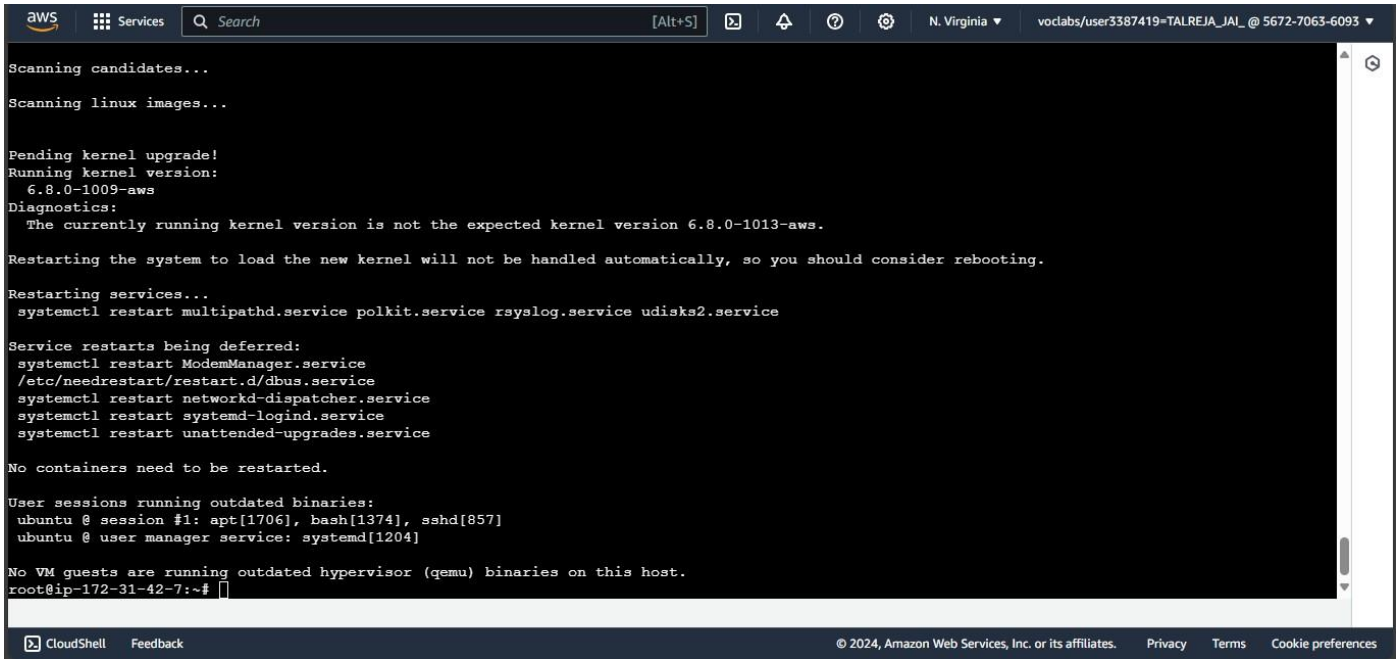
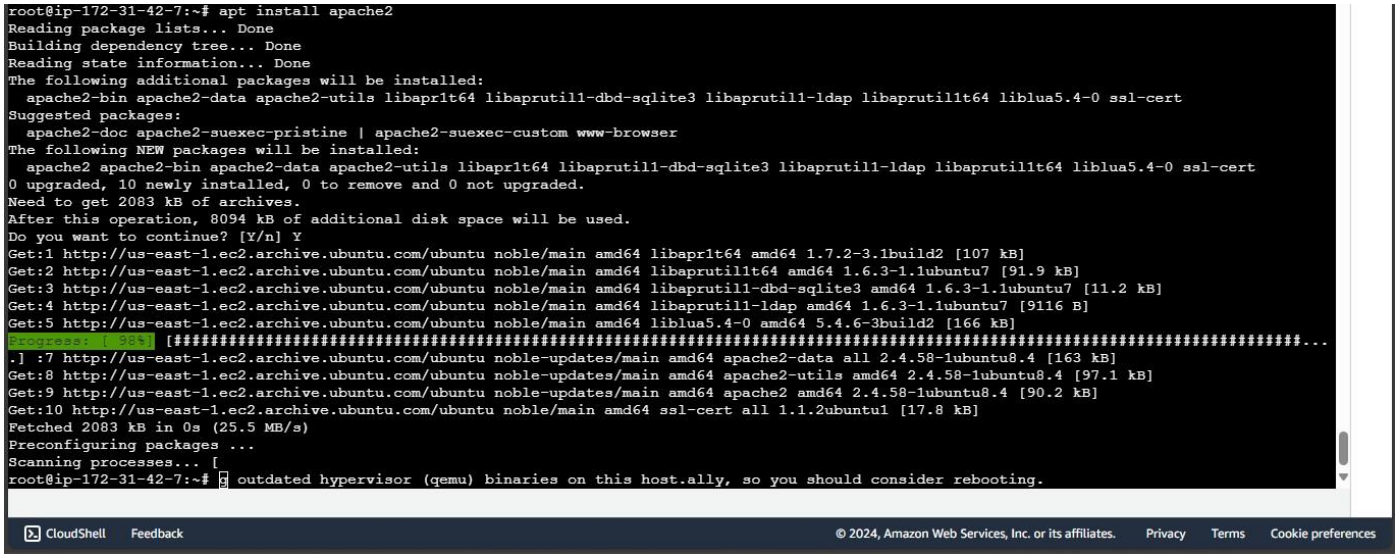
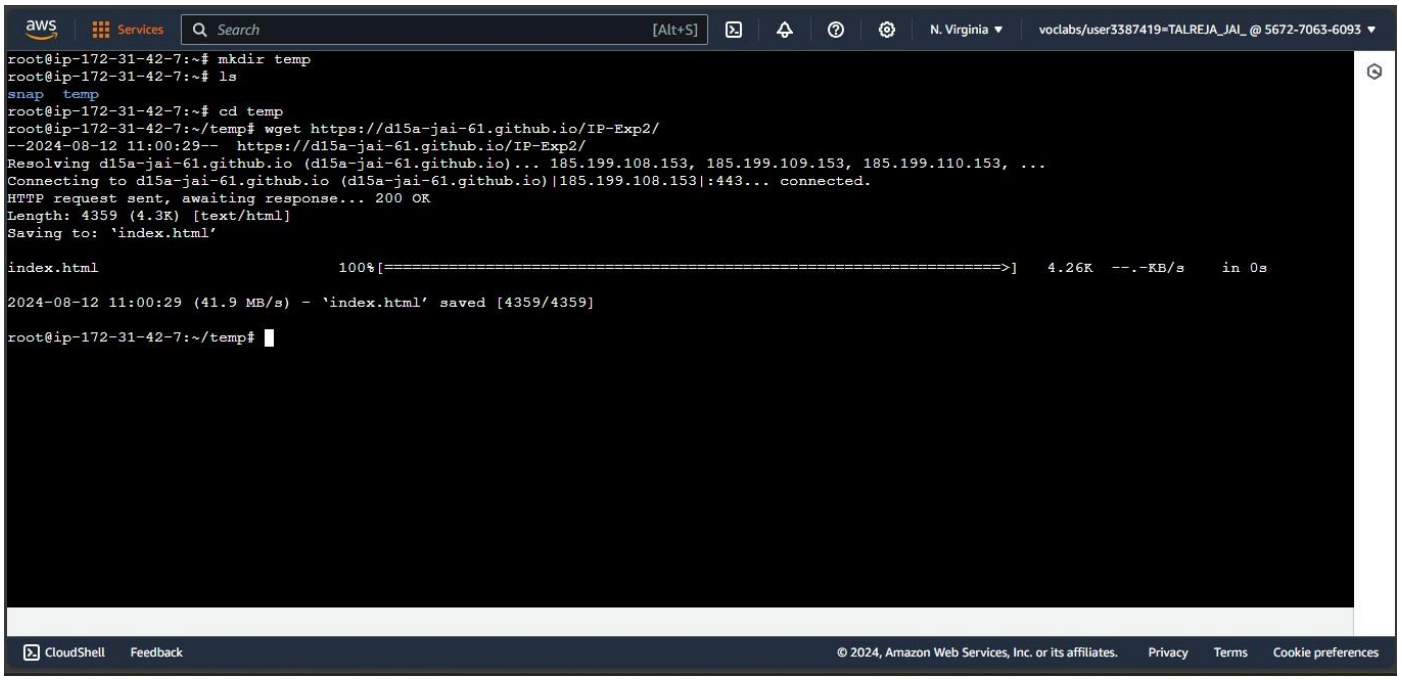
Jai Navani 31 D15A
Experiment-1

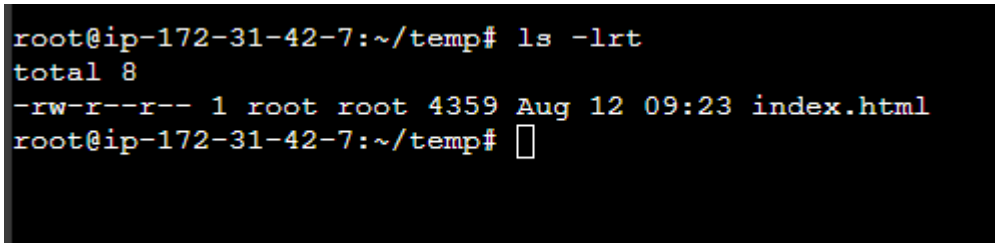
Screenshots for hosting webpage on EC2 instance :-

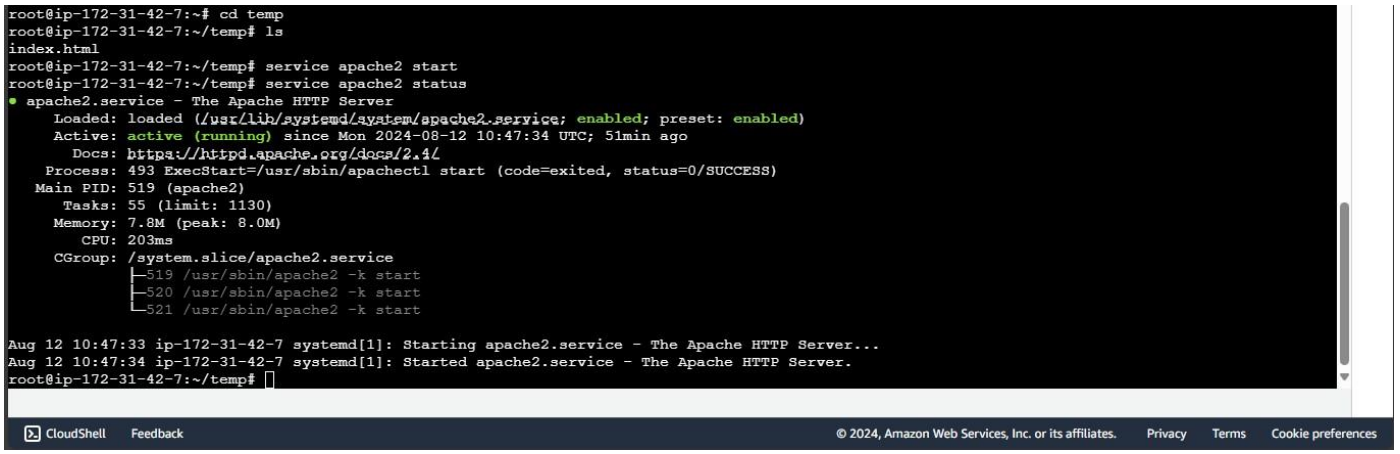




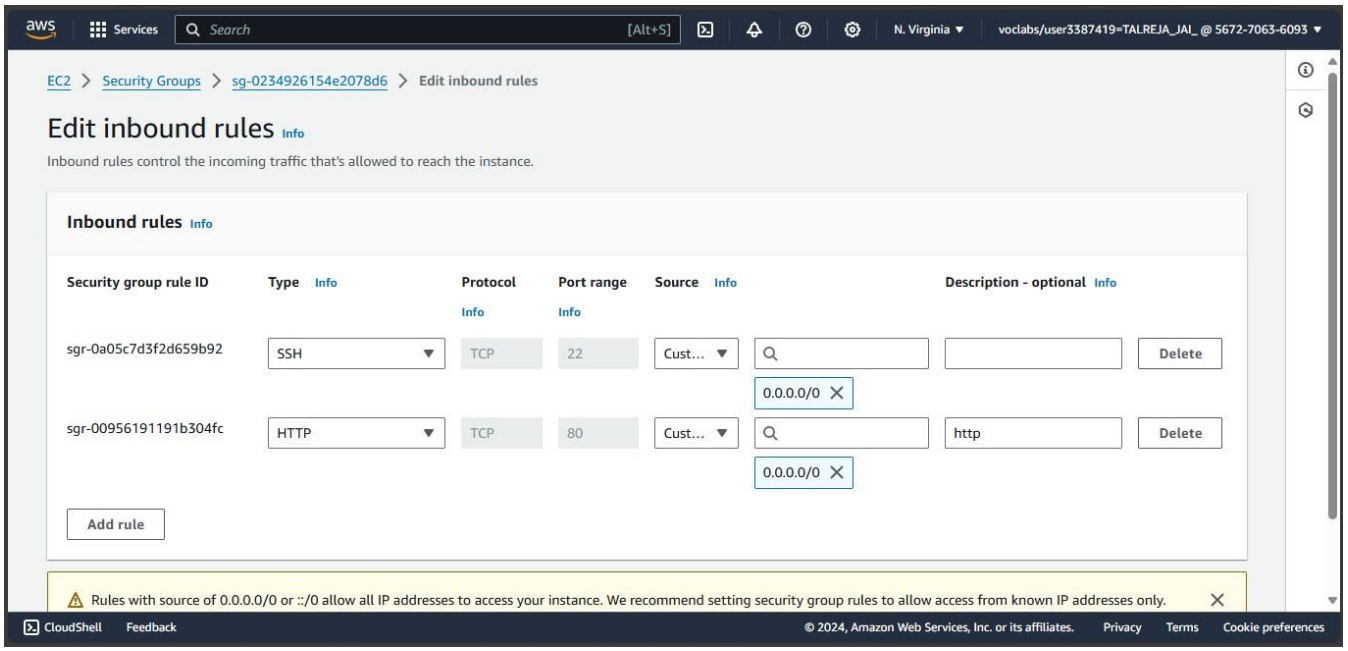
Jai Navani 31 D15A

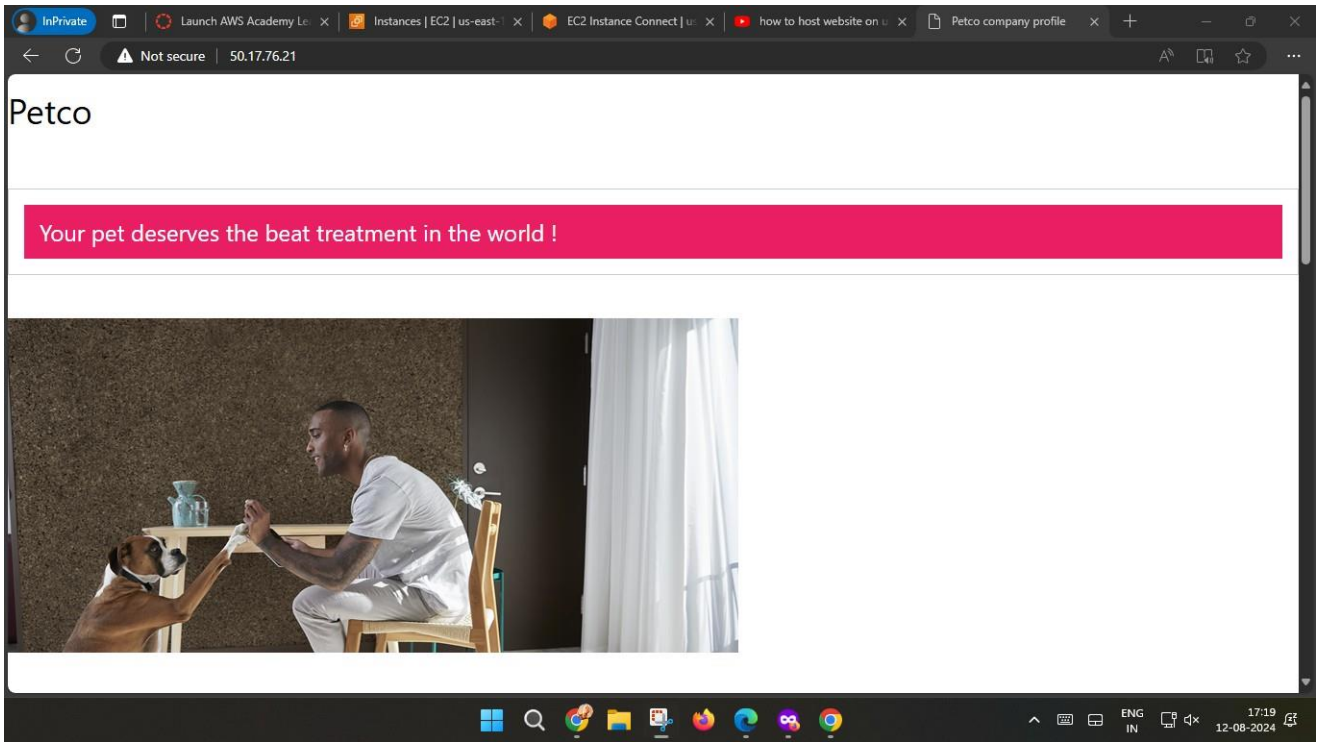




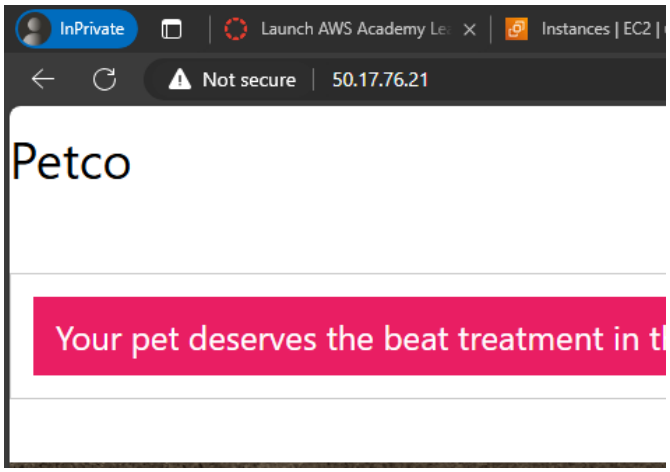


Jai Navani 31 D15A

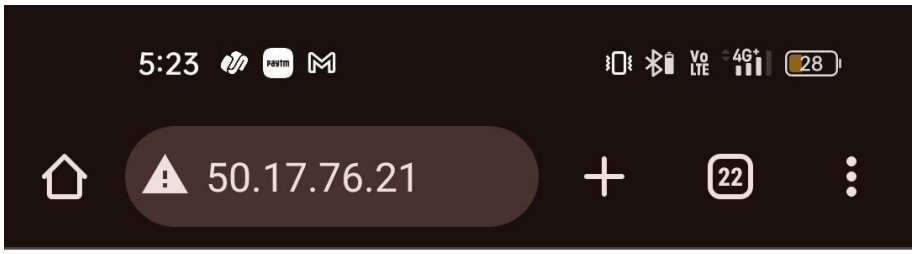




Jai Navani 31 D15A



Jai Navani 31 D15A



Petco

Your pet deserves the beat treatment in the world !

About us

Petco is a category-defining health and wellness company focused on improving the lives of pets, pet parents and our own Petco partners. Since our founding in 1965, we've been trailblazing new standards in pet care, delivering comprehensive wellness solutions through our products and services, and creating communities that deepen the pet-pet parent bond.

We employ more than 29,000 partners nationwide and operate more than 1,500 Petco locations across the U.S., Mexico and Puerto Rico — including a growing network of more than 200 in-store veterinary hospitals — and offer a complete online resource for pet health and wellness at petco.com and on the Petco app.

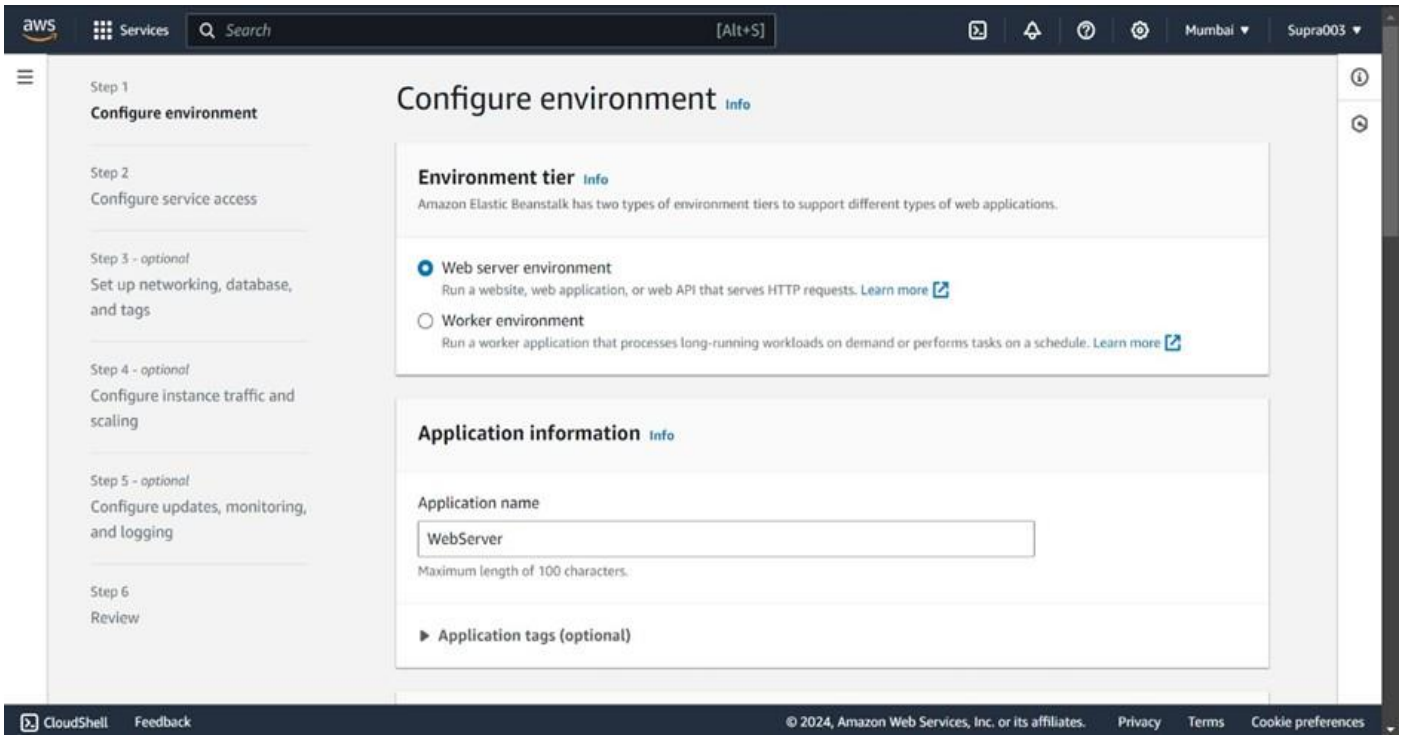
In tandem with Petco Love, an independent nonprofit organization, we have helped find homes for more than 7 million animals through in-store adoption events.

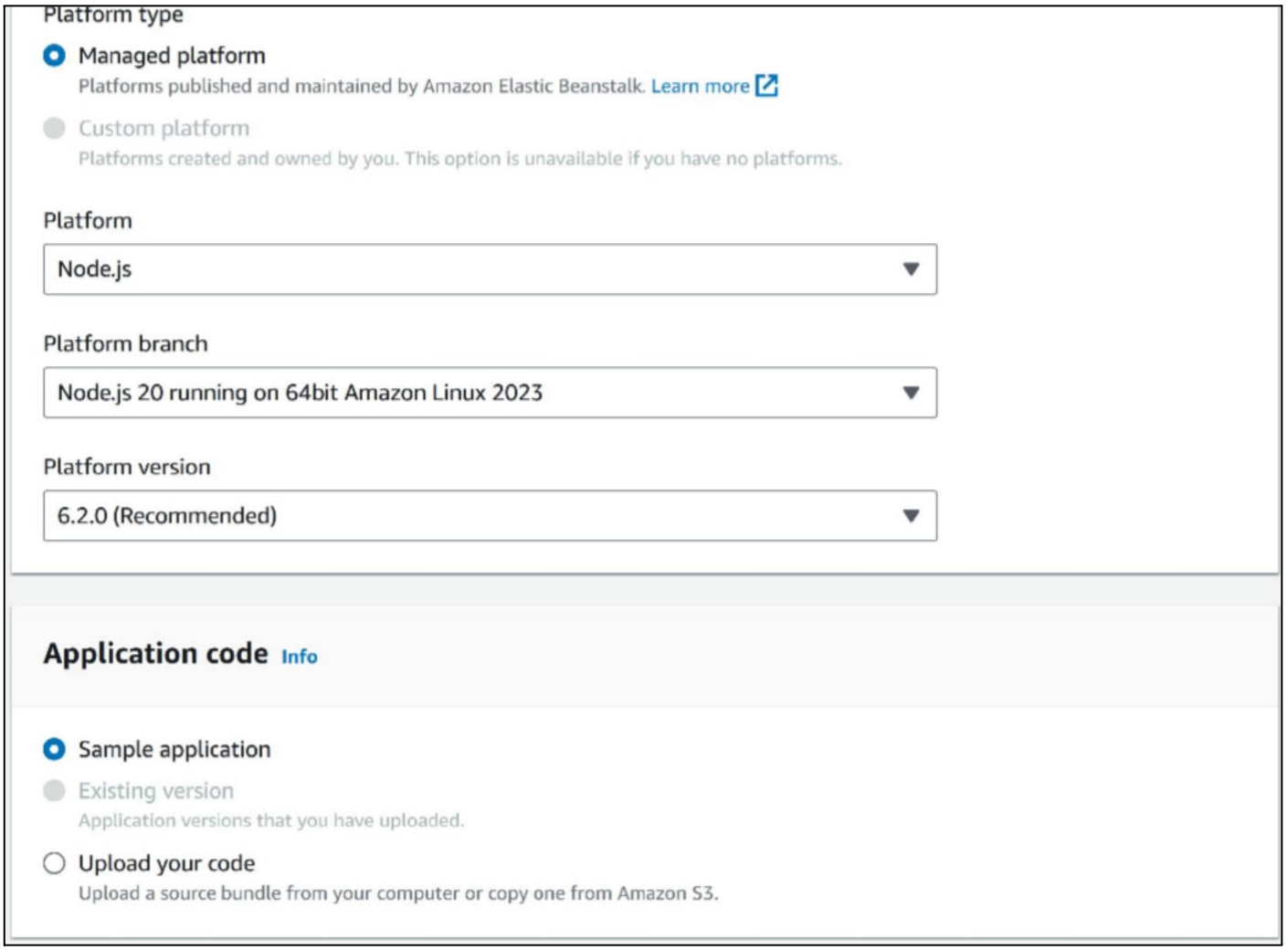
Working at Petco

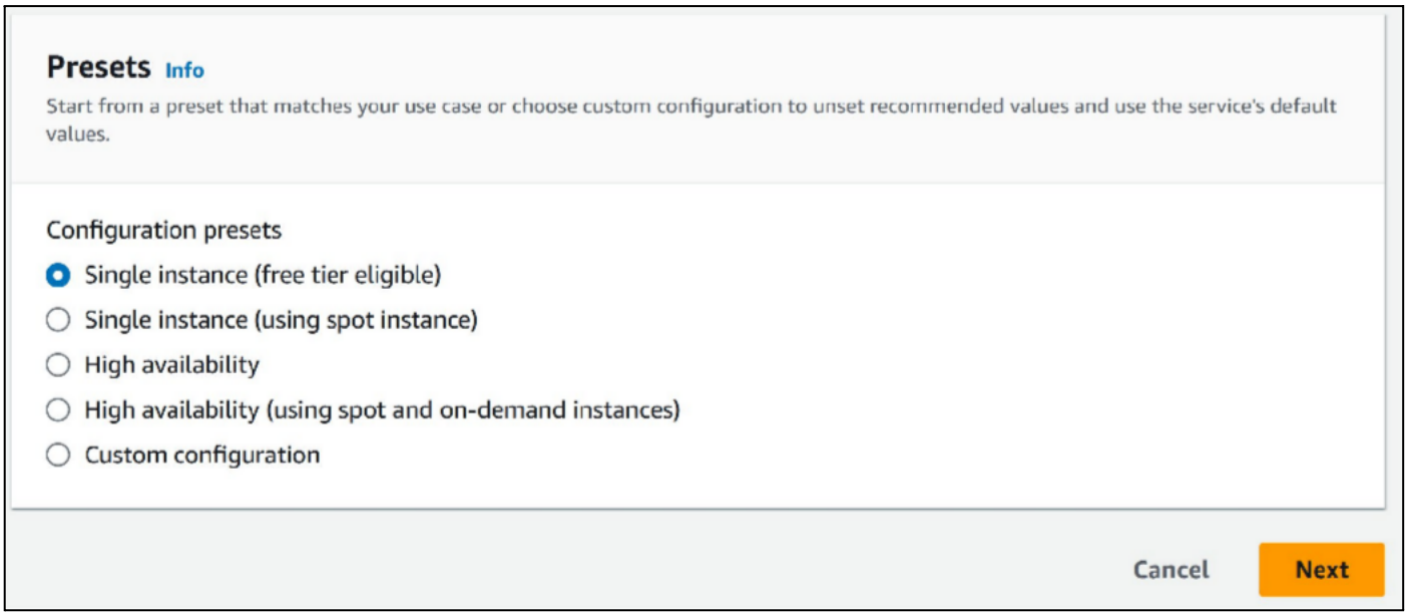
Jai Navani 31 D15A

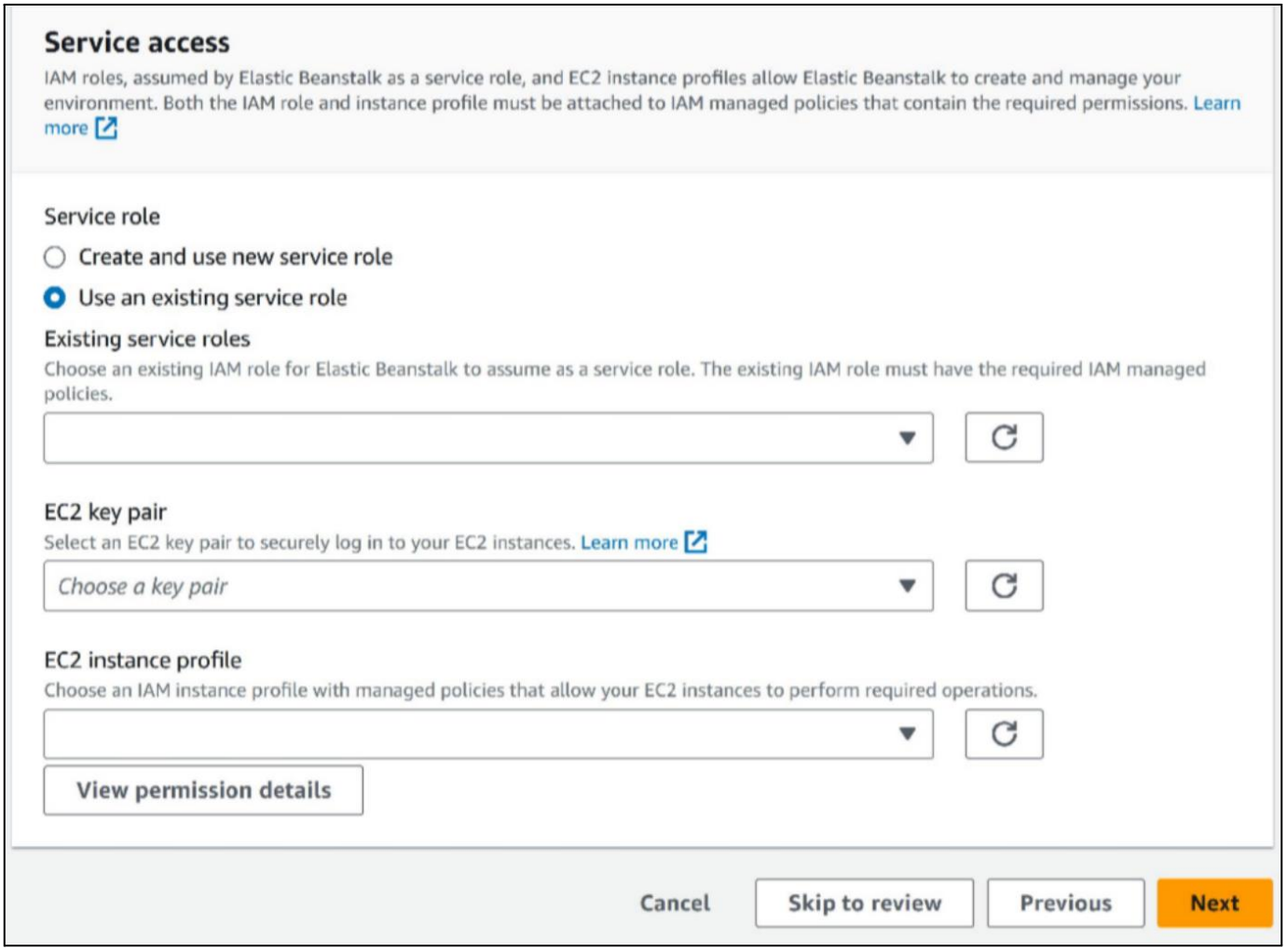
Experiment-2

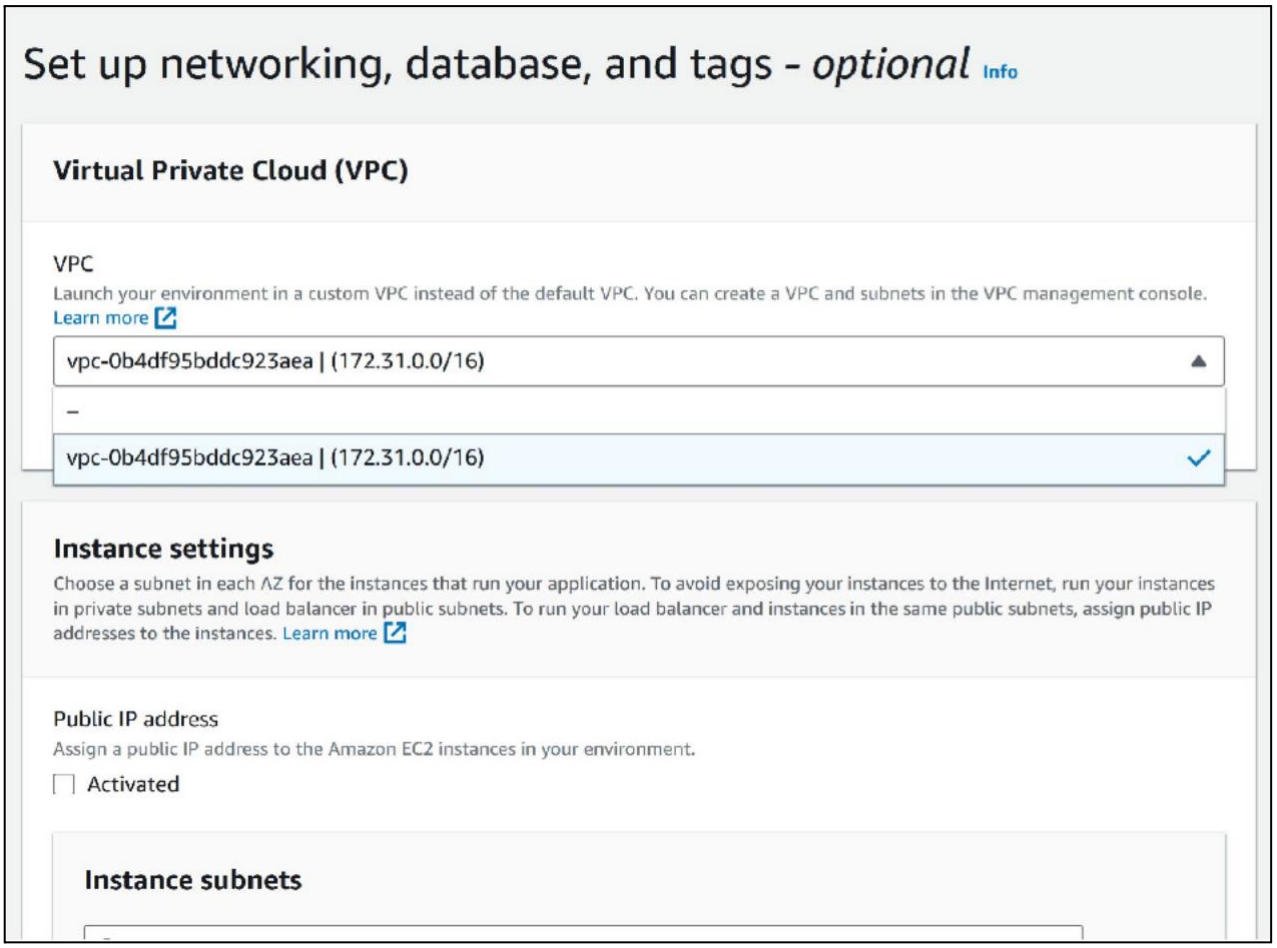
Screenshots for setting up and deploying Elastic Beanstalk application :-

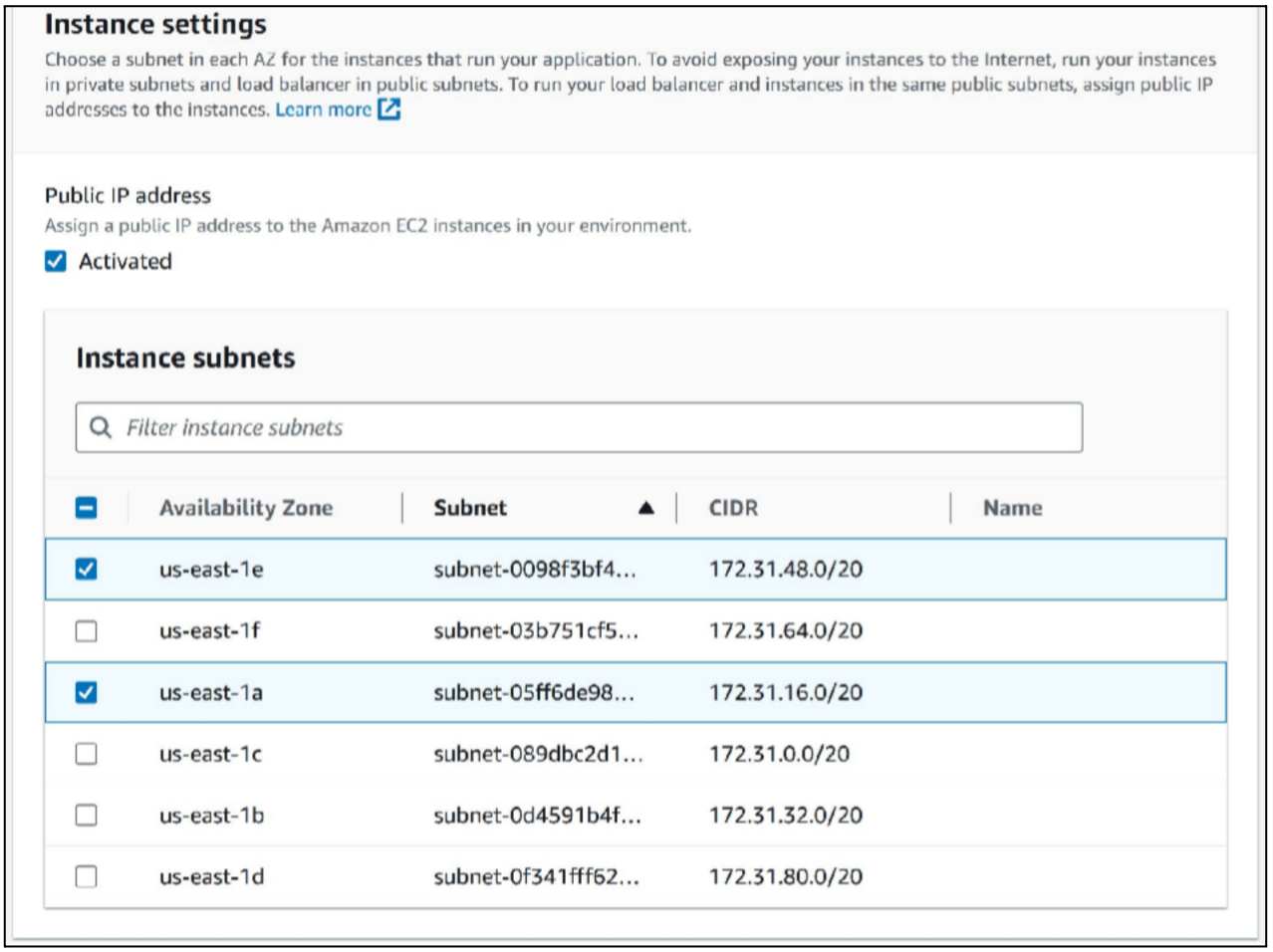





Presets Info
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
Configuration presets
Single instance (free tier eligible)
Single instance (using spot instance)
High availability
High availability (using spot and on-demand instances)
Custom configuration
Cancel
Next


Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more
Service role
Create and use new service role
Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.
EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. Learn more
Choose a key pair
EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
View permission details
Cancel
Skip to review
Previous
Next


Set up networking, database, and tags - optional Info
Virtual Private Cloud (VPC)
VPC
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.
Learn more
vpc-0b4df95bddc923aea | (172.31.0.0/16)
–
vpc-0b4df95bddc923aea | (172.31.0.0/16)
Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
Activated
Instance subnets


Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
Activated
Instance subnets
Filter instance subnets
Availability Zone
Subnet
CIDR
Name
us-east-1e subnet-0098f3bf4... 172.31.48.0/20
us-east-1f subnet-03b751cf5... 172.31.64.0/20
us-east-1a subnet-05ff6de98... 172.31.16.0/20
us-east-1c subnet-089dbc2d1... 172.31.0.0/20
us-east-1b subnet-0d4591b4f... 172.31.32.0/20
us-east-1d subnet-0f341fff62... 172.31.80.0/20

☐	us-east-1f	subnet-03b751cf5...	172.31.64.0/20
☐	us-east-1a	subnet-05ff6de98...	172.31.16.0/20
☐	us-east-1c	subnet-089dbc2d1...	172.31.0.0/20
☐	us-east-1b	subnet-0d4591b4f...	172.31.32.0/20
☐	us-east-1d	subnet-0f341fff62...	172.31.80.0/20

Enable database

Restore a snapshot - *optional*

Restore an existing snapshot from a previously used database.

Snapshot

None

Database settings

Choose an engine and instance type for your environment's database.

Engine

Engine version

Tags

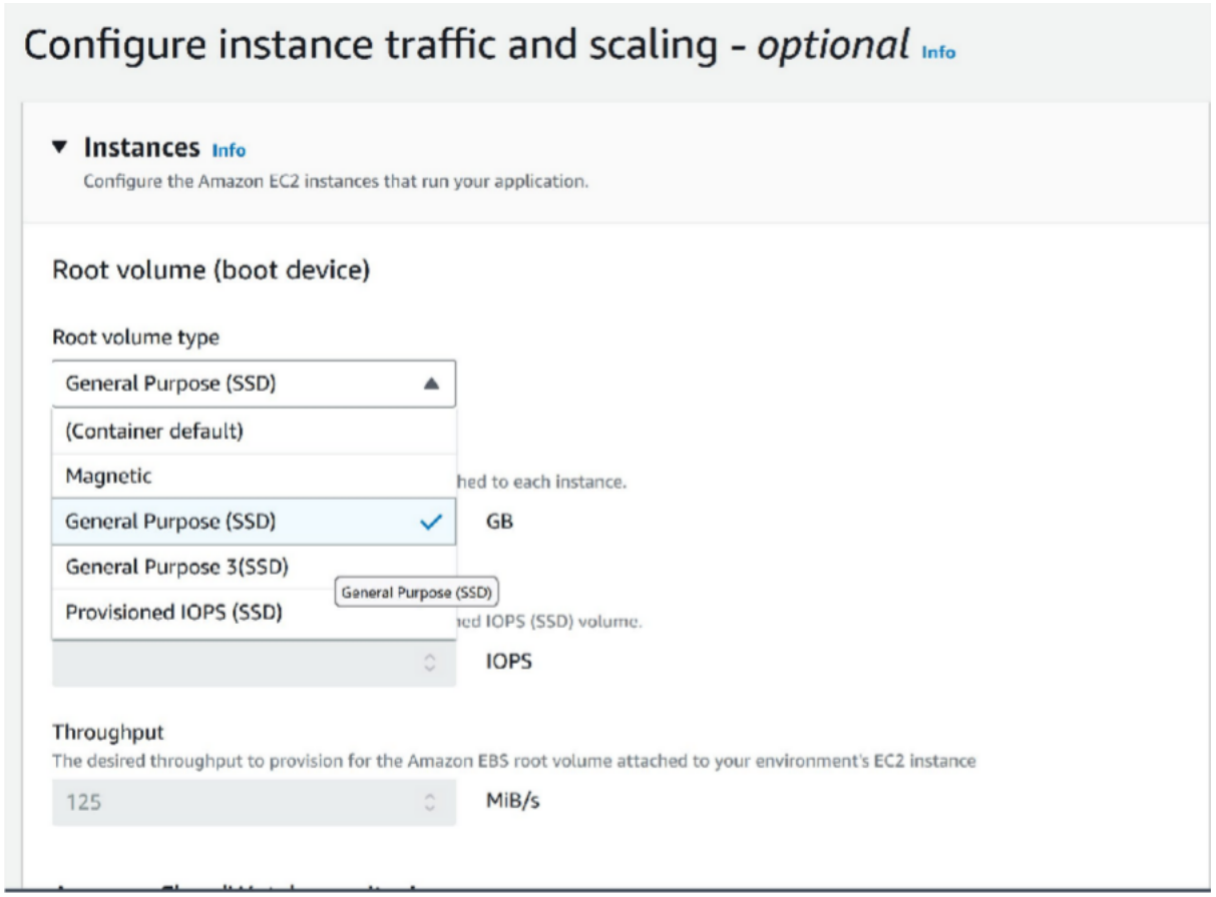
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel | Skip to review | Previous | Next



Root volume type

General Purpose (SSD)

Size

The number of gigabytes of the root volume attached to each instance.

8 GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125 MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

5 minute

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more

IMDSv1

With the current setting, the environment enables only IMDSv2.

☑ Deactivated

EC2 security groups

Select security groups to control traffic.

EC2 security groups (3)

Filter security groups

	Group name	Group ID	Name
☑	default	sg-0b6f2684b3eea9987	
☐	launch-wizard-1	sg-0234926154c2078d6	
☐	launch-wizard-2	sg-0f76834772eb69439	

Capacity Info

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type

Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. Learn more

Single instance

Instances

1 Min

1 Max

Fleet composition

Spot instances are launched at the lowest available price. Learn more

◉ On-Demand instance

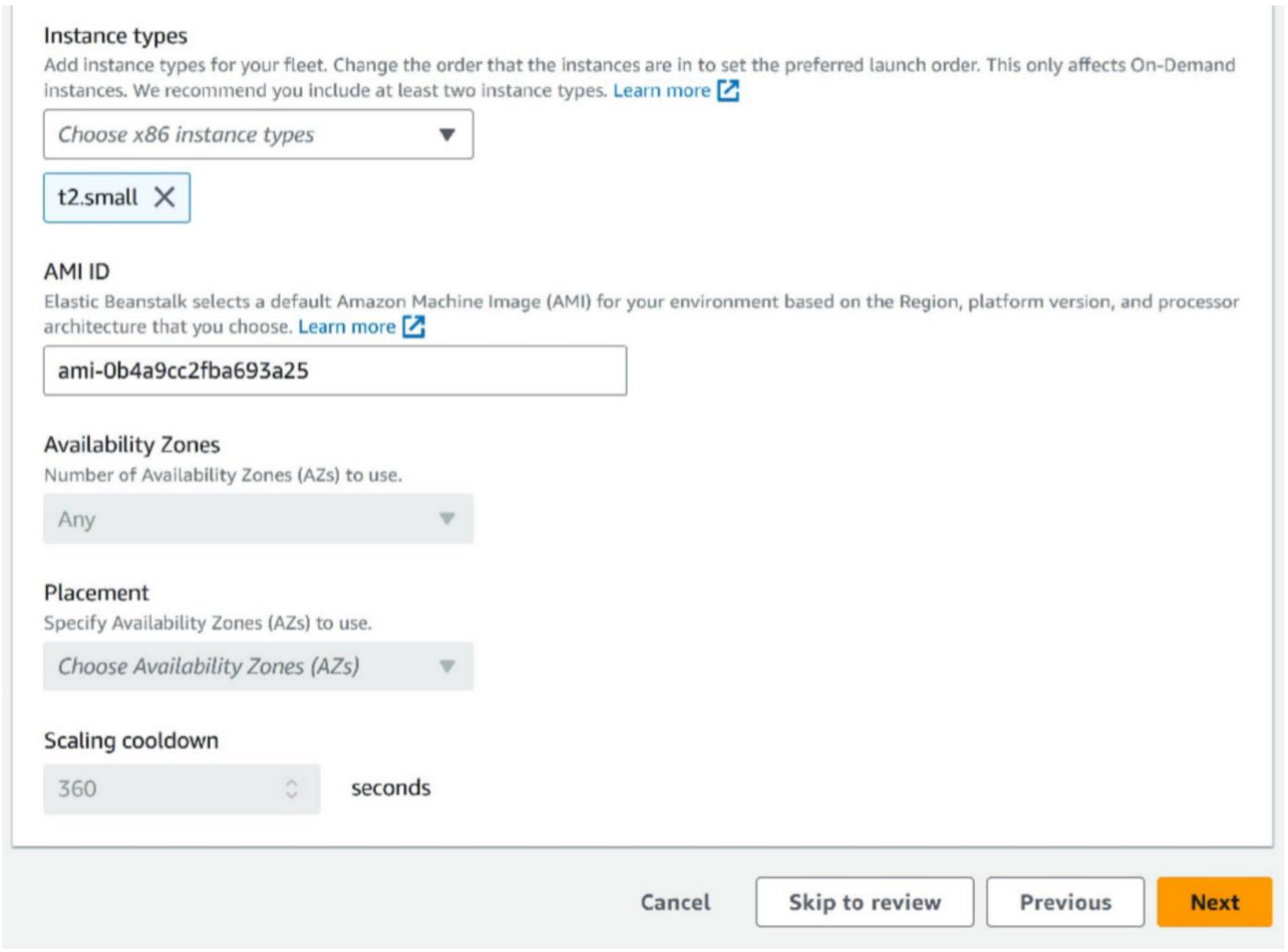
○ Spot instance

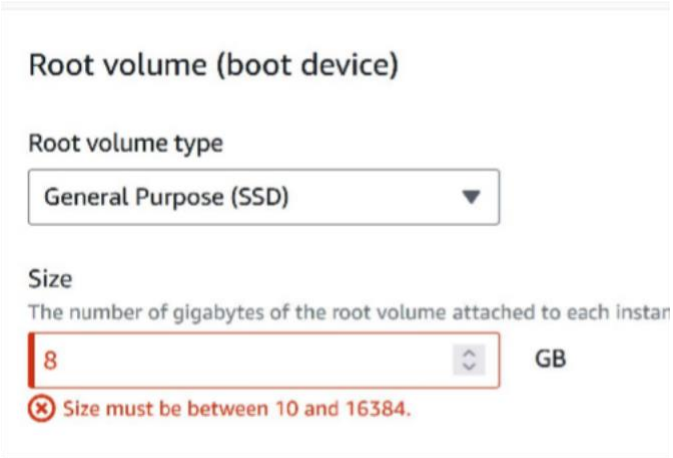
Maximum spot price

The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.

○ Default

○ Set your maximum price


Instance types
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. Learn more
Choose x86 instance types
t2.small
AMI ID
Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. Learn more
ami-0b4a9cc2fba693a25
Availability Zones
Number of Availability Zones (AZs) to use.
Any
Placement
Specify Availability Zones (AZs) to use.
Choose Availability Zones (AZs)
Scaling cooldown
360
seconds
Cancel
Skip to review
Previous
Next


Root volume (boot device)
Root volume type
General Purpose (SSD)
Size
The number of gigabytes of the root volume attached to each instar
8
GB
Size must be between 10 and 16384.


Root volume type
General Purpose (SSD)
Size
The number of gigabytes of the root volume attached to each instance.
16
GB

Configure updates, monitoring, and logging - *optional* Info

Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing

System

- (•) Basic
- ( ) Enhanced

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

- [ ] Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

Managed platform updates Info

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

- [ ] Activated

Weekly update window

Tuesday at 01 : 14 UTC

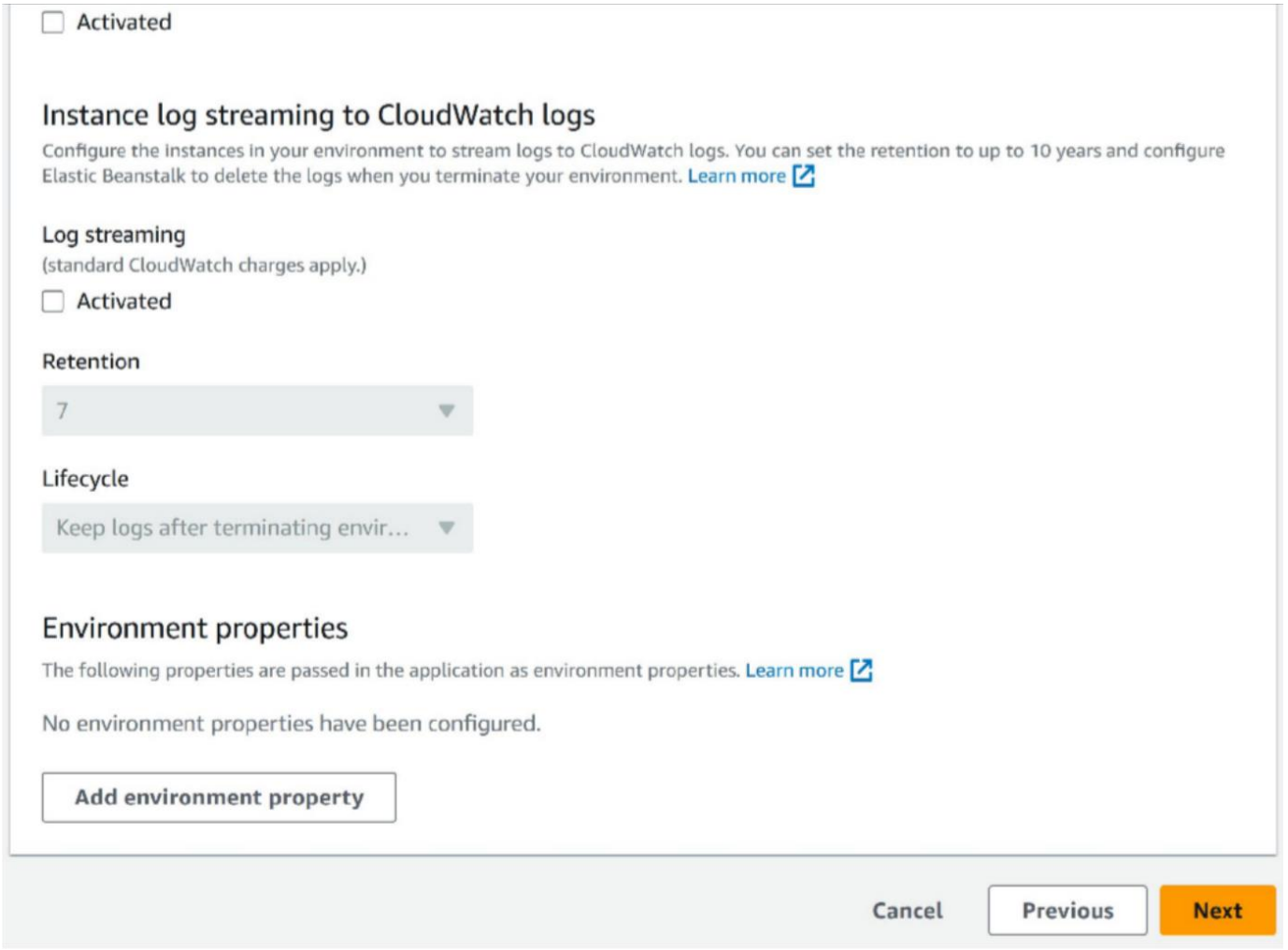
Update level

Minor and patch

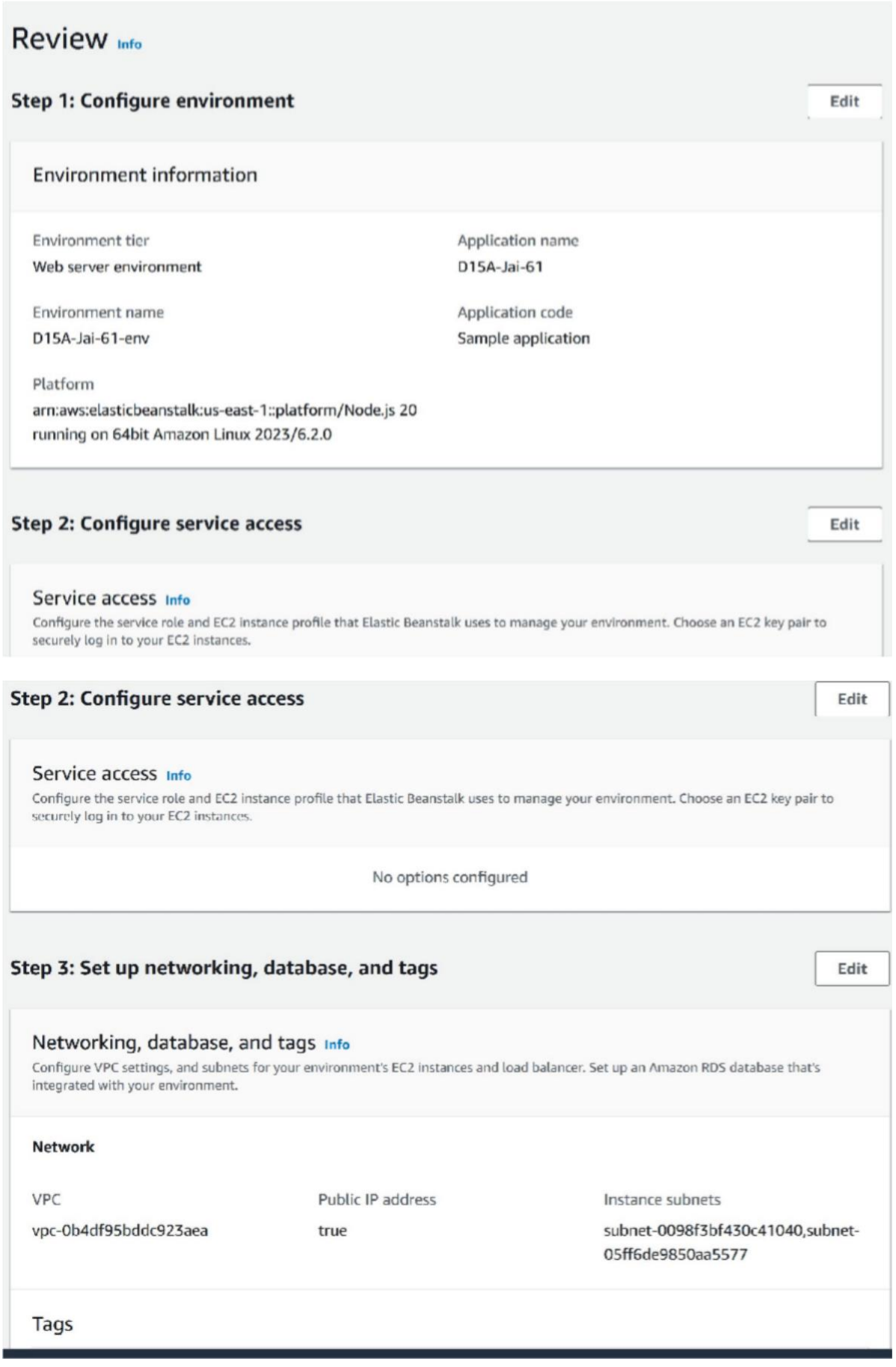
Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

- [ ] Activated


Activated
Instance log streaming to CloudWatch logs
Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more
Log streaming
(standard CloudWatch charges apply.)
Activated
Retention
7
Lifecycle
Keep logs after terminating envir...
Environment properties
The following properties are passed in the application as environment properties. Learn more
No environment properties have been configured.
Add environment property
Cancel
Previous
Next

Elastic Beanstalk application created, review screen before finalization :-



Step 4: Configure instance traffic and scaling

Edit

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic.Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

Root volume type	Instance size	IMDSv1
gp2	16	Deactivated

EC2 Security Groups

sg-0b6f2684b3eea9987

Capacity

Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	**Capacity rebalancing**	**Scaling cooldown**
0	Deactivated	360
Processor type	**Instance types**	**AMI ID**
x86_64	t2.small	ami-0b4a9cc2fba693a25

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
basic	—	—
Log streaming	**Retention**	**Lifecycle**
Deactivated	7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Deactivated	100	Percentage
Command timeout	**Deployment policy**	**Health threshold**
600	AllAtOnce	Ok
Ignore health check	**Instance replacement**	

Ignore health check
false

Instance replacement
false

Platform software

Lifecycle
false

Log streaming
Deactivated

Proxy server
nginx

Logs retention
7

Rotate logs
Deactivated

Update level
minor

X-Ray enabled
Deactivated

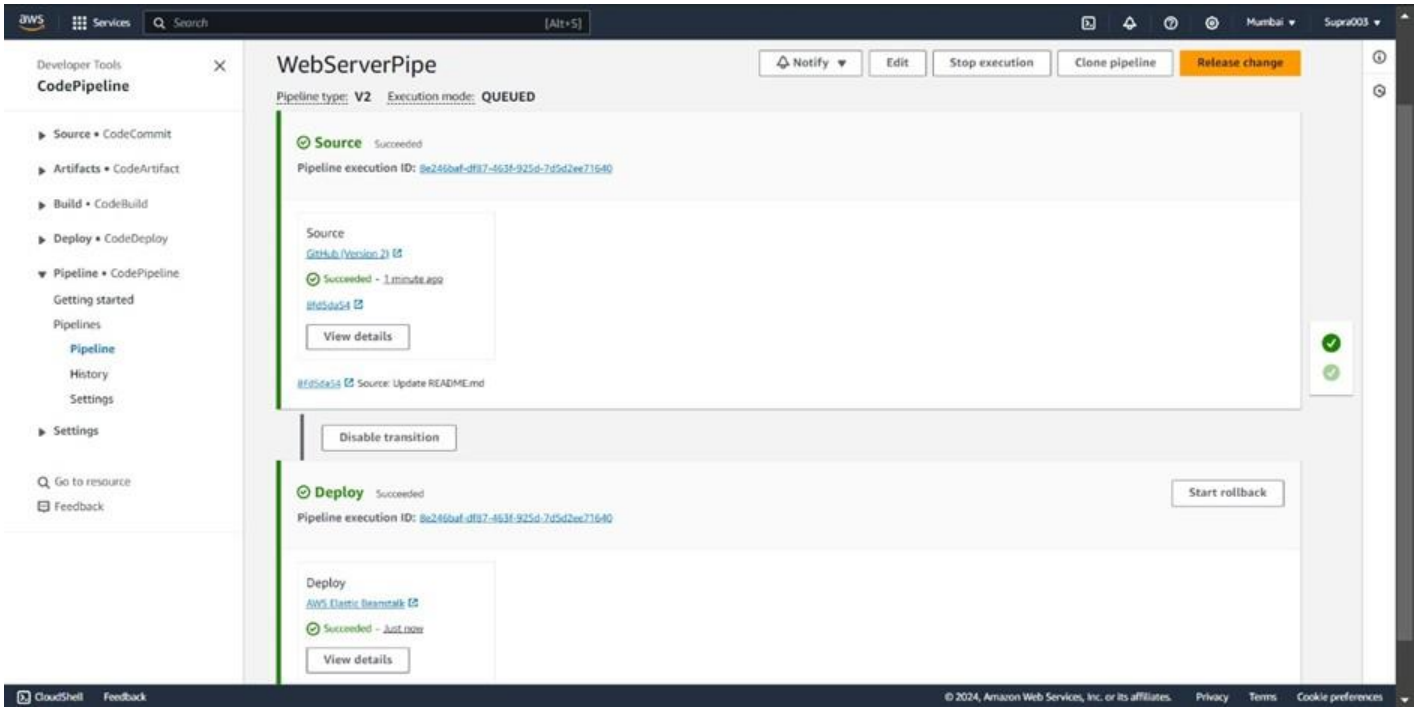
Environment properties

Key	Value

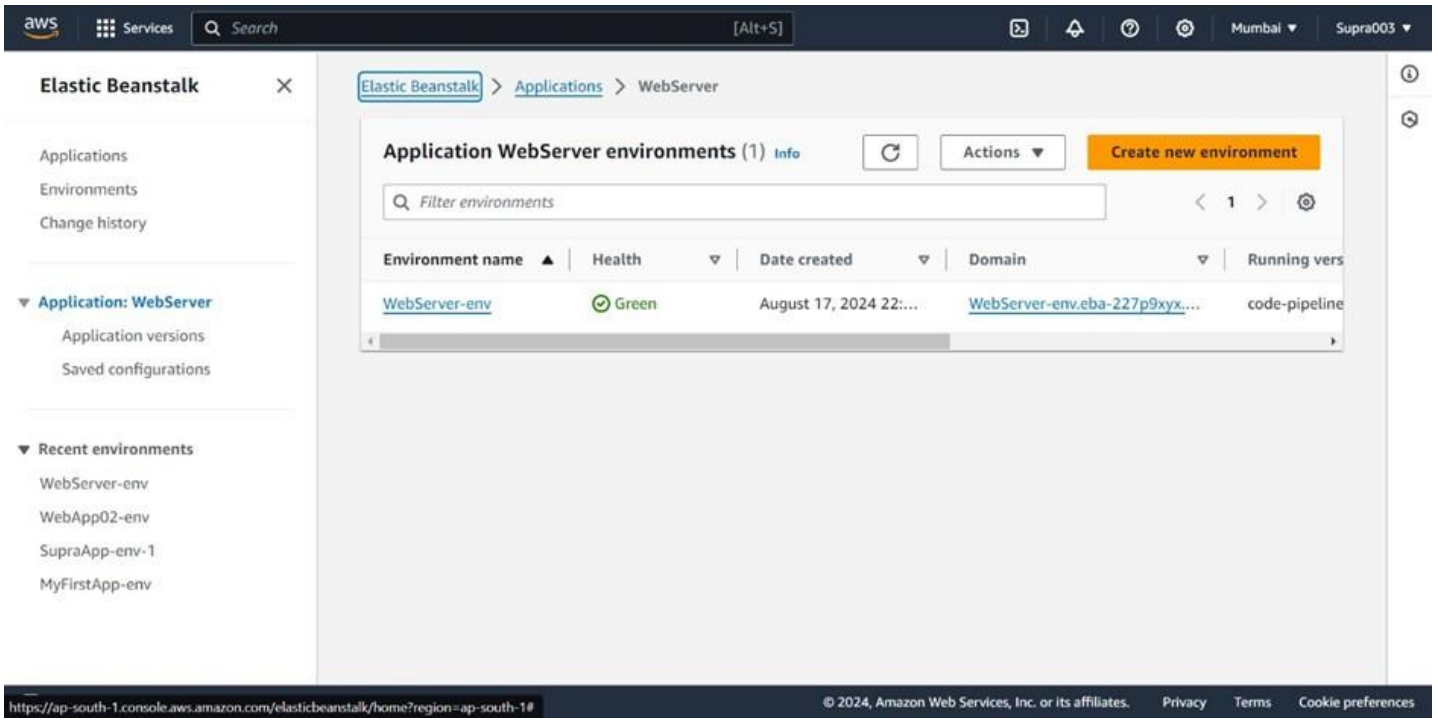
No environment properties

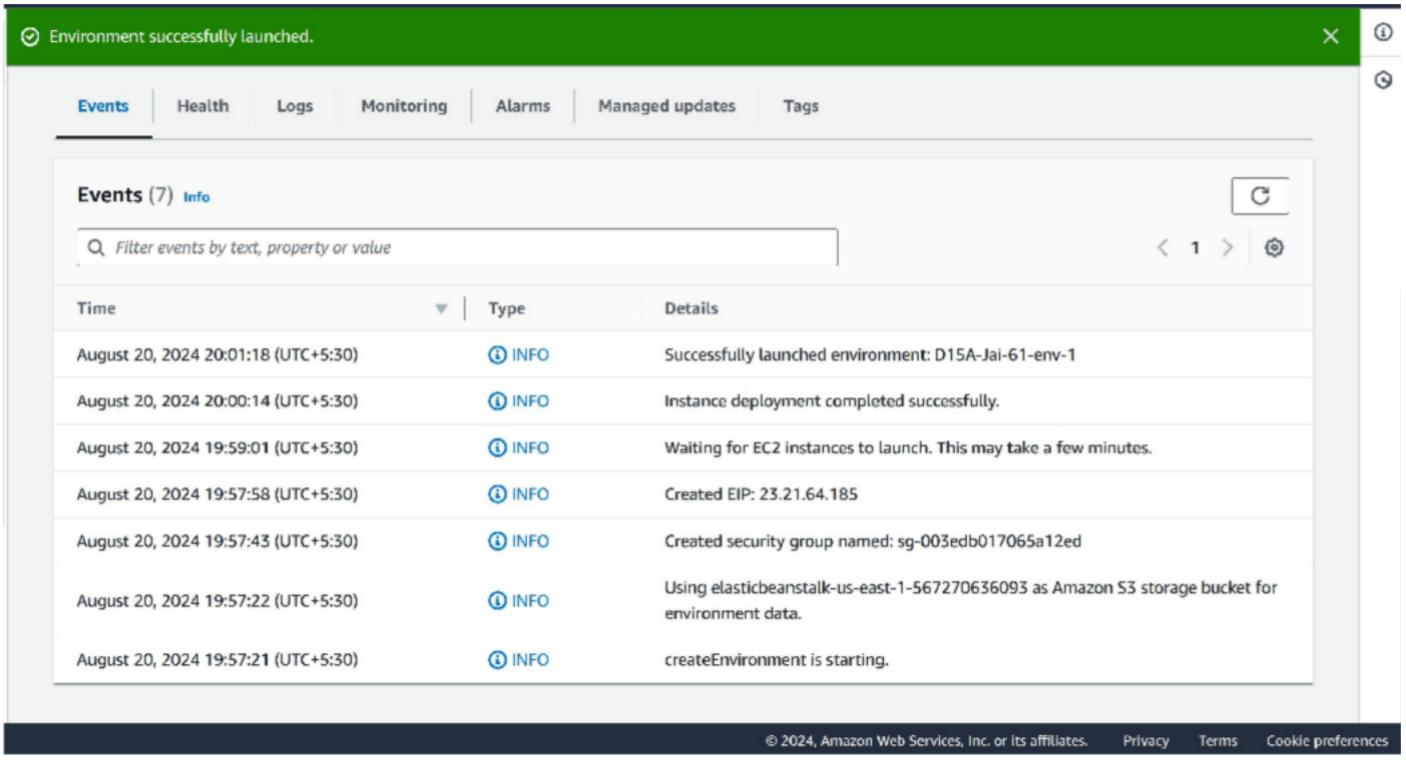
There are no environment properties defined

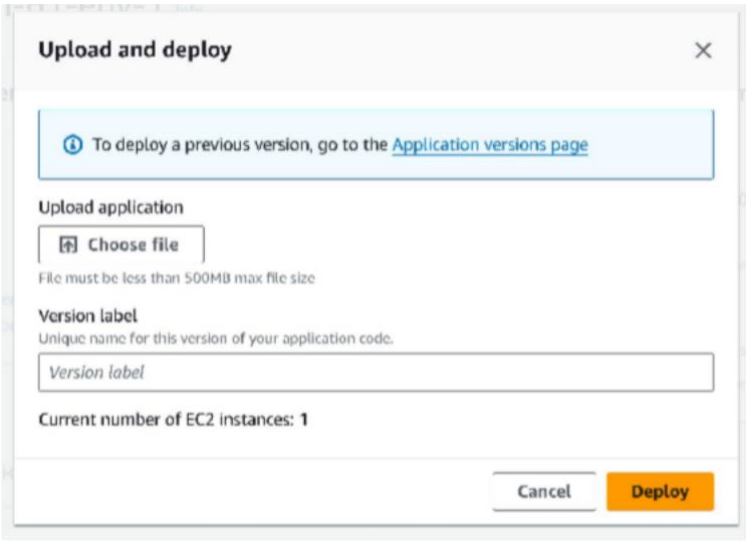
Cancel | Previous | Submit

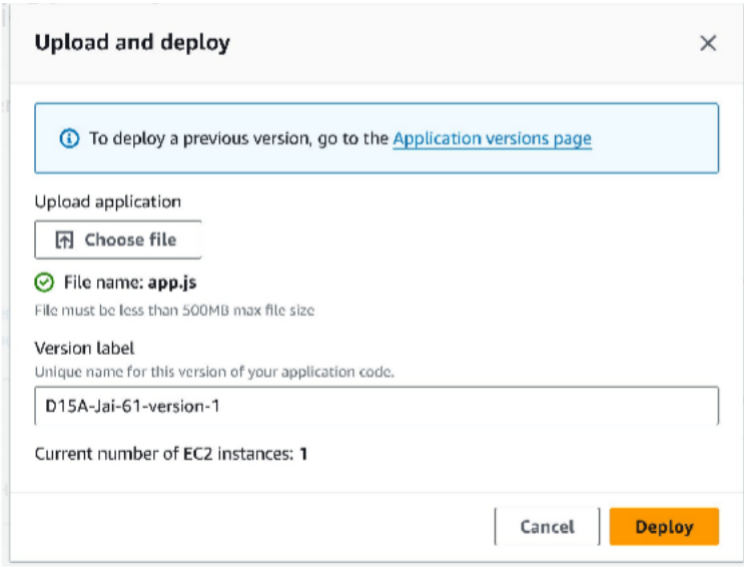


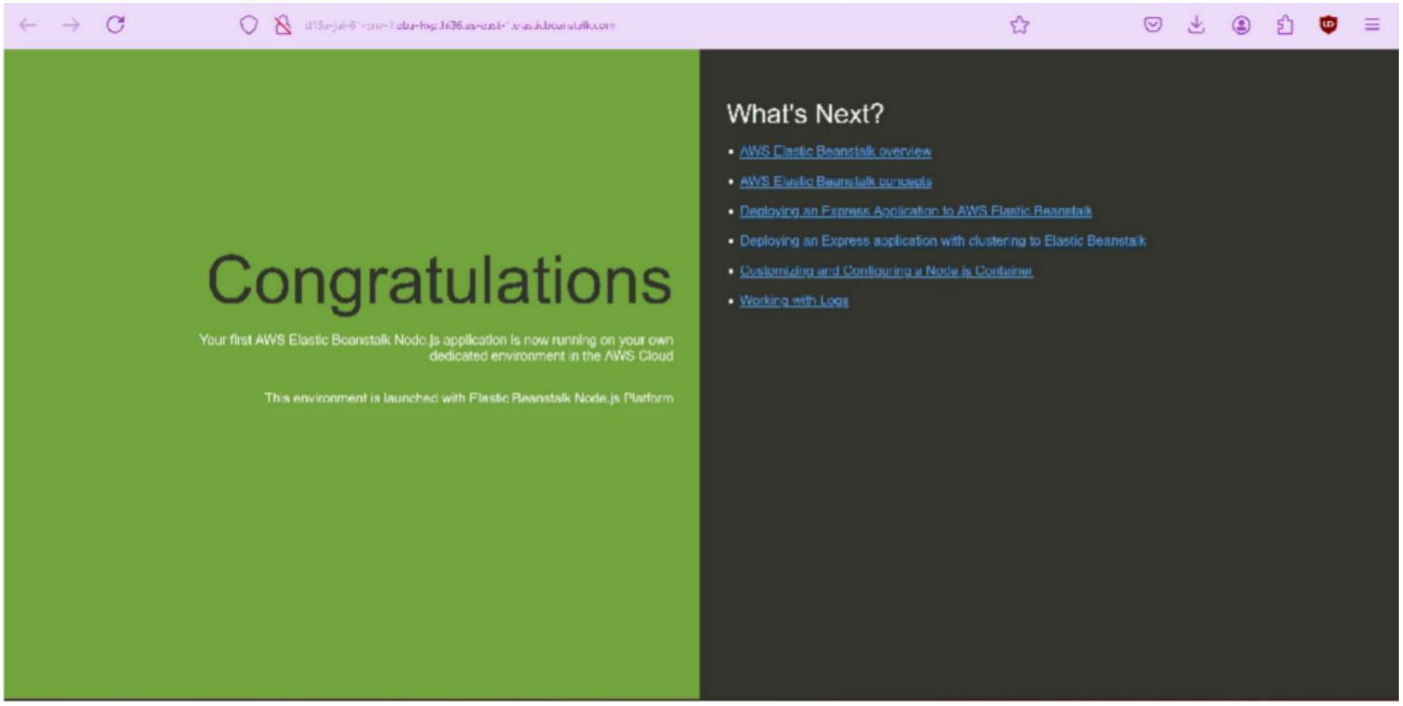
Jai Navani 31 D15A





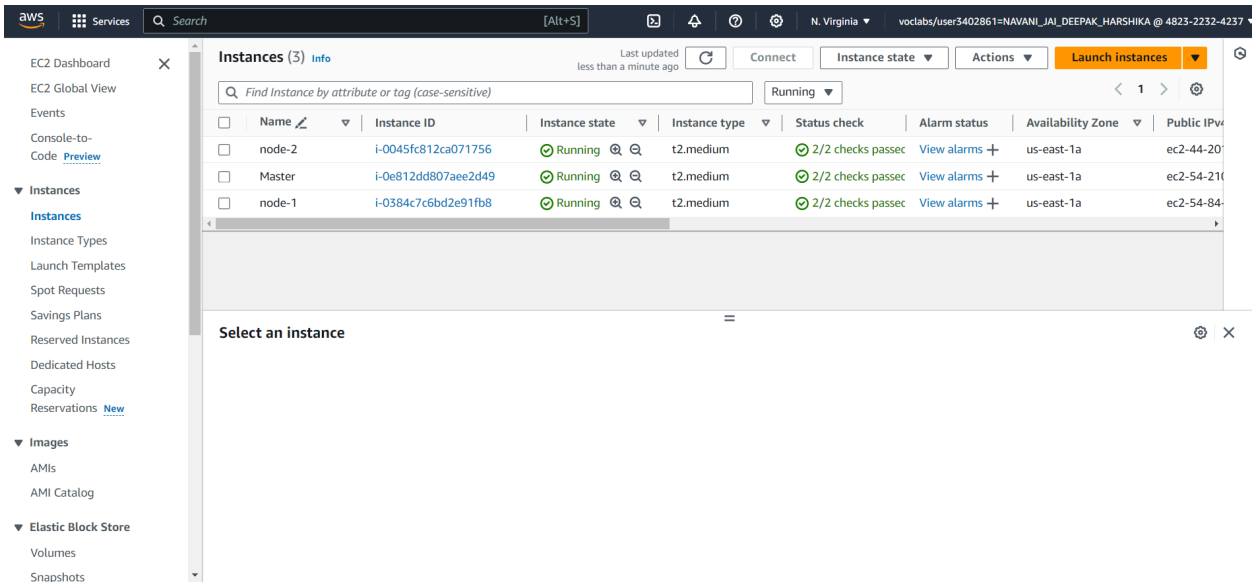

Upload and deploy
To deploy a previous version, go to the Application versions page
Upload application
Choose file
File must be less than 500MB max file size
Version label
Unique name for this version of your application code.
Version label
Current number of EC2 instances: 1
Cancel
Deploy


Upload and deploy
To deploy a previous version, go to the Application versions page
Upload application
Choose file
File name: app.js
File must be less than 500MB max file size
Version label
Unique name for this version of your application code.
D15A-Jai-61-version-1
Current number of EC2 instances: 1
Cancel
Deploy


Congratulations
Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud
This environment is launched with Elastic Beanstalk Node.js Platform
What's Next?
AWS Elastic Beanstalk overview
AWS Elastic Beanstalk tutorials
Deploying an Express Application to AWS Elastic Beanstalk
Deploying an Express application with clustering to Elastic Beanstalk
Customizing and Configuring a Node.js Container
Working with Logs

Experiment no:3
Advance-devops
Jai navani
D15-A

Step 1: create instances (1 master and 2 nodes)



Connect the instances:

Install docker in all instances

```
[root@ip-172-31-88-48 ec2-user]# yum install docker -y
Last metadata expiration check: 0:04:33 ago on Sun Sep 22 07:37:42 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    Repository     Size
================================================================================
Installing:
 docker                  x86_64        25.0.6-1.amzn2023.0.2      amazonlinux    44 M
Installing dependencies:
 containerd              x86_64        1.7.20-1.amzn2023.0.1      amazonlinux    35 M
 iptables-libs           x86_64        1.8.8-3.amzn2023.0.2       amazonlinux   401 k
 iptables-nft            x86_64        1.8.8-3.amzn2023.0.2       amazonlinux   183 k
 libcgroup               x86_64        3.0-1.amzn2023.0.1         amazonlinux    75 k
 libnetfilter_conntrack  x86_64        1.0.8-2.amzn2023.0.2       amazonlinux    58 k
 libnfnetlink            x86_64        1.0.1-19.amzn2023.0.2      amazonlinux    30 k
 libnftnl                x86_64        1.2.2-2.amzn2023.0.2       amazonlinux    84 k
 pigz                    x86_64        2.5-1.amzn2023.0.3         amazonlinux    83 k
 runc                    x86_64        1.1.13-1.amzn2023.0.1      amazonlinux   3.2 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
```

After installation start the docker:

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64          docker-25.0.6-1.amzn2023.0.2.x86_64          iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64         libcgroup-3.0-1.amzn2023.0.1.x86_64          libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64        libnftnl-1.2.2-2.amzn2023.0.2.x86_64         pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-88-48 ec2-user]# systemctl start docker
[root@ip-172-31-88-48 ec2-user]#
[root@ip-172-31-88-48 ec2-user]#
[root@ip-172-31-88-48 ec2-user]# systemctl start docker
```

Code for installation of kubernete:

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/

enabled=1

gpgcheck=1

gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key

exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni

EOF

# Set SELinux in permissive mode (effectively disabling it)

sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

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

Output (in all instances):

```
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64                  kubeadm-1.31.1-150500.1.1.x86_64
  kubectl-1.31.1-150500.1.1.x86_64                     kubelet-1.31.1-150500.1.1.x86_64                    kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-88-48 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
```

```
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-88-241 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
[root@ip-172-31-88-241 ec2-user]#
```

i-0384c7c6bd2e91fb8 (node-1)

PublicIPs: 54.84.57.3 PrivateIPs: 172.31.88.241

```
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-89-83 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
[root@ip-172-31-89-83 ec2-user]#
```

i-0045fc812ca071756 (node-2)

PublicIPs: 44.201.84.150 PrivateIPs: 172.31.89.83

Kubeadm init:
(after intialization):

```
Your Kubernetes control-plane has initialized successfully!

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

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

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

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.88.48:6443 --token 2nrq25.jqlp7gerx8nty6ys \
        --discovery-token-ca-cert-hash sha256:d969d1bb086d72a8c952b2b6904c7c6f8c7e42a1d12f6ef1a82a46935363e411
```

```
[root@ip-172-31-88-241 ec2-user]# sudo yum install iproute
Last metadata expiration check: 0:29:35 ago on Sun Sep 22 07:54:40 2024.
Package iproute-5.10.0-2.amzn2023.0.5.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-88-241 ec2-user]# kubeadm join 172.31.88.48:6443 --token 2nrq25.jqlp7gerx8nty6ys --discovery-token-ca-cert-hash sha256:d969d1bb086d72a8c952b2b6904
c7c6f8c7e42a1d12f6ef1a82a46935363e411
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
```

Jai Navani **EXPERIMENT 4**
D15A 31

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

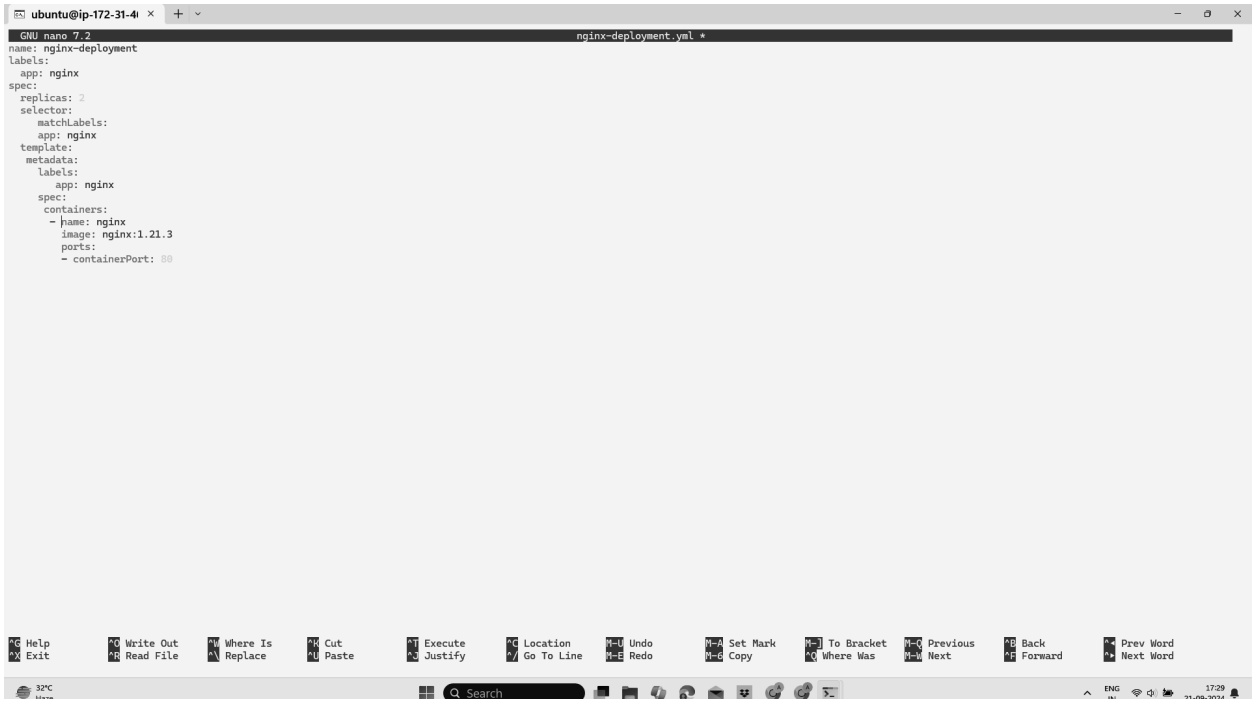
1. If you haven't already set up a Kubernetes cluster (e.g., with kubeadm), use minikube or any managed Kubernetes service (like EKS, GKE, etc.) to get a cluster running.

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

kubectl get nodes

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

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



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

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

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

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

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

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

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

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

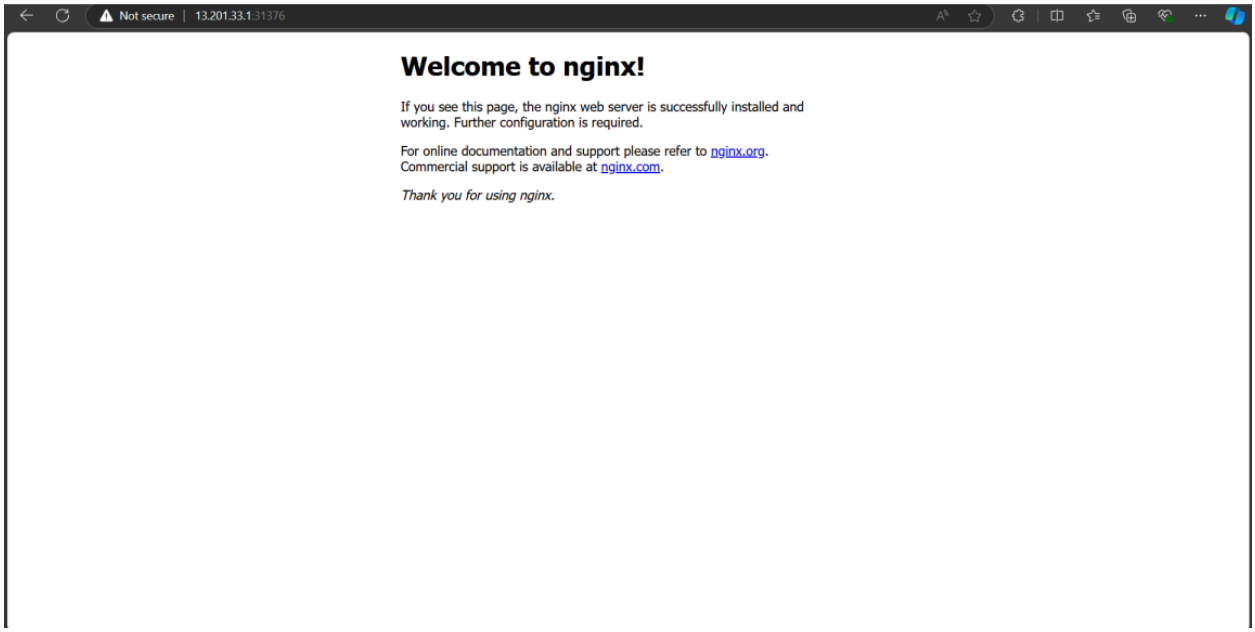
1. Forward the Service Port: Use the following command to forward a local port to the service's target port. kubectl port-forward service/ :
This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

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

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

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

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


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

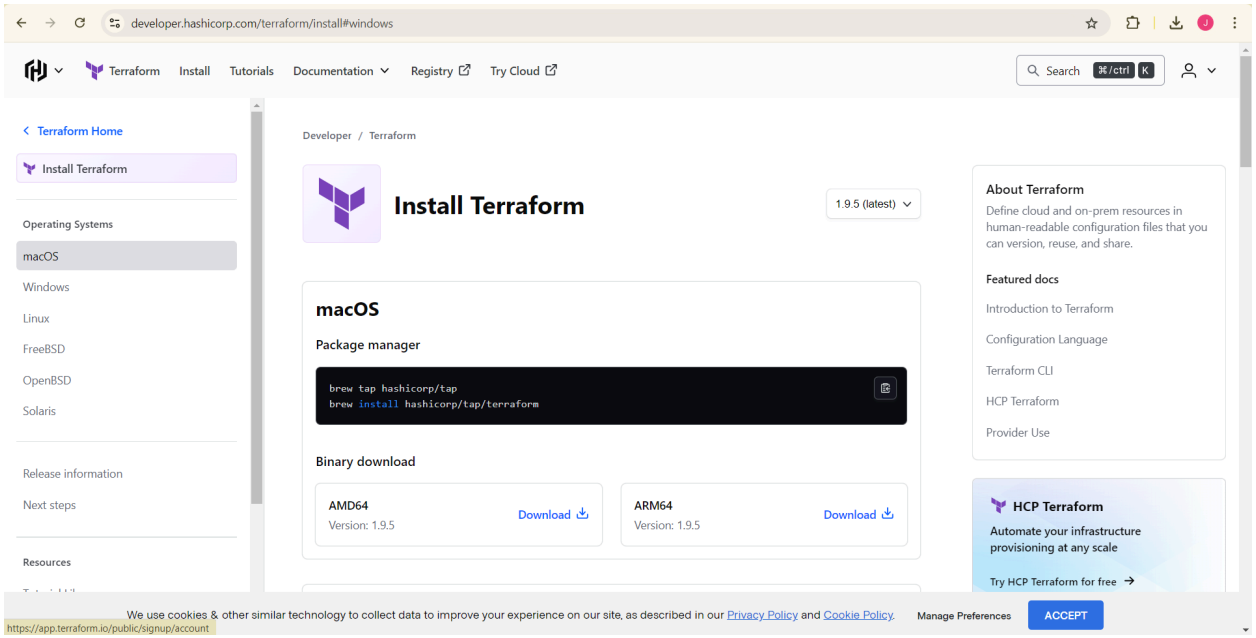
Advance devops Exp:5

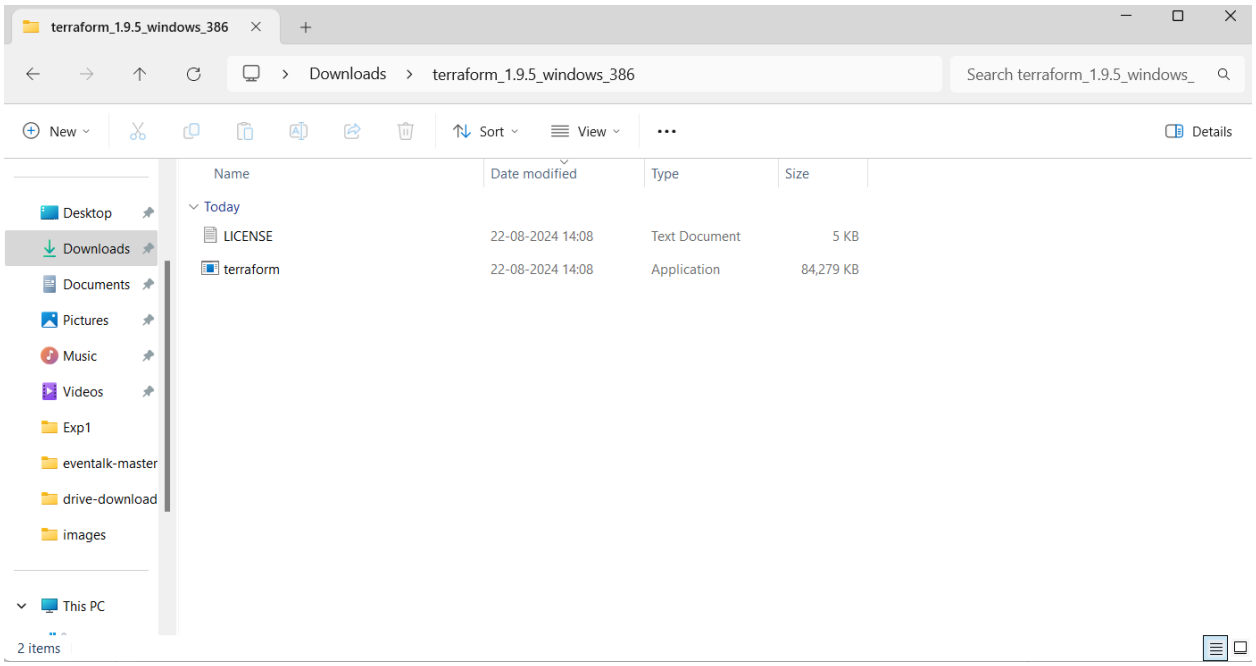
Jai Navani

D15 A-31

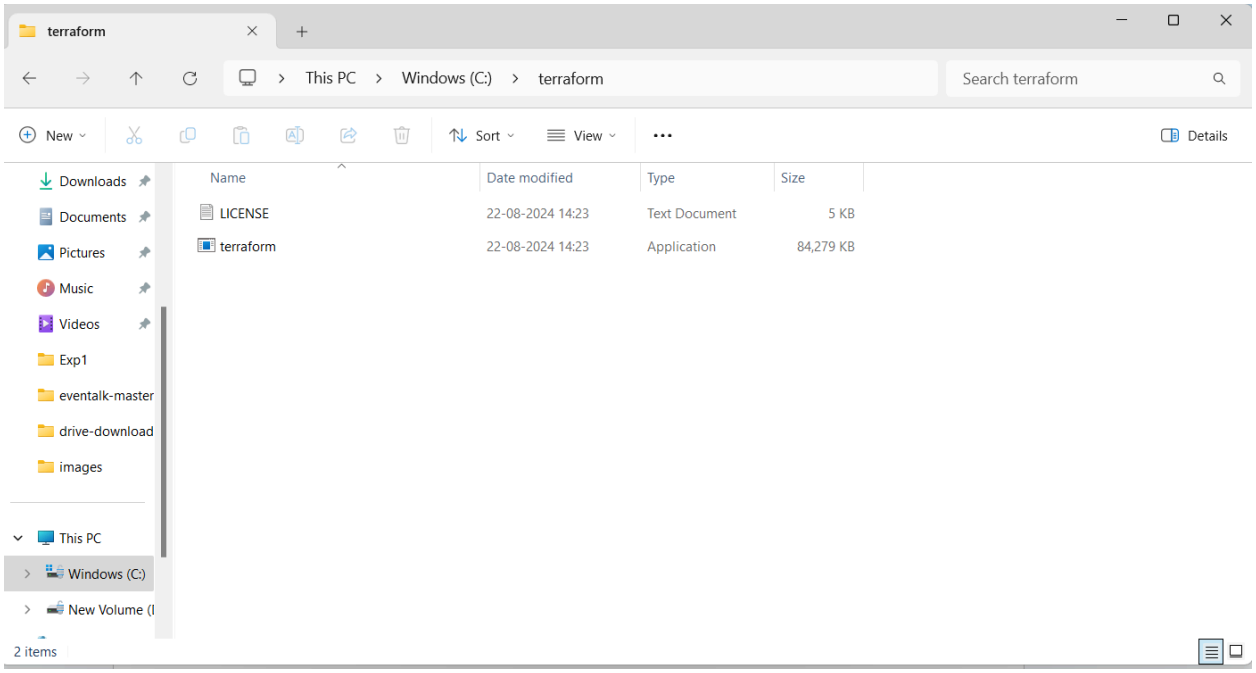
Aim: To understand terraform lifecycle, core concepts/terminologies and install it on a linux machine and windows

Step1: Download Terraform from the official website

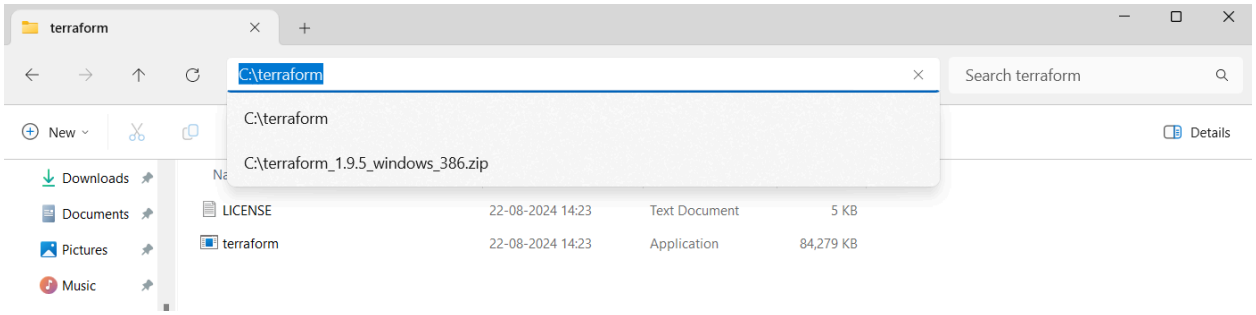




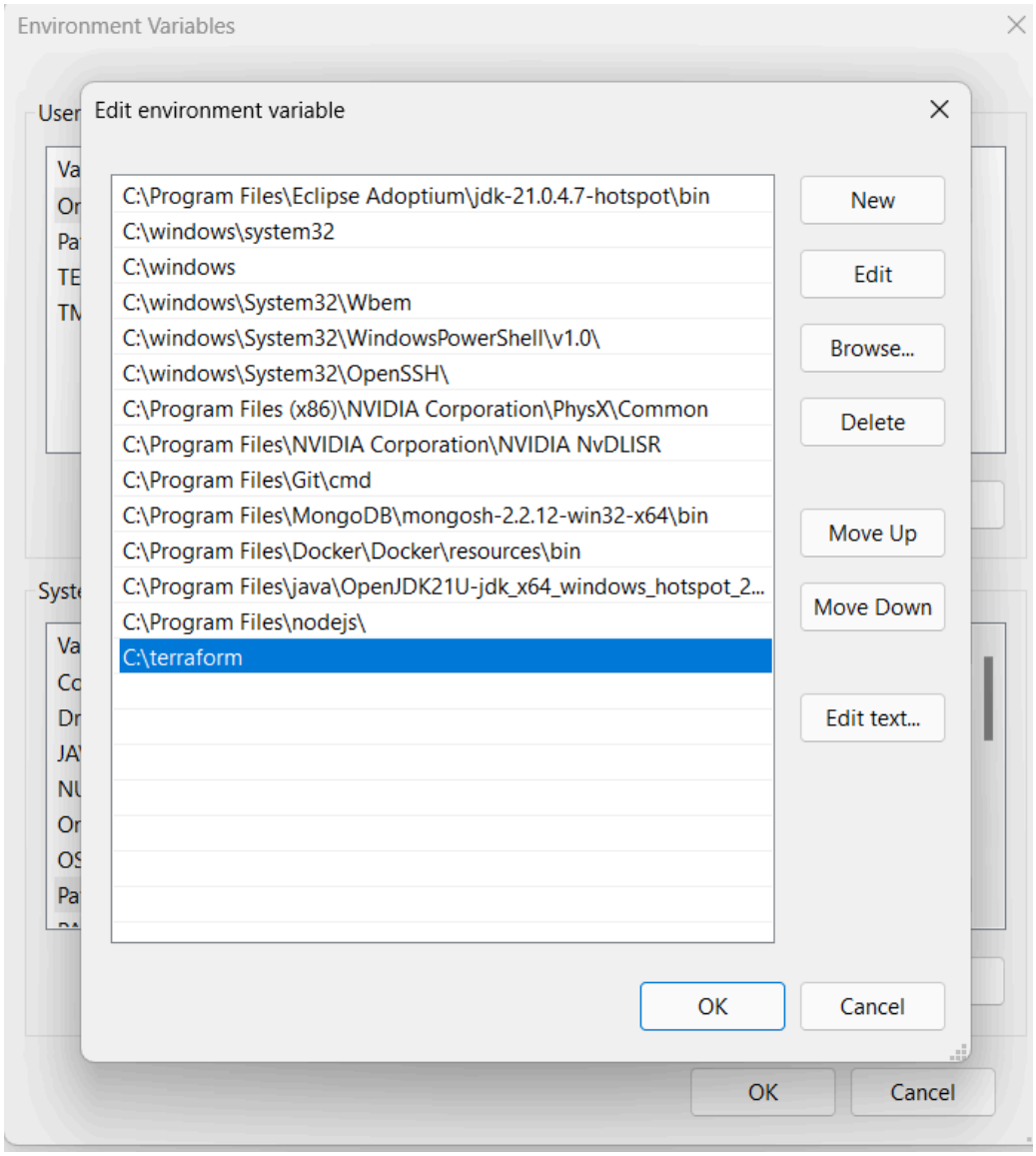
Step 2: Copy and extract Terraform from the downloads and paste it in the C drive



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



Step 4: Set the environment variables for terraform



Step 5: Check weather the terraform is installed

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

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

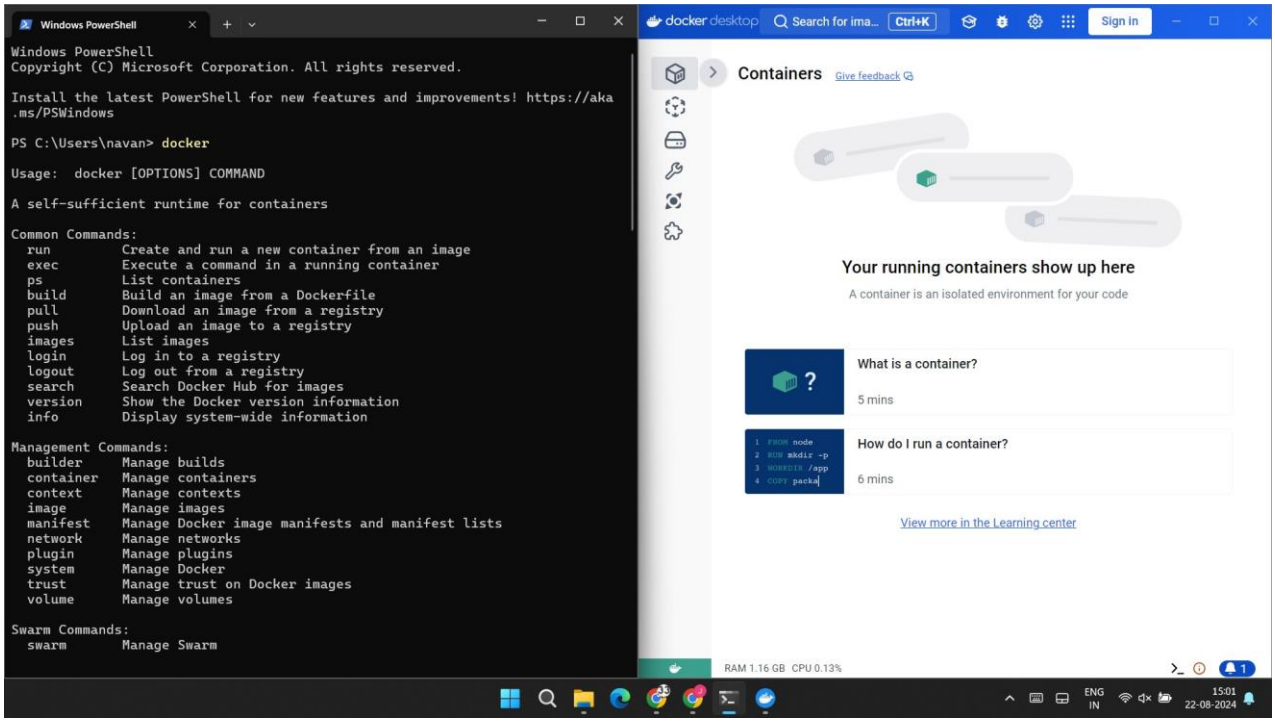
Advance devops Exp:6

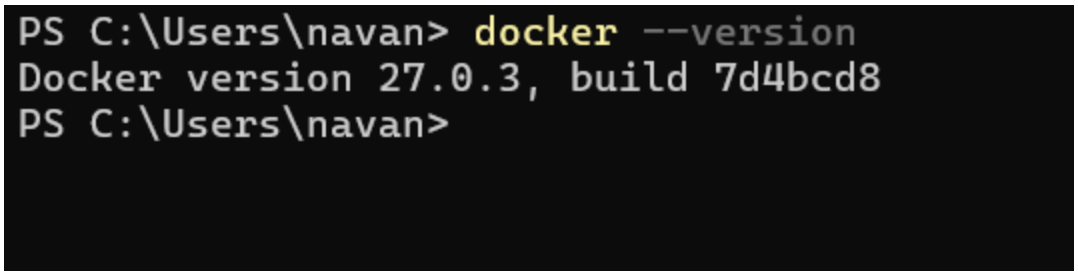
Jai Navani

D15 A-31

Aim: Creating docker image using Terraform

Step 1: Install docker Desktop after installation check the functionality



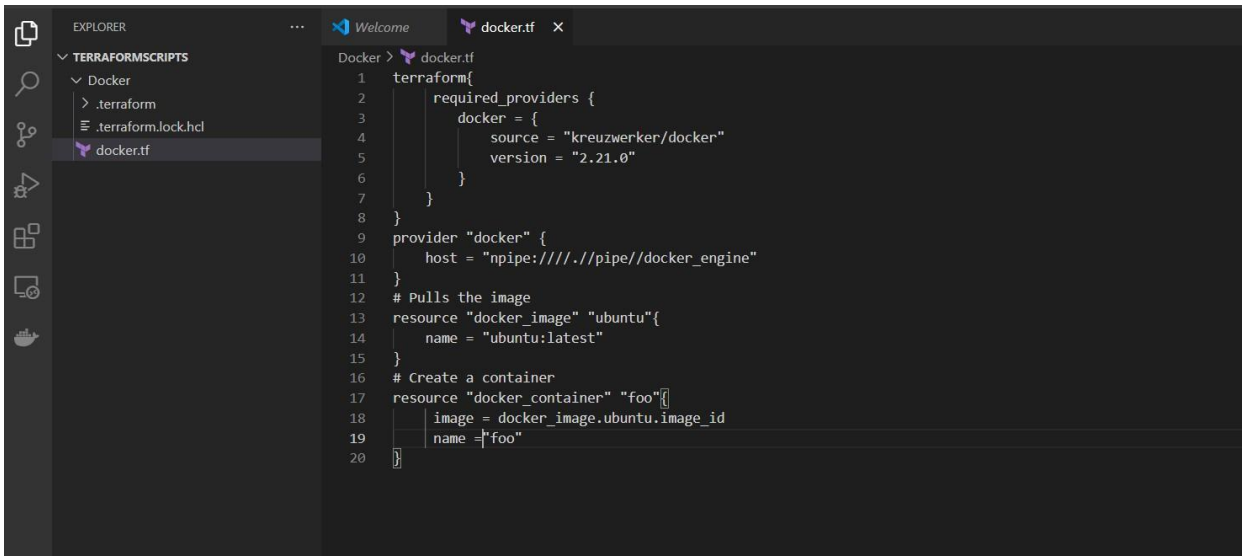


Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

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

Script:

```
terraform{
    required_providers {
      docker = {
          source = "kreuzwerker/docker"
          version = "2.21.0"
      }
   }
}
provider "docker" {
   host = "npipe:////.//pipe//docker_engine"
}
# Pulls the image
resource "docker_image" "ubuntu"{
   name = "ubuntu:latest"
}
# Create a container
resource "docker_container" "foo"{
    image = docker_image.ubuntu.image_id
    name ="foo"
}
```



Step 3: Execute terraform init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

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

Step 4: Execute Terraform plan to see the available resources

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

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

Terraform will perform the following actions:

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

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

      + healthcheck (known after apply)

      + labels (known after apply)
    }

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

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

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

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

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

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

```
    }

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

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

  Enter a value: yes

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

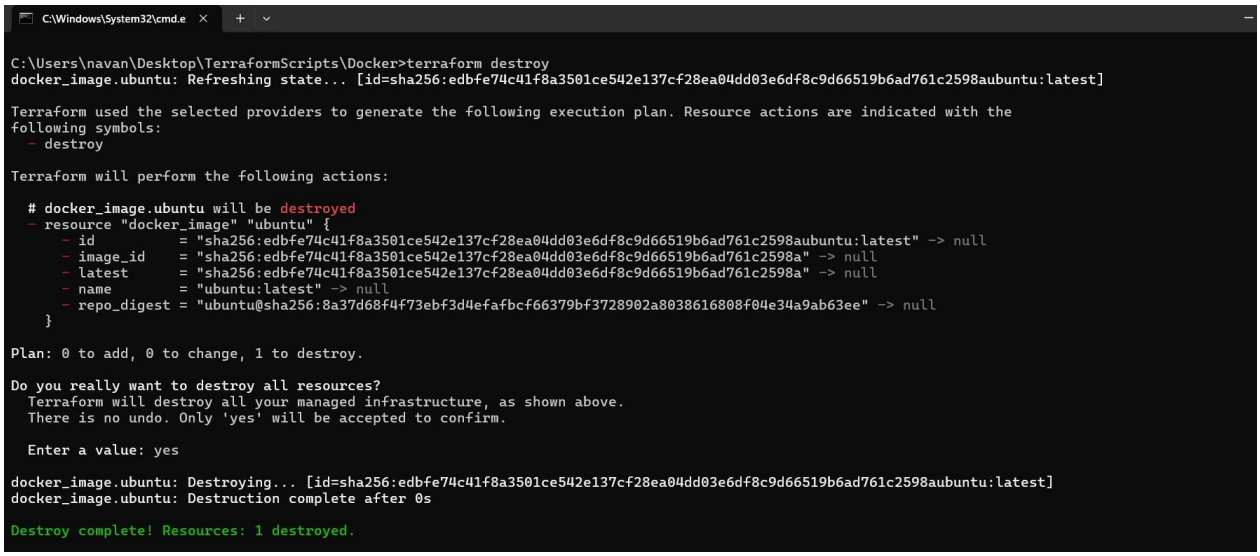
Docker images,before Executing Apply step:

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

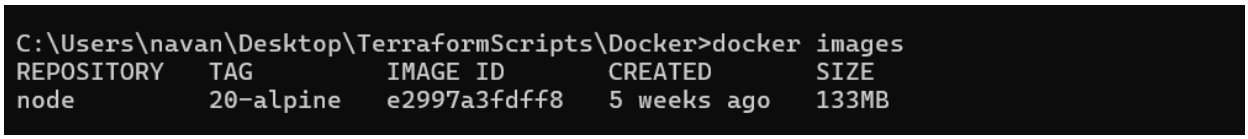
```
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28e
3e6df8c9d66519b6ad761c2598aubuntu:latest]

Note: Objects have changed outside of Terraform

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

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


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

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

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

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

Terraform will perform the following actions:

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

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

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

  Enter a value:
```



Docker images after executing destroy step

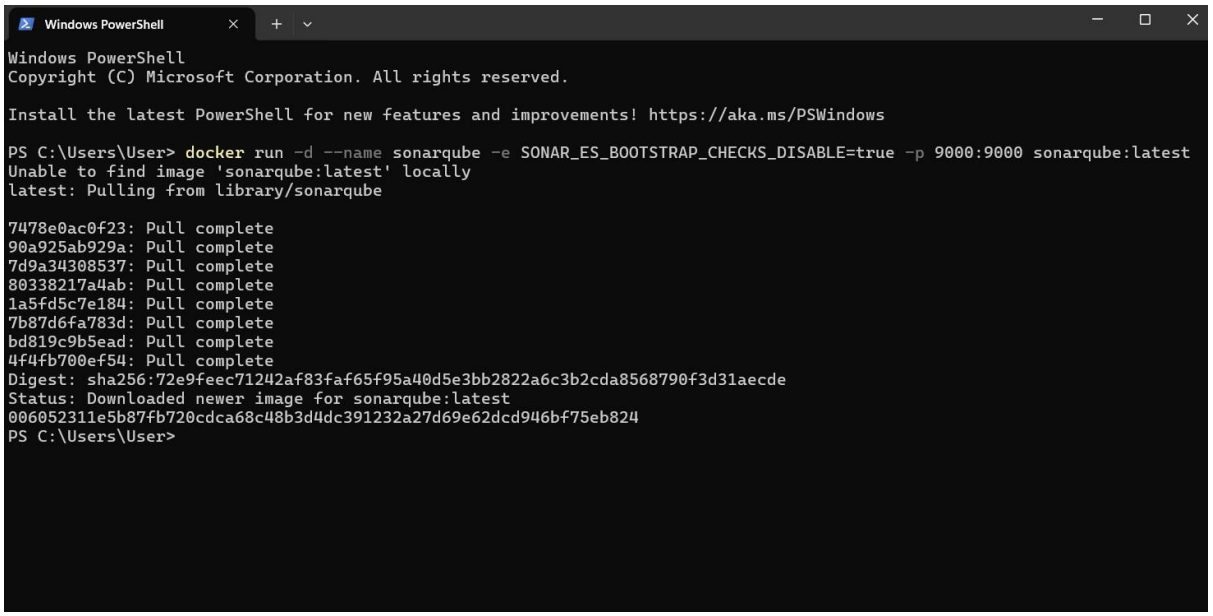


Jai Navani
D15A
31

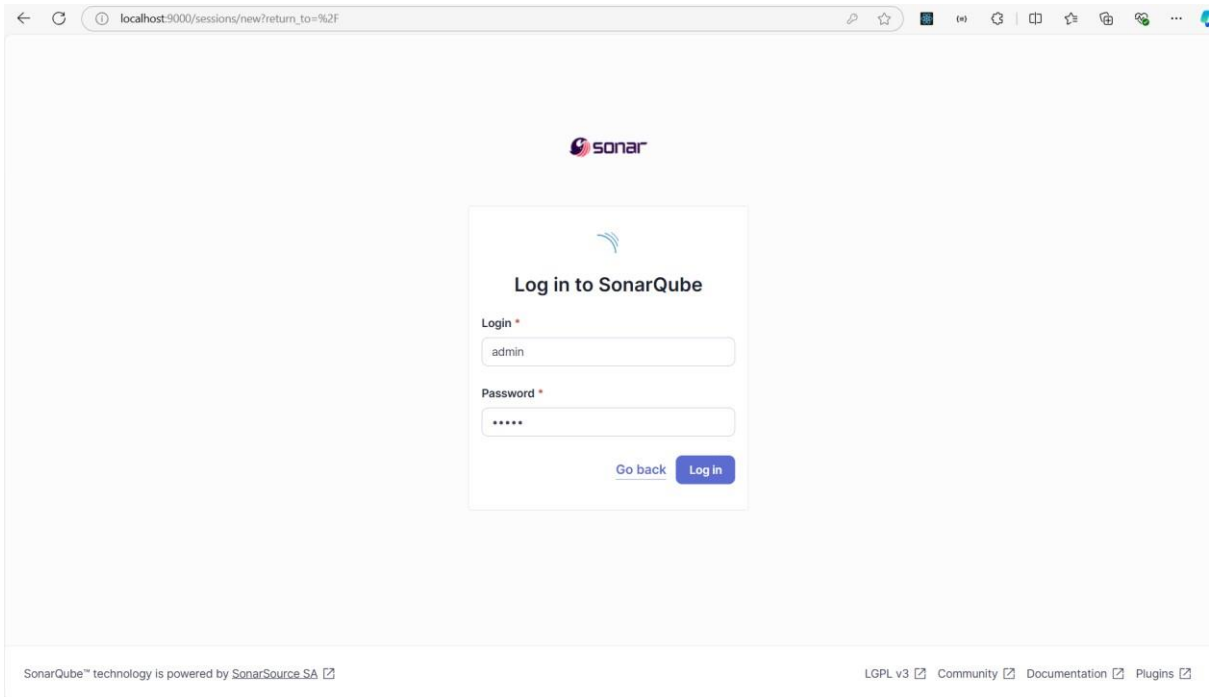
EXPERIMENT 7

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

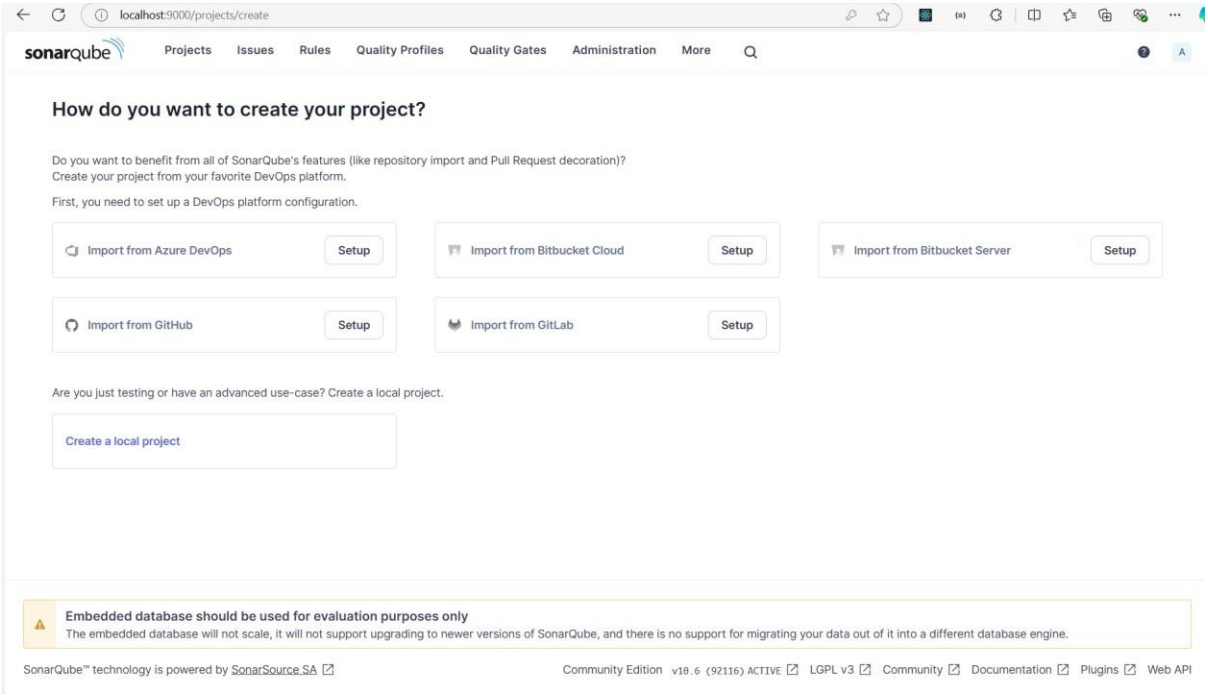
Step 1: Open Windows PowerShell and run the following command –
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
WARNING: Run the following command only once

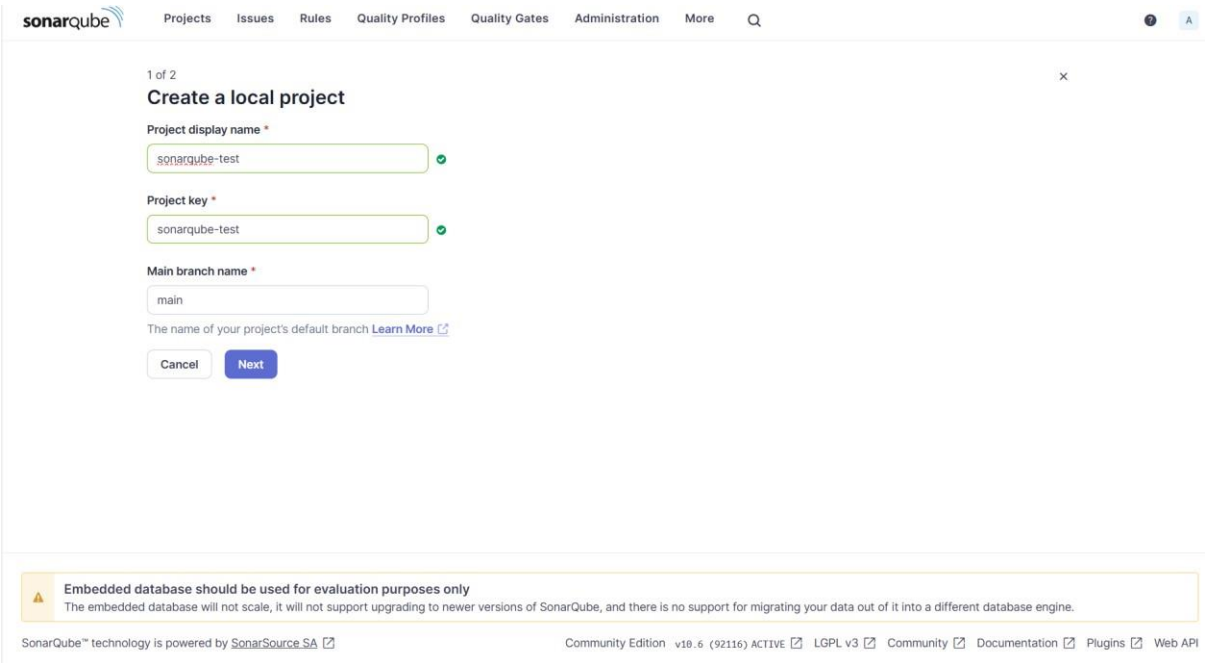


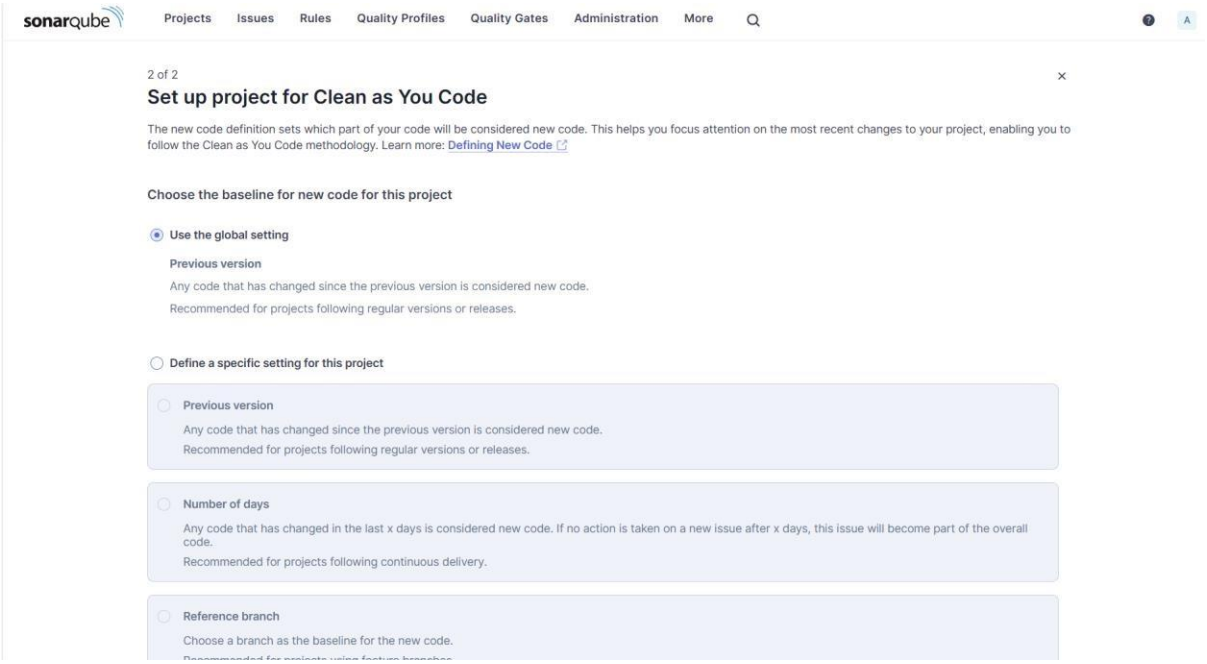
Step 2: Visit http://localhost:9000/ to open SonarQube. Login with username: admin and password: admin.



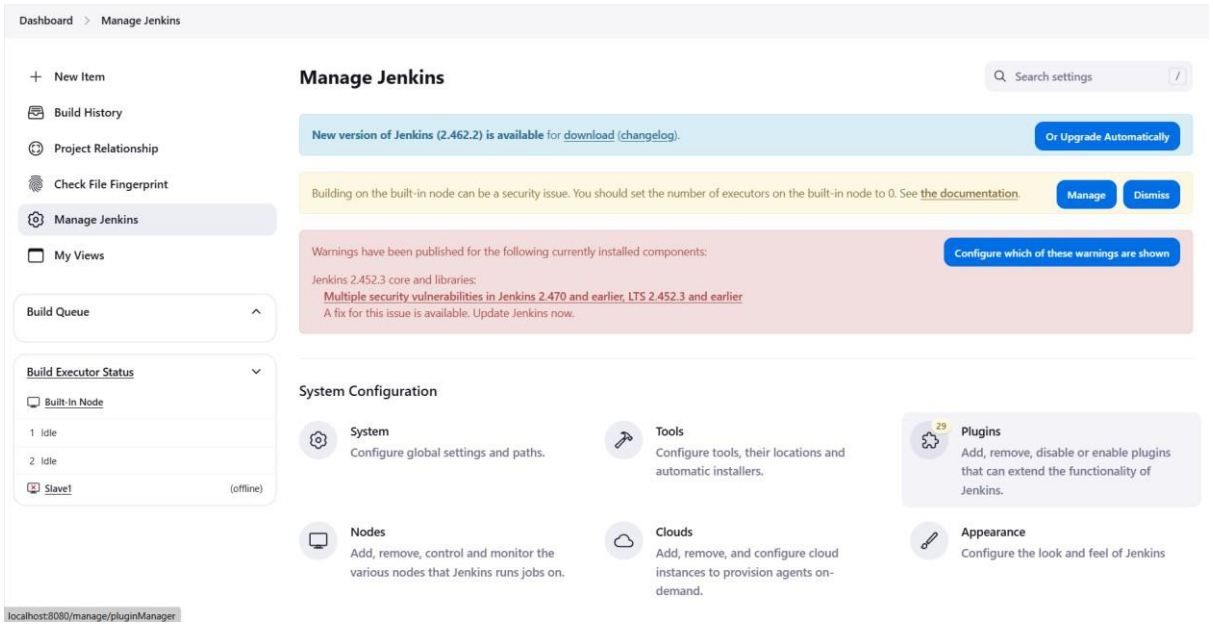
Step 3: Click on create a local project and name the project as sonarqube-test and key as sonarqube-test and click on the next button. In the next step select the “Use the global setting” option and click on create project.

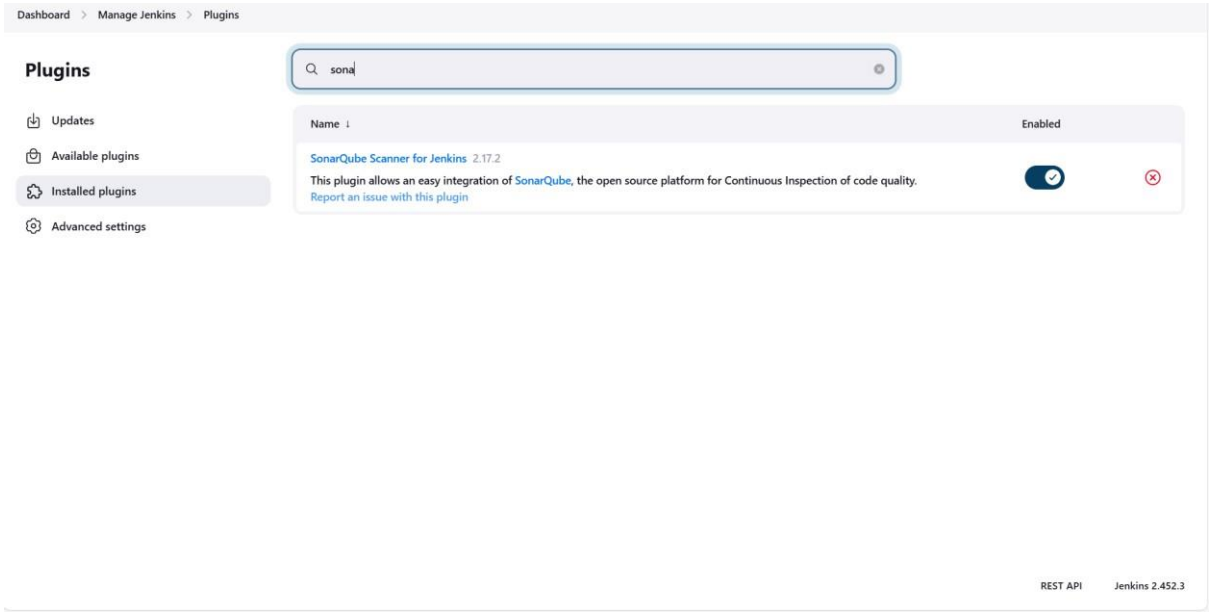

localhost:9000/projects/create
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
How do you want to create your project?
Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite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


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

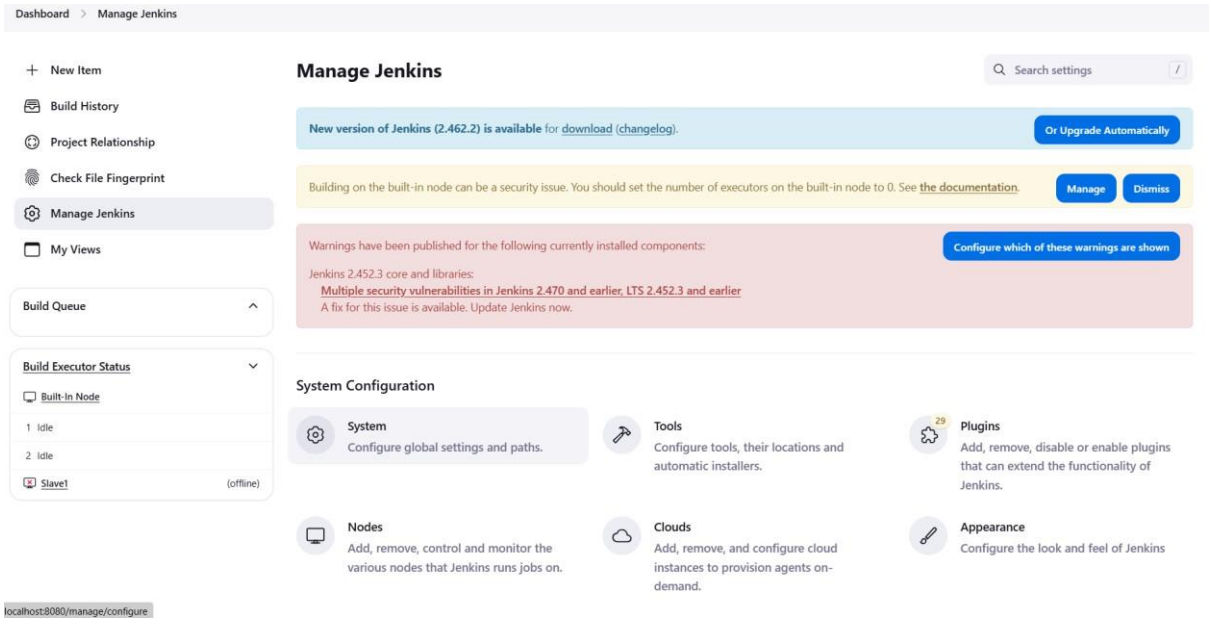


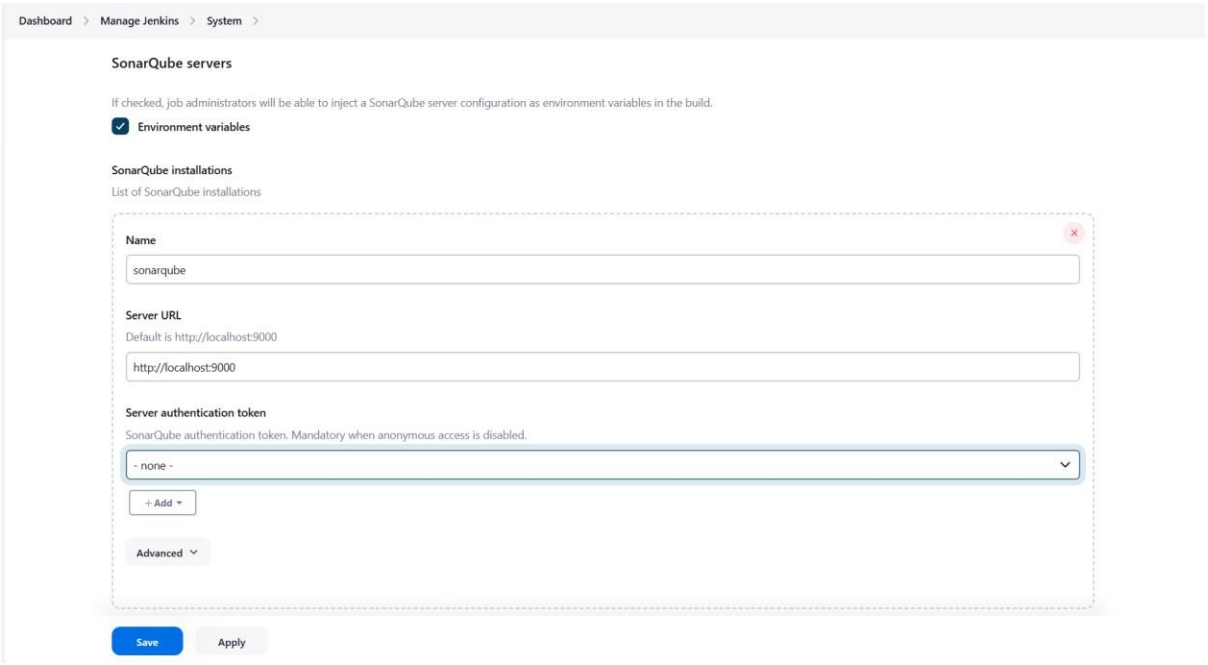
Step 4: Open Jenkins using http://localhost:8080/ and select Manage Jenkins, then select the Plugins and select available plugins from sidebar and search for SonarQube Scanner and install it. Once installed you can view the installed plugin in installed plugins section in sidebar.



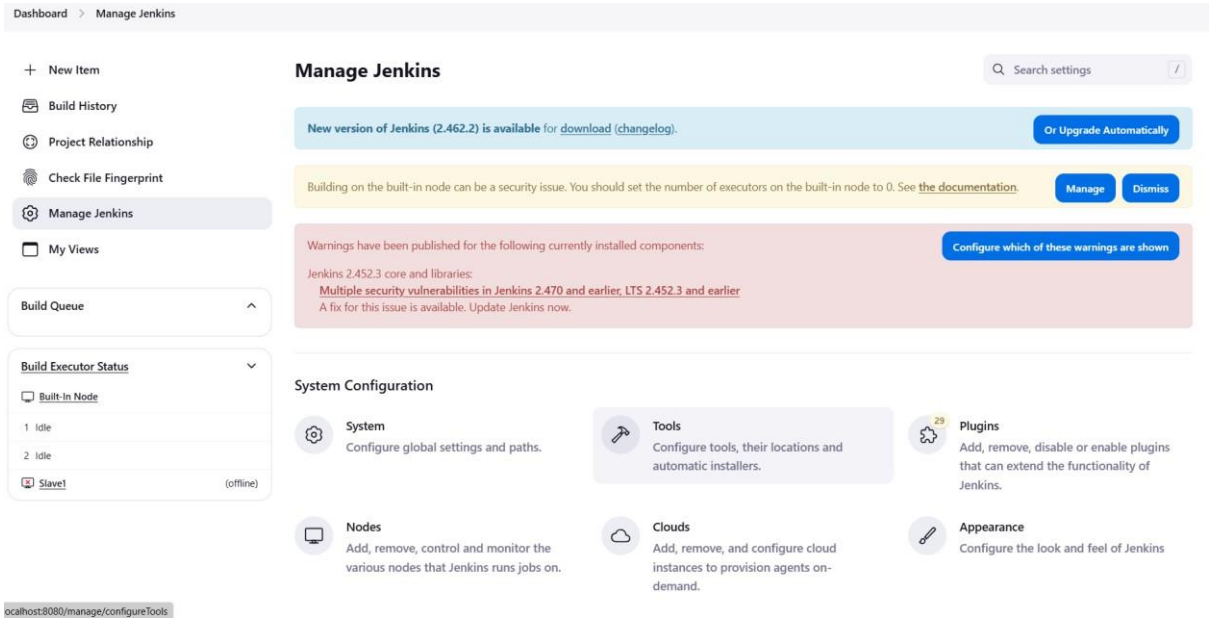


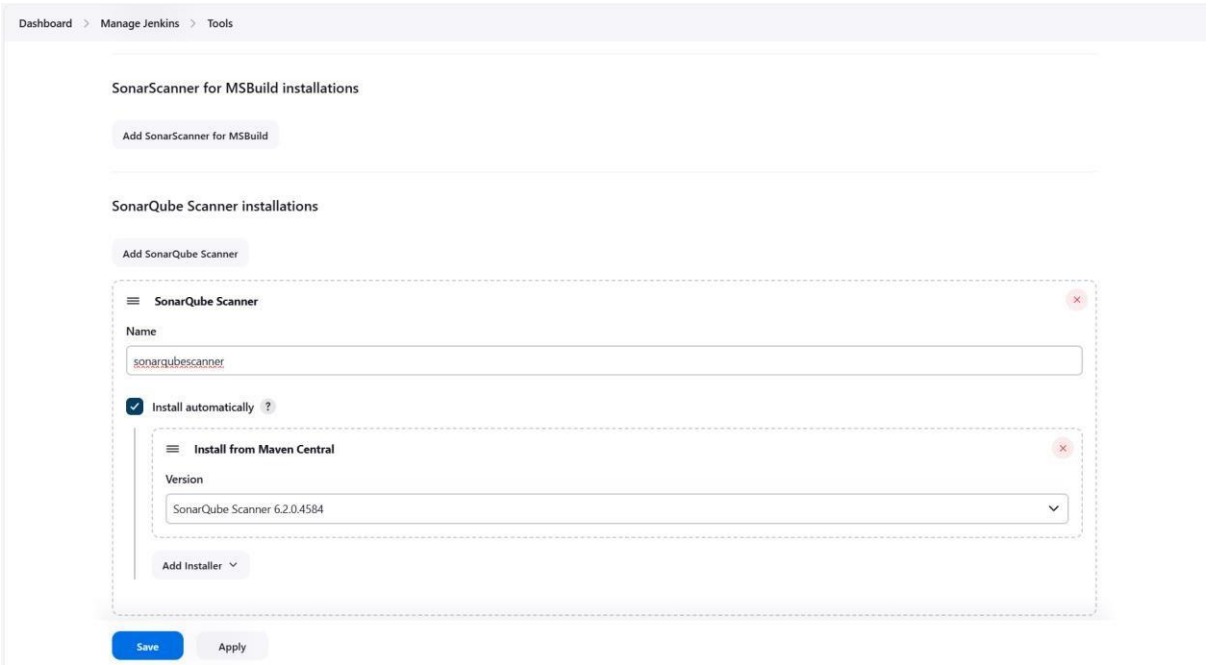
Step 5: Select Manage Jenkins, then select the System and then scroll down to SonarQube Server. Name the server as sonarqube and set the server url as http://localhost:9000/ then click on save.



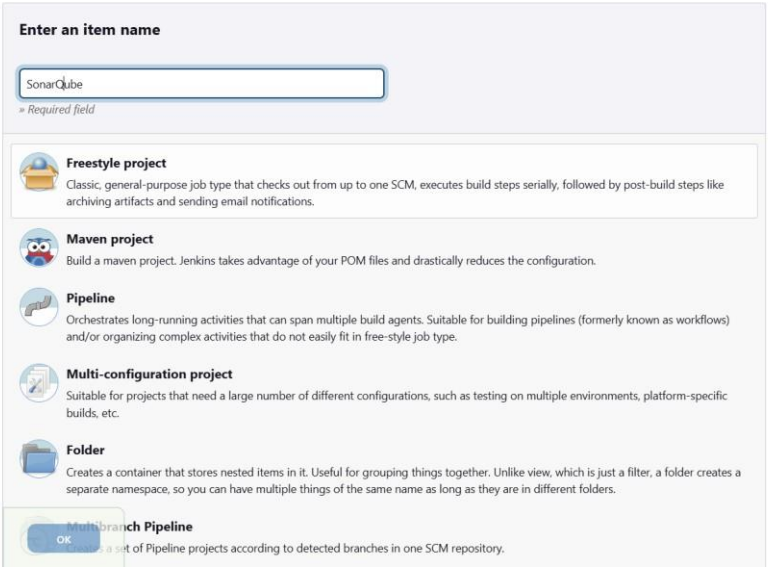


Step 6: Go to Jenkins Dashboard and select Manage Jenkins, then select the Tools and then scroll down to SonarQube Scanner installations. Name the sonarqube scanner as sonarqubescanner and select install automatically then click on save.

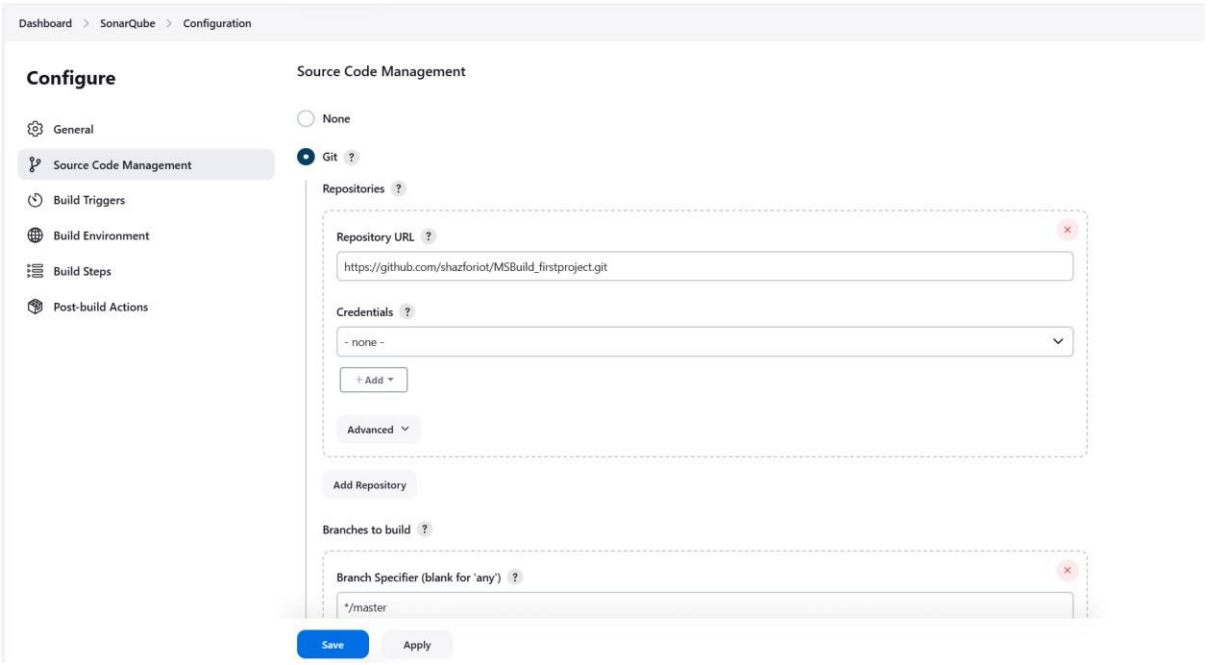




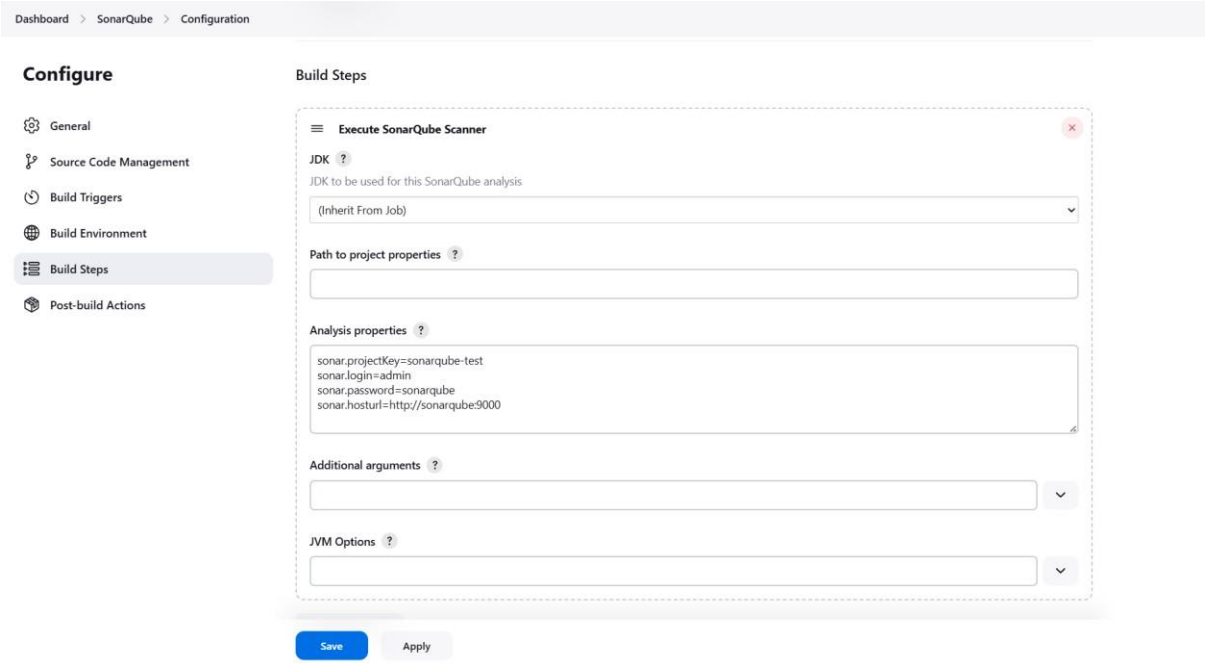
Step 7: Go to Jenkins dashboard and click on New Item and select Freestyle project and name it as SonarQube and then click on ok.

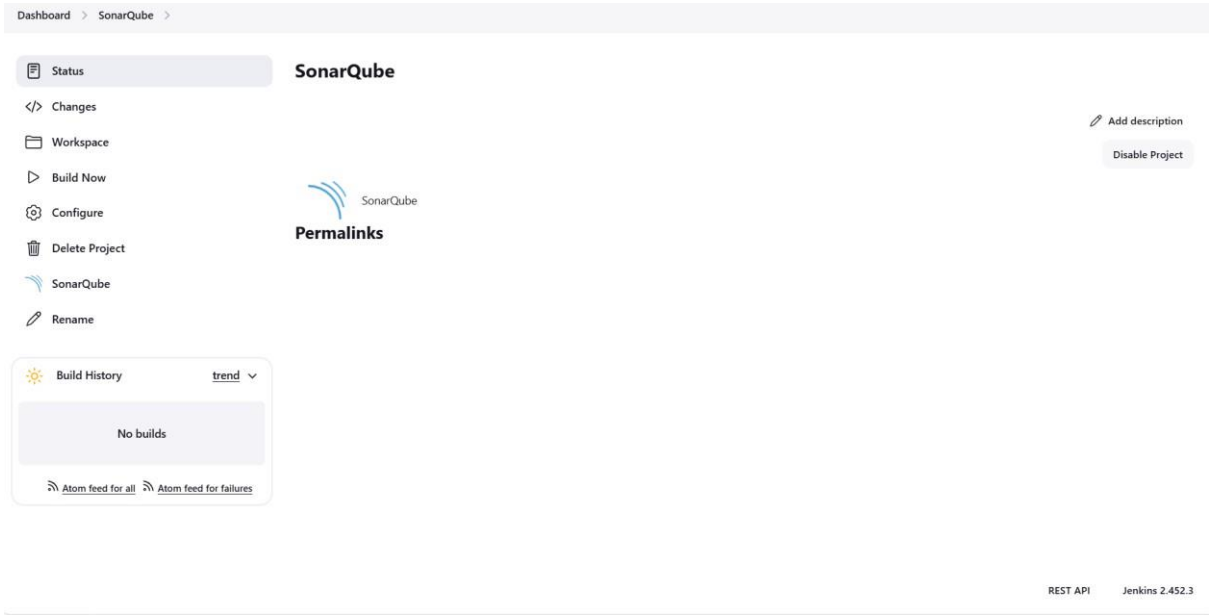


Step 8: For configuration, Select git and paste the following git repository in the repository url.
https://github.com/shazforiot/MSBuild_firstproject
This is a simple Hello world project

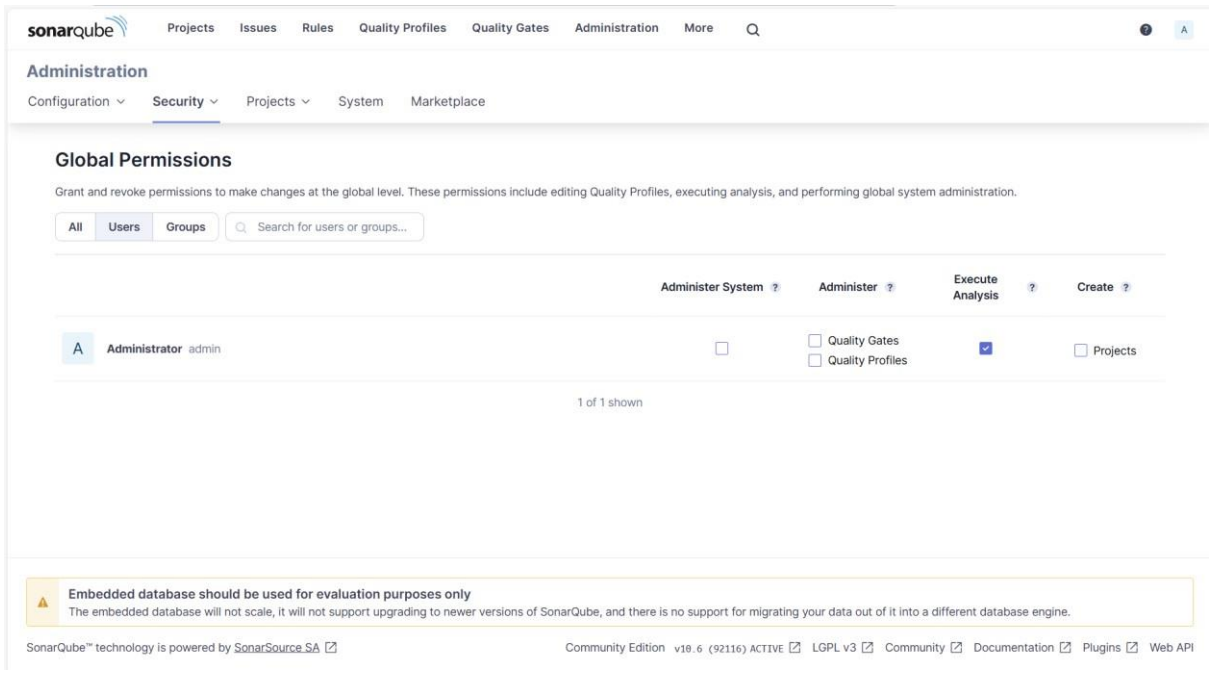


Step 9: Under the Build steps select "Execute SonarQube Scanner" option and under Analysis Properties write the following -
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=sonarqube
sonar.hosturl=http://sonarqube:9000
Then click on the save button.

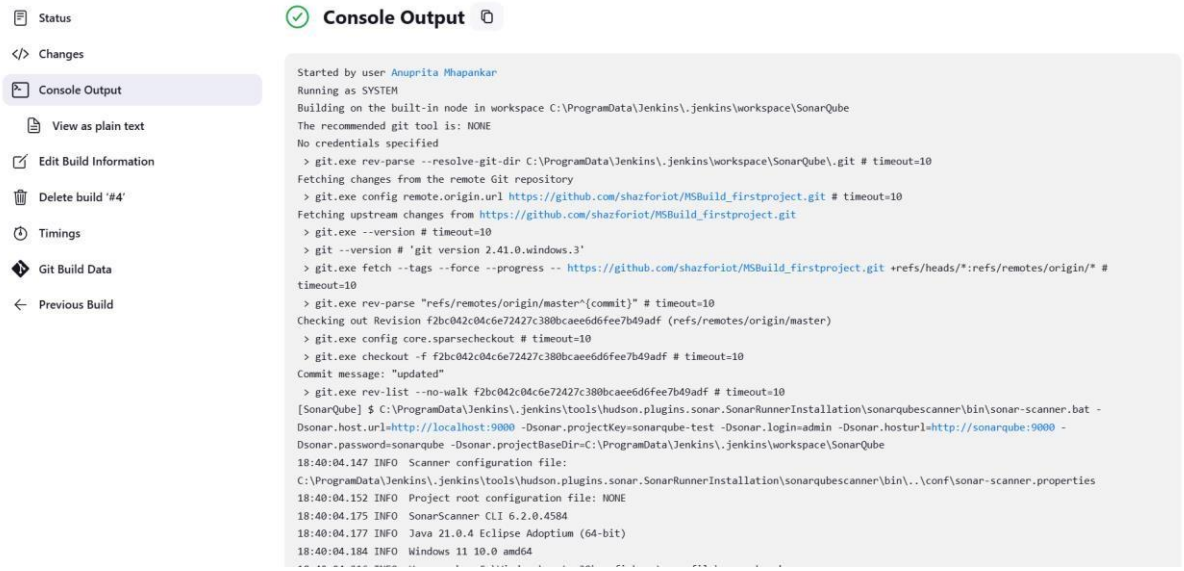


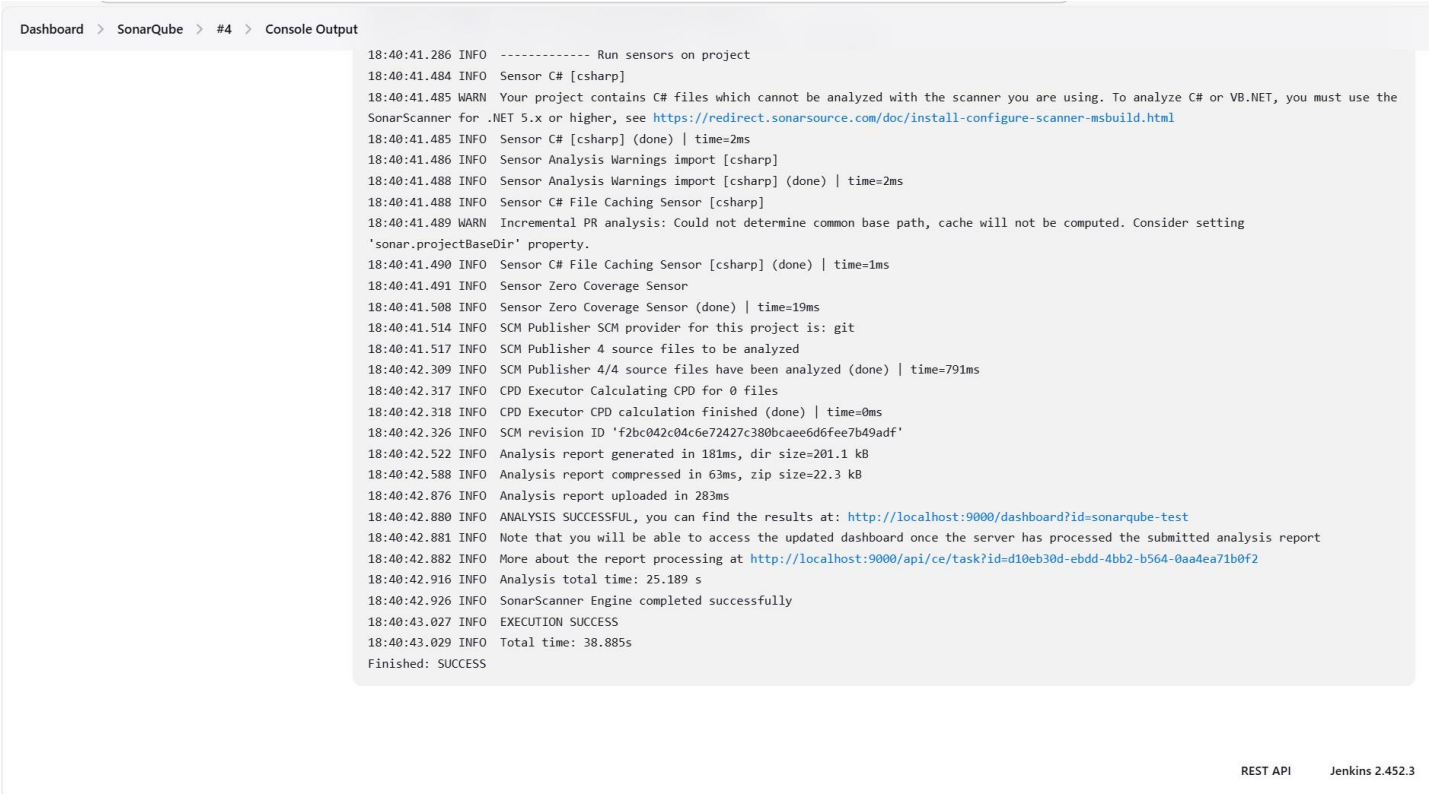


Step 10: Visit http://localhost:9000/admin/permissions and select the Users tab and for Administrator select the checkbox Execute Analysis.

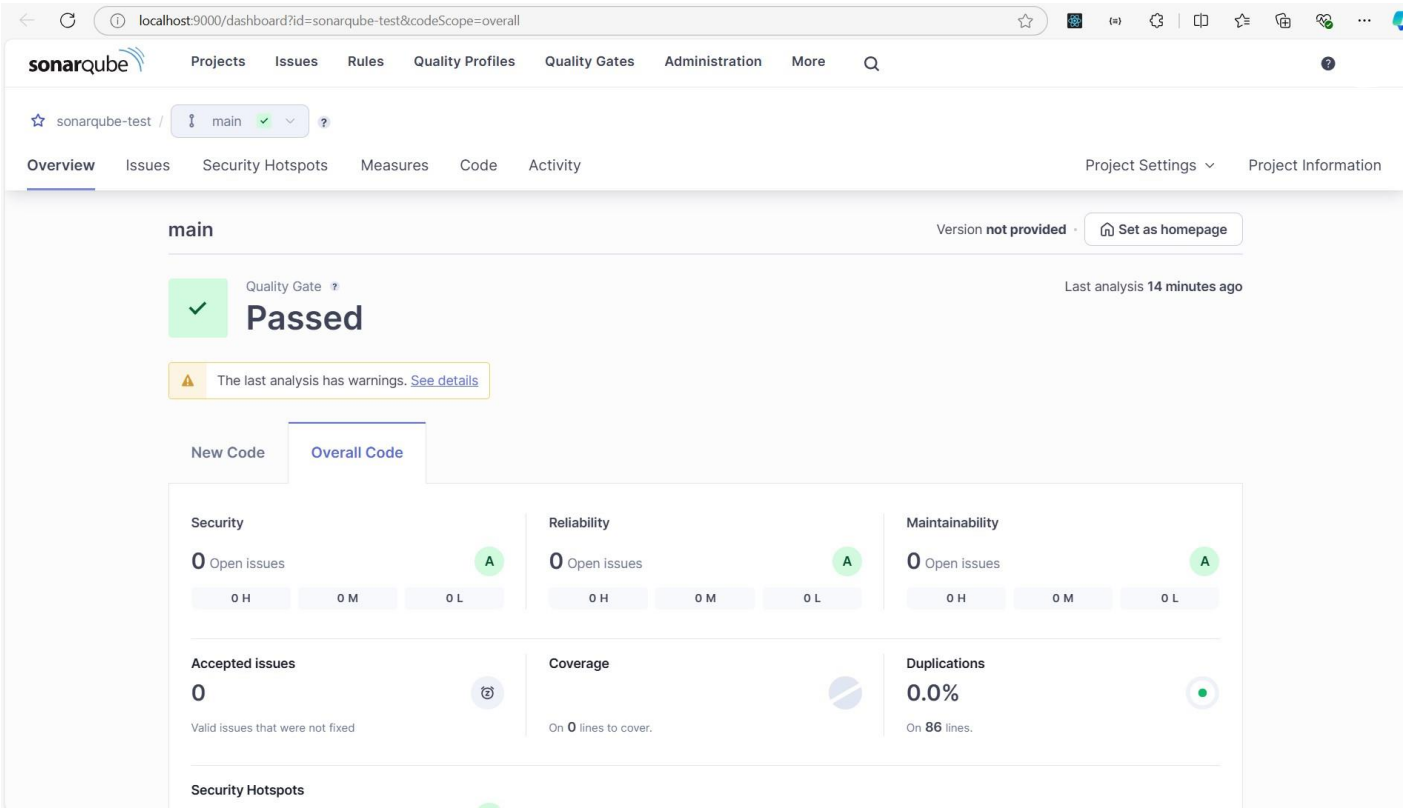


Step 11: Now, come back to Jenkins and click on Build Now. The build is success.





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


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

Advance-Devops
Experiment no:8

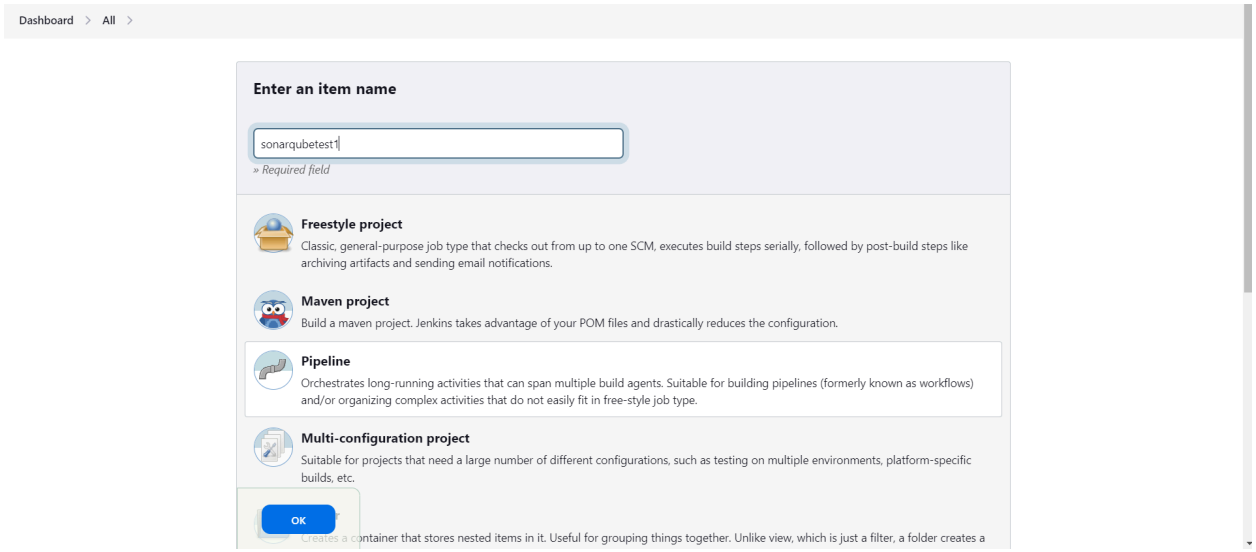
D15-A
Jai navani-31

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

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

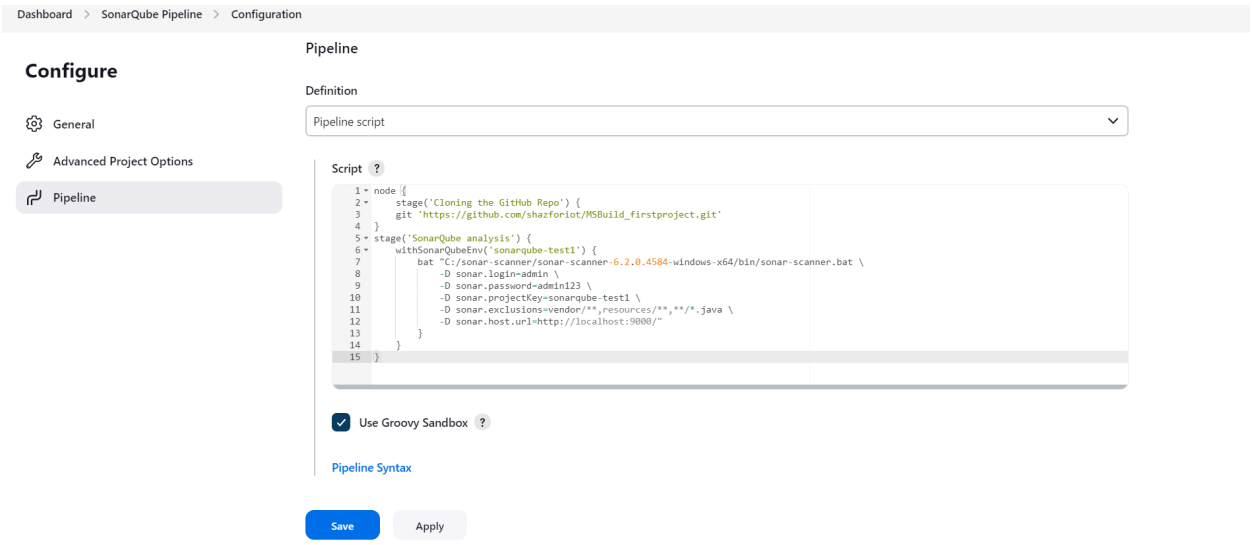
- Login to SonarQube using username admin and password admin.
- Create a manual project in SonarQube with the name sonarqube-test1

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



Under Pipeline Script, enter the following -

```
node {
        stage('Cloning the GitHub Repo') {
                git 'https://github.com/shazforiot/GOL.git'
        }
stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
        sh "<PATH_TO_SONARQUBE_FOLDER>//bin//sonar-scanner \
                -D sonar.login=<SonarQube_USERNAME> \
                -D sonar.password=<SonarQube_PASSWORD> \
                -D sonar.projectKey=<Project_KEY> \
                -D sonar.exclusions=vendor/**,resources/**,**/*.java \
                -D sonar.host.url=http://127.0.0.1:9000/"
                }
        }
}
```



Save the changes and go to build now:

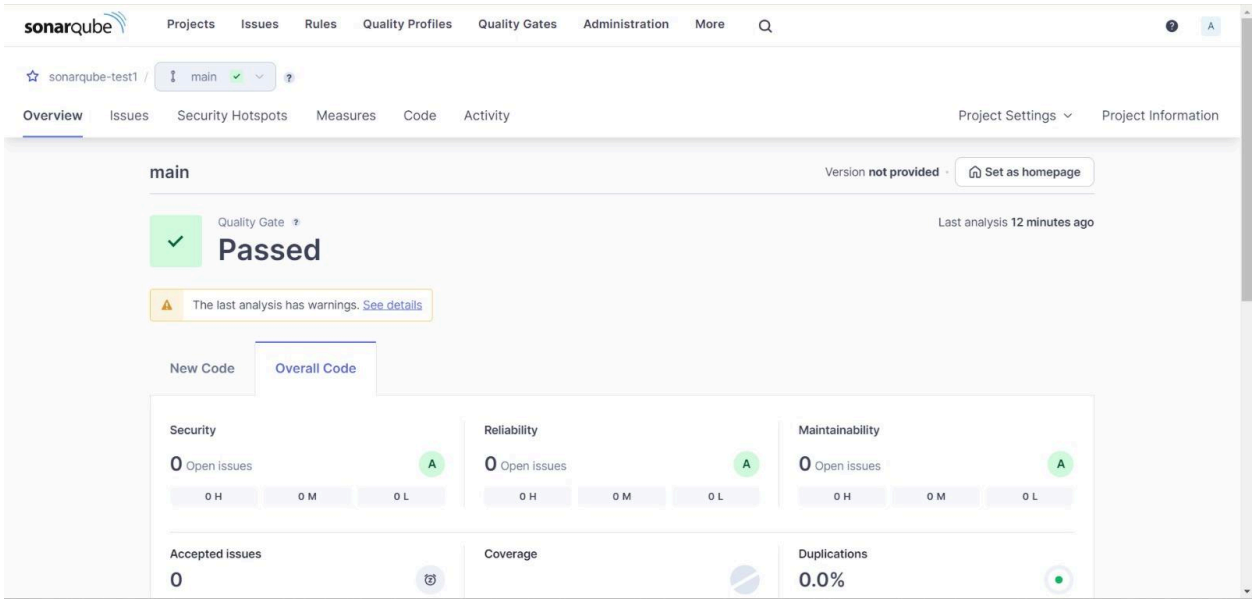
Console output:

Console Output

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

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

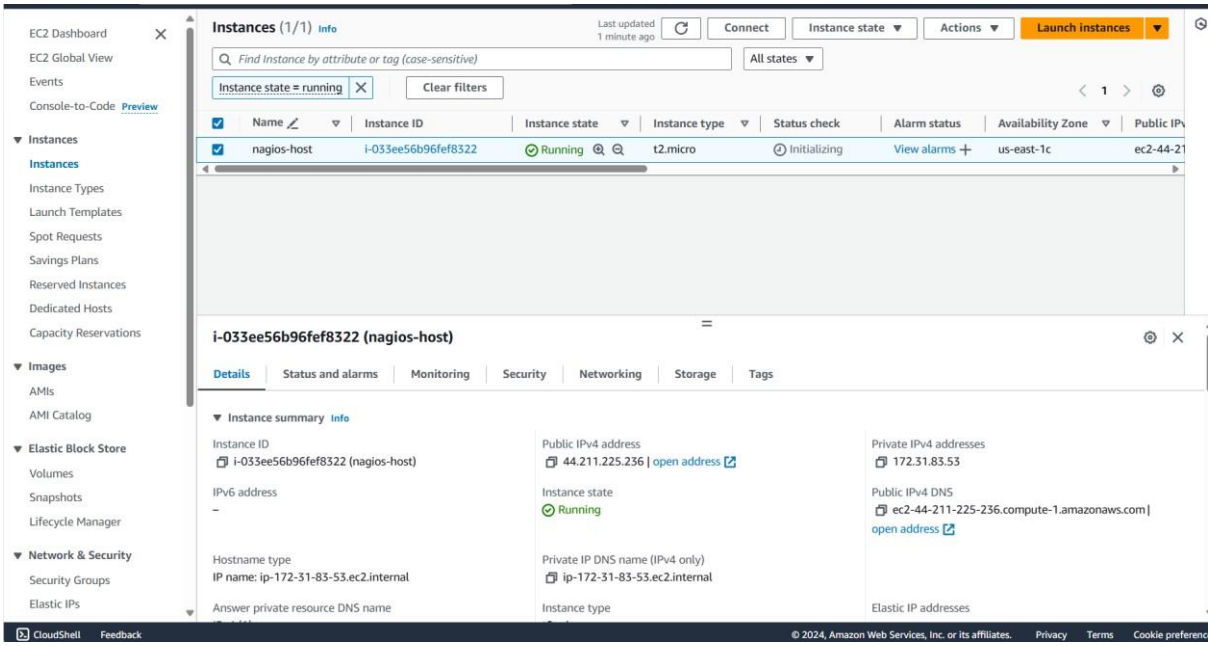
Output:



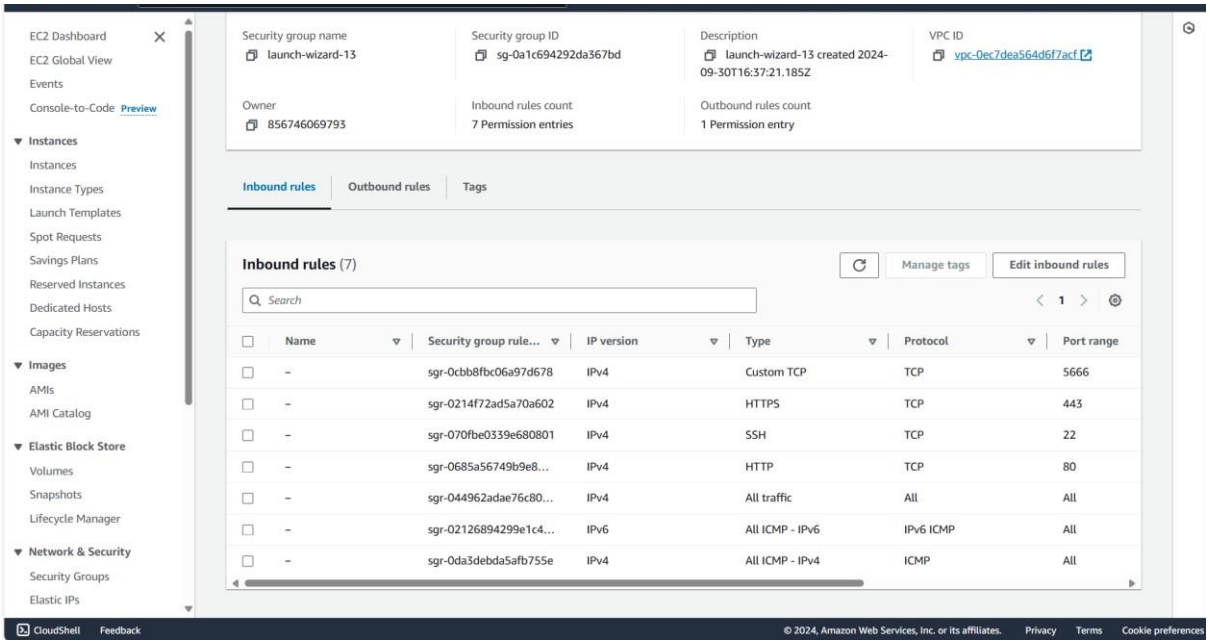
Experiment 9

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

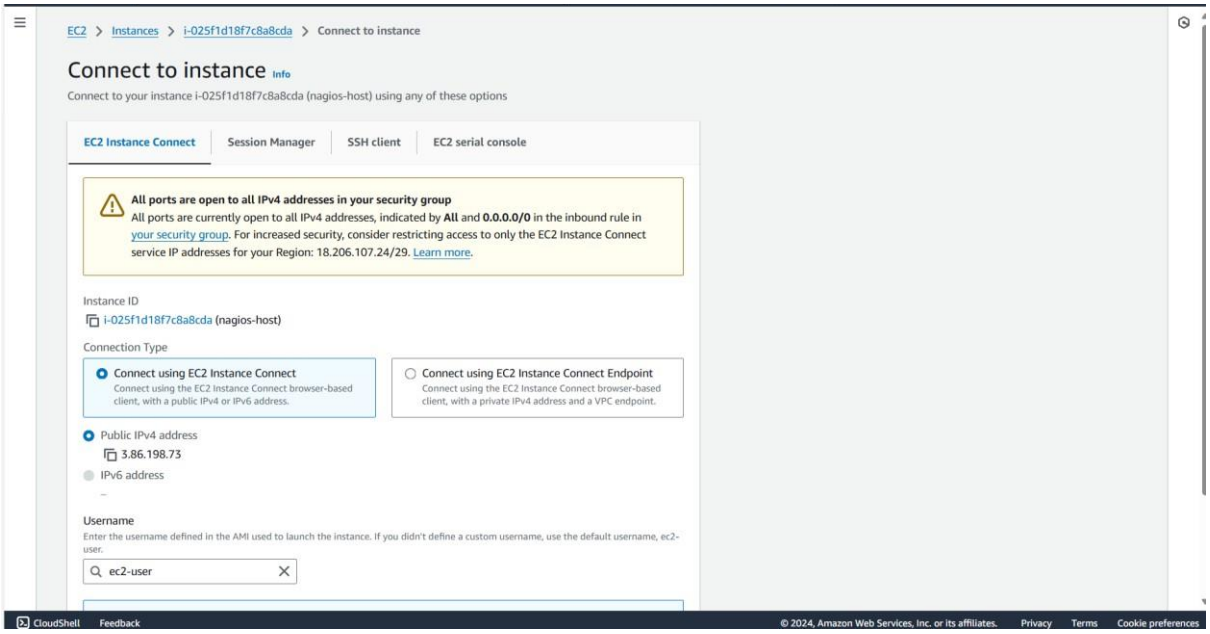
Step 1: Create an EC2 Instance and name it as nagios-host



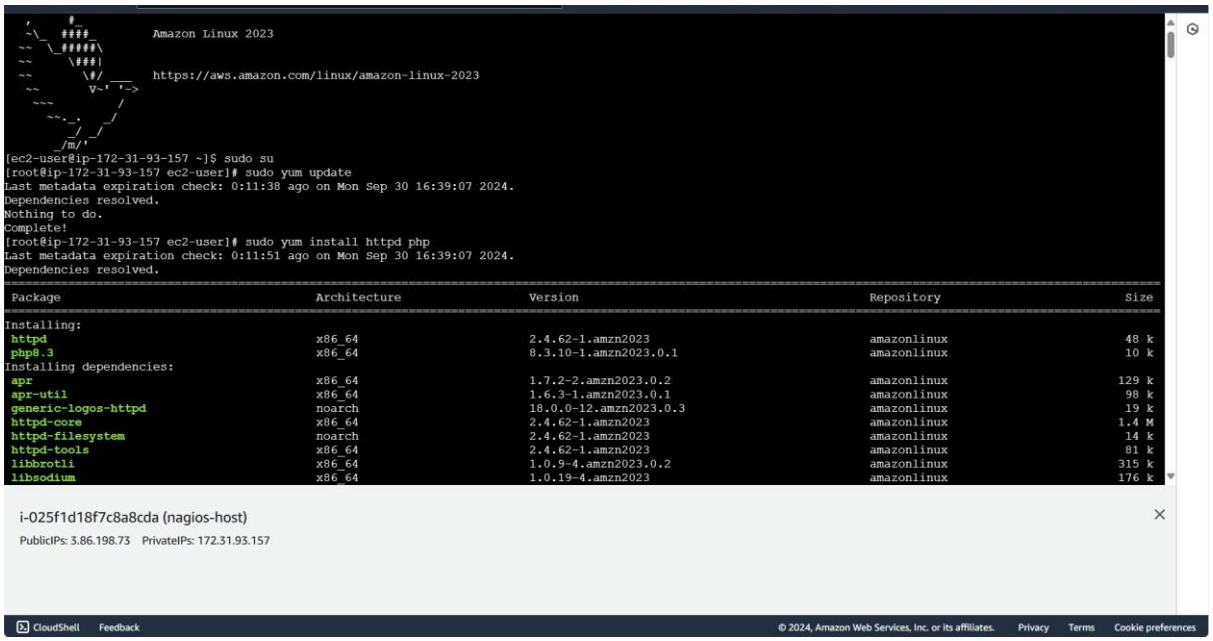
Step 2: Under the security groups, click on edit inbound rules and set as shown in the figure below

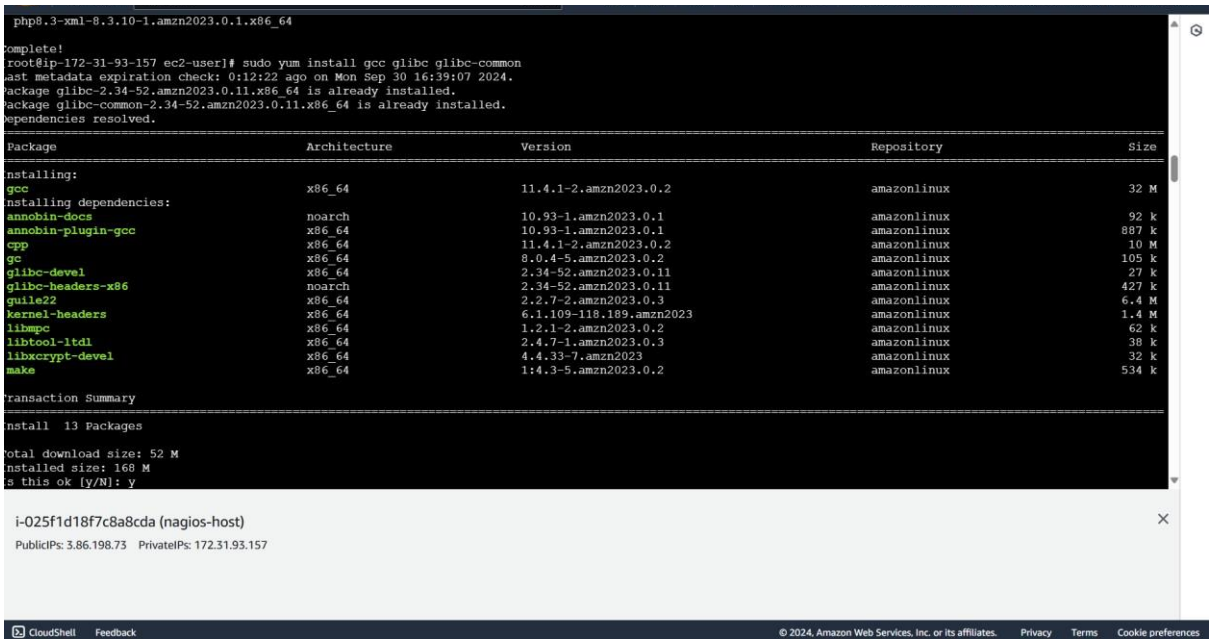


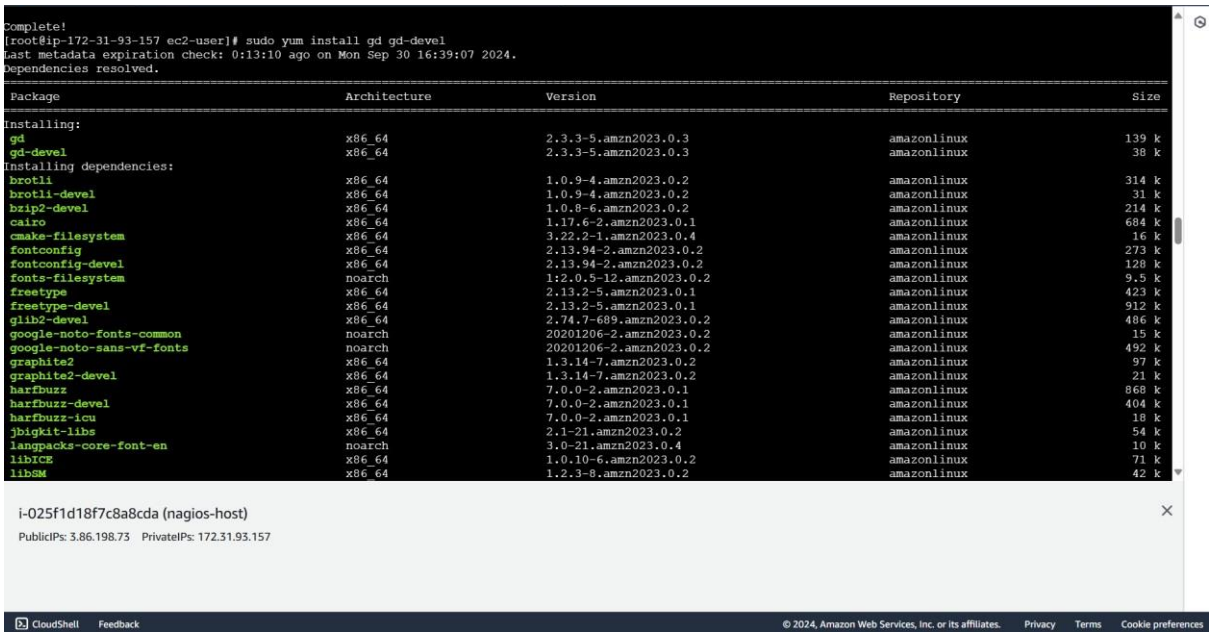
Step 3: Then select the instance nagios-host and then connect the instance.



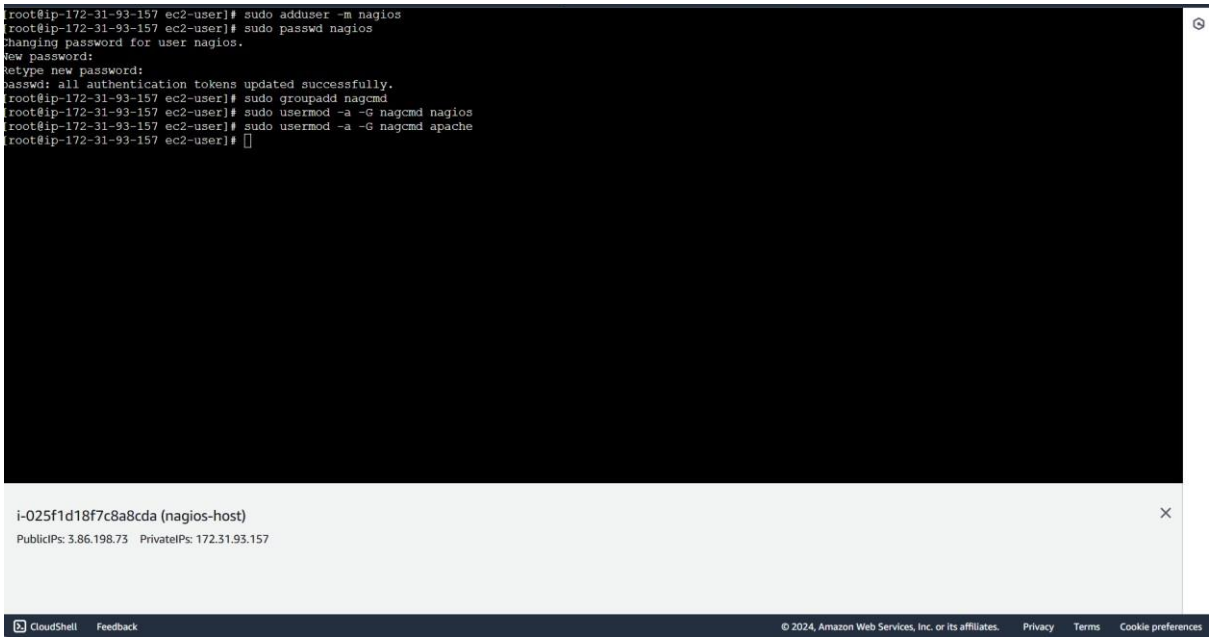
Step 3: Now, run the following commands -
sudo su
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel



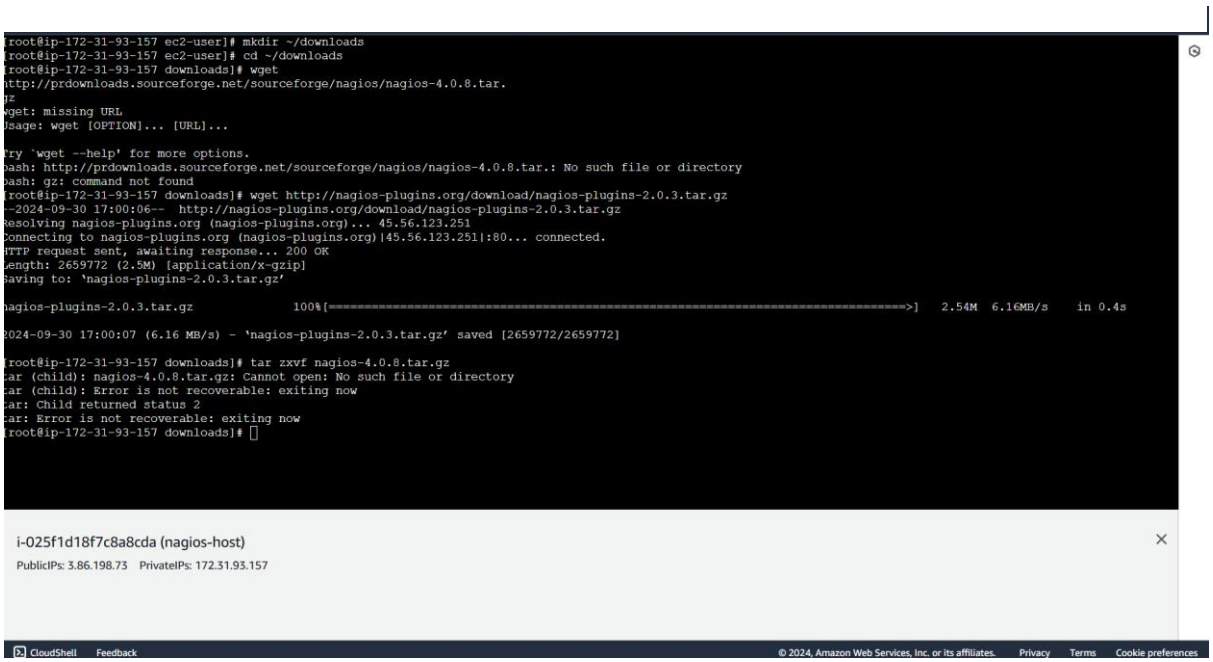




Step 4: Create a new nagios user with its password.
sudo adduser -m nagios
sudo passwd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache



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



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

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

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

2024-09-30 17:03:05 (2.21 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

--2024-09-30 17:03:05--  http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz.1'

nagios-plugins-2.0.3.tar.gz.1           100%[=============================================================================>]   2.54M  7.26MB/s    in 0.3s

2024-09-30 17:03:05 (7.26 MB/s) - 'nagios-plugins-2.0.3.tar.gz.1' saved [2659772/2659772]
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157


```
nagios-plugins-2.0.3/plugins-scripts/check_ifoperstatus.pl
nagios-plugins-2.0.3/plugins-scripts/Makefile.am
nagios-plugins-2.0.3/plugins-scripts/subst.in
nagios-plugins-2.0.3/plugins-scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins-scripts/check_log.sh
nagios-plugins-2.0.3/plugins-scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins-scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins-scripts/utils.pm.in
nagios-plugins-2.0.3/plugins-scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins-scripts/t/
nagios-plugins-2.0.3/plugins-scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins-scripts/t/check_file_age.t
nagios-plugins-2.0.3/plugins-scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins-scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/utils.t
nagios-plugins-2.0.3/plugins-scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins-scripts/check_wave.pl
nagios-plugins-2.0.3/plugins-scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins-scripts/utils.sh.in
nagios-plugins-2.0.3/plugins-scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requires
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requires
[root@ip-172-31-93-157 downloads]#
```

i-025f1d18f7c8a8cda (nagios-host)

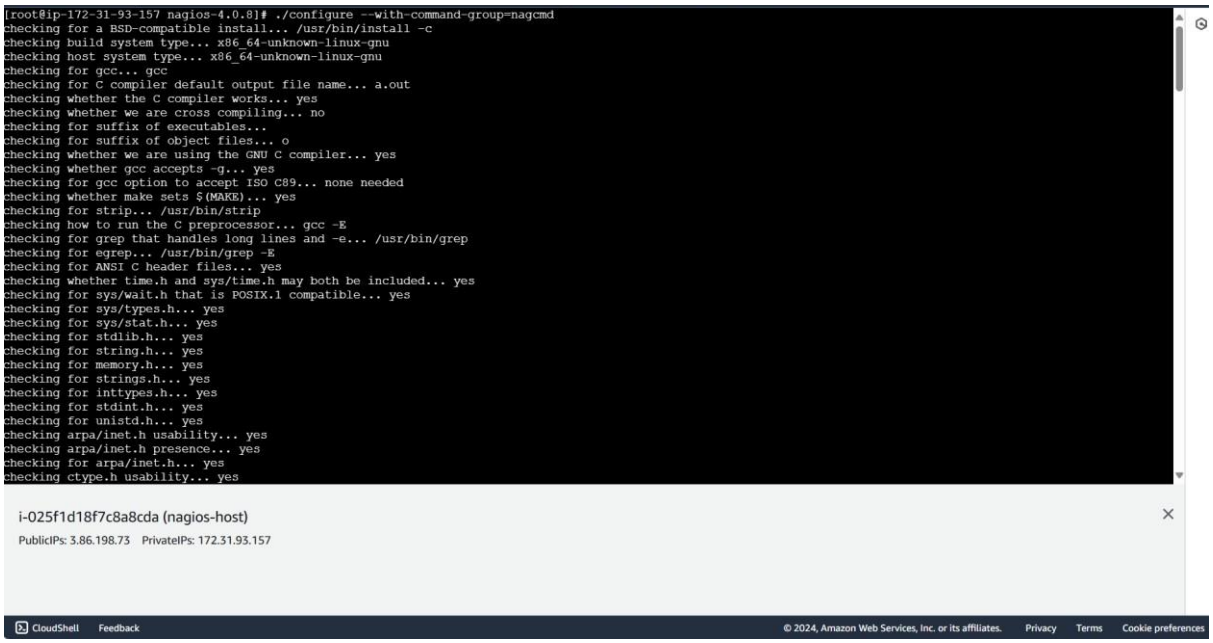
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157


```
nagios-plugins-2.0.3/plugins-scripts/Makefile.am
nagios-plugins-2.0.3/plugins-scripts/subst.in
nagios-plugins-2.0.3/plugins-scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins-scripts/check_log.sh
nagios-plugins-2.0.3/plugins-scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins-scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins-scripts/utils.pm.in
nagios-plugins-2.0.3/plugins-scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins-scripts/t/
nagios-plugins-2.0.3/plugins-scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins-scripts/t/check_file_age.t
nagios-plugins-2.0.3/plugins-scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins-scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/utils.t
nagios-plugins-2.0.3/plugins-scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins-scripts/check_wave.pl
nagios-plugins-2.0.3/plugins-scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins-scripts/utils.sh.in
nagios-plugins-2.0.3/plugins-scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requires
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requires
[root@ip-172-31-93-157 downloads]# cd nagios-4.0.8
[root@ip-172-31-93-157 nagios-4.0.8]#
```

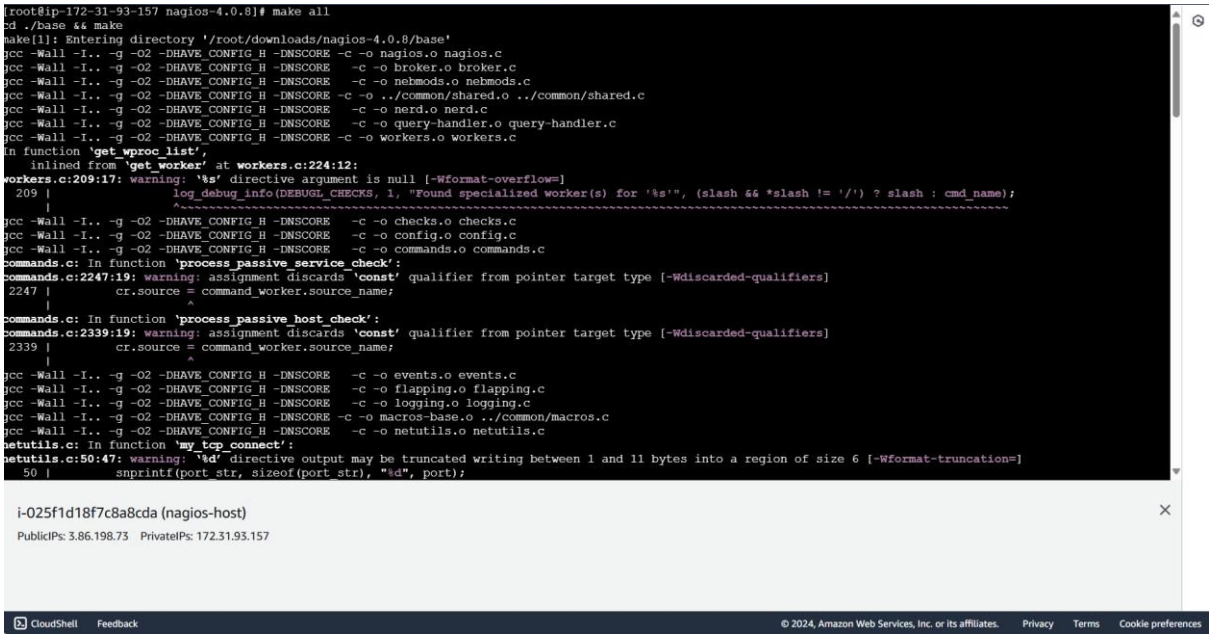
i-025f1d18f7c8a8cda (nagios-host)

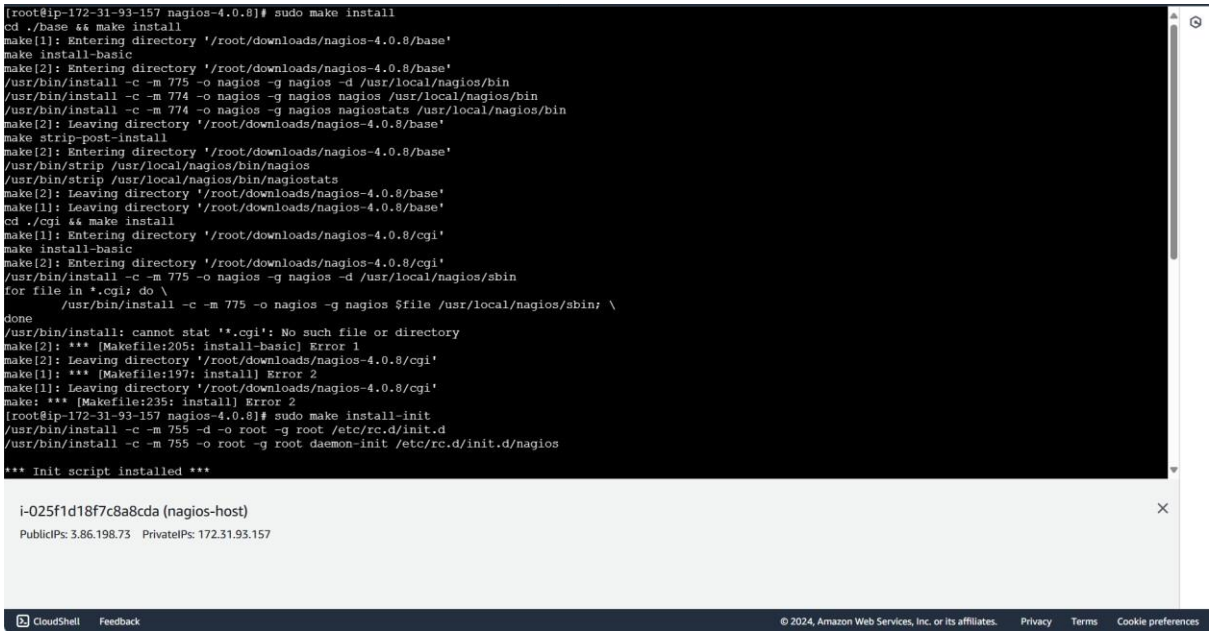
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

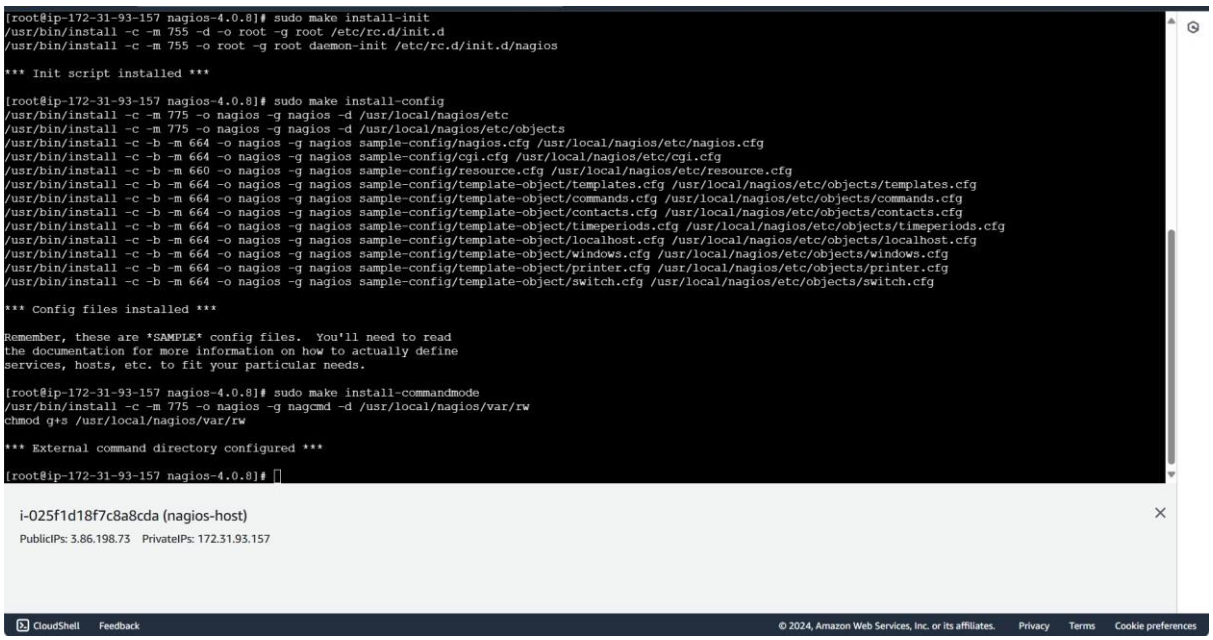
Step 6: Now to run the configuration script run the following command.
./configure --with-command-group=nagcmd



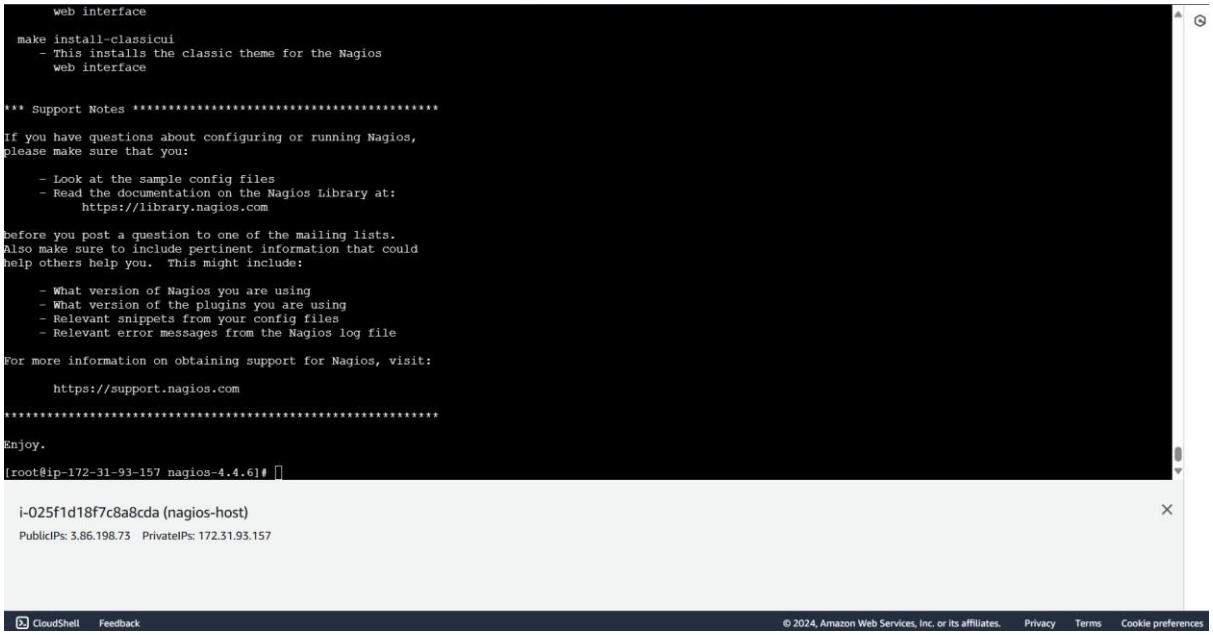
Step 7: Now, to compile the source code run the following command -
make all



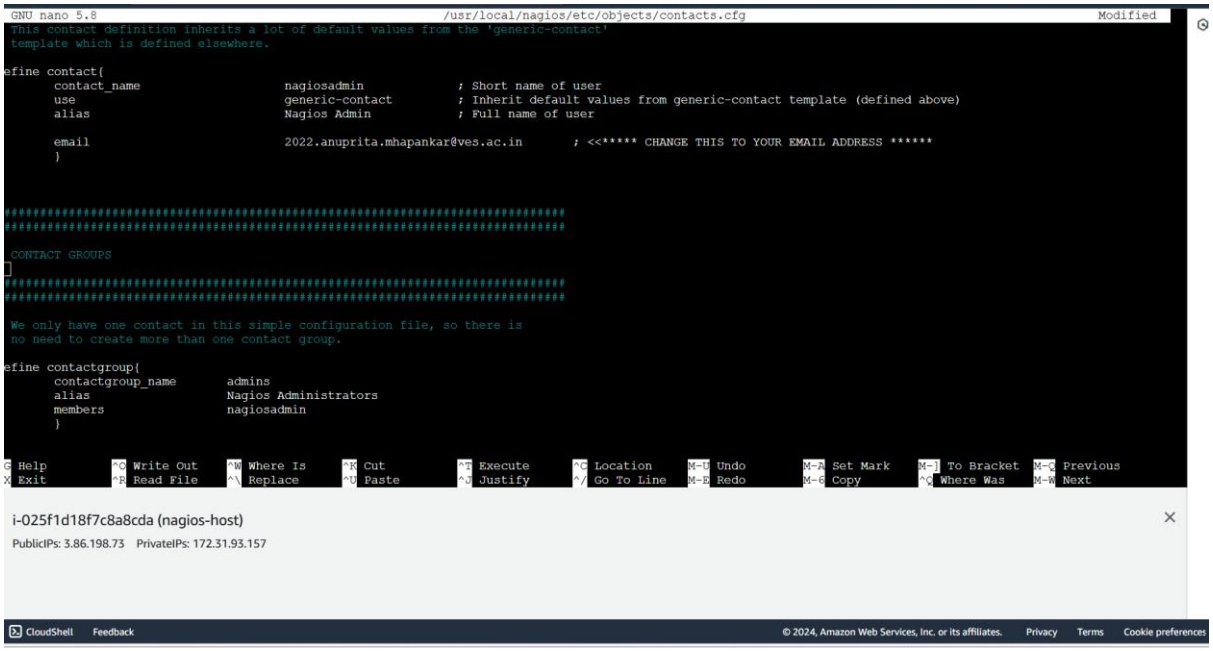




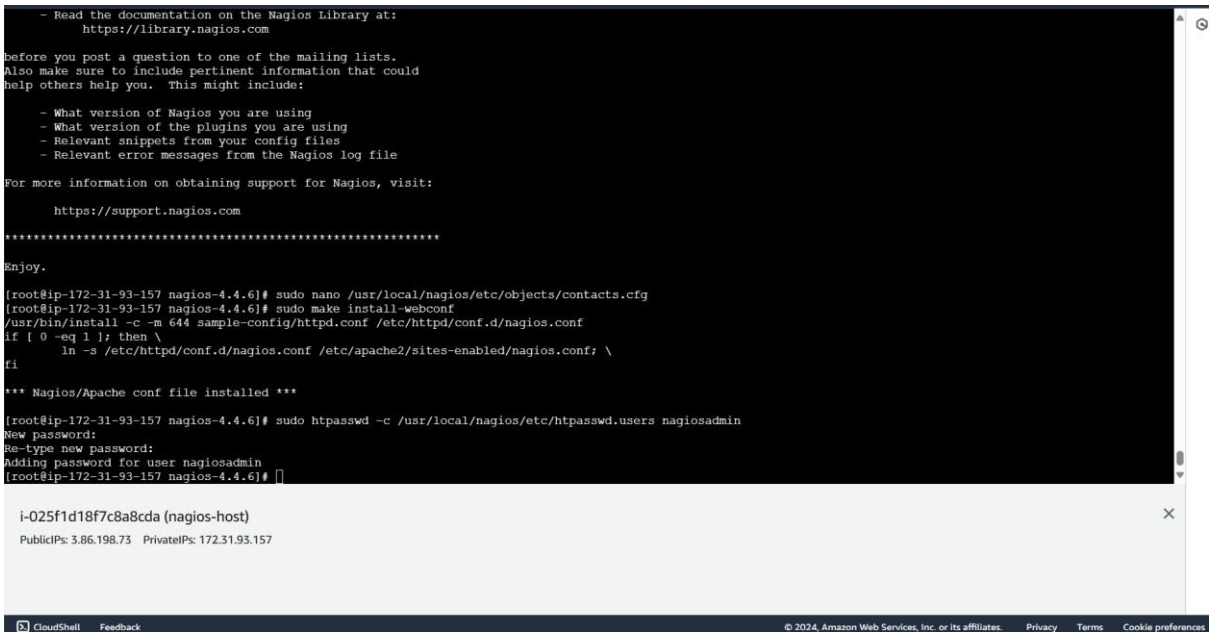
To resolve the errors run the following commands -
sudo yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel
rm -rf nagios-4.0.8
cd ~/downloads/nagios-4.4.6
./configure --with-command-group=nagcmd
make all
sudo make install

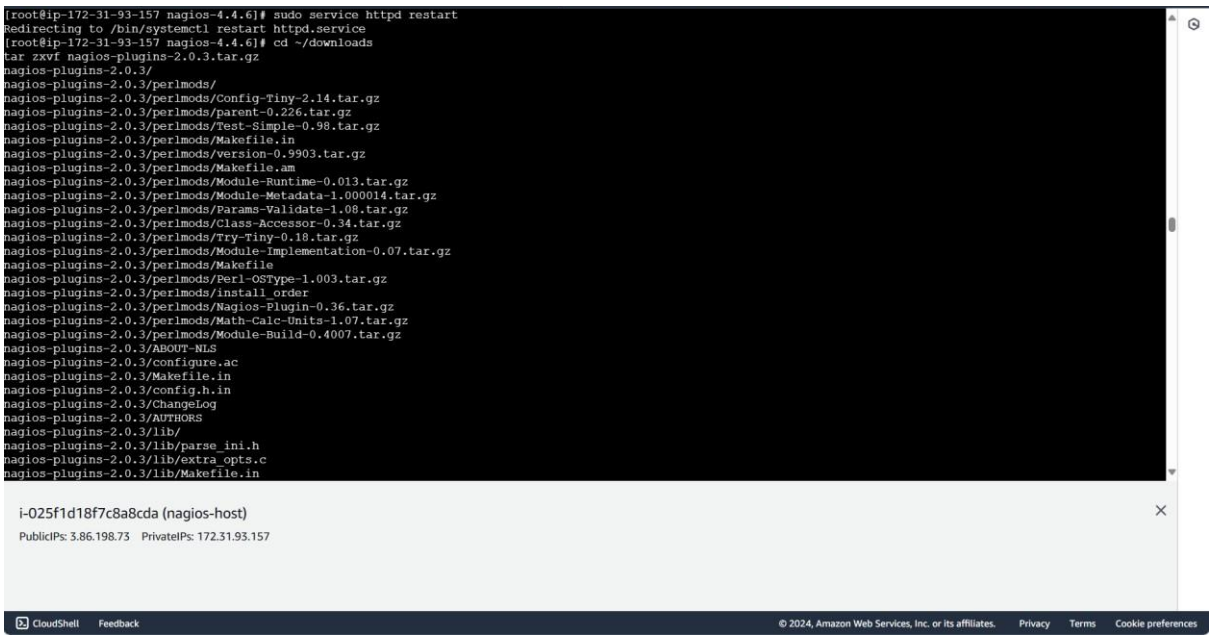


Step 8: Edit the config file and change the email address.
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

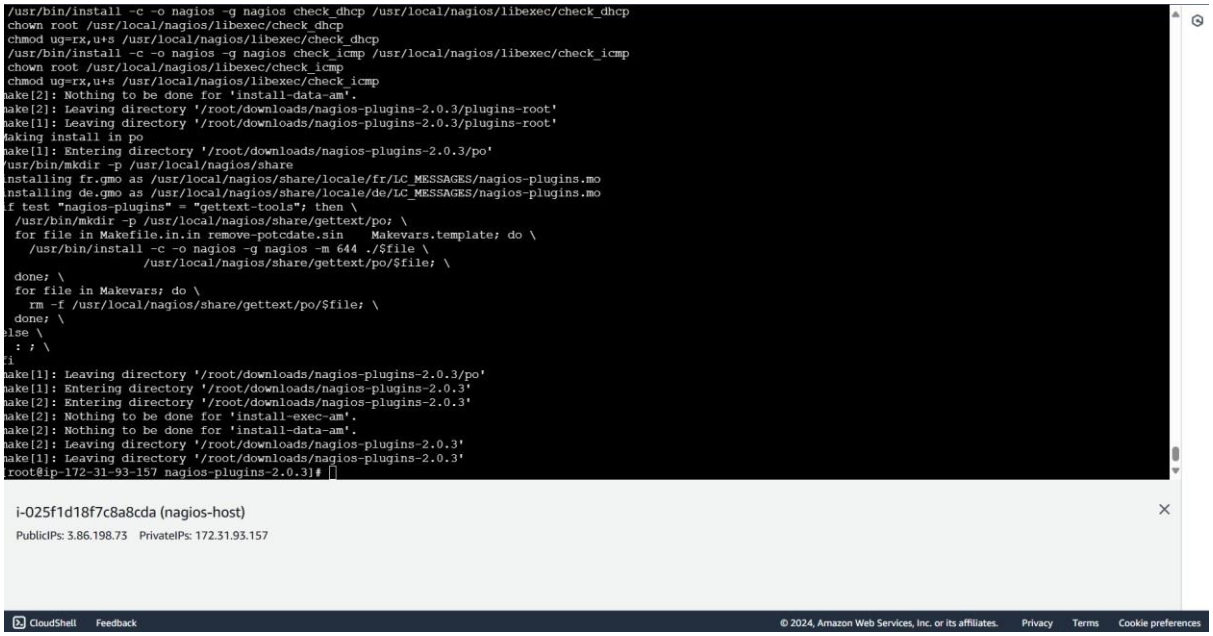


Step 9: Now run the following commands –
sudo make install-webconf
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
sudo service httpd restart
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz





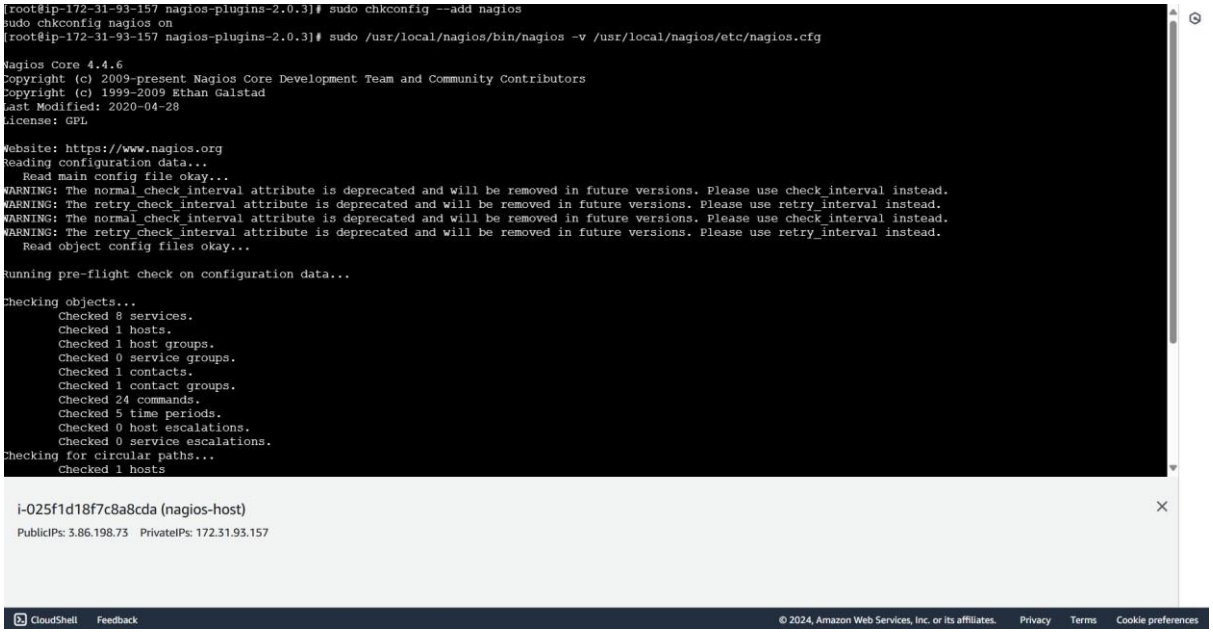
Step 10: Compile and install plugins
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install



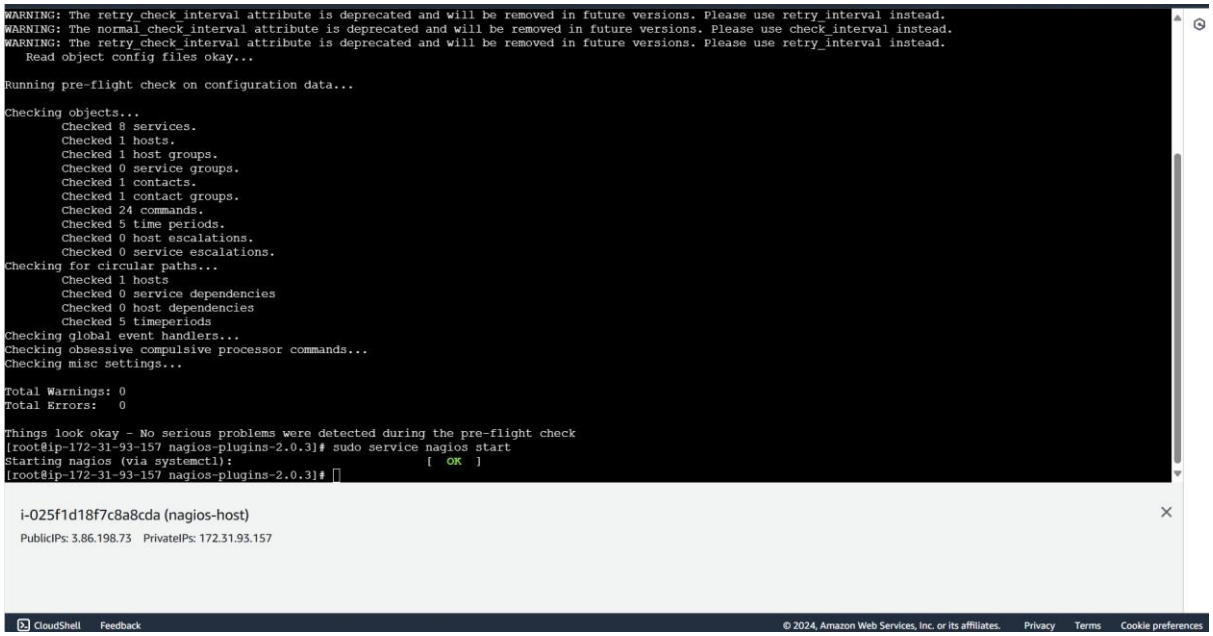
Step 11: To start nagios run the following commands –
sudo chkconfig --add nagios

sudo chkconfig nagios on

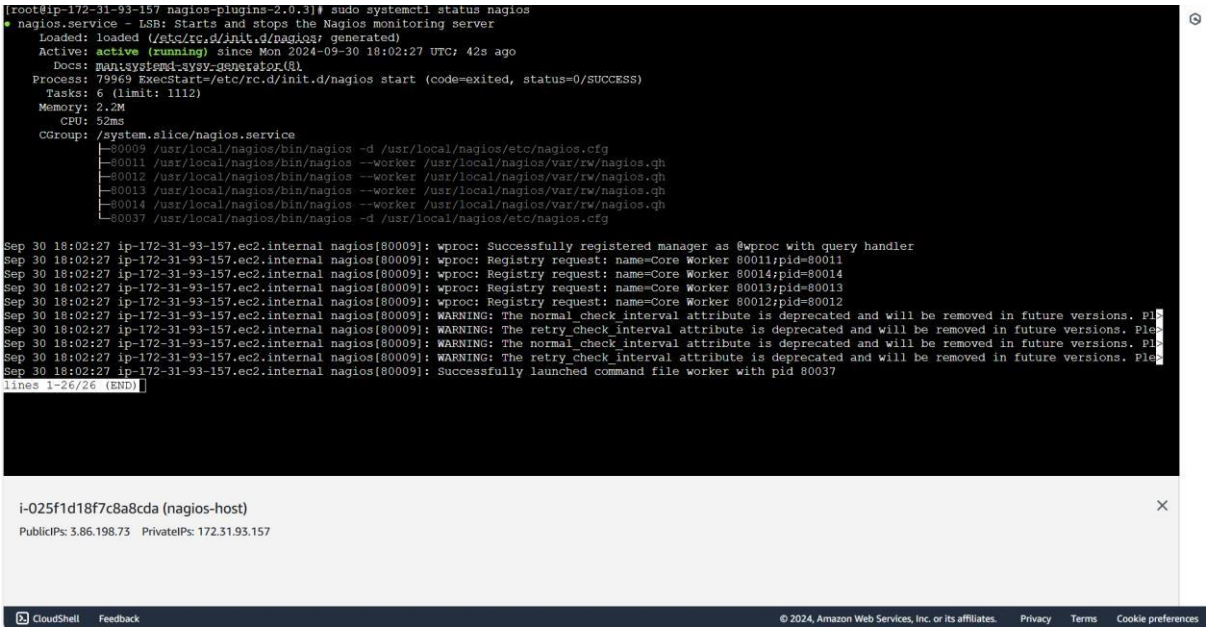
Verify using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



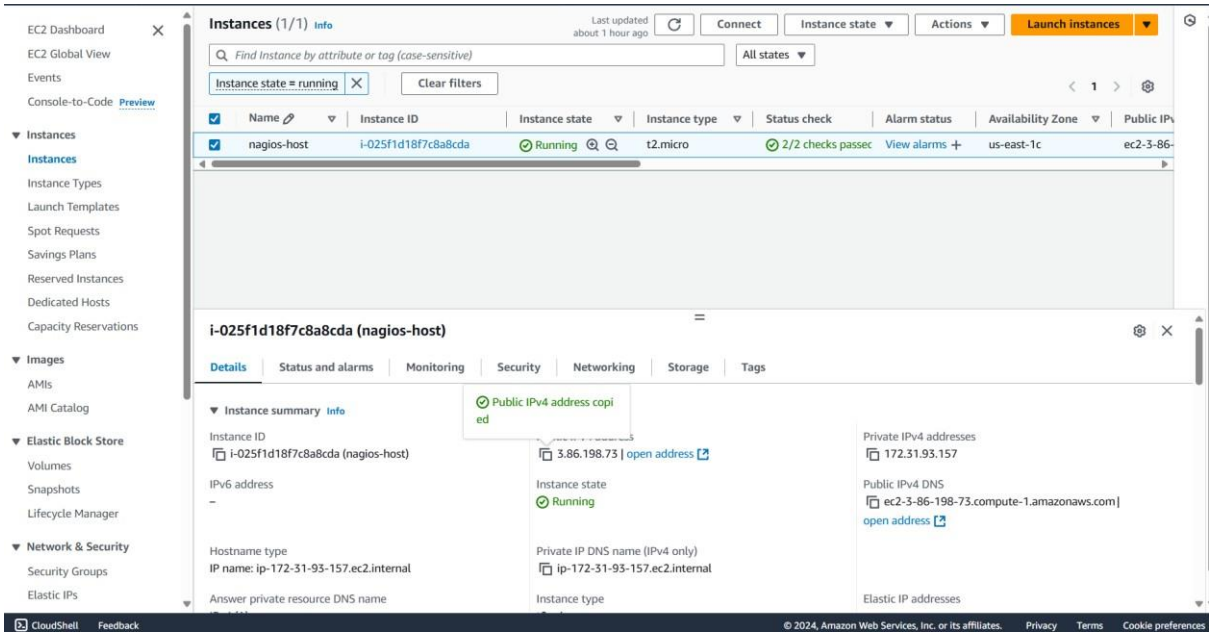
If there are no errors run the following command –
sudo service nagios start



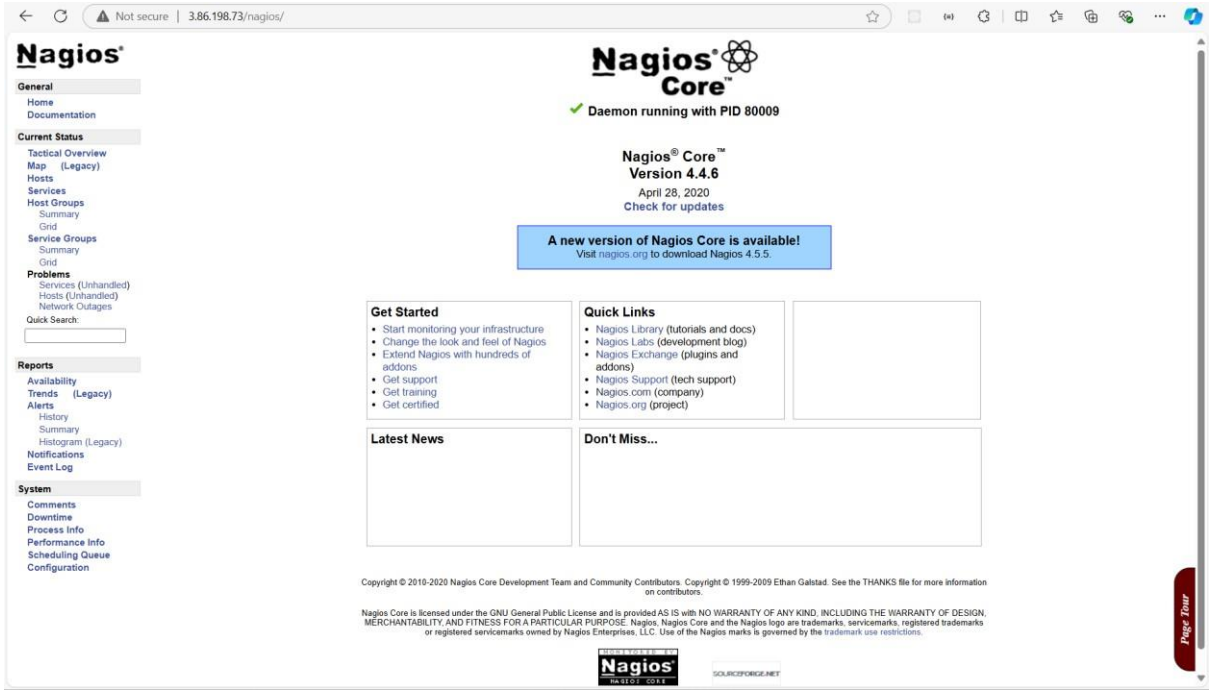
Check status using the following command -
sudo systemctl status nagios



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



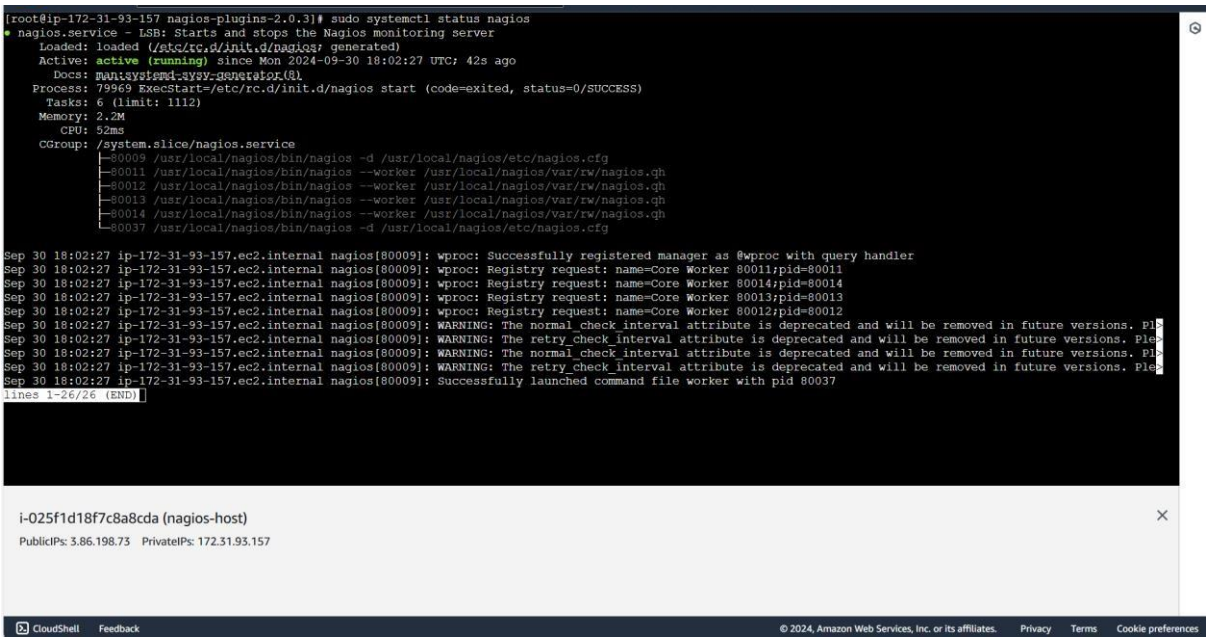
Step 13: Now visit http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.



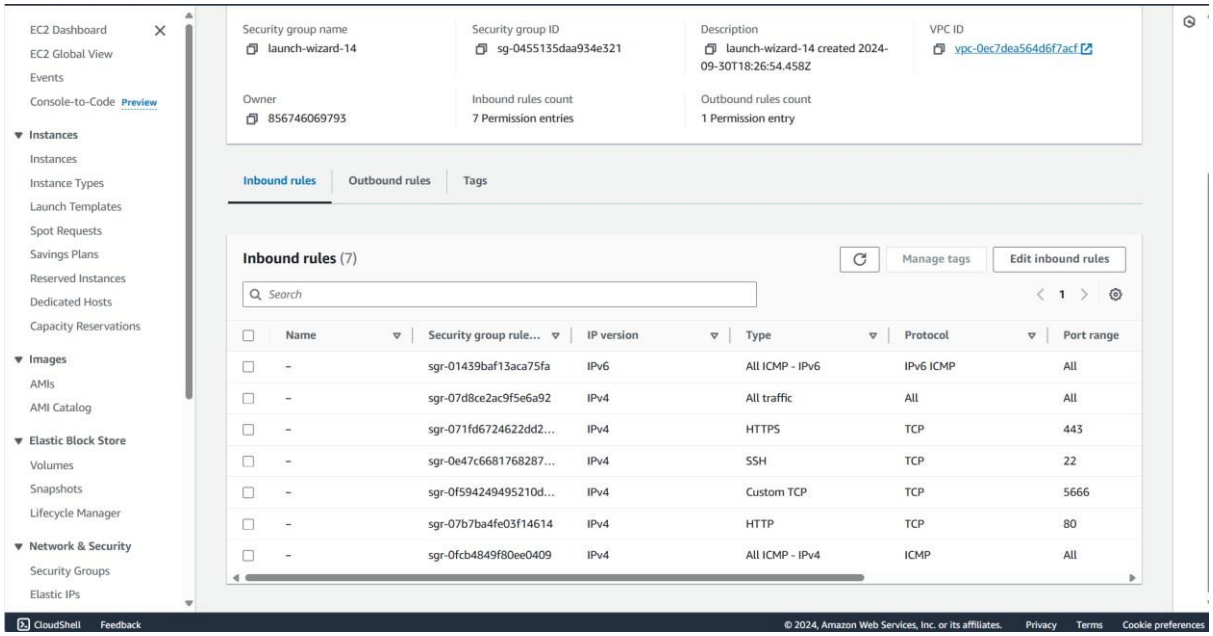
Experiment 10

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

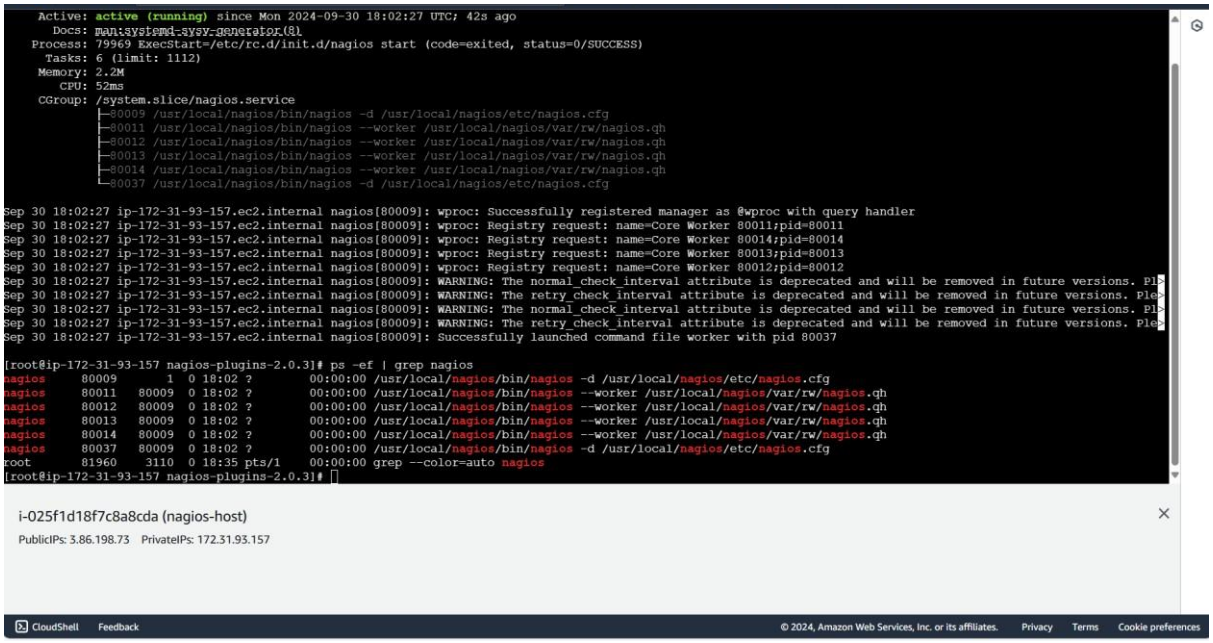
Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -
sudo systemctl status nagios
on the nagios-host instance.



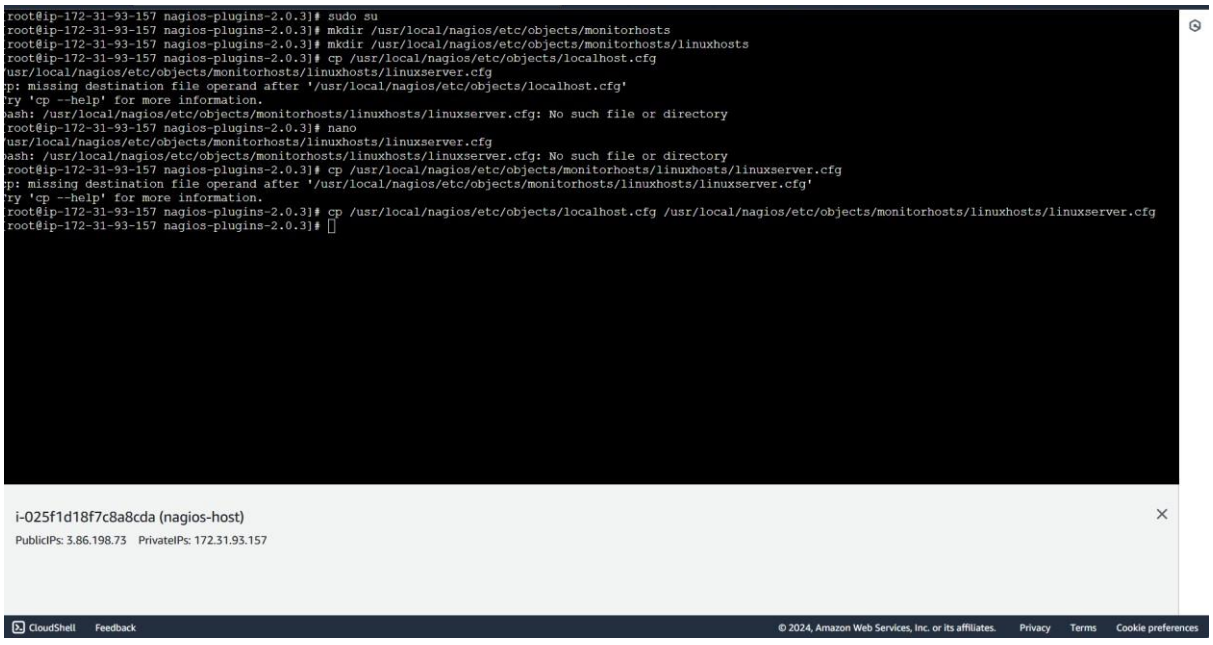
Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.
For now, leave this machine as it is, and go back to your nagios-host machine.



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

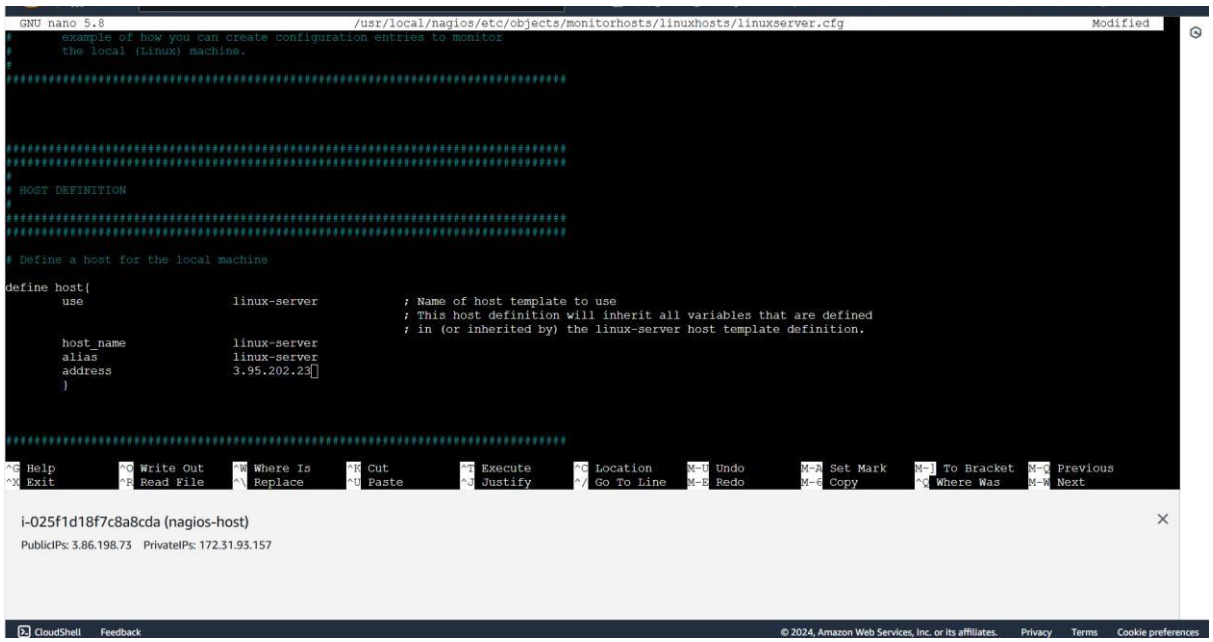


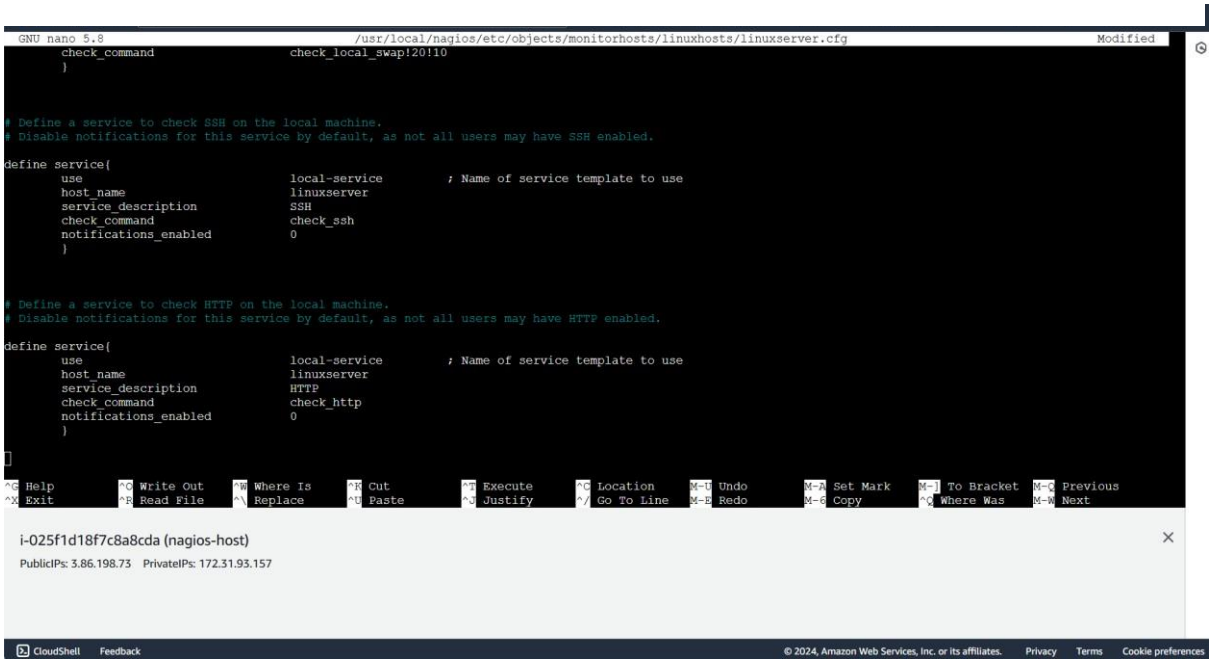
Step 4: Now, run the following commands -
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp/usr/local/nagios/etc/objects/localhost.cfg/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg



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

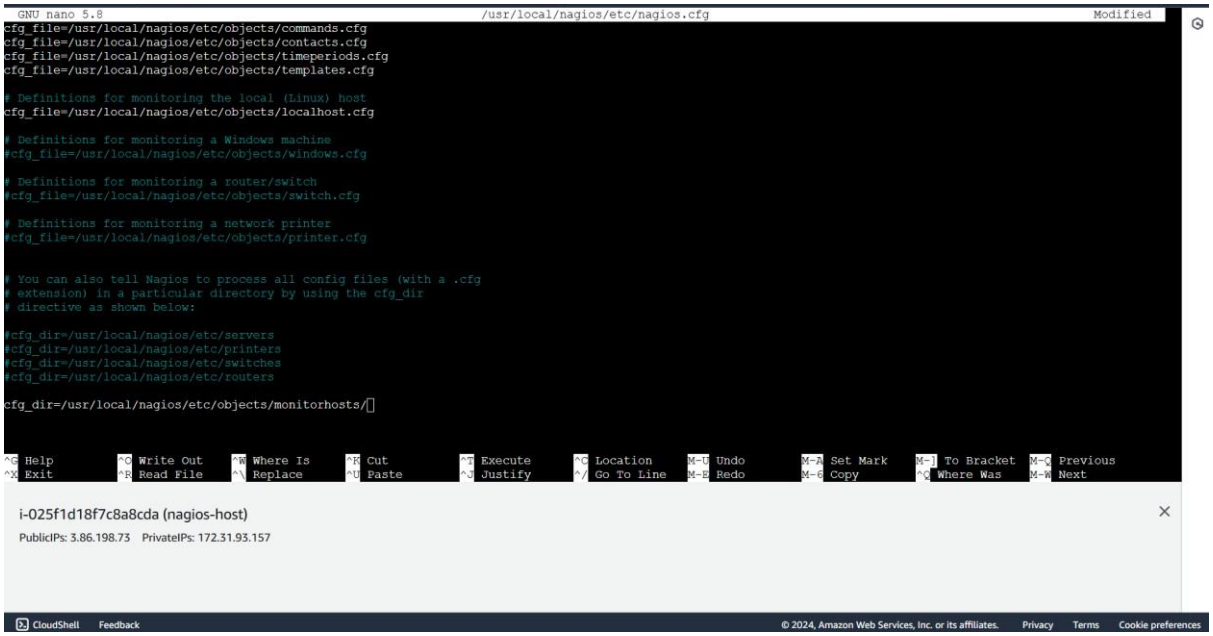
Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
Change address to the public IP address of your LINUX CLIENT.
Change hostgroup_name under hostgroup to linux-servers1





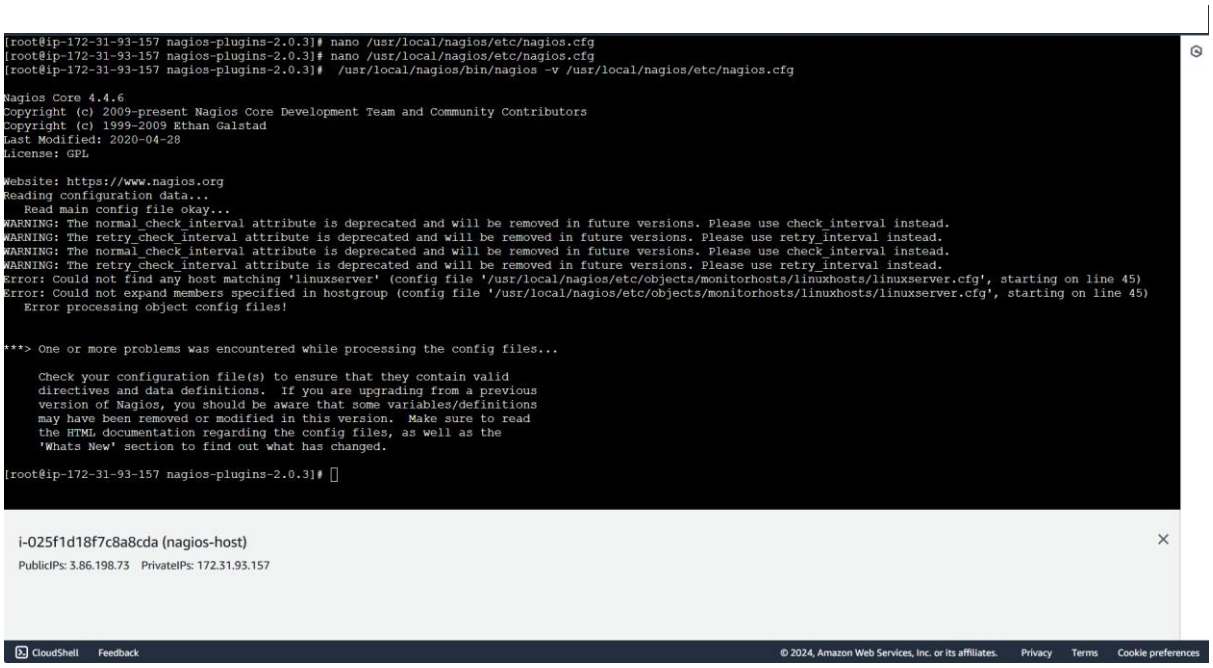
Step 6: Open Nagios config file and add the following line -
nano /usr/local/nagios/etc/nagios.cfg

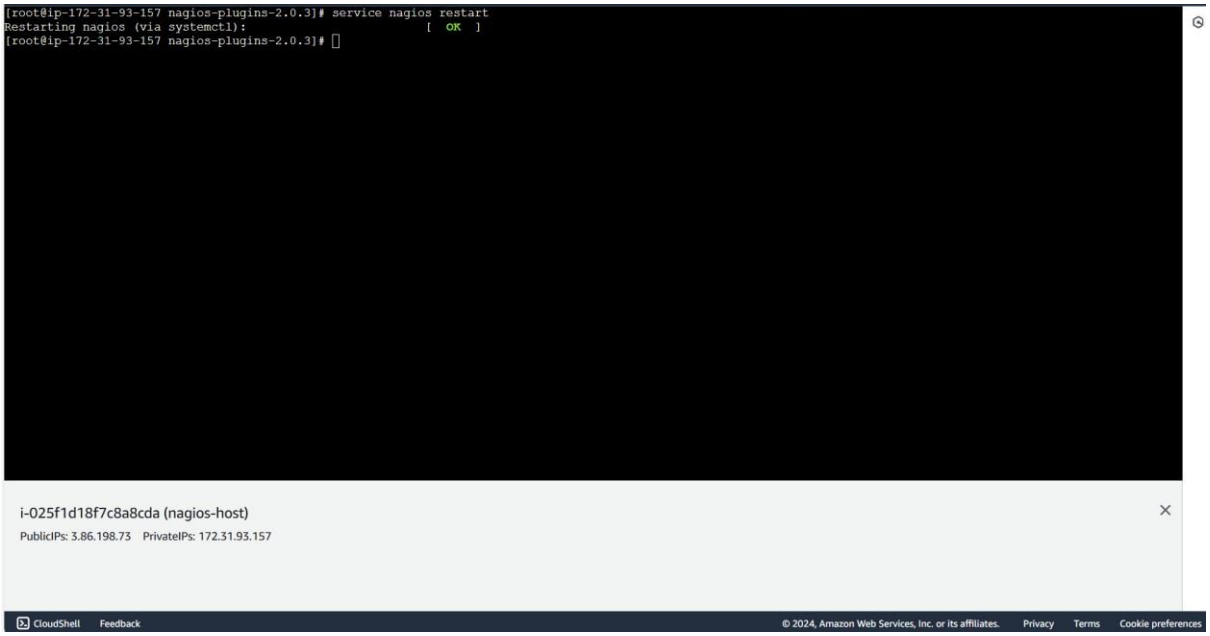
Then add this line -
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



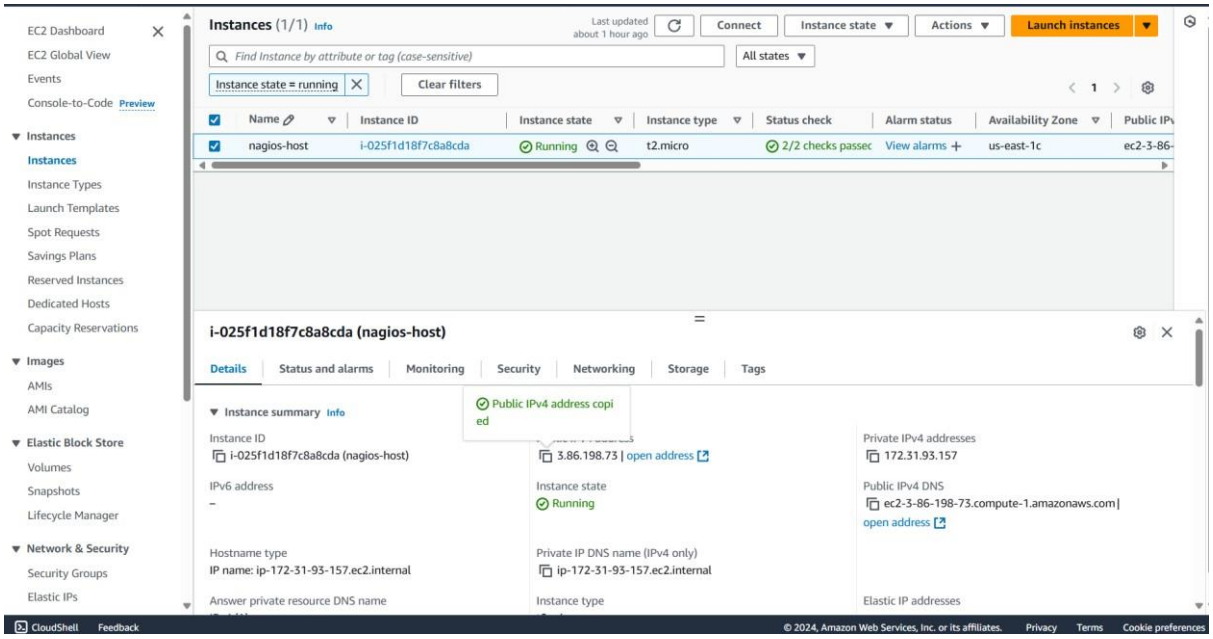
Step 8: Verify configuration files using the following command -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

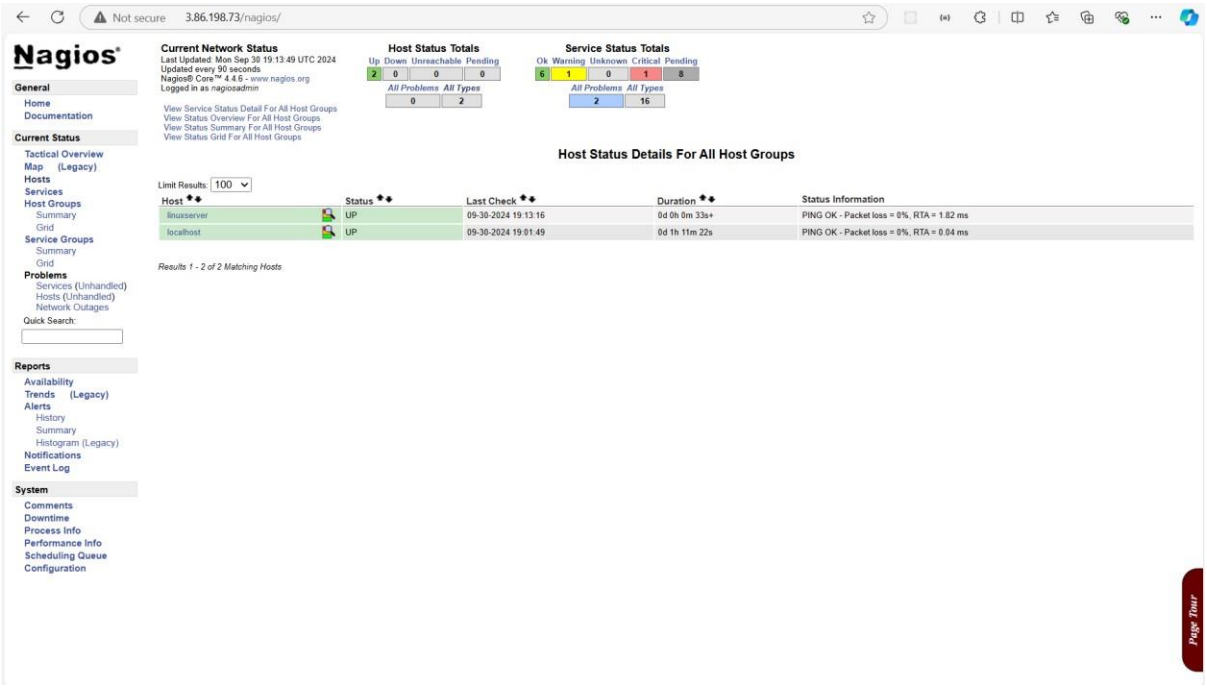
If there are no errors, run the following command -
sudo service nagios start



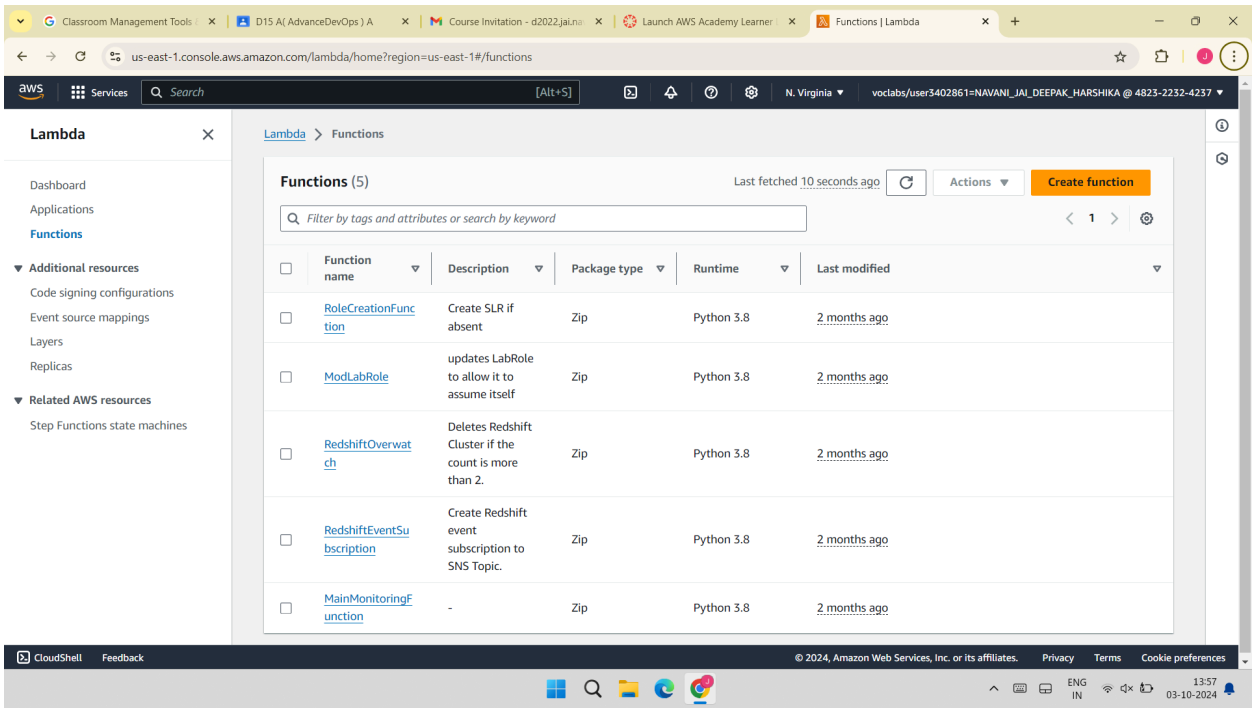


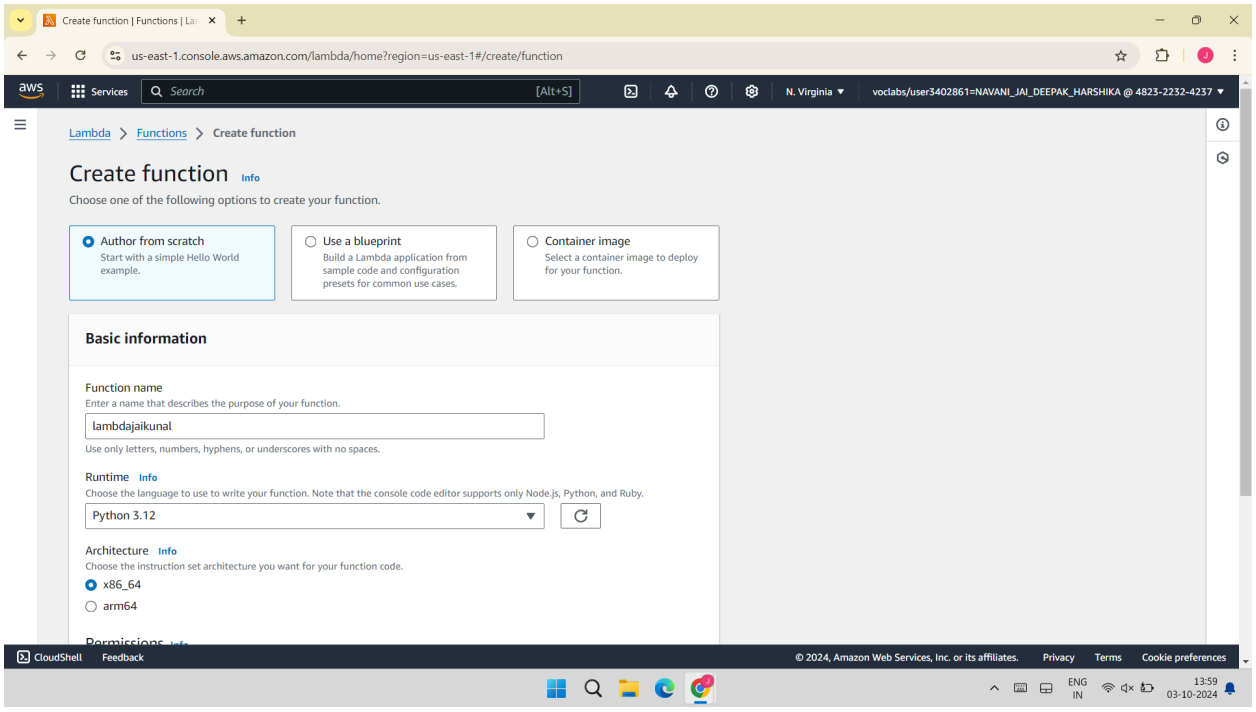
Step 9: After entering the correct credentials, you will see this page.

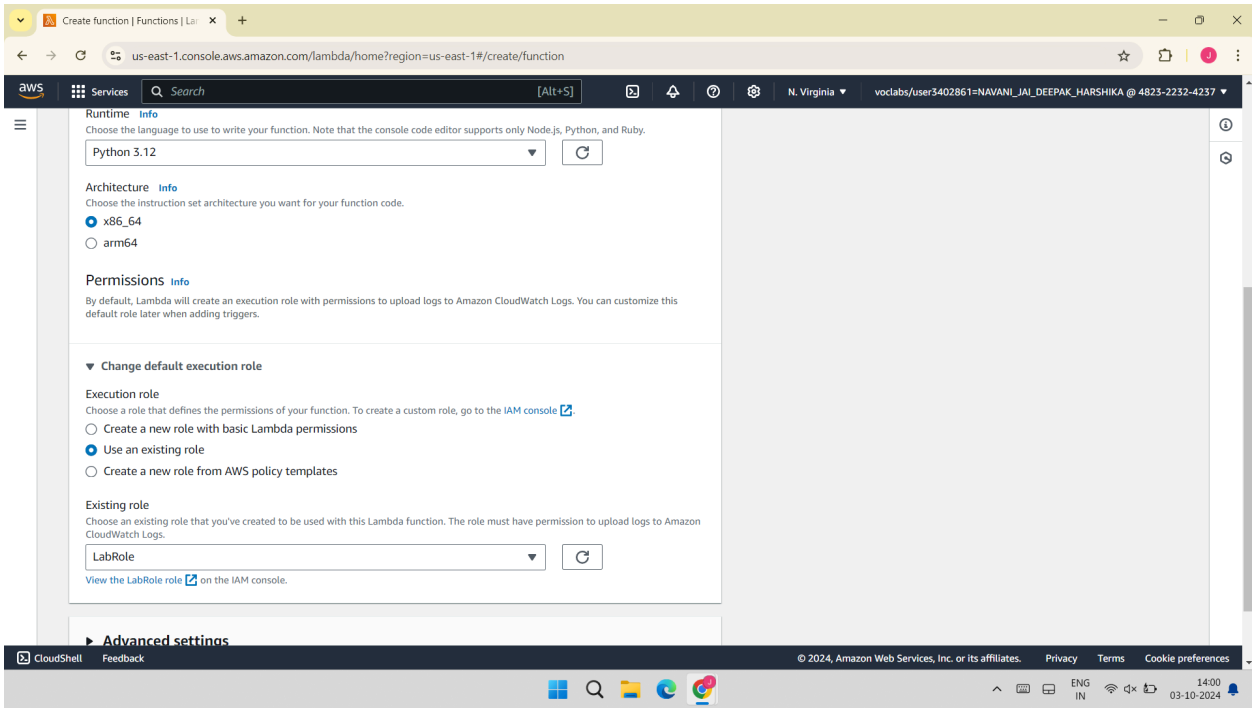


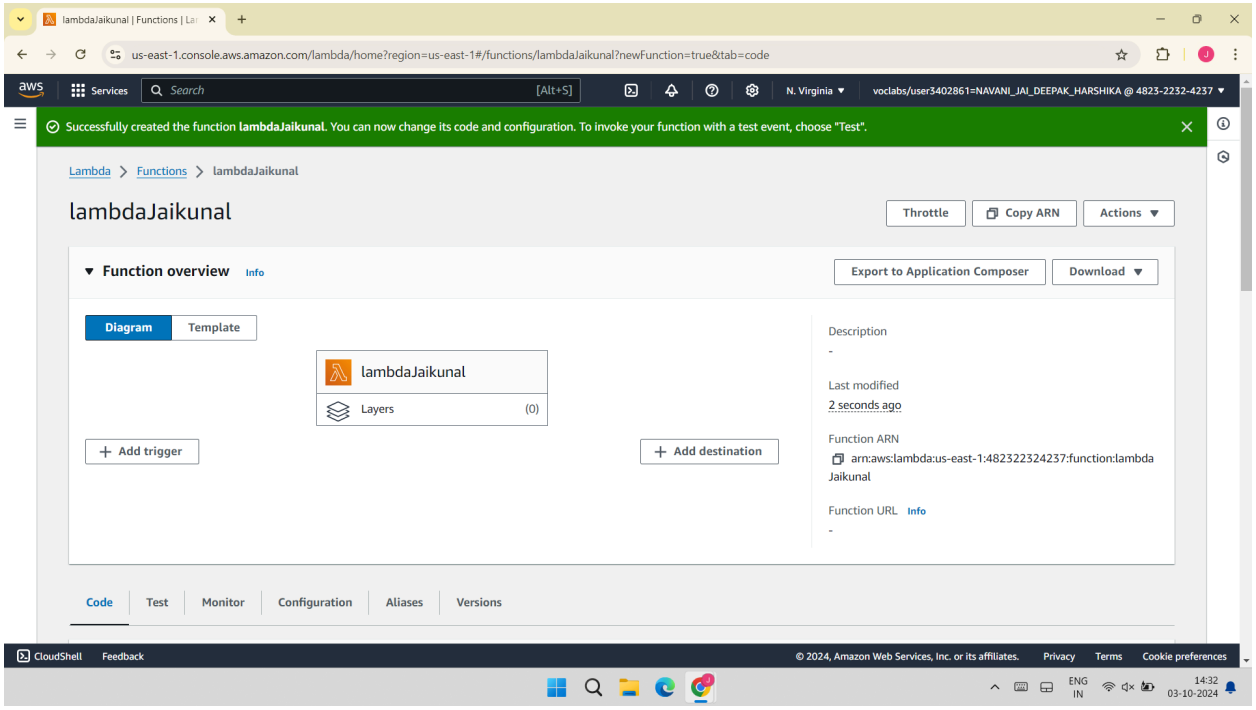

Not secure
3.86.198.73/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts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

Advance devops- experiment 11
Jai navani
D15-A(31)






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


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

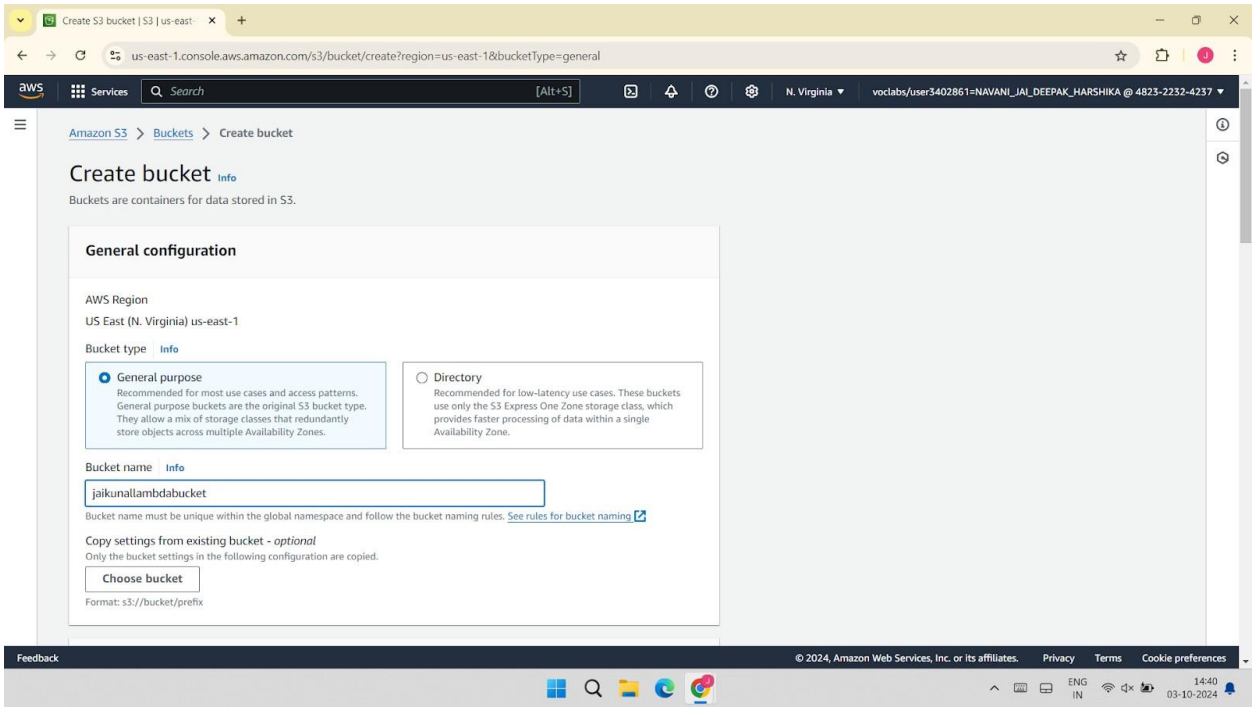
Advanced DevOps Exp-12

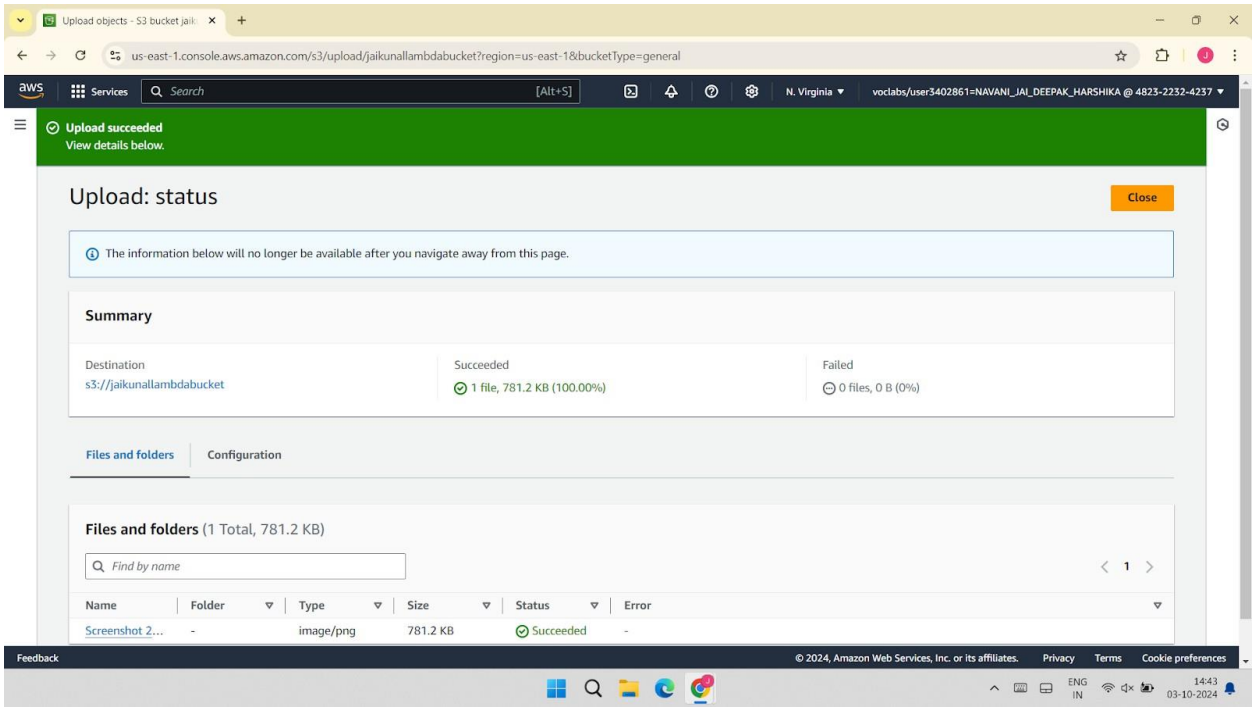
Jai Navani

D15A 31

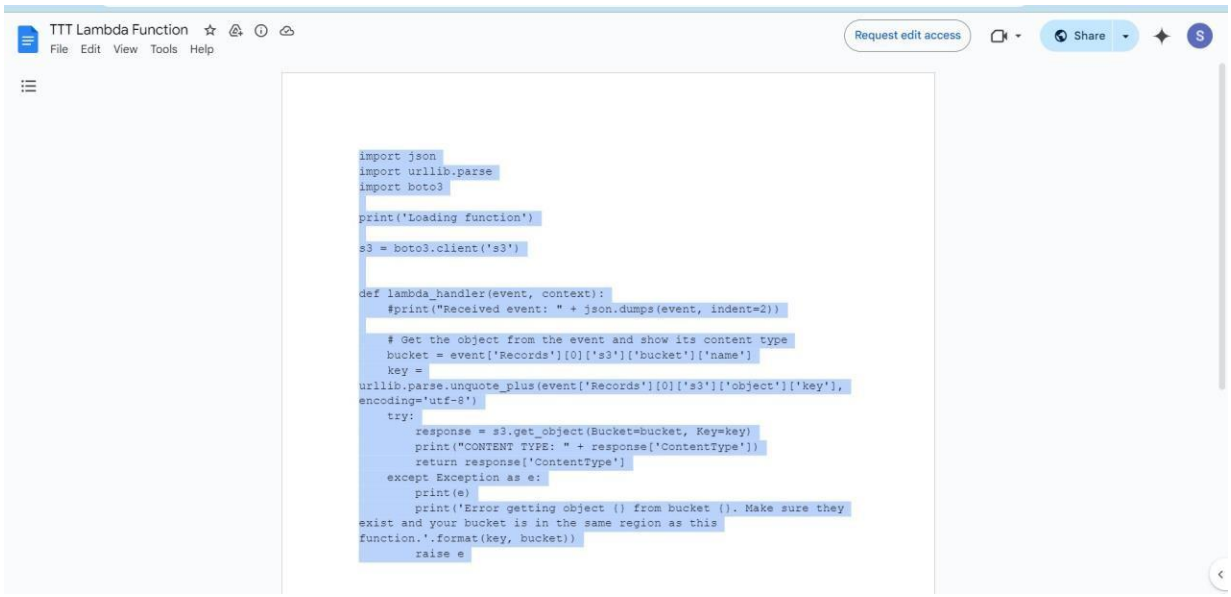
Procedure:-

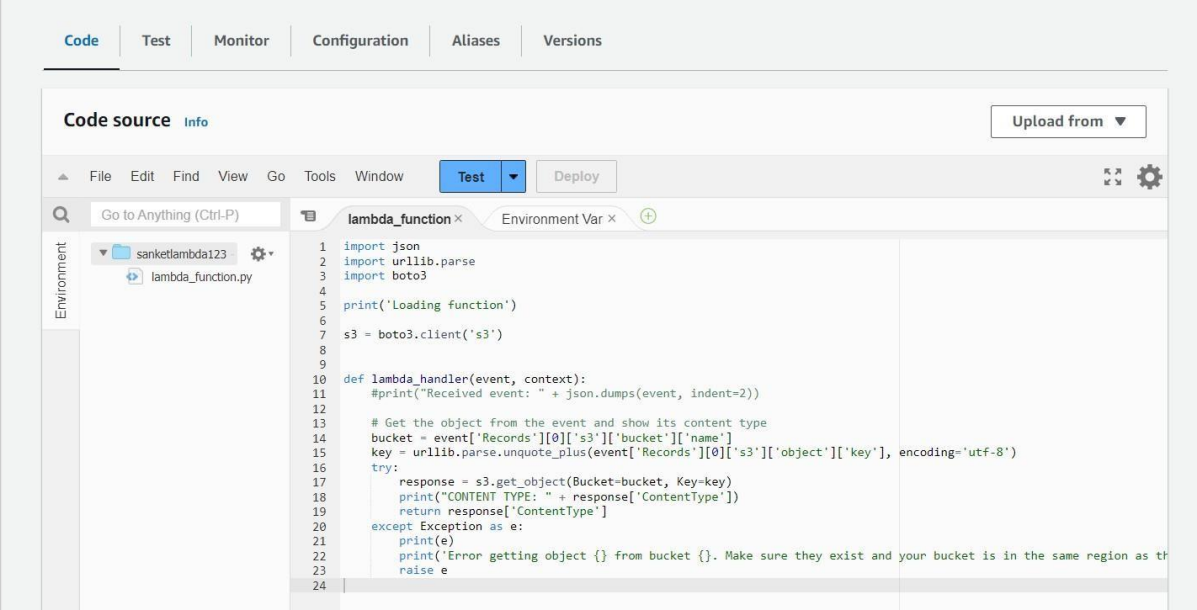
1. Create an S3 bucket of the same location as that of the Lambda function



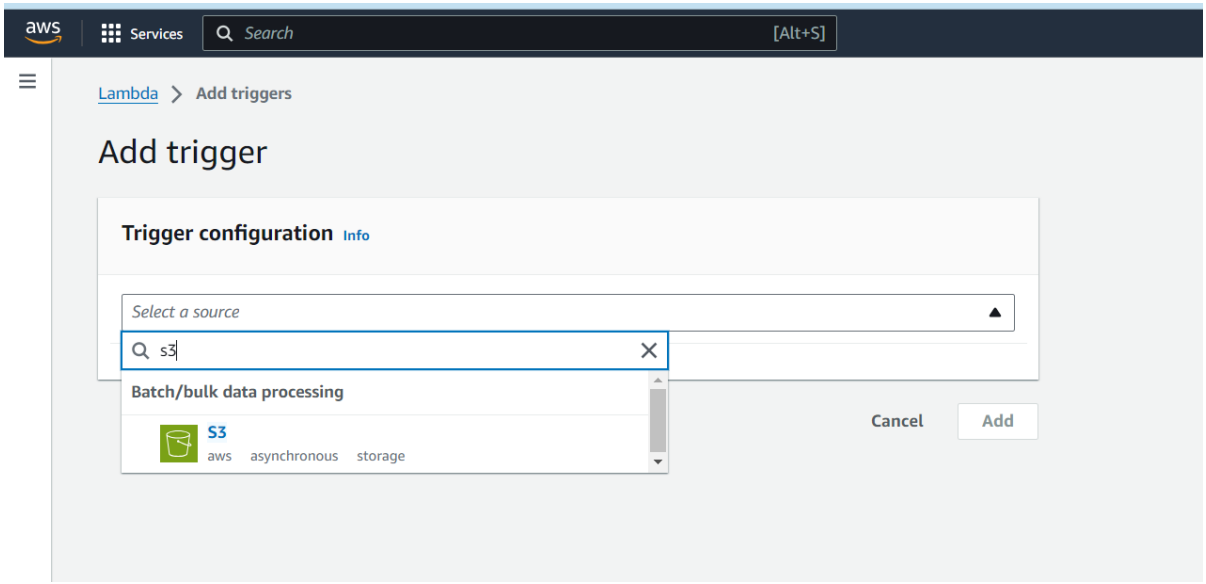

Upload objects - S3 bucket jaik
us-east-1.console.aws.amazon.com/s3/upload/jaikunallambdabucket?region=us-east-1&bucketType=general
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237
Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://jaikunallambdabucket
Succeeded
1 file, 781.2 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 781.2 KB)
Find by name
1
Name
Folder
Type
Size
Status
Error
Screenshot 2...
-
image/png
781.2 KB
Succeeded
-
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
14:43
03-10-2024

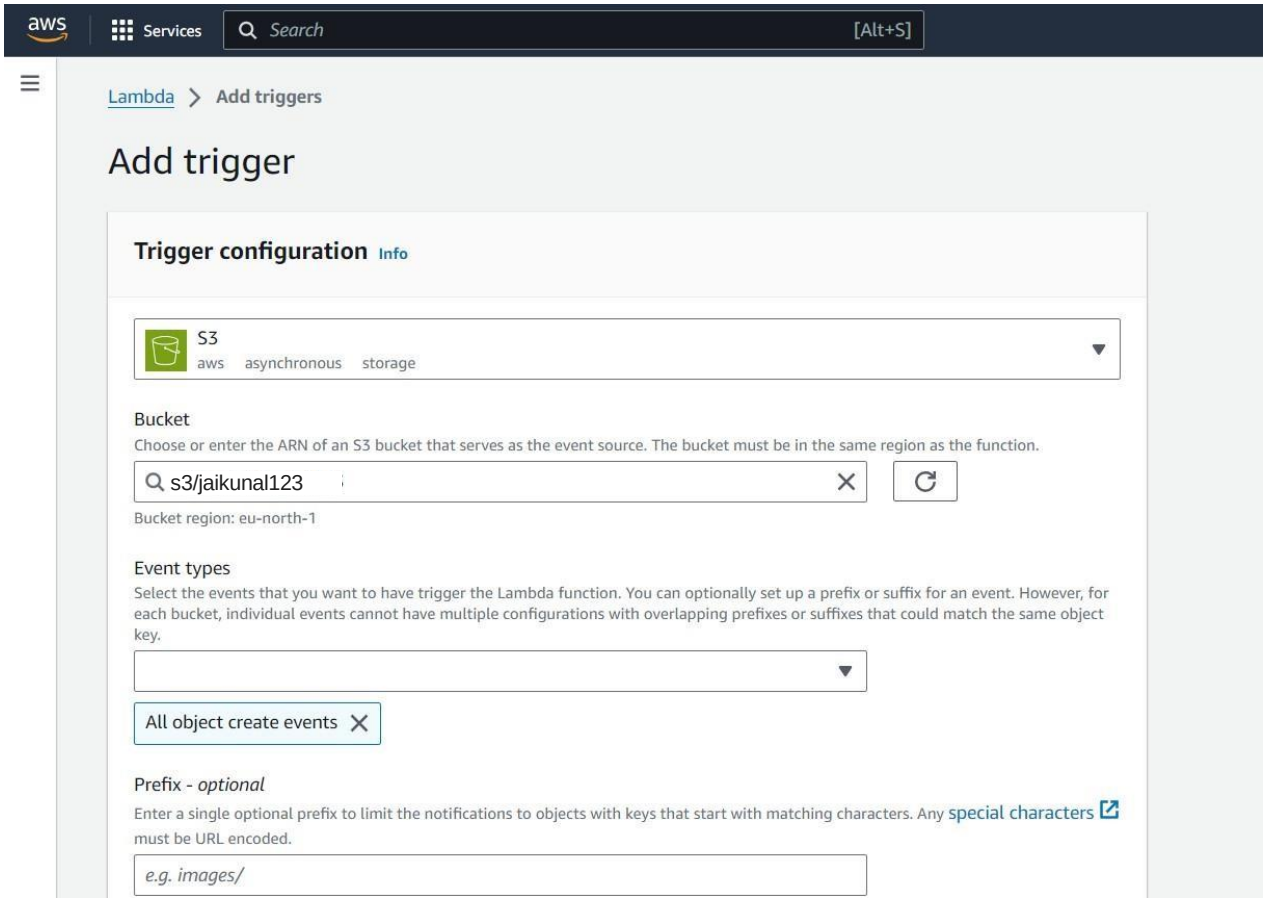
2. After creating the Lambda function copy a code available on the internet which allows the Lambda function to access the S3 bucket contents.

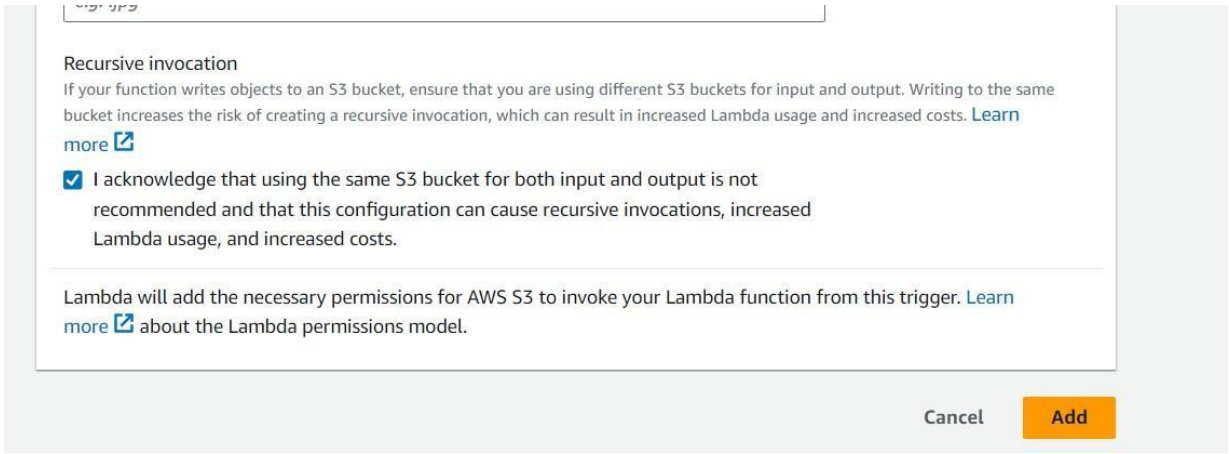




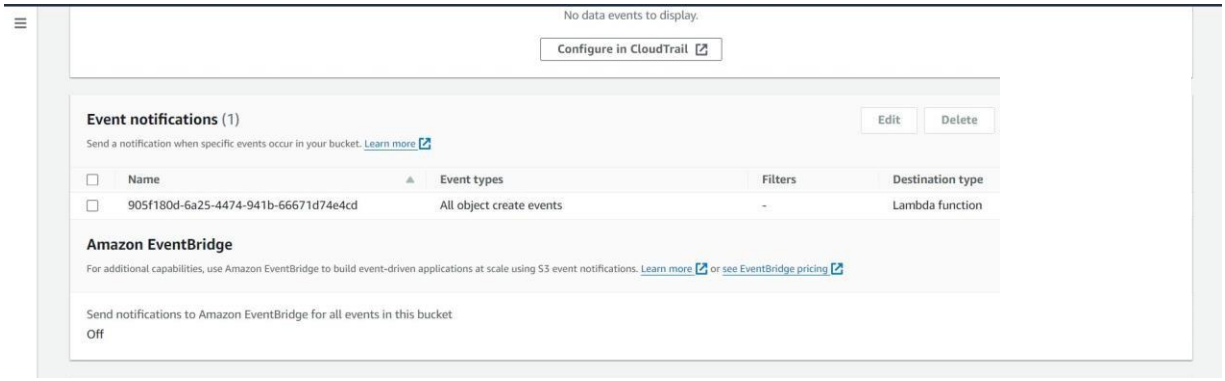
3. Add a trigger to the Lambda function so any changes in the S3 bucket will be first visible to the user.

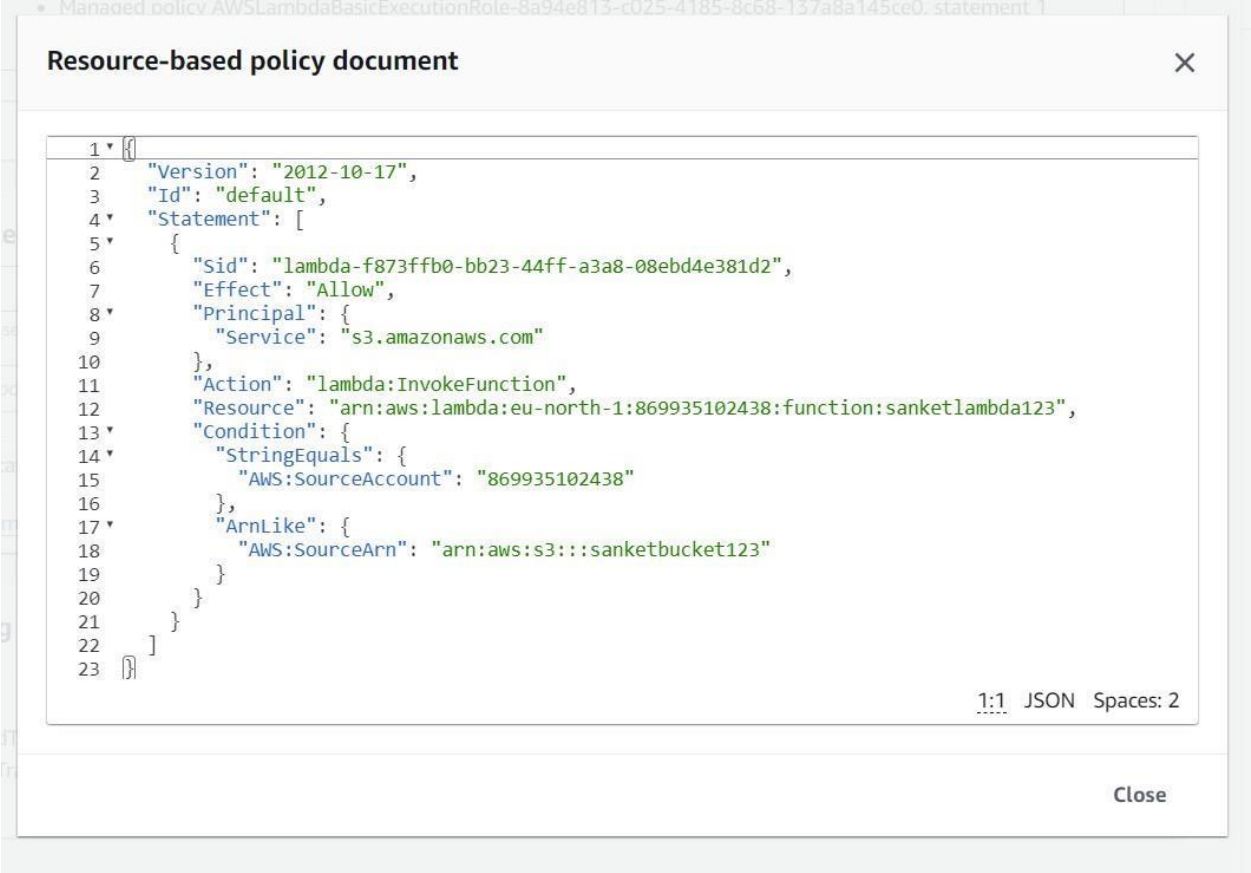



aws
Services
Search
[Alt+S]
Lambda
Add triggers
Add trigger
Trigger configuration
Info
S3
aws
asynchronous
storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/jaikunal123
Bucket region: eu-north-1
Event types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

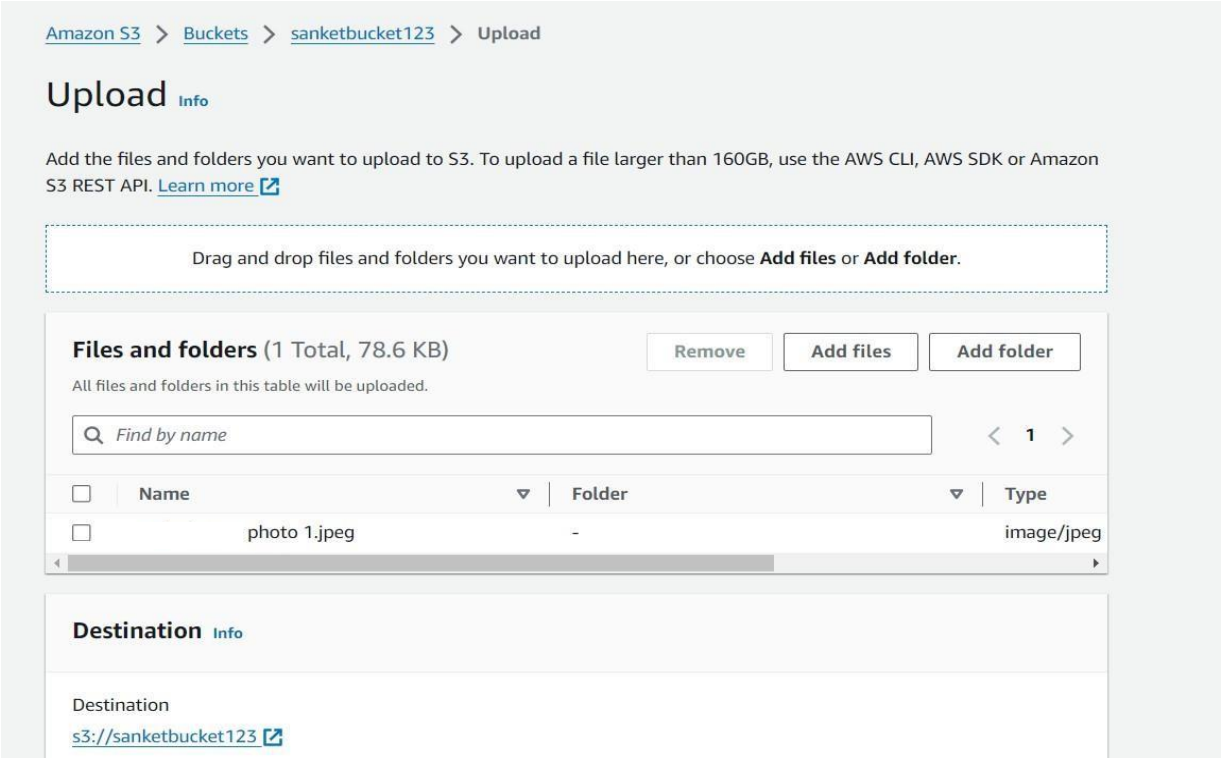

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more
I acknowledge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

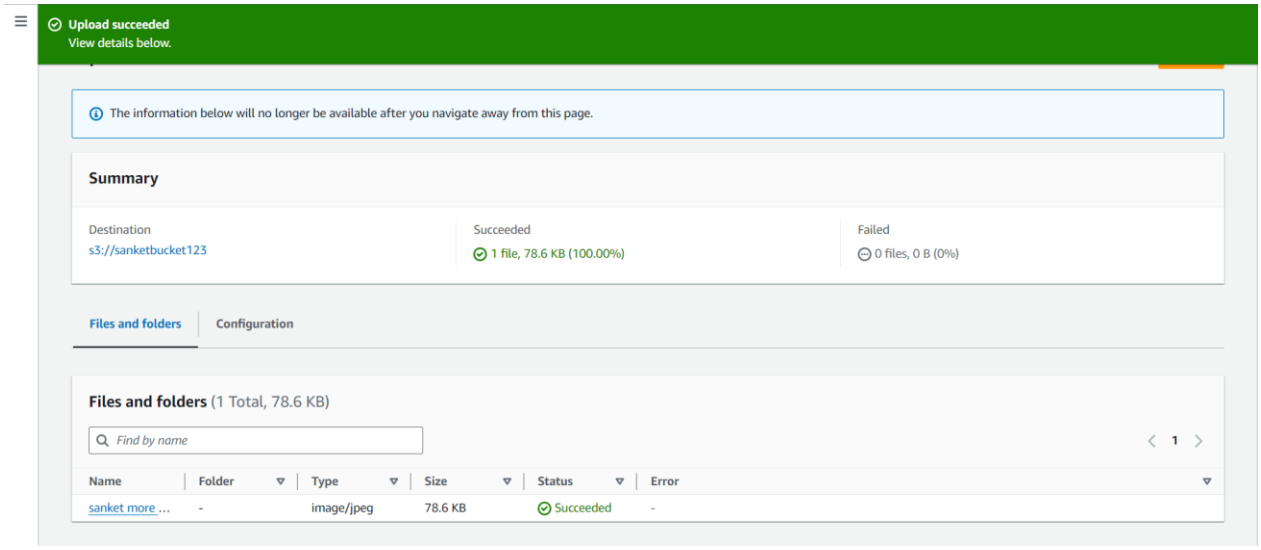
4. In the event notification of the S3 bucket we can see that it has been connected to the Lambda function .



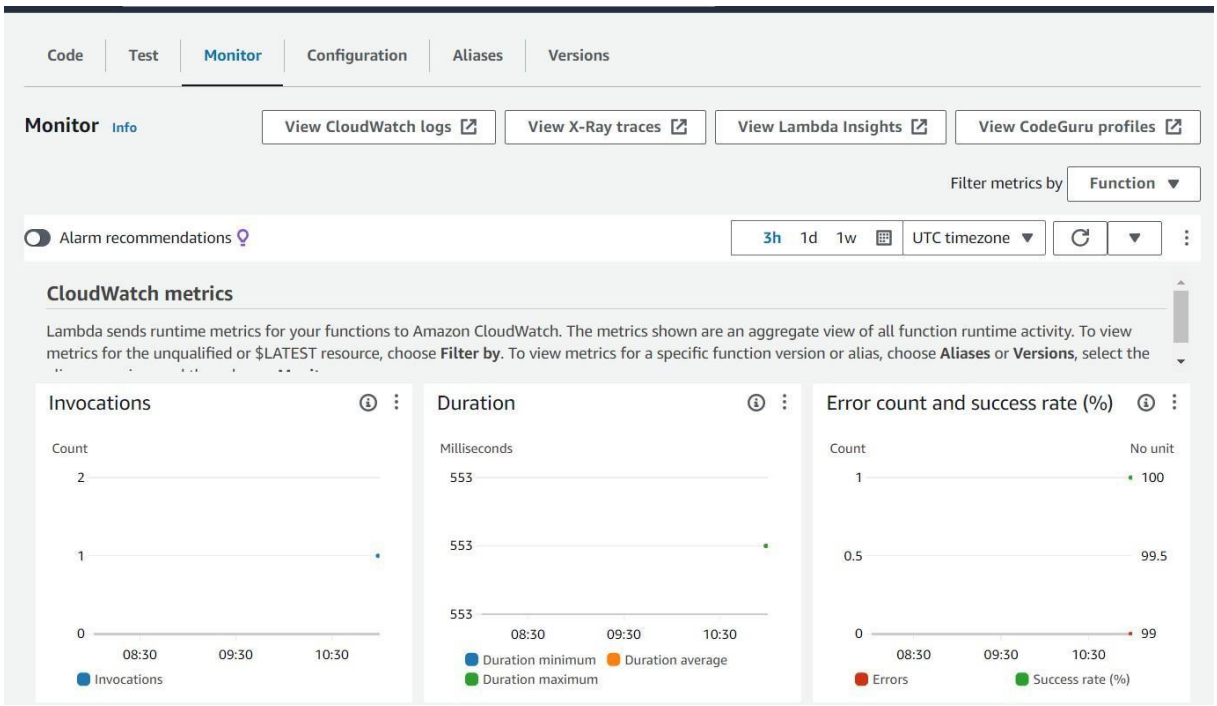


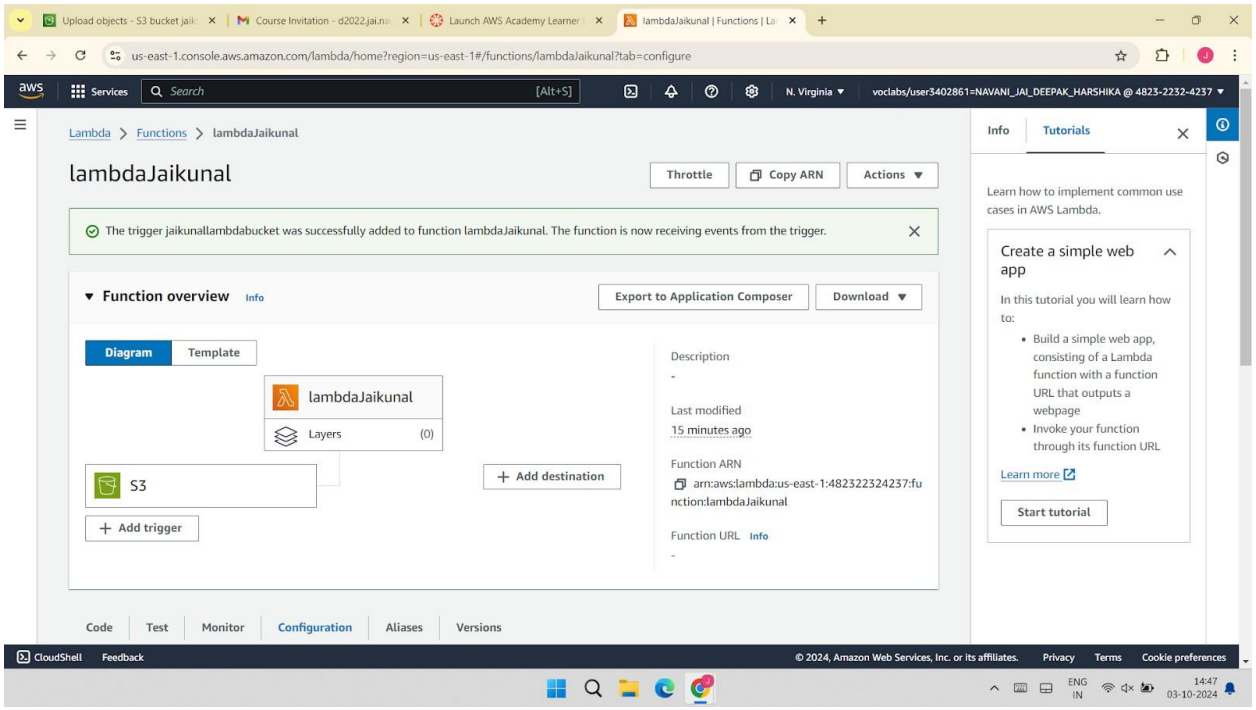
5. Upload a photo to the S3 bucket

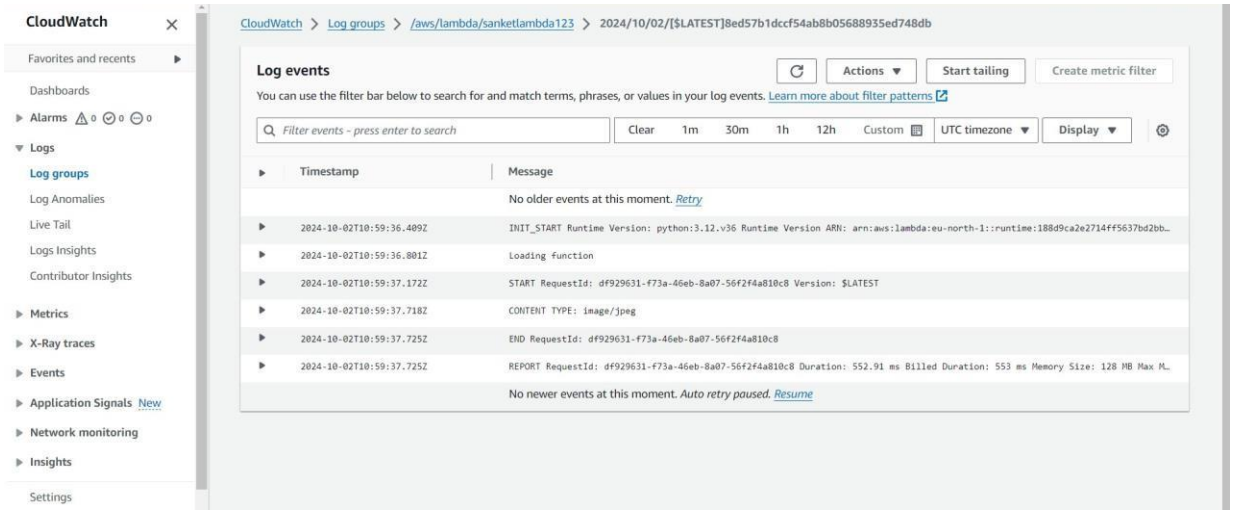




6. Now run the function and in the cloud watch logs of AWS you can see the message printed and all the other details of the working of the Lambda function.




us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/lambdaJaikunal?tab=configure
Lambda > Functions > lambdaJaikunal
lambdaJaikunal
Throttle
Copy ARN
Actions
The trigger jaikunallambdabucket was successfully added to function lambdaJaikunal. The function is now receiving events from the trigger.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
lambdaJaikunal
Layers
(0)
S3
Add destination
Add trigger
Description
-
Last modified
15 minutes ago
Function ARN
arn:aws:lambda:us-east-1:482322324237:function:lambdaJaikunal
Function URL
Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


CloudWatch
Favorites and recents
Dashboards
Alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
X-Ray traces
Events
Application Signals New
Network monitoring
Insights
Settings
CloudWatch > Log groups > /aws/lambda/sanketlambda123 > 2024/10/02/[$LATEST]8ed57b1dccf54ab8b05688935ed748db
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
2024-10-02T10:59:36.409Z INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:eu-north-1::runtime:188d9ca2e2714ff5637bd2bb...
2024-10-02T10:59:36.801Z Loading function
2024-10-02T10:59:37.172Z START RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Version: $LATEST
2024-10-02T10:59:37.718Z CONTENT TYPE: image/jpeg
2024-10-02T10:59:37.725Z END RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8
2024-10-02T10:59:37.725Z REPORT RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Duration: 552.91 ms Billed Duration: 553 ms Memory Size: 128 MB Max M...
No newer events at this moment. Auto retry paused. Resume

Advance devops assignment no. 4

(05/05) R.

Q.1) Use S3 bucket and host video Streaming.

Ans: 1) log into AWS console

- Go to AWS management console
- Enter your login credentials

2) Create an S3 bucket

- On the console, search for S3 in the search bar and select S3 from the results.
- click Create bucket
- Give your bucket a unique name.
- Choose a region
- Scroll down and uncheck Block all public access
- Confirm by checking the acknowledgement box
- Click create bucket.

3) Upload your video file to S3

- Click on your newly created bucket
- Click on the upload button.
- Add your video file from your computer
- Click upload to start the upload process.

Set Permission for public Access.
- Confirm the action by clicking make public again.

5. Get the video URL.

- After making the file public, click on the video file
- You will see a URL for the video under Object URL.
- Copy the URL

6. Edit the Bucket policy

```
{
  "Version": 2012-10-17",
  "Statement": [
    {
      "Sid": PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action":" S3:GetObject",
      "Resource": " arn:aws:S3:::video-bucket/* "
    }
  ]
}
```

Q.2) Discuss BMW and Hot Star case studing using AWS.

→ BMW: A connected driving Expreince BMW is a leader in car technology and AWS has helped them a lot in creating connected cars.

- Connected cars features: BMW has developed features like remote diagonostic, over the air-updates and real-time traffic info. AWS provides the necessary tools and infrastructure to make the features work.

- Improving Customer experience: They use AWS to enhance the customers interact with their cars for example BMW has added voice activated assistance and personalized suggestions using services like Amazon lex.

- Better manufactering: BMW uses AWS to improve its manufactering. By analyzing data from machines, they can find problem early and inrease efficieny.

- A streaming revolution: Hotstar is India's biggest video streaming platforms offering a huge range of movies, TV shows and sports AWS has been key of Hotstar success helping them.

- Manage huge traffic: During big events like IPL many people watch at once. AWS helps Hotstar handle this stage in traffic without slowing down.

- Quality streaming: AWS services like Amazon Kinesis ensures that users highly stream quality.

- Personalized Recommendation: Hotstar uses AWS machine learning tools to suggest contents to users making their viewing experience more enjoyable

Key AWS services:

- Compute : Amazon EC2

- Storage : Amazon S3, Amazon EFS

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

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

Advantages of Kubernetes:

1) Portability: Application can be moved easily between different environment without major changes.

2) Scalability: Kubernetes can automatically scale applications up or down based on traffic and demand.

3) Reliability: It features self-healing capabilities, meaning it can restart failed containers and balance workload.

Disadvantages of Kubernetes:

1) Complexity: It can be complicated to set up and manage. espically for those new to container

2) Step learning curve: Requires time and knowledge to fully understand and utilizes its features.

3) Resource Intensive: It may require more computing resources than simpler solutions.

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

Q.4) What are Nagios and explain how Nagios are used in E-services?

Ans Nagios is an open source monitoring tools that helps the organization keeps track of their IT infrastructure, including servers, networks and applications. It provides a way to ensure that systems are running smoothly and alerts users if any issues arise.

Key feature of Nagios:

- Monitoring: Tracks the performance and availability of servers, application and network devices.
- Alerts: sends notification via email or SMS when problems occur, so teams can respond quickly.

How Nagios is used in E-services:

Nagios plays a vital role in the operation of e-services by ensuring that online systems are reliable and efficient.

- Infrastructure monitoring: Nagios monitors services, databases and other IT components to ensure they are operational. If a server goes down, Nagios alerts the IT team immediately.

04/05

Advance devops (assignment no 2)

Create a REST API with serverless framework

Creating REST API with serverless framework efficient way to deploy serverless application that can scale automatically without managing servers. A powerful tool that deployment of services and serverless application across various cloud periods such as AWS, and google cloud.

ii) Serverless architecture: This design model allows developers to build application without waring about underlying infrastructure enabling focus on code & bussiness logic.

iii) REST API: Representational state transfer is architecture style for designing network application directly from your terminal.

Steps for creating RESTART for serverless framework

Install Serverless framework:

You start by installing Serverless framework CLI globally using node modules. This allows you to manage serverless application directly from your terminal

Creating a node js serverless project:

A directory is created for your project, where you will initialize a serverless service. This service will house all your lambda functions configuration and could services. Using the command serverless create your AWS

Node js microservices that will eventually deploy to AWS lamda.

3) Project structure:-
The project scoldfold creates essential files like handler.js (which contains code for lambda function and serverless.YML.

4) Create a REST API Resource:
In the serverless yml file you define function that handles port request of HTTP.

5) Deploy the service:
with the SLS deploy commands: serverless framework packages your application, uploads necessary resource to AWS and set up the infrastructure

6) Testing the API: once deployed you can test REST API using tools like Postman by making POST request to generated API.

7) Storing Data in Dynamods: To store submitted candidate data you integrate AWS DynamoDB as database

8) Adding more functionality: Adding functionalities like install candidates get candidates by ID.

9) AWS IAM permissions
you need to ensure that serverless framework is given right permissions to interact with AWS resources like DynomoDB.

» Monitoring and maintainance.
fter deployment Serverless framework provides servic
formation like deployed endpoints, API Key, log
reams.

Case study for sonarqube.

reating your own profile in Sonarqube for testing
roject quality. Use sonarqube to analyze your code.
Install java ide and analyze java code

Sonarqube is an open source platform used for
continous inspection of quality. It detects bug, code
smells and security vulnerabilities in project acro
programming languages.

» Profile Creation in Sonarqube
Quality profiles in sonarqube are essentials
configuration that define rules applied during code
analysis. Each project has a quality profile for
every supported language with default being
Sonar way profile comes built in for all languages
Custom profiles can be created by copying or
extending existing ones copying creates you can
activate or deactivate rules, prioritize certain
rules and configure parameter to profile to
specific projects

2) Using Sonarcloud to analyse github code:
Sonarcloud is a cloud based counter part of sonarqube that triggers directly with github, Bit Bucket, and github repositories. To get started with sonarcloud via github signup product page and connect your github organization or personal account. Once connect, sonarcloud mirrors your gith setup with each projet corresponding to the github repos:- After setting up organization choose subscription plan. Next input repositories into your sonarcloud organization where each github repo to a sonarcloud project. Define new cloud to focus on recent changes and choose between automatic analysis or CI based analysis. Automatic analysis happens directly in sonarcloud. while CI based analysis integrates with your build process once the analysis results can be viewed in both sonar cloud and github including security import issue.

3) Sonarlint in Java IDE:
Sonarlint is an IDE that performs on the fly code analysis as you write code. It helps developers in the developing environment such as Intellic idea or Eclipse. To set it up install the sonarlint plugin. configure the connection with sonarqube or Sonarcloud and select the project profile to analyse Java code in code quality, promoting clean & maintainable code from begining.

Analyzing python projects with Sonarqube.

Sonarqube supports Python test coverage reporting but it requires third party tool like coverage you need to enable and adjust your build process so that coverage tools runs before Sonnar scanner and ensures report file is saved in diff path. for set up, you can use TOX; and coverage to py to configure and run test. In your tox.ini: include configuration for py test and coverage to generate report in xml format. The build process can also be automated using github Actions which install dependencies runs test and invokes Sonarqube scan. Ensure report in XML format and place where scanner can access it.

Analyzing Node.js projects with SonarQube.

For nodejs project Sonarqube can analyse Java script and typescript code. Similar to the python setup you can configure Sonarqube to analyze nodejs project by installing the appropriate plugin and using sonar scanner to scan the projects. Sonarqube will check the code against industry standard rules and best practices. flagging issues related to securing vulnerabilities bugs and performances optimization

3) At a large organization your centralized operation team may get many repatabile Infrastru requests you can use form to build a self service infrastructure modes that lets product team manage their own infrastructure independently you can create and use Terraform modules that codify the standards for deploying & managing servers in your organization, allowing teams to efficiently deploy services

Ans Implementing a self service infrastructure model using Terraform can transform how large organizati manage their infrastrudure independently organizatio can enhance efficiency reduce benefits and ensure compliance with established needs.

- The need for self service infrastructure:

In large Organization, centralized operations teams often face an overwhelming number of repetitude requests. This can lead to deploy the service delivering and frustration among the product teams who need to move quality. A self service models allows teams to provision and manage their infracstructure without replying on the operations team for every request.

- Benefits of wing terraform.

1. Modularity & Reusability:

Terraform modules encapsulate standard configuration for various infrastructure components. Teams can reuse these modules across different projects reducing redundancy and minimizing the risks of errors.

Standarizations
By identifying and practices within modules organization can ensure that all deployment comply with internal policies and standards.
This conticiency helps maintain security and operational integrity accross the organization.

Increased Efficiency:
Product teams can deploy services quality by using predefined modules, significantly reducing the time spent on infrastructure setup.
This allows them to focus on developing features rather than mangging infrastructure.

Integration with ticketing systems.

Terraform cloud can integrate with ticketing system like service now. to automate the generation of infrastructure requests.
This integration streandines workflow by allowing teams to initiate request directly from their ticketing platform, reducing manual intervention

Implementation steps:

1. Identify infrastructure components
Begin by identifying which components of your infrastructure can be modularized.

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

3. Establish governance and Best Practices
- Define guidelines for module usage, versions and documentation to ensure clarity and maintainability.
- Encourage teams to contribute to module develop and share improvements

4. Testing and validation

- Implement a testing framework to validate module functionality before development.
- Best practices for module management
- Utilize the terraform register.
- leverage existing community modules from the Terraform register.