PRACTICAL 01

Name : Sonam Chhabaidiya
class : D15A
Roll No:- 09

Date
Page

Adv. deVops
Experiment 01

Aim :- To understand the benefits of cloud infrastructure and setup AWS cloud 9 IDE, launch AWS cloud 9 IDE and perform collaboration.

Theory :- Amazon web services (AWS) is a comprehensive cloud computing platform provided by Amazon. It offers wide range of cloud-based services, including computer power, storage and databases, machine learning and more.

Amazon EC2 is a core service with AWS that provides scalable virtual servers known as EC2 instance. These instances are designed to handle various workloads from basic web application to high performance computing tasks. EC2 allows users to launch and manage virtual servers in the cloud providing flexibility in terms of computing power and operating system.

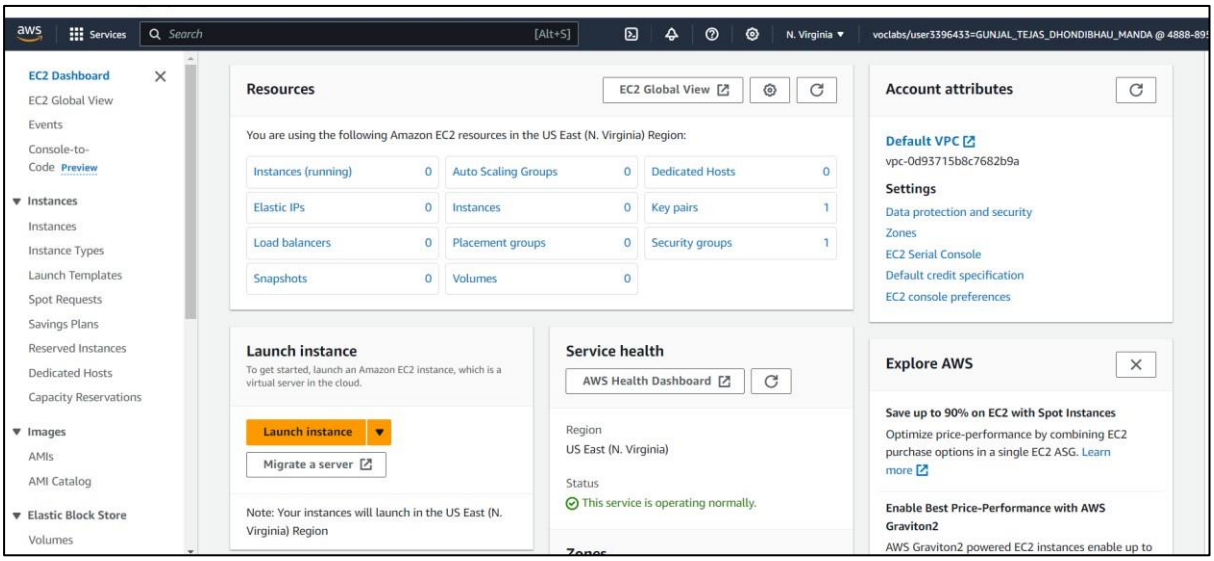
An EC2 instance is a virtual server with in the Amazon EC2 service. It represents a single unit of computing capacity in the AWS cloud.

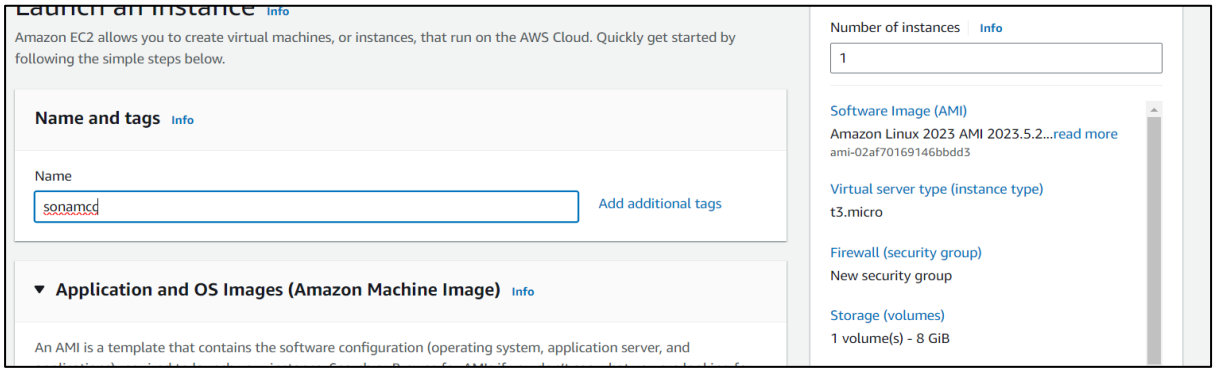
Conclusion :- Thus, we have understood the benefits of cloud infra and setup AWS cloud 9 IDE. Due to private access, the S3 website was not accessible.

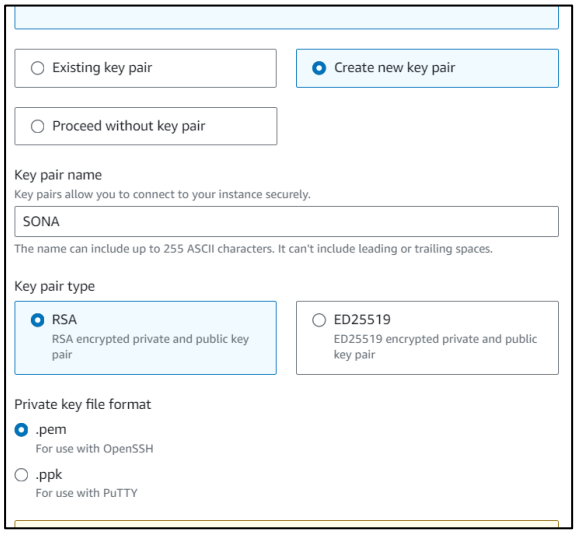
Teacher's Sign.: _____

SONAM CHHABAIDIAY D15A 09

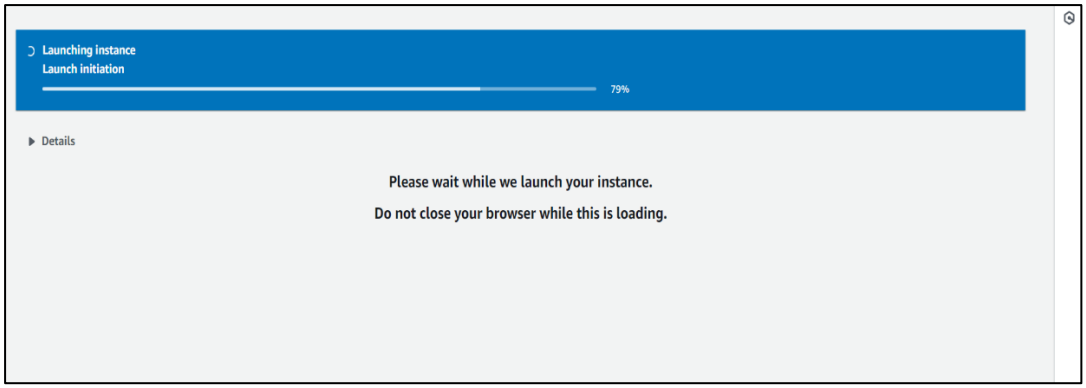
EC2 INSTANCES

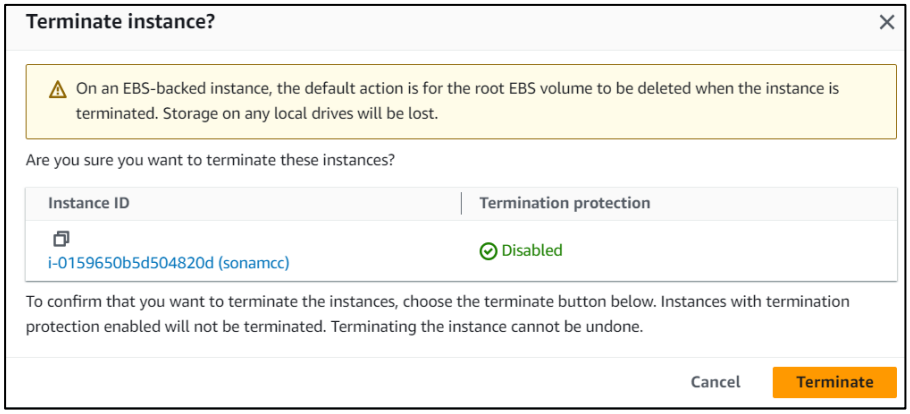


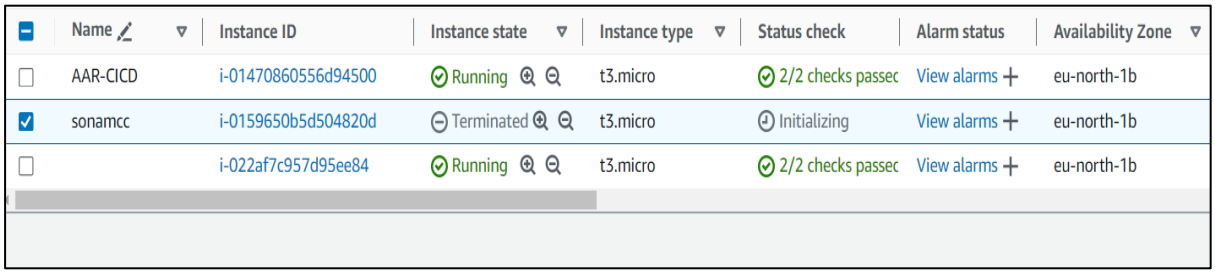




SONAM CHHABAIDIAY D15A 09



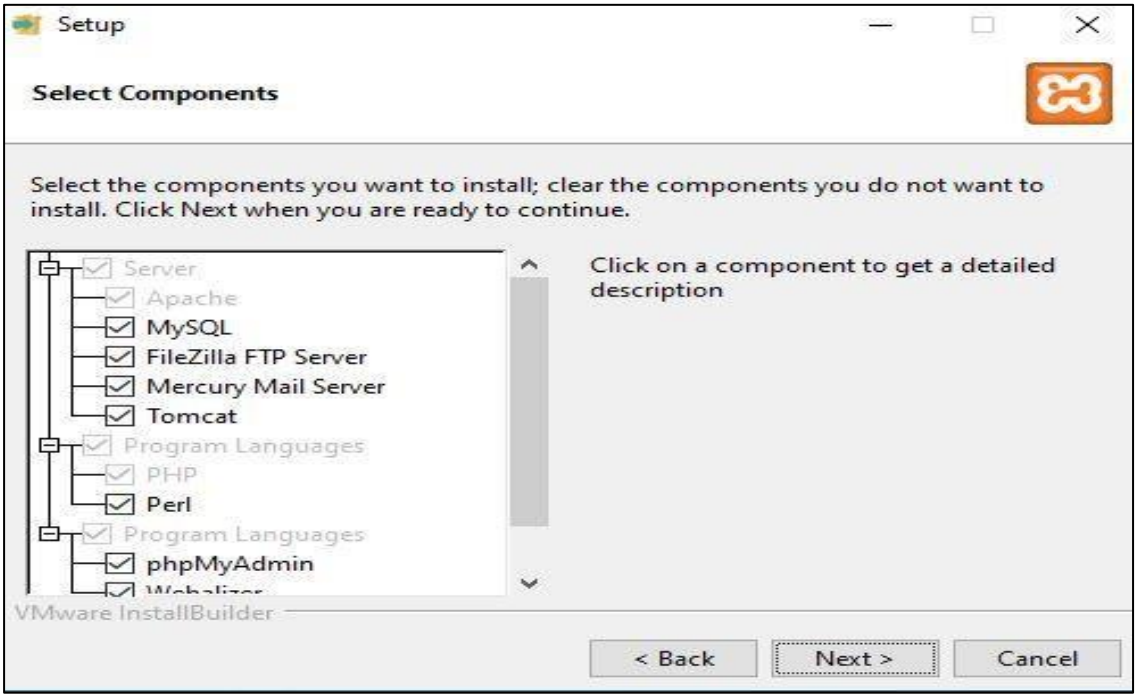


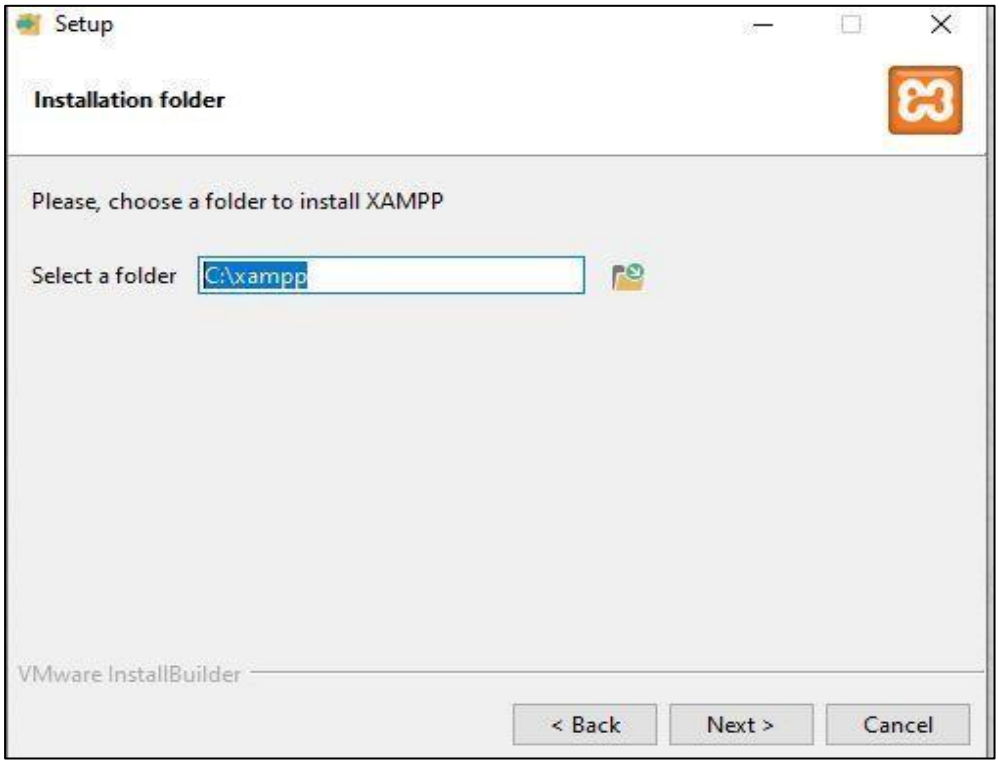


```
ubuntu@ip-172-31-5-93:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 39 not upgraded.
Need to get 1680 kB/2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Fetched 1680 kB in 0s (17.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1t64:amd64.
(Reading database ... 67739 files and directories currently installed.)
Preparing to unpack .../0-libapr1t64_1.7.2-3.1build2_amd64.deb ...
Unpacking libapr1t64:amd64 (1.7.2-3.1build2) ...
Selecting previously unselected package libaprutil1t64:amd64.
Preparing to unpack .../1-libaprutil1t64_1.6.3-1.1ubuntu7_amd64.deb ...
```

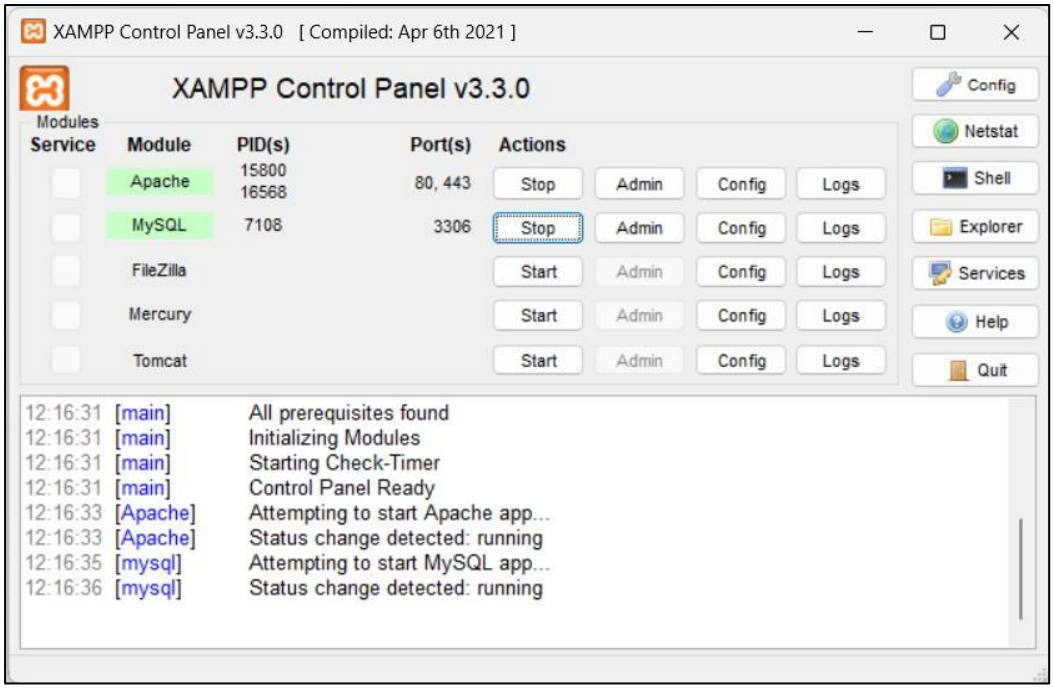
Part 1: a) To develop a website and host it on your local machine on a VM

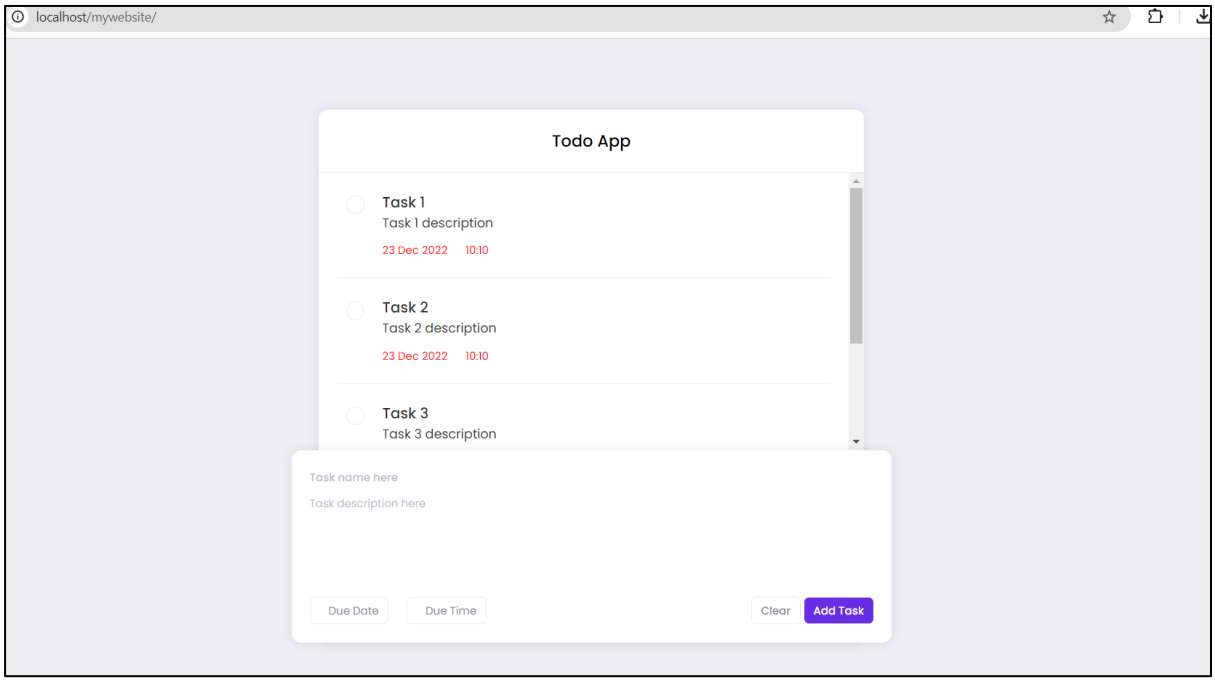
- Steps to host Website using Xampp Server


Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Server
Apache
MySQL
FileZilla FTP Server
Mercury Mail Server
Tomcat
Program Languages
PHP
Perl
Program Languages
phpMyAdmin
Click on a component to get a detailed description
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Installation folder
Please, choose a folder to install XAMPP
Select a folder
C:\xampp
VMware InstallBuilder
< Back
Next >
Cancel

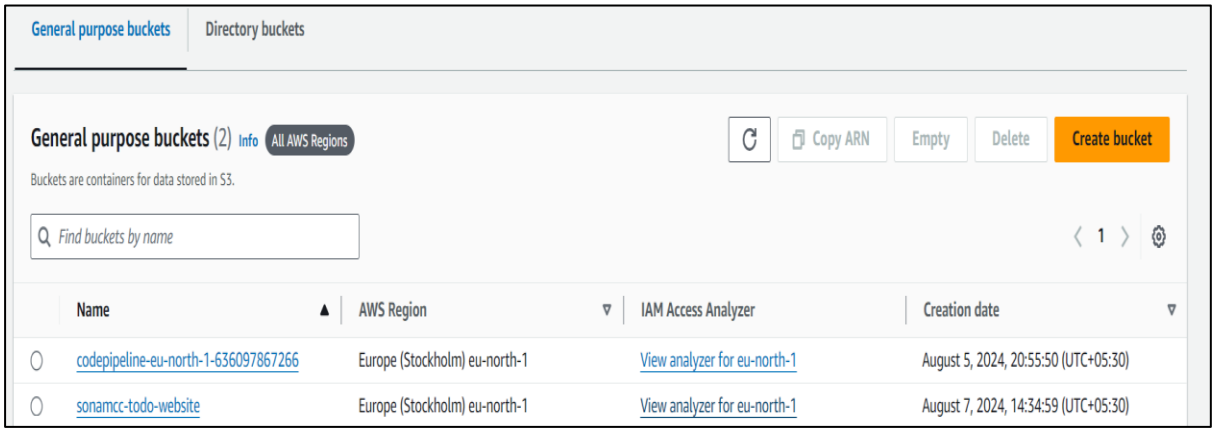
SONAM CHHABAIDIAY D15A 09

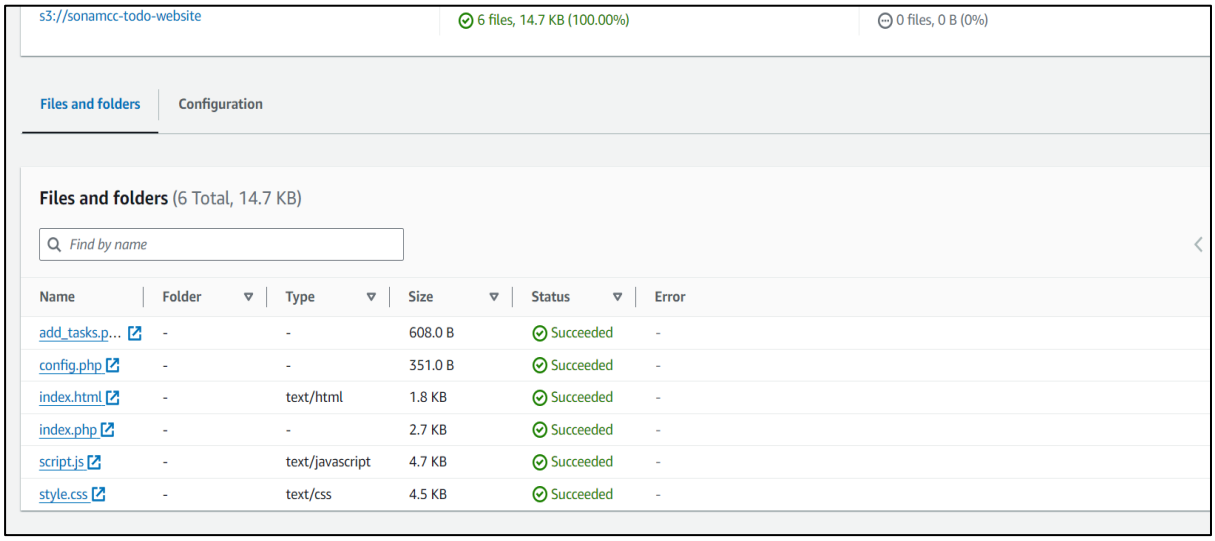




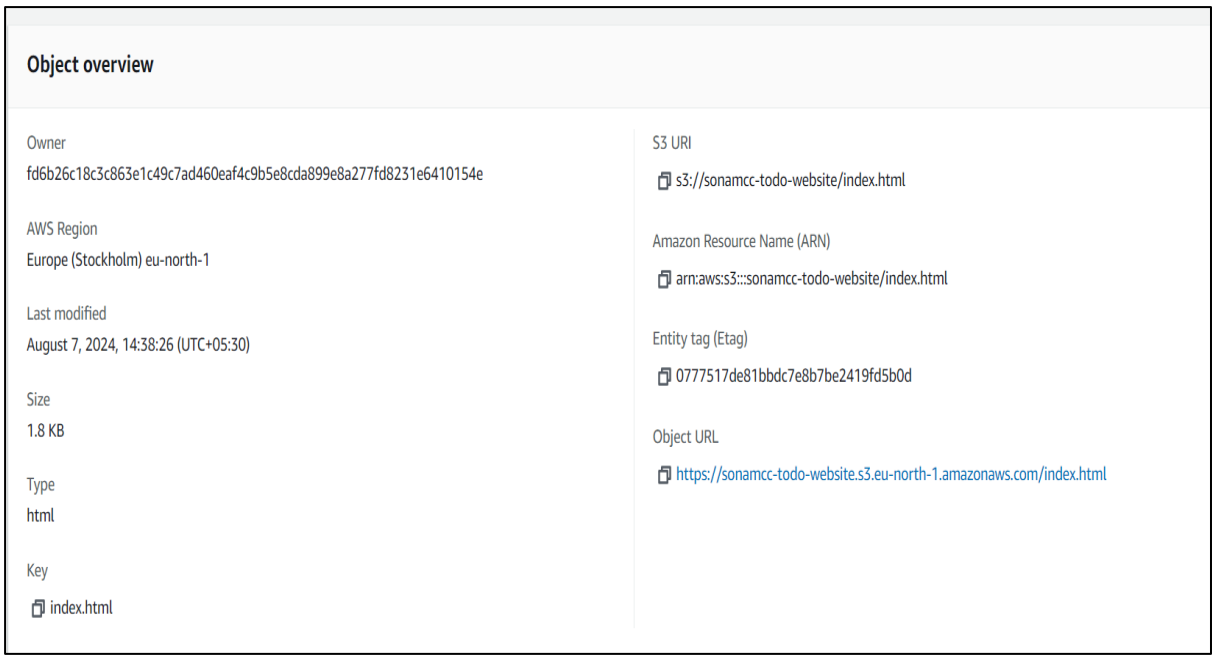
Part 1: b) Hosting a static website on Amazon S3

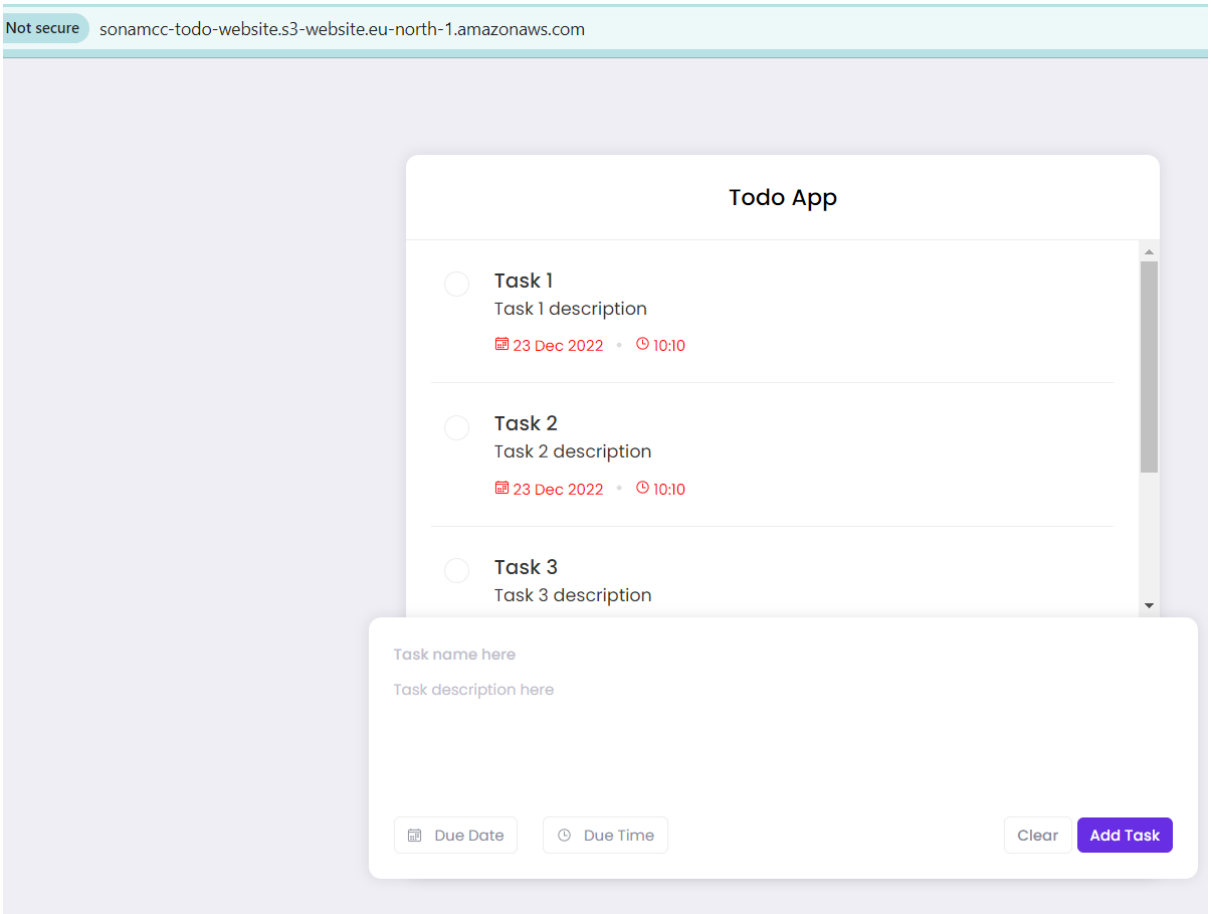
- Steps to host Website using Amazon S3





SONAM CHHABAIDIAY D15A 09





Roll NO: 09

Date
Page

Adv. DevOps.

Experiment : 02

Aim :- To Build a Application using AWS code Build and Deploy on S3/SEBS using AWS code Pipeline, Deploy Sample Application on EC2 instance using AWS codeDeploy.

Theory :-

AWS CodePipeline

- AWS codepipeline is a fully managed continuous Integration & continuous delivery service that automates the build, test & deploy phases of your-release process,
- It allows you to define a sequence of actions, such as source retrieval, building the code, running tests & deploying the application.
- It integrates with other AWS services, making it a versatile tool for automating application deployments.

AWS codeBuild

- AWS codeBuild is a fully managed build service in the cloud.
- It compiles your source code, runs unit tests & produces artifacts that are ready to deploy.
- codeBuild scales continuously and process multiple builds concurrently, so your builds are never left waiting in a queue.

Teacher's Sign.: _____

Date
Page

AWS CodeDeploy

- AWS codeDeploy automates application deployments, to various compute services such as EC2, Lambda & servers.
- It handles the complexity of updating your applications across multiple instances, with zero downtime.
- Deployment types.

[i] In-place Deployment :- Update the application on each instance in place.

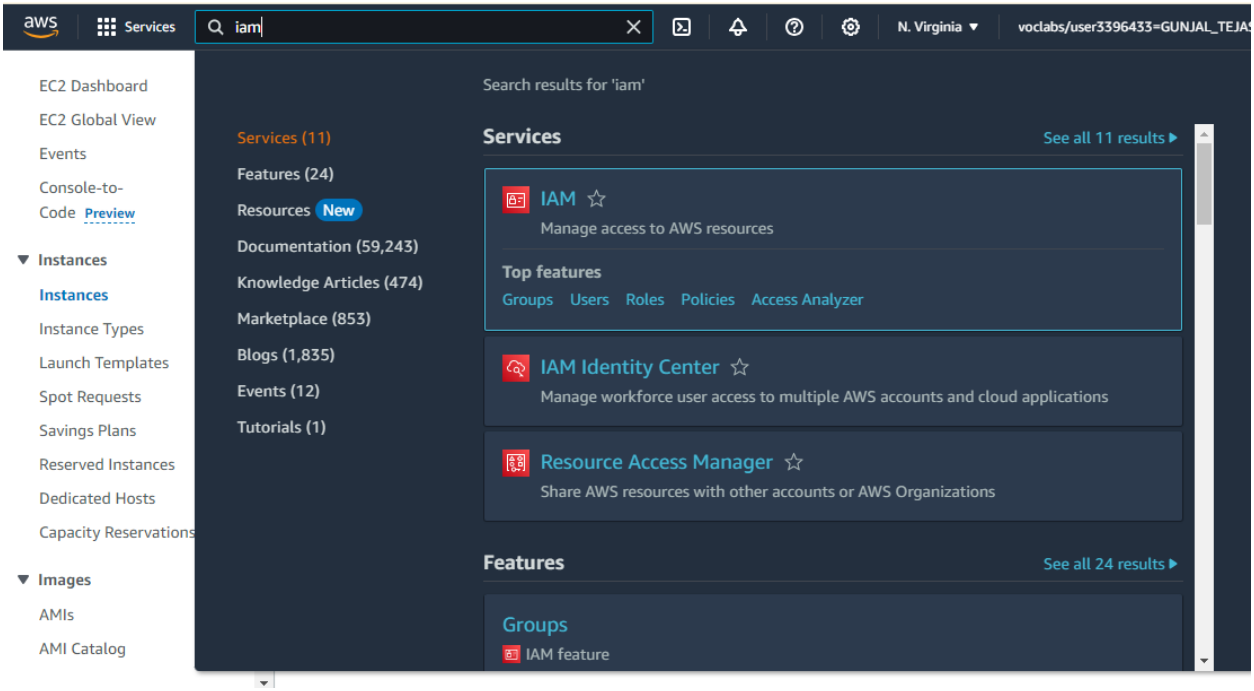
[ii] Blue/green Deployment :- Deploys the application on new instances, allowing to test new version before routing traffic to it.

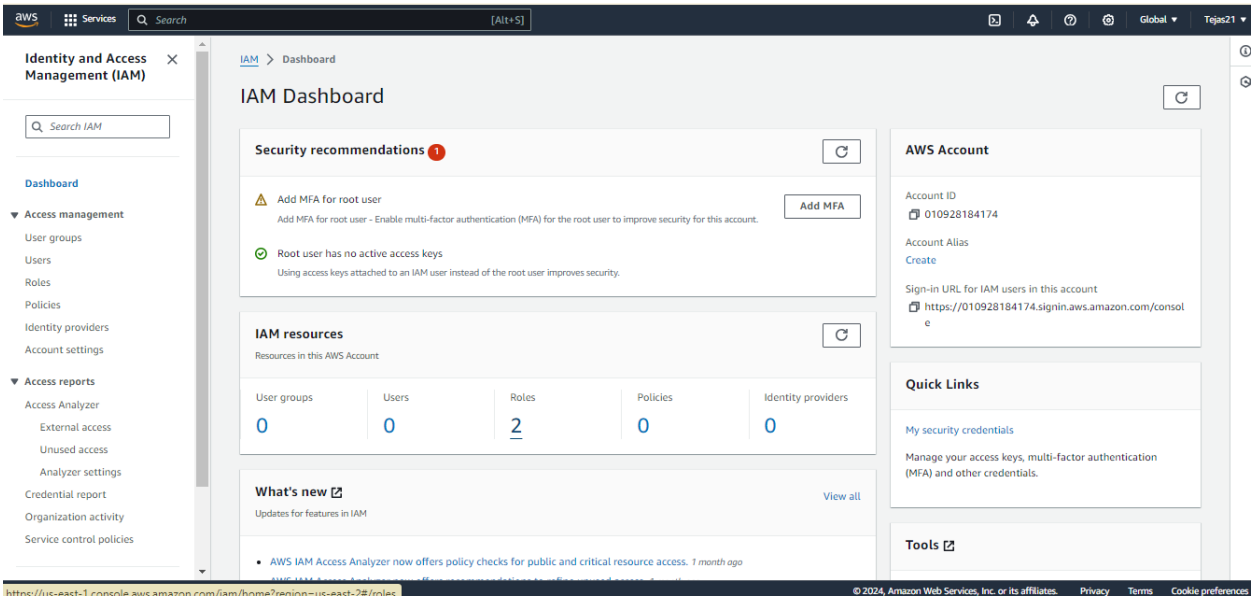
Conclusion :-

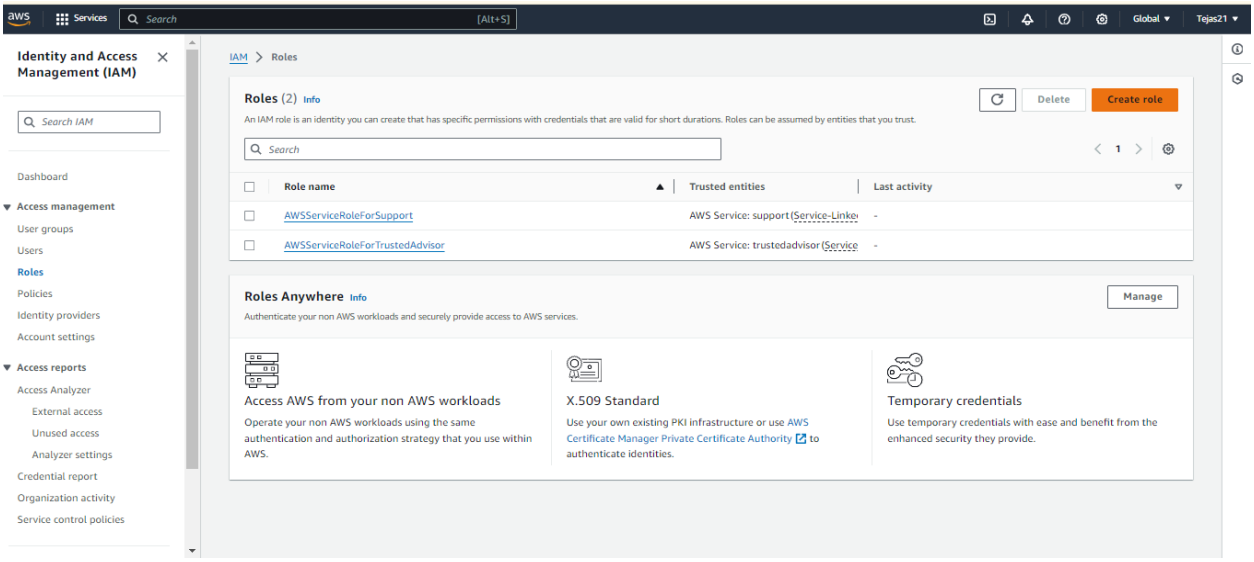
This experiment demonstrates the integration of various AWS service to build a robust CI/CD pipeline. By automating the build, test, & deployment process using AWS codeBuild, CodePipeline & codeDeploy, we can achieve faster release, reduce errors & improve the overall quality of applications.

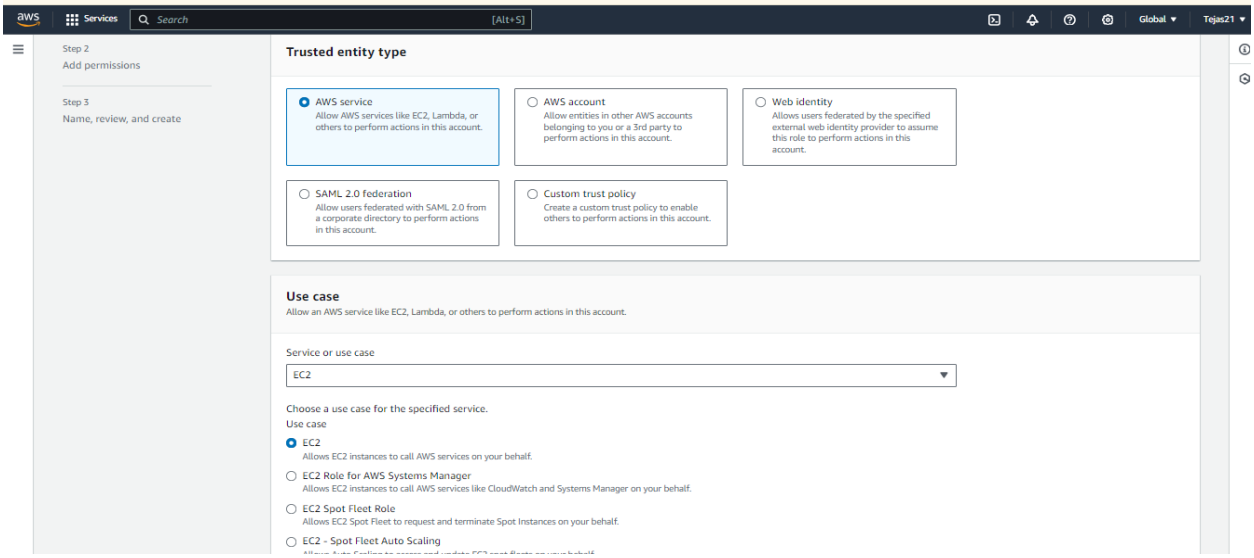
Teacher's Sign.: _____

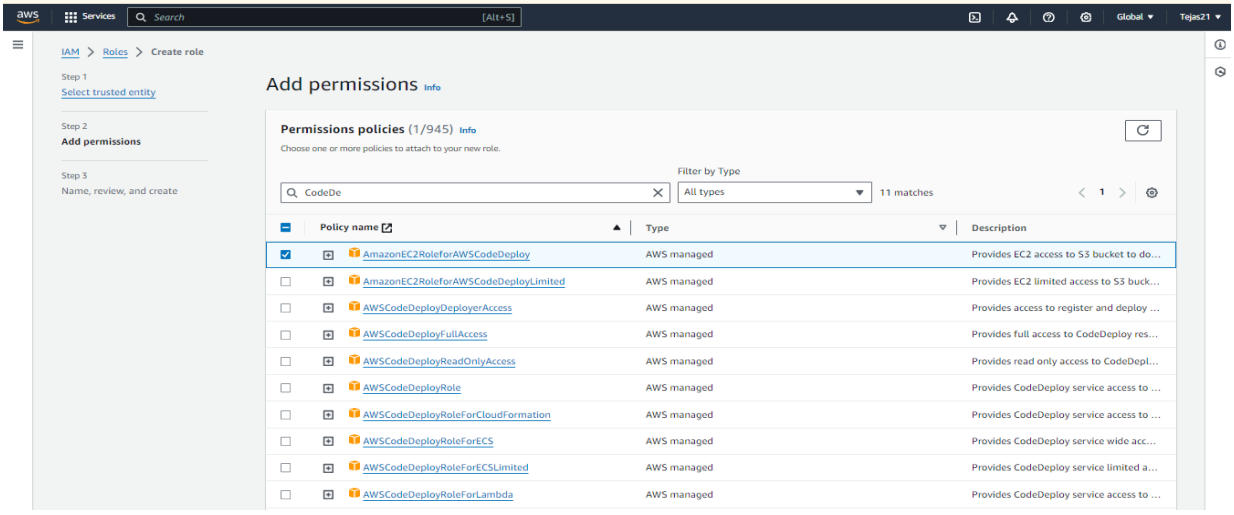
➢ Creating two roles in AWS IAM for CI/CD pipeline setup

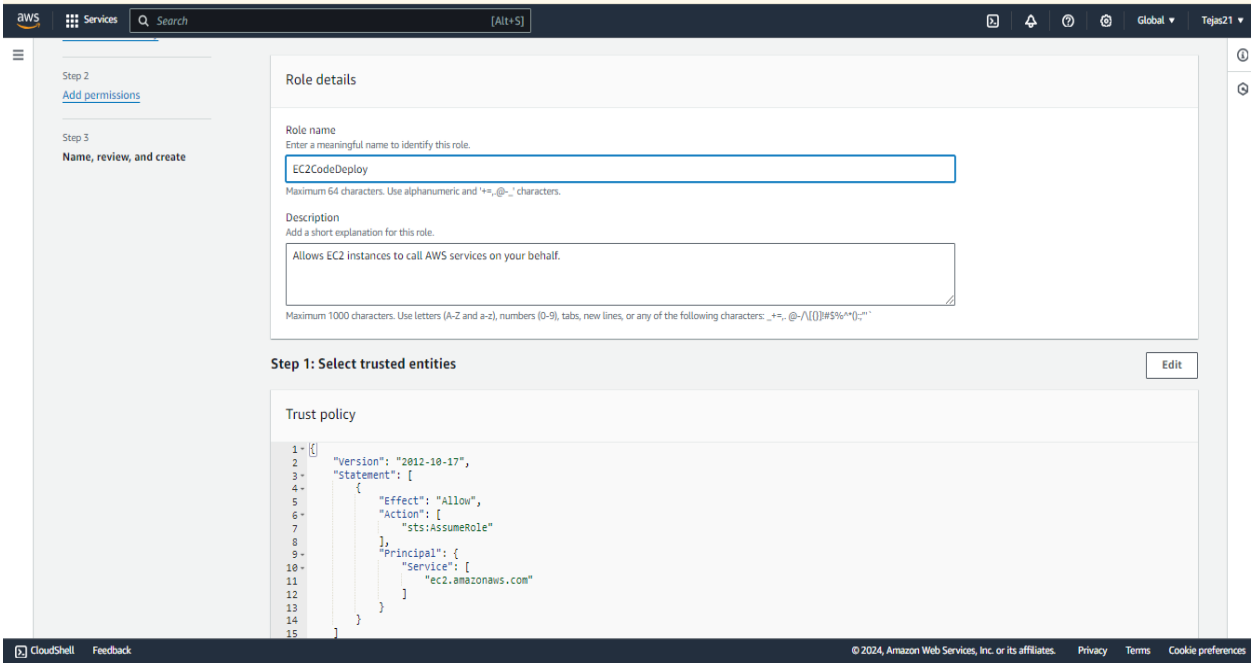


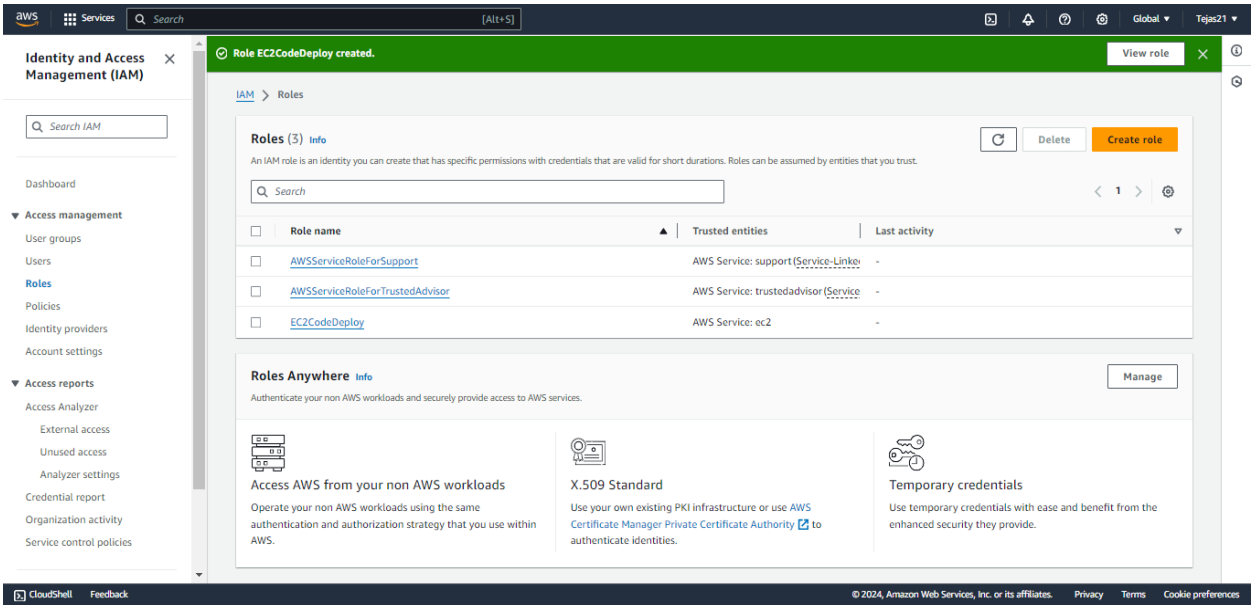


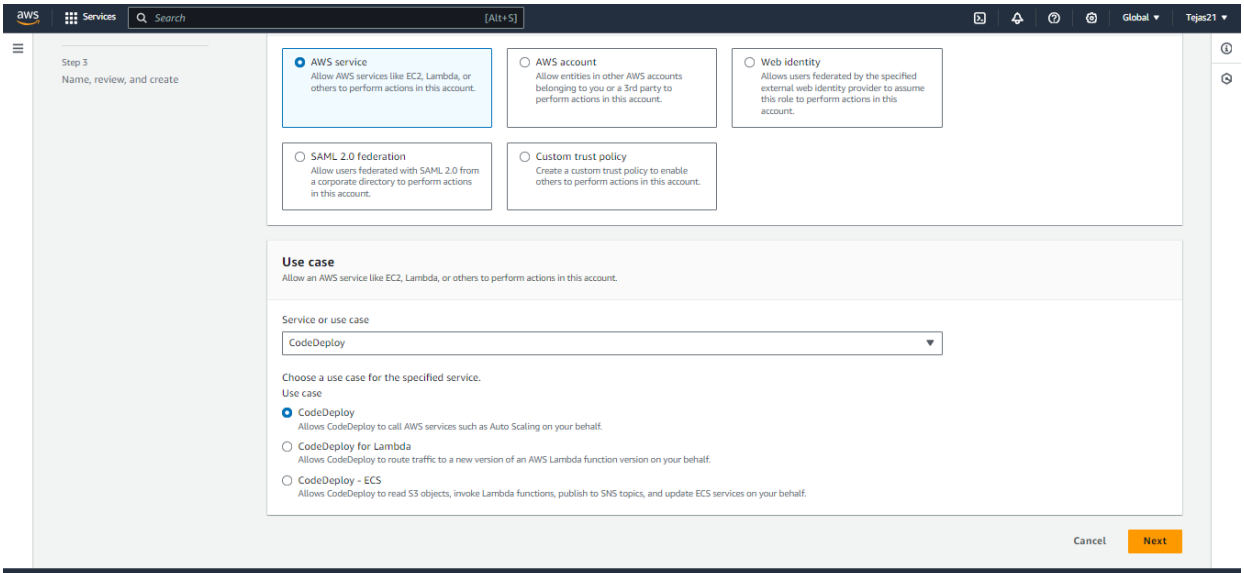

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
IAM > Roles
Roles (2) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
Role name
Trusted entities
Last activity
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
Roles Anywhere Info
Authenticate your non AWS workloads and securely provide access to AWS services.
Manage
Access AWS from your non AWS workloads
Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.
X.509 Standard
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.
Temporary credentials
Use temporary credentials with ease and benefit from the enhanced security they provide.

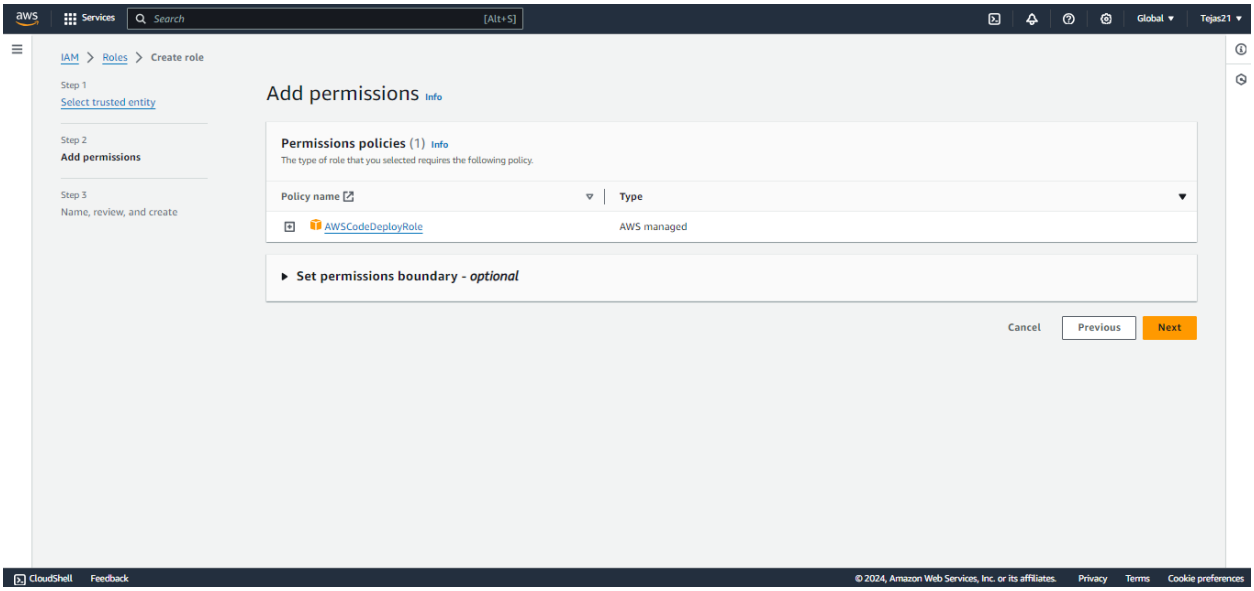

Step 2
Add permissions
Step 3
Name, review, and create
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
EC2
Choose a use case for the specified service.
Use case
EC2
Allows EC2 instances to call AWS services on your behalf.
EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
EC2 - Spot Fleet Auto Scaling

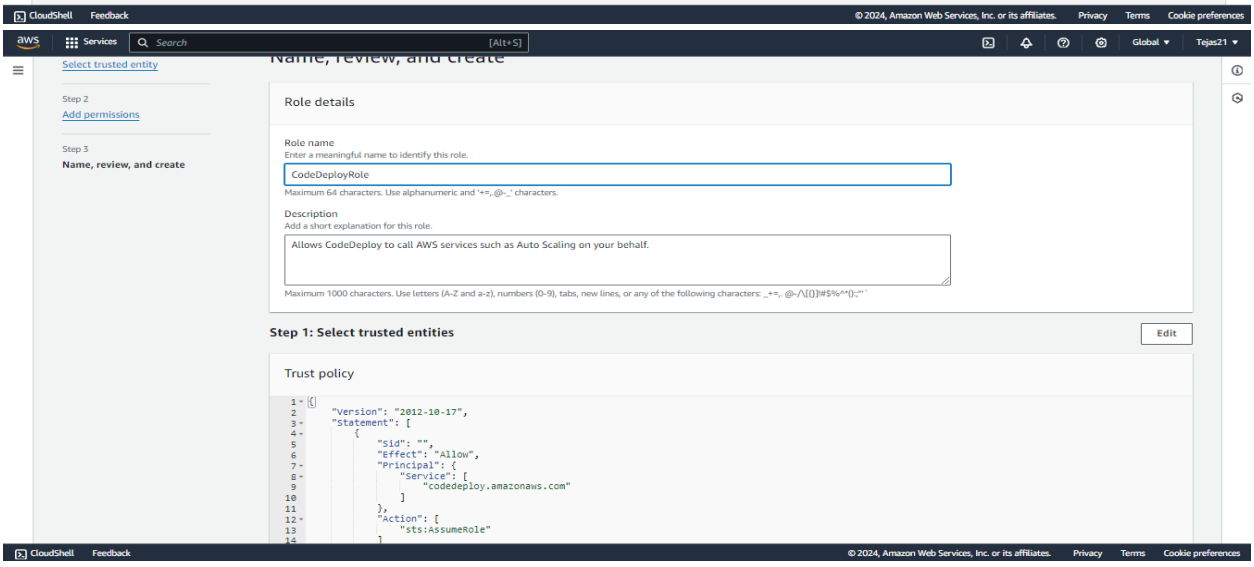

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (1/945) Info
Choose one or more policies to attach to your new role.
Filter by Type
CodeDe
All types
11 matches
Policy name
Type
Description
AmazonEC2RoleforAWSCodeDeploy
AWS managed
Provides EC2 access to S3 bucket to do...
AmazonEC2RoleforAWSCodeDeployLimited
AWS managed
Provides EC2 limited access to S3 buck...
AWSCodeDeployDeployerAccess
AWS managed
Provides access to register and deploy ...
AWSCodeDeployFullAccess
AWS managed
Provides full access to CodeDeploy res...
AWSCodeDeployReadOnlyAccess
AWS managed
Provides read only access to CodeDepl...
AWSCodeDeployRole
AWS managed
Provides CodeDeploy service access to ...
AWSCodeDeployRoleForCloudFormation
AWS managed
Provides CodeDeploy service access to ...
AWSCodeDeployRoleForECS
AWS managed
Provides CodeDeploy service wide acc...
AWSCodeDeployRoleForECSLimited
AWS managed
Provides CodeDeploy service limited a...
AWSCodeDeployRoleForLambda
AWS managed
Provides CodeDeploy service access to ...


Step 2
Add permissions
Step 3
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
EC2CodeDeploy
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=, .@-/\[{}]!#$%^*();:"'`
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"sts:AssumeRole"
],
"Principal": {
"Service": [
"ec2.amazonaws.com"
]
}
}
]
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

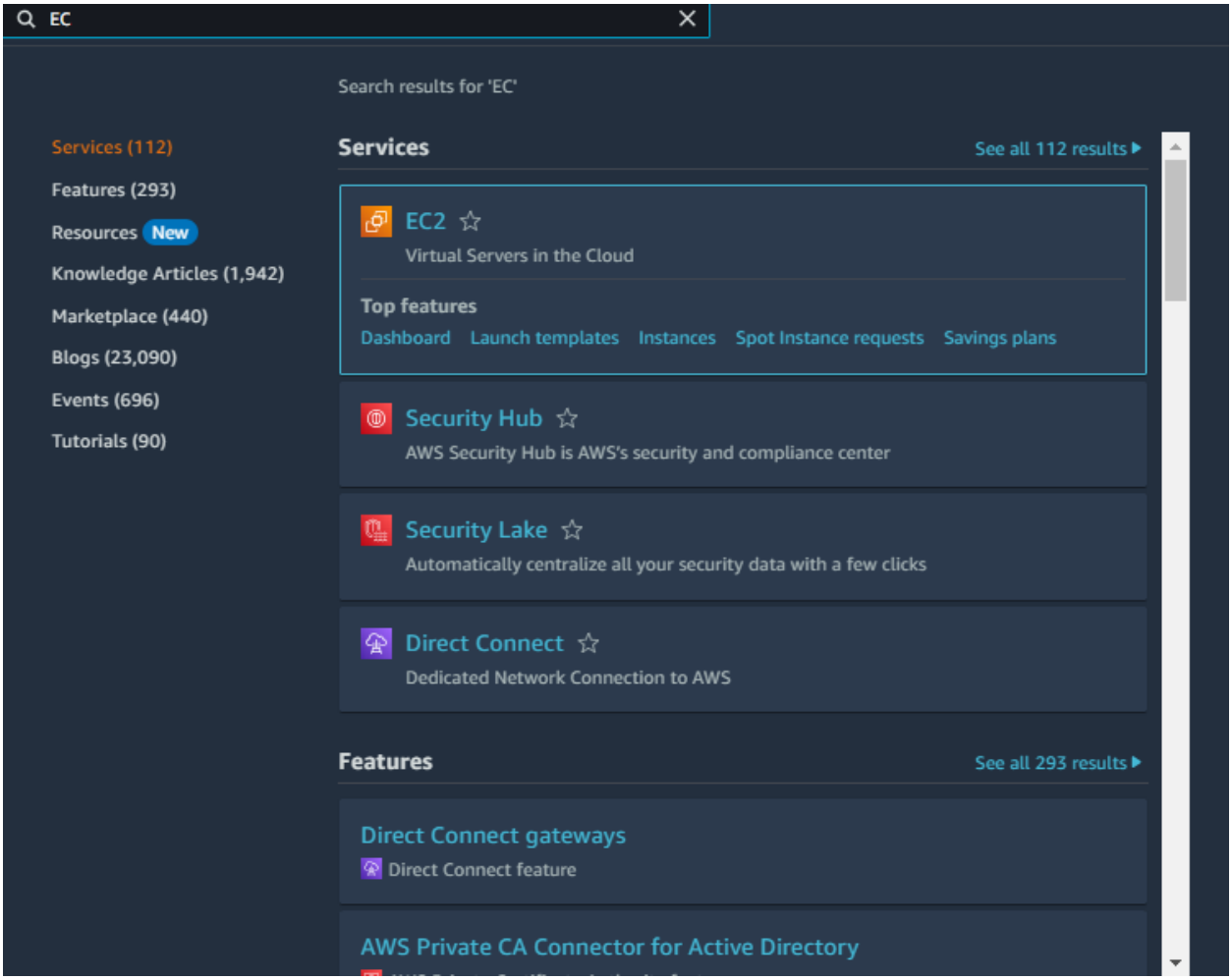

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Role EC2CodeDeploy created.
View role
IAM > Roles
Roles (3) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
Role name
Trusted entities
Last activity
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
EC2CodeDeploy
AWS Service: ec2
Roles Anywhere Info
Manage
Authenticate your non AWS workloads and securely provide access to AWS services.
Access AWS from your non AWS workloads
Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.
X.509 Standard
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.
Temporary credentials
Use temporary credentials with ease and benefit from the enhanced security they provide.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

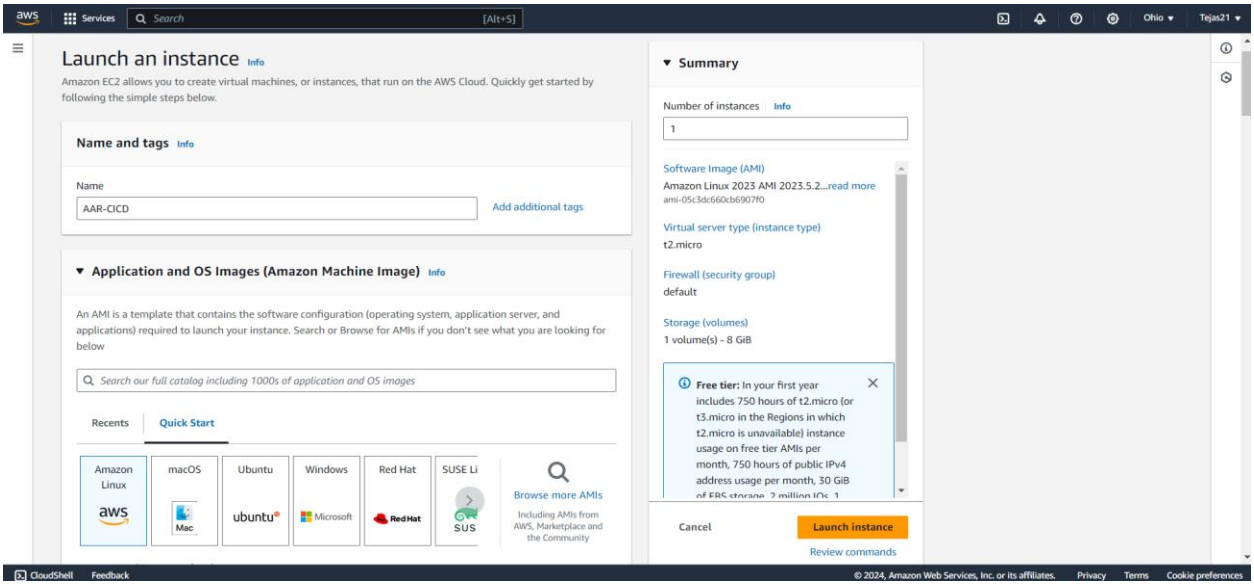

Step 3
Name, review, and create
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
CodeDeploy
Choose a use case for the specified service.
Use case
CodeDeploy
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
CodeDeploy for Lambda
Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.
CodeDeploy - ECS
Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.
Cancel
Next

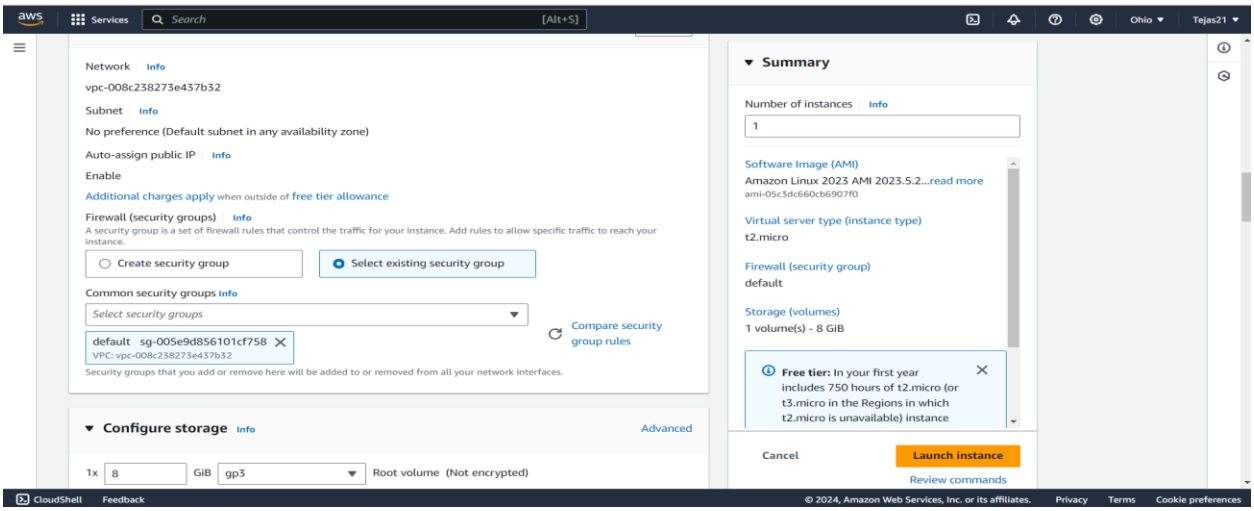

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (1) Info
The type of role that you selected requires the following policy.
Policy name
Type
AWSCodeDeployRole
AWS managed
Set permissions boundary - optional
Cancel
Previous
Next

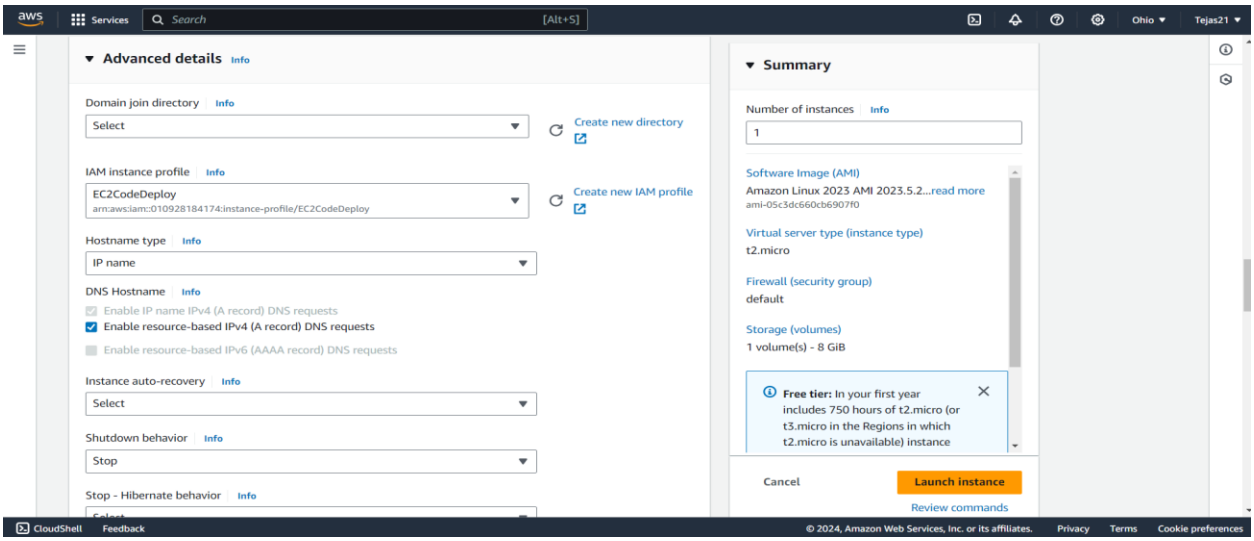

Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
CodeDeployRole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "",
"Effect": "Allow",
"Principal": {
"Service": [
"codedeploy.amazonaws.com"
]
},
"Action": [
"sts:AssumeRole"

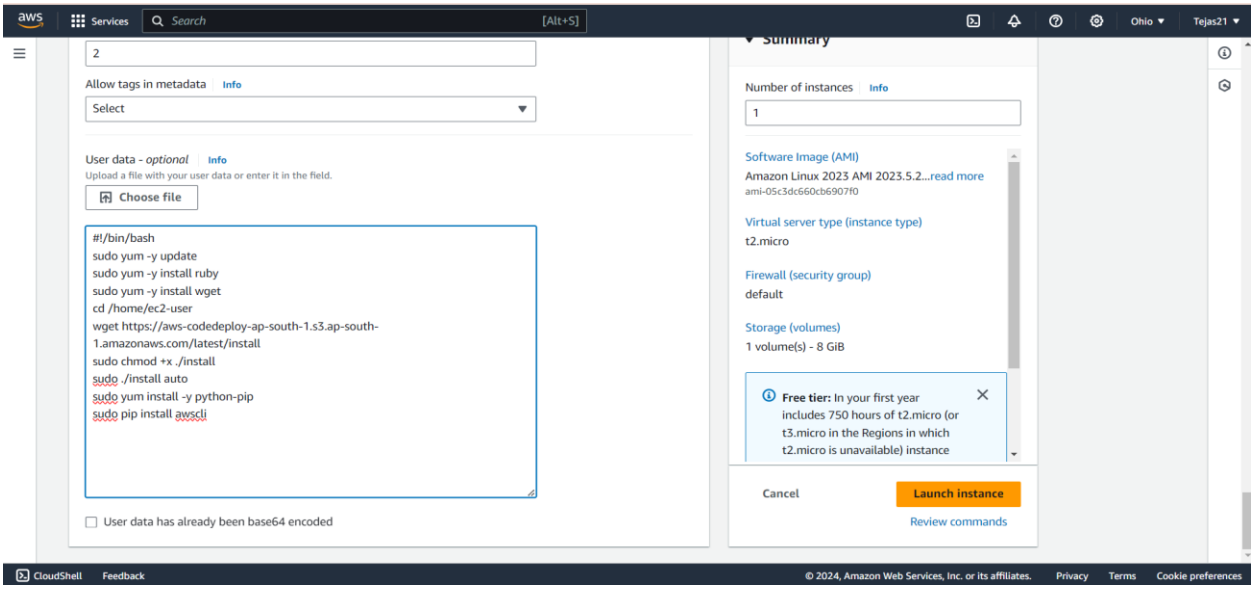
- Launching & Setting up EC2 Instance for Deployment

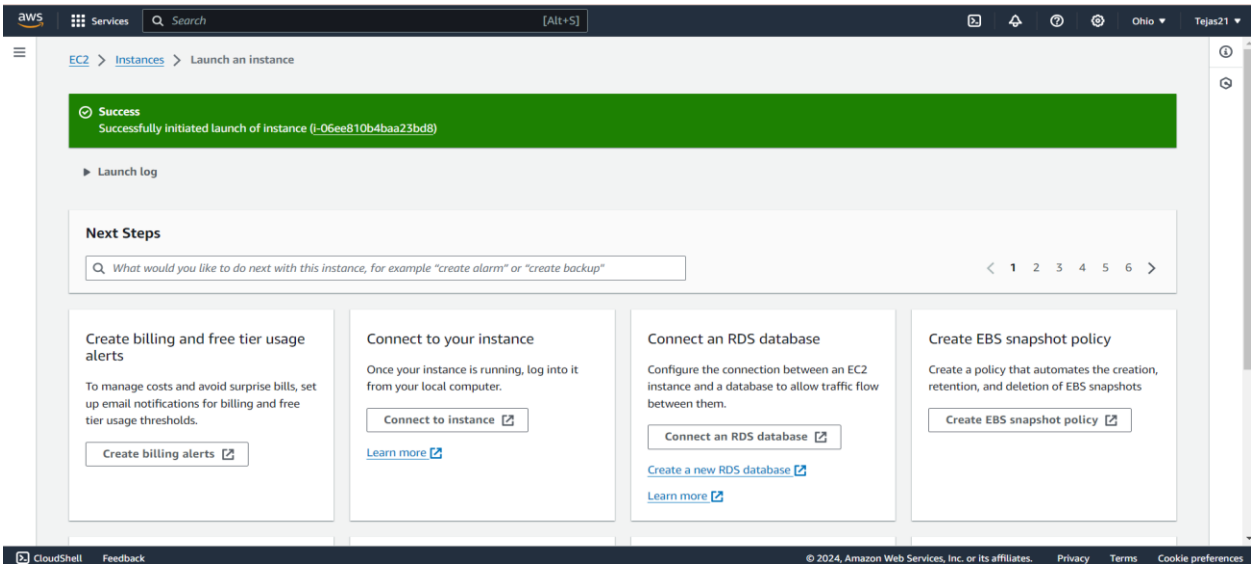


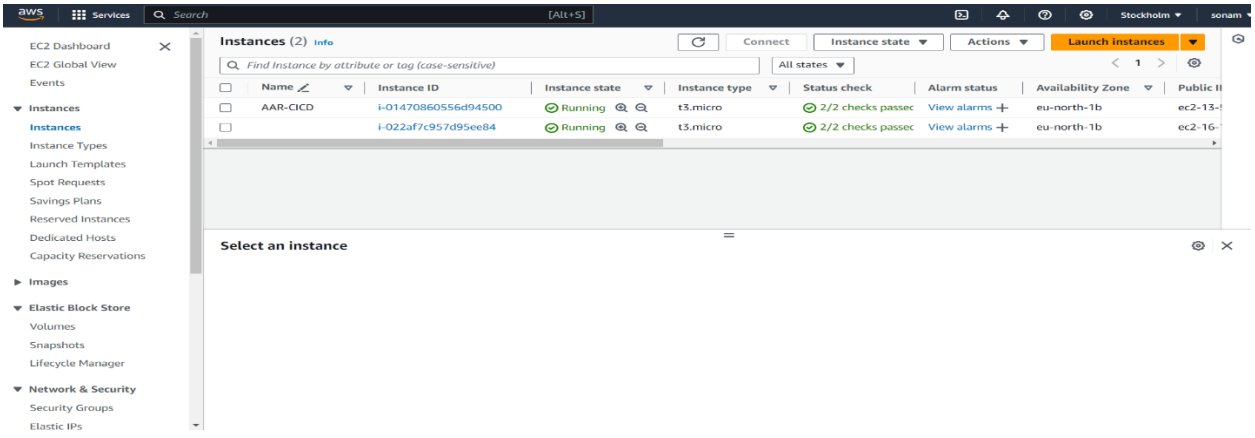


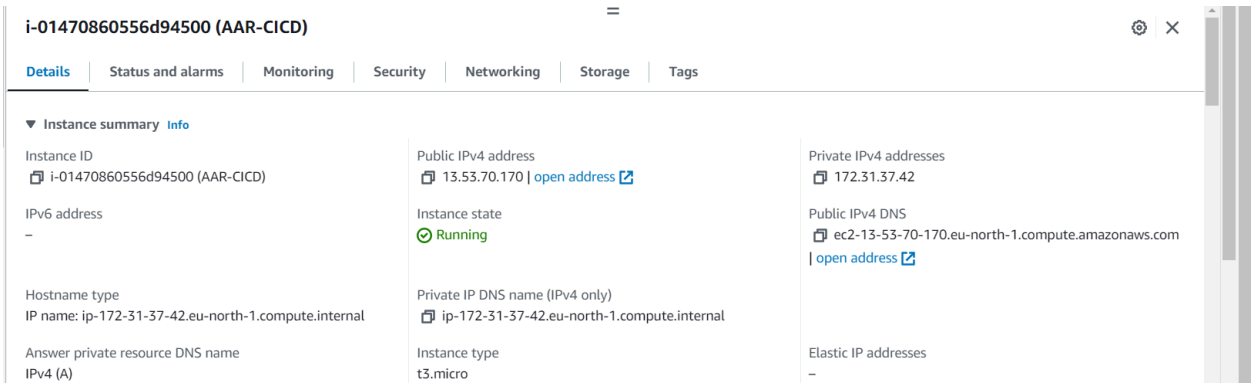

Network Info
vpc-008c238273e437b32
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Common security groups Info
Select security groups
default sg-005e9d856101cf758
VPC: vpc-008c238273e437b32
Security groups that you add or remove here will be added to or removed from all your network interfaces.
Compare security group rules
Configure storage Info
Advanced
1x 8 GiB gp3 Root volume (Not encrypted)
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-05c3dc660cb6907f0
Virtual server type (instance type)
t2.micro
Firewall (security group)
default
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Advanced details Info
Domain join directory Info
Select
Create new directory
IAM instance profile Info
EC2CodeDeploy
arn:aws:iam::010928184174:instance-profile/EC2CodeDeploy
Create new IAM profile
Hostname type Info
IP name
DNS Hostname Info
Enable IP name IPv4 (A record) DNS requests
Enable resource-based IPv4 (A record) DNS requests
Enable resource-based IPv6 (AAAA record) DNS requests
Instance auto-recovery Info
Select
Shutdown behavior Info
Stop
Stop - Hibernate behavior Info
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-05c3dc660cb6907f0
Virtual server type (instance type)
t2.micro
Firewall (security group)
default
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

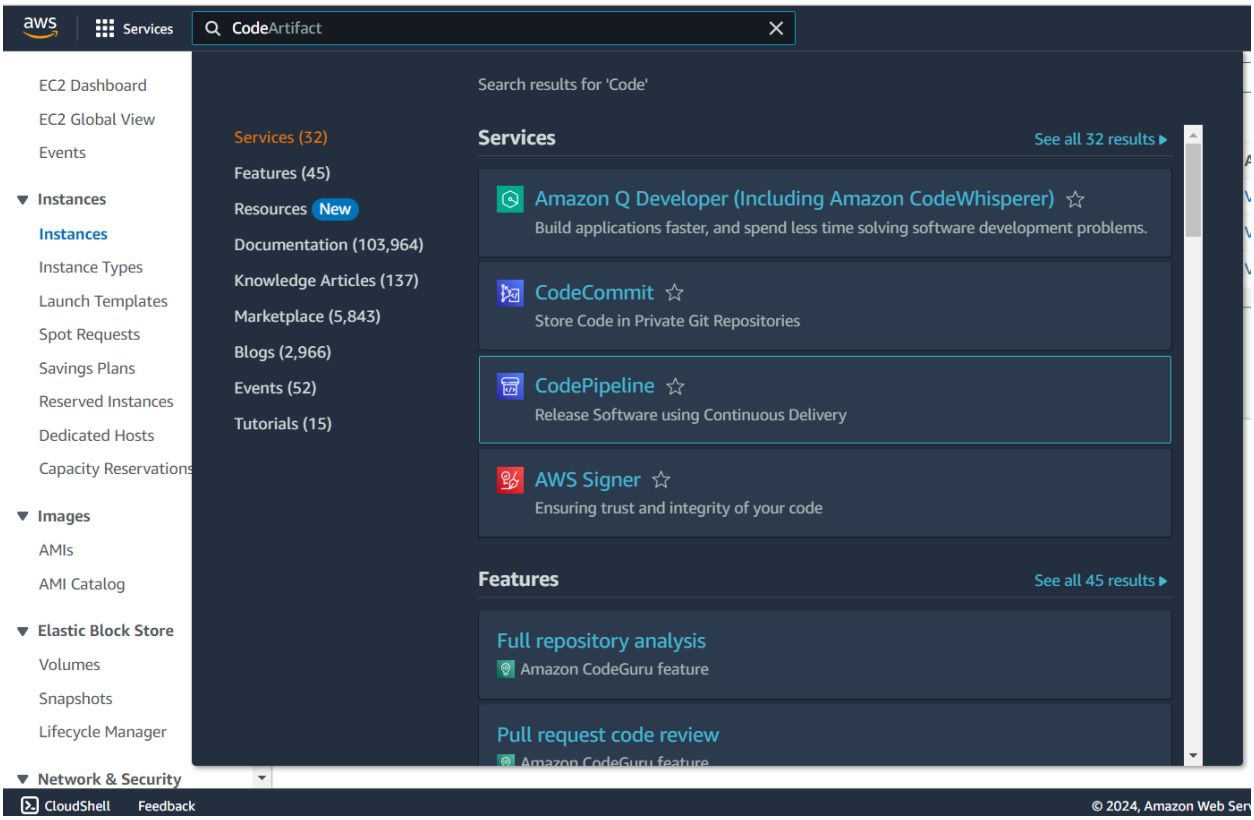

2
Allow tags in metadata Info
Select
User data - optional Info
Upload a file with your user data or enter it in the field.
Choose file
#!/bin/bash
sudo yum -y update
sudo yum -y install ruby
sudo yum -y install wget
cd /home/ec2-user
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
sudo yum install -y python-pip
sudo pip install awscli
User data has already been base64 encoded
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-05c3dc660cb6907f0
Virtual server type (instance type)
t2.micro
Firewall (security group)
default
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

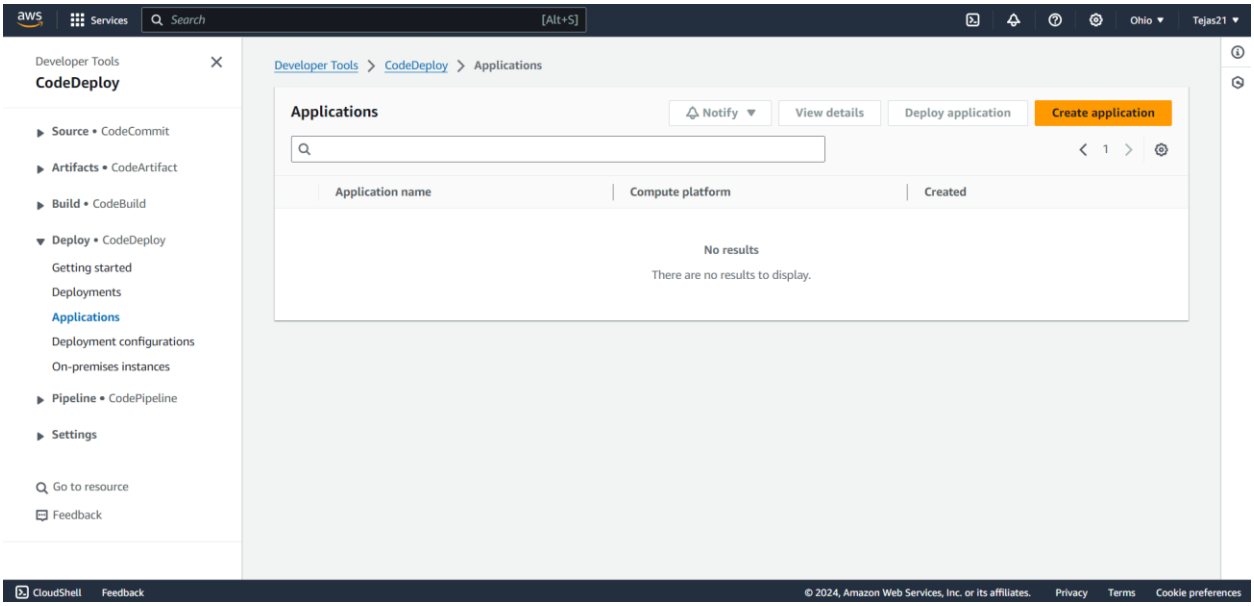

Services
Search
[Alt+S]
Ohio
Tejas21
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-06ee810b4baa23bd8)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

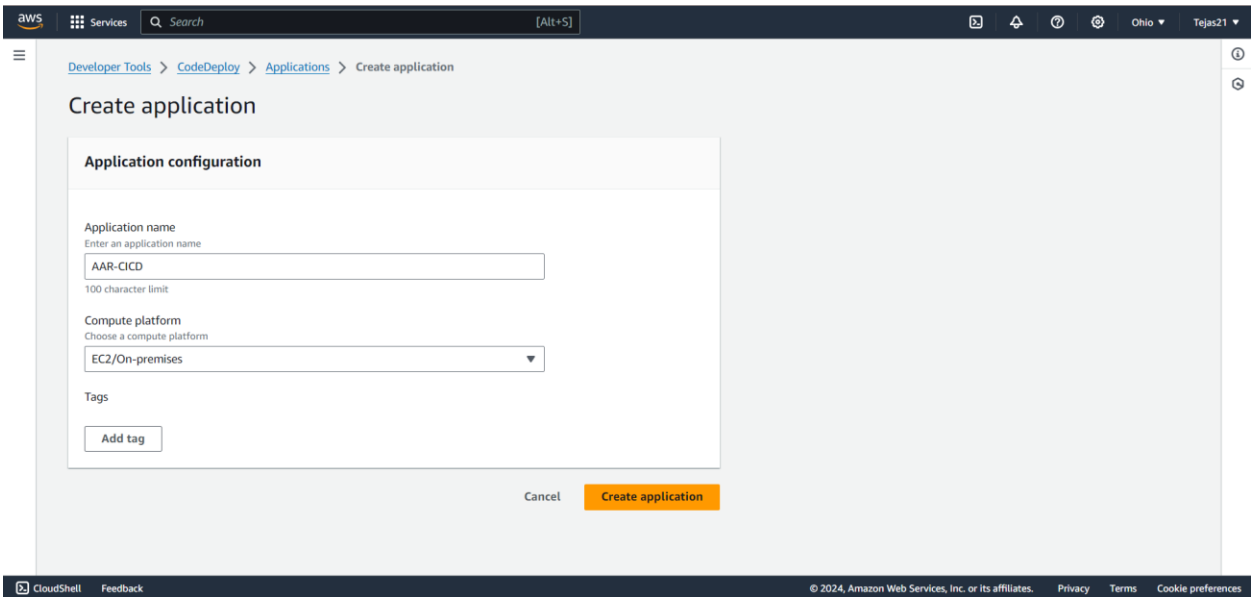

Services
Search
[Alt+S]
Stockholm
sonam
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
AAR-CICD
i-01470860556d94500
Running
t3.micro
2/2 checks passed
View alarms
eu-north-1b
i-022af7c957d95ee84
Running
t3.micro
2/2 checks passed
View alarms
eu-north-1b
Select an instance

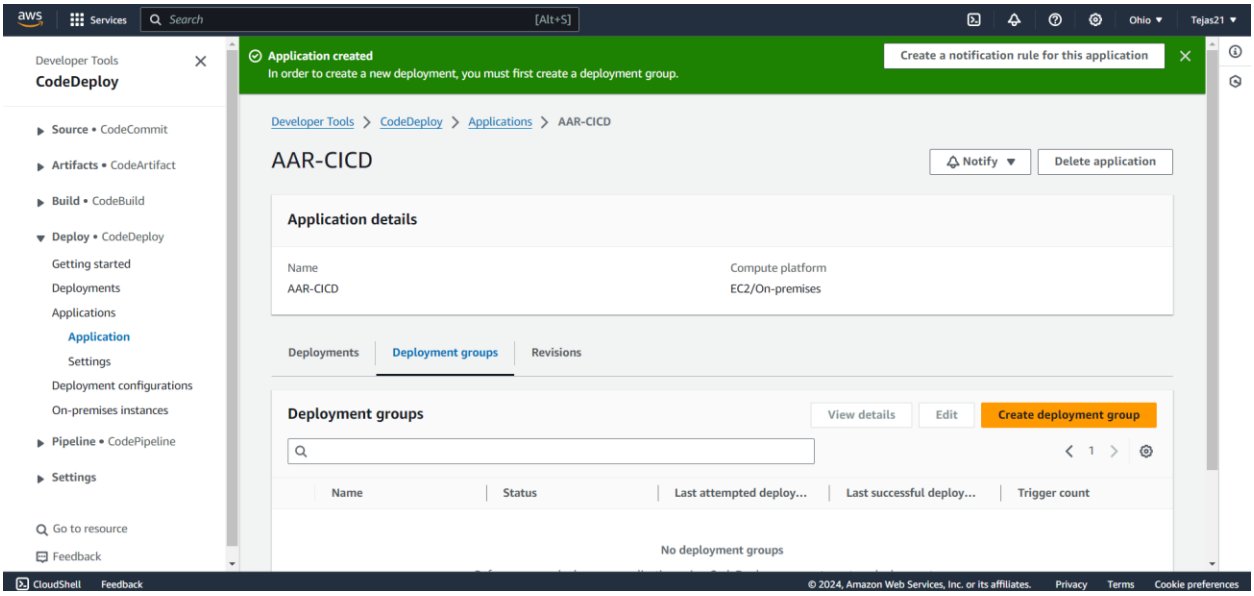

i-01470860556d94500 (AAR-CICD)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-01470860556d94500 (AAR-CICD)
Public IPv4 address
13.53.70.170 | open address
Private IPv4 addresses
172.31.37.42
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-13-53-70-170.eu-north-1.compute.amazonaws.com | open address
Hostname type
IP name: ip-172-31-37-42.eu-north-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-37-42.eu-north-1.compute.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t3.micro
Elastic IP addresses
–

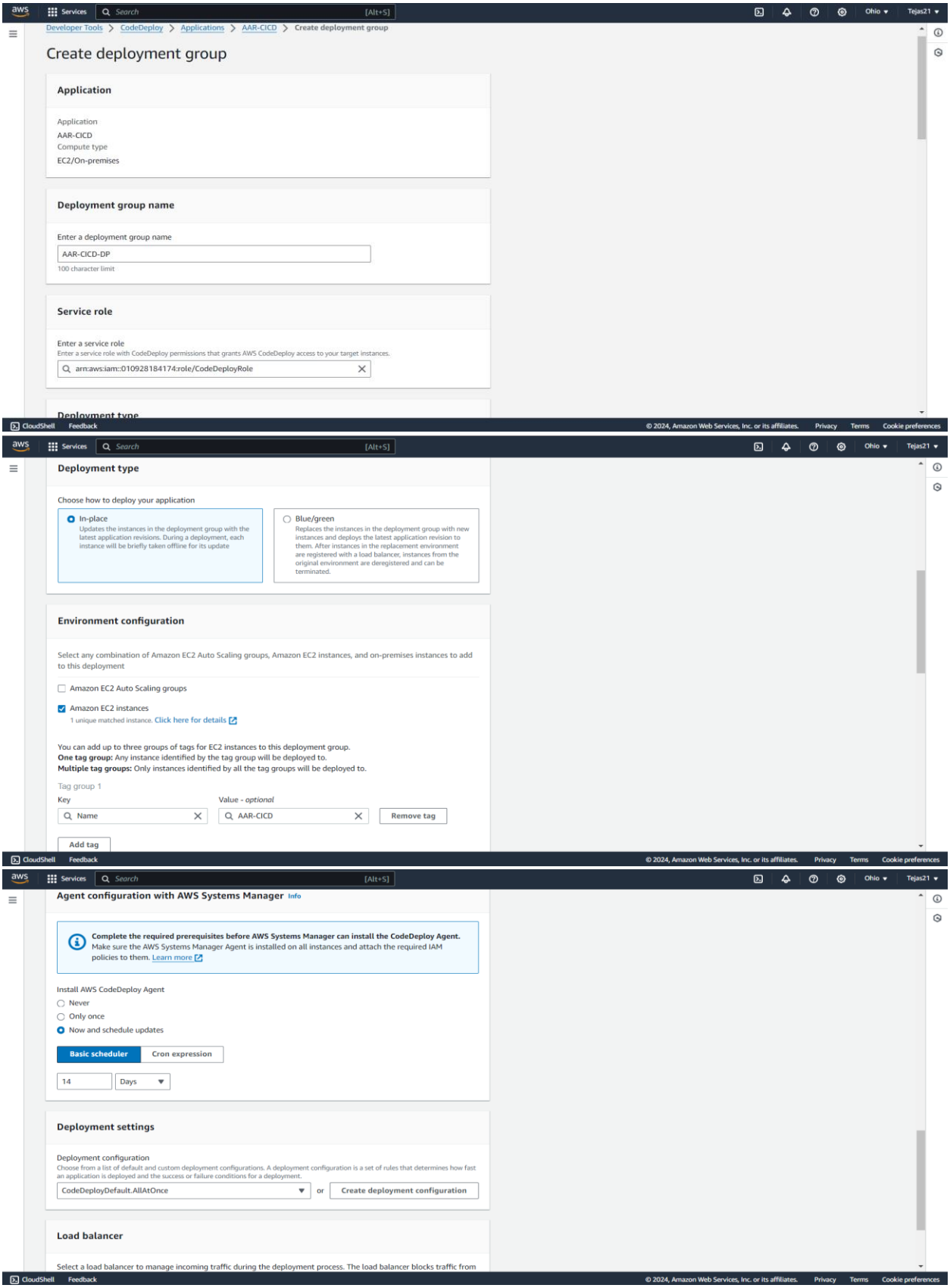
➢ Creating an application and deployment group in AWS CodeDeploy



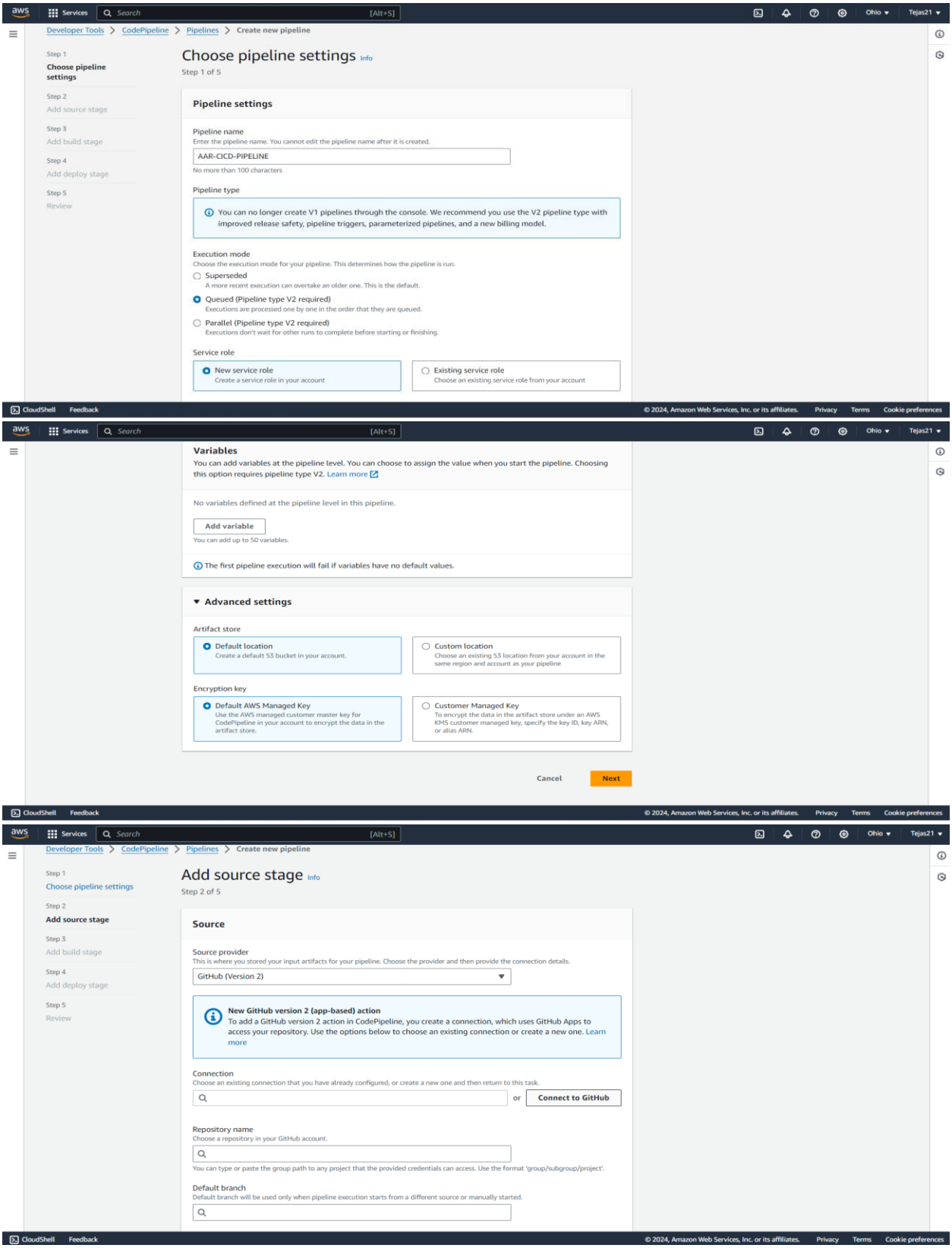



Services
Search
[Alt+S]
Ohio
Tejas21
Developer Tools > CodeDeploy > Applications > Create application
Create application
Application configuration
Application name
Enter an application name
AAR-CICD
100 character limit
Compute platform
Choose a compute platform
EC2/On-premises
Tags
Add tag
Cancel
Create application
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

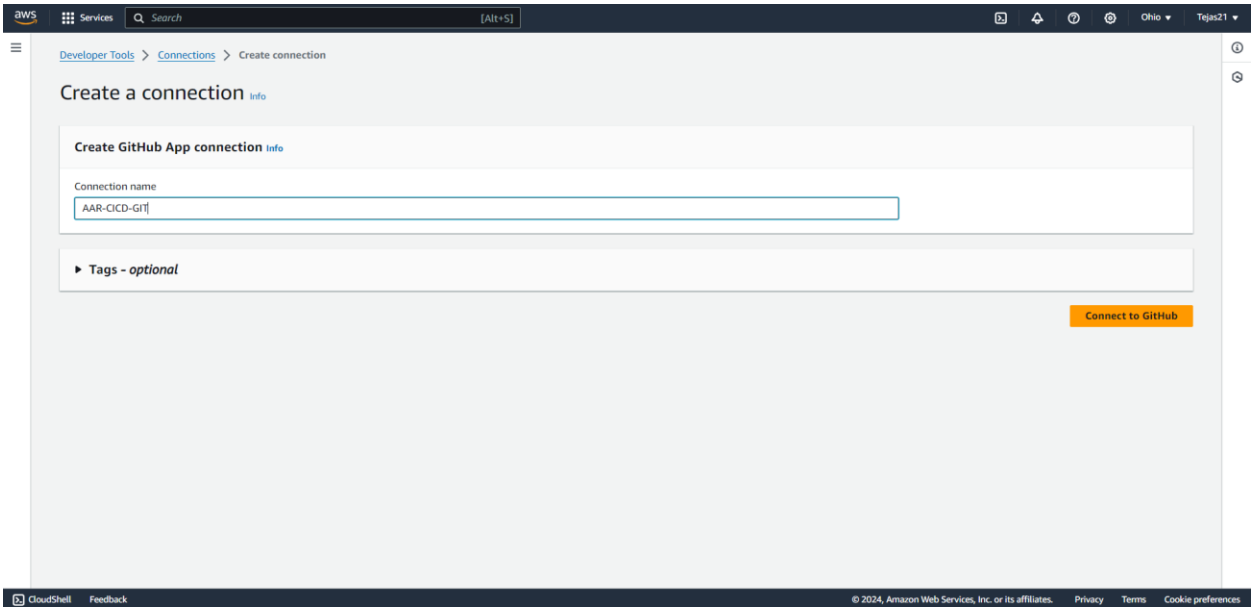

Services
Search
[Alt+S]
Ohio
Tejas21
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Application
Settings
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Go to resource
Feedback
Application created
In order to create a new deployment, you must first create a deployment group.
Create a notification rule for this application
Developer Tools > CodeDeploy > Applications > AAR-CICD
AAR-CICD
Notify
Delete application
Application details
Name
AAR-CICD
Compute platform
EC2/On-premises
Deployments
Deployment groups
Revisions
Deployment groups
View details
Edit
Create deployment group
Name
Status
Last attempted deploy...
Last successful deploy...
Trigger count
No deployment groups
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

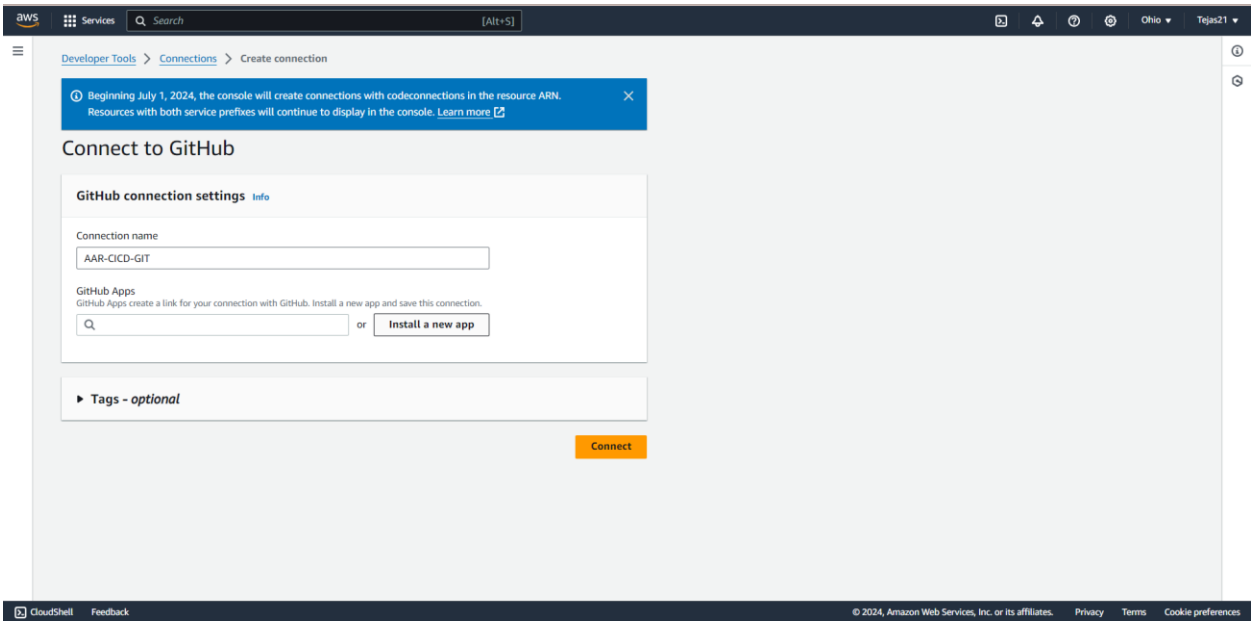

Services
Search
[Alt+S]
Ohio
Tejas21
Developer Tools > CodeDeploy > Applications > AAR-CICD > Create deployment group
Create deployment group
Application
Application
AAR-CICD
Compute type
EC2/On-premises
Deployment group name
Enter a deployment group name
AAR-CICD-DP
100 character limit
Service role
Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::010928184174:role/CodeDeployRole
Deployment type
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Deployment type
Choose how to deploy your application
In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update
Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.
Environment configuration
Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment
Amazon EC2 Auto Scaling groups
Amazon EC2 instances
1 unique matched instance. Click here for details
You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.
Tag group 1
Key
Value - optional
Name
AAR-CICD
Remove tag
Add tag
Agent configuration with AWS Systems Manager Info
Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent.
Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more
Install AWS CodeDeploy Agent
Never
Only once
Now and schedule updates
Basic scheduler
Cron expression
14
Days
Deployment settings
Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.
CodeDeployDefault.AllAtOnce
or
Create deployment configuration
Load balancer
Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from

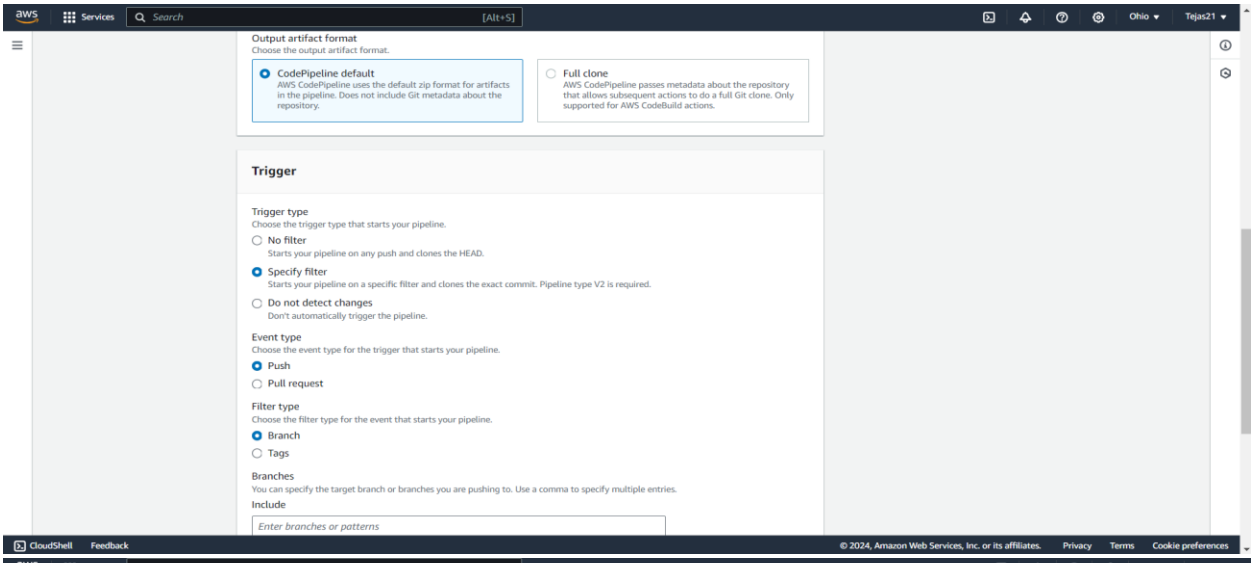
- Setting up a CICD pipeline for deploying applications on EC2 using GitHub.

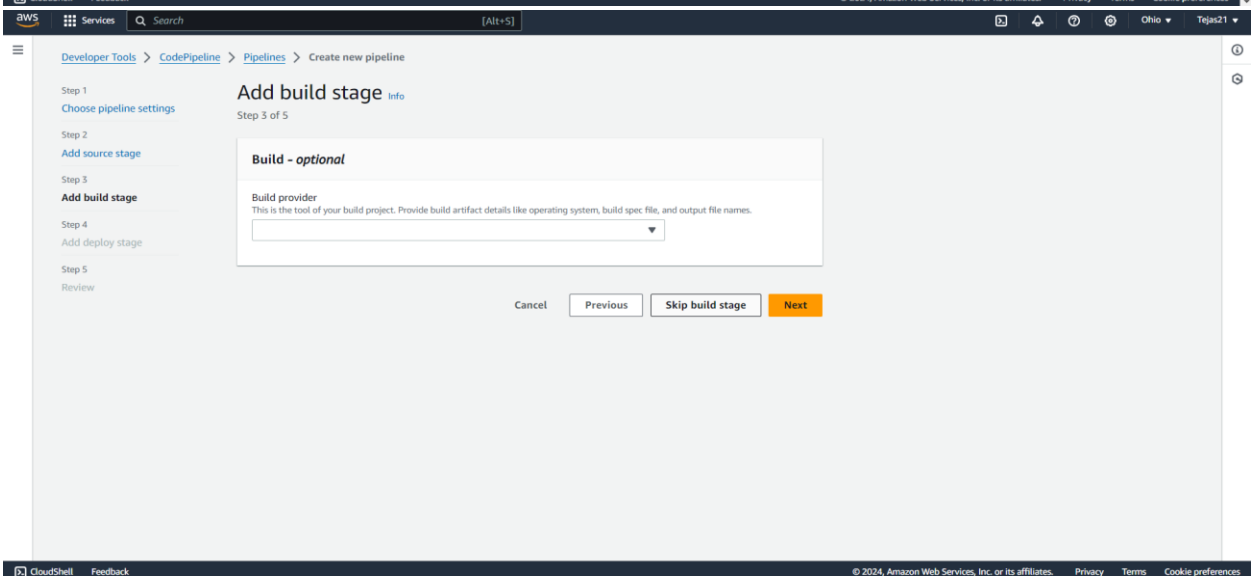


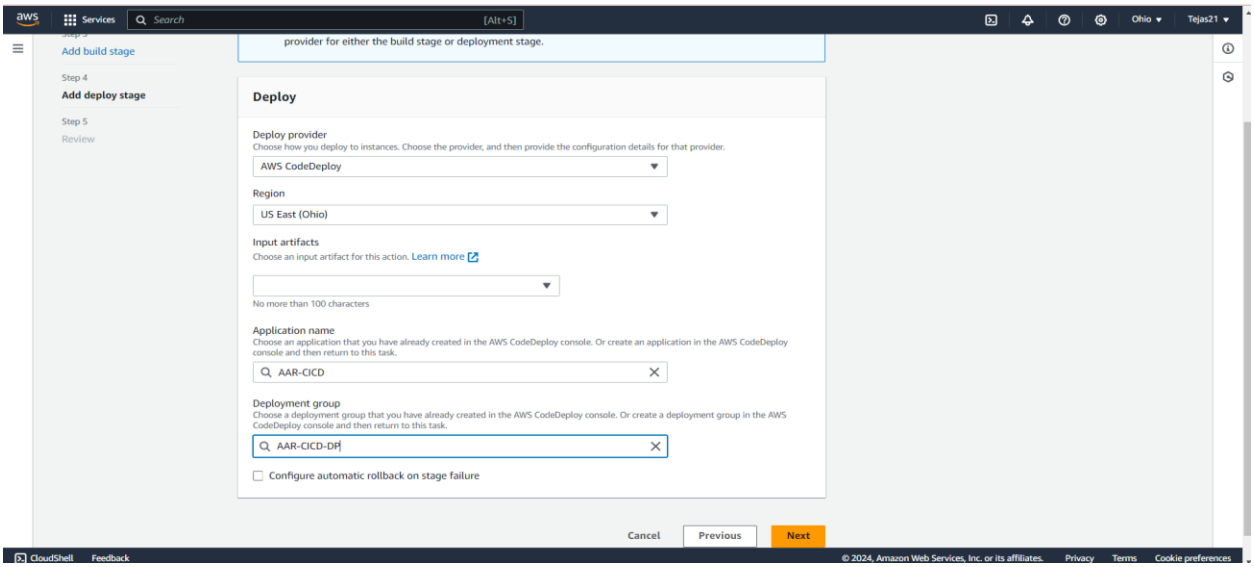
- Connecting and setting up GitHub for CI/CD deployment on AWS CodeDeploy.

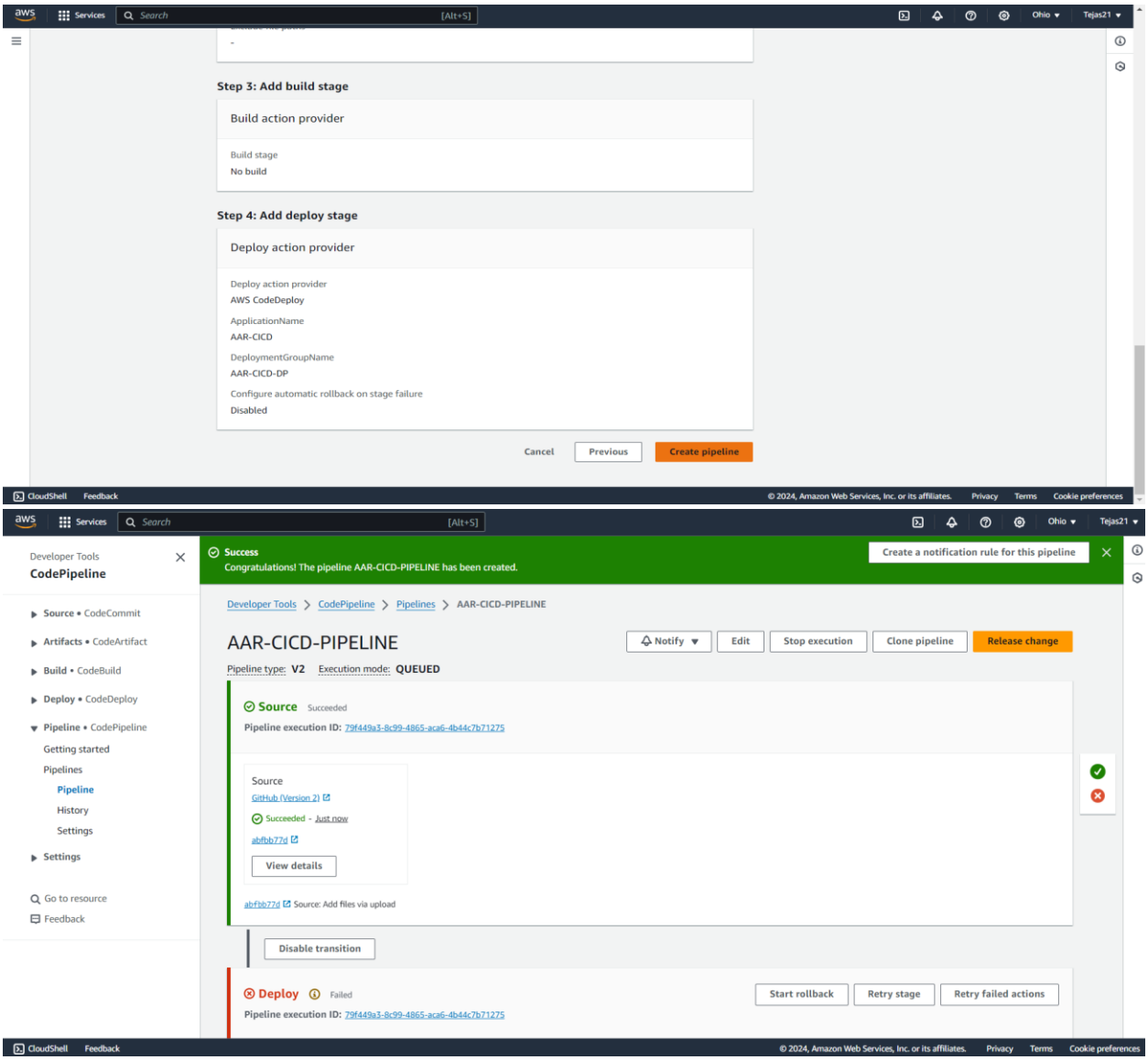


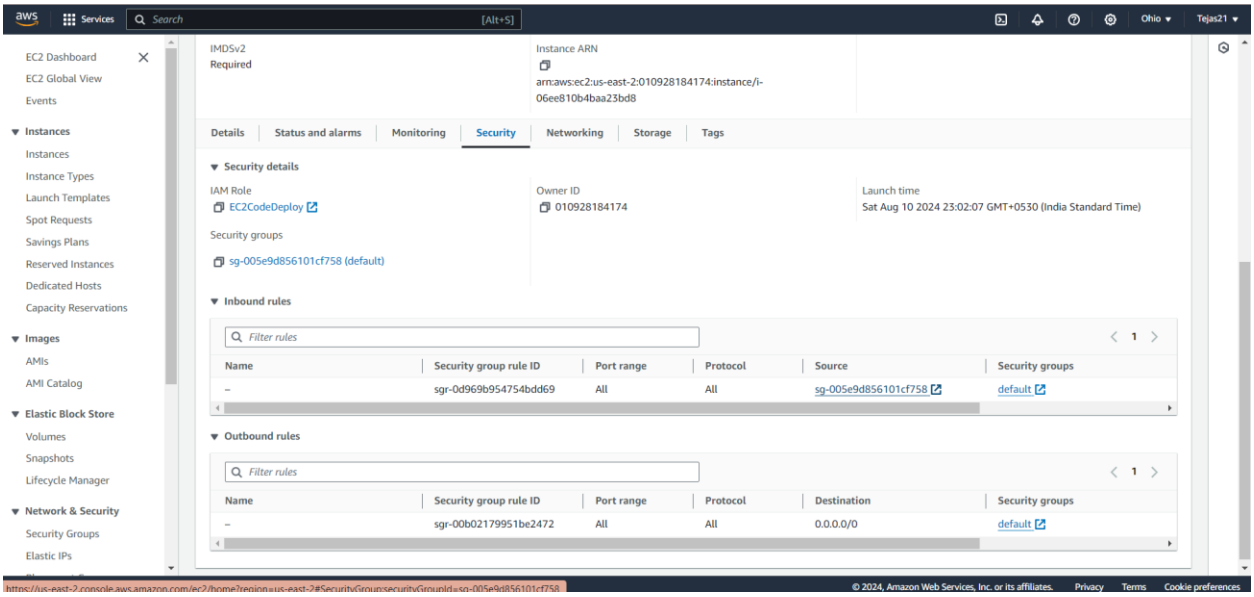


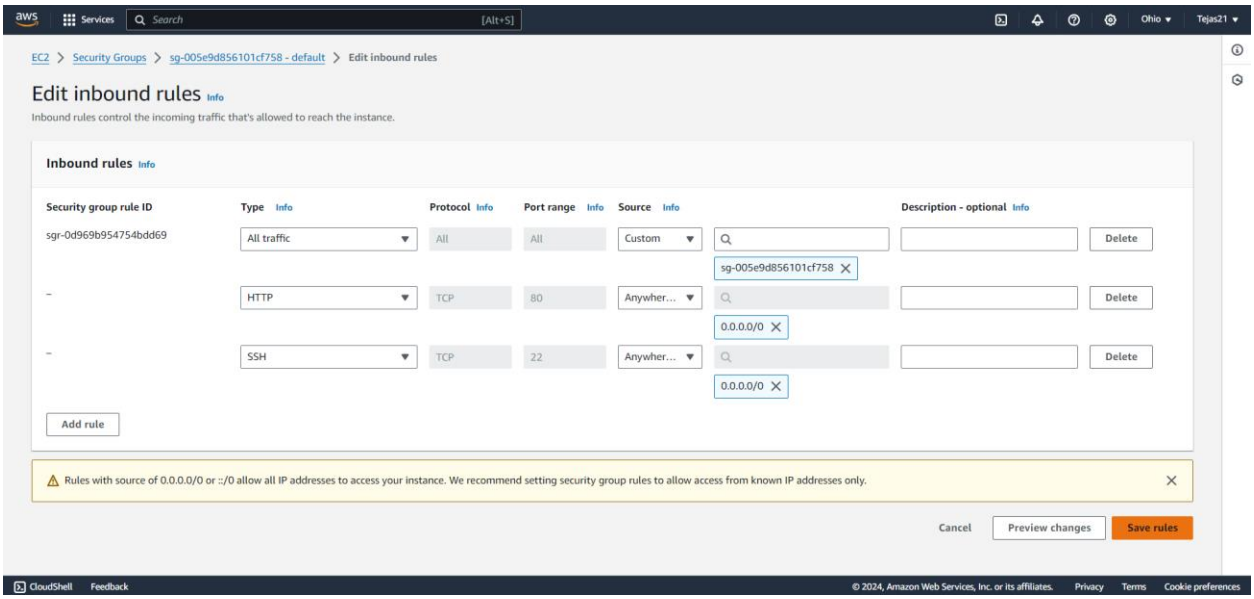

Output artifact format
Choose the output artifact format.
CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.
Trigger
Trigger type
Choose the trigger type that starts your pipeline.
No filter
Starts your pipeline on any push and clones the HEAD.
Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
Do not detect changes
Don't automatically trigger the pipeline.
Event type
Choose the event type for the trigger that starts your pipeline.
Push
Pull request
Filter type
Choose the filter type for the event that starts your pipeline.
Branch
Tags
Branches
You can specify the target branch or branches you are pushing to. Use a comma to specify multiple entries.
Include
Enter branches or patterns
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

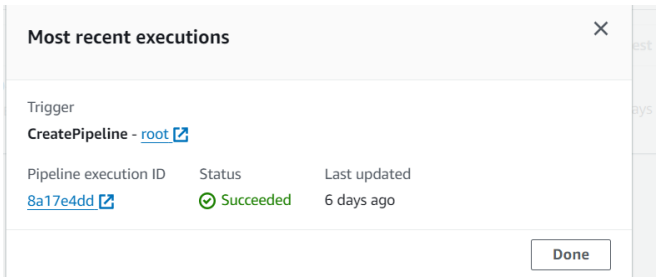

Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add build stage Info
Step 3 of 5
Build - optional
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
Cancel
Previous
Skip build stage
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

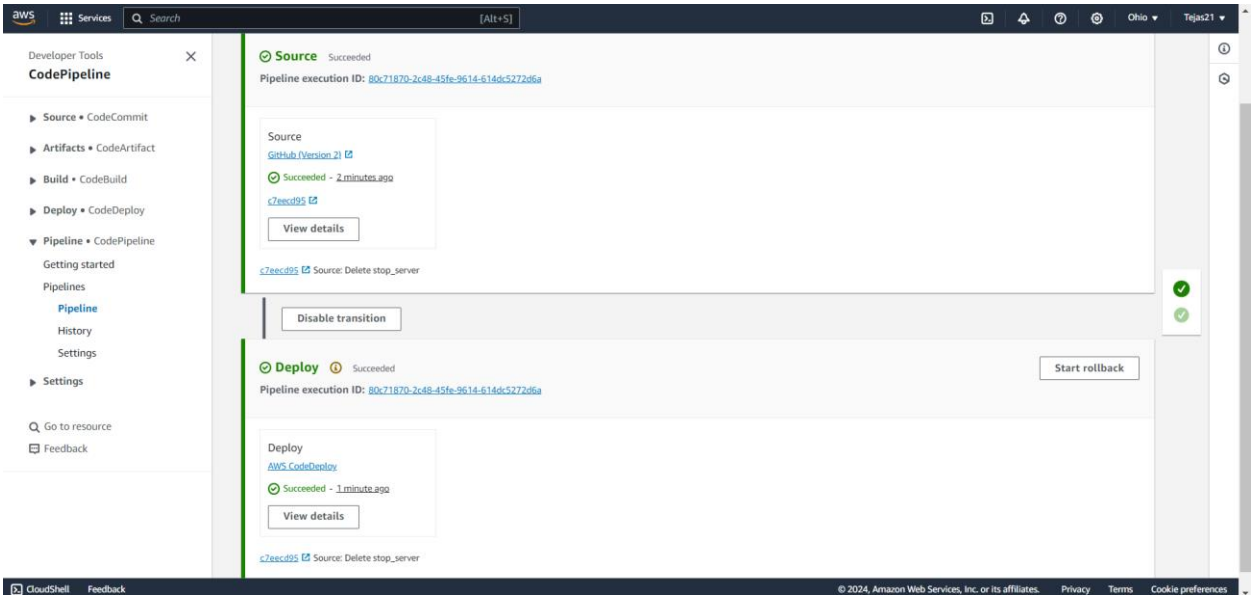

provider for either the build stage or deployment stage.
Add build stage
Step 4
Add deploy stage
Step 5
Review
Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy
Region
US East (Ohio)
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.
AAR-CICD
Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.
AAR-CICD-DP
Configure automatic rollback on stage failure
Cancel
Previous
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

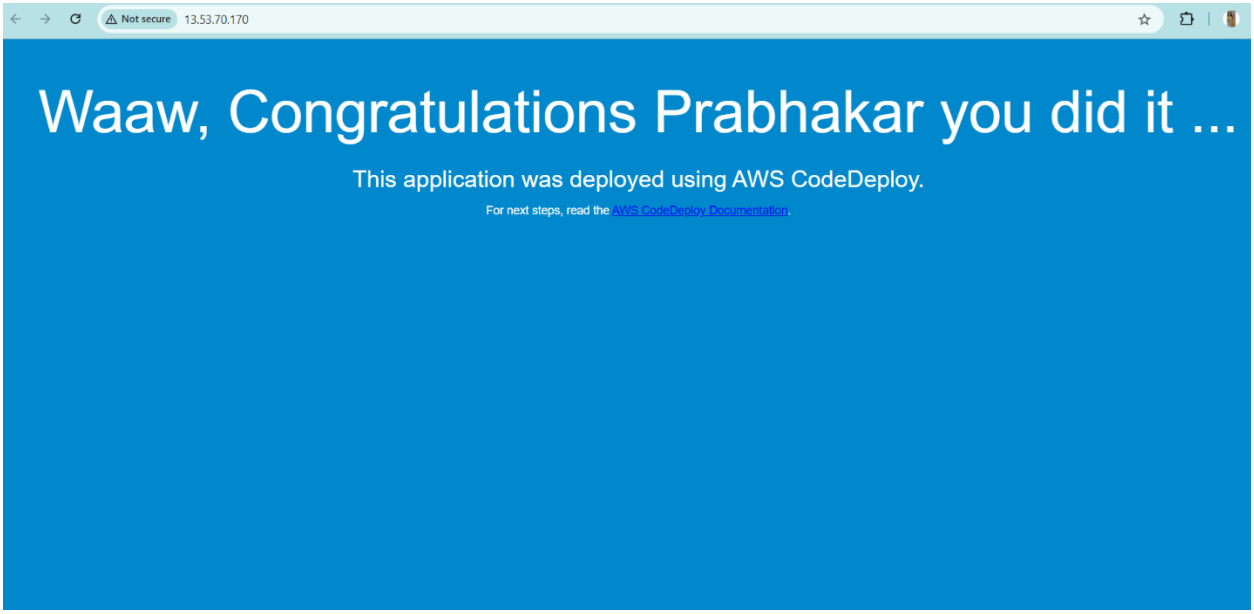

Step 3: Add build stage
Build action provider
Build stage
No build
Step 4: Add deploy stage
Deploy action provider
Deploy action provider
AWS CodeDeploy
ApplicationName
AAR-CICD
DeploymentGroupName
AAR-CICD-DP
Configure automatic rollback on stage failure
Disabled
Cancel
Previous
Create pipeline
Success
Congratulations! The pipeline AAR-CICD-PIPELINE has been created.
Create a notification rule for this pipeline
Developer Tools
CodePipeline
Developer Tools > CodePipeline > Pipelines > AAR-CICD-PIPELINE
AAR-CICD-PIPELINE
Notify
Edit
Stop execution
Clone pipeline
Release change
Pipeline type: V2
Execution mode: QUEUED
Source
Succeeded
Pipeline execution ID: 79f449a3-8c99-4865-aca6-4b44c7b71275
Source
GitHub (Version 2)
Succeeded - Just now
abfbb77d
View details
abfbb77d Source: Add files via upload
Disable transition
Deploy
Failed
Pipeline execution ID: 79f449a3-8c99-4865-aca6-4b44c7b71275
Start rollback
Retry stage
Retry failed actions


Services
Search
[Alt+S]
Ohio
Tejas21
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
IMDSv2
Required
Instance ARN
arn:aws:ec2:us-east-2:010928184174:instance/i-06ee810b4baa23bd8
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Security details
IAM Role
EC2CodeDeploy
Owner ID
010928184174
Launch time
Sat Aug 10 2024 23:02:07 GMT+0530 (India Standard Time)
Security groups
sg-005e9d856101cf758 (default)
Inbound rules
Filter rules
Name
Security group rule ID
Port range
Protocol
Source
Security groups
sgr-0d969b954754bdd69
All
All
sg-005e9d856101cf758
default
Outbound rules
Destination
sgr-00b02179951be2472
0.0.0.0/0
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Ohio
Tejas21
EC2
Security Groups
sg-005e9d856101cf758 - default
Edit inbound rules
Edit inbound rules
Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules
Info
Security group rule ID
Type
Protocol
Port range
Source
Description - optional
sgr-0d969b954754bdd69
All traffic
All
All
Custom
sg-005e9d856101cf758
Delete
HTTP
TCP
80
Anywher...
0.0.0.0/0
SSH
22
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Preview changes
Save rules
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Most recent executions
Trigger
CreatePipeline - root
Pipeline execution ID
8a17e4dd
Status
Succeeded
Last updated
6 days ago
Done


Developer Tools
CodePipeline
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Settings
Go to resource
Feedback
Source Succeeded
Pipeline execution ID: 80c71870-2c48-45fe-9614-614dc5272d6a
Source
GitHub (Version 2)
Succeeded - 2 minutes ago
c7eecd95
View details
c7eecd95 Source: Delete stop_server
Disable transition
Deploy Succeeded
Start rollback
Pipeline execution ID: 80c71870-2c48-45fe-9614-614dc5272d6a
Deploy
AWS CodeDeploy
Succeeded - 1 minute ago
View details
c7eecd95 Source: Delete stop_server
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
13.53.70.170
Waaw, Congratulations Prabhakar you did it ...
This application was deployed using AWS CodeDeploy.
For next steps, read the AWS CodeDeploy Documentation.

- After committing changes in GitHub, the pipeline automatically starts deploying the updates, and the changes are applied without any manual intervention.

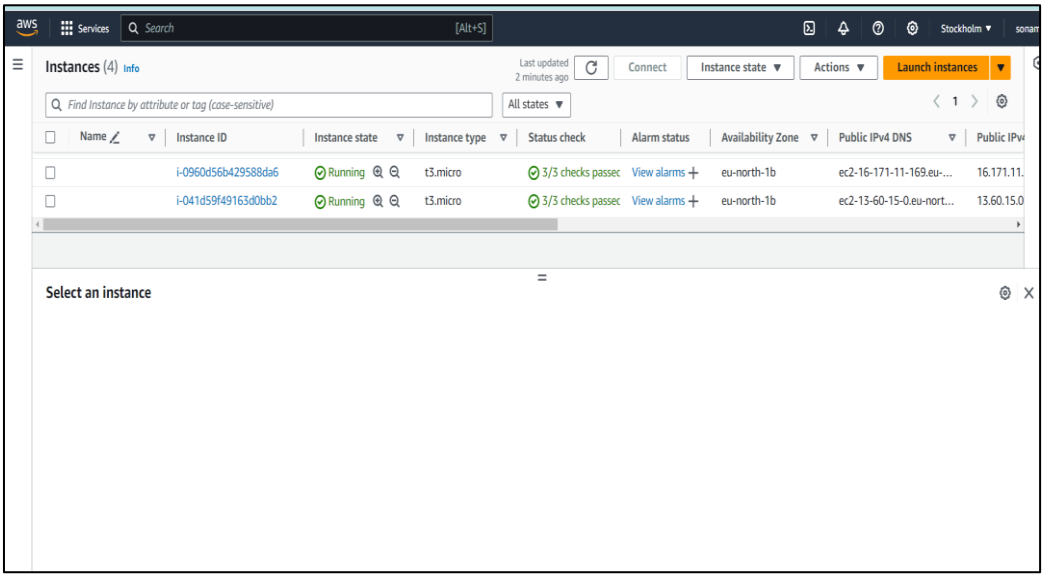
EXPERIMENT 3

Aim:

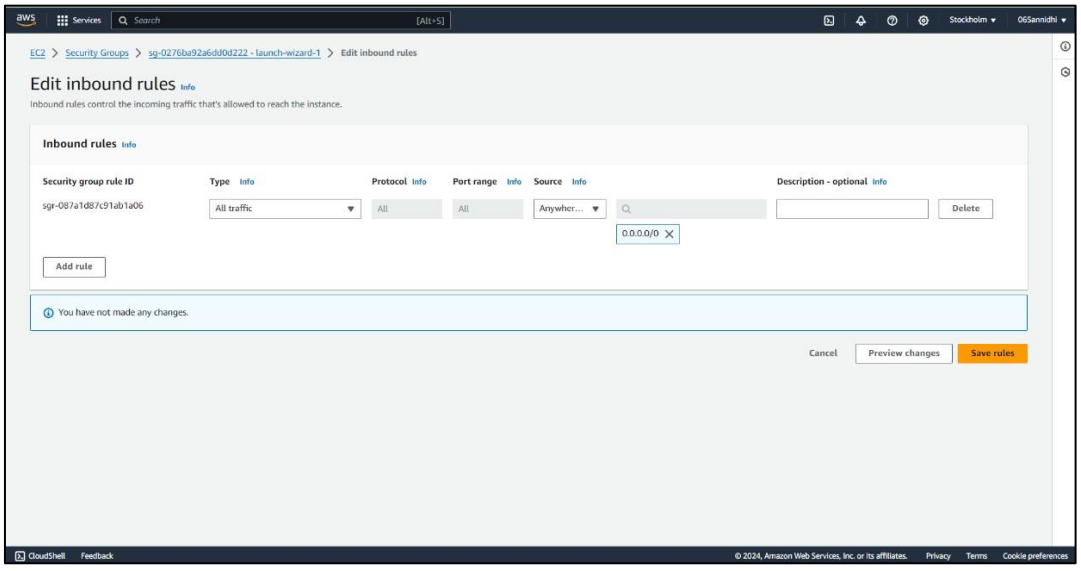
To understand the Kubernetes Cluster Architecture, install and Spin Up a KubernetesCluster on Linux Machines/Cloud Platforms.

Implementation

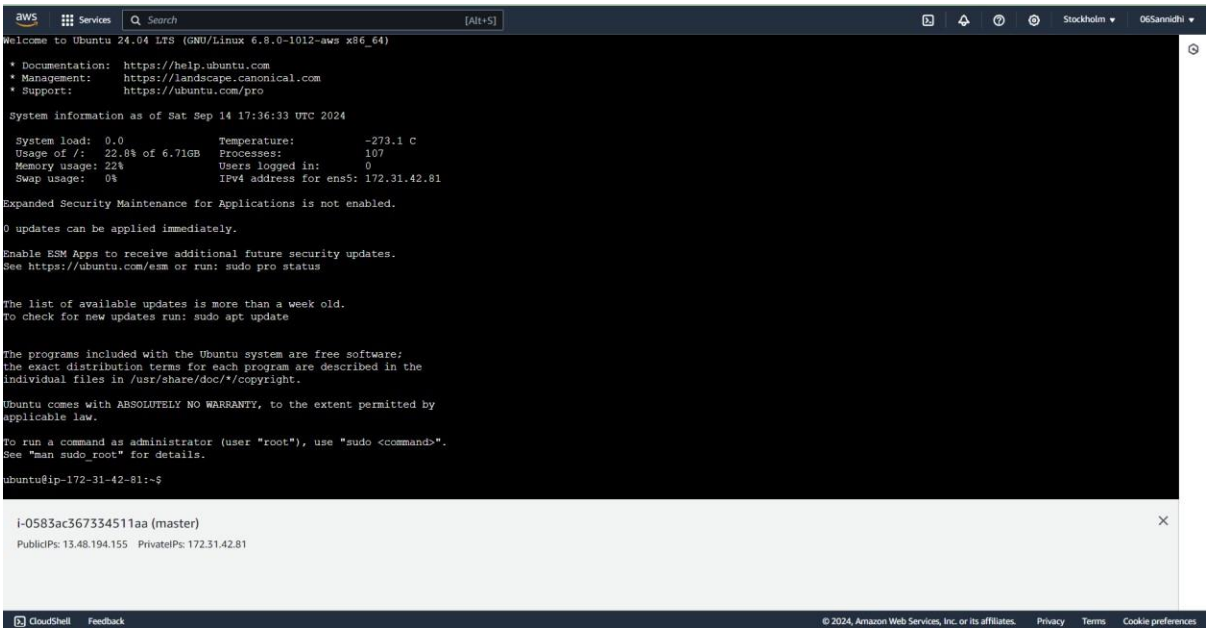
1) Create EC2 Ubuntu Instances on AWS. (Master and Worker)

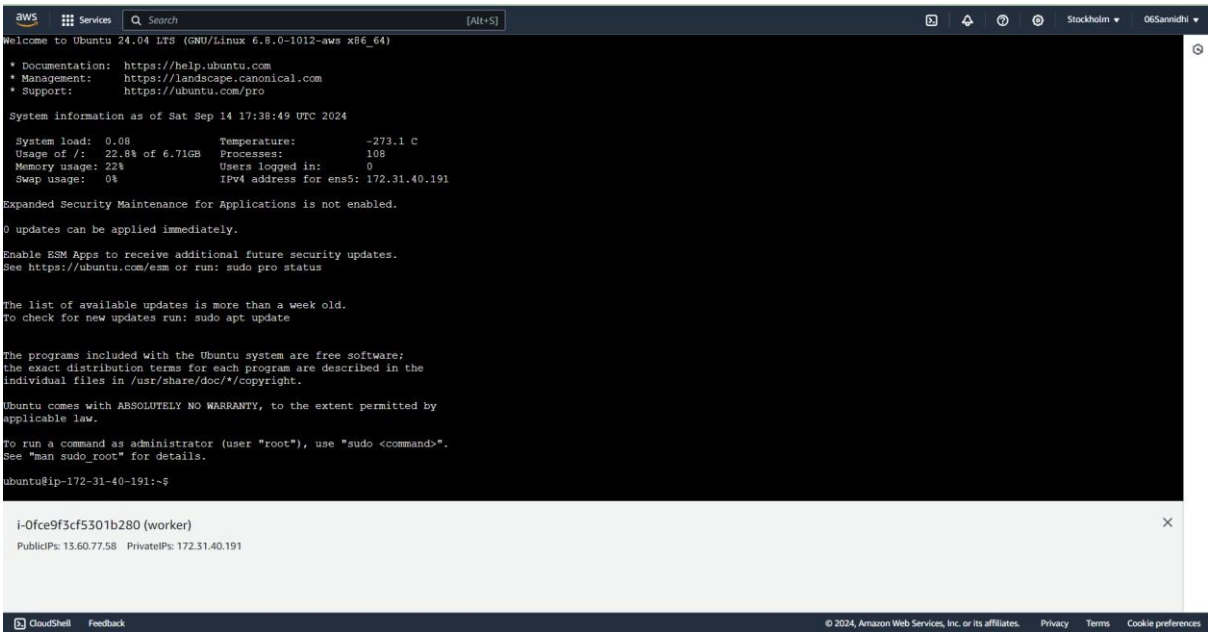


2) Edit the Security Group Inbound Rules to allow SSH



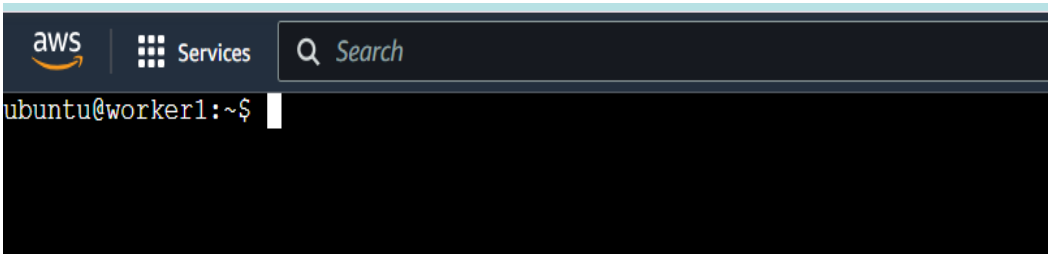
3) AWS CLI for master and worker instances





4) Assign Unique Hostname for Each Server Node

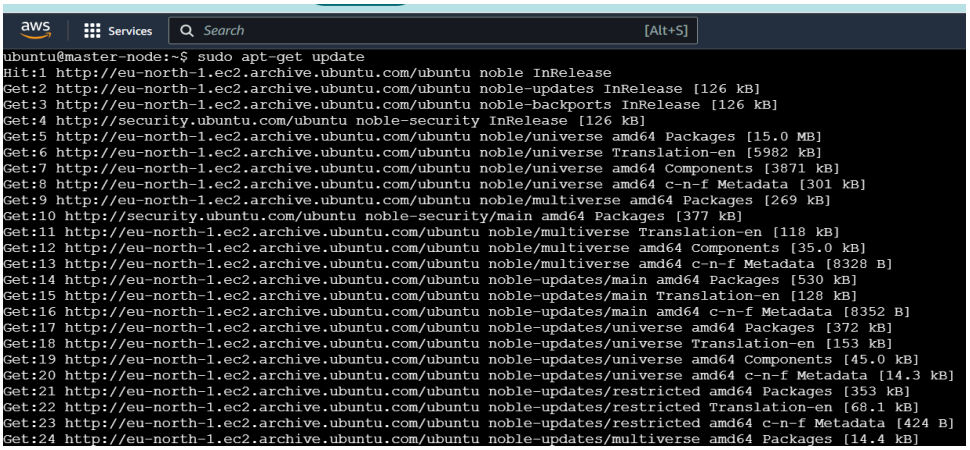
$ sudo hostnamectl set-hostname master-node
$ sudo hostnamectl set-hostname worker-1



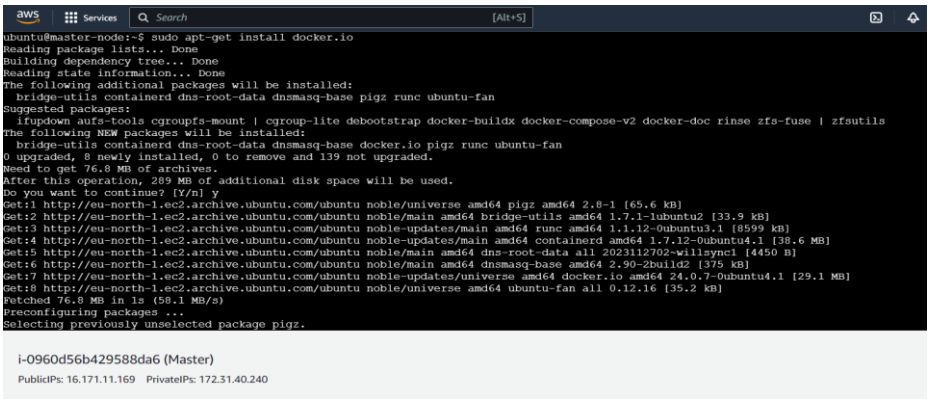
Set up Docker (both master and worker)

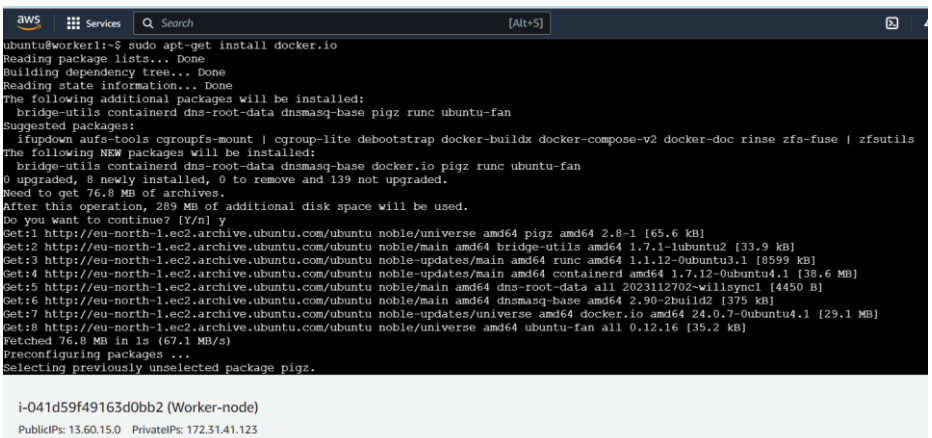
5) Install Docker

$ sudo apt-get update

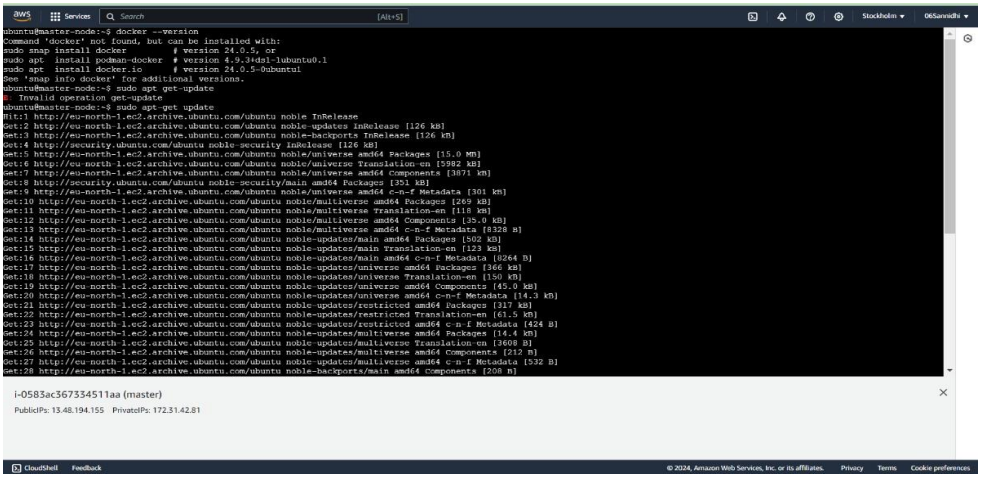


$ sudo apt-get install docker.io





$ docker —version

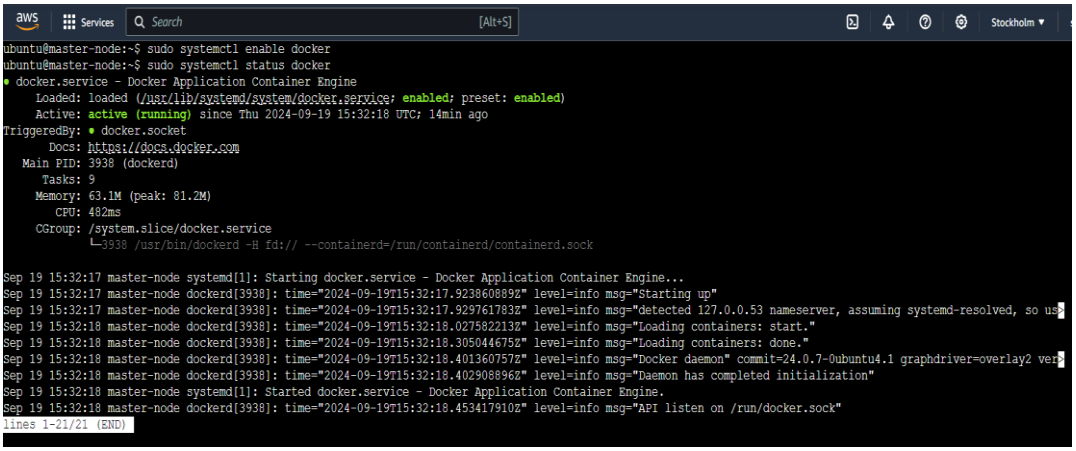


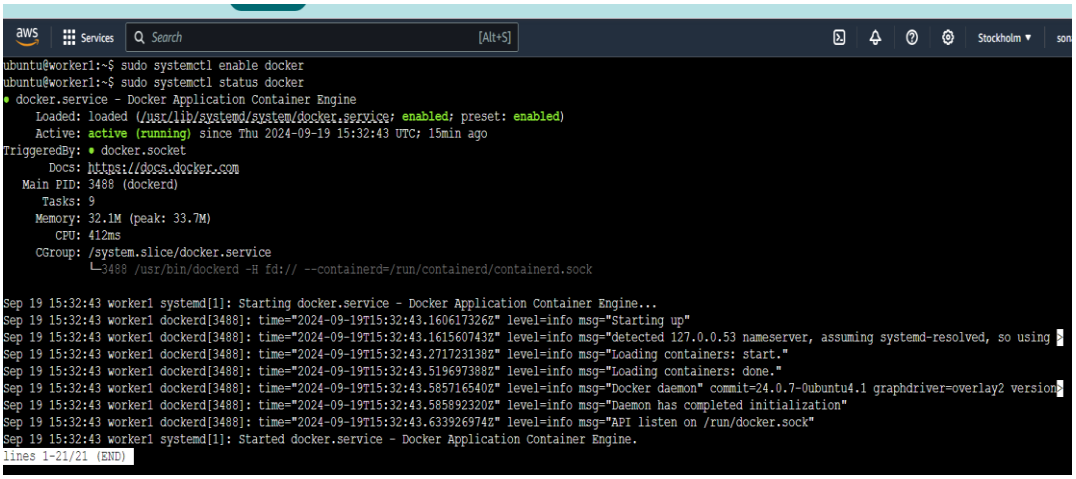
6) Start and Enable Docker

$ sudo systemctl enable docker

$ sudo systemctl status docker

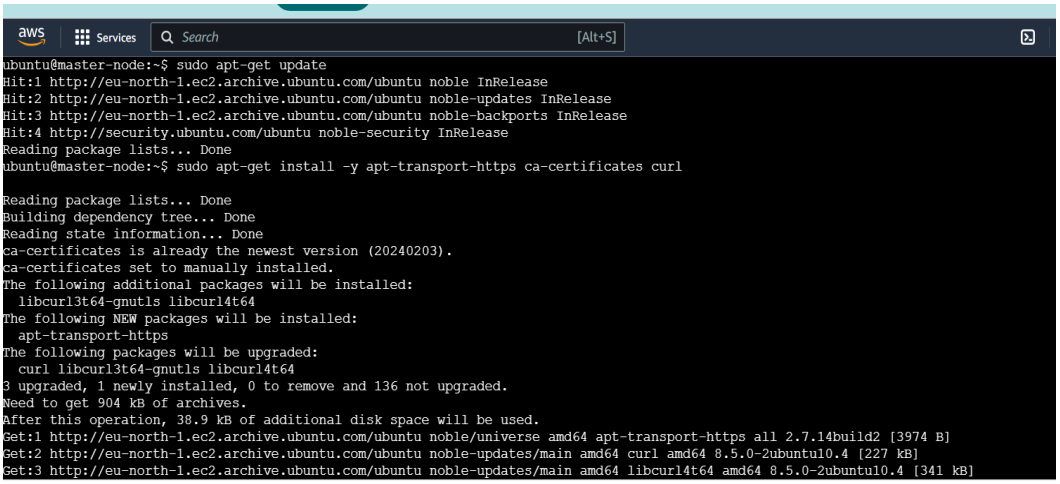
$ sudo systemctl start docker

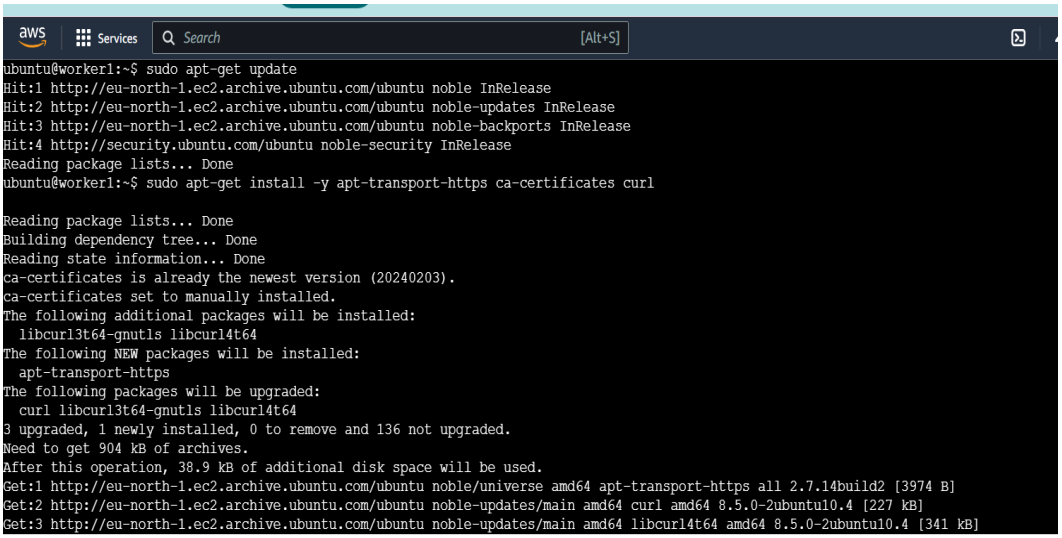




7) Install Kubernetes(both master and worker node)

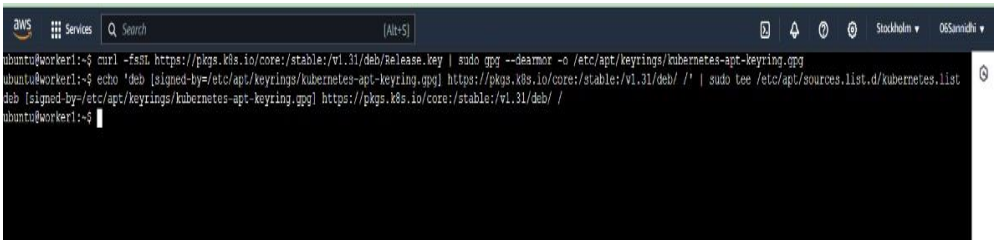
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https ca-certificates curl

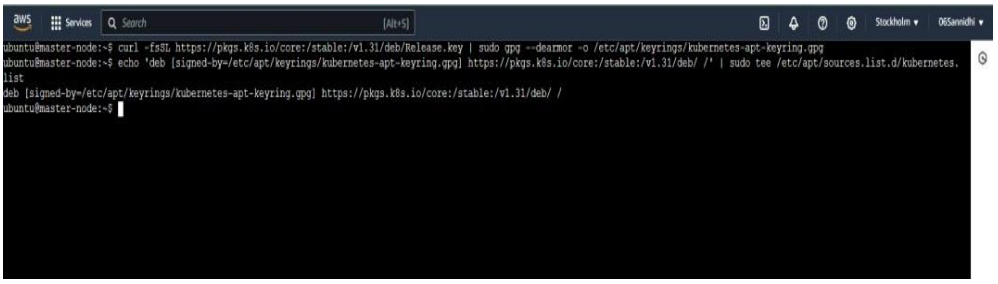




$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
(download the google cloud public signing key)
(curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg)

$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
(add the Kubernetes apt repository:)
(echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /"
| sudo tee /etc/apt/sources.list.d/kubernetes.list)

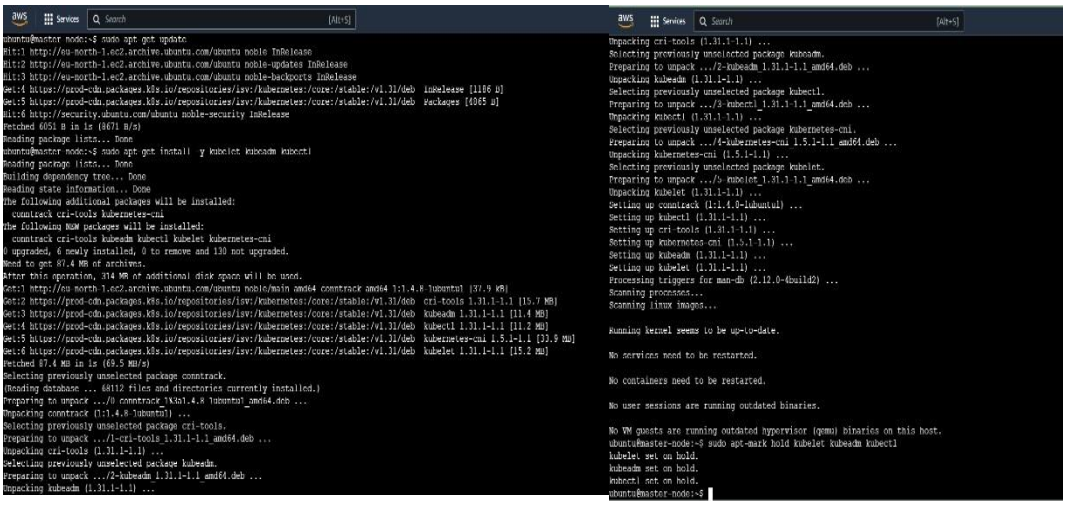


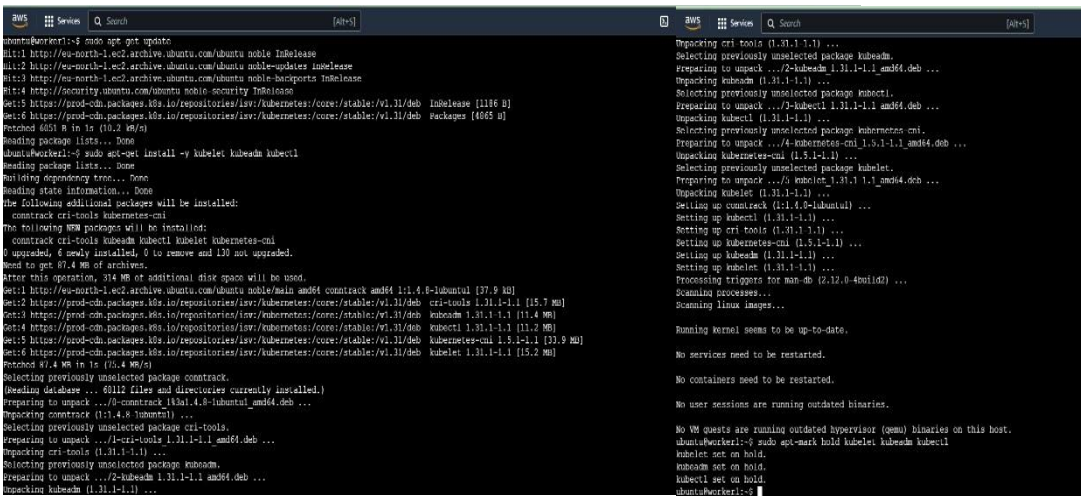


$ sudo apt-get update

$ sudo apt-get install -y kubelet kubeadm kubectl

$ sudo apt-mark hold kubelet kubeadm kubectl





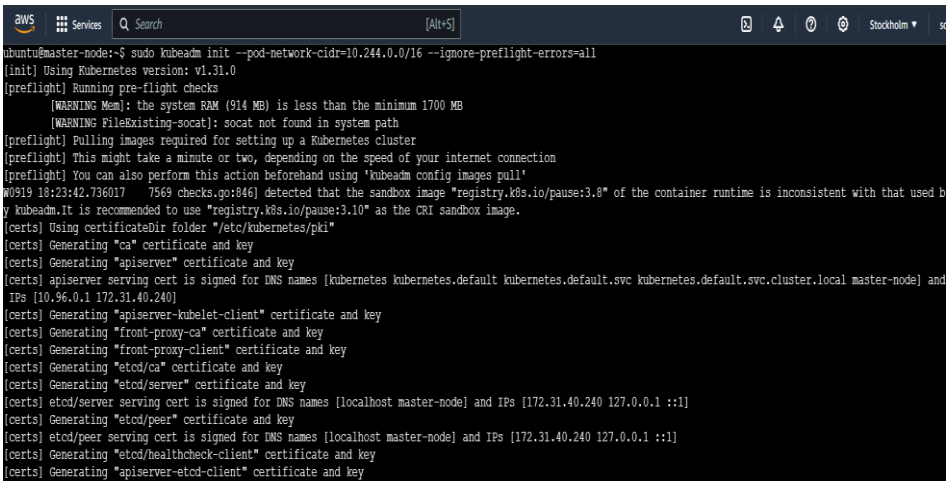
Kubernetes Deployment (master only)

8) Begin Kubernetes Deployment

$ sudo swapoff –a

9) Initialize Kubernetes on Master Node

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



10) Deploy Pod Network to Cluster

$ mkdir -p $HOME/.kube

$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

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

$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

```
kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 \
        --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0d484df29bd6bccc7808e713a1c52fa
ubuntu@master-node:~$   mkdir -p $HOME/.kube
ubuntu@master-node:~$  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master-node:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master-node:~$ ^C
ubuntu@master-node:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@master-node:~$
```

$ kubectl get pods --all-namespaces

```
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                READY   STATUS             RESTARTS        AGE
kube-flannel   kube-flannel-ds-gmnqm               1/1     Running            0               4m57s
kube-system    coredns-7c65d6cfc9-bb6x4            1/1     Running            0               15m
kube-system    coredns-7c65d6cfc9-zfsvw            1/1     Running            0               15m
kube-system    etcd-master-node                    1/1     Running            0               15m
kube-system    kube-apiserver-master-node          1/1     Running            0               15m
kube-system    kube-controller-manager-master-node 1/1     Running            0               15m
kube-system    kube-proxy-k2ksj                    0/1     CrashLoopBackOff   6 (2m40s ago)   15m
kube-system    kube-scheduler-master-node          1/1     Running            0               15m
ubuntu@master-node:~$
```

Join Worker Node to Cluster (on worker node)

11) sudo kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0d484df29bd6bccc7808e713a1c52fa –ignore-preflight-errors=all

```
ubuntu@worker1:~$ sudo kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0
4df29bd6bccc7808e713a1c52fa --ignore-preflight-errors=all
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 502.220002ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

12) $ kubectl get nodes (on master node)

```
aws  Services  Search                                                    [Alt+S]
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                READY   STATUS             RESTARTS        AGE
kube-flannel   kube-flannel-ds-gmnqm               1/1     Running            0               4m57s
kube-system    coredns-7c65d6cfc9-bb6x4            1/1     Running            0               15m
kube-system    coredns-7c65d6cfc9-zfsvw            1/1     Running            0               15m
kube-system    etcd-master-node                    1/1     Running            0               15m
kube-system    kube-apiserver-master-node          1/1     Running            0               15m
kube-system    kube-controller-manager-master-node 1/1     Running            0               15m
kube-system    kube-proxy-k2ksj                    0/1     CrashLoopBackOff   6 (2m40s ago)   15m
kube-system    kube-scheduler-master-node          1/1     Running            0               15m
ubuntu@master-node:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE     VERSION
master-node   Ready    control-plane   39m     v1.31.1
worker1       Ready    <none>          9m44s   v1.31.1
ubuntu@master-node:~$
```

we now have a Kubernetes cluster running across AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed acrossthese machines.

Conclusion

Successfully understood the Kubernetes cluster architecture and deployed a Kubernetes cluster on Linux machines/cloud platforms, demonstrating seamless setupand orchestration.

EXPERIMENT 4

Aim

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

Implementation

1) In your Kubernetes cluster check if all nodes are connected.
   $ kubectl get nodes

```
ubuntu@master-node:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE   VERSION
master-node   Ready    control-plane   54m   v1.31.1
worker1       Ready    <none>          36m   v1.31.1
ubuntu@master-node:~$
```

2) Create deploy.yaml file
   $ sudo nano deploy.yaml

3) Copy nginx-deployement.yaml file from Kubernetes site and paste in above file.

4) Create deployment
   $ kubectl apply -f

5) Check if deployment is created and all pods are running
   $ kubectl get deploy.po

6) Expose the app to internet using load balancer service.
   $ kubectl expose deployment.apps/nginx-deployment –type= ”load balancer”

7) Kubectl get svc

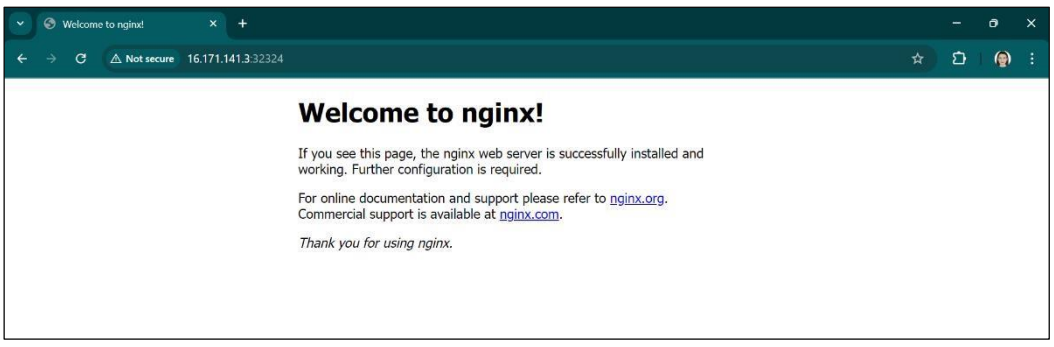
```
aws  Services  Search  [Alt+S]  Stockholm
ubuntu@master-node:~$ sudo nano deploy.yaml
ubuntu@master-node:~$ kubectl create -f deploy.yaml
error: the path "deploy.yaml" does not exist
ubuntu@master-node:~$ sudo nano deploy.yaml
ubuntu@master-node:~$ sudo nano deploy.yaml
ubuntu@master-node:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master-node:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           68s
ubuntu@master-node:~$  kubectl expose deployment.apps/nginx-deployment -type= ”load balancer”
error: there is no need to specify a resource type as a separate argument when passing arguments in resource/name form (e.g. 'kubectl get resource/<resource_name>
' instead of 'kubectl get resource resource/<resource_name>'
ubuntu@master-node:~$ kubectl get deployments
kubectl get pods
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           2m53s
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-ctbgj    1/1     Running   0          2m53s
nginx-deployment-d556bf558-jbjjs    1/1     Running   0          2m53s
nginx-deployment-d556bf558-vprbk    1/1     Running   0          2m53s
ubuntu@master-node:~$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-d556bf558-ctbgj   1/1     Running   0          3m10s
pod/nginx-deployment-d556bf558-jbjjs   1/1     Running   0          3m10s
pod/nginx-deployment-d556bf558-vprbk   1/1     Running   0          3m10s
i-0960d56b429588da6 (Master)
PublicIPs: 16.171.11.169  PrivateIPs: 172.31.40.240
```

```
 GNU nano 7.2                                  deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
                                [ Read 21 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark   M-] To Bracket  M-Q Previous
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy       ^Q Where Was    M-W Next
```

```
nginx-deployment-d556bf558-vprbk   1/1     Running   0          2m53s
ubuntu@master-node:~$ kubectl get all
NAME                                   READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-d556bf558-ctbgj   1/1     Running   0          3m10s
pod/nginx-deployment-d556bf558-jbjjs   1/1     Running   0          3m10s
pod/nginx-deployment-d556bf558-vprbk   1/1     Running   0          3m10s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   117m

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   3/3     3            3           3m10s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-d556bf558   3         3         3       3m10s
ubuntu@master-node:~$ kubectl expose deployment nginx-deployment --type=LoadBalancer --name=nginx-service
service/nginx-service exposed
ubuntu@master-node:~$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP        118m
nginx-service   LoadBalancer   10.98.229.217   <pending>     80:31665/TCP   18s
```

8)

8) deployed our Nginx server on our EC2 instance.



Conclusion

Successfully installed Kubectl and executed commands to manage the Kubernetes cluster, deploying the first application seamlessly.

EXPERIMENT 05

Roll No: 09
class: D15A

Date
Page

Experiment : 05

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

Theory :-

Terraform is an infrastructure as code (IaC) tool developed by HashiCorp that allows users to define and manage cloud and on-premises resources using declarative configuration files. Understanding its lifecycle and core concepts is essential for effectively utilizing Terraform in managing infrastructure.

Terraform Lifecycle

Terraform manages resources through a defined lifecycle that includes three primary stages:

1. Apply (Create) :- This stage involves creating the resources as defined in the configuration files. Terraform communicates with the cloud provider's API to provision the resource.

Update : In this stage, Terraform makes modifications to existing resources. This can involve incremental changes or complete replacements, depending on the nature of the update.

Teacher's Sign.: _____

3. Destroy :- This final stage removes the resources from the environment, ensuring that it is no longer managed by Terraform.

Terraform also provides a lifecycle meta-argument that allows users to control these stages more granularly. options within this meta-argument include:-

- create_before_destroy :- Ensures a new instance is created before the old one is destroyed, which is useful for maintaining service availability during updates.

- prevent_destroy :- Prevents accidental deletion of critical resources.

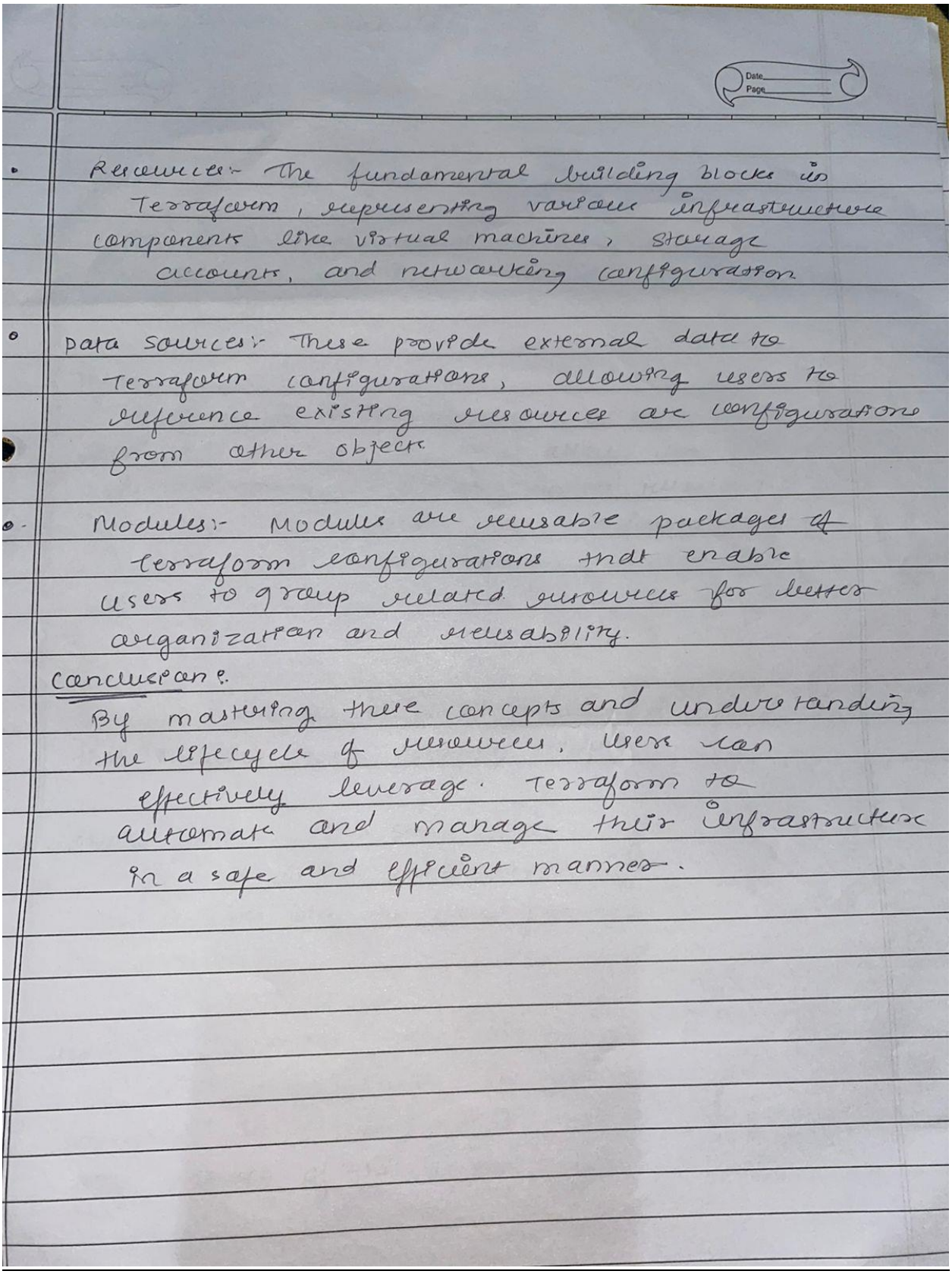
- ignore_changes :- Allows Terraform to ignore changes made outside of its management for specified fields or entire objects.

Core concepts

Several key concepts underpin Terraform's functionality

- Configuration files :- These files, written in HashiCorp Configuration language (HCL), define the desired state of the infrastructure.

Teacher's Sign.: _____



- Resources:- The fundamental building blocks in Terraform, representing various infrastructure components like virtual machines, storage accounts, and networking configurations.

- Data sources:- These provide external data to Terraform configurations, allowing users to reference existing resources or configurations from other objects.

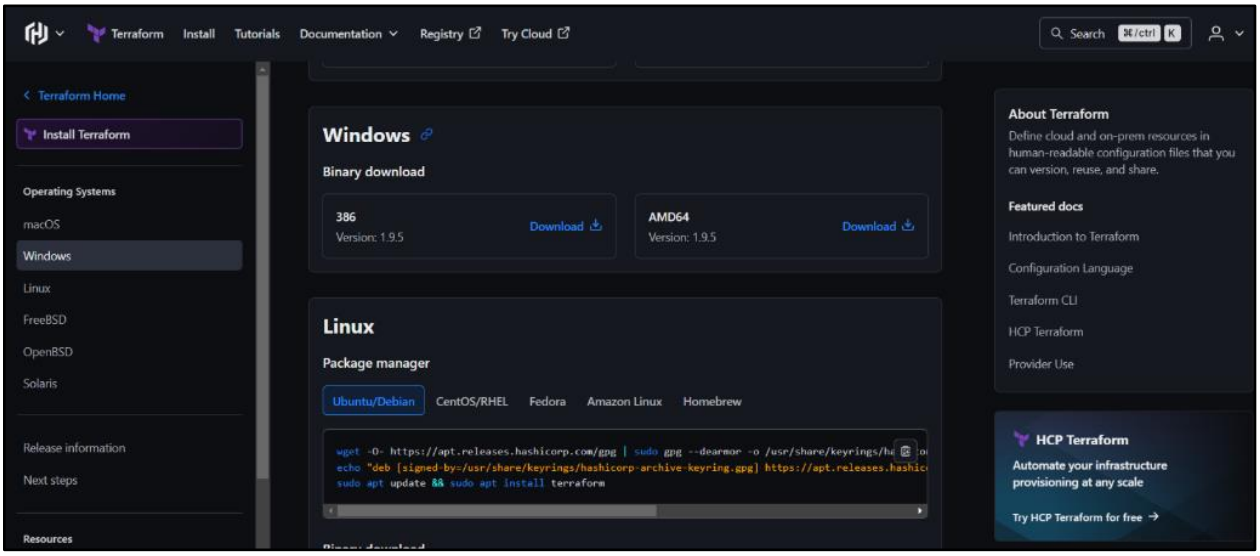
- Modules:- Modules are reusable packages of Terraform configurations that enable users to group related resources for better organization and reusability.

Conclusion:

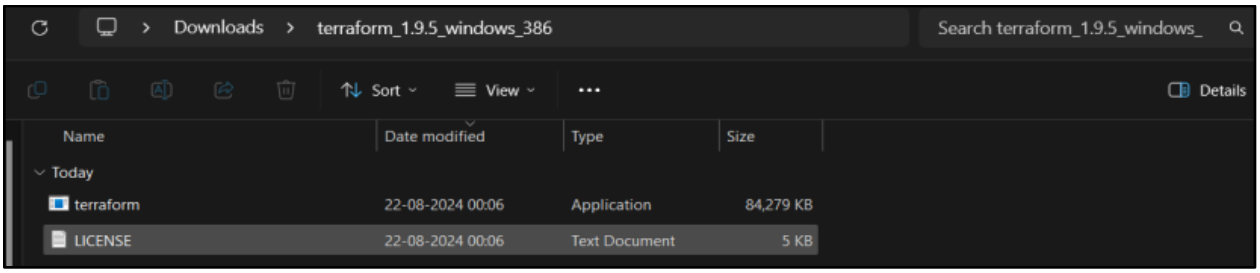
By mastering these concepts and understanding the lifecycle of resources, users can effectively leverage Terraform to automate and manage their infrastructure in a safe and efficient manner.

Implementation:

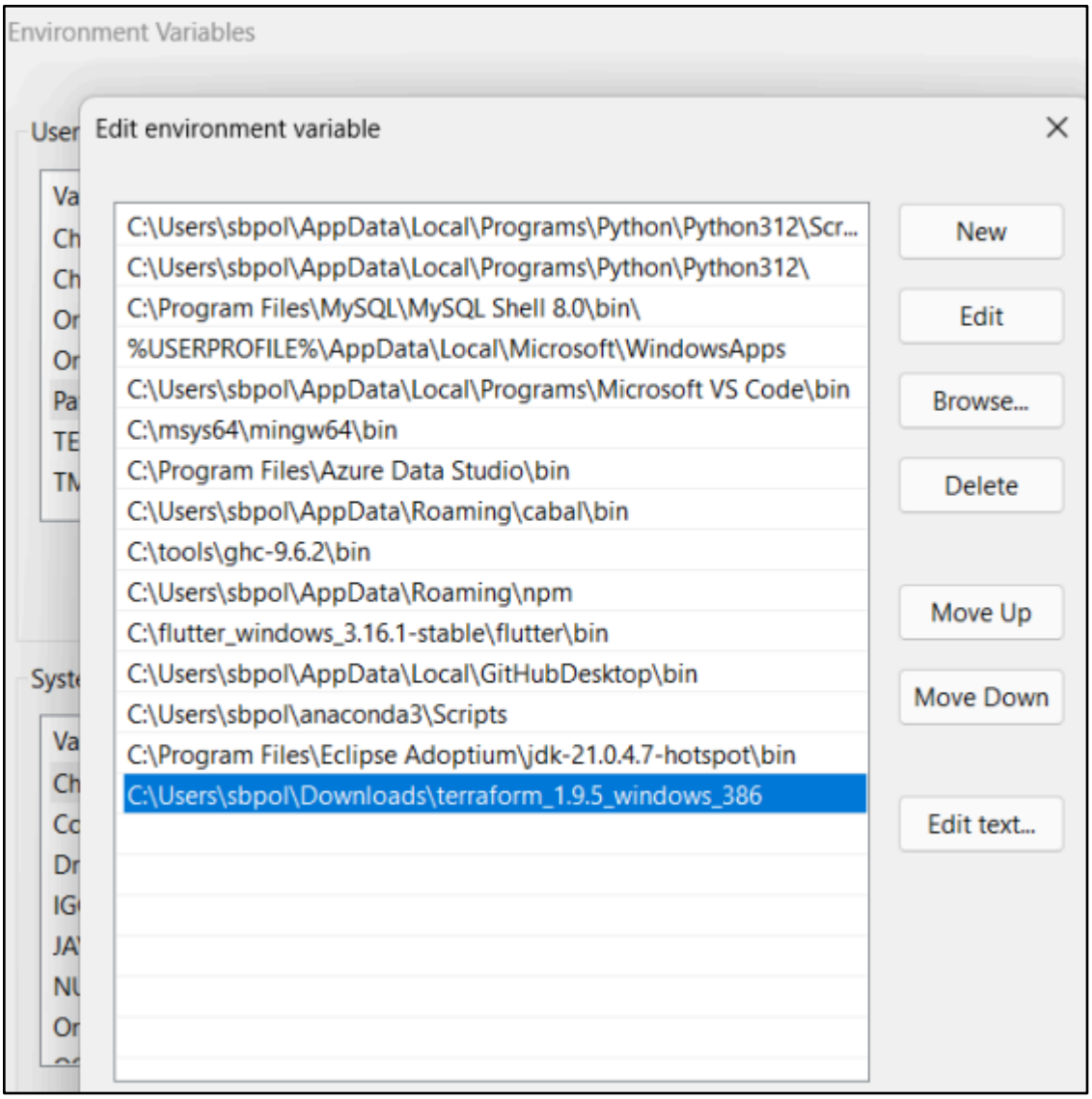
Step 1: Go to Terraform website and download 386 .



Step 2 : Extract the zip file and copy the path of the file.



Step 3: Edit environment variables and paste copied path.



Step 4 : Make sure Terraform is installed successfully.

```
(base) PS C:\Users\sbpol> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
```

```
Install the latest PowerShell for new features an

Loading personal and system profiles took 1476ms.
(base) PS C:\Users\sbpol> terraform --version
Terraform v1.9.5
on windows_386
```

EXPERIMENTS 06

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

Part A: Creating docker image using terraform

Step 1: Check Docker functionality

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

C:\Users\student>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  pull        Download an image from a registry
  push        Upload an image to a registry
  images      List images
  login       Log in to a registry
  logout      Log out from a registry
  search      Search Docker Hub for images
  version     Show the Docker version information
  info        Display system-wide information

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx
  checkpoint  Manage checkpoints
  compose*    Docker Compose
  container   Manage containers
  context     Manage contexts
  debug*      Get a shell into any image or container
  desktop*    Docker Desktop commands (Alpha)
  dev*        Docker Dev Environments
  extension*  Manages Docker extensions
  feedback*   Provide feedback, right in your terminal!
```

Check for the docker version with the following command.

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

C:\Users\student>
```

Create a folder named ‘Terraform Scripts’ in which we save our different typesof scripts which will be further used in this experiment.

Step 2:

Creating a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.
Creating a new docker.tf file using Atom editor and write the following contents into.

This will create a Ubuntu Linux container

docker.tf

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

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

# Pull the image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

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

Step 3: Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform plan

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "sleep",
          + "3600",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

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

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

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

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform apply

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "sleep",
          + "3600",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
```

```
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

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

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

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

  Enter a value: yes
```

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 9s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Before Executing Apply step:

```
PS C:\Users\Admin\TerraformScripts\Docker> docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
```

After Executing Apply step:

```
PS C:\Users\Admin\TerraformScripts\Docker> docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
ubuntu                              latest    edbfe74c41f8   3 weeks ago     78.1MB
```

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

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Terraform will perform the following actions:

  # docker_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach            = false -> null
      - command           = [
          - "sleep",
          - "3600",
        ] -> null
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      - entrypoint        = [] -> null
      - env               = [] -> null
      - gateway           = "172.17.0.1" -> null
      - group_add         = [] -> null
      - hostname          = "01adf07e5918" -> null
      - id                = "01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24" -> null
      - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - init              = false -> null
      - ip_address        = "172.17.0.2" -> null
      - ip_prefix_length  = 16 -> null
      - ipc_mode          = "private" -> null
      - links             = [] -> null
      - log_driver        = "json-file" -> null
      - log_opts          = {} -> null
      - logs              = false -> null
      - max_retry_count   = 0 -> null
```

```
      - memory            = 0 -> null
      - memory_swap       = 0 -> null
      - must_run          = true -> null
      - name              = "foo" -> null
      - network_data      = [
          - {
              - gateway                   = "172.17.0.1"
              - global_ipv6_prefix_length = 0
              - ip_address                = "172.17.0.2"
              - ip_prefix_length          = 16
              - network_name              = "bridge"
                # (2 unchanged attributes hidden)
            },
        ] -> null
      - network_mode      = "default" -> null
      - privileged        = false -> null
      - publish_all_ports = false -> null
      - read_only         = false -> null
      - remove_volumes    = true -> null
      - restart           = "no" -> null
      - rm                = false -> null
      - runtime           = "runc" -> null
      - security_opts     = [] -> null
      - shm_size          = 64 -> null
      - start             = true -> null
      - stdin_open        = false -> null
      - stop_timeout      = 0 -> null
      - storage_opts      = {} -> null
      - sysctls           = {} -> null
      - tmpfs             = {} -> null
      - tty               = false -> null
        # (8 unchanged attributes hidden)
    }
```

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

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

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

  Enter a value: yes

docker_container.foo: Destroying... [id=01adf07e5918931fee9b90073726a0367037923dd92032ce0e15bbb764a6f24]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

Docker images After Executing Destroy step

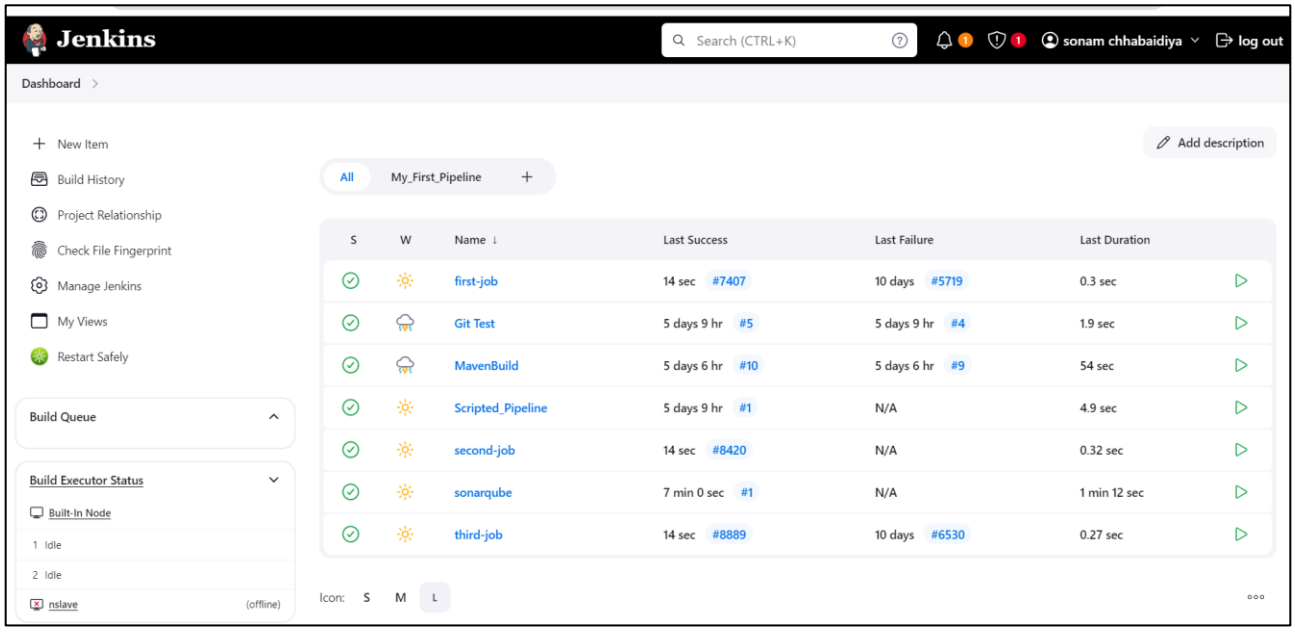
```
PS C:\Users\Admin\TerraformScripts\Docker> docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
```

Sonam chhabaidiya D15A 09

ADVANCE DEVOPS EXP-7

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

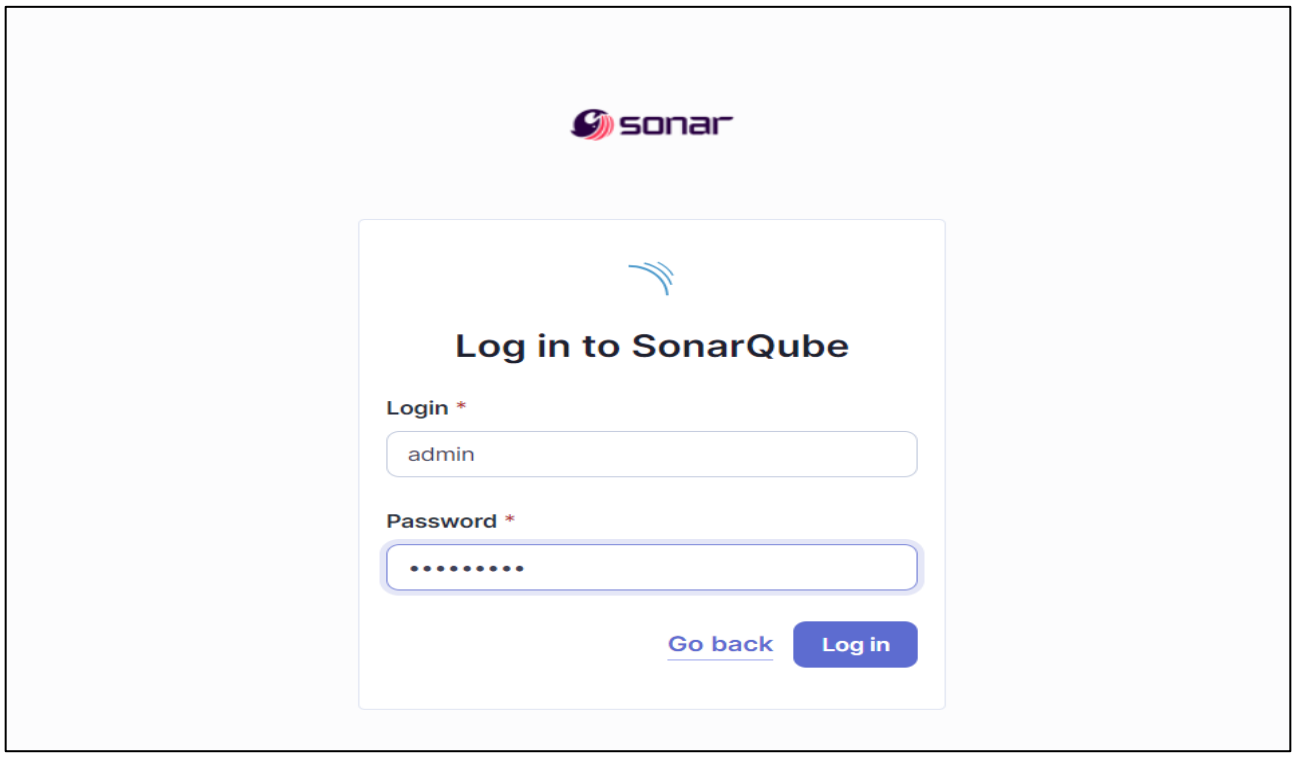
Step-1: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is atfor you.



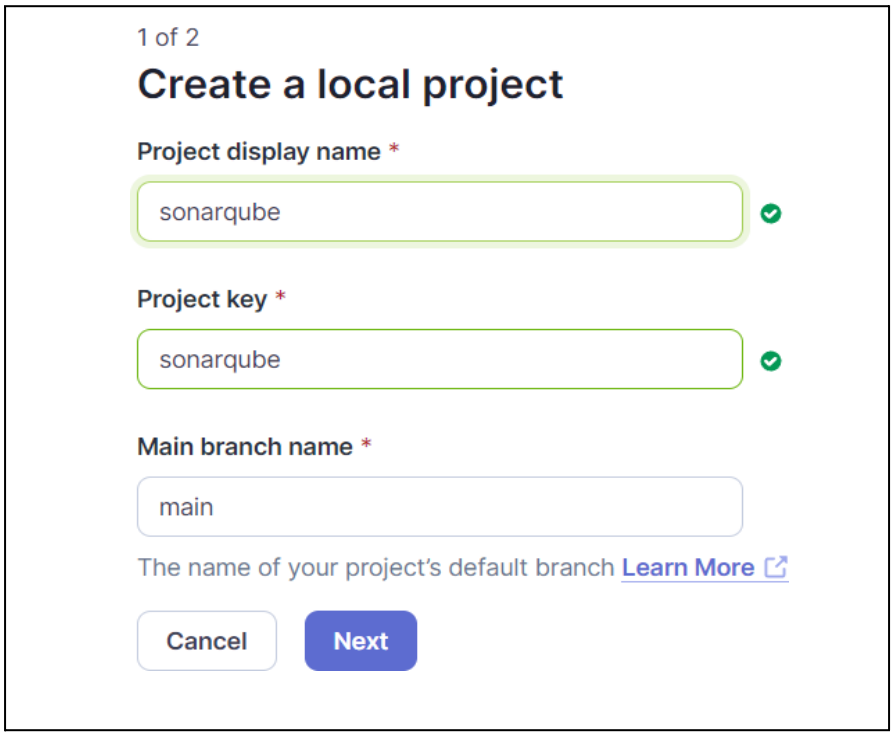
Step-2: Run SonarQube in a Docker container using this command :- a]docker -vb]
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

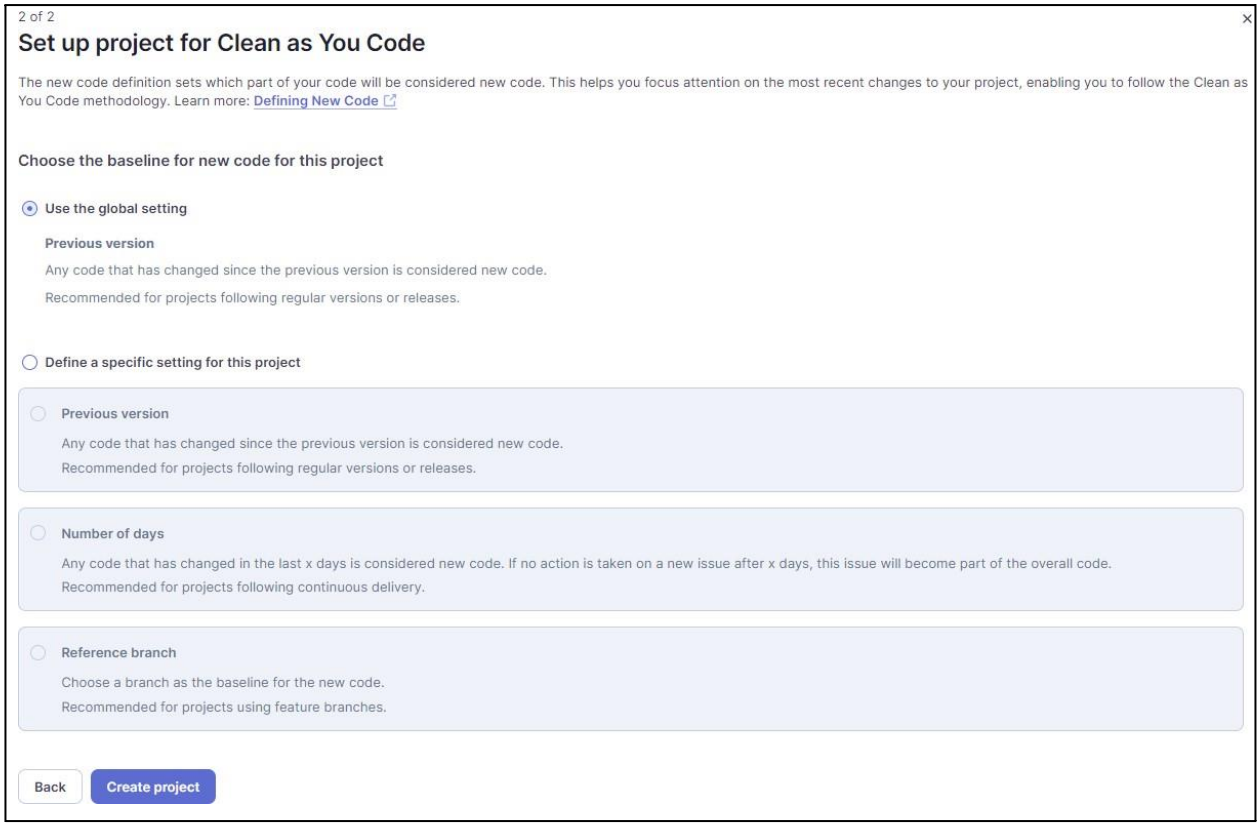
```
C:\Windows\System32>docker -v
Docker version 27.2.0, build 3ab4256

C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
1b0713c1db8319472cb04c7bb7753025f7e75b97c484e85babb3c5161d8416f0
```

Step-3: Once the container is up and running, you can check the status of SonarQubeat localhost port 9000. The login id is “admin” and the password is also “admin”.

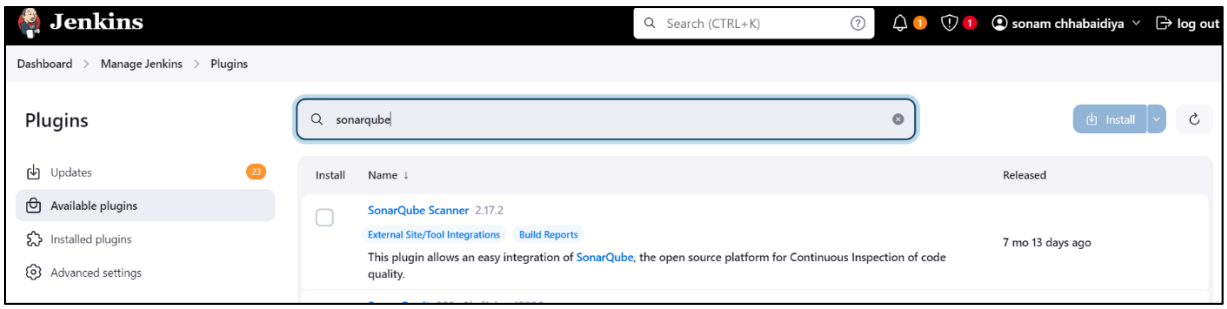


Step-4: Create a local project in SonarQube with the name sonarqube

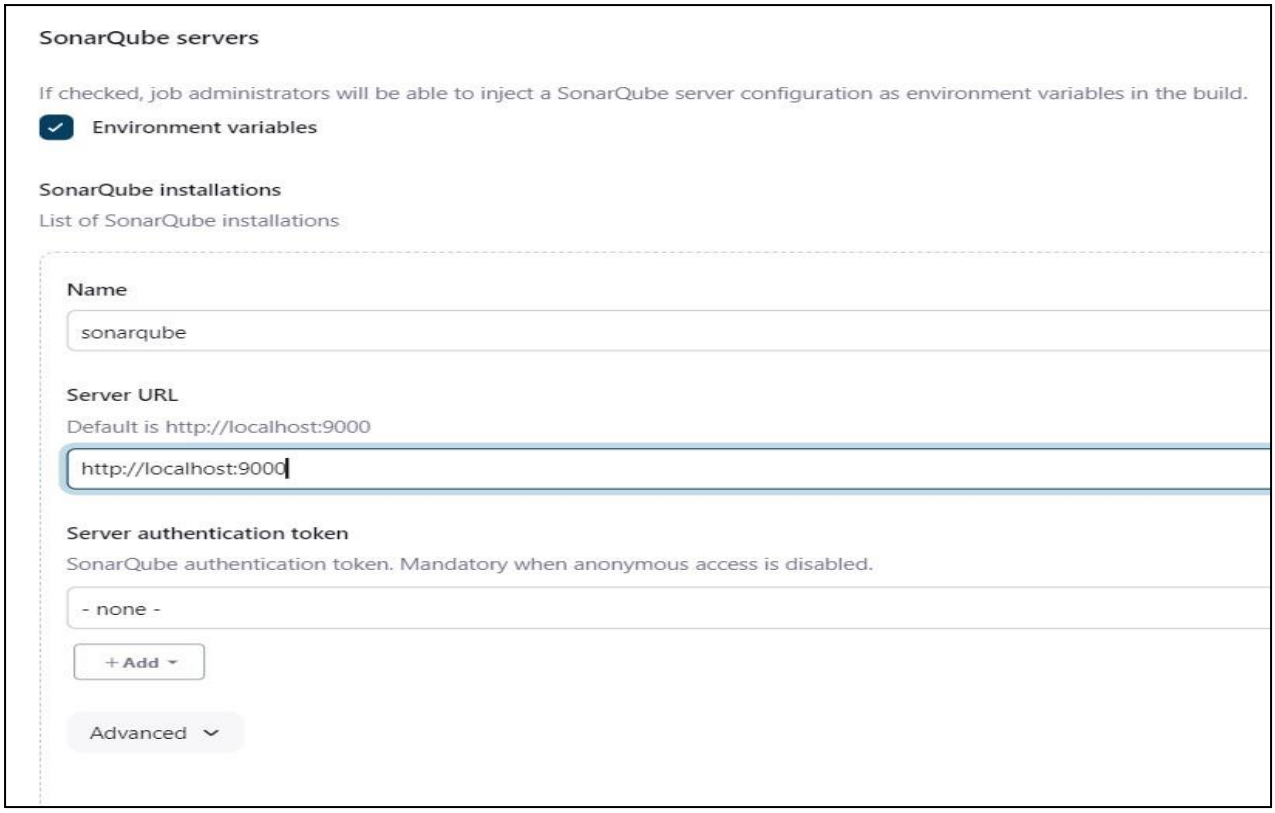




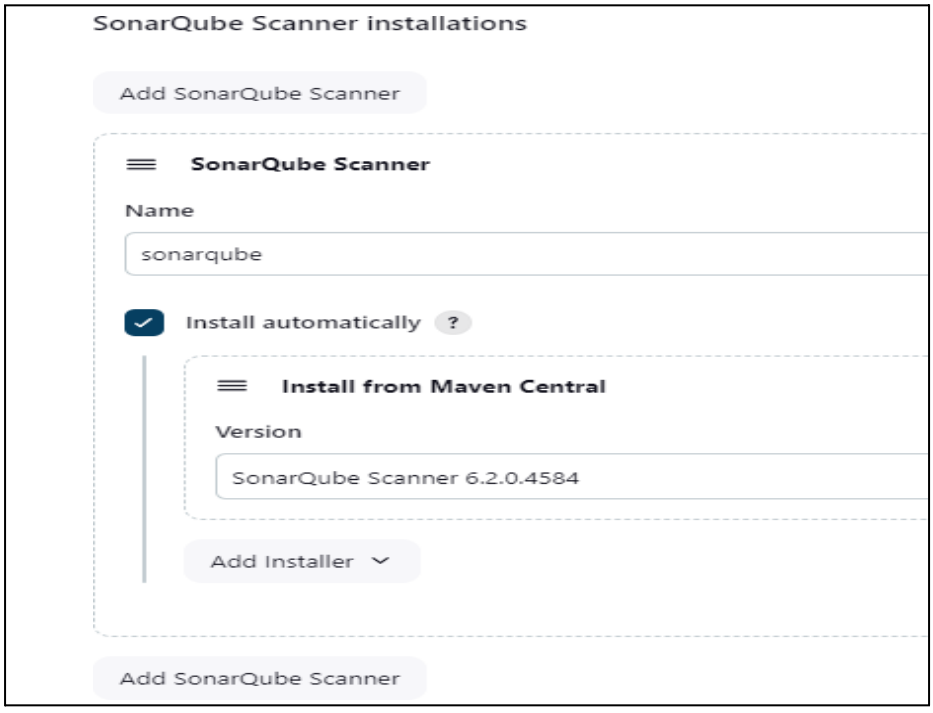
Step-5: Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins → Plugins and search for SonarQube Scanner in Available Plugins and install it.



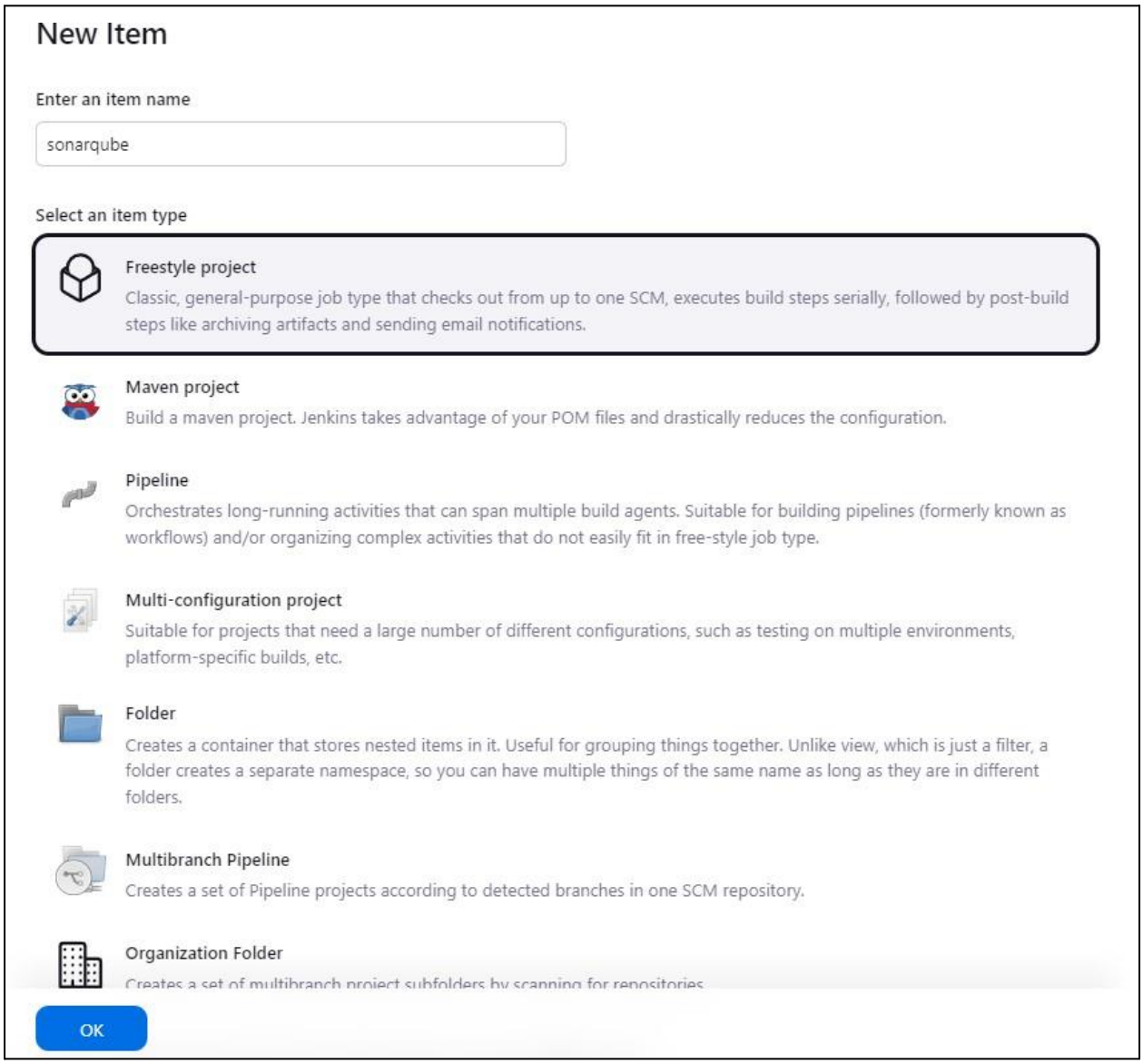
Step-6: Under 'Manage Jenkins → System', look for SonarQube Servers and enterthese details. Name : sonarqube, Server URL : http://localhost:9000



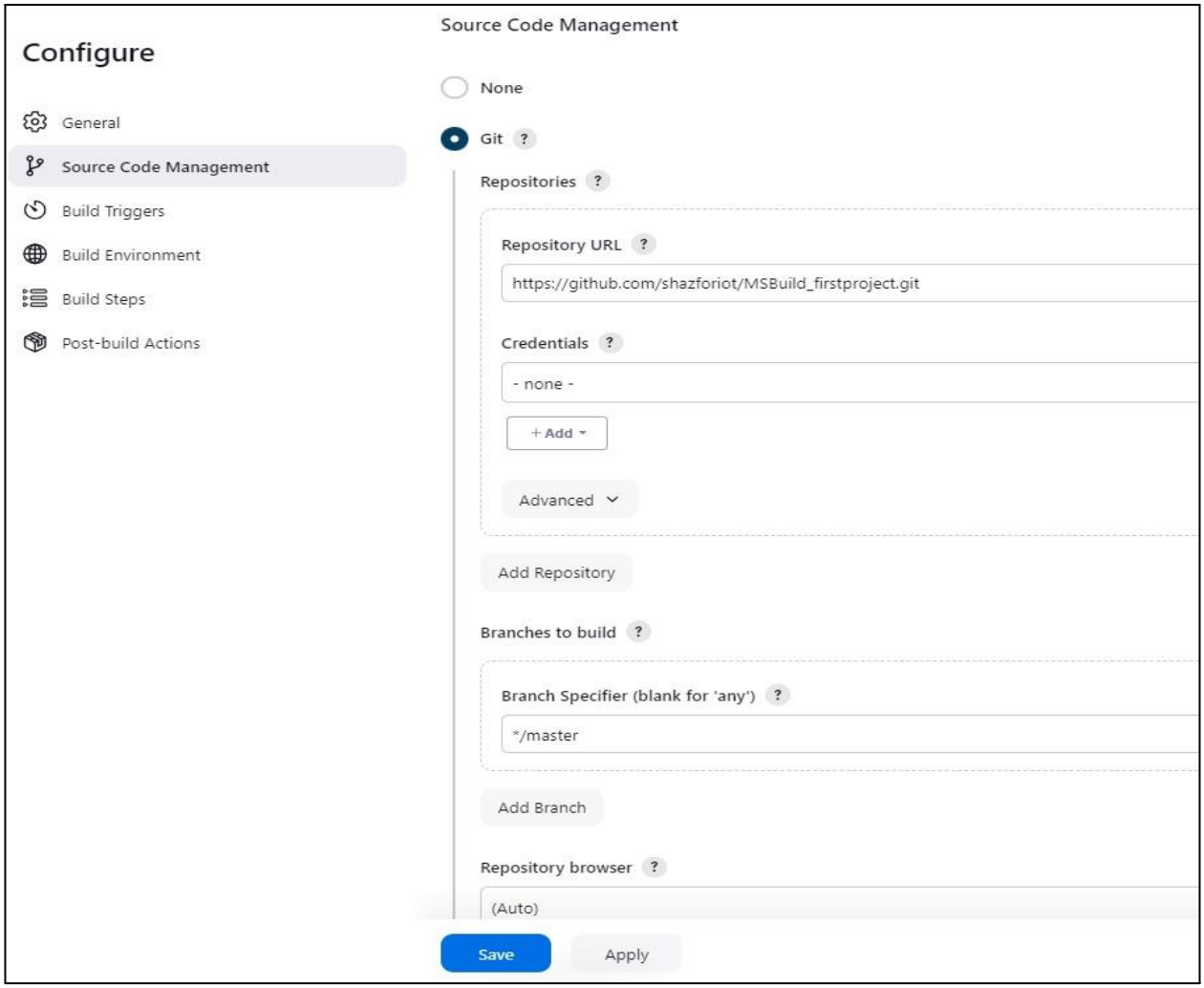
Step-7: Search for SonarQube Scanner under Global Tool Configuration. Choose thelatest configuration and choose Install automatically. Manage Jeknins → Tools → SonarQube Scanner Installation.



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



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



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

sonar.projectKey=sonarqube

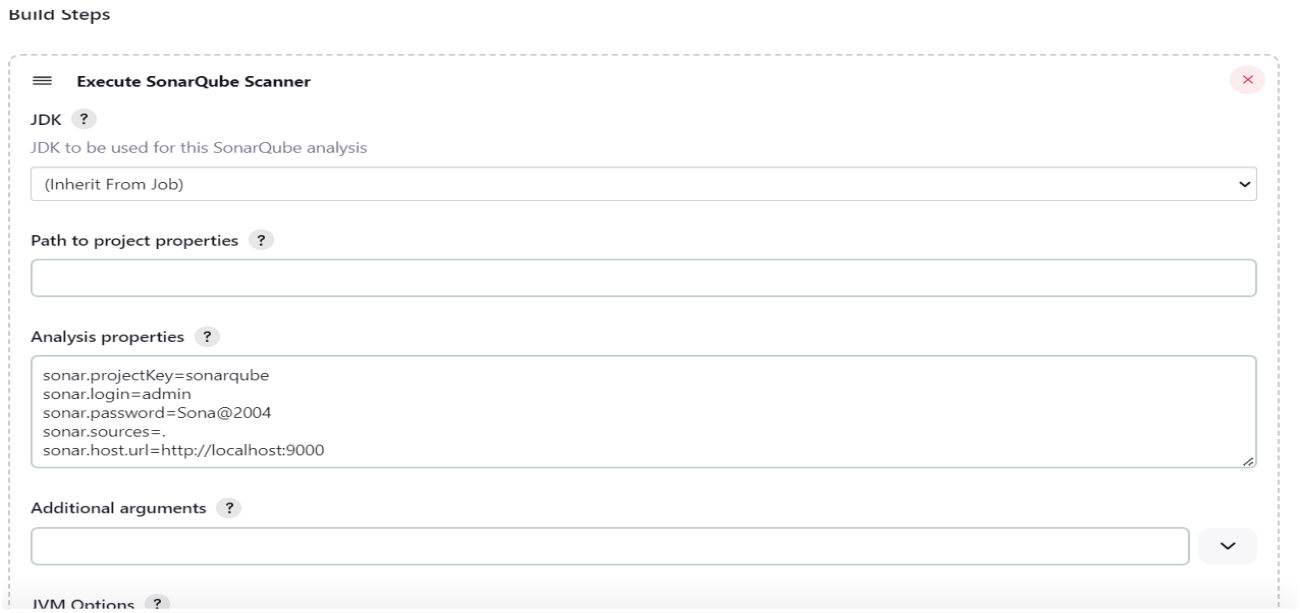
sonar.login=admin

sonar.password=Sona@2004

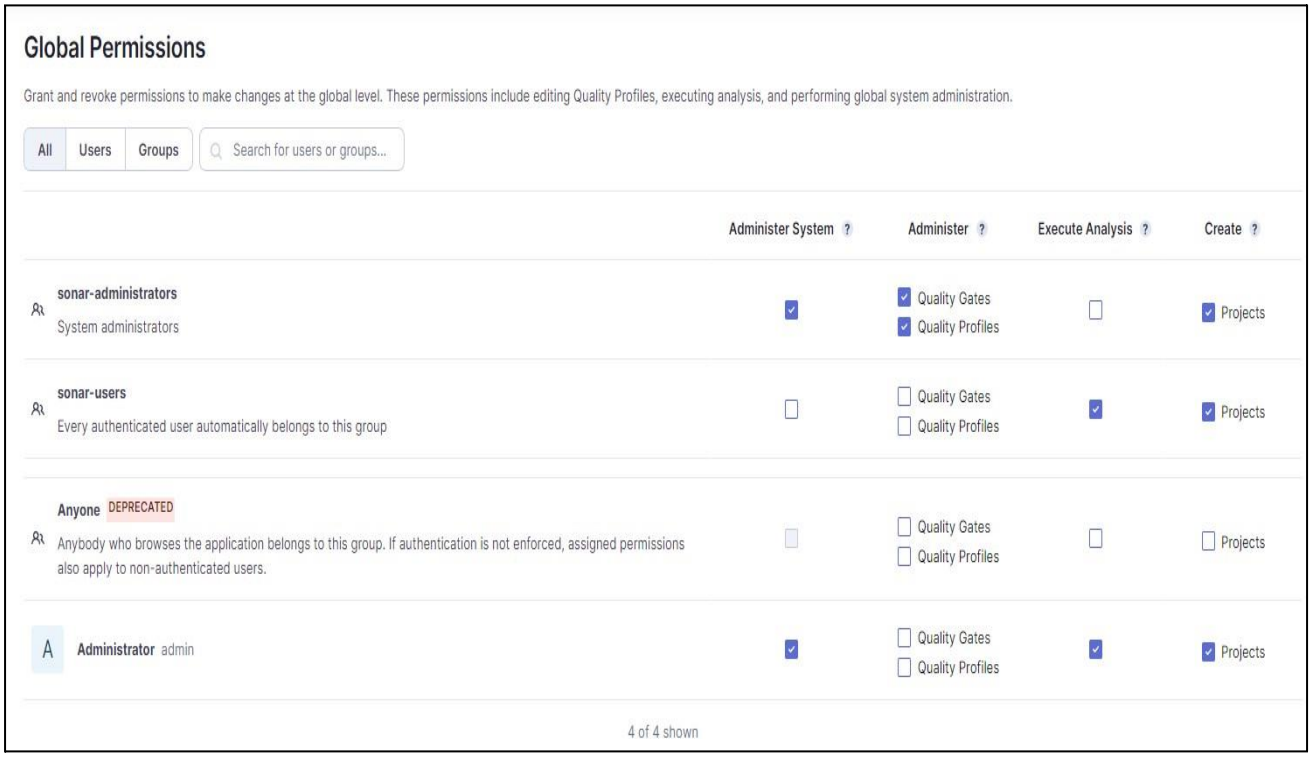
sonar.sources=.

sonar.host.url=http://localhost:9000

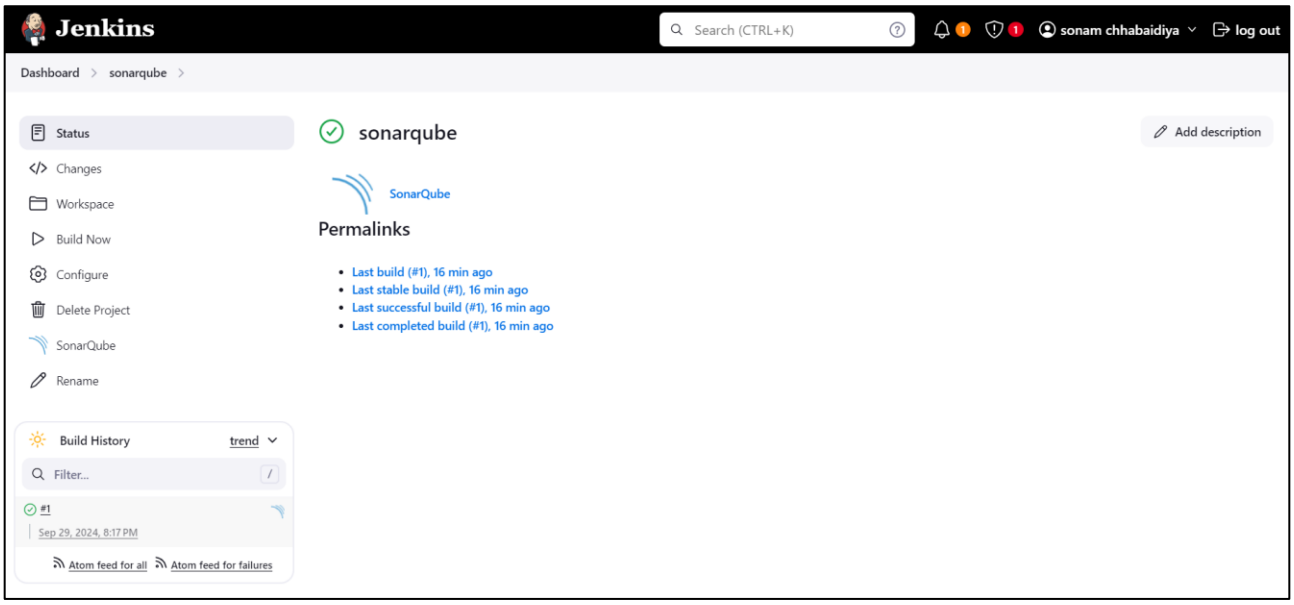
Sonam chhabaidiya D15A 09

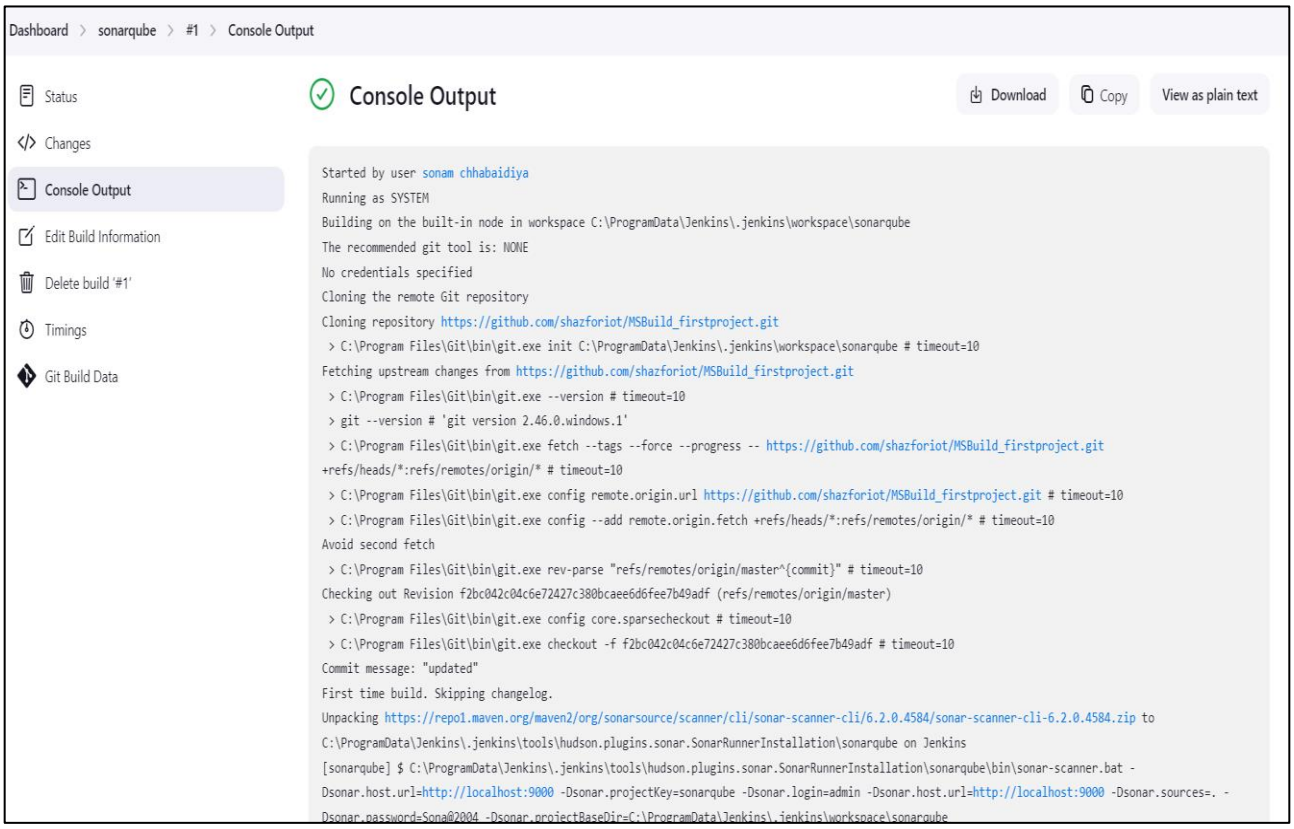


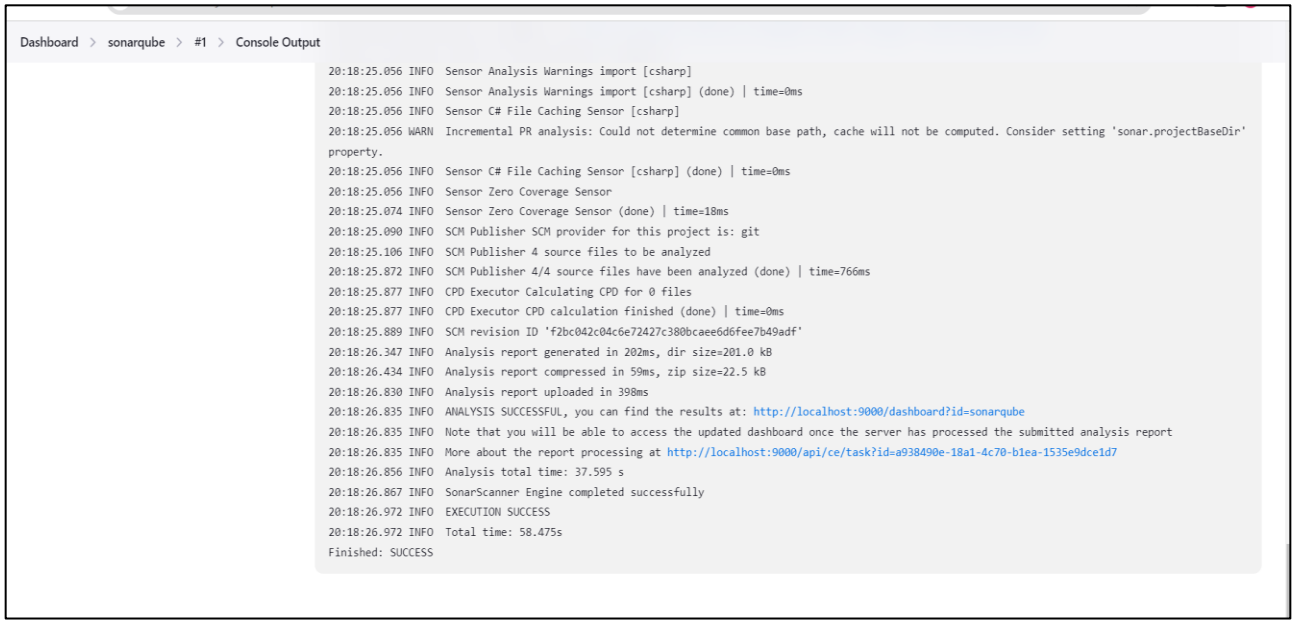
Step-11: Go to http://localhost:9000/admin/permissions and allow Execute Permissionsto the Admin user.



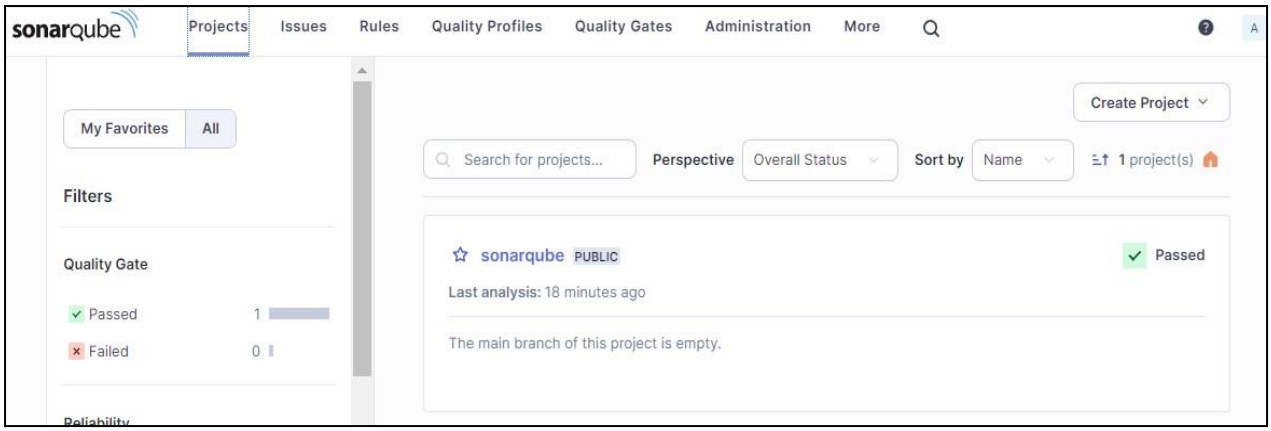
Step-12: Run The Build and check the console output.



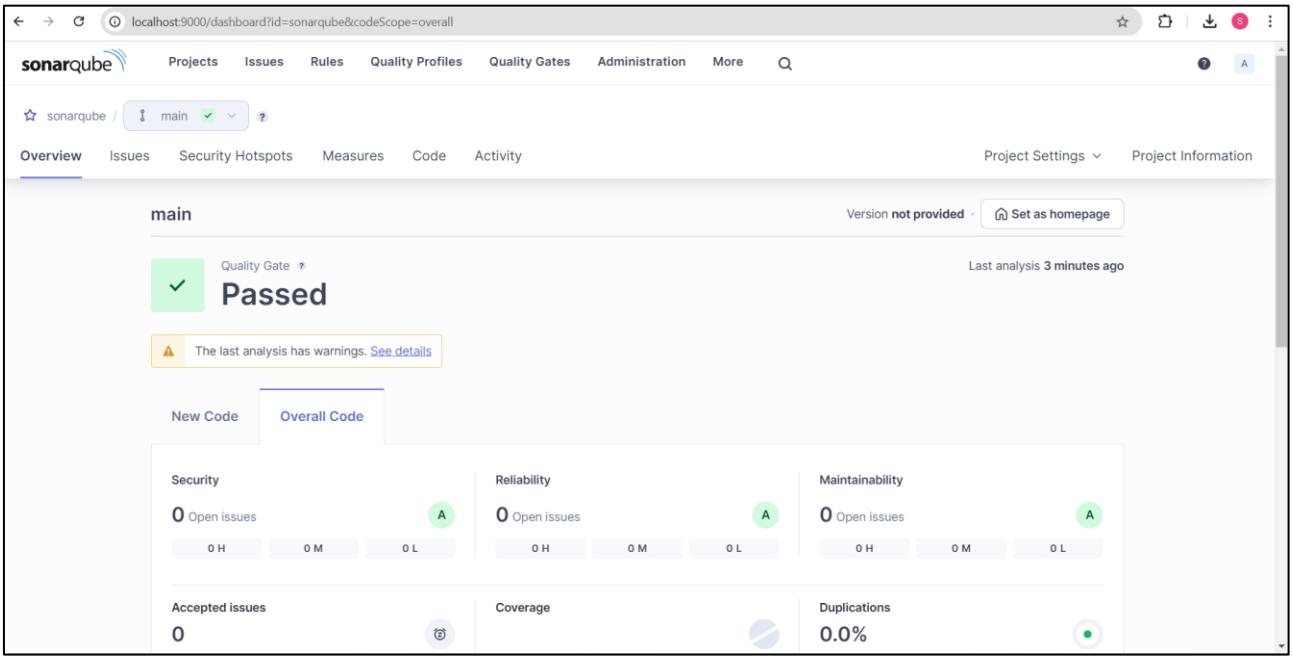




Step-13: Once the build is complete, check the project in SonarQube.



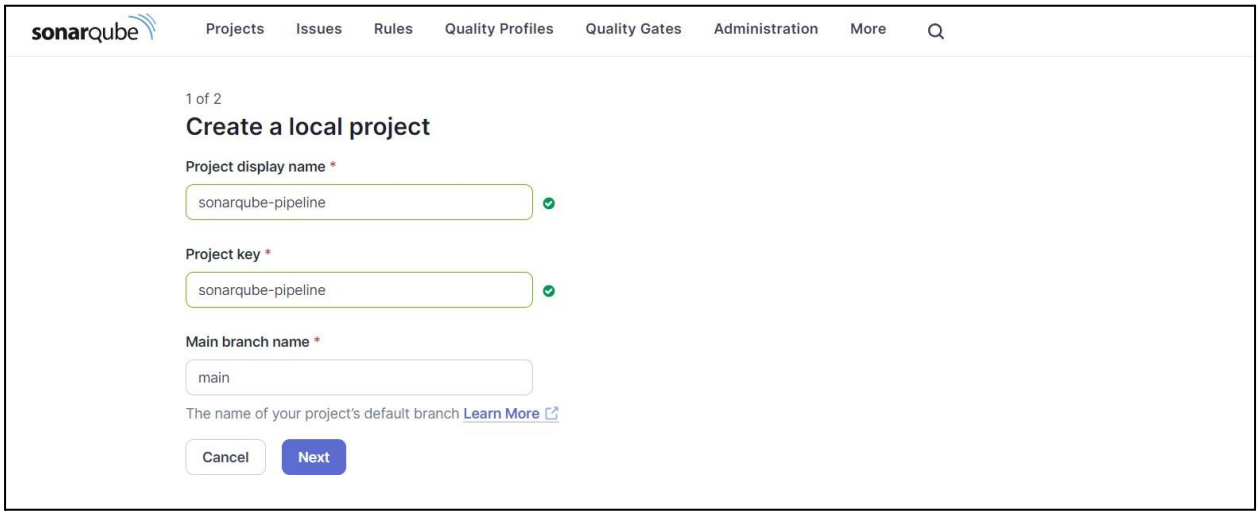
Sonam chhabaidiya D15A 09

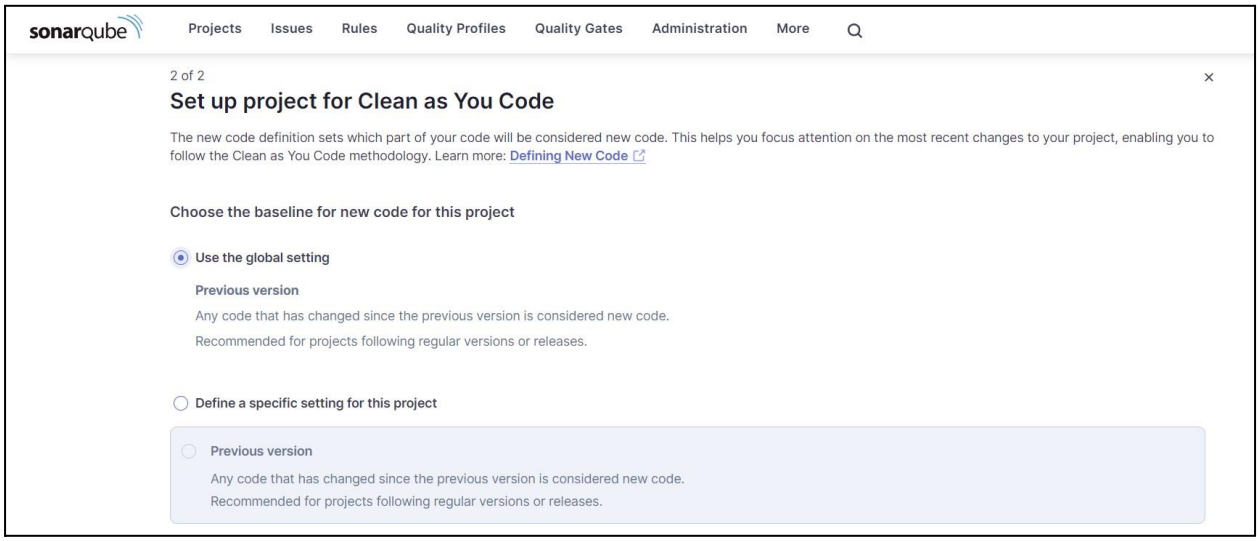


ADVANCE DEVOPS
EXPERIMENT 08

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

Step 1: Log in to sonarqube portal and create a local project.



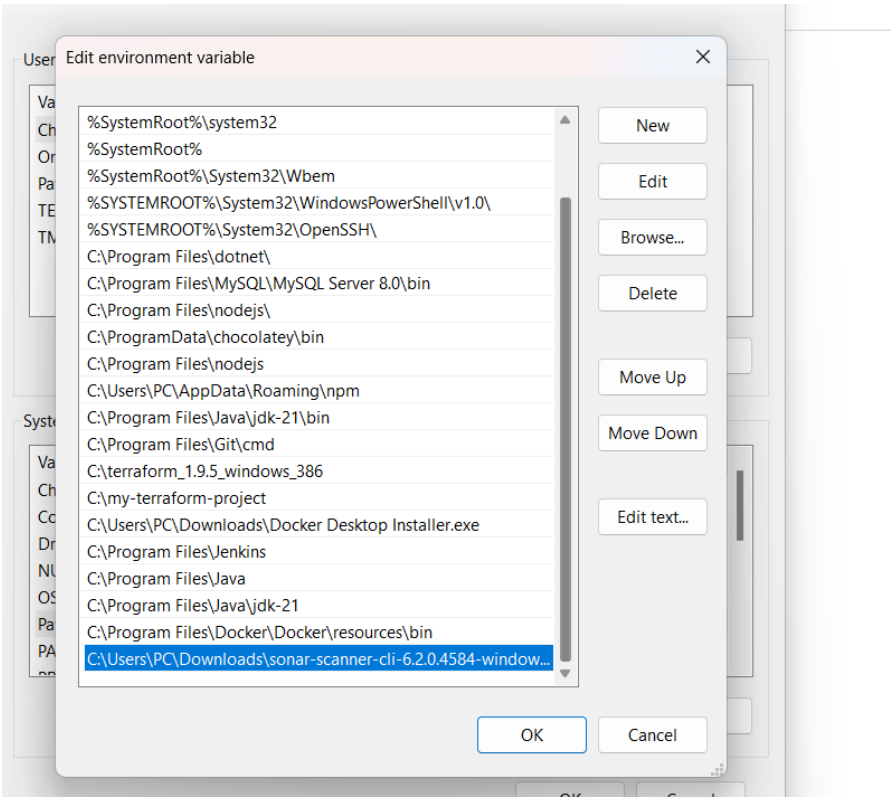


Step 2: Go to download sonarscanner to download sonar scanner

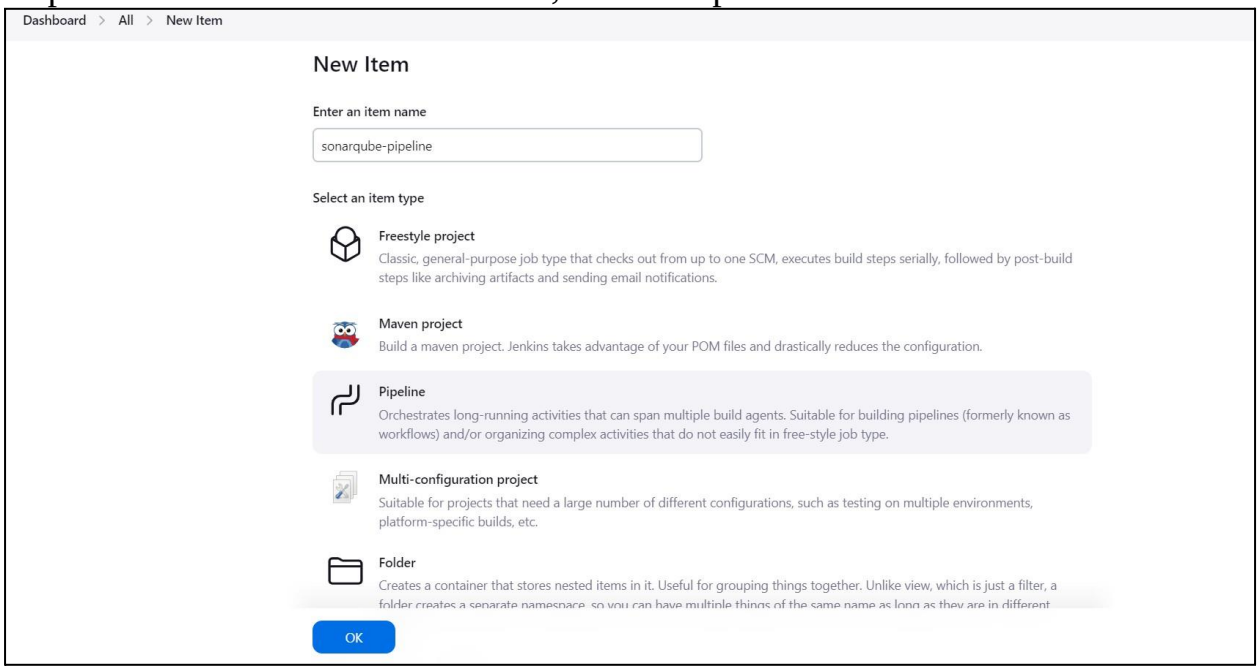
Sonam chhabaidiya D15A 09

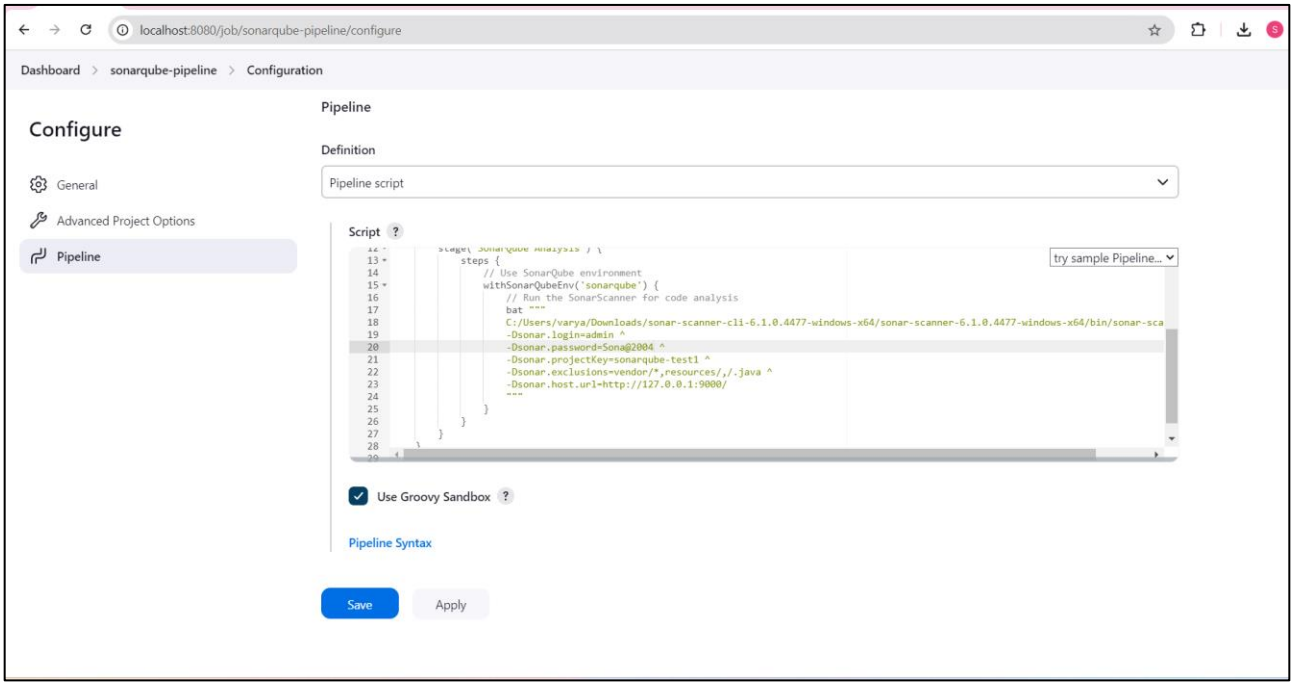
After the download is complete, extract the file and copy the path to bin folderGo to environment variables, system variables and click on path

Add a new path, paste the path copied earlier.

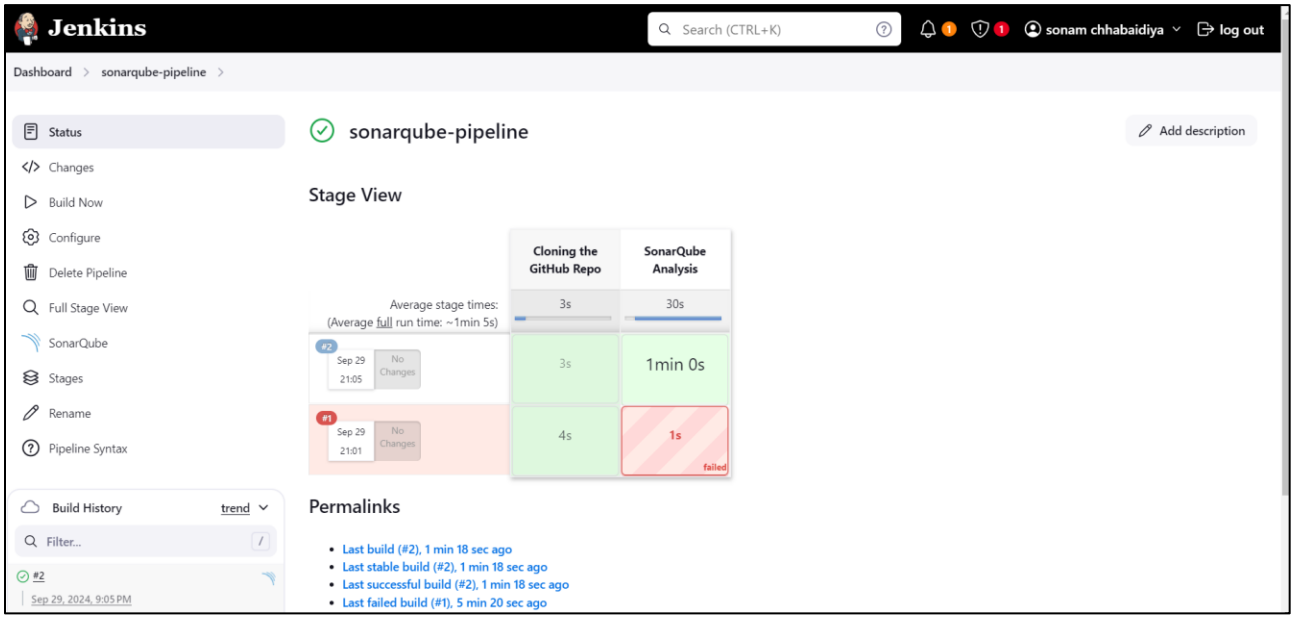


Step 3: Create a New Item in Jenkins, choose Pipeline.

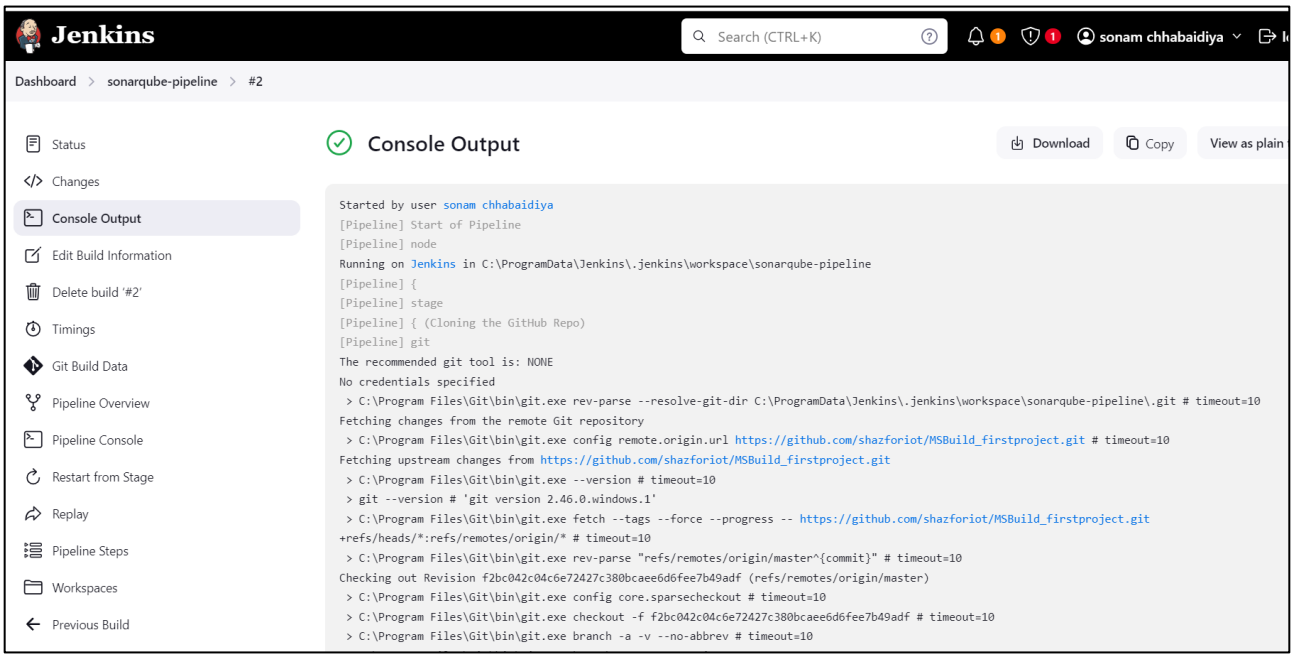




Step 4: Save the pipeline and build it.

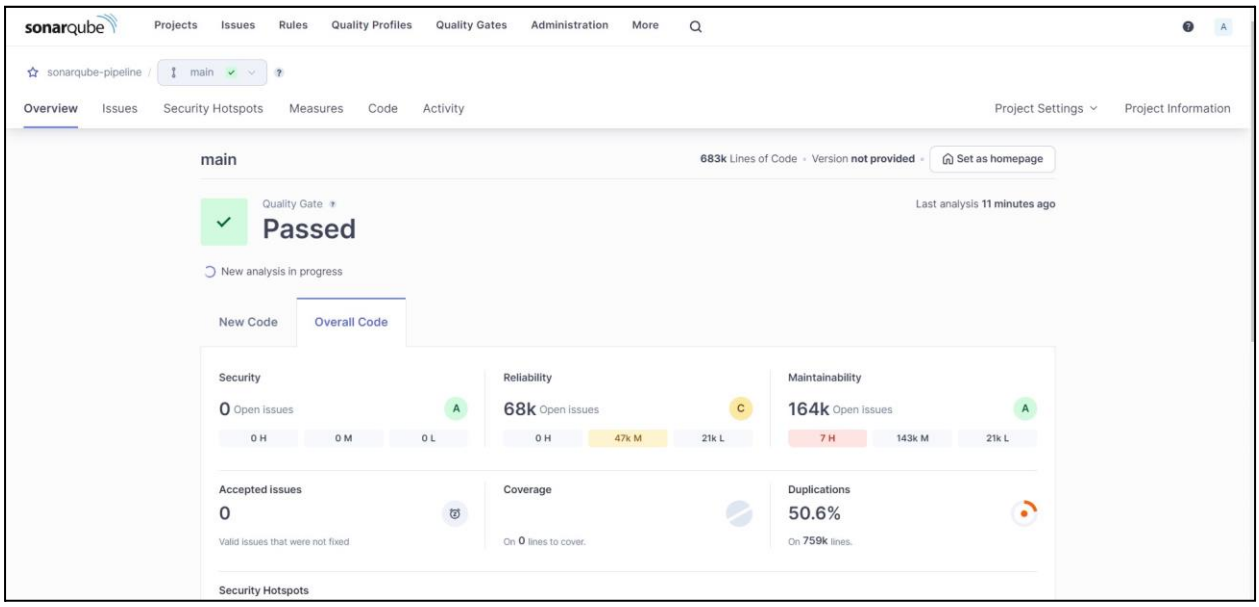


Console output:



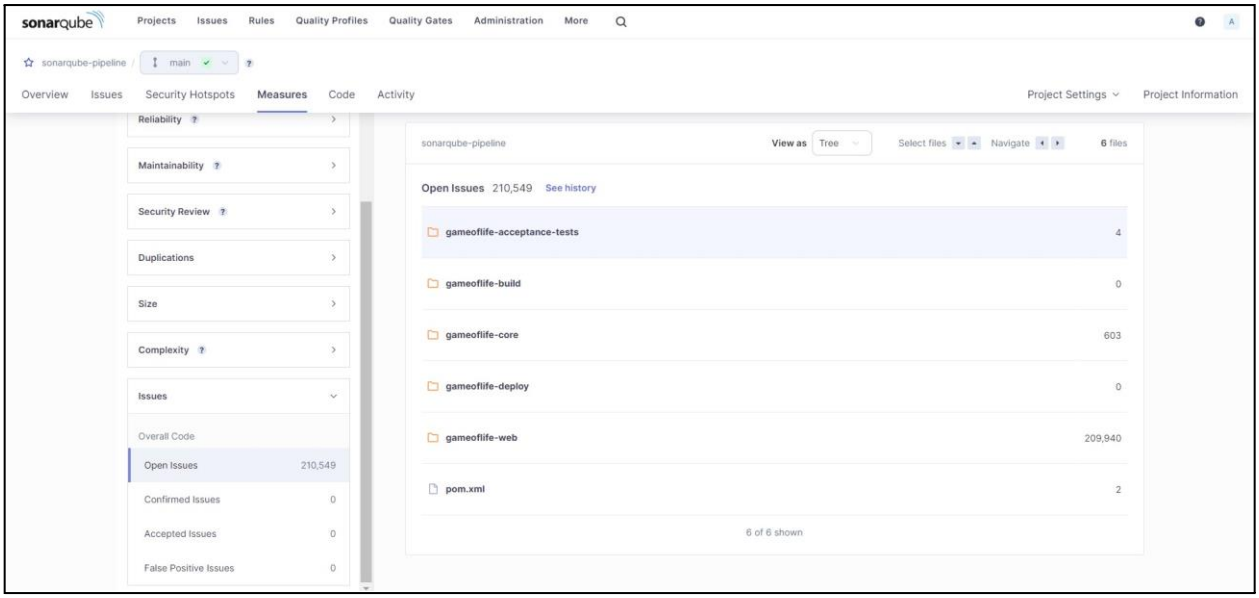
Sonam chhabaidiya D15A 09

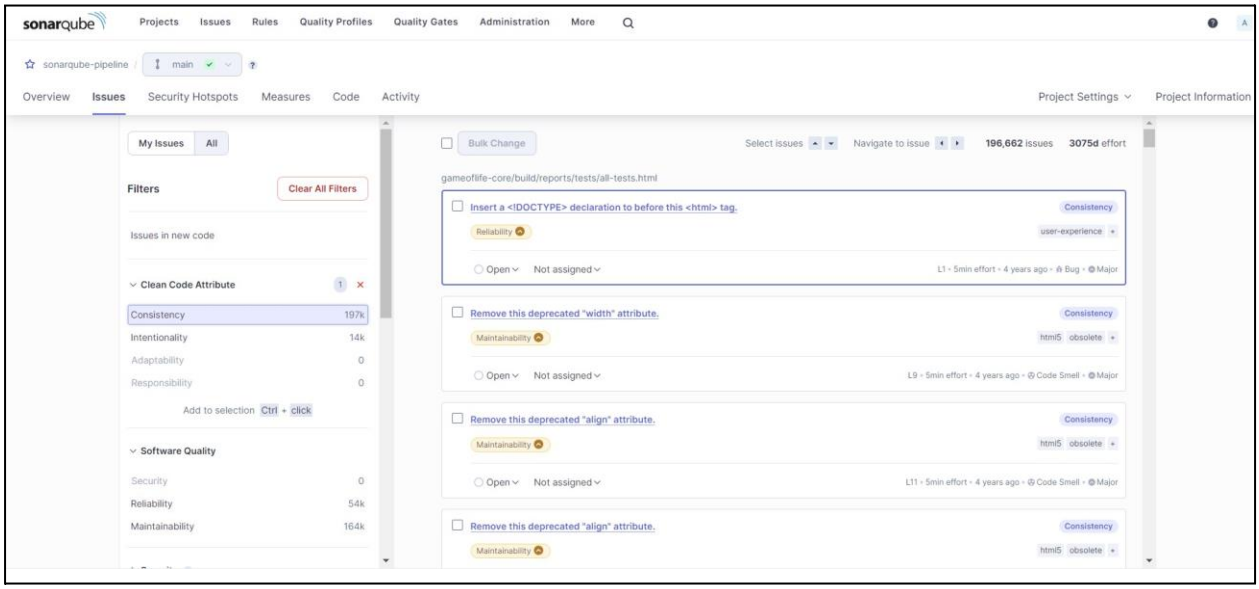
Step 5: After that, check the project in SonarQube



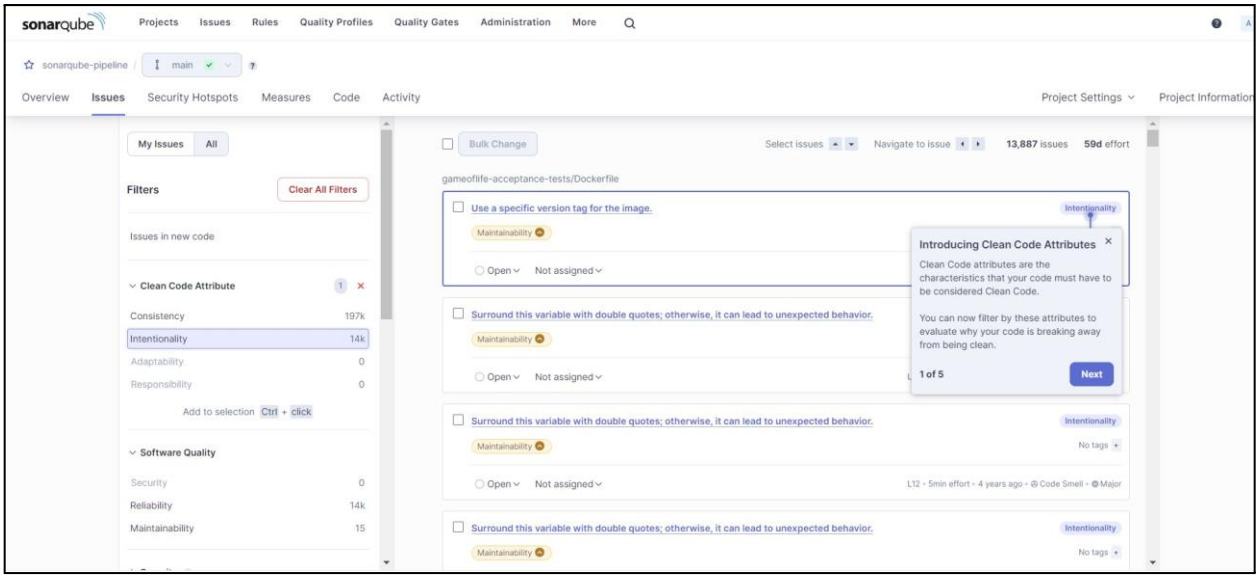
Under different tabs, check all different issues with the code.

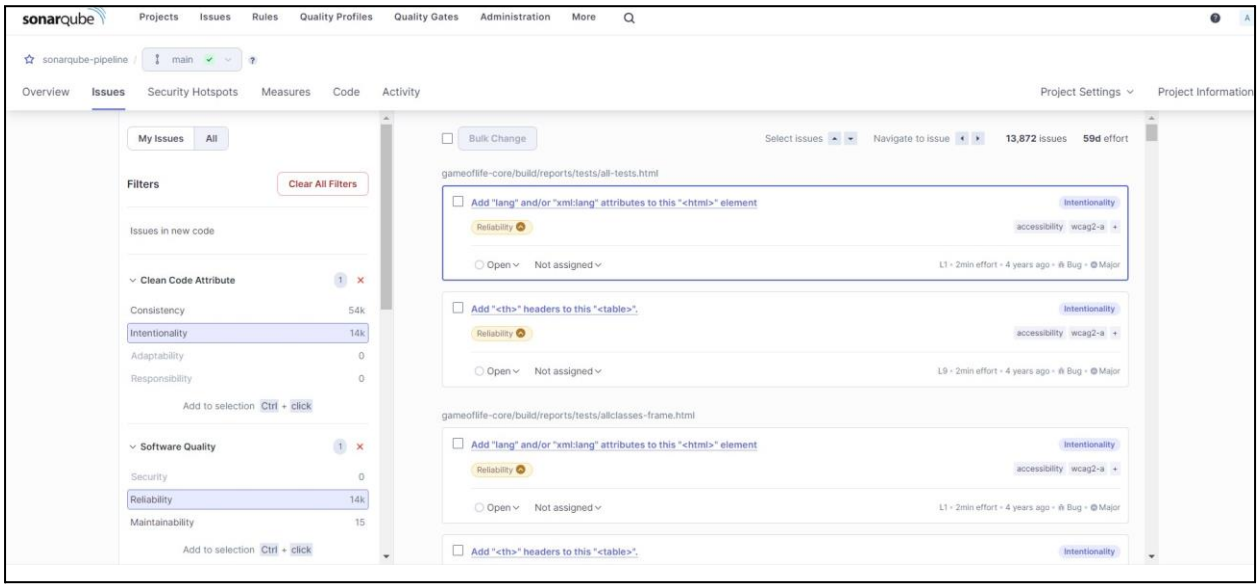
Sonam chhabaidiya D15A 09



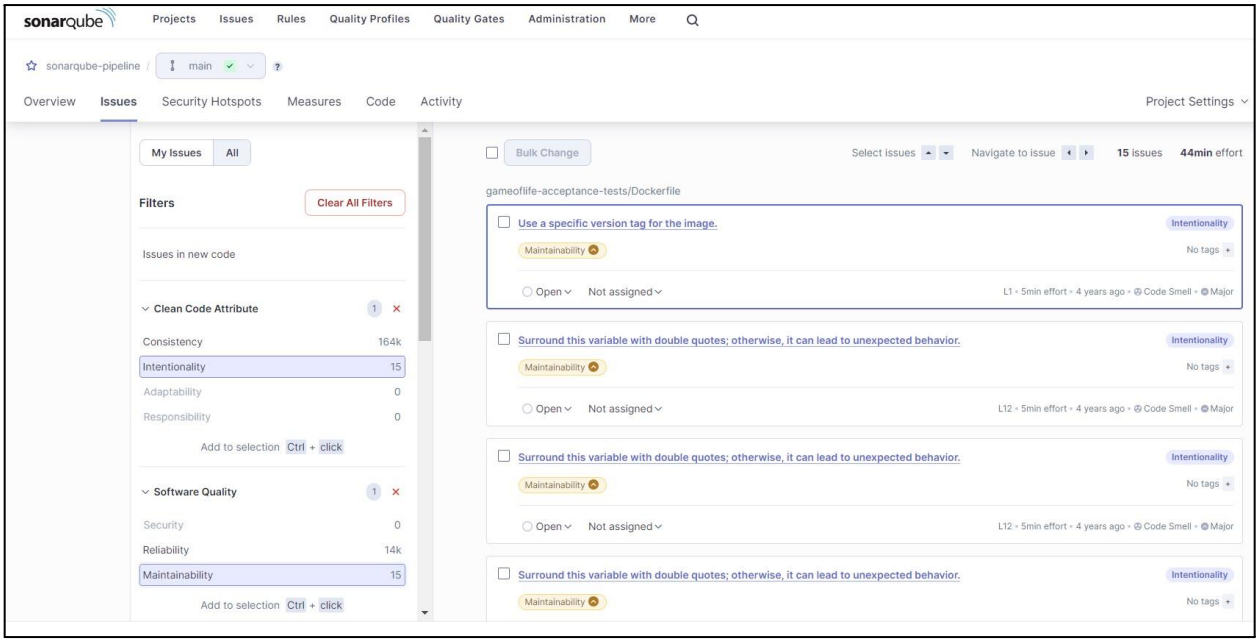


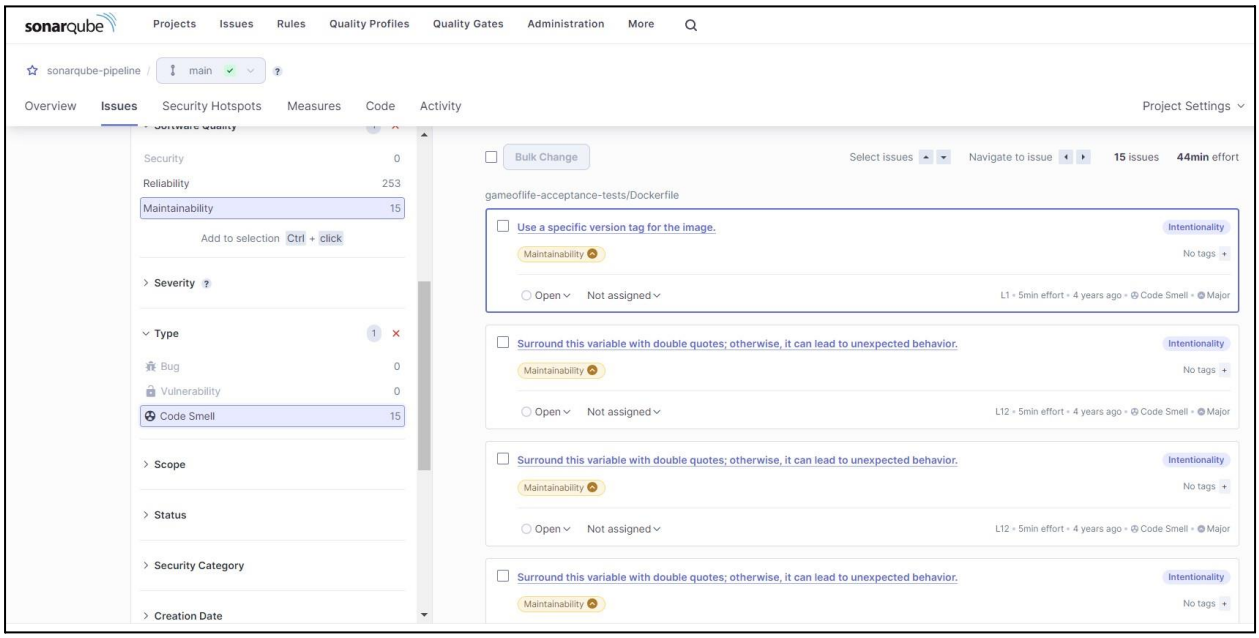
Sonam chhabaidiya D15A 09



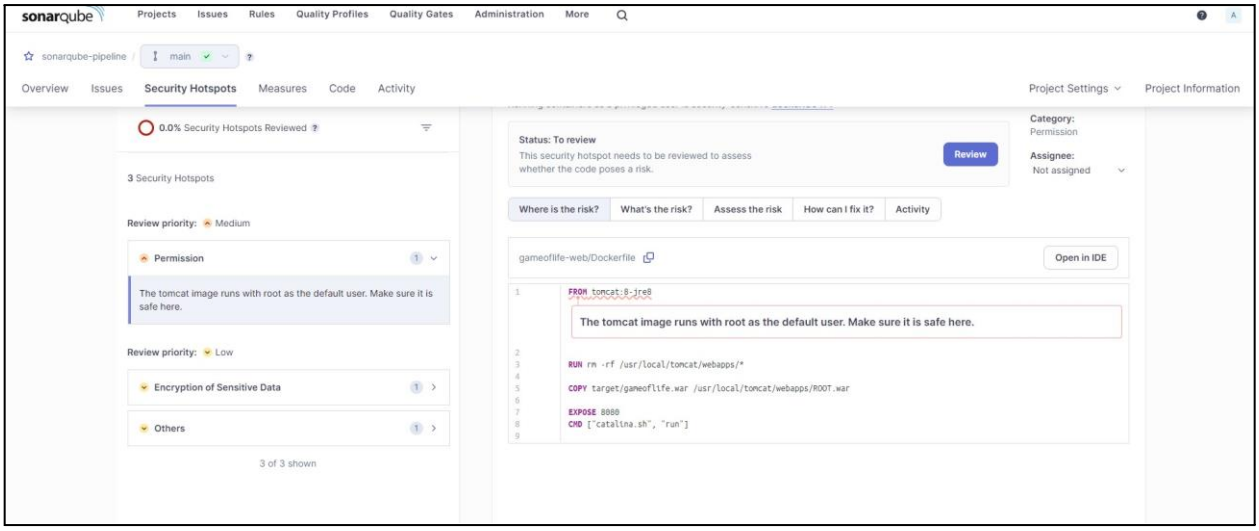


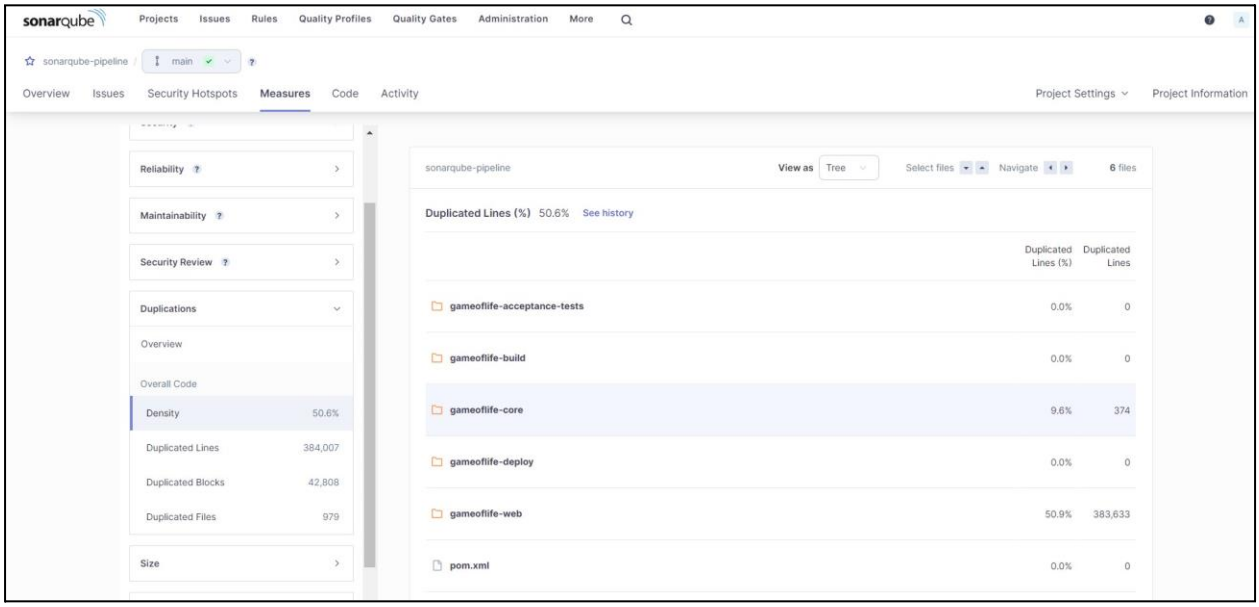
Sonam chhabaidiya D15A 09



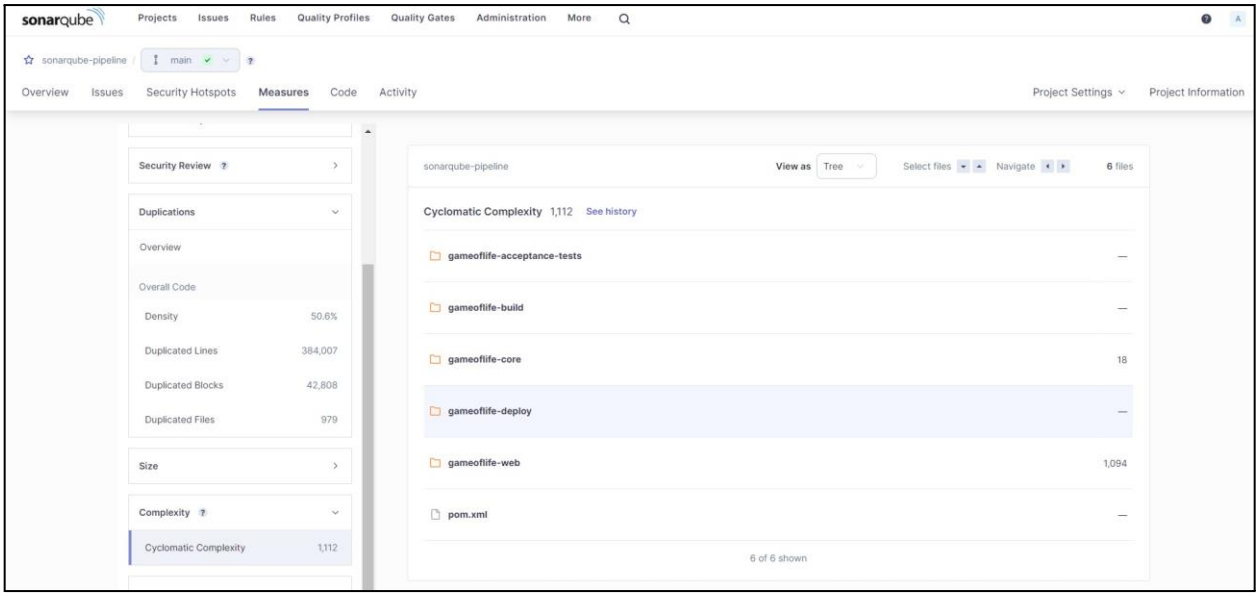


Sonam chhabaidiya D15A 09





Sonam chhabaidiya D15A 09

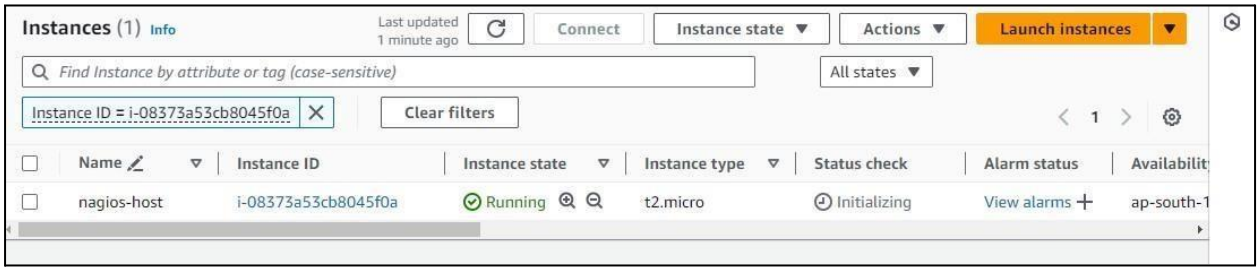


Sonam chhabaidiya D15A 09

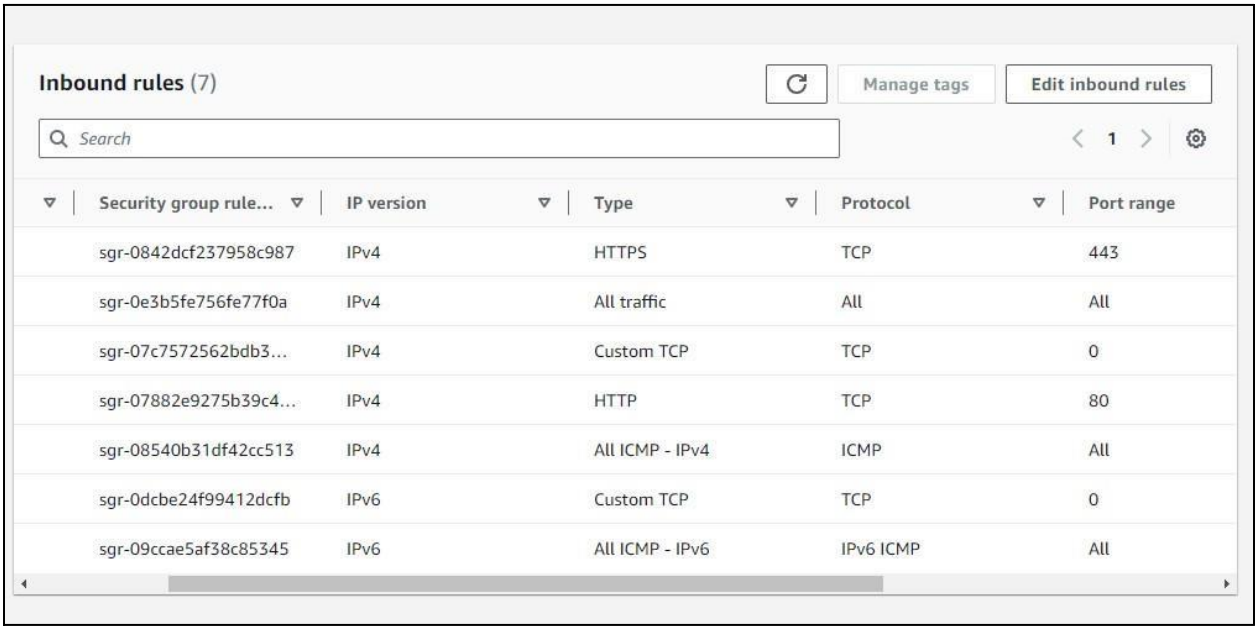
ADVANCE DEVOPS EXPERIMENT 9

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

Step 1: Create an Amazon Linux EC2 instance and name it as nagios-host



Step 2: Edit the following inbound rules of the specified security groups and ensure HTTP.HTTPS,SSH,ICMP are accessible from anywhere



Step 3:Connect to your EC2 instance via the connect option available in EC2 instances menu

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:23 ago on Thu Sep 26 08:42:17 2024.
Dependencies resolved.
```

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

Step 4: Update and install the required packages Use the following commands: **sudo yum update sudo yum install httpd php sudo yum install gcc glibc glibc-common sudo yum install gd gd-devel**

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:32 ago on Thu Sep 26 08:42:17 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

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

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:27 ago on Thu Sep 26 08:42:17 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k

Step 5:Create a new nagios user by writing the following commands **sudo adduser -m nagios sudo passwd nagios**

```
Complete!
[ec2-user@ip-172-31-33-14 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-33-14 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-33-14 ~]$
```

Step 6: Create a new user group using **sudo groupadd nagcmd** and Add users to the group using the following commands:

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
Complete!
[ec2-user@ip-172-31-33-14 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-33-14 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-33-14 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-33-14 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-33-14 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-33-14 ~]$ mkdir downloads
[ec2-user@ip-172-31-33-14 ~]$ cd downloads
[ec2-user@ip-172-31-33-14 downloads]$ wget https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz/download?use
_mirror=excellmedia
--2024-09-26 09:15:54--  https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz/download?use_mirror=excellmedi
a
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?ts=gAAAAABm9SZKLFW7LwD1QAJ2jNzqmSJwAPA1mQ-
eAJYK8z5NmrV_ifVkhbsV-qOfPsLUyICC6yvdHu6UeeIyvNZsVGUtir9BeQ%3D%3D&use_mirror=excellmedia&r= [following]
```

Step 7: Create a directory for Nagios downloads using the following commands-

Commands - **mkdir ~/downloads**

cd ~/downloads

Also download Nagios and plugin source files

Commands - **wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz**

```
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-26 09:38:43--  https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://excellmedia.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1 [following]
--2024-09-26 09:38:45--  https://excellmedia.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1
Resolving excellmedia.dl.sourceforge.net (excellmedia.dl.sourceforge.net)... 202.153.32.19, 2401:fb00:0:1fe:8000::5
Connecting to excellmedia.dl.sourceforge.net (excellmedia.dl.sourceforge.net)|202.153.32.19|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz                 100%[=========================================================================>]   1.72M  8.14MB/s    in 0.2s

2024-09-26 09:38:45 (8.14 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

[ec2-user@ip-172-31-33-14 downloads]$ ls
'download?use_mirror=excellmedia'   nagios-4.0.8.tar.gz
[ec2-user@ip-172-31-33-14 downloads]$ tar -xzf nagios-4.0.8.tar.gz
[ec2-user@ip-172-31-33-14 downloads]$
```

Step 8-Extract the nagios source file with the following commands
tar zxvf nagios-4.4.6.tar.gz cd nagios-4.4.6

Then run the configuration script with the following command
/configure --with-command-group=nagcmd

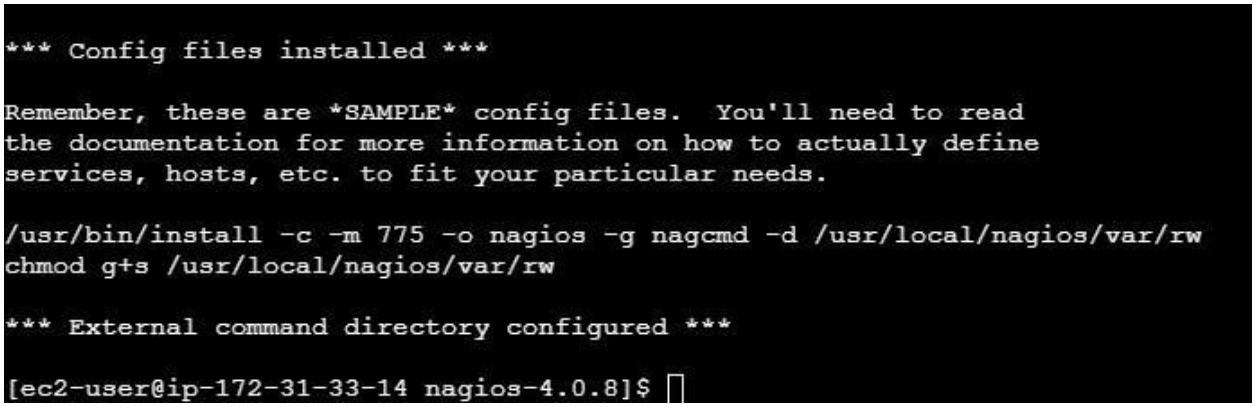
```
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  ${prefix}/var/nagios.lock
   Check result directory:  ${prefix}/var/spool/checkresults
          Init directory:  /etc/rc.d/init.d
  Apache conf.d directory:  /etc/httpd/conf.d
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                HTML URL:  http://localhost/nagios/
                 CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/bin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.

[ec2-user@ip-172-31-33-14 nagios-4.0.8]$
```

Step 9-Compile the source code with the following commands make all

Step 10-Install binaries,init script and sample config files

Commands -

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



Step 11-Edit the Config File to Change the Email Address Commands -

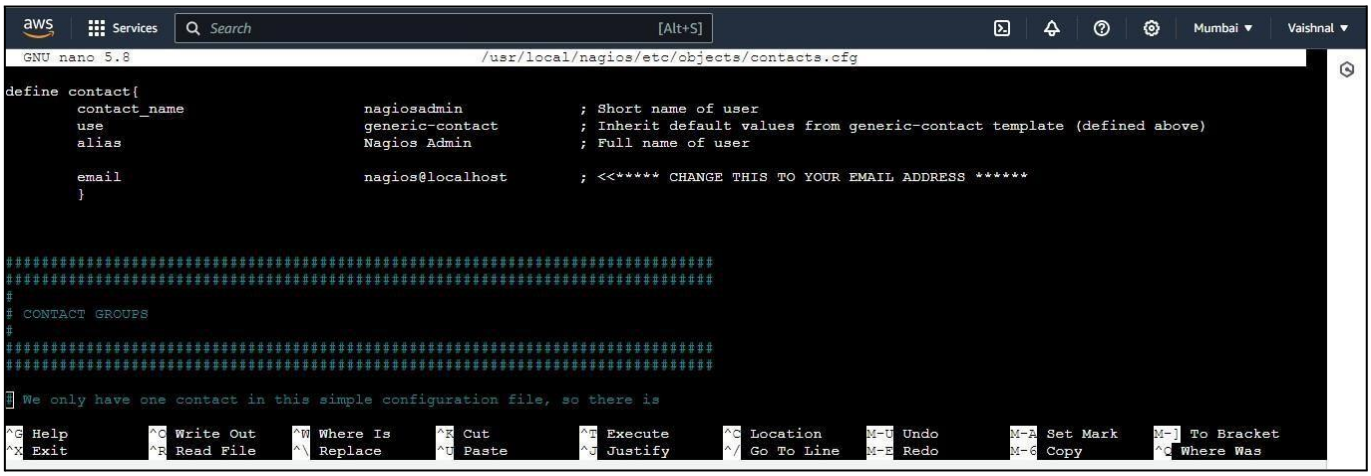
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

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

Step 12-Configure the Web Interface

Commands - **sudo make install-webconf**

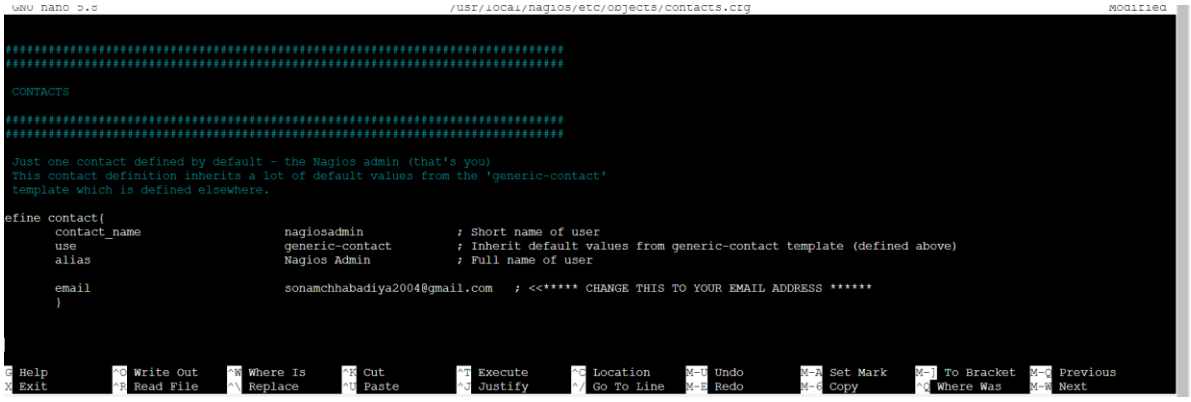
Sonam chhabaidiya D15A 09



Step 13-Create a Nagios Admin Account

Commands -

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin ● You will be prompted to enter and confirm the password for the nagiosadmin user



Step 14-. Extract the Plugins Source File

Commands - **cd**

~/downloads

tar zxvf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

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

[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf

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

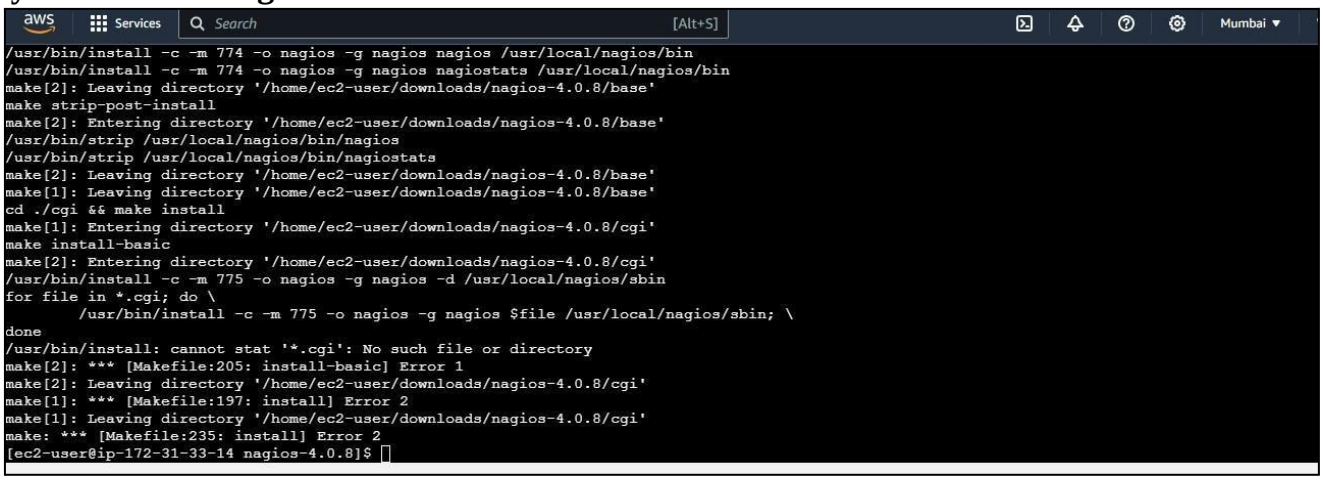
Step 15-19. Compile and Install Plugins

Commands -

./configure --with-nagios-user=nagios --with-nagios-group=nagios make
sudo make install

```
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

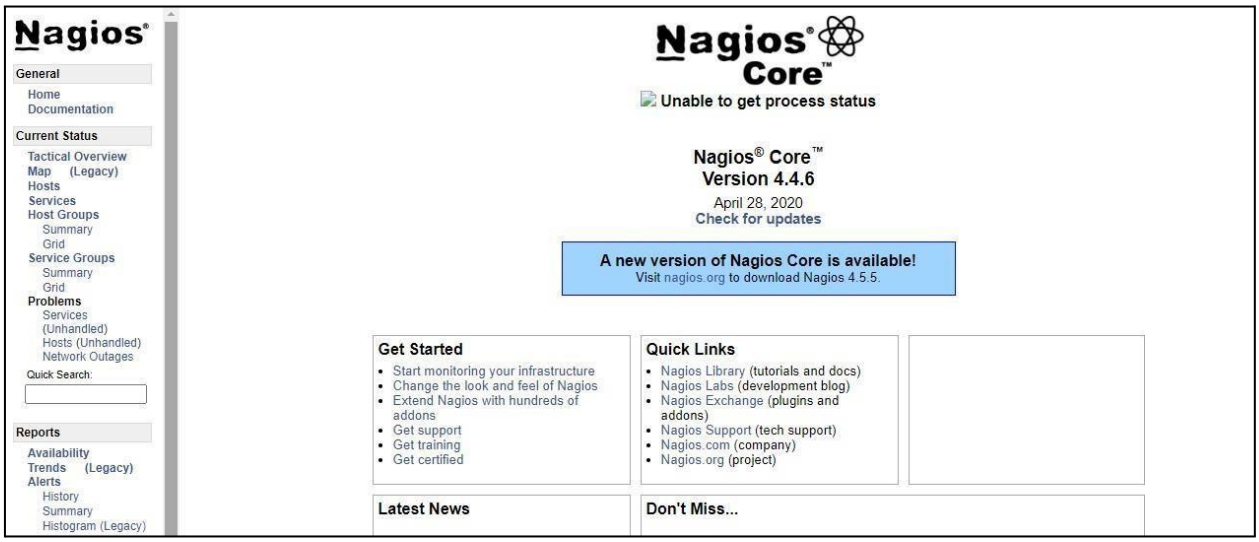
Step 16-Start Nagios
Commands - **sudo chkconfig**
--add nagios sudo chkconfig
nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg sudo
systemctl start nagios



Step 17-Access Nagios Web Interface

- Copy the Public IP address of your EC2 instance.
- Open your browser and navigate to http:///nagios.
- Enter the username nagiosadmin and the password you set in Step 16.

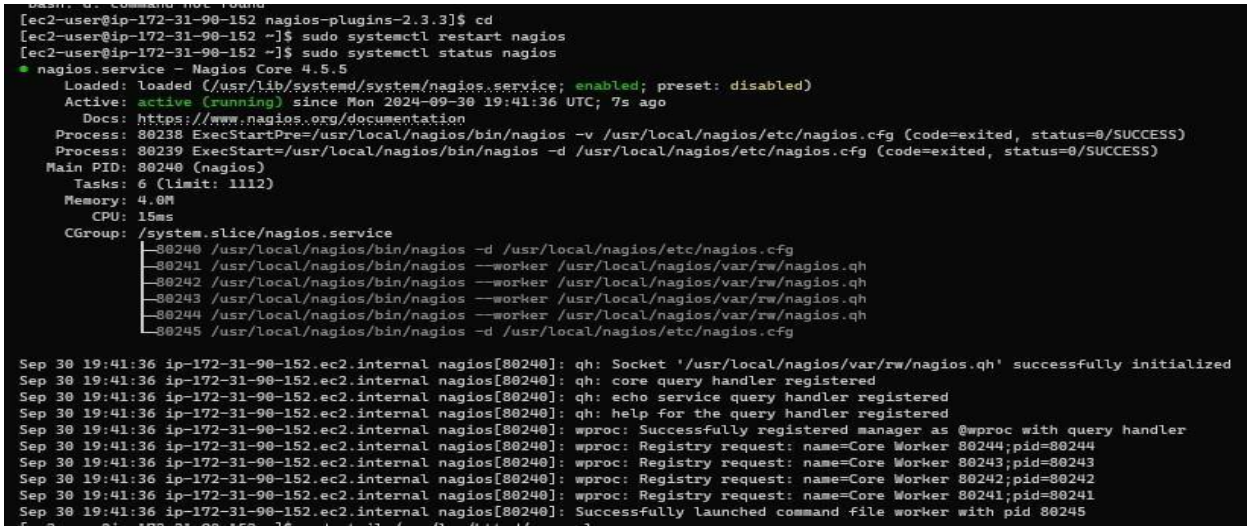
Sonam chhabaidiya D15A 09



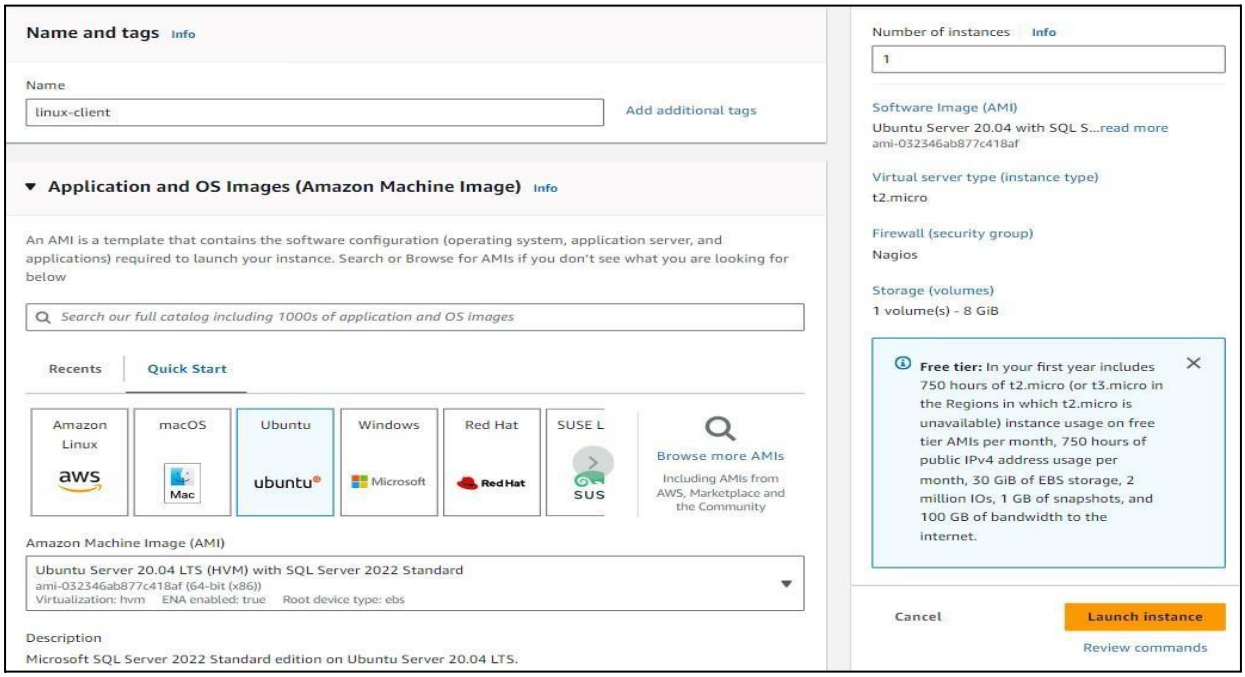
ADVANCE DEVOPS EXP-10

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

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



Step-2. Create an Ubuntu 20.04 Server EC2 Instance



Step-3: Verify Nagios Process on the Server

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

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

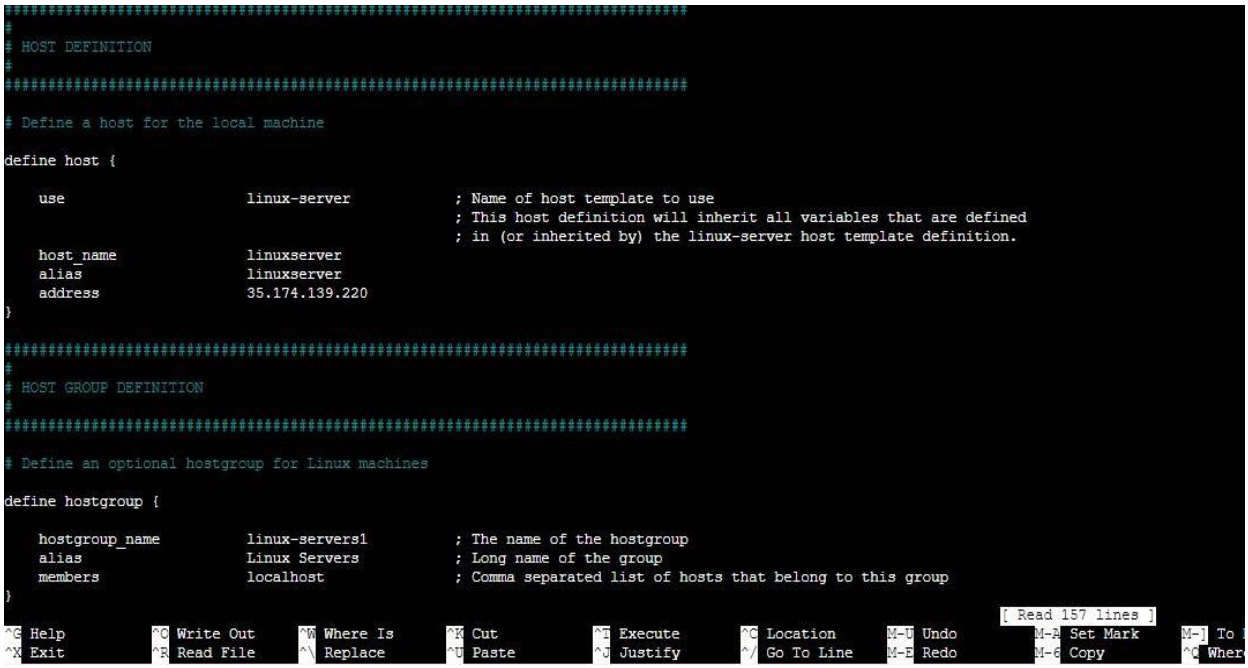
```
[ec2-user@ip-172-31-80-215 nagios-plugins-2.3.3]$ sudo su
[root@ip-172-31-80-215 nagios-plugins-2.3.3]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-215 nagios-plugins-2.3.3]# cp /usr/local/nagios/etc/objects/localhost.cfg
cp: missing destination file operand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

i-0ae1aae975bae3b7a (nagios-host)

Step-5: Edit the Configuration File

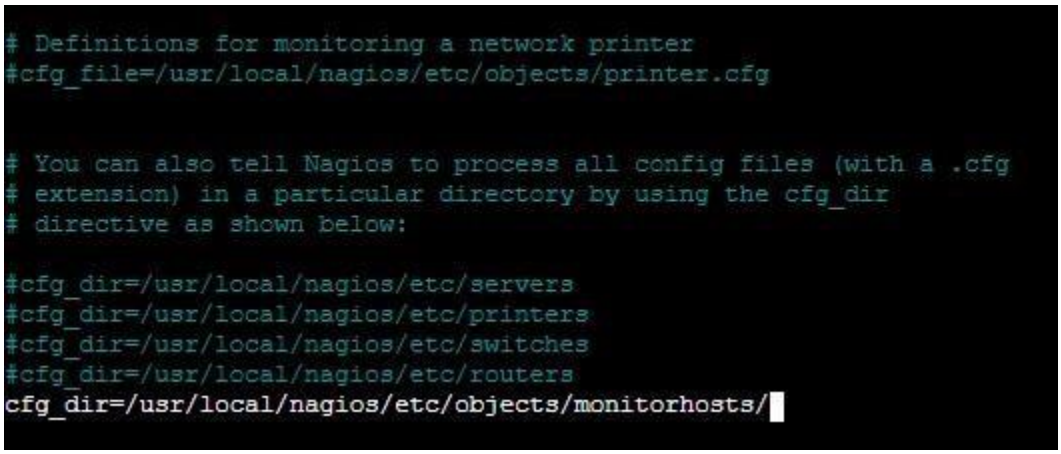
sudo nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

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



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

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



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

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

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0
```

Step-8: Restart Nagios Service sudo
systemctl restart nagios

Step-9:. SSH into the Client Machine

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

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

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

Step-11: Edit NRPE Configuration File

Commands -

sudo nano /etc/nagios/nrpe.cfg

Add your Nagios host IP address under `allowed_hosts`:

`allowed_hosts=<Nagios_Host_IP>`

```
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,35.174.139.220



# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed.  This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments

dont_blame_nrpe=0
```

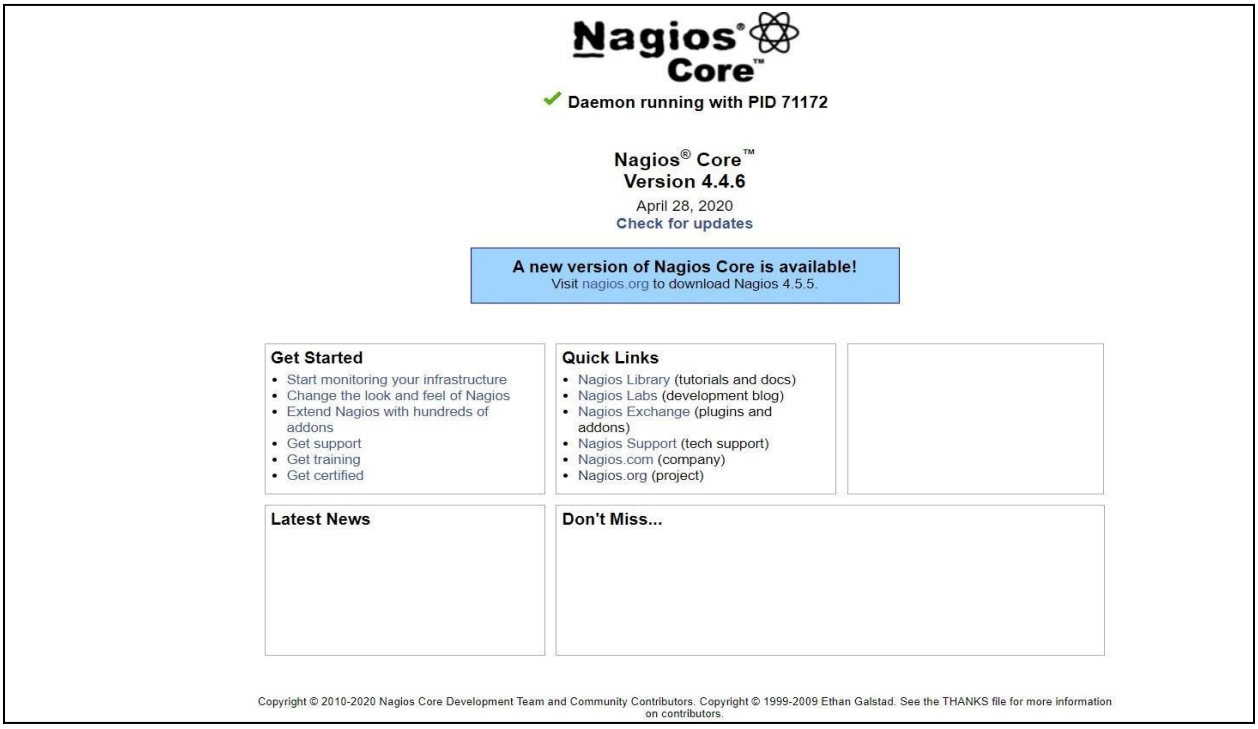
Step-12: Restart NRPE Server

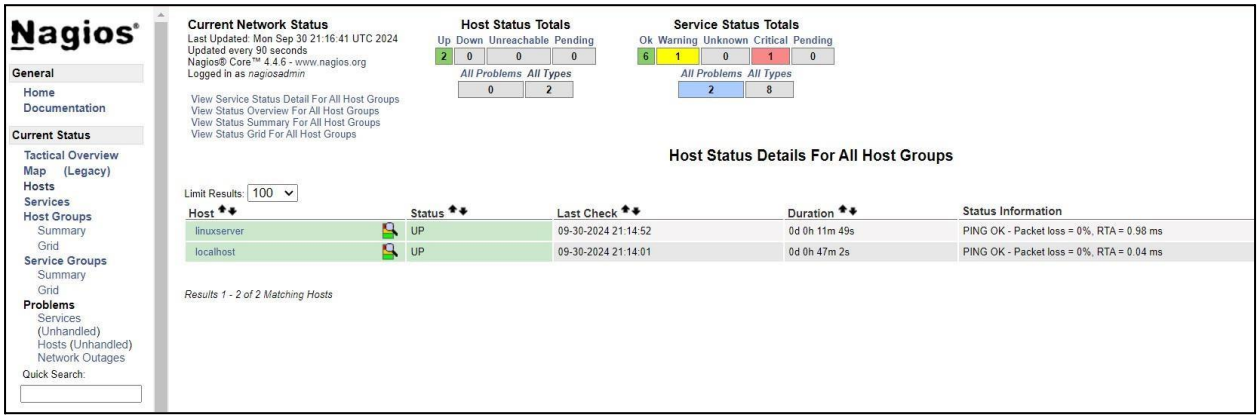
Commands -
sudo systemctl restart nagios-nrpe-server

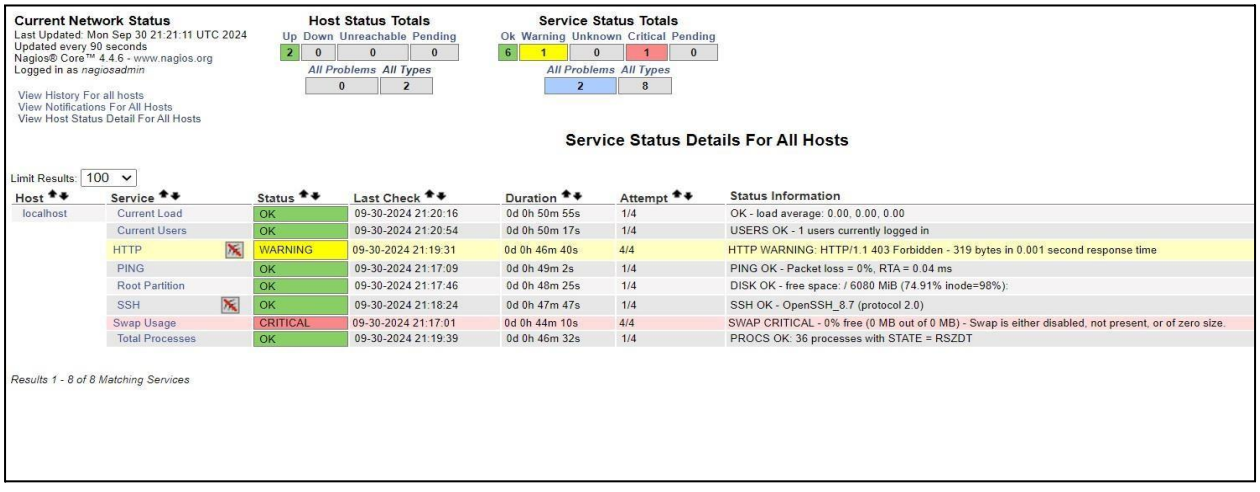
Step-13: Check Nagios Dashboard

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

Sonam Chhabaidiya D15A 09







Sonam Chhabaidiya D15A09

ADVANCED DEVOPS
Experiment 11

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

Theory:

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS).

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

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

Features of AWS Lambda

● AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.

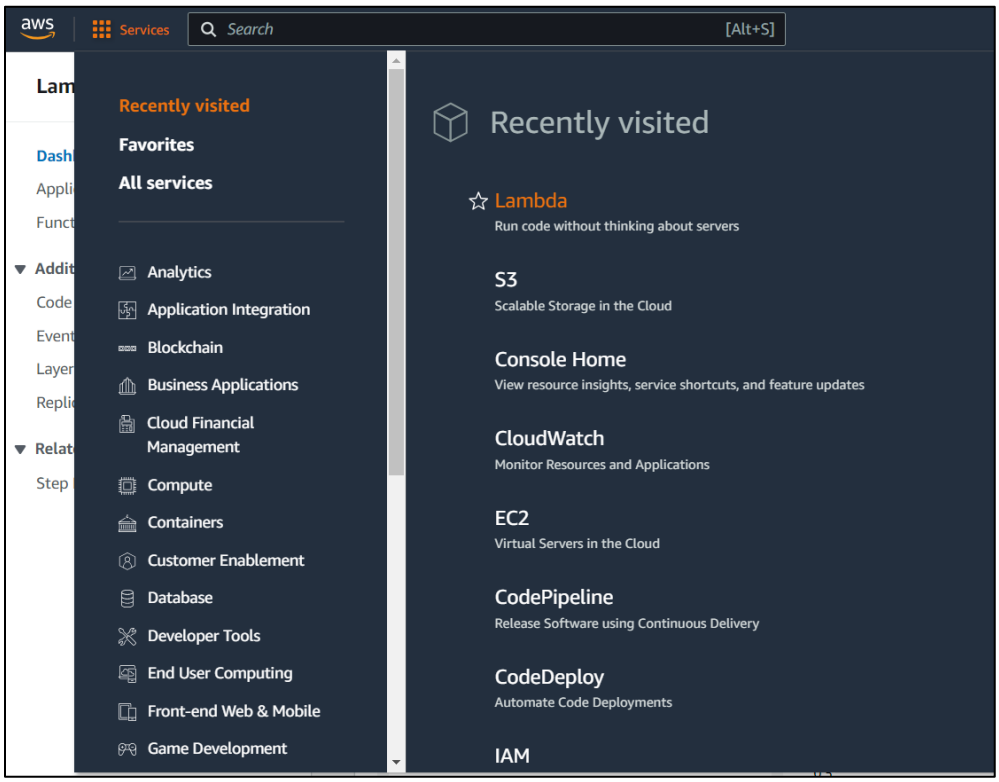
● It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.

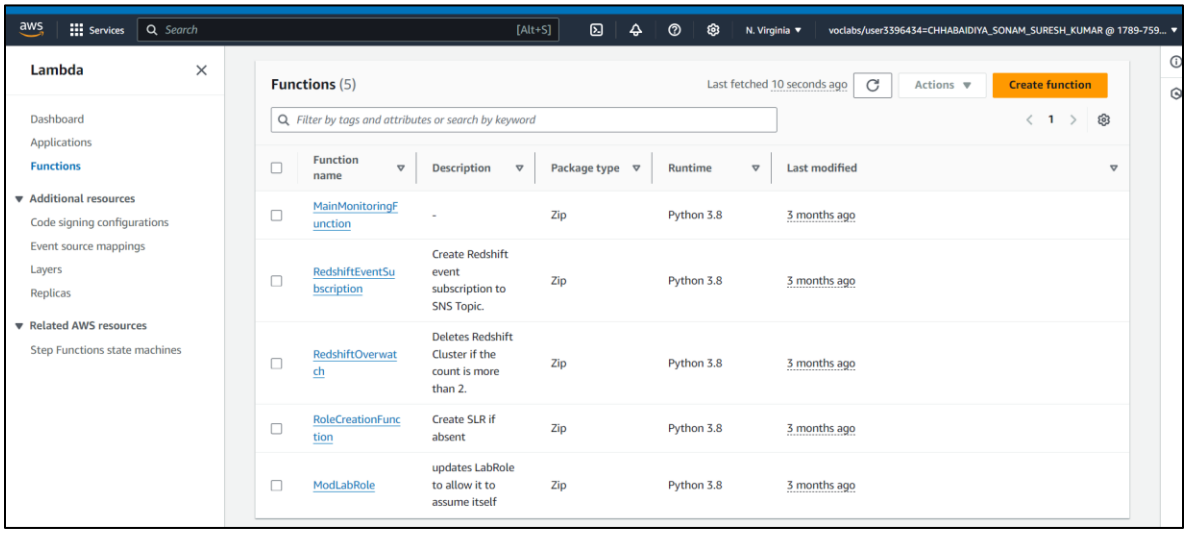
● You don’t need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.

● AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.

● It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

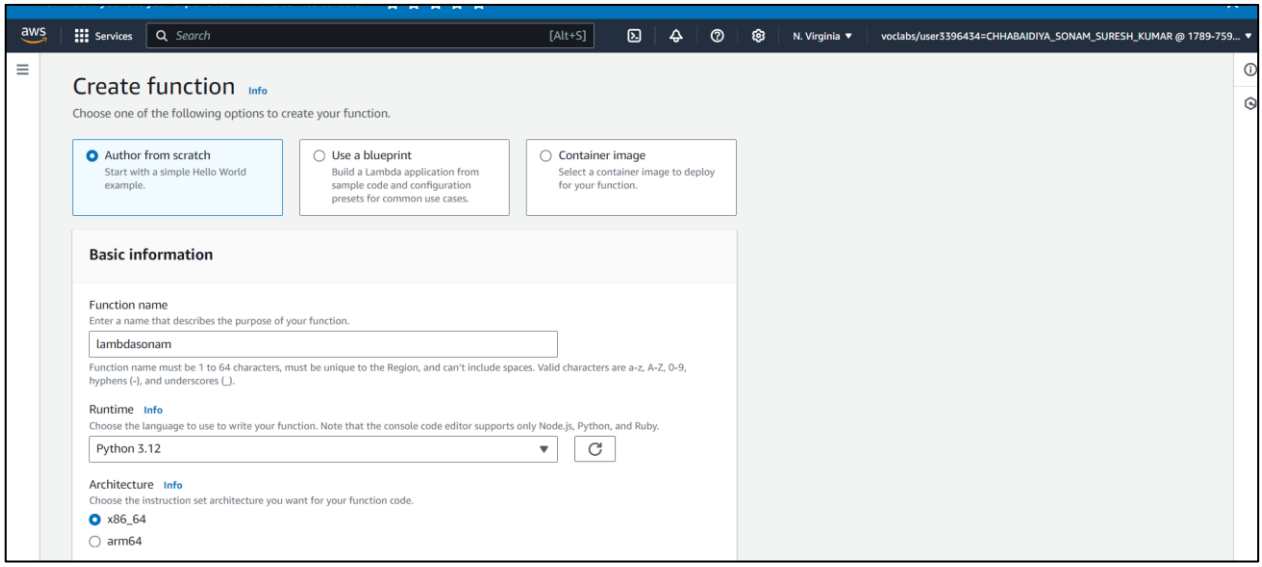
1. Go to your AWS account and search for Lamda



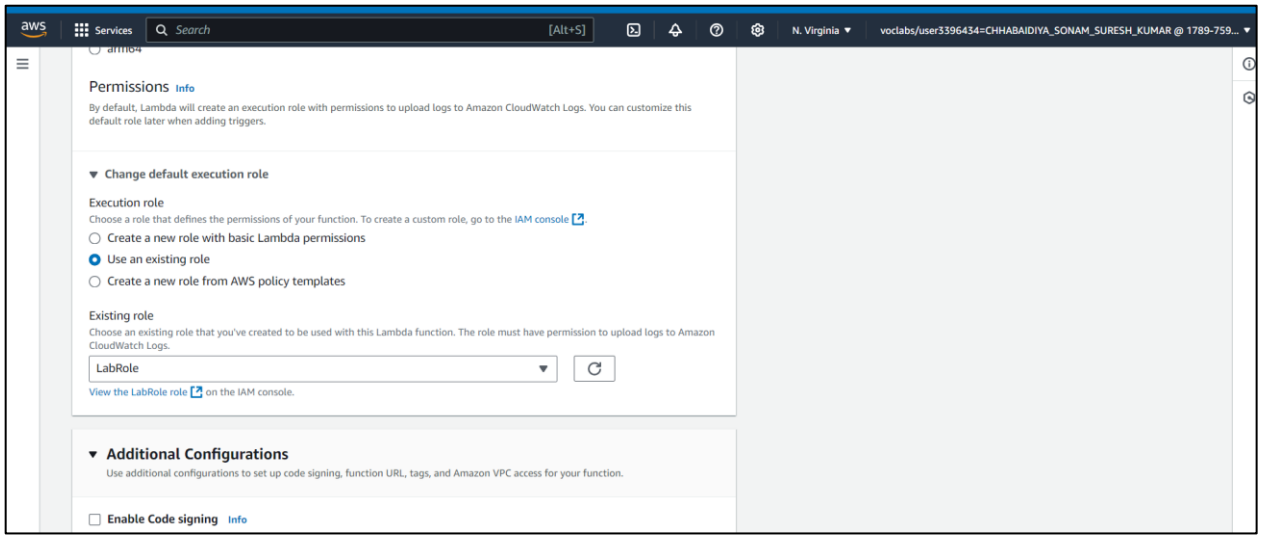


Sonam Chhabaidiya D15A09

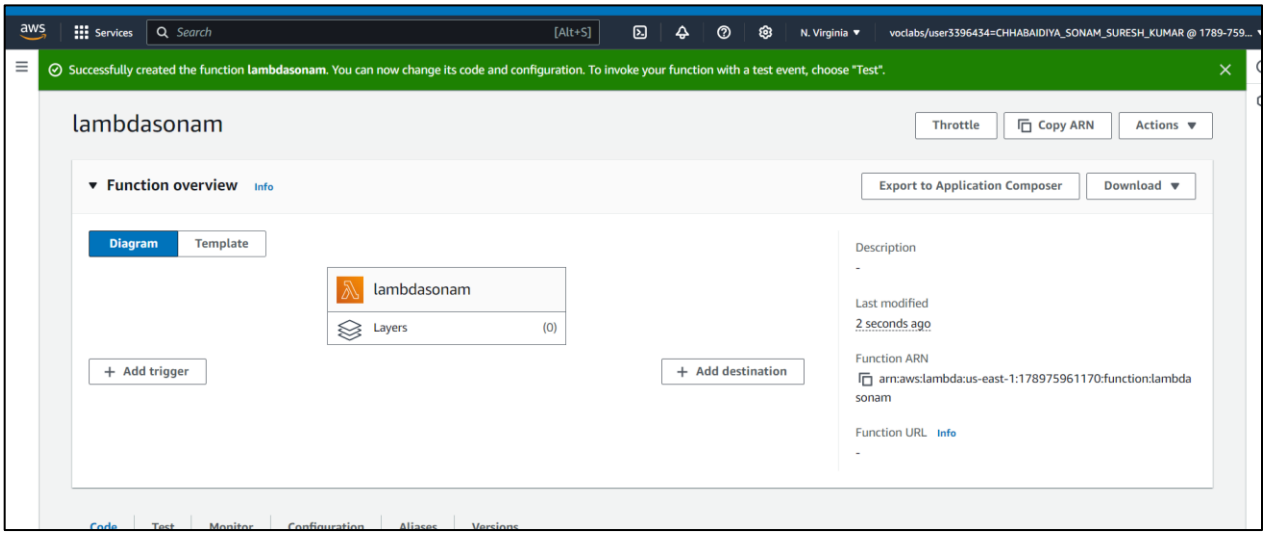
2. Create a new Lambda function

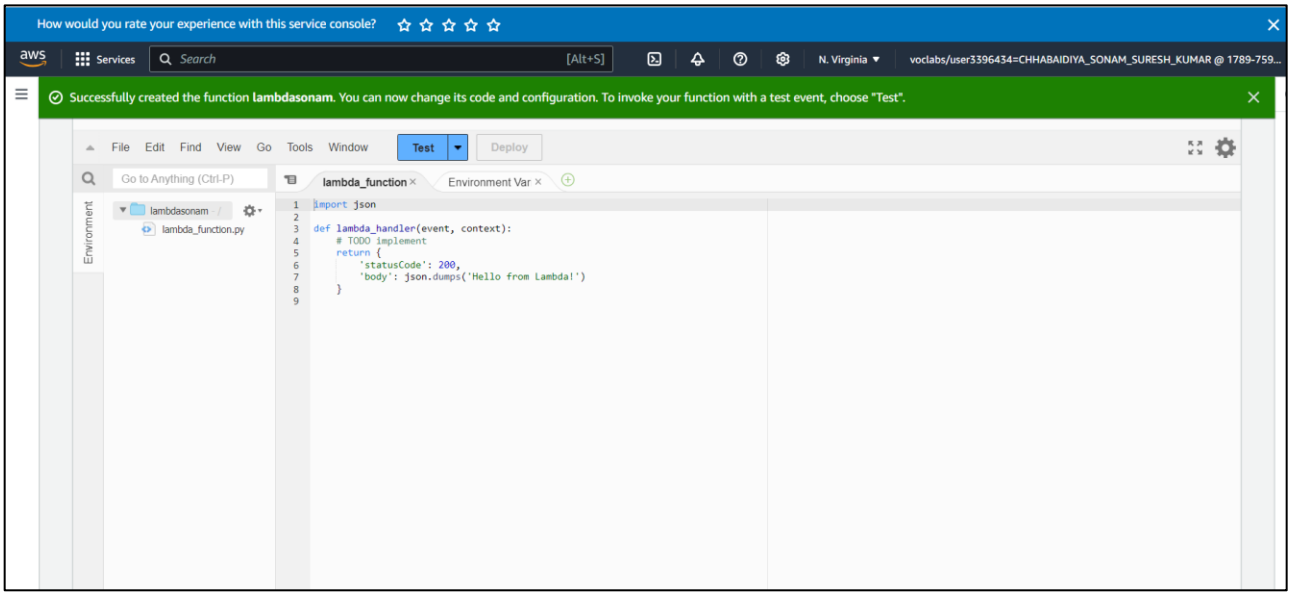


Use Labrole as user role.

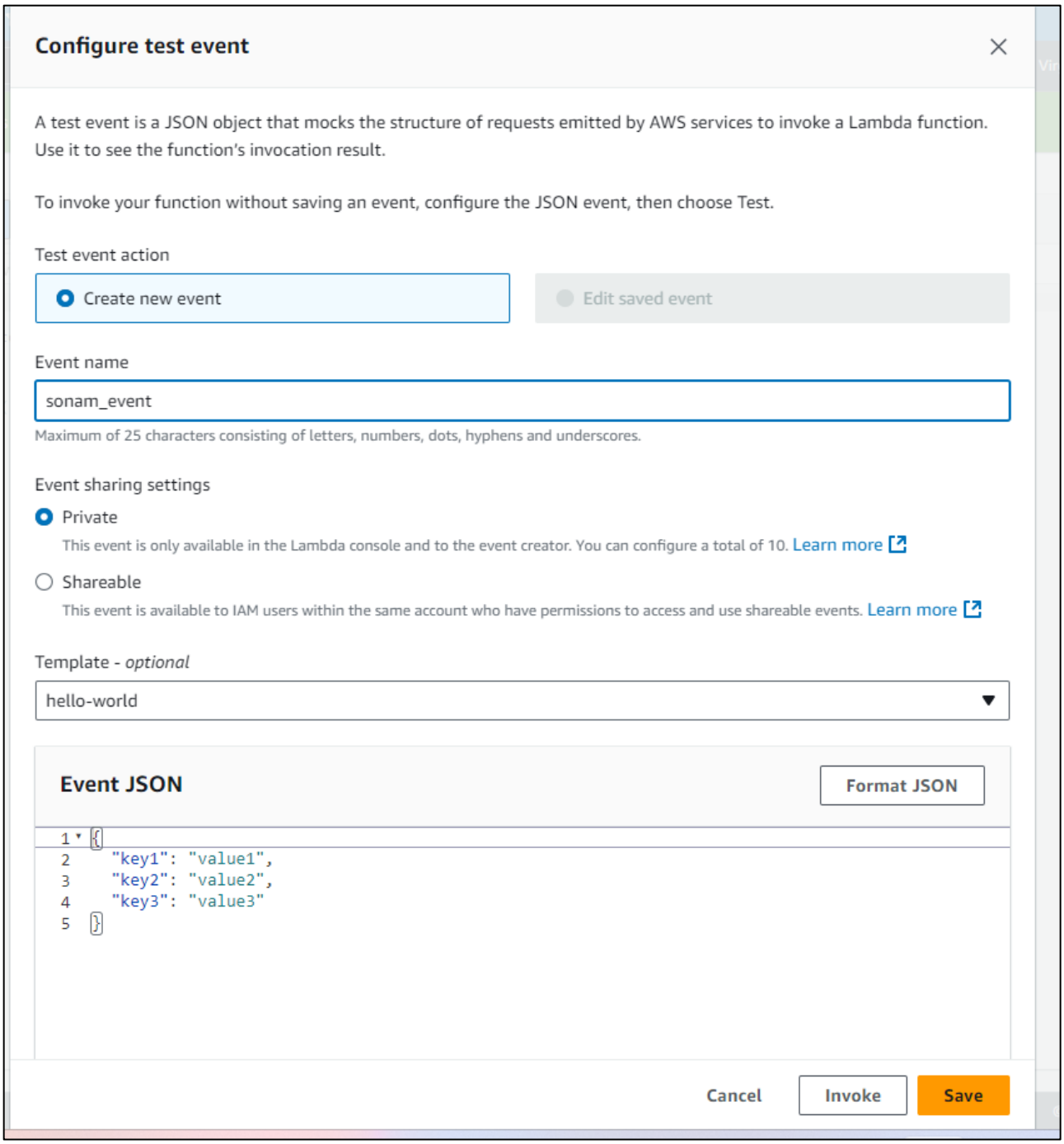


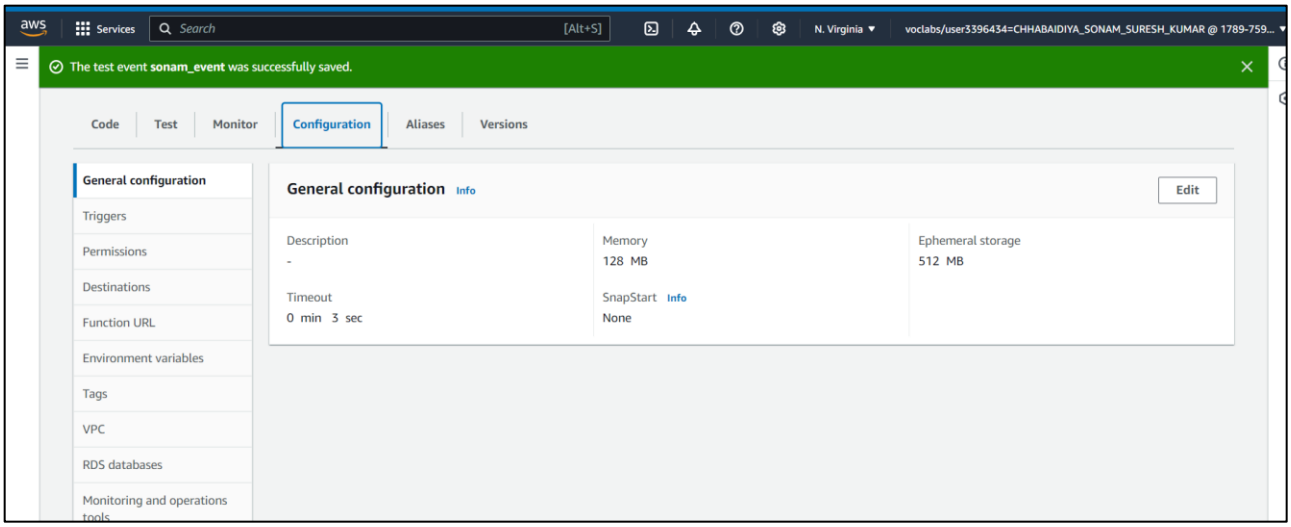
Sonam Chhabaidiya D15A09



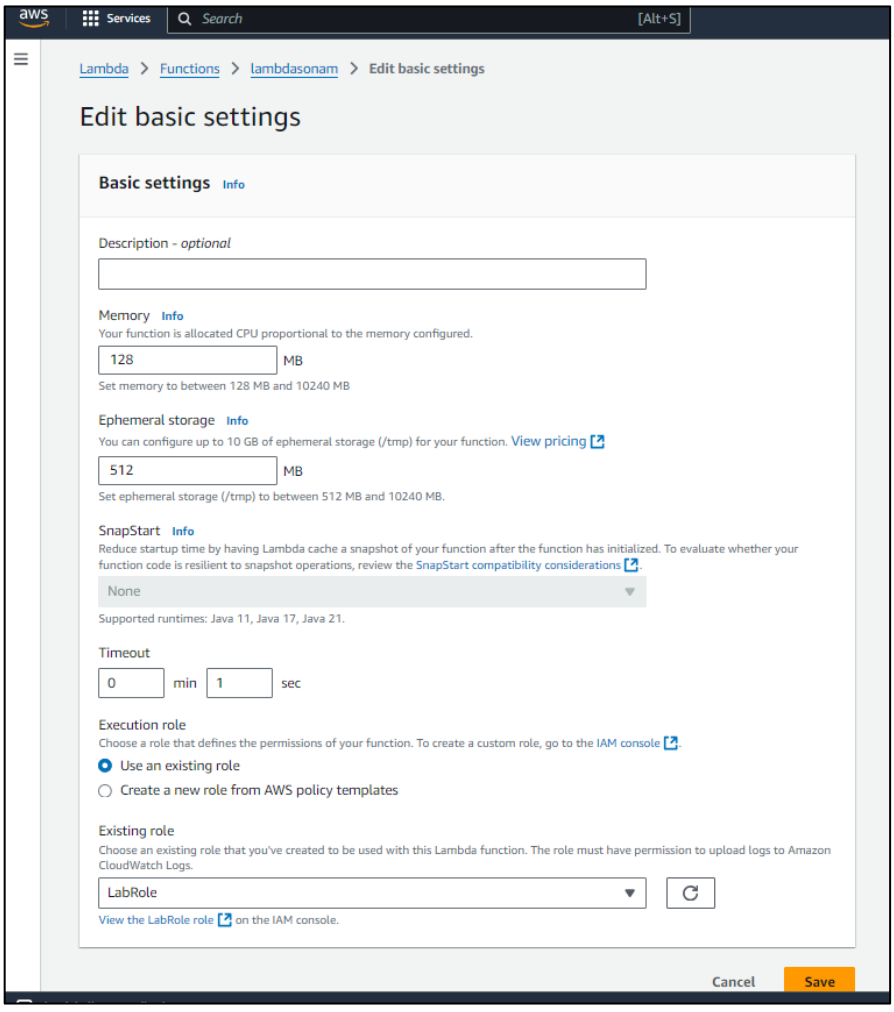


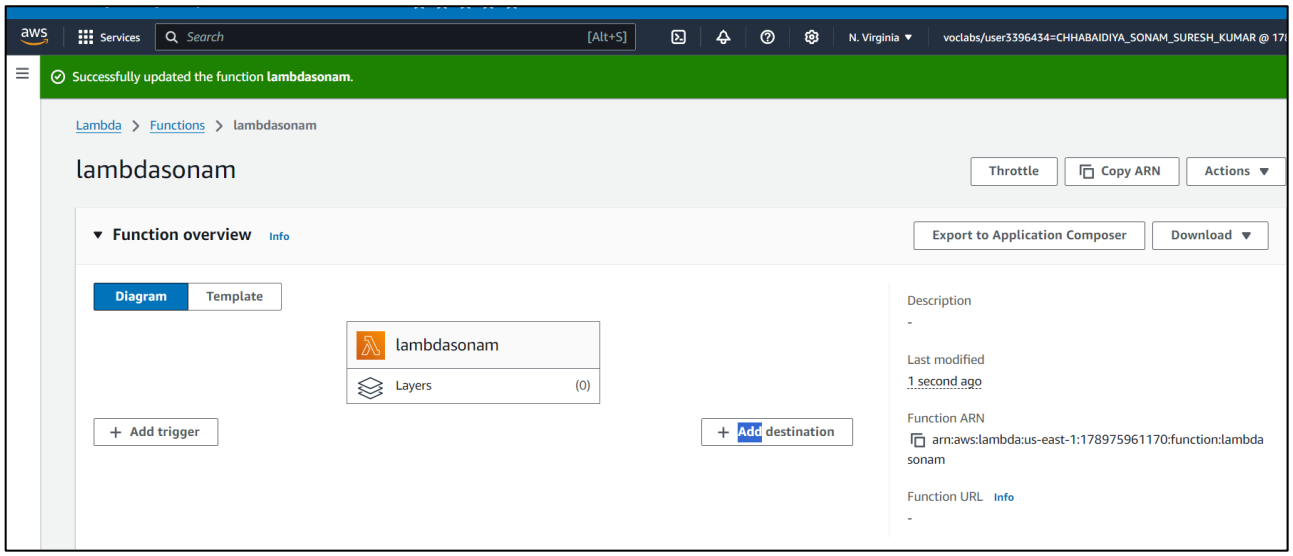
3. Create a new test event and save it.



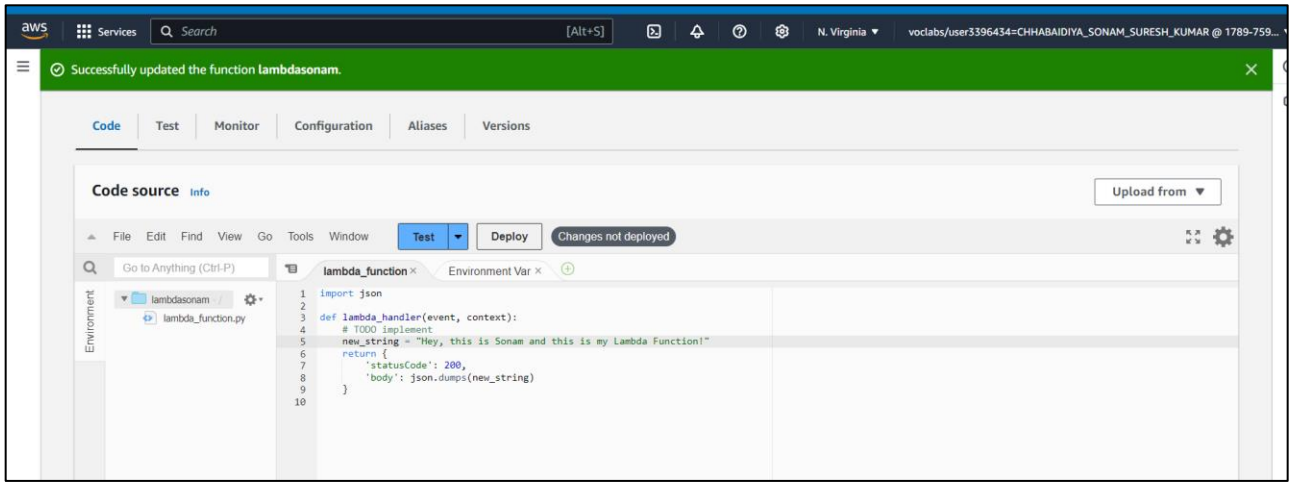


4. Change the basic settings

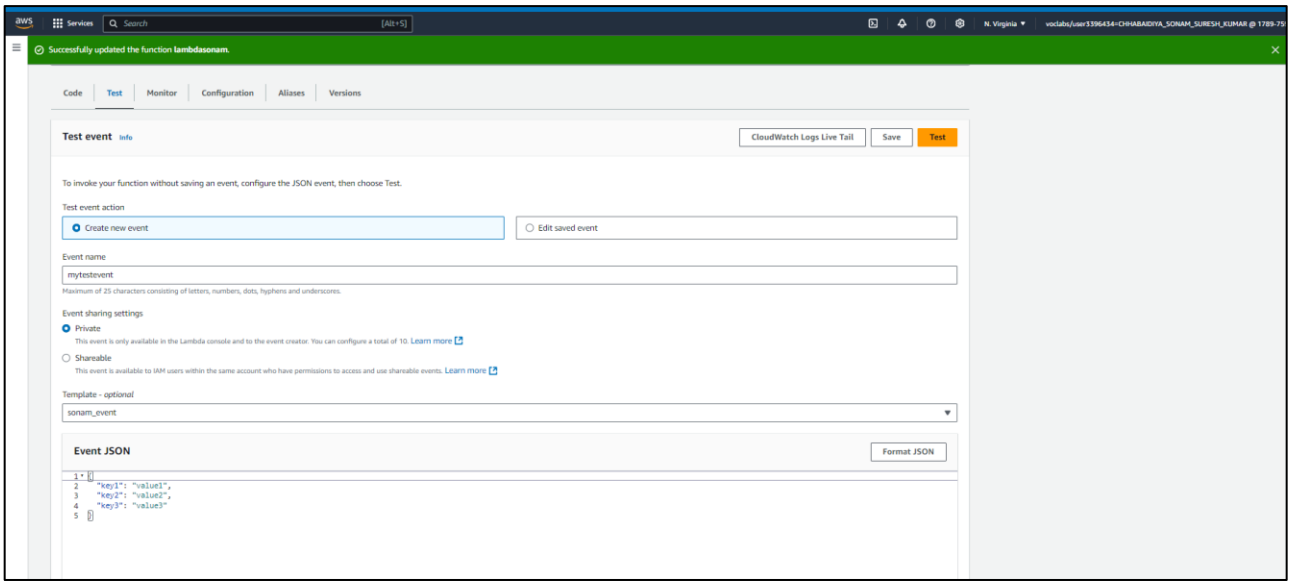




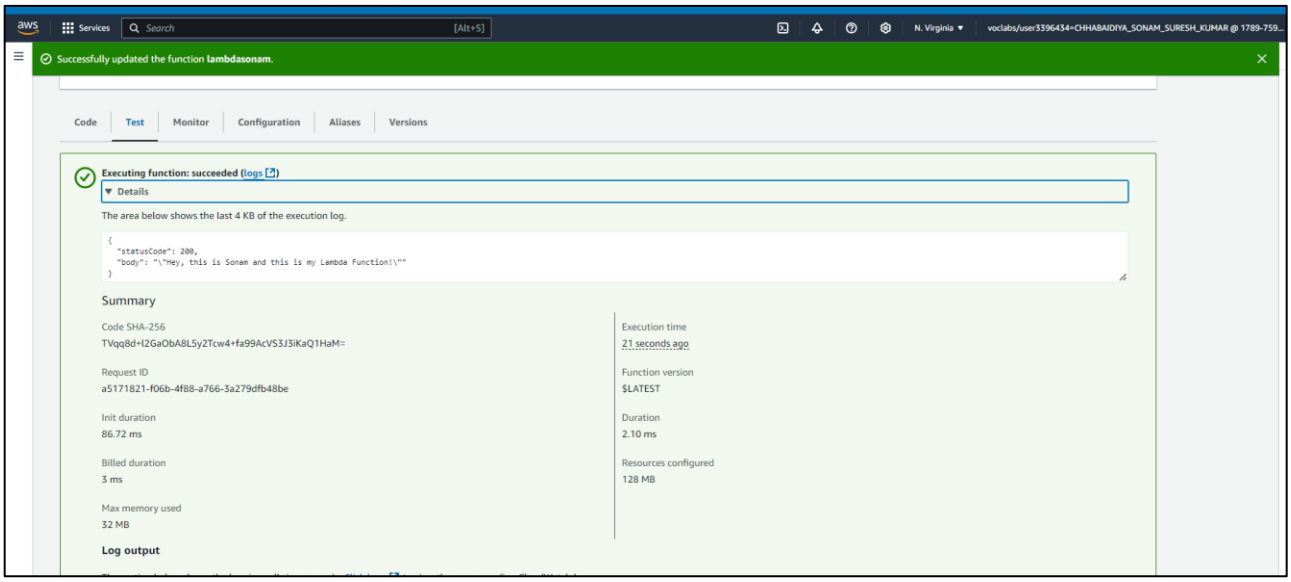
5. Edit the code as required and ctrl+ s to save it. Click on Deploy



6. Create a new test and click on Test



7. After this, the result can be seen



Sonam Chhabaidiya D15A09

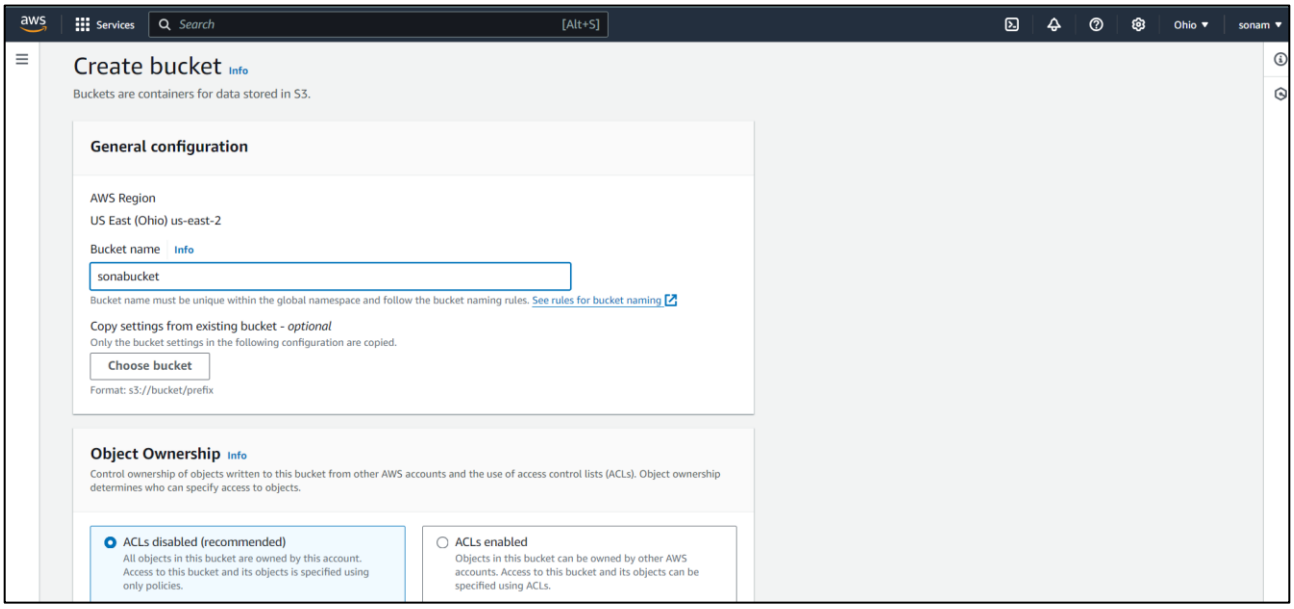
Sonam Chhabaidiya D15A09

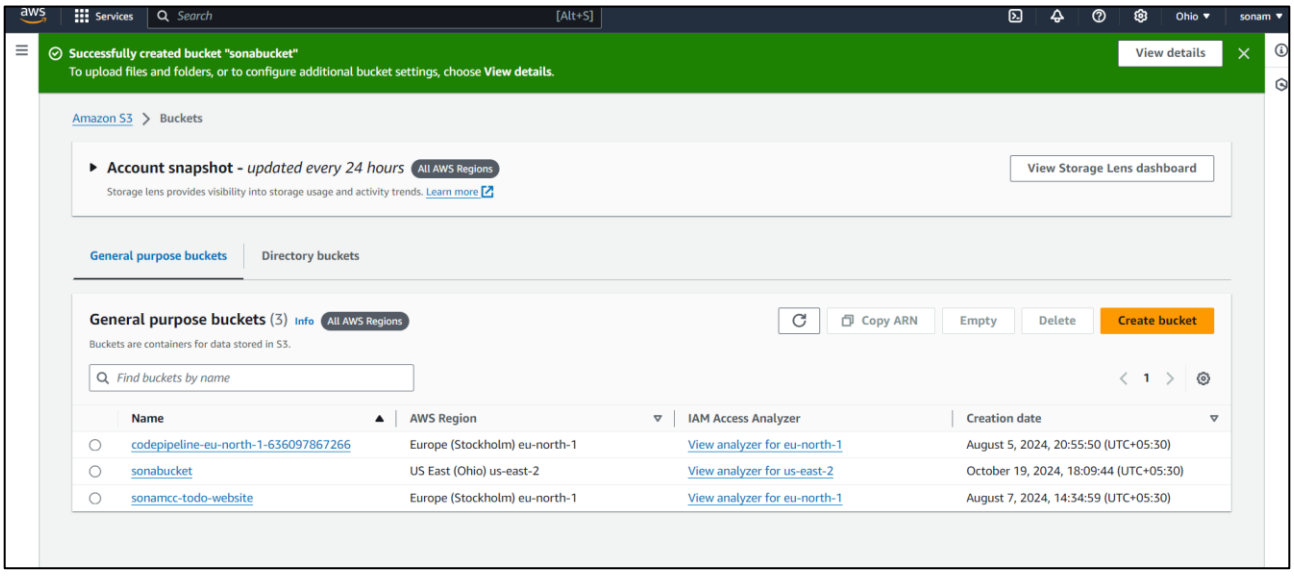
Sonam Chhabaidiya
D15A - 09

ADVANCE DEVOPS
Experiment No 12

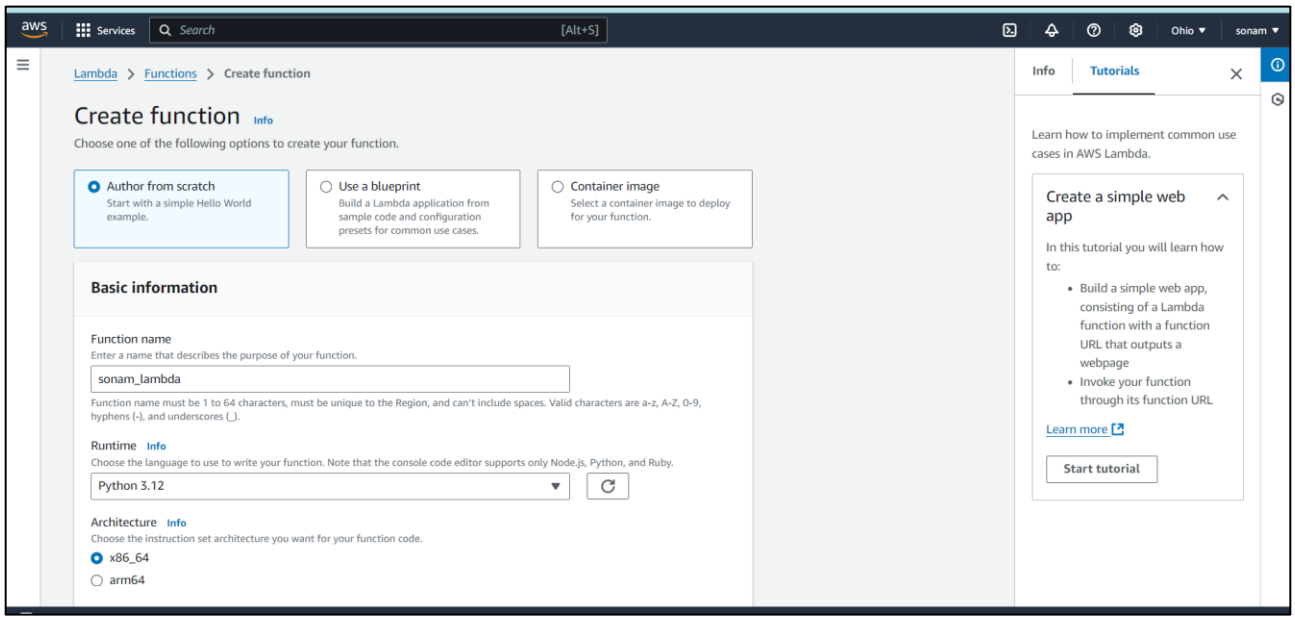
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

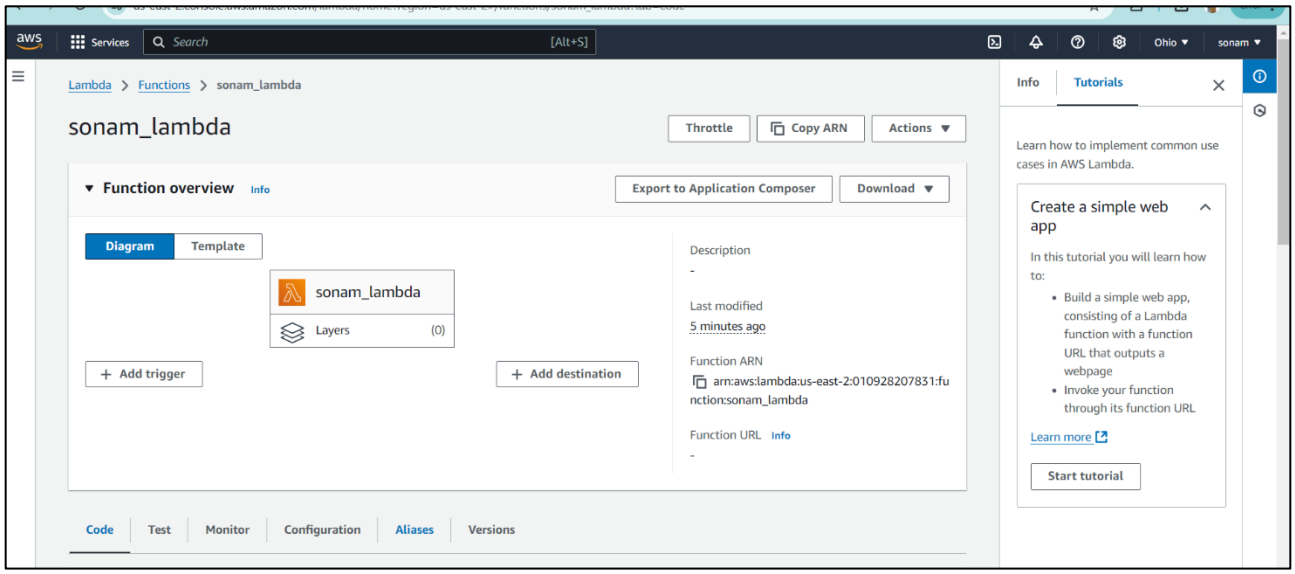
1. Create an AWS S3 bucket and save it.



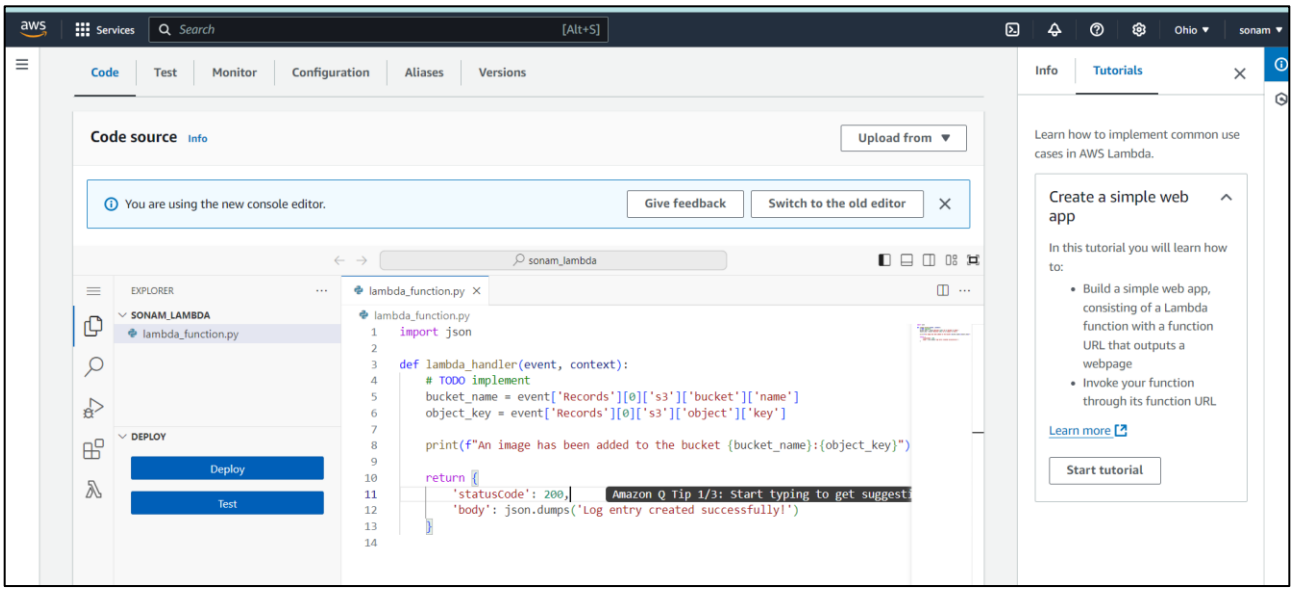


2. Create an AWS Lambda Function, use python 3.11

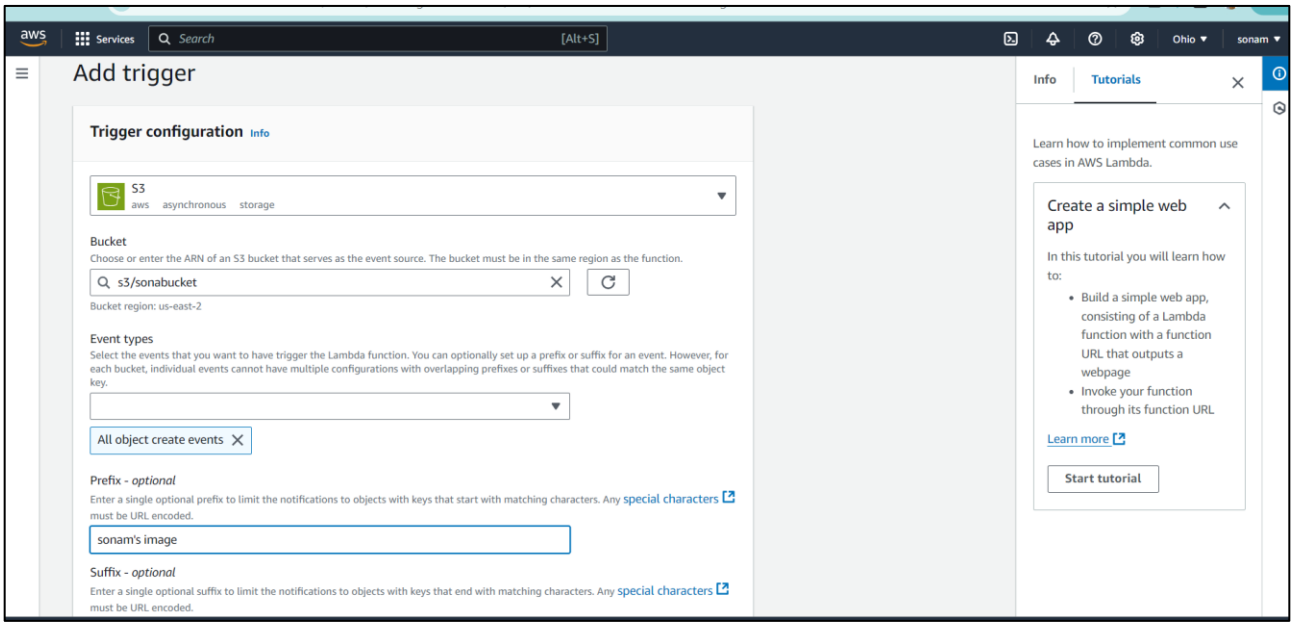


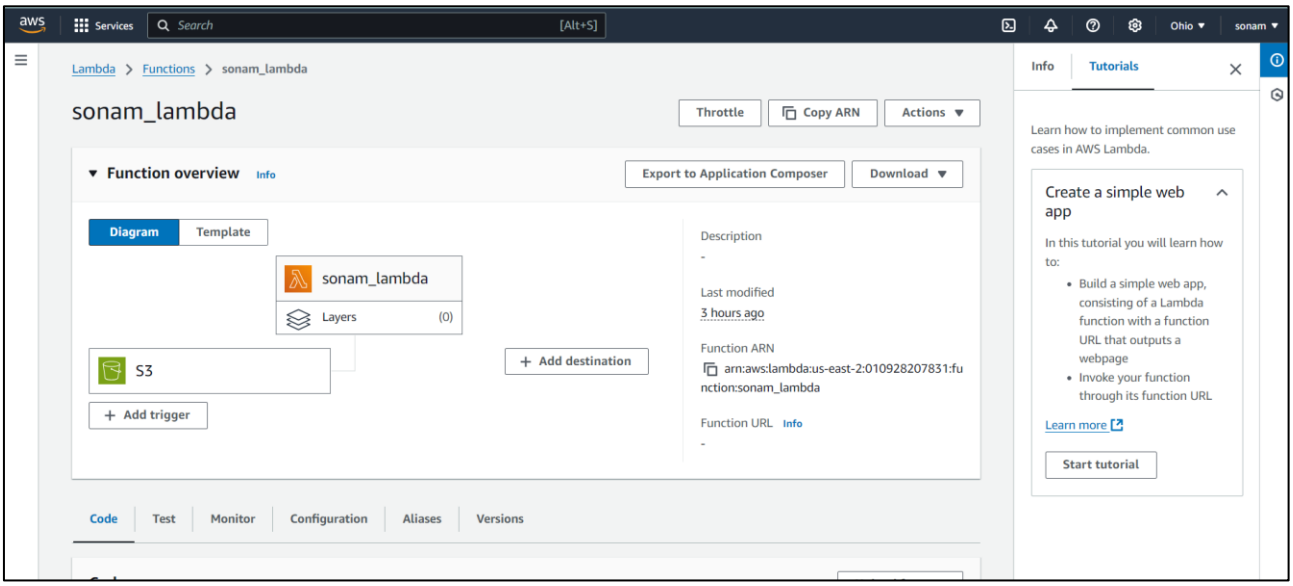


3. Change the code such that it prints “An image has been added to the bucket” when triggered.

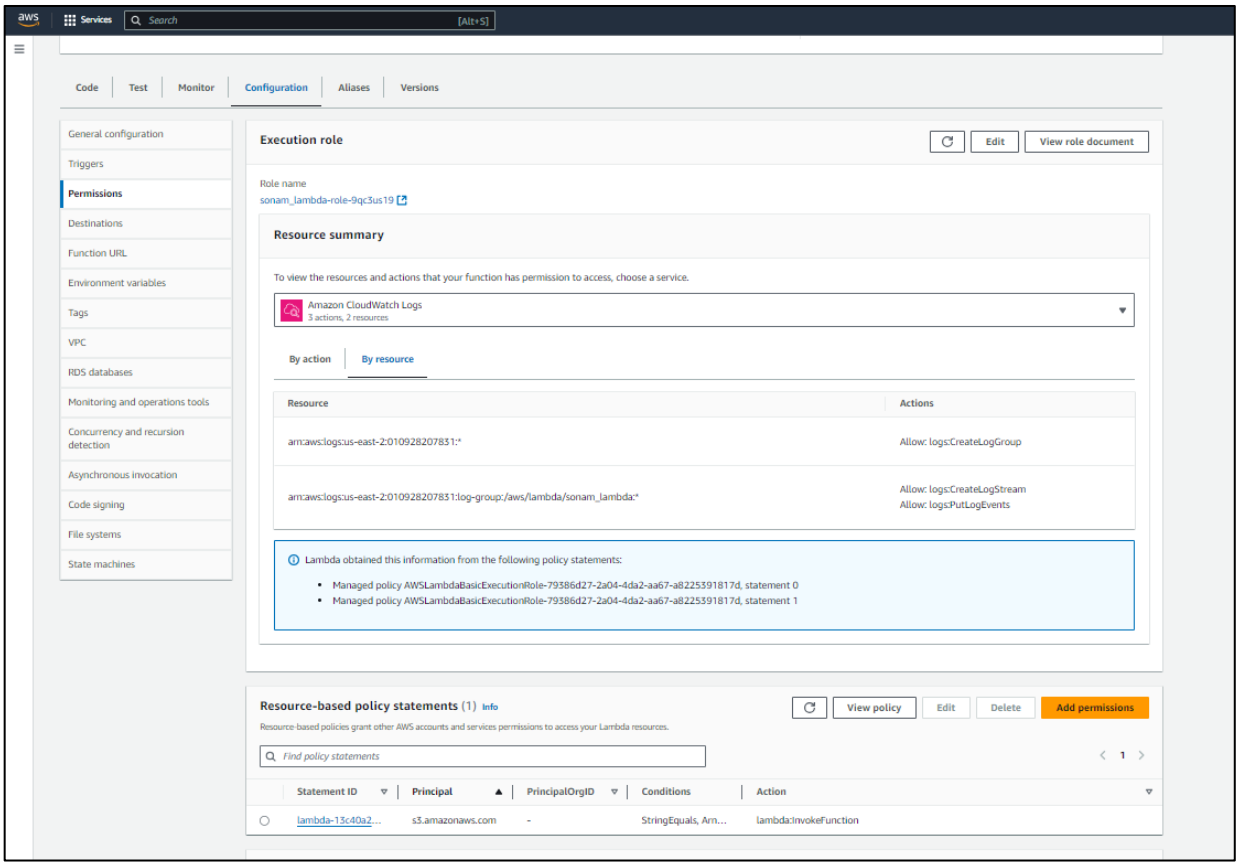


4. Add S3 bucket in your triggers and select the name of your S3 bucket.

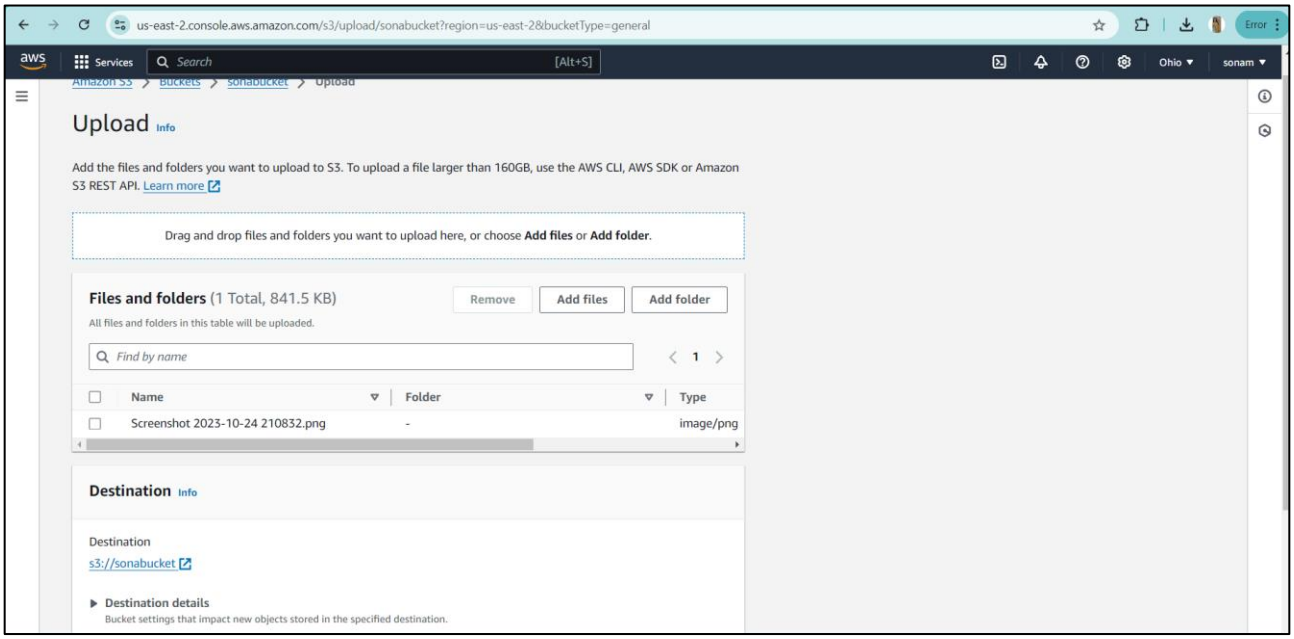


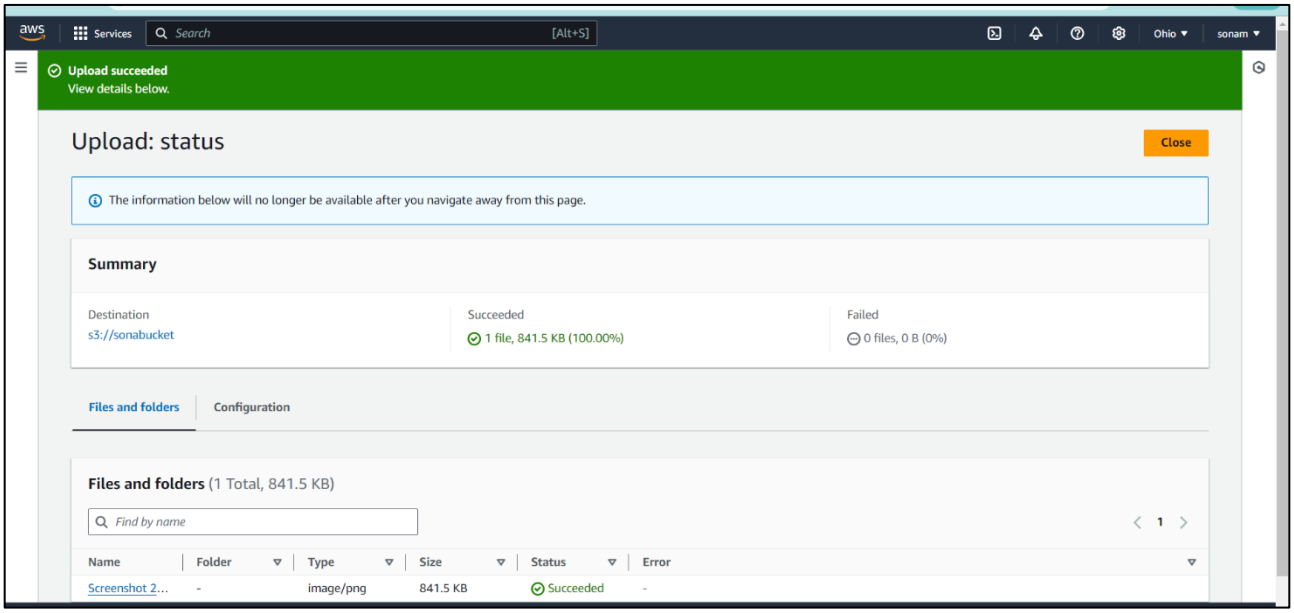


5. Go to your S3 bucket



6. Upload an image inside the s3 bucket





7. Go back to your Lambda function and under monitor select ‘open cloudwatch logs’ Here you can see the message: An image has been added to the bucket.