PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)

Semester: V
Academic Year: 2025-26
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor:Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID: 24204006

EXPERIMENT NO. 01

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

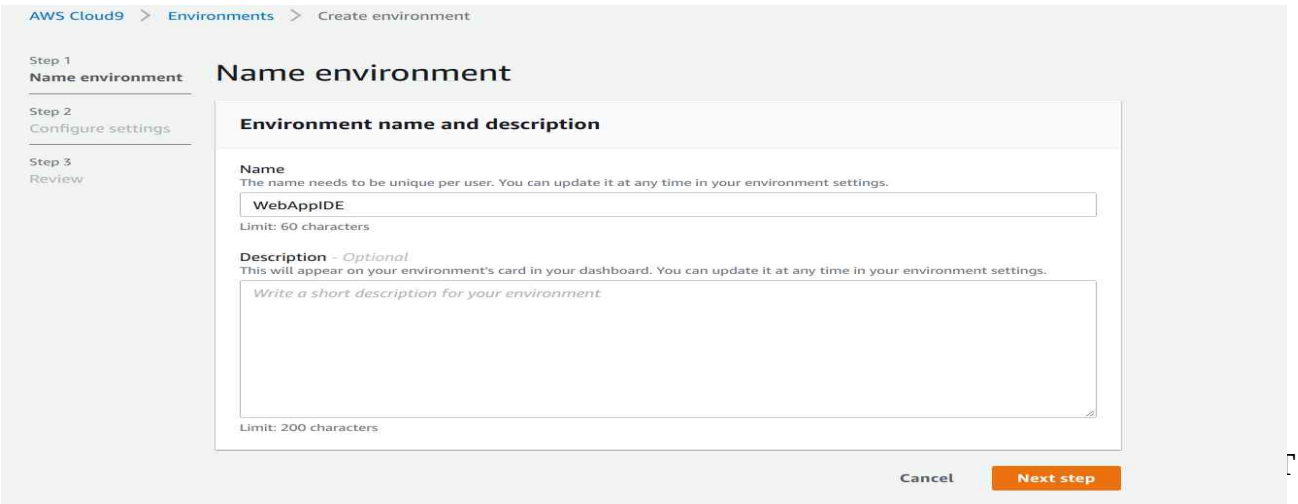
Steps:

1. Login with your AWS account.

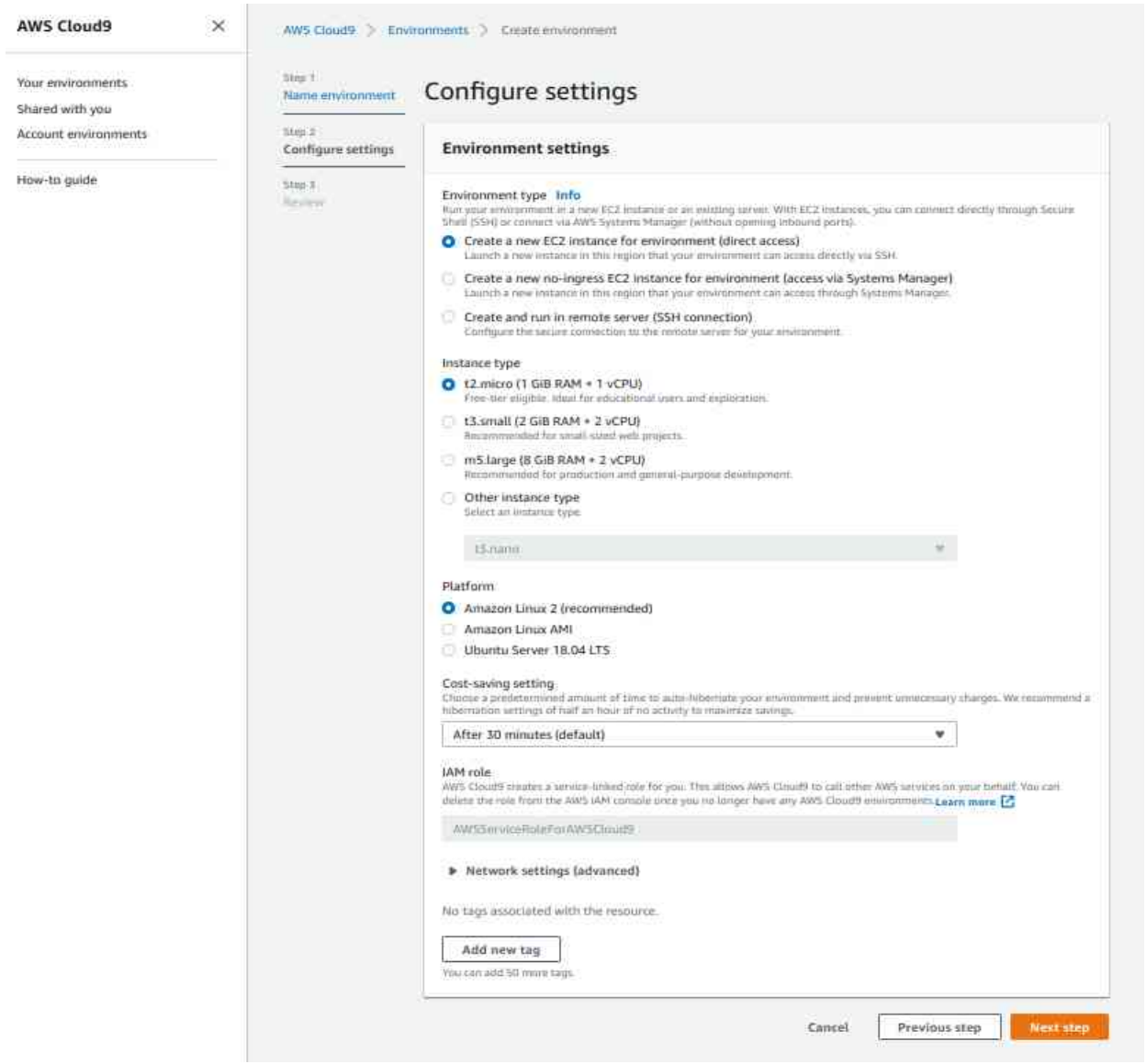
2. Navigate to Cloud 9 service from Developer tools section as below:

3. Click on Create Environment :

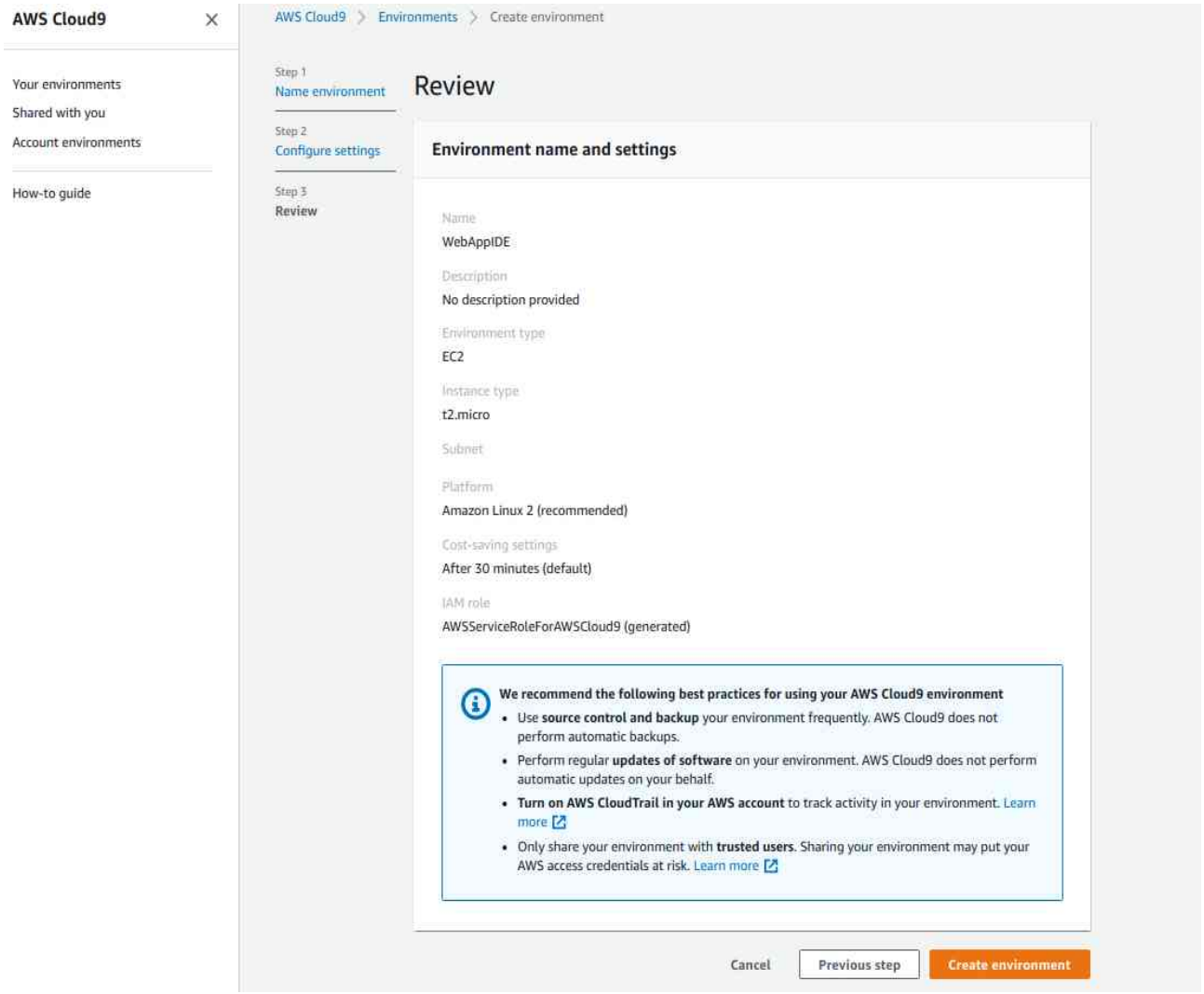
4. Provide name for the Enviornment (Huzaifa) and click on next.

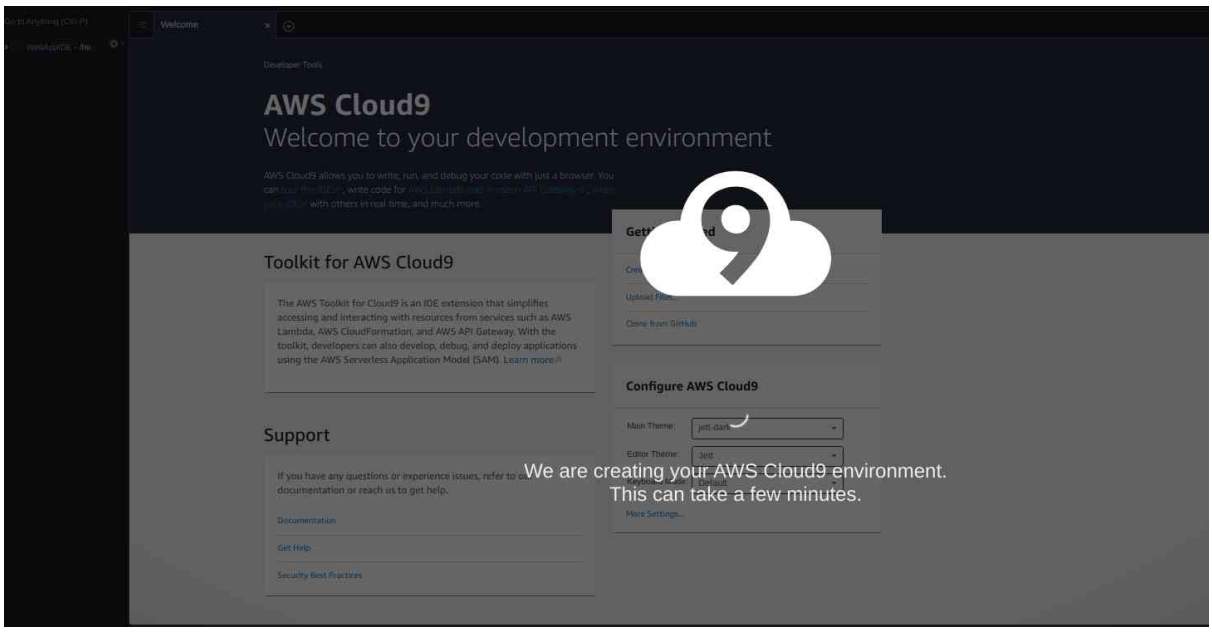


5. Keep all the Default settings as shown in below then Click On next:



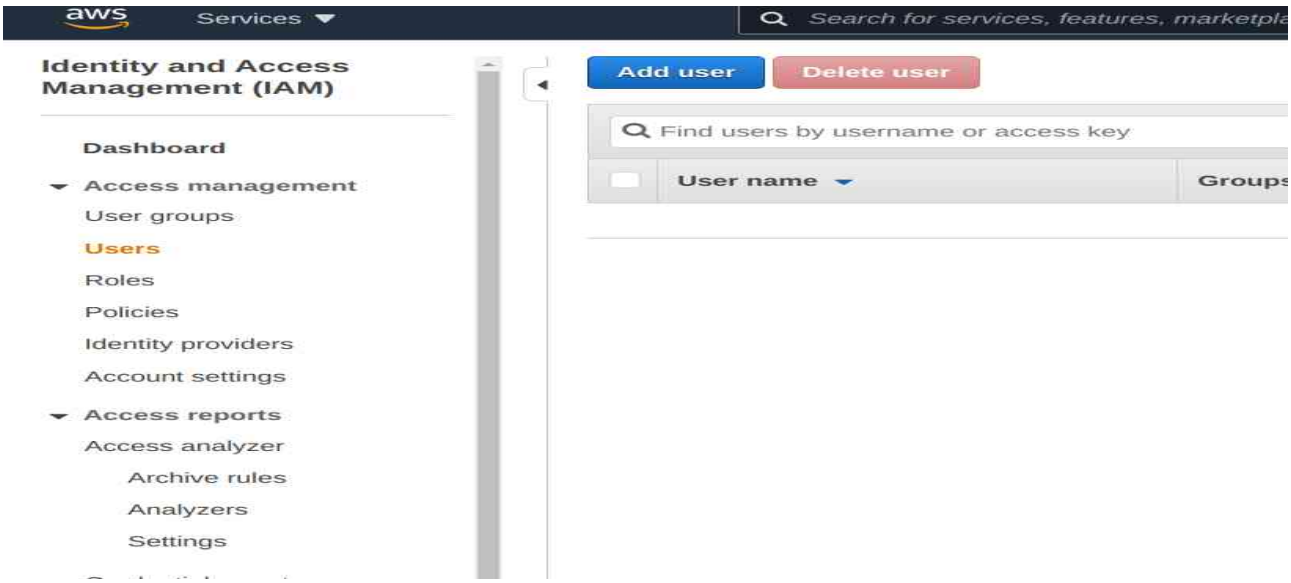
6. Review the Enviornment name and Settings and click on Create Enviornment:


AWS Cloud9
Your environments
Shared with you
Account environments
How-to guide
AWS Cloud9 > Environments > Create environment
Step 1
Name environment
Step 2
Configure settings
Step 3
Review
Review
Environment name and settings
Name
WebAppIDE
Description
No description provided
Environment type
EC2
Instance type
t2.micro
Subnet
Platform
Amazon Linux 2 (recommended)
Cost-saving settings
After 30 minutes (default)
IAM role
AWSServiceRoleForAWSCloud9 (generated)
We recommend the following best practices for using your AWS Cloud9 environment
Use source control and backup your environment frequently. AWS Cloud9 does not perform automatic backups.
Perform regular updates of software on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
Turn on AWS CloudTrail in your AWS account to track activity in your environment. Learn more
Only share your environment with trusted users. Sharing your environment may put your AWS access credentials at risk. Learn more
Cancel
Previous step
Create environment

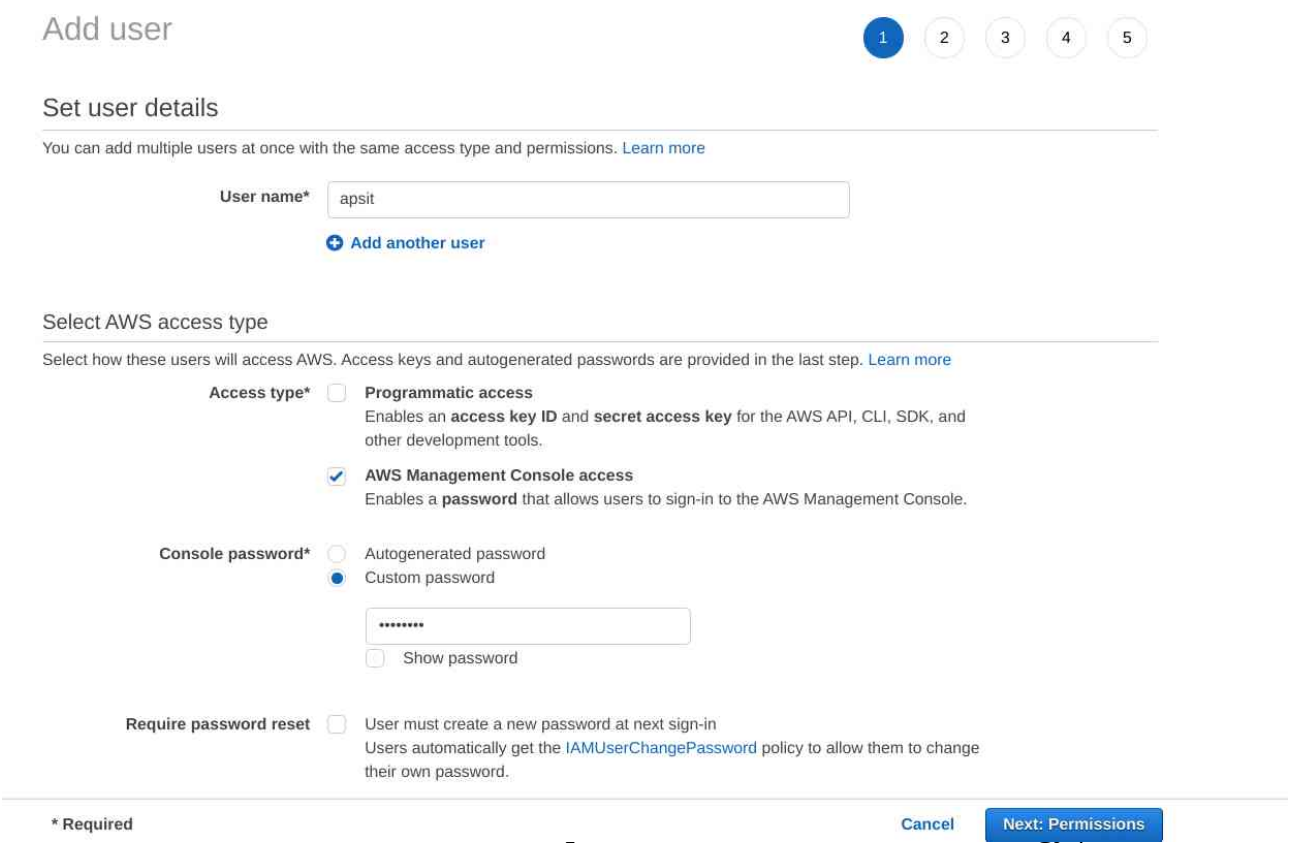

AWS Cloud9
Welcome to your development environment
Toolkit for AWS Cloud9
Support
Configure AWS Cloud9
We are creating your AWS Cloud9 environment.
This can take a few minutes.

It will take few minutes to create aws instance for your Cloud 9 Enviornment.

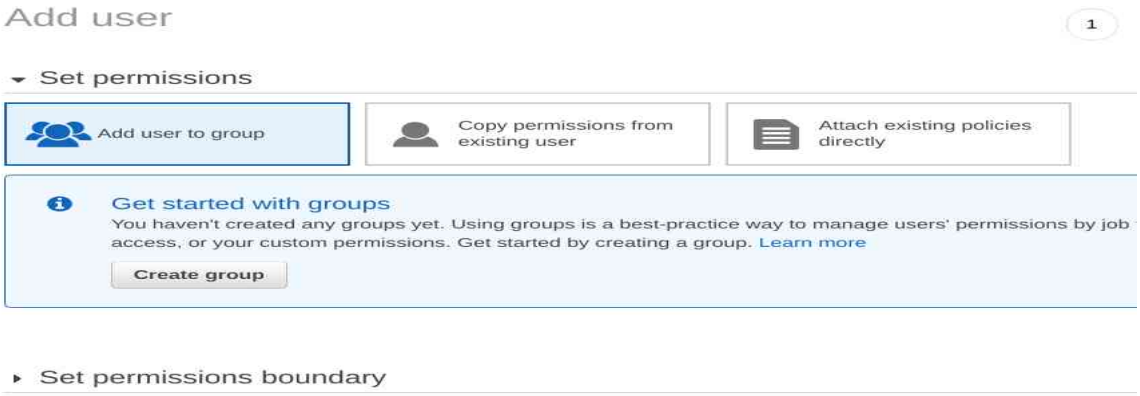
7. Till that time open IAM Identity and Access Management in order to Add user In other tab Click On Add User.



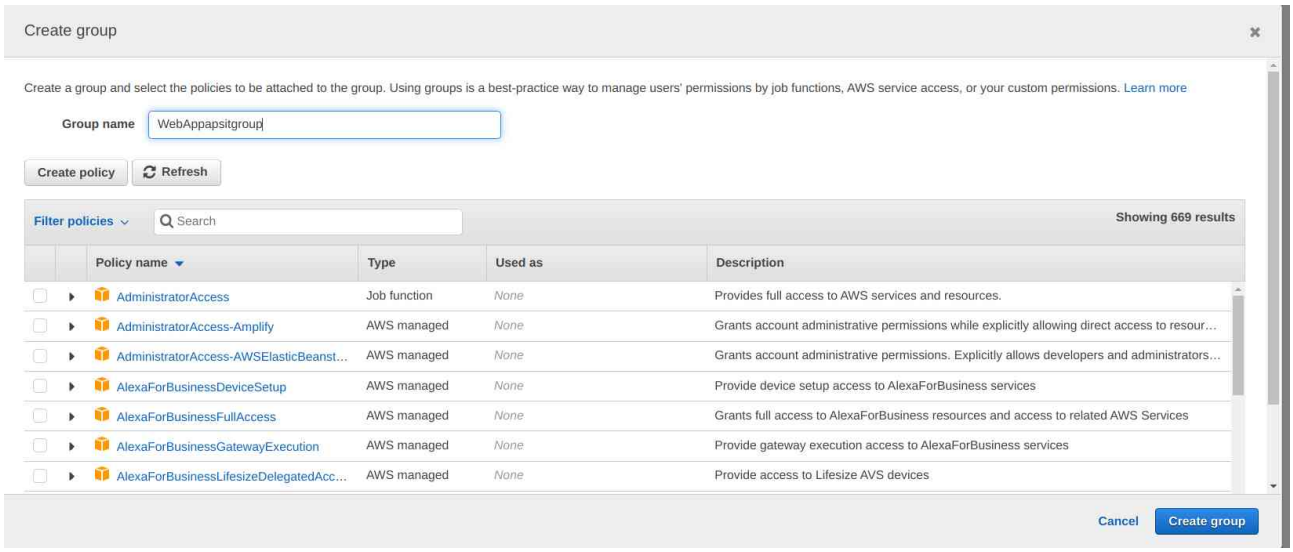
8. Add user provide manual password if you want and click on Next permission tab Give Custom Password.



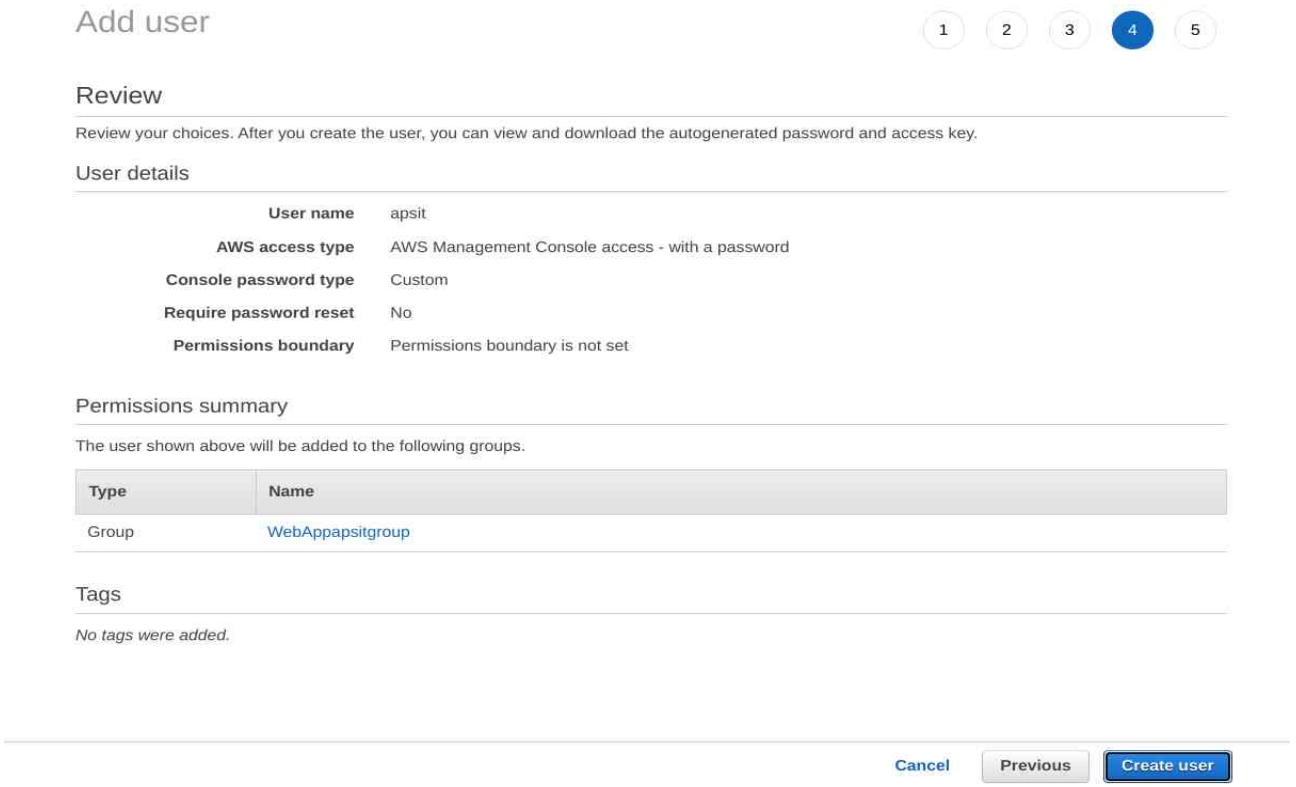
9.Click on Create group To Add User In A Group And Give Permission To That Group



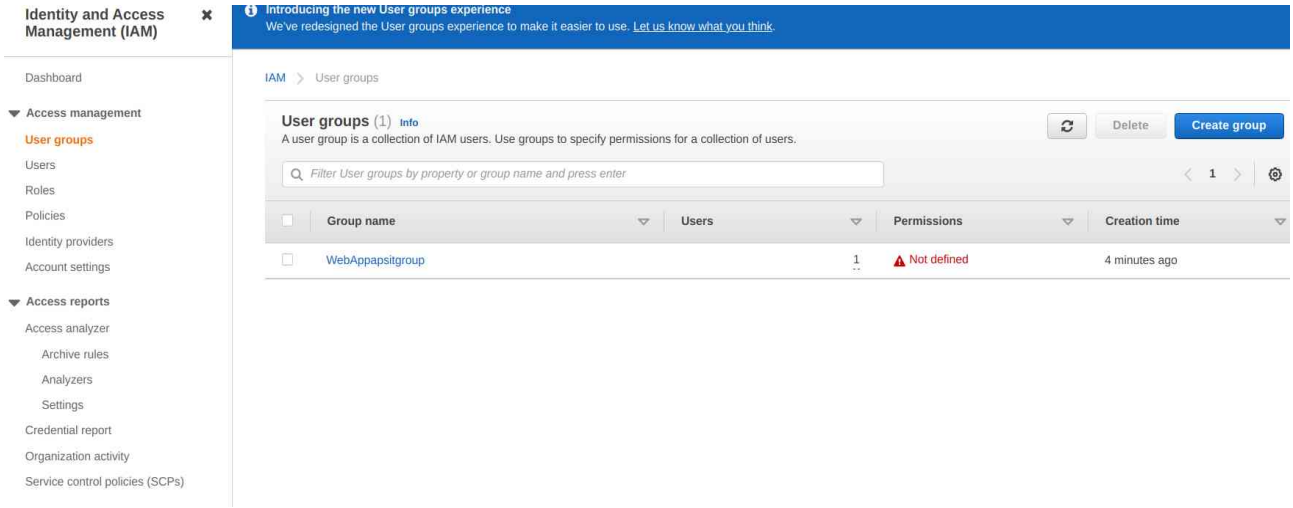
10. Provide group name and click on create group.



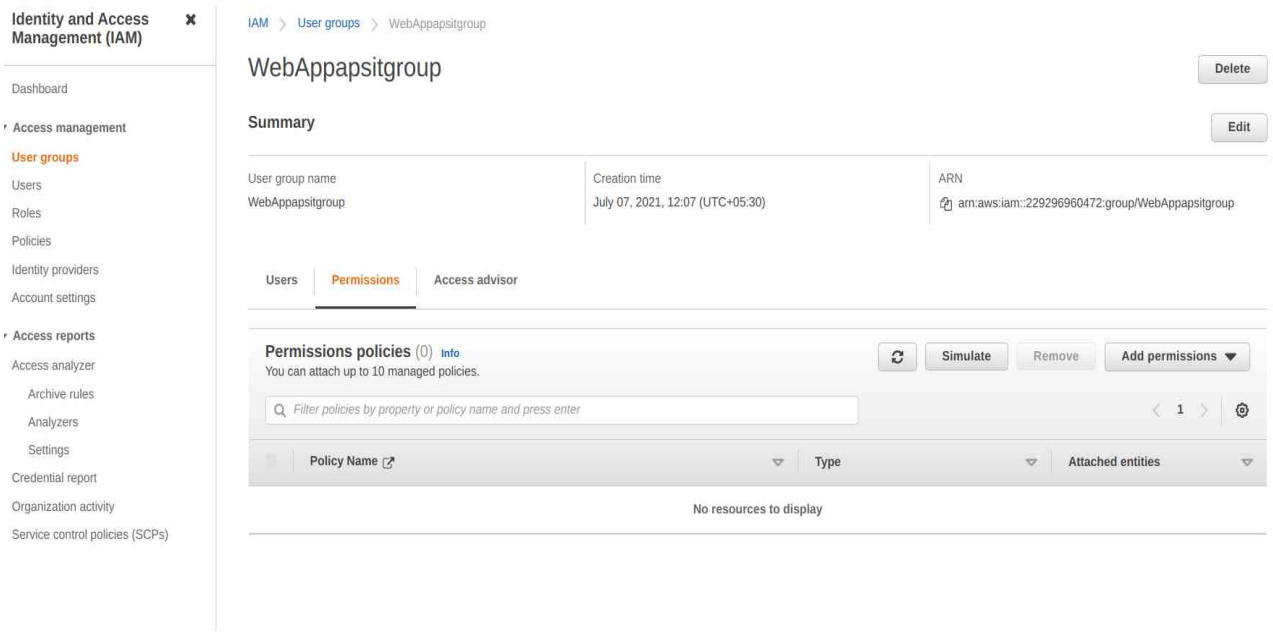
11. After that group is created click on next if u want to provide tag else click on Review for user settings and click on create user as shown in fig.



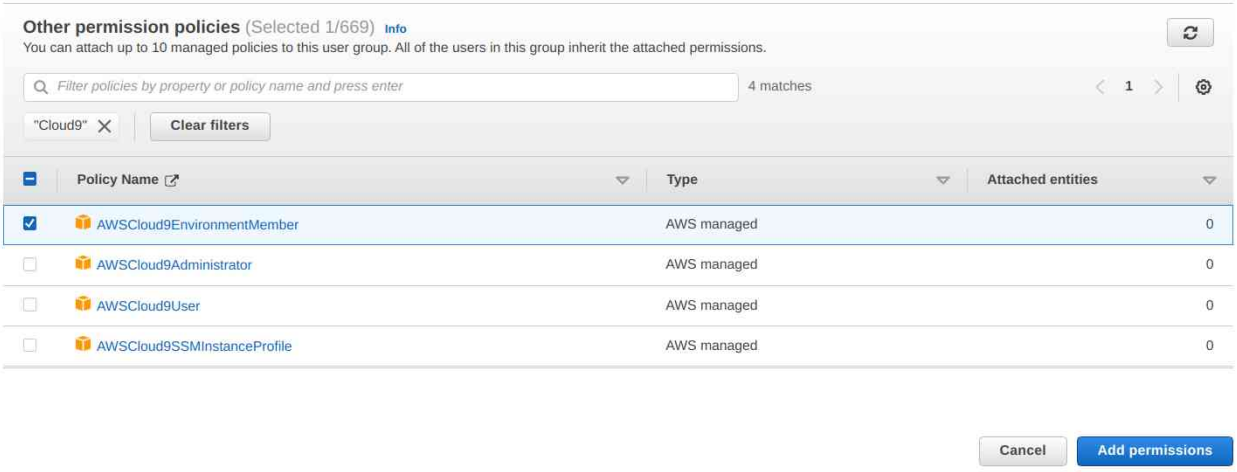
12. Now close that window and Navigate to user Groups from left pane in IAM.



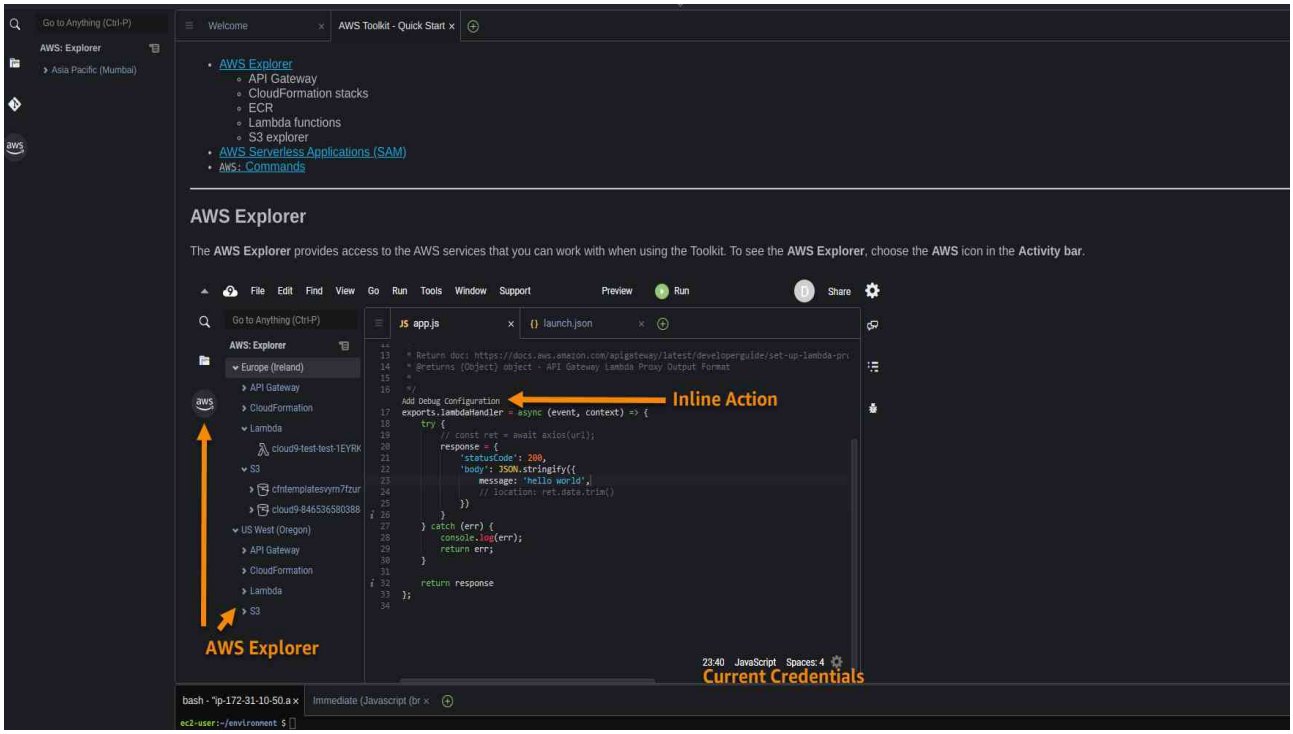
13. click on your group name which you have created and navigate to permission tab as shown:



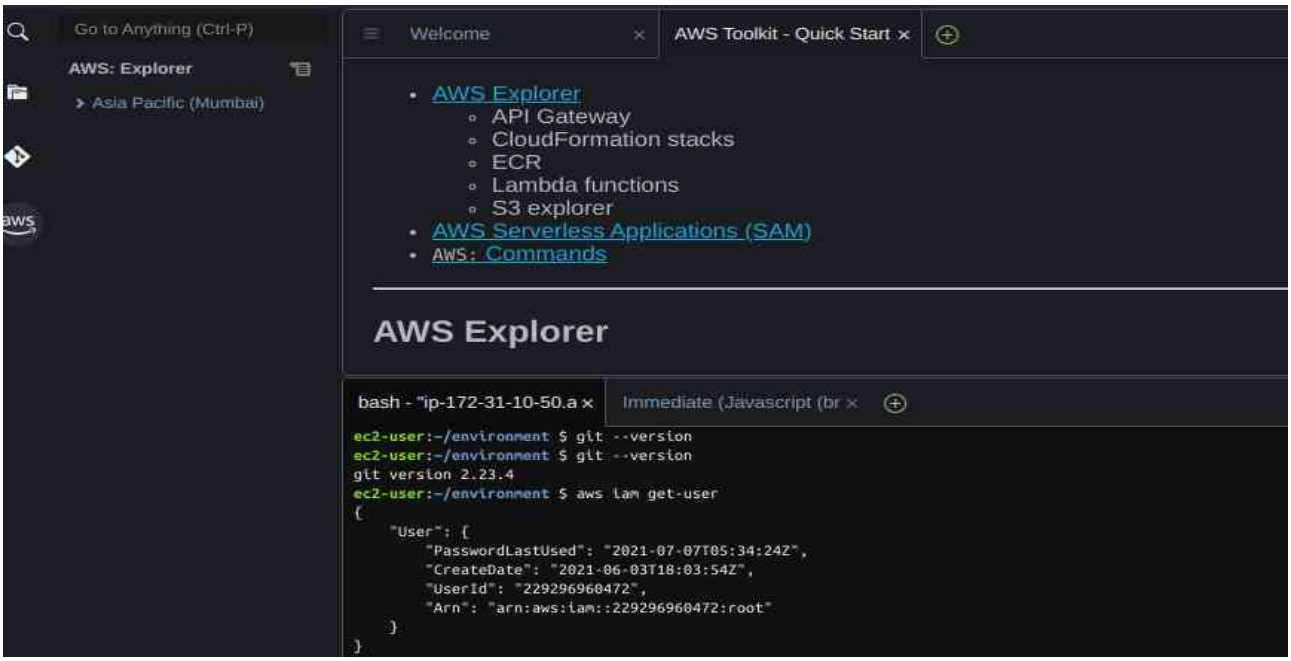
14. Now click on Add permission and select Attach Policy after that search for Cloud9 related policy and select Awscloud9EnviornmentMember policy and add it.



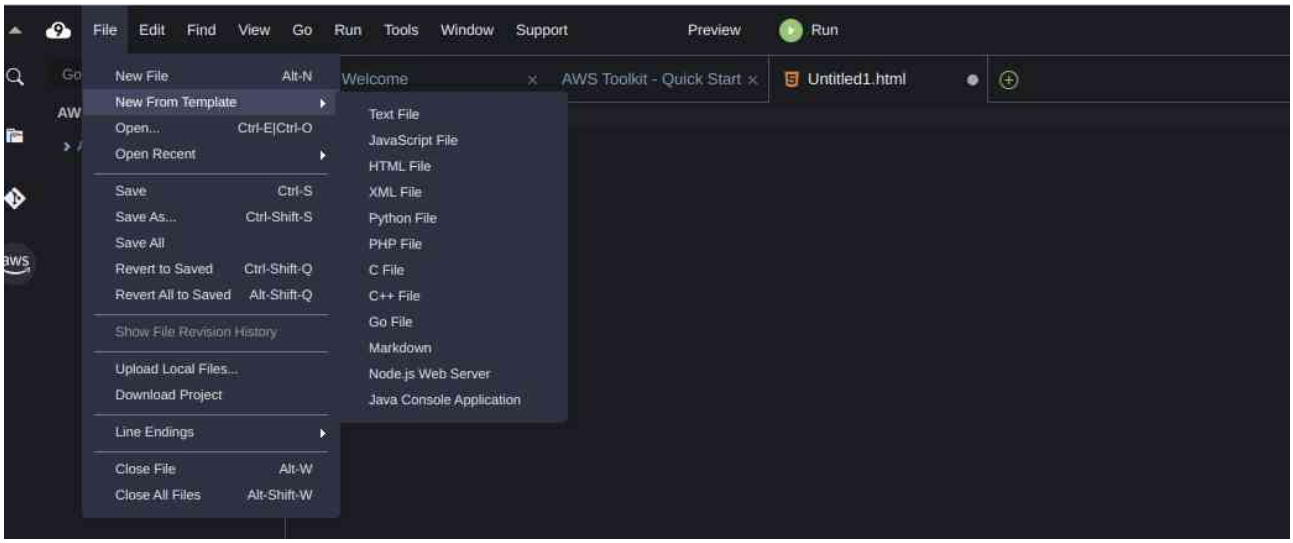
15. now we move towards our cloud9 IDE Enviornment tab it shows as shown :



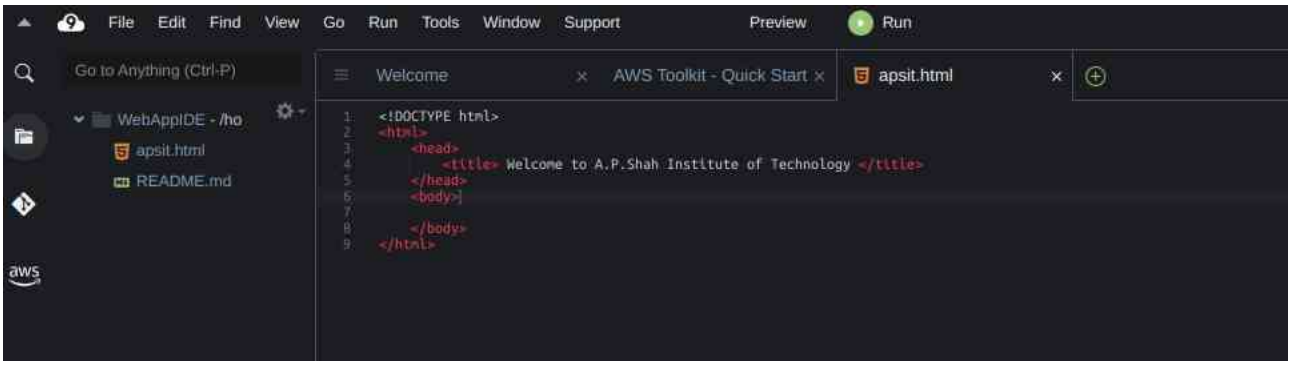
16. If you check at bottom side Cloud9 IDE also giving you and aws CLI for command operations: as we here checked git version, iam user details and so on...



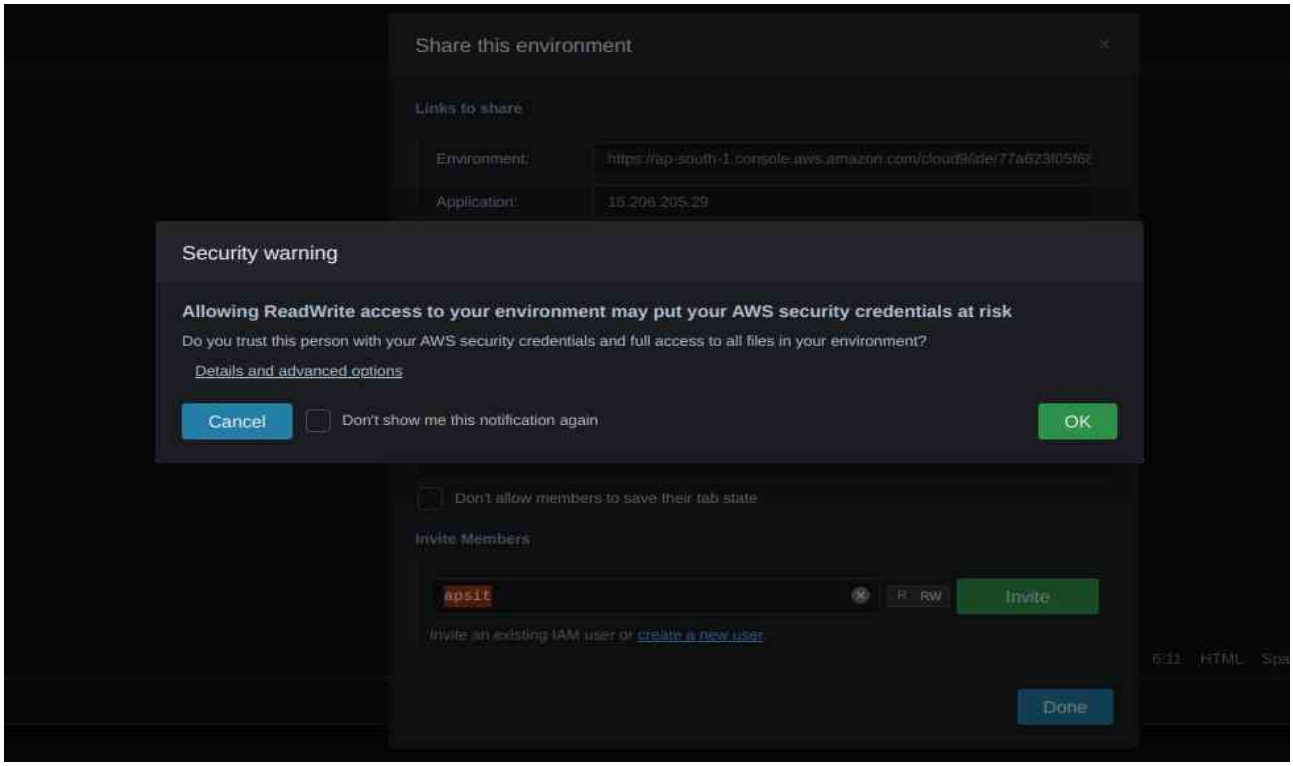
17. Now we will setup collaborative enviornment Click on File you can create new file or choose from template, here an opting html file to collaborate.



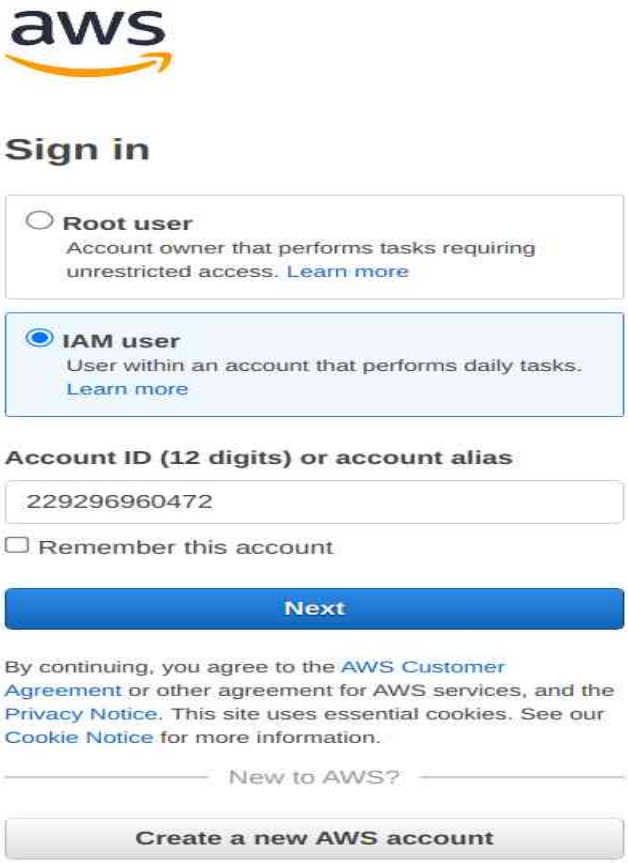
18. Edit html file and save it



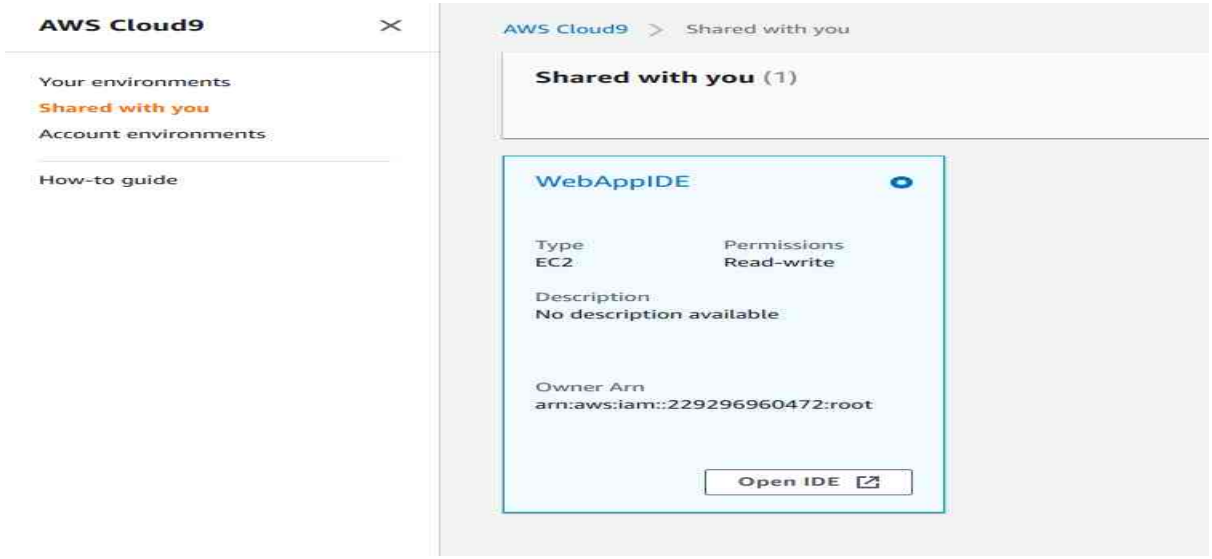
19. now in order to share this file to collaborate with other members of your team click on Share option on Right Pane and username which you created in IAM before into Invite members and enable permission as RW (Read and Write) and click on Done. Click OK for Security warning.



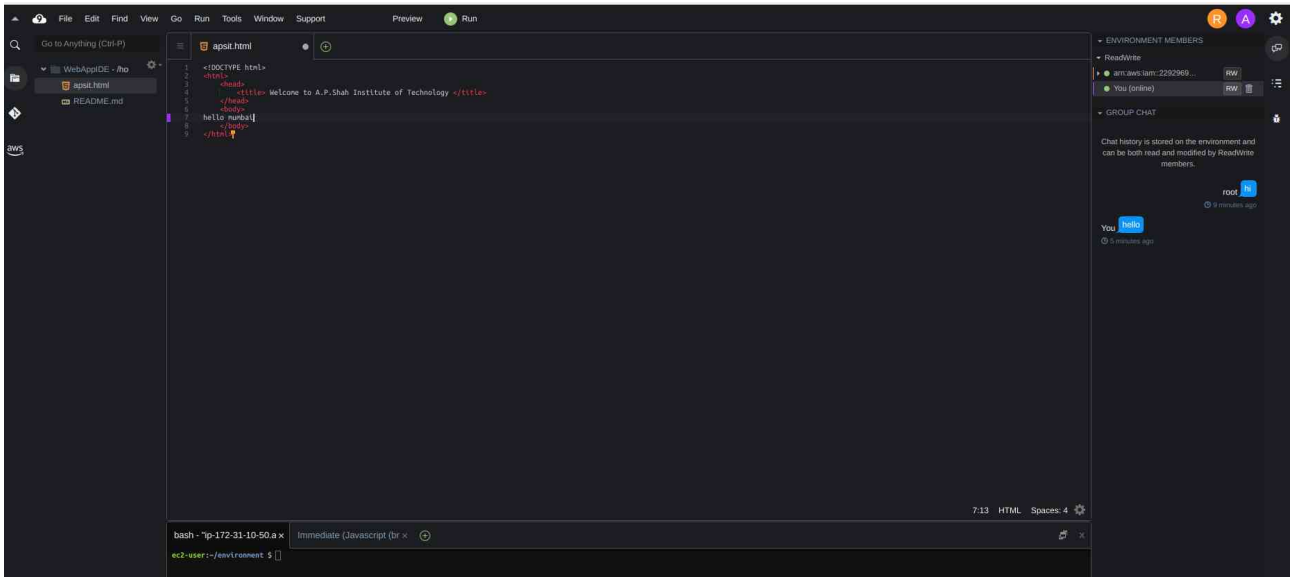
20. Now Open your Browsers Incognito Window and login with IAM user which you configured before.

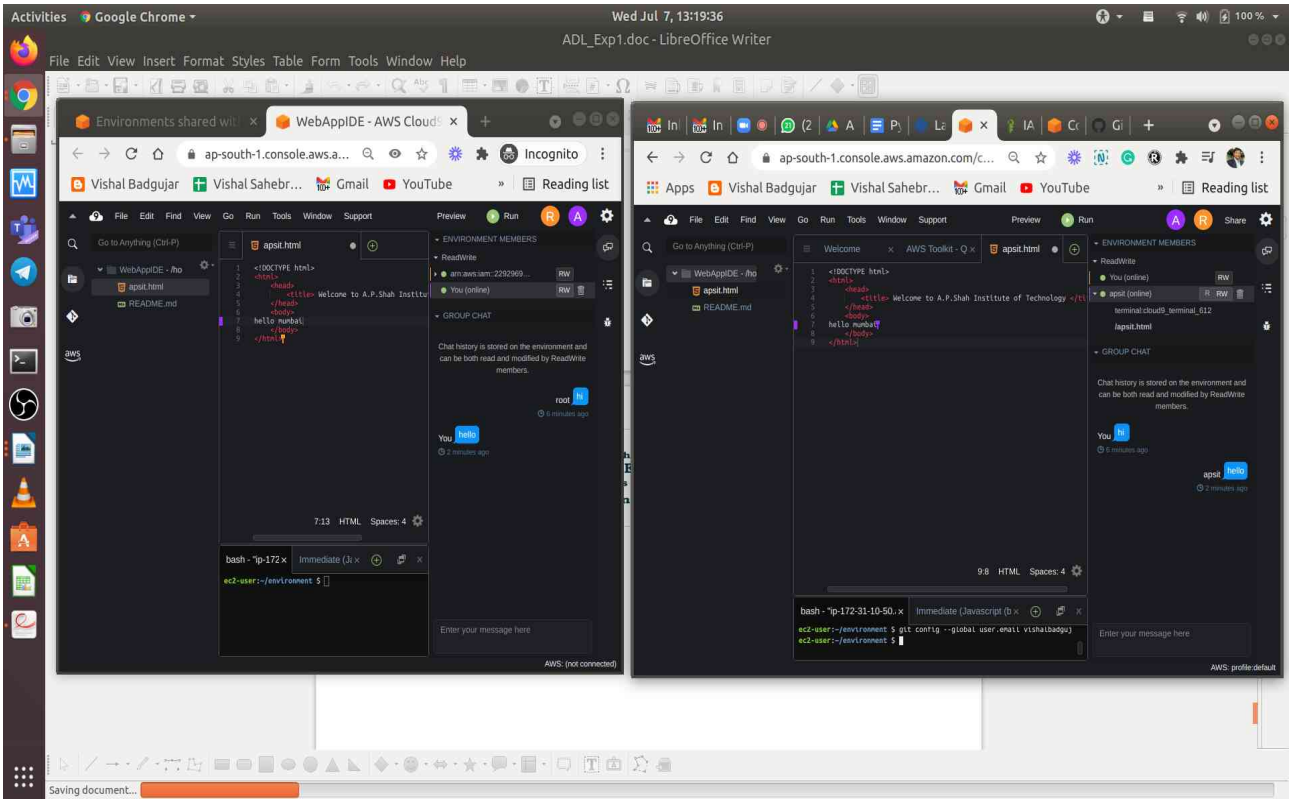


21. After Successful login with IAM user open Cloud9 service from dashboard services and click on shared with you enviornment to collaborate.

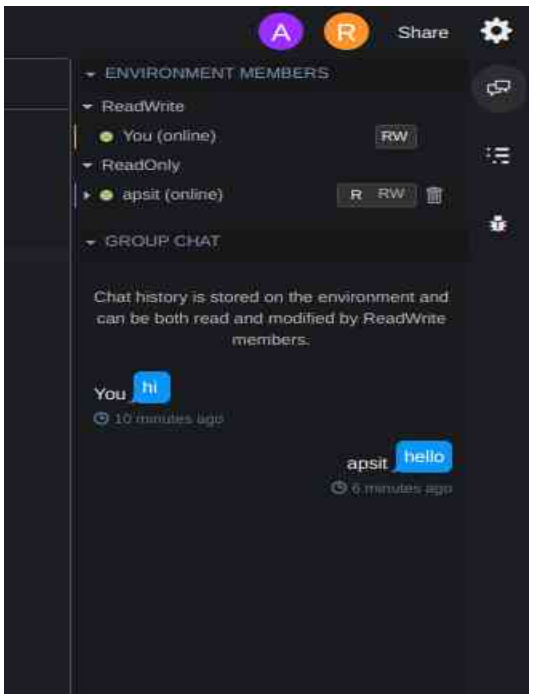


22. Click on Open IDE you will same interface as your other member have to collaborate in real time, also you all within team can do group chats as shown below:





24. you can also explore settings where you can update permissions of your temmates as from RW to R only or you can remove user from an environment.



For more info related to AWS-Cloud 9 you all can refer following Docs.

https://docs.aws.amazon.com/cloud9/latest/user-guide/aws-cloud9-ug.pdf

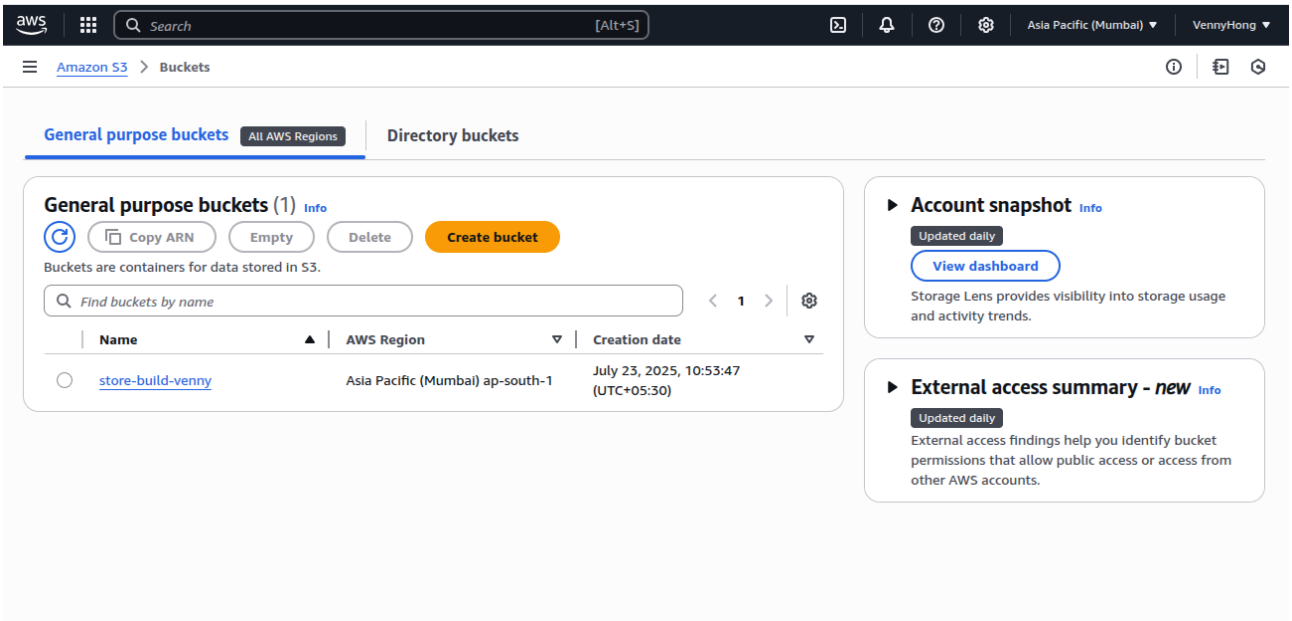
Conclusion: We Have Learned About benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration With A Teammates. Also We Have Leaner To Create User And User Group Also How To Add User To A User Group Was Learned.

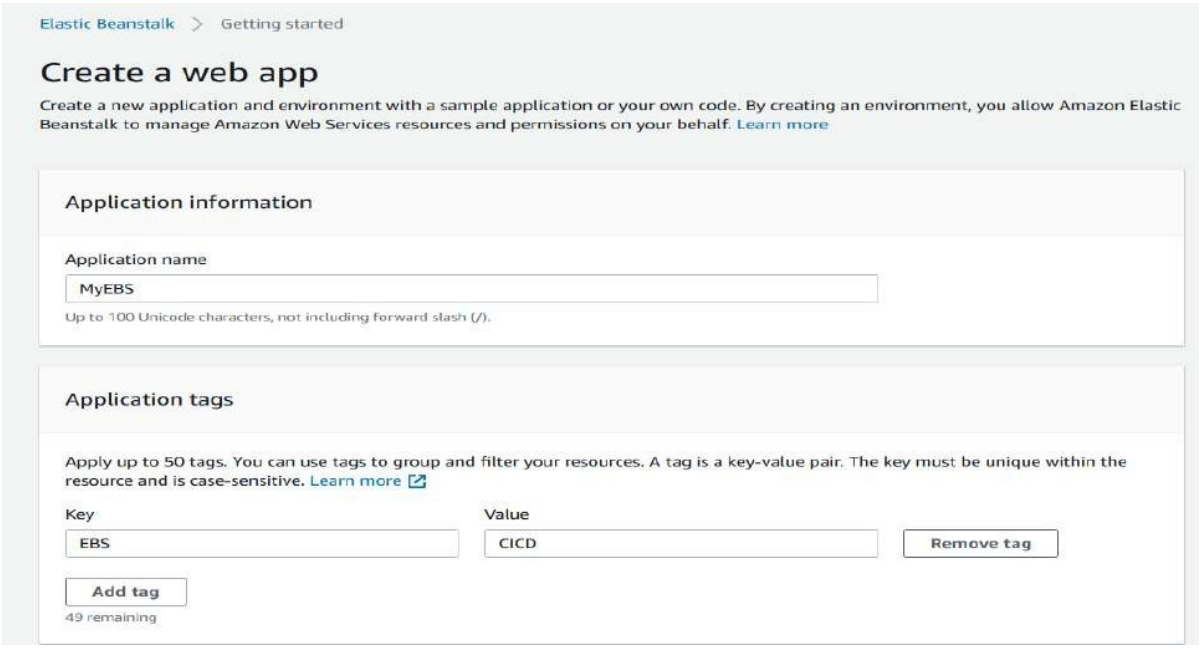
Semester: V
Academic Year: 2021-22
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Manjusha K.

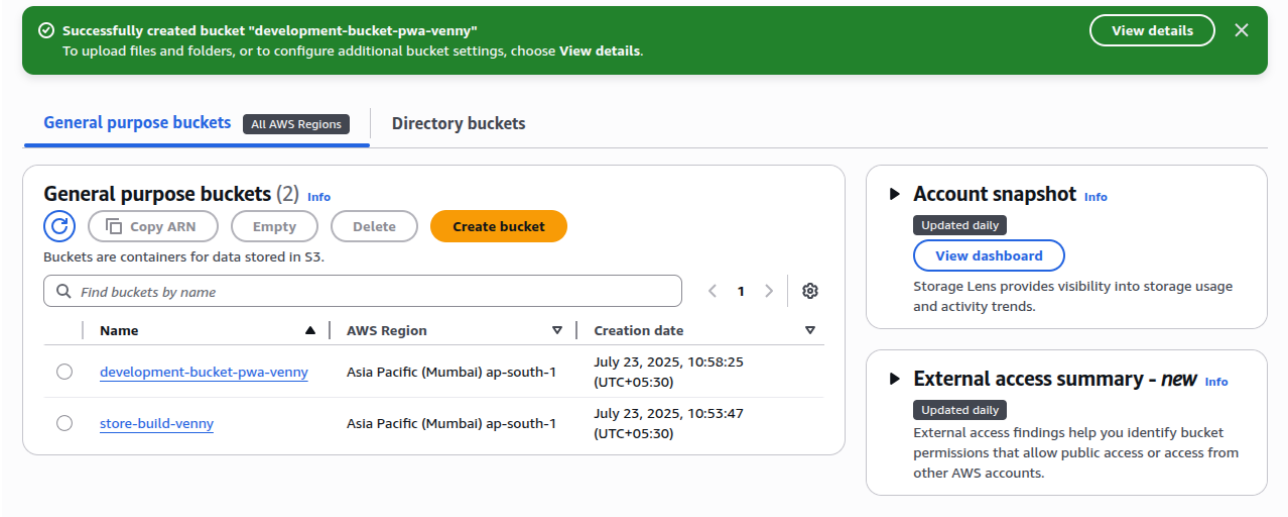
Name of Student:Huzaifa Bubere
ID:24204006

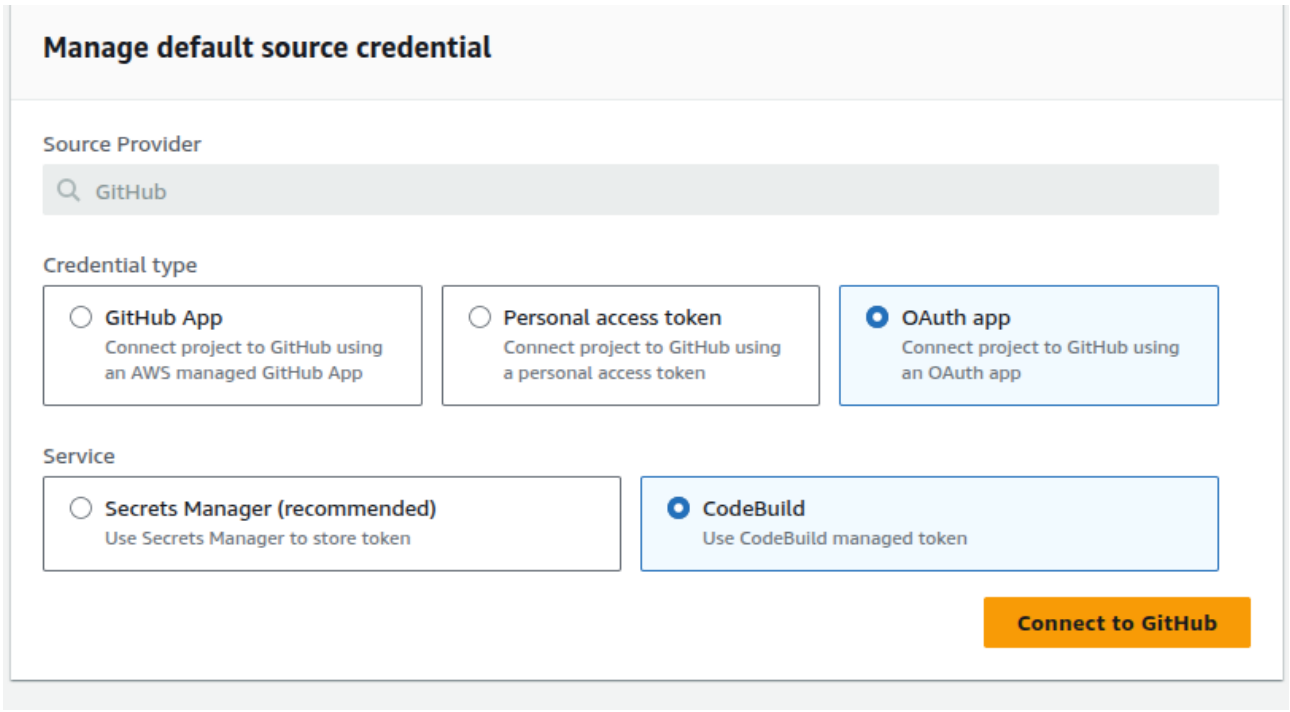
EXPERIMENT NO. 02

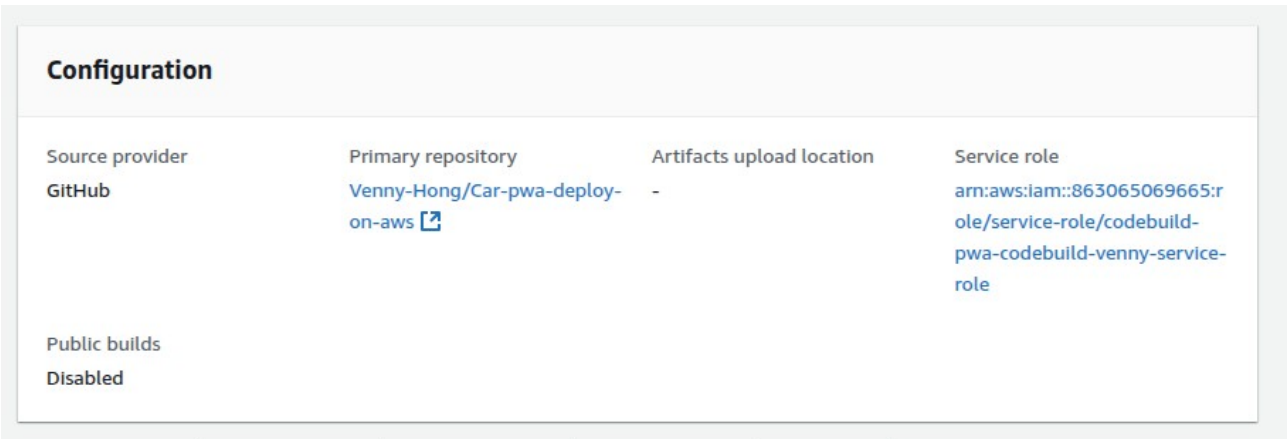
Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

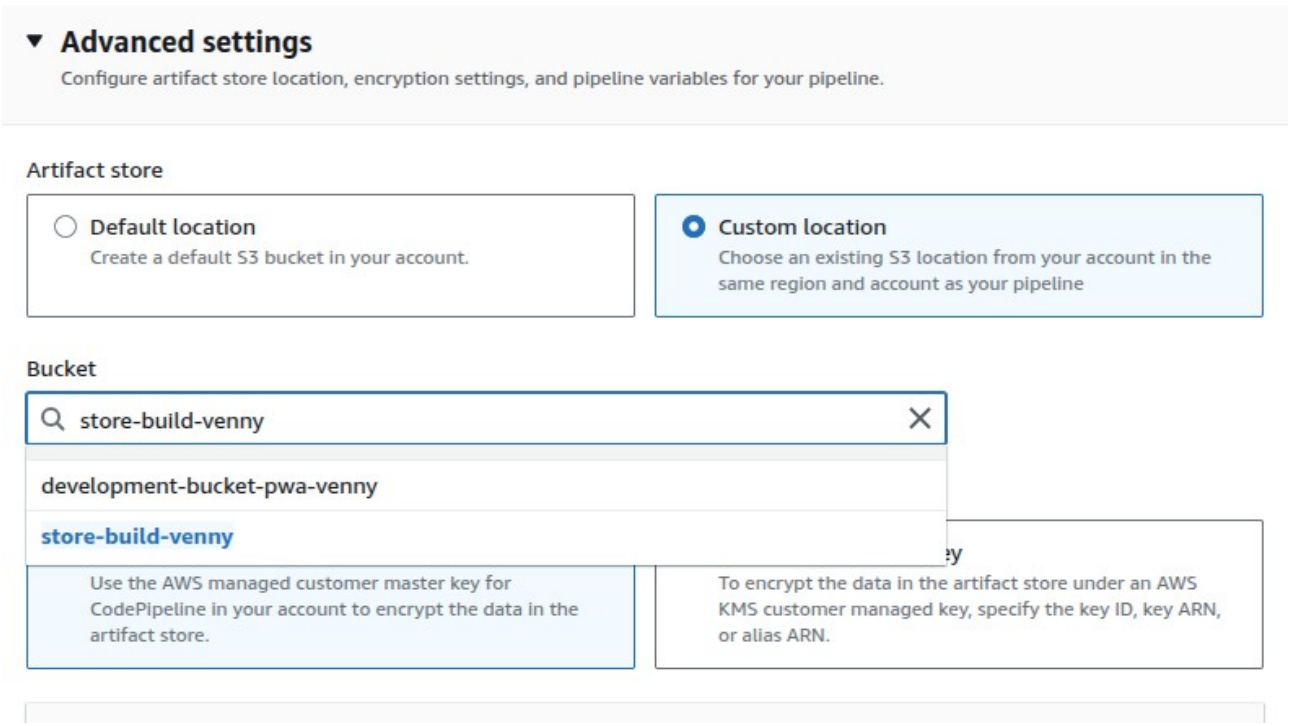


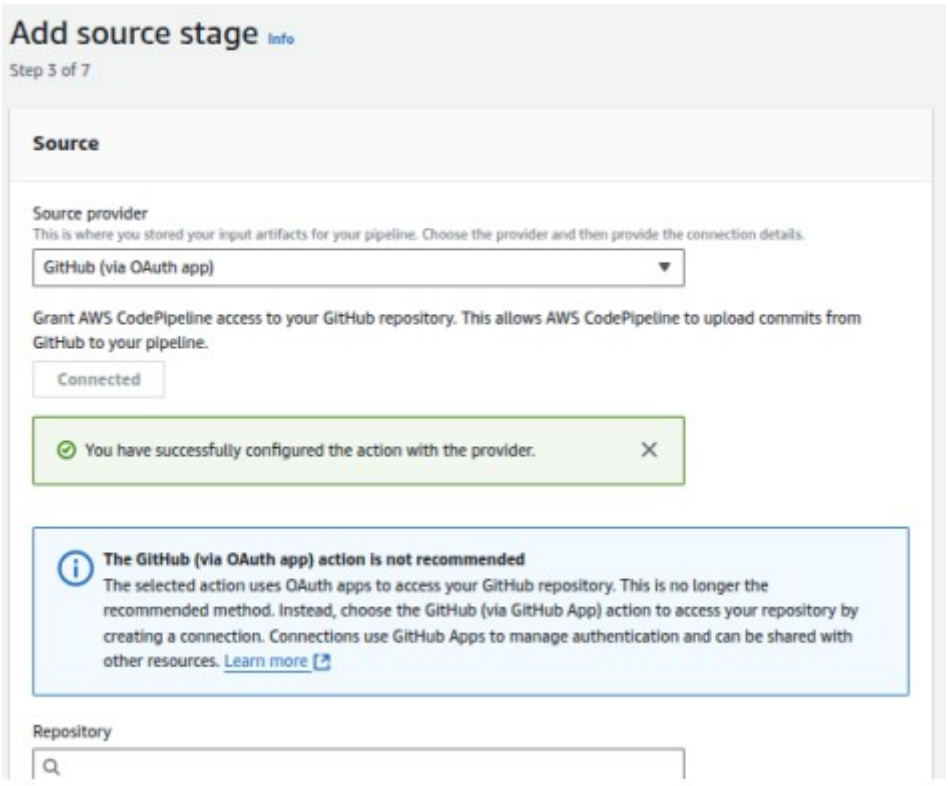

Elastic Beanstalk > Getting started
Create a web app
Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. Learn more
Application information
Application name
MyEBS
Up to 100 Unicode characters, not including forward slash (/).
Application tags
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more
Key
Value
EBS
CICD
Remove tag
Add tag
49 remaining

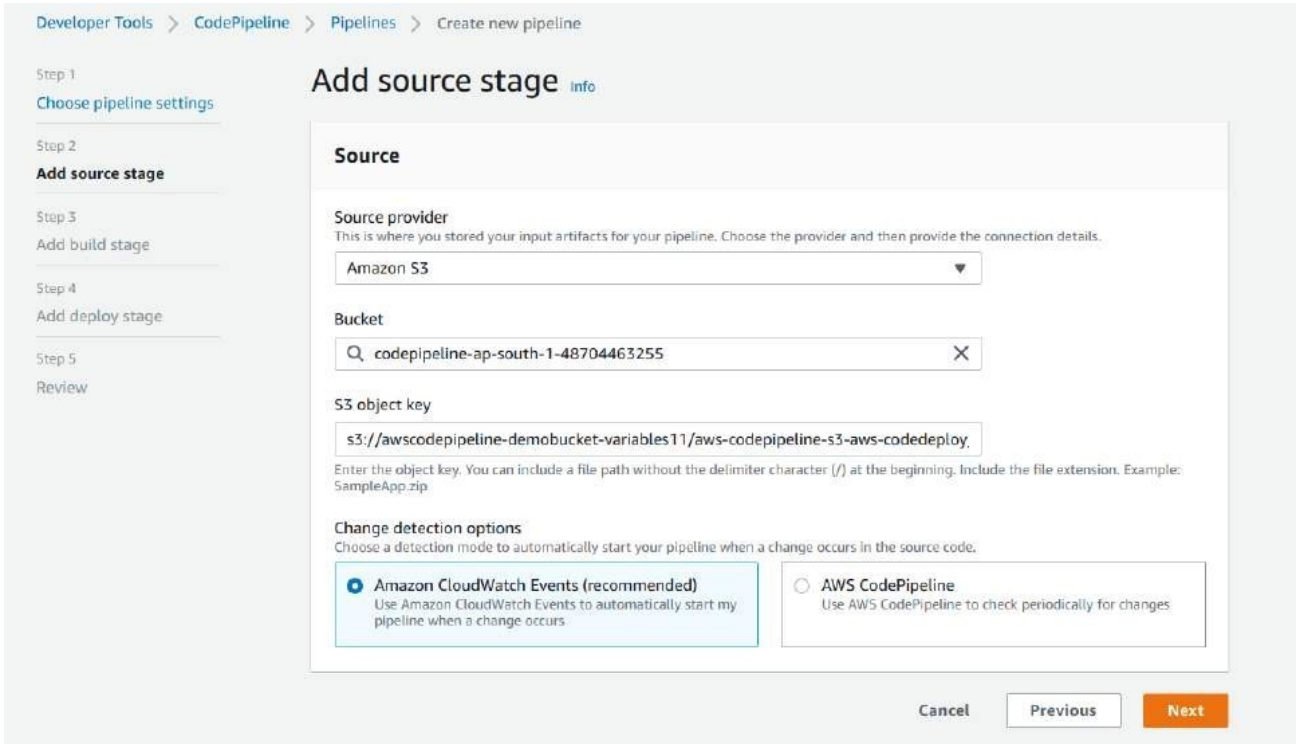

Successfully created bucket "development-bucket-pwa-venny"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
General purpose buckets
All AWS Regions
Directory buckets
General purpose buckets (2) Info
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3.
Find buckets by name
Name
AWS Region
Creation date
development-bucket-pwa-venny
Asia Pacific (Mumbai) ap-south-1
July 23, 2025, 10:58:25 (UTC+05:30)
store-build-venny
Asia Pacific (Mumbai) ap-south-1
July 23, 2025, 10:53:47 (UTC+05:30)
Account snapshot Info
Updated daily
View dashboard
Storage Lens provides visibility into storage usage and activity trends.
External access summary - new Info
Updated daily
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

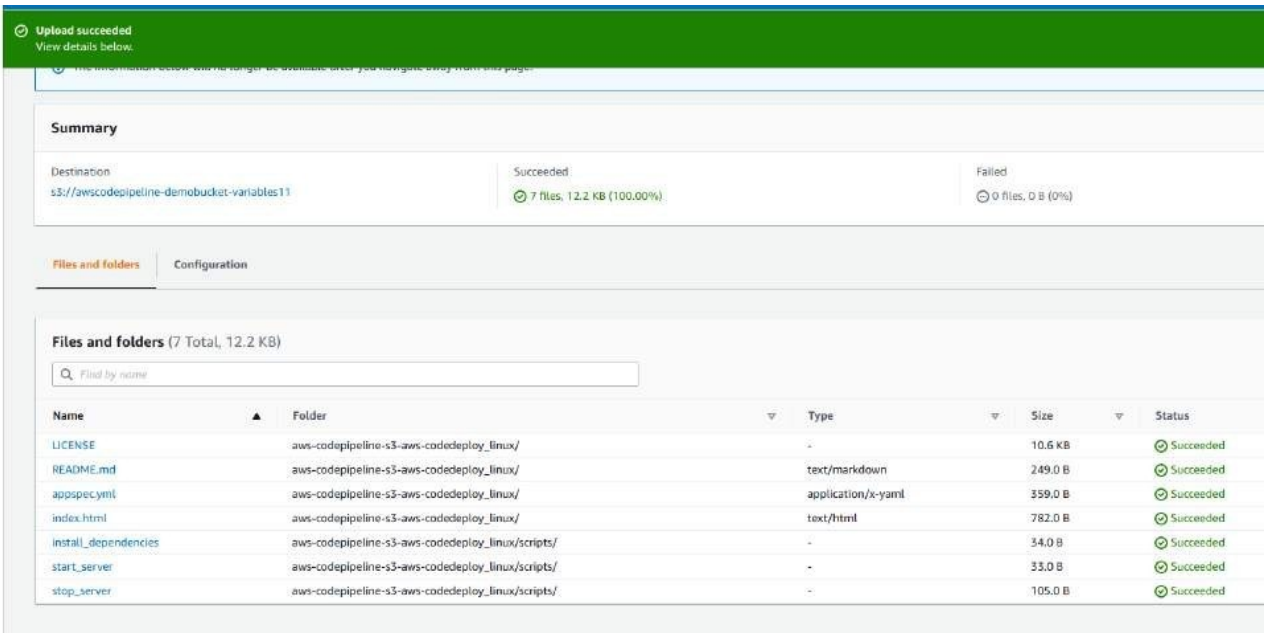

Manage default source credential
Source Provider
GitHub
Credential type
GitHub App
Connect project to GitHub using an AWS managed GitHub App
Personal access token
Connect project to GitHub using a personal access token
OAuth app
Connect project to GitHub using an OAuth app
Service
Secrets Manager (recommended)
Use Secrets Manager to store token
CodeBuild
Use CodeBuild managed token
Connect to GitHub

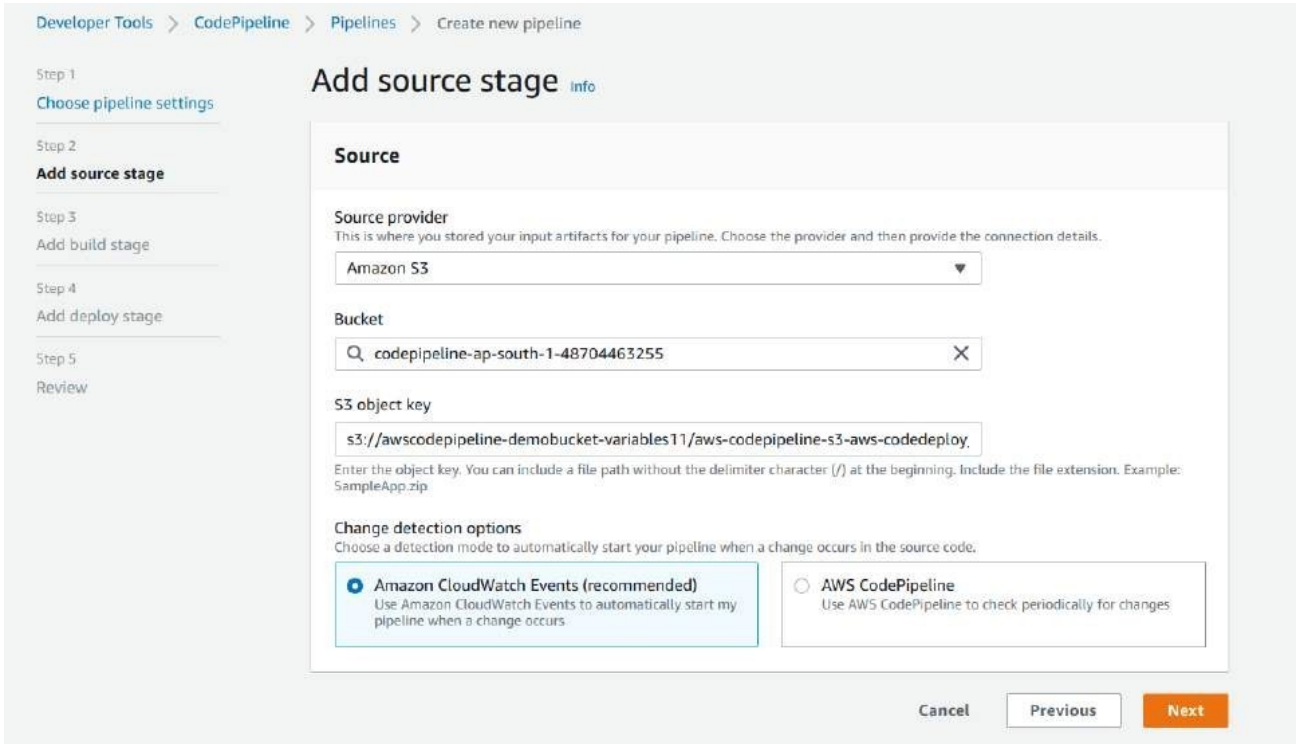

Configuration
Source provider
GitHub
Primary repository
Venny-Hong/Car-pwa-deploy-on-aws
Artifacts upload location
-
Service role
arn:aws:iam::863065069665:role/service-role/codebuild-pwa-codebuild-venny-service-role
Public builds
Disabled

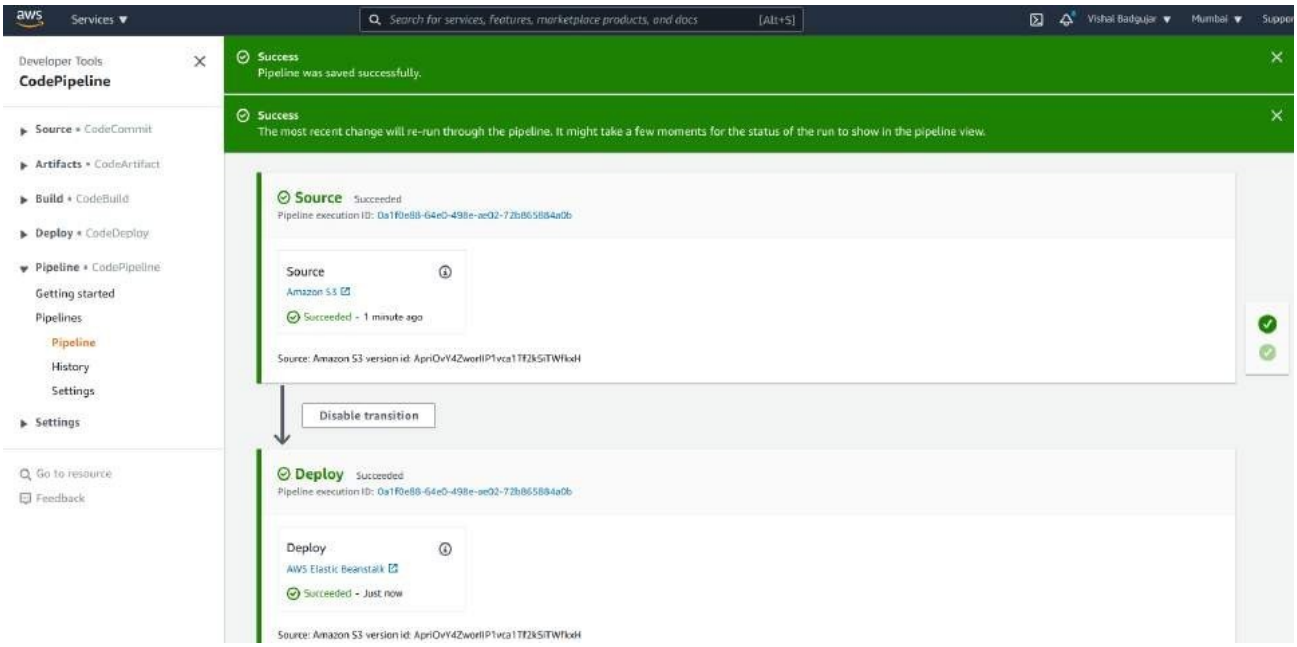

Advanced settings
Configure artifact store location, encryption settings, and pipeline variables for your pipeline.
Artifact store
Default location
Create a default S3 bucket in your account.
Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline
Bucket
store-build-venny
development-bucket-pwa-venny
store-build-venny
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.


Add source stage Info
Step 3 of 7
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (via OAuth app)
Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.
Connected
You have successfully configured the action with the provider.
The GitHub (via OAuth app) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (via GitHub App) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more
Repository


Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage info
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
Amazon S3
Bucket
codepipeline-ap-south-1-48704463255
S3 object key
s3://awscodepipeline-demobucket-variables11/aws-codepipeline-s3-aws-codedeploy.
Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
AWS CodePipeline
Use AWS CodePipeline to check periodically for changes
Cancel
Previous
Next


Upload succeeded
View details below.
Summary
Destination
s3://awscodepipeline-demobucket-variables11
Succeeded
7 files, 12.2 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (7 Total, 12.2 KB)
Find by name
Name
Folder
Type
Size
Status
LICENSE
aws-codepipeline-s3-aws-codedeploy_linux/
-
10.6 KB
Succeeded
README.md
aws-codepipeline-s3-aws-codedeploy_linux/
text/markdown
249.0 B
Succeeded
appspec.yml
aws-codepipeline-s3-aws-codedeploy_linux/
application/x-yaml
359.0 B
Succeeded
index.html
aws-codepipeline-s3-aws-codedeploy_linux/
text/html
782.0 B
Succeeded
install_dependencies
aws-codepipeline-s3-aws-codedeploy_linux/scripts/
-
34.0 B
Succeeded
start_server
aws-codepipeline-s3-aws-codedeploy_linux/scripts/
-
33.0 B
Succeeded
stop_server
aws-codepipeline-s3-aws-codedeploy_linux/scripts/
-
105.0 B
Succeeded


Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage Info
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
Amazon S3
Bucket
codepipeline-ap-south-1-48704463255
S3 object key
s3://awscodepipeline-demobucket-variables11/aws-codepipeline-s3-aws-codedeploy.
Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
AWS CodePipeline
Use AWS CodePipeline to check periodically for changes
Cancel
Previous
Next


Developer Tools
CodePipeline
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Settings
Success
Pipeline was saved successfully.
Success
The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.
Source Succeeded
Source
Amazon S3
Succeeded - 1 minute ago
Disable transition
Deploy Succeeded
Deploy
AWS Elastic Beanstalk
Succeeded - Just now

Conclusion

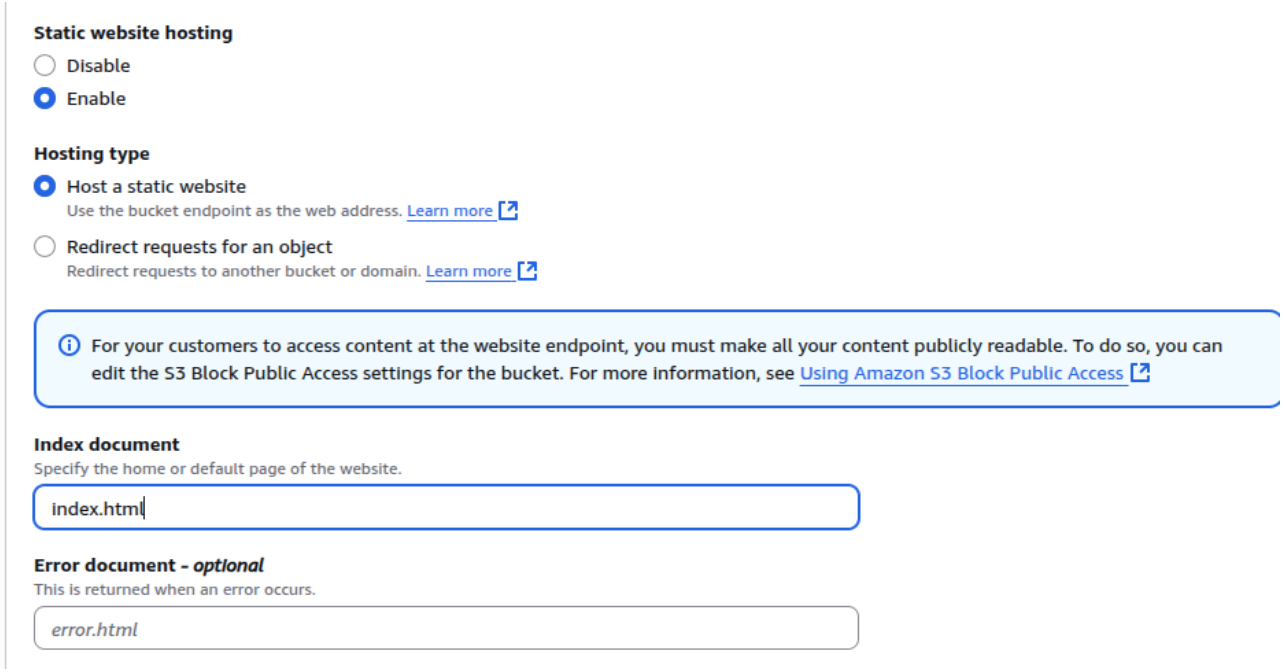
In this experiment, we successfully built and deployed a sample application using AWS CodePipeline. We created a deployment environment, uploaded the sample code to Amazon S3, and configured a pipeline to automate the deployment process. The pipeline retrieved the source code from S3 and deployed it to an Elastic Beanstalk environment. This demonstrated how AWS CodePipeline integrates with S3, Code Build, and Elastic Beanstalk to achieve continuous deployment, simplifying the release process and ensuring efficient application delivery.

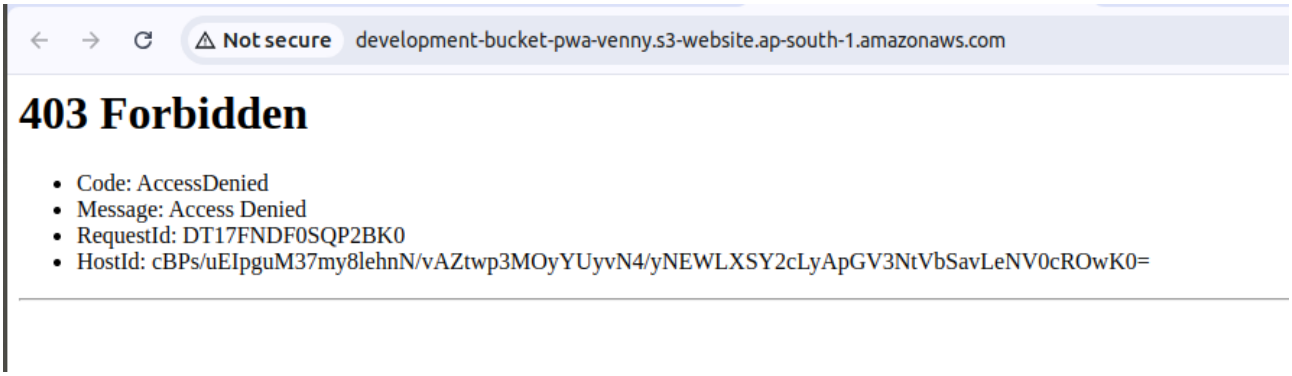
Semester: V
Academic Year: 2021-22
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Manjusha K.

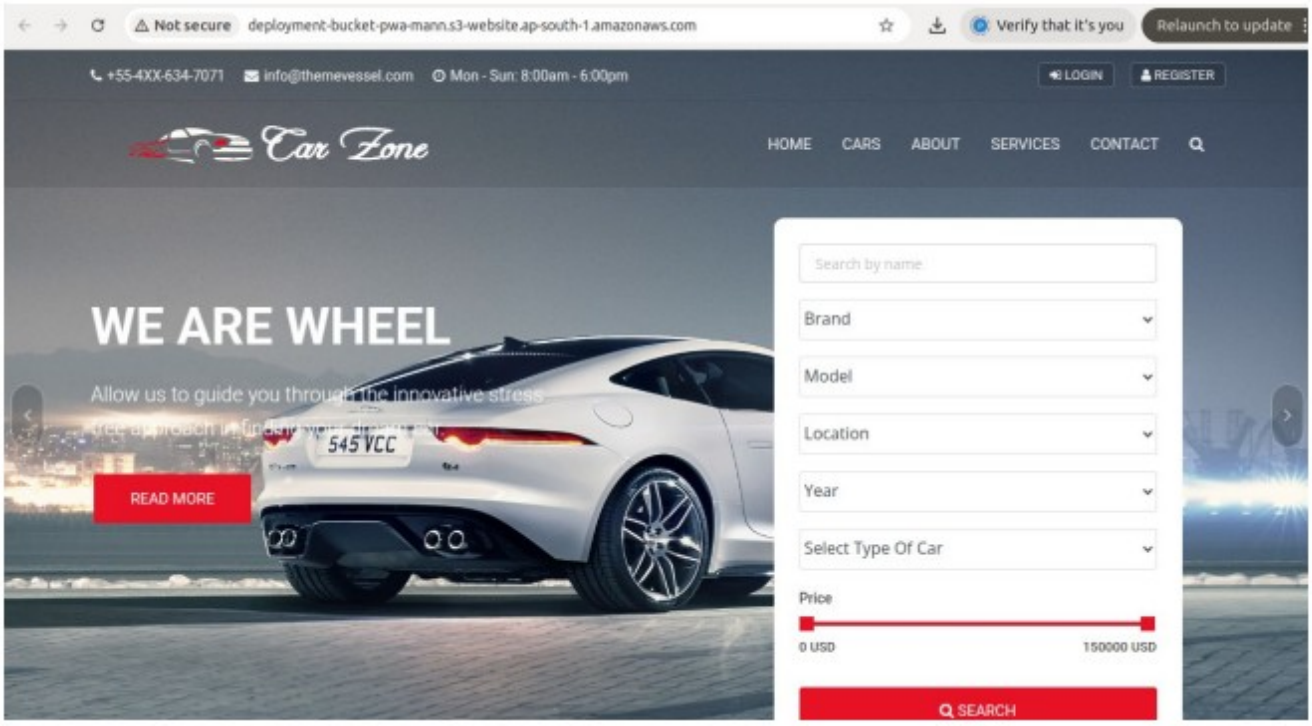
Name of Student:Huzaifa Bubere
ID:24204006

EXPERIMENT NO. 03

Aim: To deploy Sample Application on EC2 instance using AWS Code Deploy.




Not secure
development-bucket-pwa-venny.s3-website.ap-south-1.amazonaws.com
403 Forbidden
Code: AccessDenied
Message: Access Denied
RequestId: DT17FNDF0SQP2BK0
HostId: cBPs/uEIpguM37my8lehnN/vAZtwp3MOyYUyvN4/yNEWLXSY2cLyApGV3NtVbSavLeNV0cROwK0=


Not secure
deployment-bucket-pwa-mann.s3-website.ap-south-1.amazonaws.com
Verify that it's you
Relaunch to update
+55-4XX-634-7071
info@themevessel.com
Mon - Sun: 8:00am - 6:00pm
LOGIN
REGISTER
Car Zone
HOME
CARS
ABOUT
SERVICES
CONTACT
WE ARE WHEEL
READ MORE
Search by name
Brand
Model
Location
Year
Select Type Of Car
Price
0 USD
150000 USD
SEARCH

Conclusion

In this experiment, we successfully built and deployed a sample application using AWS CodePipeline. We created a deployment environment, uploaded the sample code to Amazon S3, and configured a pipeline to automate the deployment process. The pipeline retrieved the source code from S3 and deployed it to an Elastic Beanstalk environment. This demonstrated how AWS CodePipeline integrates with S3, Code Build, and Elastic Beanstalk to achieve continuous deployment, simplifying the release process and ensuring efficient application delivery.

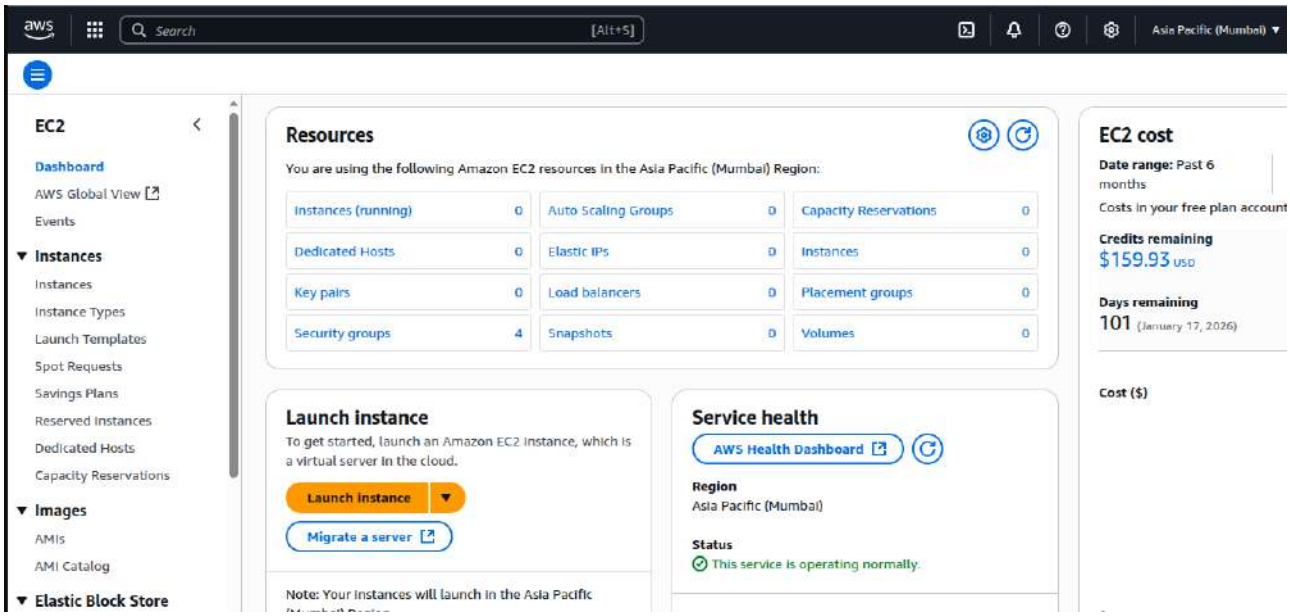
Semester: V
Academic Year: 2025-26
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor:Prof.Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID: 24204006

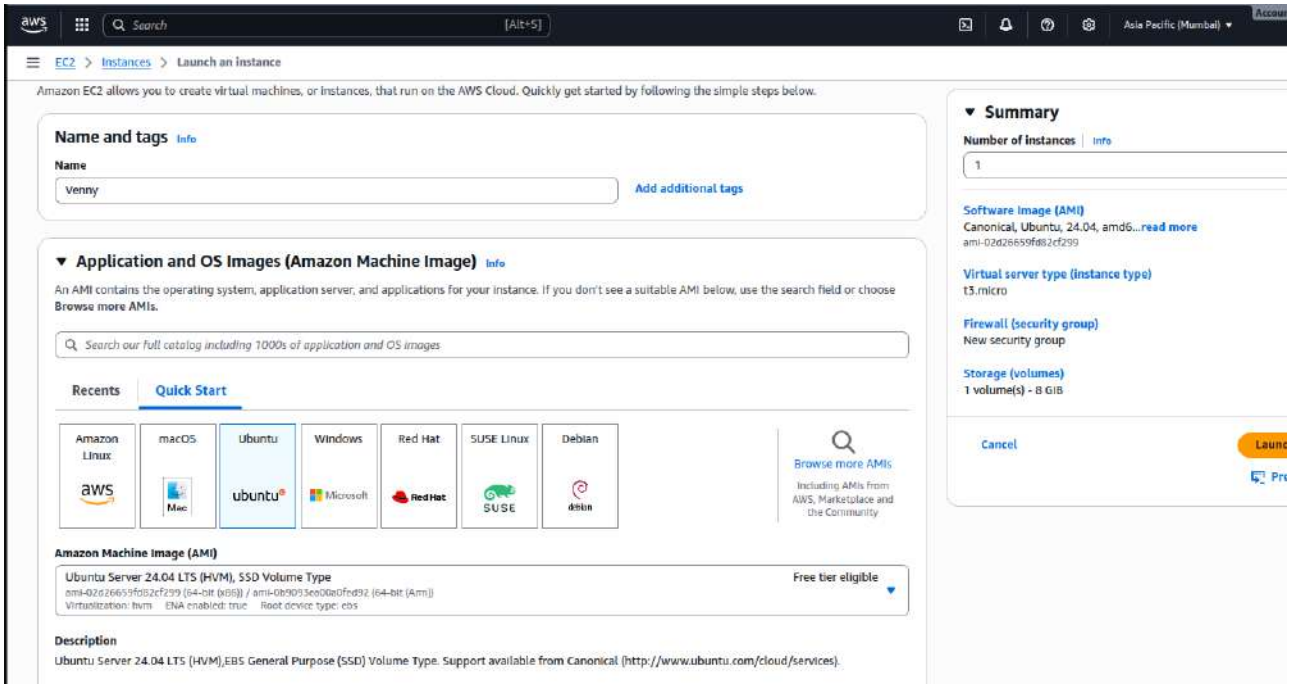
EXPERIMENT NO. 04

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

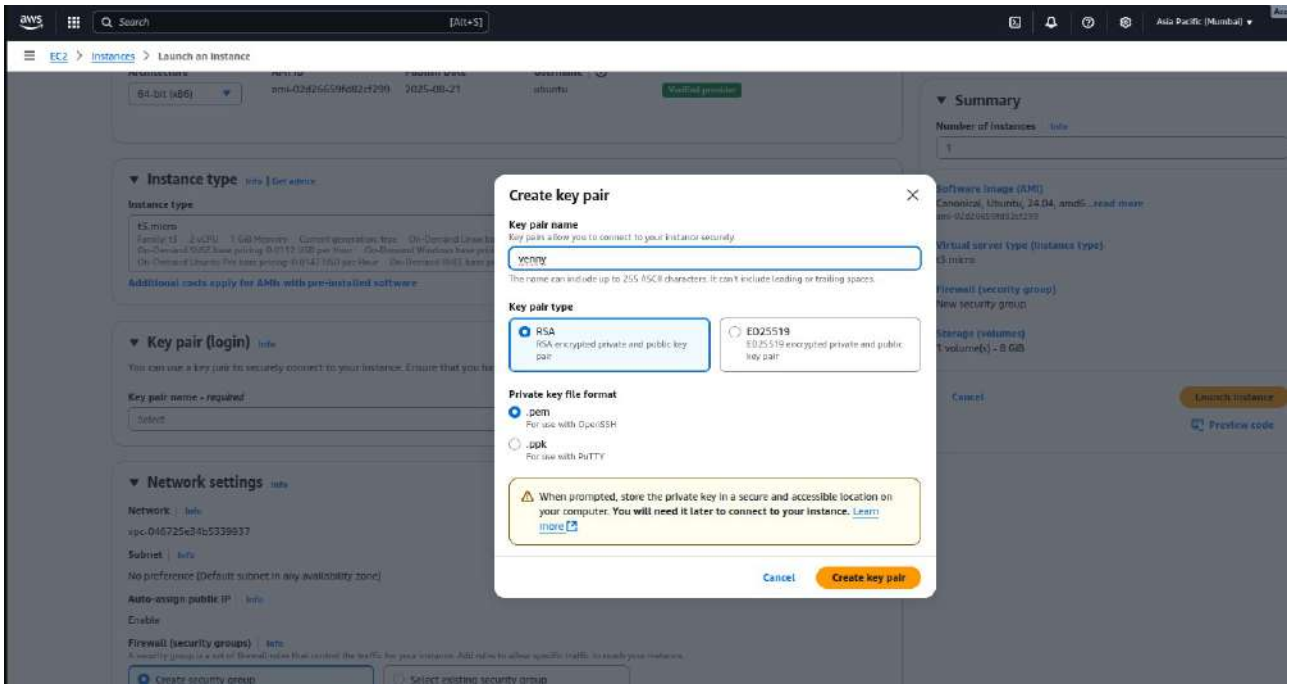
STEP 1: Check security group,delete all SG only keep default



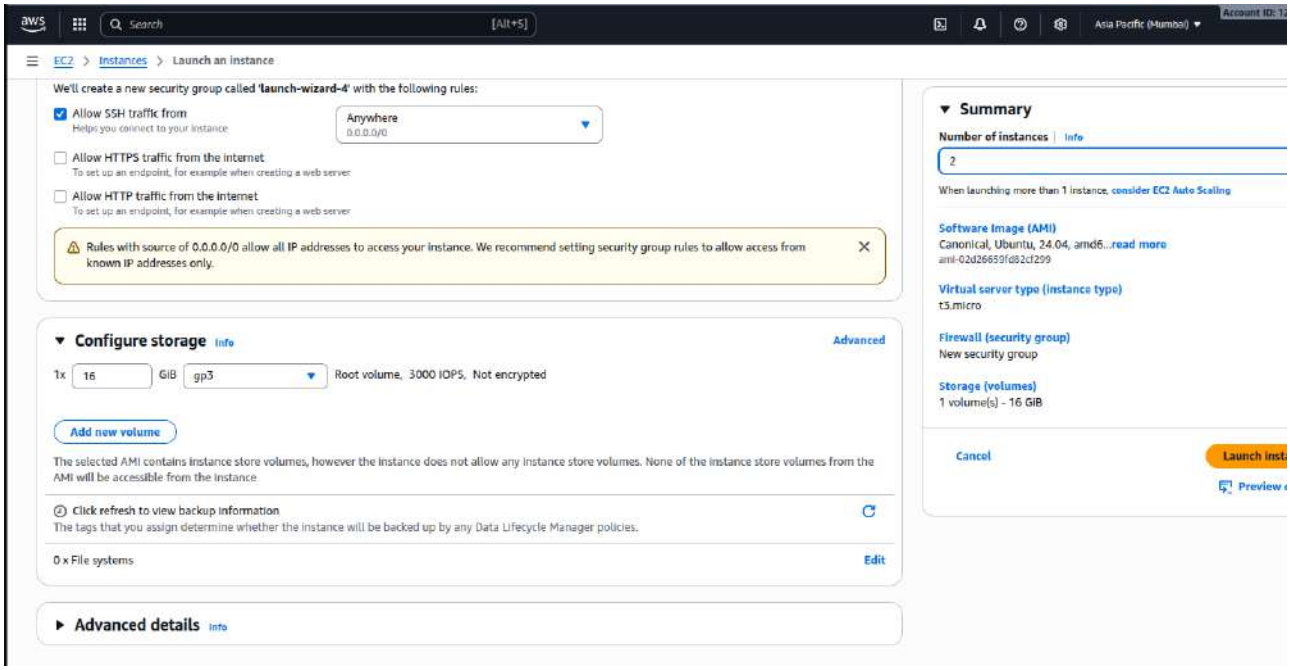
Create 2 instance



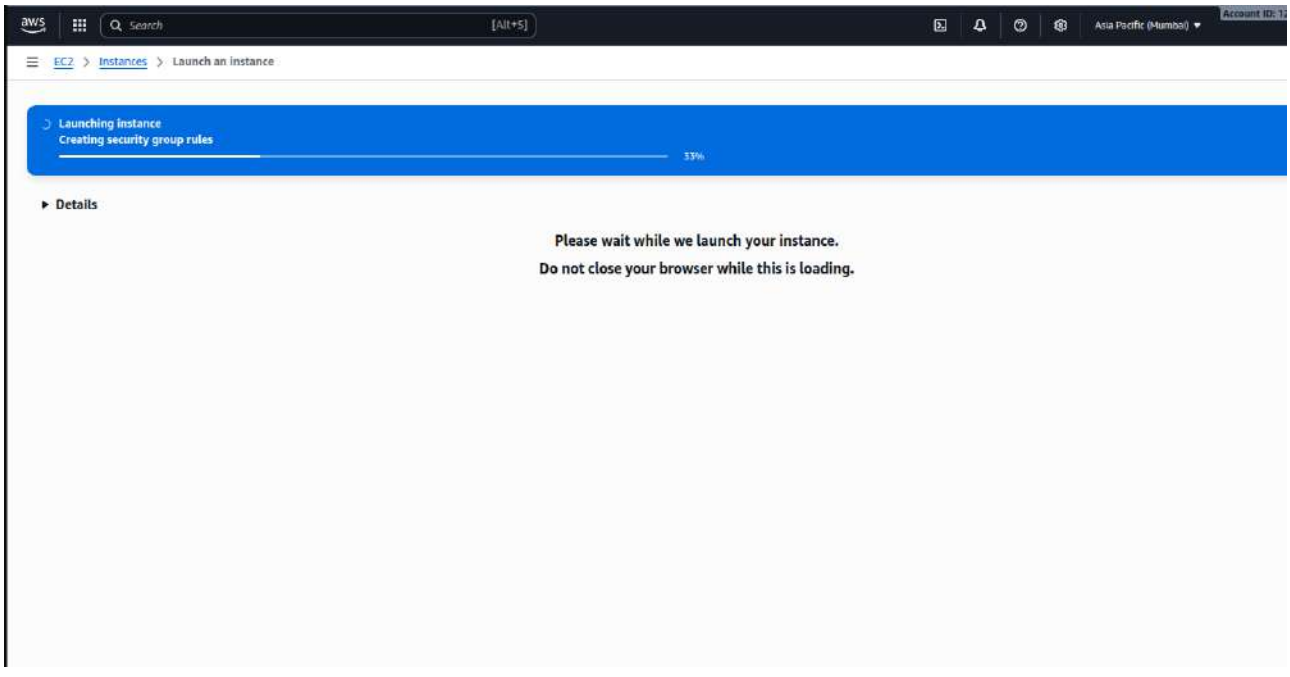
Create key pair



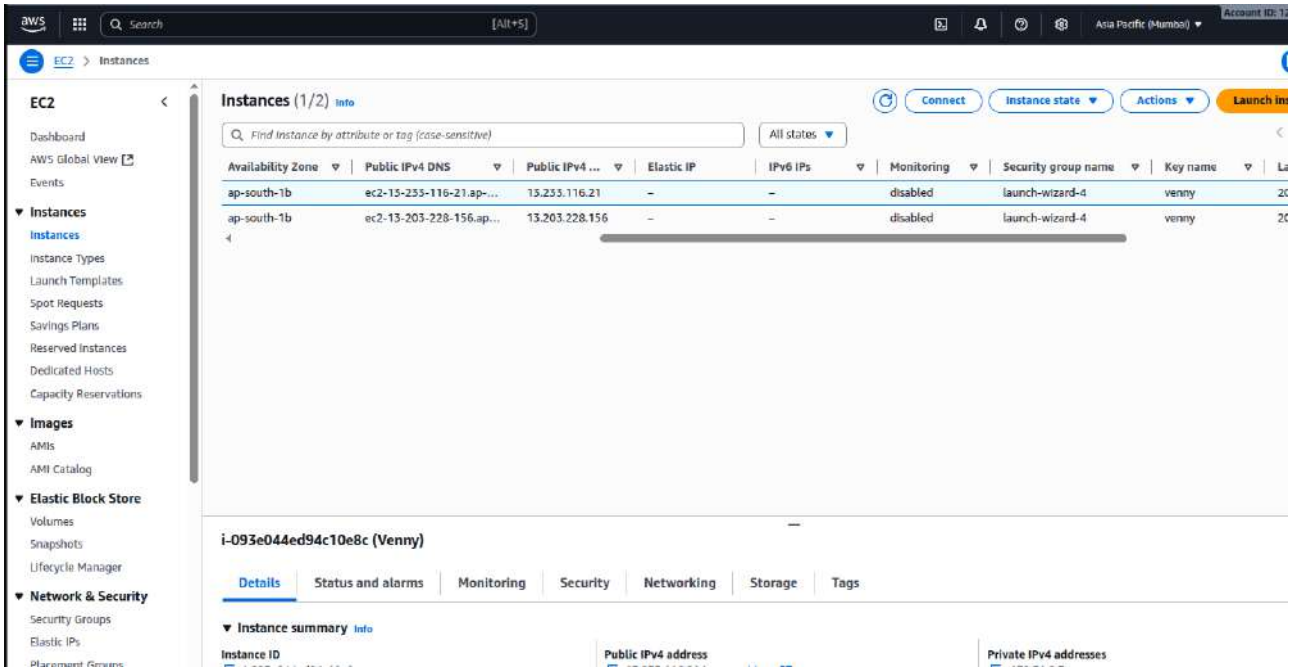
Create security group and allow traffic



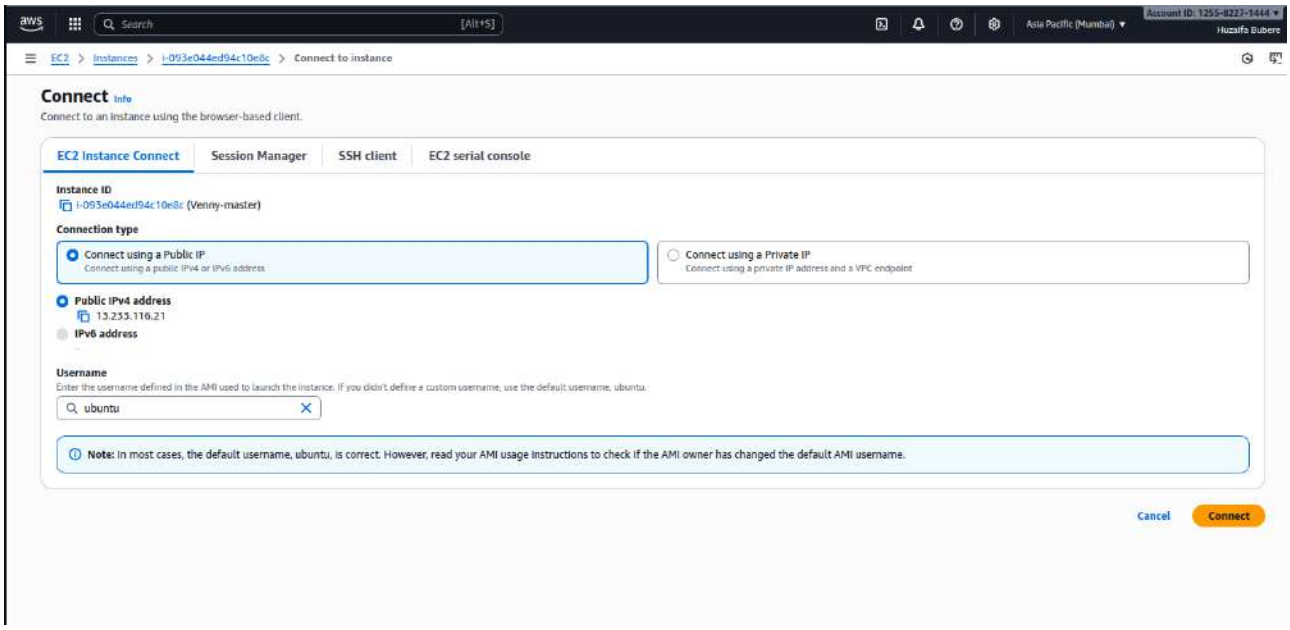
Launch Instance



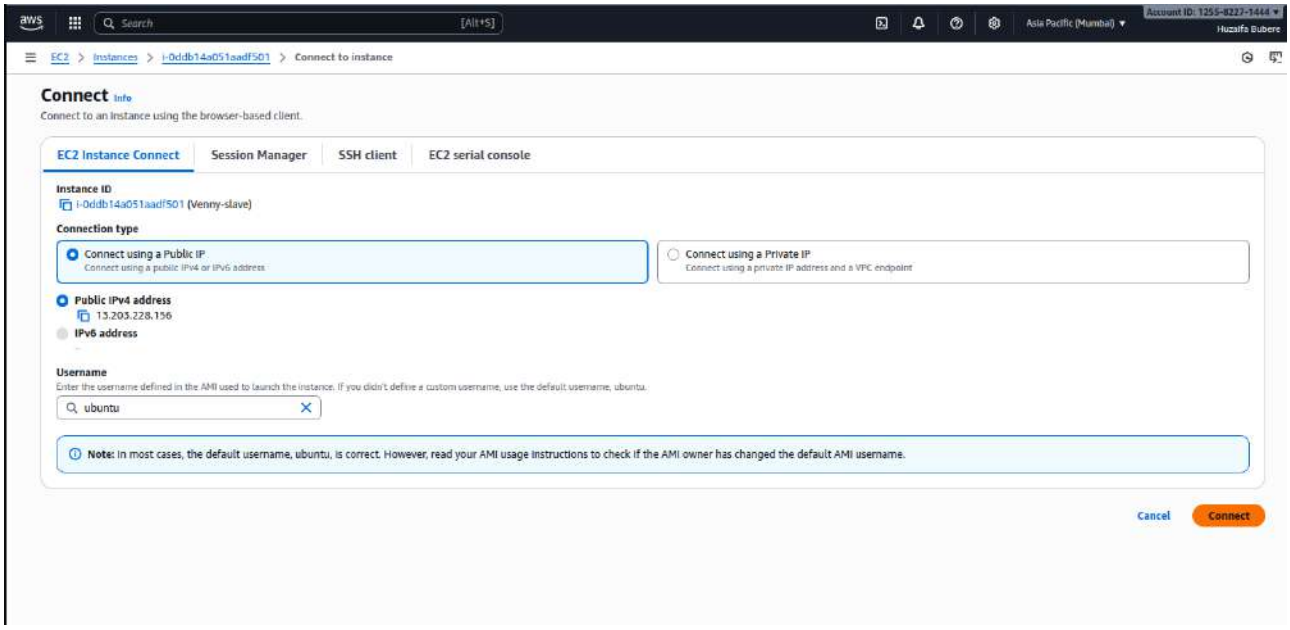
Check security group of both instances



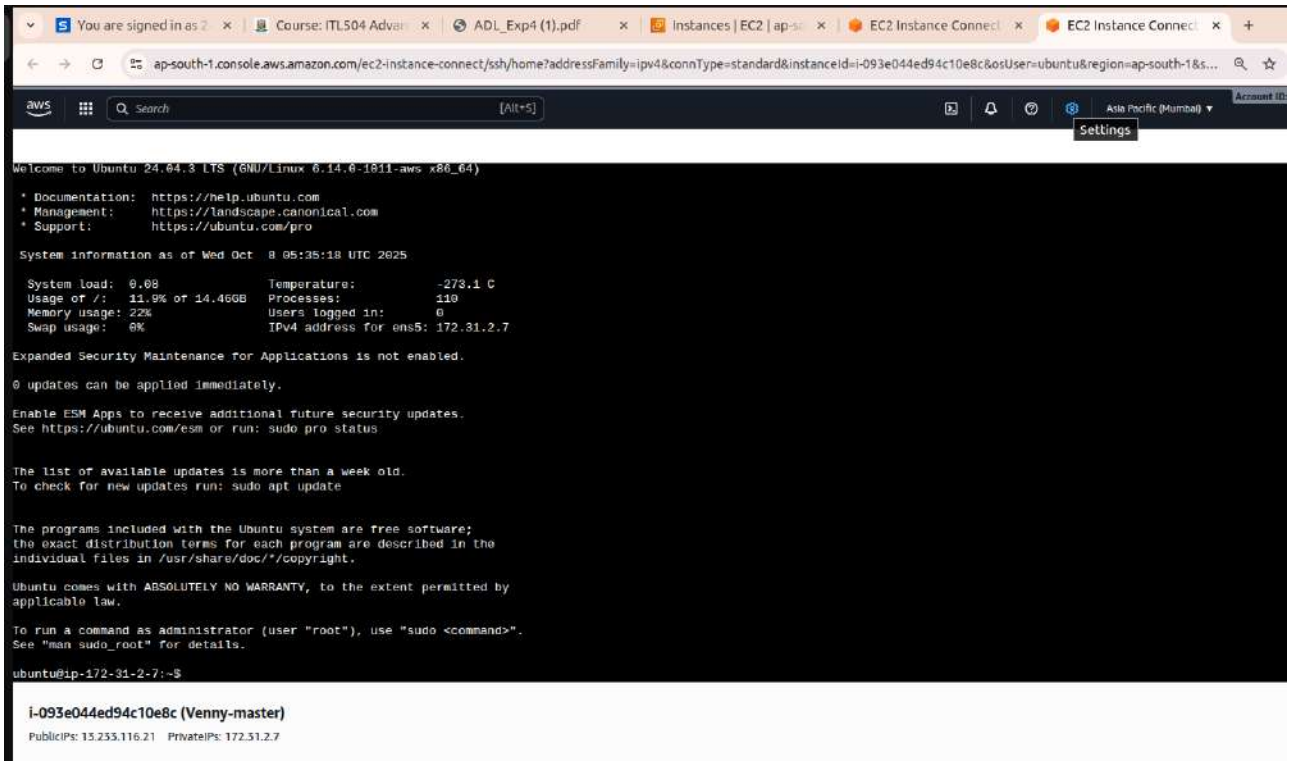
Click on connect



Sameway connect to slave-node



After Connecting



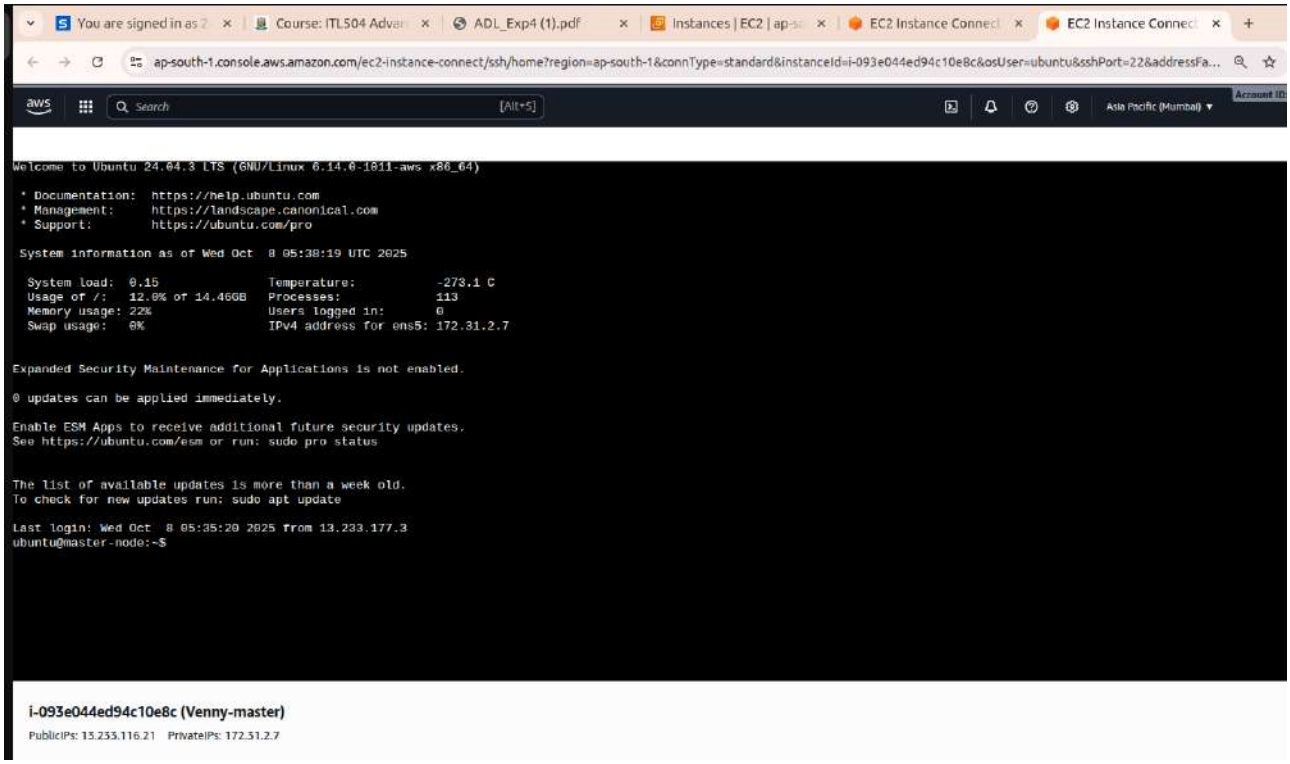
Step 2:

Assign Unique Hostname for Each Server Node

$ sudo hostnamectl set-hostname master-node

Than exit

Refresh



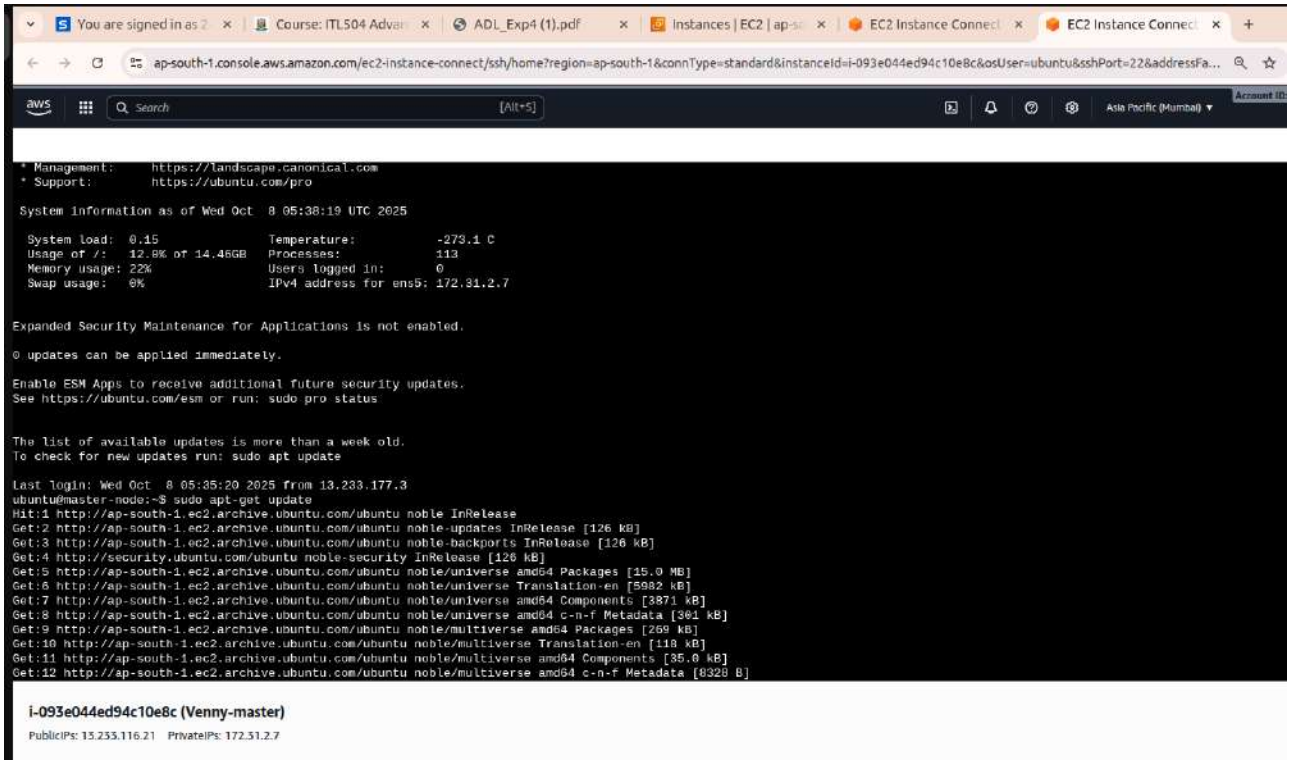
Next, set a worker node hostname by entering the following on the worker server:

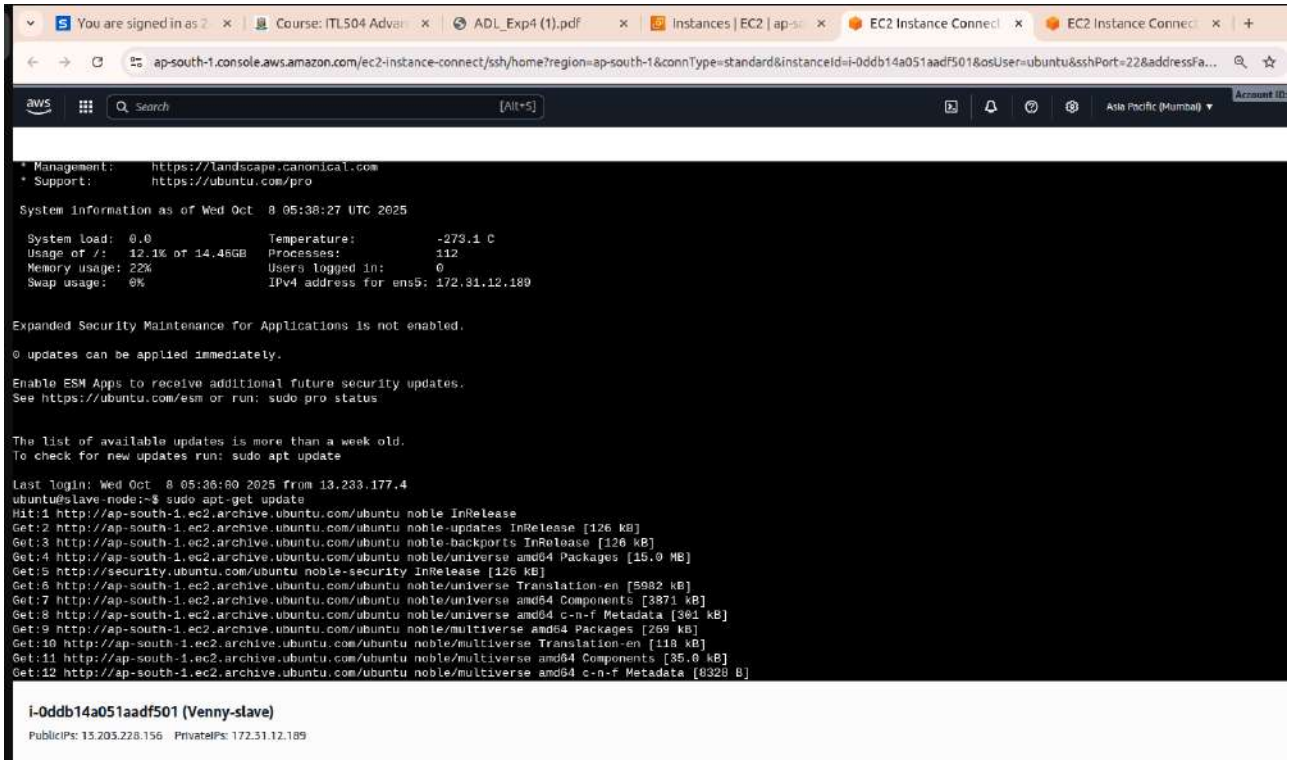
$ sudo hostnamectl set-hostname worker1

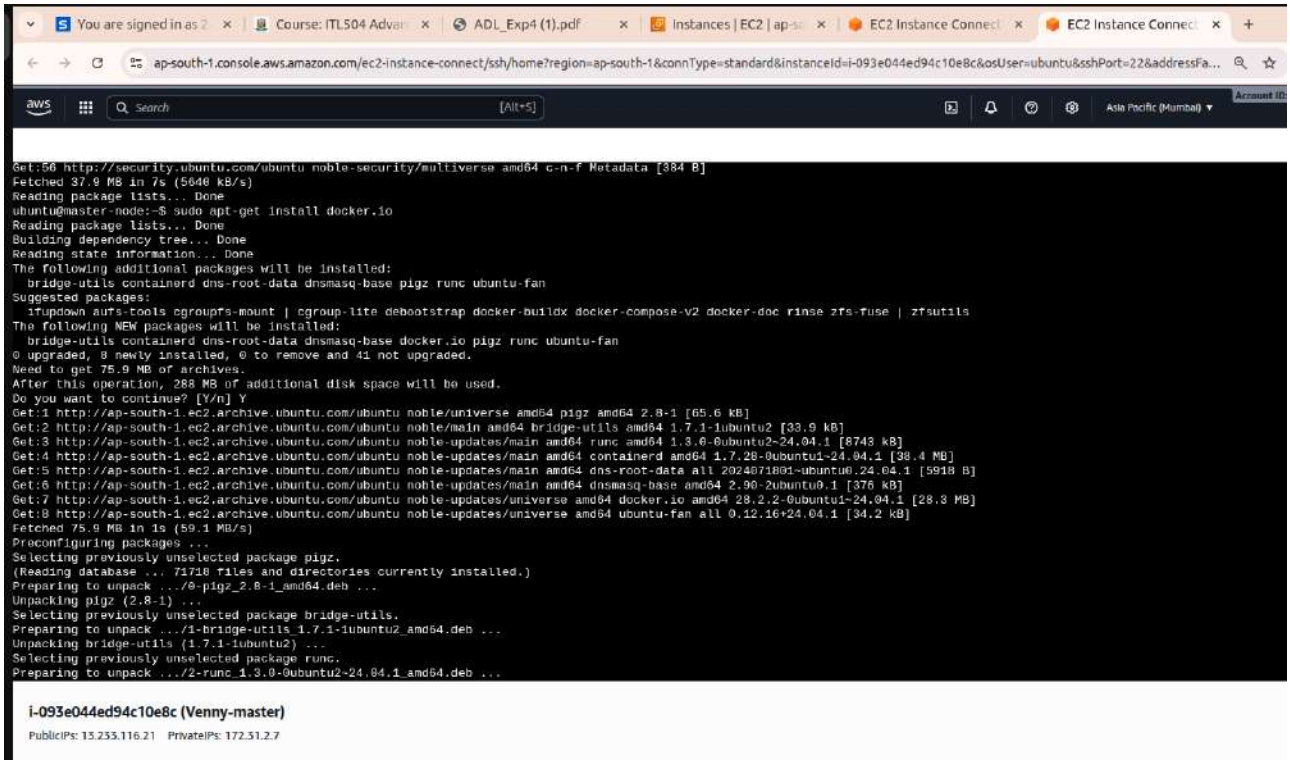
STEP 3: On both master and worker1

$ sudo apt-get update

STEP 4: On both master and worker1


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-093e044ed94c10e8c&osUser=ubuntu&sshPort=22&addressFa...
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Wed Oct 8 05:38:19 UTC 2025
System load: 0.15 Temperature: -273.1 C
Usage of /: 12.8% of 14.46GB Processes: 113
Memory usage: 22% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.2.7
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Wed Oct 8 05:35:20 2025 from 13.233.177.3
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
i-093e044ed94c10e8c (Venny-master)
PublicIPs: 13.233.116.21 PrivateIPs: 172.31.2.7


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0ddb14a051aadf501&osUser=ubuntu&sshPort=22&addressFa...
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Wed Oct 8 05:38:27 UTC 2025
System load: 0.0 Temperature: -273.1 C
Usage of /: 12.1% of 14.46GB Processes: 112
Memory usage: 22% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.12.189
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Wed Oct 8 05:36:00 2025 from 13.233.177.4
ubuntu@slave-node:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
i-0ddb14a051aadf501 (Venny-slave)
PublicIPs: 13.203.228.156 PrivateIPs: 172.31.12.189



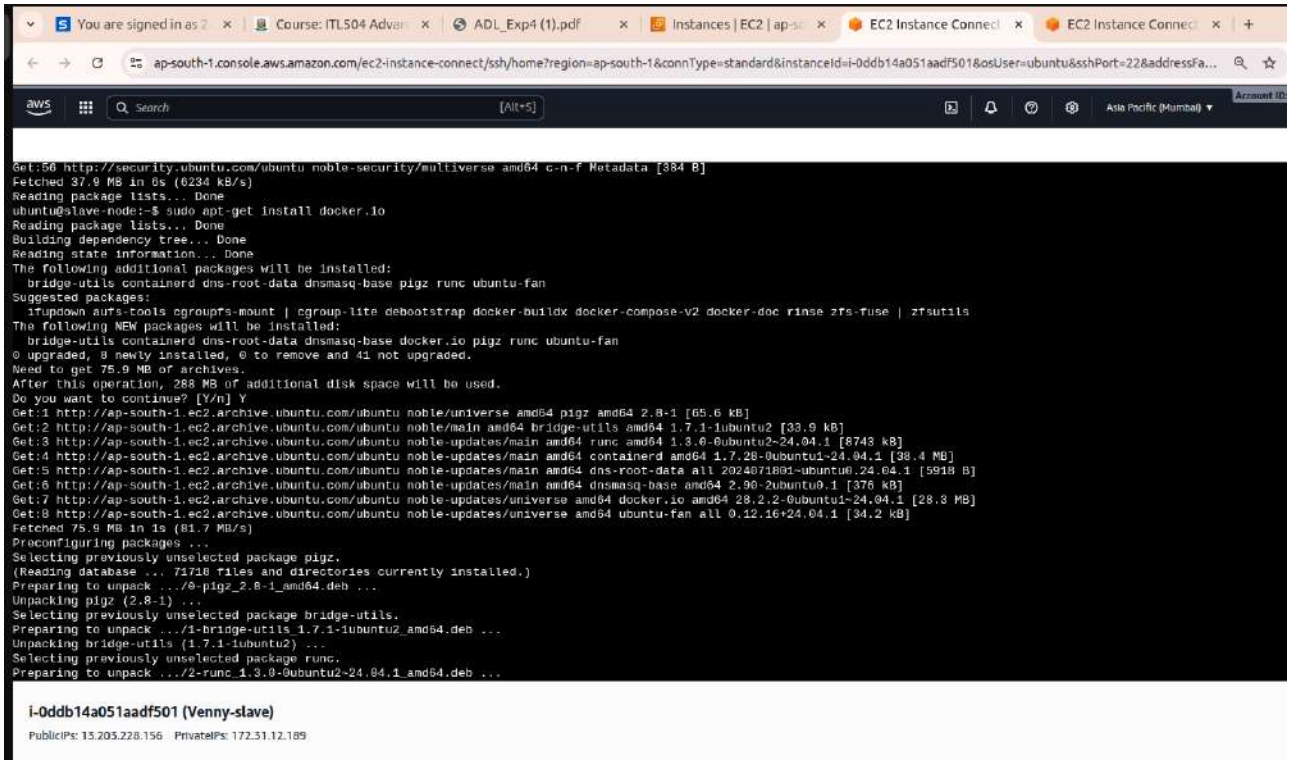
Install docker sudo apt-get install docker.io

STEP 5 :

Start and Enable Docker Set Docker to launch at boot by entering the following:

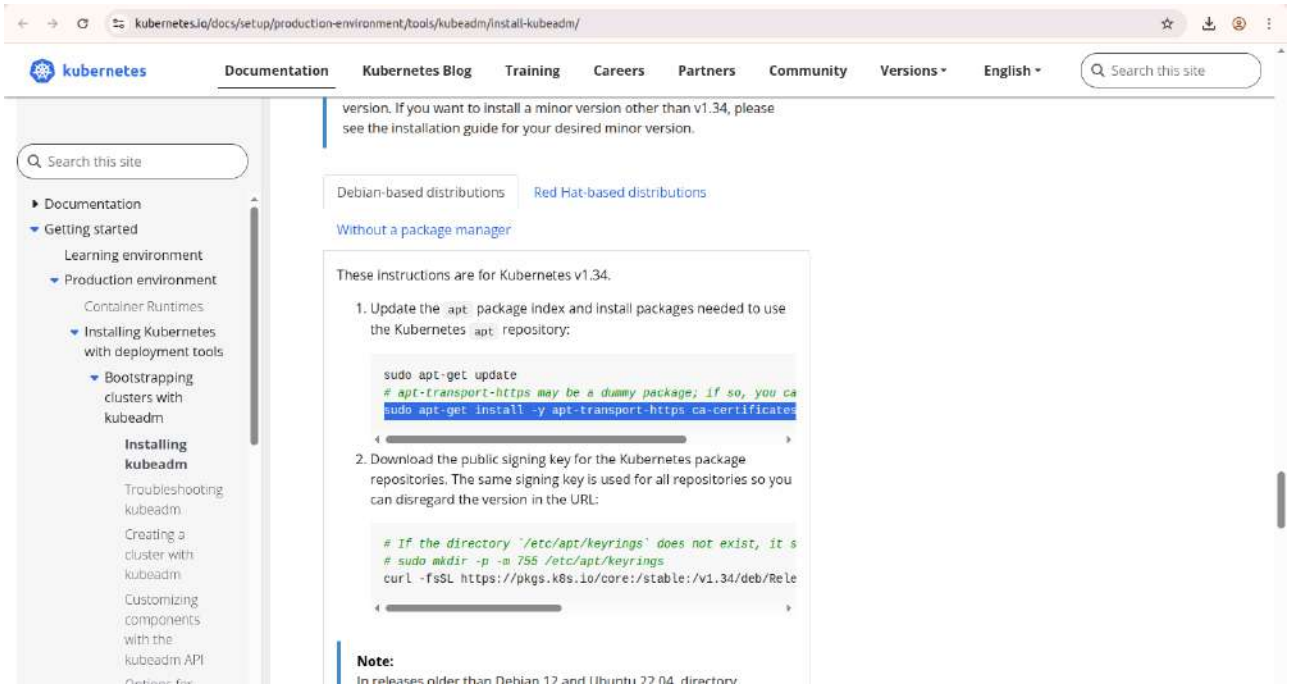
$ sudo systemctl enable docker

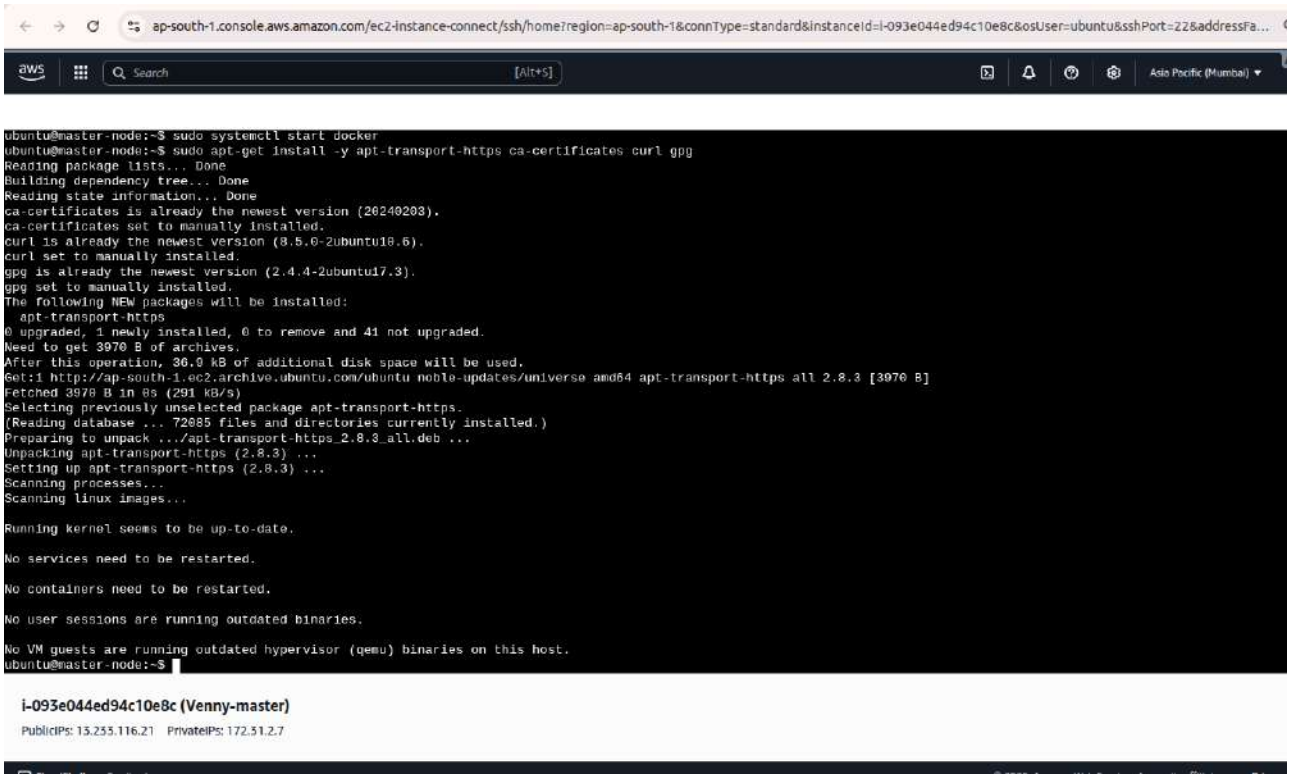
$ sudo systemctl status docker

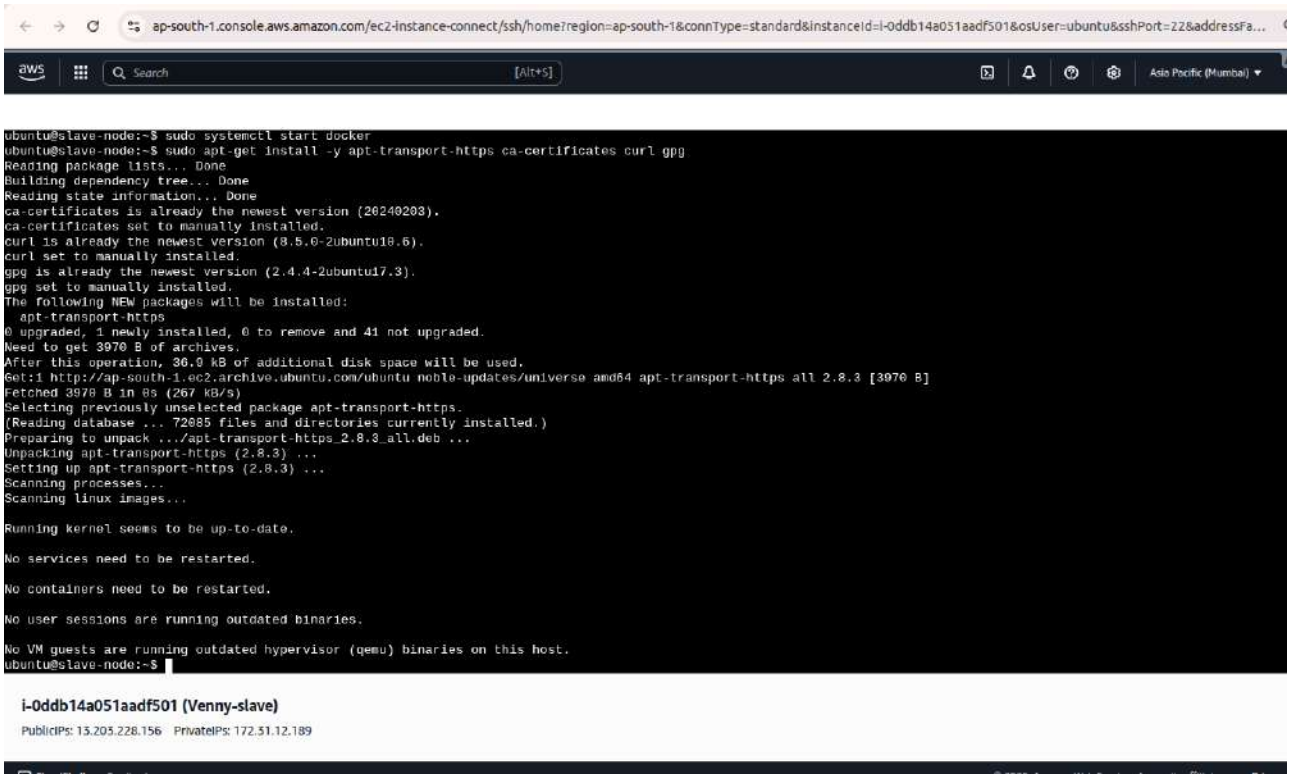


STEP 6 Install Kubernetes

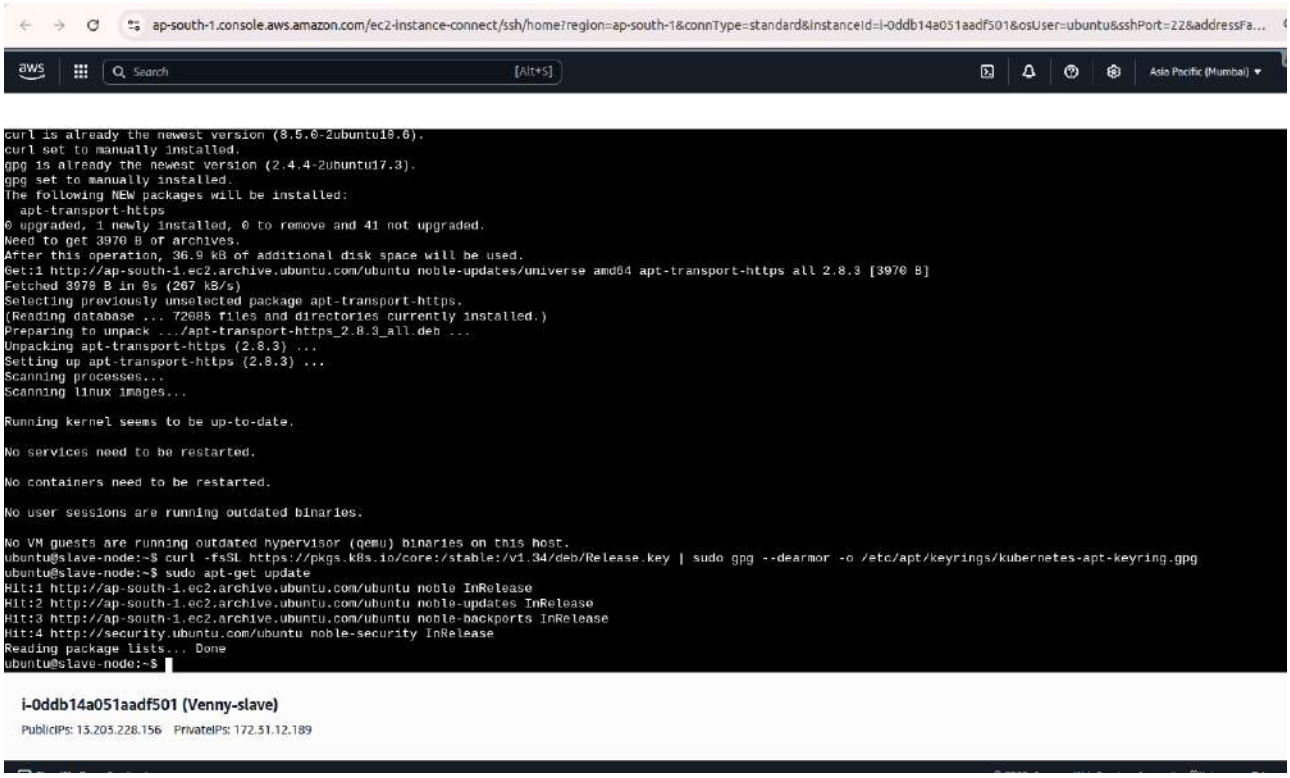
https://kubernetes.io/docs/setup/productionenvironment/tools/kubeadm/install-kubeadm/

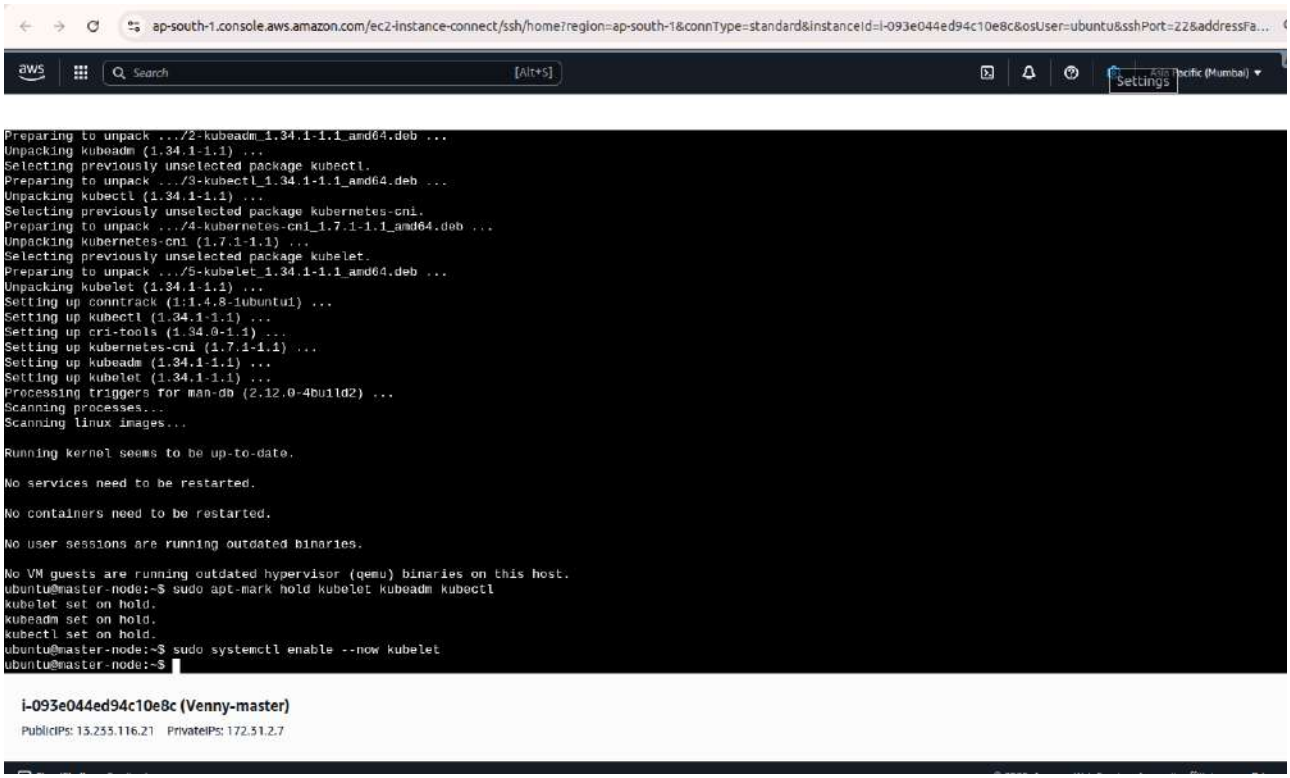

kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
kubernetes
Documentation
Kubernetes Blog
Training
Careers
Partners
Community
Versions
English
Search this site
version. If you want to install a minor version other than v1.34, please see the installation guide for your desired minor version.
Debian-based distributions
Red Hat-based distributions
Without a package manager
These instructions are for Kubernetes v1.34.
1. Update the apt package index and install packages needed to use the Kubernetes apt repository:
sudo apt-get update
# apt-transport-https may be a dummy package; if so, you ca
sudo apt-get install -y apt-transport-https ca-certificates
2. Download the public signing key for the Kubernetes package repositories. The same signing key is used for all repositories so you can disregard the version in the URL:
If the directory `/etc/apt/keyrings` does not exist, it s
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.34/deb/Rele
Note:
Documentation
Getting started
Learning environment
Production environment
Container Runtimes
Installing Kubernetes with deployment tools
Bootstrapping clusters with kubeadm
Installing kubeadm
Troubleshooting kubeadm
Creating a cluster with kubeadm
Customizing components with the kubeadm API

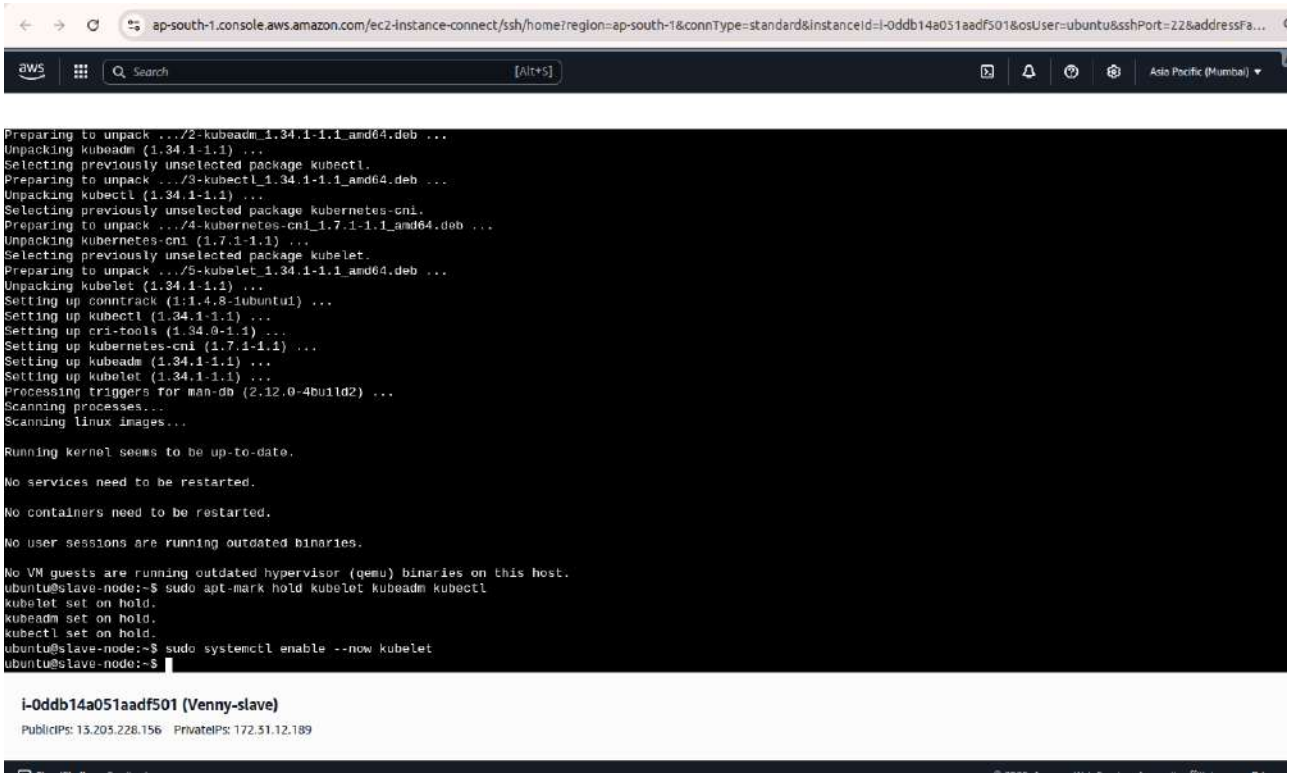

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-093e044ed94c10e8c&osUser=ubuntu&sshPort=22&addressFa...
aws
Search
[Alt+S]
Asia Pacific (Mumbai)
ubuntu@master-node:~$ sudo systemctl start docker
ubuntu@master-node:~$ sudo apt-get install -y apt-transport-https ca-certificates curl gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gpg is already the newest version (2.4.4-2ubuntu17.3).
gpg set to manually installed.
The following NEW packages will be installed:
apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 41 not upgraded.
Need to get 3970 B of archives.
After this operation, 36.0 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 apt-transport-https all 2.8.3 [3970 B]
Fetched 3970 B in 0s (291 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 72085 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.8.3_all.deb ...
Unpacking apt-transport-https (2.8.3) ...
Setting up apt-transport-https (2.8.3) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$
i-093e044ed94c10e8c (Venny-master)
PublicIPs: 13.233.116.21 PrivateIPs: 172.51.2.7

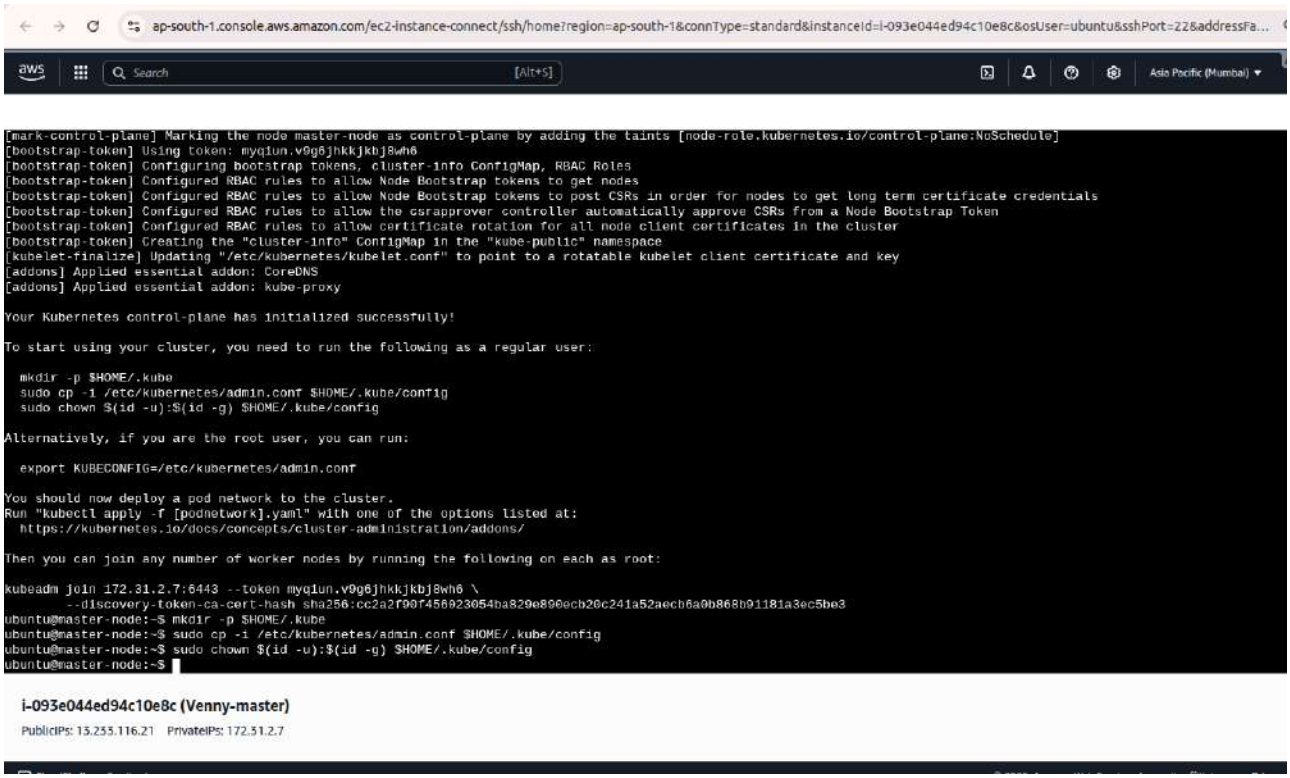


Signining key

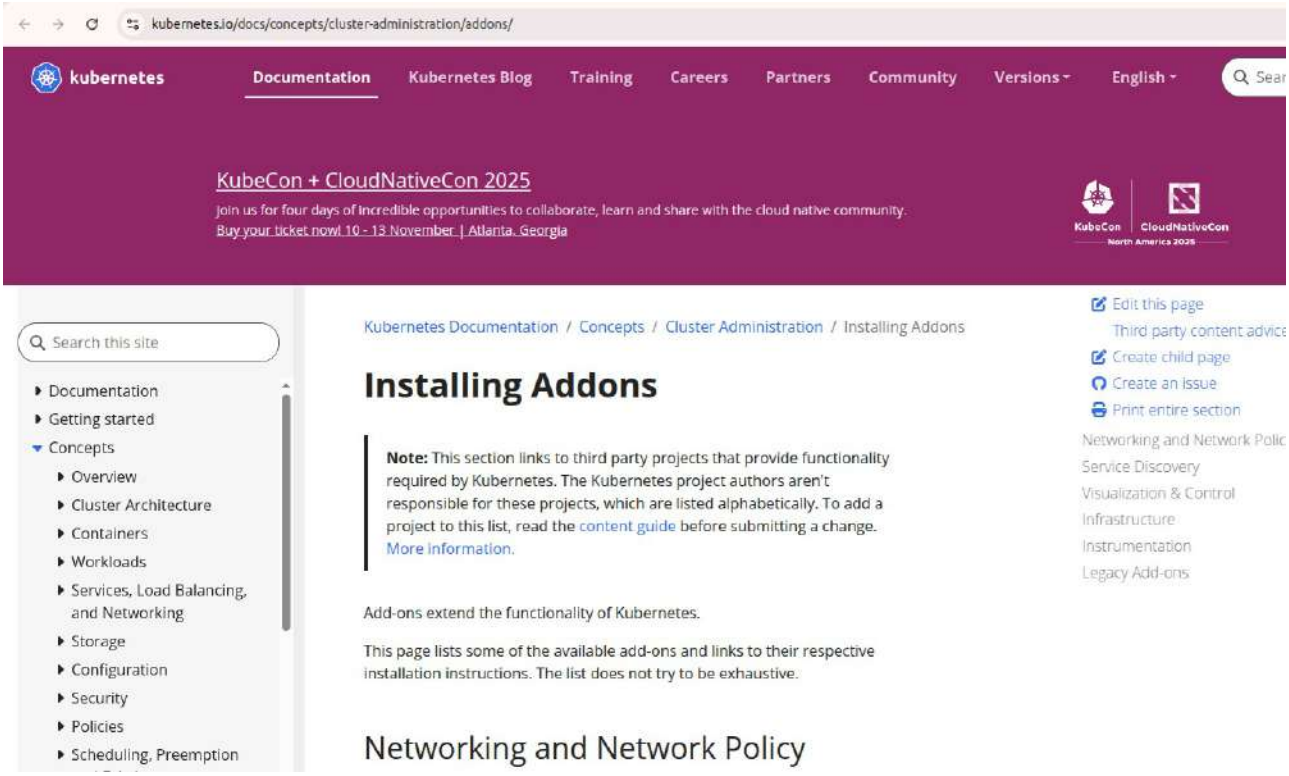



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-093e044ed94c10e8c&osUser=ubuntu&sshPort=22&addressFa...
Search
[Alt+S]
Settings
Preparing to unpack .../2-kubeadm_1.34.1-1.1_amd64.deb ...
Unpacking kubeadm (1.34.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.34.1-1.1_amd64.deb ...
Unpacking kubectl (1.34.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.7.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.7.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.34.1-1.1_amd64.deb ...
Unpacking kubelet (1.34.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.34.1-1.1) ...
Setting up cri-tools (1.34.0-1.1) ...
Setting up kubernetes-cni (1.7.1-1.1) ...
Setting up kubeadm (1.34.1-1.1) ...
Setting up kubelet (1.34.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@master-node:~$ sudo systemctl enable --now kubelet
ubuntu@master-node:~$
i-093e044ed94c10e8c (Venny-master)
PublicIPs: 13.233.116.21 PrivateIPs: 172.51.2.7

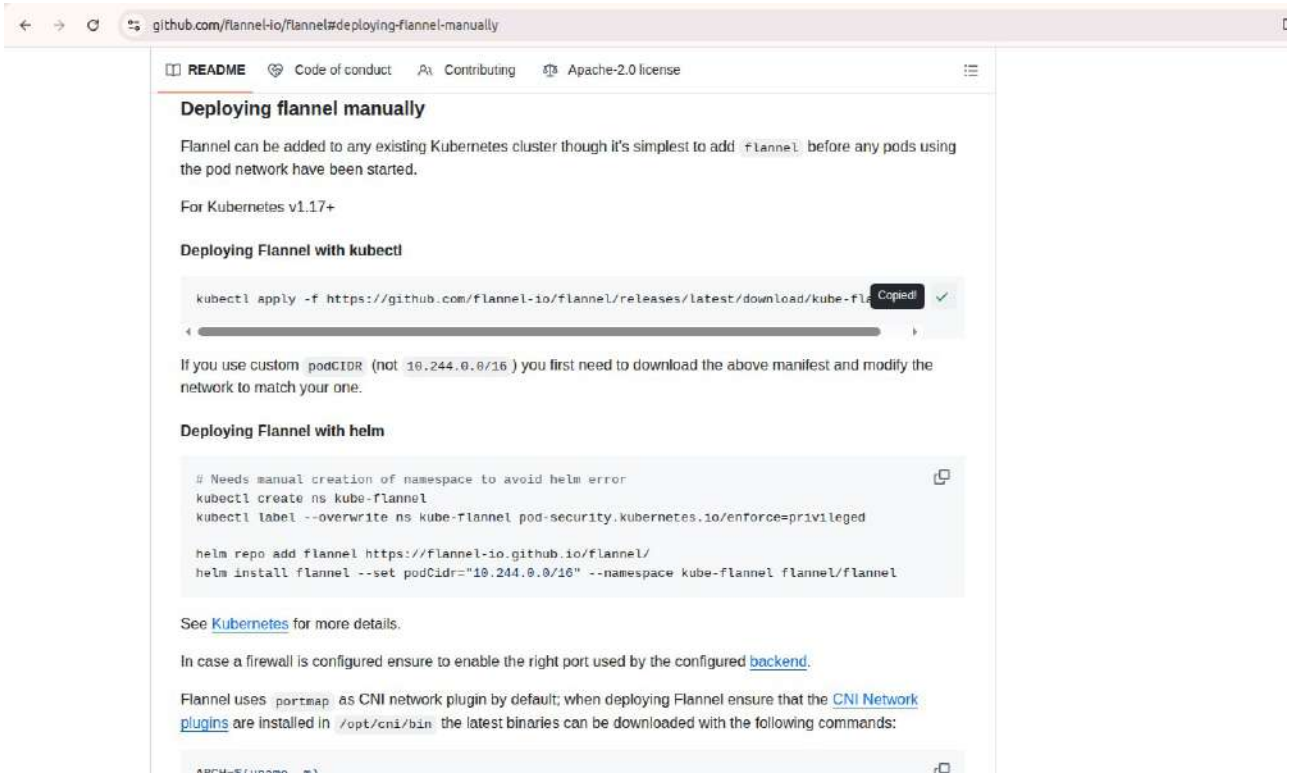

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0ddb14a051aadf501&osUser=ubuntu&sshPort=22&addressFa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Preparing to unpack .../2-kubeadm_1.34.1-1.1_amd64.deb ...
Unpacking kubeadm (1.34.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.34.1-1.1_amd64.deb ...
Unpacking kubectl (1.34.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.7.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.7.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.34.1-1.1_amd64.deb ...
Unpacking kubelet (1.34.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.34.1-1.1) ...
Setting up cri-tools (1.34.0-1.1) ...
Setting up kubernetes-cni (1.7.1-1.1) ...
Setting up kubeadm (1.34.1-1.1) ...
Setting up kubelet (1.34.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@slave-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@slave-node:~$ sudo systemctl enable --now kubelet
ubuntu@slave-node:~$
i-0ddb14a051aadf501 (Venny-slave)
PublicIPs: 13.205.228.156 PrivateIPs: 172.51.12.189



STEP 10 Copy weblink from masternode https://kubernetes.io/docs/concepts/cluster-administration/addons/



goto flannel copy this command and paste on master



Conclusion: In this experiment We Have Learnt About Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Semester: V
Academic Year: 2025-26
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID: 24204006

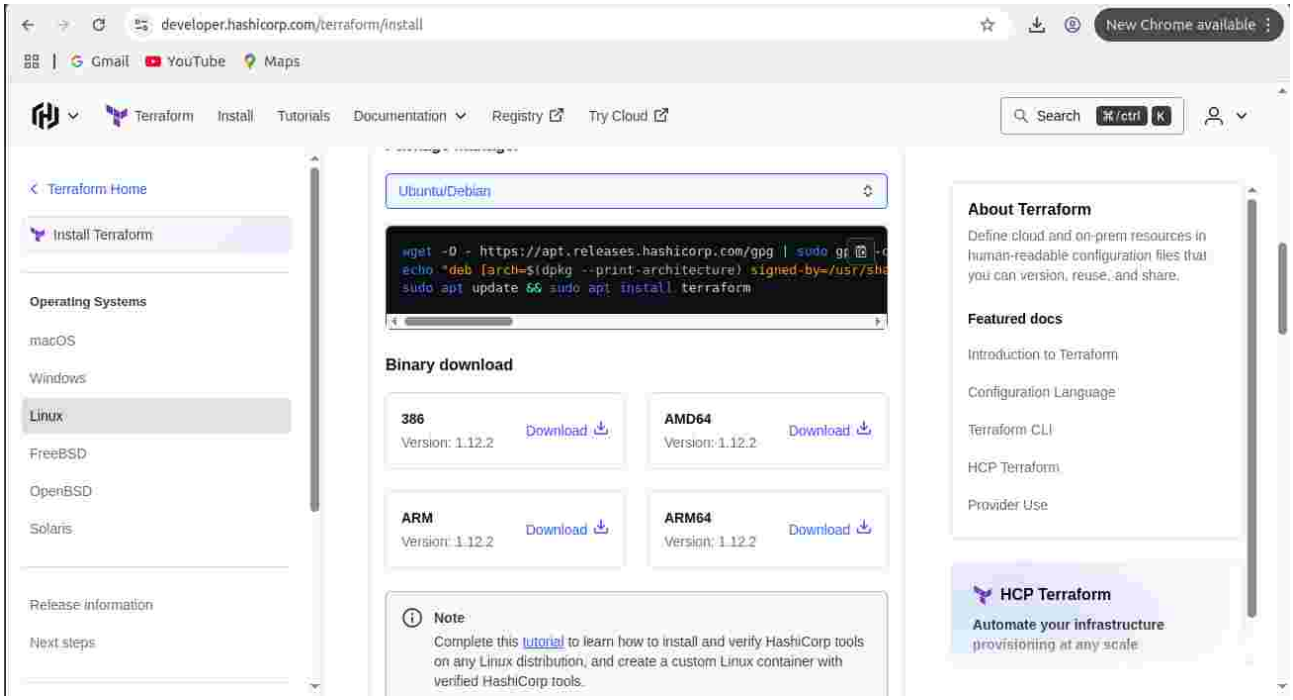
EXPERIMENT NO. 05

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

Terraform Installation Steps on Ubunu18.04

Step: 1 Terraform uses HashiCorp Configuration Language (HCL) to manage environments of Operators and Infrastructure teams. To download go to site https://www.terraform.io/downloads.html

Select the appropriate package for your operating system and architecture.



Step:2 unzip the archive by using below command

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads$ unzip terraform_1.12.2_linux_amd64.zip
Archive:  terraform_1.12.2_linux_amd64.zip
  inflating: LICENSE.txt
  inflating: terraform
```

The archive will extract a single binary called **terraform**.

Step 3: Change the directory to unzipped folder

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads$ cd terraform_1.12.2_linux_amd64
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

and Move the terraform binary to a directory included in your system's PATH in my case */usr/*local/ bin/

Step 4: To check whether Terraform is installed, run:

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo mv terraform /usr/local/bin/
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ terraform -v
Terraform v1.12.2
on linux_amd64
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

Conclusion: In This Experiment We have Learnt About Terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.

Semester: V **Name of Student: Huzaifa Bubere**
Academic Year: 2022-23 **Student ID: 24204006**
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Vishal Badgujar

EXPERIMENT NO. 06

Aim: To Build, change, and destroy AWS infrastructure Using Terraform.

Pre-requistes:

1. Install the AWS CLI version 2 on Linux

Follow these steps from the command line to install the AWS CLI on Linux.

Install curl on linux

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.24).
0 upgraded, 0 newly installed, 0 to remove and 59 not upgraded.
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

apsit@apsit:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo curl "https://awscli.amazonaws.com/awscli-exe-linux-x86
_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 59.0M  100 59.0M    0     0  6899k      0  0:00:08  0:00:08 --:--:-- 11.2M
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

apsit@apsit:~$ sudo apt install unzip

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-21ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 59 not upgraded.
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

apsit@apsit:~$ sudo unzip awscliv2.zip

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/README.md
  inflating: aws/install
   creating: aws/dist/awscli/
   creating: aws/dist/dateutil/
   creating: aws/dist/docutils/
   creating: aws/dist/lib-dynload/
   creating: aws/dist/prompt_toolkit-3.0.51.dist-info/
   creating: aws/dist/wheel-0.45.1.dist-info/
```

apsit@apsit:~$ sudo ./aws/install

apsit@apsit:~$ aws --version

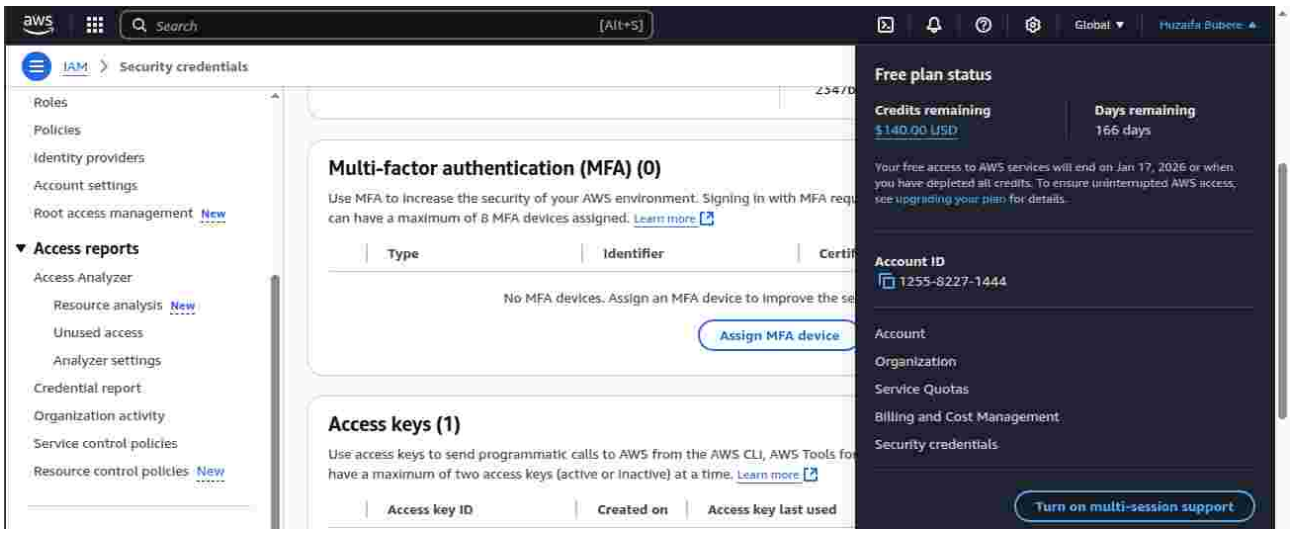
it should display the below outout.

aws-cli/2.1.29 Python/3.8.8 Linux/5.4.0-1038-aws exe/x86_64.ubuntu.18 prompt/off

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ sudo ./aws/install
Found preexisting AWS CLI installation: /usr/local/aws-cli/v2/current. Please rerun install script with --update flag.
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ aws --version
aws-cli/2.13.12 Python/3.11.4 Linux/5.4.0-150-generic exe/x86_64.ubuntu.18 prompt/off
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$
```

2. Create a new access key if you don't have one. Make sure you download the keys in your local machine.

Login to AWS console, click on username and go to My security credentials.



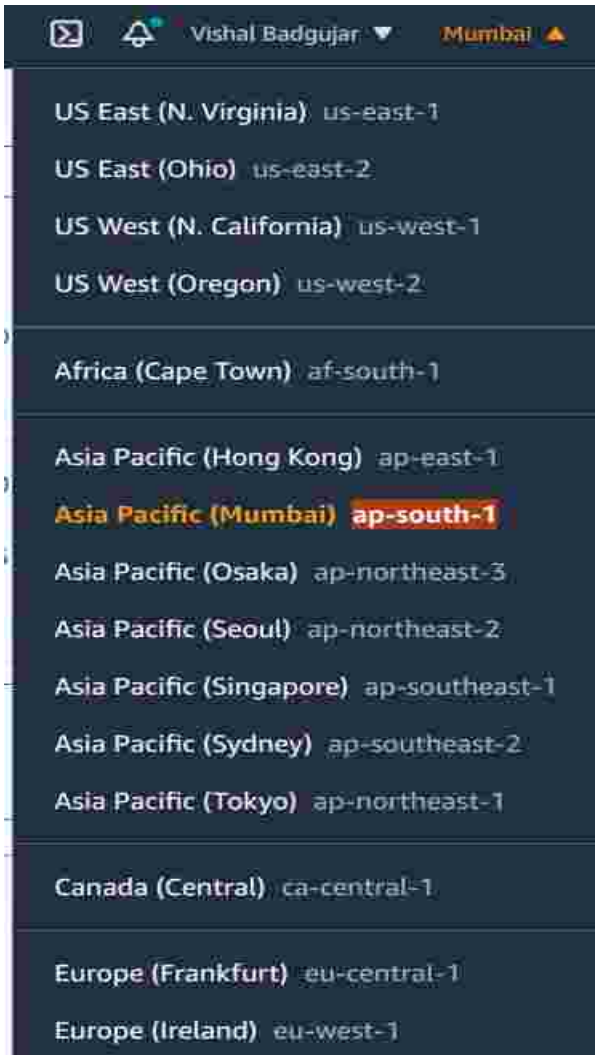
Continue on security credentials, click on access keys

Perform below commands in Linux where you have installed Terraform

First setup your access keys, secret keys and region code locally.

apsit@apsit:~$aws configure

You can check region as shown in below image :



```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ aws configure
AWS Access Key ID [****************NM34]: AKIAR2PKJMPKNQIWRU7Z
AWS Secret Access Key [****************fleU]: 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
Default region name [us-east-1]: ap-south-1
Default output format [None]:
```

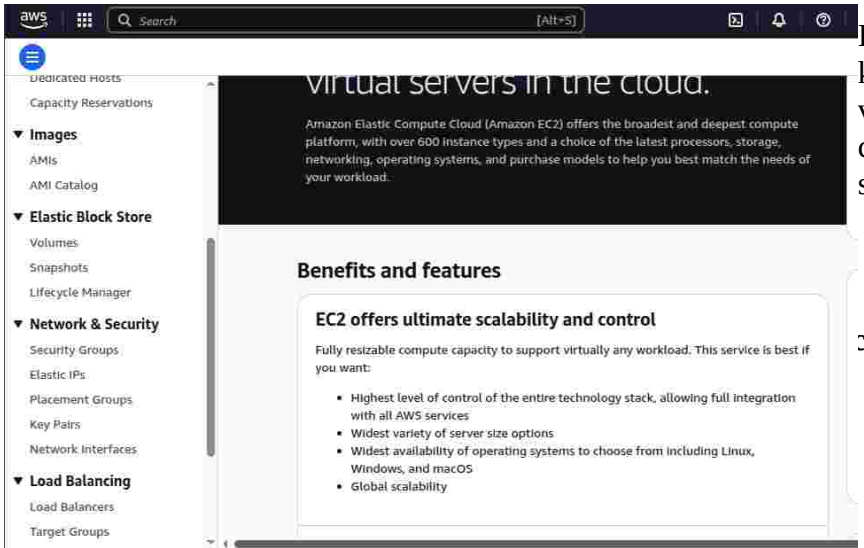
Create one Directory for Terraform project in which all files of terraform we can save

apsit@apsit:~$ cd ~
apsit@apsit:~$ mkdir project-terraform
apsit@apsit:~$ cd project-terraform

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/Downloads/terraform_1.12.2_linux_amd64$ cd
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ mkdir project-terraform
mkdir: cannot create directory 'project-terraform': File exists
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ mkdir projects-terraform
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ cd projects-terraform
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$
```

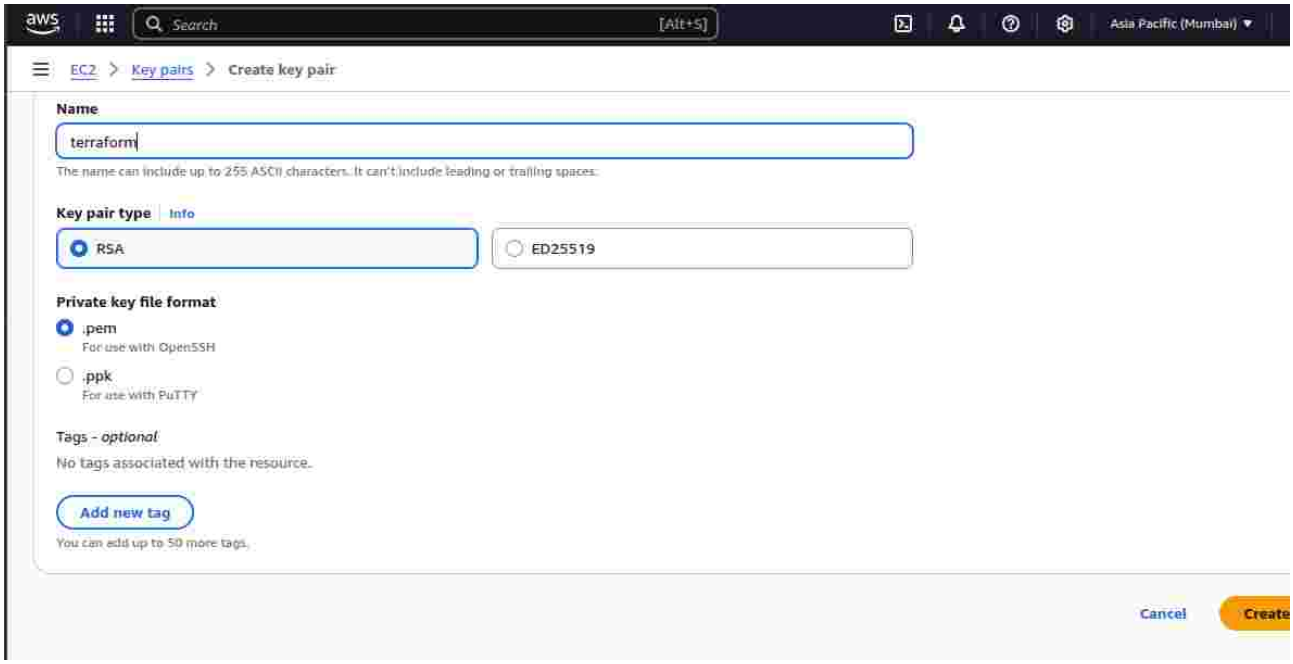
Create Terraform Files

vishal@apsit:~$ sudo nano variables.tf

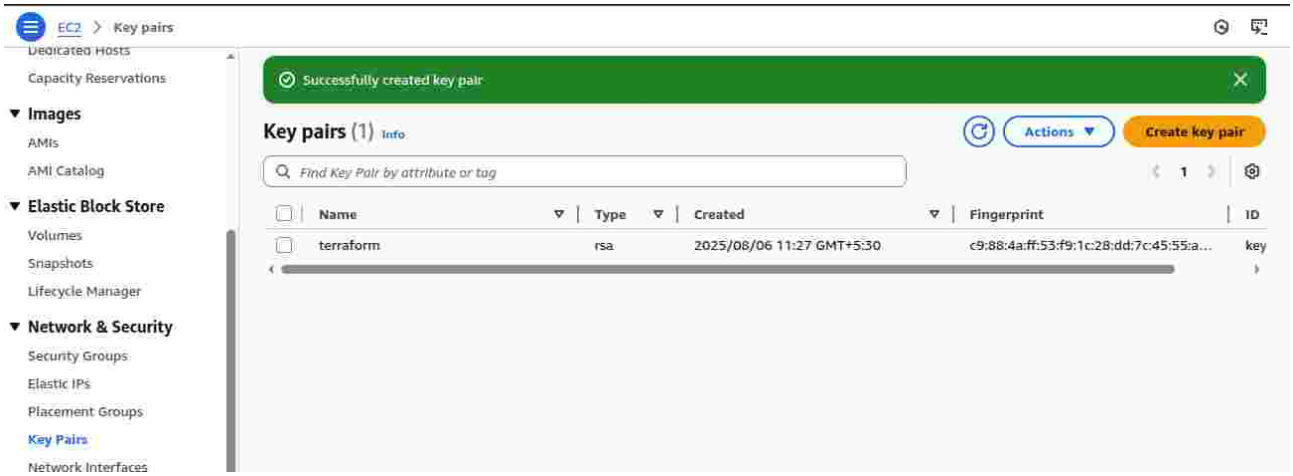


In order to provide key name in variables first create key pair as shown:

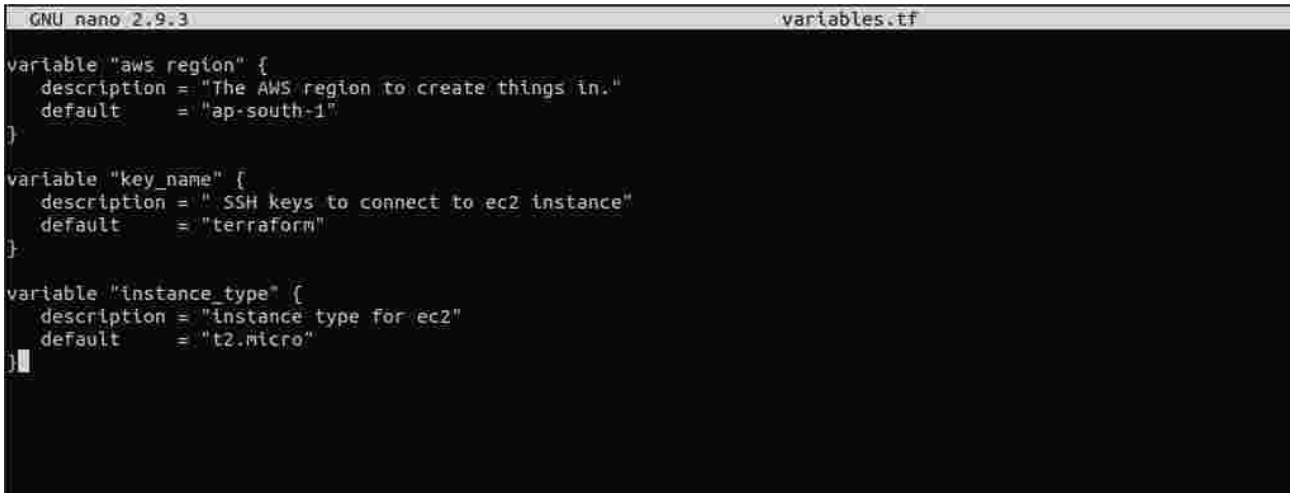
Give name to key pair file as **terraform**



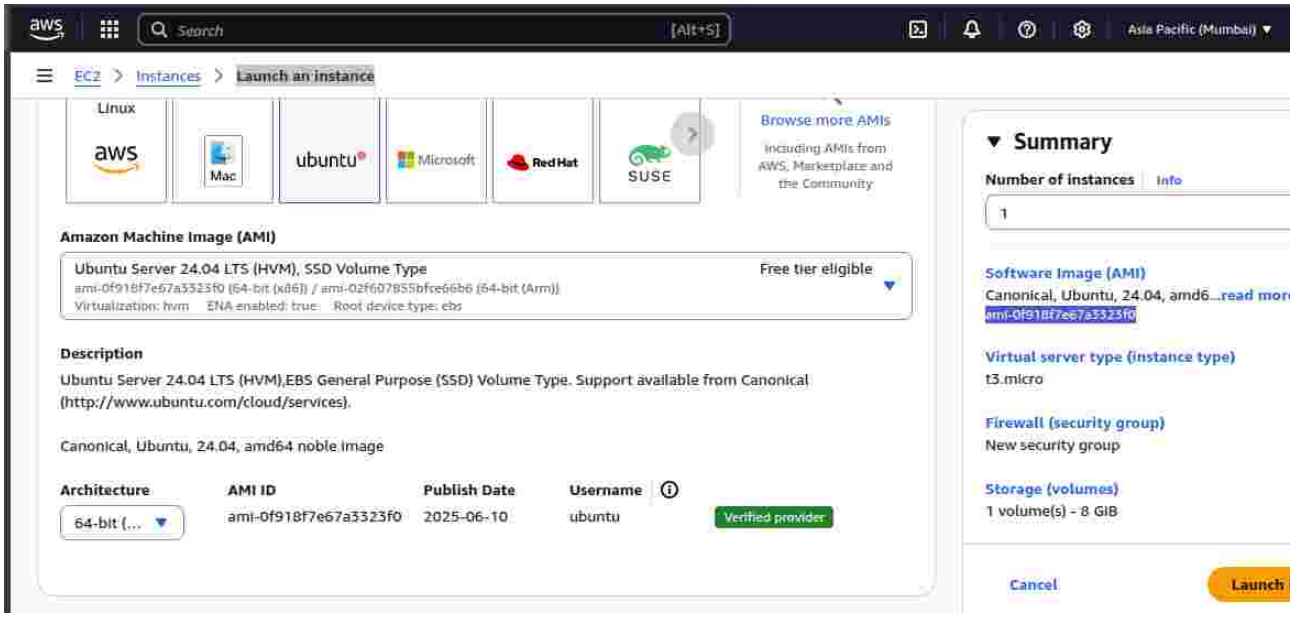
Key pair is generated



Use your Region and Key name in variable.tf as shown and provide instance type which you want to create.



After creating variable terraform file note down the AMI ID of instance which u want to create which we will use to configure our instance in main.tf file.



Now create main.tf file:

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$ sudo nano main.tf
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$
```

```
provider "aws" {
  region = var.aws_region
}

#Create security group with firewall rules
resource "aws_security_group" "security_jenkins_port" {
  name        = "security_jenkins_port"
  description = "security group for jenkins"

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

 ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
# outbound from jenkis server
  egress {
    from_port   = 0
    to_port     = 65535
```

```
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }

 tags= {
  Name = "security_jenkins_port"
 }
}

resource "aws_instance" "myFirstInstance" {
 ami           = "ami-0f918f7e67a3323f0"
 key_name = var.key_name
 instance_type = var.instance_type
 security_groups= [ "security_jenkins_port"]
 tags= {
  Name = "jenkins_instance"
 }
}
```

Put AMI-ID in above highlighted space and Now execute the below command:

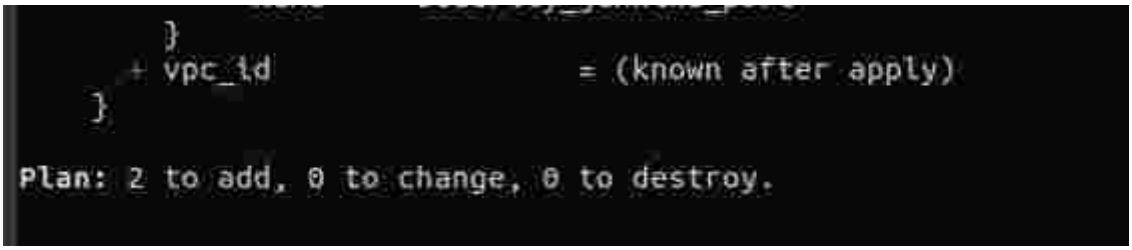
you should see like below screenshot.

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.7.0...
- Installed hashicorp/aws v6.7.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$
```

Execute the below command

the above command will show how many resources will be added.
Plan: 3 to add, 0 to change, 0 to destroy.

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$ 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:

  # aws_instance.myFirstInstance will be created
  + resource "aws_instance" "myFirstInstance" {
      + ami                                  = "ami-0f918f7e67a3323f0"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "terraform"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
```



Execute the below command

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~/projects-terraform$ 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:

  # aws_instance.myFirstInstance will be created
  + resource "aws_instance" "myFirstInstance" {
      + ami                                  = "ami-0f918f7e67a3323f0"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "terraform"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

Provide the value as Yes for applying terraform

```
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
```

Plan: 3 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

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Now login to EC2 console, to see the new instances up and running, you can see Jenkins_instance is up and running which we deploy from terraform.

You can also check the security group resource details which you created from terraform :

Terraform destroy

you can also destroy or delete your instance by using terraform destroy command :

```
vishal@apsit:~/project-terraform$ terraform destroy
```

Now you can see instance which you created by using terraform is deleted successfully from aws console also you can check it will removed successfully:

All the Resources including Security groups, EC2 instances using terraform will be deleted. In this way we can automate infrastructure set up using terrform in aws cloud.

Conclusion: In this Experiment We have Learnt About Build, change, and destroy AWS infrastructure Using Terraform.

Semester: V
Academic Year: 2025-26
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor:Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID:24204006

EXPERIMENT NO. 07

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

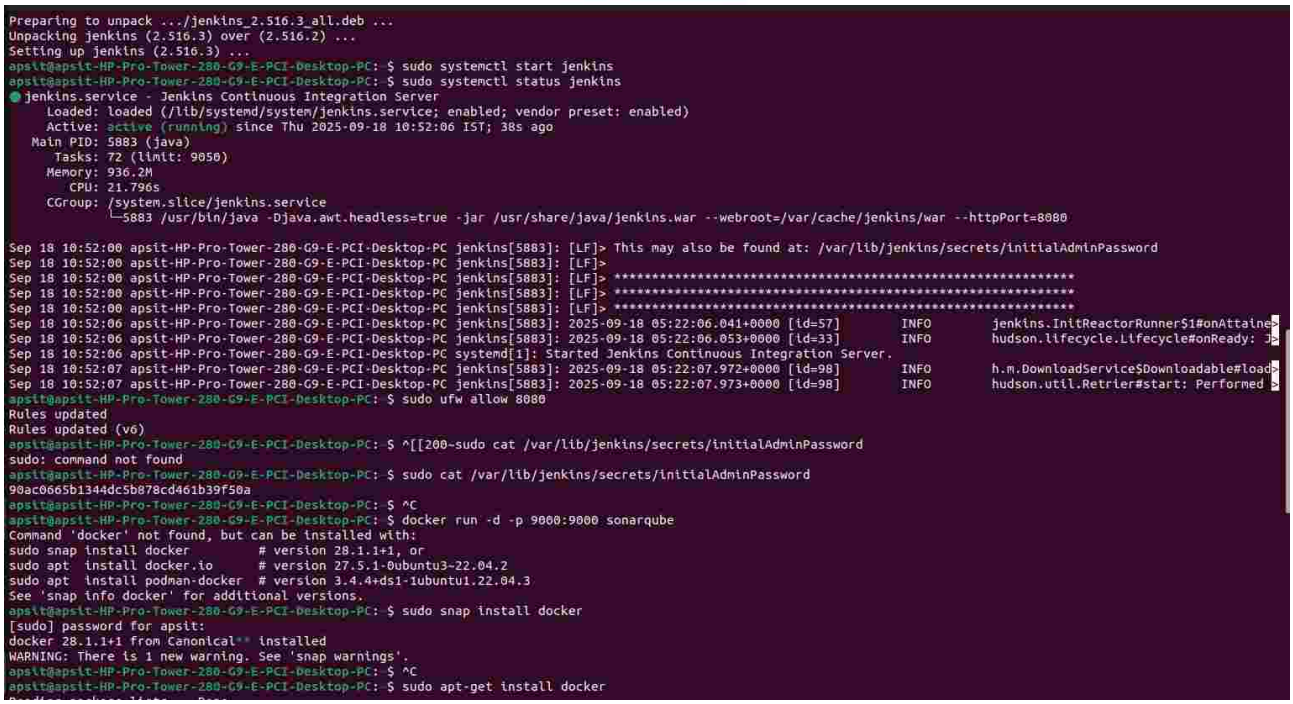
Steps:

1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.

2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.

.

```
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
[sudo] password for apsit:
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo apt update-
E: Invalid operation update-
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo apt update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:4 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:5 https://pkg.jenkins.io/debian-stable binary/ Packages [29.7 kB]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,214 B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:8 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [54.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```



1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.

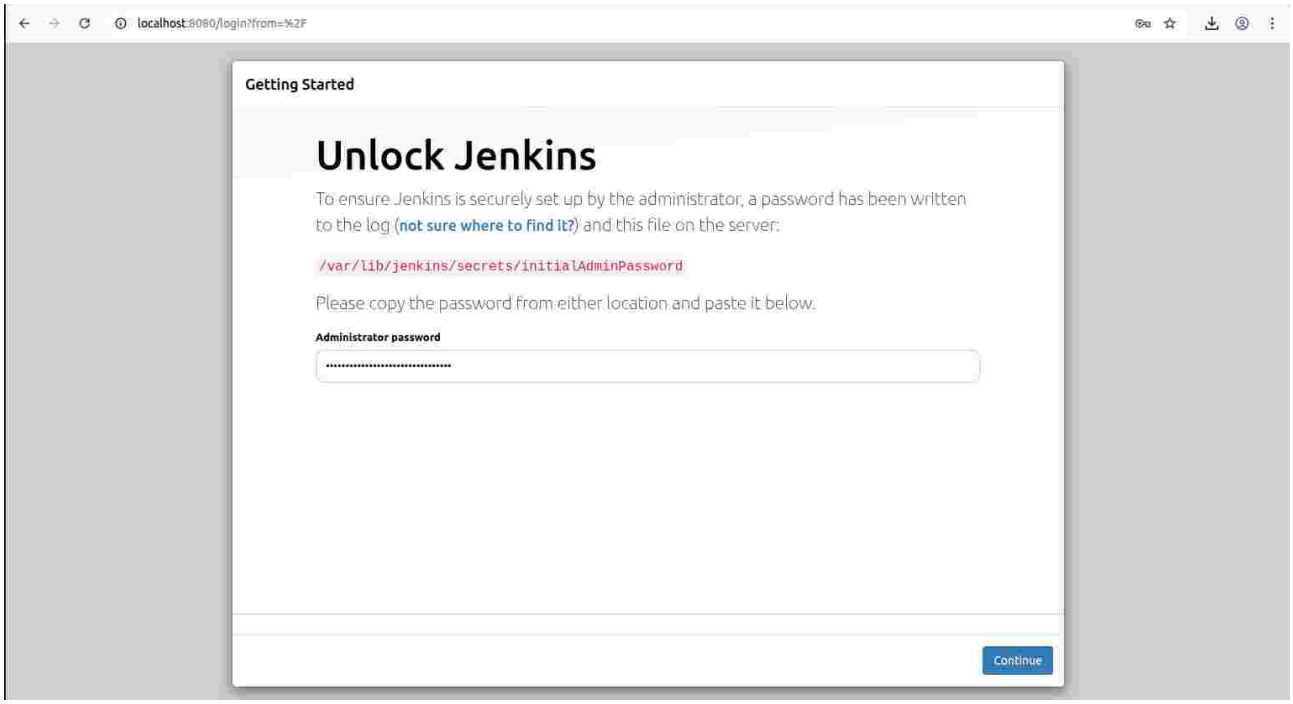
Installation of Jenkins

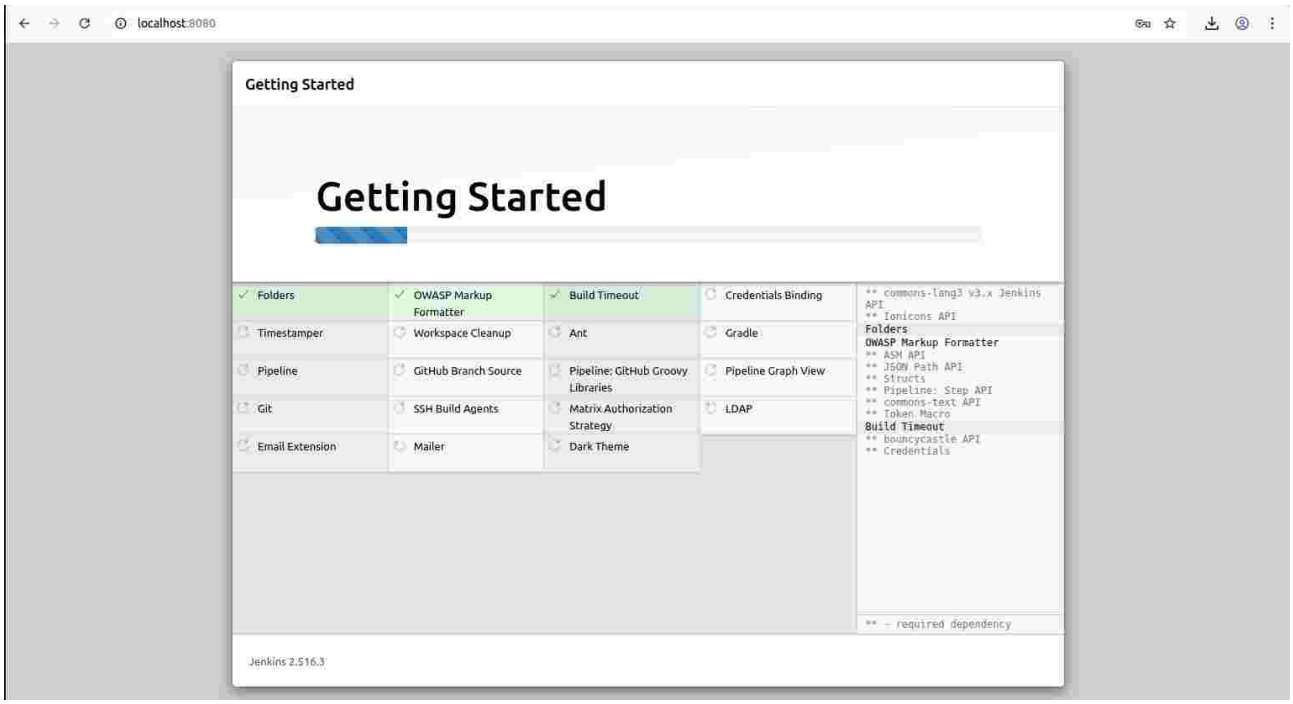
The version of Jenkins included with the default Ubuntu packages is often behind the latest available version from the project itself. To take advantage of the latest fixes and features, you can use the project-maintained packages to install Jenkins.

manjusha@apsit:~$ wget -q -O -
https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

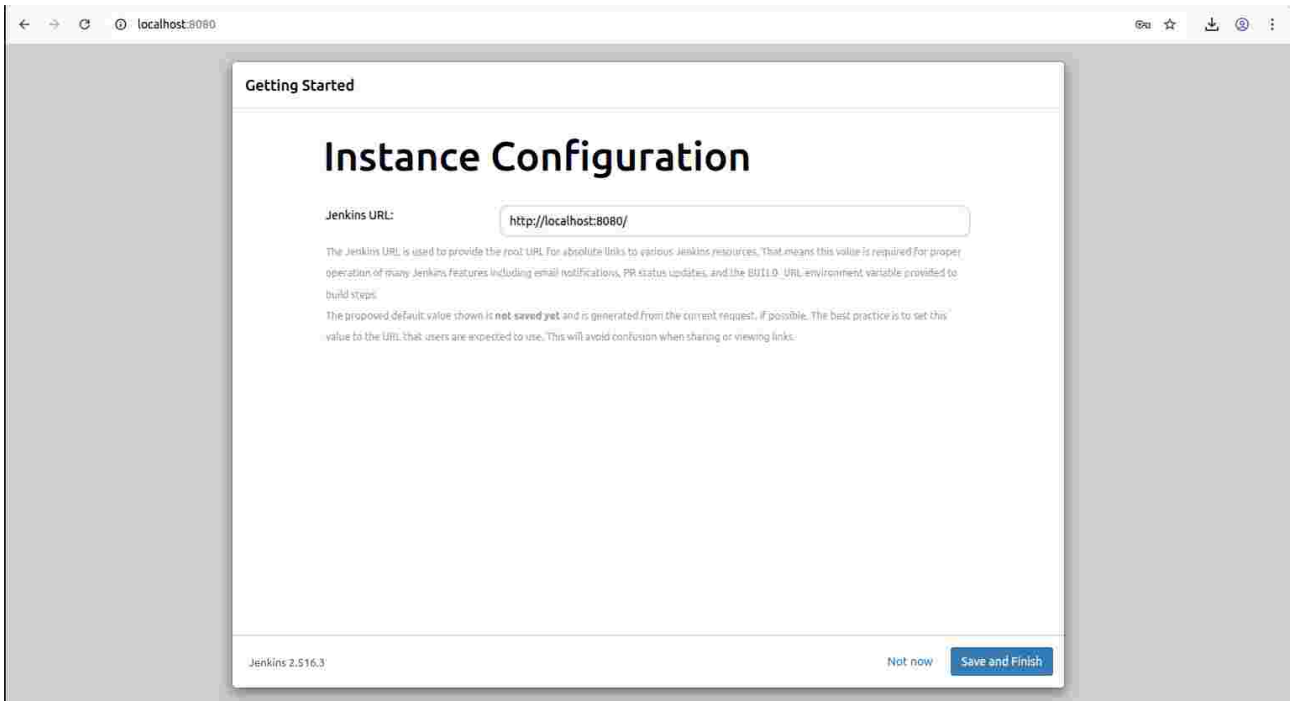
When the key is added, the system will return OK. Next, append the Debian package repository address to the server's sources.list:

manjusha@apsit:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable

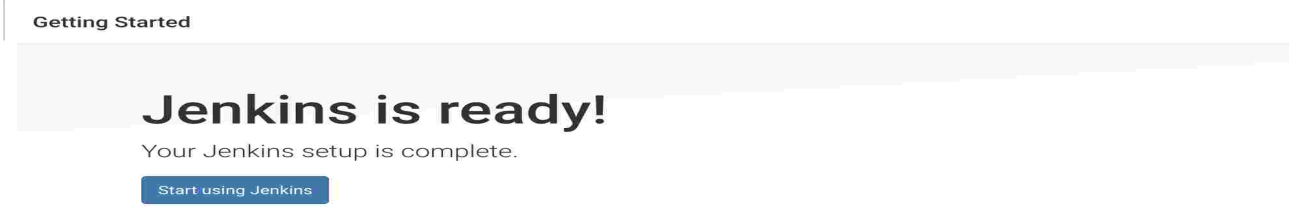

localhost:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue


localhost:8080
Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
LDAP
Email Extension
Mailer
Dark Theme
Jenkins 2.516.3

When the installation is complete, you will be prompted to set up the first administrative user. It's possible to skip this step and continue as admin using the initial password we used above, but we'll take a moment to create the user.



After confirming the appropriate information, click Save and Finish. You will see a confirmation page confirming that "Jenkins is Ready!":
Click Start using Jenkins to visit the main Jenkins dashboard:

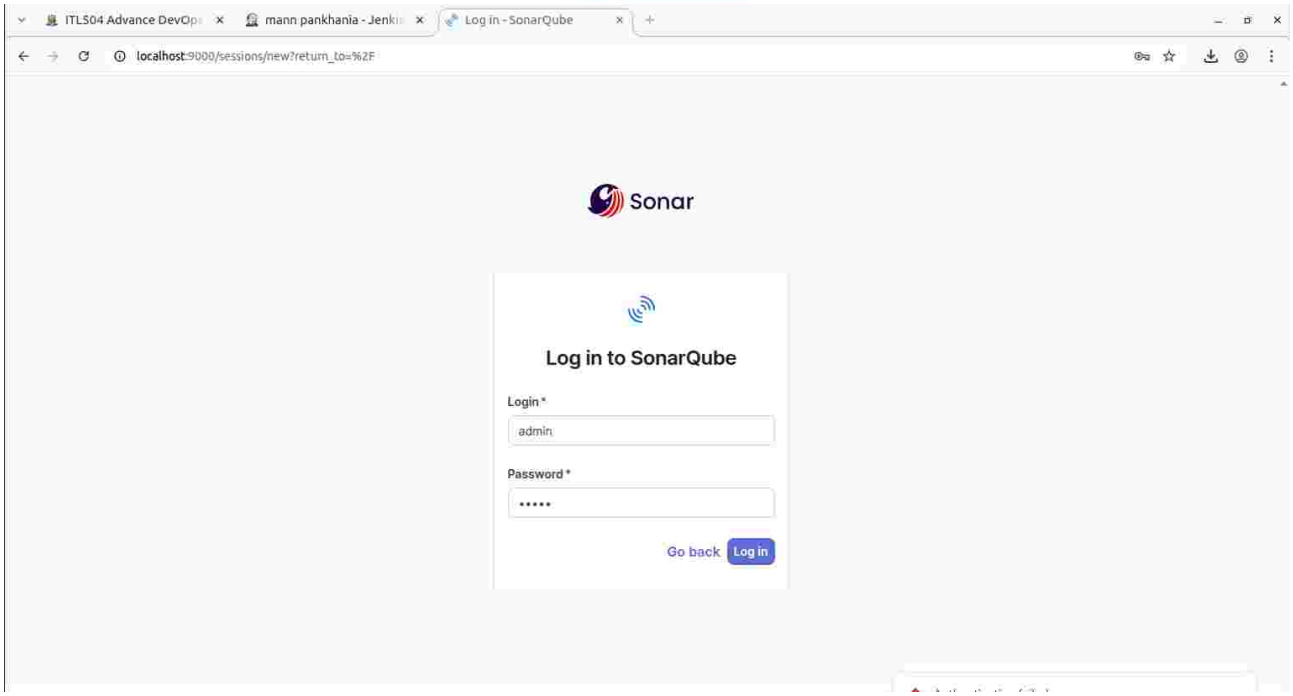


SonarQube Setup

Before proceeding with the integration, we will setup SonarQube Instance. we are using SonarQube Docker Container.

manjusha@apsit:~$docker run -d -p 9000:9000 sonarqube

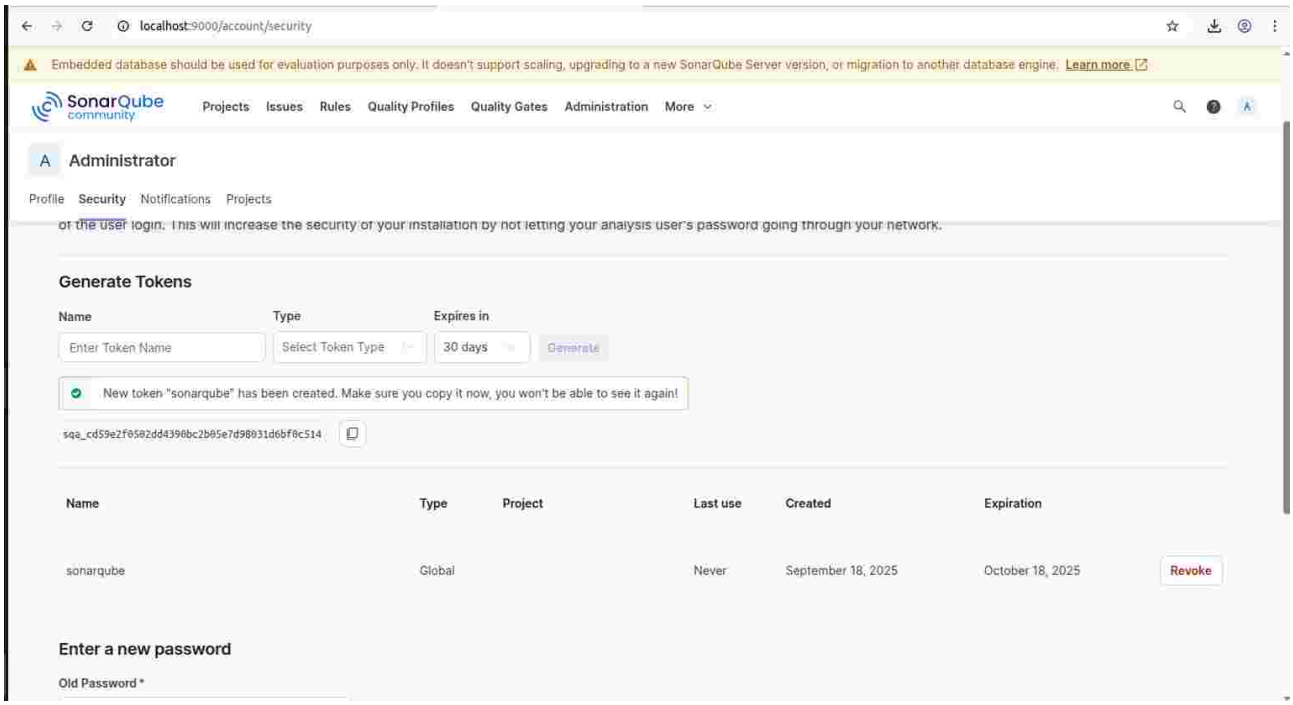
In the above command, we are forwarding port 9000 of the container to the port 9000 of the host machine as SonarQube is will run on port 9000. Then, from the browser, enter http://localhost:9000. After That, you will see the SonarQube is running. Then, login using default credentials (admin:admin).

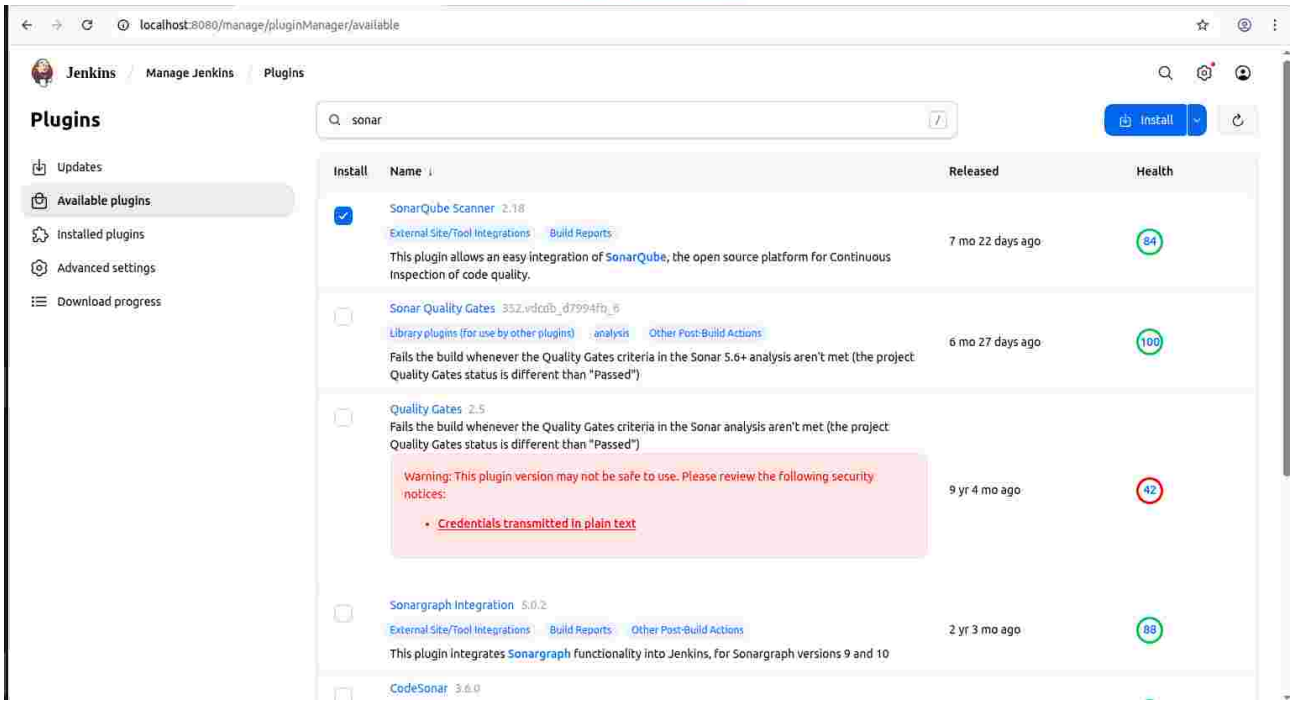


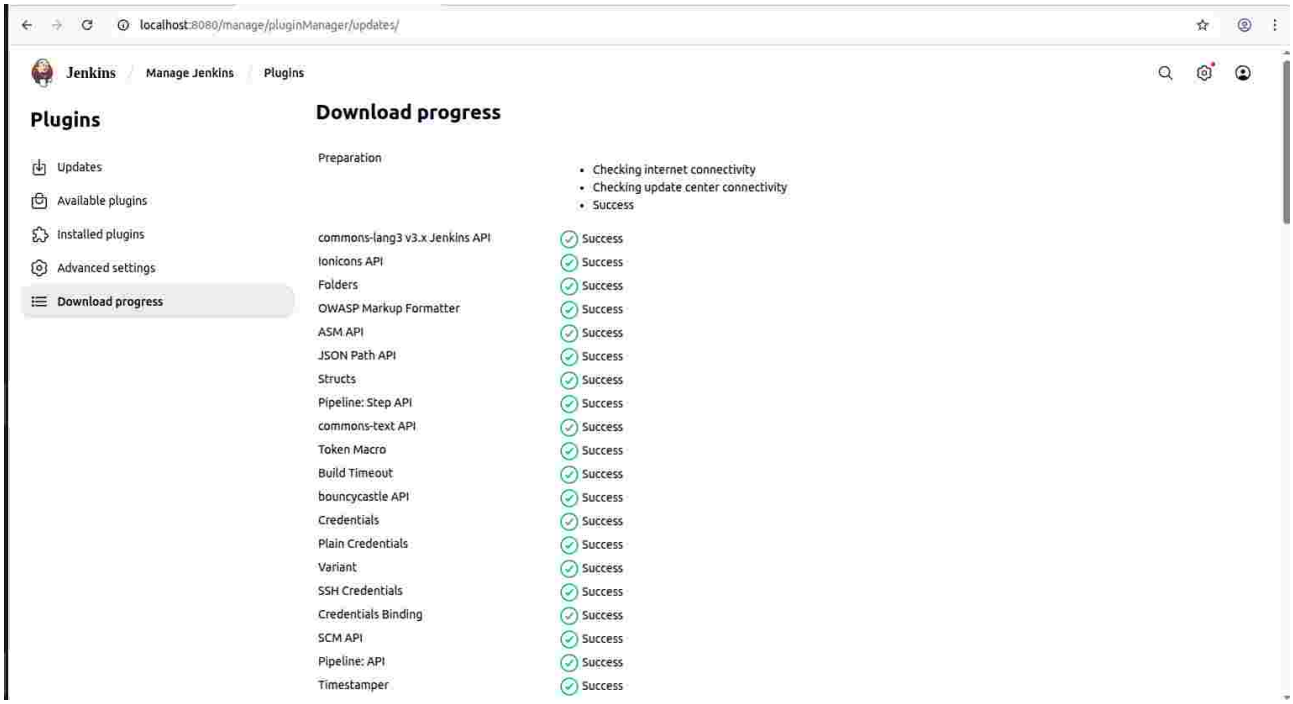
Generate User Token

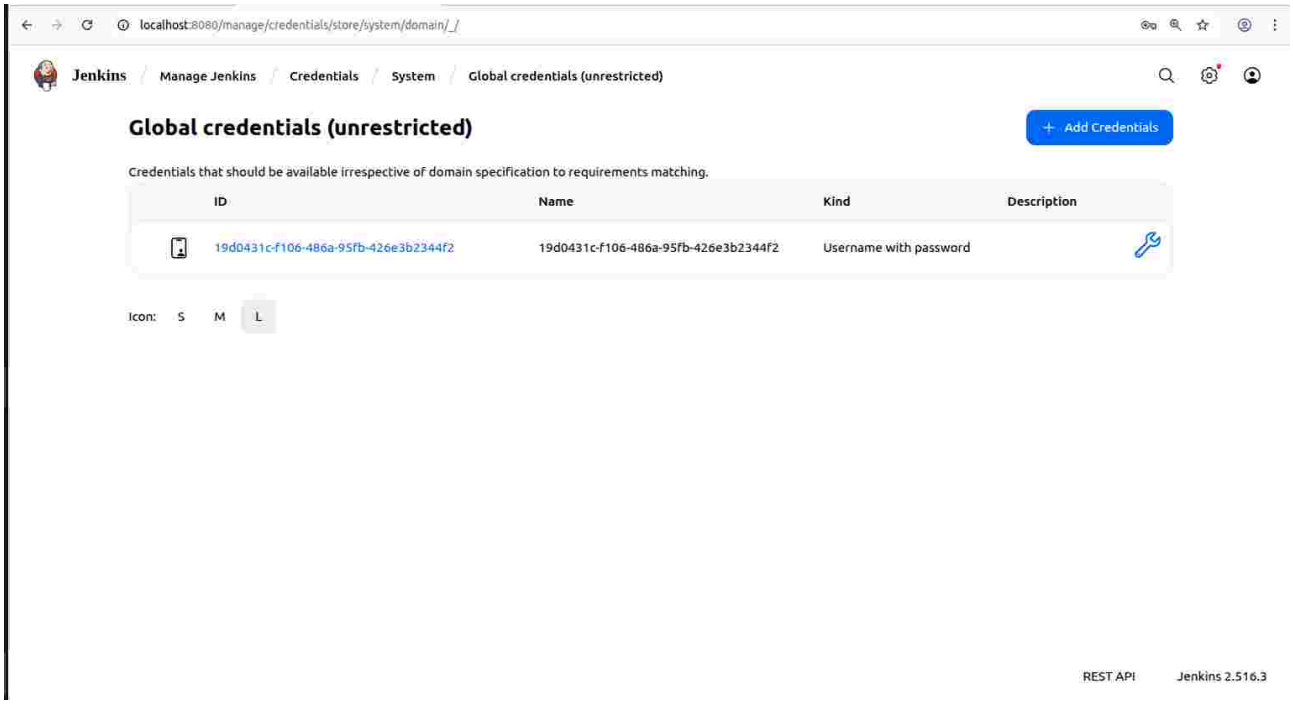
Now, we need to get the SonarQube user token to make connection between Jenkins and SonarQube. For the same, go to **Administration> User > My Account > Security** and then, from the bottom of the page you can create new tokens by clicking the Generate Button. Copy the Token and keep it safe.

C96798e9bd081e117189b516c868ddb7d87ee785 SonarQube


localhost:9000/account/security
Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more
SonarQube community
Projects Issues Rules Quality Profiles Quality Gates Administration More
A Administrator
Profile Security Notifications Projects
of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.
Generate Tokens
Name
Type
Expires in
Enter Token Name
Select Token Type
30 days
Generate
New token "sonarqube" has been created. Make sure you copy it now, you won't be able to see it again!
sqa_cd59e2f0502dd4396bc2b05e7d98031d6bf0c514
Name Type Project Last use Created Expiration
sonarqube Global Never September 18, 2025 October 18, 2025 Revoke
Enter a new password
Old Password *


localhost:8080/manage/pluginManager/available
Jenkins
Manage Jenkins
Plugins
Plugins
sonar
Install
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Install
Name
Released
Health
SonarQube Scanner 2.18
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
7 mo 22 days ago
84
Sonar Quality Gates 352.vdcd0_d7994fb_6
Library plugins (for use by other plugins)
analysis
Other Post-Build Actions
Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
6 mo 27 days ago
100
Quality Gates 2.5
Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")
Warning: This plugin version may not be safe to use. Please review the following security notices:
Credentials transmitted in plain text
9 yr 4 mo ago
42
Sonargraph Integration 5.0.2
External Site/Tool Integrations
Build Reports
Other Post-Build Actions
This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10
2 yr 3 mo ago
88
CodeSonar 3.6.0


localhost:8080/manage/pluginManager/updates/
Jenkins
Manage Jenkins
Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
commons-lang3 v3.x Jenkins API Success
Ionicons API Success
Folders Success
OWASP Markup Formatter Success
ASM API Success
JSON Path API Success
Structs Success
Pipeline: Step API Success
commons-text API Success
Token Macro Success
Build Timeout Success
bouncycastle API Success
Credentials Success
Plain Credentials Success
Variant Success
SSH Credentials Success
Credentials Binding Success
SCM API Success
Pipeline: API Success
Timestamper Success



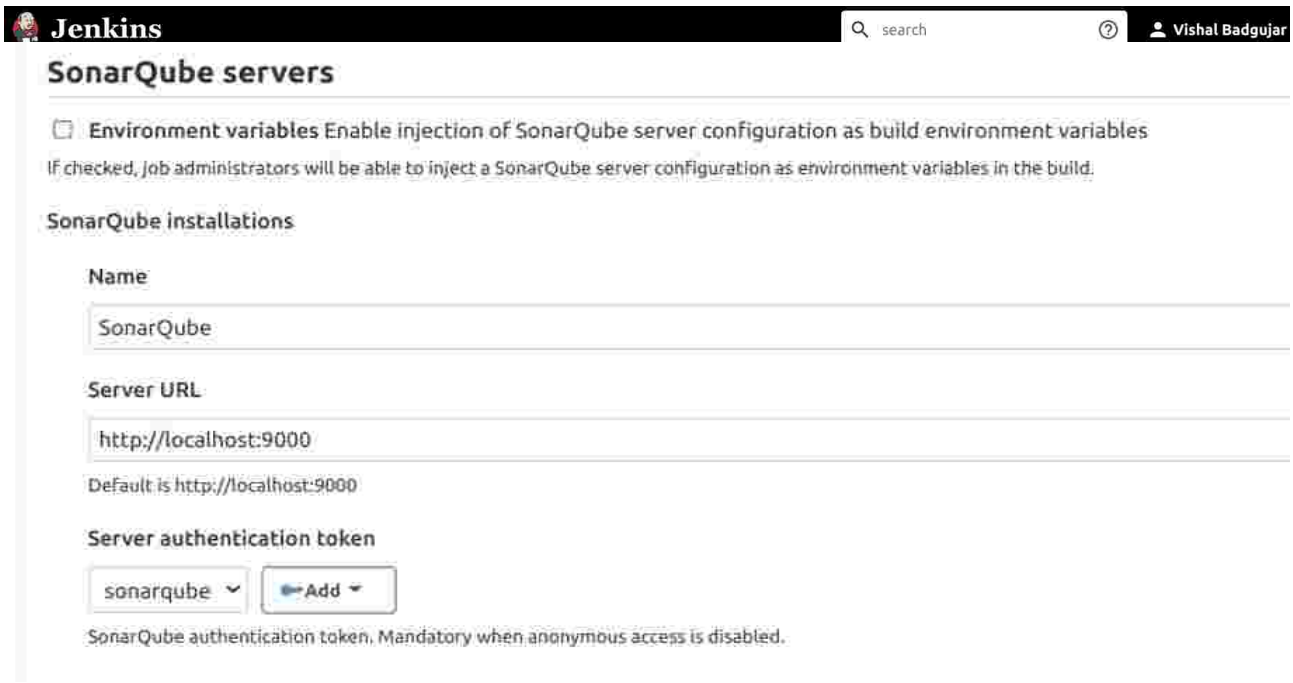
2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.

Jenkins Setup for SonarQube

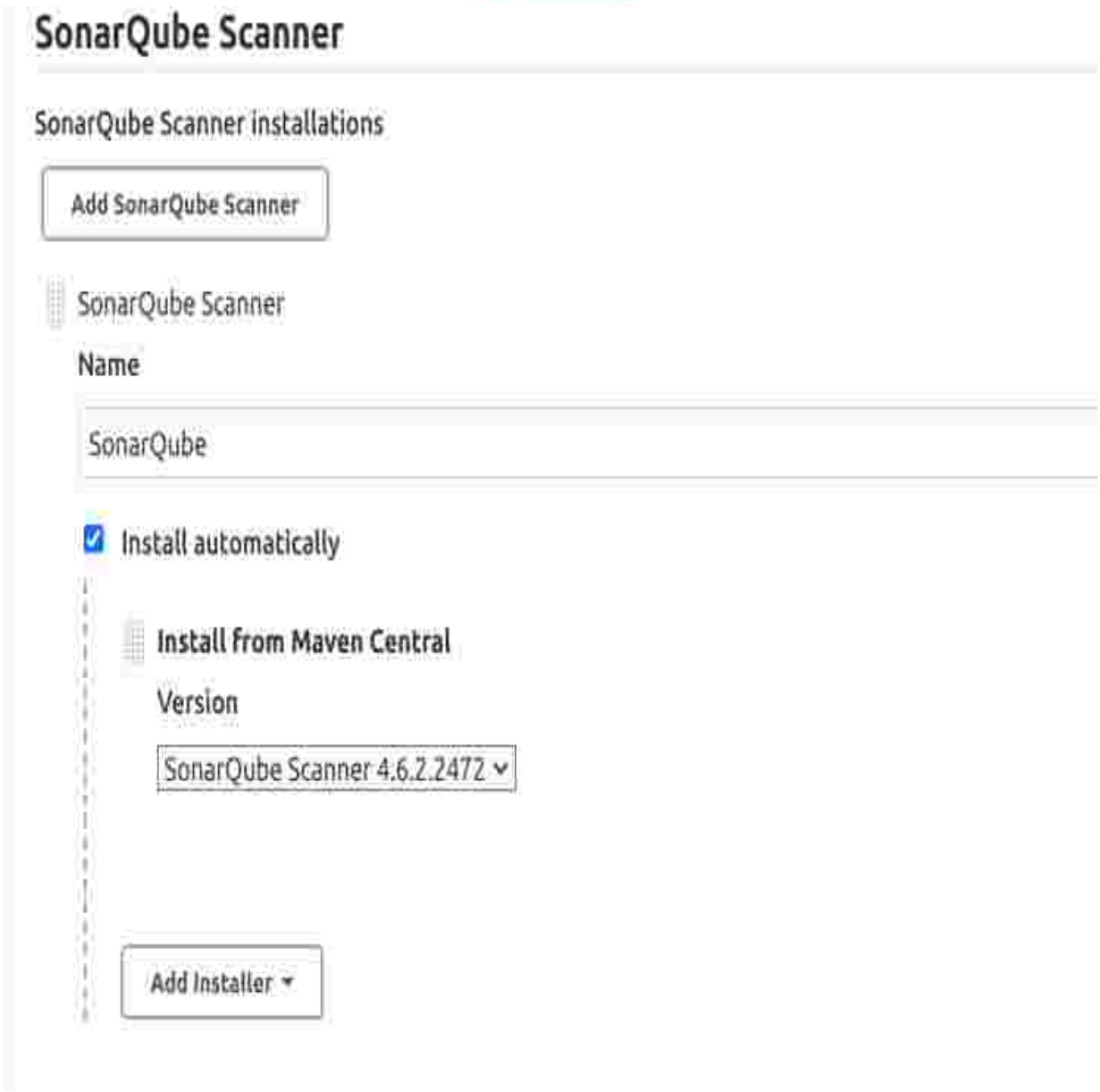
Before all, we need to install the SonarQube Scanner plugin in Jenkins. For the same, go to **Manage Jenkins > Plugin Manager > Available.** From here, type SonarQube Scanner then select and install.

Tool Configuration SonarQube Scanner

Now, we need to configure the Jenkins plugin for SonarQube Scanner to make a connection with the SonarQube Instance. For that, got to **Manage Jenkins > Configure System > SonarQube Server.** Then, Add SonarQube. In this, give the Installation Name, Server URL then Add the Authentication token in the Jenkins Credential Manager and select the same in the configuration.

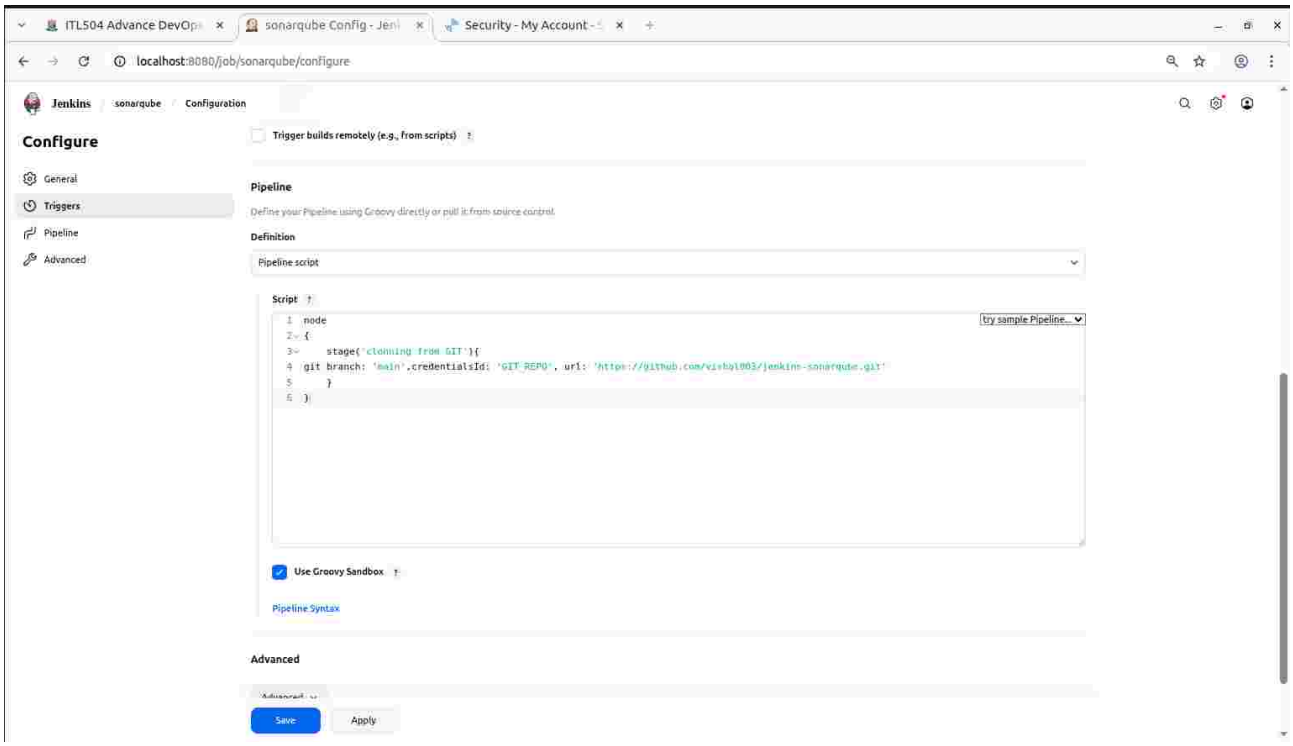


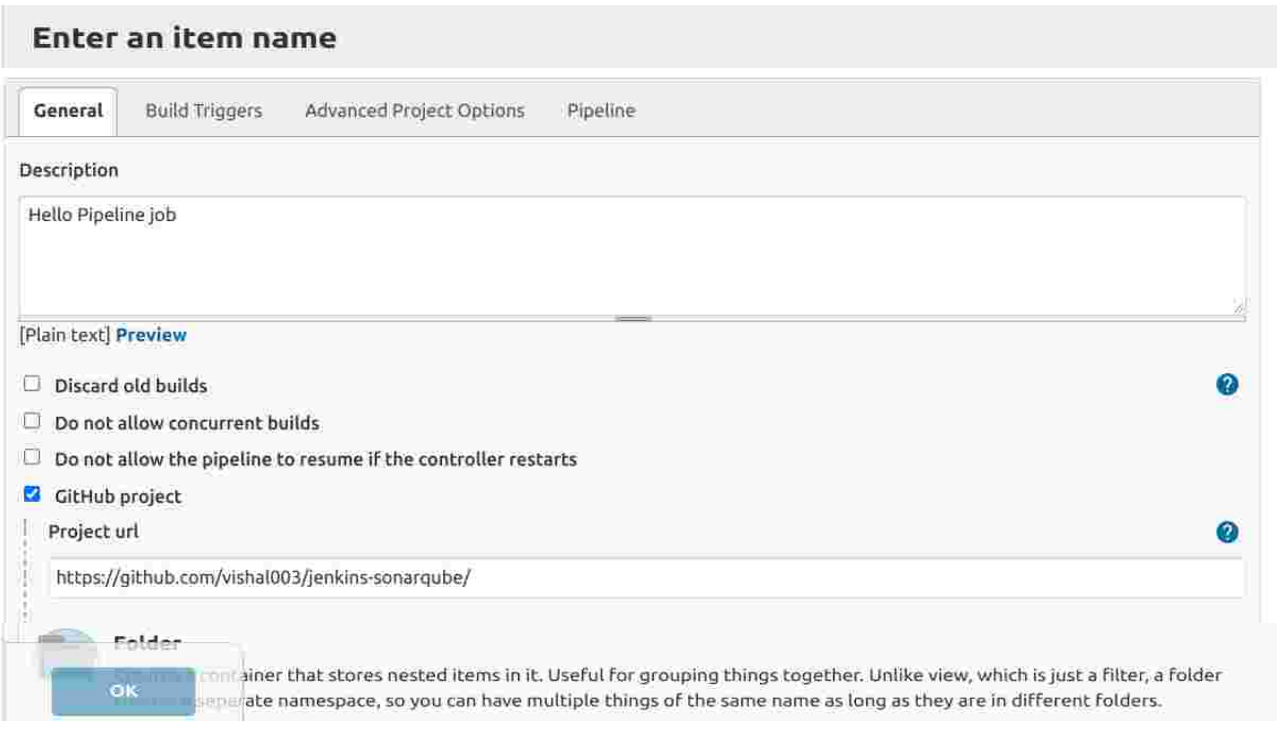
Then, we need to set-up the SonarQube Scanner to scan the source code in the various stage. For the same, go to **Manage Jenkins > Global Tool Configuration > SonarQube Scanner**. Then, Click **Add SonarQube Scanner Button**. From there, give some name of the scanner type and **Add Installer** of your choice. In this case, I have selected SonarQube Scanner from Maven Central.



SonarQube Scanner in Jenkins Pipeline

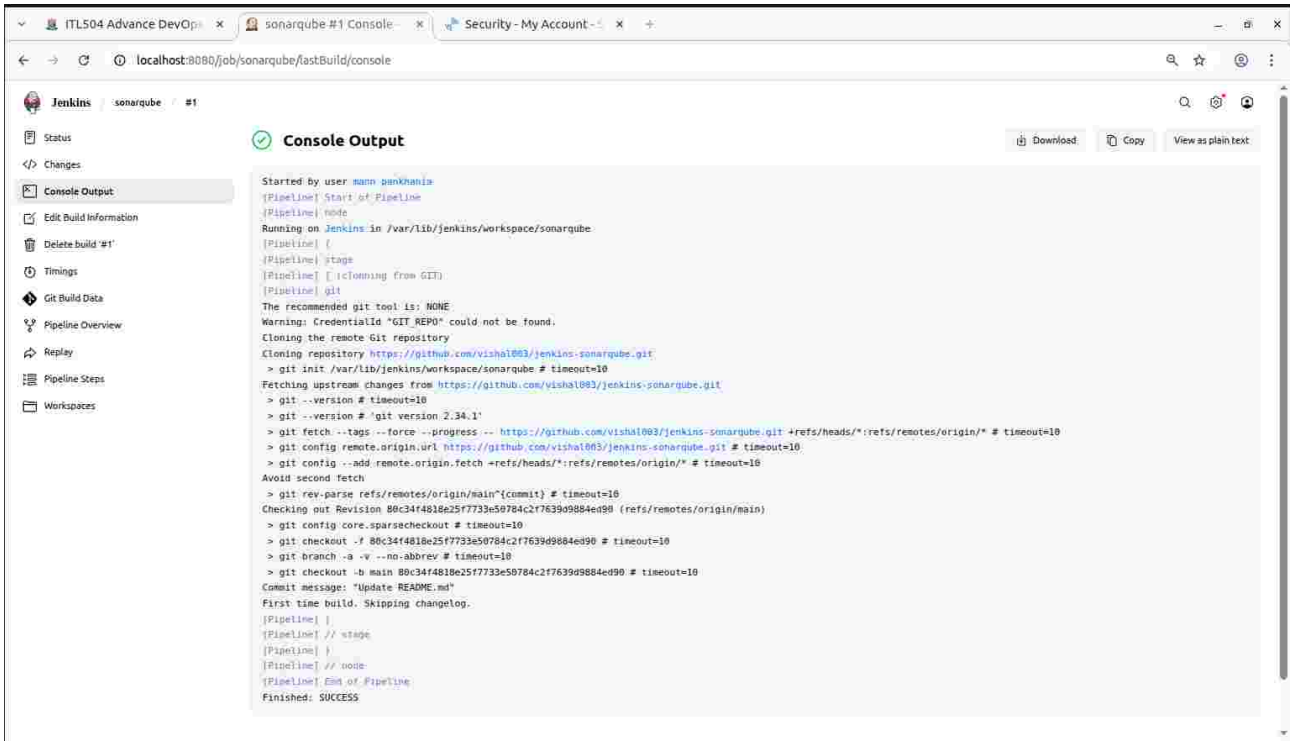
Now, It's time to integrate the SonarQube Scanner in the Jenkins Pipeline. For the same, we are going to add one more stage in the Jenkinsfile called SonarQube and inside that, I am adding the following settings and code.


ITL504 Advance DevOp
sonarqube Config - Jen
Security - My Account -
localhost:8080/job/sonarqube/configure
Jenkins
sonarqube
Configuration
Configure
General
Triggers
Pipeline
Advanced
Trigger builds remotely (e.g., from scripts)
Pipeline
Define your Pipeline using Groovy directly or pull it from source control
Definition
Pipeline script
Script
try sample Pipeline...
1 node
2 {
3 stage('clonning from GIT'){
4 git branch: 'main',credentialsId: 'GIT_REPO', url: 'https://github.com/vishal003/jenkins-sonarqube.git'
5 }
6 }
Use Groovy Sandbox
Pipeline Syntax
Advanced
Save
Apply


Enter an item name
General
Build Triggers
Advanced Project Options
Pipeline
Description
Hello Pipeline job
[Plain text] Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url
https://github.com/vishal003/jenkins-sonarqube/
Folder
OK
ainer that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder
ate namespace, so you can have multiple things of the same name as long as they are in different folders.

Github Configuration in Jenkins Pipeline

Git Clonning into Jenkins



Successfully Build Github Repository in Jenkins

Conclusion: In This Experiment We Have Learnt About Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Semester: V
Academic Year: 2025-26
Class / Branch: TE IT-C
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID:24204006

EXPERIMENT NO. 08

Aim: To 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.

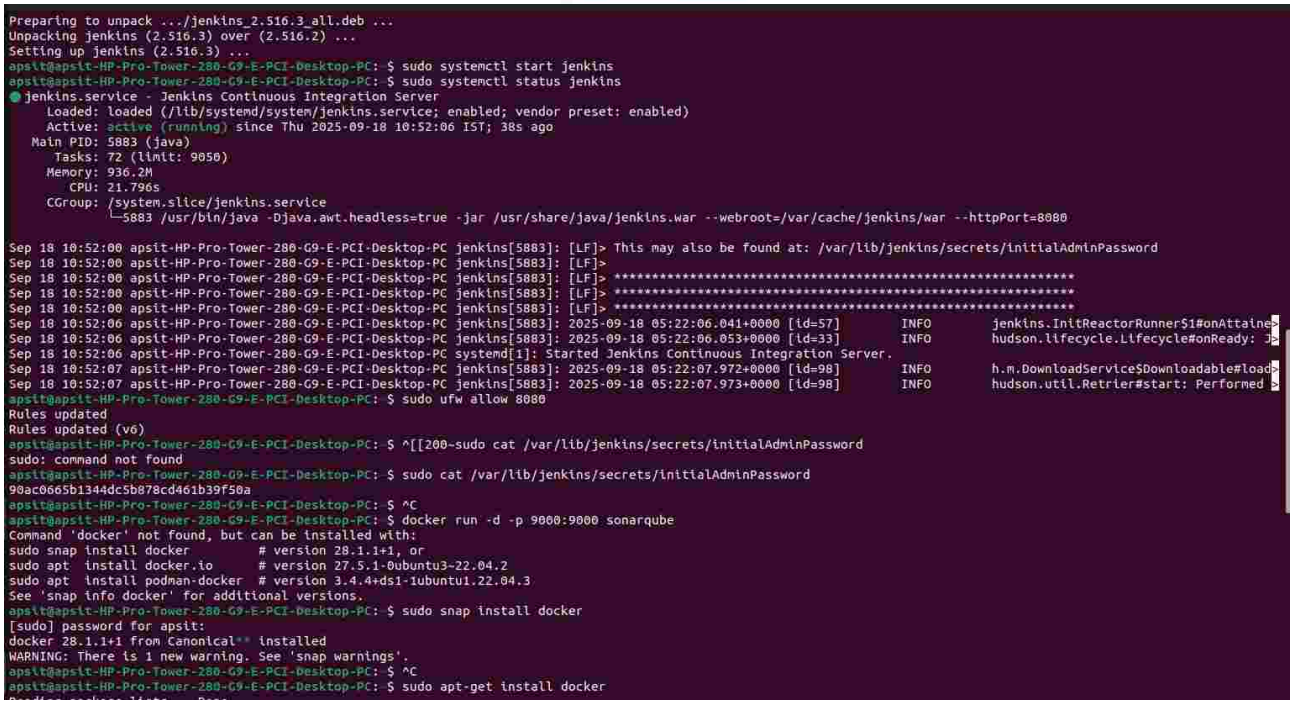
Steps:

1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.

2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.

.

```
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
[sudo] password for apsit:
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo apt update~
E: Invalid operation update~
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC: $ sudo apt update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:4 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:5 https://pkg.jenkins.io/debian-stable binary/ Packages [29.7 kB]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,214 B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:8 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [54.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```



1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.

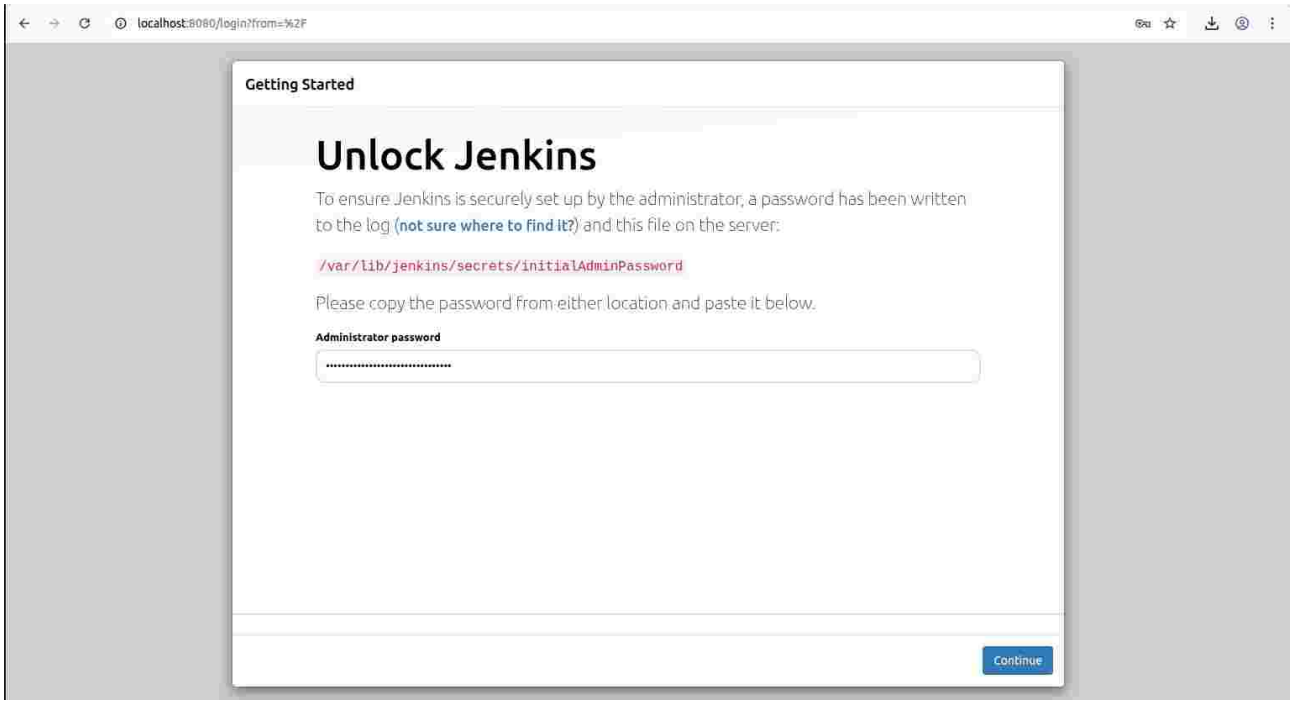
Installation of Jenkins

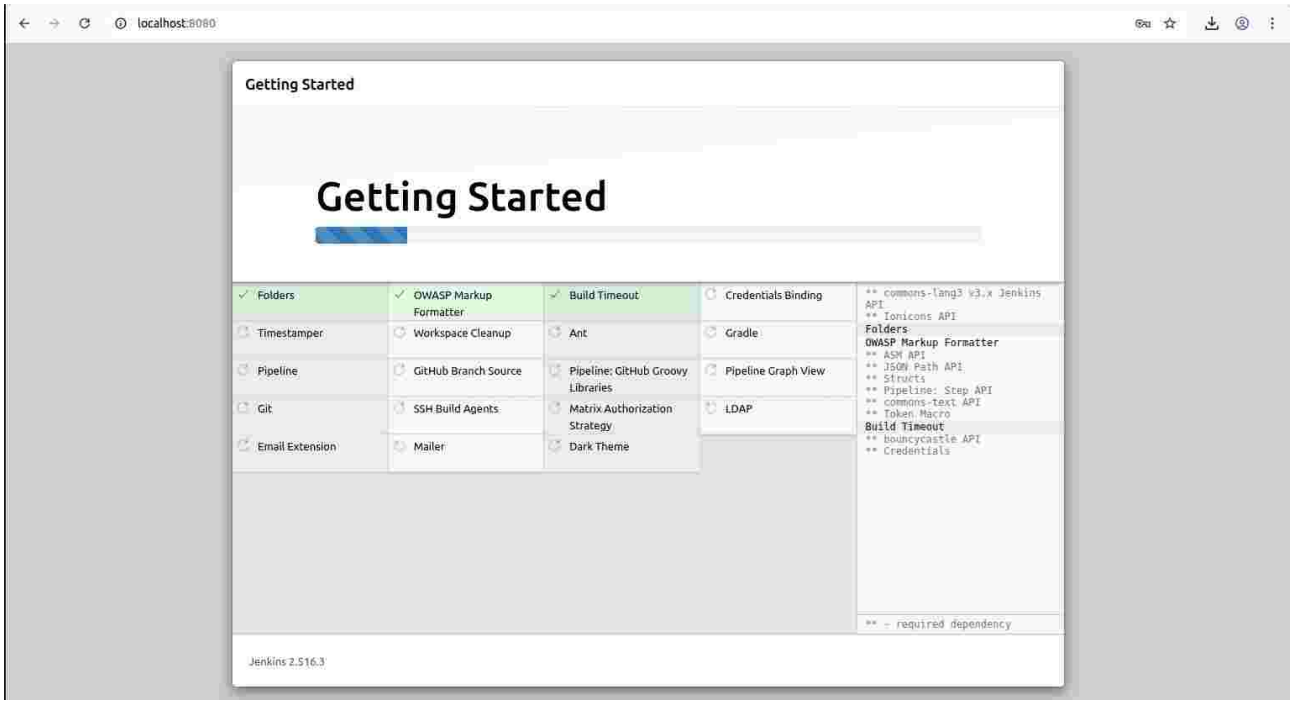
The version of Jenkins included with the default Ubuntu packages is often behind the latest available version from the project itself. To take advantage of the latest fixes and features, you can use the project-maintained packages to install Jenkins.

manjusha@apsit:~$ wget -q -O -
https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

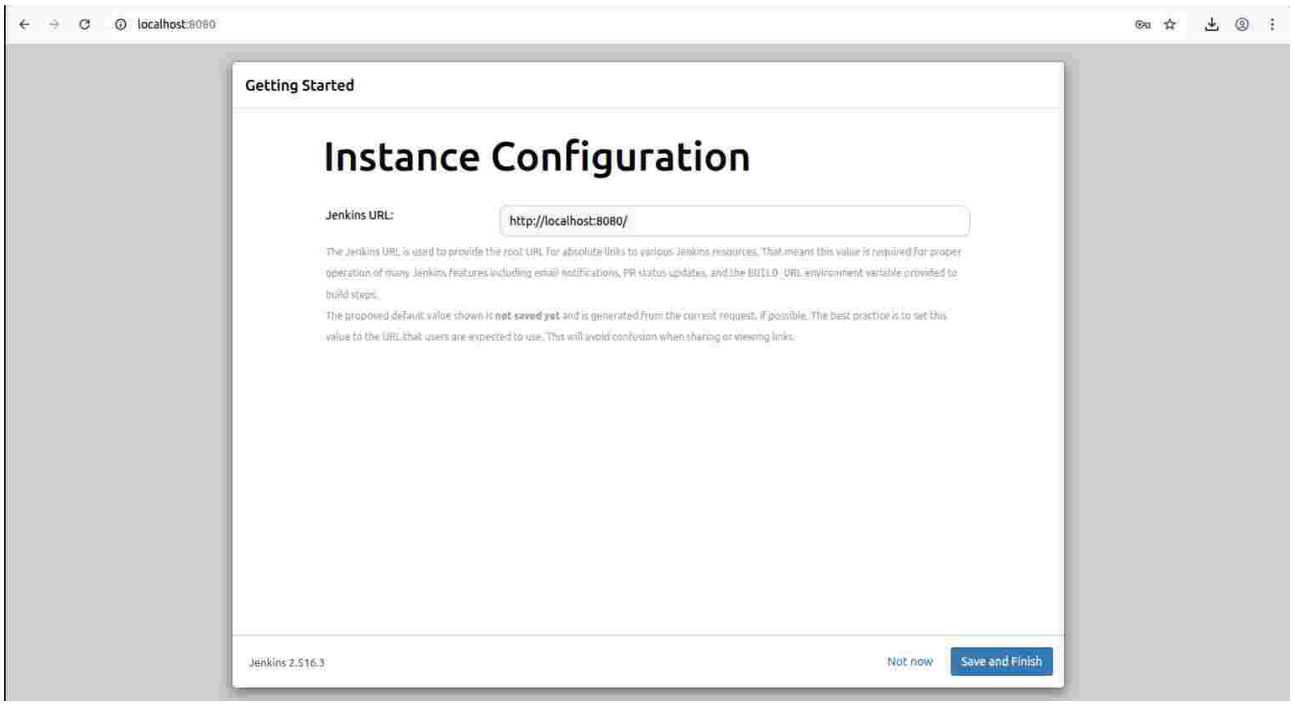
When the key is added, the system will return OK. Next, append the Debian package repository address to the server's sources.list:

manjusha@apsit:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable

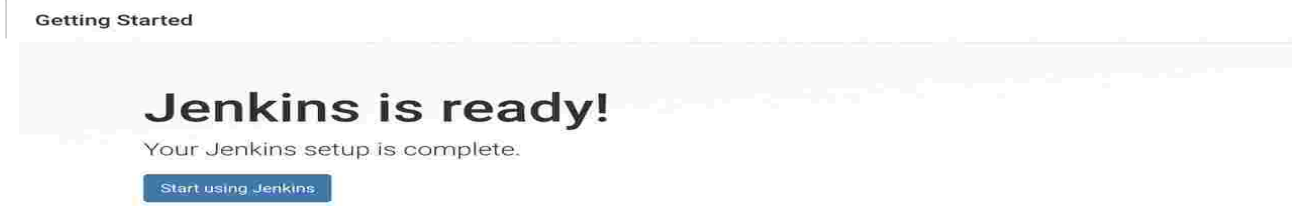

localhost:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue


localhost:8080
Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
LDAP
Email Extension
Mailer
Dark Theme
Jenkins 2.516.3

When the installation is complete, you will be prompted to set up the first administrative user. It's possible to skip this step and continue as admin using the initial password we used above, but we'll take a moment to create the user.



After confirming the appropriate information, click Save and Finish. You will see a confirmation page confirming that "Jenkins is Ready!":
Click Start using Jenkins to visit the main Jenkins dashboard:

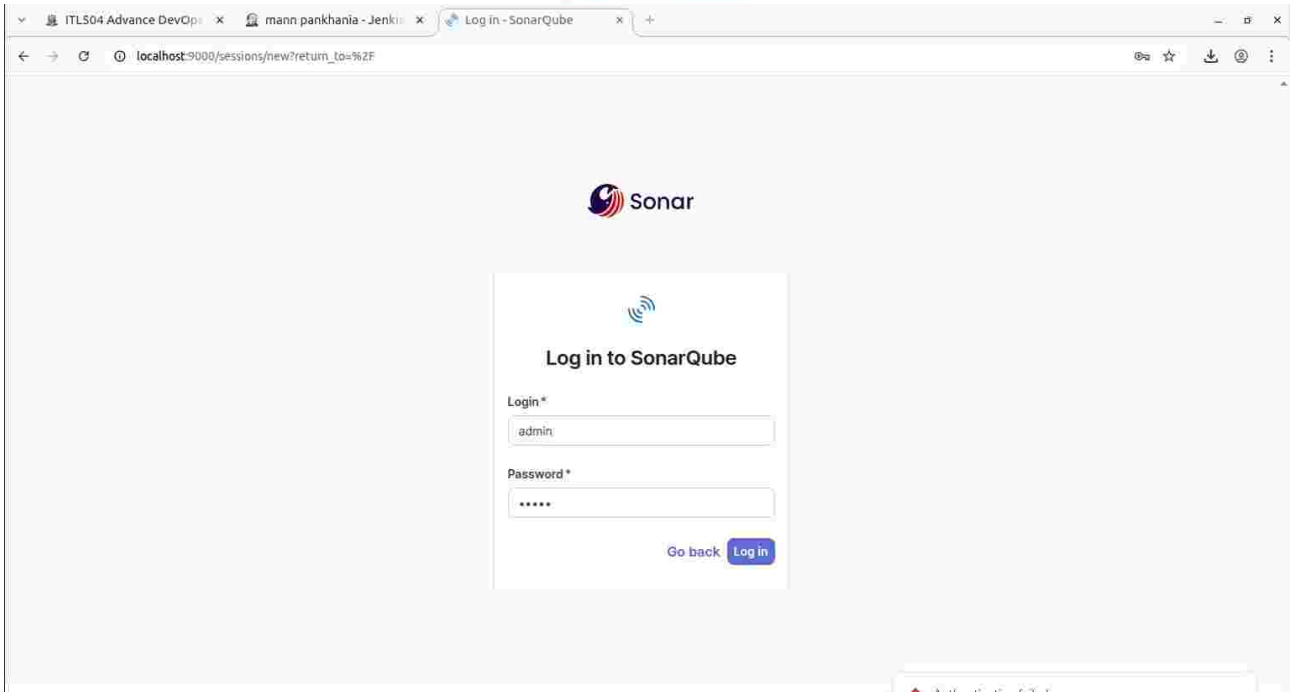


SonarQube Setup

Before proceeding with the integration, we will setup SonarQube Instance. we are using SonarQube Docker Container.

manjusha@apsit:~$docker run -d -p 9000:9000 sonarqube

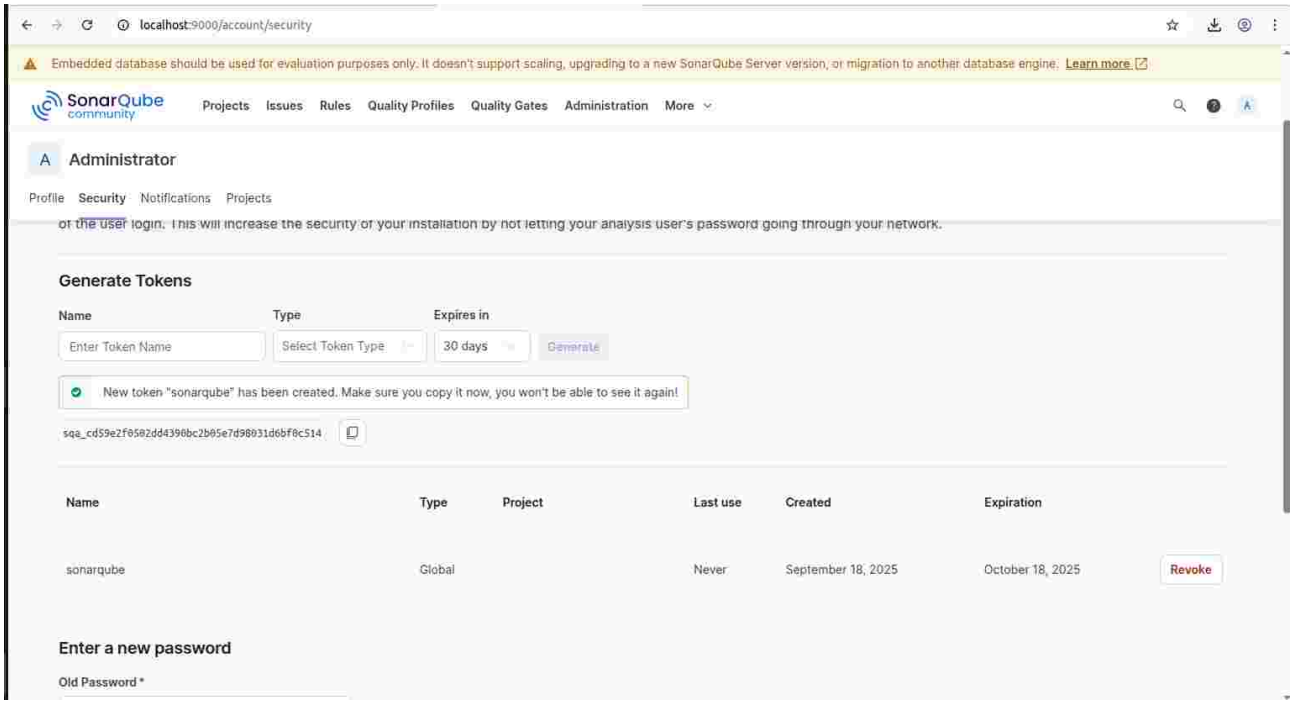
In the above command, we are forwarding port 9000 of the container to the port 9000 of the host machine as SonarQube is will run on port 9000. Then, from the browser, enter http://localhost:9000. After That, you will see the SonarQube is running. Then, login using default credentials (admin:admin).

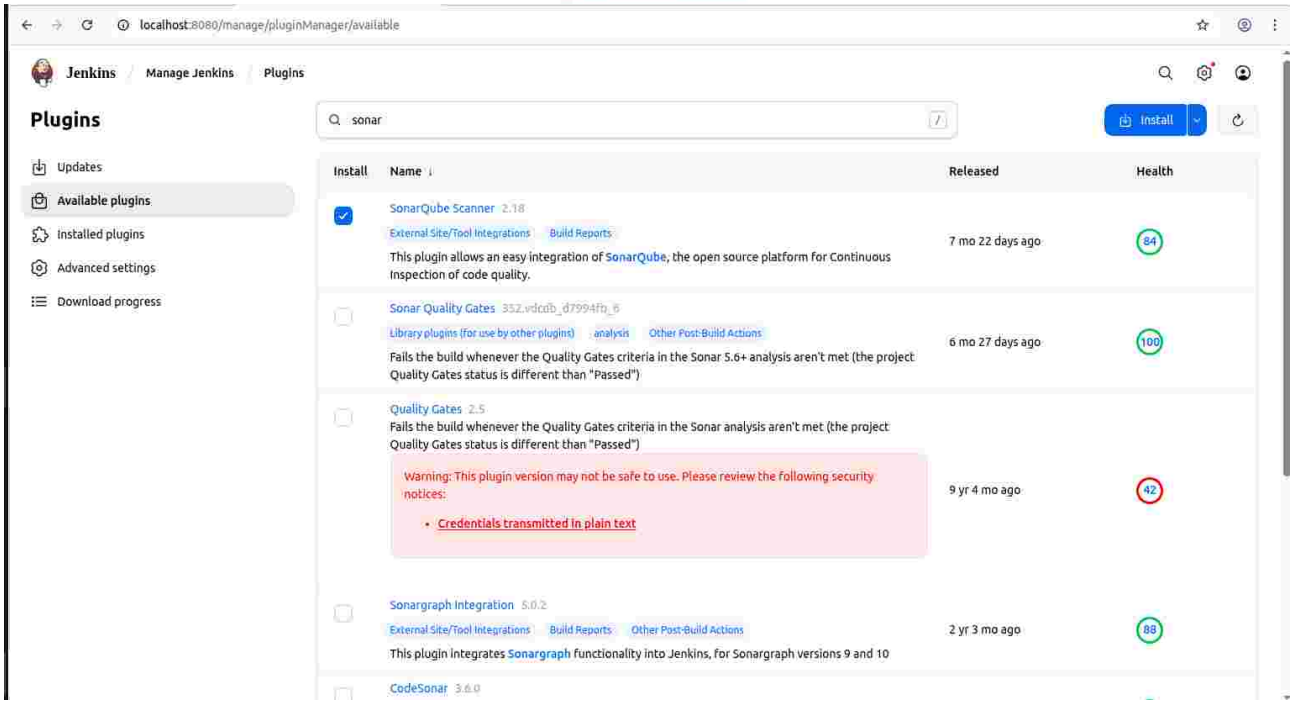


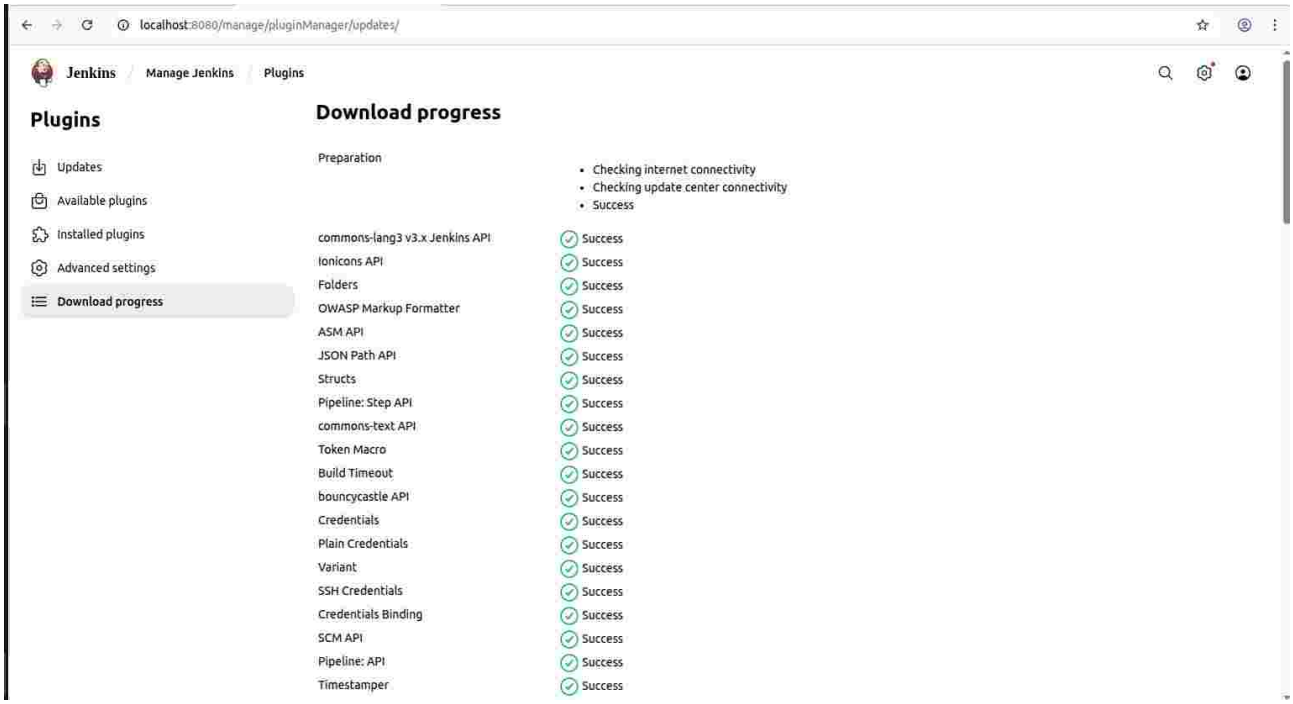
Generate User Token

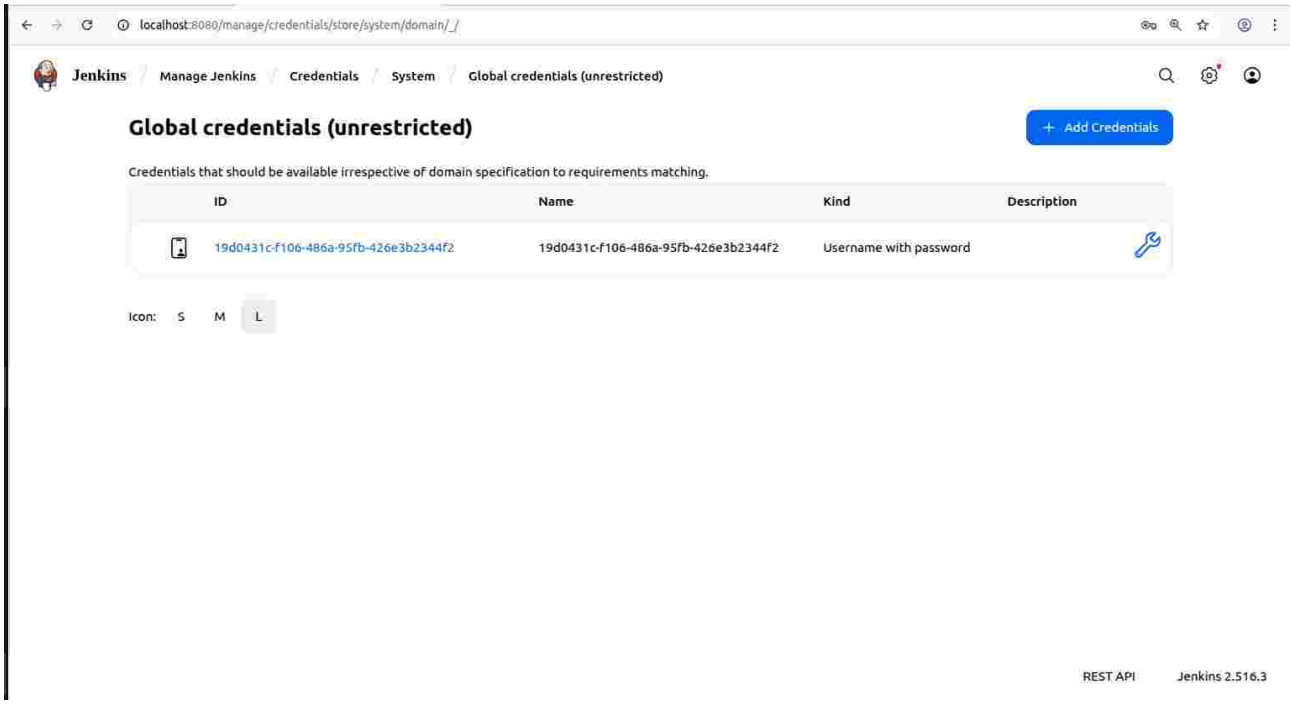
Now, we need to get the SonarQube user token to make connection between Jenkins and SonarQube. For the same, go to **Administration> User > My Account > Security** and then, from the bottom of the page you can create new tokens by clicking the Generate Button. Copy the Token and keep it safe.

C96798e9bd081e117189b516c868ddb7d87ee785 SonarQube


localhost:9000/account/security
Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more
SonarQube community
Projects Issues Rules Quality Profiles Quality Gates Administration More
A Administrator
Profile Security Notifications Projects
of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.
Generate Tokens
Name
Type
Expires in
Enter Token Name
Select Token Type
30 days
Generate
New token "sonarqube" has been created. Make sure you copy it now, you won't be able to see it again!
sqa_cd59e2f0502dd4390bc2b05e7d98031d6bf0c514
Name Type Project Last use Created Expiration
sonarqube Global Never September 18, 2025 October 18, 2025 Revoke
Enter a new password
Old Password *


localhost:8080/manage/pluginManager/available
Jenkins
Manage Jenkins
Plugins
Plugins
sonar
Install
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Install
Name
Released
Health
SonarQube Scanner 2.18
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
7 mo 22 days ago
84
Sonar Quality Gates 352.vdcd0_d7994fb_6
Library plugins (for use by other plugins)
analysis
Other Post-Build Actions
Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
6 mo 27 days ago
100
Quality Gates 2.5
Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")
Warning: This plugin version may not be safe to use. Please review the following security notices:
Credentials transmitted in plain text
9 yr 4 mo ago
42
Sonargraph Integration 5.0.2
External Site/Tool Integrations
Build Reports
Other Post-Build Actions
This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10
2 yr 3 mo ago
88
CodeSonar 3.6.0


localhost:8080/manage/pluginManager/updates/
Jenkins
Manage Jenkins
Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
commons-lang3 v3.x Jenkins API Success
Ionicons API Success
Folders Success
OWASP Markup Formatter Success
ASM API Success
JSON Path API Success
Structs Success
Pipeline: Step API Success
commons-text API Success
Token Macro Success
Build Timeout Success
bouncycastle API Success
Credentials Success
Plain Credentials Success
Variant Success
SSH Credentials Success
Credentials Binding Success
SCM API Success
Pipeline: API Success
Timestamper Success



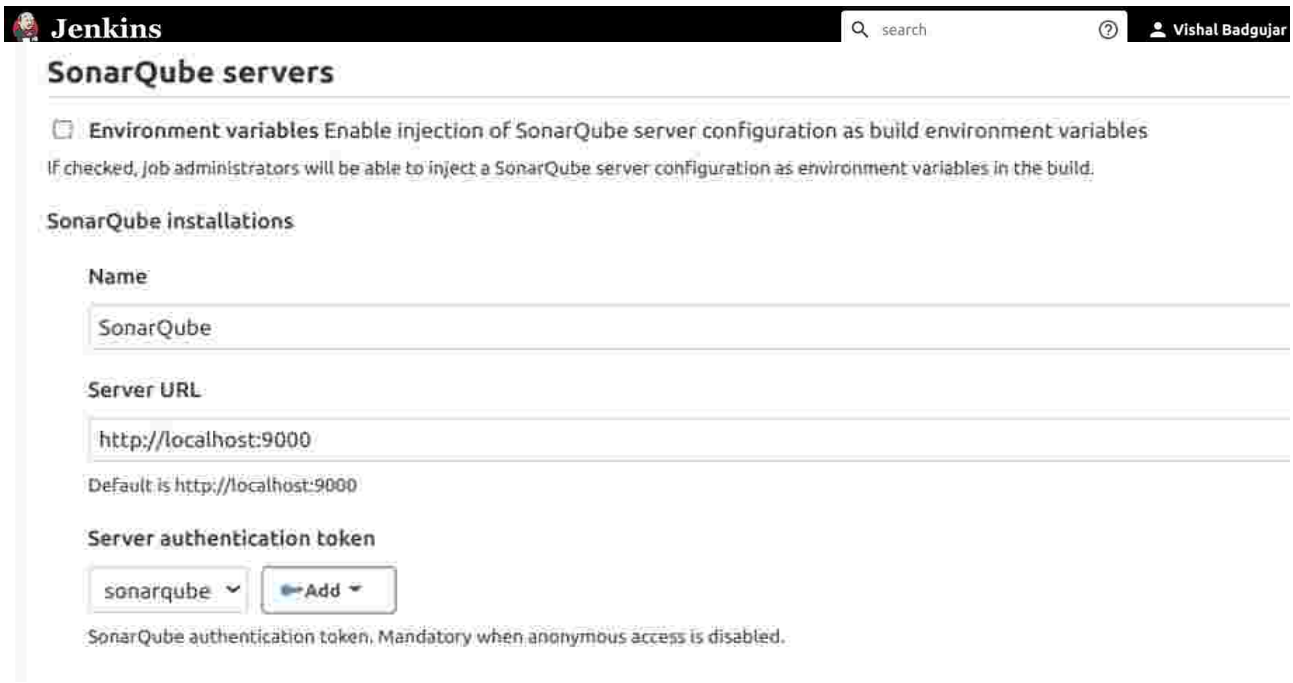
2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.

Jenkins Setup for SonarQube

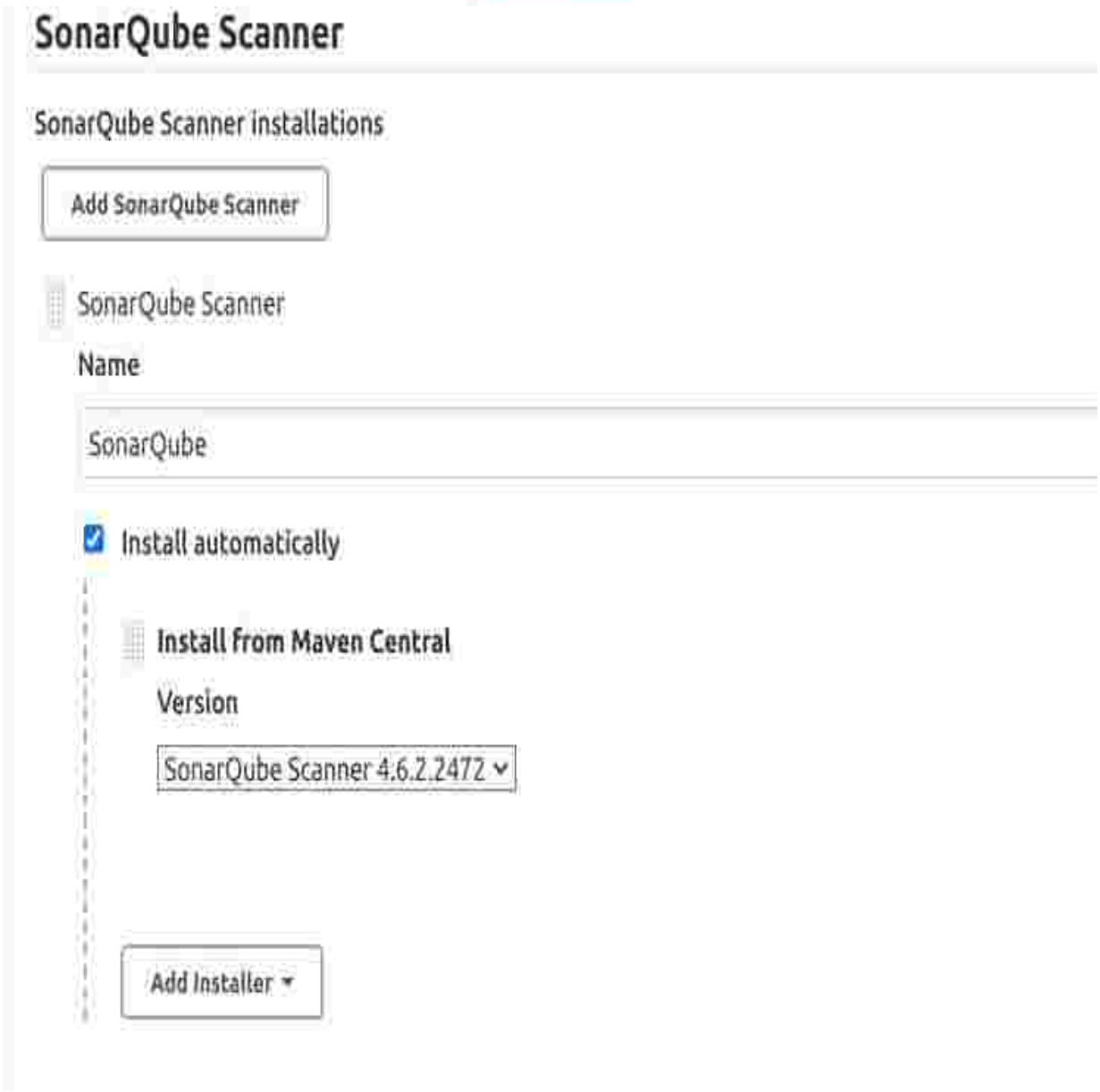
Before all, we need to install the SonarQube Scanner plugin in Jenkins. For the same, go to **Manage Jenkins > Plugin Manager > Available.** From here, type SonarQube Scanner then select and install.

Tool Configuration SonarQube Scanner

Now, we need to configure the Jenkins plugin for SonarQube Scanner to make a connection with the SonarQube Instance. For that, got to **Manage Jenkins > Configure System > SonarQube Server.** Then, Add SonarQube. In this, give the Installation Name, Server URL then Add the Authentication token in the Jenkins Credential Manager and select the same in the configuration.

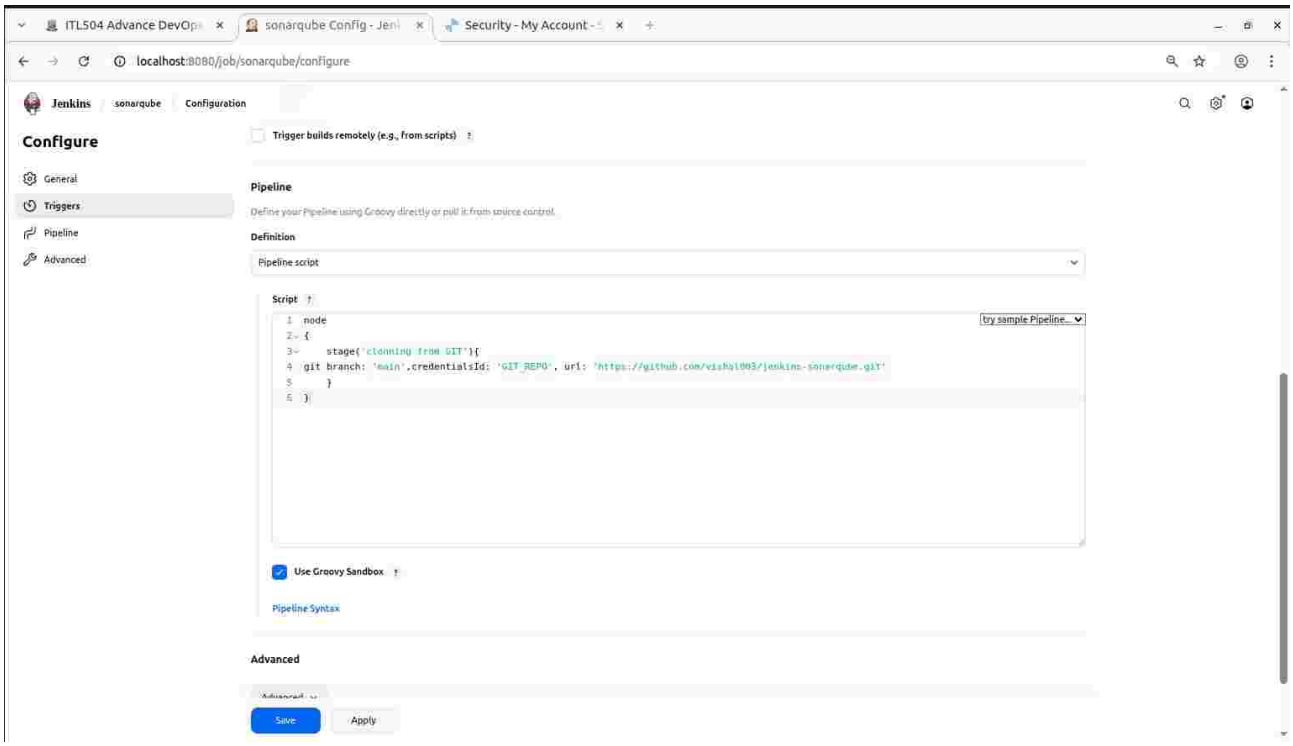


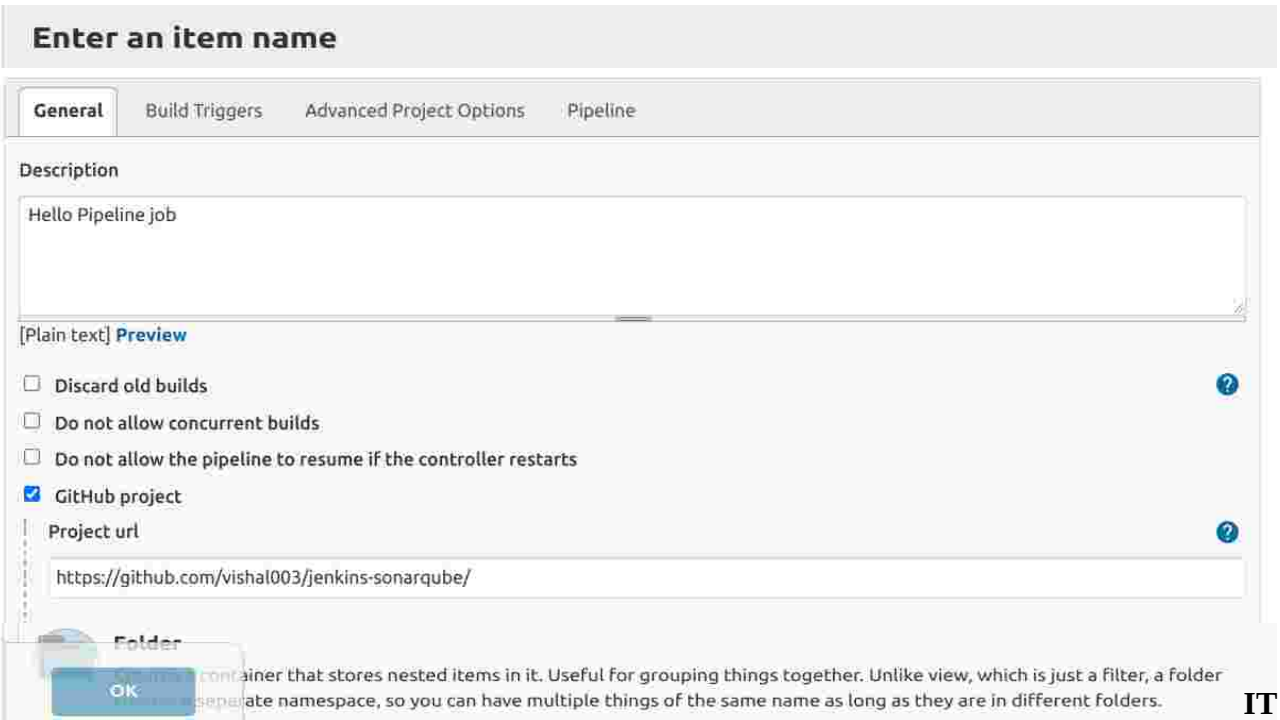
Then, we need to set-up the SonarQube Scanner to scan the source code in the various stage. For the same, go to **Manage Jenkins > Global Tool Configuration > SonarQube Scanner**. Then, Click **Add SonarQube Scanner Button**. From there, give some name of the scanner type and **Add Installer** of your choice. In this case, I have selected SonarQube Scanner from Maven Central.



SonarQube Scanner in Jenkins Pipeline

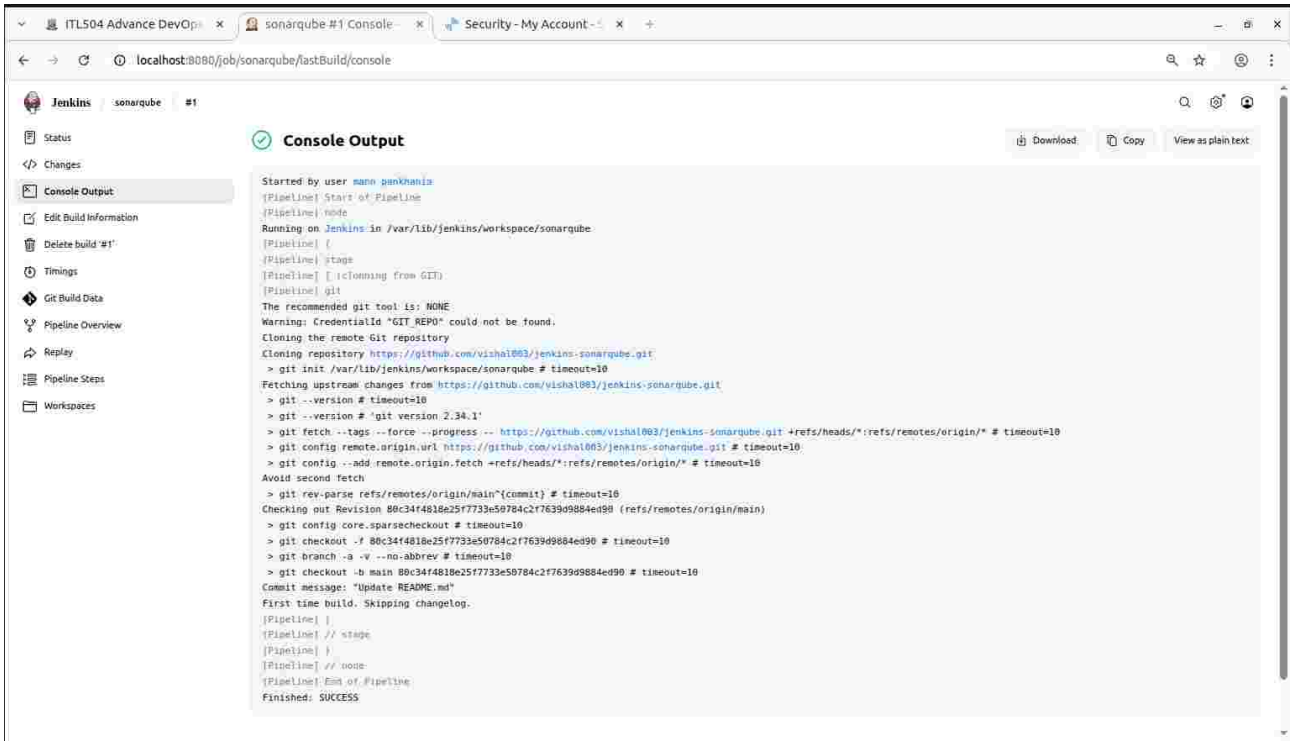
Now, It's time to integrate the SonarQube Scanner in the Jenkins Pipeline. For the same, we are going to add one more stage in the Jenkinsfile called SonarQube and inside that, I am adding the following settings and code.


ITL504 Advance DevOp
sonarqube Config - Jen
Security - My Account -
localhost:8080/job/sonarqube/configure
Jenkins
sonarqube
Configuration
Configure
General
Triggers
Pipeline
Advanced
Trigger builds remotely (e.g., from scripts)
Pipeline
Define your Pipeline using Groovy directly or pull it from source control
Definition
Pipeline script
Script
try sample Pipeline...
1 node
2 {
3 stage('clonning from GIT'){
4 git branch: 'main',credentialsId: 'GIT_REPO', url: 'https://github.com/vishal003/jenkins-sonarqube.git'
5 }
6 }
Use Groovy Sandbox
Pipeline Syntax
Advanced
Save
Apply


Enter an item name
General
Build Triggers
Advanced Project Options
Pipeline
Description
Hello Pipeline job
[Plain text] Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url
https://github.com/vishal003/jenkins-sonarqube/
Folder
ainer that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder
ate namespace, so you can have multiple things of the same name as long as they are in different folders.
OK

Github Configuration in Jenkins Pipeline

Git Clonning into Jenkins



Successfully Build Github Repository in Jenkins

Conclusion:

In this experiment, we learnt and understood how to 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.

Academic Year: 2022-23 Semester: V Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL) **Name: Huzaifa Bubere**
Subject Lab Incharge: Prof. Manjusha K. **Student Id: 24204006**

EXPERIMENT NO. 09

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

Theory:

Continuous monitoring is a process to detect, report, respond all the attacks which occur in its infrastructure. Once the application is deployed into the server, the role of continuous monitoring comes in to play. The entire process is all about taking care of the company's infrastructure and respond appropriately.

Why We Need Nagios tool?

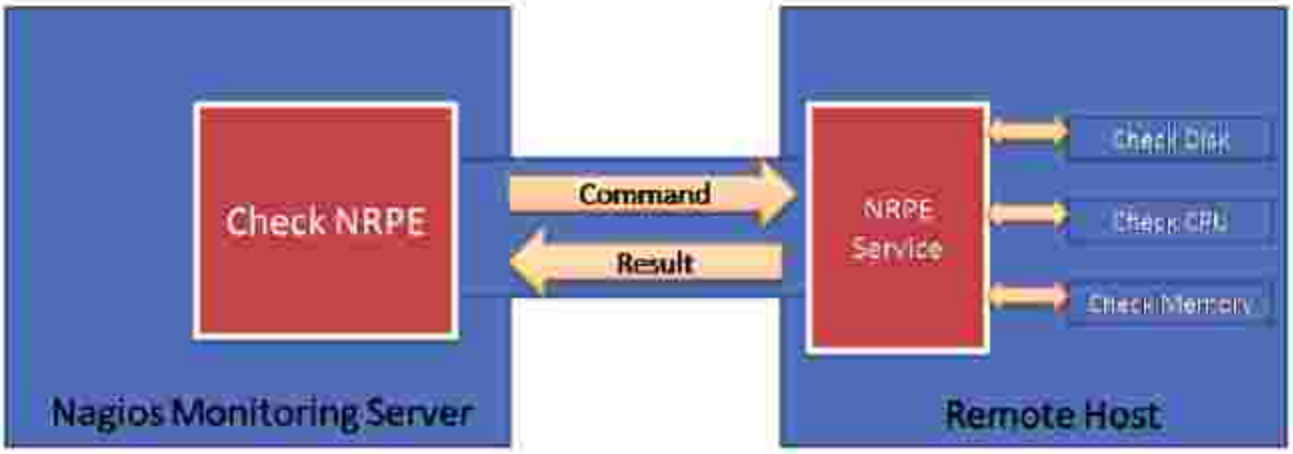
Here, are the important reasons to use Nagios monitoring tool:

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

• Nagios is the most popular, open source, powerful monitoring system for any kind of infrastructure. It enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. Nagios has the capability of monitoring application, services, entire IT infrastructure.

• NRPE is known as Nagios Remote Plugin Executor. The NRPE add-on is designed to execute plugins on remote Nix systems. In this setup, NRPE daemon is installed on the remote system to which services need to monitor through Nagios server.

• NRPE runs as a daemon on remote systems and waits for Nagios requests. When Nagios server needs to check the status of any resources or applications to that remote host, sends and commands signal, which command definition is stored on NRPE service. NRPE takes Nagios server request and execute the command on the local system and sends the result back to Nagios.



1. Pre-requistes:

First requirement is to install Apache and PHP first. Use the following commands to complete it. And use commands to install required packages for Nagios.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo apt-get install wget build-essential unzip openssl libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1.1).
wget is already the newest version (1.20.3-1ubuntu2.1).
The following packages were automatically installed and are no longer required:
  libgcc1:i386 libhttp-parser2.7.1 libssl1.0.0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libc-bin libc-dev-bin libc6 libc6:i386 libc6-dbg libc6-dev libc6-i386 libcbor0.6 libcrypt-dev libcrypt1 libcrypt1:i386 libfido2-1 libgssapi-krb5-2 libicu66
  libk5crypto3 libkrb5-3 libkrb5support0 libnghttp2-14 libnode-dev libnode64 libssl1.1 libuv1 libuv1-dev locales manpages-dev node-gyp node-semver nodejs
  openssh-client openssh-server openssh-sftp-server
Suggested packages:
  glibc-doc glibc-doc:i386 locales:i386 krb5-doc krb5-user libssl-doc keychain libpam-ssh monkeysphere ssh-askpass molly-guard
Recommended packages:
  libidn2-0 libidn2-0:i386
The following packages will be REMOVED:
  libssl1.0-dev nodejs-dev
The following NEW packages will be installed:
  libcbor0.6 libcrypt-dev libcrypt1 libcrypt1:i386 libfido2-1 libicu66 libnode-dev libnode64 libssl-dev
The following packages will be upgraded:
  libc-bin libc-dev-bin libc6 libc6:i386 libc6-dbg libc6-dev libc6-i386 libgssapi-krb5-2 libk5crypto3 libkrb5-3 libkrb5support0 libnghttp2-14 libssl1.1 libuv1
  libuv1-dev locales manpages-dev node-gyp node-semver nodejs openssh-client openssh-server openssh-sftp-server openssl unzip
25 upgraded, 9 newly installed, 2 to remove and 1830 not upgraded.
Need to get 22.4 MB/58.4 MB of archives.
After this operation, 40.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main i386 libcrypt1 i386 1:4.4.10-10ubuntu4 [90.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libcbor0.6 amd64 0.6.0-0ubuntu1 [21.1 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libfido2-1 amd64 1.3.1-1ubuntu2 [47.9 kB]
```

2 – Create Nagios User

Create a new user account for Nagios in your system and assign a password

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo adduser venny
Adding user `venny' ...
Adding new group `venny' (1004) ...
Adding new user `venny' (1003) with group `venny' ...
Creating home directory `/home/venny' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for venny
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo groupadd vennycmd
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo usermod -a -G vennycmd venny
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo usermod -a -G vennycmd www-data
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$
```

Step 3 – Install Nagios Core Service

After installing required dependencies and adding user accounts and Nagios core installation. Download latest Nagios core service from the official site.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:~$ cd /opt/
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt$ sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.3.tar.gz
--2025-09-24 11:04:54--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.3.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: 11302228 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.3.tar.gz.1'

nagios-4.4.3.tar.gz.1                   100%[=============================================================================>]  10.78M  2.48MB/s    in 4.3s

2025-09-24 11:04:59 (2.48 MB/s) - 'nagios-4.4.3.tar.gz.1' saved [11302228/11302228]

nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt$ sudo tar xzf nagios-4.4.3.tar.gz
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt$ cd nagios-4.4.3
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo ./configure --with-command-group=vennycmd
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
```

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make all
cd ./base && make
make[1]: Entering directory '/opt/nagios-4.4.3/base'
make -C ../lib
make[2]: Entering directory '/opt/nagios-4.4.3/lib'
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c squeue.c -o squeue.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c kvvec.c -o kvvec.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c iocache.c -o iocache.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c lobroker.c -o lobroker.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c bitmap.c -o bitmap.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c dkhash.c -o dkhash.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c runcmd.c -o runcmd.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c nsutils.c -o nsutils.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c fanout.c -o fanout.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c pqueue.c -o pqueue.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c worker.c -o worker.o
worker.c: In function 'enter_worker':
```

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make install-daemoninit
/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

*** Init script installed ***

nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg
```

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g vennycmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo make install-exfoliation

*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme
```

Now copy event handlers scripts under libexec directory. These binaries provides multiple events triggers for your Nagios web interface.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo chown -R nagios:nagios/usr/local/nagios/libexec/eventhandlers
chown: missing operand after 'nagios:nagios/usr/local/nagios/libexec/eventhandlers'
Try 'chown --help' for more information.
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo chown -R venny:nagios/usr/local/nagios/libexec/eventhandlers
chown: missing operand after 'venny:nagios/usr/local/nagios/libexec/eventhandlers'
Try 'chown --help' for more information.
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo chown -R venny:nagios /usr/local/nagios/libexec/eventhandlers
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$
```

Step 4 – Setup Apache with Authentication

Now create an Apache configuration file for your Nagios server as below:

Add below lines to nagios.conf file.

To setup apache authentication for user nagiosadmin

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo a2enconf nagios
Conf nagios already enabled
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo a2enmod cgi rewrite
Module cgi already enabled
Module rewrite already enabled
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo service apache2 restart
Job for apache2.service failed because the control process exited with error code.
See "systemctl status apache2.service" and "journalctl -xe" for details.
```

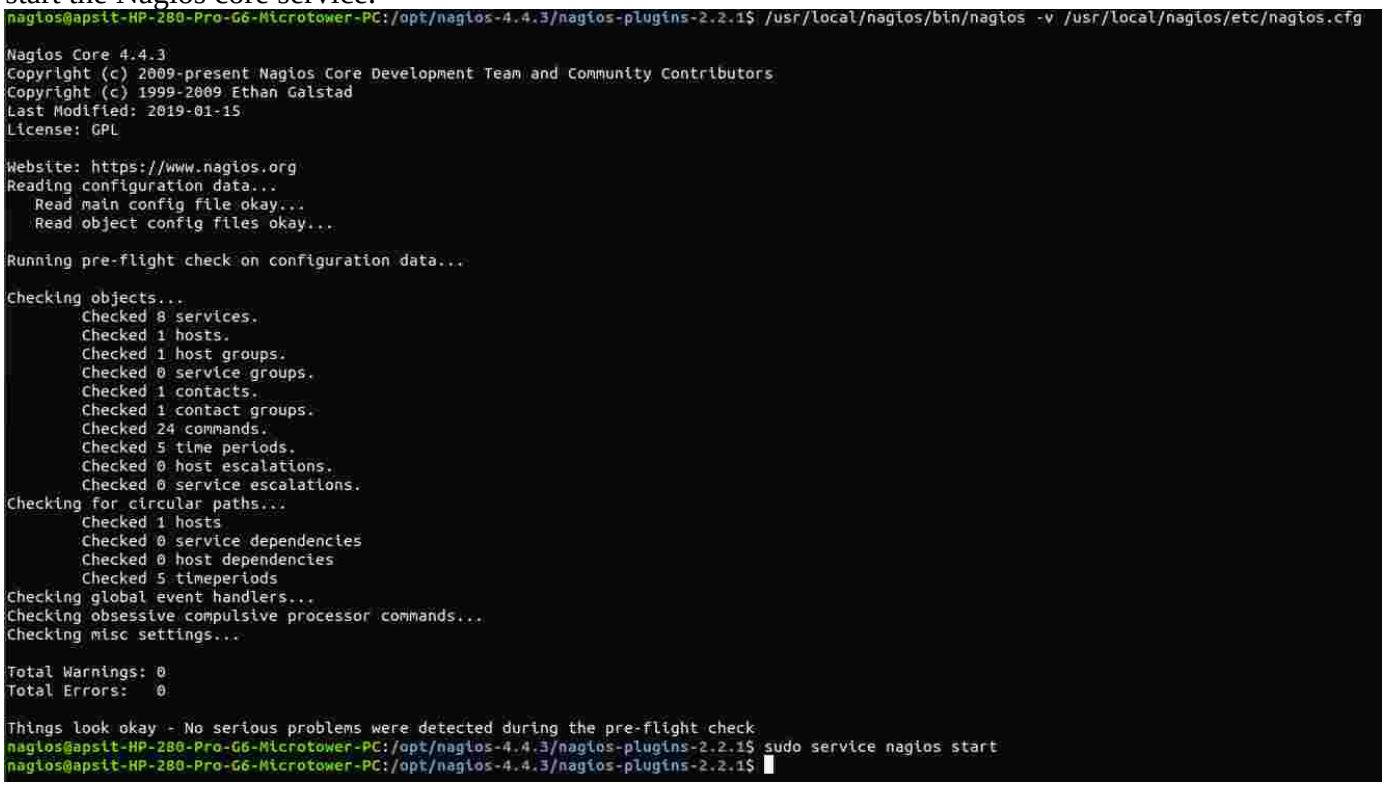
Step 5 – Installing Nagios Plugins

After installing and configuring Nagios core service, Download latest nagios-plugins source and install using following commands.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ sudo tar xzf nagios-plugins-2.2.1.tar.gz
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ cd nagios-plugins2.2.1
-bash: cd: nagios-plugins2.2.1: No such file or directory
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ ls
aclocal.m4       common          configure        doxy.conf      indent.sh   LICENSE       module                    pkginfo        subst     THANKS          xdata
autoconf-macros  config.guess    configure.ac     functions      INSTALLING  Makefile      nagios-plugins-2.2.1      pkginfo.in     subst.in  t-tap
base             config.log      contrib          html           install-sh  Makefile.in   nagios-plugins-2.2.1.tar.gz README.md     t         update-version
cgi              config.status   CONTRIBUTING.md  include        LEGAL       make-tarball  nagios.spec               sample-config  tap       UPGRADING
Changelog        config.sub      docs             indent-all.sh  lib         mkpackage     nagios.sysconfig          startup        test      worker
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3$ cd nagios-plugins-2.2.1
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo ./configure --with-nagios-user=nagios --with-nagios- group=nagios
configure: WARNING: unrecognized options: --with-nagios-
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
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
```

Step 6 – Verify Settings

Use the Nagios commands to verify the Nagios installation and configuration file. After successfully verify start the Nagios core service.



Step 7 – Access Nagios Web Interface

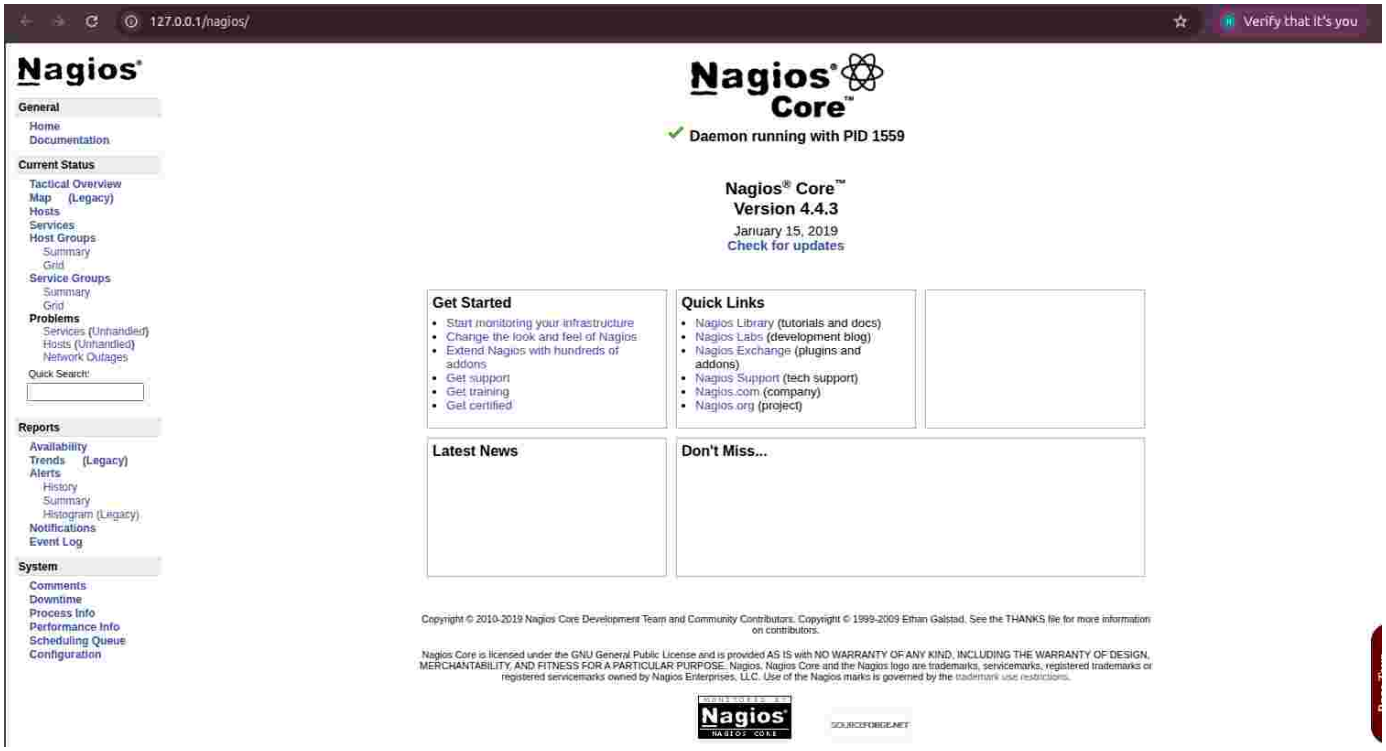
Access your nagios setup by access nagios server using hostname or ip address followed by /nagios. http://127.0.0.1/nagios/

Prompting for Apache Authentication Password –

username: nagiosadmin

Password : 123456 (which you enter while configuration)

Nagios After login screen –



Conclusion: We have successfully installed and configured Nagios Monitoring Server core service in our system now we need to install NRPE on all remote Linux systems to monitor with Nagios

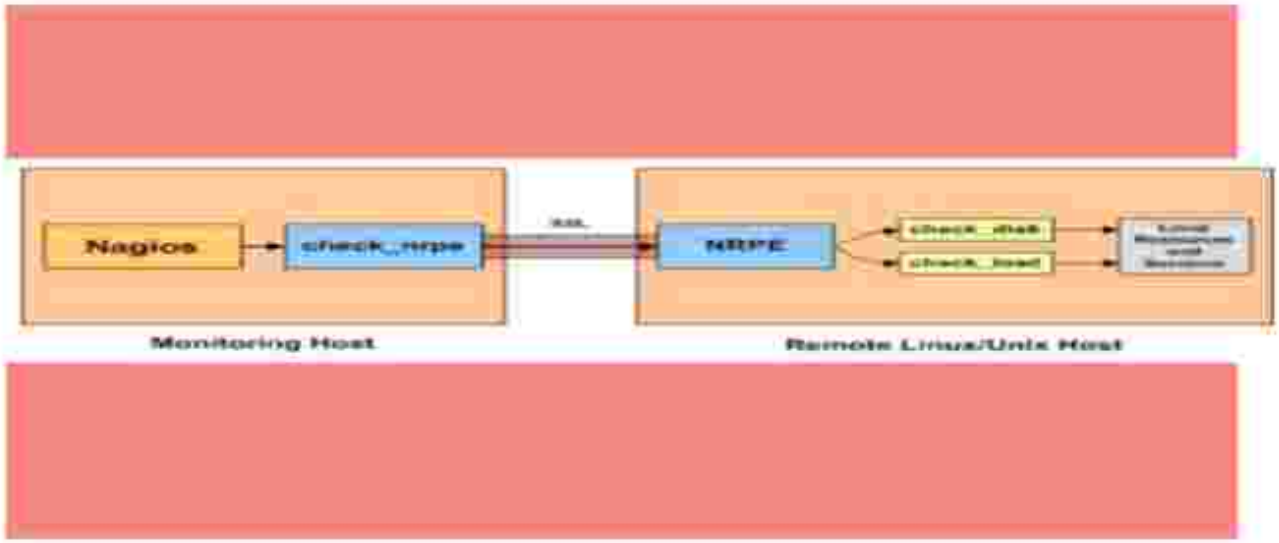
Academic Year: 2025-26 Semester: V Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL) **Name: Huzaifa Bubere**
Subject Lab Incharge: Prof Vishal Badgujar. **Student Id: 24204006**

EXPERIMENT NO. 10

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

Theory:

Monitoring remote Linux/Unix hosts is to use the NRPE addon. NRPE allows you to execute plugins on remote Linux/Unix hosts. This is useful if you need to monitor local resources/attributes like disk usage, CPU load, memory usage, etc. on a remote host.



Note: To perform this experiment Experiment 9 is pre-requisite where we have configured Nagios on Linux System. Here In this Experiment we will Add a Linux Host to Nagios for Monitoring purpose.

Step 1 – Configure NRPE on Linux Host:
Follow the below steps to install and configure NRPE on client machine and check connectivity with Nagios server.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo apt-get install nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree
Reading state information... Done
nagios-nrpe-server is already the newest version (4.0.0-2ubuntu1).
nagios-plugins is already the newest version (2.2-3ubuntu3.18.04.1).
The following packages were automatically installed and are no longer required:
  libgcc1:i386 libhttp-parser2.7.1 libssl1.0.0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1829 not upgraded.
```

Step 1.2 – Configure NRPE

After successfully installing NRPE service, Edit nrpe configuration file /etc/nagios/nrpe.cfg in your favorite editor and add your nagios service ip in allowed hosts.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo nano /etc/nagios/nrpe.cfg
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo /etc/init.d/nagios-nrpe-server restart
[ ok ] Restarting nagios-nrpe-server (via systemctl): nagios-nrpe-server.service.
```

allowed_hosts=127.0.0.1, 192.168.86.13, 192.168.86.30

After making above changes in nrpe configuration file, Lets restart NRPE service as per your system

Step 1.3 – Verify Connectivity from Nagios

Now run the below command from Nagios server to make sure your nagios is able to connect nrpe client on remote Linux system. Here 192.168.86.13 is your remote Linux system ip.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ /usr/lib/nagios/plugins/check_nrpe -H 192.168.86.13
NRPE v4.0.0
```

Step 2 – Add Linux Host in Nagios

First create a configuration file using below values. for example you Linux hosts ip is . We also need to define a service with host. So add a ping check service, which will continuously check that host is up or not.

```
# ------------------------------
# Host definition
# ------------------------------
define host {
    use             linux-server
    host_name           Linux_Host_001
    alias           Linux Host 001
    address           192.168.86.13
```

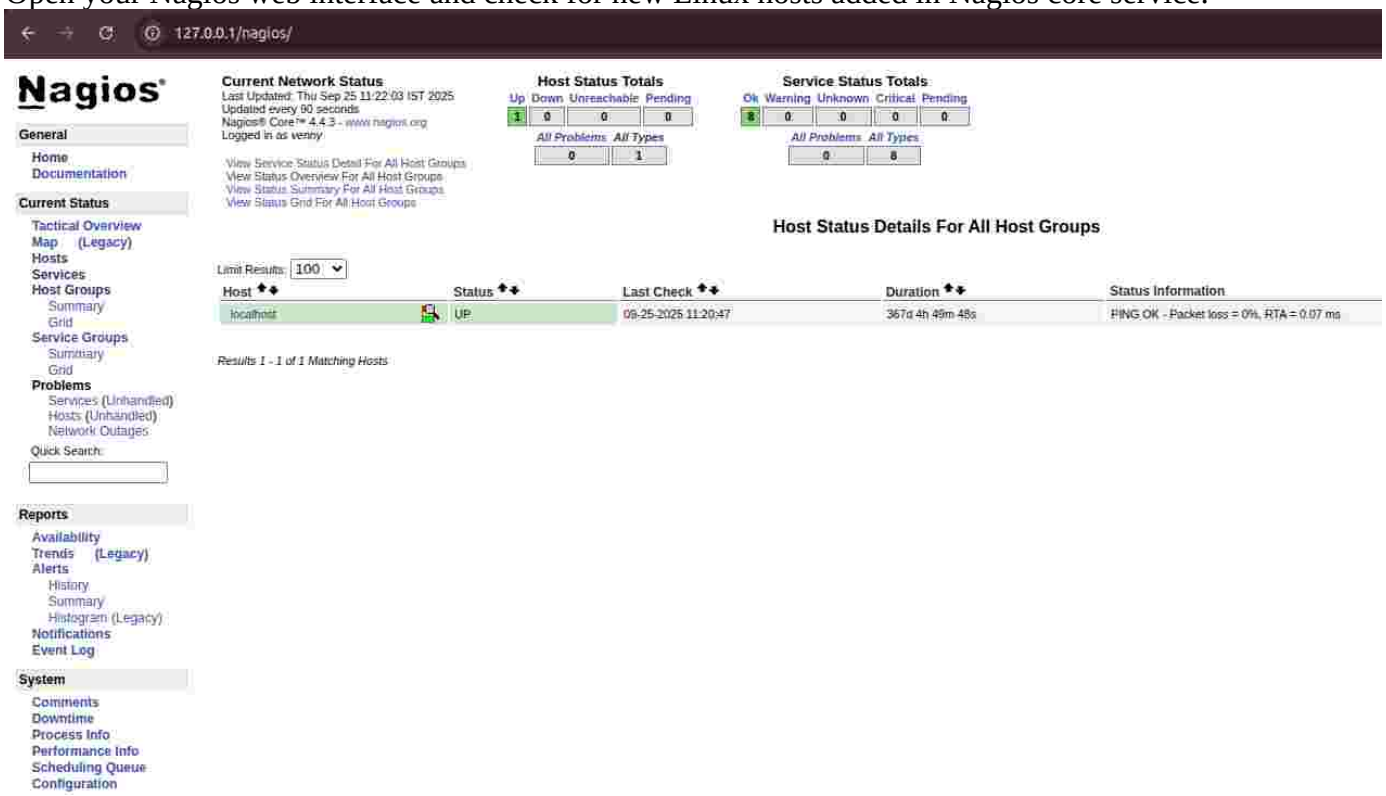
```
    register                1
}


# ------------------------------
# Service definition (PING check)
# ------------------------------
define service {
    use                     generic-service
    host_name               Linux_Host_001
    service_description     PING
    check_command           check_ping!100.0,20%!500.0,60%
    max_check_attempts      2
    check_interval          2
    retry_interval          2
    check_period            24x7
    check_freshness         1
    contact_groups          admins
    notification_interval   2
    notification_period     24x7
    notifications_enabled   1
    register                1
}
```

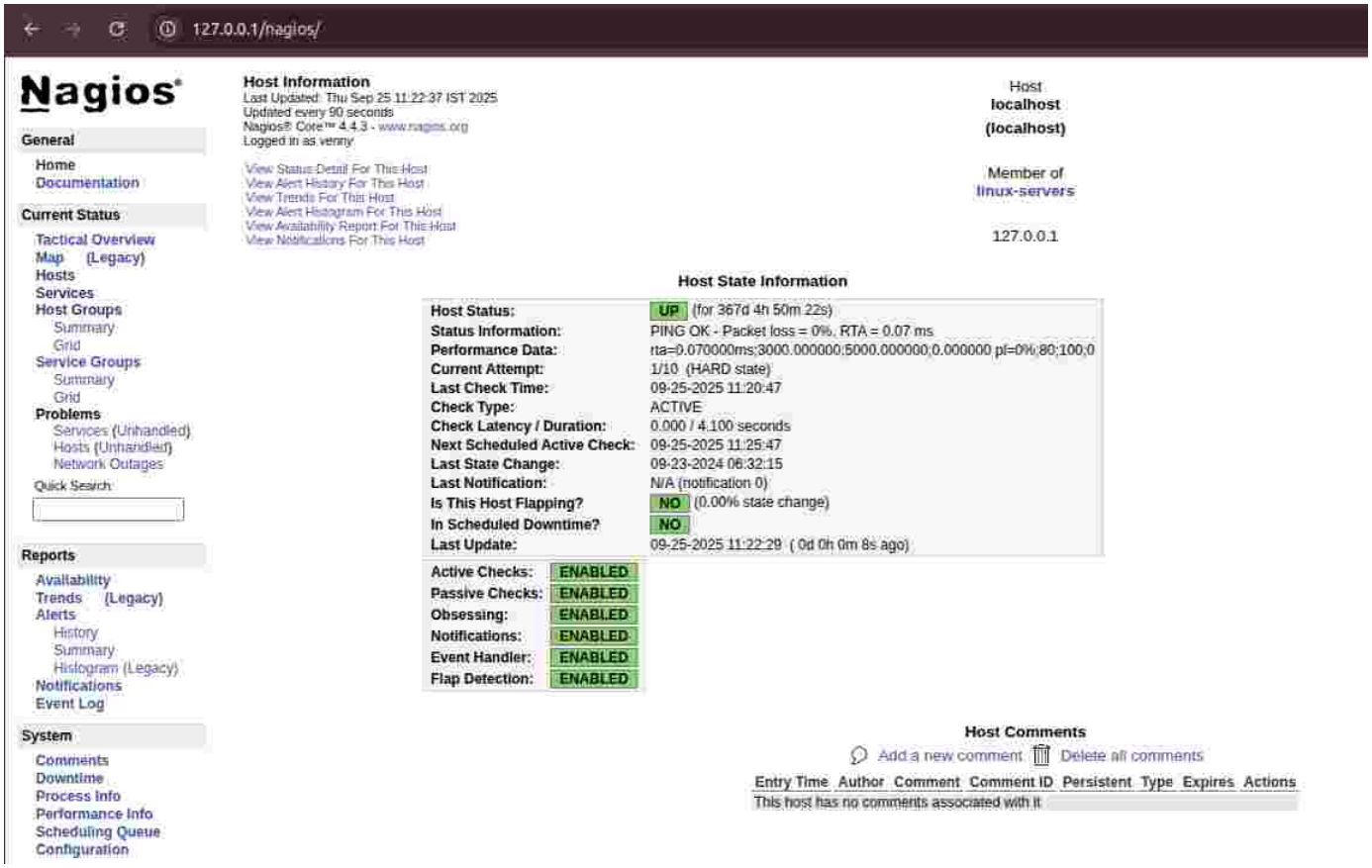
Now verify configuration files using following command. If there are no errors found in configuration, restart nagios service.

```
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo service nagios restart
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo service nagios start
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
htpasswd: password verification error
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users venny
New password:
Re-type new password:
Adding password for user venny
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart apache2
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart nagios
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart apache2
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo service nagios start
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ cat /usr/local/nagios/etc/htpasswd.users
venny:$apr1$cAKlWL7h$vGtferAkxLlRCX0oe4Fg6.
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo nano /usr/local/nagios/etc/cgi.cfg
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart nagios
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart apache2
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ ls /etc/apache2/conf-available/
 charset.conf    javascript-common.conf       nagios.conf         other-vhosts-access-log.conf   serve-cgi-bin.conf
 gitweb.conf     localized-error-pages.conf  'nagios.conf:wq!'    security.conf
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo nano /etc/apache2/conf-available/nagios.conf
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo a2enconf nagios
Conf nagios already enabled
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo a2enmod cgi
Module cgi already enabled
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo systemctl restart apache2
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$ sudo service nagios start
nagios@apsit-HP-280-Pro-G6-Microtower-PC:/opt/nagios-4.4.3/nagios-plugins-2.2.1$
```

Step 3 – Check Host in Nagios Web Interface

Open your Nagios web interface and check for new Linux hosts added in Nagios core service.





Conclusion: We have successfully perform Port, Service monitoring, Linux server monitoring using Nagios.

Semester: V
Academic Year: 2022-23
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID: 24204006

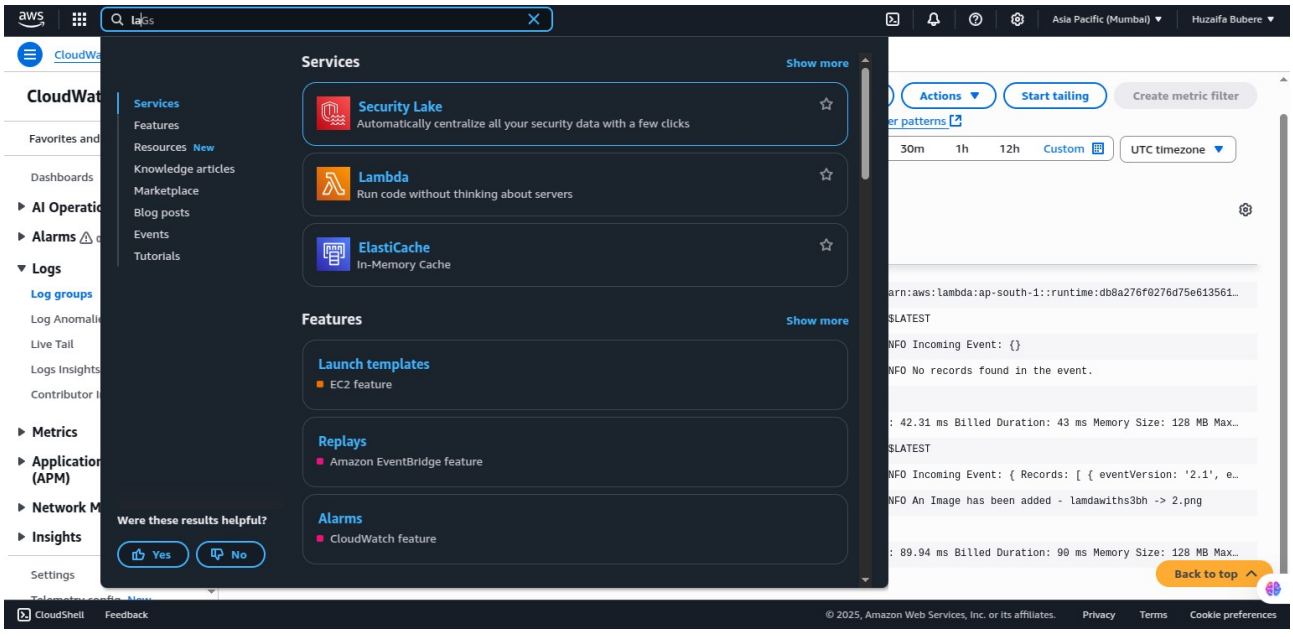
EXPERIMENT NO. 11

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

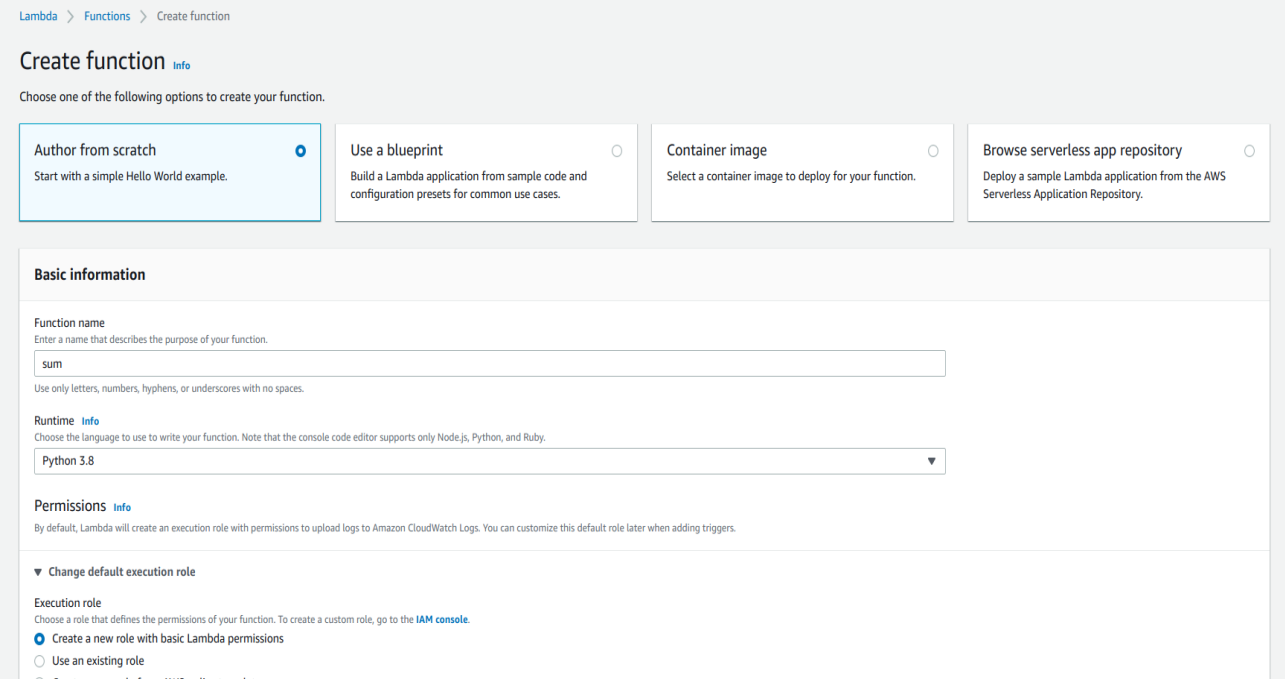
Steps: First Lambda functions using Python

1. Open Aws Console and search for Lambda Service and open home screen of Lambda.

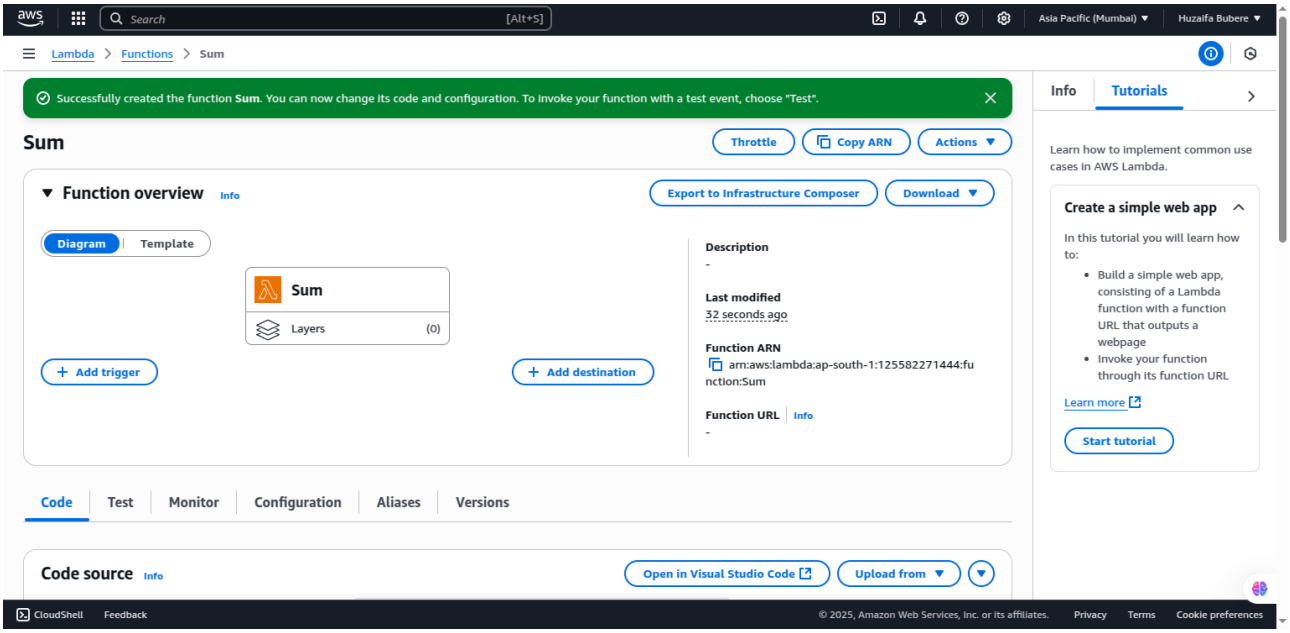
2. Choose region in which you need to create Lambda function as it is region specific.



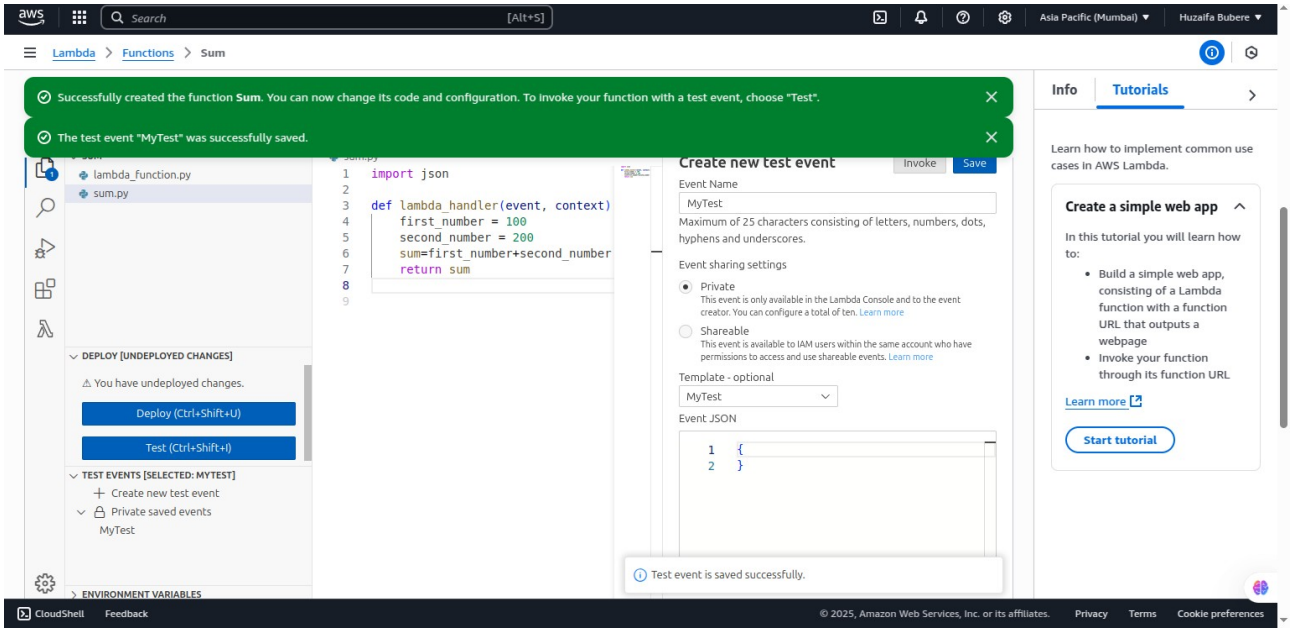
3. Create sum as a Lambda Function in Python Language so select latest version of Python and choose role with basic Lambda Permission to allow cloudwatch for monitoring.



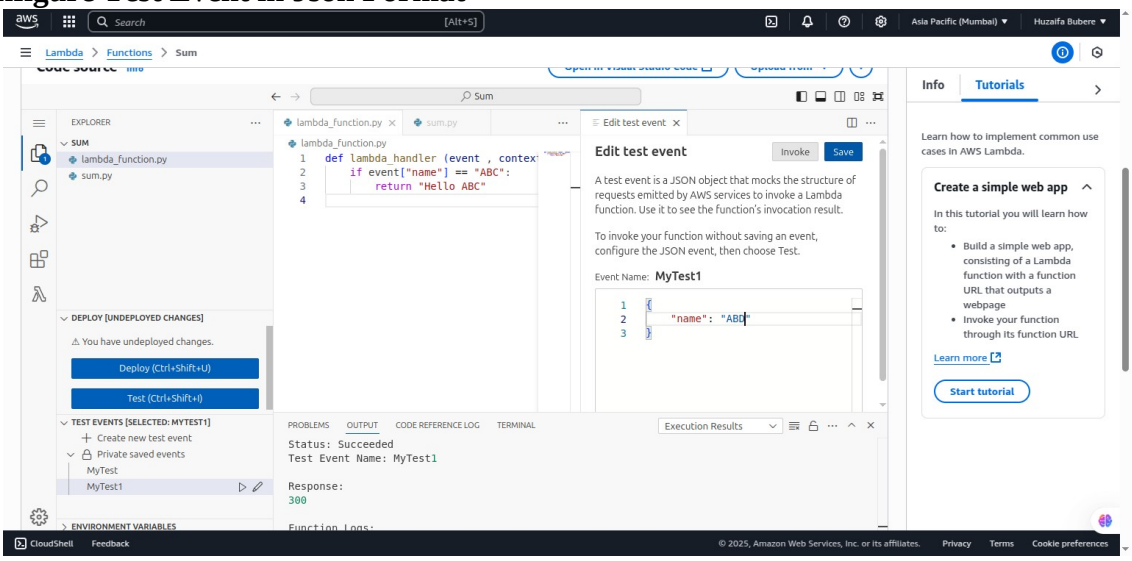
4. Lambda sum function is created successfully



Write a sample python code for sum of two numbers:



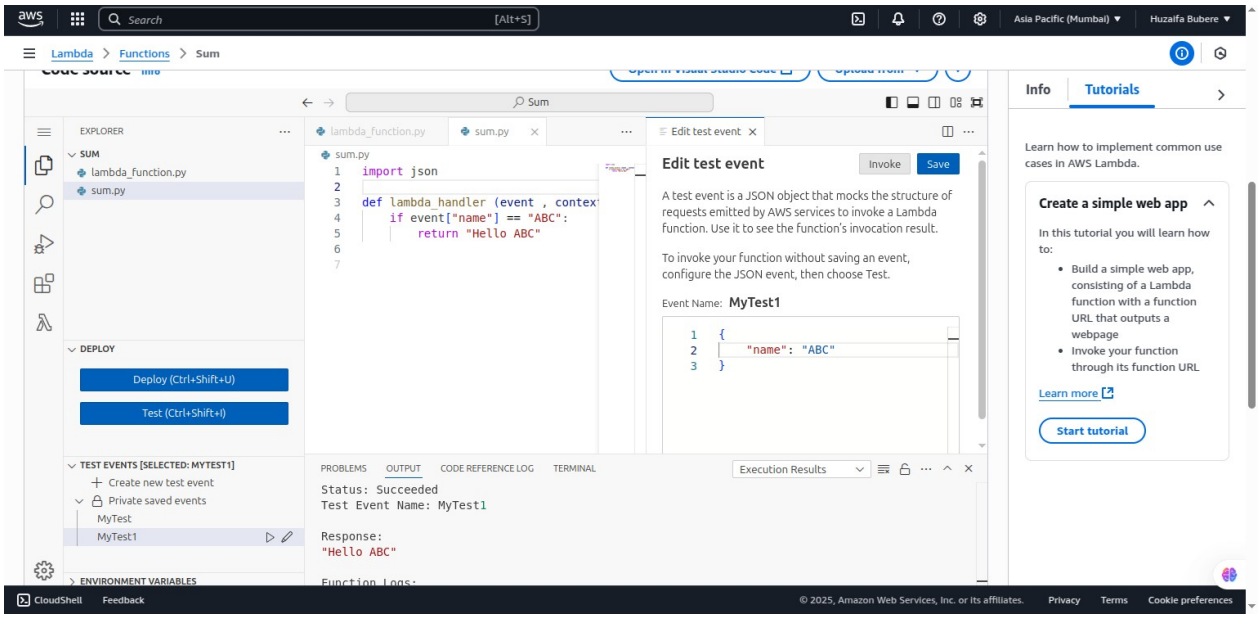
6. Configure Test Event in Json Format



Write a sample Second sample python Code:

Configure Test Event

If condition met returns a value as apsit



Conclusion: Write your own findings.

Semester: V
Academic Year: 2022-23
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor:Prof. Vishal Badgujar

Name of Student: Huzaifa Bubere
Student ID: 24204006

EXPERIMENT NO. 12

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

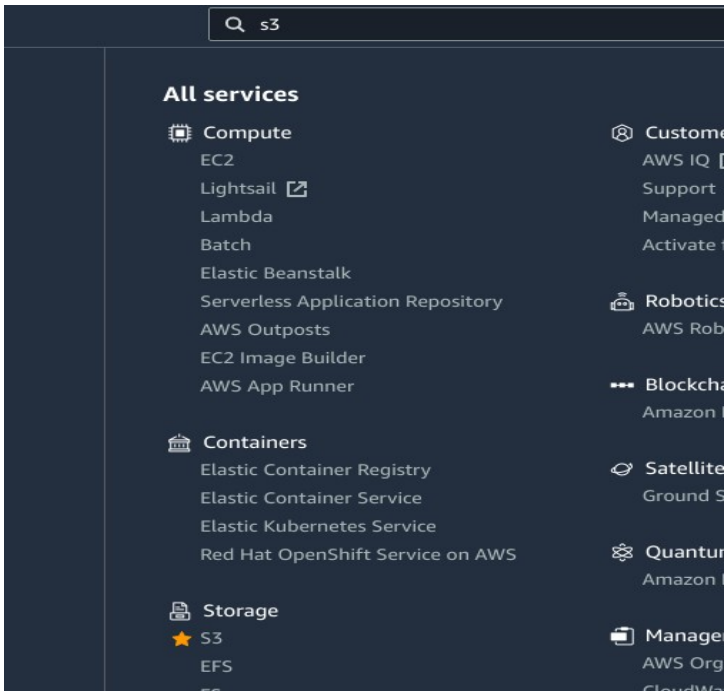
Theory:

Creating S3 Bucket

Let us start first by creating a s3 bucket in AWS console using the steps given below –

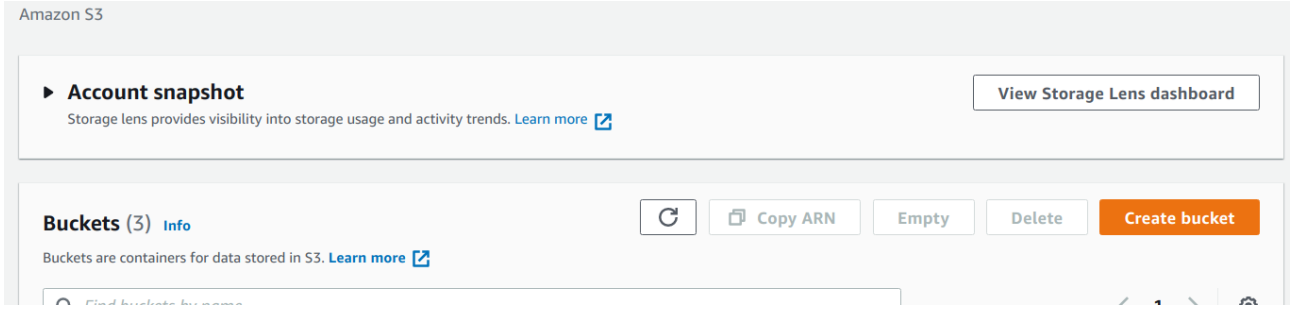
Step 1

Go to Amazon services and click **S3** in storage section as highlighted in the image given below –



Step 2

Click **S3** storage and **Create bucket** which will store the files uploaded.

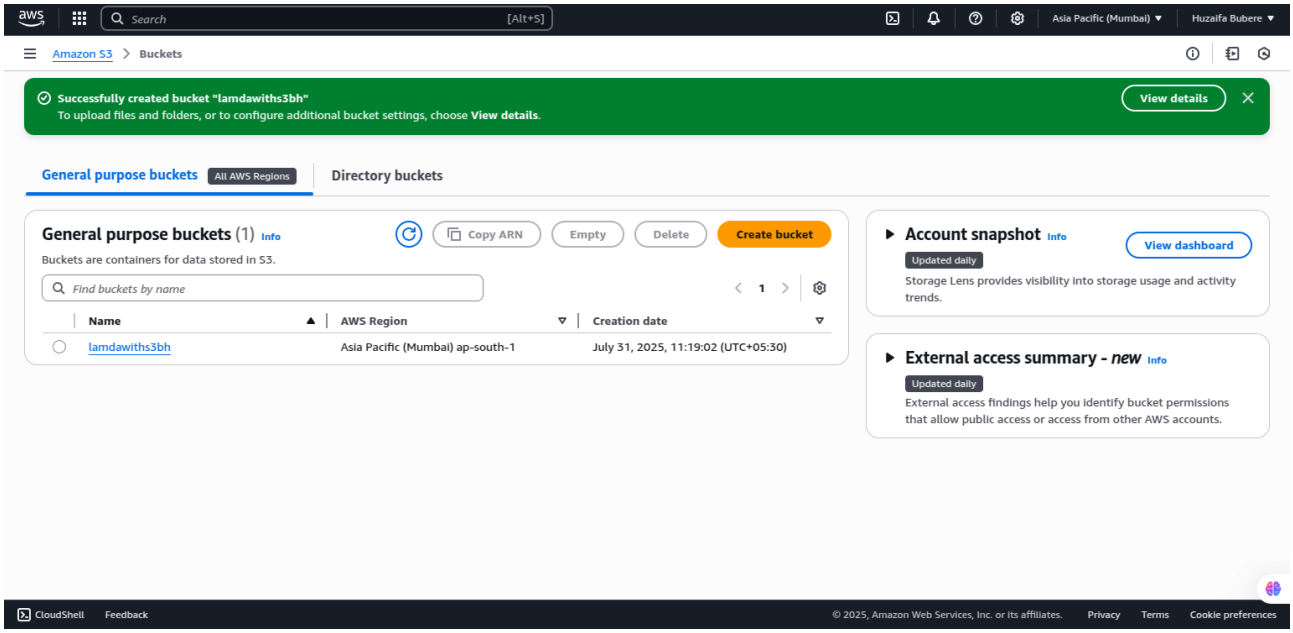


Step 3

Once you click Create bucket button, you can see a screen as follows –

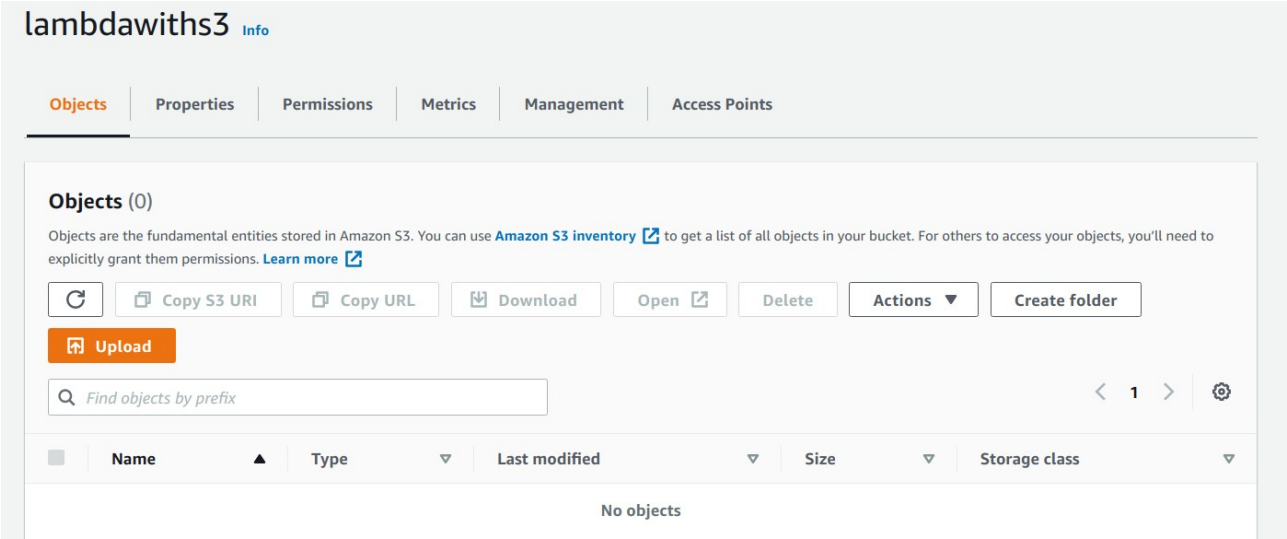
Step 4

Enter the details Bucket name, Select the Region and click Create button at the bottom left side. Thus, we have created bucket with name :



Step 5

Now, click the bucket name and it will ask you to upload files as shown below –



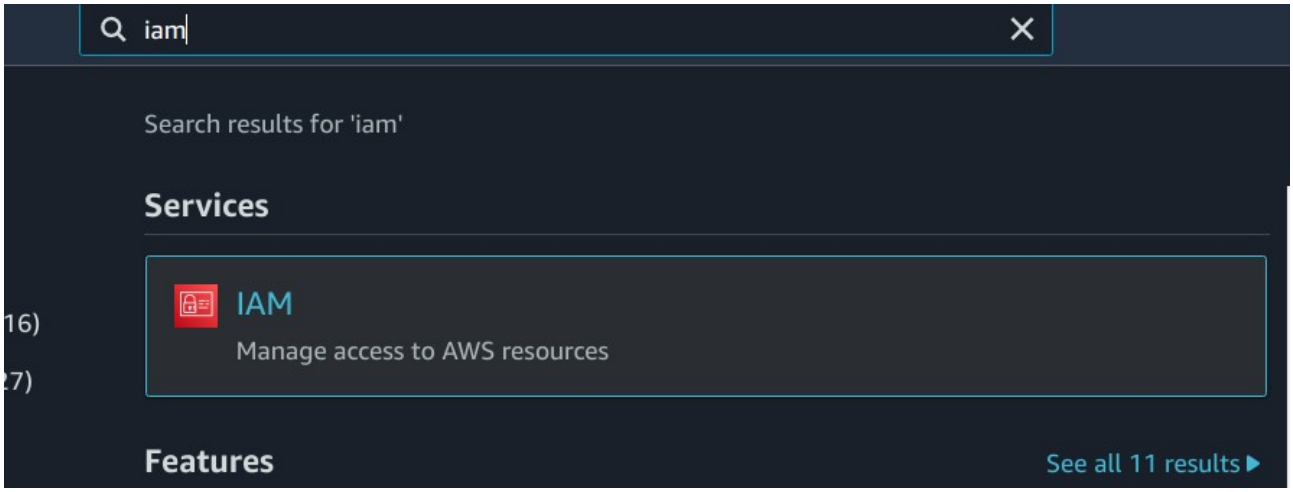
Thus, we are done with bucket creation in S3.

Create Role that Works with S3 and Lambda

To create role that works with S3 and Lambda, please follow the Steps given below –

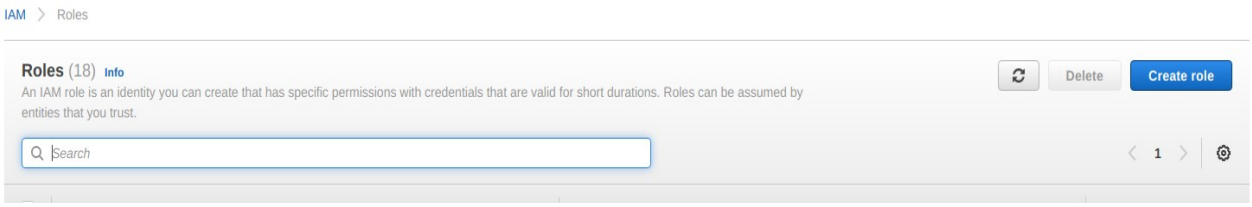
Step 1

Go to AWS services and select IAM as shown below –



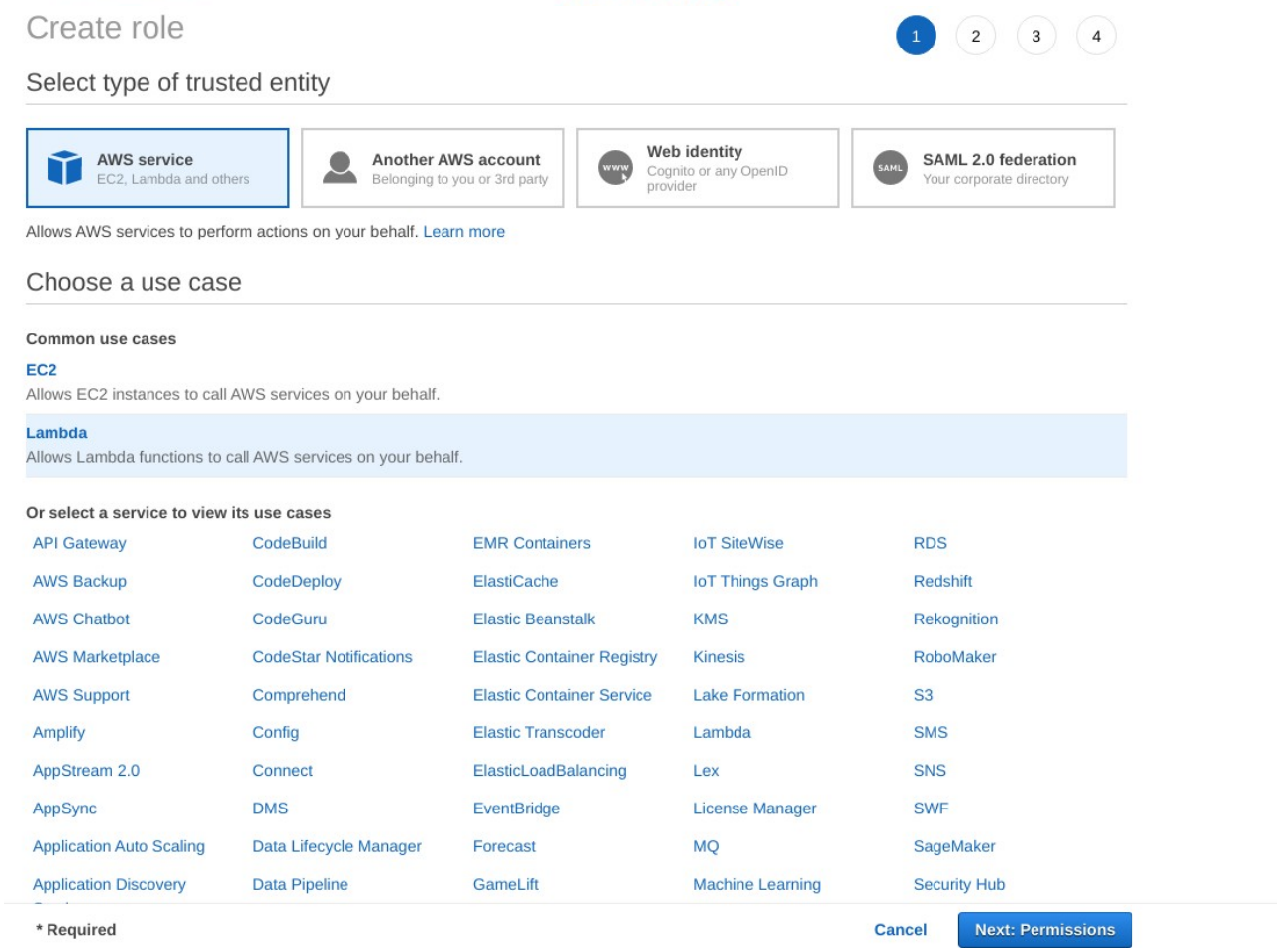
Step 2

Now, click **IAM -> Roles** as shown below –



Step 3

Now, click **Create role** and choose the services that will use this role. Select Lambda and click **Permission** button.



Step 4

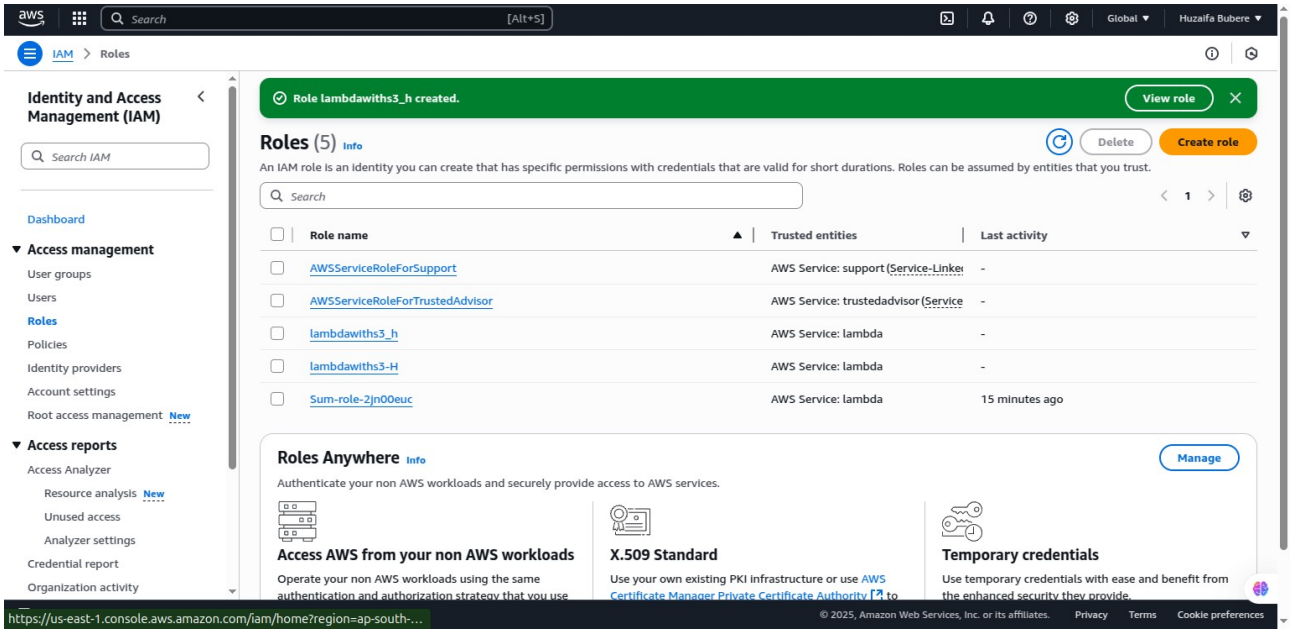
Add the permission from below and click Review.

AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.

Step 5

Observe that we have chosen the following permissions –

Observe that the Policies that we have selected are **AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.**



Step 6

Now, enter the Role name, Role description and click Create Role button at the bottom.

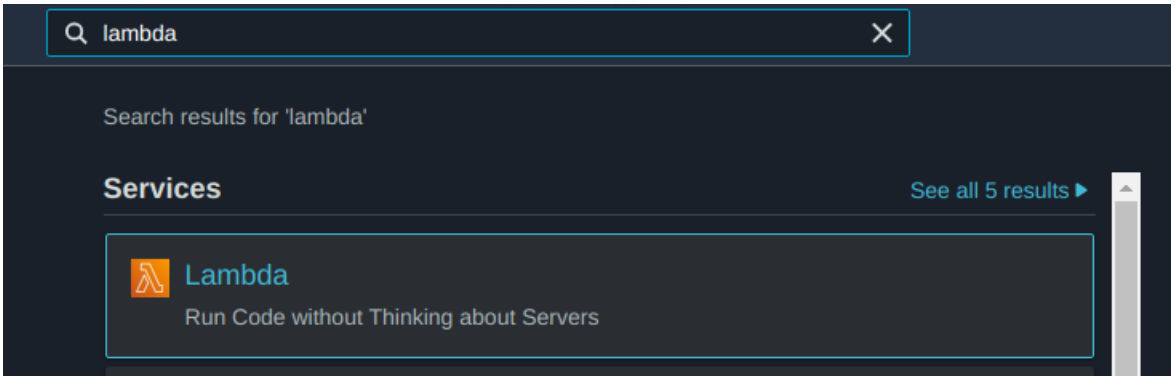
Thus, our role named lambdawiths3service is created.

Create Lambda function and Add S3 Trigger

In this section, let us see how to create a Lambda function and add a S3 trigger to it. For this purpose, you will have to follow th Steps given below –

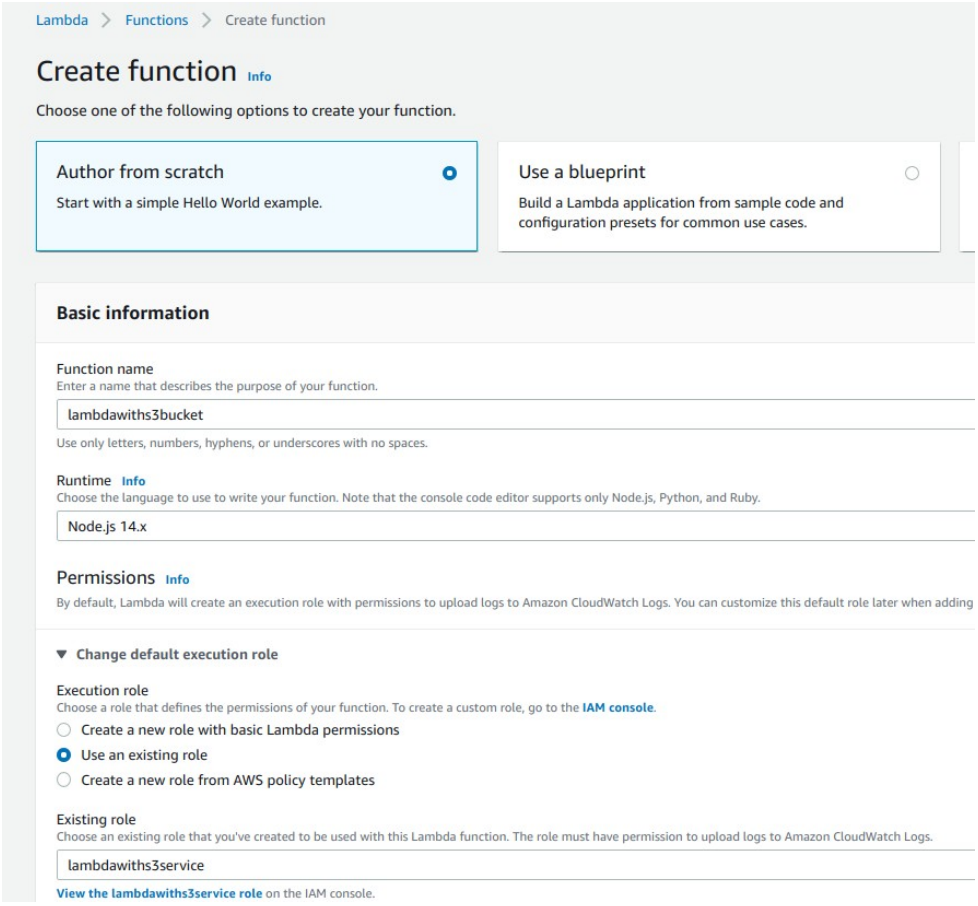
Step 1

Go to AWS Services and select Lambda as shown below –



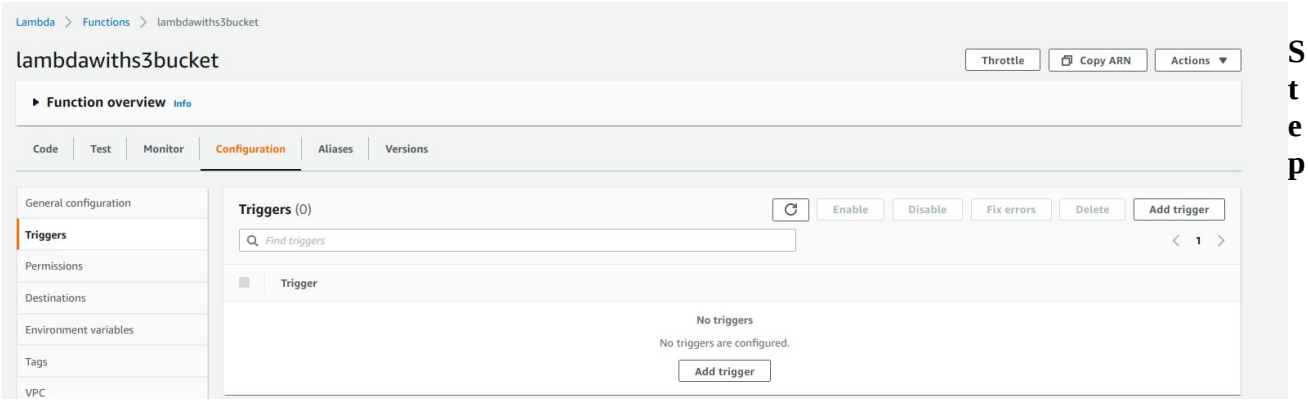
Step 2

Click **Lambda** and follow the process for adding **Name**. Choose the **Runtime, Role** etc. and create the function. The Lambda function that we have created is shown in the screenshot below –



Step 3

Now let us add the S3 trigger.



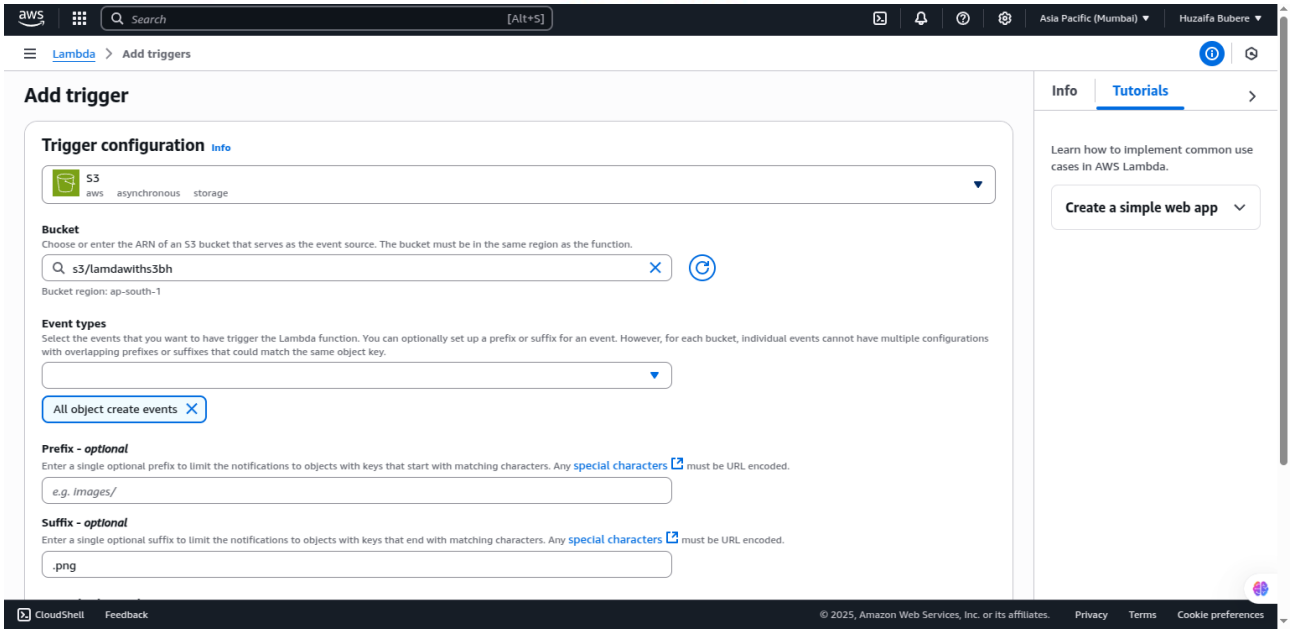
Step 4

Choose the trigger from above and add the details as shown below –

You can add Prefix and File pattern which are used to filter the files added. For Example, to trigger lambda only for .jpg images. as we need to trigger Lambda for all jpg image files uploaded. Click Add button to add the trigger.

Step 5

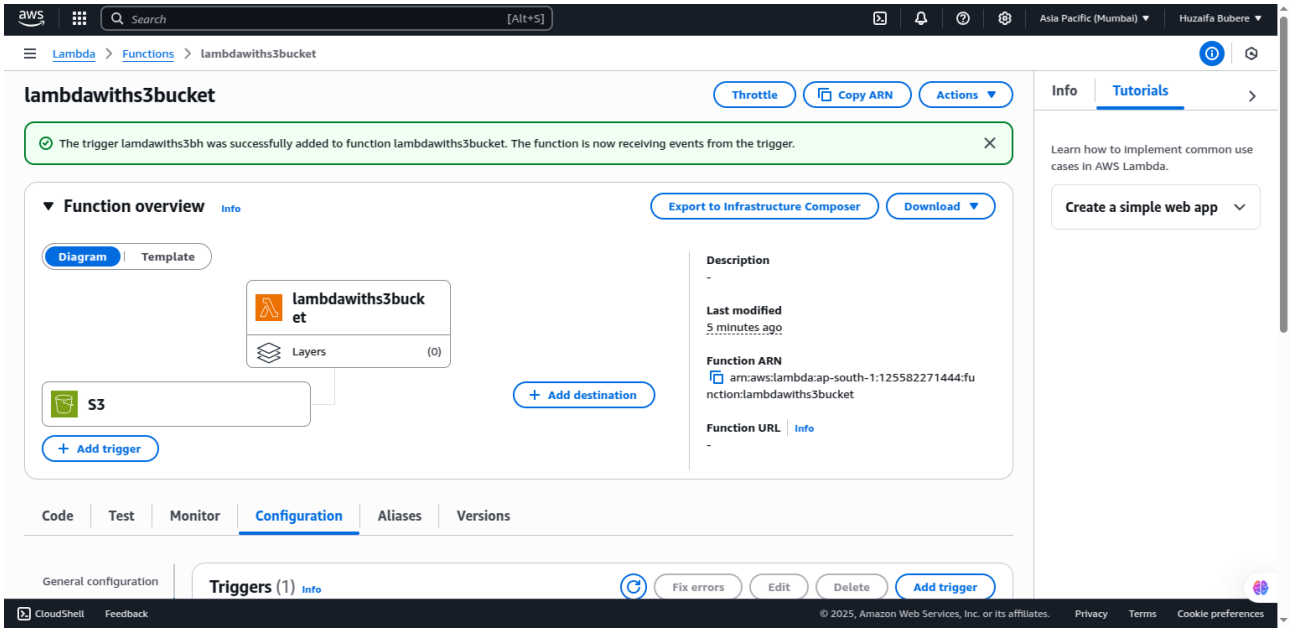
You can find the the trigger display for the Lambda function as shown below –



Step 6

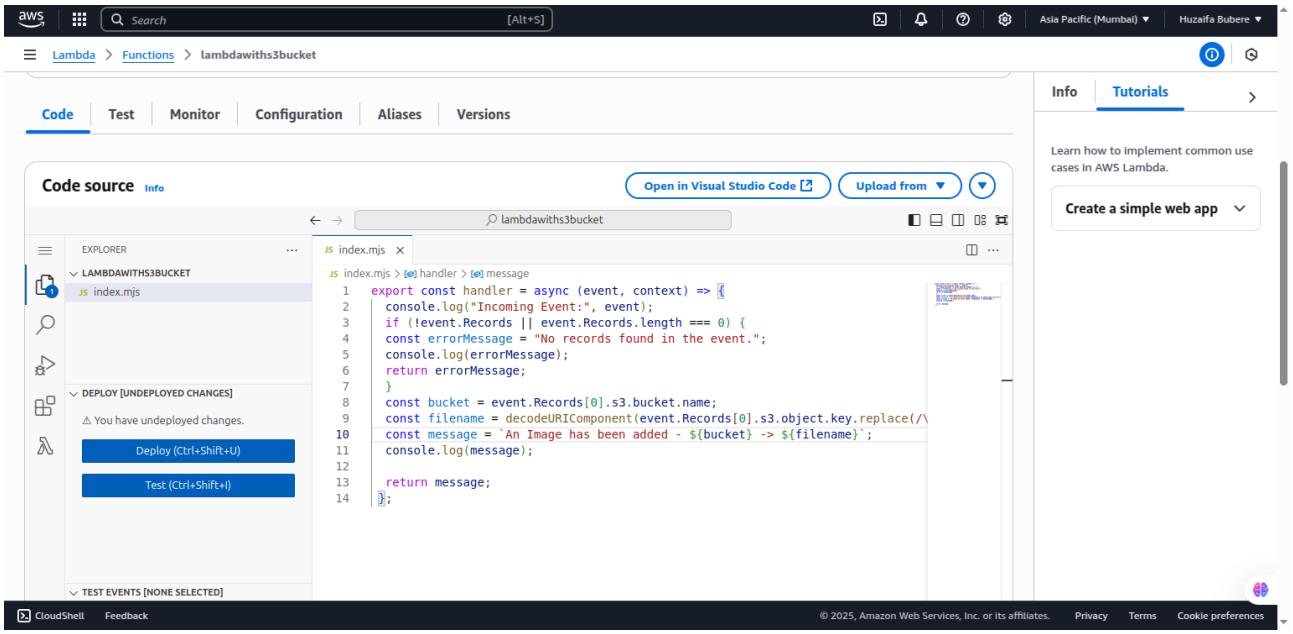
Let's add the details for the aws lambda function. Here, we will use the online editor to add our code and use nodejs as the runtime environment.

To trigger S3 with AWS Lambda, we will have to use S3 event in the code as shown below



Step 7:

let us save the changes and test the lambda function with S3upload.

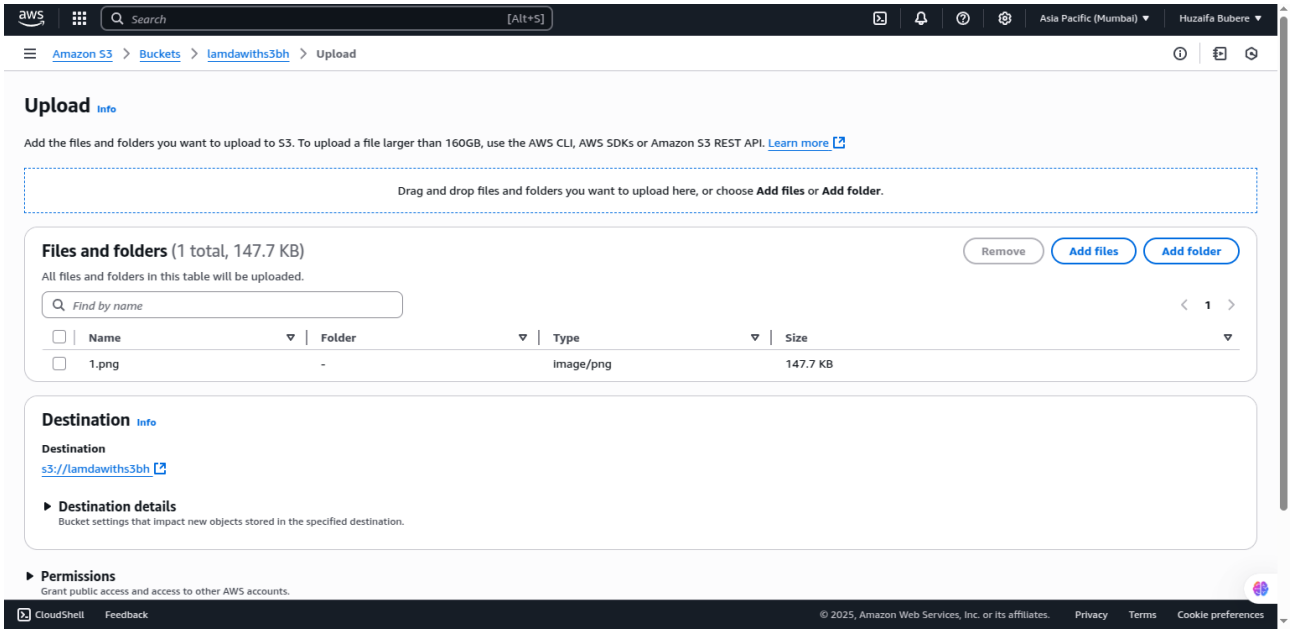


Step 8:

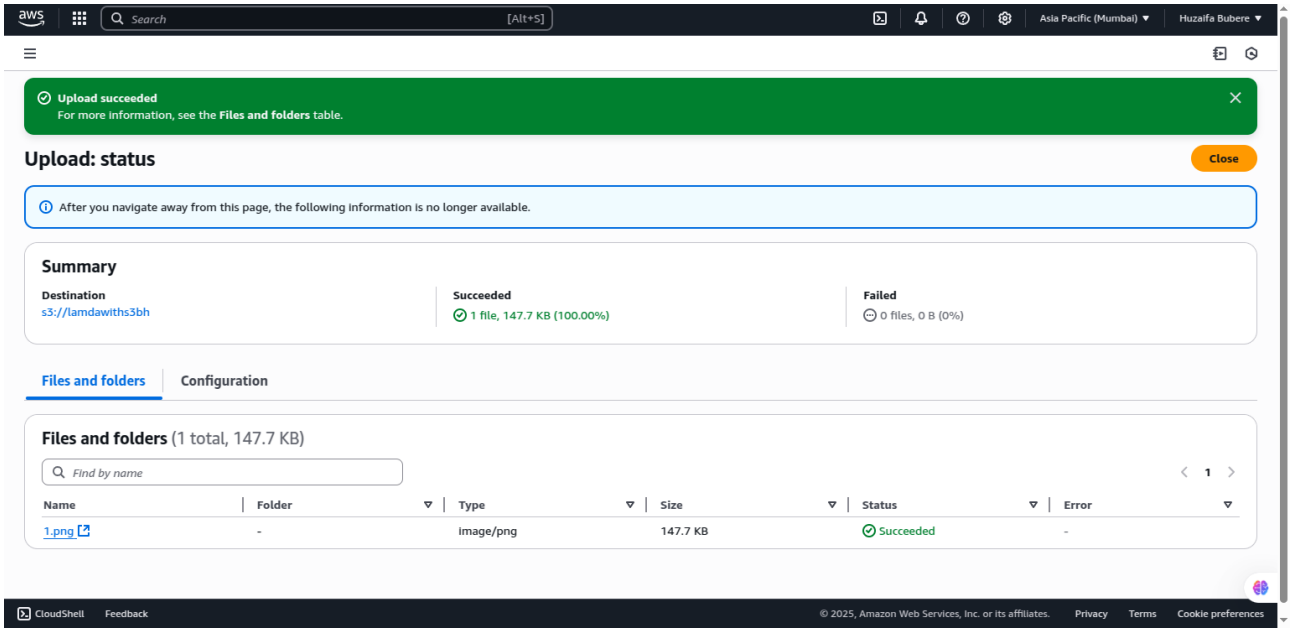
Now, save the Lambda function. Open S3 from Amazon services and open the bucket we created earlier namely lambdawiths3.

Upload the image in it as shown below –

Click **Add files** to add files. You can also drag and drop the files. Now, click **Upload** button.



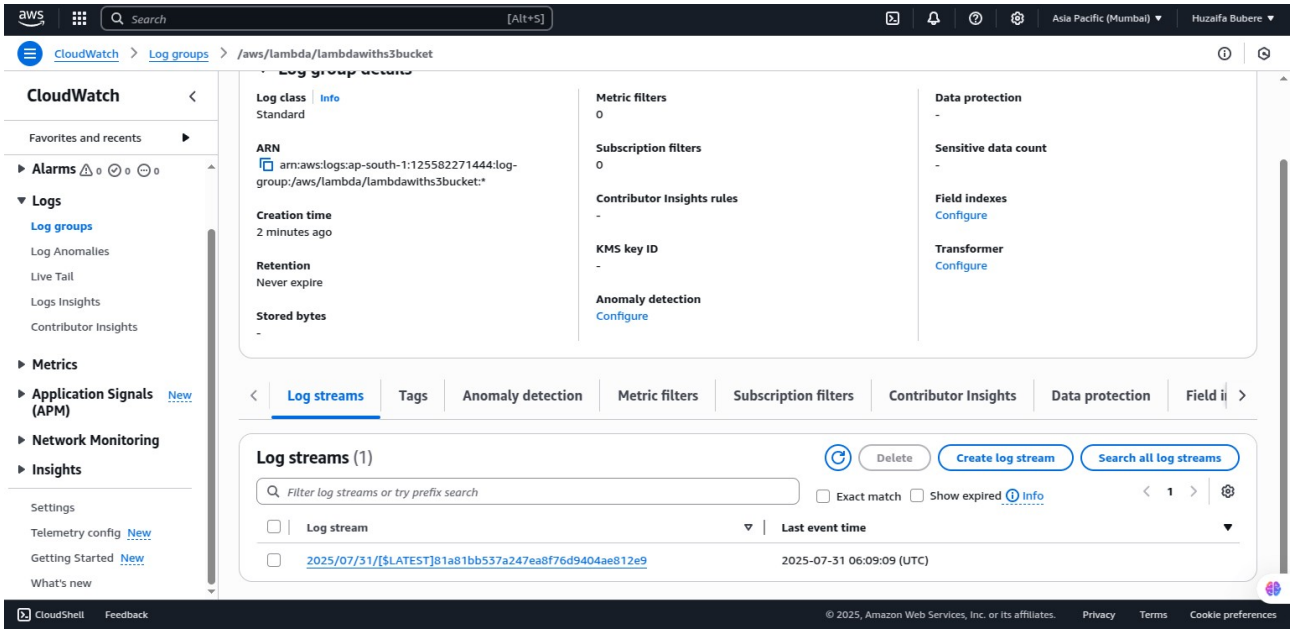
Thus, we have uploaded one image in our S3 bucket.

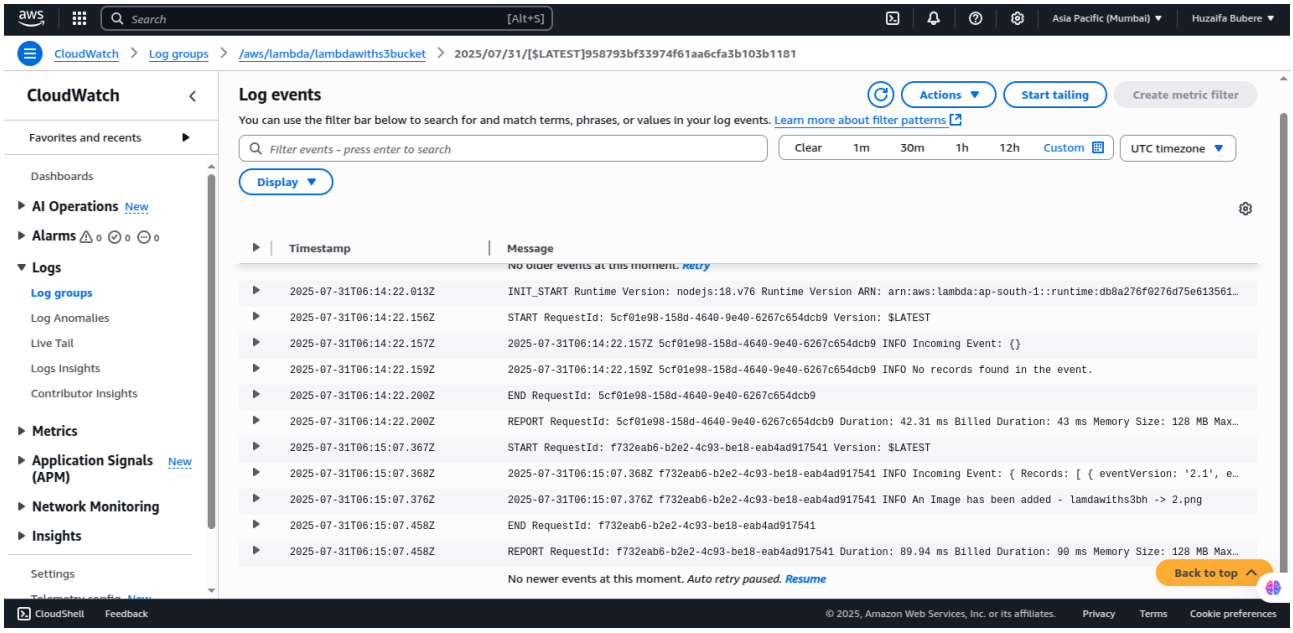


Step 9

To see the trigger details, go to AWS service and select CloudWatch. Open the logs for the Lambda

AWS Lambda function gets triggered when file is uploaded in S3 bucket and the details are logged in Cloudwatch as shown below –





An image has been Added -> 2.png you can see in cloudwatch logs.

Conclusion: Write your own findings.