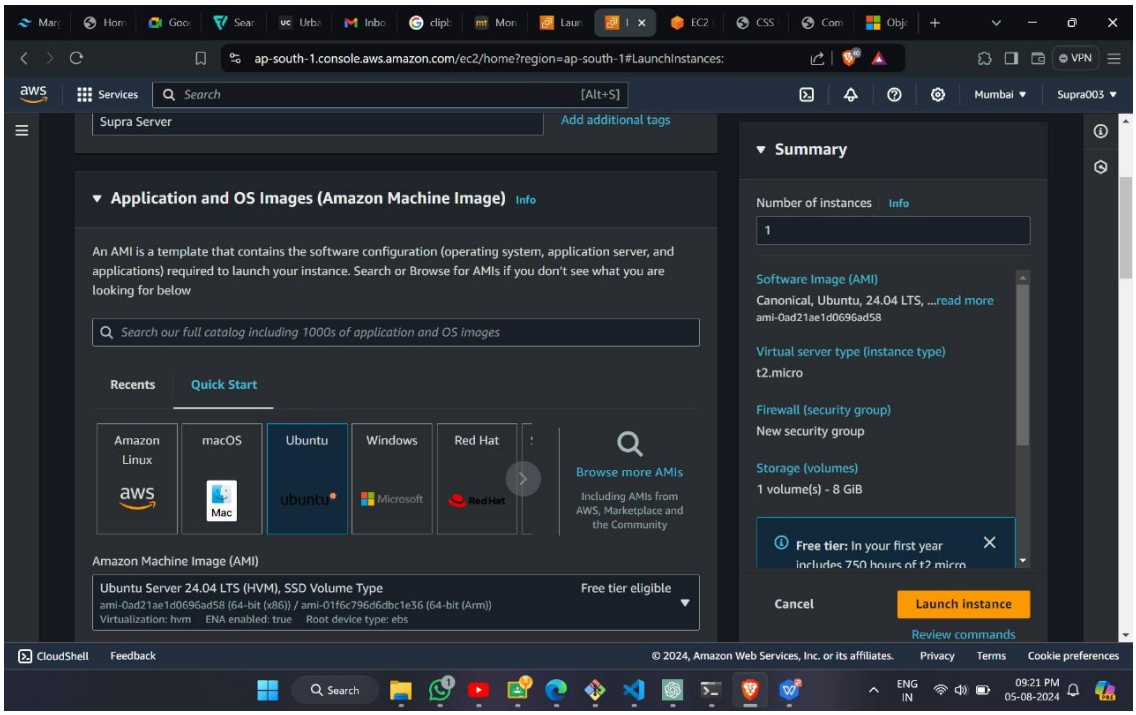
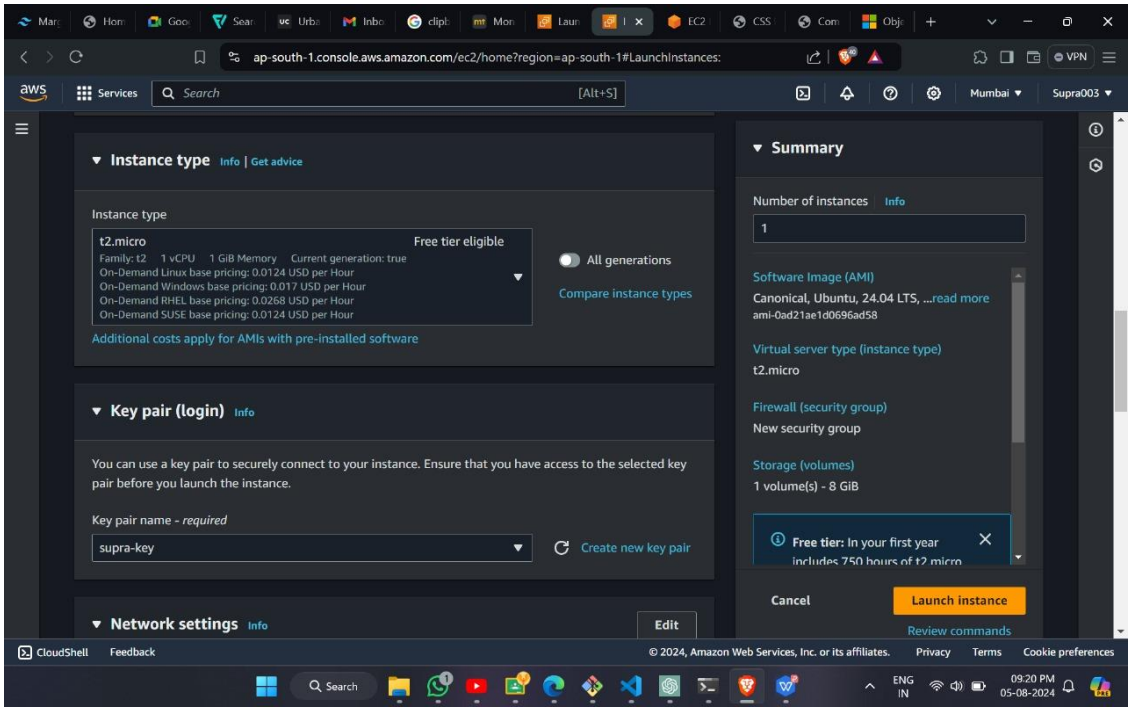
Experiment 01

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

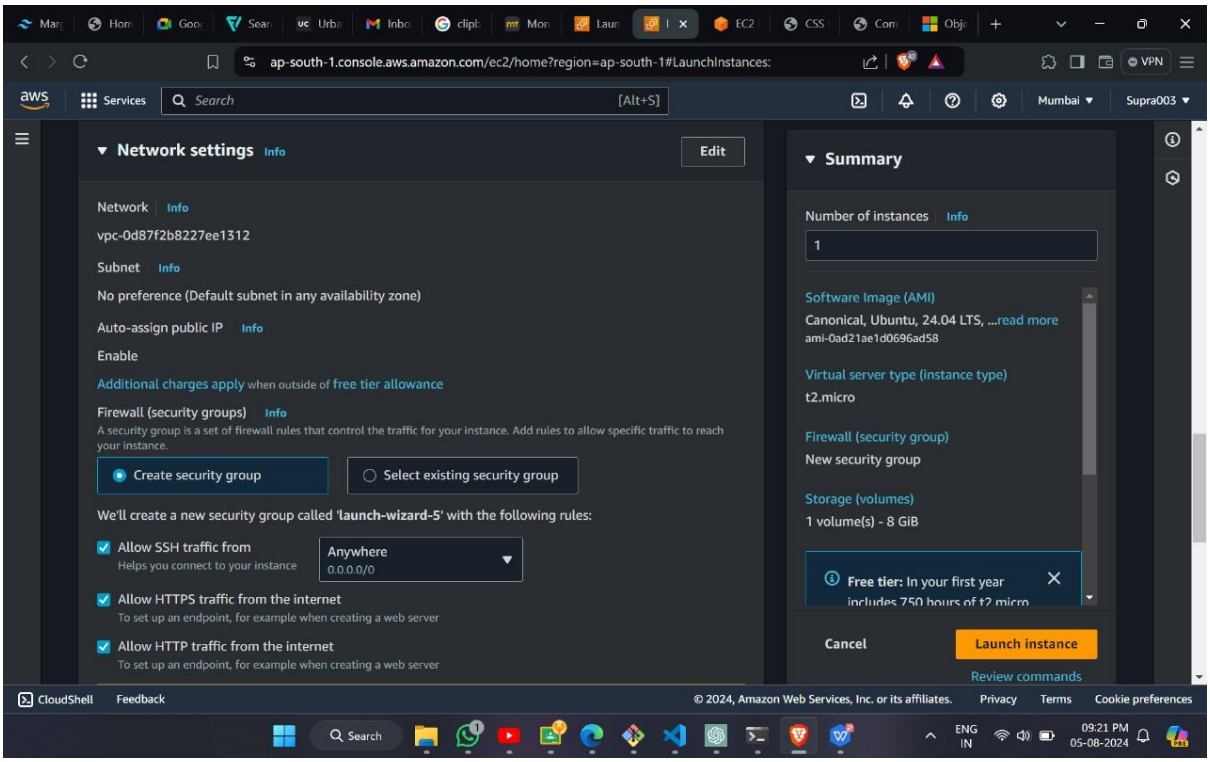
1. Give name to EC2 instance



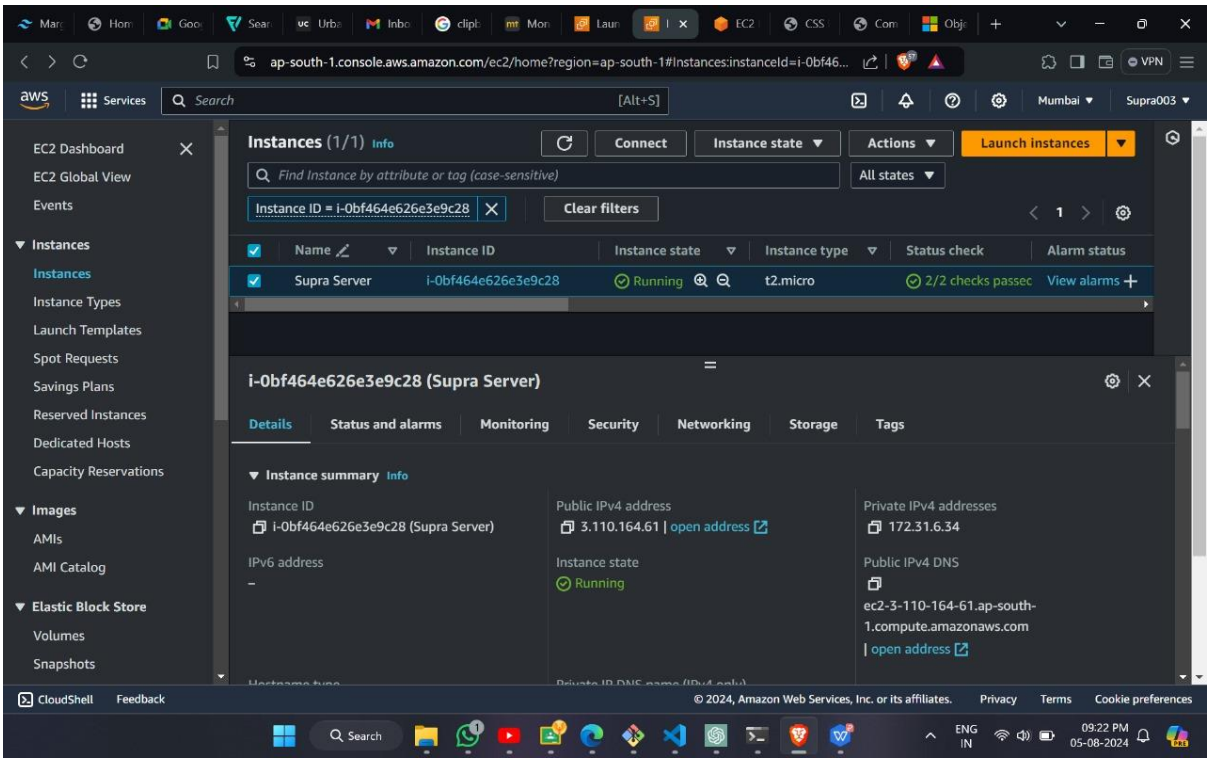
2. Create a key-pair login.



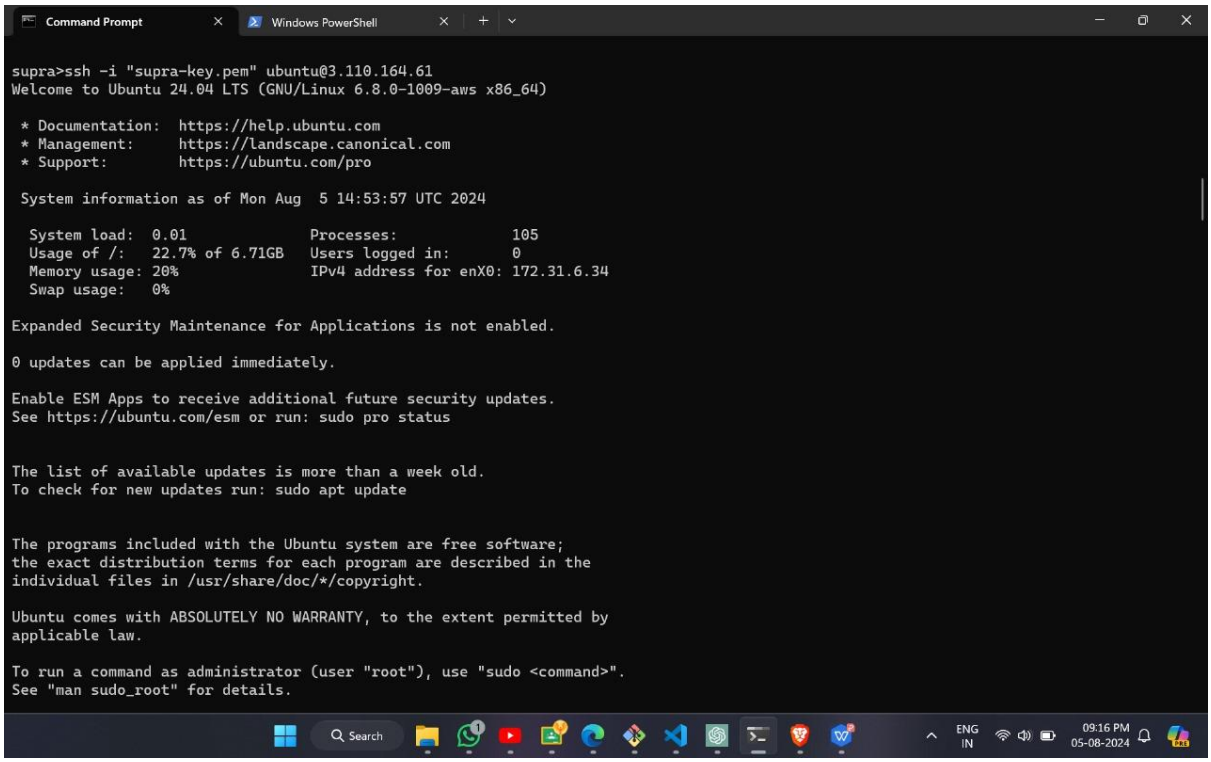
3. Edit network settings. And Launch Instance.



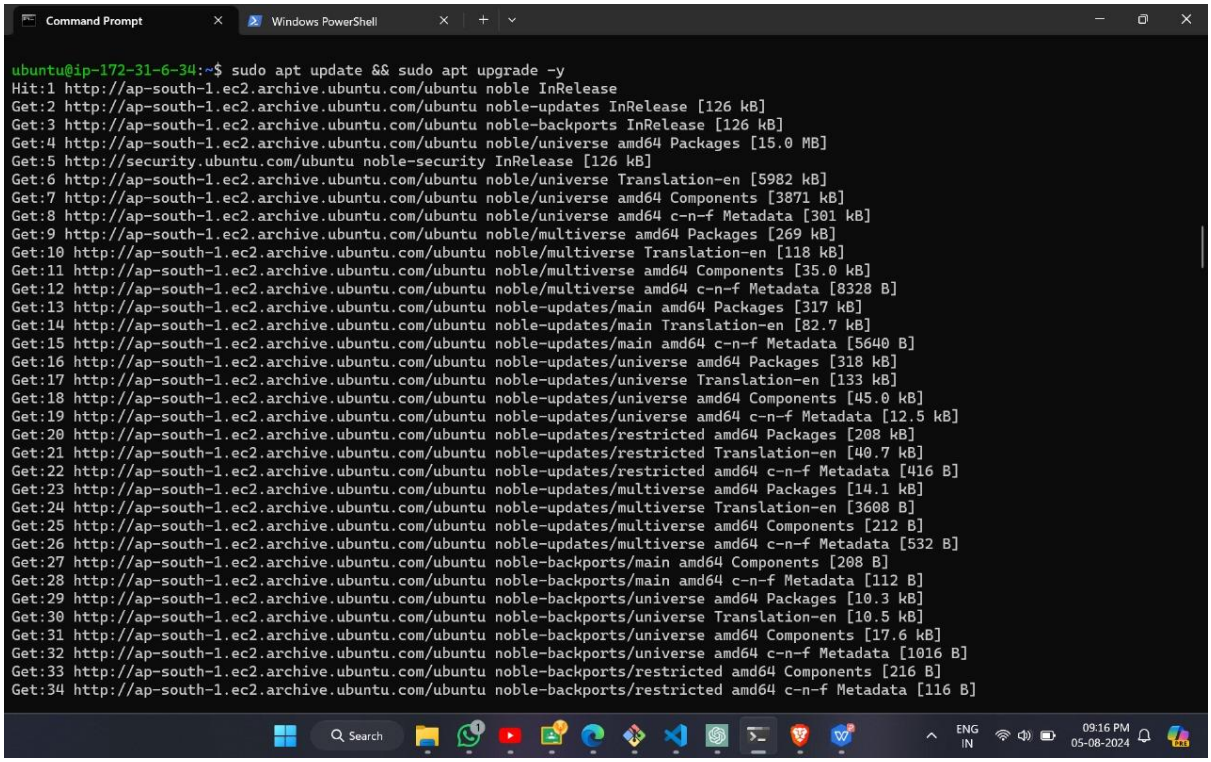
4. Instance successfully lanunched.



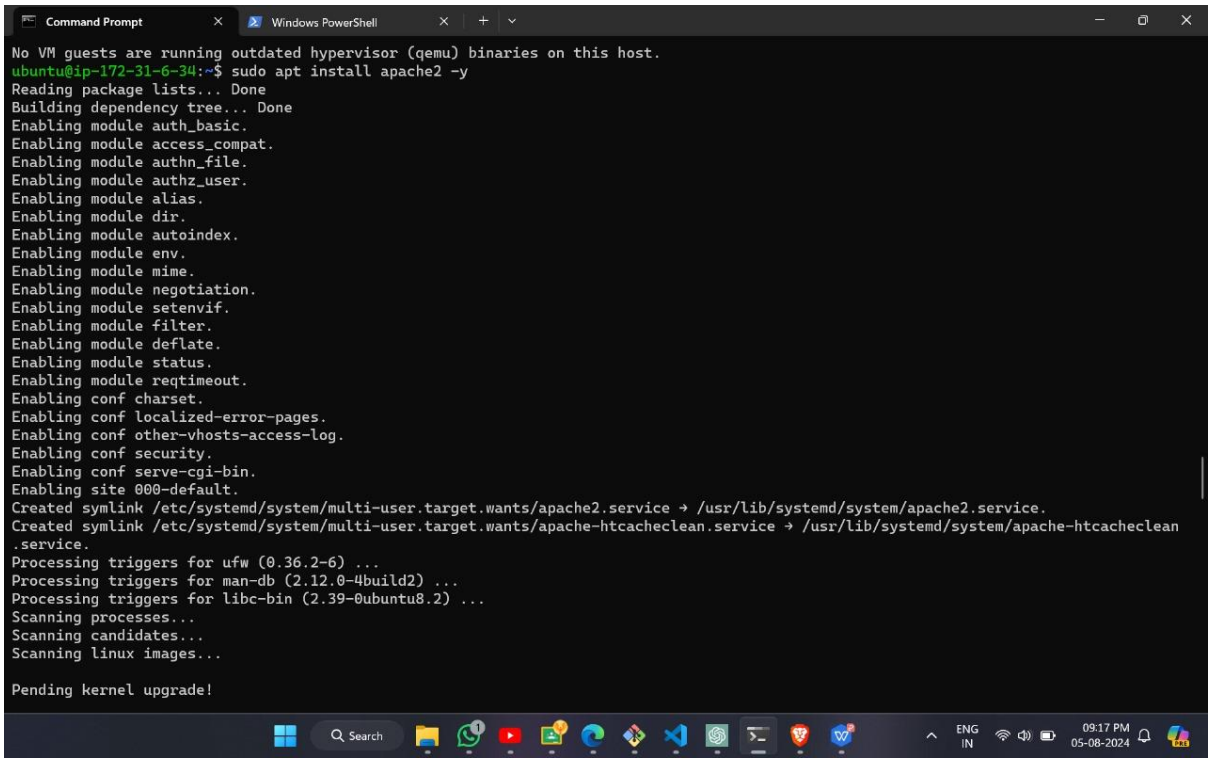
5. Create a virtual environment with key pair permission.



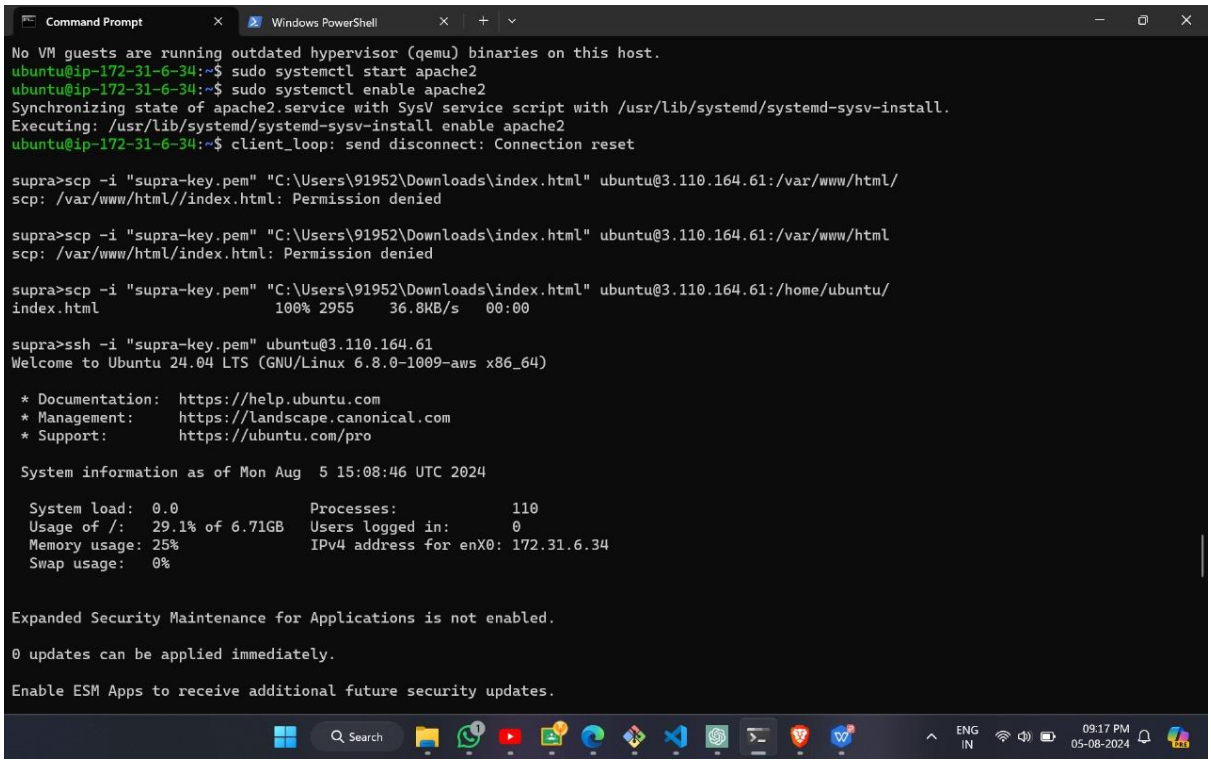
6. Install apache on virtual machine.



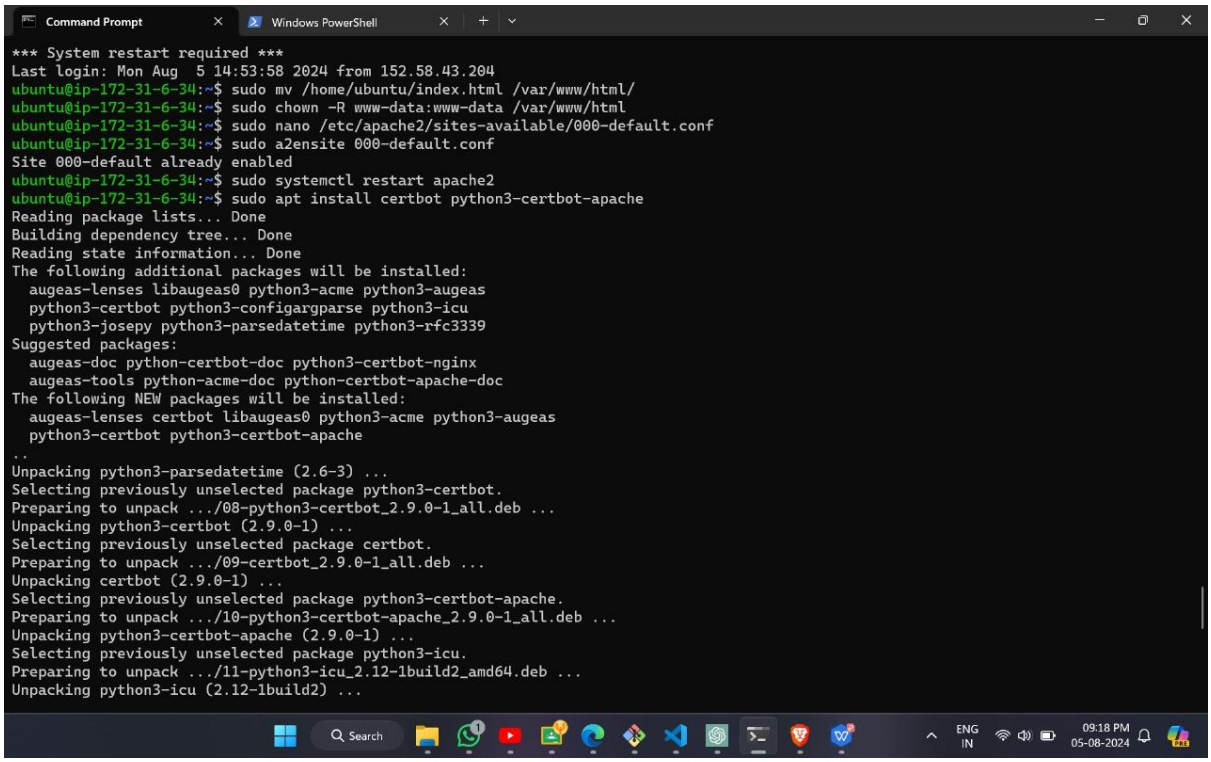
7. Update and upgrade it.



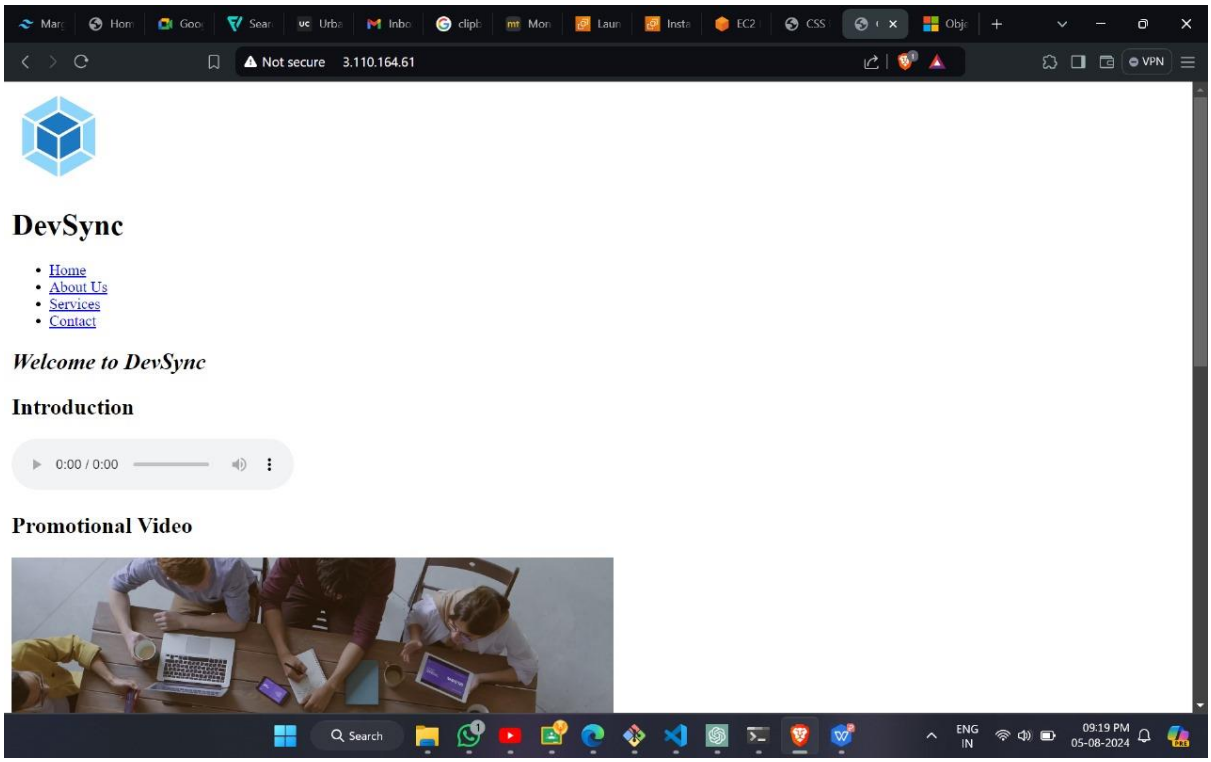
8. Start the apche server.



9. Give file location to be executed.

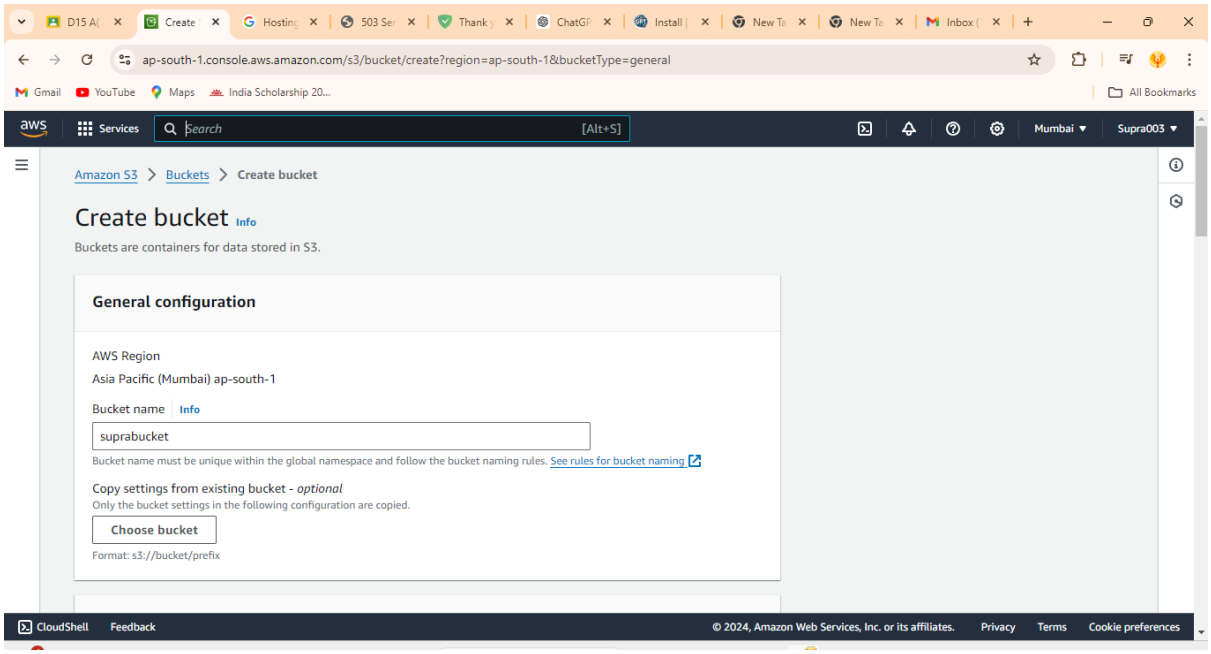


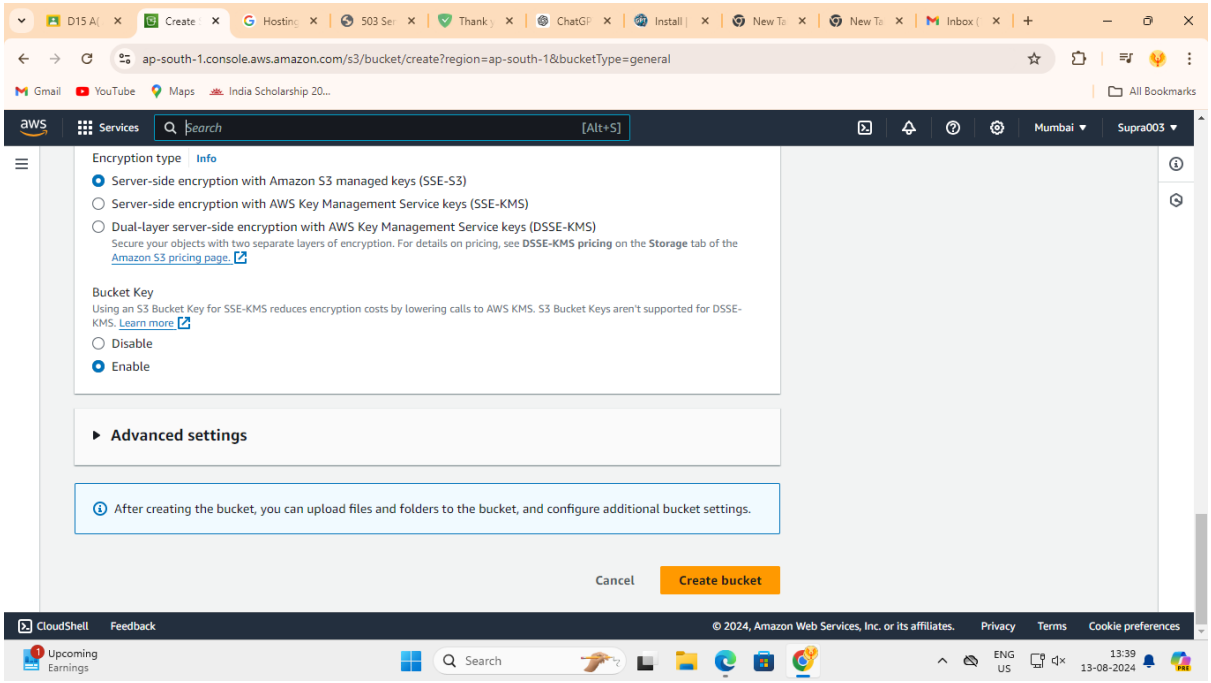
10. Output of the code on local machine.

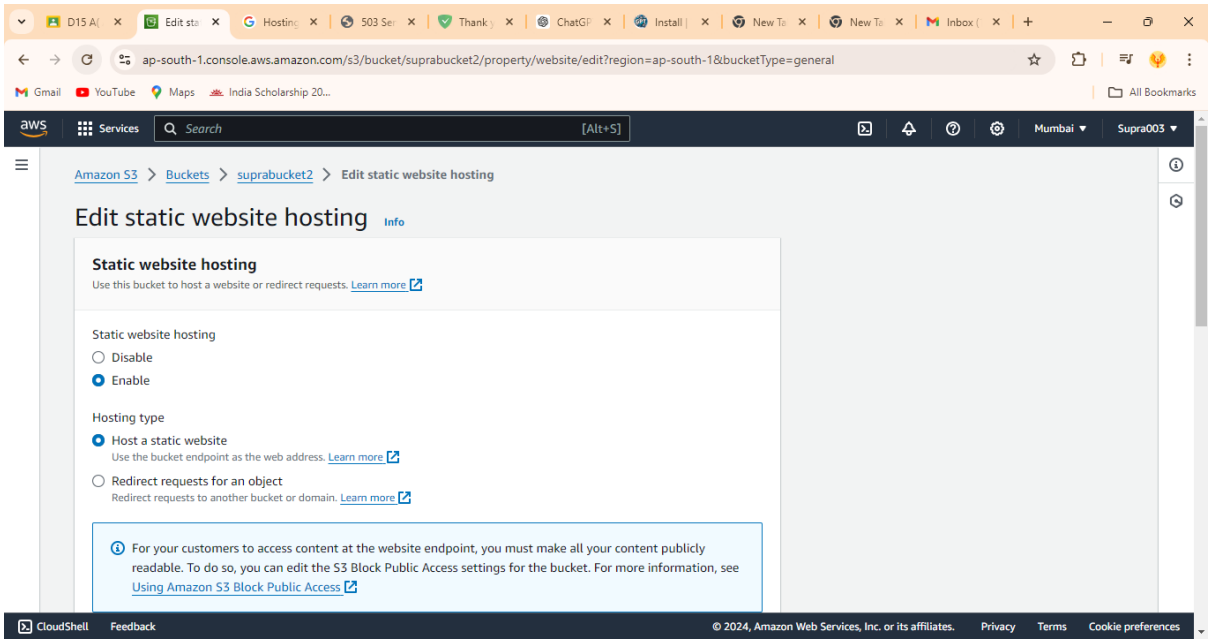


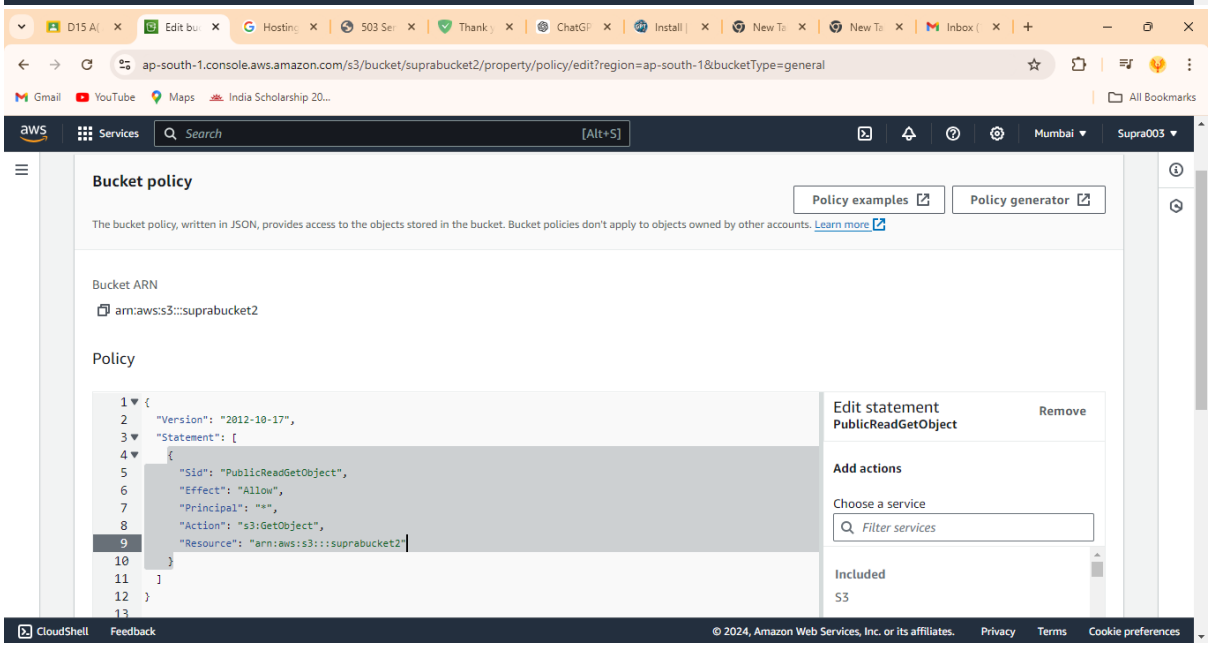
Experiment 01-(a)

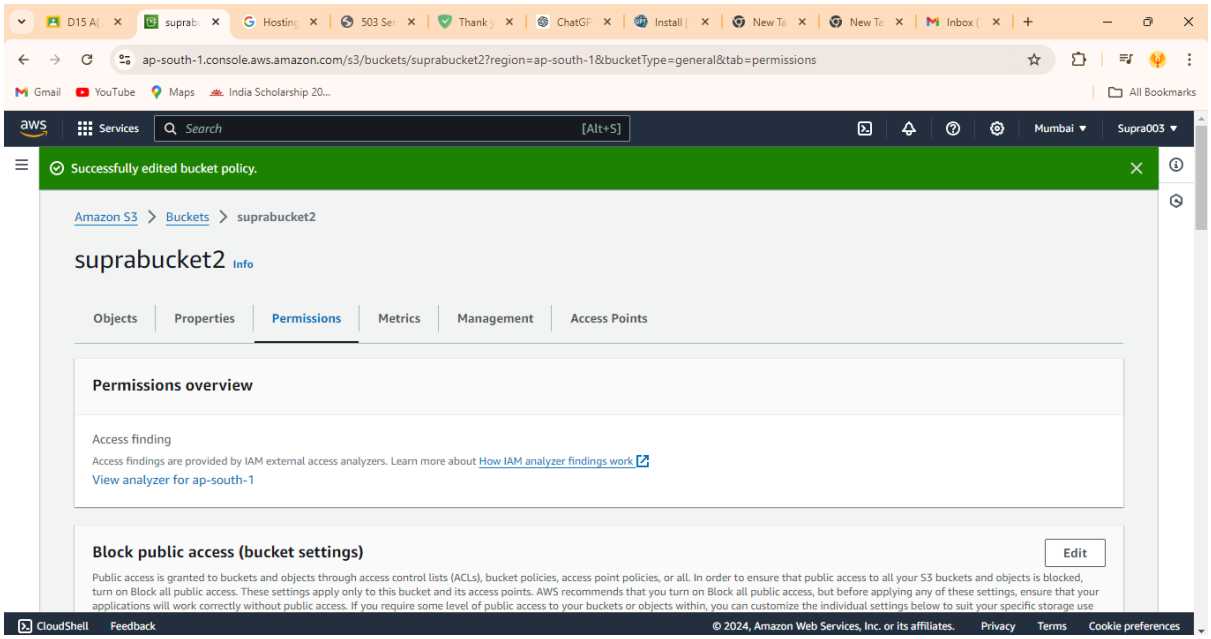
Hosting a static website on Amazon S3

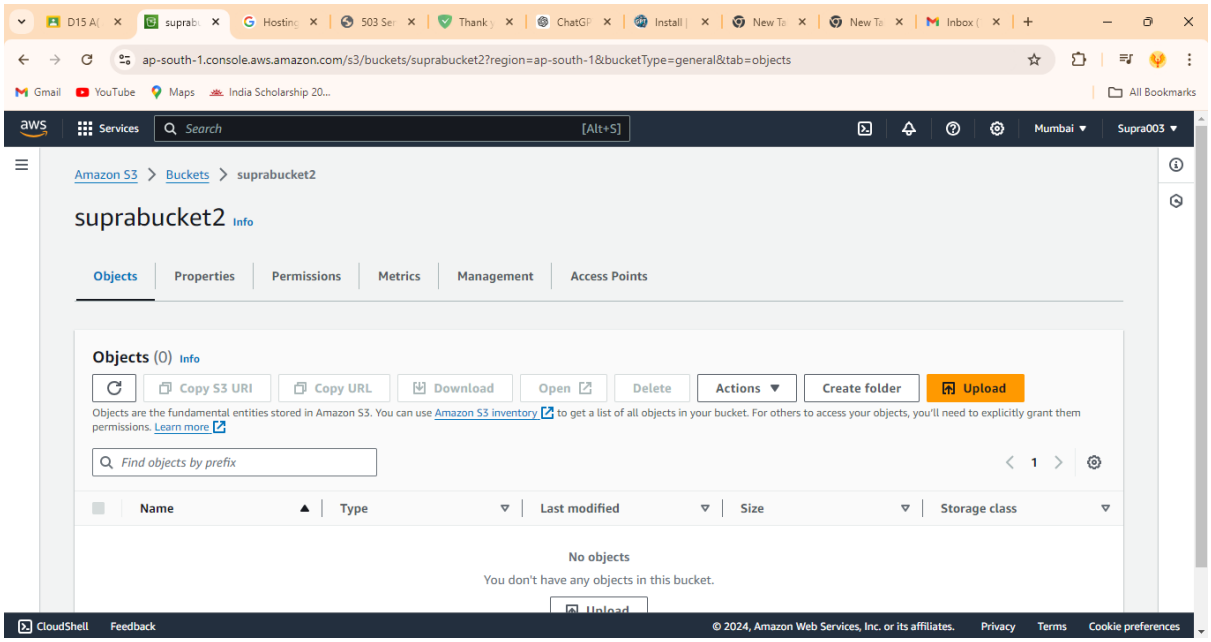


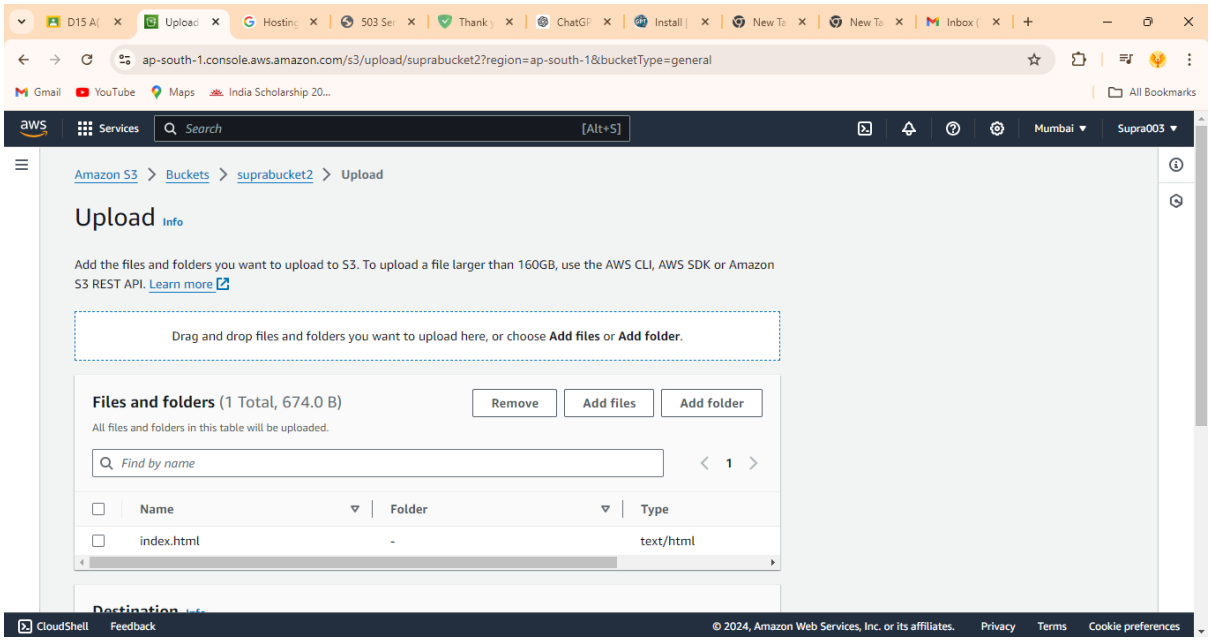


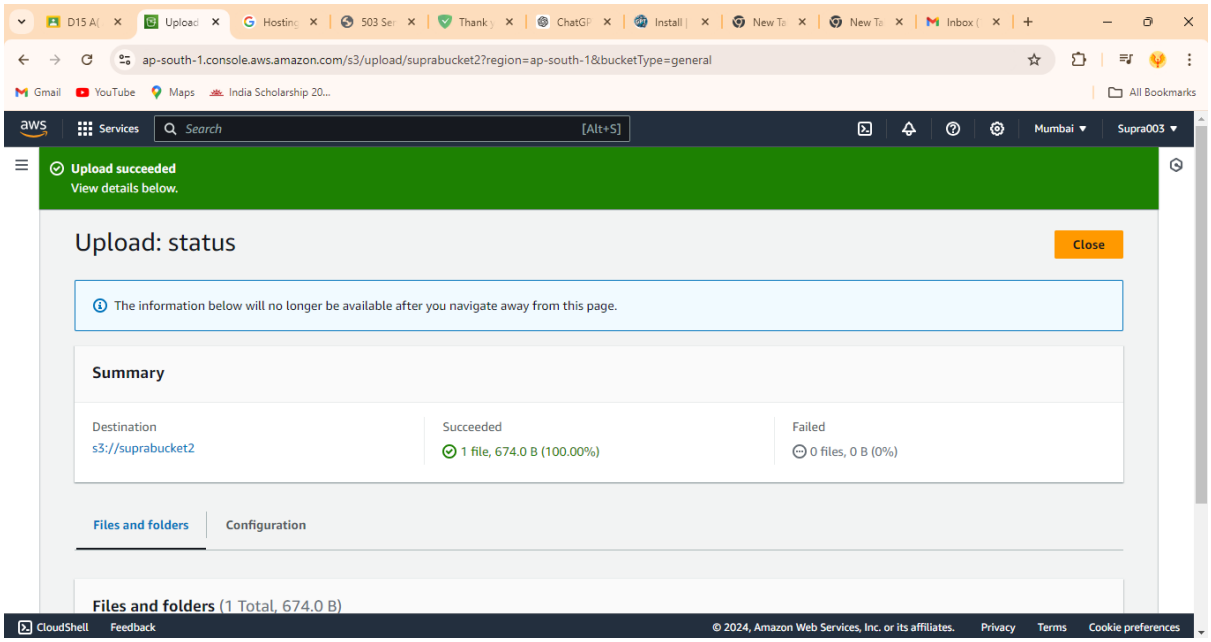

ap-south-1.console.aws.amazon.com/s3/bucket/suprabucket2/property/website/edit?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Amazon S3 > Buckets > suprabucket2 > Edit static website hosting
Edit static website hosting
Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

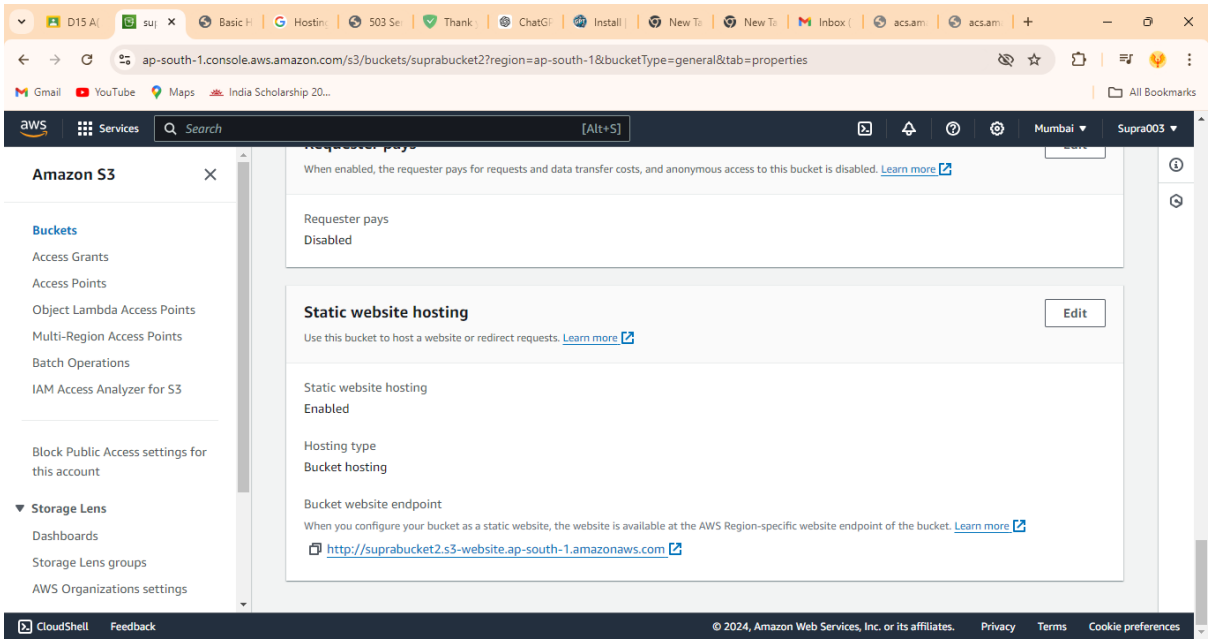

ap-south-1.console.aws.amazon.com/s3/bucket/suprabucket2/property/policy/edit?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Bucket policy
Policy examples
Policy generator
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
Bucket ARN
arn:aws:s3:::suprabucket2
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "PublicReadGetObject",
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::suprabucket2"
}
]
}
Edit statement
PublicReadGetObject
Remove
Add actions
Choose a service
Filter services
Included
S3
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

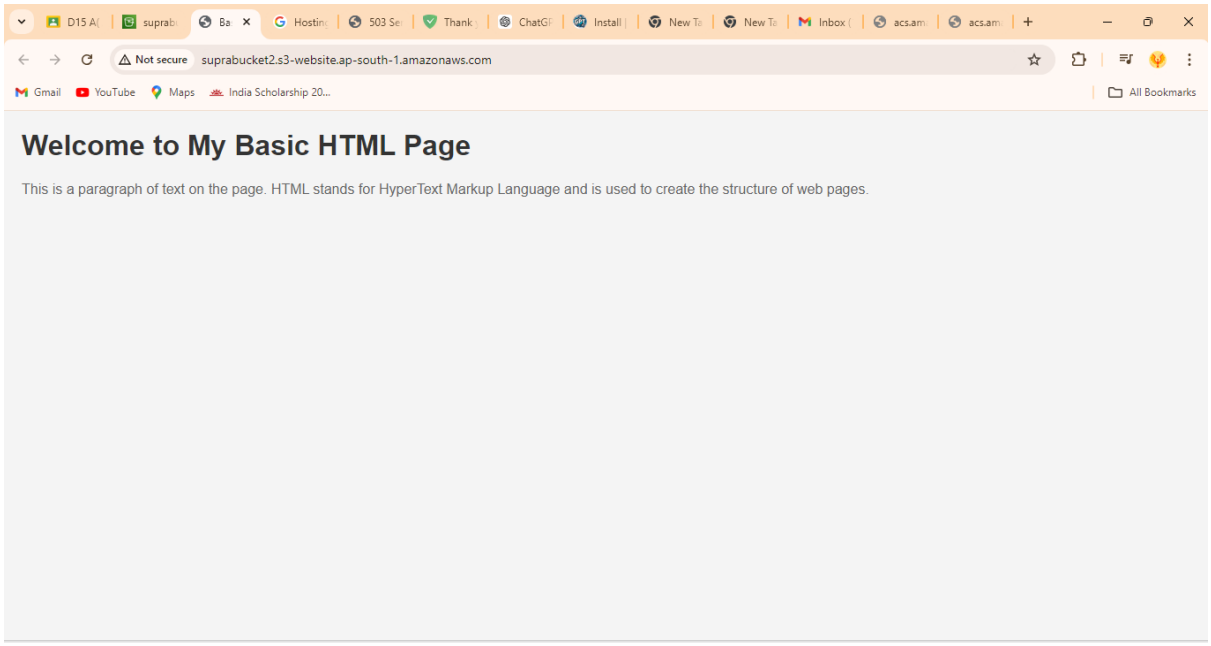

ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=permissions
Successfully edited bucket policy.
Amazon S3 > Buckets > suprabucket2
suprabucket2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for ap-south-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=objects
Amazon S3 > Buckets > suprabucket2
suprabucket2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/upload/suprabucket2?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Amazon S3 > Buckets > suprabucket2 > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 674.0 B)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
Name
Folder
Type
index.html
-
text/html
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/upload/suprabucket2?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
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://suprabucket2
Succeeded
1 file, 674.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 674.0 B)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=properties
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more
Requester pays
Disabled
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Edit
Static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://suprabucket2.s3-website.ap-south-1.amazonaws.com
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
suprabucket2.s3-website.ap-south-1.amazonaws.com
Welcome to My Basic HTML Page
This is a paragraph of text on the page. HTML stands for HyperText Markup Language and is used to create the structure of web pages.

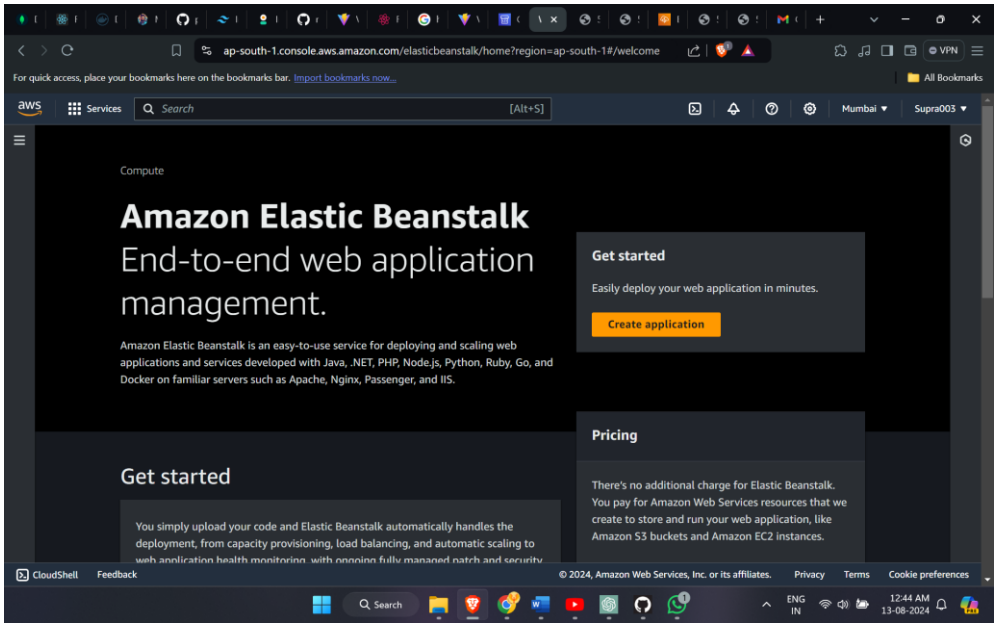
Experiment 02

Name: Prajyot Shinde **Roll no: 57**

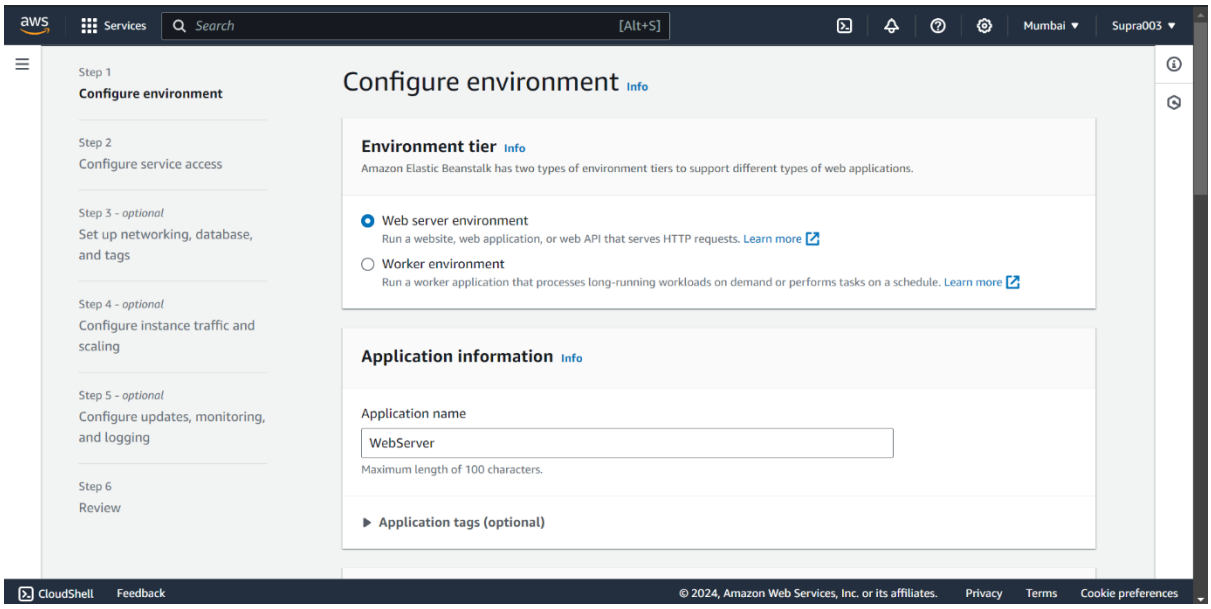
Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Step-1: Beanstalk Deployment

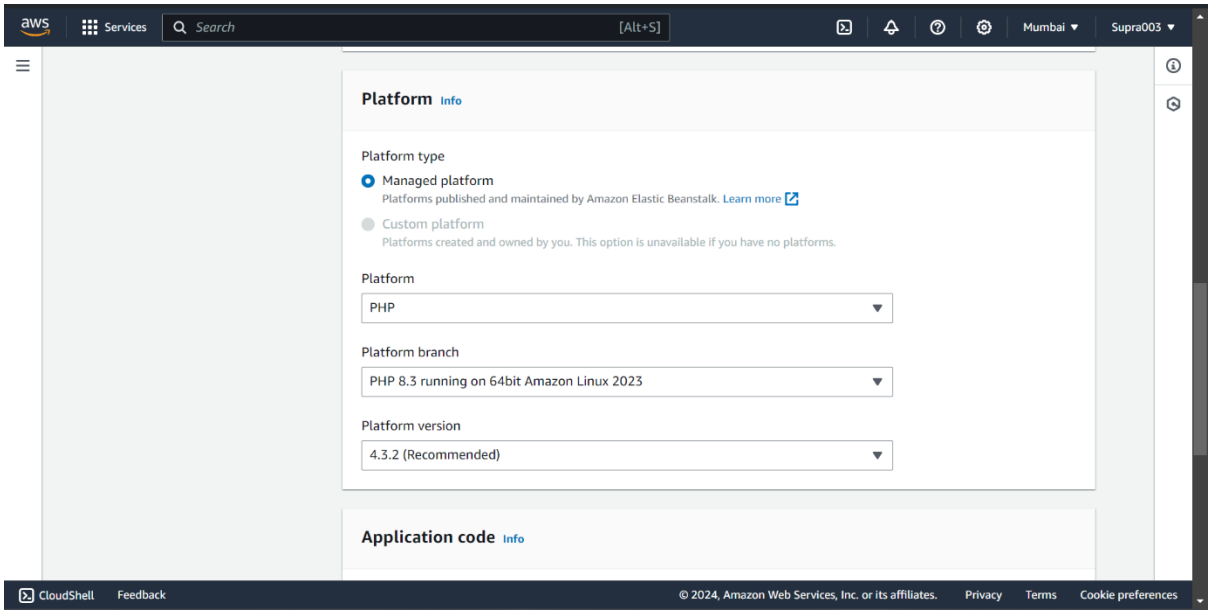
1. Login to your AWS account and search for Elastic Beanstalk in the search box



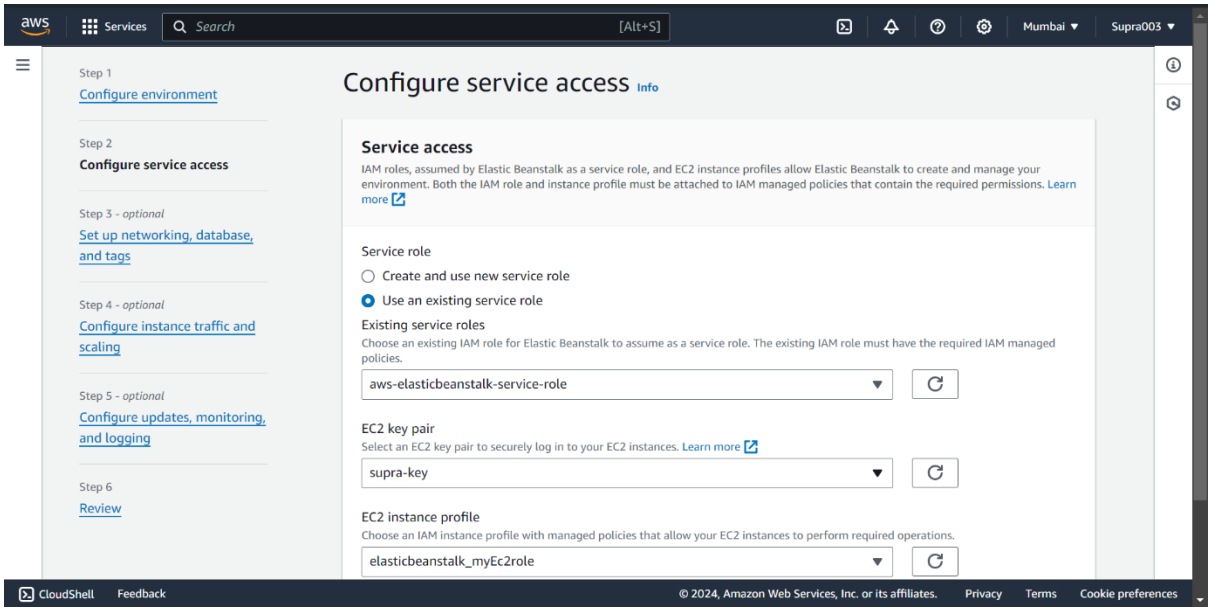
2. Open up Elastic Beanstalk and name your web app.

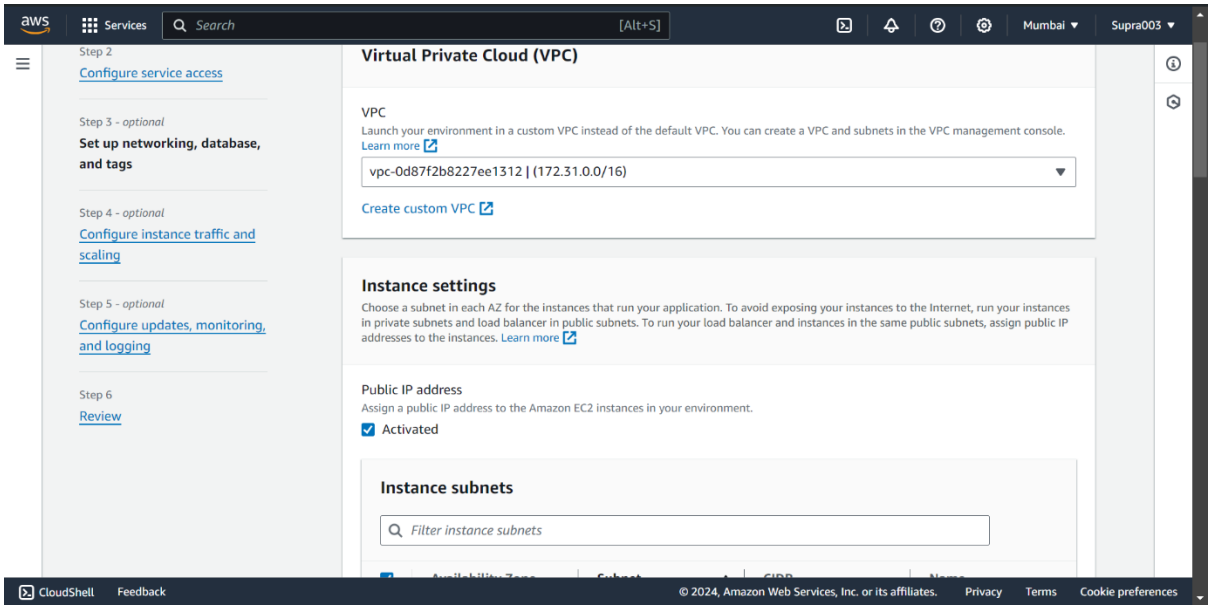


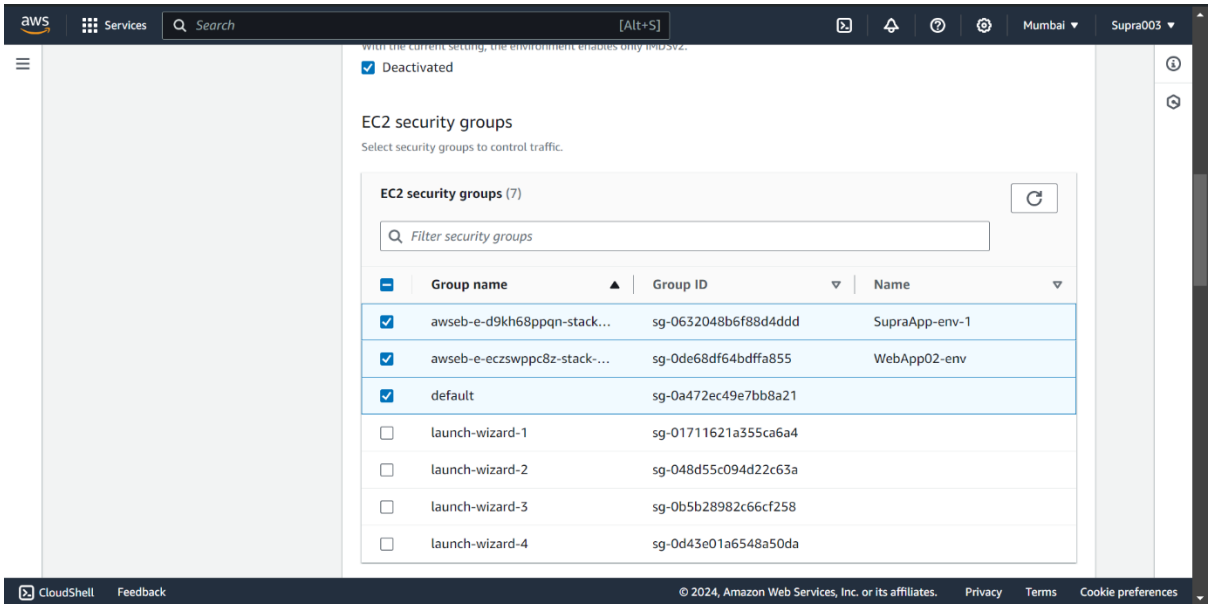
3. Choose Python from the drop-down menu and then click Create Application.

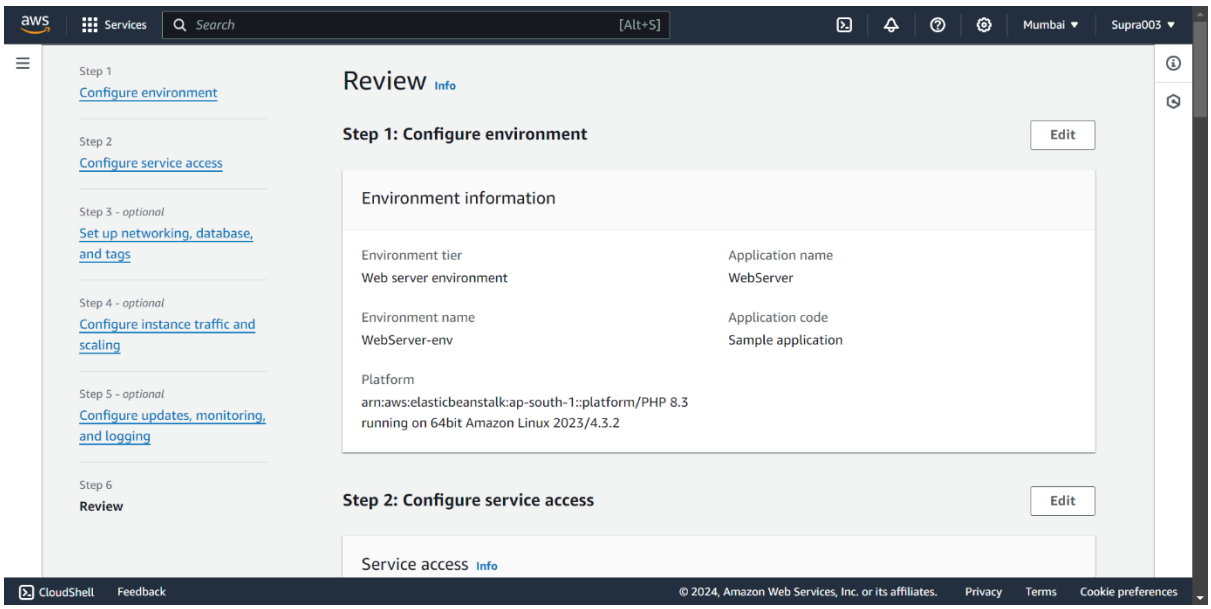


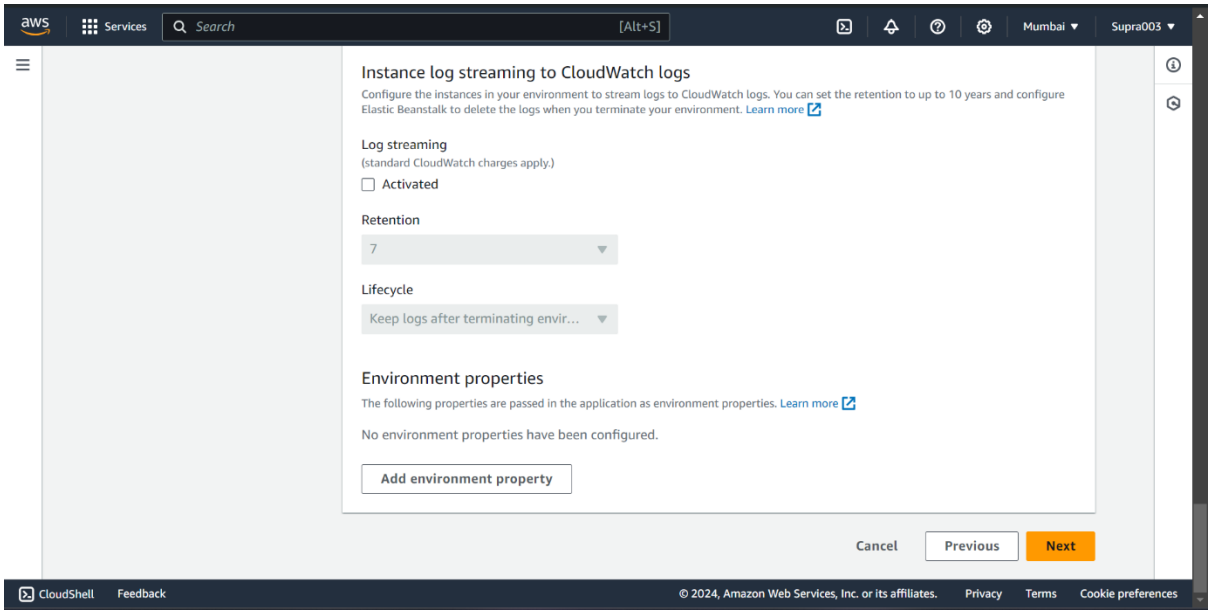
4. Choose the proper options

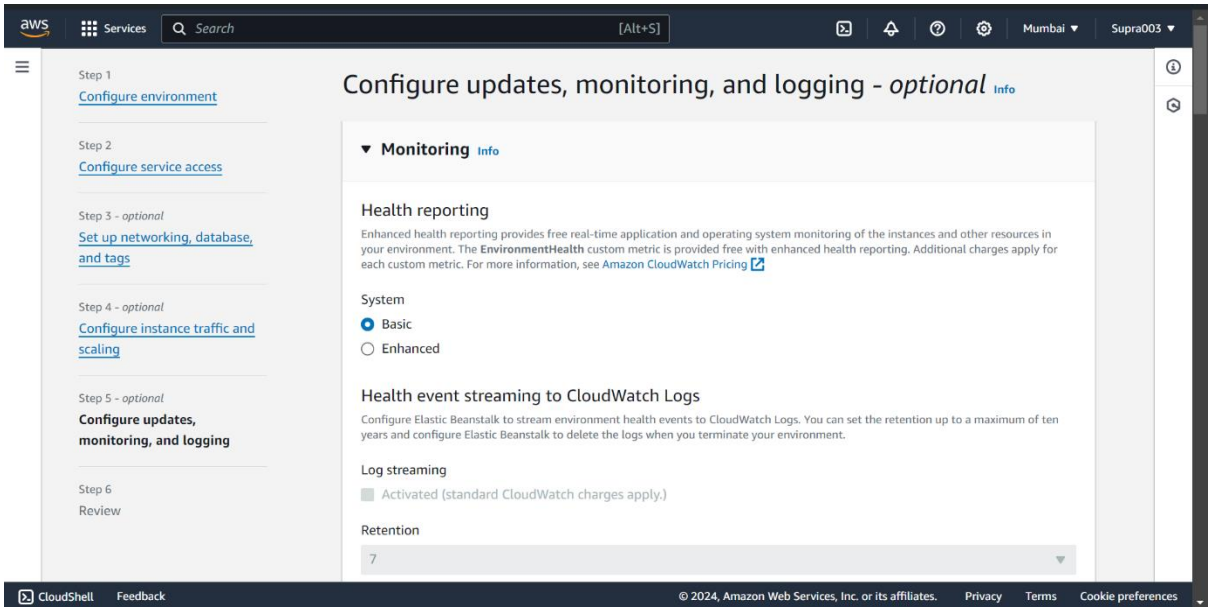



Services
Search
[Alt+S]
Mumbai
Supra003
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Virtual Private Cloud (VPC)
VPC
Launch your 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-0d87f2b8227ee1312 | (172.31.0.0/16)
Create custom VPC
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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Mumbai
Supra003
Deactivated
EC2 security groups
Select security groups to control traffic.
EC2 security groups (7)
Filter security groups
Group name
Group ID
Name
awseb-e-d9kh68ppqn-stack...
sg-0632048b6f88d4ddd
SupraApp-env-1
awseb-e-eczswppc8z-stack-...
sg-0de68df64bdffa855
WebApp02-env
default
sg-0a472ec49e7bb8a21
launch-wizard-1
sg-01711621a355ca6a4
launch-wizard-2
sg-048d55c094d22c63a
launch-wizard-3
sg-0b5b28982c66cf258
launch-wizard-4
sg-0d43e01a6548a50da
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Mumbai
Supra003
Step 1
Configure environment
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Review
Info
Step 1: Configure environment
Edit
Environment information
Environment tier
Web server environment
Application name
WebServer
Environment name
WebServer-env
Application code
Sample application
Platform
arn:aws:elasticbeanstalk:ap-south-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2
Step 2: Configure service access
Edit
Service access
Info
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

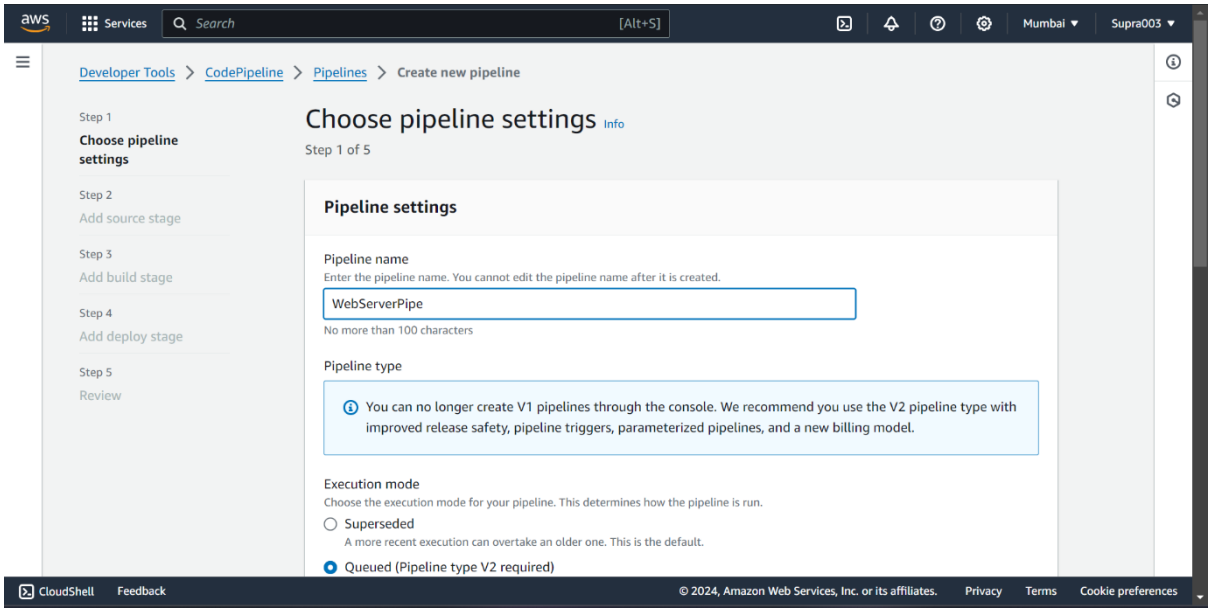

aws
Services
Search
[Alt+S]
Mumbai
Supra003
Instance log streaming to CloudWatch logs
Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn 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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Mumbai
Supra003
Step 1
Configure environment
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

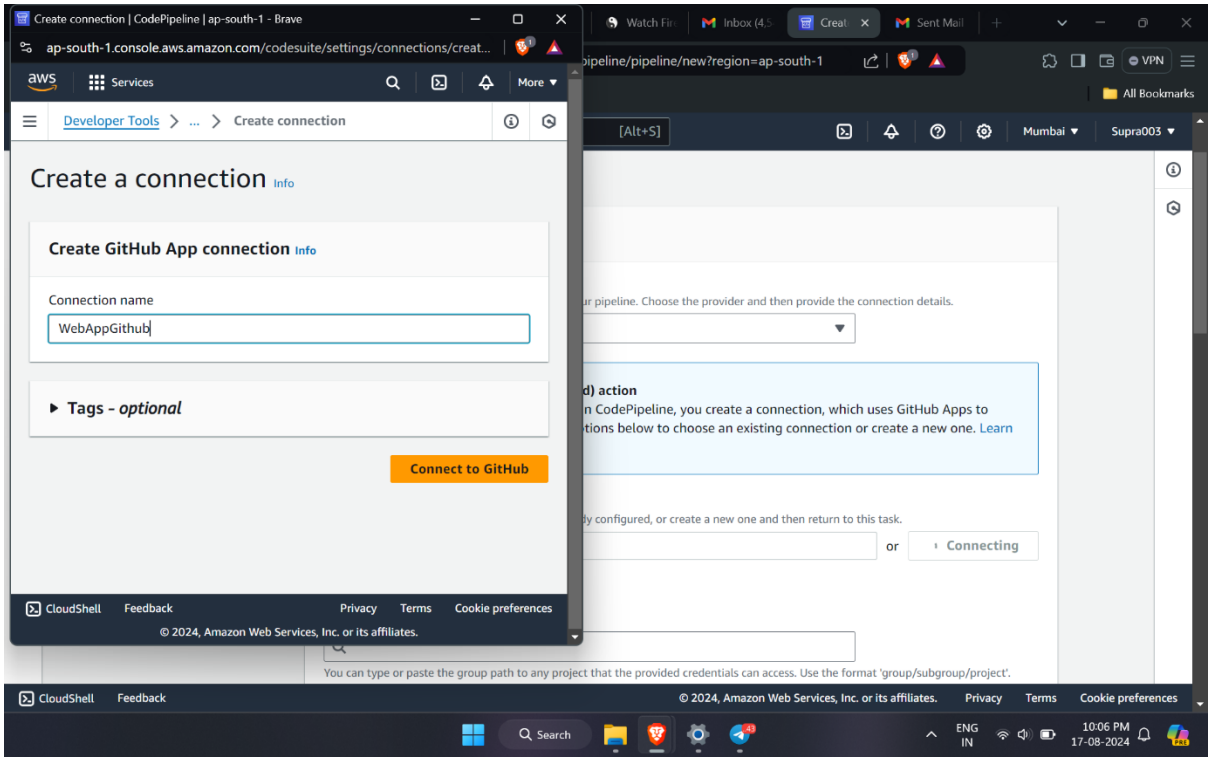
Step-2: Creating CodePipeline

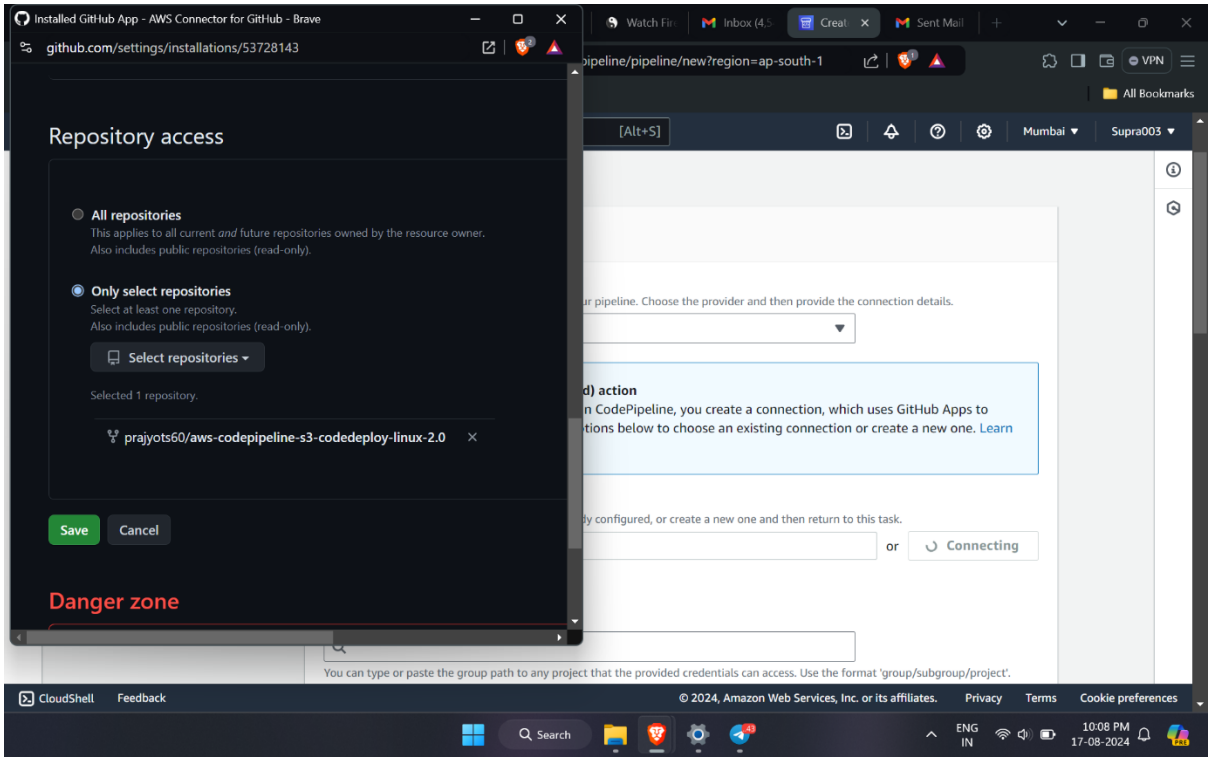
Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

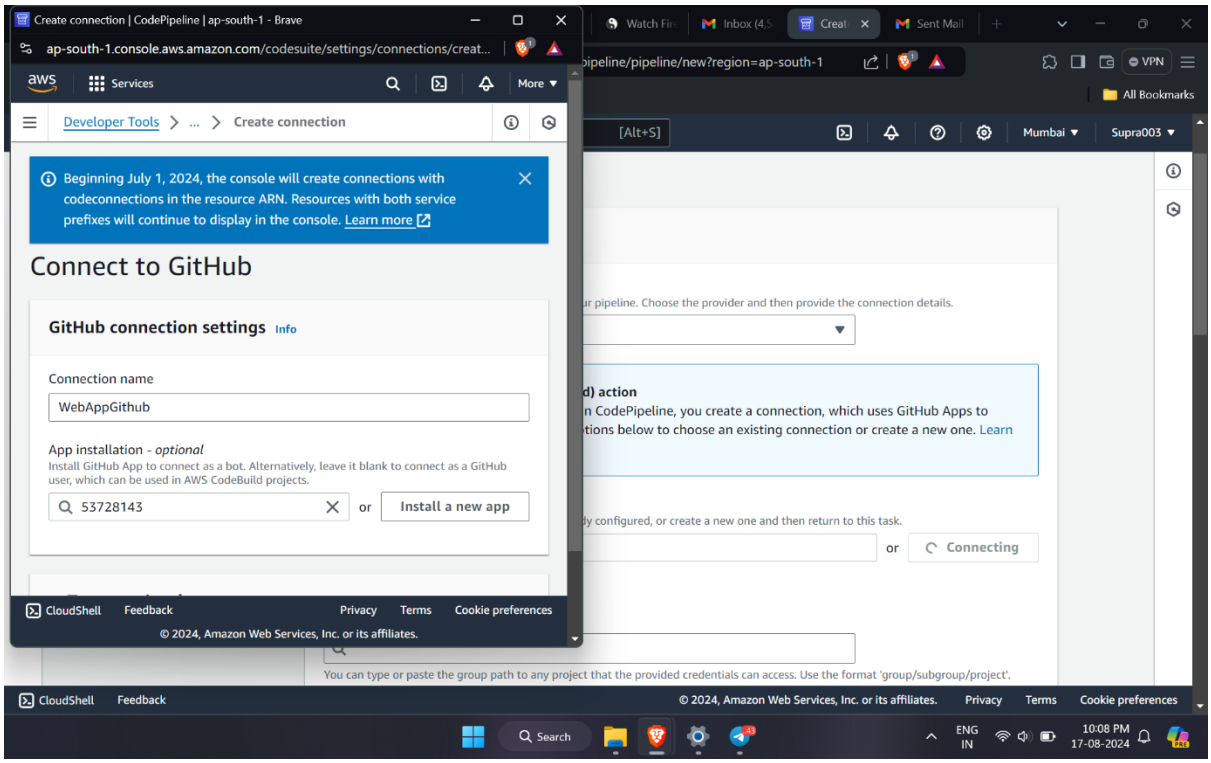
Create a CodePipeline and give pipeline name

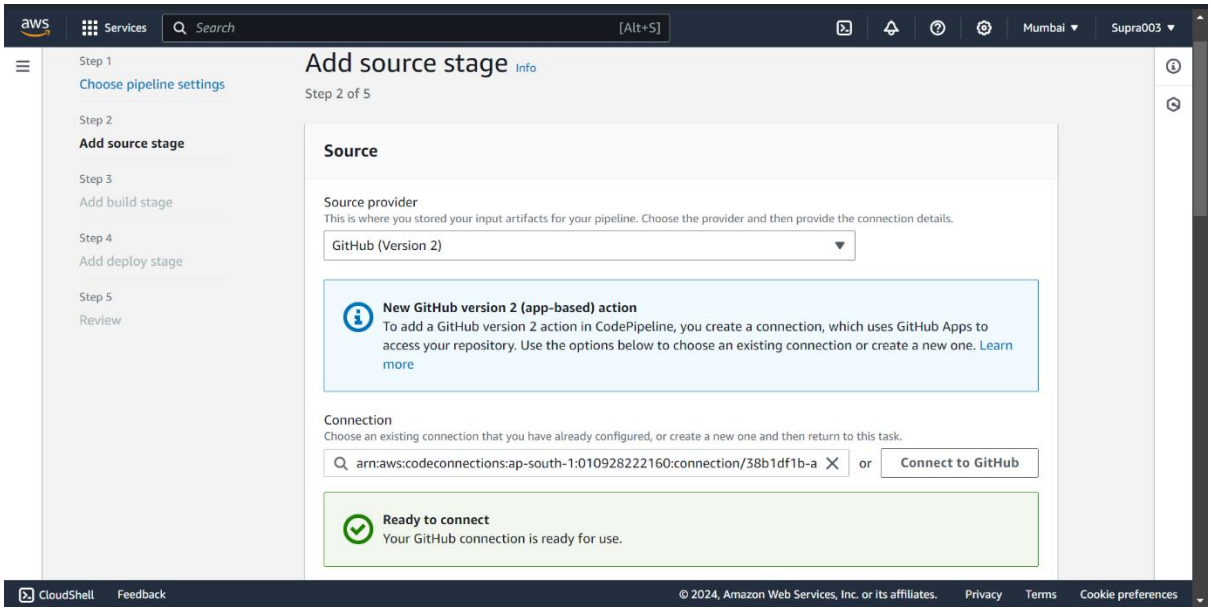


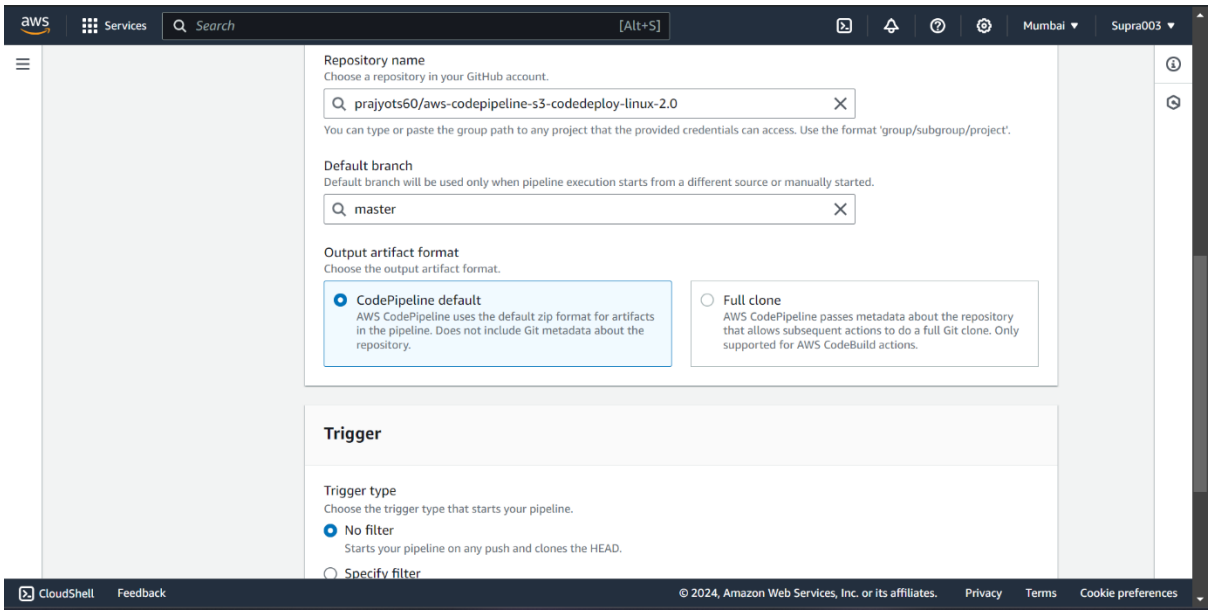
Create a Github connection



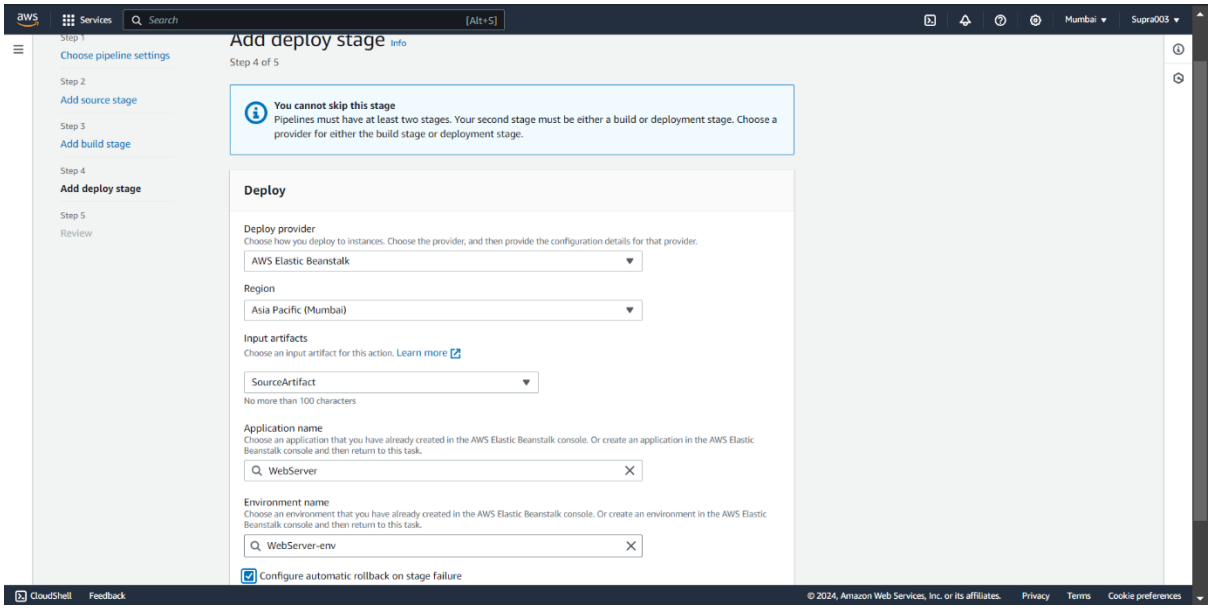

Installed GitHub App - AWS Connector for GitHub - Brave
github.com/settings/installations/53728143
Repository access
All repositories
This applies to all current and future repositories owned by the resource owner.
Also includes public repositories (read-only).
Only select repositories
Select at least one repository.
Also includes public repositories (read-only).
Select repositories
Selected 1 repository.
prajyots60/aws-codepipeline-s3-codedeploy-linux-2.0
Save
Cancel
Danger zone
pipeline/pipeline/new?region=ap-south-1
[Alt+S]
Mumbai
Supra003
ur pipeline. Choose the provider and then provide the connection details.
d) action
n CodePipeline, you create a connection, which uses GitHub Apps to
tions below to choose an existing connection or create a new one. Learn
dy configured, or create a new one and then return to this task.
or
Connecting
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
10:08 PM
17-08-2024

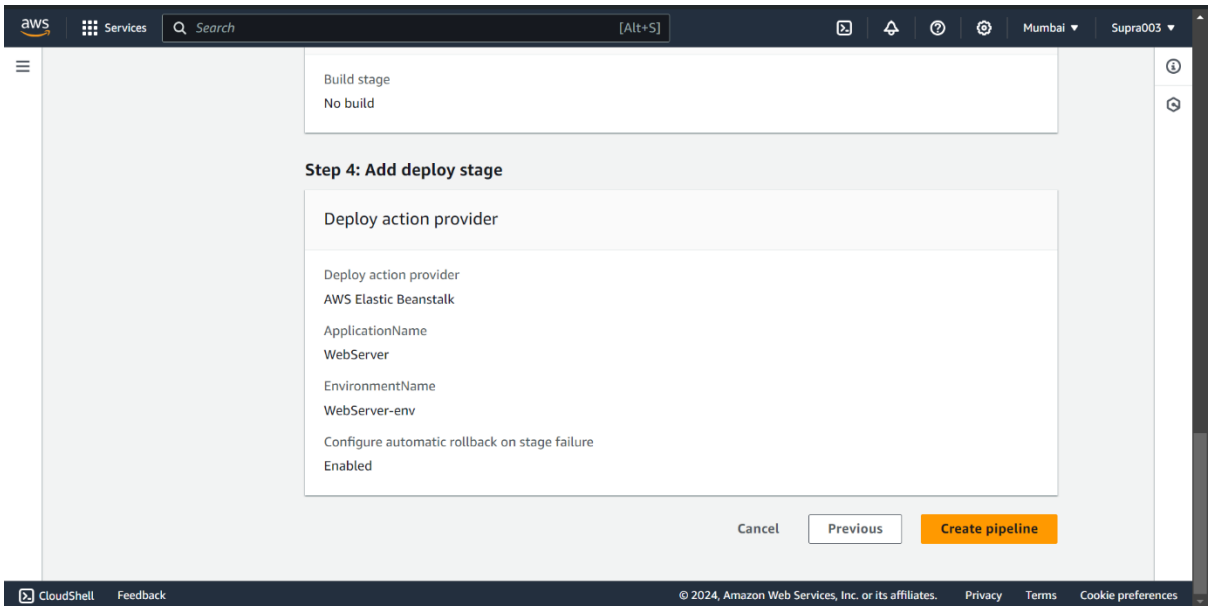

Create connection | CodePipeline | ap-south-1 - Brave
ap-south-1.console.aws.amazon.com/codesuite/settings/connections/creat...
aws
Services
More
Developer Tools
...
Create connection
Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. Learn more
Connect to GitHub
GitHub connection settings
Info
Connection name
WebAppGithub
App installation - optional
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.
53728143
or
Install a new app
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.
pipeline/pipeline/new?region=ap-south-1
[Alt+S]
Mumbai
Supra003
ur pipeline. Choose the provider and then provide the connection details.
d) action
n CodePipeline, you create a connection, which uses GitHub Apps to
tions below to choose an existing connection or create a new one. Learn
dy configured, or create a new one and then return to this task.
or
Connecting
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
10:08 PM
17-08-2024


aws
Services
Search
[Alt+S]
Mumbai
Supra003
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage Info
Step 2 of 5
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 2)
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more
Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.
arn:aws:codeconnections:ap-south-1:010928222160:connection/38b1df1b-a
or
Connect to GitHub
Ready to connect
Your GitHub connection is ready for use.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

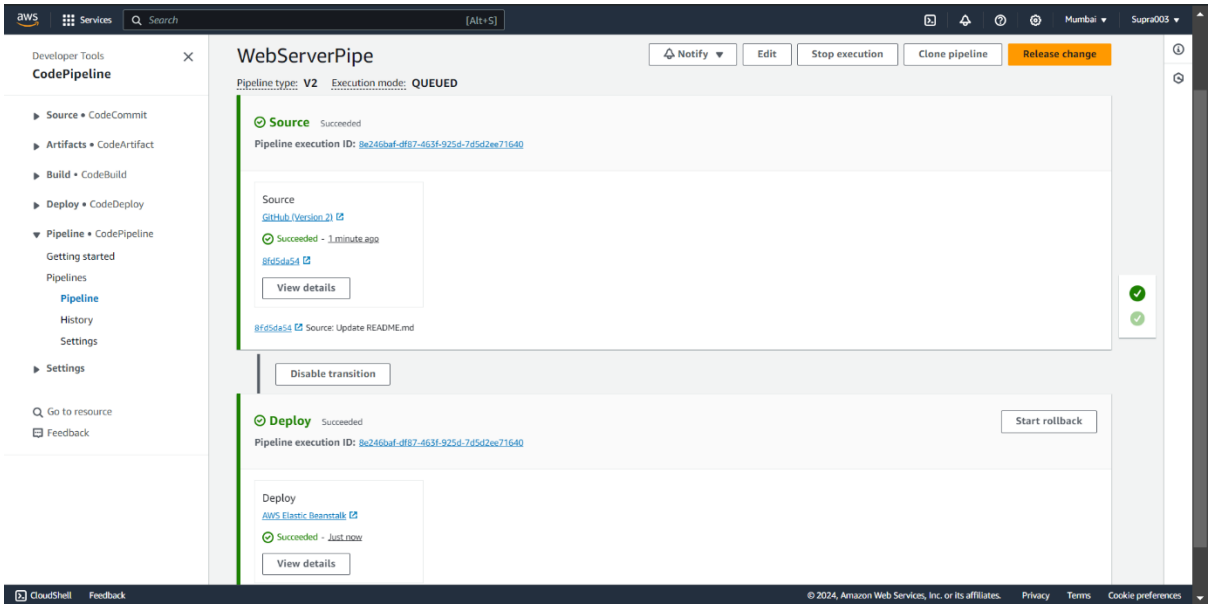

aws
Services
Search
[Alt+S]
Mumbai
Supra003
Repository name
Choose a repository in your GitHub account.
prajyots60/aws-codepipeline-s3-codedeploy-linux-2.0
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.
Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.
master
Output artifact format
Choose the output artifact format.
CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.
Trigger
Trigger type
Choose the trigger type that starts your pipeline.
No filter
Starts your pipeline on any push and clones the HEAD.
Specify filter
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

For deploy choose elastic beanstalk

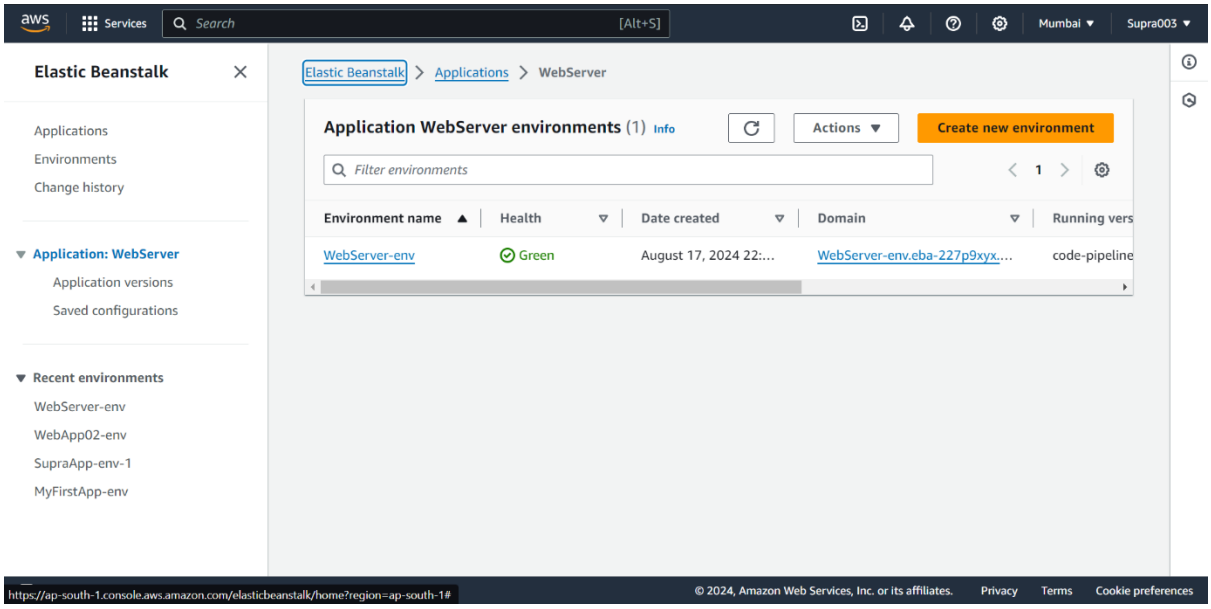


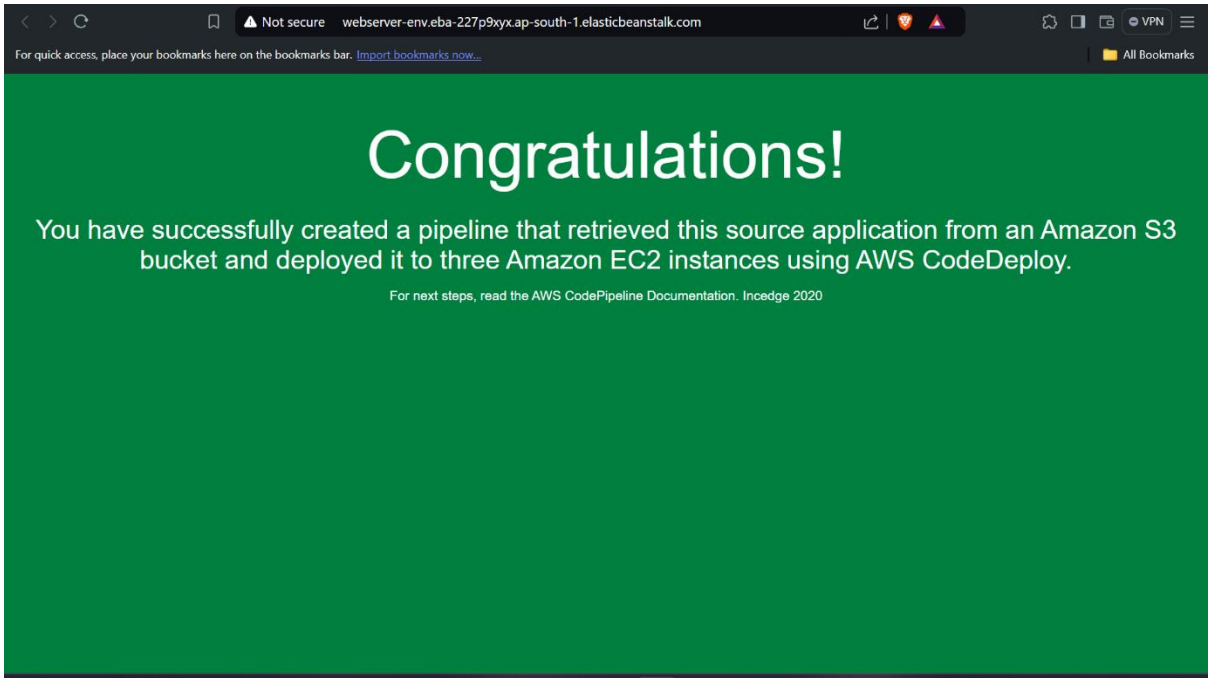


Make connection between code pipeline and beanstalk. And Green means successfully connected.



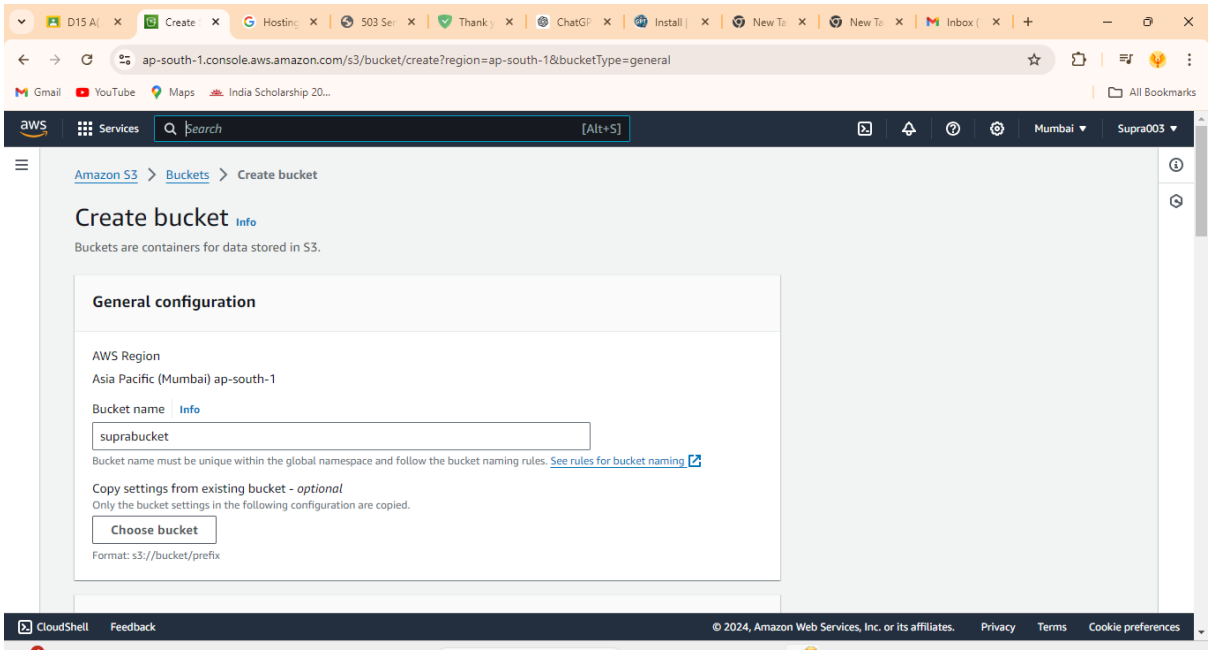
Click on domain to view you hosting site

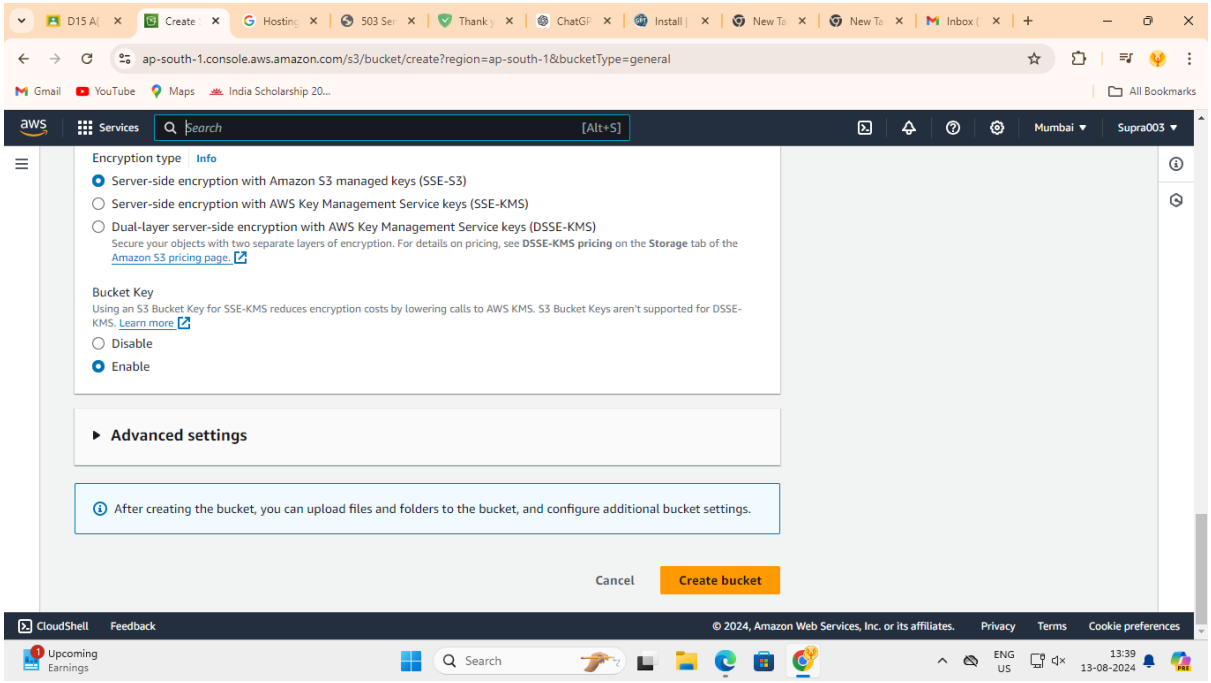


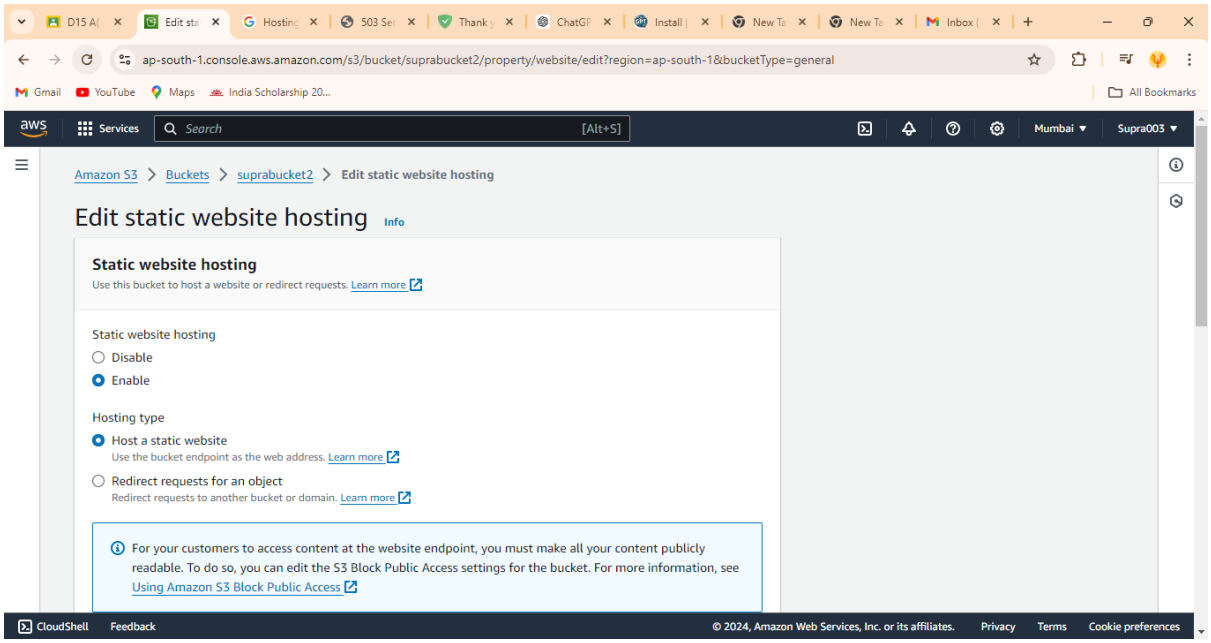

Not secure
webserver-env.eba-227p9xyx.ap-south-1.elasticbeanstalk.com
VPN
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
All Bookmarks
Congratulations!
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.
For next steps, read the AWS CodePipeline Documentation. Incedge 2020

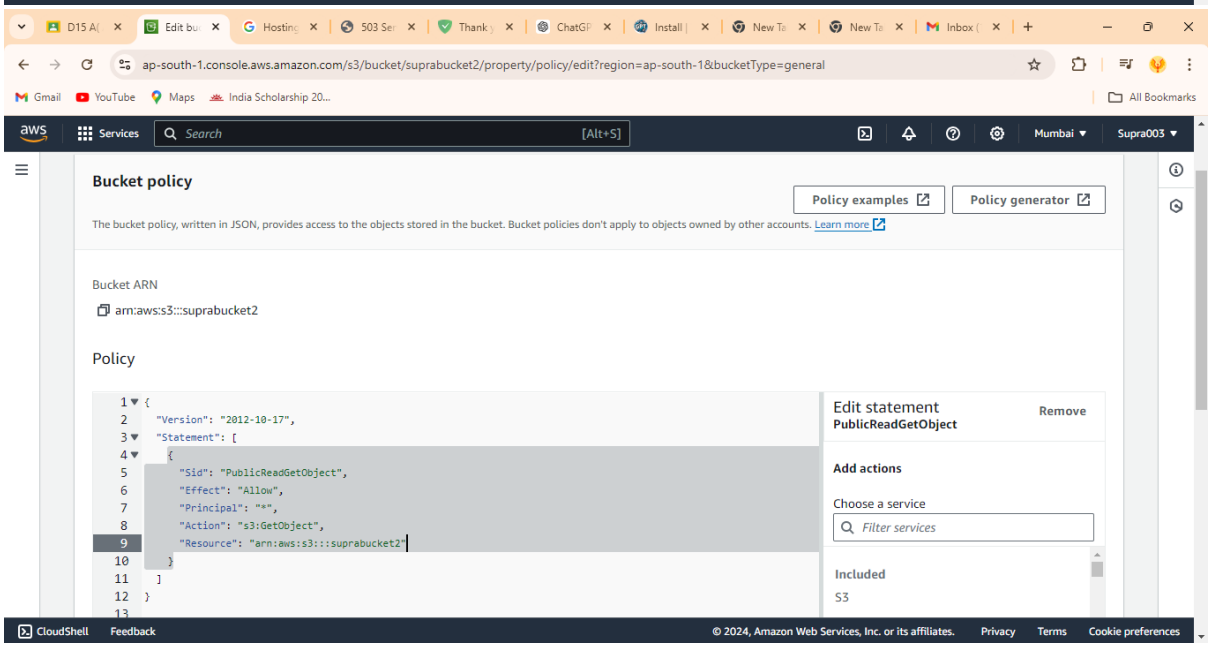
Experiment 01-(a)

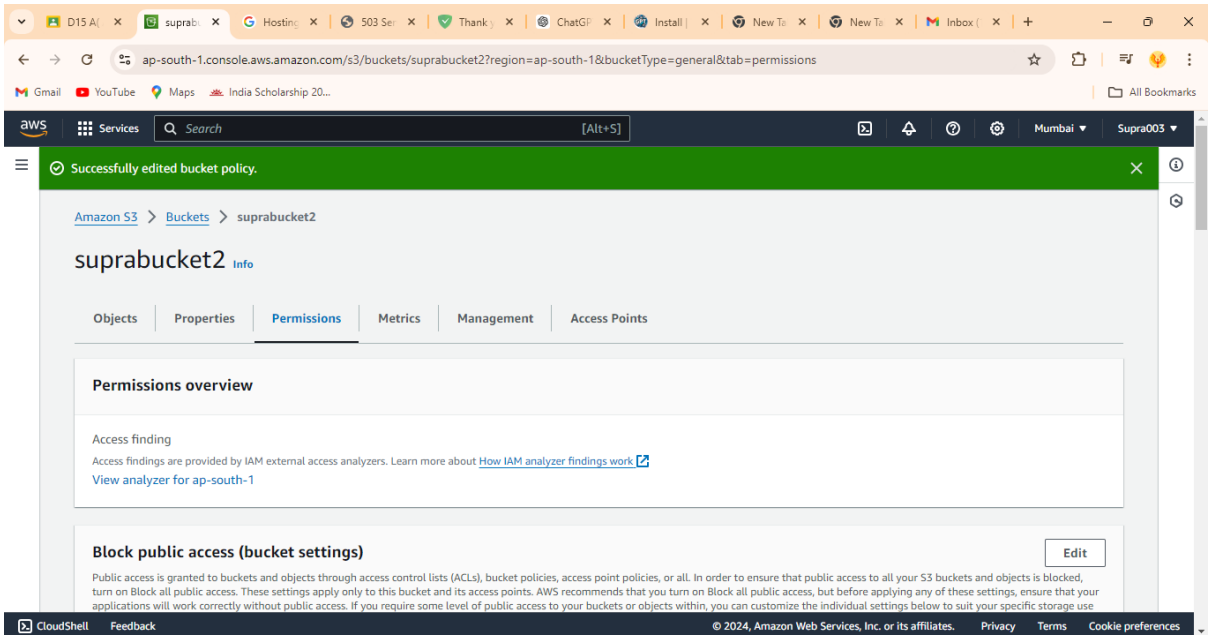
Hosting a static website on Amazon S3

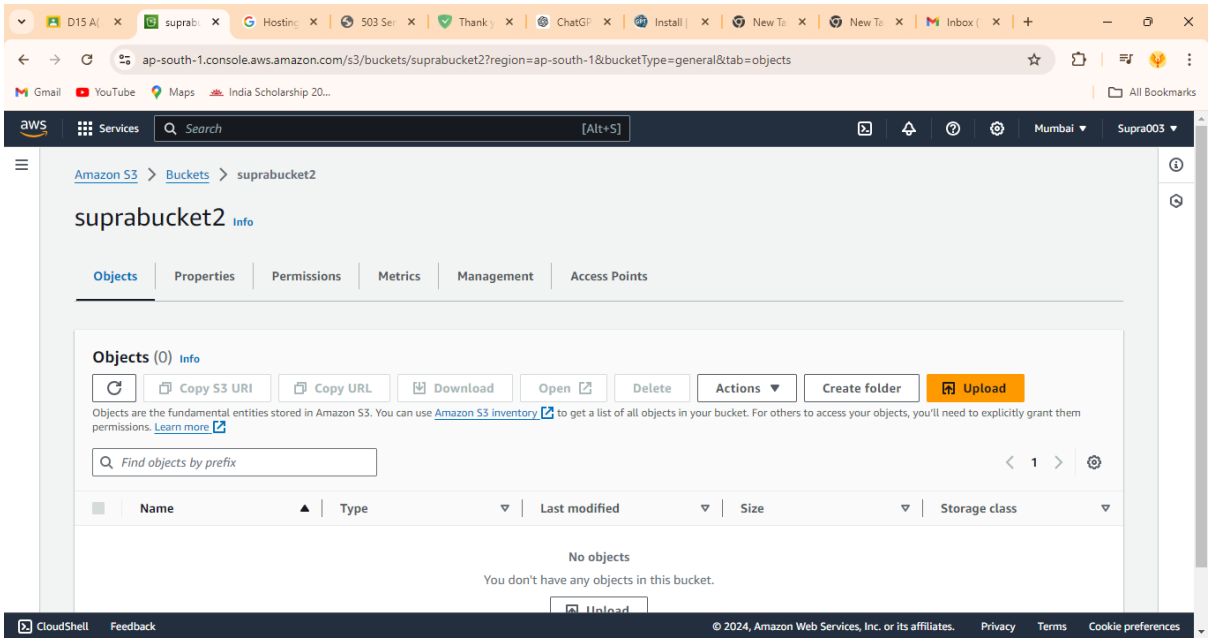


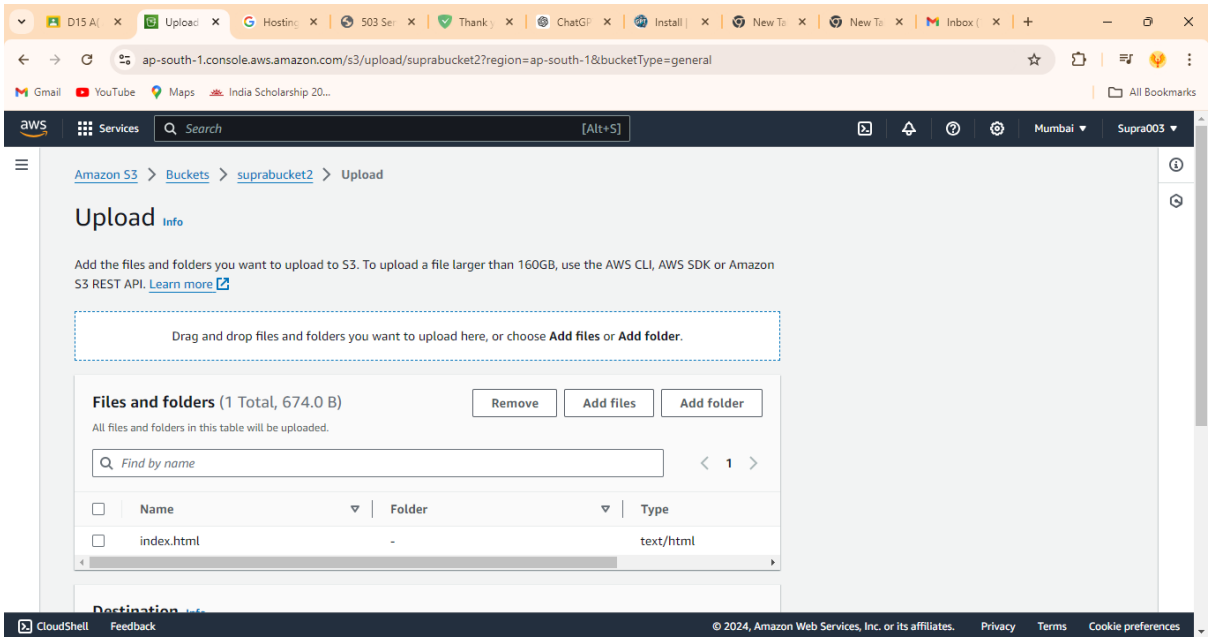


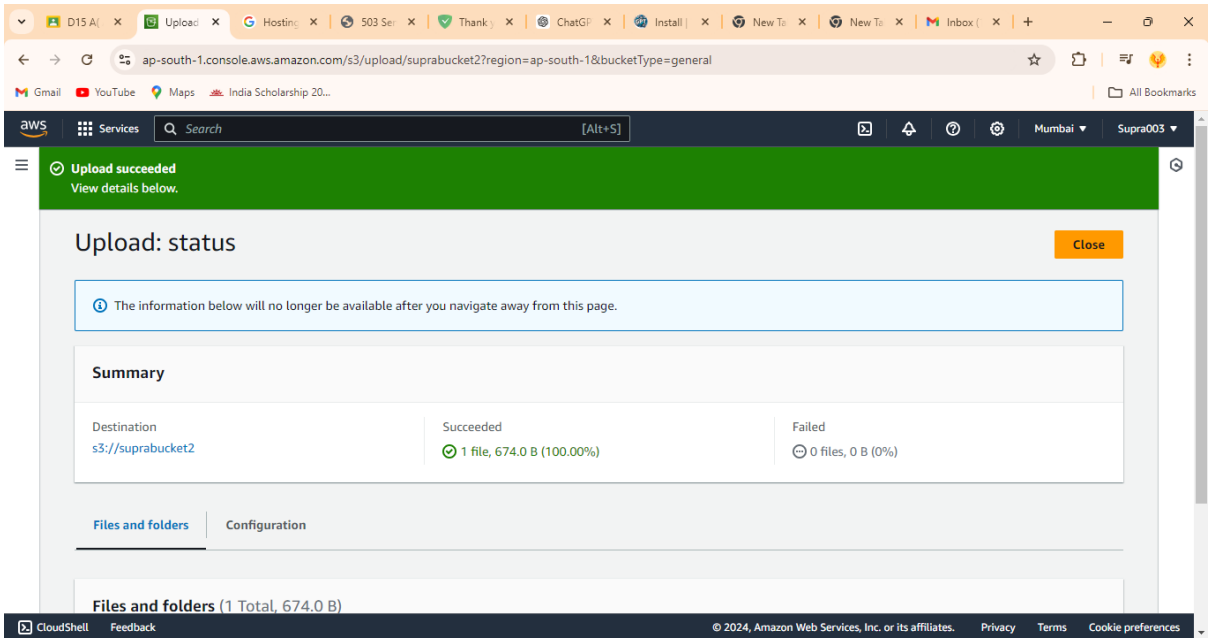

ap-south-1.console.aws.amazon.com/s3/bucket/suprabucket2/property/website/edit?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Amazon S3 > Buckets > suprabucket2 > Edit static website hosting
Edit static website hosting
Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

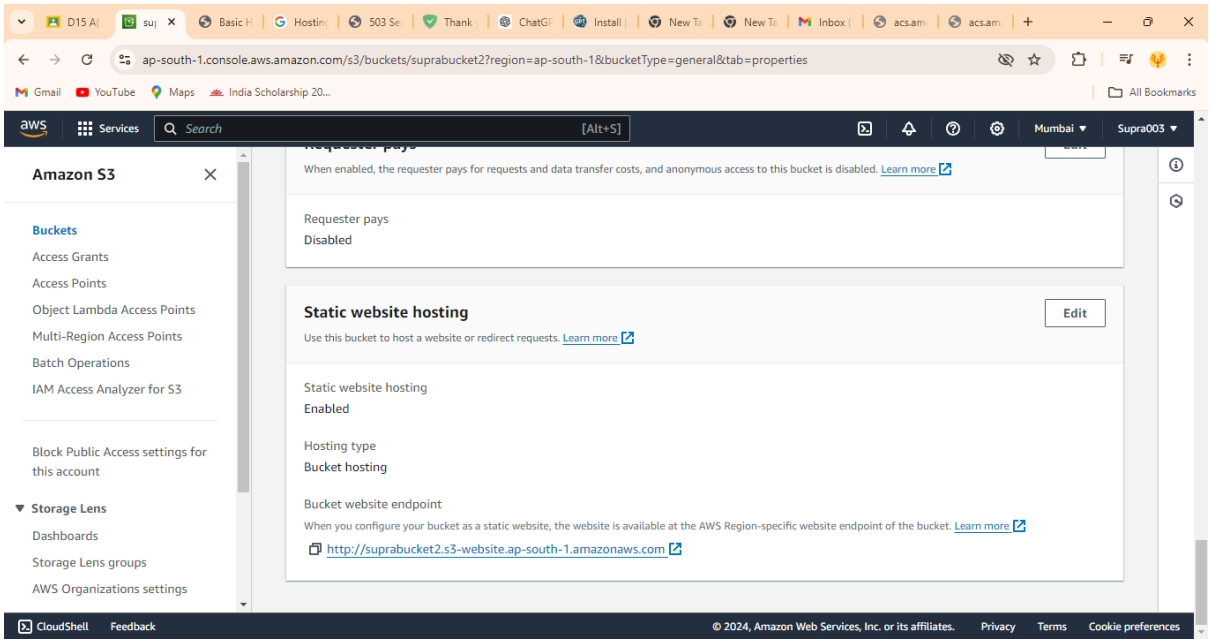

ap-south-1.console.aws.amazon.com/s3/bucket/suprabucket2/property/policy/edit?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Bucket policy
Policy examples
Policy generator
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
Bucket ARN
arn:aws:s3:::suprabucket2
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "PublicReadGetObject",
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::suprabucket2"
}
]
}
Edit statement
PublicReadGetObject
Remove
Add actions
Choose a service
Filter services
Included
S3
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

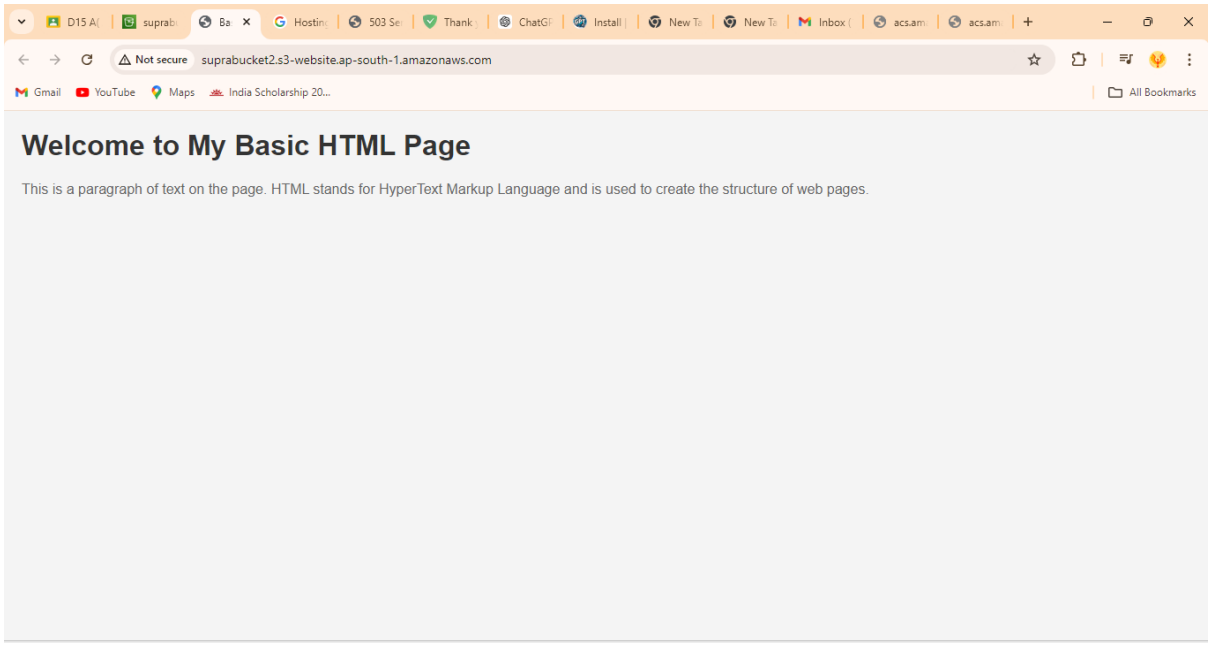

ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=permissions
Successfully edited bucket policy.
Amazon S3 > Buckets > suprabucket2
suprabucket2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for ap-south-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=objects
Amazon S3 > Buckets > suprabucket2
suprabucket2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/upload/suprabucket2?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
Amazon S3 > Buckets > suprabucket2 > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 674.0 B)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
Name
Folder
Type
index.html
-
text/html
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/upload/suprabucket2?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
Supra003
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://suprabucket2
Succeeded
1 file, 674.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 674.0 B)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/s3/buckets/suprabucket2?region=ap-south-1&bucketType=general&tab=properties
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more
Requester pays
Disabled
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Edit
Static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://suprabucket2.s3-website.ap-south-1.amazonaws.com
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
suprabucket2.s3-website.ap-south-1.amazonaws.com
Welcome to My Basic HTML Page
This is a paragraph of text on the page. HTML stands for HyperText Markup Language and is used to create the structure of web pages.

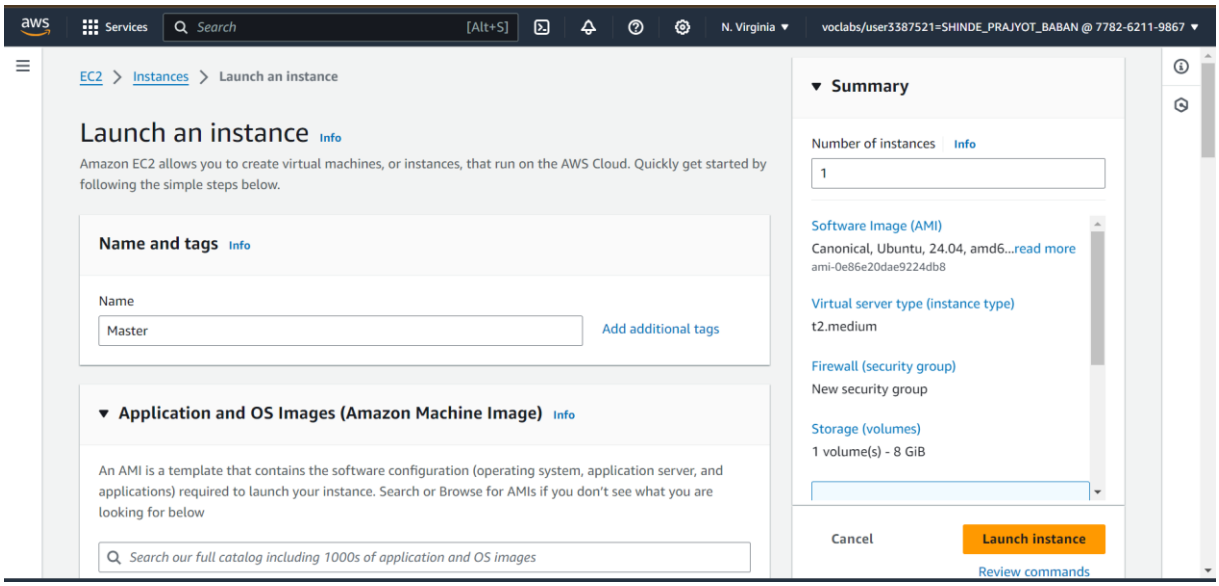
ADVANCE DEVOPS EXP-3

Name : Prajyot Shinde

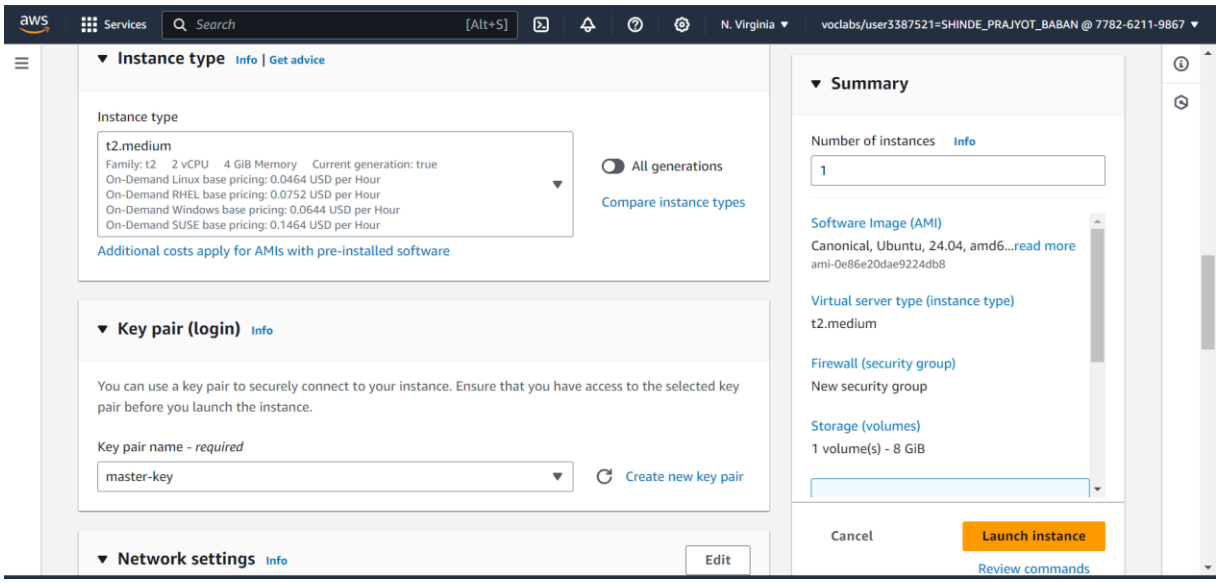
Roll No : 57

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

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

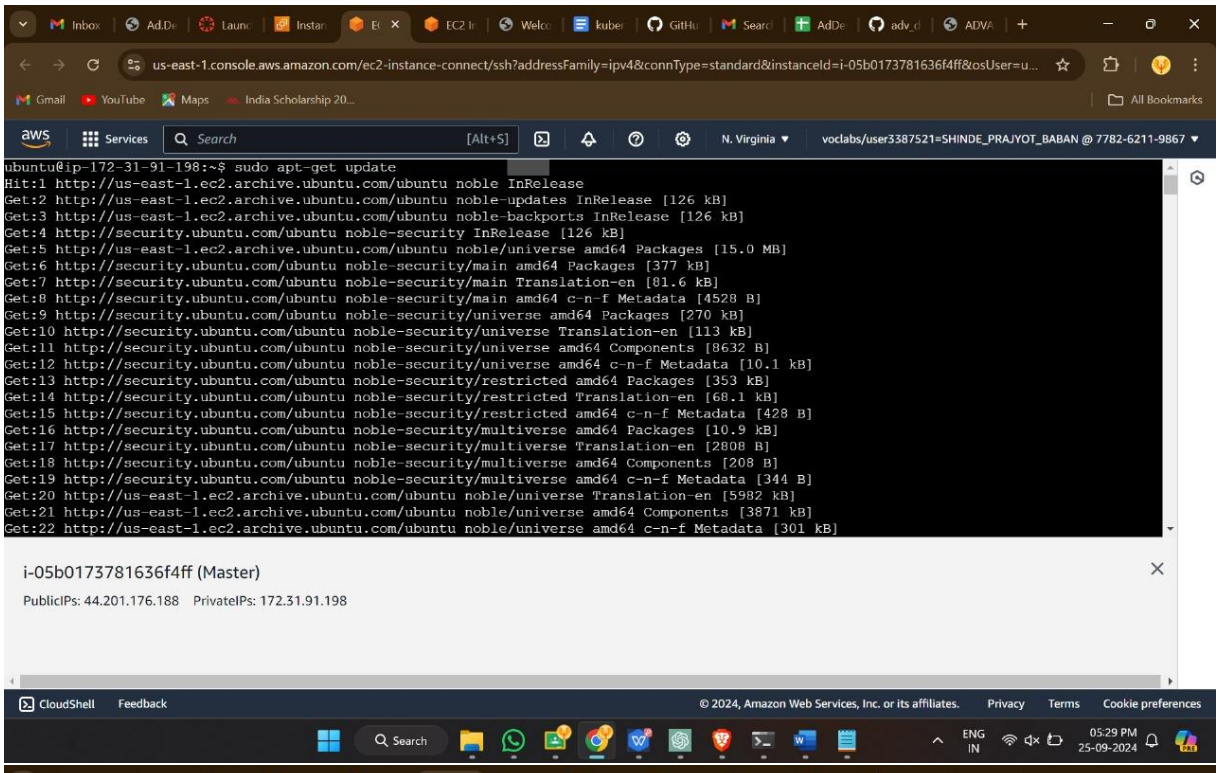


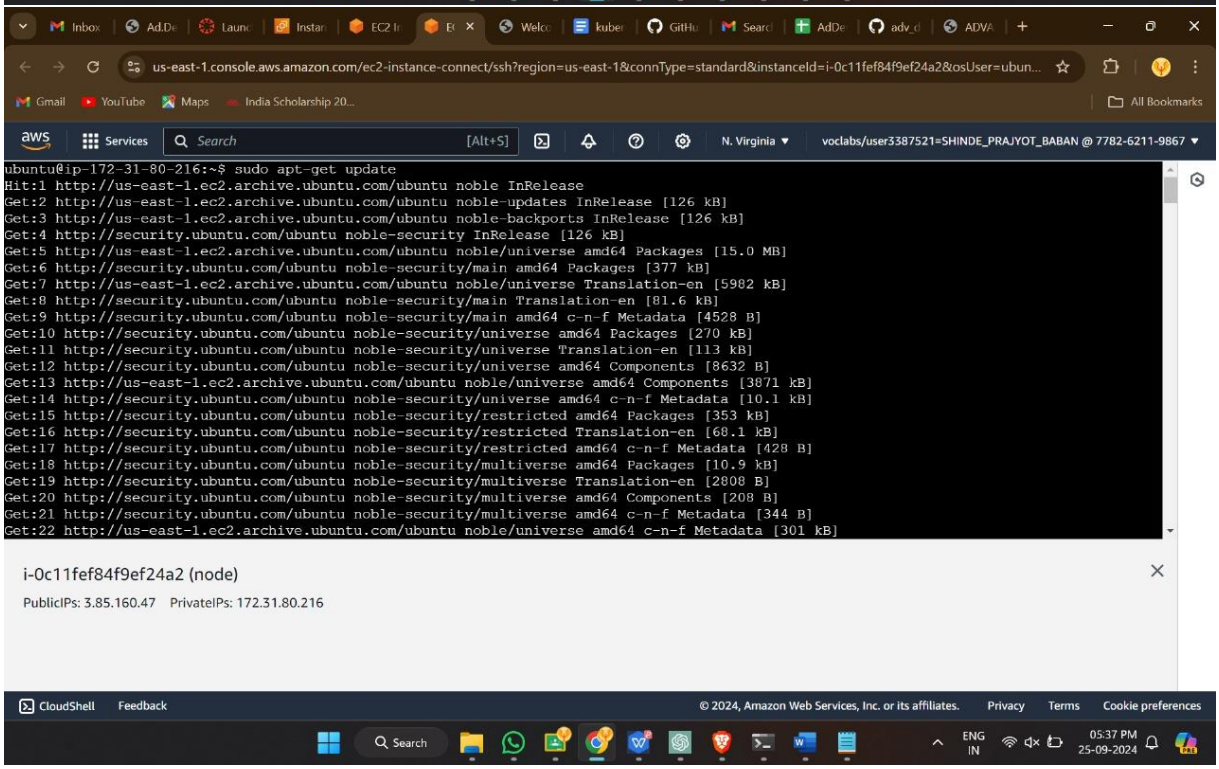
Generate a key pair for the same:



Step2:

Open Master and node on EC2 terminal:

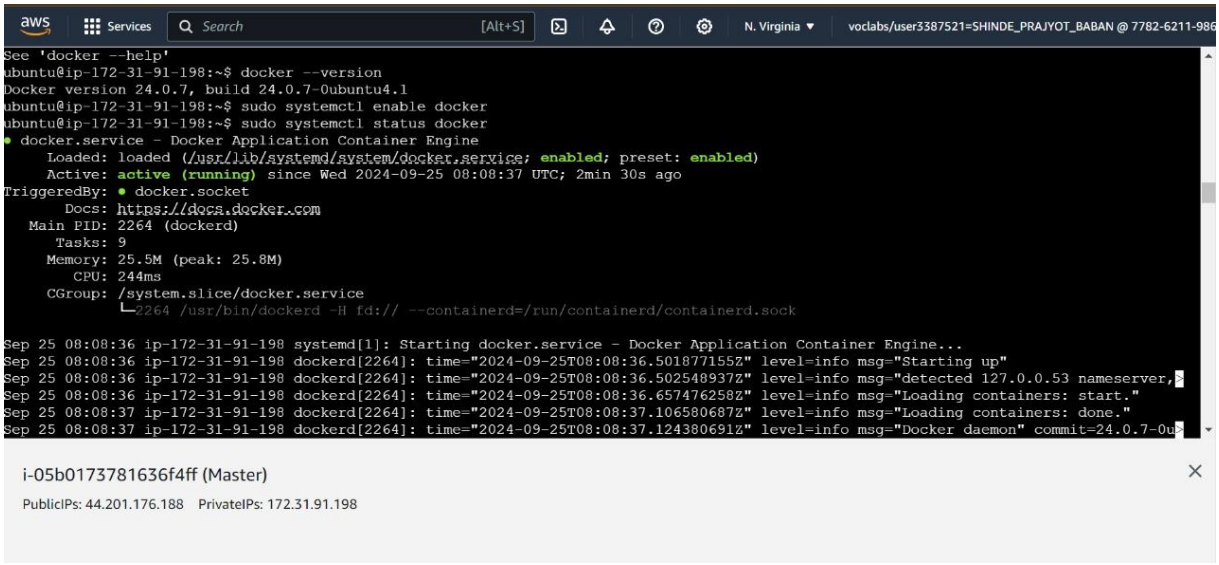




Step 3:

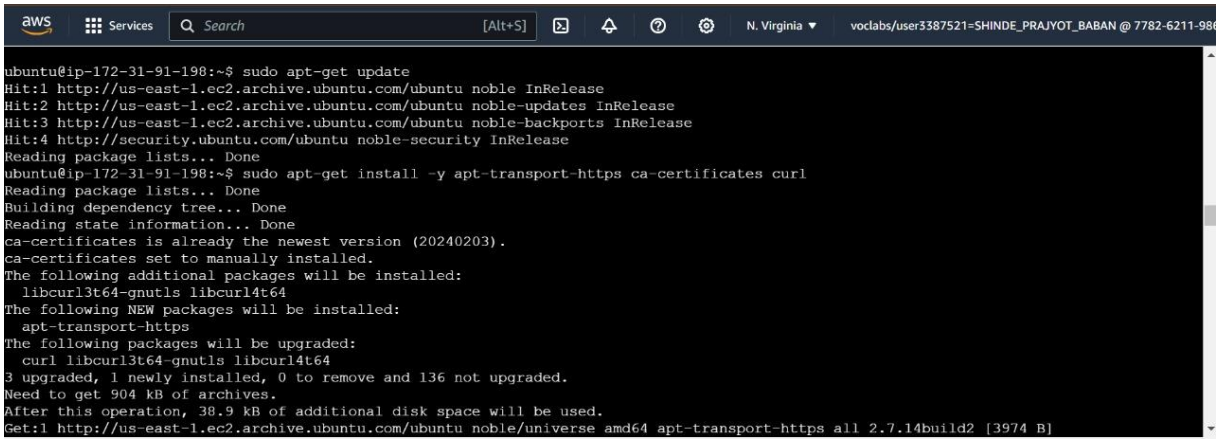
Install Docker

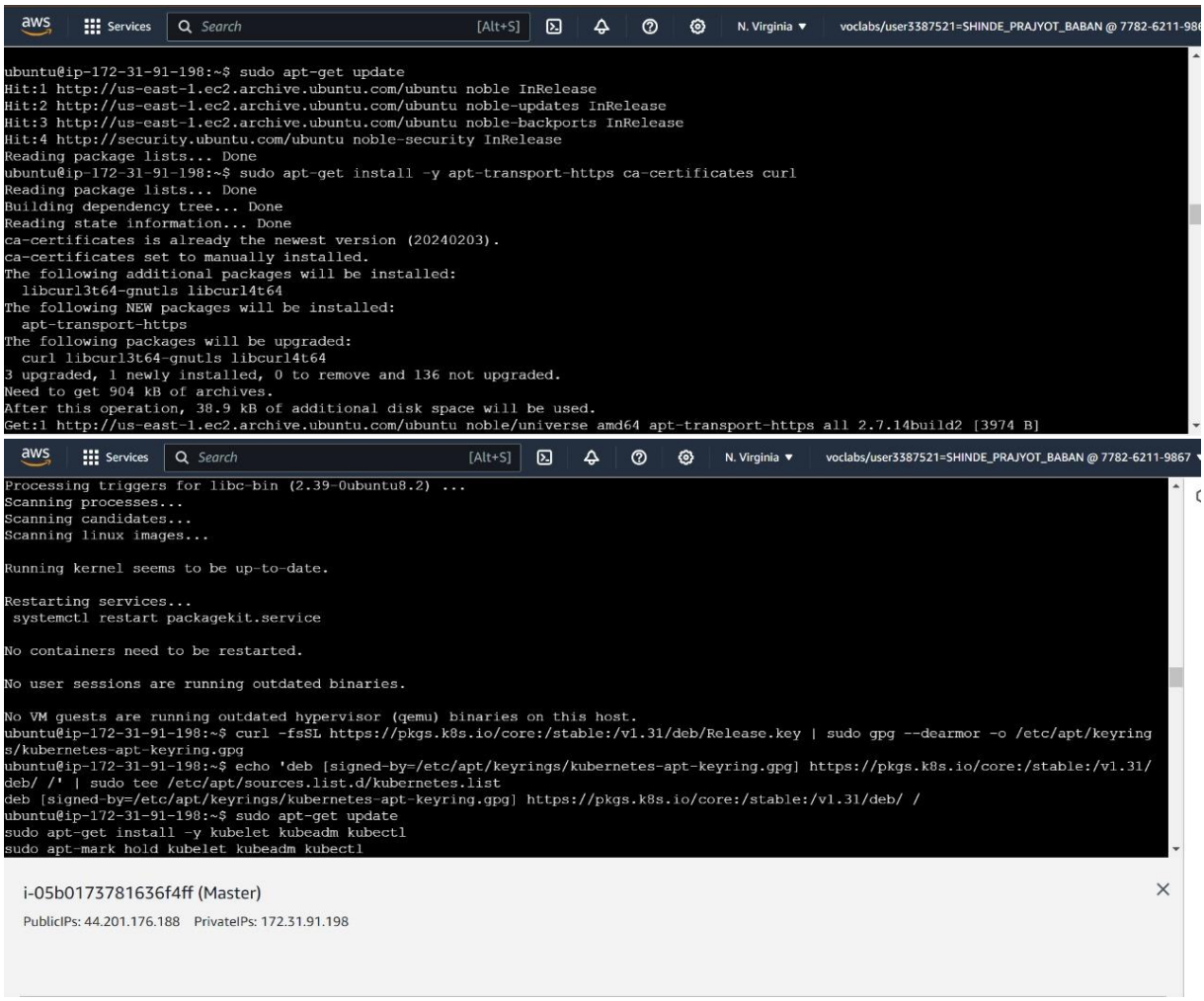
```
ubuntu@ip-172-31-91-198:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 139 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```



Step 4:

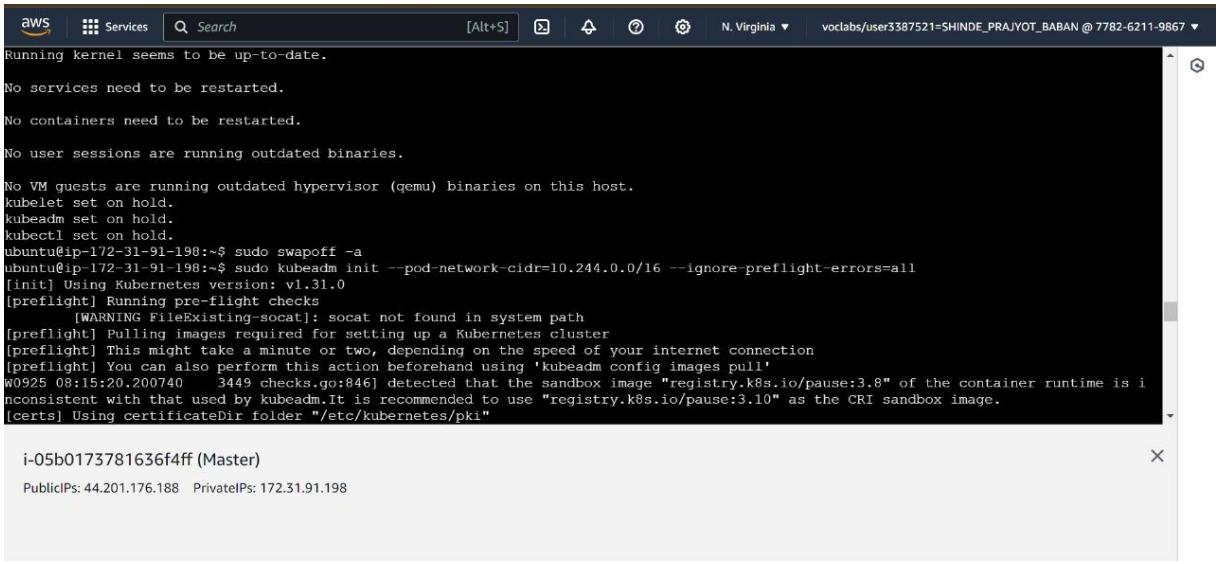
Install kubeadm, kubelet, kubectl:





Step5:

Disable Swap (Kubernetes requires swap to be off):



Step 6:

Initialize the Kubernetes Cluster on Master Node On the master node: sudo

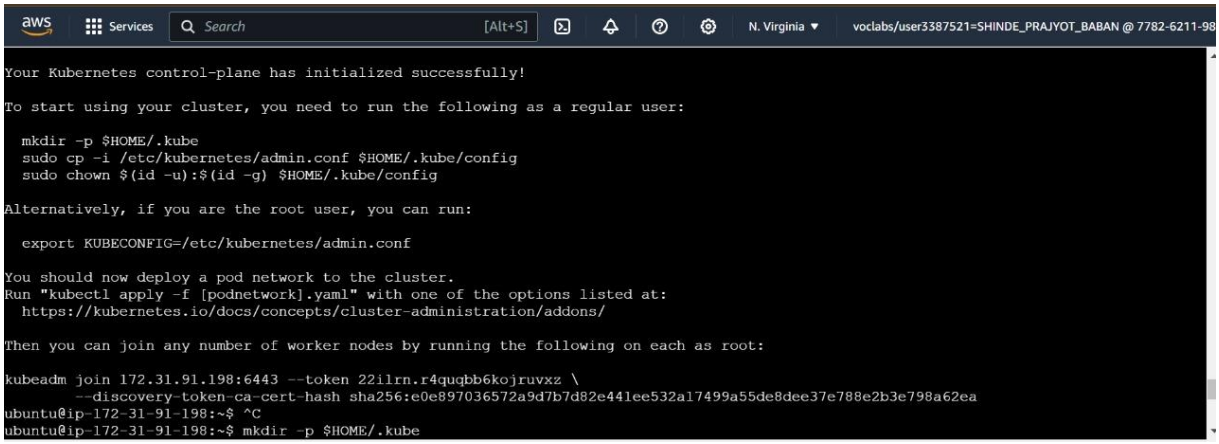
kubeadm init --pod-network-cidr=10.244.0.0/16

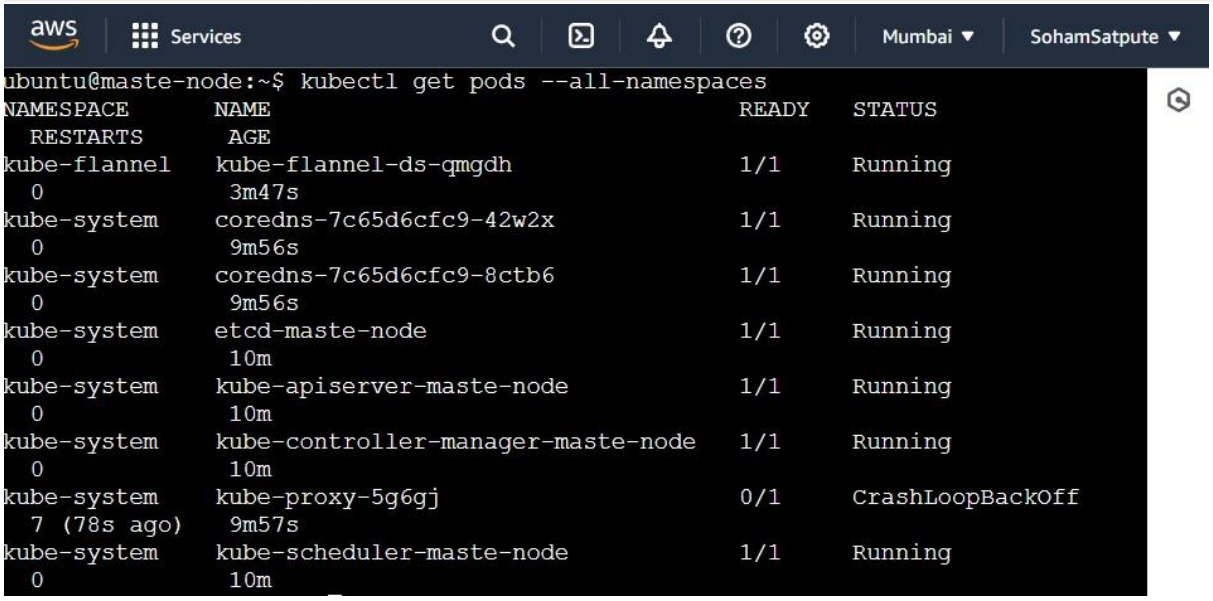
Set up kubectl on the master node:

mkdir -p $HOME/.kube sudo cp -i

/etc/kubernetes/admin.conf $HOME/.kube/config sudo

chown $(id -u):$(id -g) $HOME/.kube/config





Step 7:

Join Worker Nodes to the Cluster On the worker nodes, run the command provided by the master node during initialization:

Step 8:

Verify the Cluster Once the worker node joins, check the status on the master node

ADVANCE DEVOPS EXP 4

Name :- Prajyot Shinde

Roll no :- 57

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

Step 1: As the cluster is up and running, we can deploy our nginx server on this cluster.

```
ubuntu@ip-172-31-91-198:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE     VERSION
ip-172-31-80-216  Ready    <none>          14s     v1.31.1
ip-172-31-91-198  Ready    control-plane   3m35s   v1.31.1
```

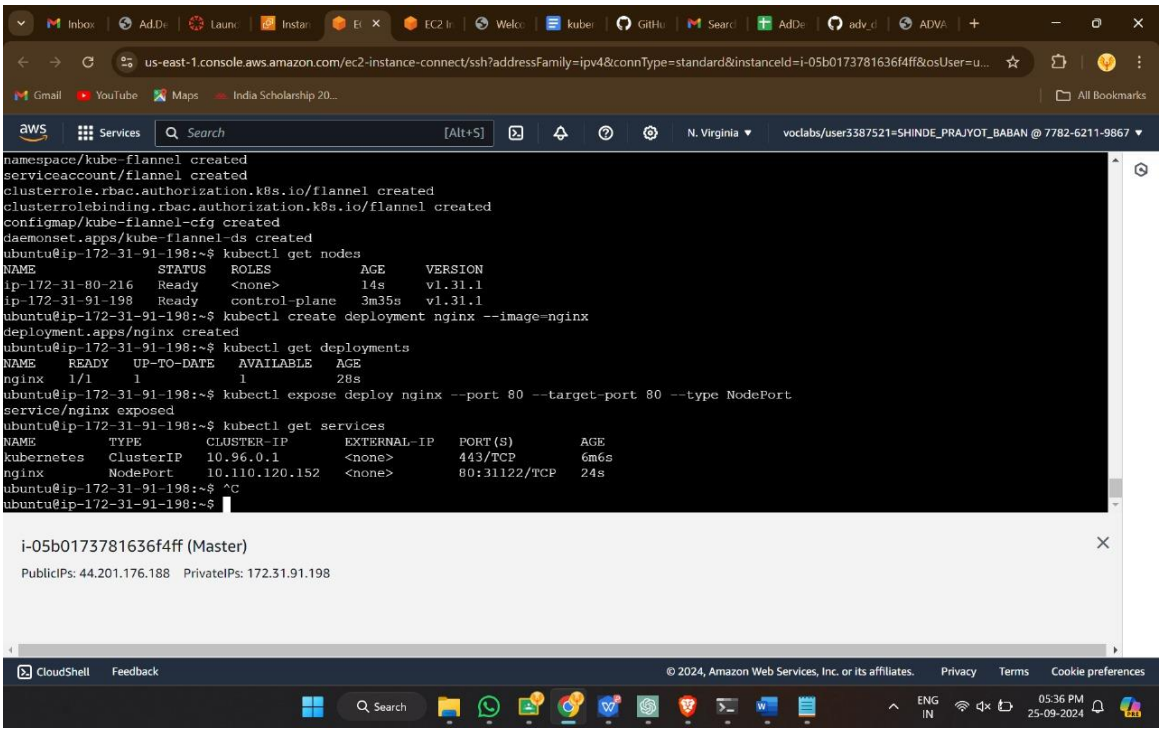
Apply this deployment file using this command to create a deployment.

$kubectl create deployment nginx --image=nginx

```
ubuntu@ip-172-31-91-198:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
```

Step 2: Verify the deployment using the command:

$kubectl get deployments

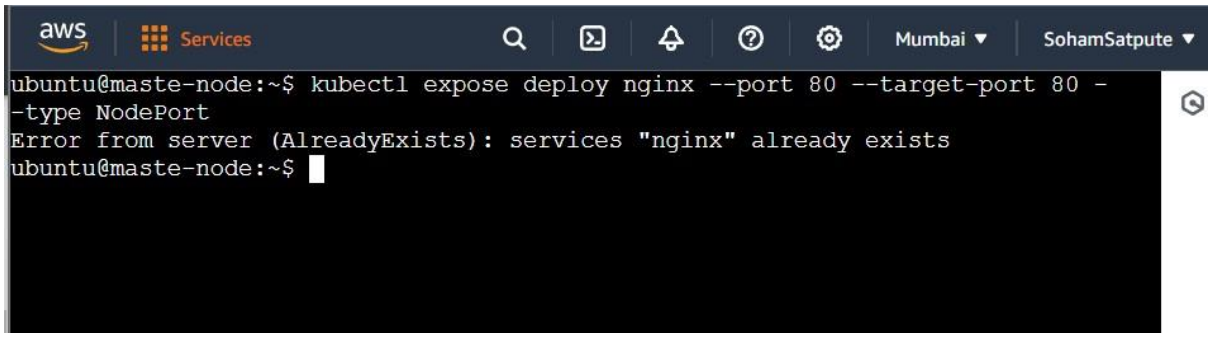


Step 3: Next, run the following command to create a service named nginx that will

expose the app publicly. It will do so through a NodePort, a scheme that will make

the pod accessible through an arbitrary port opened on each node of the clusterwith this servicetype, Kubernetes will assign this service on ports on the **30000+**range.

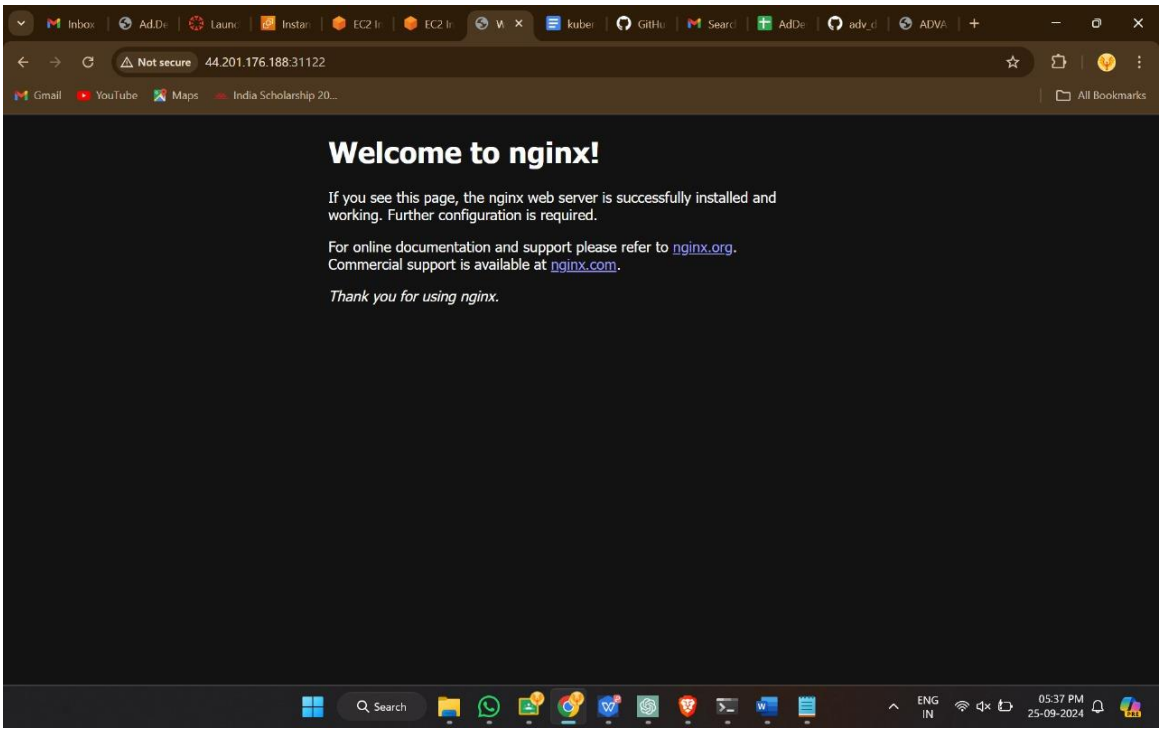
$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort



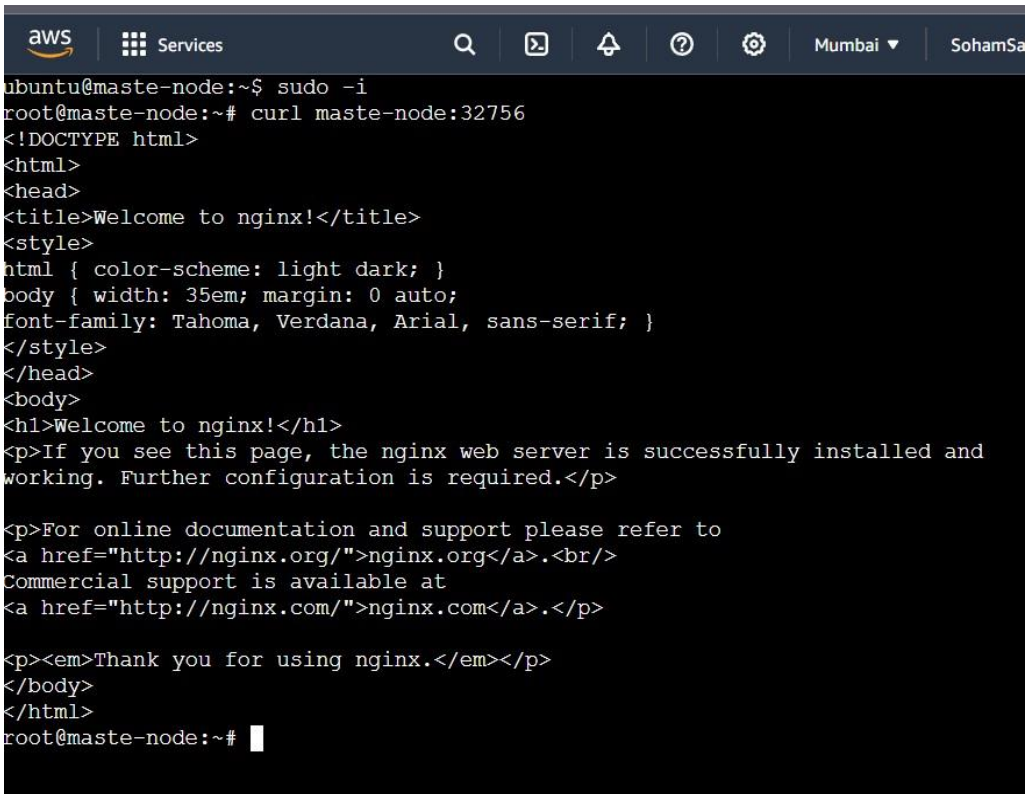
Step 4: Run this command to see a summary of the service and the ports exposed.

$kubectl get services

Step 5: Add the port which is displayed i.e. 32756(in our case) in the inbound rules of the security group.



Step 6: Now you can verify that the Nginx page is reachable on all nodes using thecurl command.As you can see, the "**WELCOME TO NGINX!**" page can be reached.



Step 7: To test that everything is working, visithttp://worker_1_ip:nginx_port or http://worker_2_ip:nginx_port through a browser on your local machine. You will see Nginx's familiar welcomepage.

http://13.127.63.136:32756/

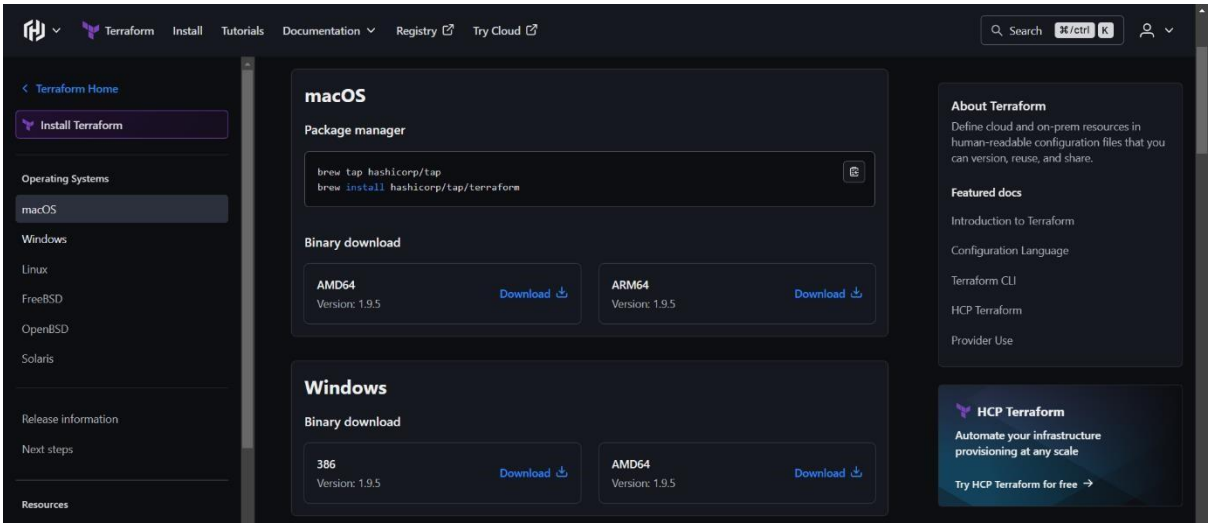
Experiment no 5

Name :Prajyot Shinde

Roll no :- 57/D15A

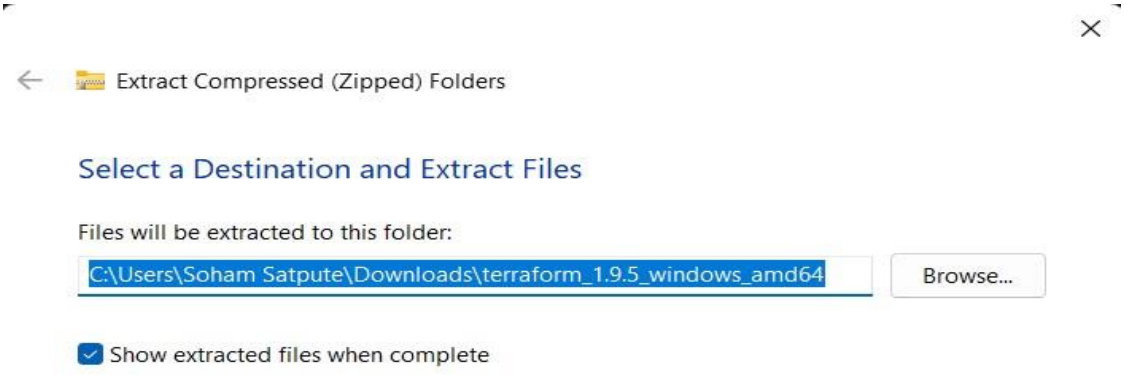
Aim :-Installation and Configuration of Terraform in Windows

Installation for Windows :-



Step1: In your Downloads, right-click on the downloaded Terraform binary file and select "**Extract All**" .

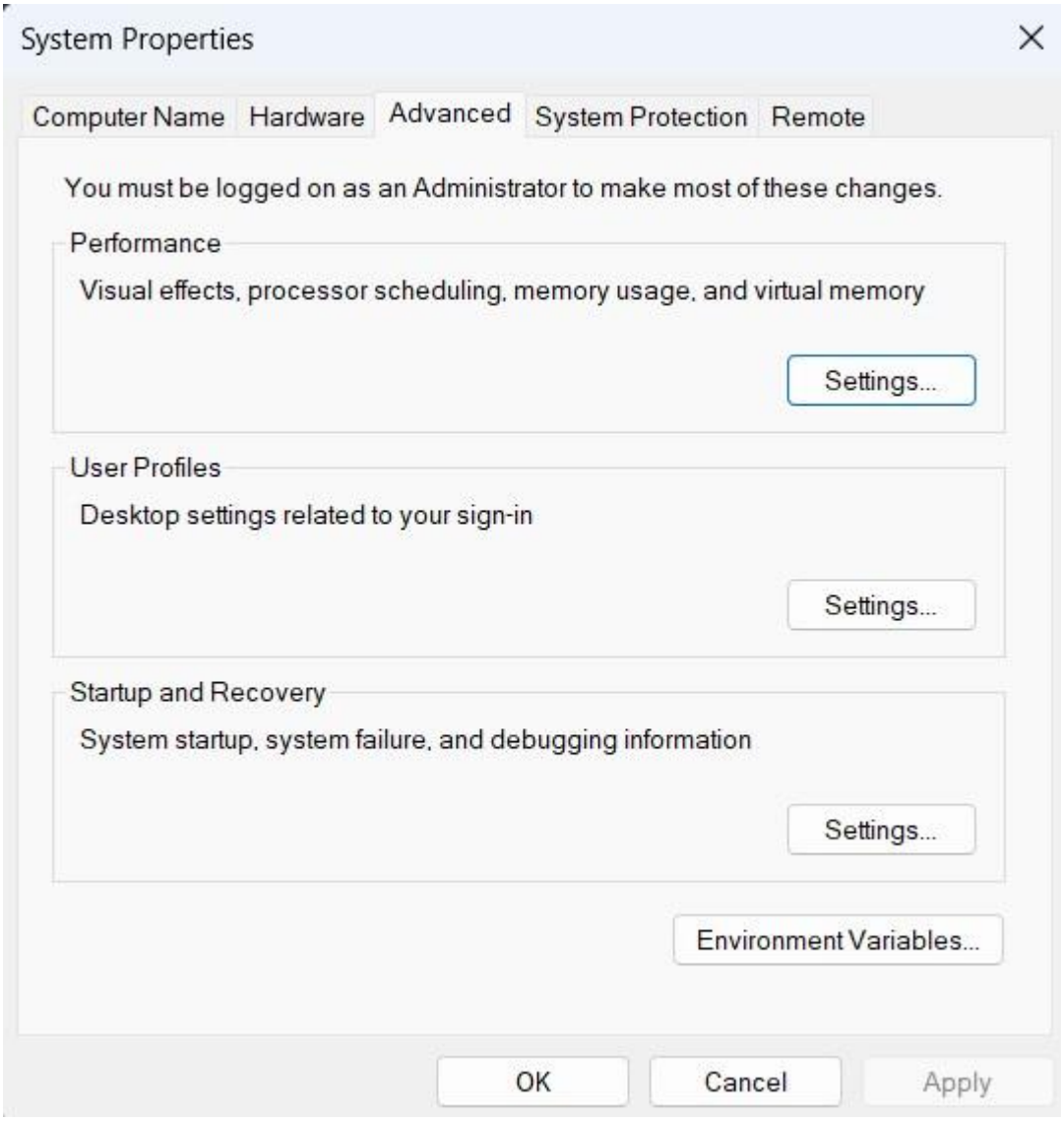
The following window will pop up:



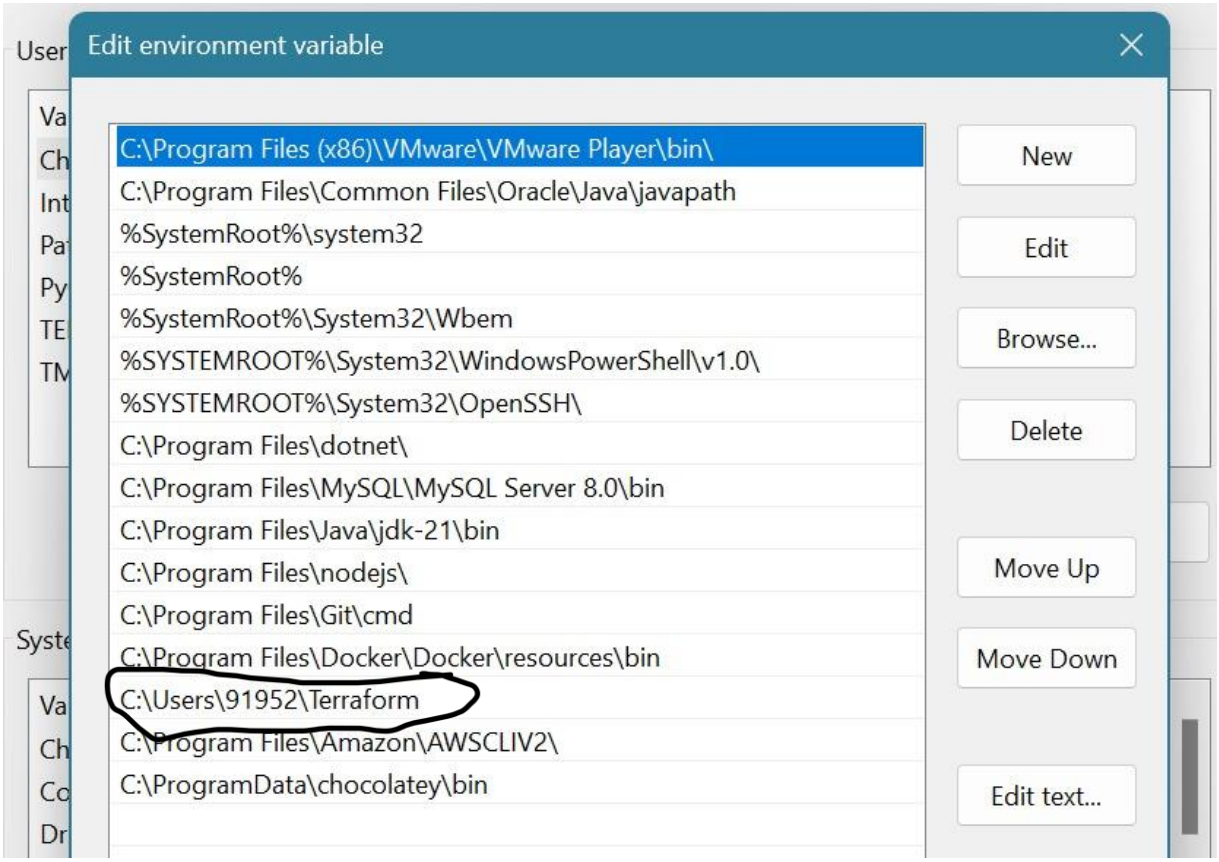
Step 2 :

Set Terraform Path to System Environment Variables

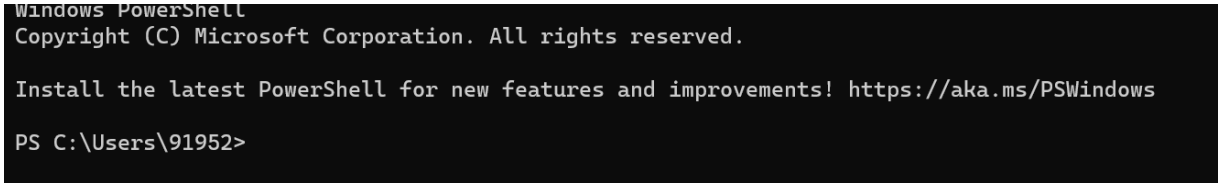
Click on the "**Environment variables**" in the system properties:



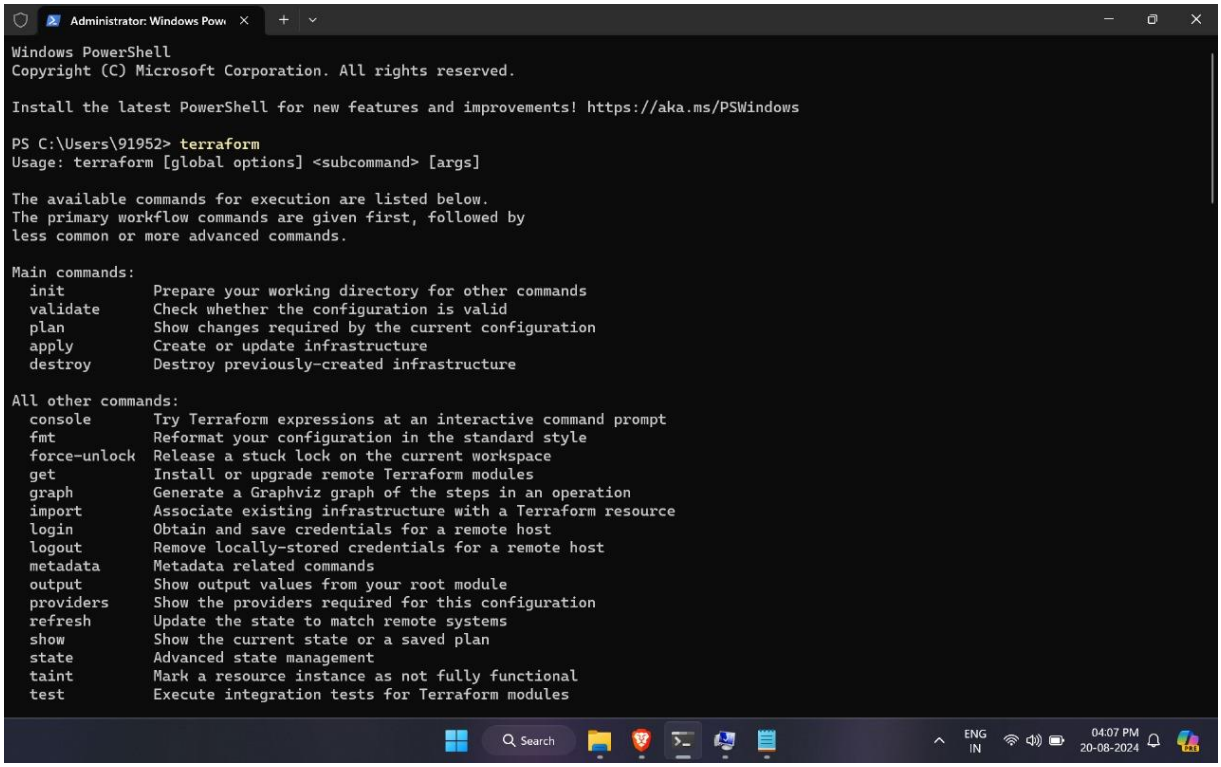
Step 3: Set the path



Step 4: Navigate to the folder path C:\terraform in a new command prompt window and type the terraform -version to verify the installed version.



Step 5: To see more Terraform commands, you can simply type terraform in the terminal.


Administrator: Windows Pow
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\91952> terraform
Usage: terraform [global options] <subcommand> [args]
The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.
Main commands:
init Prepare your working directory for other commands
validate Check whether the configuration is valid
plan Show changes required by the current configuration
apply Create or update infrastructure
destroy Destroy previously-created infrastructure
All other commands:
console Try Terraform expressions at an interactive command prompt
fmt Reformat your configuration in the standard style
force-unlock Release a stuck lock on the current workspace
get Install or upgrade remote Terraform modules
graph Generate a Graphviz graph of the steps in an operation
import Associate existing infrastructure with a Terraform resource
login Obtain and save credentials for a remote host
logout Remove locally-stored credentials for a remote host
metadata Metadata related commands
output Show output values from your root module
providers Show the providers required for this configuration
refresh Update the state to match remote systems
show Show the current state or a saved plan
state Advanced state management
taint Mark a resource instance as not fully functional
test Execute integration tests for Terraform modules
Search
ENG
IN
04:07 PM
20-08-2024

Experiment no 6

Name :- Prajyot Shinde

Roll_No :- 57

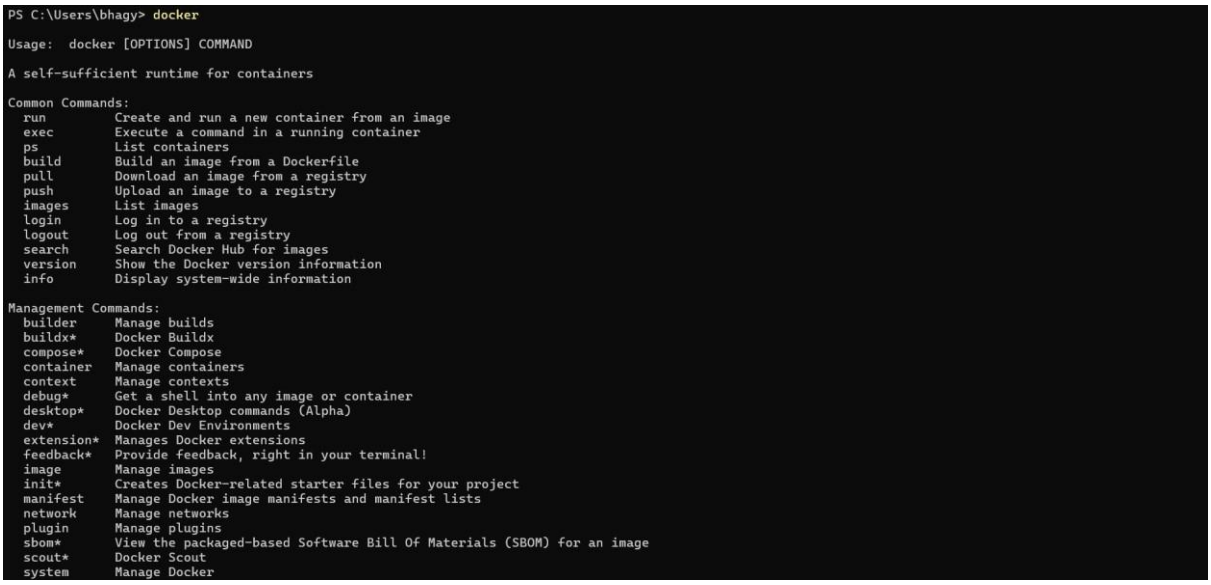
Aim :-
To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker) fdp

Part A:Creating docker image using terraform

Prerequisite:

1) Download and Install Docker Desktop from https://www.docker.com/

Step 1: Check Docker functionality



Check for the docker version with the following command

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.

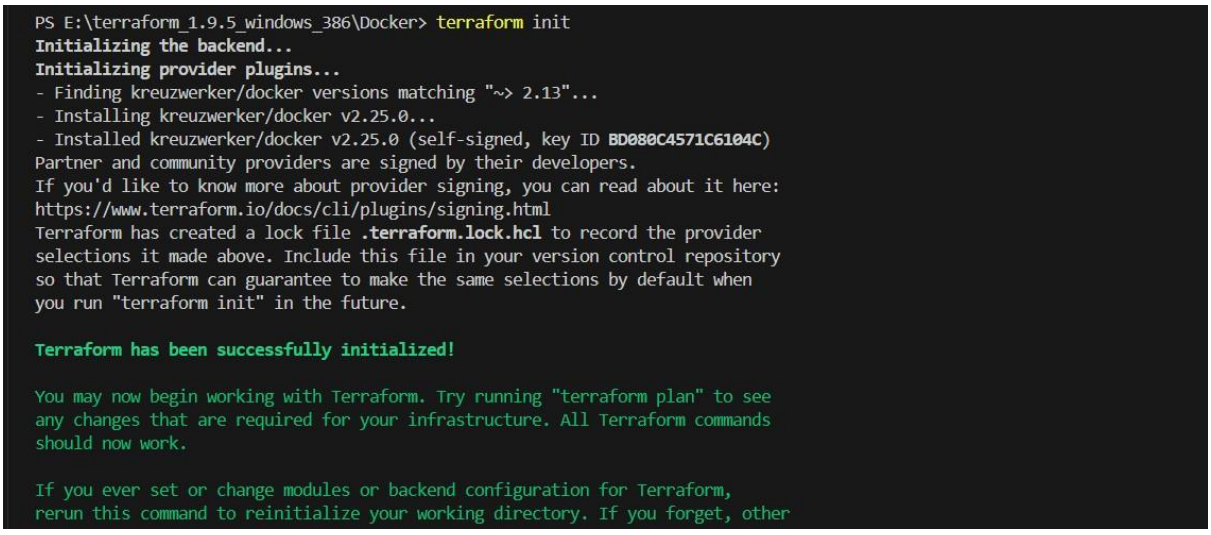
Script:-

```
terraform {
required_providers {
  docker = {
source = "kreuzwerker/docker"
version = "2.25.0"
  }
 }
}

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

resource "docker_image" "ubuntu" {
name = "ubuntu:latest"
}

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

docker.tf

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.25.0"
    }
  }
}

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

resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

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

Step 3: Execute Terraform Init command to initialize the resources



```
PS E:\terraform_1.9.5_windows_386\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 2.13"...
- Installing kreuzwerker/docker v2.25.0...
- Installed kreuzwerker/docker v2.25.0 (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
```

Step 4: Execute Terraform plan to see the available resources

```
PS E:\terraform_1.9.5_windows_386\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)
      + container_read_refresh_timeout_milliseconds = 15000
      + 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)
```

Step 5 : Type terraform apply to apply changes .

```
PS E:\terraform_1.9.5_windows_386\Docker> terraform apply

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

Terraform will perform the following actions:

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

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

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

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

  Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 8s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:l
atest]
docker_container.foo: Creating...
```

Ln 22, Col 1 Spaces: 4 UTF-8 CRLF Terraform Prettier

Docker images , Before Executing Apply Step

```
PS E:\terraform_1.9.5_windows_386\Docker> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

Docker images , After Executing Apply Step

```
PS E:\terraform_1.9.5_windows_386\Docker> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    edbfe74c41f8   3 weeks ago   78.1MB
```

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

```
PS E:\terraform_1.9.5_windows_386\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

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

Plan: 0 to add, 0 to change, 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: yes

docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

Docker images After Executing Destroy step

```
PS E:\terraform_1.9.5_windows_386\Docker> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

Prajyot Shinde 57/D15A

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

Theory:

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

What problems does SAST solve?

SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise. It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

Why is SAST important?

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.

Thus, integrating static analysis into the SDLC can yield dramatic results in the overall quality of the code developed.

What are the key steps to run SAST effectively?

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

1. Finalize the tool. Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.

2. Create the scanning infrastructure, and deploy the tool. This step involves

handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.

3. Customize the tool. Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security

vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.

4. Prioritize and onboard applications. Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk

applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.

5. Analyze scan results. This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation.

6. Provide governance and training. Proper governance ensures that your development teams are employing the scanning tools properly. The software

security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

Integrating Jenkins with SonarQube: Windows installation

Step 1 Install JDK 1.8

Step 2 download and install jenkins
installing-the-default-jre-jdk Step 1 Install JDK 1.8
sudo apt-get install openjdk-8-jre sudo apt install default-jre /

how-to-install-jenkins-on-ubuntu-20-04

Open SSH

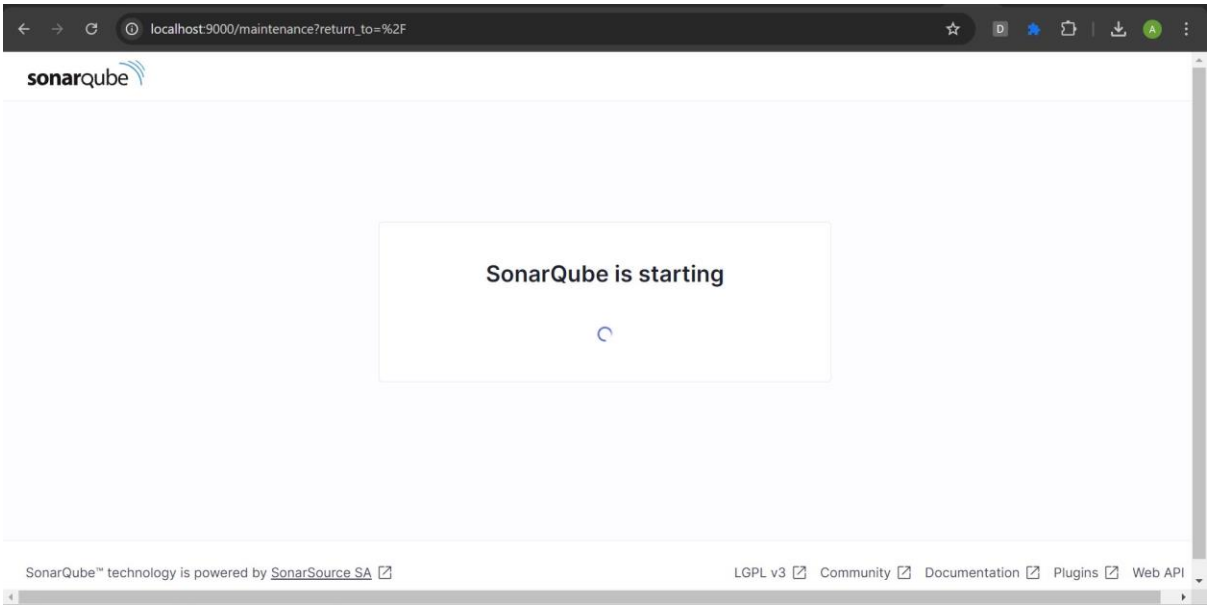
Prerequisites:

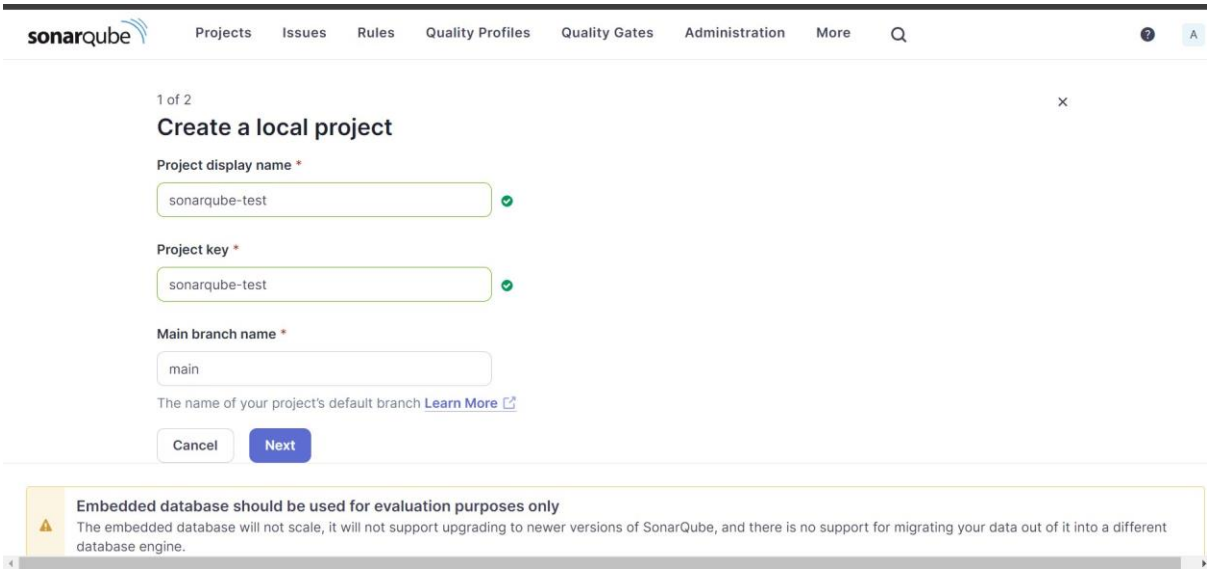
- Jenkins installed
- Docker Installed (for SonarQube)

(sudo apt-get install docker-ce=5:20.10.15~3-0~ubuntu-jammy docker-ce-cli=5:20.10.15~3-0~ubuntu-jammy containerd.io docker-compose-plugin)

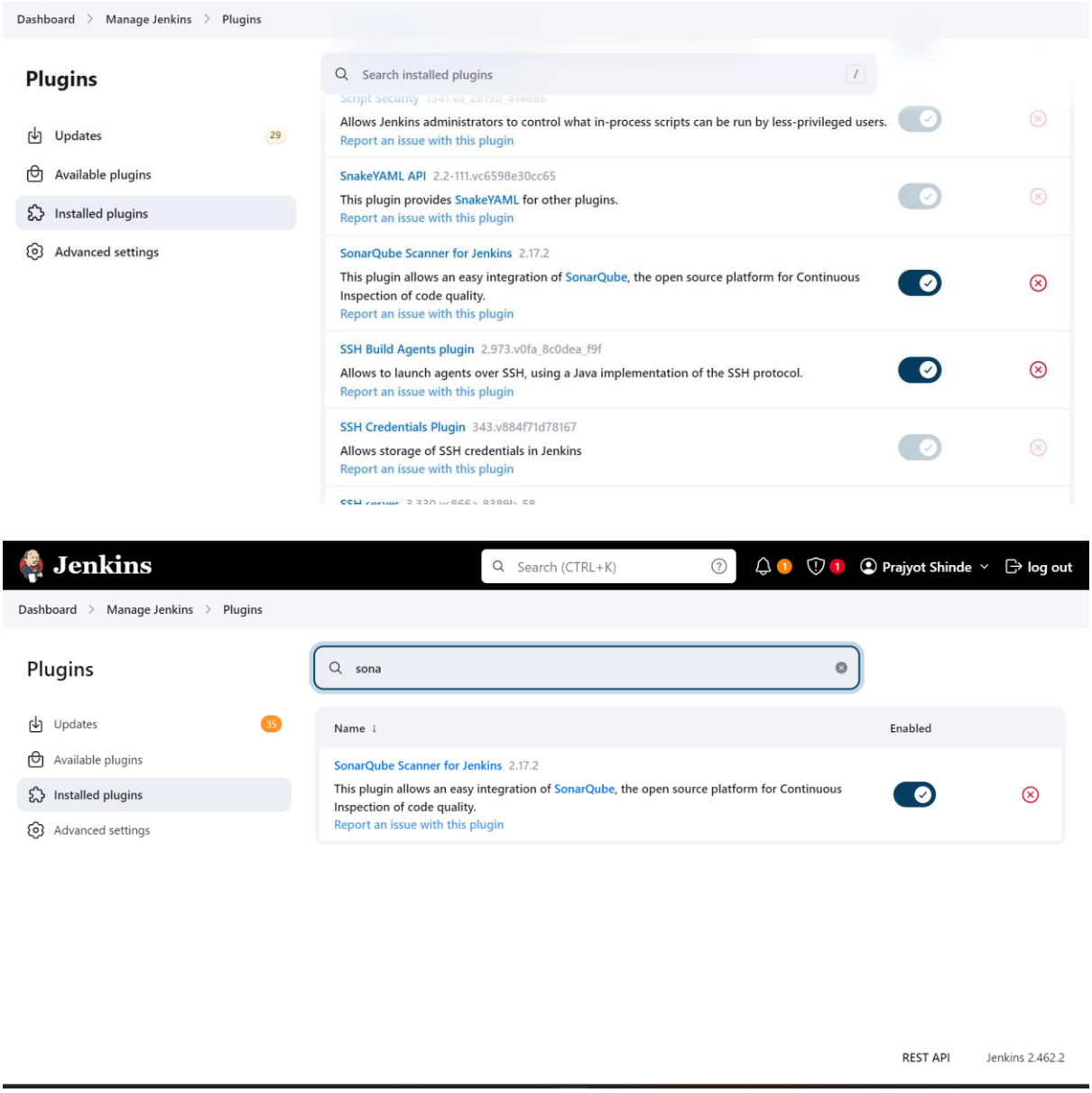
- SonarQube Docker Image

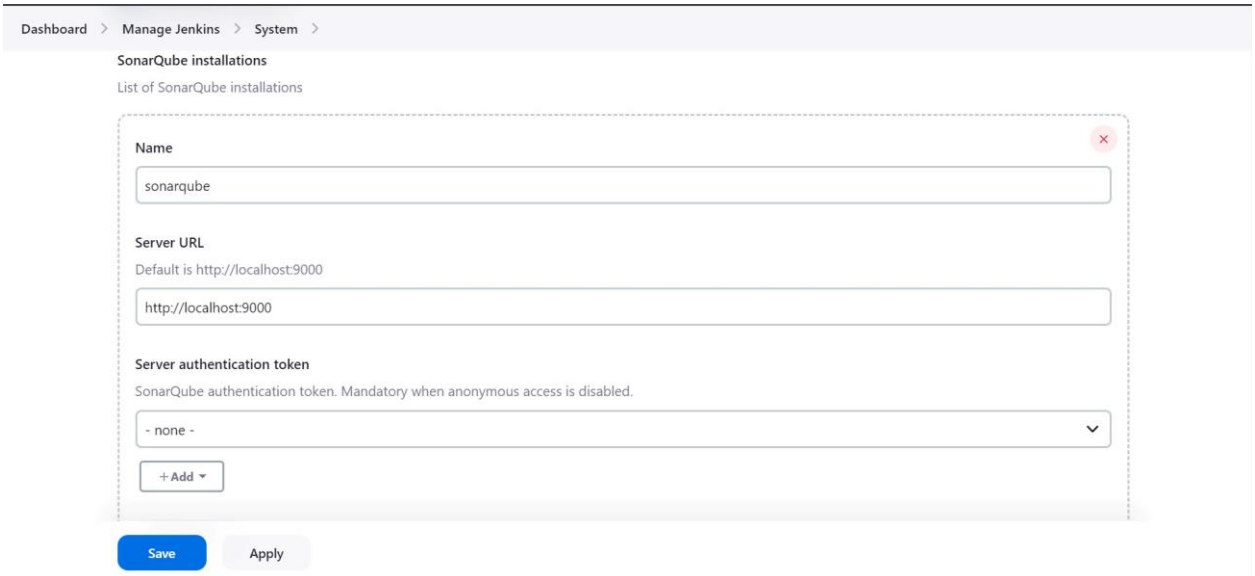
```
C:\Users\91952>docker run -d --name sonarqube -p 9000:9000 sonarqube
0392bc54d6d9ecd3e651c04433976aae2ab159b63bc07776b81a03ebf6e754b5

C:\Users\91952>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
postgres     latest    026d0ab72b34   7 weeks ago     611MB
sonarqube    latest    72e9feec7124   2 months ago    1.92GB

C:\Users\91952>
```

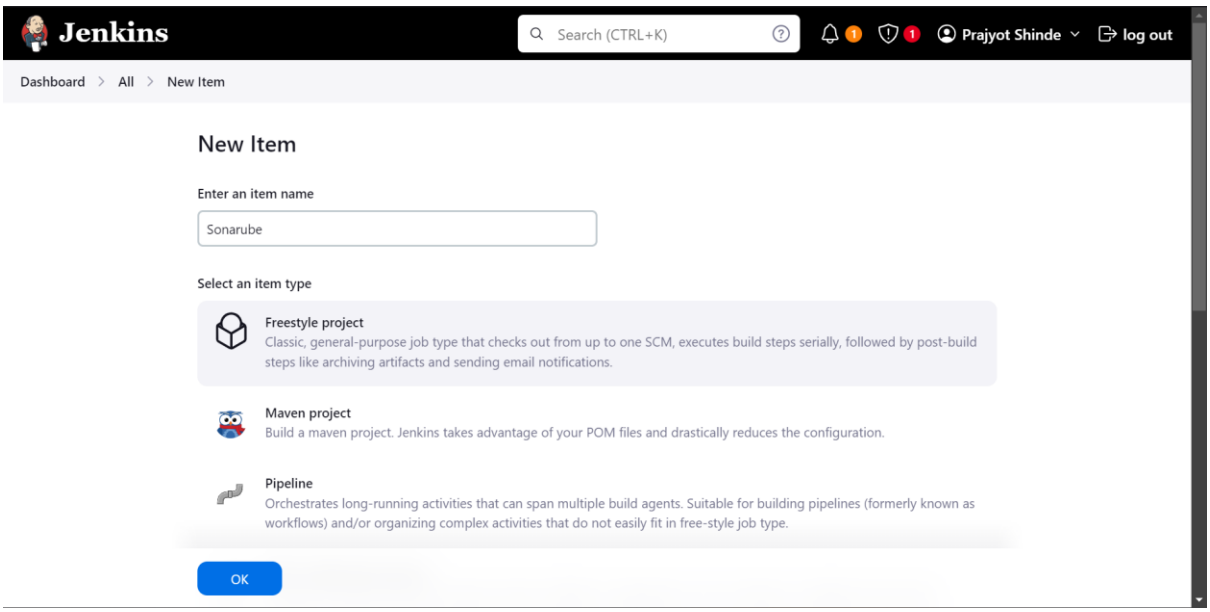


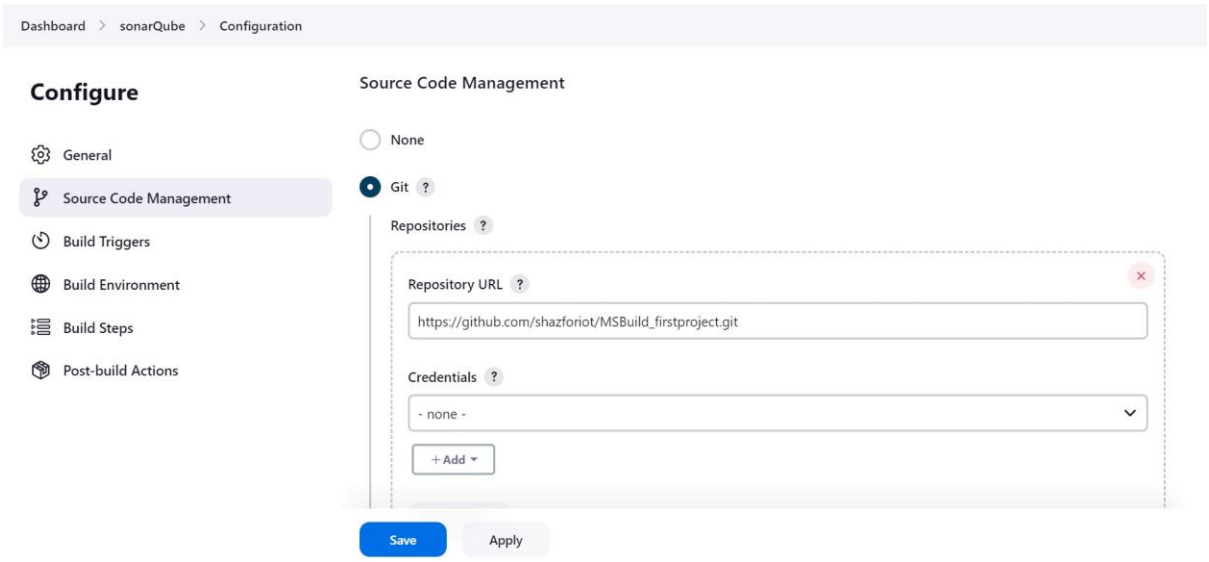


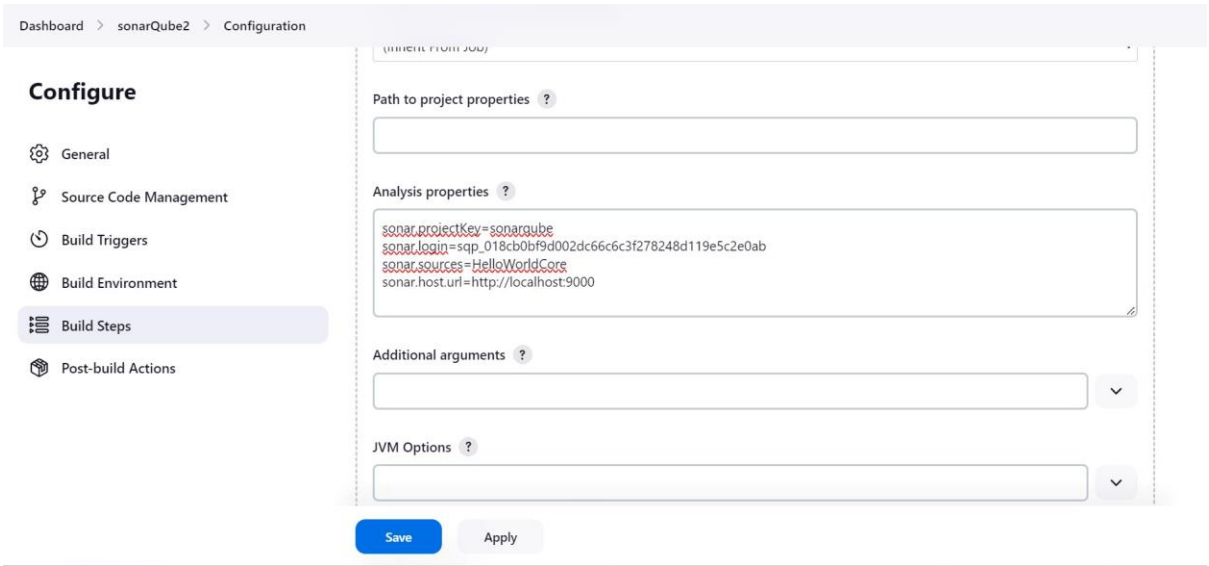
Ok so go to Jenkins dashboard and then go to Manage jenkins then plugins , install plugins and then install “sonarqube scanner”

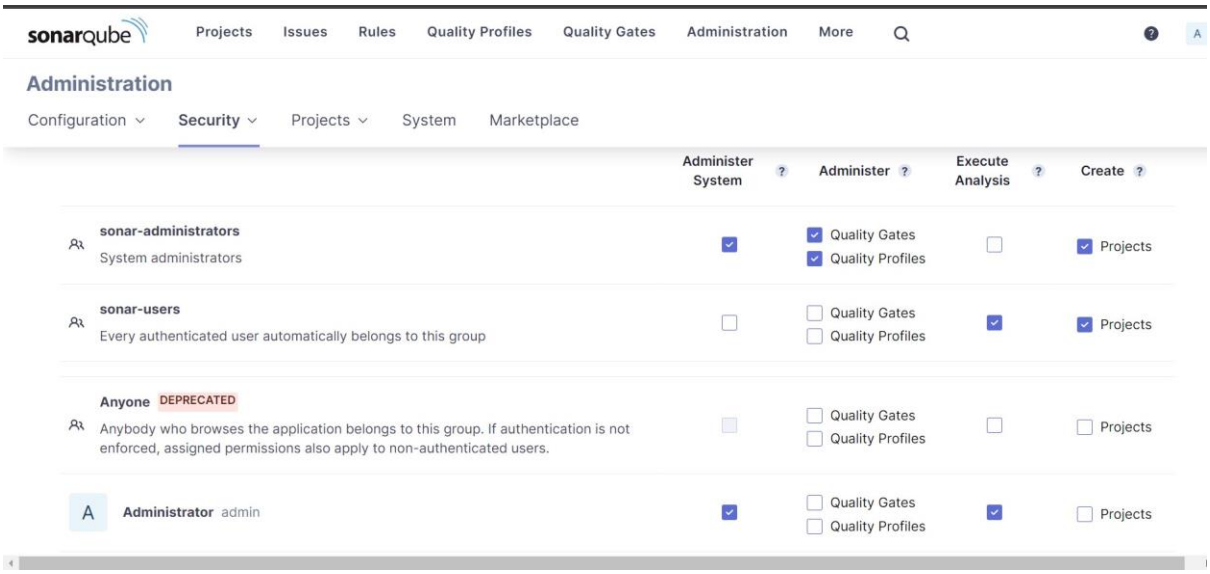



Dashboard > Manage Jenkins > System >
SonarQube installations
List of SonarQube installations
Name
sonarqube
Server URL
Default is http://localhost:9000
http://localhost:9000
Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -
+ Add
Save
Apply

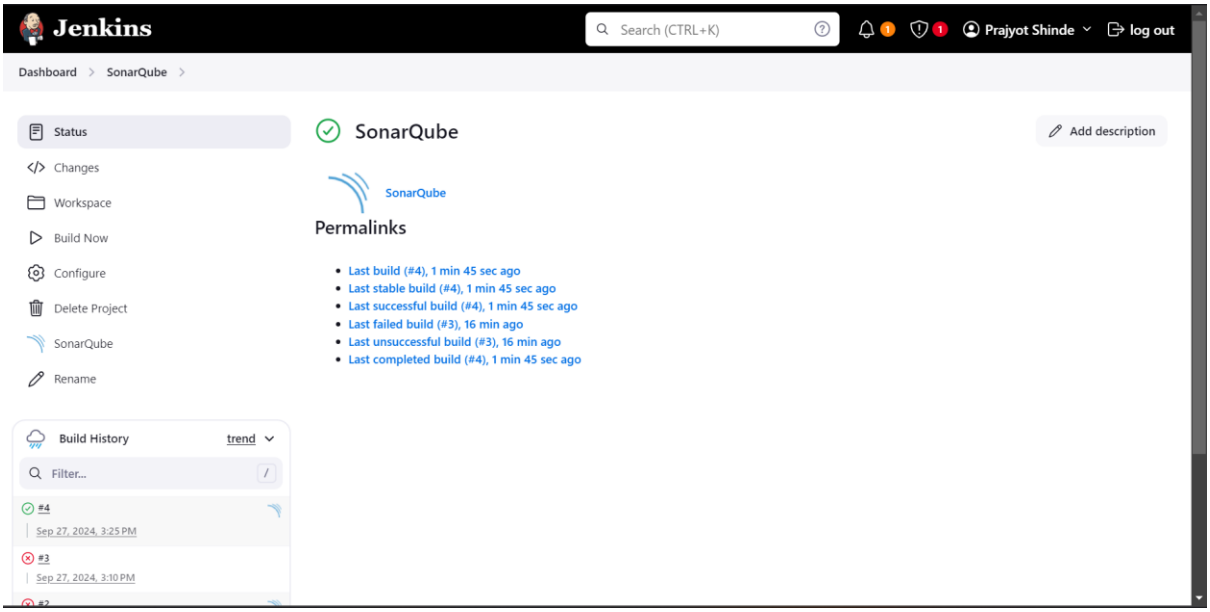

Jenkins
Search (CTRL+K)
Prajyot Shinde
log out
Dashboard > All > New Item
New Item
Enter an item name
Sonarube
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
OK

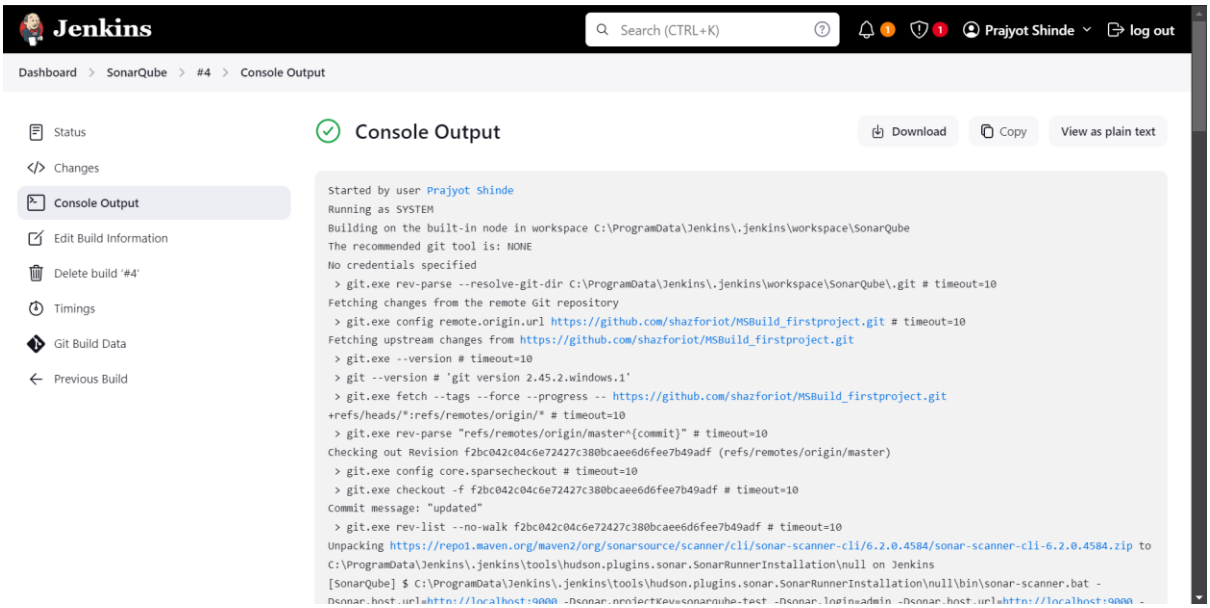

Dashboard > sonarQube > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/shazforiot/MSBuild_firstproject.git
Credentials
- none -
+ Add
Save
Apply

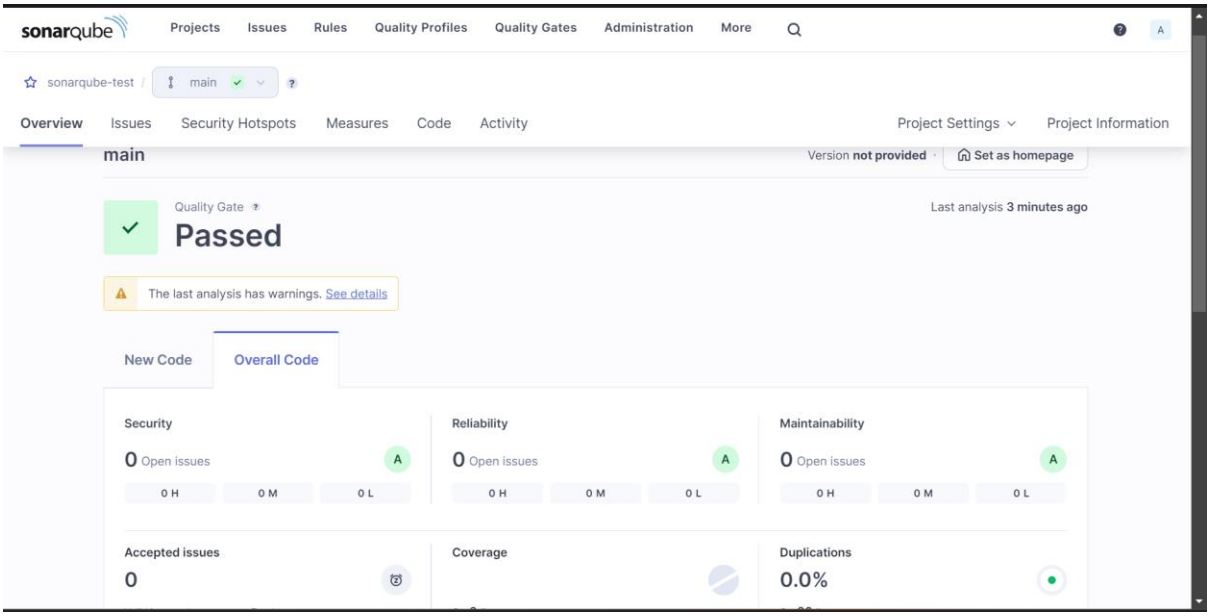

Dashboard > sonarQube2 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Path to project properties
Analysis properties
sonar.projectKey=sonarqube
sonar.login=sqp_018cb0bf9d002dc66c6c3f278248d119e5c2e0ab
sonar.sources=HelloWorldCore
sonar.host.url=http://localhost:9000
Additional arguments
JVM Options
Save
Apply


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Administration
Configuration
Security
Projects
System
Marketplace
Administer System
Administer
Execute Analysis
Create
sonar-administrators
System administrators
Quality Gates
Quality Profiles
Projects
sonar-users
Every authenticated user automatically belongs to this group
Quality Gates
Quality Profiles
Projects
Anyone DEPRECATED
Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.
Quality Gates
Quality Profiles
Projects
A
Administrator admin
Quality Gates
Quality Profiles
Projects

After 3 failures finally there was 4th success which is shown down below in image





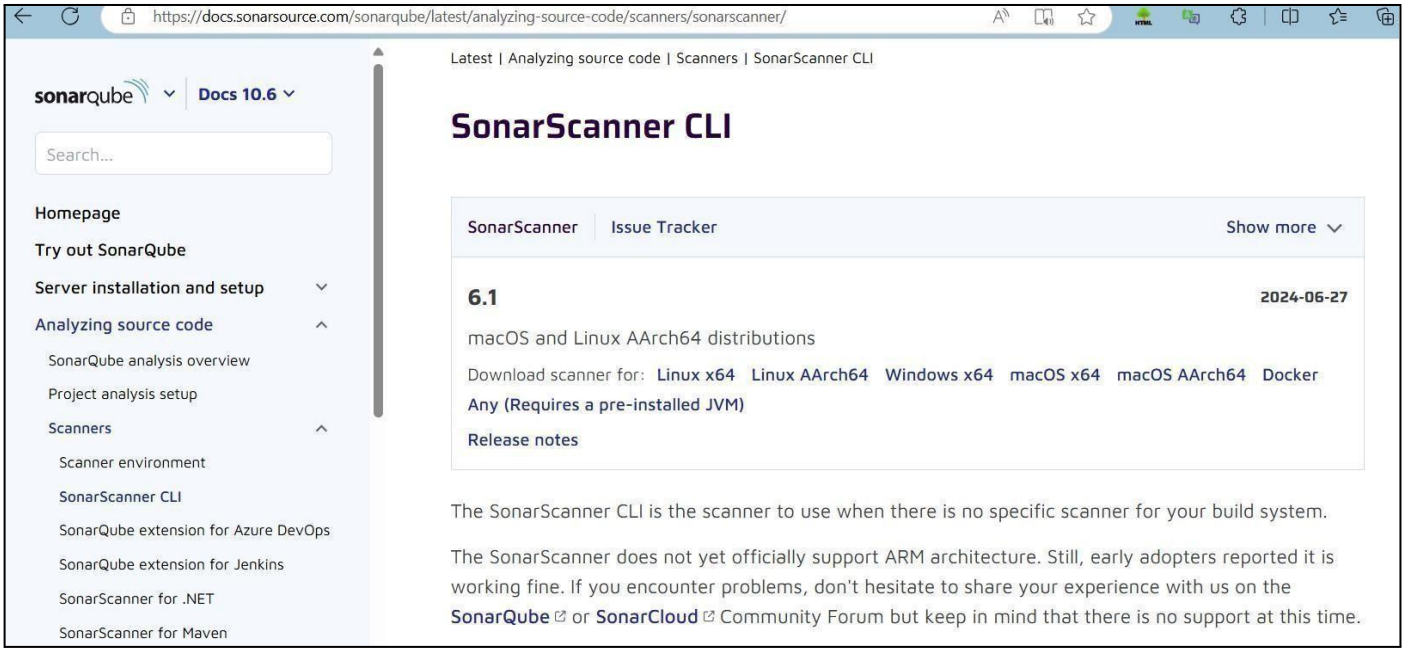

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
sonarqube-test
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
main
Version not provided
Set as homepage
Quality Gate
Passed
Last analysis 3 minutes ago
The last analysis 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
Coverage
Duplications
0.0%

ADVANCE DEVOPS EXP 8

NAME: PRAJYOT SHINDE 57 / D15A

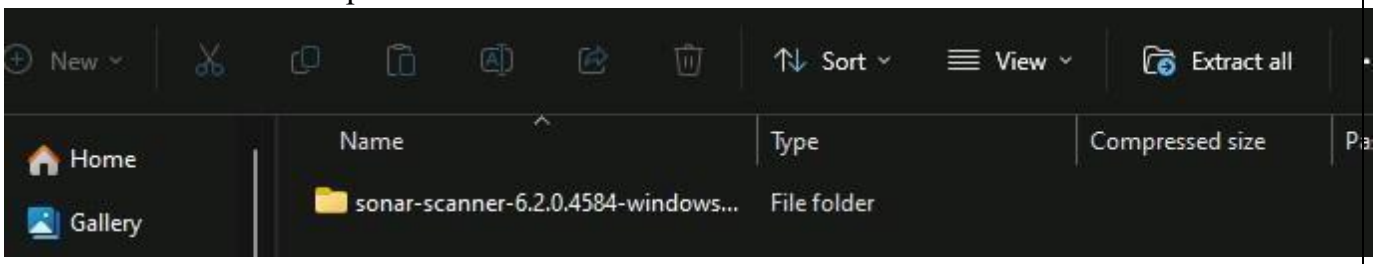
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner https://docs.sonarsource.com/sonarqube/latest/analyzingsource-code/scanners/sonarscan



ner/ Visit this link and download the sonarqube scanner CLI.

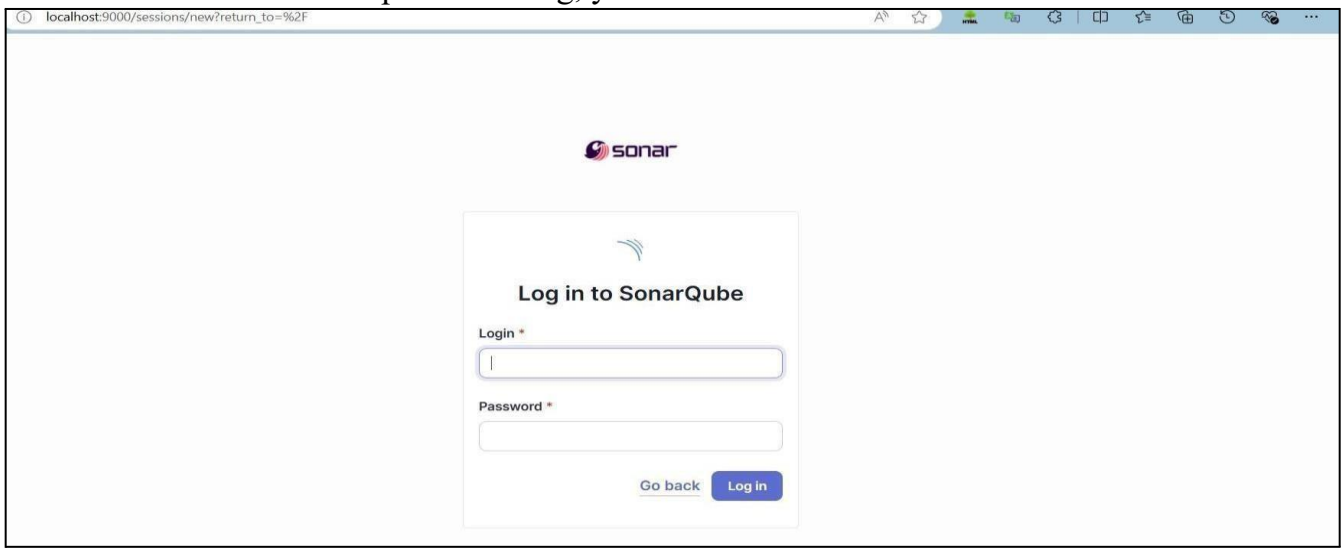
Extract the downloaded zip file in a folder.



1. Install sonarqube image Command: **docker pull sonarqube**

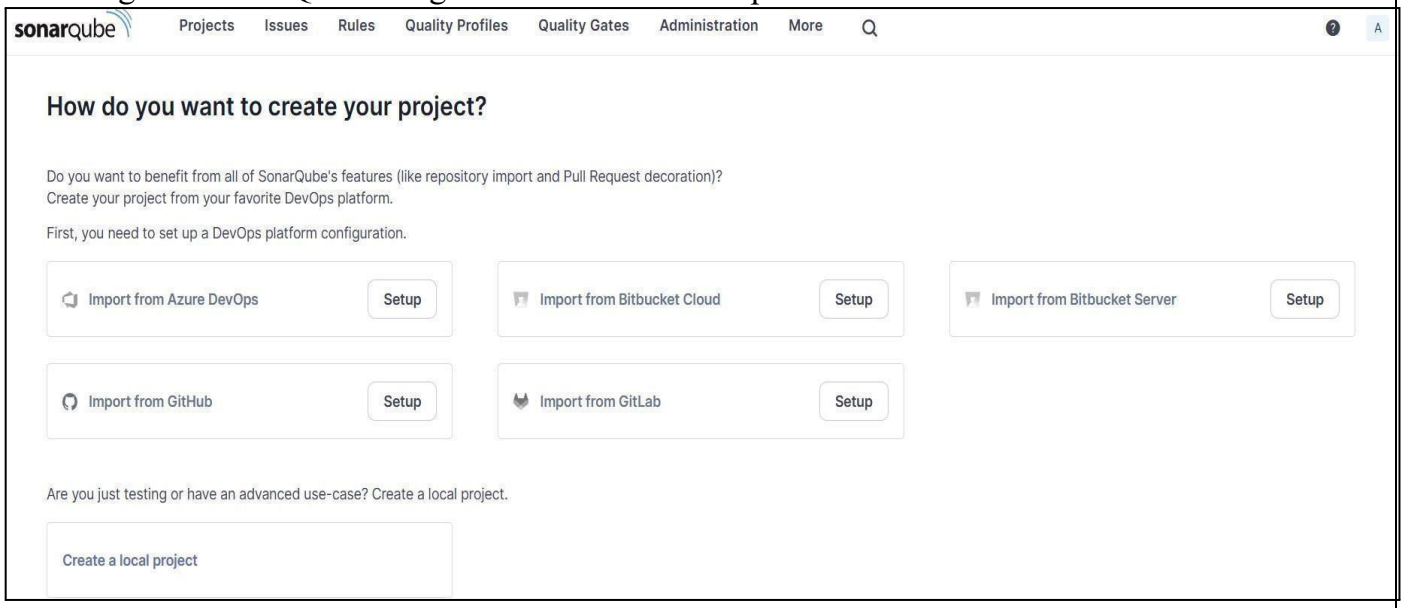
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindo

PS C:\Users\Soham Satpute> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

2. Once the container is up and running, you can check the status of

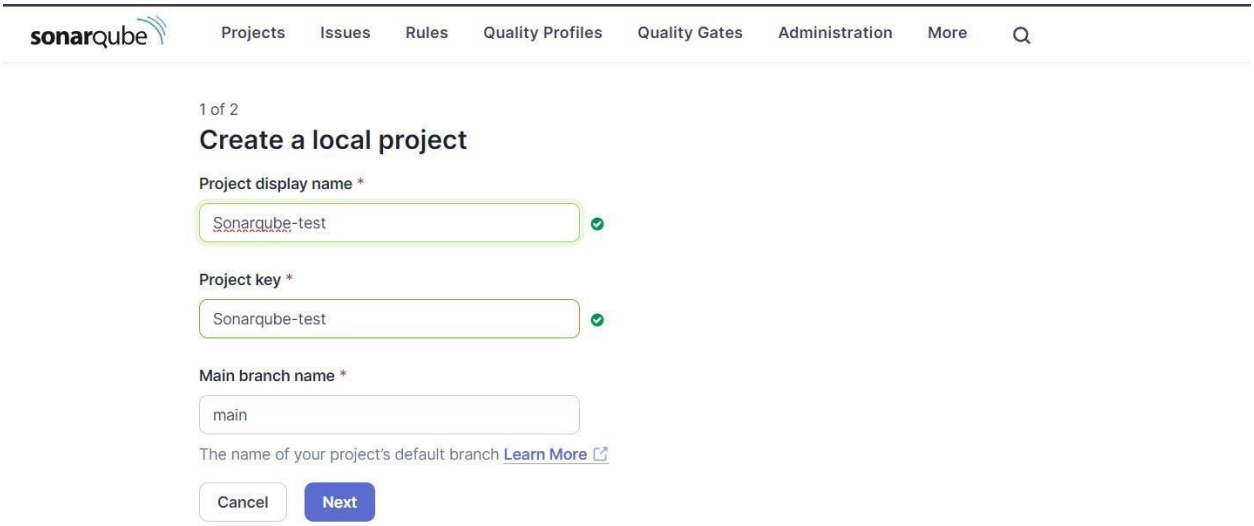


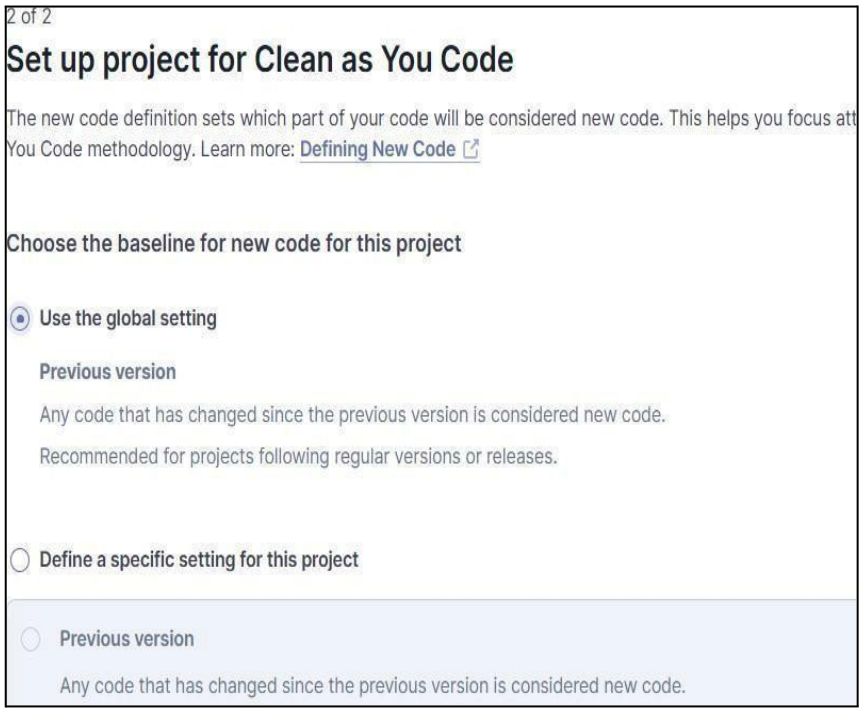
SonarQube at localhost port 9000.

3. Login to SonarQube using username admin and password admin.

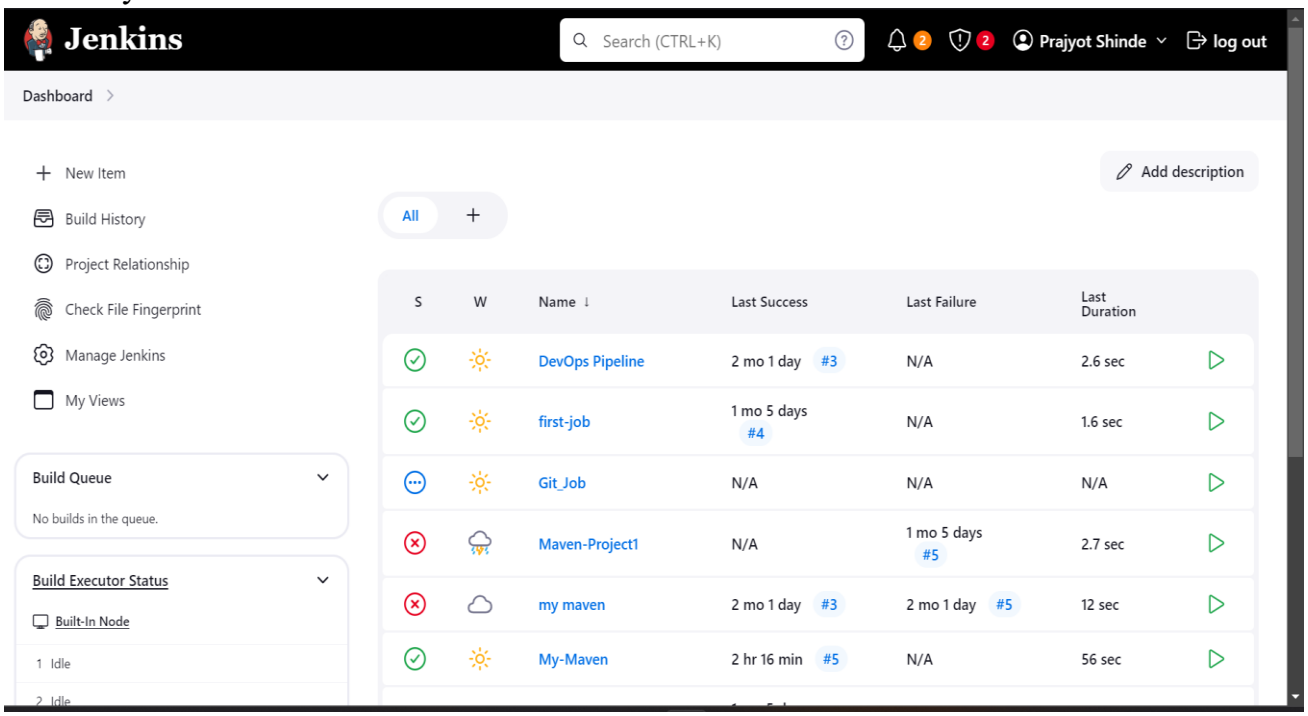


4. Create a manual project in SonarQube with the name sonarqube

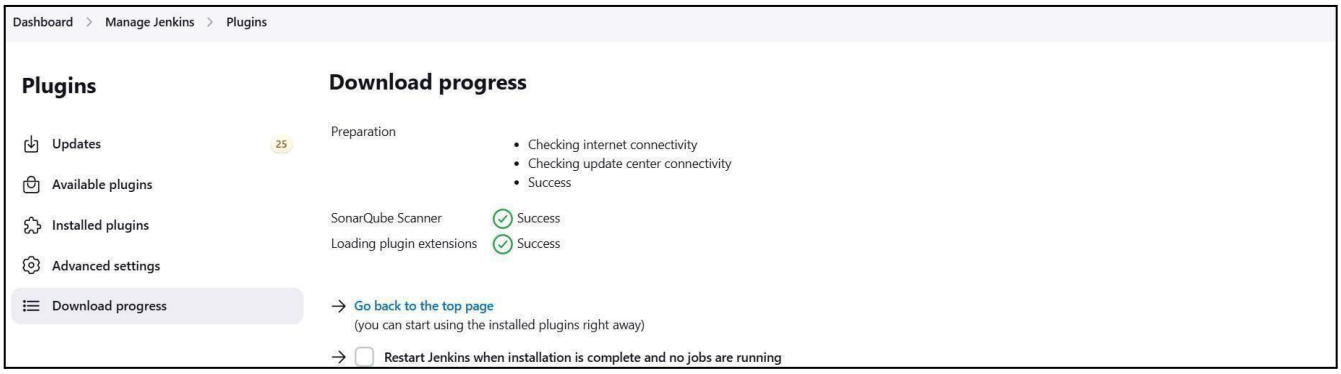

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
1 of 2
Create a local project
Project display name *
Sonarqube-test
Project key *
Sonarqube-test
Main branch name *
main
The name of your project's default branch Learn More
Cancel
Next


2 of 2
Set up project for Clean as You Code
The new code definition sets which part of your code will be considered new code. This helps you focus att
You Code methodology. Learn more: Defining New Code
Choose the baseline for new code for this project
Use the global setting
Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
Define a specific setting for this project
Previous version
Any code that has changed since the previous version is considered new code.

5. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



6. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



7.Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube**

Servers and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me

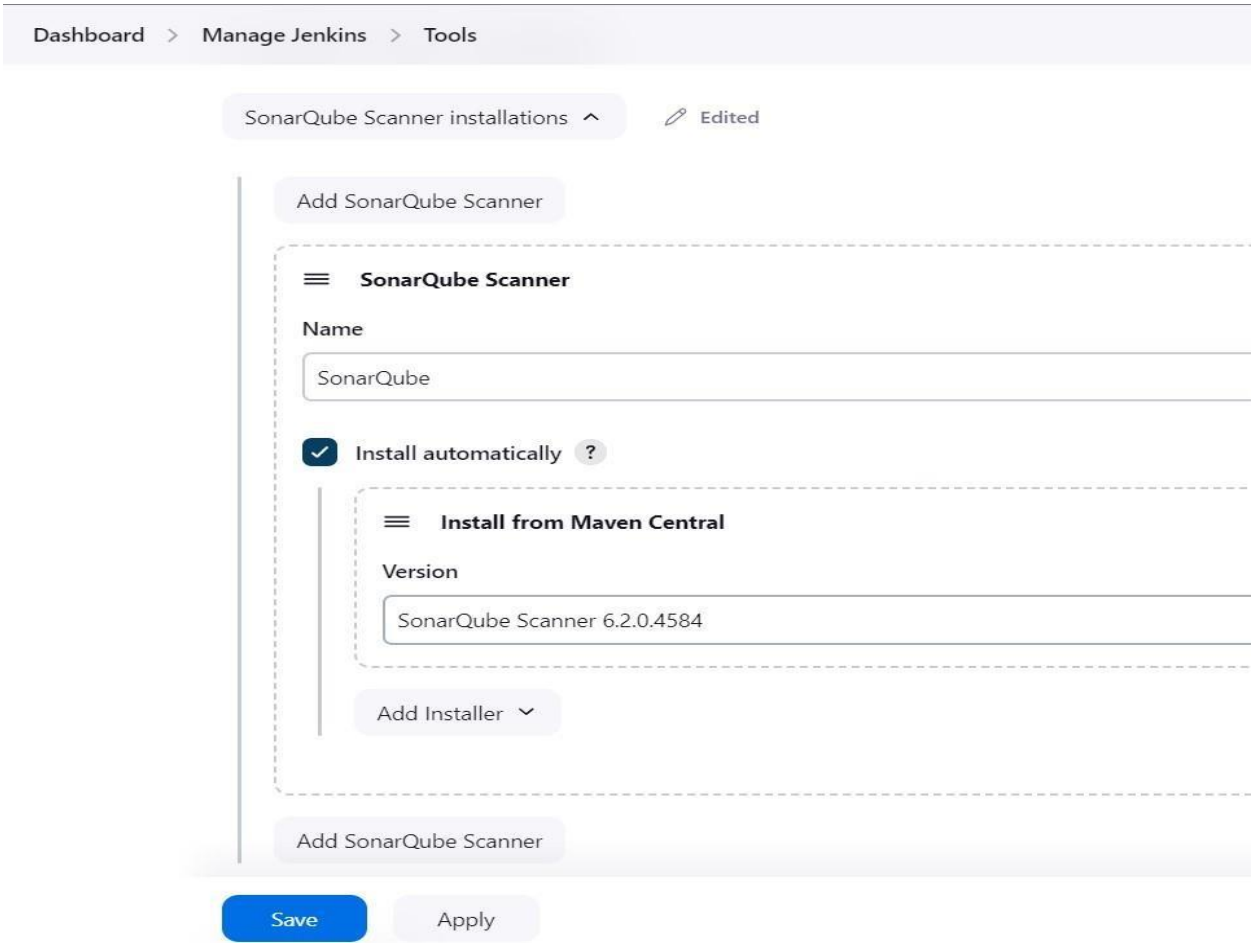
adv_devops_7_sonarqube

In **Server URL** Default is **http://localhost:9000**

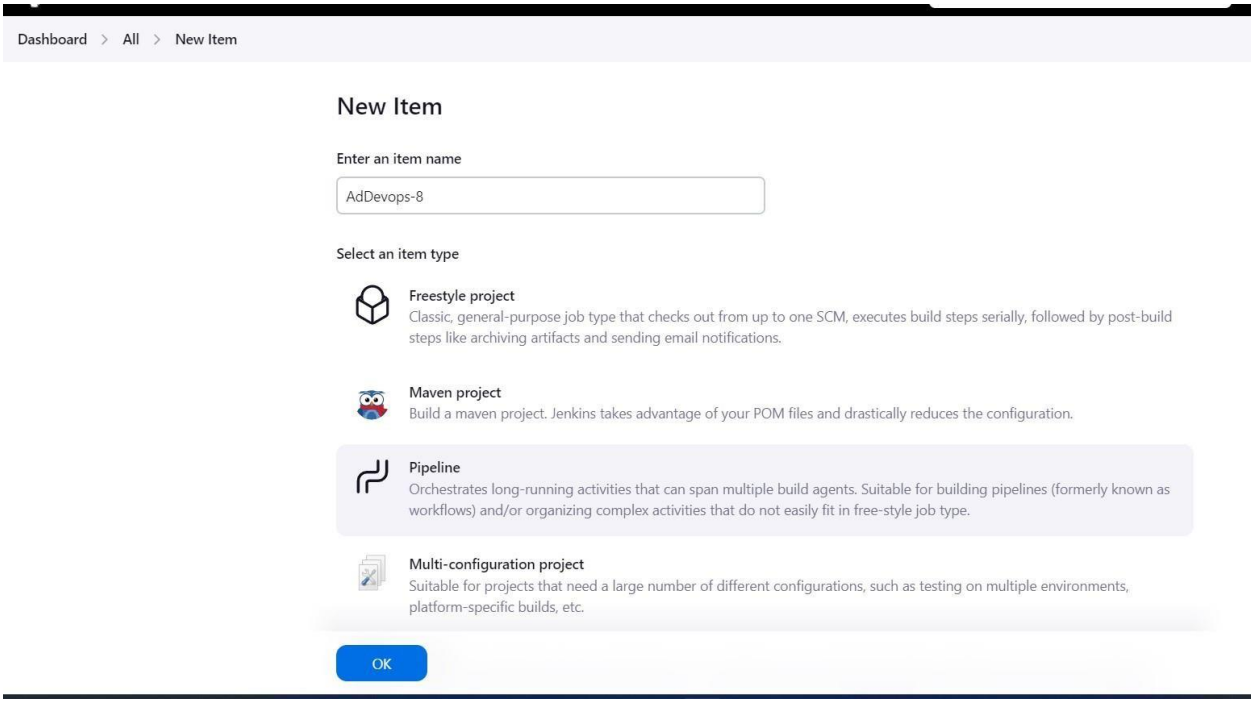
8. Search for SonarQube Scanner under Global Tool Configuration.
Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

Check the “Install automatically” option. → Under name any name as identifier → Check



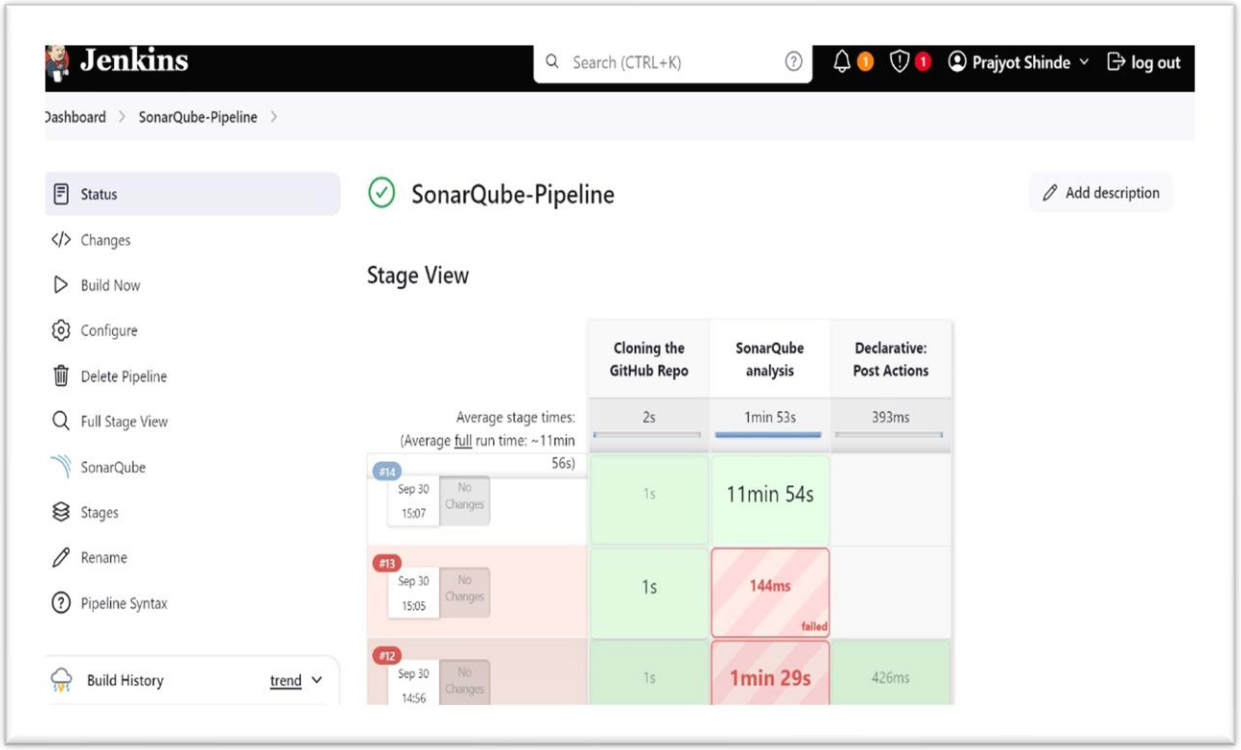
9. After configuration, create a New Item → choose a pipeline project.



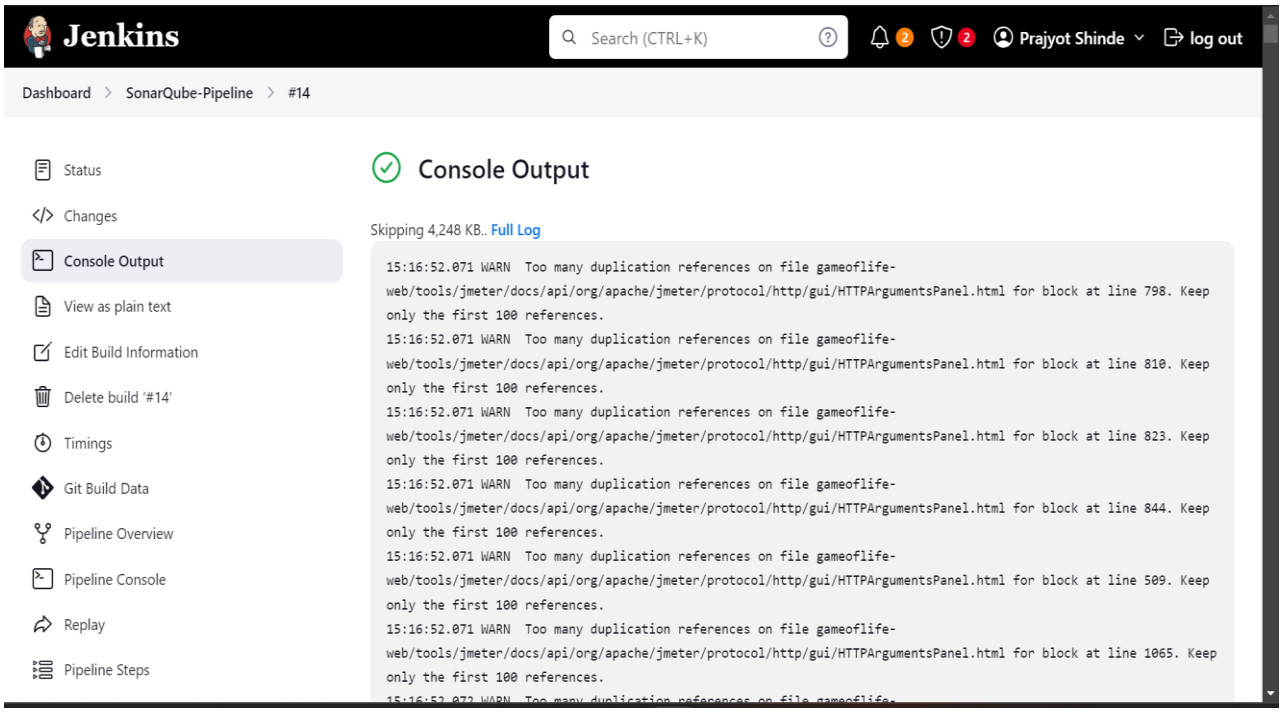
10. Under Pipeline script, enter the following:

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

  analysis') {

    withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenk
    i ns>') { sh """
        <PATH_TO_SONARQUBE_SCANNER_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=<SonarQube_URL>(default: http://localhost:9000/)
      """
    }
  }
}
```

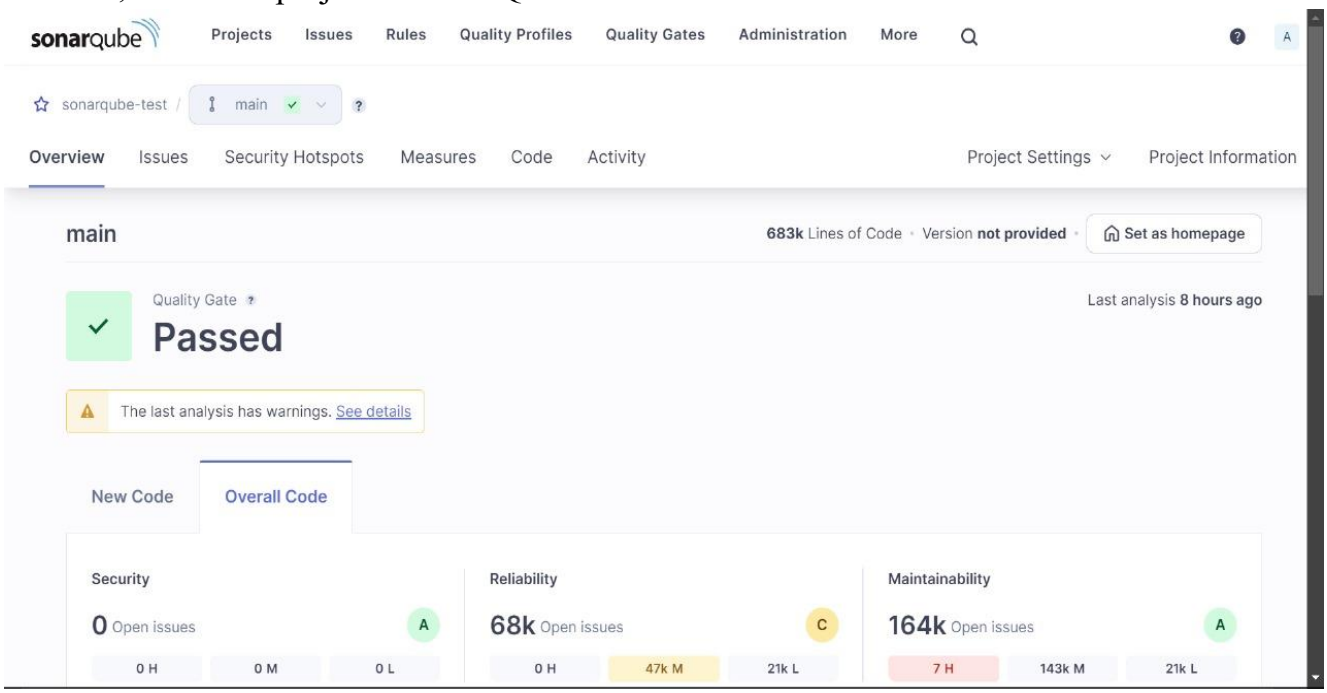
It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.



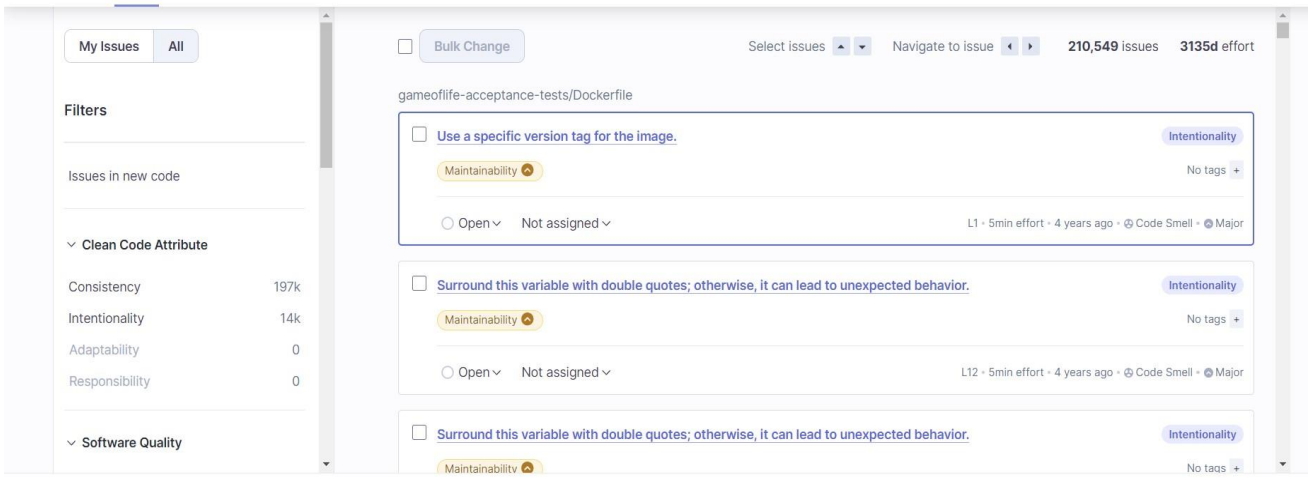
11.Check console



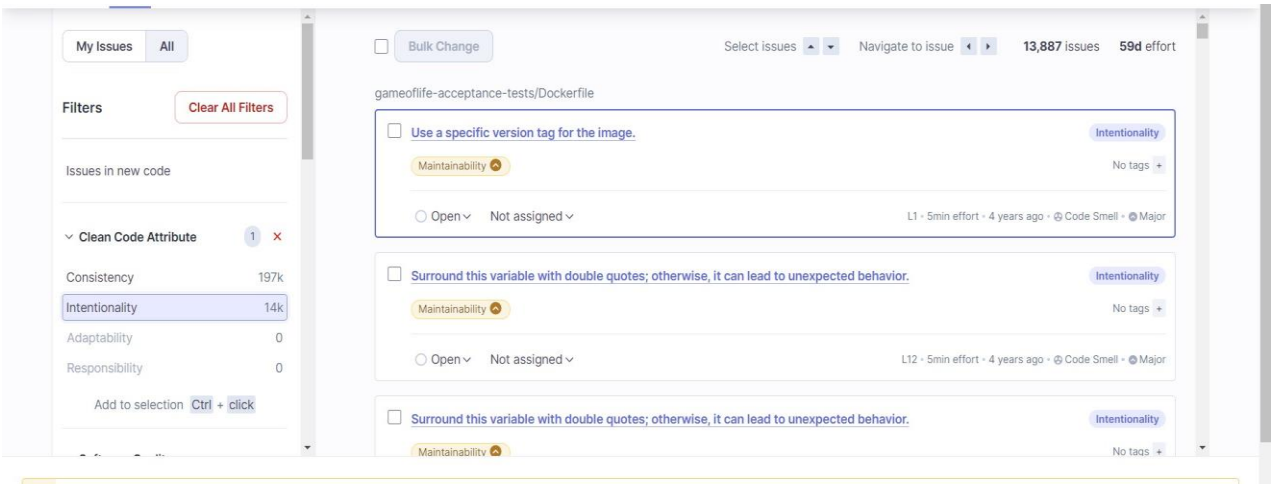
12.Now, check the project in SonarQube:



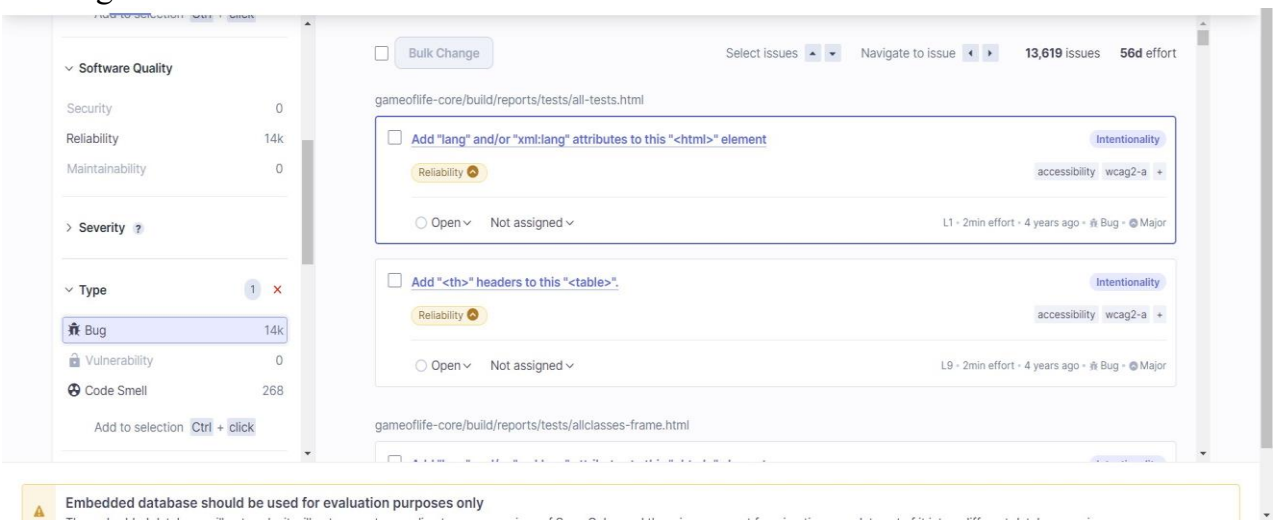
13.code problems consistency:



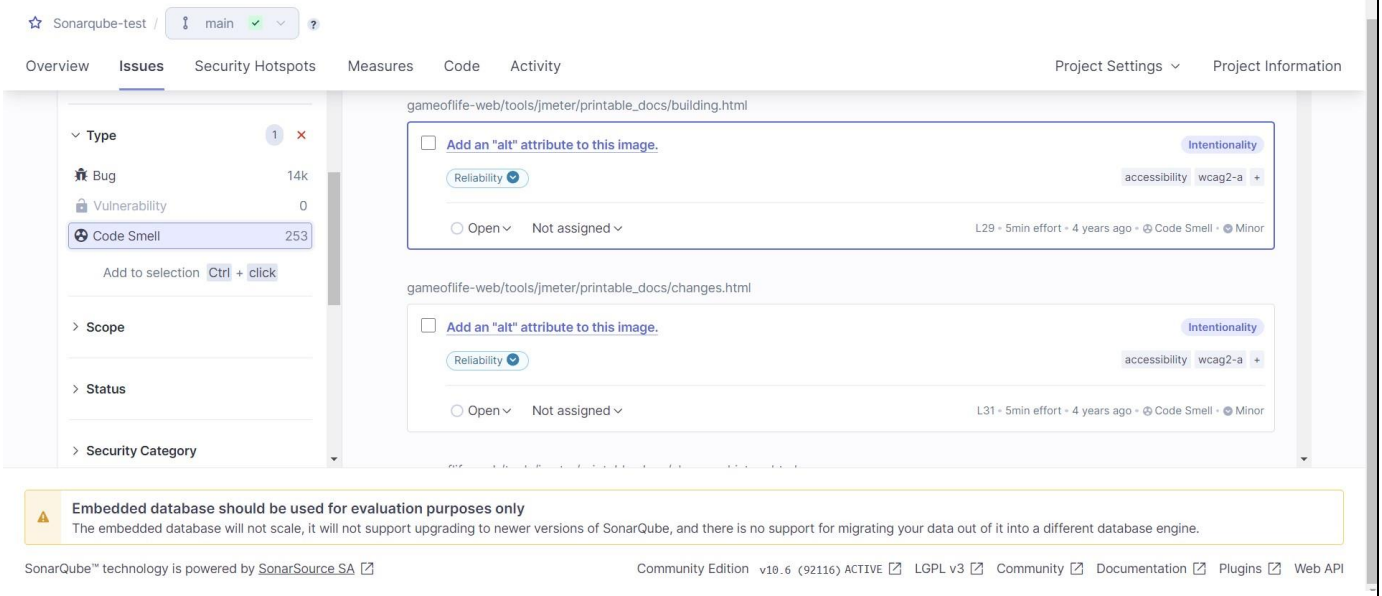
14.Intentionality:



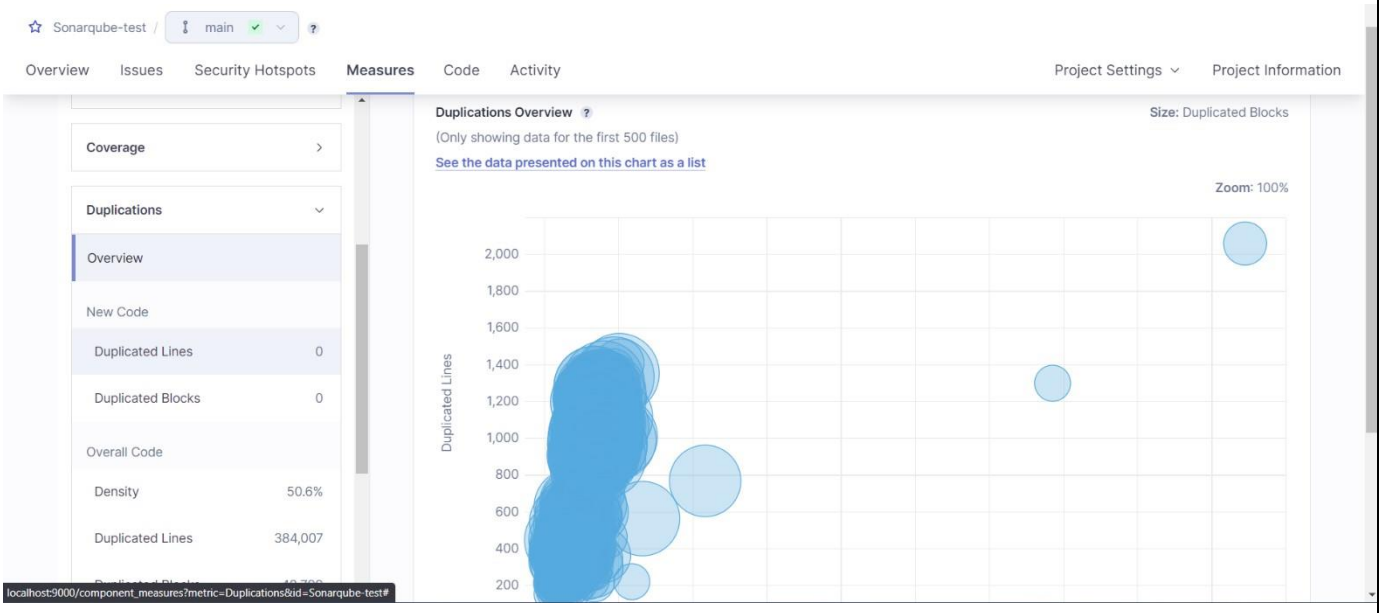
15.Bugs



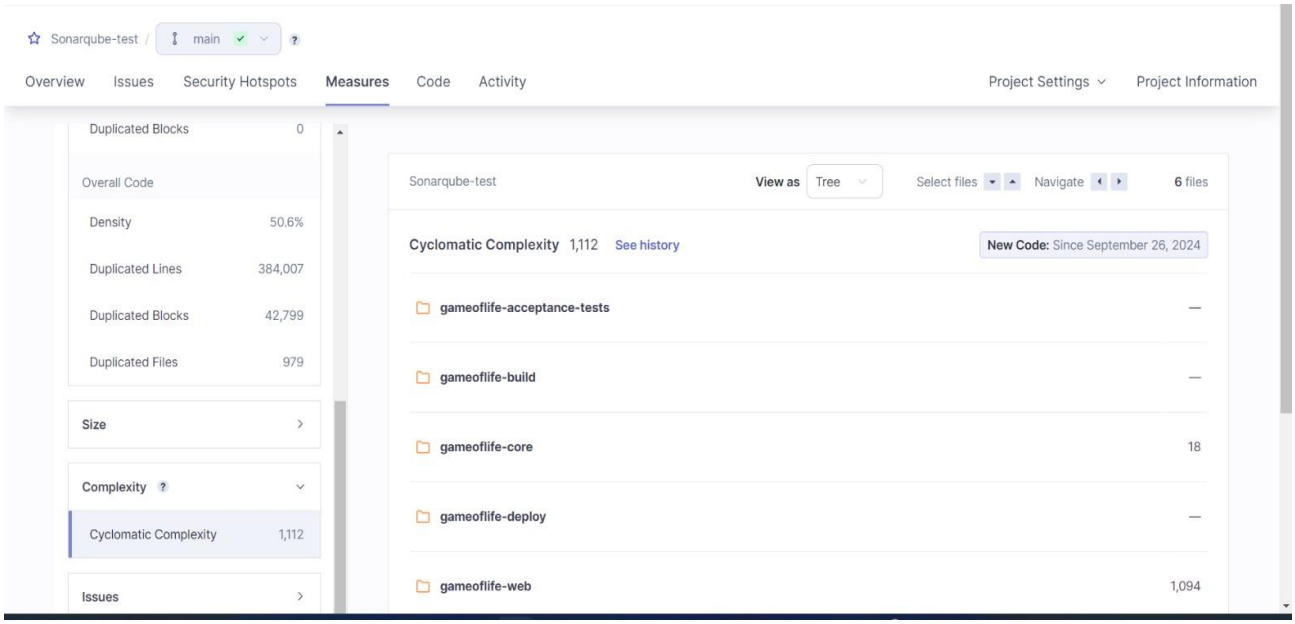
Code smells:



Duplications:



Cyclomatic Complexities:



In this way, we have integrated Jenkins with SonarQube for SAST.

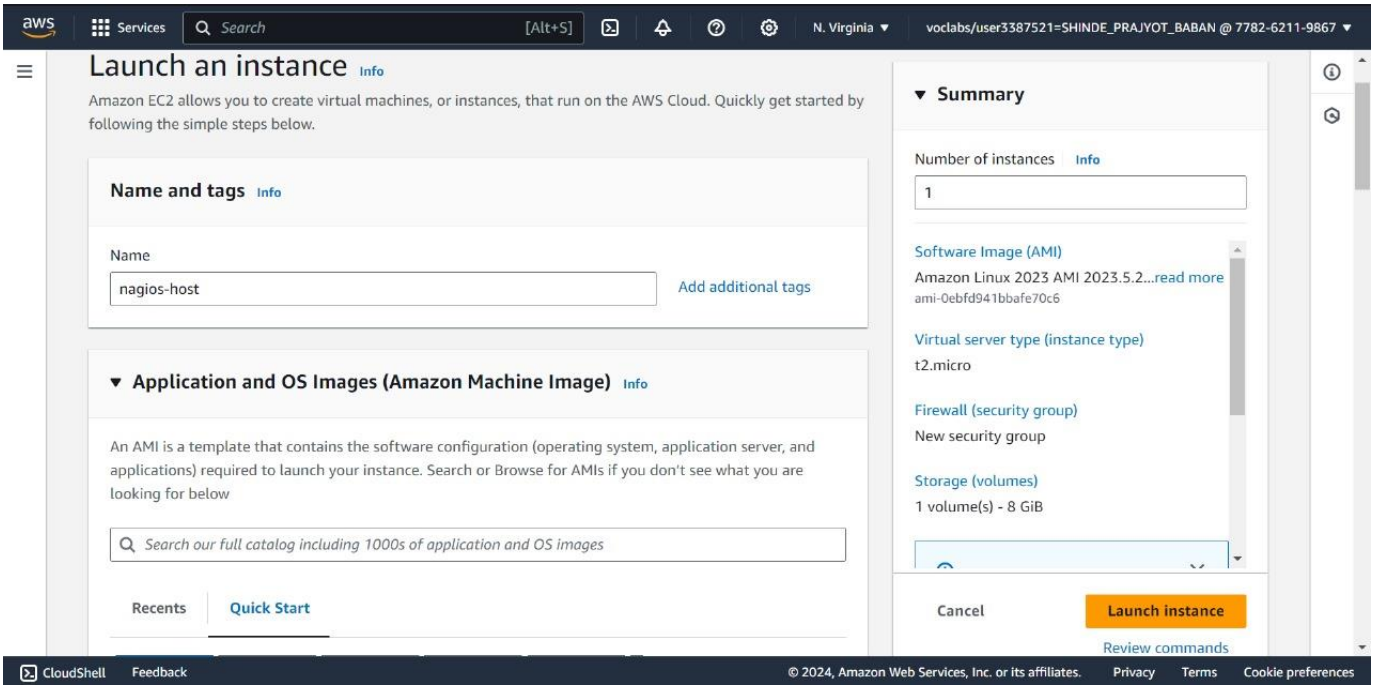
ADVANCE DEVOPS EXP 9

Name :- Prajyot Shinde

Roll no :- 57

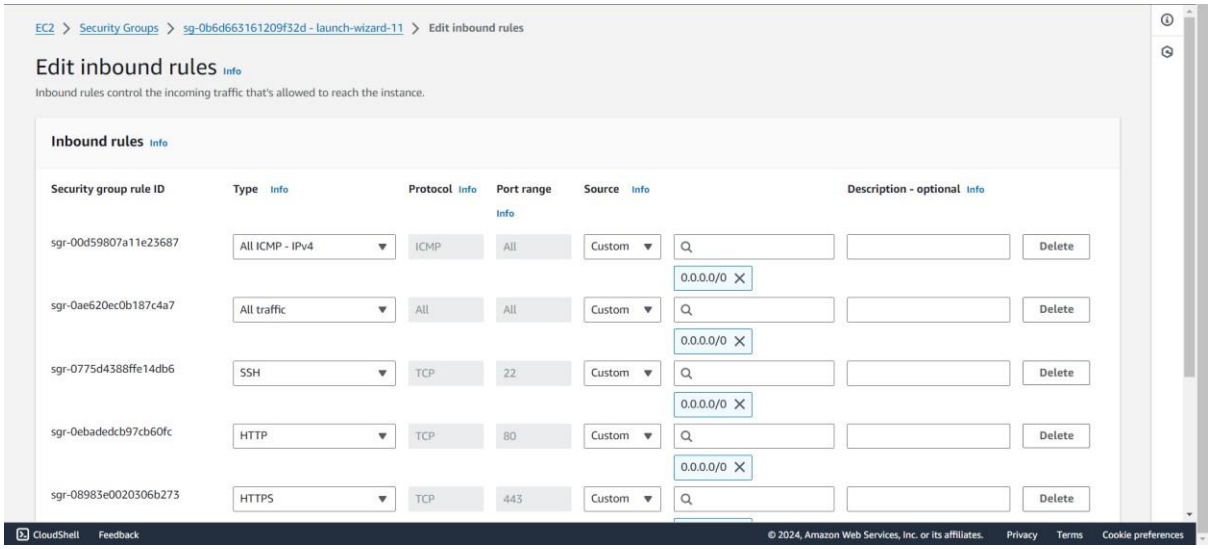
Aim :- To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

1. Create an Amazon Linux EC2 Instance



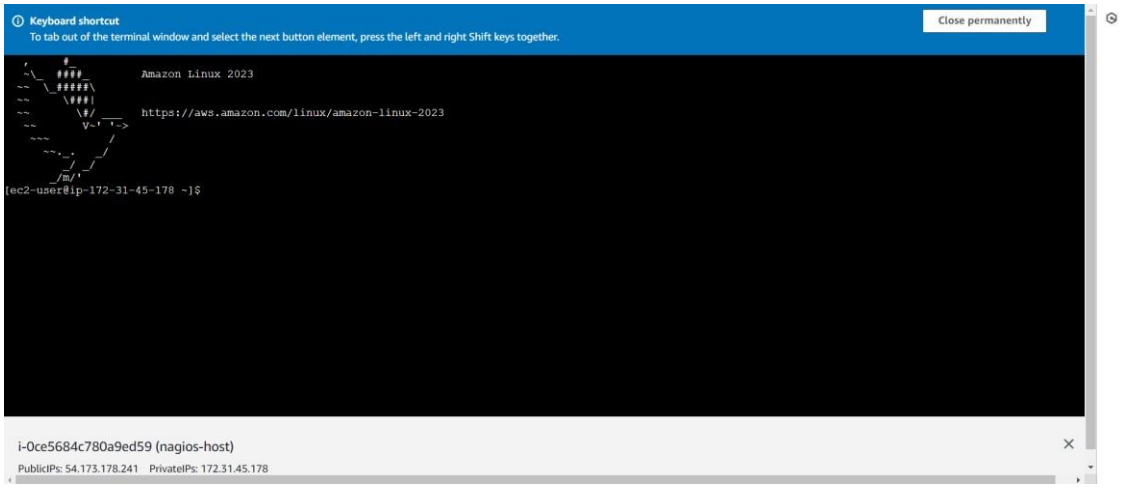
2. Configure Security Group

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

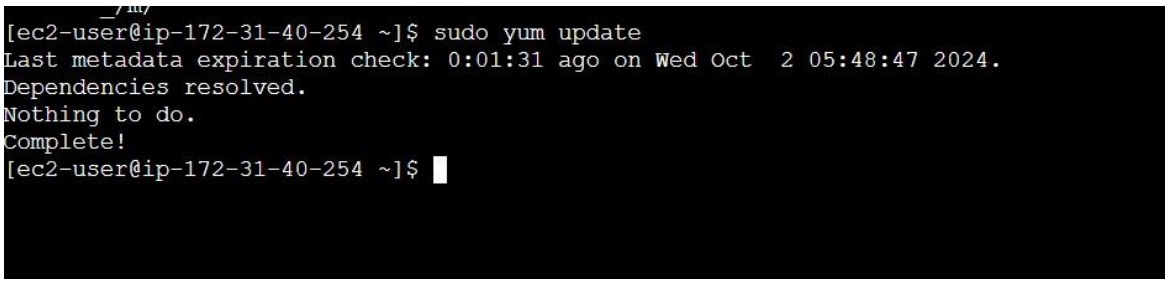


You have to edit the inbound rules of the specified Security Group for this.

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



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



sudo yum install httpd php

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:59 ago on Wed Oct  2 05:48:47 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
php8.3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 k
Installing dependencies:				
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libxslt	x86_64	1.1.34-5.amzn2023.0.2	amazonlinux	241 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.4	amazonlinux	9.8 k
php8.3-cli	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	3.7 M
php8.3-common	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	737 k
php8.3-process	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	45 k

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:41 ago on Wed Oct  2 05:48:47 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	32 M
Installing dependencies:				
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux	92 k
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux	887 k
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	10 M
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux	105 k
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux	27 k
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 k
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libxcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 k

Transaction Summary

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:46 ago on Wed Oct  2 05:48:47 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912 k
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21 k
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868 k
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404 k
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18 k

5. **Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.** sudo adduser -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-40-254 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-40-254 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-254 ~]$
```

6. **Create a new user group** sudo groupadd nagcmd

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$
```

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

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

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-40-254 ~]$
```

8. **Create a new directory for Nagios downloads** mkdir ~/downloads cd ~/downloads

```
[ec2-user@ip-172-31-40-254 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-40-254 downloads]$
```

9. **Use wget to download the source zip files.**

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

```
[ec2-user@ip-172-31-40-254 downloads]$ Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
-bash: Wget: command not found
--2024-10-02 06:15:45--  https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'

nagios-plugins-2.3.3.tar.gz              0%[                                                                                ]       0  --.-KB/s
nagios-plugins-2.3.3.tar.gz             23%[=================>                                                              ] 632.00K  3.02MB/s
nagios-plugins-2.3.3.tar.gz            100%[===================================================================================>]   2.65M  8.10MB/s    in 0.3s

2024-10-02 06:15:46 (8.10 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]

[ec2-user@ip-172-31-40-254 downloads]$
```

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

```
[ec2-user@ip-172-31-40-254 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-10-02 06:17:24--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz                      0%[                                                                                ]       0  --.-KB/s
nagios-4.4.6.tar.gz                      4%[==>                                                                             ] 495.62K  2.40MB/s
nagios-4.4.6.tar.gz                     30%[=========>                                                                      ]   3.26M  7.99MB/s
nagios-4.4.6.tar.gz                     63%[====================>                                                           ]   6.91M  11.0MB/s
nagios-4.4.6.tar.gz                     96%[==============================>                                                 ]  10.46M  12.6MB/s
nagios-4.4.6.tar.gz                    100%[===================================================================================>]  10.81M  12.9MB/s    in 0.8s

2024-10-02 06:17:25 (12.9 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

[ec2-user@ip-172-31-40-254 downloads]$
```

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

tar zxvf nagios-4.4.6.tar.gz cd
nagios-4.4.6

```
[ec2-user@ip-172-31-40-254 downloads]$ tar zxvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
nagios-4.4.6/autoconf-macros/ax_nagios_get_files
nagios-4.4.6/autoconf-macros/ax_nagios_get_inetd
nagios-4.4.6/autoconf-macros/ax_nagios_get_init
nagios-4.4.6/autoconf-macros/ax_nagios_get_os
nagios-4.4.6/autoconf-macros/ax_nagios_get_paths
nagios-4.4.6/autoconf-macros/ax_nagios_get_ssl
nagios-4.4.6/base/
nagios-4.4.6/base/.gitignore
nagios-4.4.6/base/Makefile.in
nagios-4.4.6/base/broker.c
```

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

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

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
```

12. Compile the source code.

make all

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
    inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |                 log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o commands.o commands.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o netutils.o netutils.c
netutils.c: In function 'my_tcp_connect':
netutils.c:50:47: warning: '%d' directive output may be truncated writing between 1 and 11 bytes into a region of size 6 [-Wformat-truncation=]
   50 |         snprintf(port_str, sizeof(port_str), "%d", port);
      |                                               ^~
netutils.c:50:46: note: directive argument in the range [-2147483648, 65535]
   50 |         snprintf(port_str, sizeof(port_str), "%d", port);
      |                                              ^~~~
netutils.c:50:9: note: 'snprintf' output between 2 and 12 bytes into a destination of size 6
   50 |         snprintf(port_str, sizeof(port_str), "%d", port);
      |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o notifications.o notifications.c
```

```
  make install-classicui
     - This installs the classic theme for the Nagios
       web interface


*** Support Notes *****************************************

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

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

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

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

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

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

Enjoy.
```

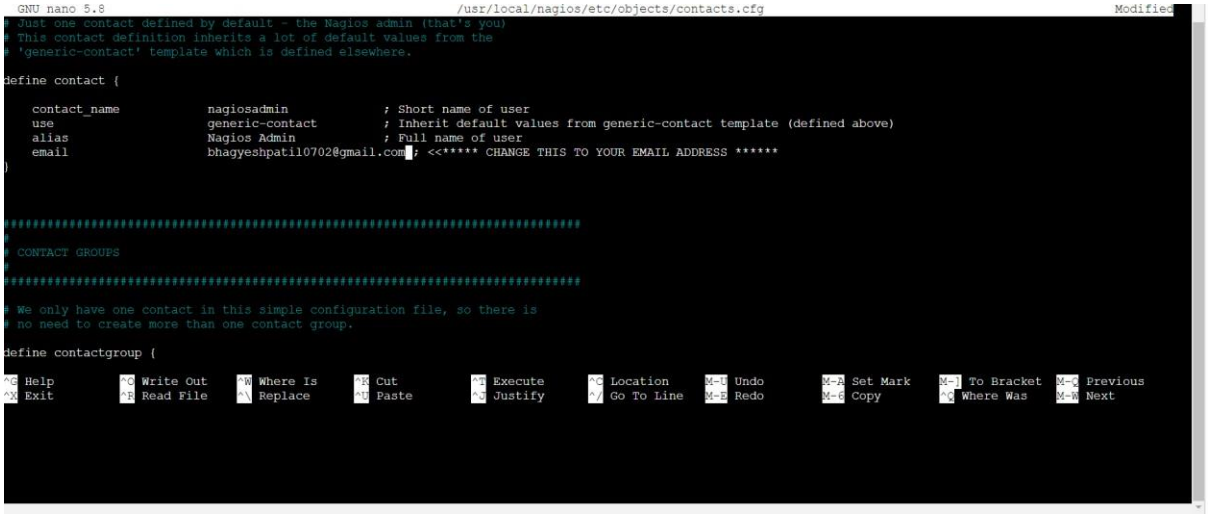
13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

./sudo make install sudo make install-init sudo make install-config sudo make install-commandmode

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
-bash: ./sudo: No such file or directory
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

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

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

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

14. Edit the config file and change the email address.

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



15. Configure the web interface.

sudo make install-webconf

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice. sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

17. Restart Apache sudo systemctl restart httpd

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

cd ~/downloads tar zxvf nagios-

plugins-2.3.3.tar.gz cd nagios-

plugins-2.3.3

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ cd ~/downloads
tar zxvf nagios-plugins-2.3.3.tar.gz
 cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.3.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/ABOUT-NLS
nagios-plugins-2.3.3/configure.ac
nagios-plugins-2.3.3/Makefile.in
nagios-plugins-2.3.3/config.h.in
nagios-plugins-2.3.3/ChangeLog
nagios-plugins-2.3.3/AUTHORS
nagios-plugins-2.3.3/lib/
nagios-plugins-2.3.3/lib/parse_ini.h
nagios-plugins-2.3.3/lib/extra_opts.c
nagios-plugins-2.3.3/lib/Makefile.in
```

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

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

group=nagios make sudo make install

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

19. Compile and install plugins sudo chkconfig --add nagios sudo
chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg sudo

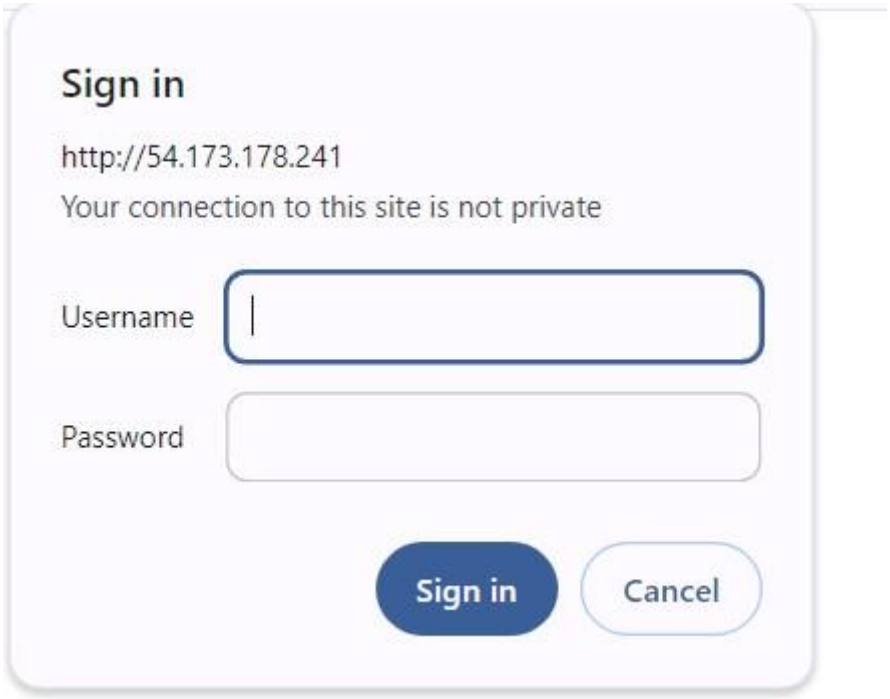
systemctl start nagios

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo systemctl start nagios
Error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.

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

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

Running pre-flight check on configuration data...

Checking objects...
       Checked 8 services.
       Checked 1 hosts.
       Checked 1 host groups.
```

20. Check the status of Nagios

```
hings look okay - No serious problems were detected during the pre-flight check
ec2-user@ip-172-31-45-178 nagios-plugins-2.3.3]$ sudo systemctl status nagios
 nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 05:37:36 UTC; 14s ago
       Docs: https://www.nagios.org/documentation
    Process: 67990 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 67991 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 67992 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.0M
        CPU: 16ms
     CGroup: /system.slice/nagios.service
             ├─67992 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67993 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67994 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67995 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67996 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67997 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

ct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
ct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: core query handler registered
```

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




Not secure
54.173.178.241/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
Nagios Core
Daemon running with PID 67992
Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates
A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.
Get Started
Start monitoring your infrastructure
Change the look and feel of Nagios
Extend Nagios with hundreds of addons
Get support
Get training
Get certified
Quick Links
Nagios Library (tutorials and docs)
Nagios Labs (development blog)
Nagios Exchange (plugins and addons)
Nagios Support (tech support)
Nagios.com (company)
Nagios.org (project)
Latest News
Don't Miss...
Page Tour
Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Advance DevOps Exp 10

Name:- Prajyot Shinde

Roll :- 57

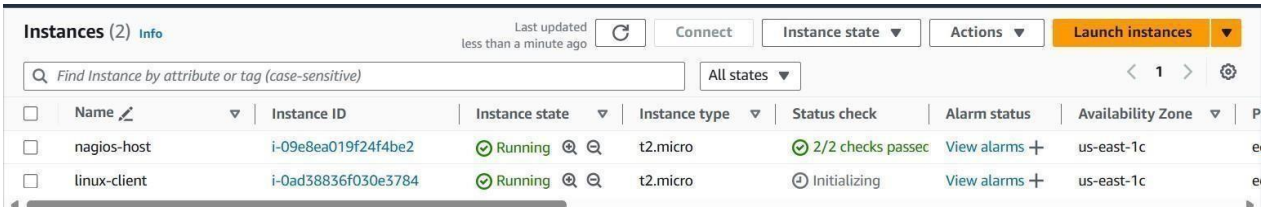
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Procedure:-

Check if the nagios service is running by executing following command sudo systemctl status nagios

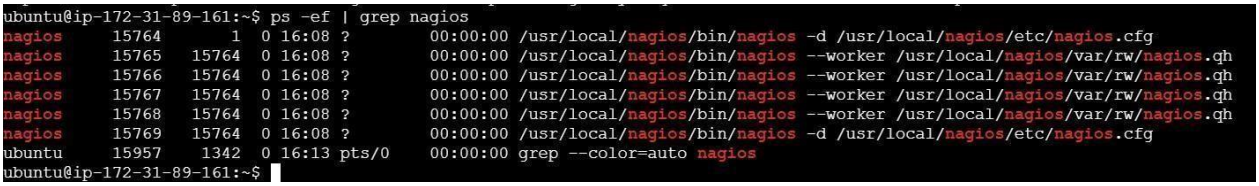
```
ubuntu@ip-172-31-89-161:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-28 16:08:58 UTC; 1min 2s ago
       Docs: https://www.nagios.org/documentation
    Process: 15743 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 15753 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 15764 (nagios)
      Tasks: 6 (limit: 1130)
     Memory: 2.4M (peak: 3.2M)
        CPU: 29ms
     CGroup: /system.slice/nagios.service
             ├─15764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─15765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─15766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─15767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─15768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─15769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: core query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: echo service query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: help for the query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15765;pid=15765
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15766;pid=15766
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15767;pid=15767
```

Now, create a new EC2 instance on AWS



Now perform the following commands on nagios-host EC2 instance.On the server, run this command

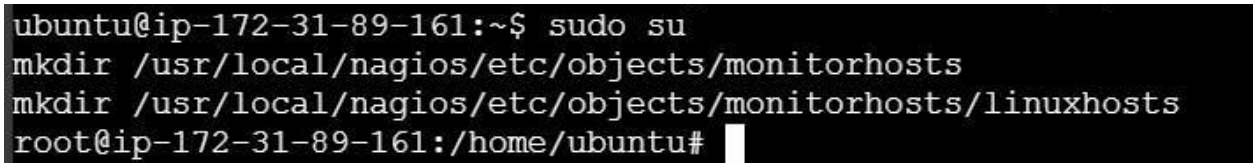
ps -ef | grep nagios



Sudo su

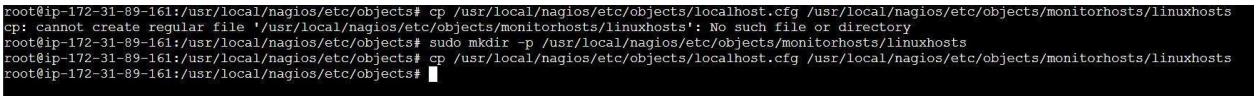
mkdir /usr/local/nagios/etc/objects/monitorhosts

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



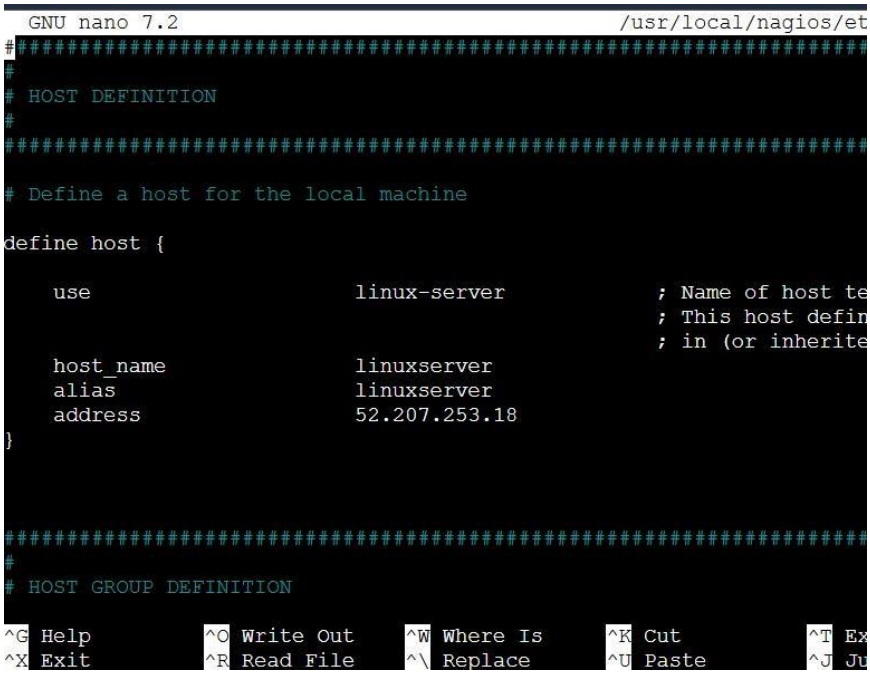
Copy localhost.cfg file to the mentioned location

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



Open the nano editor for localhost.cfg file and make these changes. Add the Ip address of the linux-client for the address field.

nano/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/localhost.cfg



Note - Here replace hostname with linuxserver

nano /usr/local/nagios/etc/nagios.cfg

Add the following line to the nagios.cfg file

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

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

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


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

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

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

After making the changes in nagios.cfg file now check validate the file by typing the following command in the terminal.

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

```
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
        Checked 16 services.
        Checked 2 hosts.
        Checked 2 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 2 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts#
```

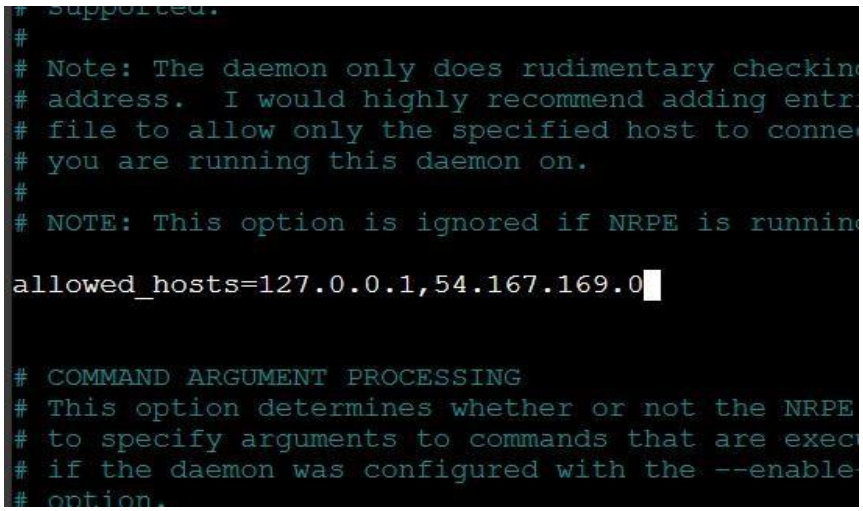
Now restart the service by using this command

service nagios restart

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# service nagios restart
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-28 17:36:35 UTC; 19s ago
       Docs: https://www.nagios.org/documentation
    Process: 1870 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 1872 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 1874 (nagios)
      Tasks: 8 (limit: 1130)
     Memory: 3.0M (peak: 3.2M)
        CPU: 24ms
     CGroup: /system.slice/nagios.service
             ├─1874 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─1875 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1876 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1877 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1878 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1879 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─1880 /usr/local/nagios/libexec/check_ping -H 52.207.253.18 -w 3000.0,80% -c 5000.0,100% -p 5
             └─1881 /usr/bin/ping -n -U -w 30 -c 5 52.207.253.18

Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: core query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: echo service query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: help for the query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Registry request: name=Core Worker 1875;pid=1875
lines 1-26
```

Now using this command update the apt repository of ubuntu (linux-client), install gcc, nagios-nrpe-server and nagios-plugin sudo apt update -y sudo apt install gcc -y sudo apt install -y nagios-nrpe-server nagios-plugins

Now open nrpe.cfg file and add the ip address of the nagios host as shown. To open the nrpe.cfg file copy this command.

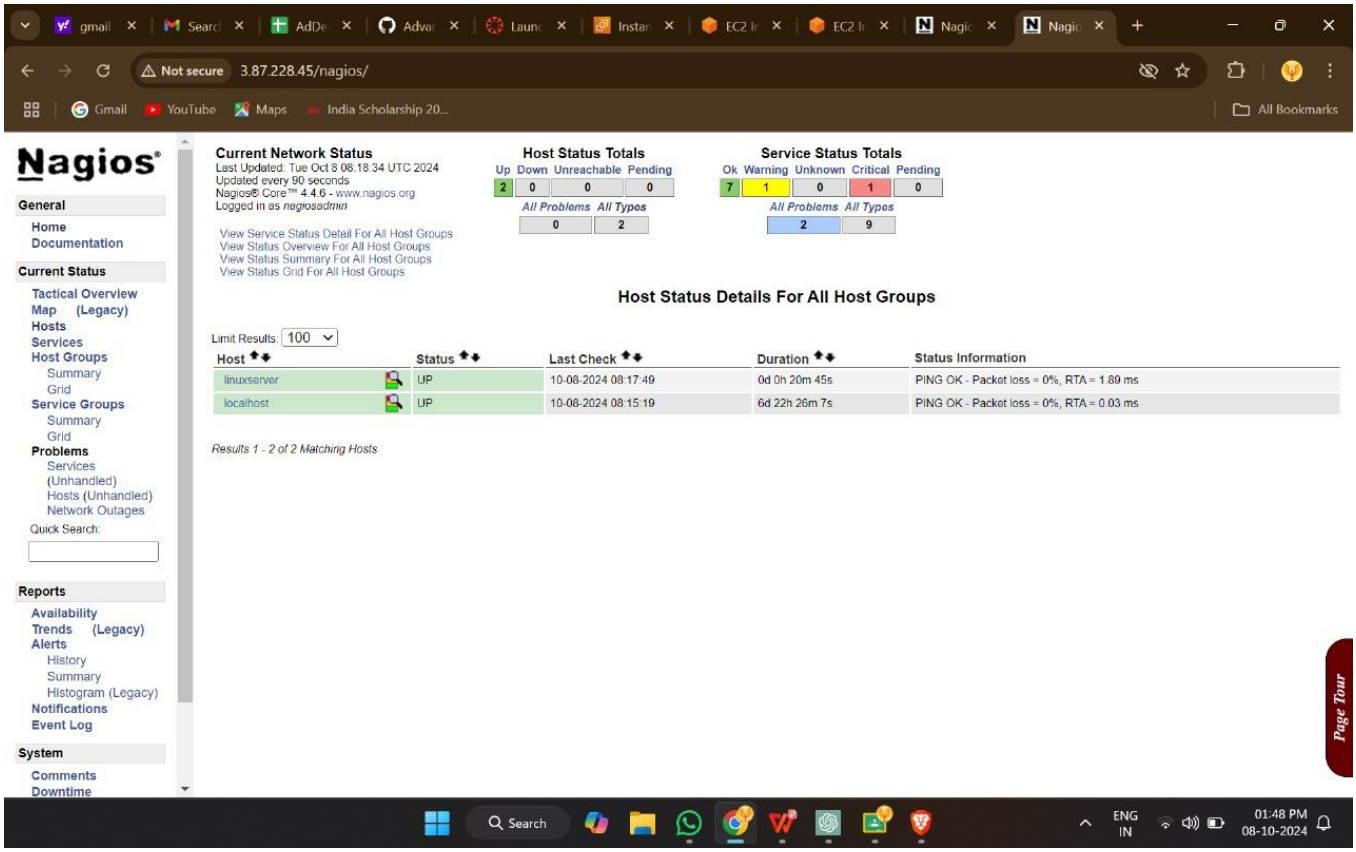


sudo nano /etc/nagios/nrpe.cfg

Now restart nrpe server by using this command

sudo systemctl restart nagios-nrpe-server

Now, check nagios dashboard, you should see linuxserver up and running, if not



Experiment 11

PRAJYOT SHINDE 57 D15A

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

AWS Lambda

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

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

Features of AWS Lambda

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

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket.

And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

Execution Model

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

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code.

AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

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

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

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

Common Use Cases for Lambda

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

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

Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to

the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

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

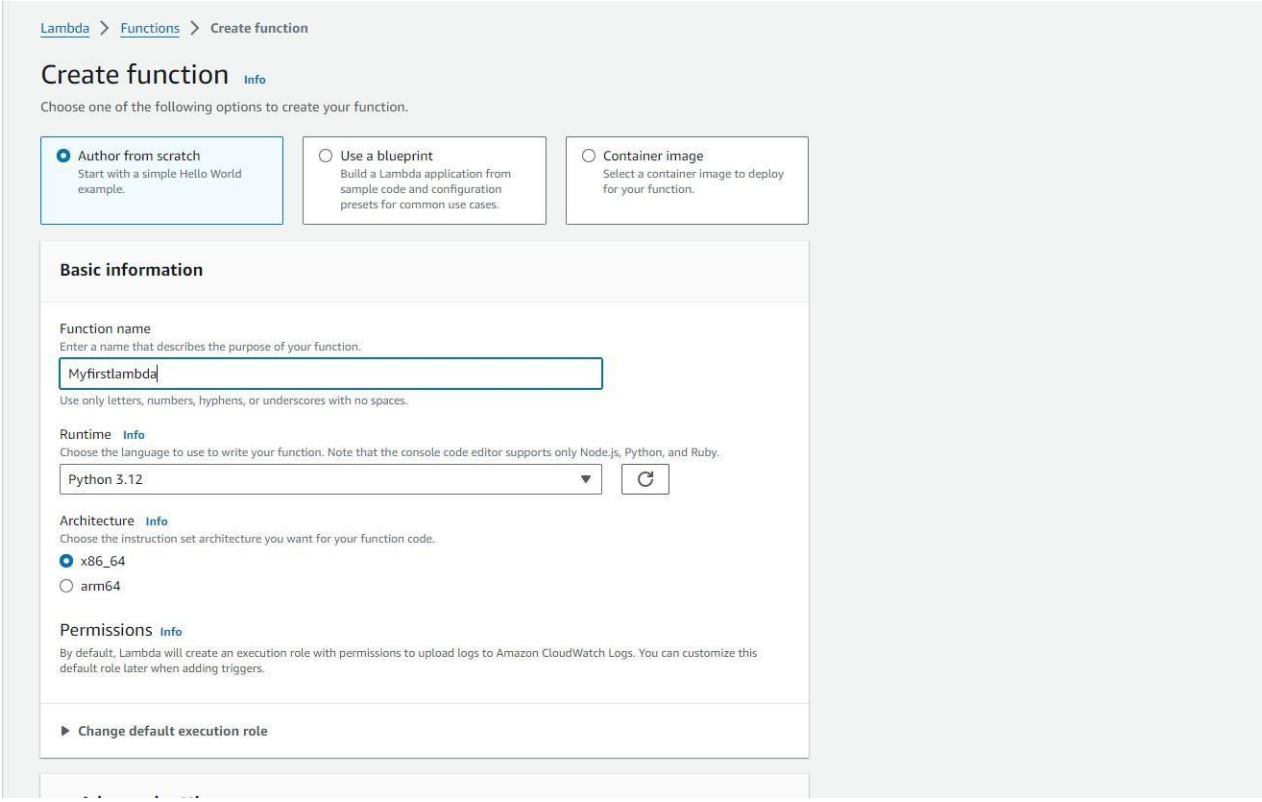
Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1.Choose a Function Creation Method:
Select Author from scratch.

2. Configure the Function:
Function name: Enter a name for your function (e.g., MyFirstLambda).
Runtime: Choose Python 3.x (the latest available version).
Permissions:Choose Create a new role with basic Lambda permissions (this creates a role with the necessary permissions).

3. Click on Create function.



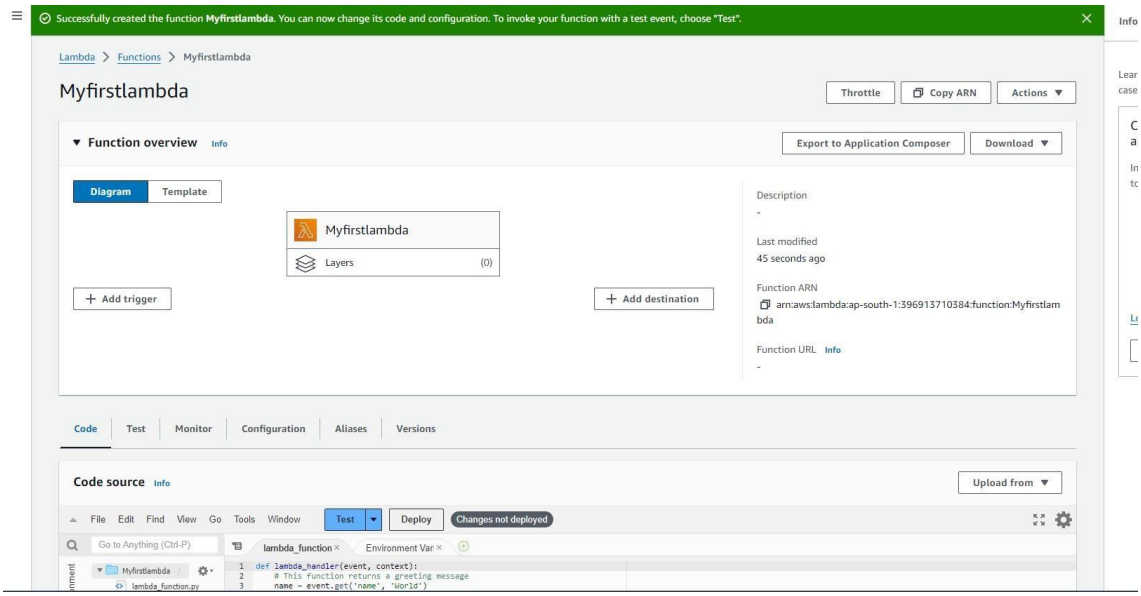
Step 2: Write Your Lambda Function Code

In the Function code section, you will see a code editor. Replace the default code with the following Python code:

python Copy code def lambda_handler(event, context): # This function returns a greeting message name = event.get('name', 'World')

```
    return {
        'statusCode': 200, 'body': f'Hello,
        {name}!'
    }
```

This function reads a name from the event and returns a greeting message. If no name is provided, it defaults to "World".



Step3: 1. Configure a Test Event:

Click on the Test button.

In the Configure test event dialog, give your event a name (e.g., TestEvent). Replace the default JSON with the following:

```
{
   "name": "Lambda User"
}
```

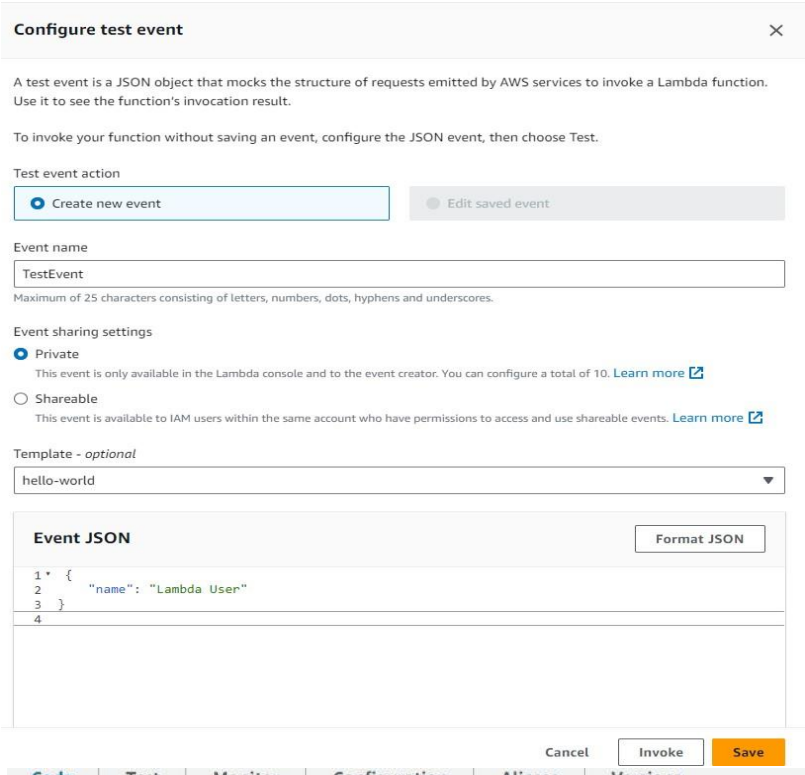
2. Run the Test:

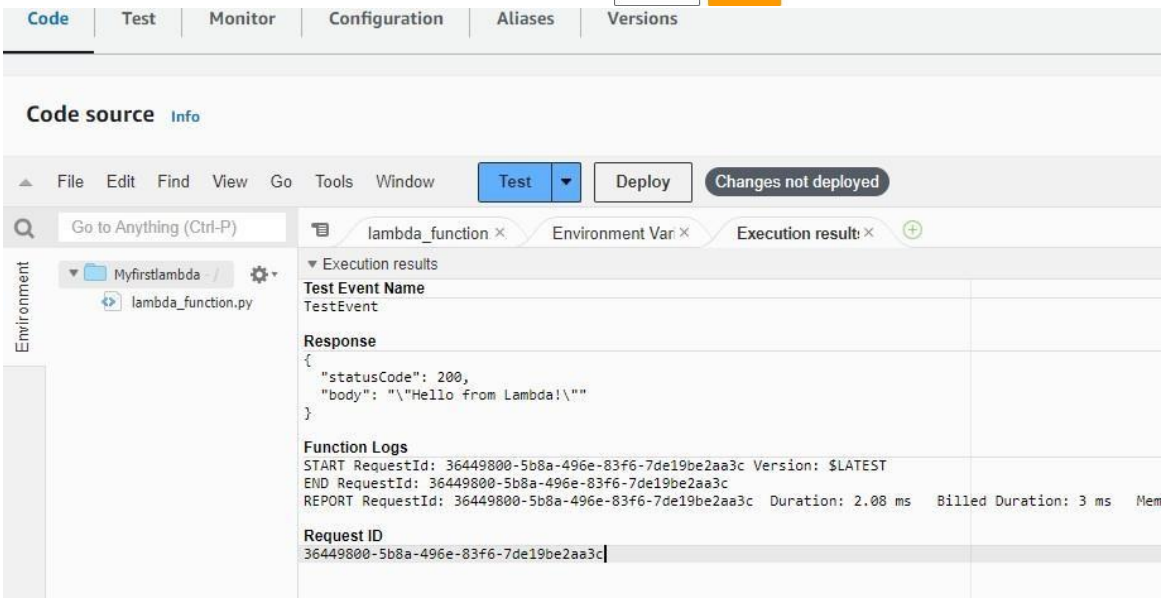
Click on the Test button again to execute your Lambda function.

You should see the execution results below the code editor, including the response: json

Copy code

```
{
   "statusCode": 200,
   "body": "Hello, Lambda User!"
}
```





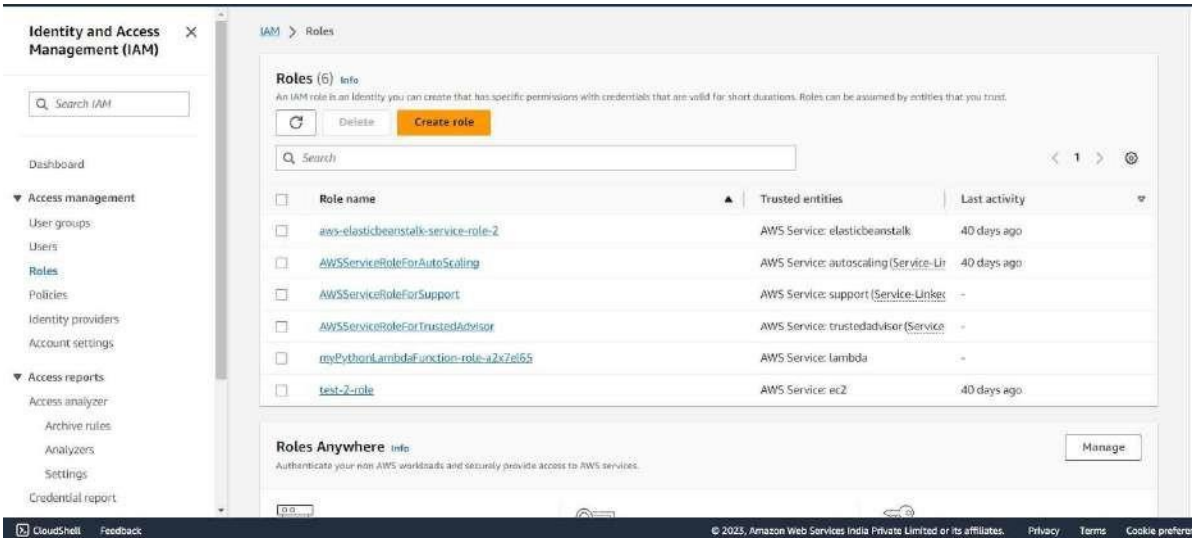
Conclusion:

AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database updates, and scales based on the demand

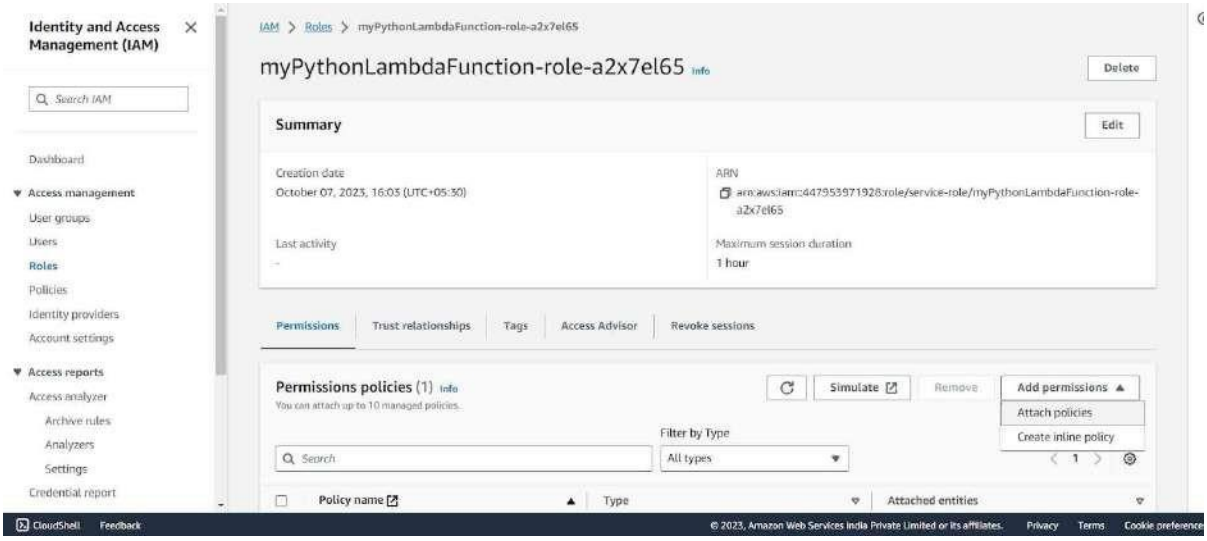
Adv. DevOps Exp. 12

Prajyot Shinde
D15A - 57

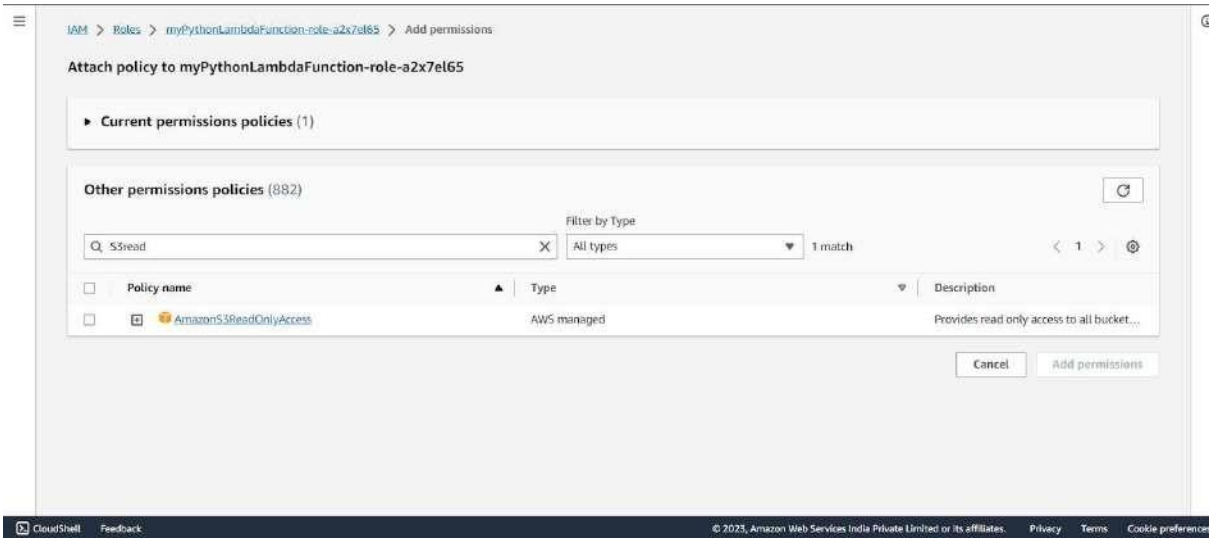
Step 1: Open the IAM (user)



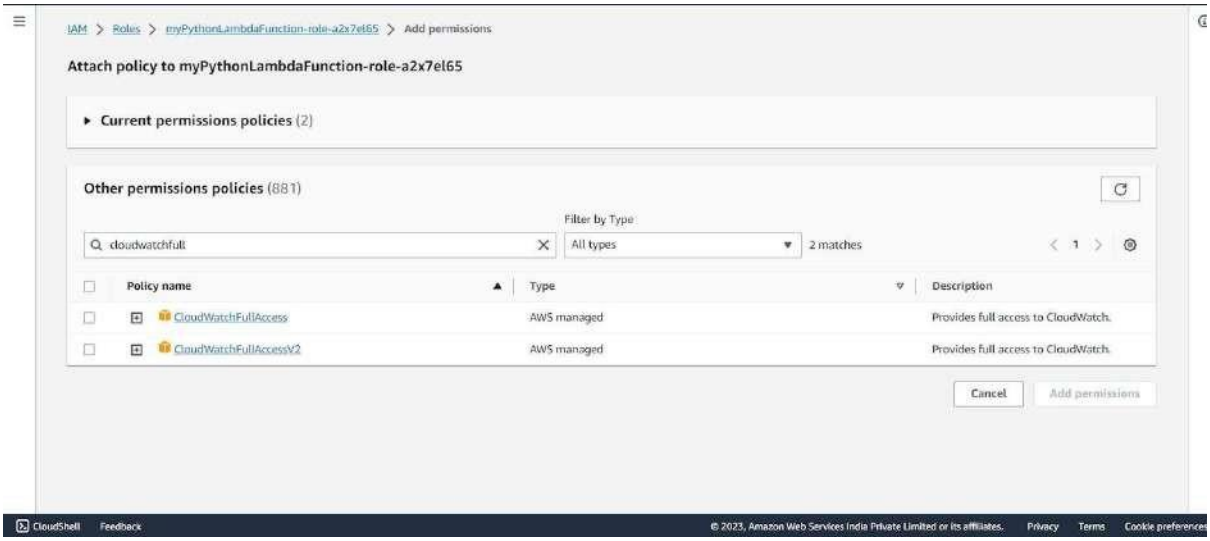
Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.



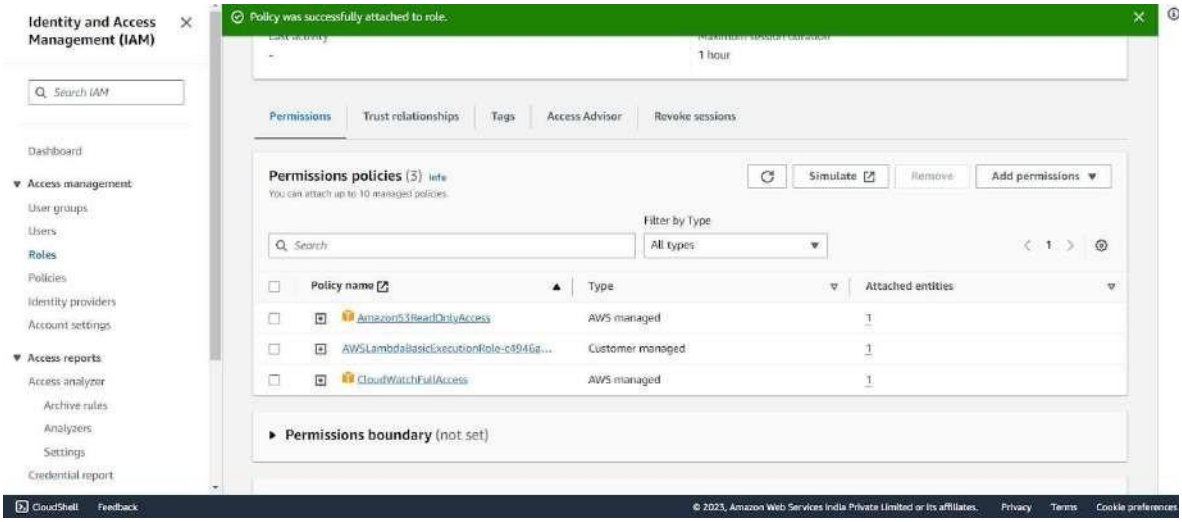
S3-ReadOnly



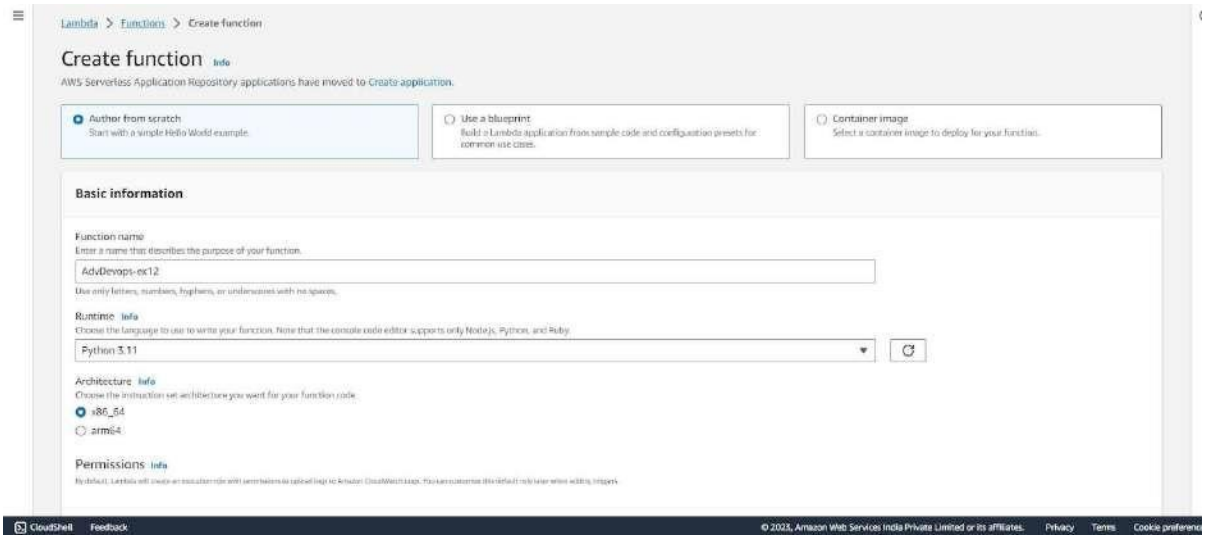
CloudWatchFull



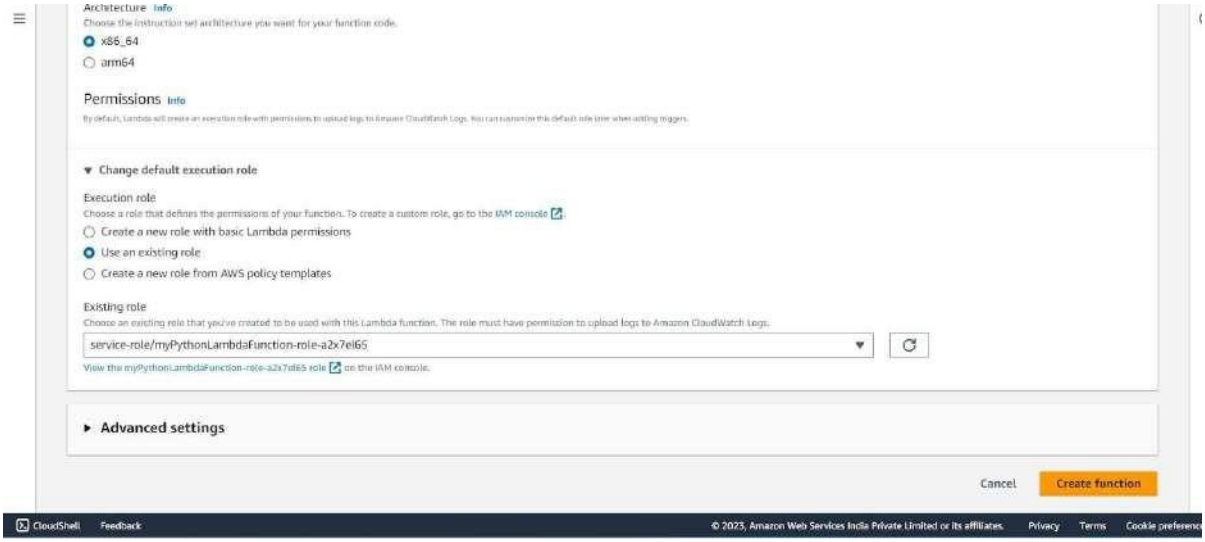
After successful attachment of policy you will see something like this you will be able to see the updated policies.



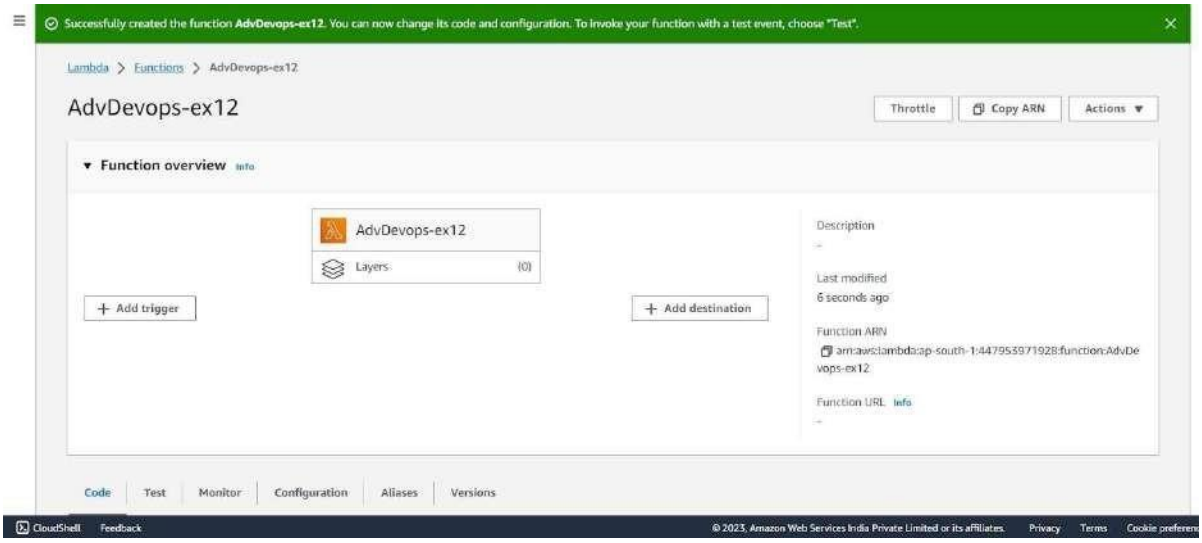
Step 3: Open up AWS Lambda and create a new Python function.



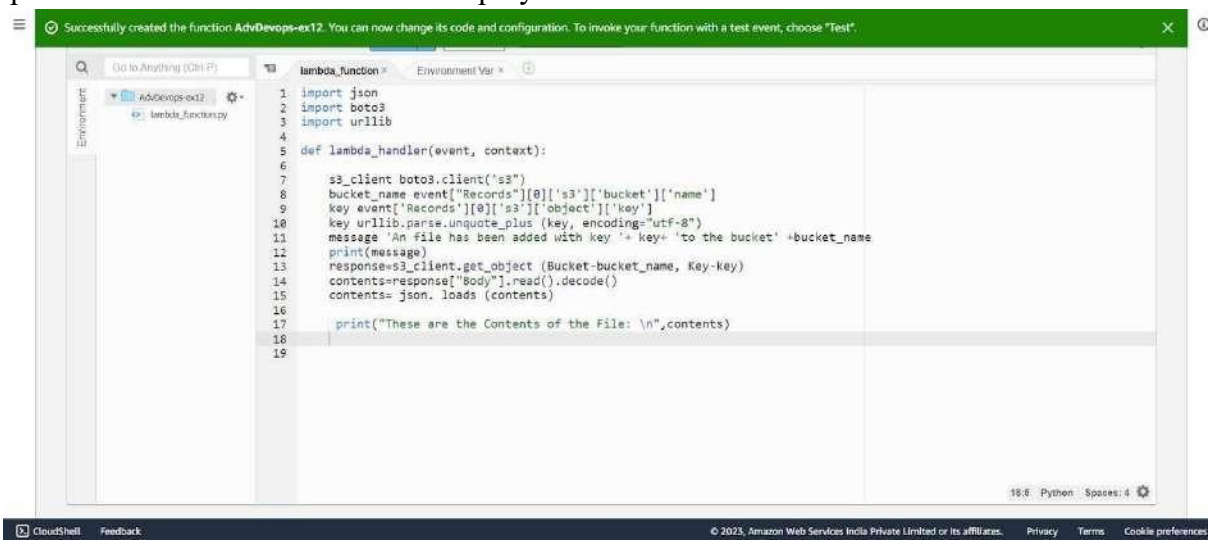
Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.



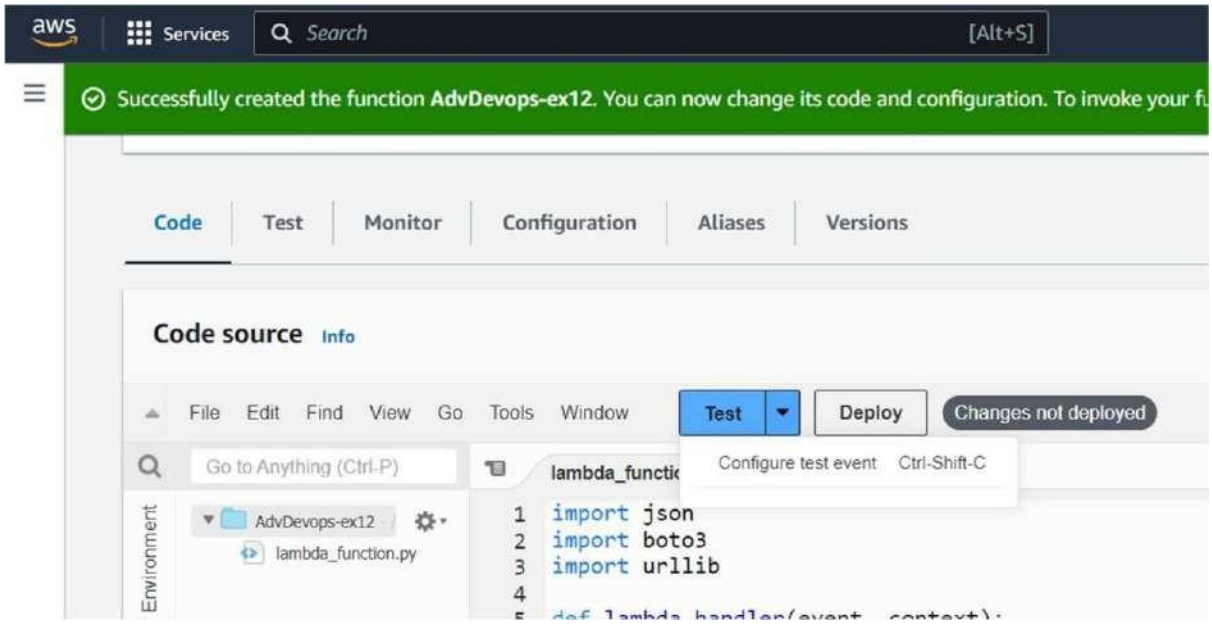
Step 4: The function is up and running.

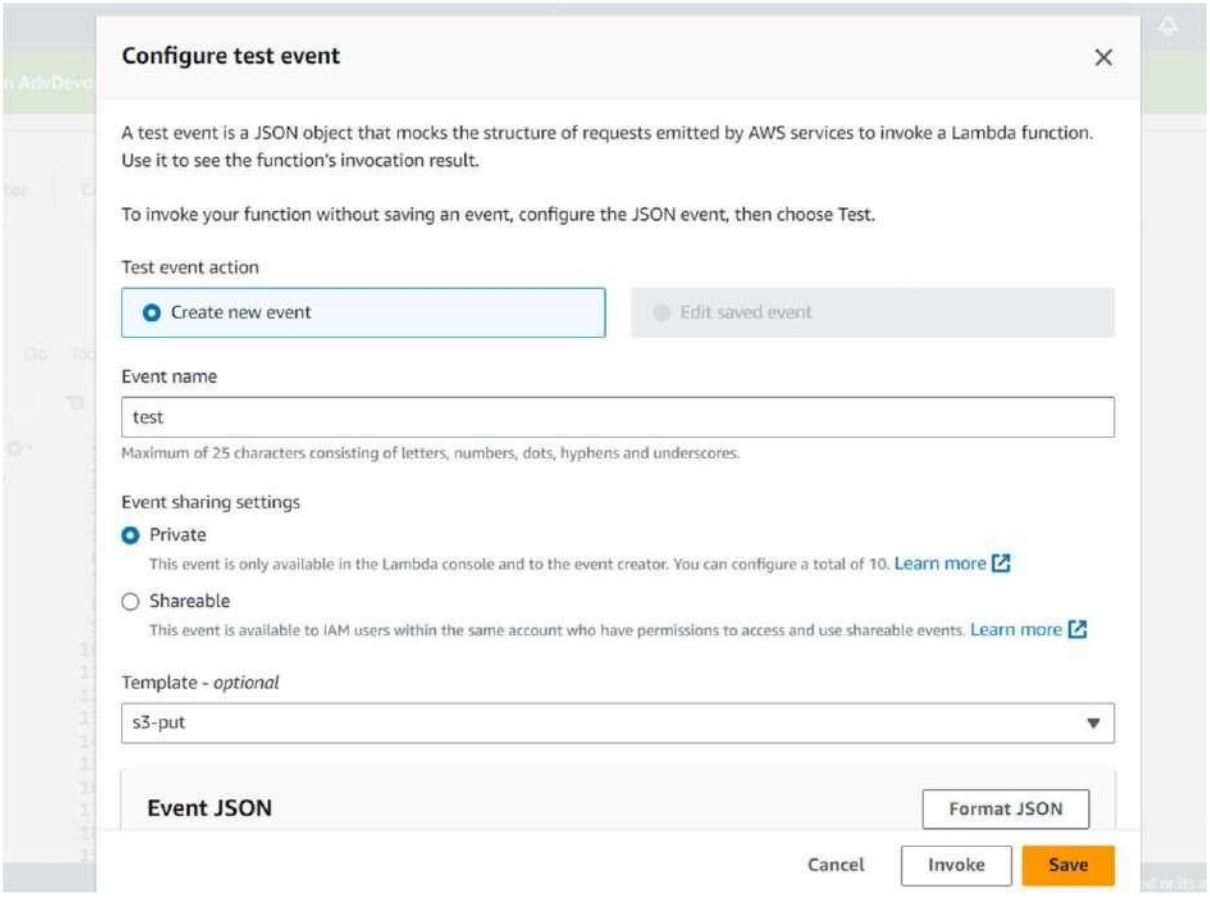


Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.



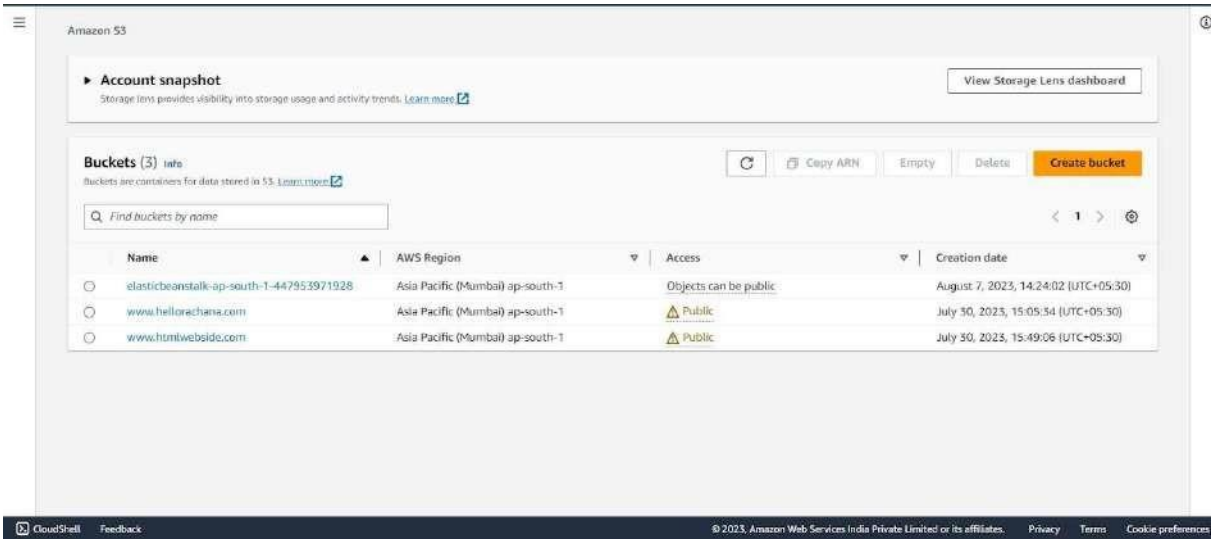
Step 6: Click on Test and choose the ‘S3 Put’ Template.



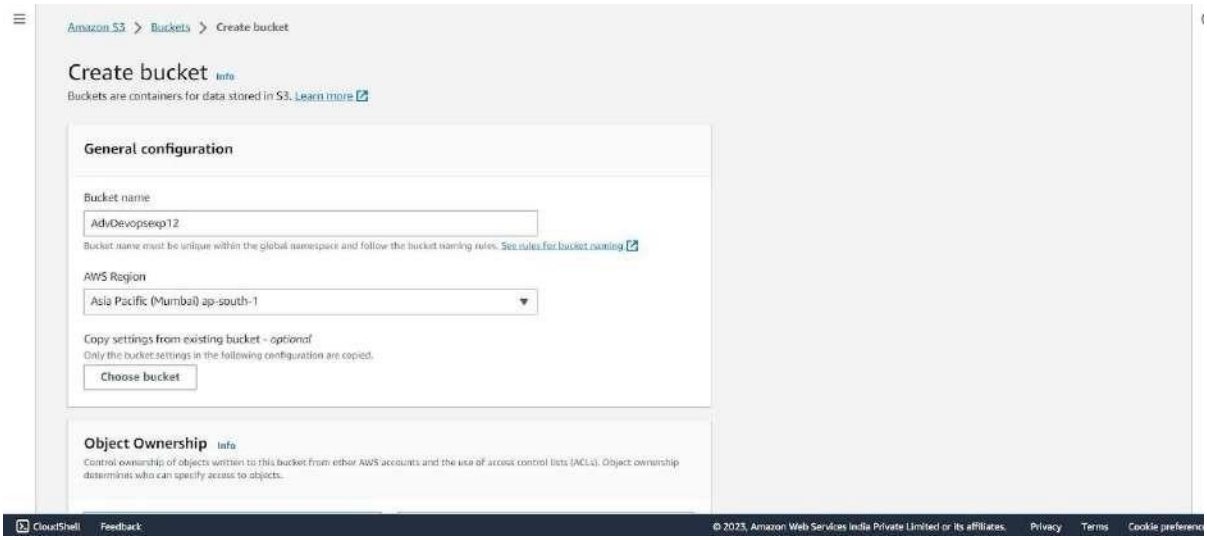


And Save it.

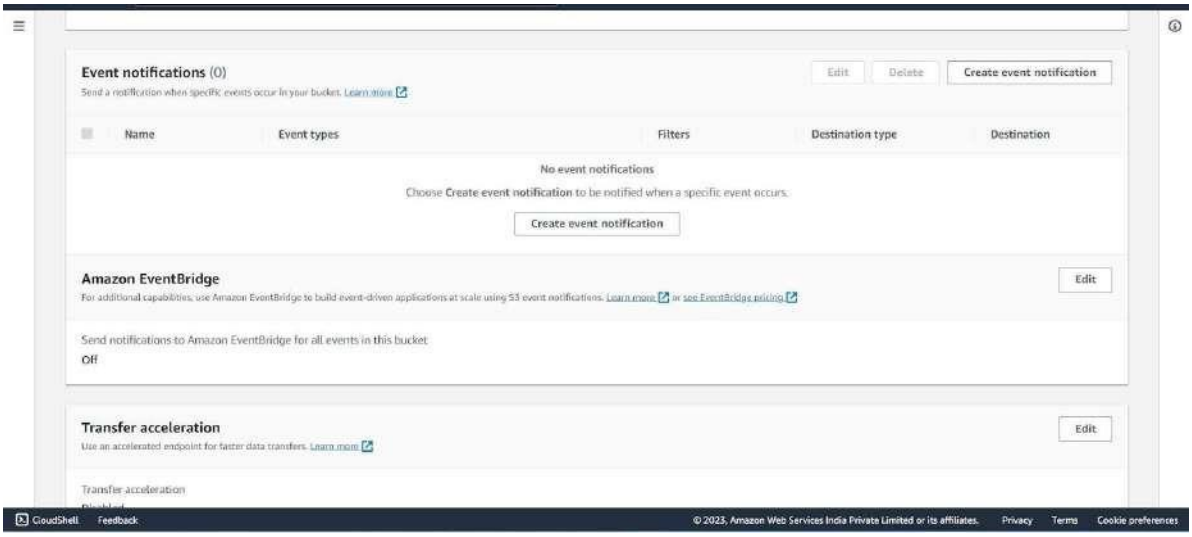
Step 7: Open up the S3 Console and create a new bucket.



Step 8: With all general settings, create the bucket in the same region as the function.

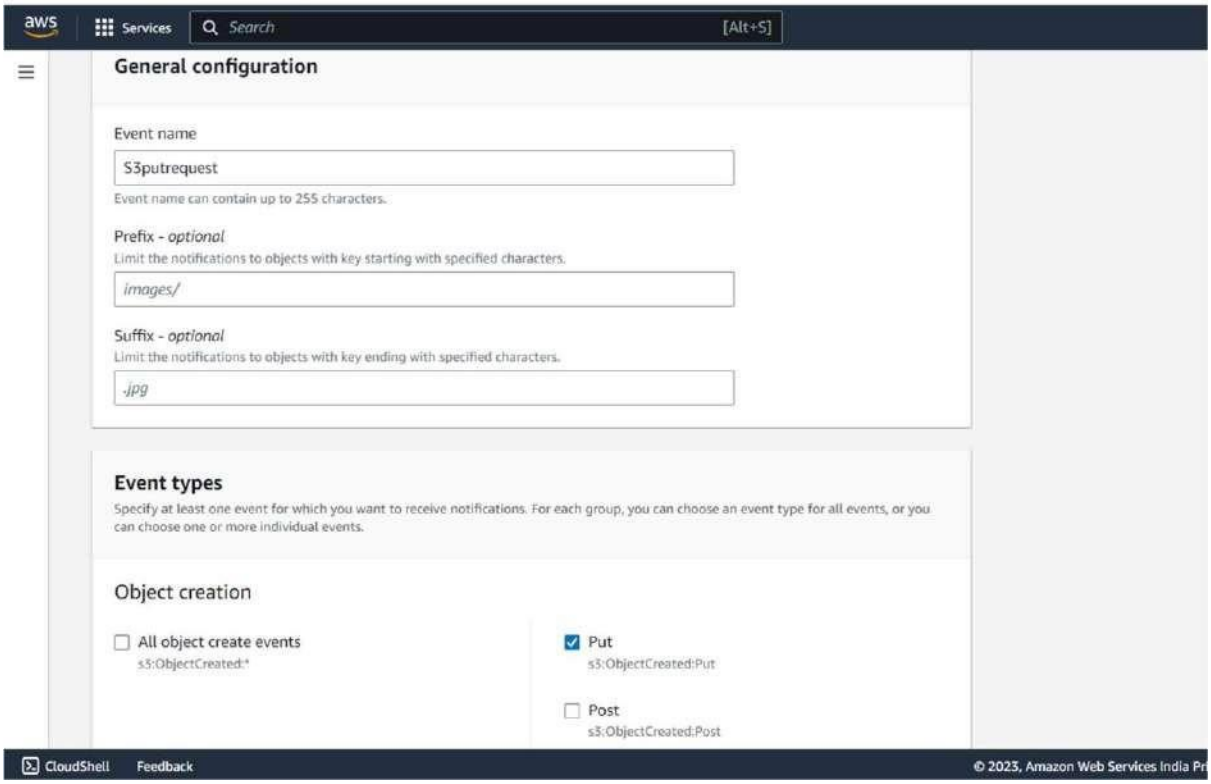


Step 9: Click on the created bucket and under properties, look for events.

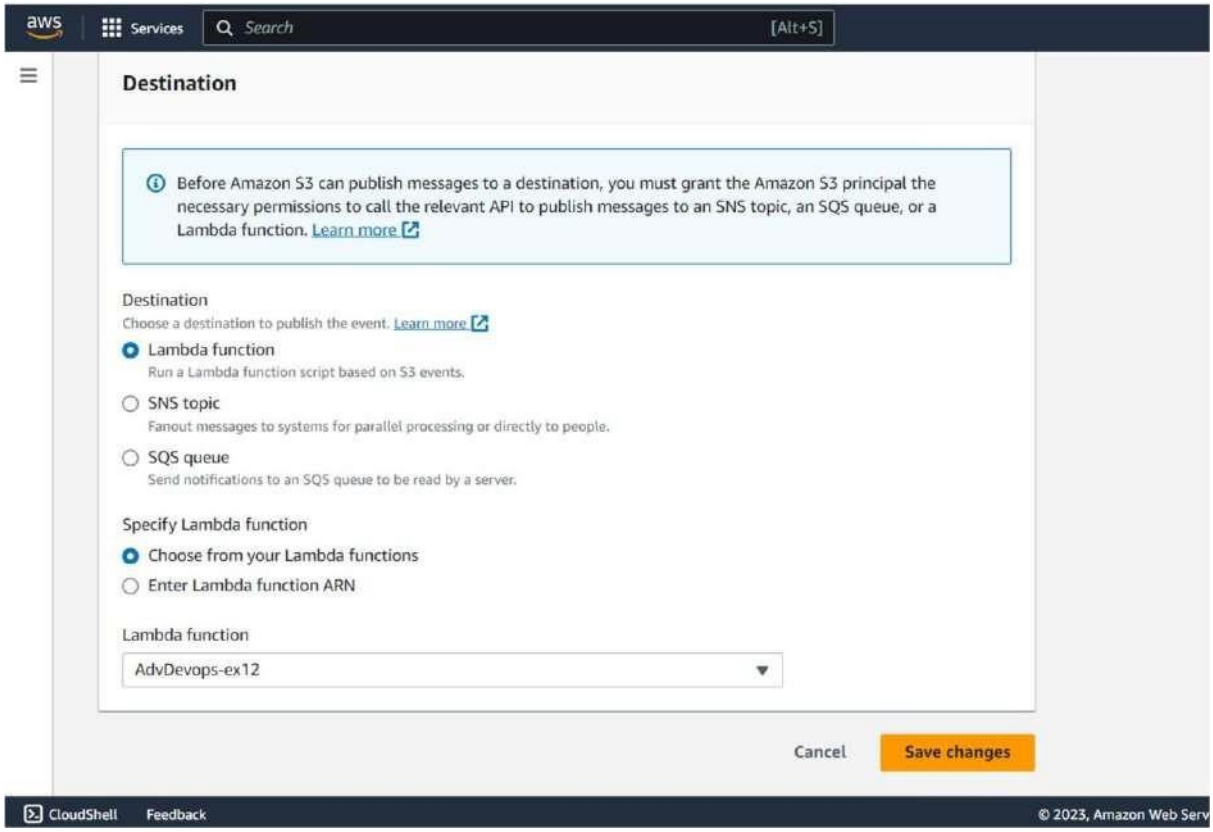


Click on Create Event Notification.

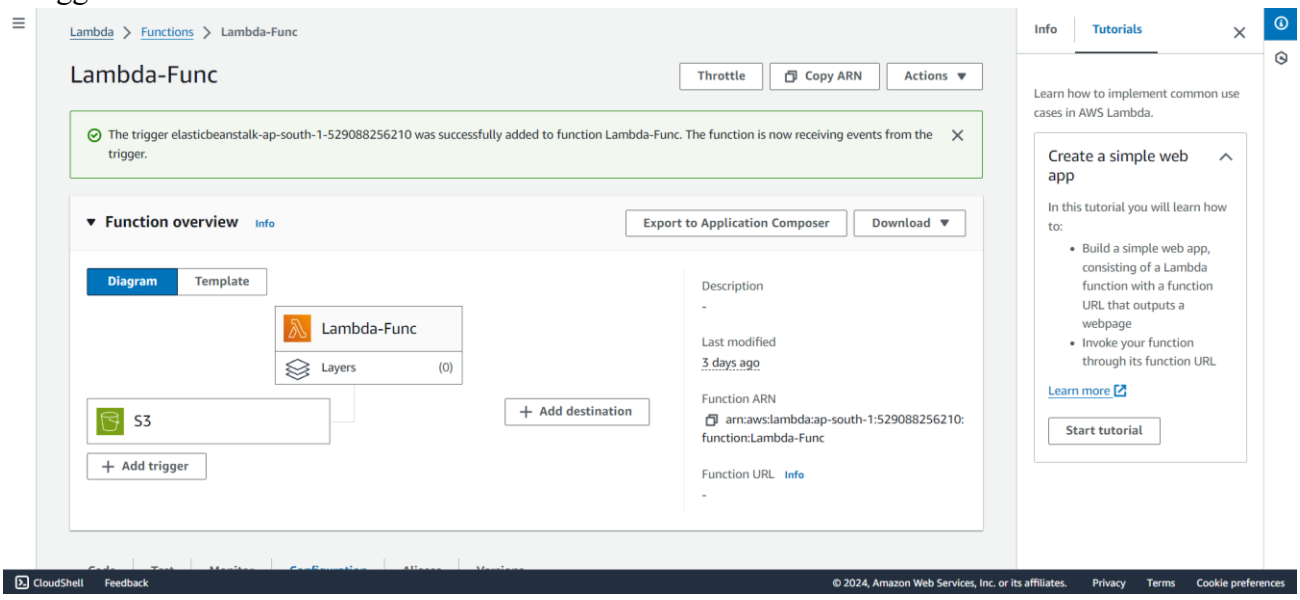
Step 10: Mention an event name and check Put under event types.



Choose Lambda function as destination and choose your lambda function and save the changes.



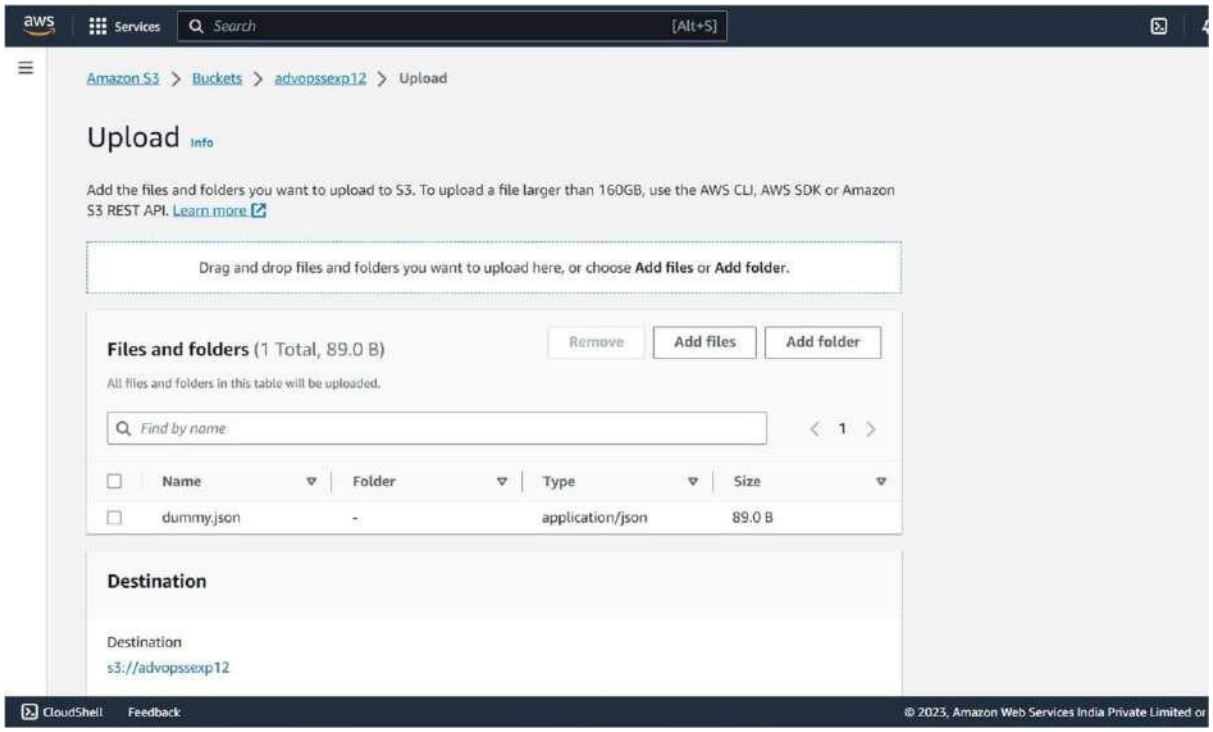
Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



Step 12: Now, create a dummy JSON file locally.

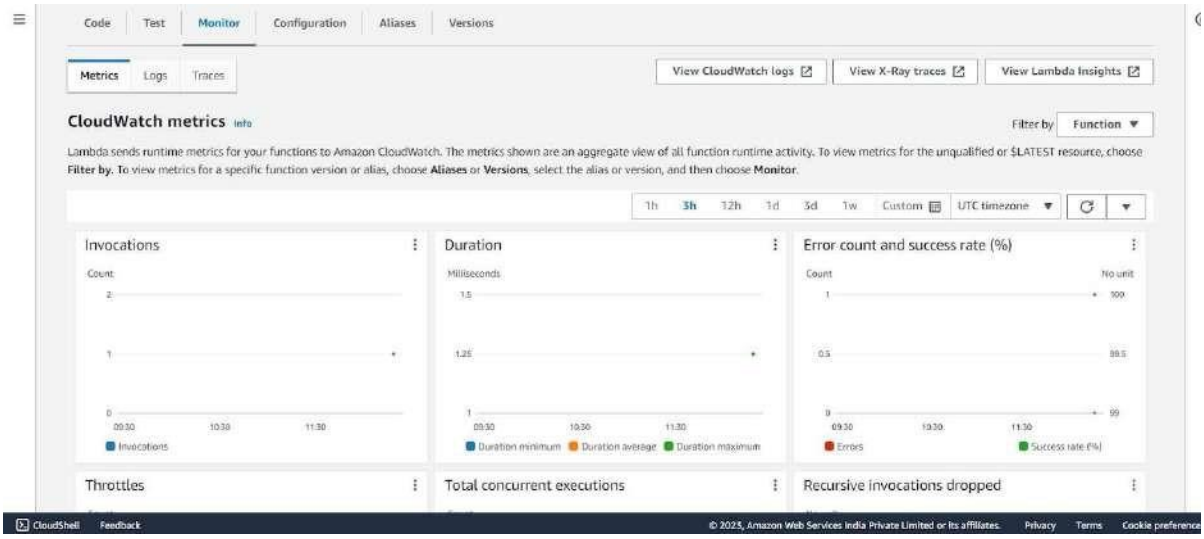
Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.

Step 14: Select the dummy data file from your computer and click Upload.

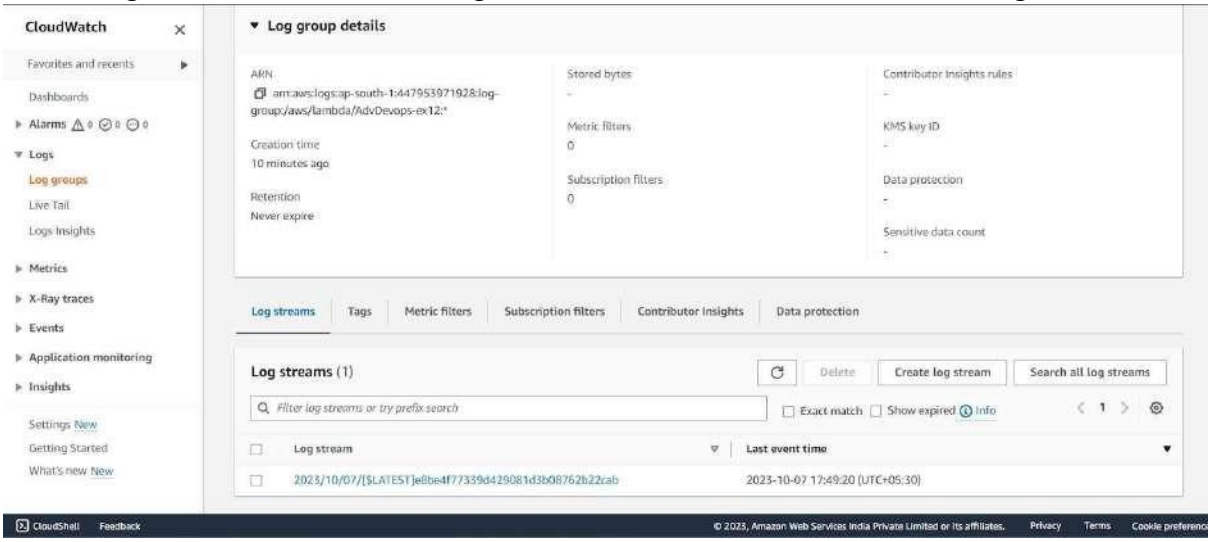


Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

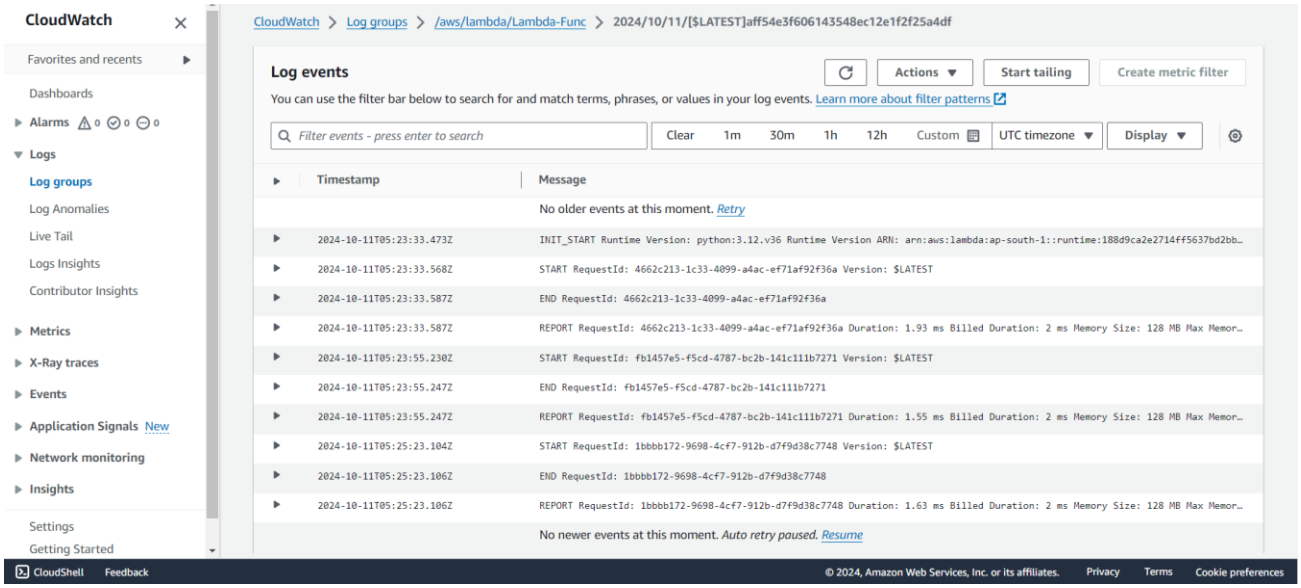
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.



Conclusion: Thus, we have created a Lambda function which logs “An Image has been added” once you add an object to a specific bucket in S3.