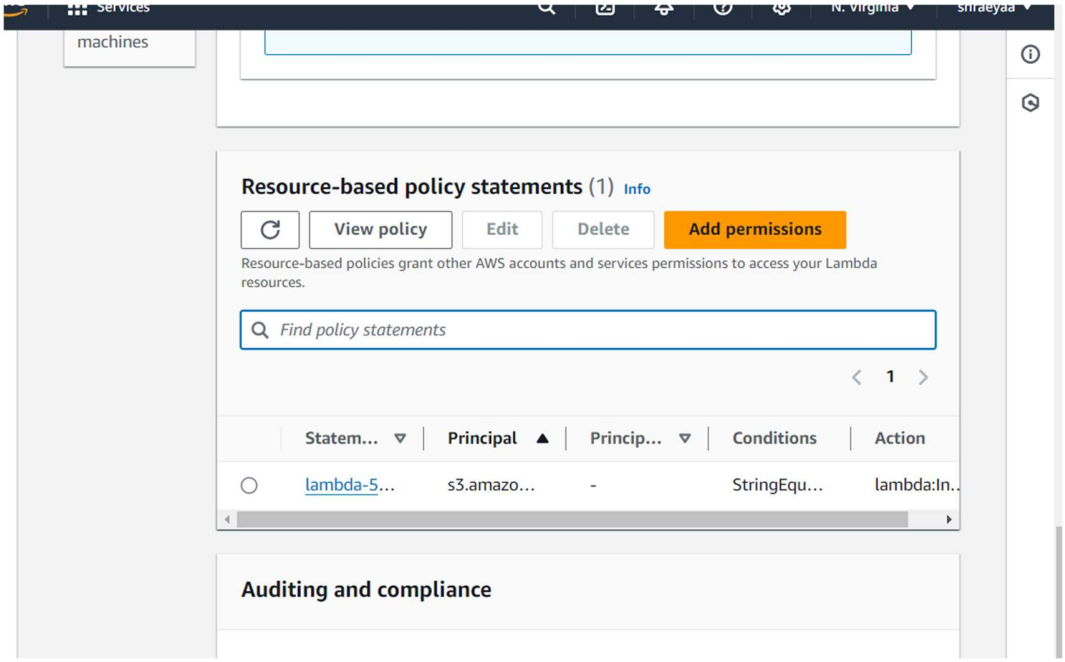

Successfully created the function shraeyaaimageloader. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > shraeyaaimageloader
shraeyaaimageloader
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
shraeyaa imagelo ader
Layers (0)
+ Add trigger
+ Add destination
Description
-
Last modified
1 second ago
Function ARN
arn:aws:lambda:us-east-1:022499027707:function:shraeyaaimageloader
Function URL Info
-

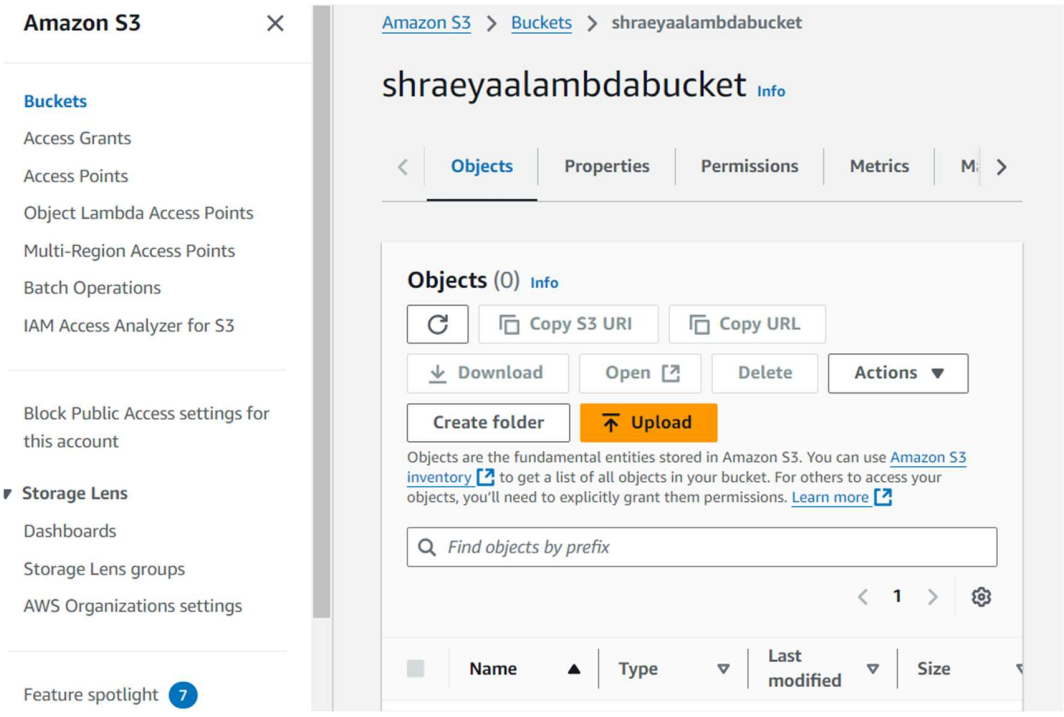

Successfully updated the function shraeyaaimageloader.
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Environment
shraeyaaimageloadi
lambda_function.py
import json
def lambda_handler(event, context):
# TODO implement
bucket_name=event['Records'][0]['s3']['bucket']['name']
object_key= event['Records'][0]['s3']['object']['key']
print(f"An IMage has been added to the bucket {bucket_name};{object_key}")
return {
'statusCode': 200,
'body': json.dumps('log entry created successfully')
}


Lambda > Add triggers
Add trigger
Trigger configuration Info
S3
aws asynchronous storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/shraeyaalambdabucket
Use: "s3/shraeyaalambdabucket"
codepipeline-eu-north-1-117394251014
elasticbeanstalk-eu-north-1-022499027707
shraeyaalambdabucket
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters


shraeyaaimageloader
Throttle
Copy ARN
Actions
The trigger shraeyaalambdabucket was successfully added to function shraeyaaimageloader. The function is now receiving events from the trigger.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
shraeyaa imagelo ader
Layers (0)
S3
Add destination
Add trigger
Description
-
Last modified
6 minutes ago
Function ARN
arn:aws:lambda:us-east-1:022499027707:function:shraeyaaimageloader
Function URL
Info
-


Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
RDS databases
Monitoring and
Triggers (1)
Info
Fix errors
Edit
Delete
Add trigger
Find triggers
1
Trigger
S3: shraeyaalambdabucket
arn:aws:s3:::shraeyaalambdabucket
Details


machines
Resource-based policy statements (1) Info
View policy
Edit
Delete
Add permissions
Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.
Find policy statements
1
Statem...
Principal
Princip...
Conditions
Action
lambda-5...
s3.amazo...
-
StringEqu...
lambda:In...
Auditing and compliance


Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
Amazon S3 > Buckets > shraeyaalambdabucket
shraeyaalambdabucket Info
Objects
Properties
Permissions
Metrics
Objects (0) 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
1
Name
Type
Last modified
Size

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 31.9 KB)

Remove | Add files | Add folder

All files and folders in this table will be uploaded.

Find by name

1

Name	Folder	Type	Size
download (15).jpg	-	image/jpeg	31.9 KB

Destination Info

Destination

s3://shraeyaalambdabucket

Upload succeeded
View details below.

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://shraeyaalambdabucket	1 file, 31.9 KB (100.00%)	0 files, 0 B (0%)

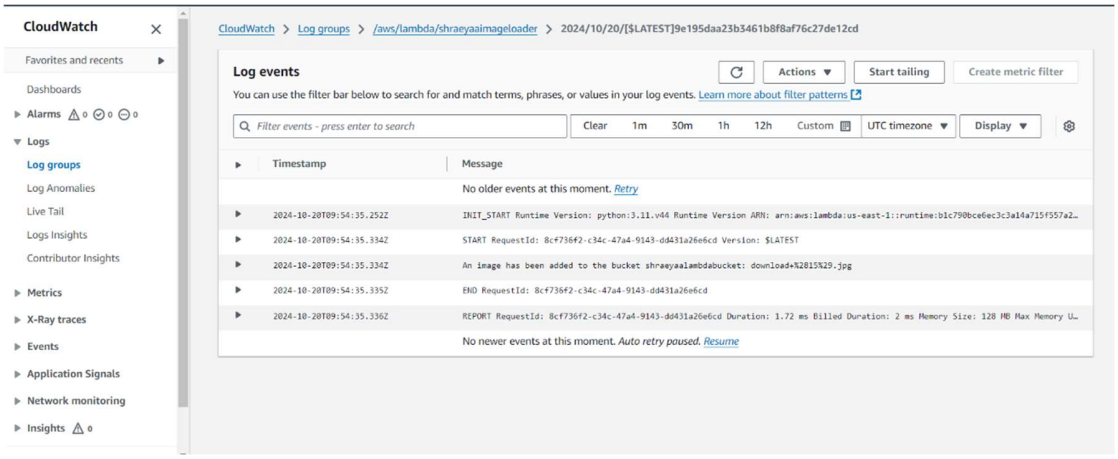
Files and folders | Configuration

Files and folders (1 Total, 31.9 KB)

Find by name

1

Name	Folder	Type	Size	Status	Error
download (...	-	image/jpeg	31.9 KB	Succeeded	-



Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.