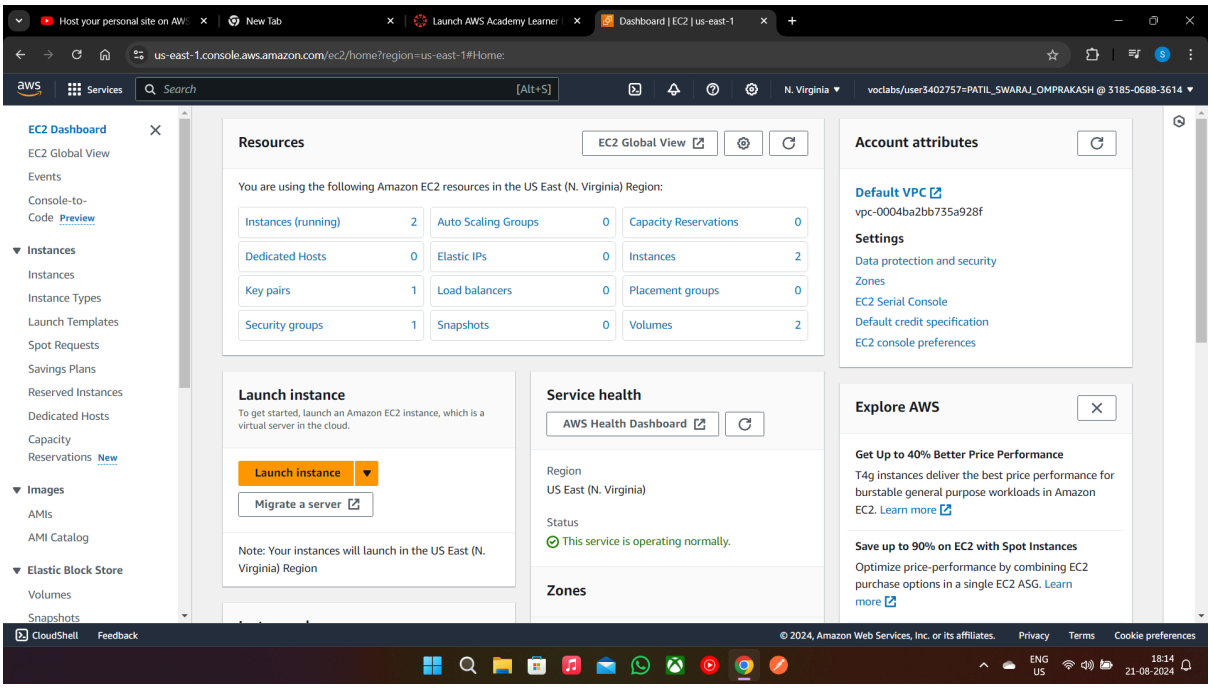
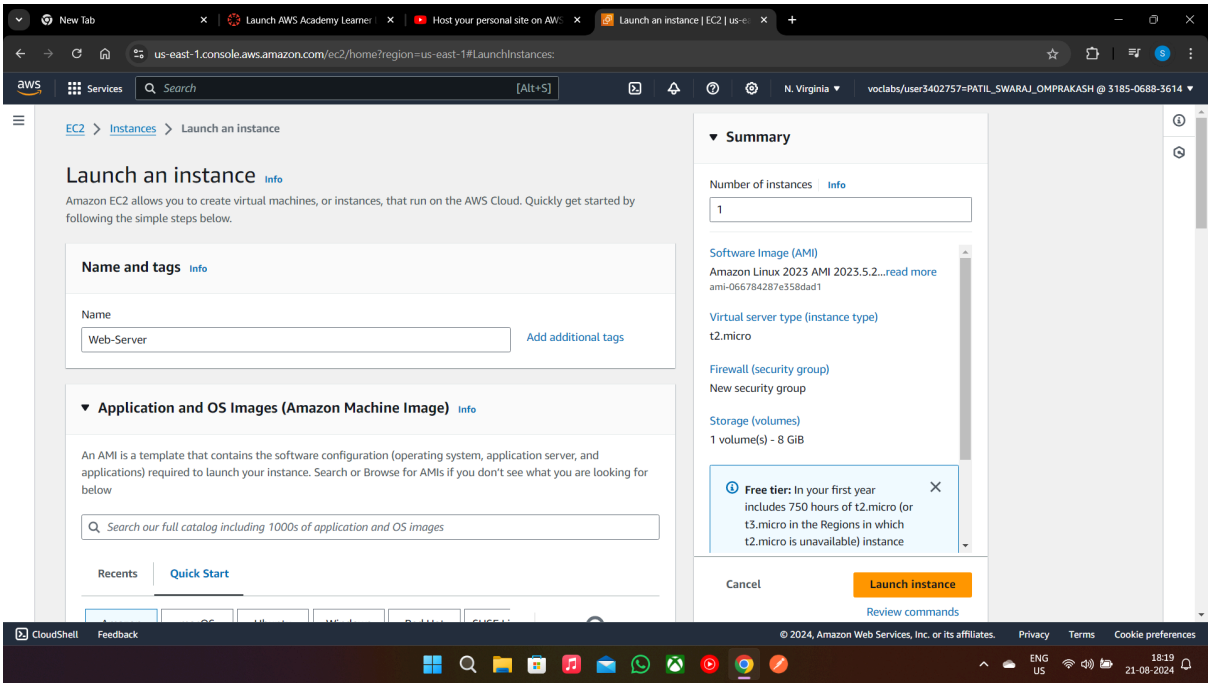
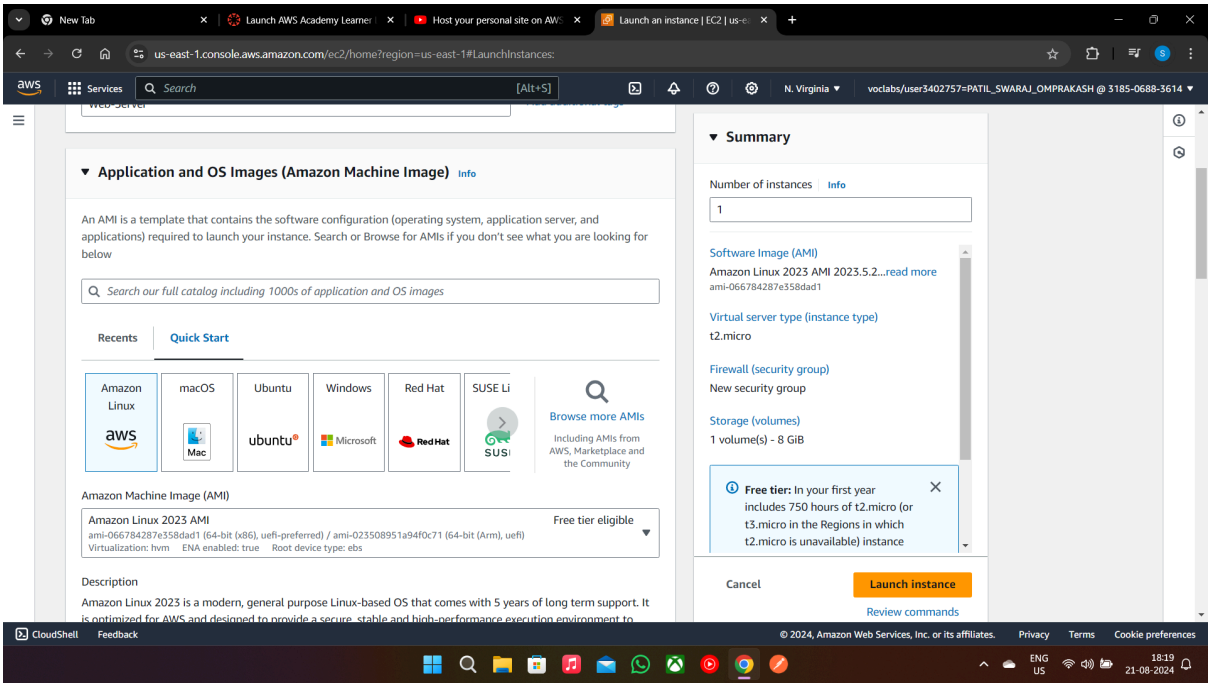
EXPERIMENT 1

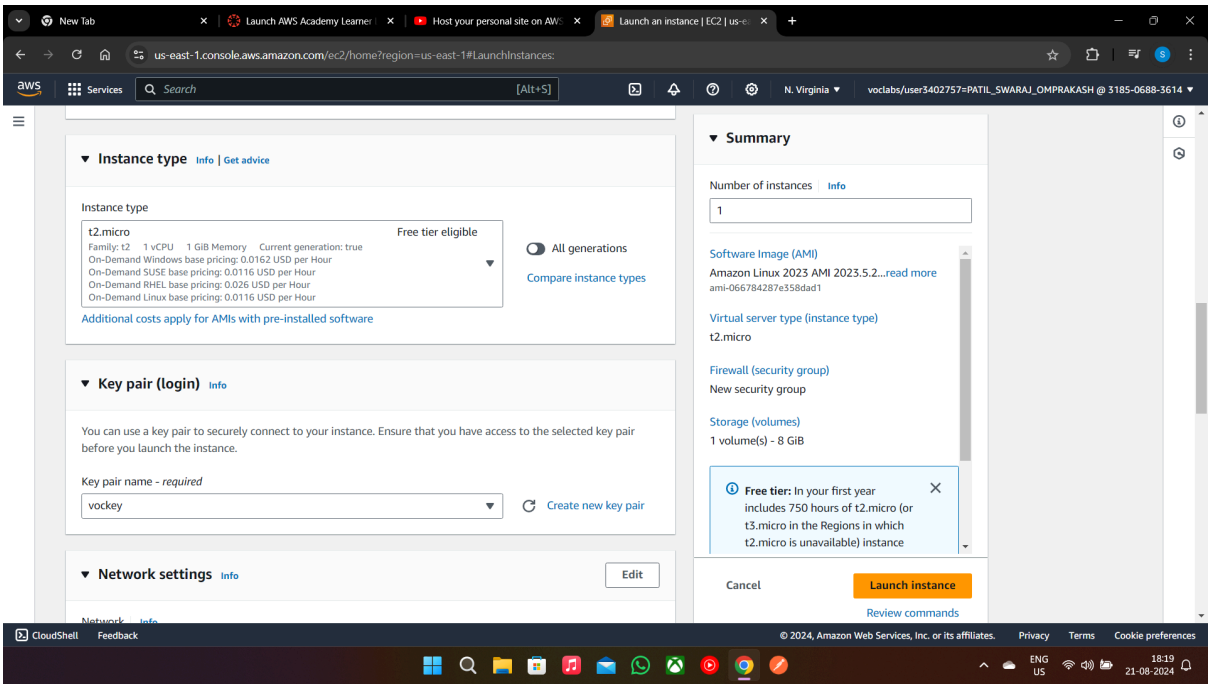
1) To Devlop A Website And Host it on a VM.

Steps:
Create An Instance:

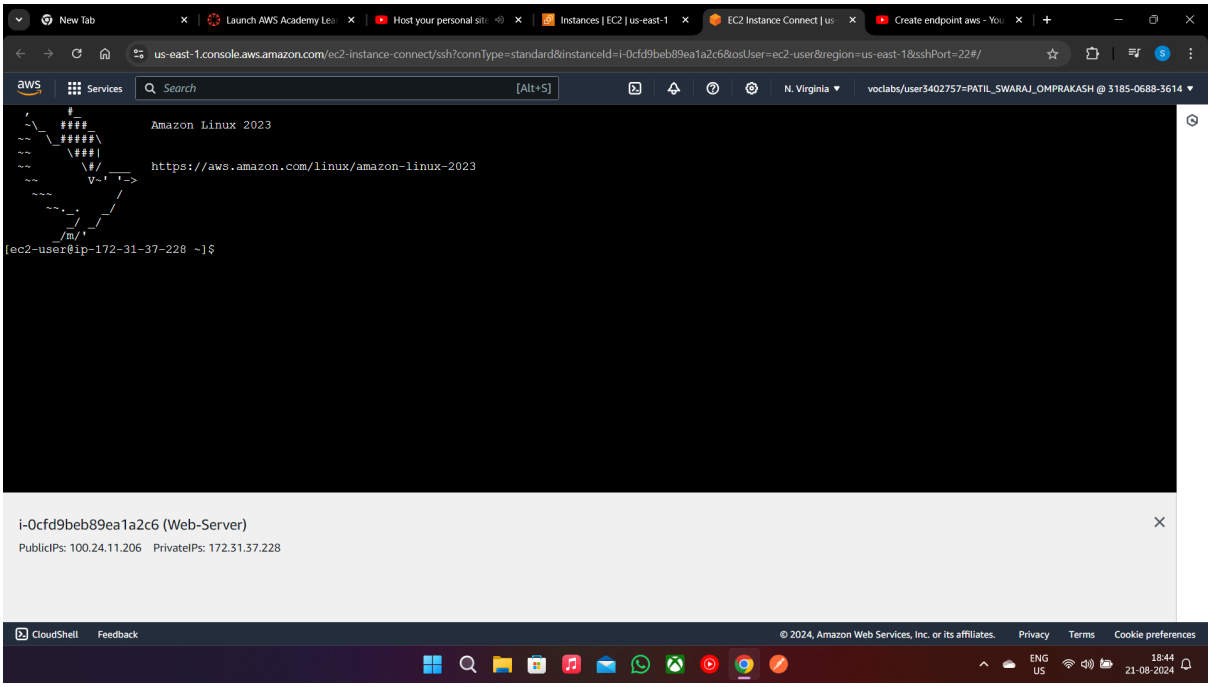




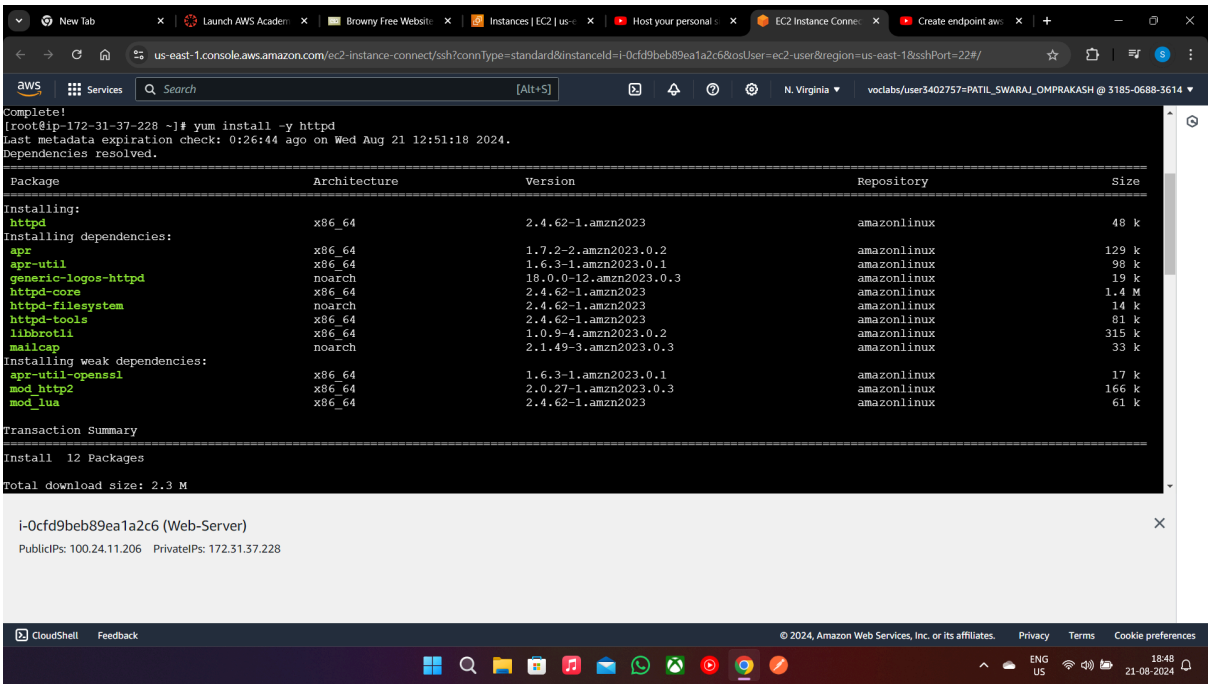

New Tab
Launch AWS Academy Learner
Host your personal site on AW
Launch an instance | EC2 | us-e
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH @ 3185-0688-3614
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
Free tier eligible
ami-066784287e358dad1 (64-bit (x86), uefi-preferred) / ami-023508951a94f0c71 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-066784287e358dad1
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
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
ENG
US
18:19
21-08-2024

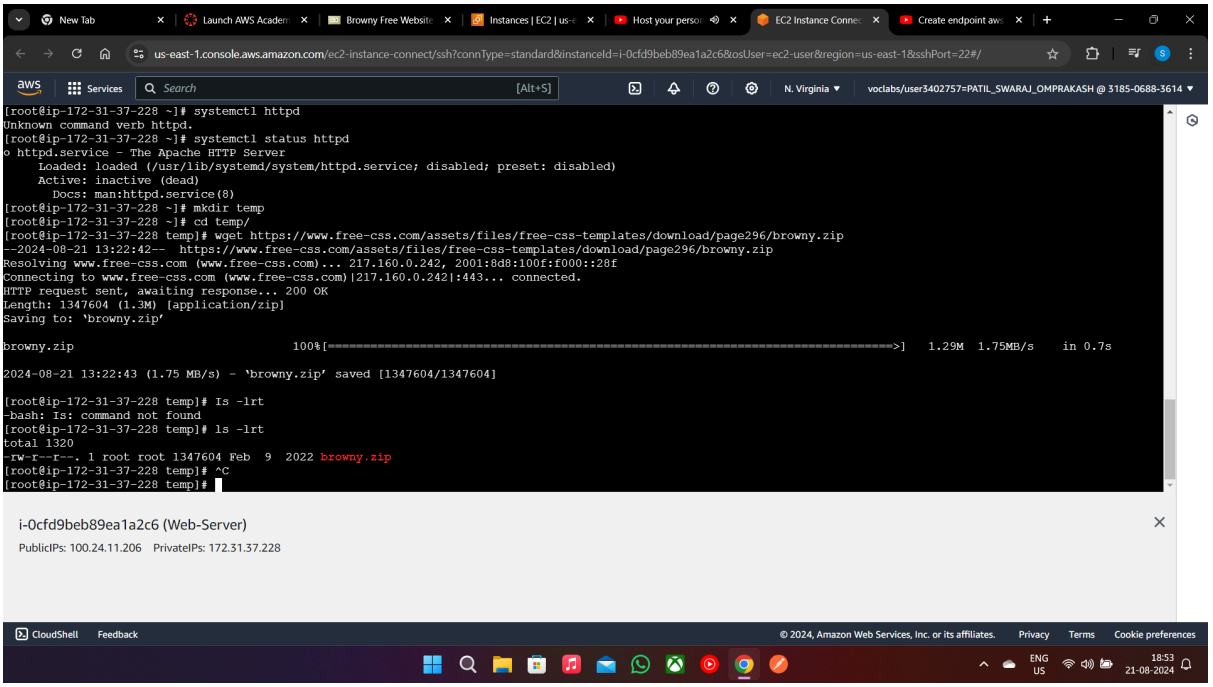

New Tab
Launch AWS Academy Learner
Host your personal site on AW
Launch an instance | EC2 | us-e
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH @ 3185-0688-3614
Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
vockey
Create new key pair
Network settings Info
Edit
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-066784287e358dad1
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
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
ENG
US
18:19
21-08-2024

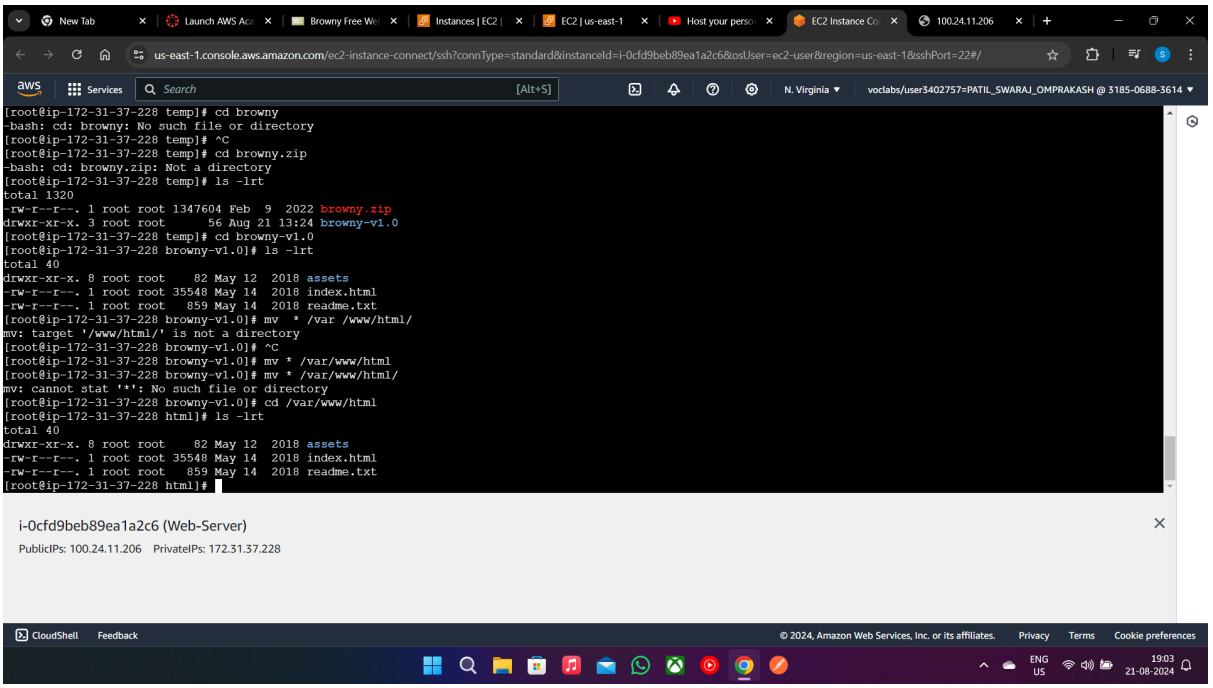
Step 2: Connect To the Instance:



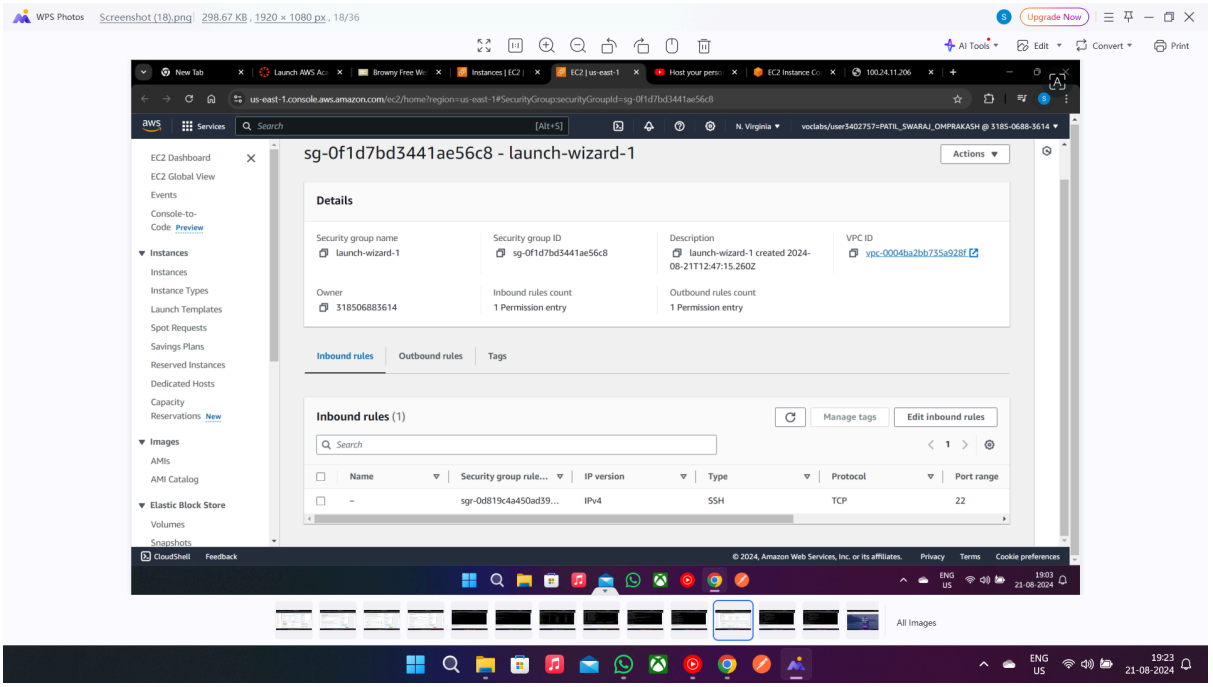
Type the Required Commands:

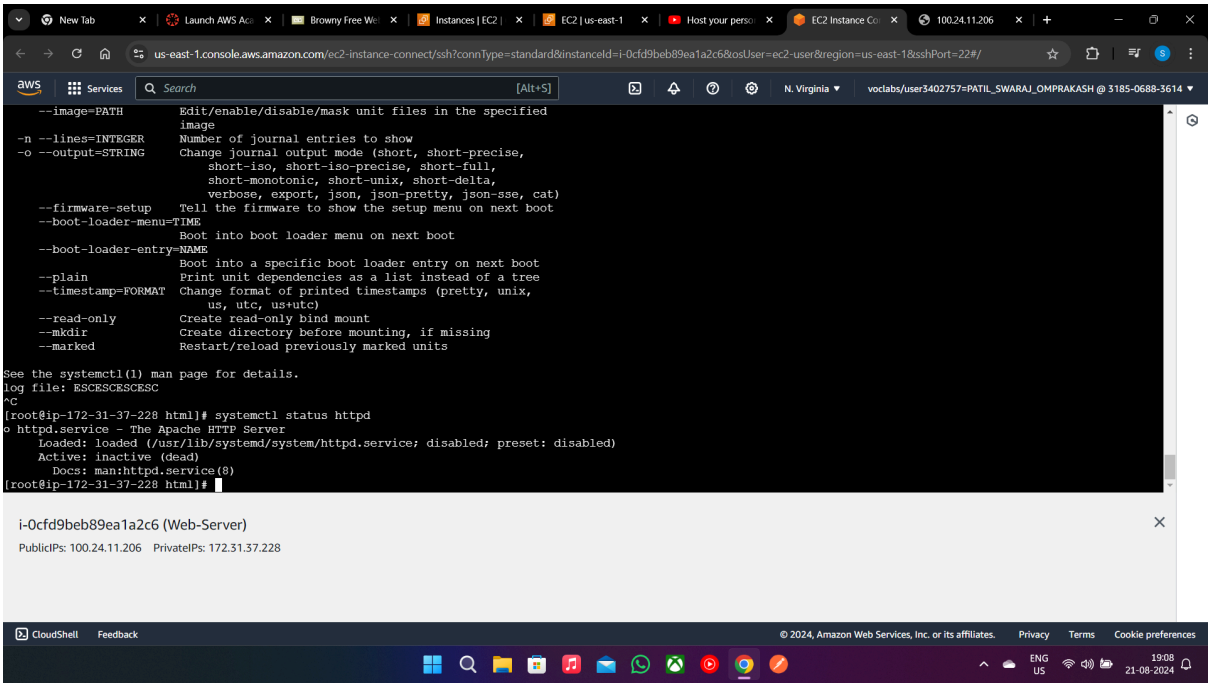



New Tab
Launch AWS Academ
Browny Free Website
Instances | EC2 | us-e
Host your perso
EC2 Instance Connec
Create endpoint aws
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0cfd9beb89ea1a2c6&osUser=ec2-user&region=us-east-1&sshPort=22#/
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH @ 3185-0688-3614
[root@ip-172-31-37-228 ~]# systemctl httpd
Unknown command verb httpd.
[root@ip-172-31-37-228 ~]# systemctl status httpd
o httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
Active: inactive (dead)
Docs: man:httpd.service(8)
[root@ip-172-31-37-228 ~]# mkdir temp
[root@ip-172-31-37-228 ~]# cd temp/
[root@ip-172-31-37-228 temp]# wget https://www.free-css.com/assets/files/free-css-templates/download/page296/browny.zip
--2024-08-21 13:22:42-- https://www.free-css.com/assets/files/free-css-templates/download/page296/browny.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d8:100f:f000::28f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1347604 (1.3M) [application/zip]
Saving to: 'browny.zip'
browny.zip 100%[>] 1.29M 1.75MB/s in 0.7s
2024-08-21 13:22:43 (1.75 MB/s) - 'browny.zip' saved [1347604/1347604]
[root@ip-172-31-37-228 temp]# Is -lrt
-bash: Is: command not found
[root@ip-172-31-37-228 temp]# ls -lrt
total 1320
-rw-r--r--. 1 root root 1347604 Feb 9 2022 browny.zip
[root@ip-172-31-37-228 temp]# ^C
[root@ip-172-31-37-228 temp]#
i-0cfd9beb89ea1a2c6 (Web-Server)
PublicIPs: 100.24.11.206 PrivateIPs: 172.31.37.228
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
US
18:53
21-08-2024

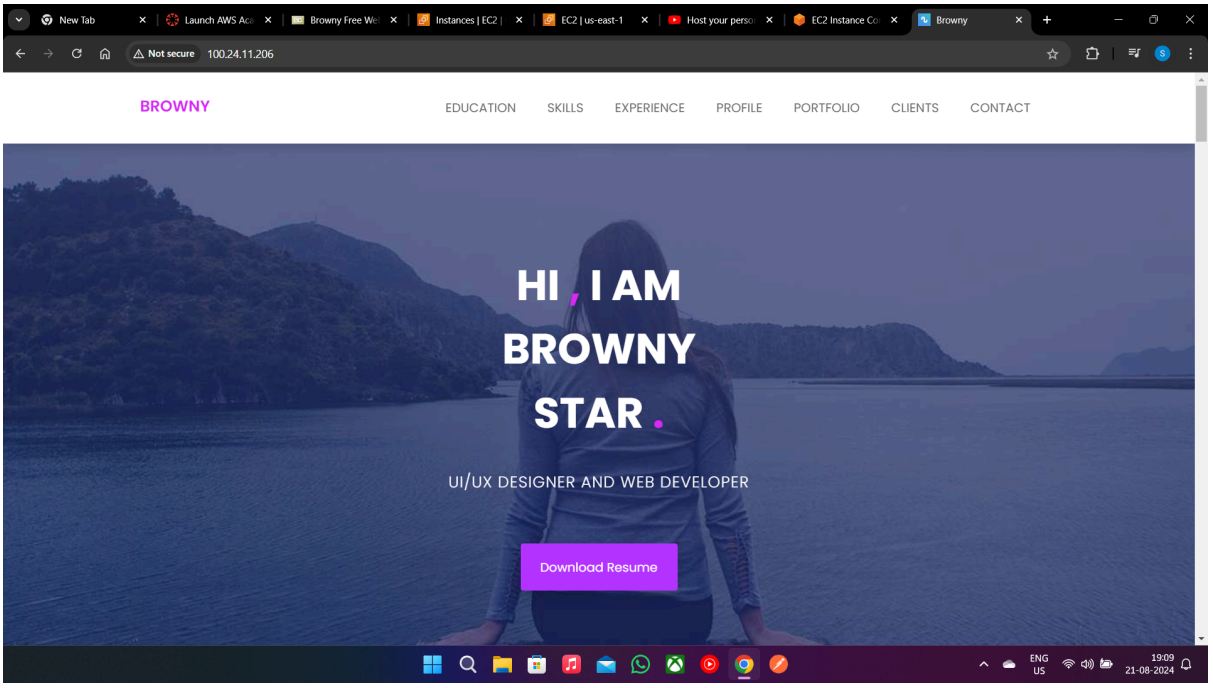

New Tab
Launch AWS Ac
Browny Free We
Instances | EC2 |
EC2 | us-east-1
Host your perso
EC2 Instance Co
100.24.11.206
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0cfd9beb89ea1a2c6&osUser=ec2-user&region=us-east-1&sshPort=22#/
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH @ 3185-0688-3614
[root@ip-172-31-37-228 temp]# cd browny
-bash: cd: browny: No such file or directory
[root@ip-172-31-37-228 temp]# ^C
[root@ip-172-31-37-228 temp]# cd browny.zip
-bash: cd: browny.zip: Not a directory
[root@ip-172-31-37-228 temp]# ls -lrt
total 1320
-rw-r--r--. 1 root root 1347604 Feb 9 2022 browny.zip
drwxr-xr-x. 3 root root 56 Aug 21 13:24 browny-v1.0
[root@ip-172-31-37-228 temp]# cd browny-v1.0
[root@ip-172-31-37-228 browny-v1.0]# ls -lrt
total 40
drwxr-xr-x. 8 root root 82 May 12 2018 assets
-rw-r--r--. 1 root root 35548 May 14 2018 index.html
-rw-r--r--. 1 root root 859 May 14 2018 readme.txt
[root@ip-172-31-37-228 browny-v1.0]# mv * /var /www/html/
mv: target '/www/html/' is not a directory
[root@ip-172-31-37-228 browny-v1.0]# ^C
[root@ip-172-31-37-228 browny-v1.0]# mv * /var/www/html
[root@ip-172-31-37-228 browny-v1.0]# mv * /var/www/html/
mv: cannot stat '*': No such file or directory
[root@ip-172-31-37-228 browny-v1.0]# cd /var/www/html
[root@ip-172-31-37-228 html]# ls -lrt
total 40
drwxr-xr-x. 8 root root 82 May 12 2018 assets
-rw-r--r--. 1 root root 35548 May 14 2018 index.html
-rw-r--r--. 1 root root 859 May 14 2018 readme.txt
[root@ip-172-31-37-228 html]#
i-0cfd9beb89ea1a2c6 (Web-Server)
PublicIPs: 100.24.11.206 PrivateIPs: 172.31.37.228
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
US
19:03
21-08-2024

Give it Permission For HTTP And Https:



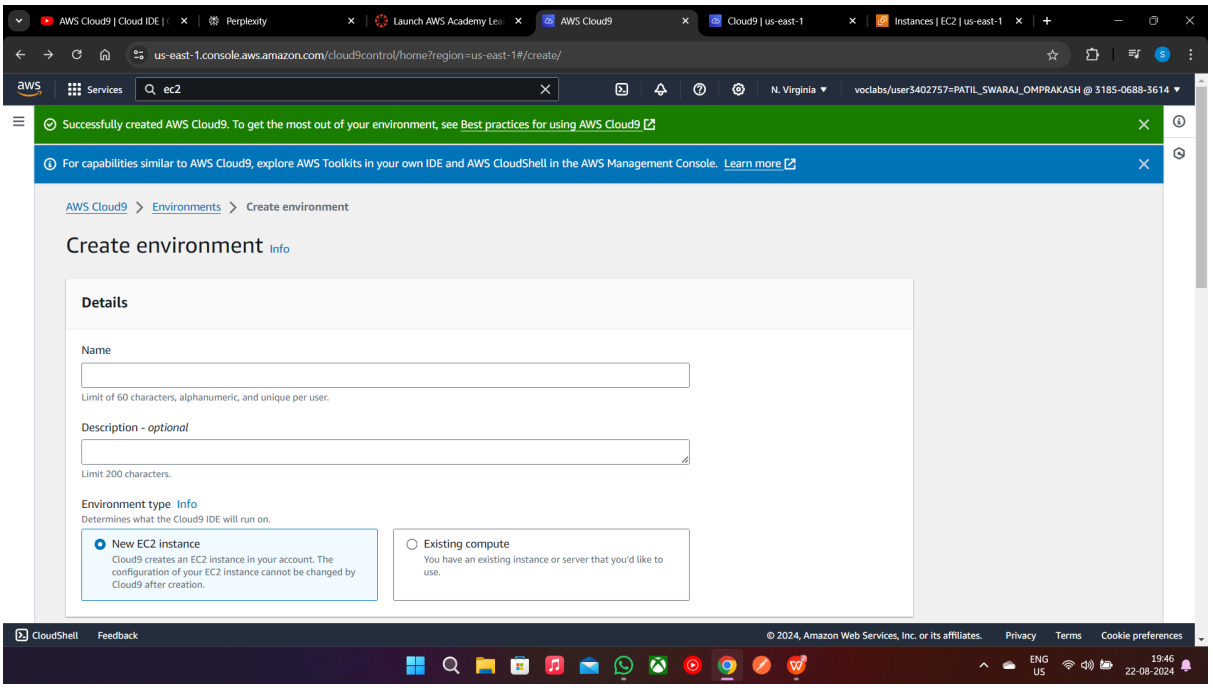


THE HOSTED WEBSITE:

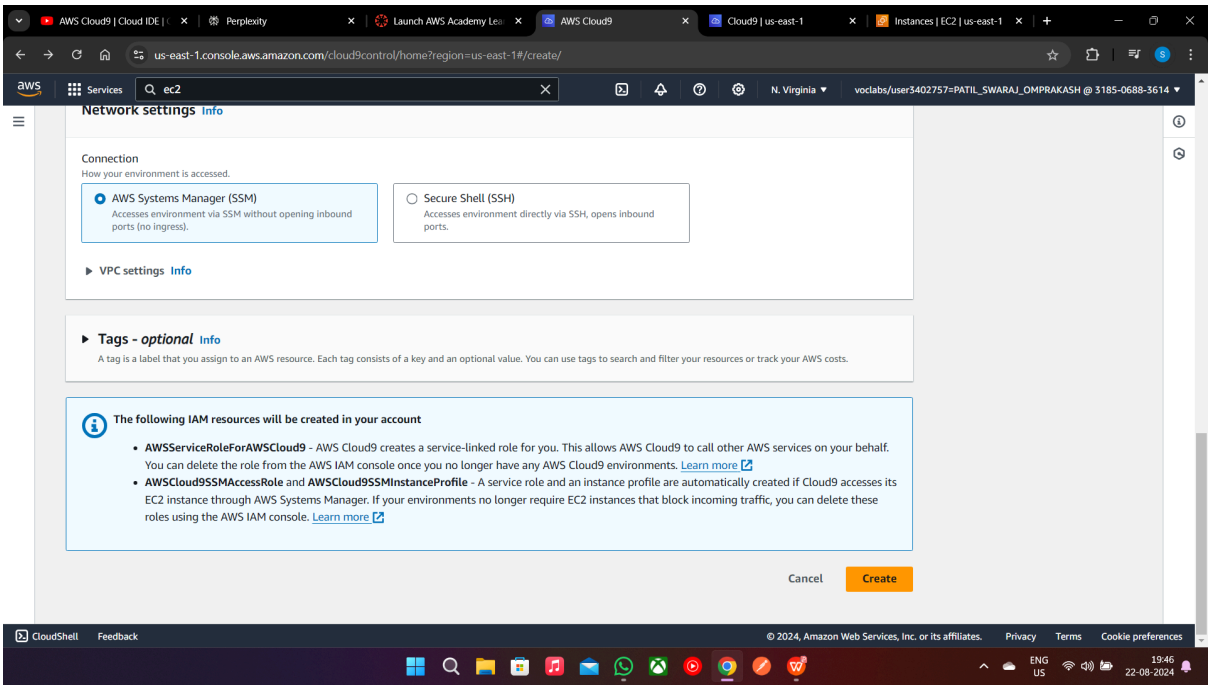


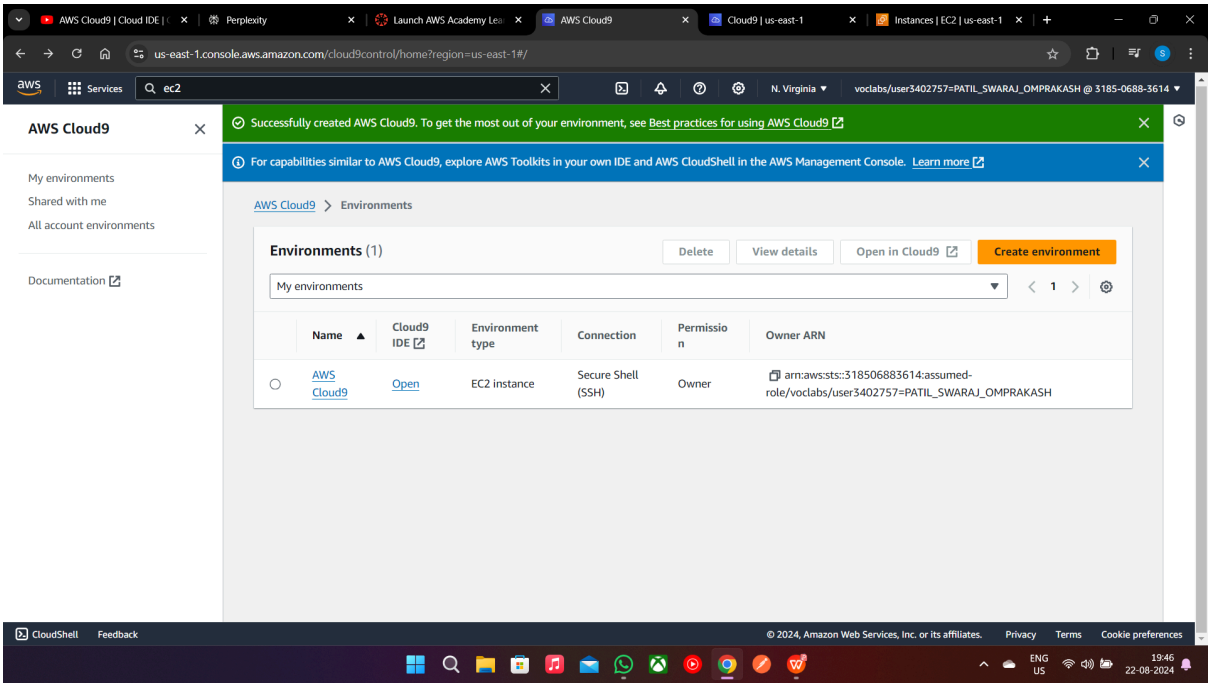
2) HOSTING THROUGH CLOUD9:

Step 1: CREATE A CLOUD9 ENVIRONMENT:

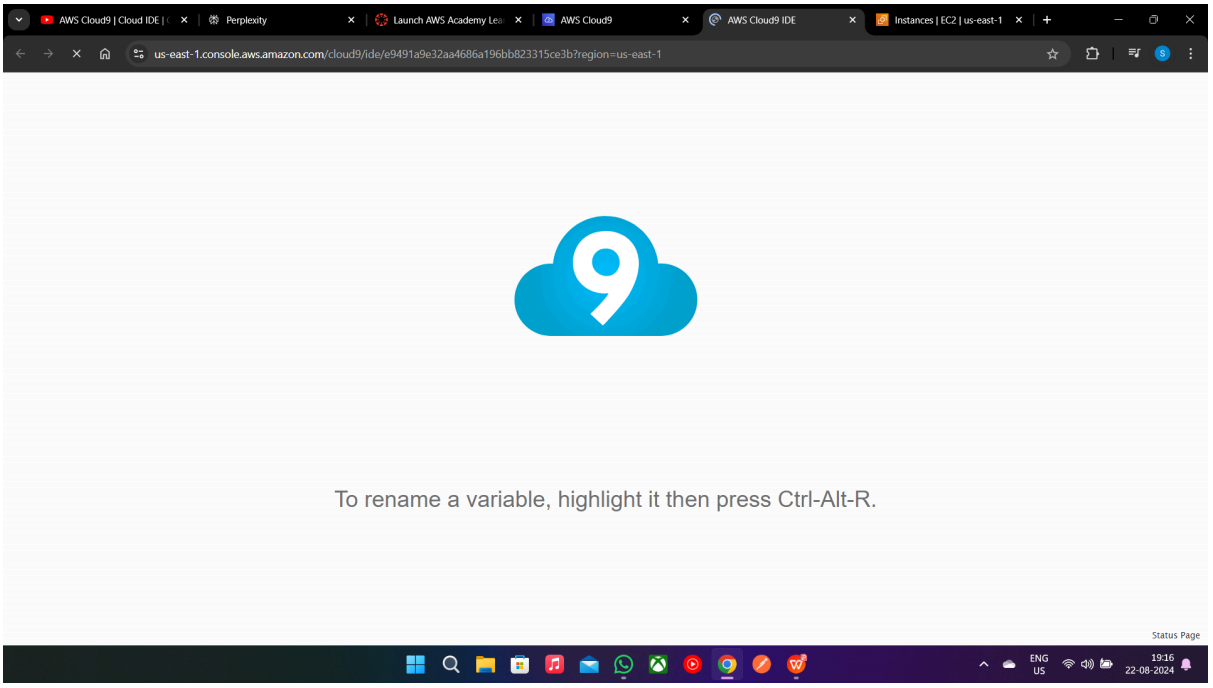


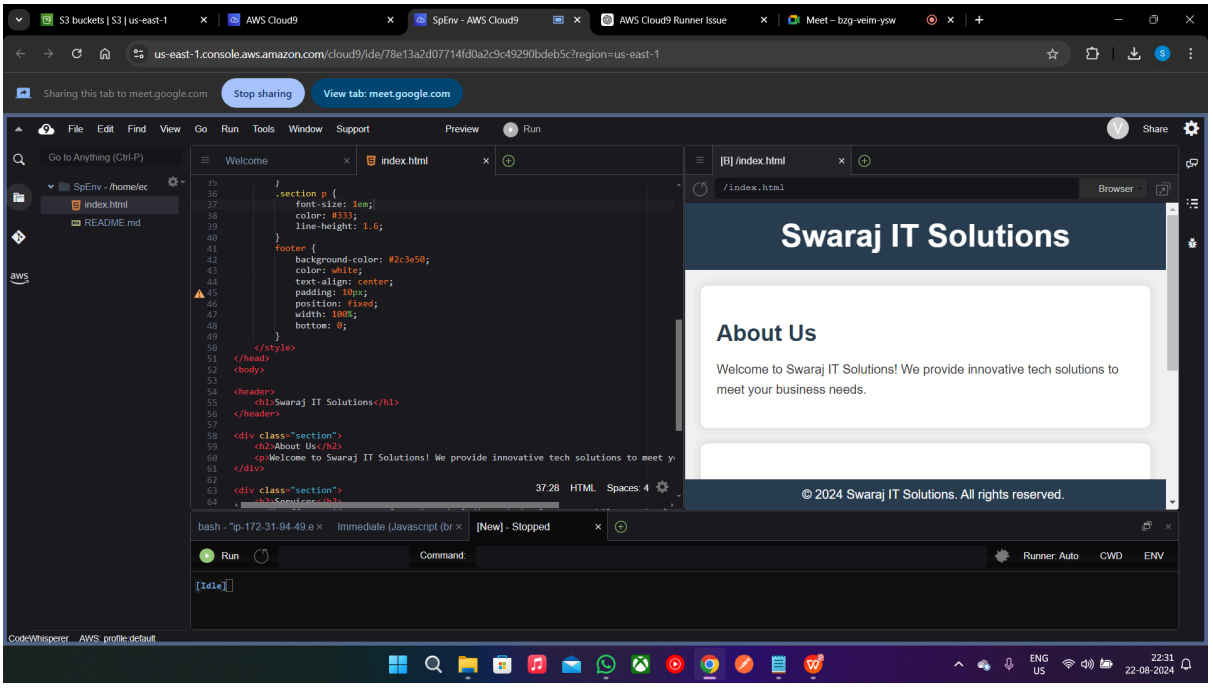
STEP 2:


us-east-1.console.aws.amazon.com/cloud9control/home?region=us-east-1#/create/
Services
ec2
N. Virginia
voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH @ 3185-0688-3614
Network settings Info
Connection
How your environment is accessed.
AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).
Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.
VPC settings Info
Tags - optional Info
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
The following IAM resources will be created in your account
AWSServiceRoleForAWSCloud9 - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
AWSCloud9SSMAccessRole and AWSCloud9SSMInstanceProfile - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. Learn more
Cancel
Create
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/cloud9control/home?region=us-east-1#/
AWS Cloud9
My environments
Shared with me
All account environments
Documentation
Successfully created AWS Cloud9. To get the most out of your environment, see Best practices for using AWS Cloud9
For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. Learn more
AWS Cloud9 > Environments
Environments (1)
Delete
View details
Open in Cloud9
Create environment
My environments
Name
Cloud9 IDE
Environment type
Connection
Permissio n
Owner ARN
AWS Cloud9
Open
EC2 instance
Secure Shell (SSH)
Owner
arn:aws:sts::318506883614:assumed-role/voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

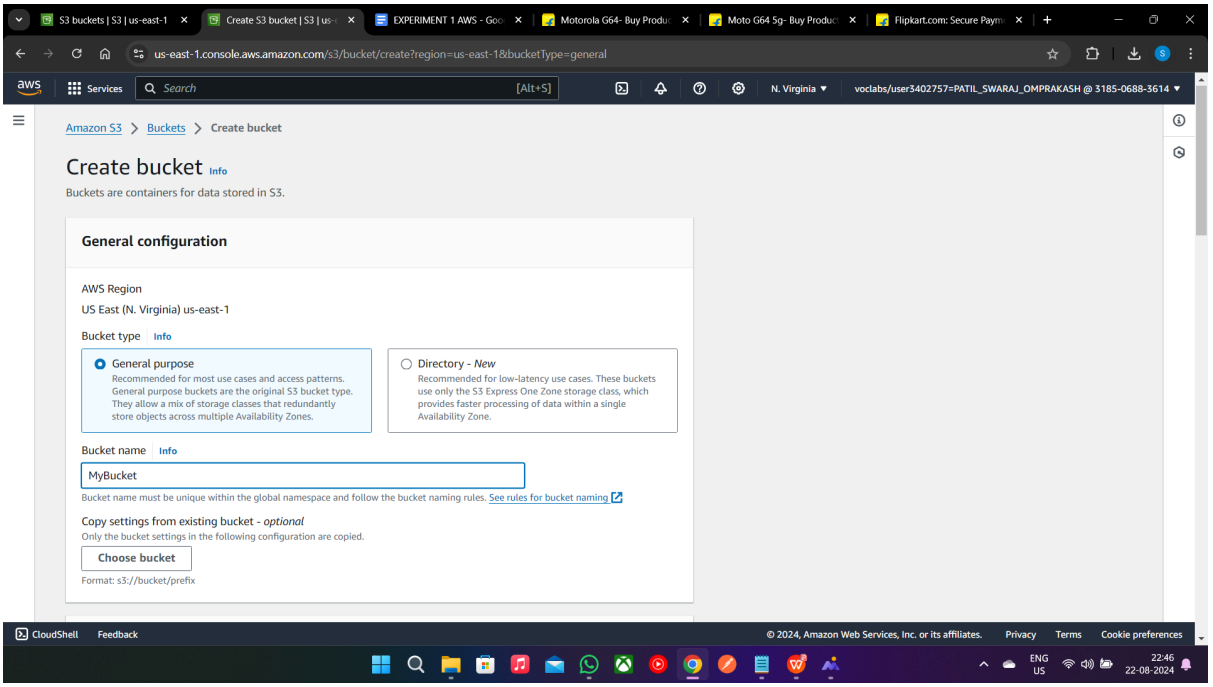
STEP 3: OPEN IN THE CLOUD9 IDE:



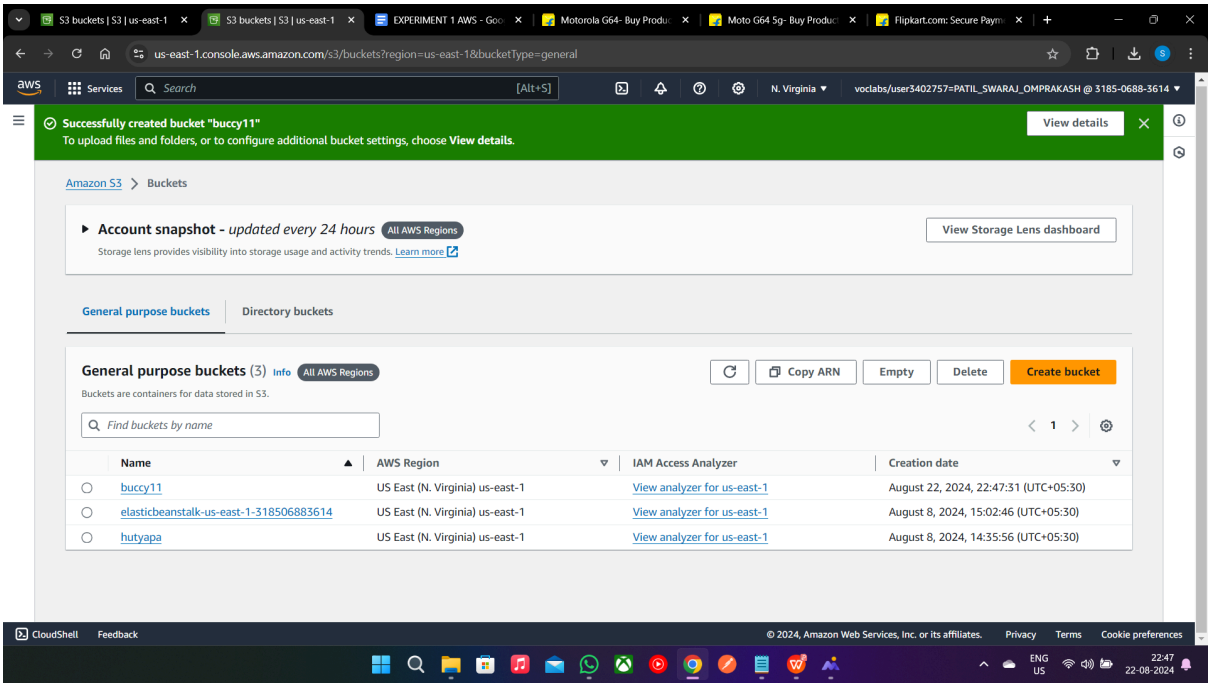


HOSTING USING S3 BUCKET:

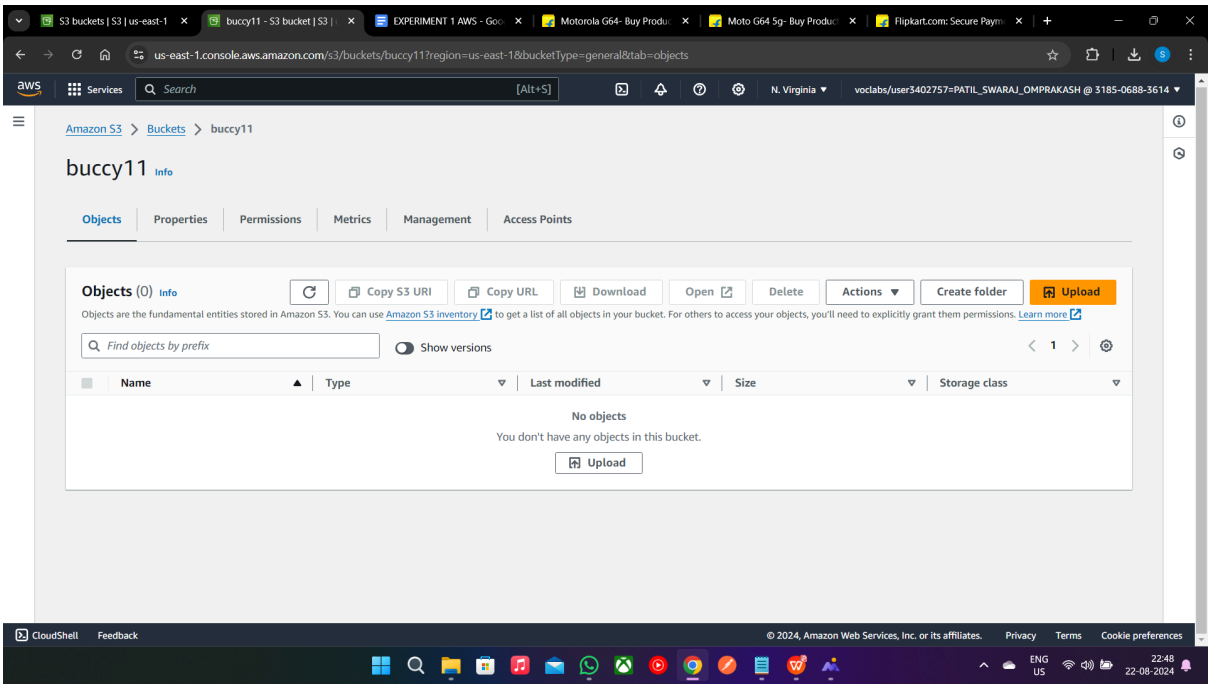
CREATE A BUCKET:

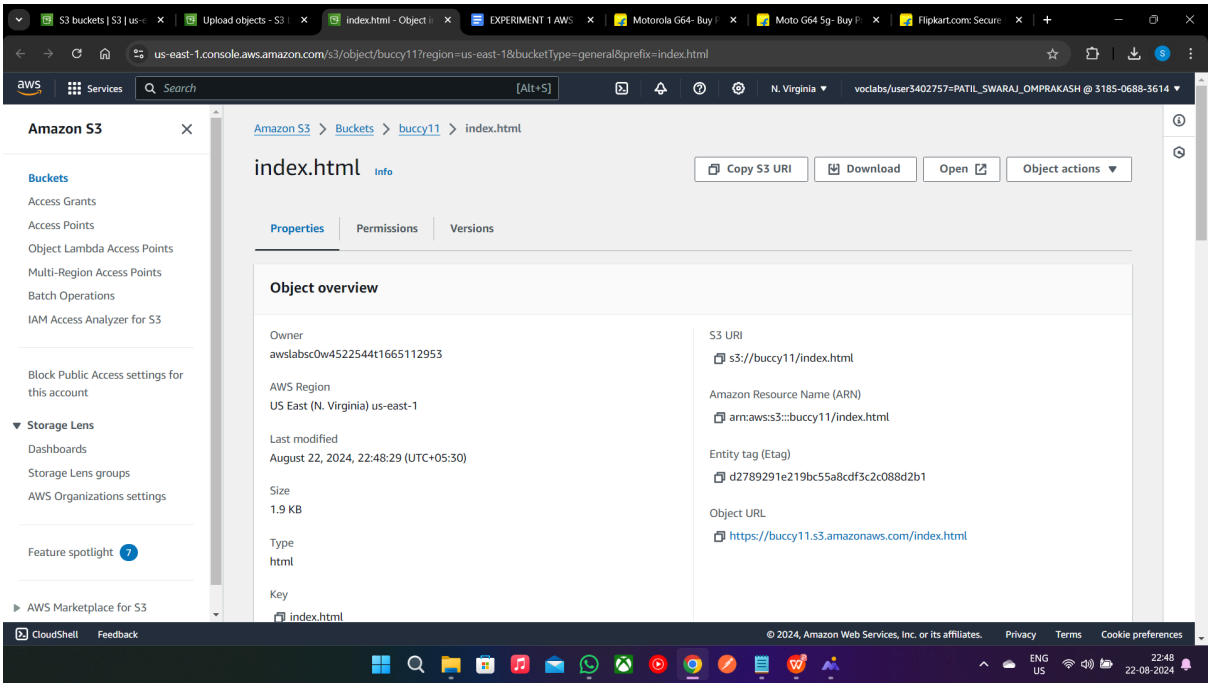


SELECT THE BUCKET:

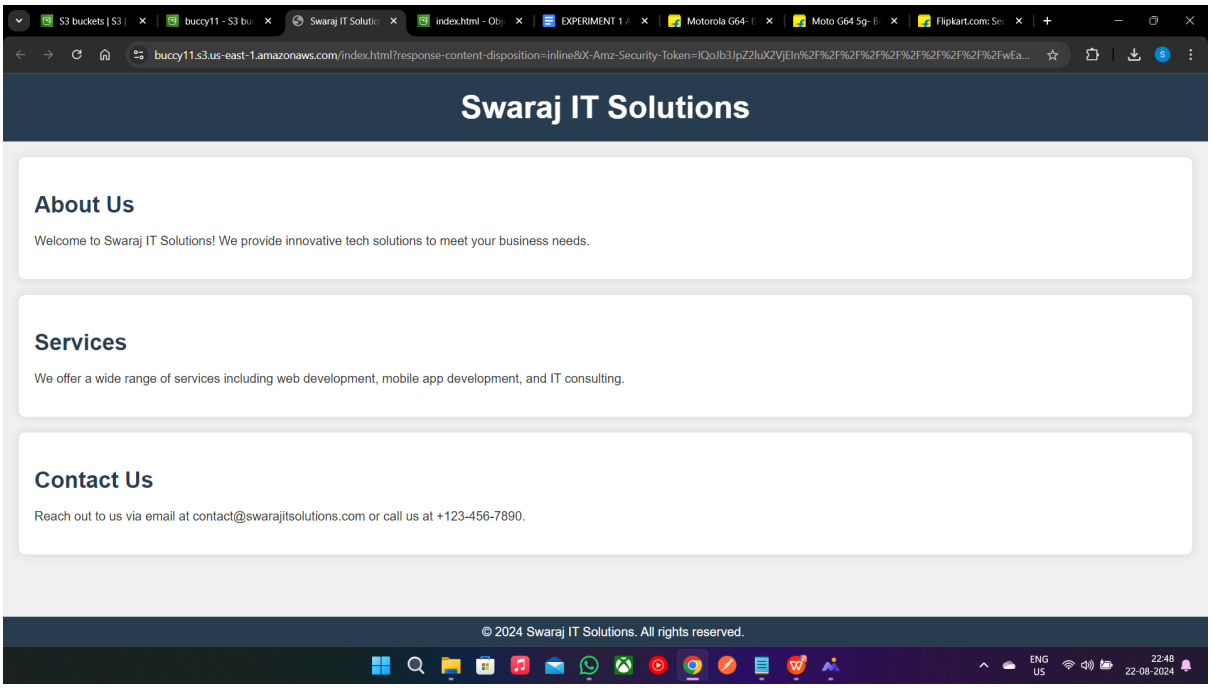


UPLOAD THE HTML FILE:





OUTPUT PAGE:


buccy11.s3.us-east-1.amazonaws.com/index.html?response-content-disposition=inline&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEln%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEa...
Swaraj IT Solutions
About Us
Welcome to Swaraj IT Solutions! We provide innovative tech solutions to meet your business needs.
Services
We offer a wide range of services including web development, mobile app development, and IT consulting.
Contact Us
Reach out to us via email at contact@swarajitsolutions.com or call us at +123-456-7890.
© 2024 Swaraj IT Solutions. All rights reserved.

ADVANCE

DEVOPS EXPERIMENT NO.2

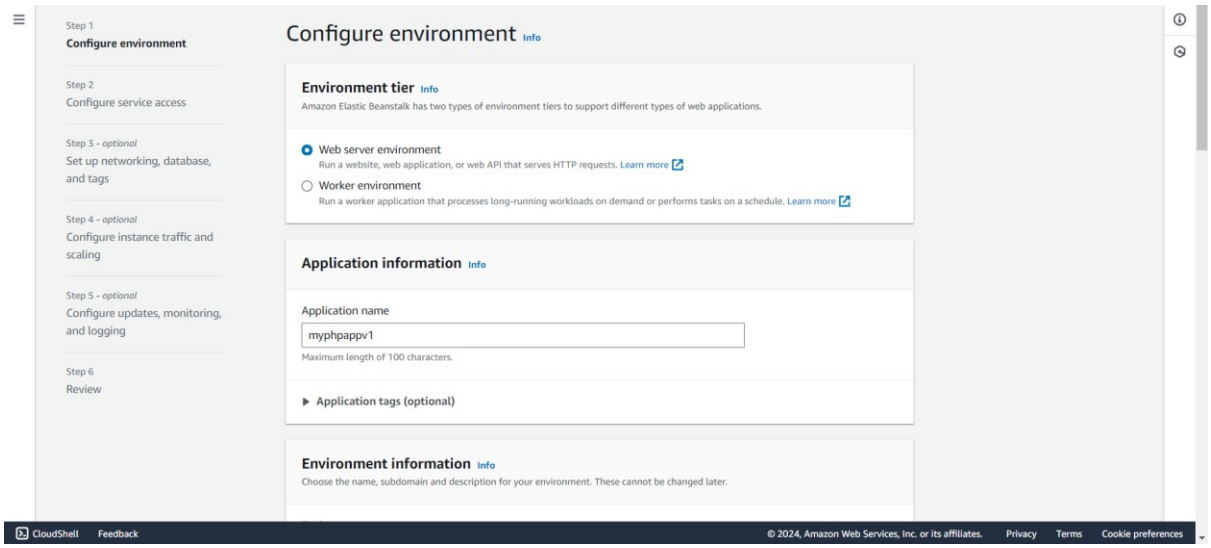
Name: Swaraj Patil

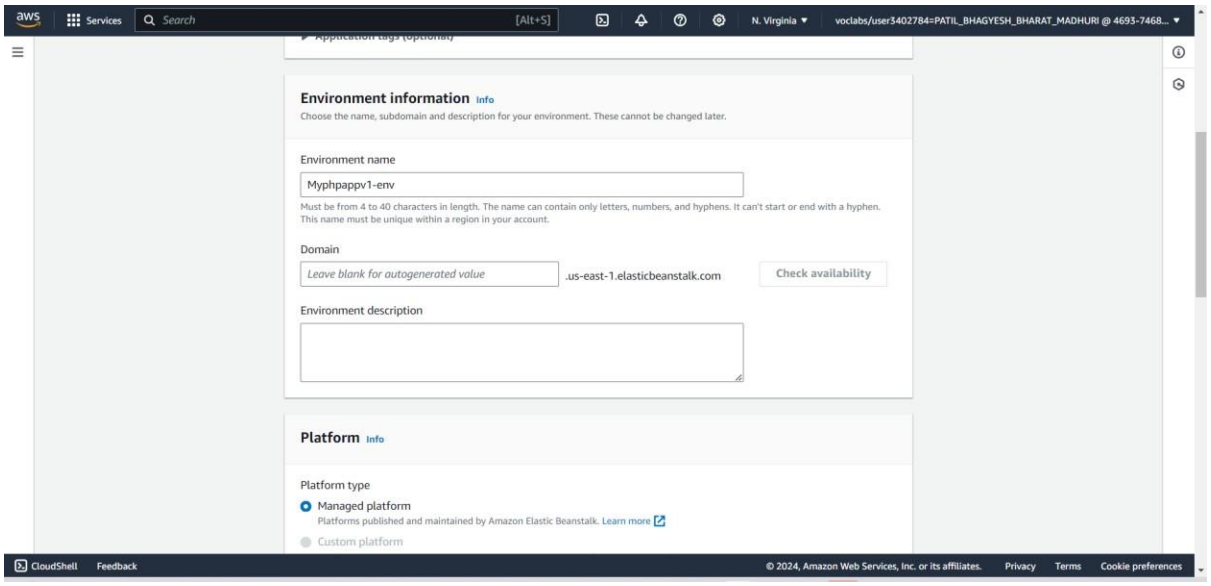
Class:D15A

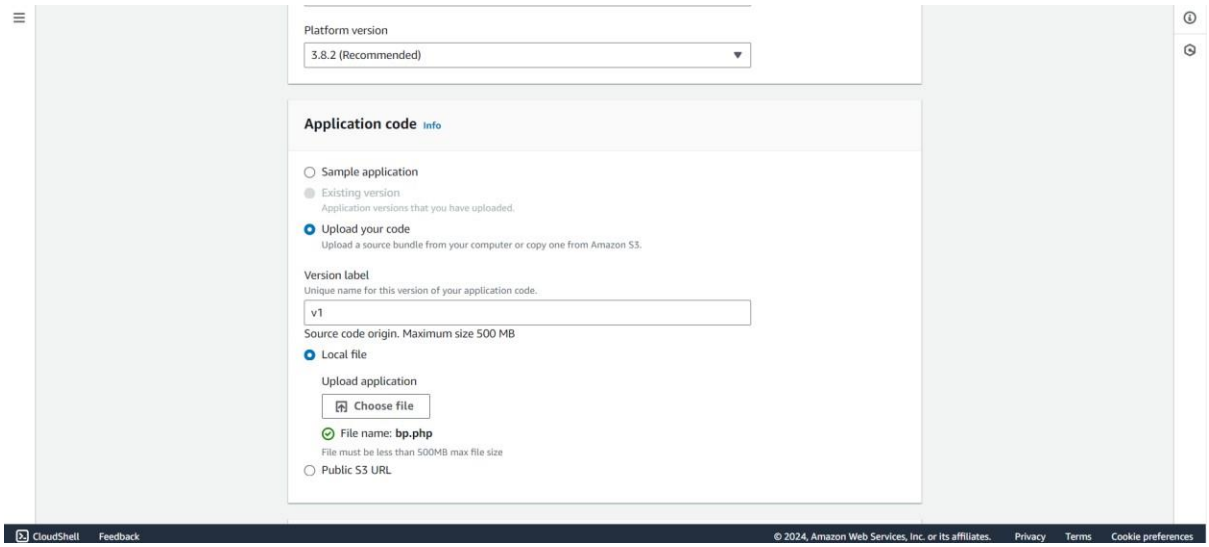
Roll No:40

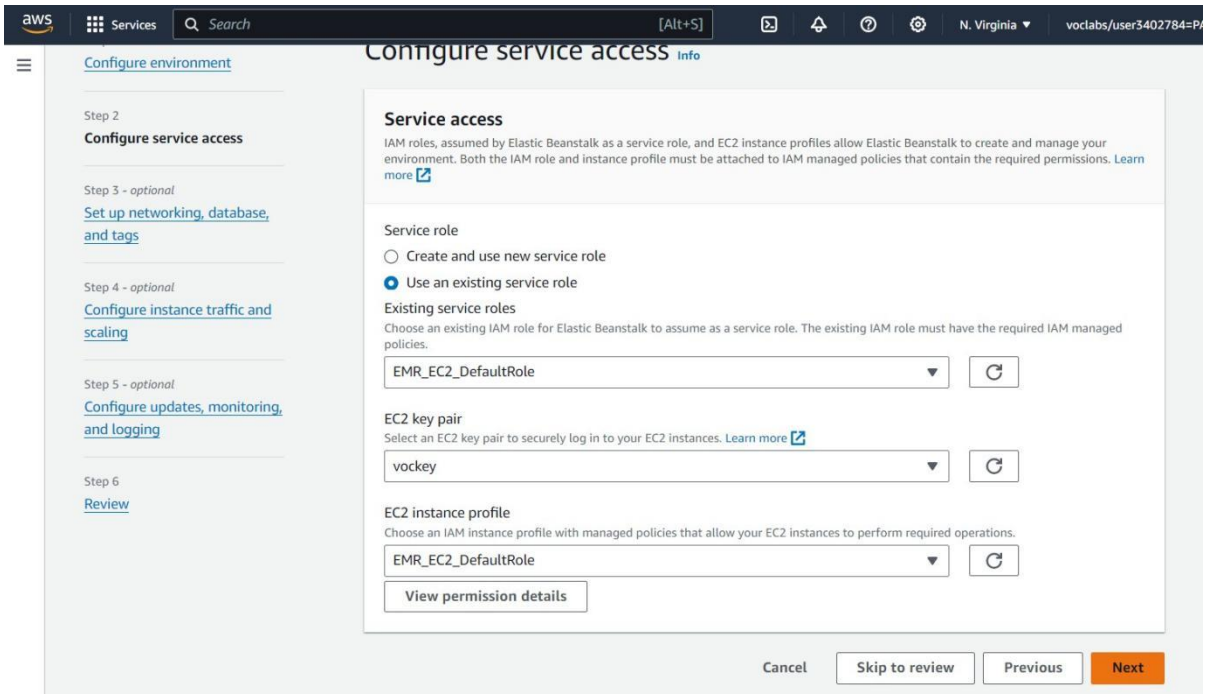
Aim:-

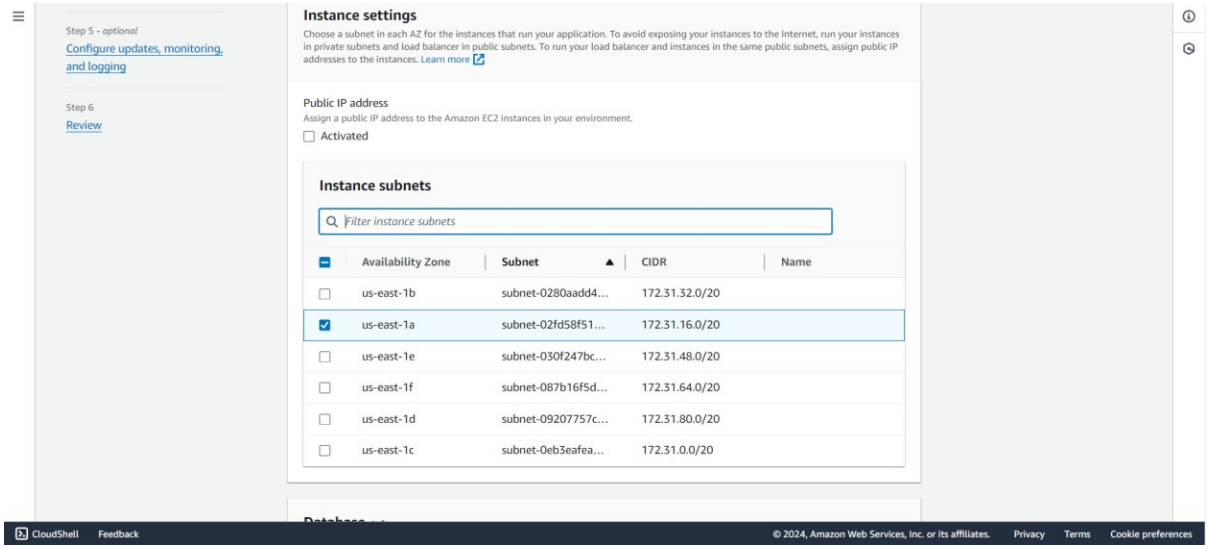
To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWSCodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy

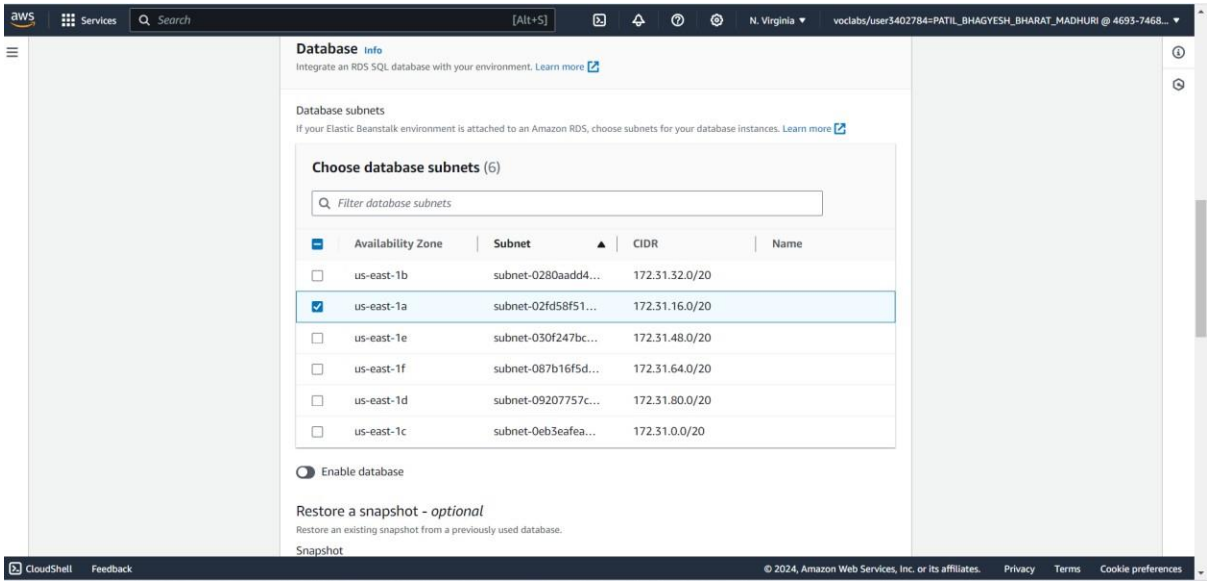


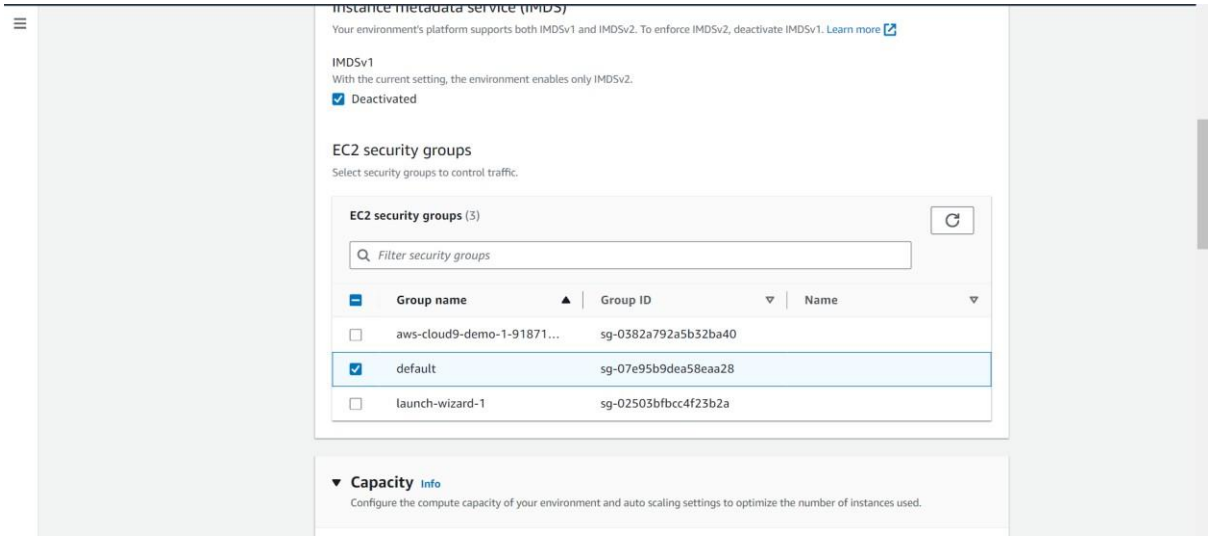


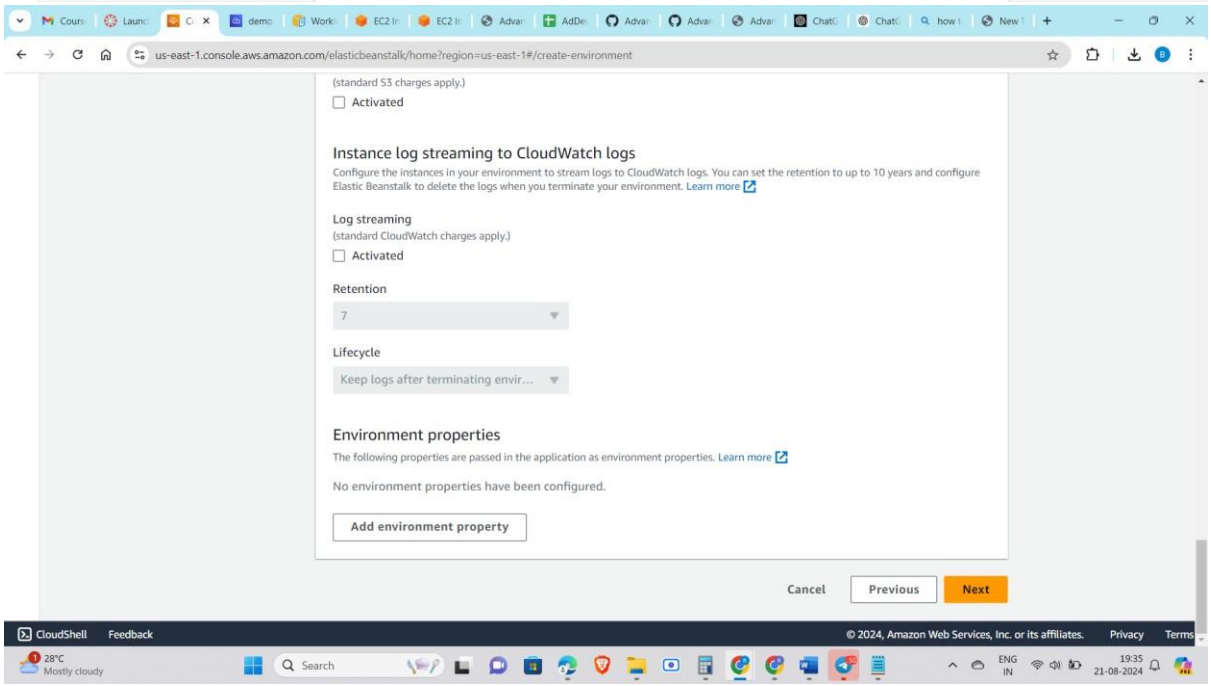

Platform version
3.8.2 (Recommended)
Application code Info
Sample application
Existing version
Application versions that you have uploaded.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Version label
Unique name for this version of your application code.
v1
Source code origin. Maximum size 500 MB
Local file
Upload application
Choose file
File name: bp.php
File must be less than 500MB max file size
Public S3 URL
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

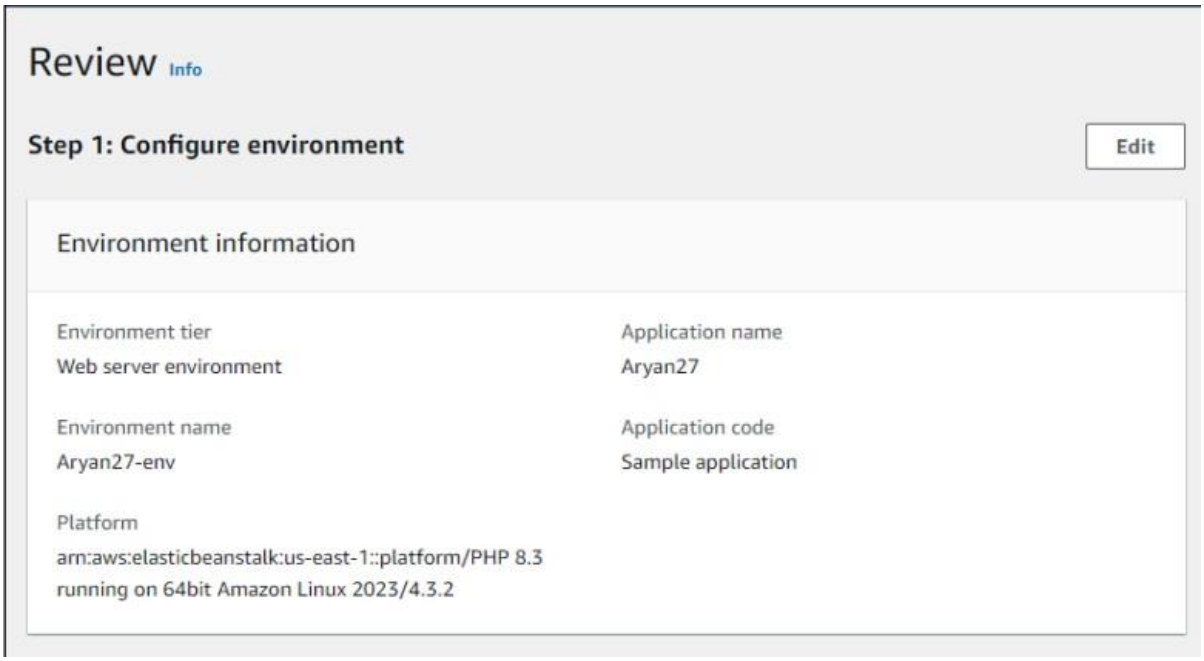

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402784=PA
Configure environment
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Configure service access Info
Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more
Service role
Create and use new service role
Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.
EMR_EC2_DefaultRole
EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. Learn more
vockey
EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
EMR_EC2_DefaultRole
View permission details
Cancel
Skip to review
Previous
Next

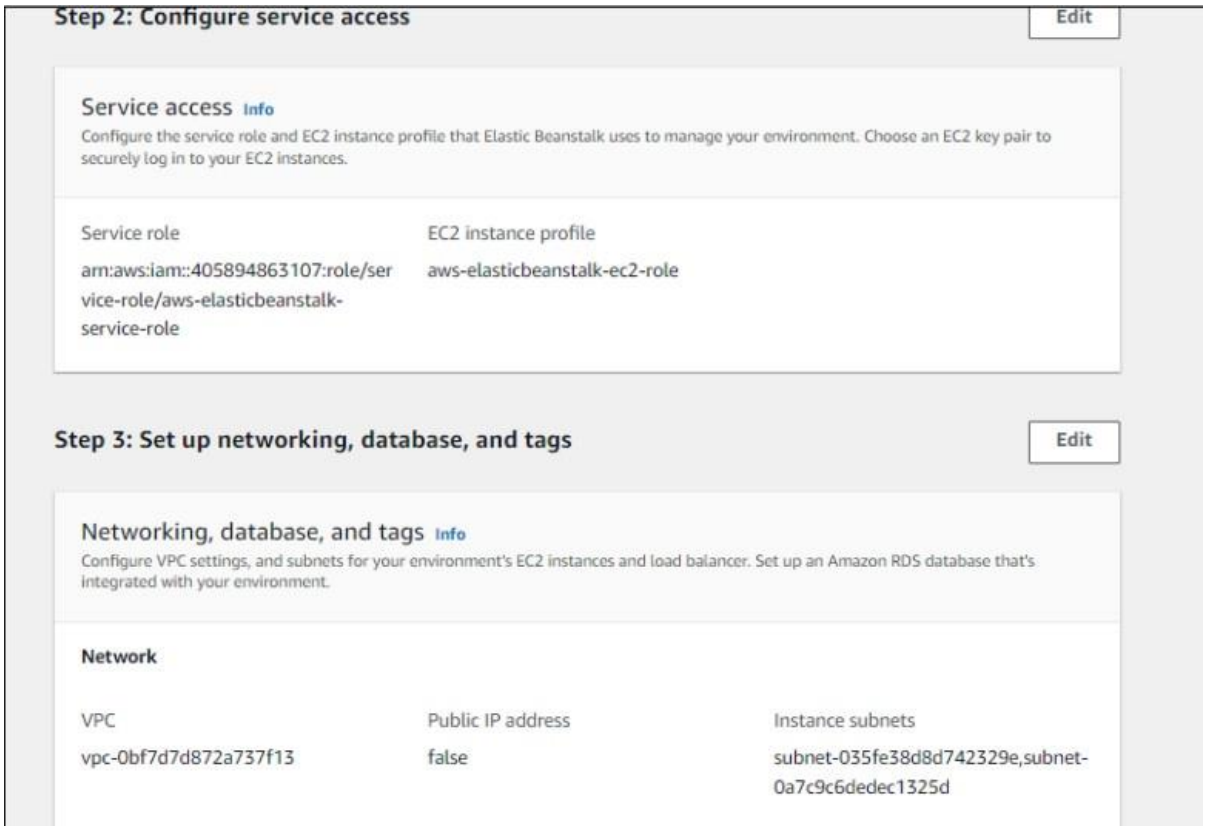

Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
Activated
Instance subnets
Filter instance subnets
Availability Zone
Subnet
CIDR
Name
us-east-1b subnet-0280aadd4... 172.31.32.0/20
us-east-1a subnet-02fd58f51... 172.31.16.0/20
us-east-1e subnet-030f247bc... 172.31.48.0/20
us-east-1f subnet-087b16f5d... 172.31.64.0/20
us-east-1d subnet-09207757c... 172.31.80.0/20
us-east-1c subnet-0eb3eafea... 172.31.0.0/20
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Database Info
Integrate an RDS SQL database with your environment. Learn more
Database subnets
If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. Learn more
Choose database subnets (6)
Filter database subnets
Availability Zone | Subnet | CIDR | Name
us-east-1b | subnet-0280aadd4... | 172.31.32.0/20
us-east-1a | subnet-02fd58f51... | 172.31.16.0/20
us-east-1e | subnet-030f247bc... | 172.31.48.0/20
us-east-1f | subnet-087b16f5d... | 172.31.64.0/20
us-east-1d | subnet-09207757c... | 172.31.80.0/20
us-east-1c | subnet-0eb3eafea... | 172.31.0.0/20
Enable database
Restore a snapshot - optional
Restore an existing snapshot from a previously used database.
Snapshot
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more
IMDSv1
With the current setting, the environment enables only IMDSv2.
Deactivated
EC2 security groups
Select security groups to control traffic.
EC2 security groups (3)
Filter security groups
Group name | Group ID | Name
aws-cloud9-demo-1-91871... | sg-0382a792a5b32ba40
default | sg-07e95b9dea58eaa28
launch-wizard-1 | sg-02503bfbcc4f23b2a
Capacity Info
Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.


us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment
(standard S3 charges apply.)
Activated
Instance log streaming to CloudWatch logs
Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more
Log streaming
(standard CloudWatch charges apply.)
Activated
Retention
7
Lifecycle
Keep logs after terminating envir...
Environment properties
The following properties are passed in the application as environment properties. Learn more
No environment properties have been configured.
Add environment property
Cancel
Previous
Next
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms


Review Info
Step 1: Configure environment
Edit
Environment information
Environment tier
Web server environment
Application name
Aryan27
Environment name
Aryan27-env
Application code
Sample application
Platform
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2


Step 2: Configure service access
Edit
Service access Info
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.
Service role
arn:aws:iam::405894863107:role/service-role/aws-elasticbeanstalk-service-role
EC2 instance profile
aws-elasticbeanstalk-ec2-role
Step 3: Set up networking, database, and tags
Edit
Networking, database, and tags Info
Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.
Network
VPC
vpc-0bf7d7d872a737f13
Public IP address
false
Instance subnets
subnet-035fe38d8d742329e,subnet-0a7c9c6dedec1325d

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

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

Monitoring

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

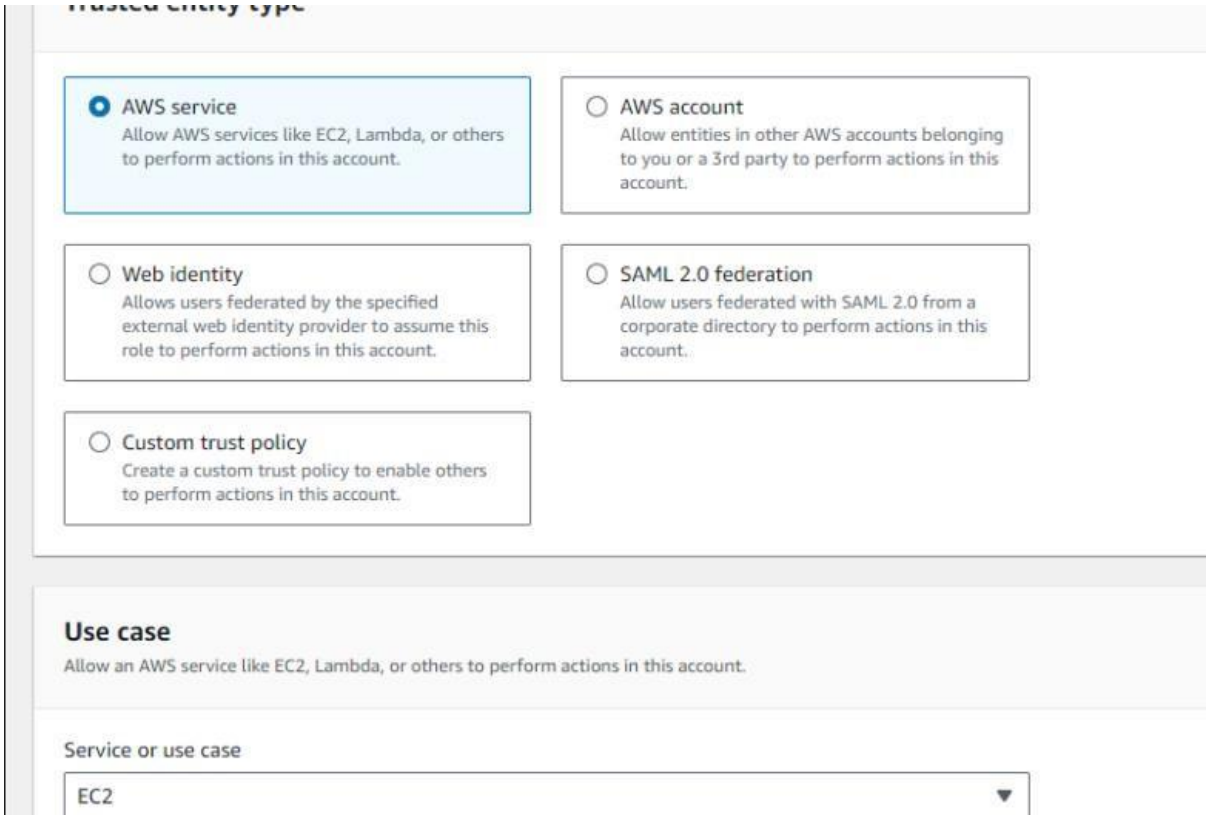
Updates

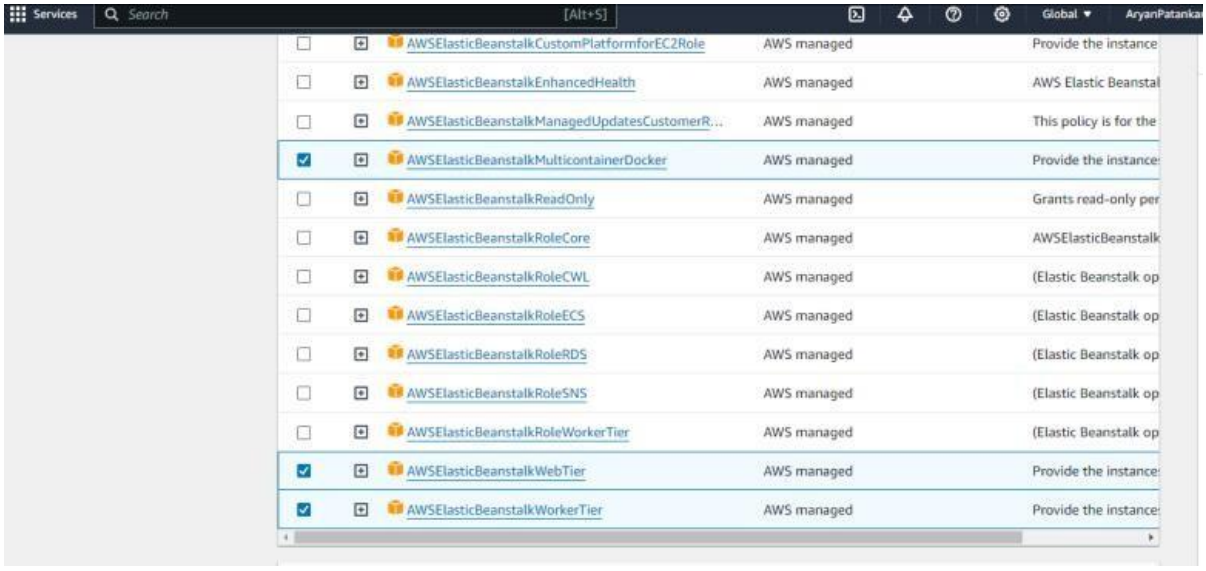
Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage

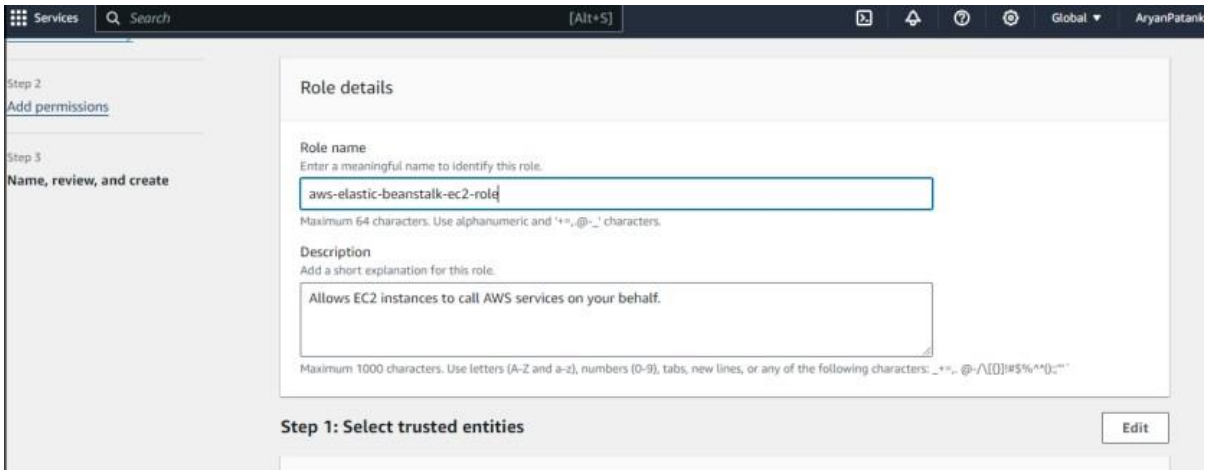
Platform software

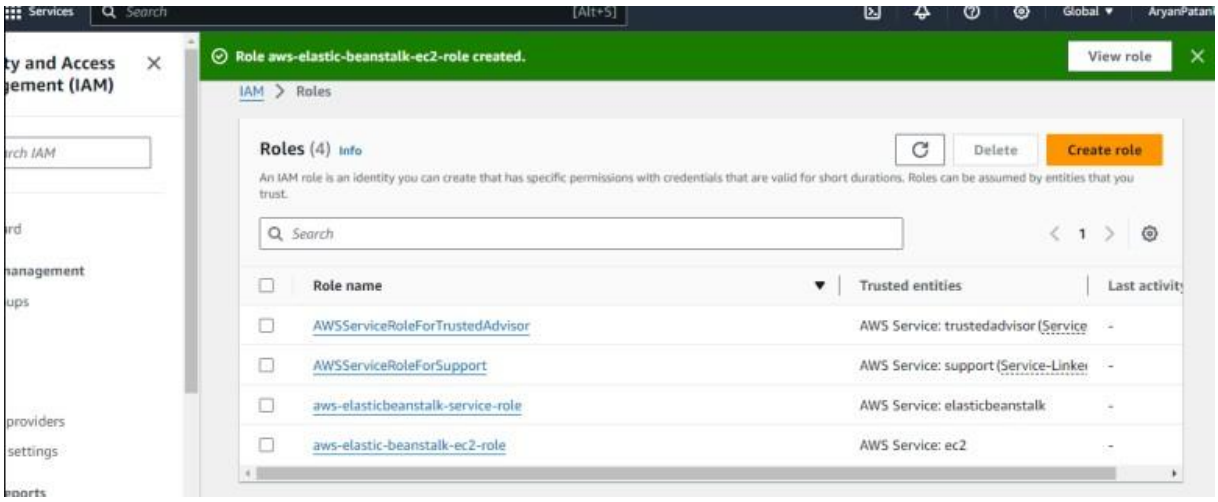
Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	**Document root**	**Max execution time**
Off	–	60
Memory limit	**Zlib output compression**	**Proxy server**
256M	Off	nginx
Logs retention	**Rotate logs**	**Update level**
7	Deactivated	minor

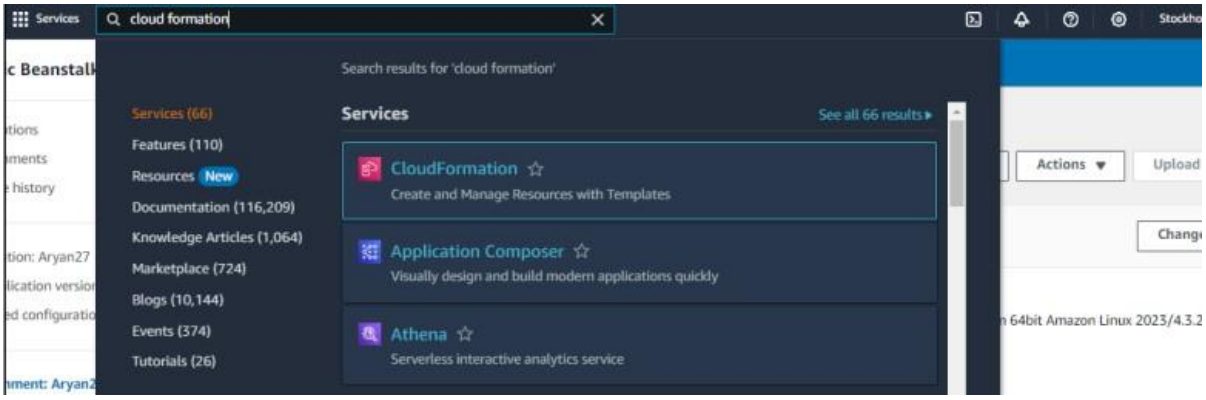
X-Ray enabled

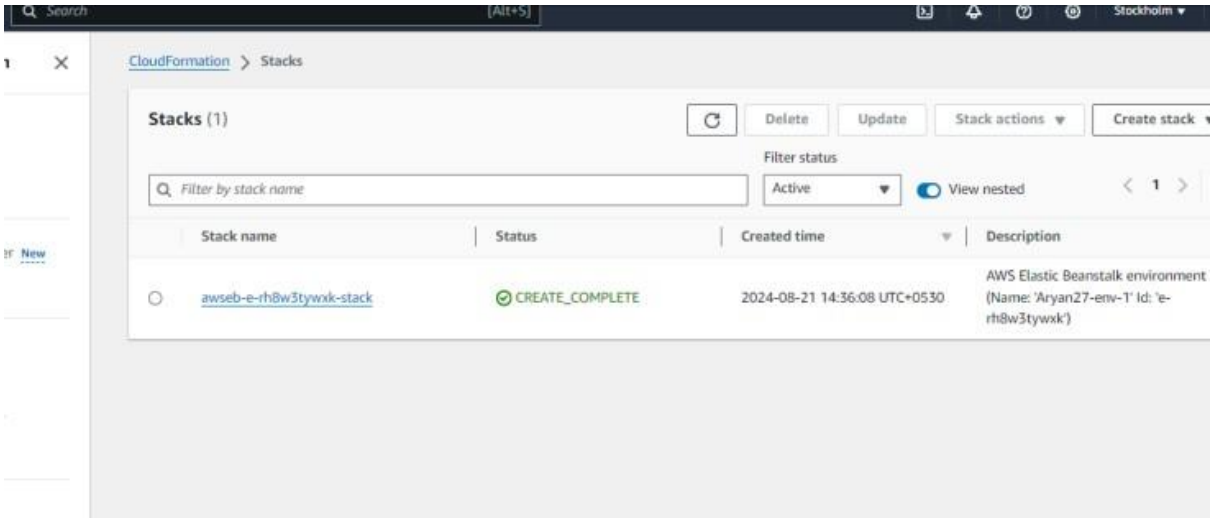

AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume 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

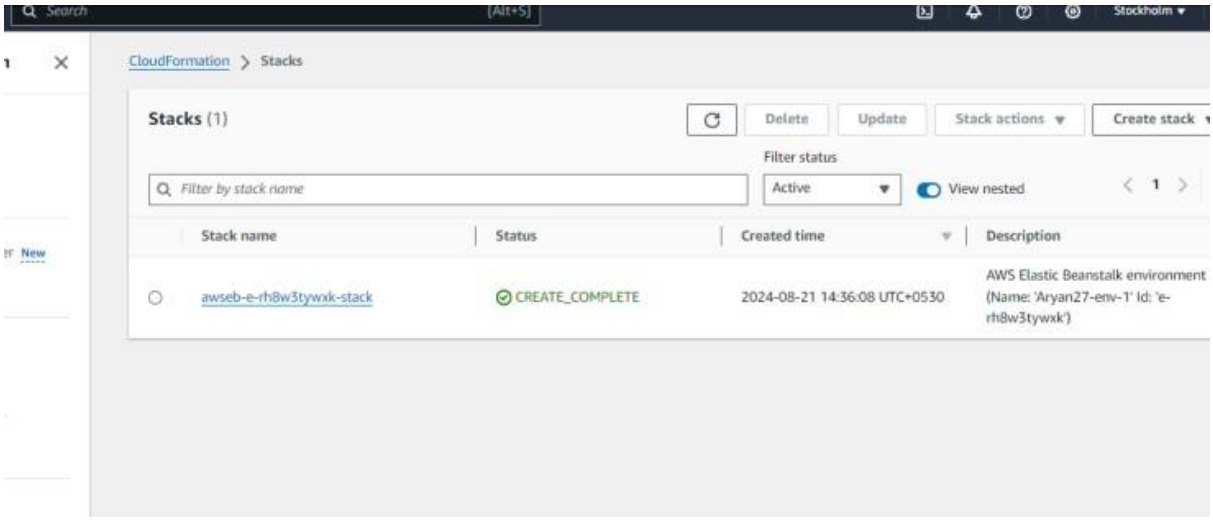

Services
Search
[Alt+S]
Global
AWSElasticBeanstalkCustomPlatformforEC2Role
AWS managed
Provide the instance
AWSElasticBeanstalkEnhancedHealth
AWS managed
AWS Elastic Beanstal
AWSElasticBeanstalkManagedUpdatesCustomerR...
AWS managed
This policy is for the
AWSElasticBeanstalkMulticontainerDocker
AWS managed
Provide the instance
AWSElasticBeanstalkReadOnly
AWS managed
Grants read-only per
AWSElasticBeanstalkRoleCore
AWS managed
AWSElasticBeanstalk
AWSElasticBeanstalkRoleCWL
AWS managed
(Elastic Beanstalk op
AWSElasticBeanstalkRoleECS
AWS managed
(Elastic Beanstalk op
AWSElasticBeanstalkRoleRDS
AWS managed
(Elastic Beanstalk op
AWSElasticBeanstalkRoleSNS
AWS managed
(Elastic Beanstalk op
AWSElasticBeanstalkRoleWorkerTier
AWS managed
(Elastic Beanstalk op
AWSElasticBeanstalkWebTier
AWS managed
Provide the instance
AWSElasticBeanstalkWorkerTier
AWS managed
Provide the instance

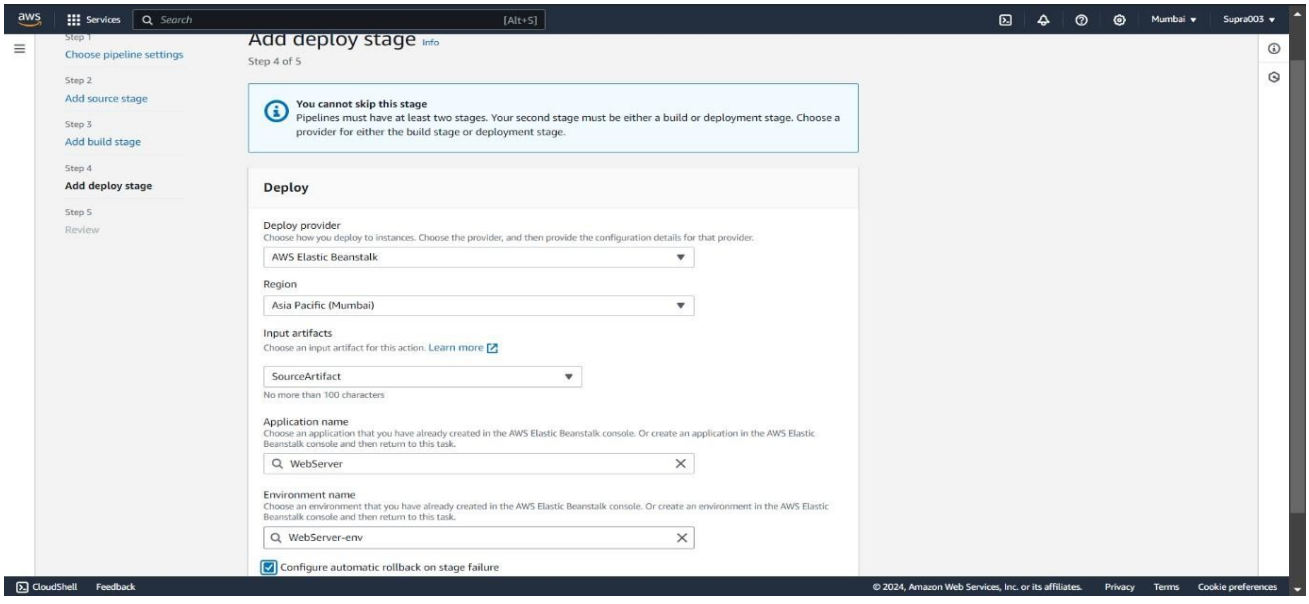

Services
Search
[Alt+S]
Global
Step 2
Add permissions
Step 3
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
aws-elastic-beanstalk-ec2-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
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

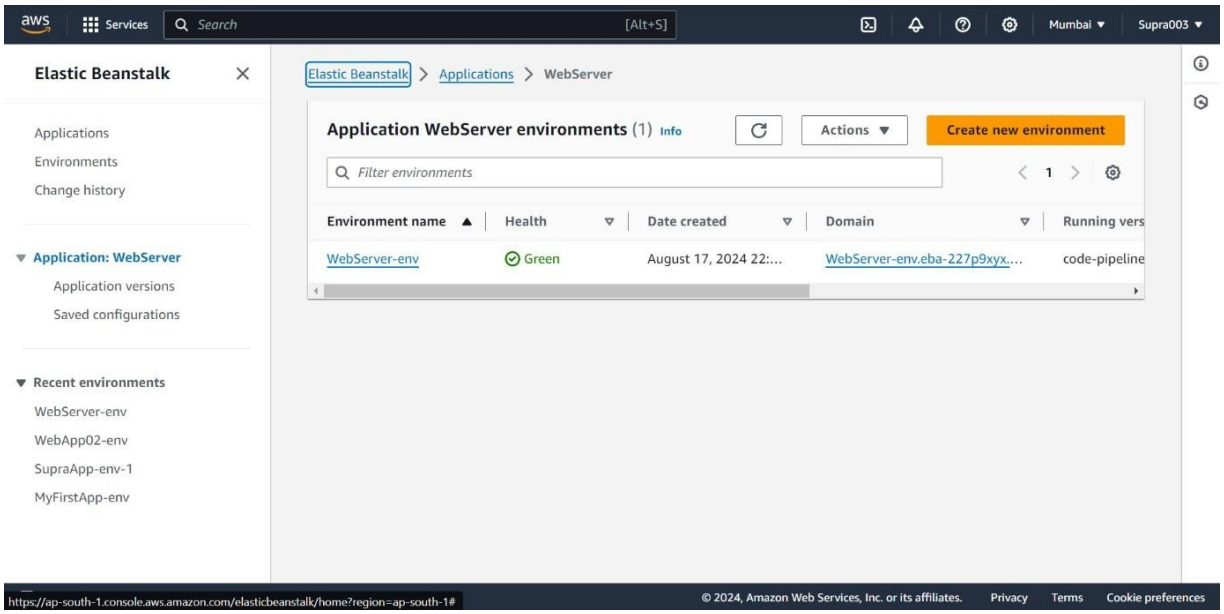

Role aws-elastic-beanstalk-ec2-role created.
View role
IAM > Roles
Roles (4) Info
Delete
Create role
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Search
Role name
Trusted entities
Last activity
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
aws-elasticbeanstalk-service-role
AWS Service: elasticbeanstalk
aws-elastic-beanstalk-ec2-role
AWS Service: ec2

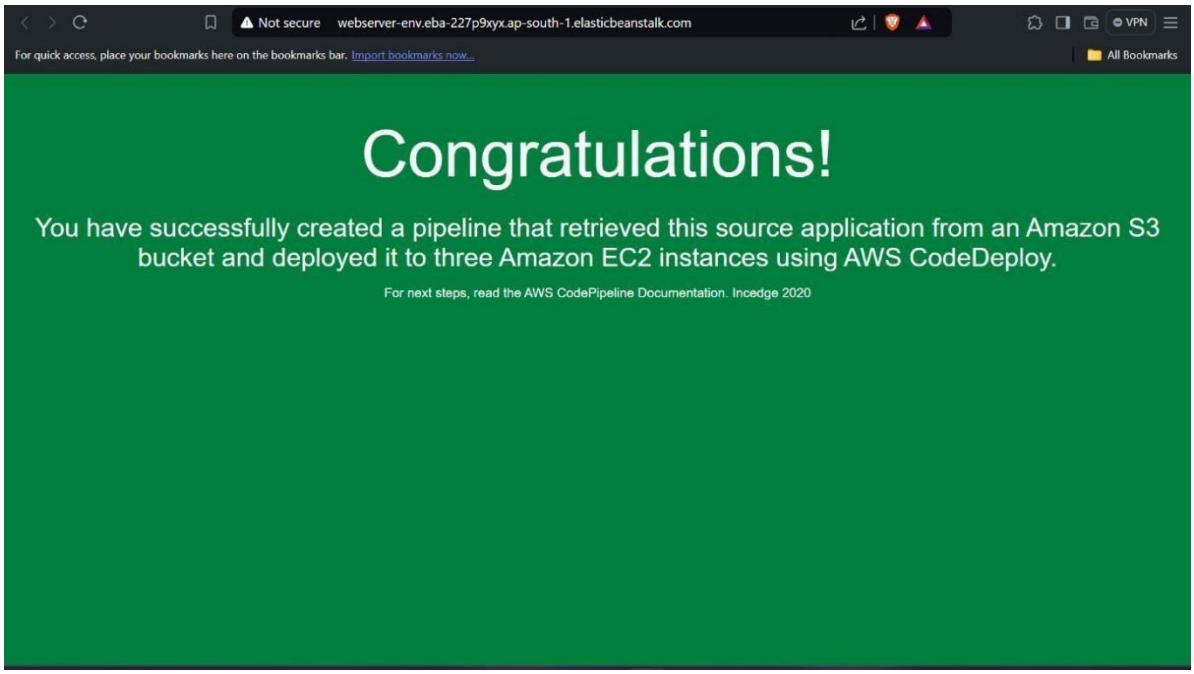

Services
cloud formation
Search results for 'cloud formation'
Services (66)
Features (110)
Resources New
Documentation (116,209)
Knowledge Articles (1,064)
Marketplace (724)
Blogs (10,144)
Events (374)
Tutorials (26)
Services
See all 66 results
CloudFormation
Create and Manage Resources with Templates
Application Composer
Visually design and build modern applications quickly
Athena
Serverless interactive analytics service
Actions
Upload


Search
[Alt+S]
Stockholm
CloudFormation > Stacks
Stacks (1)
Delete
Update
Stack actions
Create stack
Filter status
Filter by stack name
Active
View nested
1
Stack name
Status
Created time
Description
awseb-e-rh8w3tywxk-stack
CREATE_COMPLETE
2024-08-21 14:36:08 UTC+0530
AWS Elastic Beanstalk environment (Name: 'Aryan27-env-1' Id: 'e-rh8w3tywxk')

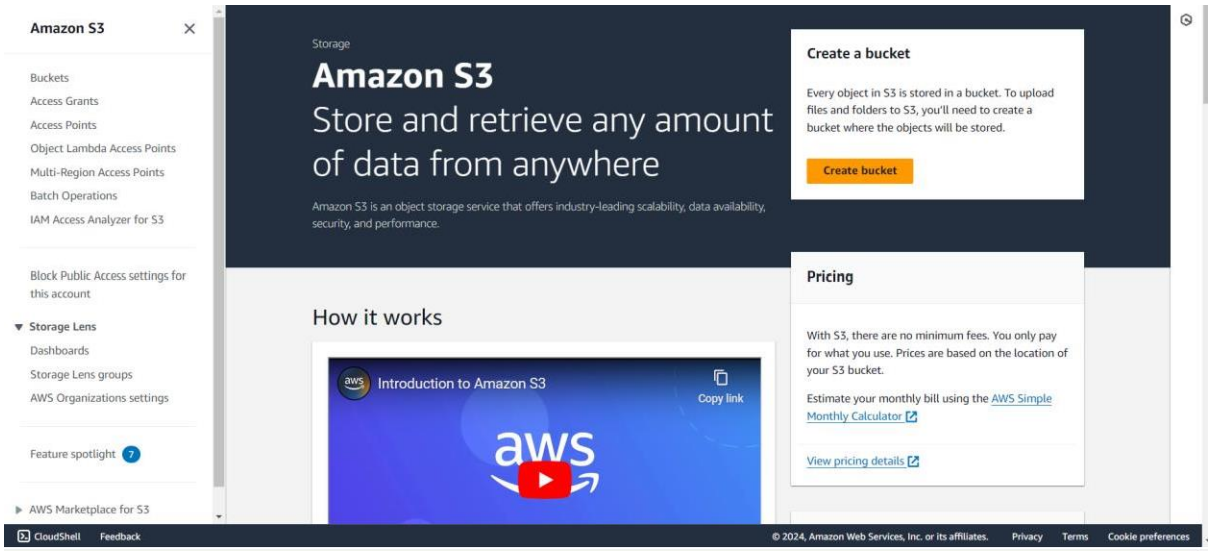

Search
[Alt+S]
Stockholm
CloudFormation > Stacks
Stacks (1)
Delete
Update
Stack actions
Create stack
Filter status
Filter by stack name
Active
View nested
1
Stack name
Status
Created time
Description
awseb-e-rh8w3tywxk-stack
CREATE_COMPLETE
2024-08-21 14:36:08 UTC+0530
AWS Elastic Beanstalk environment (Name: 'Aryan27-env-1' Id: 'e-rh8w3tywxk')


Services
Search
[Alt+S]
Mumbai
Supra003
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add deploy stage Info
Step 4 of 5
You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.
Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk
Region
Asia Pacific (Mumbai)
Input artifacts
Choose an input artifact for this action. Learn more
SourceArtifact
No more than 100 characters
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
WebServer
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
WebServer-env
Configure automatic rollback on stage failure
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Mumbai
Supra003
Elastic Beanstalk
Applications
Environments
Change history
Application: WebServer
Application versions
Saved configurations
Recent environments
WebServer-env
WebApp02-env
SupraApp-env-1
MyFirstApp-env
Elastic Beanstalk > Applications > WebServer
Application WebServer environments (1) Info
Actions
Create new environment
Filter environments
Environment name
Health
Date created
Domain
Running vers
WebServer-env
Green
August 17, 2024 22:...
WebServer-env.eba-227p9xyx....
code-pipeline
https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


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

Using s3 bucket



General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

- (●) General purpose
  Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- ( ) Directory - *New*
  Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

hutiypa

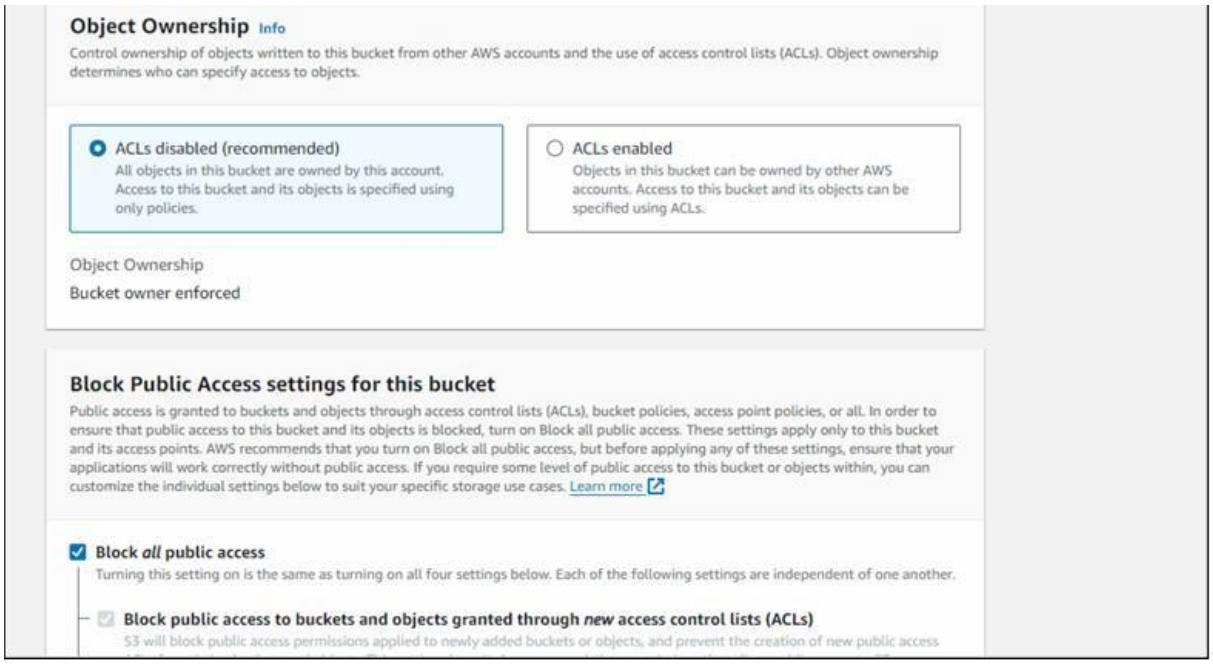
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

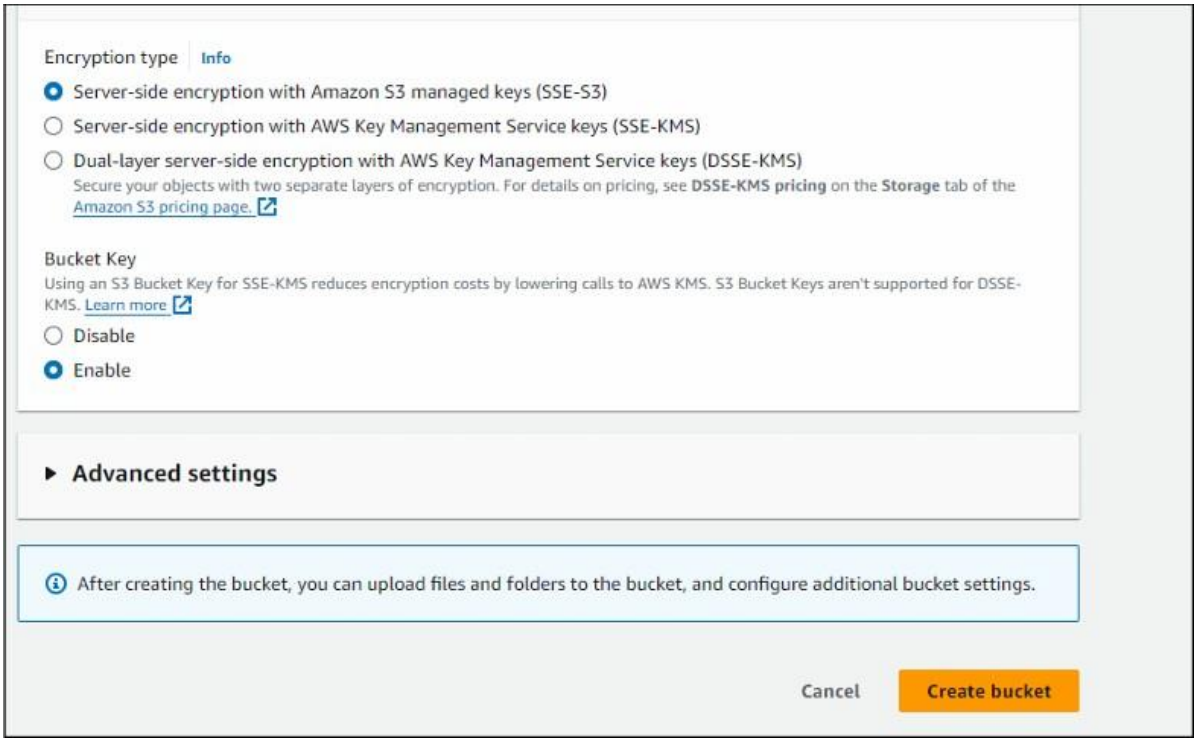
Copy settings from existing bucket - *optional*

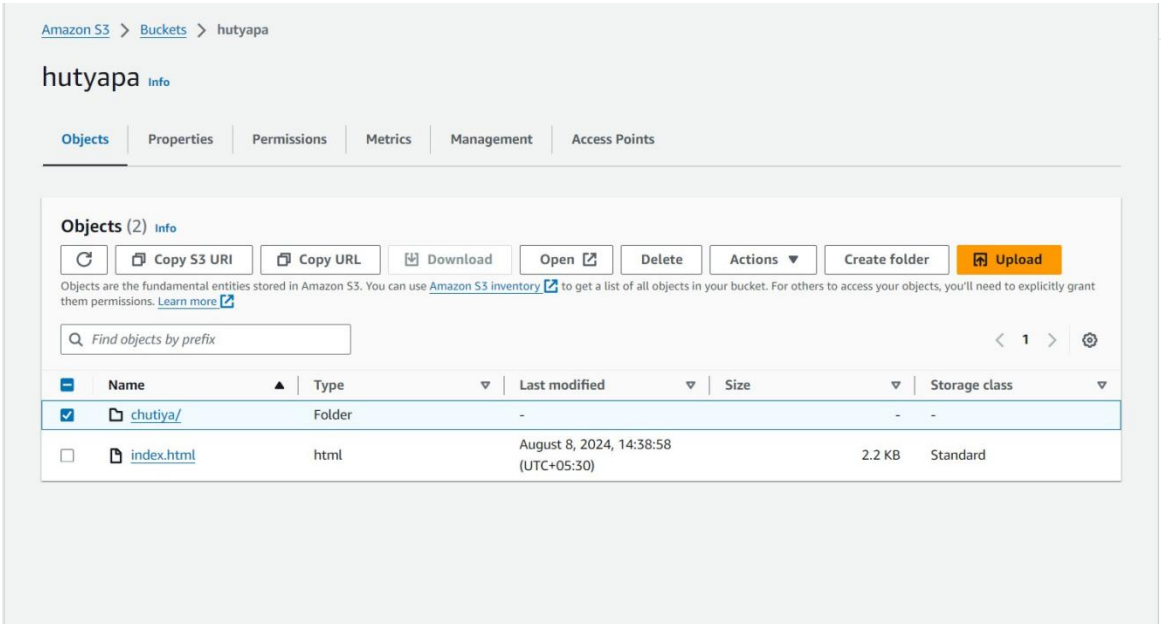
Only the bucket settings in the following configuration are copied.

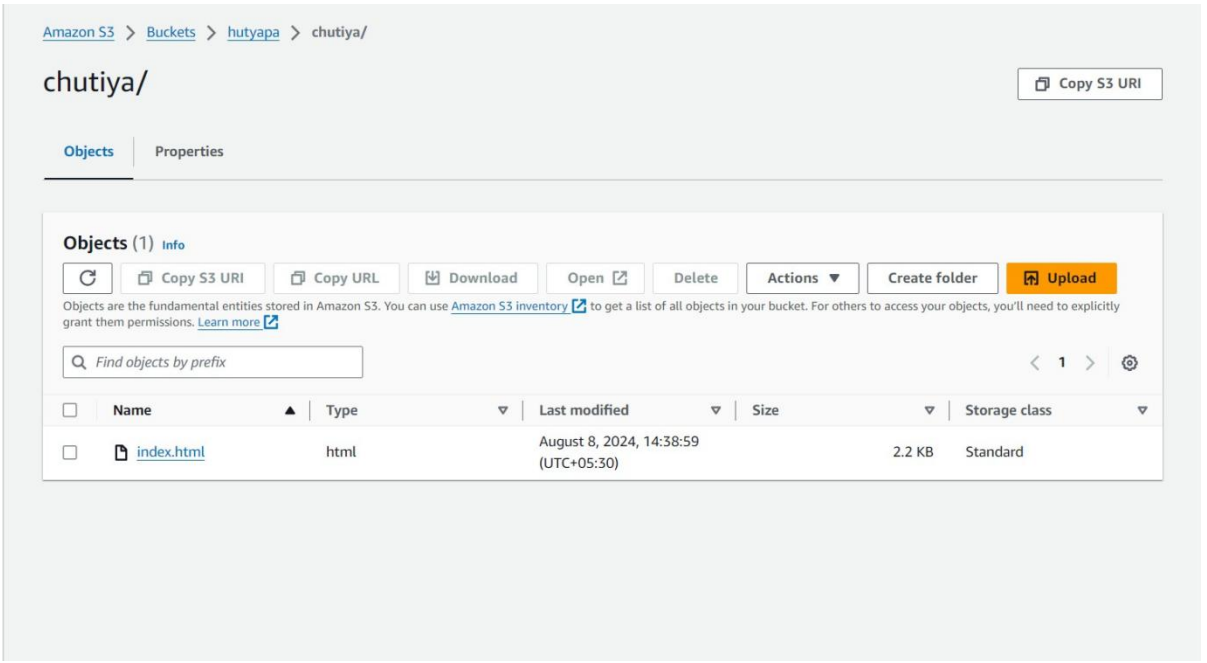
Choose bucket

Format: s3://bucket/prefix


Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access


Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket


Amazon S3 > Buckets > hutyapa
hutyapa Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (2) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
chutiya/
Folder
-
-
-
index.html
html
August 8, 2024, 14:38:58 (UTC+05:30)
2.2 KB
Standard


Amazon S3 > Buckets > hutyapa > chutiya/
chutiya/
Copy S3 URI
Objects
Properties
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
index.html
html
August 8, 2024, 14:38:59 (UTC+05:30)
2.2 KB
Standard

About This Page

This is a simple and elegant web page designed with a modern and clean UI. The layout includes a header, a content area, and a footer, all styled to provide a pleasant user experience.

Learn More

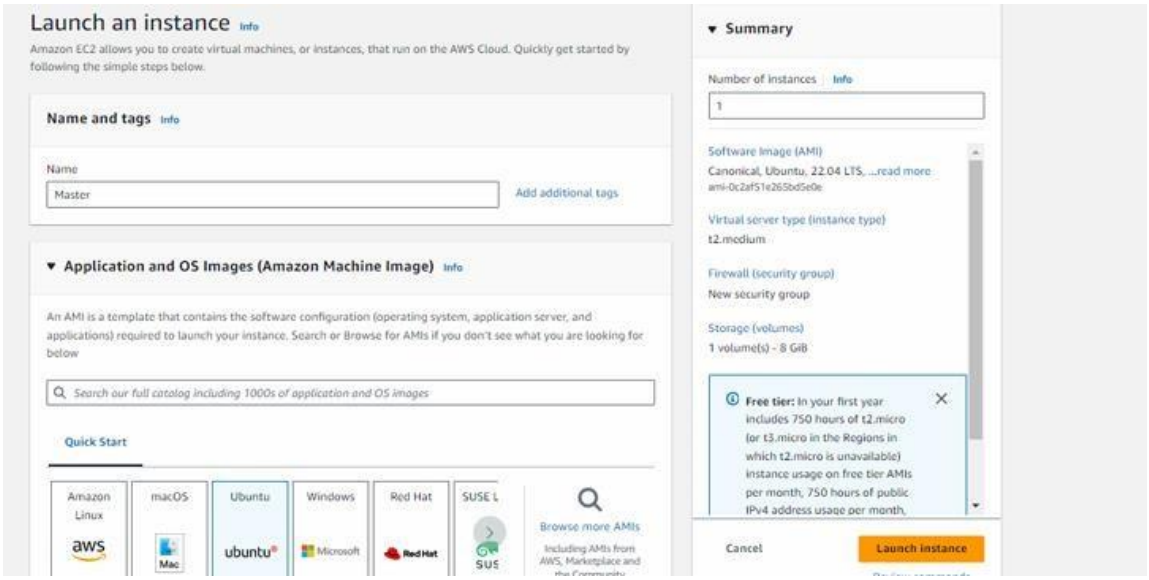
ADVANCE DEVOPS EXP 3

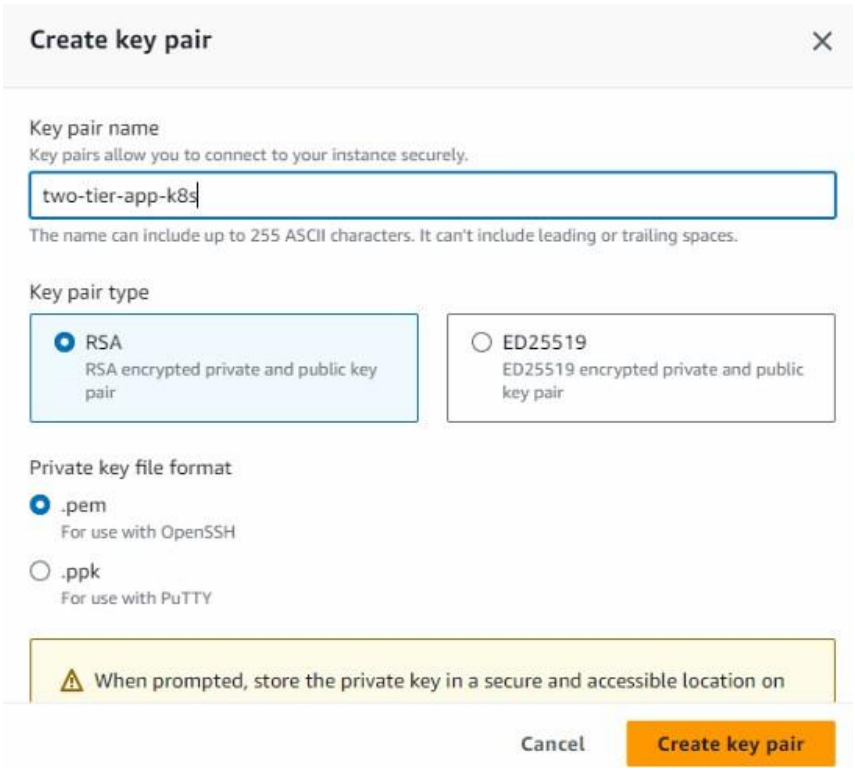
Name :- Swaraj Patil
Roll no :- 40

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

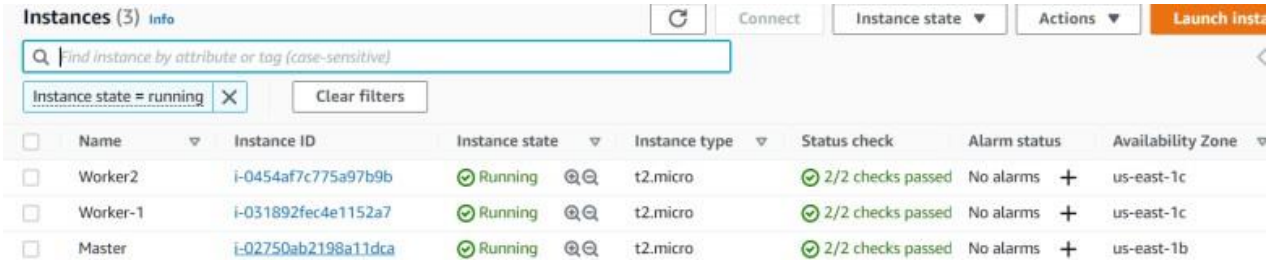
Step 1: Pre-requisites

1.1 Create 3 EC2 instances,one for the master node and two for the worker nodes.

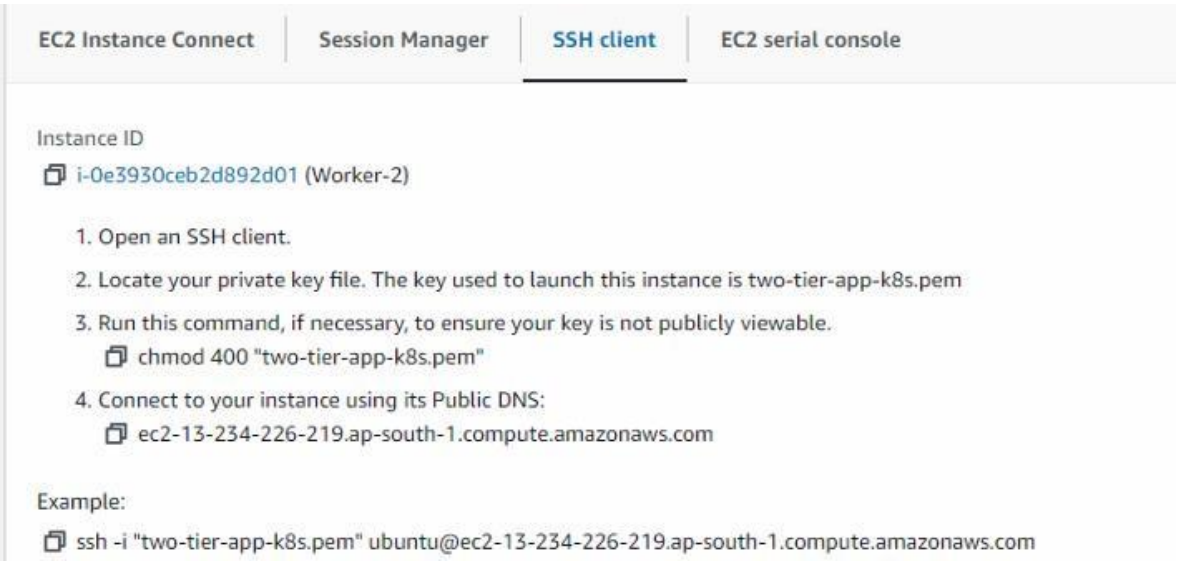




Create 3 EC2 Ubuntu Instances of Ubuntu version 20.04 and keep all the instances in the same security group on AWS. (Name 1 as Master, the other 2 as worker-1 and worker-2)



Now the ssh created, copy the text given in the example




```
acer@TMP214-53 MINGW64 ~/Downloads
$ ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-232-36-34.ap-south-1.compute.amazo
naws.com
The authenticity of host 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com (13.
232.36.34)' can't be established.
ED25519 key fingerprint is SHA256:uVGEO+FWYefj6OjOft7OSralv8NrzEi/IwxAtBY+EPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com' (
ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Sep 11 14:07:10 UTC 2024

  System load:  0.0               Processes:             106
  Usage of /:   20.7% of 7.57GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for eth0: 172.31.45.227
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

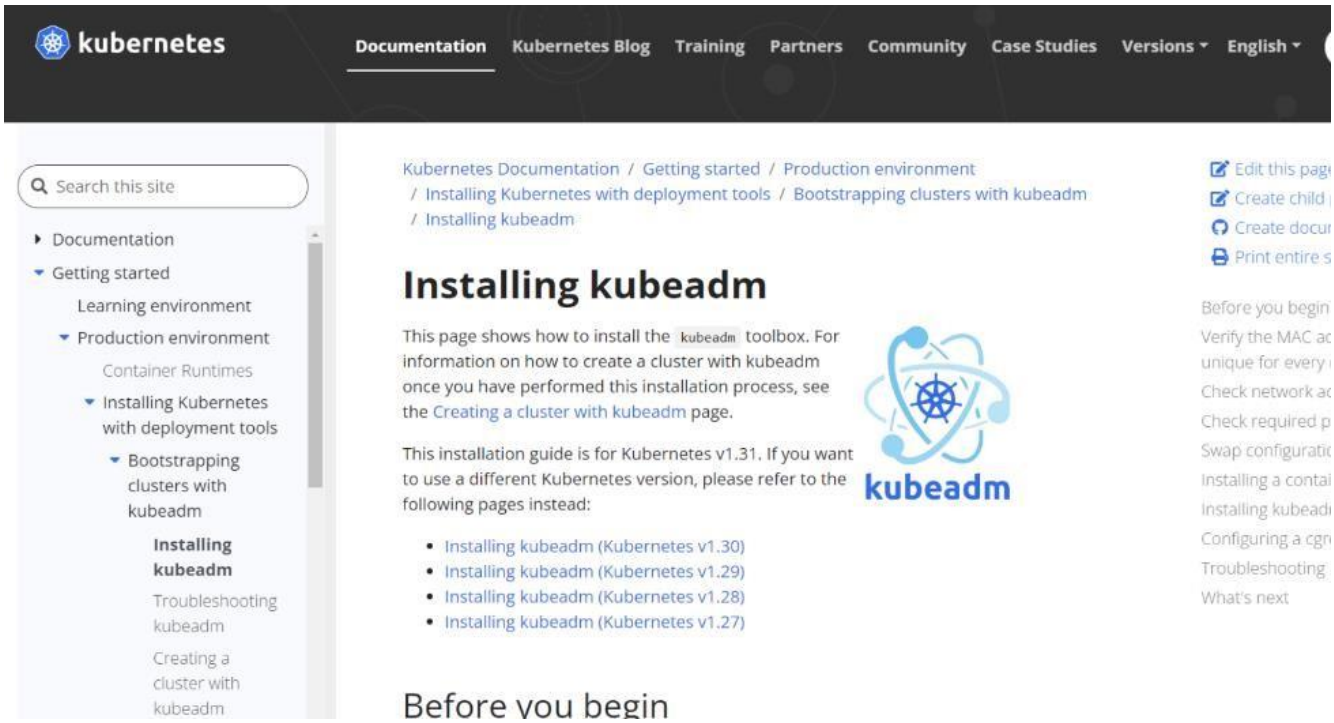
```
ubuntu@ip-172-31-81-188:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-81-188:~$
```

```
ubuntu@ip-172-31-23-53:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-23-53:~$
```

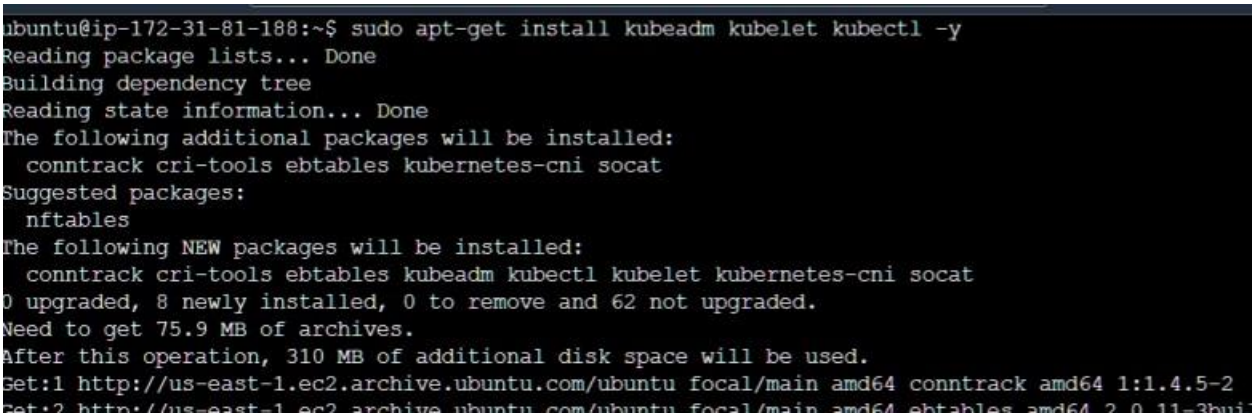
```
ubuntu@ip-172-31-21-143:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-21-143:~$
```

Kubernetes Installation

Go to official documentation off kubedam



1. $sudo apt-get install kubeadm kubelet kubectl -y

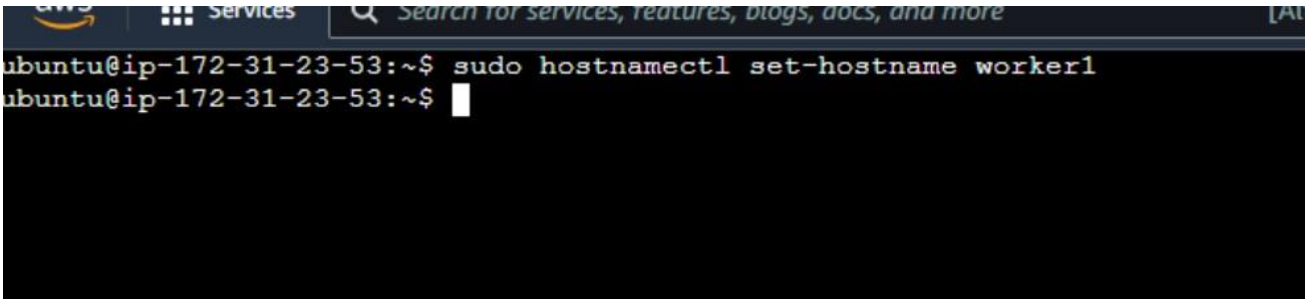


2. Verify the installation with

3. $sudo swapoff –a

```
ubuntu@ip-172-31-22-29:~$ sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

4. $sudo hostnamectl set-hostname




```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
 net.bridge.bridge-nf-call-ip6tables = 1
 net.bridge.bridge-nf-call-iptables = 1
 EOF
 sudo sysctl—system
```

```
ubuntu@ip-172-31-81-188:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the system RAM (967 MB) is less than the minimum 1700 MB
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kub
```

```
lternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

ubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu \
        --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d306f8bfcadac2d827699b0db3d87e13192873a1044f86e2
buntu@ip-172-31-81-188:~$
```

Deploy Pod Network to Cluster A Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

```
buntu@ip-172-31-81-188:~$ sudo kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
The connection to the server localhost:8080 was refused - did you specify the right host or port?
buntu@ip-172-31-81-188:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
buntu@ip-172-31-81-188:~$
```

Join Worker Nodes to the Cluster On the worker nodes, run the command provided by the master node during initialization . It looks something like this: sudo kubeadm join :6443--token --discovery-token-ca-cert-hash sha256:

```
oot@worker1:~# kubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu         --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d30 f8bf
adac2d827699b0db3d87e13192873a1044f86e2 --ignore-preflight-errors=all
preflight] Running pre-flight checks
rror execution phase preflight: couldn't validate the identity of the API Server: Get "https://172.31.81.188:6443/api/v1/namespaces/kube-public/ onfi
maps/cluster-info?timeout=10s": net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers)
o see the stack trace of this error execute with --v=5 or higher
oot@worker1:~# kubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu         --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d30 f8bf
adac2d827699b0db3d87e13192873a1044f86e2 --ignore-preflight-errors=all
preflight] Running pre-flight checks
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-start] Waiting for the kubelet to perform the TLS Bootstrap...

his 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.

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

Verify the Cluster

```
Using cluster from kubectl context: workshop.k8s.local

Validating cluster workshop.k8s.local

INSTANCE GROUPS
NAME                    ROLE    MACHINETYPE     MIN     MAX     SUBNETS
master-us-west-2a       Master  t3.medium       1       1       us-west-2a
nodes-us-west-2a        Node    t3.medium       1       1       us-west-2a

NODE STATUS
NAME                                            ROLE    READY
ip-172-20-40-55.us-west-2.compute.internal      master  True
ip-172-20-58-174.us-west-2.compute.internal     node    True

Your cluster workshop.k8s.local is ready
```

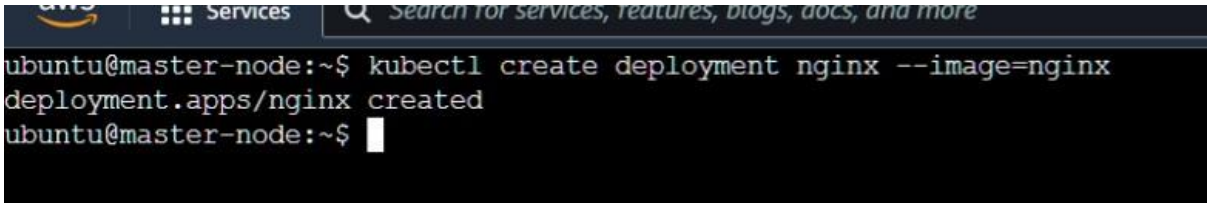
ADVANCE DEVOPS EXP 4

Name :- Swaraj Patil
Roll no :- 40

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

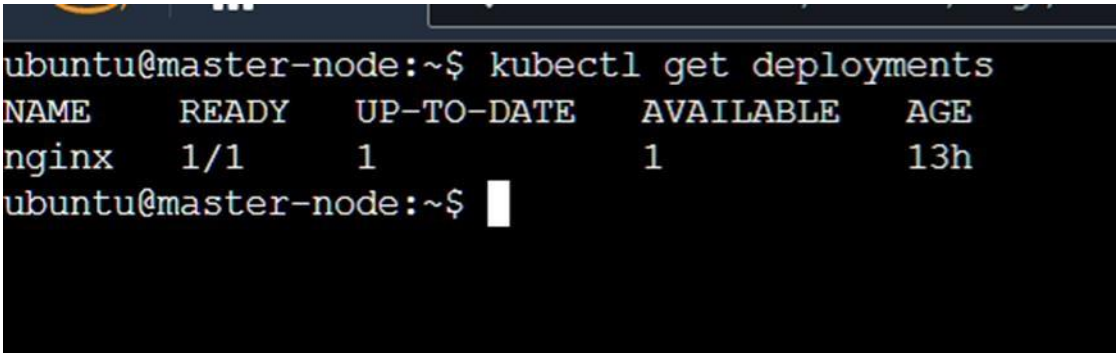
Step 1: As the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment.

$kubectl create deployment nginx --image=nginx



Step 2: Verify the deployment using the command:

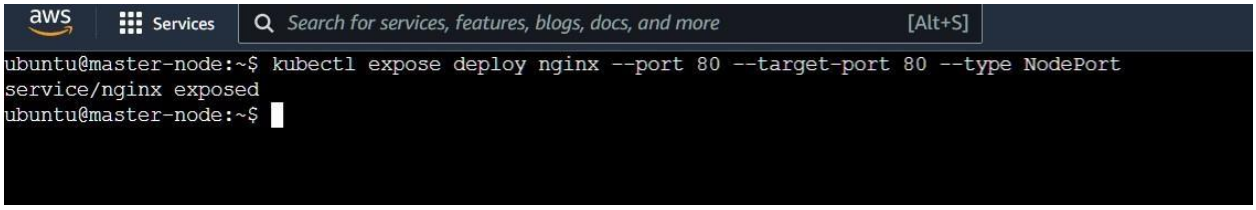
$kubectl get deployments



Step 3: Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster

with this service-type, Kubernetes will assign this service on ports on the **30000+** range.

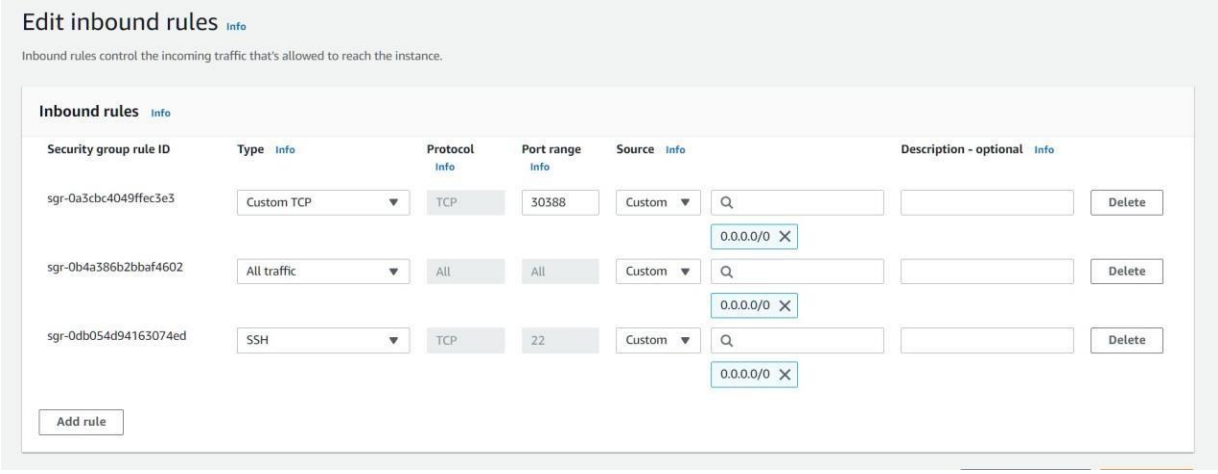
$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort



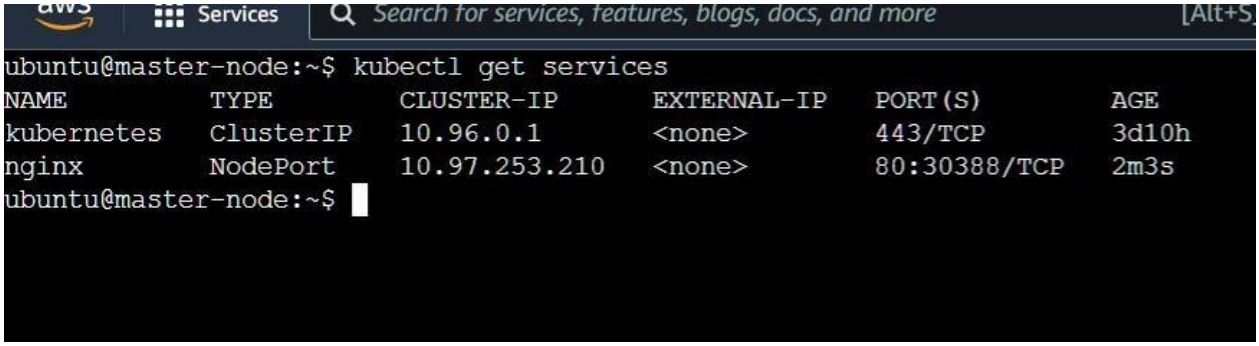
Step 4: Run this command to see a summary of the service and the ports exposed.

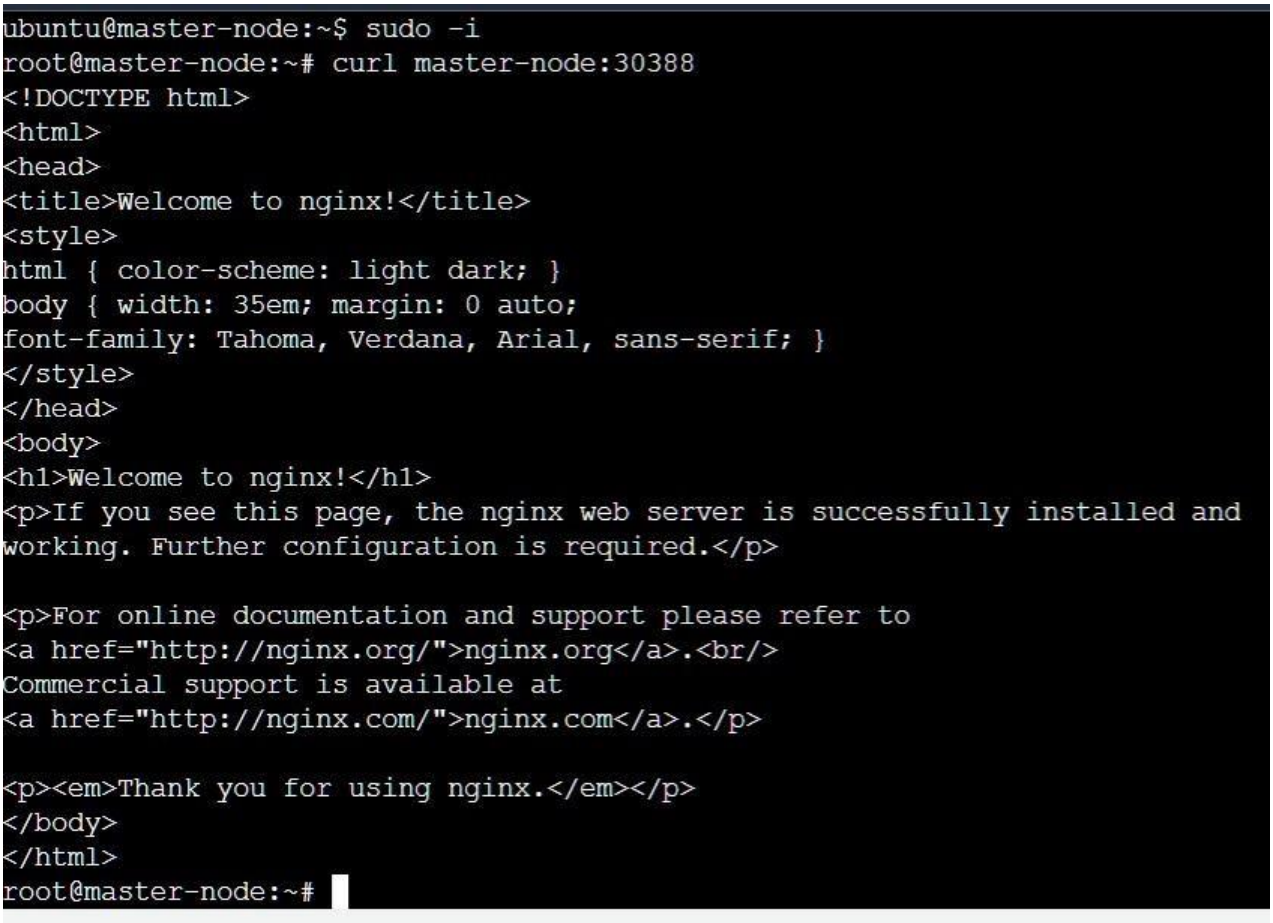
$kubectl get services

Step 5: Add the port which is displayed i.e. 30388 (in our case) in the inbound rules of the security group.



Step 6: Now you can verify that the Nginx page is reachable on all nodes using thecurl command.

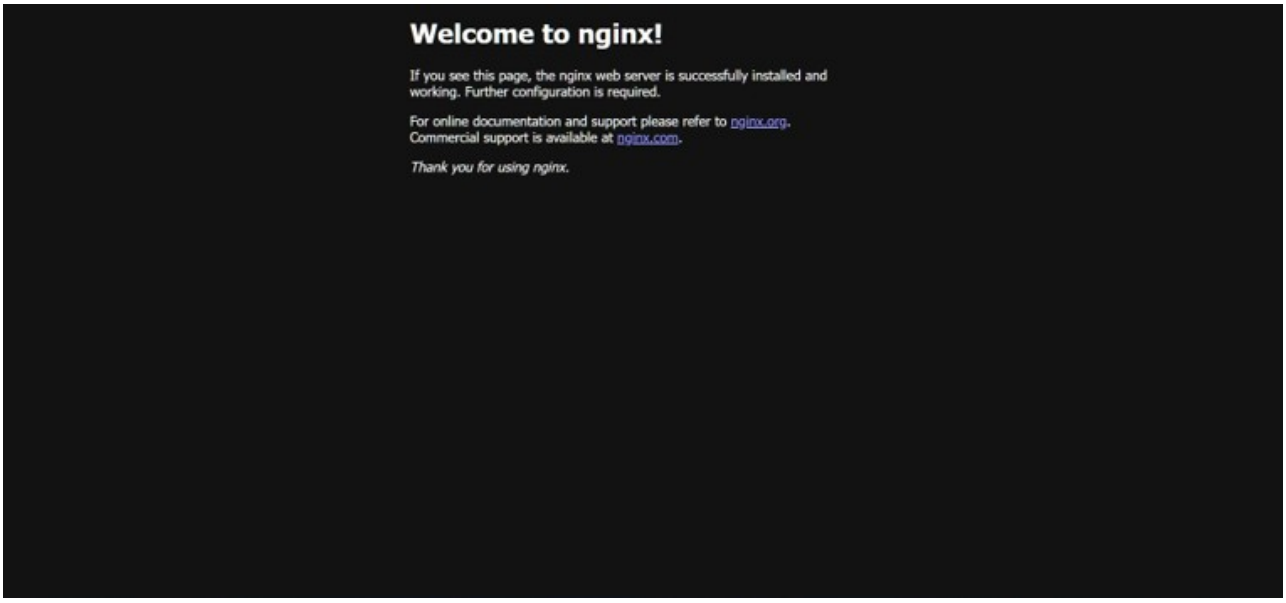




As you can see, the "**WELCOME TO NGINX!**" page can be reached.

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

http://52.90.129.234:30388



ADVANCE DEVOPS EXP-7

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

Step 1: Installing SonarQube from the Docker Image

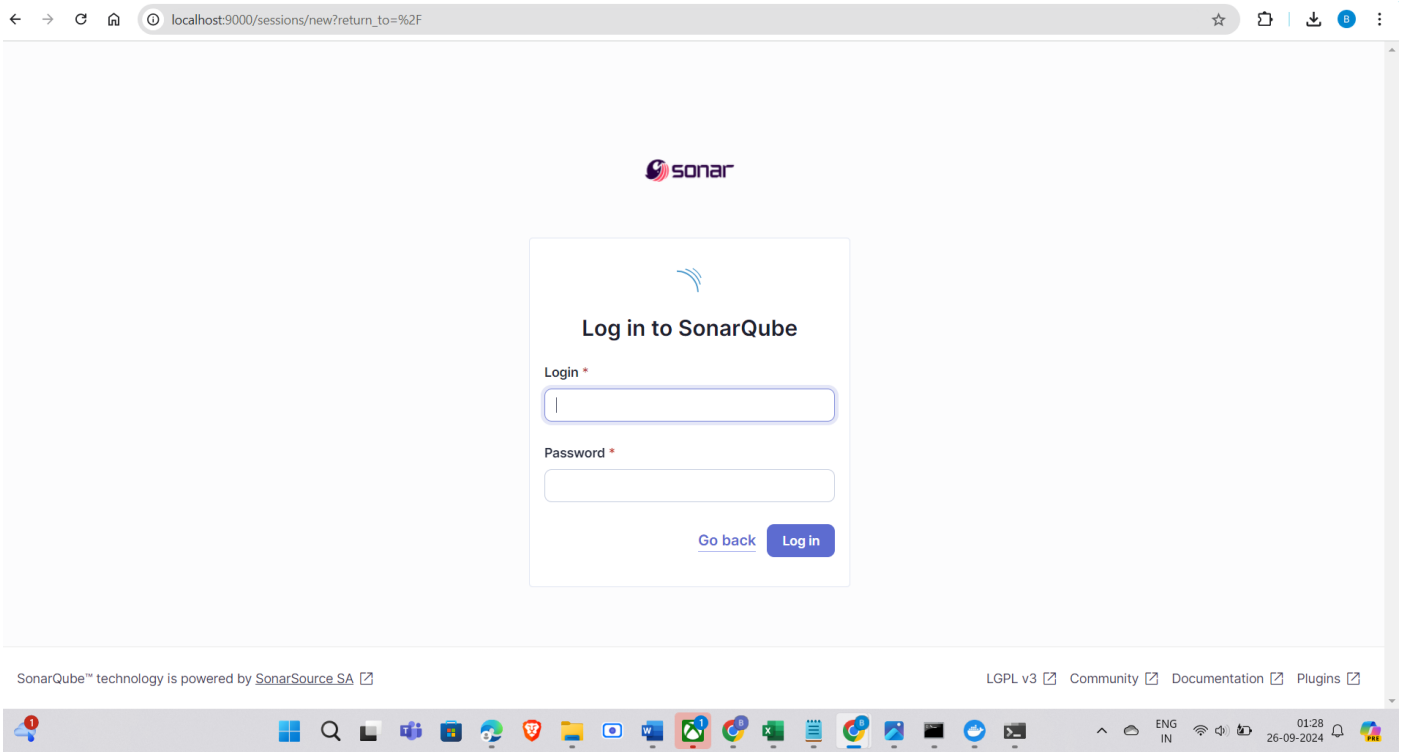
$ docker run -d --name sonarqube -e

SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000

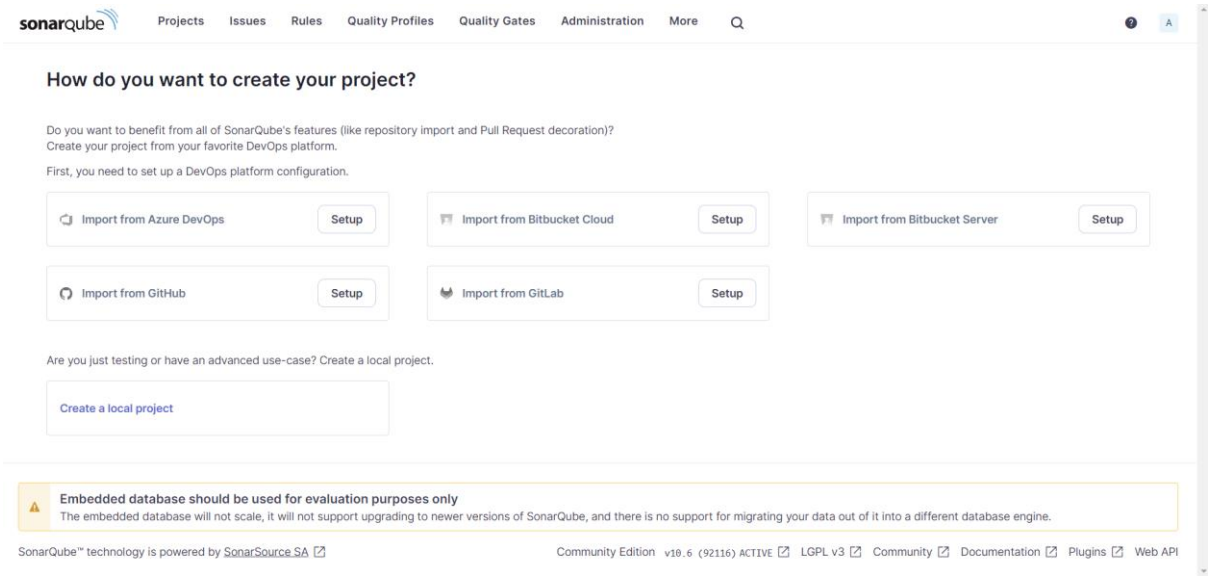
sonarqube:latest

```
PS C:\Windows\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
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
22ee358a0f88adabbcd823e6ececbb28975b9164b116b619e567666c8d3860d1
PS C:\Windows\system32>
```

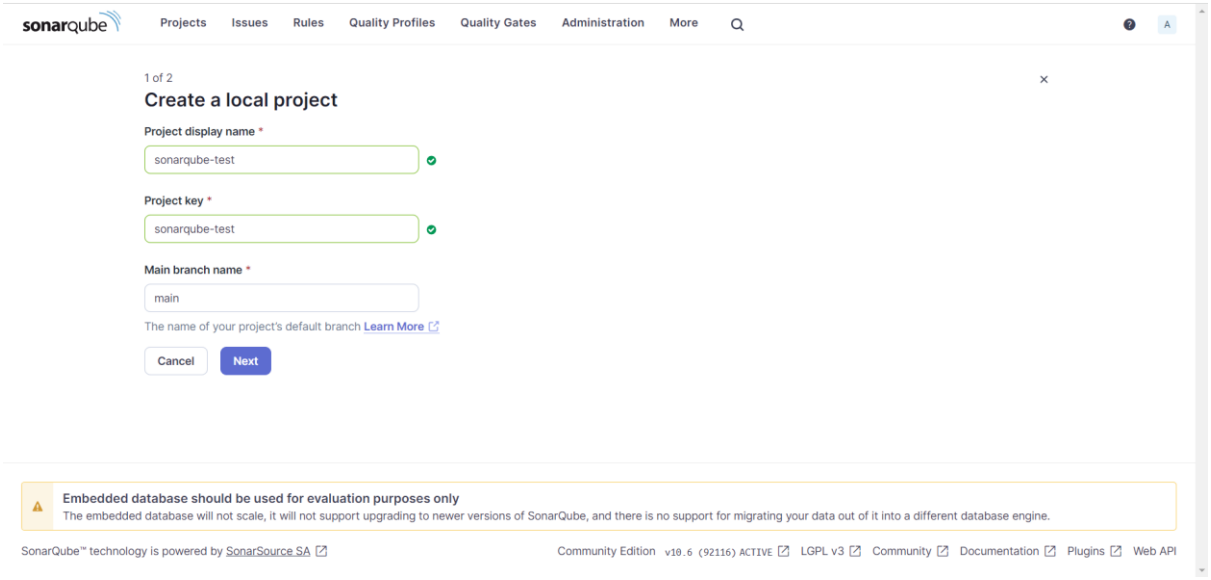
Step 2: After installation of SonarQube, go to the SonarQube page by typing: http://localhost:9000/ on your browser. If you see such page then you have successfully installed it.



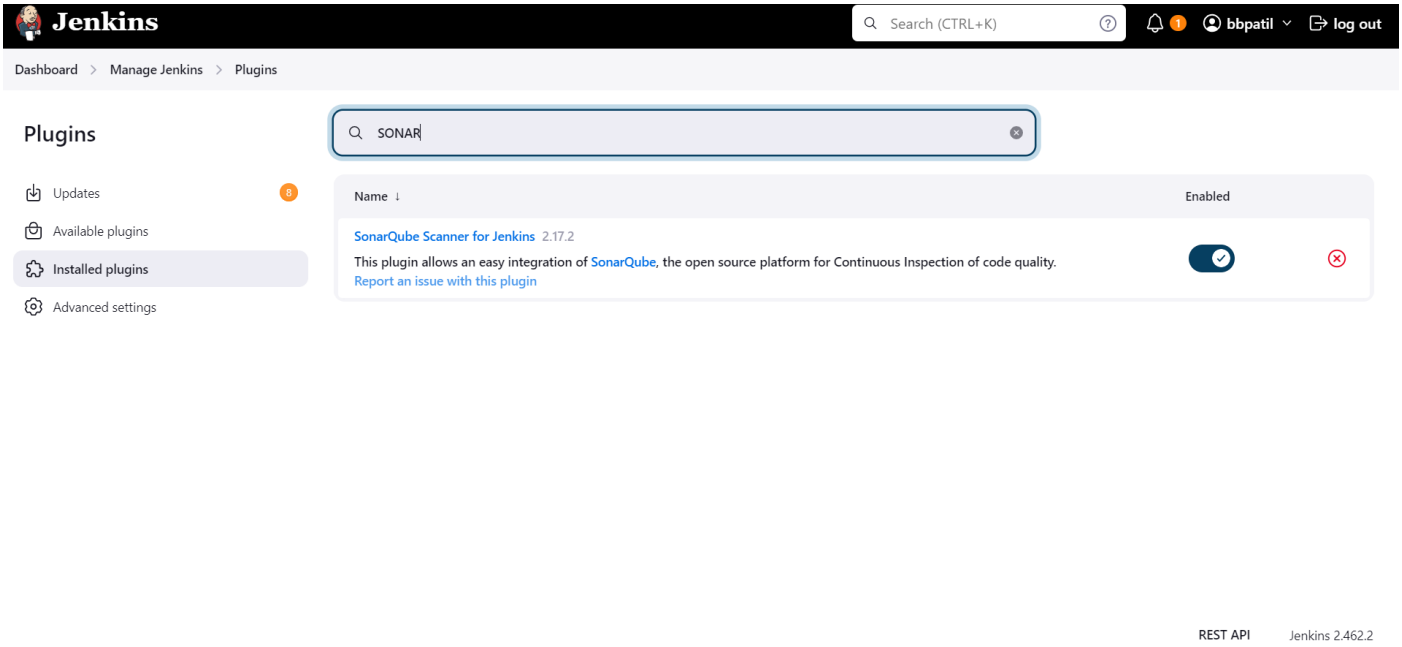
Step 3: Login using the username as “admin” and password as “admin”. And then you will see the home page of SonarQube



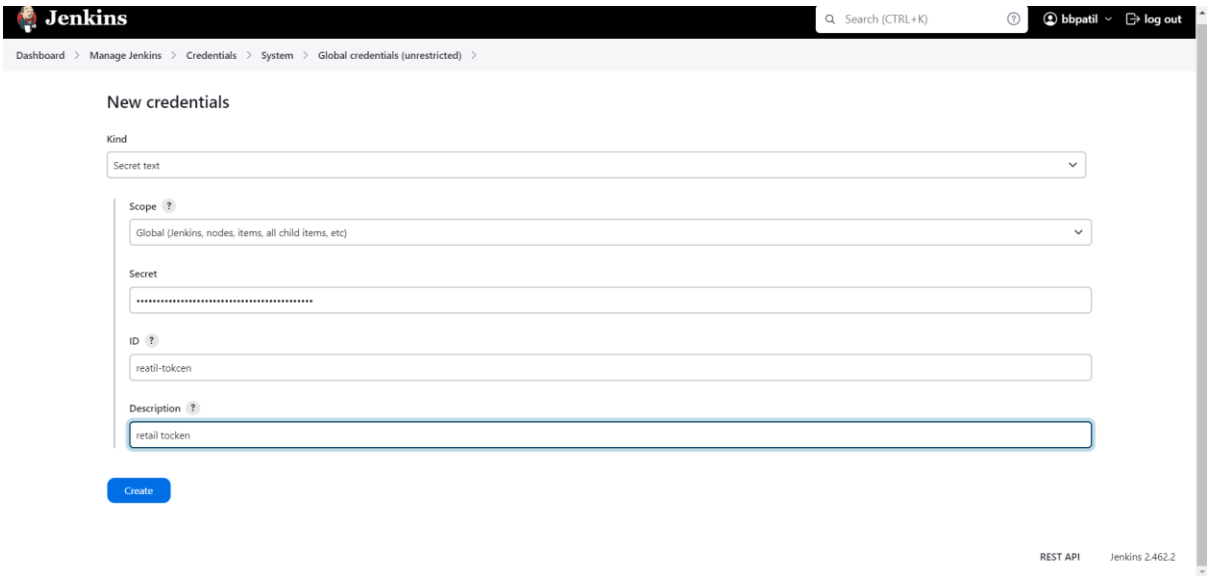
Step 4 :- give the name project



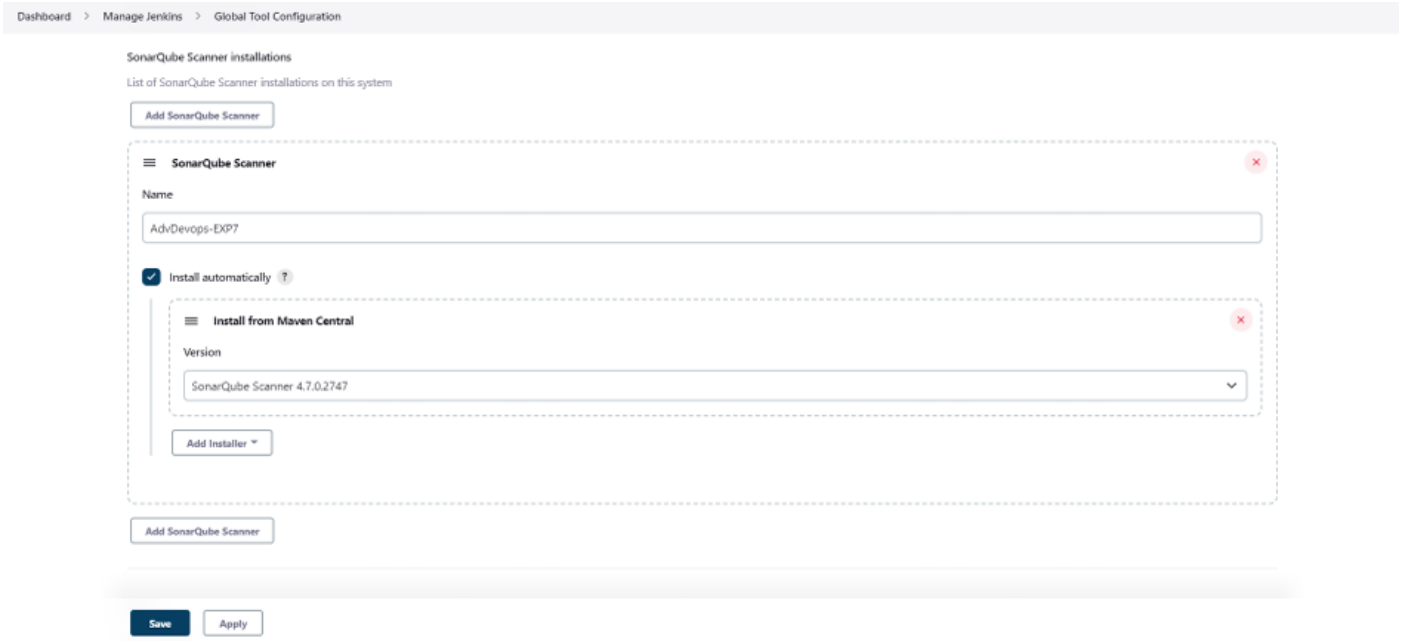
Step 5: Go to Dashboard > Manage Jenkins > Plugin Manager and search for SonarQube Scanner under Available plugins for Jenkins and install it



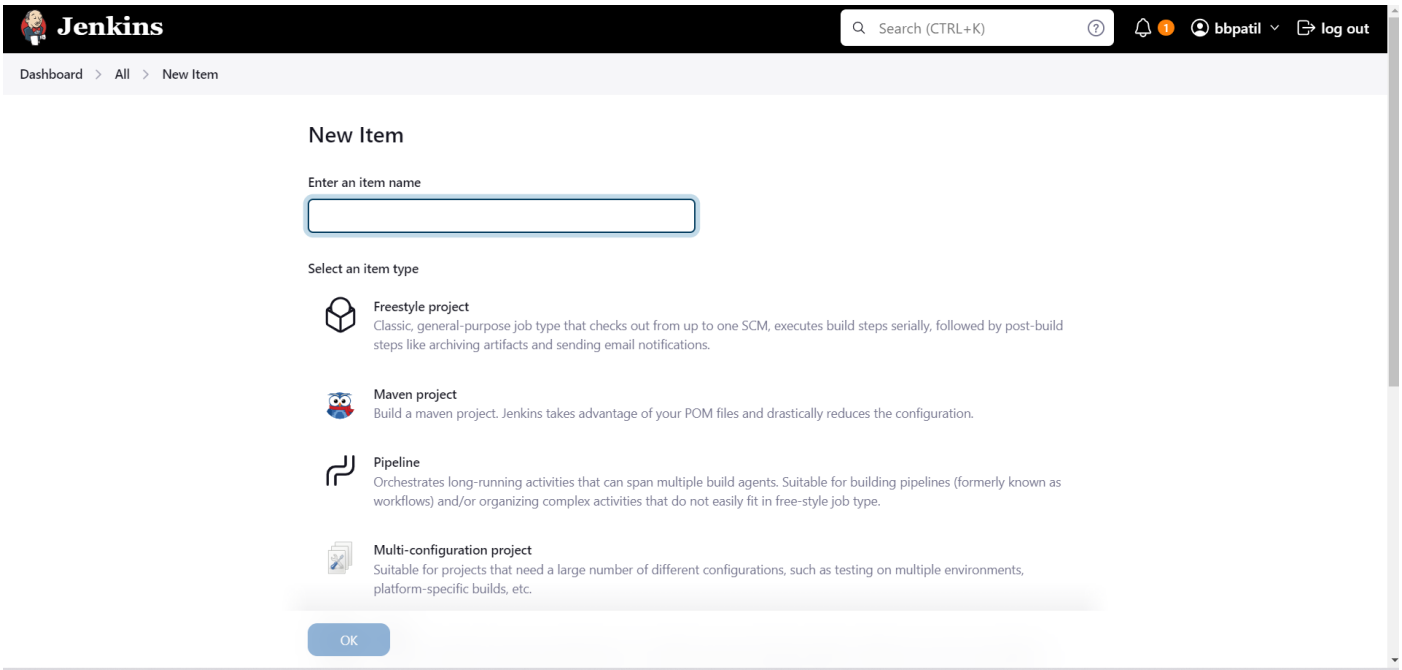
Step 6:- add authentication all details



Step 7: Search SonarQube Scanner under Dashboard > Manage Jenkins > Global Tool Configuration. Choose the latest configuration and choose Install Automatically



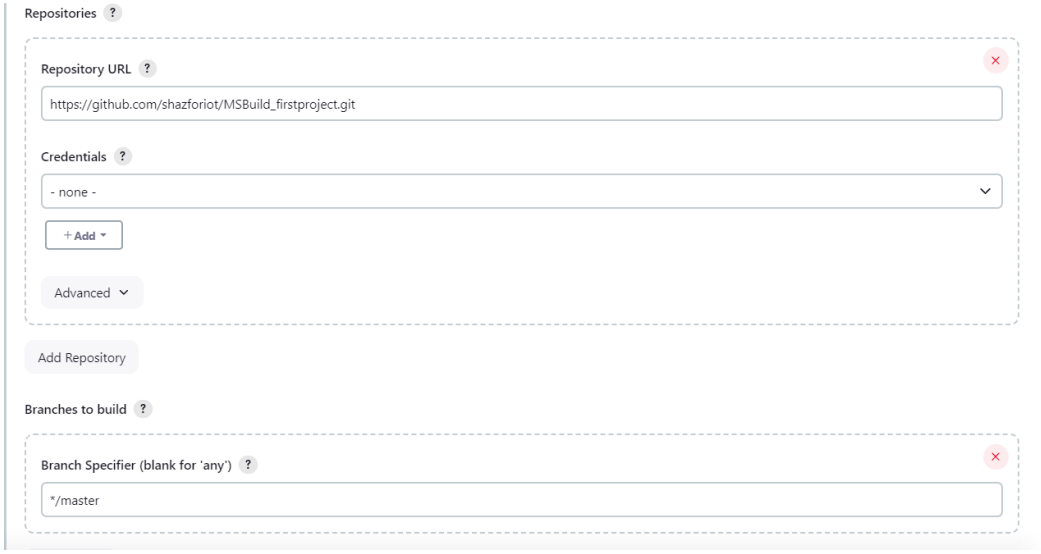
Step 8: After the configuration, create a New Item in Jenkins, choose a freestyle project.

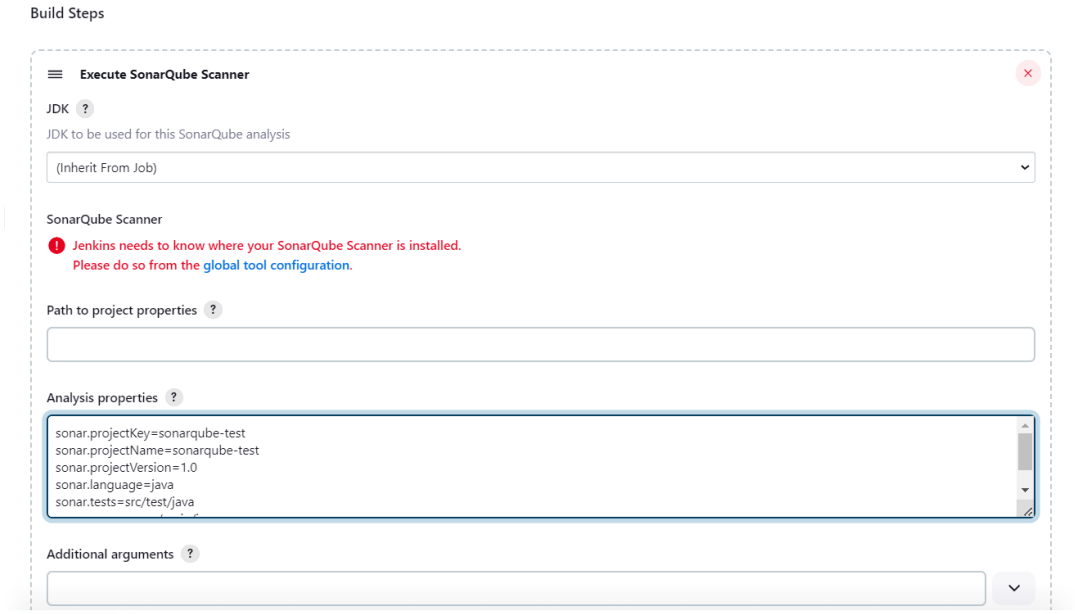


Step 9 :- add the git url and also all details

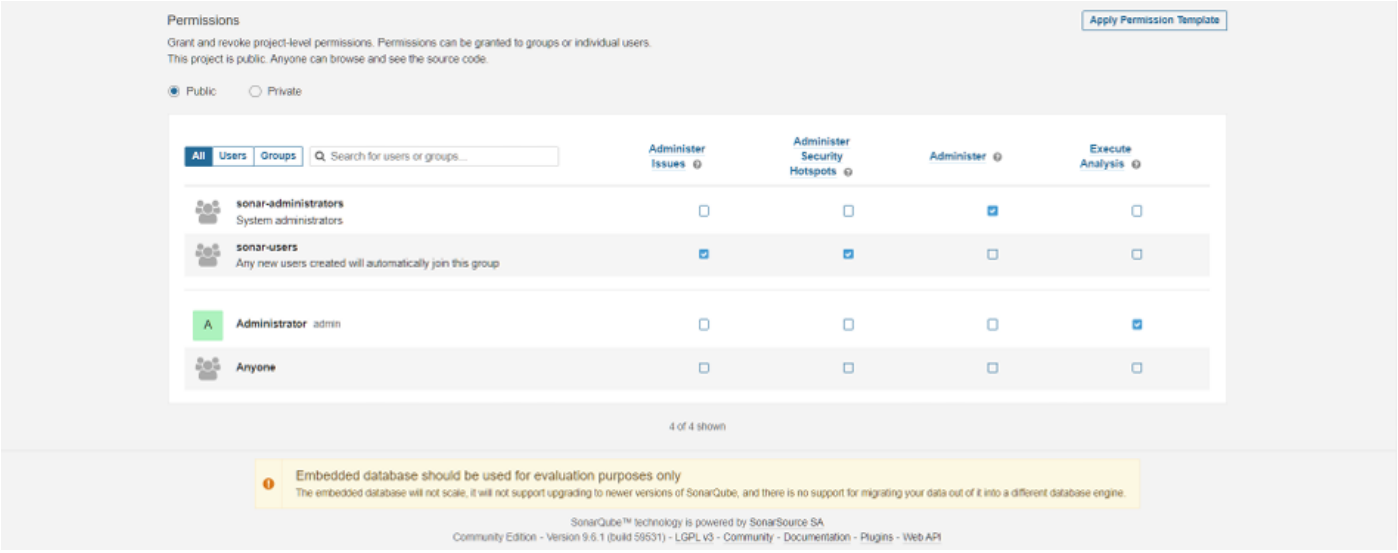
https://github.com/shazforiot/MSBuild_firstproject.git

step 10 :- under the bulid command all the details plz correctly and give Mention the SonarQube Project Key, Login, Password, and Host URL. sonar.projectKey=retail-stock
sonar.login=*your username*
sonar.password=*your password*
sonar.hosturl=http://localhost:9000

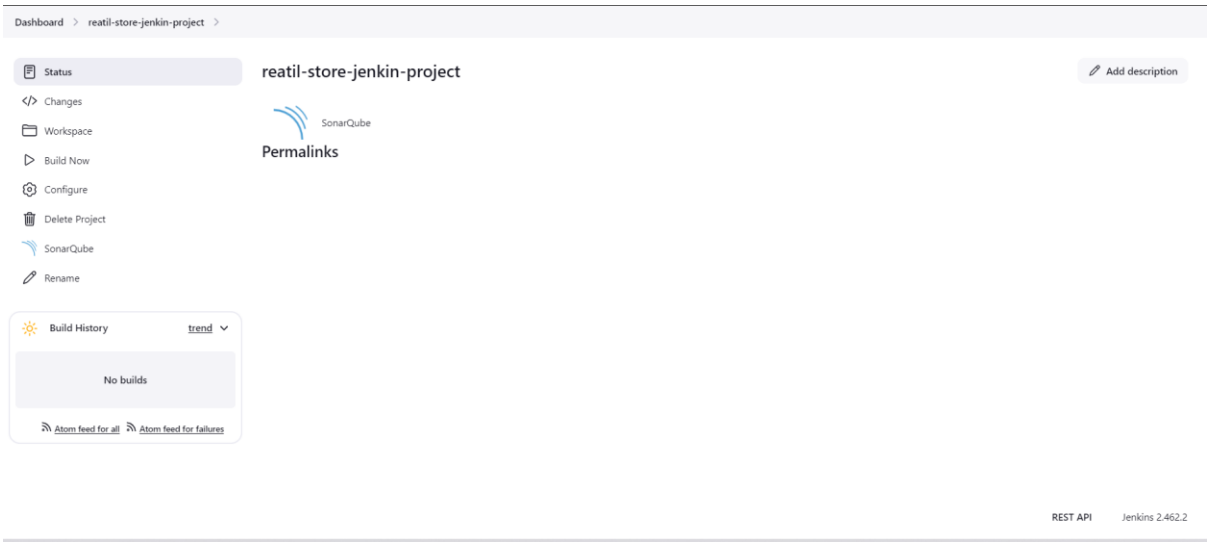




Step 11: Go to http://localhost:9000/ and enter your previously created username. Go to Permissions and grant the Admin user Execute Permissions.



Step 12: Run The Build

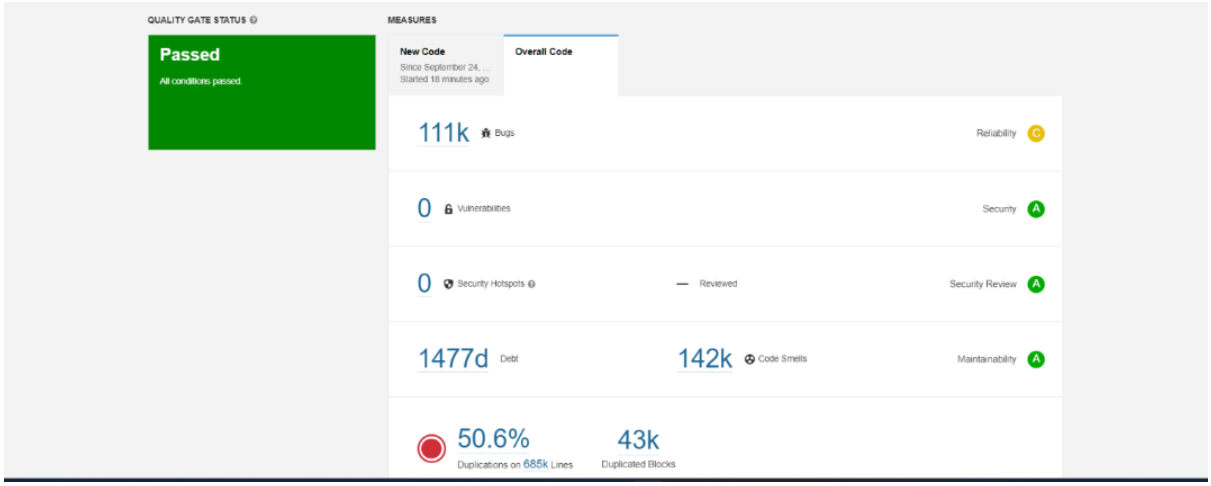


console output

```
Started by user bbpatil
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\plzhoha
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\plzhoha # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.46.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=1
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
[plzhoha] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 ******** "-Dsonar.projectKey=pipelone ok1" -Dsonar.host.url=http://localhost:9000 -Dsonar.source=HelloWorldCar
-Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\plzhoha
00:39:45.044 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
00:39:45.059 INFO  Scanner configuration file:
C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-scanner.properties
```

```
Commit message: "updated"
First time build. Skipping changelog.
[plzhoha] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 ******** "-Dsonar.projectKey=pipelone ok1" -Dsonar.host.url=http://localhost:9000 -Dsonar.source=HelloWorldCare
-Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\plzhoha
00:39:45.044 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
00:39:45.059 INFO  Scanner configuration file:
C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-scanner.properties
00:39:45.060 INFO  Project root configuration file: NONE
00:39:45.076 INFO  SonarScanner CLI 6.1.0.4477
00:39:45.077 INFO  Java 21.0.4 Eclipse Adoptium (64-bit)
00:39:45.082 INFO  Windows 11 10.0 amd64
00:39:45.109 INFO  User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
00:39:45.909 INFO  JRE provisioning: os[windows], arch[amd64]
00:39:48.935 INFO  Communicating with SonarQube Server 10.6.0.92116
00:39:49.380 INFO  Starting SonarScanner Engine...
00:39:49.381 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
00:39:50.215 INFO  Load global settings
00:39:50.408 INFO  Load global settings (done) | time=191ms
00:39:50.417 INFO  Server id: 147B411E-AZIp-aMmENycOd53ifbB
00:39:50.430 INFO  Loading required plugins
00:39:50.430 INFO  Load plugins index
00:39:50.554 INFO  Load plugins index (done) | time=123ms
00:39:50.554 INFO  Load/download plugins
00:39:51.123 INFO  Load/download plugins (done) | time=570ms
00:39:51.447 INFO  Process project properties
00:39:51.449 INFO  Process project properties (done) | time=2ms
00:39:51.475 ERROR Validation of project failed:
  o "pipelone ok1" is not a valid project key. Allowed characters are alphanumeric, '-', '_', '.' and ':', with at least one non-digit.
00:39:51.547 INFO  EXECUTION FAILURE
00:39:51.548 INFO  Total time: 6.491s
```

Due to blocking system git this screenshoot form clg pc



Name:- Swaraj Patil

Roll :- 40

Experiment no 8

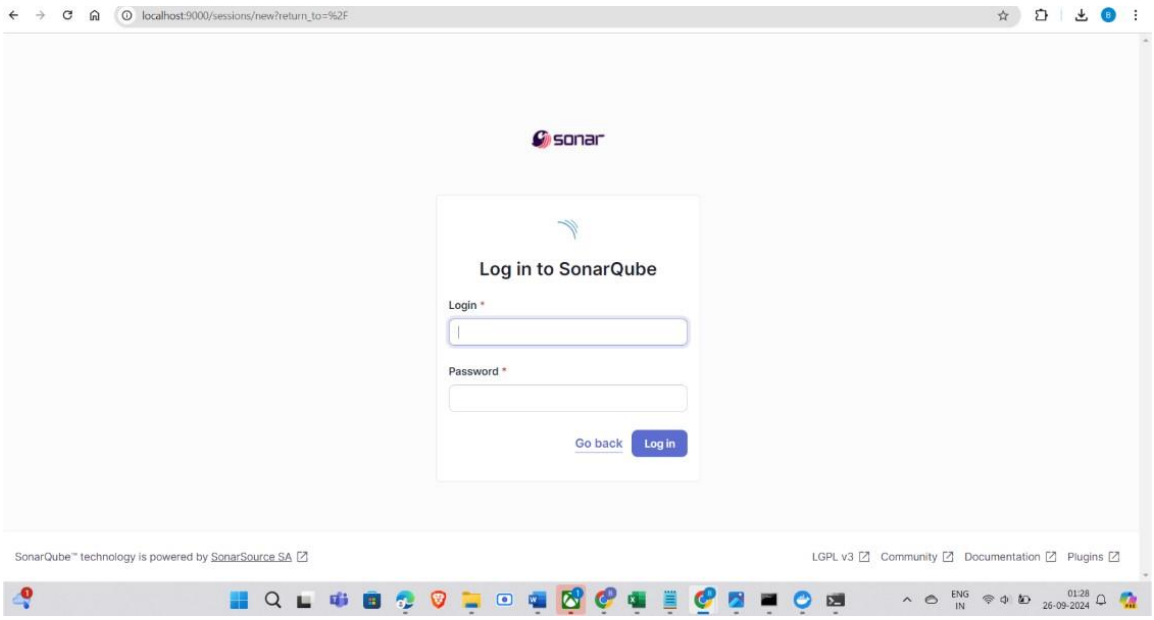
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner

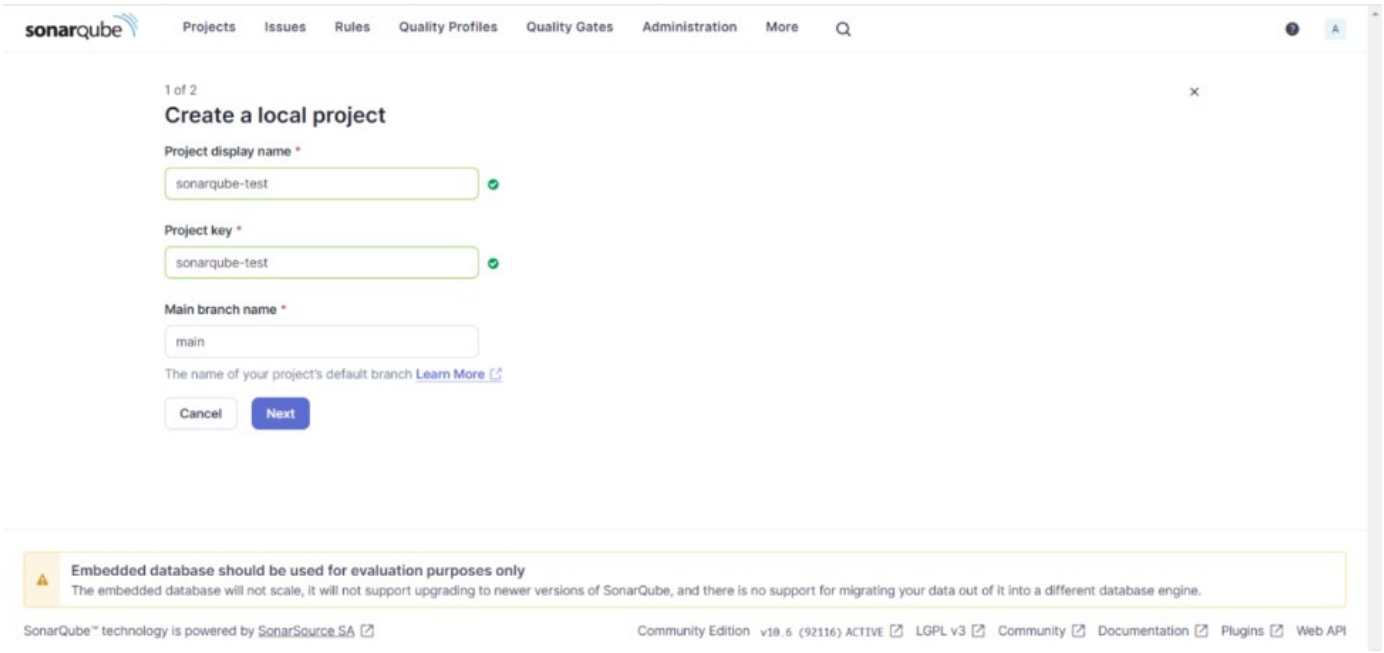
1. Command: docker pull sonarqube

```
C:\Windows\System32>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

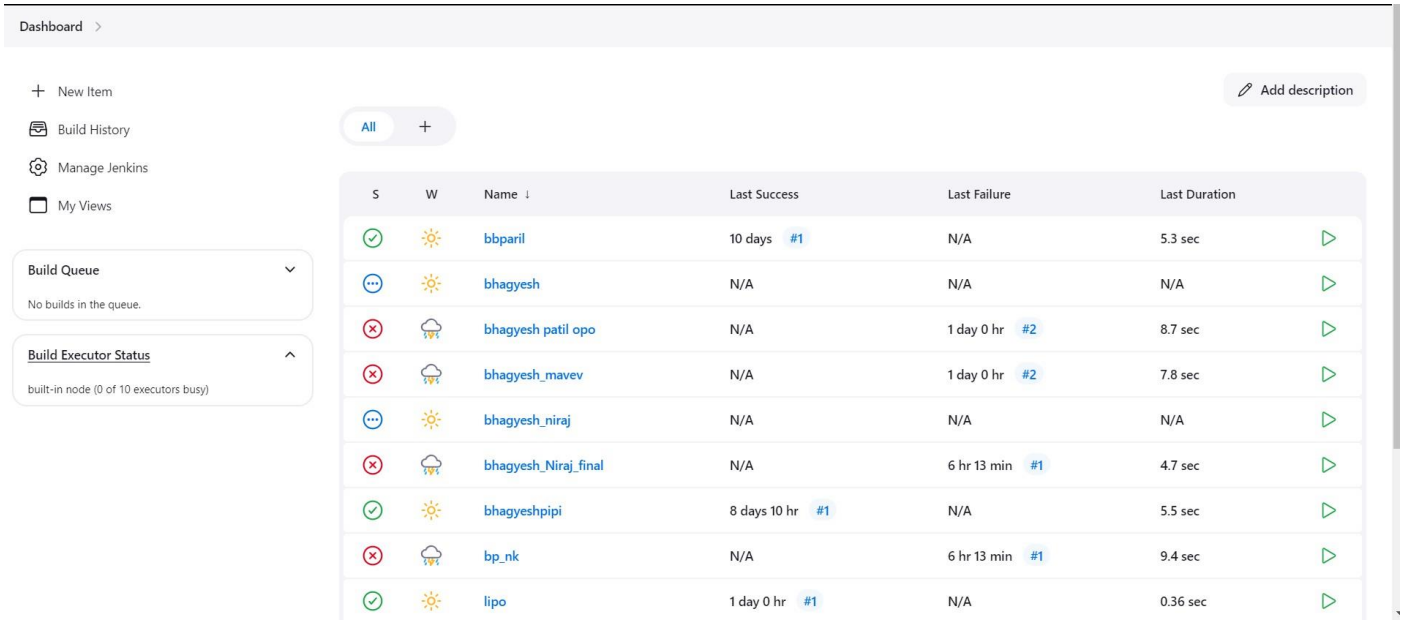
2. Login the details



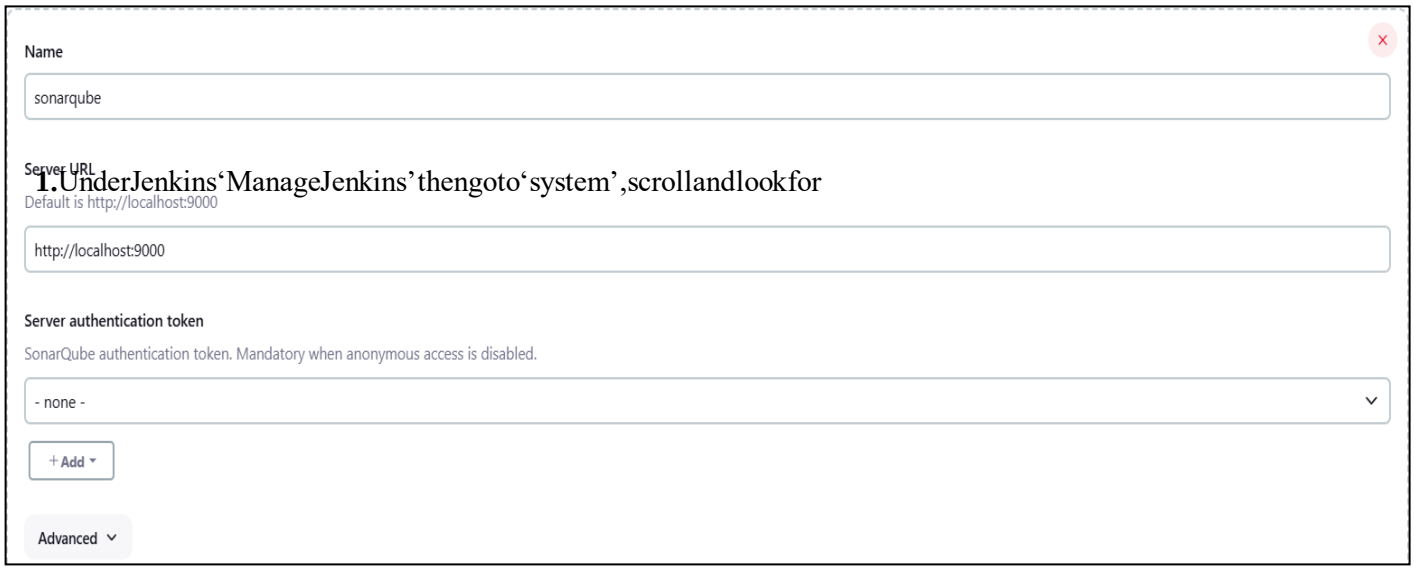
3. Give the name of the project



4. Open up Jenkins Dashboard on localhost, port 8081 or which ever portitisat for you.



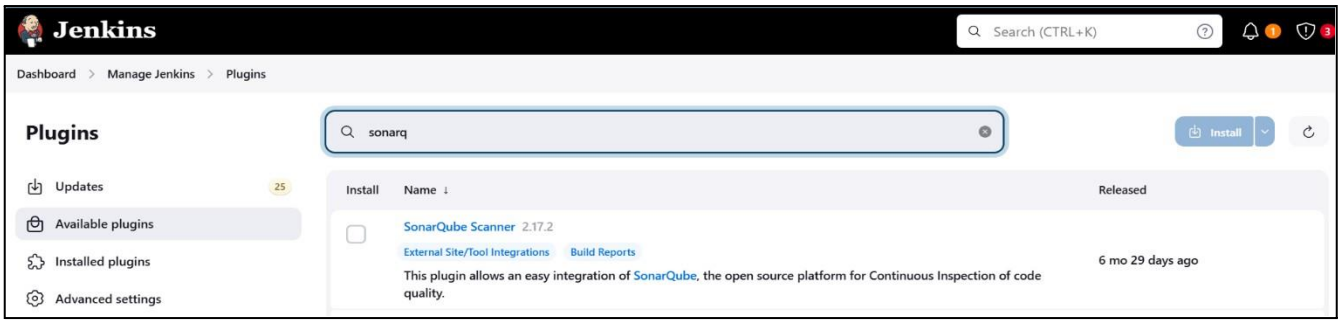
5. to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



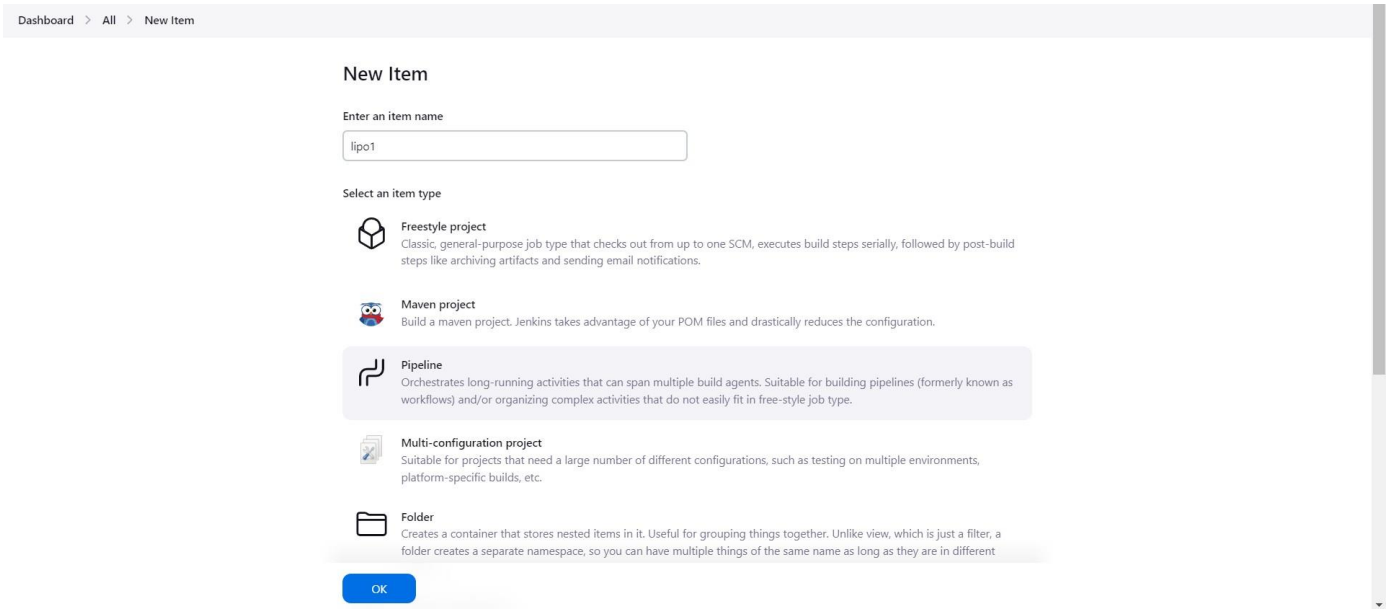
1. UnderJenkins‘ManageJenkins’thengoto‘system’,scrollandlookfor

2. SonarQubeServers
and enter the details.
Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **adv_devops_7_sonarqube**
In **Server URL** Default is **http://localhost:9000**



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



This is java code put in the pipeline script

Under Pipeline script, enter the following:

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

  stage('SonarQube analysis') {
    withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenki
    ns>') {
      sh """
        <PATH_TO_SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
        -D sonar.login=<SonarQube_USERNAME> \
        -D sonar.password=<SonarQube_PASSWORD> \
        -D sonar.projectKey=<Project_KEY> \
```

```
        -D sonar.exclusions=vendor/**,resources/**,**/*.java \
        -D sonar.host.url=<SonarQube_URL>(default:
      http://localhost:9000/)"""
    }
  }
```

Definition

Pipeline script

Script ?

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

stage('SonarQube analysis') { withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
sh """
<PATH_TO_SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
-D sonar.login=admin \
-D sonar.password=admin> \
-D sonar.projectKey=sonarqube \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=http://localhost:9000
"""
}
}
}
```

Use Groovy Sandbox ?

Pipeline Syntax

7. Build project

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times:	6s	168ms
#2 Sep 25 23:34 No Changes	7s	166ms failed
#1 Sep 25 23:34 No Changes	6s	170ms failed

Permalinks

- Last build (#2), 1 day 1 hr ago
- Last failed build (#2), 1 day 1 hr ago
- Last unsuccessful build (#2), 1 day 1 hr ago
- Last completed build (#2), 1 day 1 hr ago

This error come in pc but next step output come in clg pc

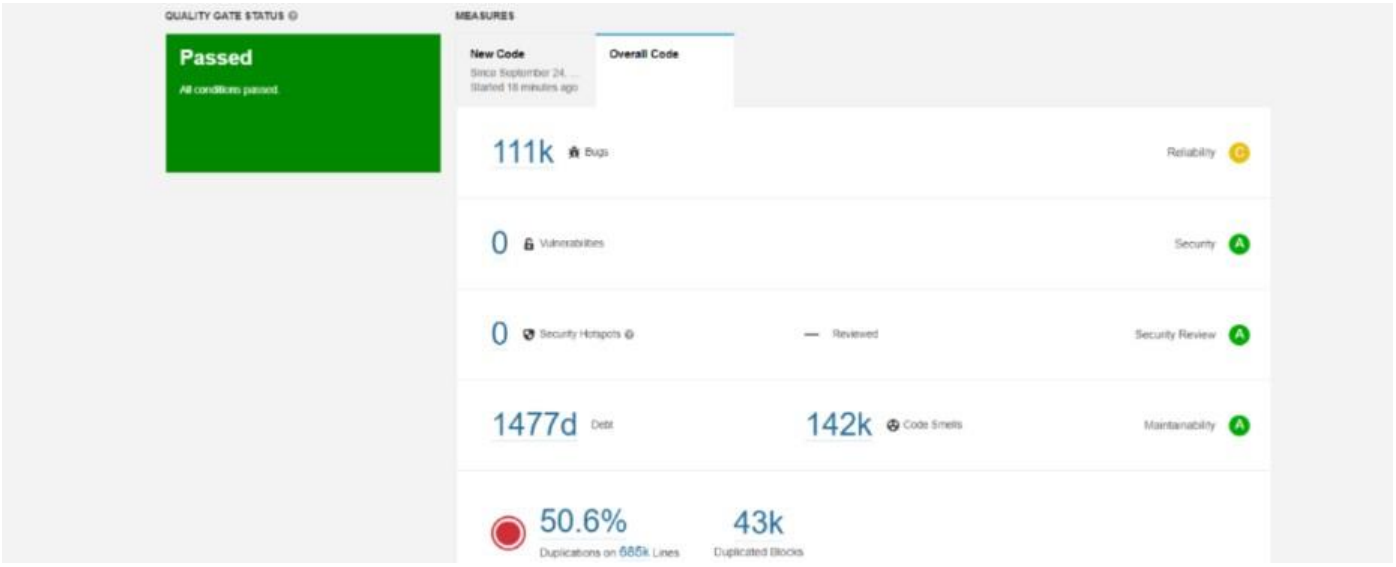
Console Output

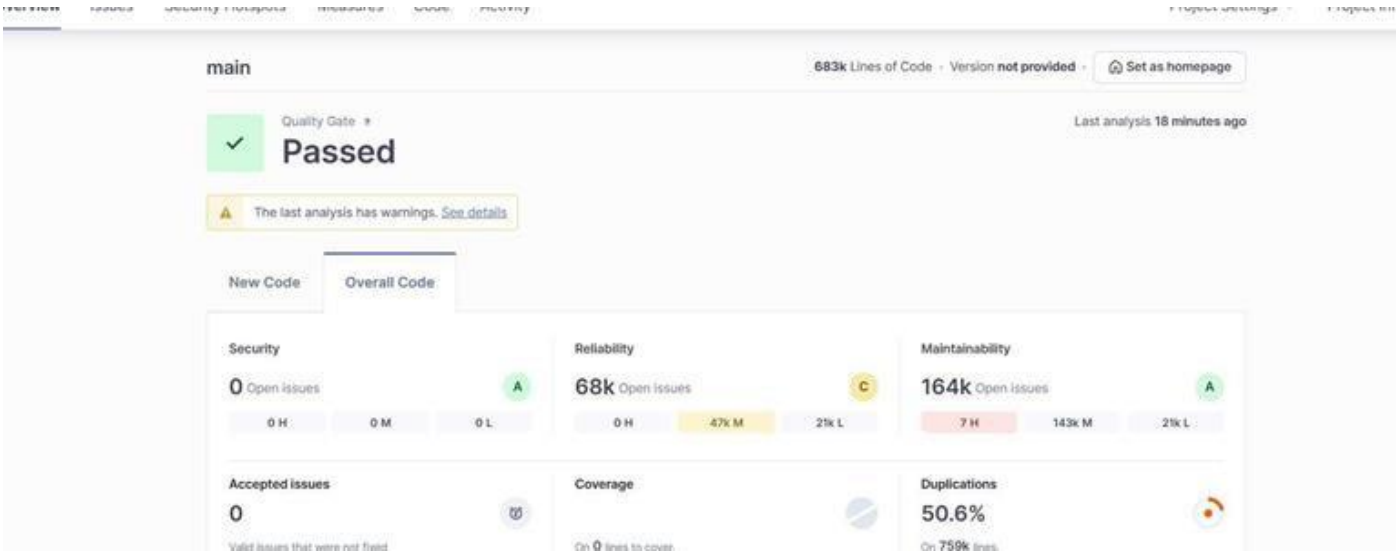
Download Copy View as plain text

```
Started by user bbpatil
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\lipo12
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/GOL.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\lipo12 # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.46.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe checkout -b master ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
Commit message: "Update Jenkinsfile"
```

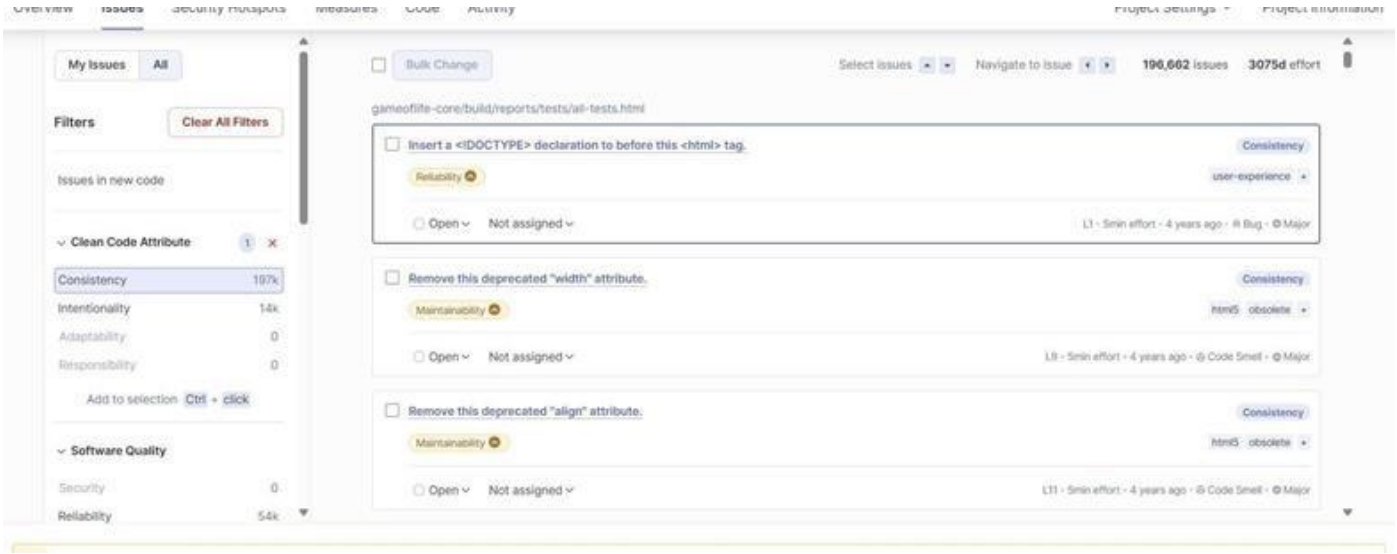
8. Check Console

9. Now, check the project in SonarQube

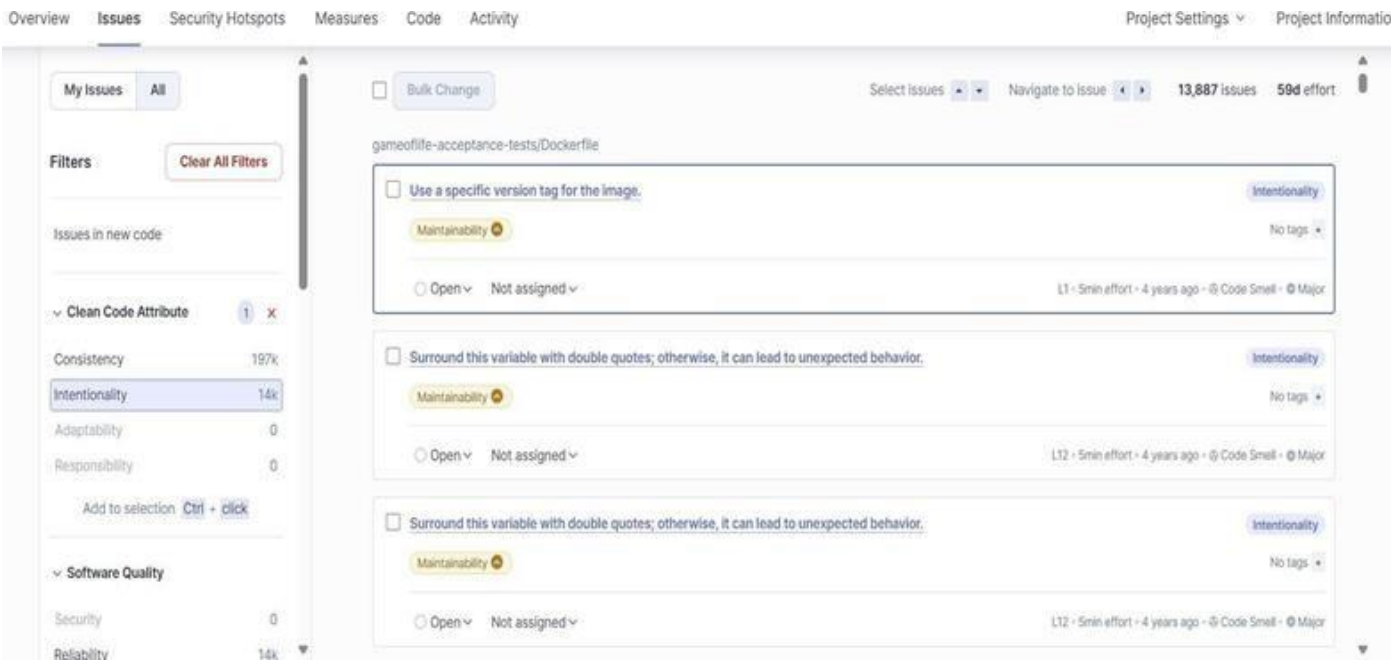




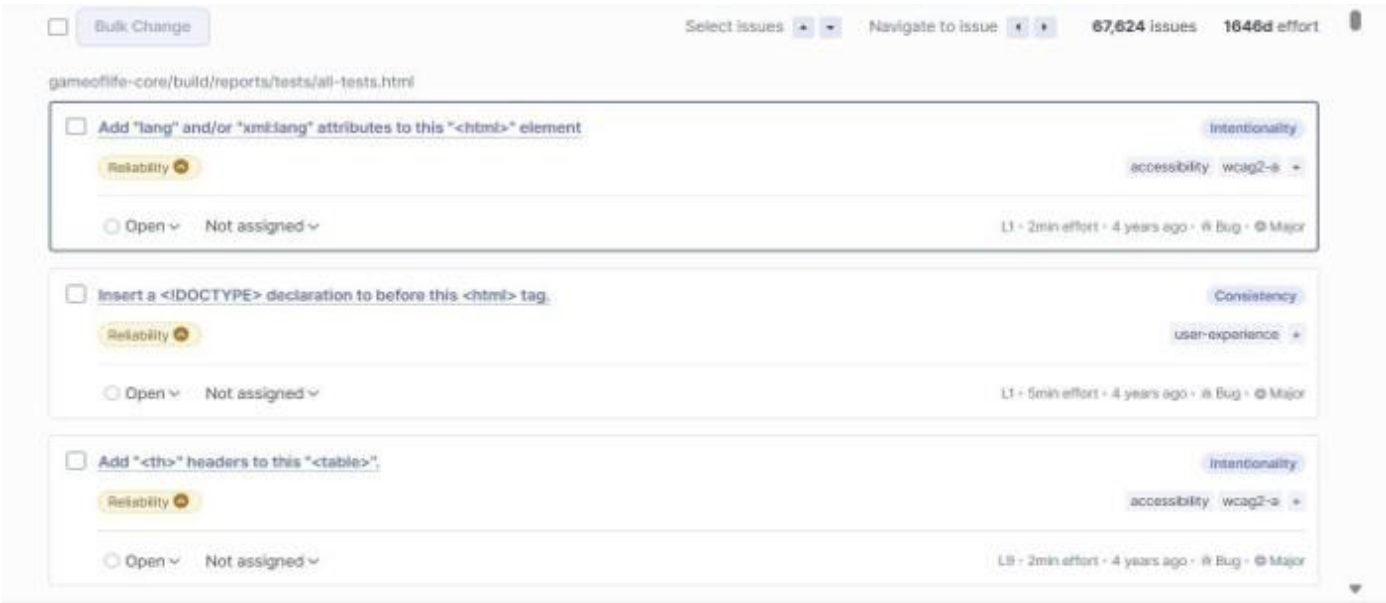
10 .Consistency


Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
My Issues
All
Bulk Change
Select issues
Navigate to issue
196,662 issues
3075d effort
Filters
Clear All Filters
gameoflife-core/build/reports/tests/all-tests.html
Issues in new code
Insert a <!DOCTYPE> declaration to before this <html> tag.
Consistency
Reliability
user-experience
Open
Not assigned
L1 · 5min effort · 4 years ago · Bug · Major
Clean Code Attribute
Consistency 197k
Intentionality 14k
Adaptability 0
Responsibility 0
Remove this deprecated "width" attribute.
Consistency
Maintainability
html5 obsolete
Open
Not assigned
L9 · 5min effort · 4 years ago · Code Smell · Major
Add to selection Ctrl + click
Remove this deprecated "align" attribute.
Consistency
Maintainability
html5 obsolete
Software Quality
Security 0
Reliability 54k
Open
Not assigned
L11 · 5min effort · 4 years ago · Code Smell · Major

11. Intentionality

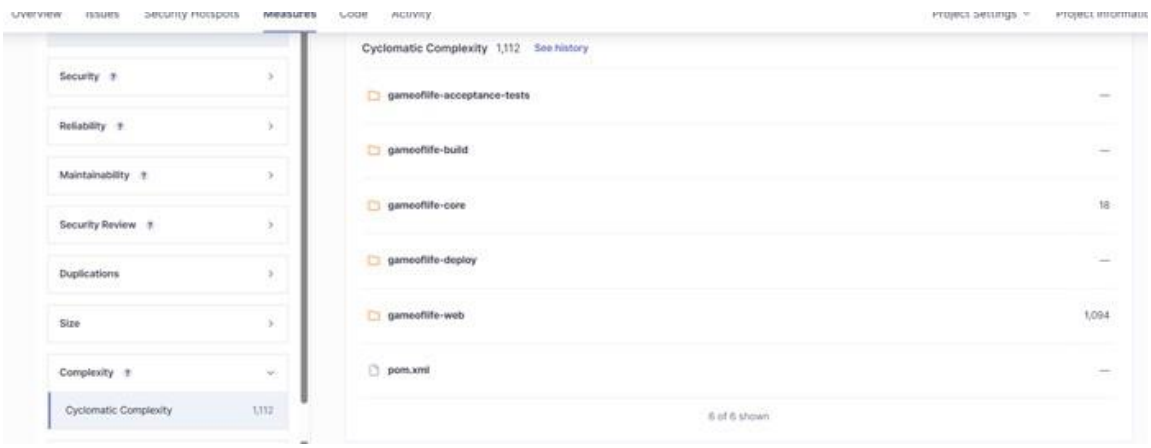


12. Bugs



13. Code smell

14. Cyclomatic Complexities



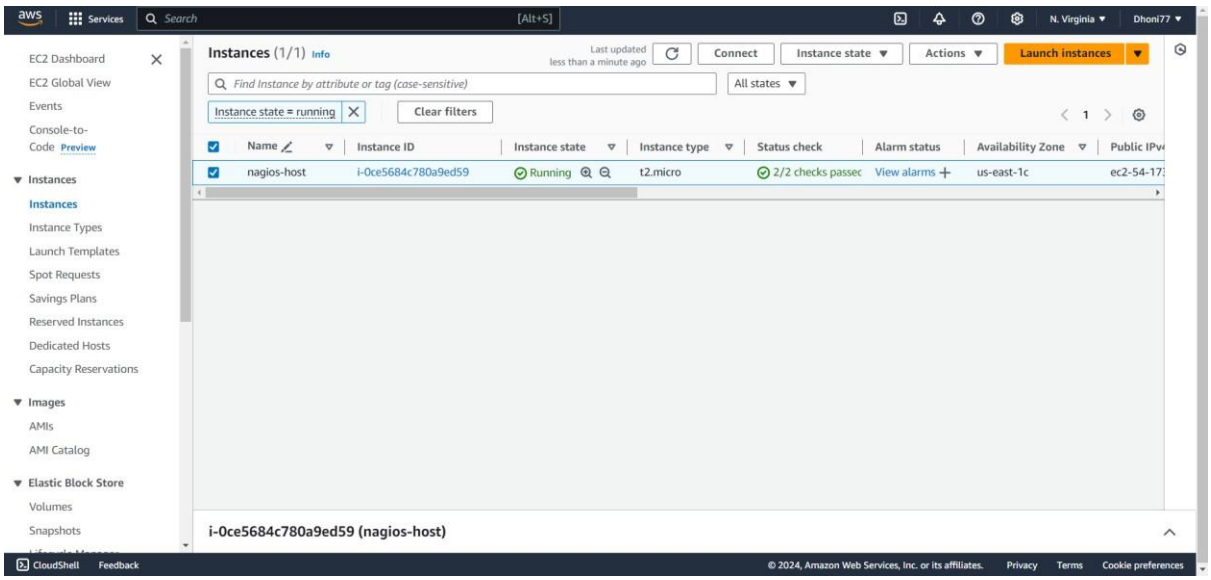
ADVANCE DEVOPS EXP 9

Name :- Swaraj Patil

Roll_no :- 40

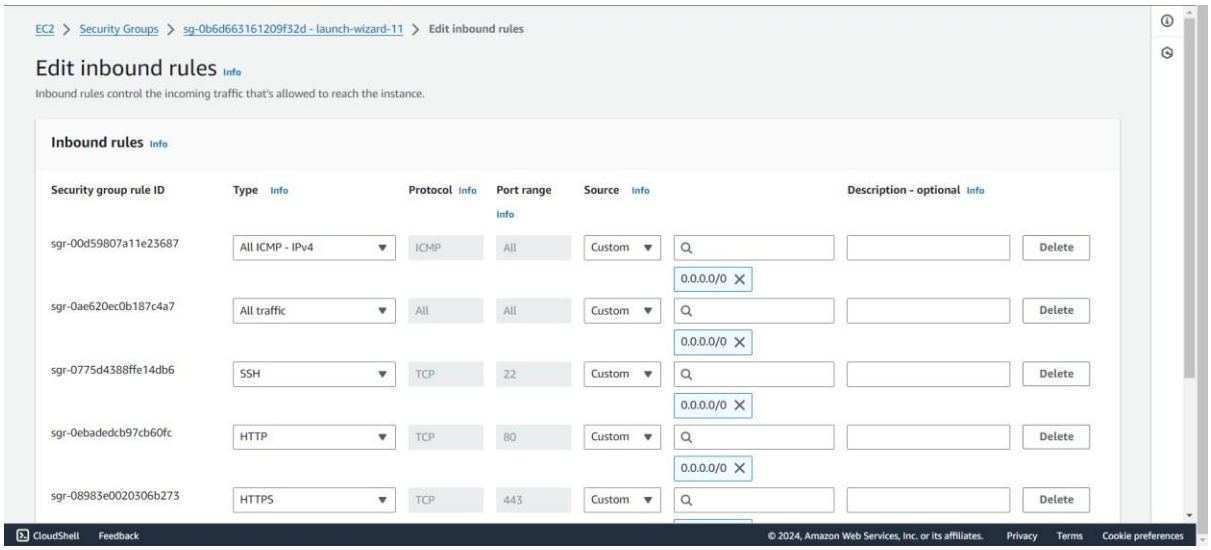
Aim :- To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

1. Create an Amazon Linux EC2 Instance



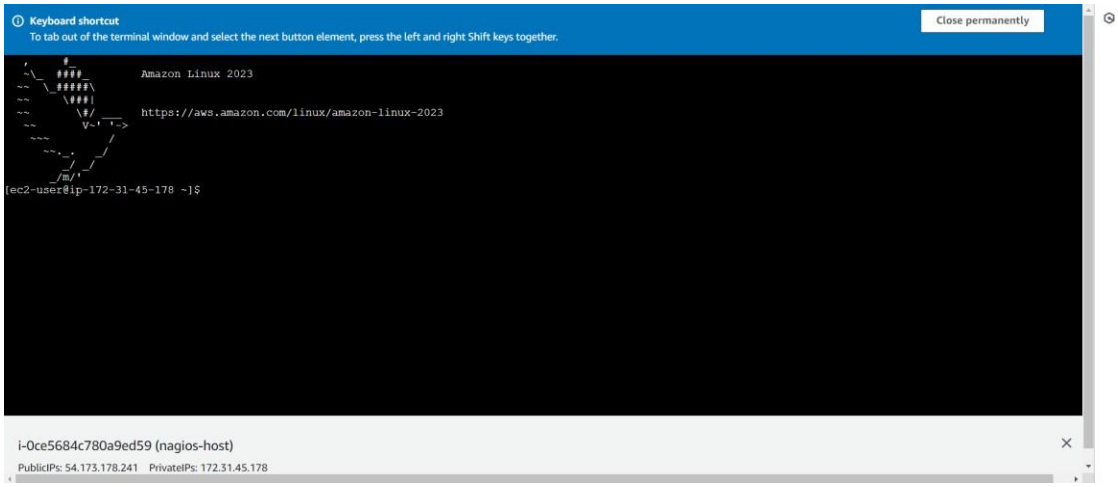
2. Configure Security Group

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



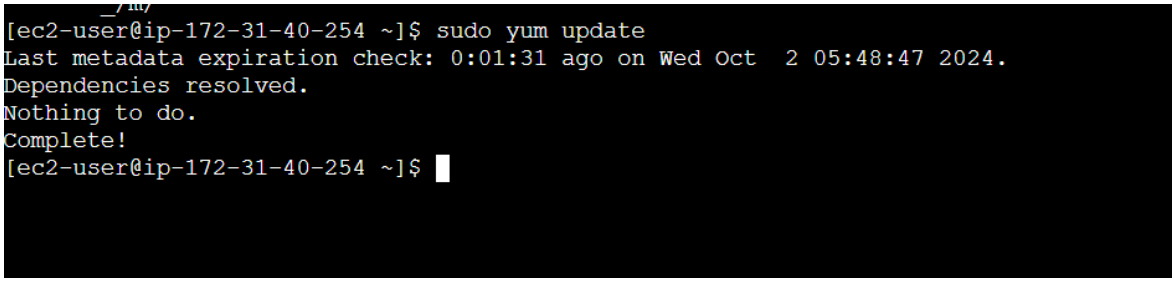
You have to edit the inbound rules of the specified Security Group for this.

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

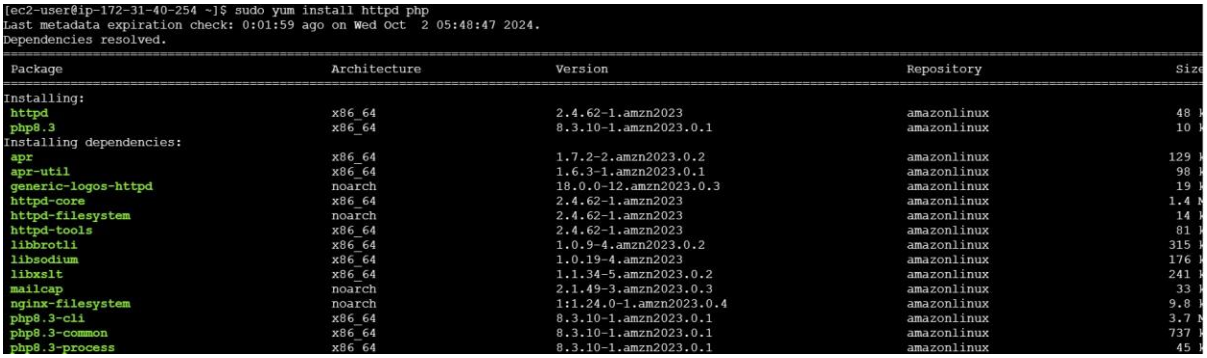


4. Update the package indices and install the following packages using yum

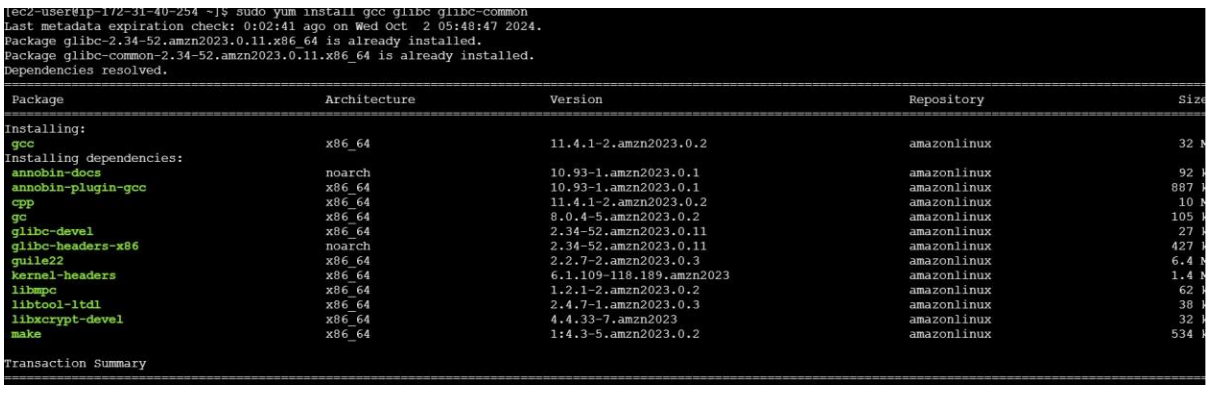
sudo yum update



sudo yum install httpd php



sudo yum install gcc glibc glibc-common



sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:46 ago on Wed Oct  2 05:48:47 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture    Version                    Repository      Size
================================================================================
Installing:
 gd                      x86_64          2.3.3-5.amzn2023.0.3       amazonlinux     139 k
 gd-devel                x86_64          2.3.3-5.amzn2023.0.3       amazonlinux      38 k
Installing dependencies:
 brotli                  x86_64          1.0.9-4.amzn2023.0.2       amazonlinux     314 k
 brotli-devel            x86_64          1.0.9-4.amzn2023.0.2       amazonlinux      31 k
 bzip2-devel             x86_64          1.0.8-6.amzn2023.0.2       amazonlinux     214 k
 cairo                   x86_64          1.17.6-2.amzn2023.0.1      amazonlinux     684 k
 cmake-filesystem        x86_64          3.22.2-1.amzn2023.0.4      amazonlinux      16 k
 fontconfig              x86_64          2.13.94-2.amzn2023.0.2     amazonlinux     273 k
 fontconfig-devel        x86_64          2.13.94-2.amzn2023.0.2     amazonlinux     128 k
 fonts-filesystem        noarch          1:2.0.5-12.amzn2023.0.2    amazonlinux     9.5 k
 freetype                x86_64          2.13.2-5.amzn2023.0.1      amazonlinux     423 k
 freetype-devel          x86_64          2.13.2-5.amzn2023.0.1      amazonlinux     912 k
 glib2-devel             x86_64          2.74.7-689.amzn2023.0.2    amazonlinux     486 k
 google-noto-fonts-common noarch         20201206-2.amzn2023.0.2    amazonlinux      15 k
 google-noto-sans-vf-fonts noarch        20201206-2.amzn2023.0.2    amazonlinux     492 k
 graphite2               x86_64          1.3.14-7.amzn2023.0.2      amazonlinux      97 k
 graphite2-devel         x86_64          1.3.14-7.amzn2023.0.2      amazonlinux      21 k
 harfbuzz                x86_64          7.0.0-2.amzn2023.0.1       amazonlinux     868 k
 harfbuzz-devel          x86_64          7.0.0-2.amzn2023.0.1       amazonlinux     404 k
 harfbuzz-icu            x86_64          7.0.0-2.amzn2023.0.1       amazonlinux      18 k
```

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

sudo adduser -m nagios

sudo passwd nagios

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

6. Create a new user group

sudo groupadd nagcmd

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

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

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

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

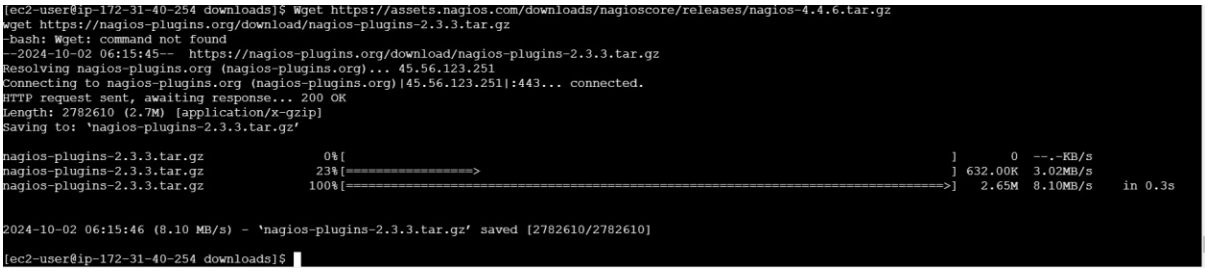
8. Create a new directory for Nagios downloads

mkdir ~/downloads
cd ~/downloads

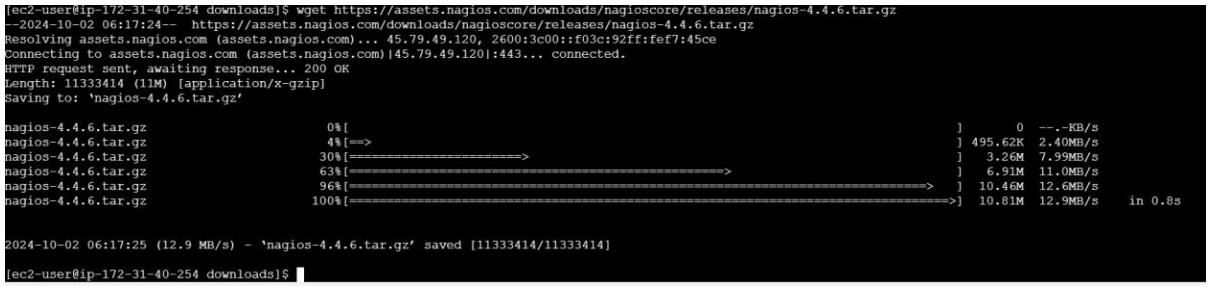
```
[ec2-user@ip-172-31-40-254 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-40-254 downloads]$
```

9. Use wget to download the source zip files.

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

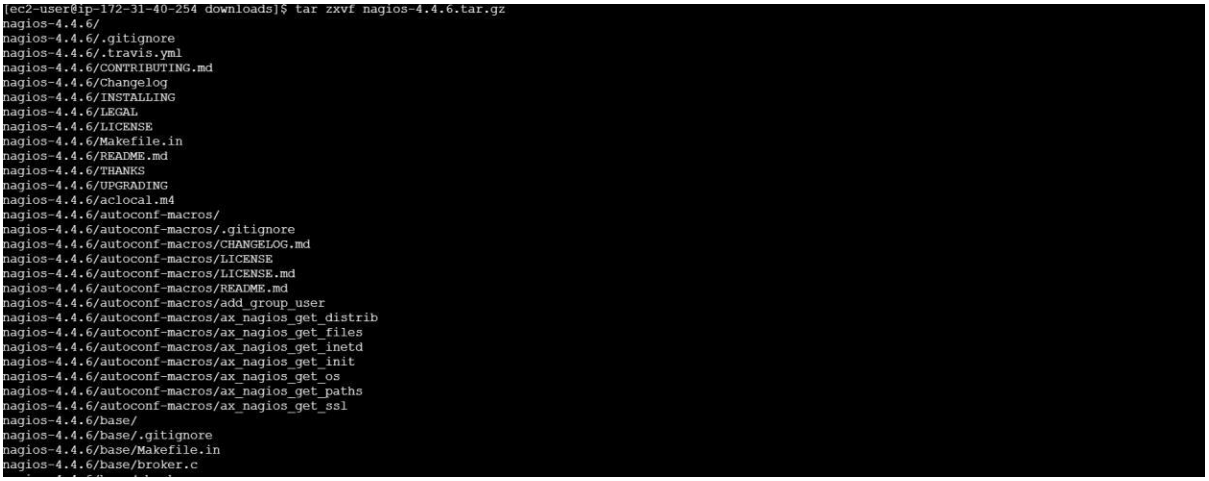


https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz



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

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



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

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

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

12. Compile the source code.

make all

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

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


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

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

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

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

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

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

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

Enjoy.
```

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

./sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

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

*** Config files installed ***

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

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

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

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

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

```
  GNU nano 5.8                                  /usr/local/nagios/etc/objects/contacts.cfg                                  Modified
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact template (defined above)
    alias                   Nagios Admin            ; Full name of user
    email                   bhagyeshpatil0702@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}



###############################################################################
#
# CONTACT GROUPS
#
###############################################################################

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {

^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
```

15. Configure the web interface.

sudo make install-webconf

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

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

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

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

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

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

17. Restart Apache

sudo systemctl restart httpd

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

cd ~/downloads

tar zxvf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

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

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

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

make sudo

make install

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

19. Compile and install plugins

sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

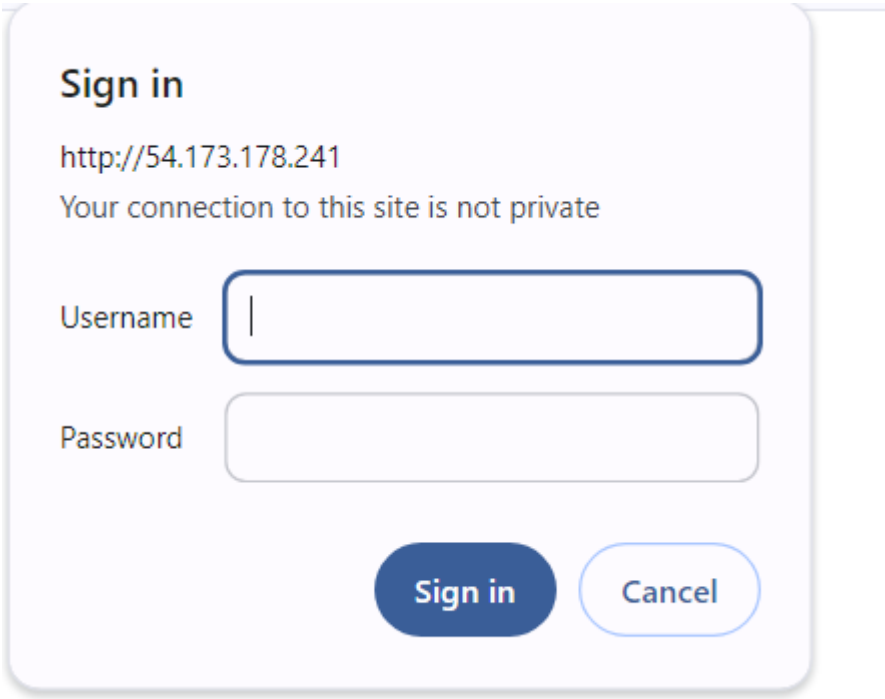
sudo systemctl start nagios

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

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

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

Running pre-flight check on configuration data...

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

20. Check the status of Nagios

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

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

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




Not secure 54.173.178.241/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Nagios Core
Daemon running with PID 67992
Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates
A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.
Get Started
Start monitoring your infrastructure
Change the look and feel of Nagios
Extend Nagios with hundreds of addons
Get support
Get training
Get certified
Quick Links
Nagios Library (tutorials and docs)
Nagios Labs (development blog)
Nagios Exchange (plugins and addons)
Nagios Support (tech support)
Nagios.com (company)
Nagios.org (project)
Latest News
Don't Miss...
Page Tour
Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Name : Swaraj Patil
Class : D15A
Roll No. : 35

EXPERIMENT 10

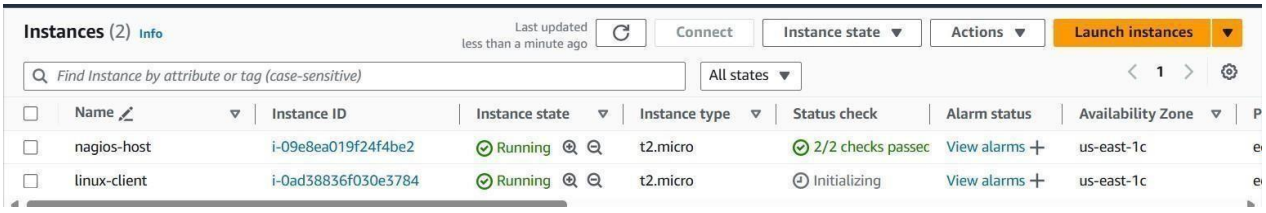
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Procedure:-

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

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

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

Now, create a new EC2 instance on AWS



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

ps -ef | grep nagios

```
ubuntu@ip-172-31-89-161:~$ ps -ef | grep nagios
nagios     15764       1  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     15765   15764  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     15766   15764  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     15767   15764  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     15768   15764  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     15769   15764  0 16:08 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ubuntu     15957    1342  0 16:13 pts/0    00:00:00 grep --color=auto nagios
ubuntu@ip-172-31-89-161:~$
```

Sudo su

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

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

```
ubuntu@ip-172-31-89-161:~$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/home/ubuntu#
```

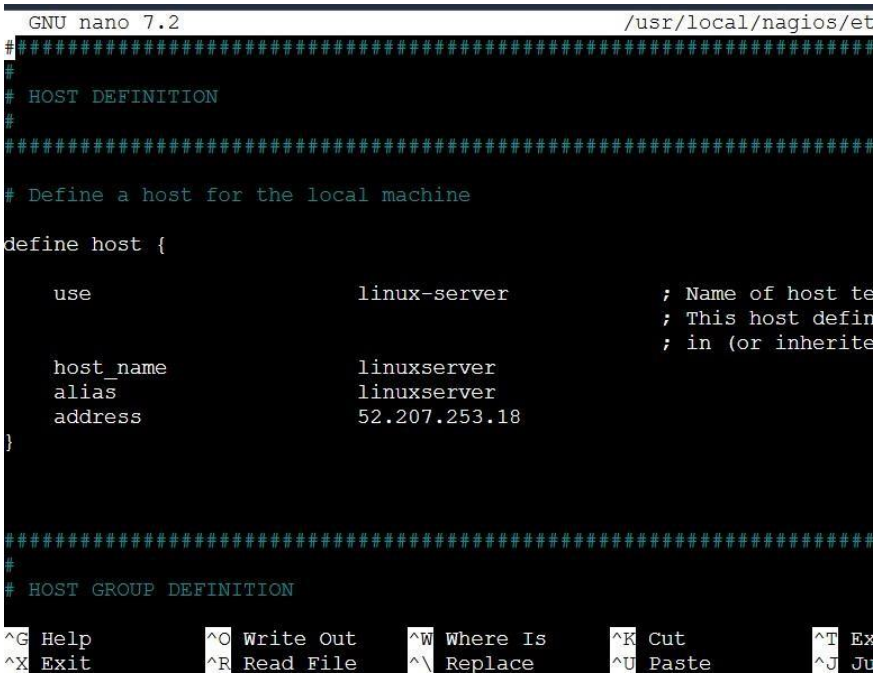
Copy localhost.cfg file to the mentioned location

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

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts': No such file or directory
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# sudo mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects#
```

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

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



Note - Here replace hostname with linuxserver

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

Add the following line to the nagios.cfg file

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

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

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


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

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

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

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

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

```
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0

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

Now restart the service by using this command

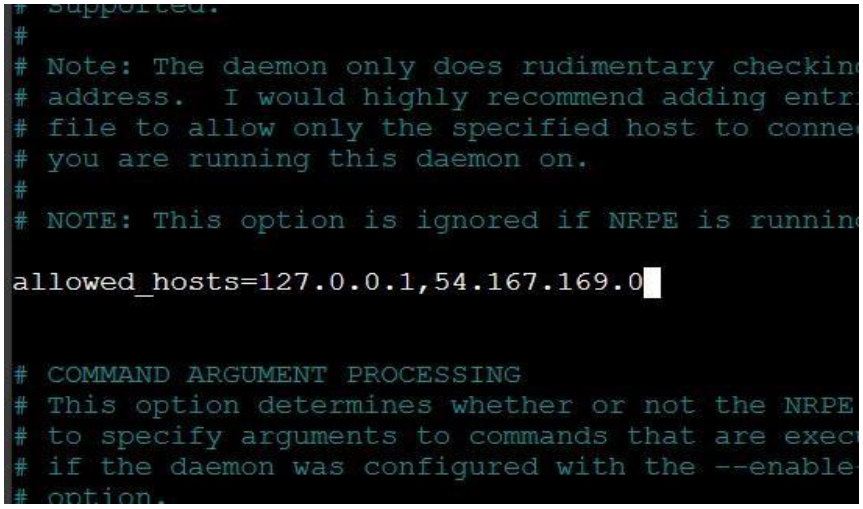
service nagios restart

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

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

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

sudo apt install -y nagios-nrpe-server nagios-plugins

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

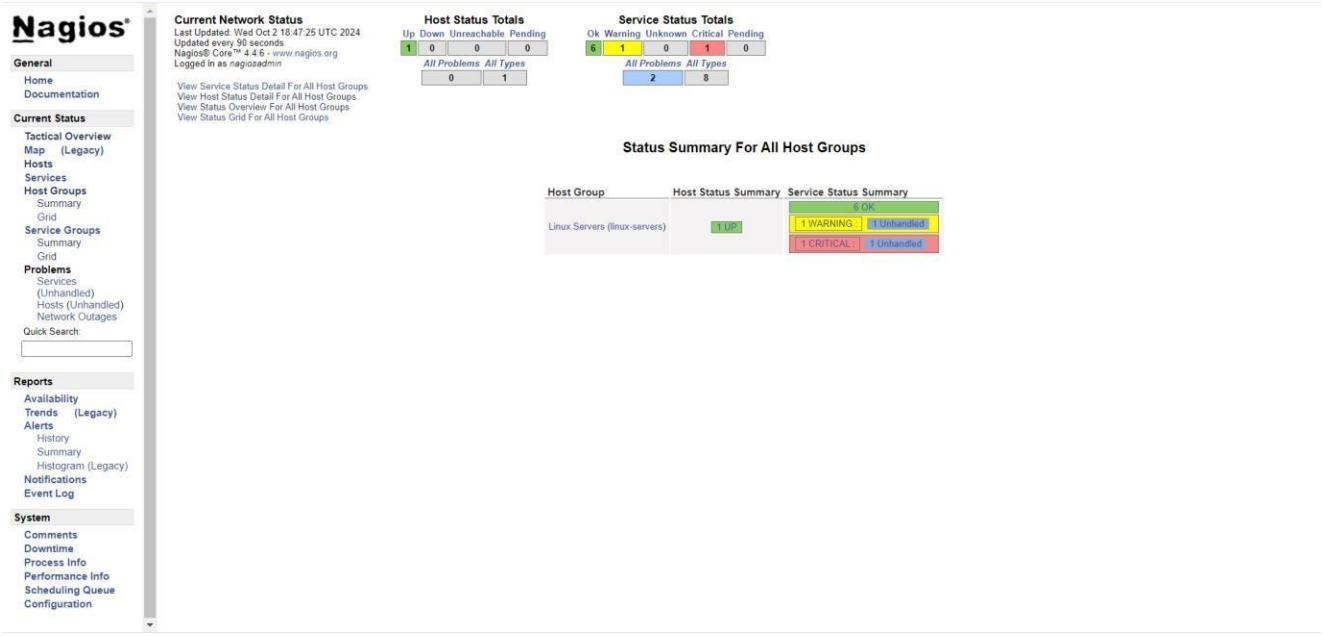


sudo nano /etc/nagios/nrpe.cfg

Now restart nrpe server by using this command

sudo systemctl restart nagios-nrpe-server

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



Experiment 11

Swaraj Patil
D15A 40

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, whichexecutes 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 ofdata 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 careof 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 canfocus 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. Youdo not have to worry about the application down.

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket.
And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'llgo over this in detail later on in this guide.

Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served,a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container withour function to serve those requests.
This has some interesting implications. Firstly, our functions are effectively stateless.
Secondly, each request (or event) is served by a single instance of a Lambda function.This means that you are not going to be handling concurrent requests in your code.
AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

The above execution model makes Lambda functions effectively stateless. This meansthat every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event.
However, due to the optimization noted above, the actual Lambda function is invoked only once per container instantiation. Recall that our functions are run inside containers.So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.
For example, the createNewDbConnection method below is called once per container instantiation and not every time the Lambda function is invoked. The myHandler function on the other hand is called on every invocation.

Common Use Cases for Lambda

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

1. Individual tasks run for a short time;
2. Each task is generally self-contained;

3. There is a large difference between the lowest and highest levels in the workloadof the application.

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

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

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

Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1.Choose a Function Creation

Method:

Select Author from scratch.

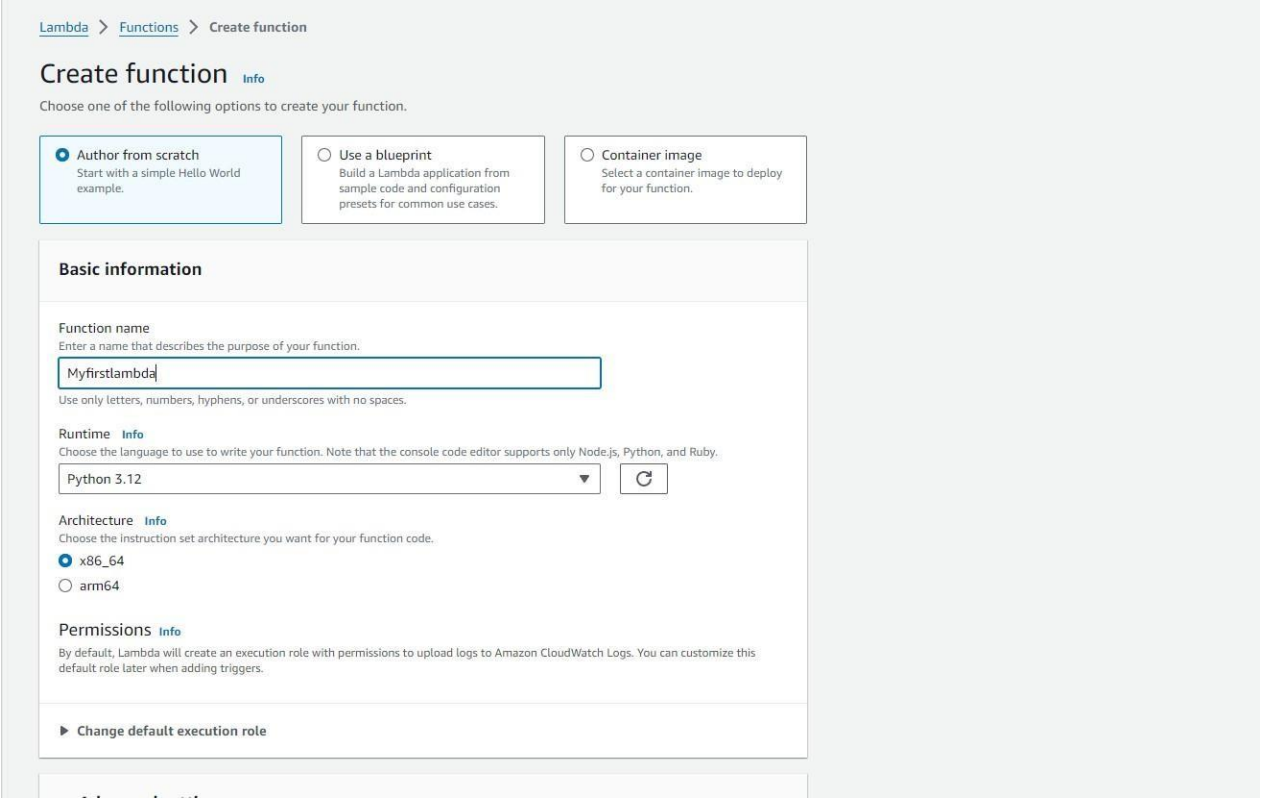
2. Configure the Function:

Function name: Enter a name for your function (e.g., MyFirstLambda).

Runtime: Choose Python 3.x (the latest available version).

Permissions:Choose Create a new role with basic Lambda permissions (this creates arole with the necessary permissions).

3. Click on Create function.



Step 2: Write Your Lambda Function Code

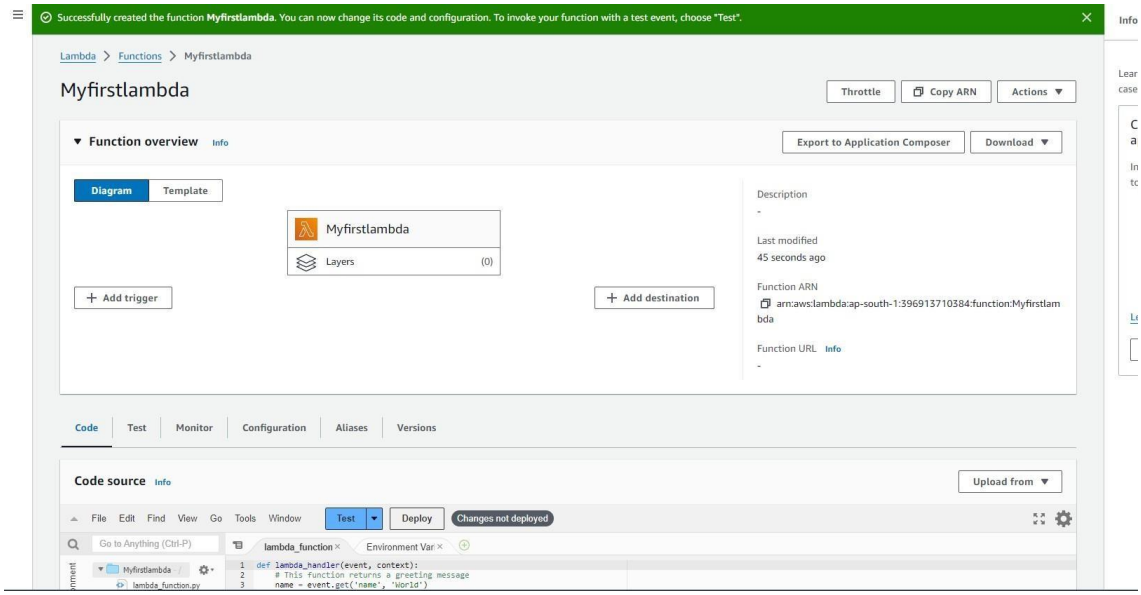
In the Function code section, you will see a code editor. Replace the default code withthe following Python code:

python

Copy code

```
def lambda_handler(event, context):
    # This function returns a greeting message
    name = event.get('name', 'World')
    return {
        'statusCode': 200, 'body':
        f'Hello, {name}!'
    }
```

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



Step3:

1. Configure a Test Event:

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

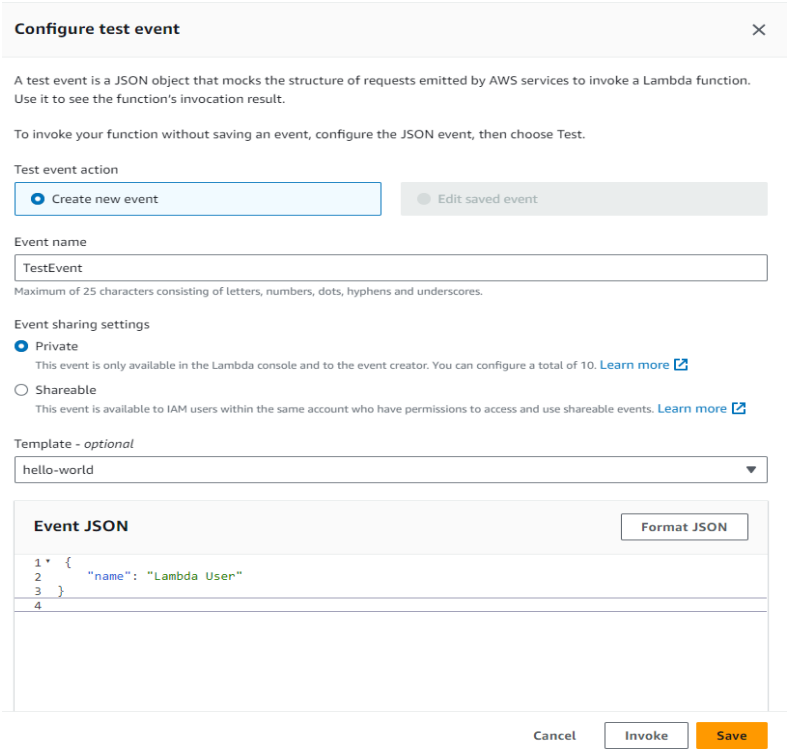
```
{
   "name": "Lambda User"
}
```

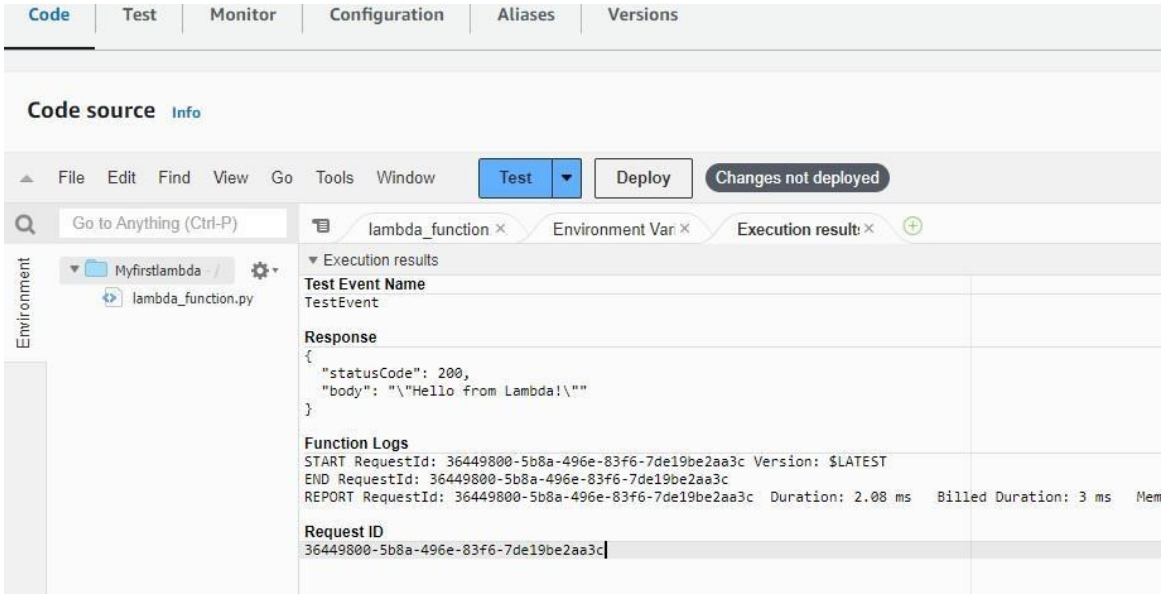
2. Run the Test:

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

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

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





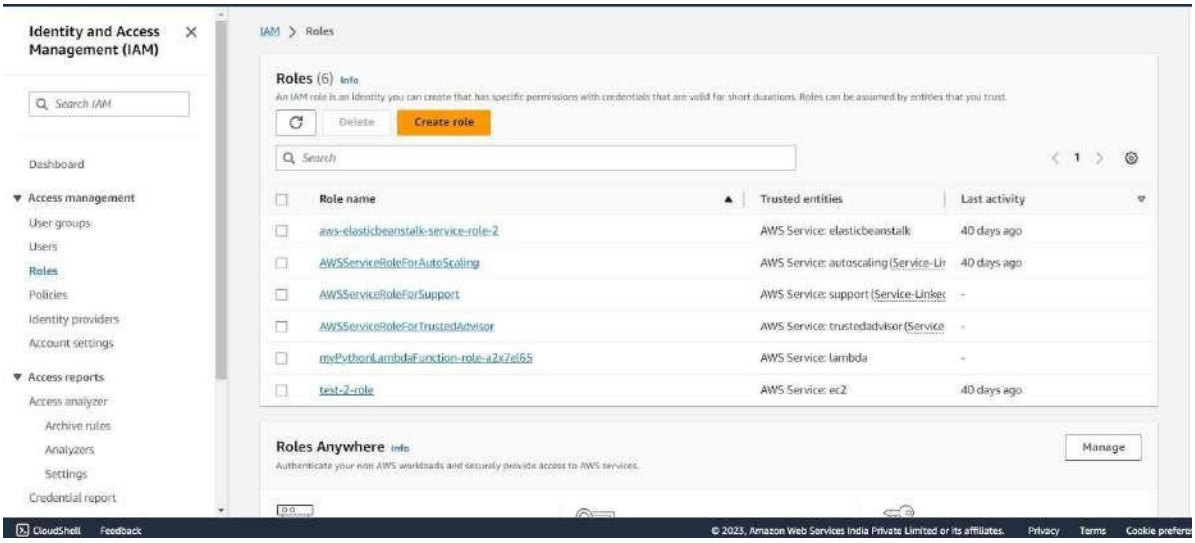
Conclusion:

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

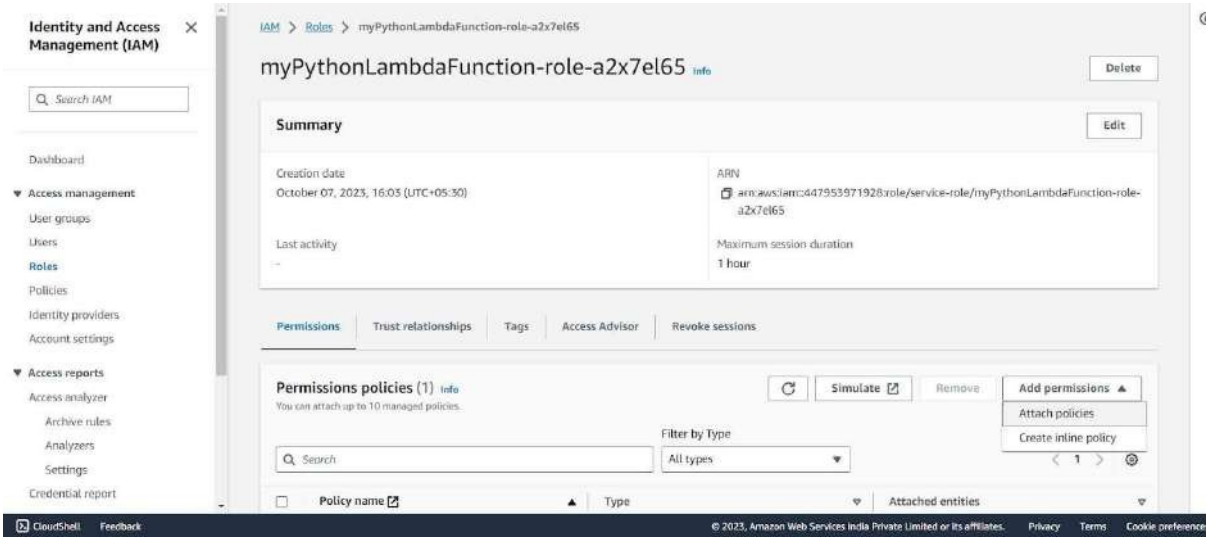
Adv. DevOps Exp. 12

Swaraj Patil
D15A 40

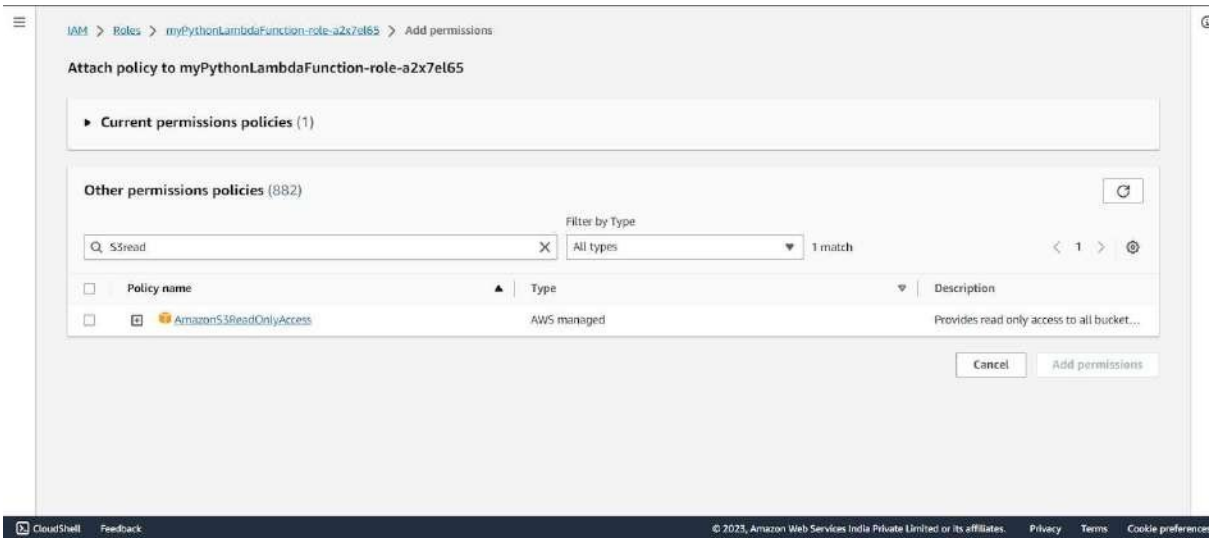
Step 1: Open the IAM (user)



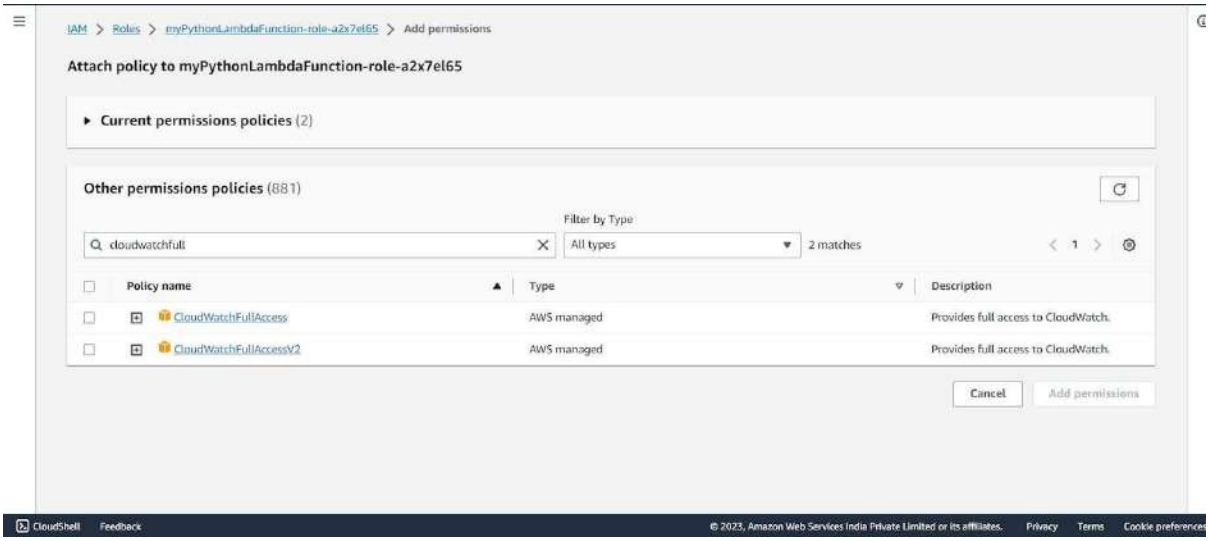
Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.



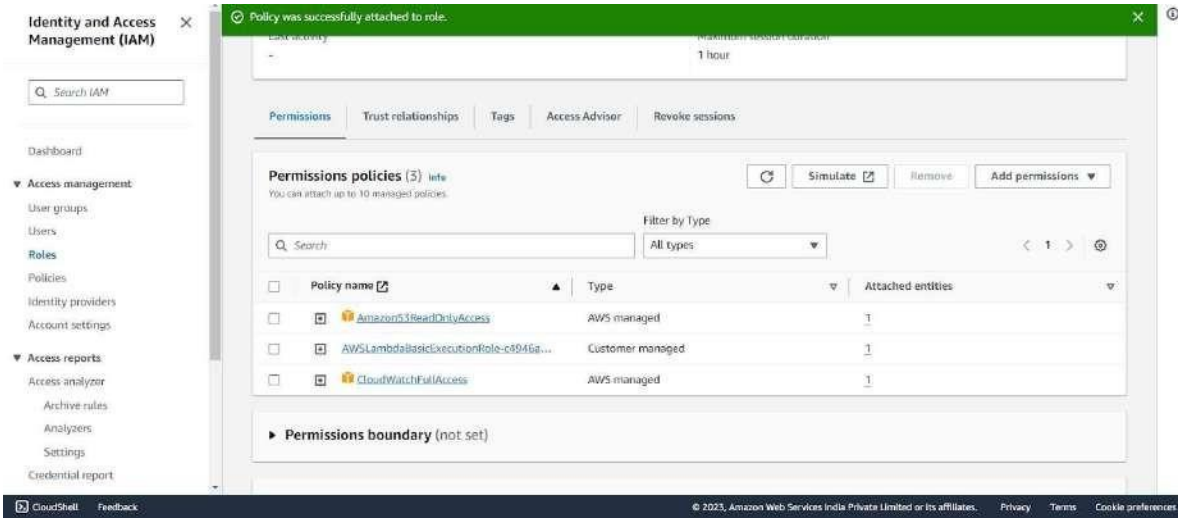
S3-ReadOnly



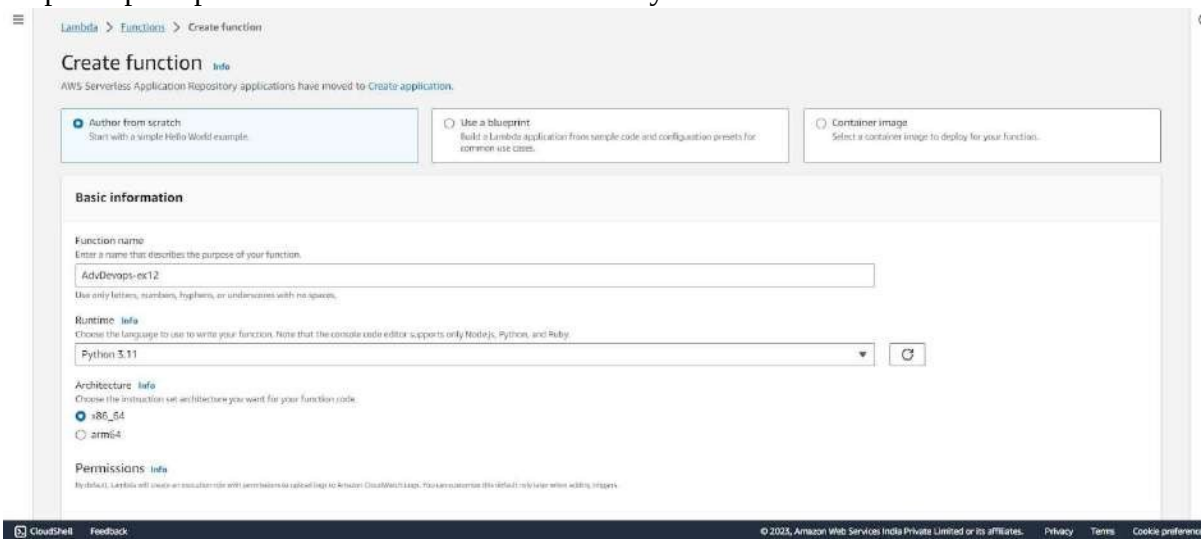
CloudWatchFull



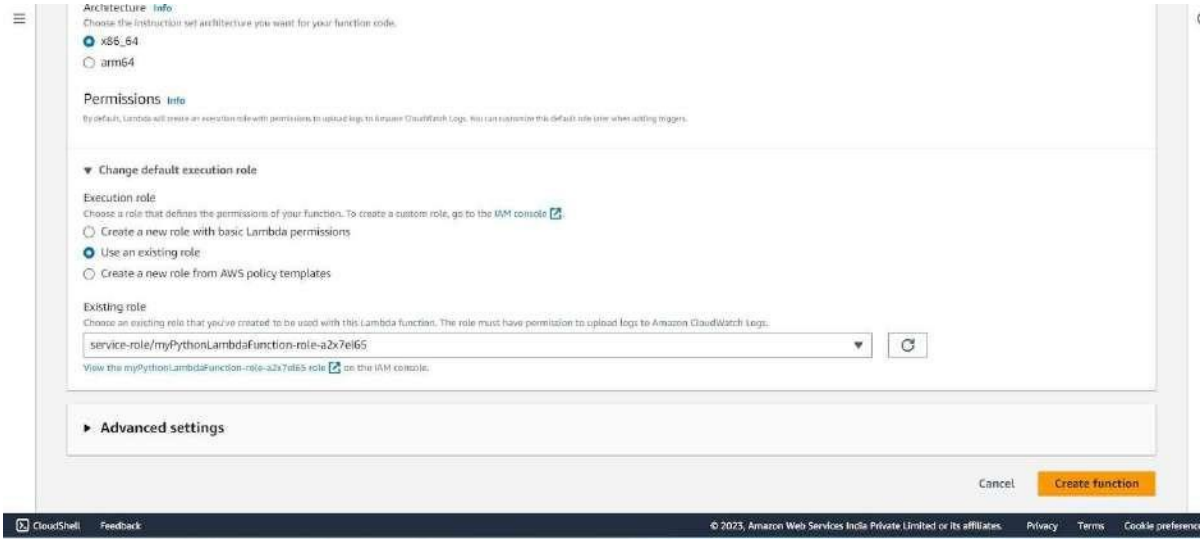
After successful attachment of policy you will see something like this you will beable to see the updated policies.



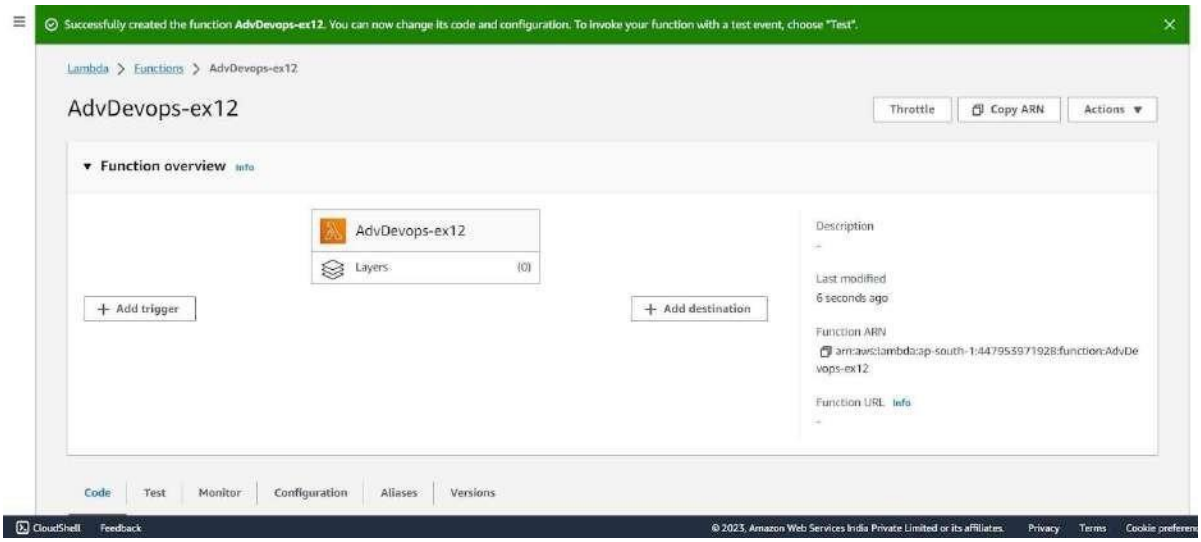
Step 3: Open up AWS Lambda and create a new Python function.



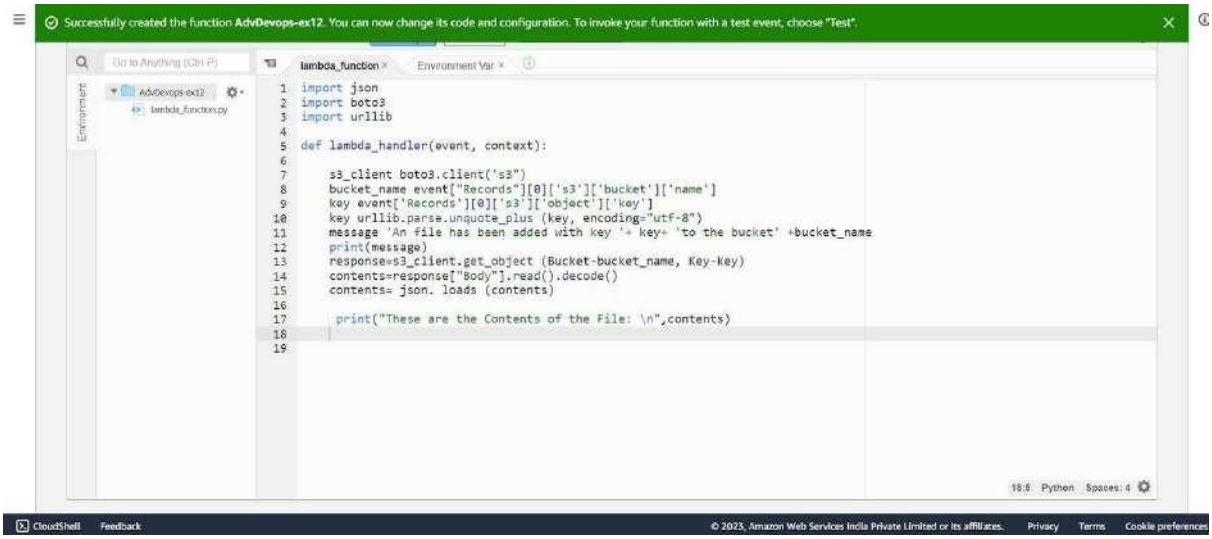
Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.



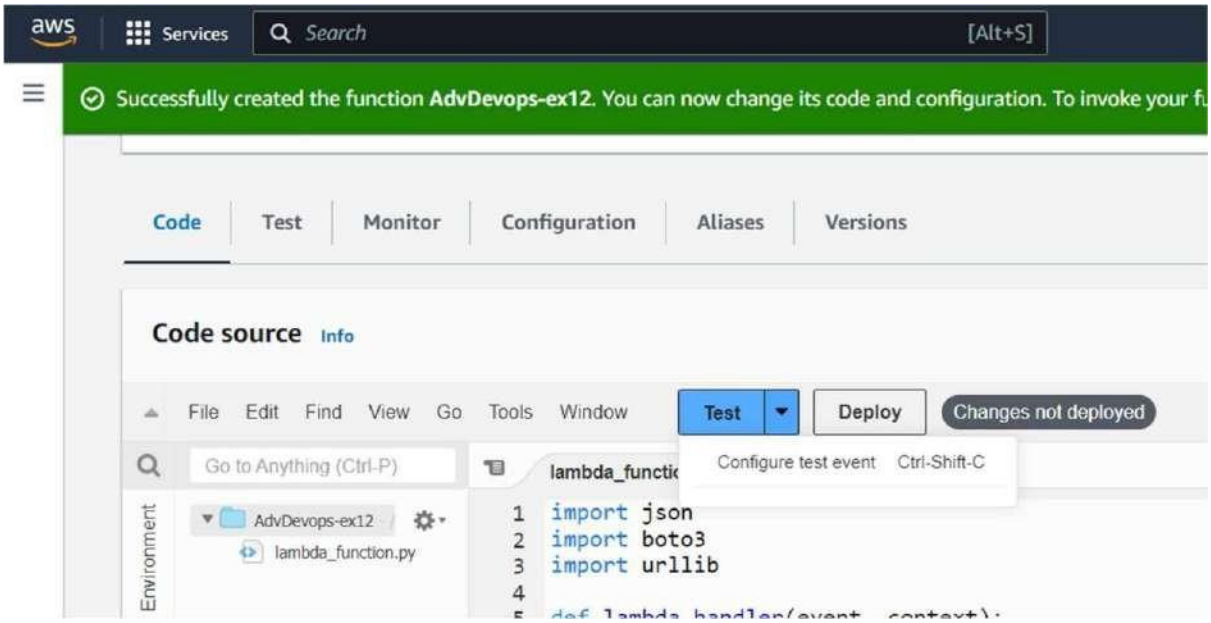
Step 4: The function is up and running.

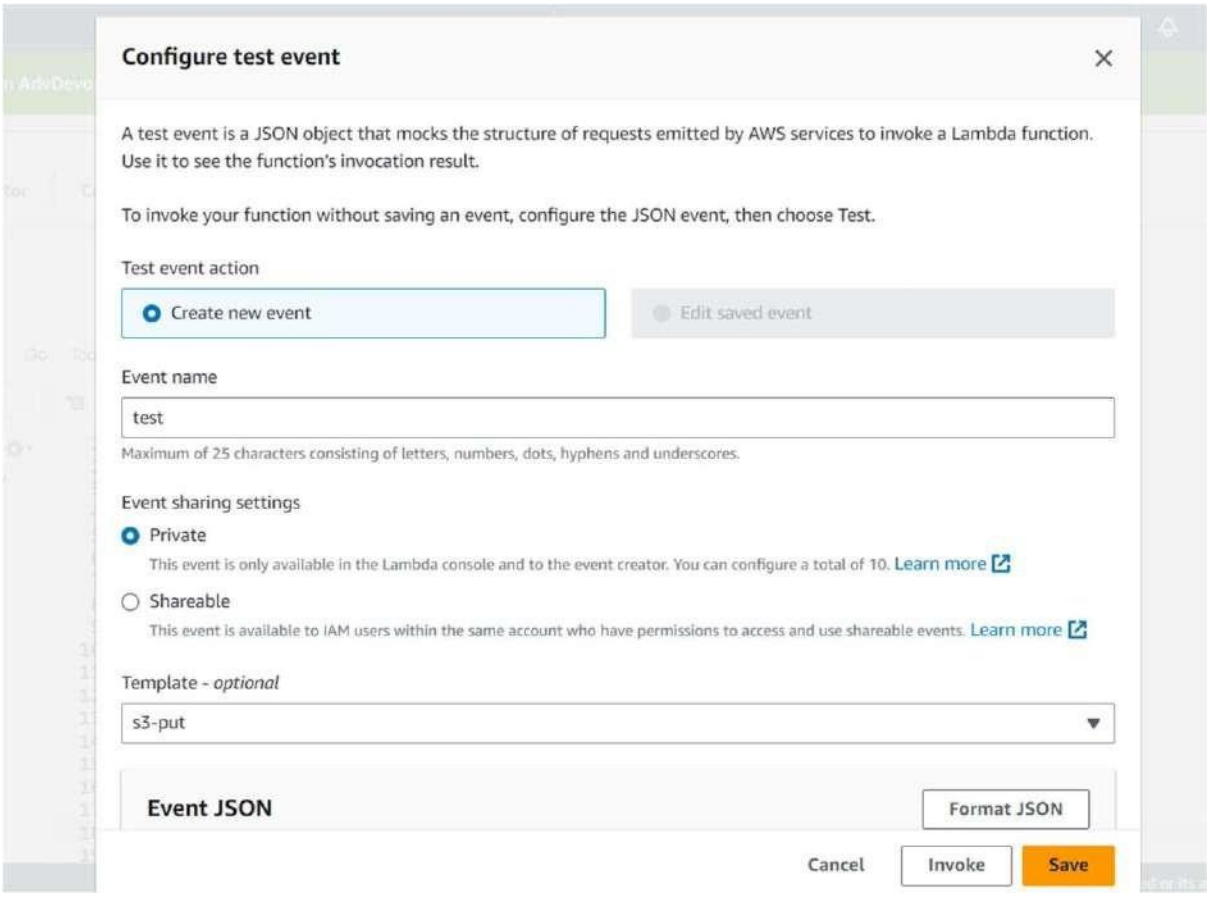


Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.



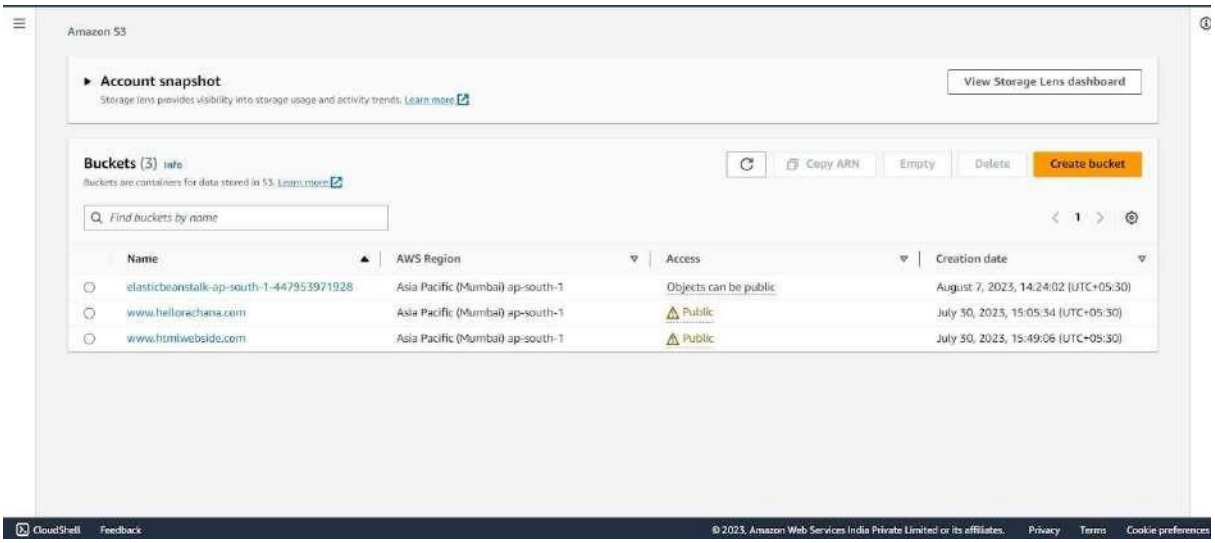
Step 6: Click on Test and choose the ‘S3 Put’ Template.



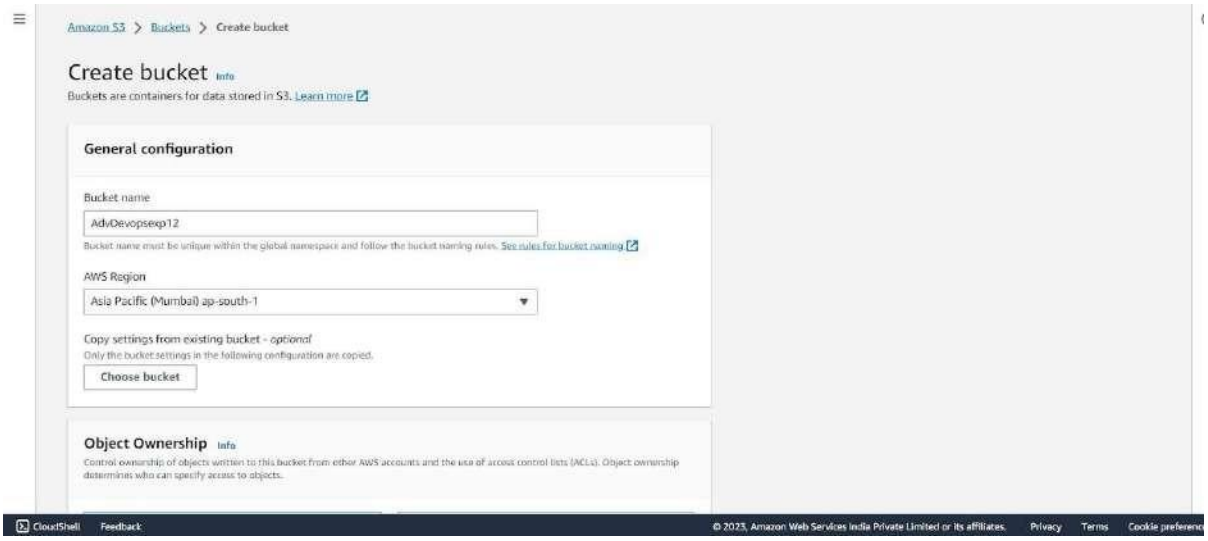


And Save it.

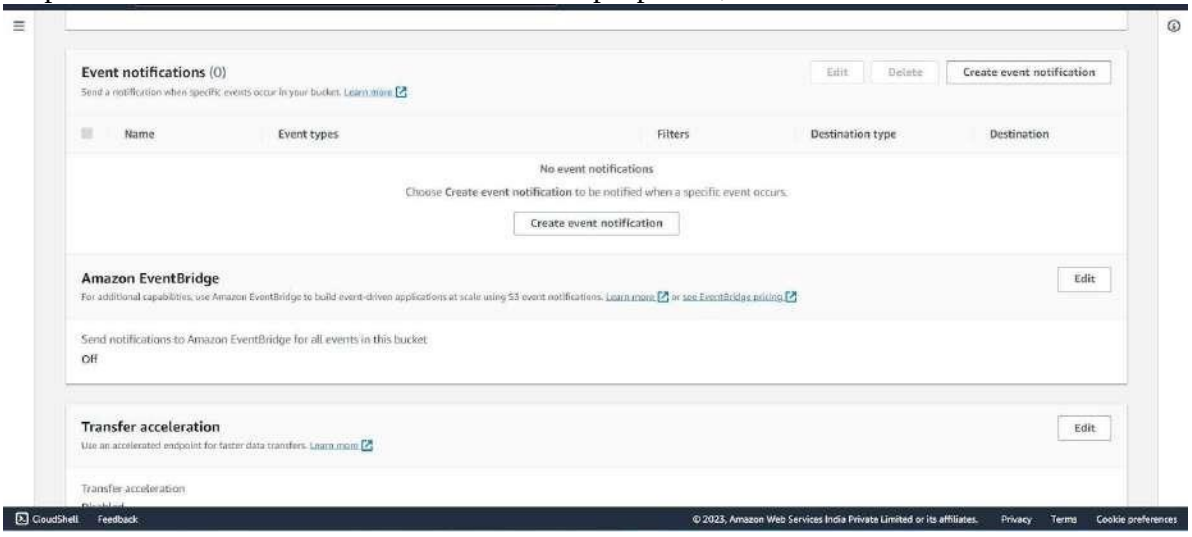
Step 7: Open up the S3 Console and create a new bucket.



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

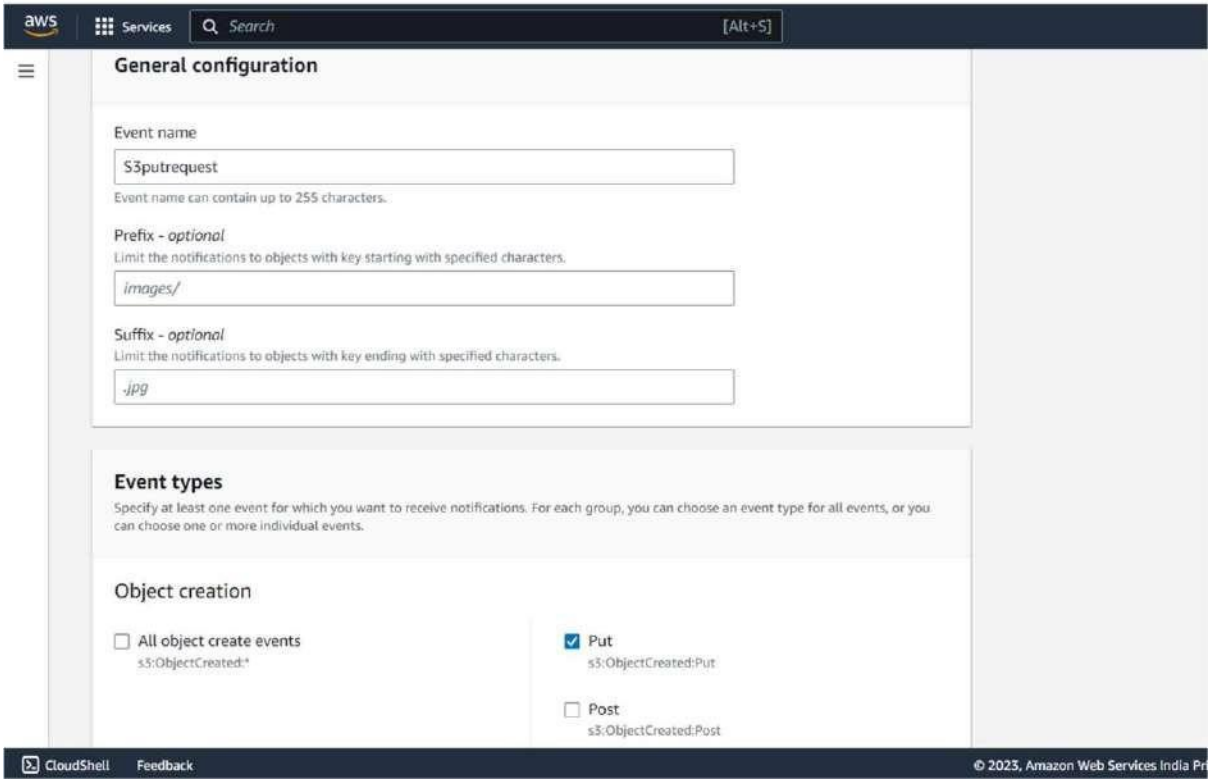


Step 9: Click on the created bucket and under properties, look for events.

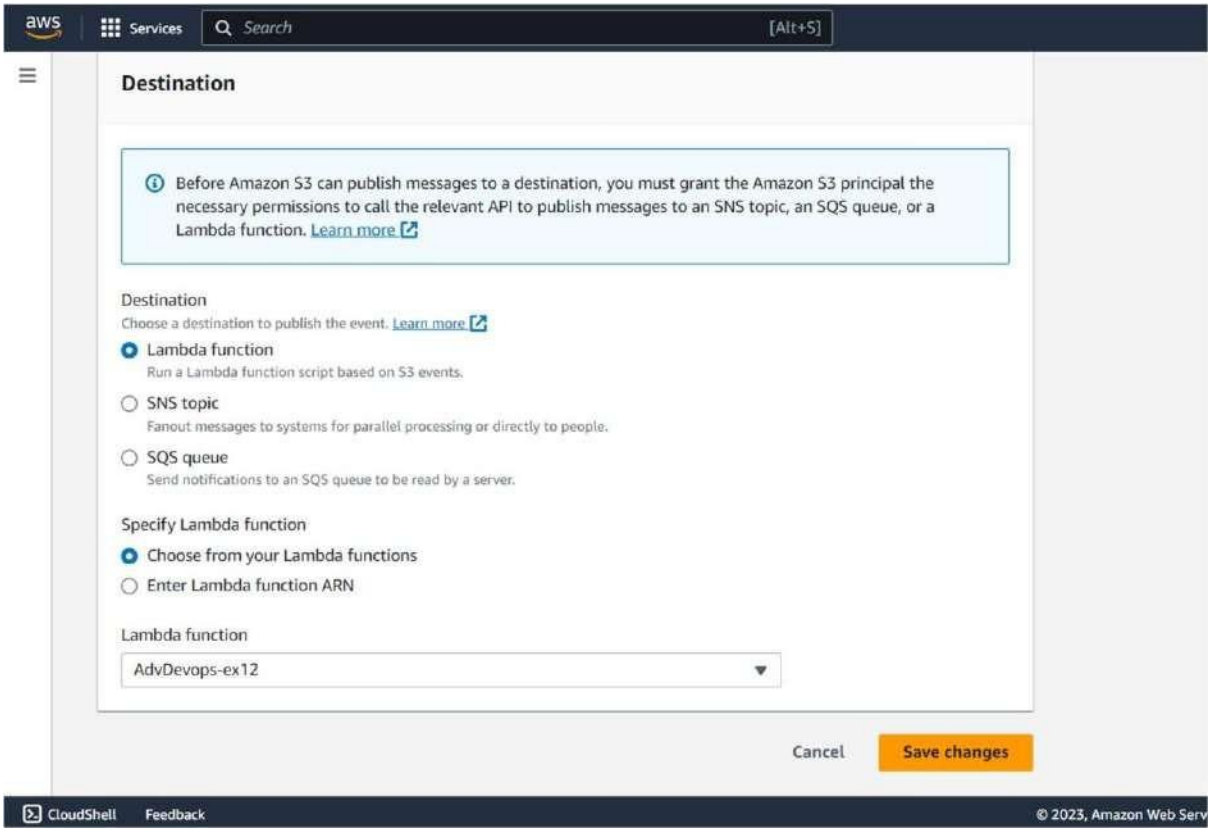


Click on Create Event Notification.

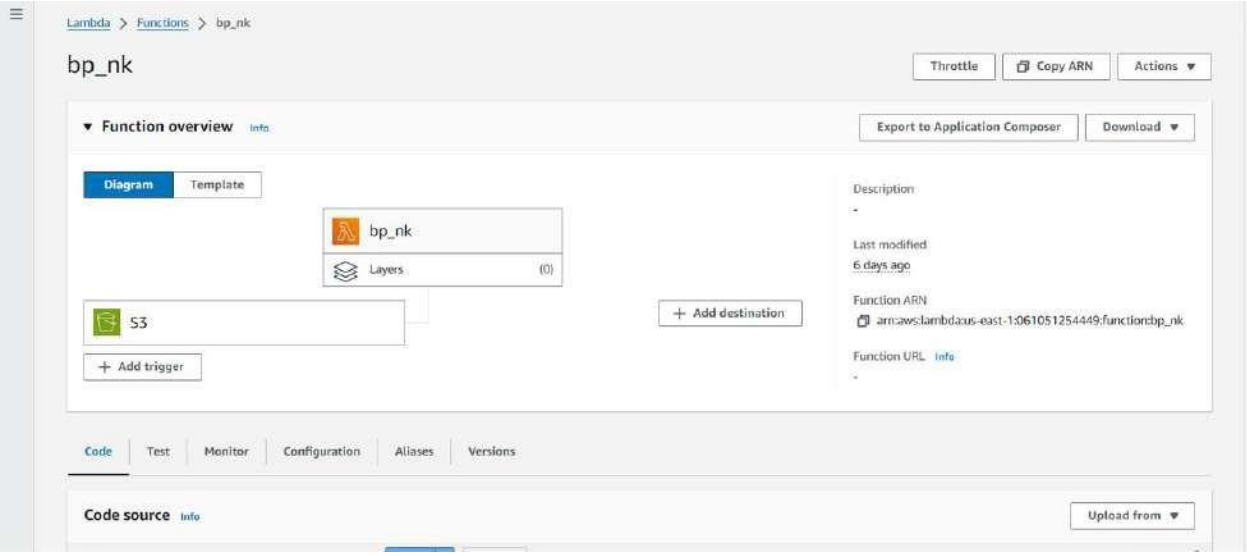
Step 10: Mention an event name and check Put under event types.



Choose Lambda function as destination and choose your lambda function and savethe changes.

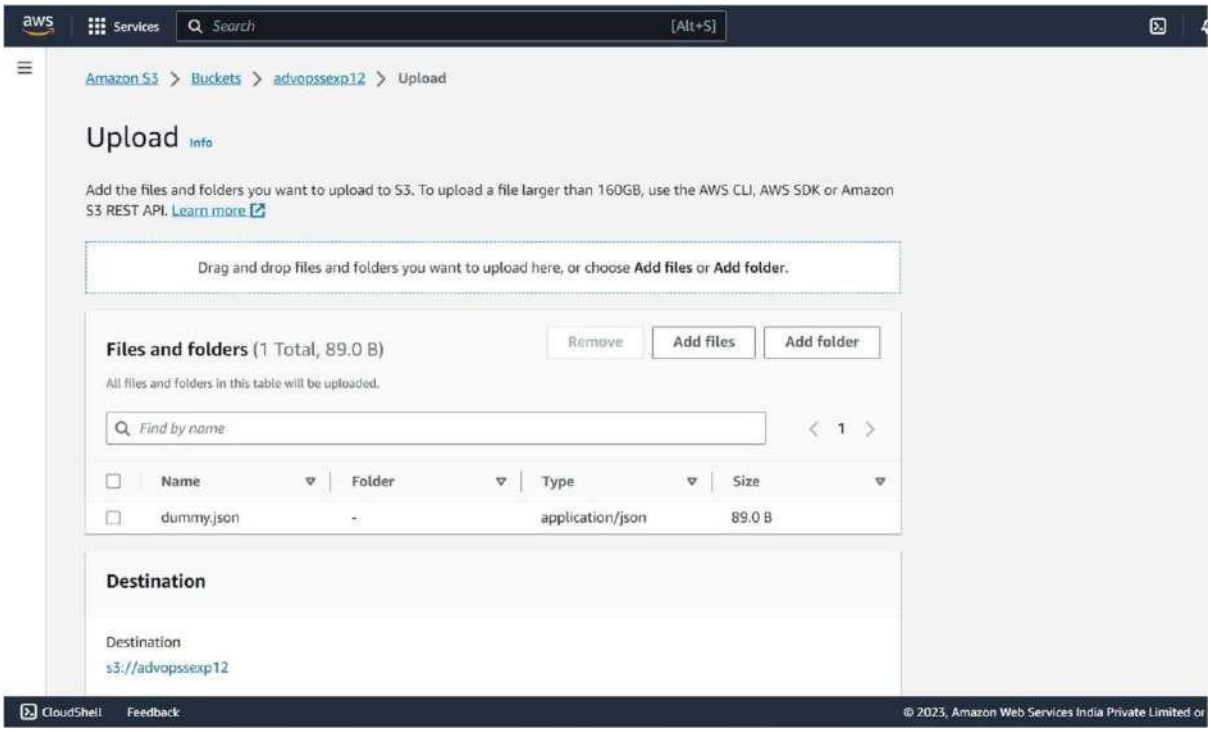


Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



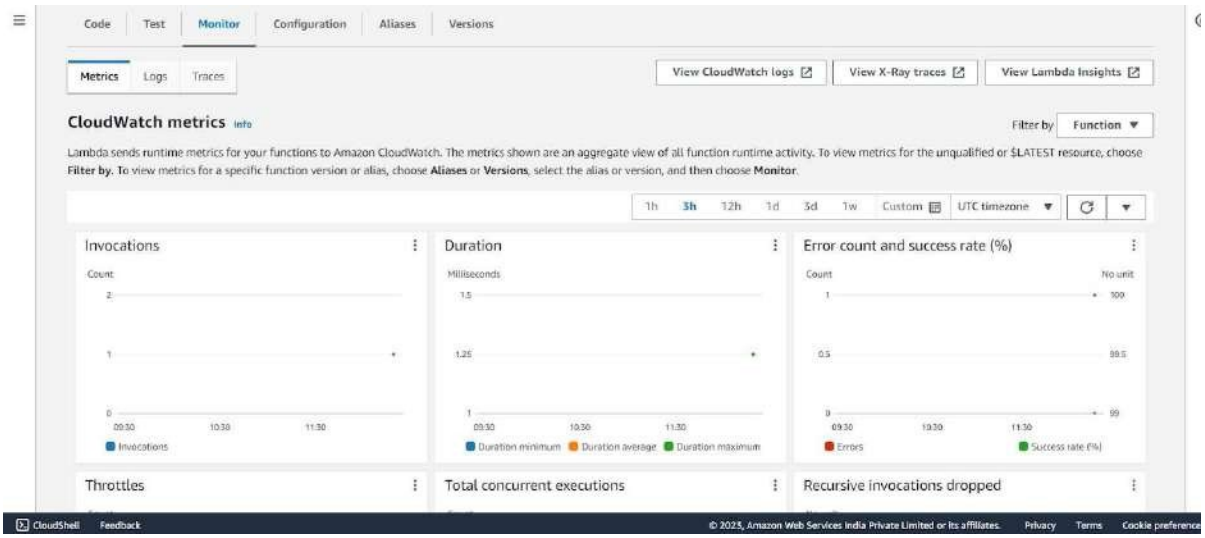
Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.
Step 14: Select the dummy data file from your computer and click Upload.

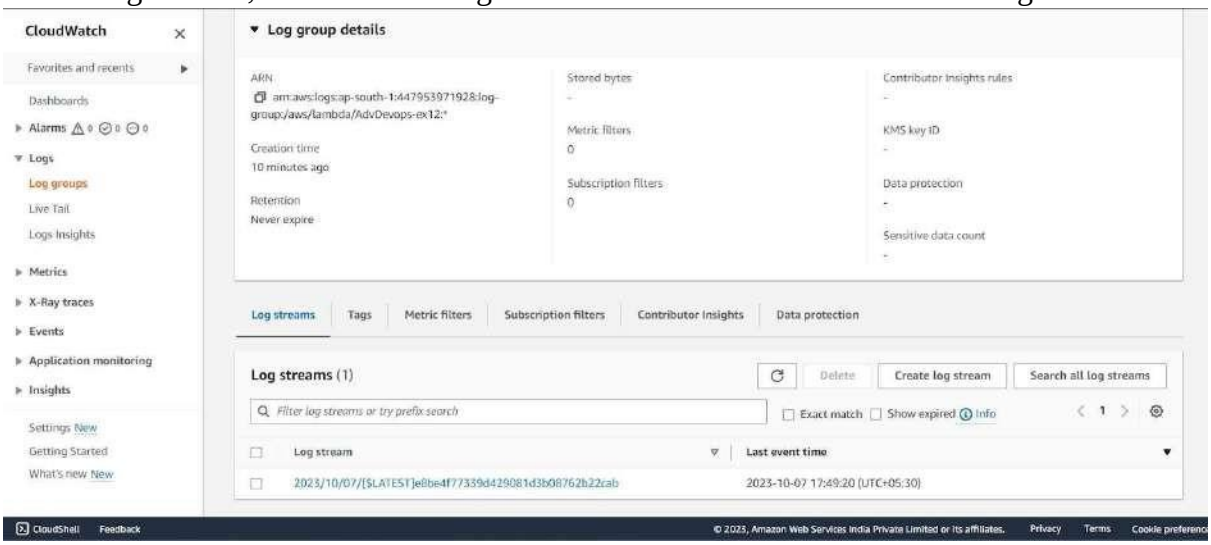


Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

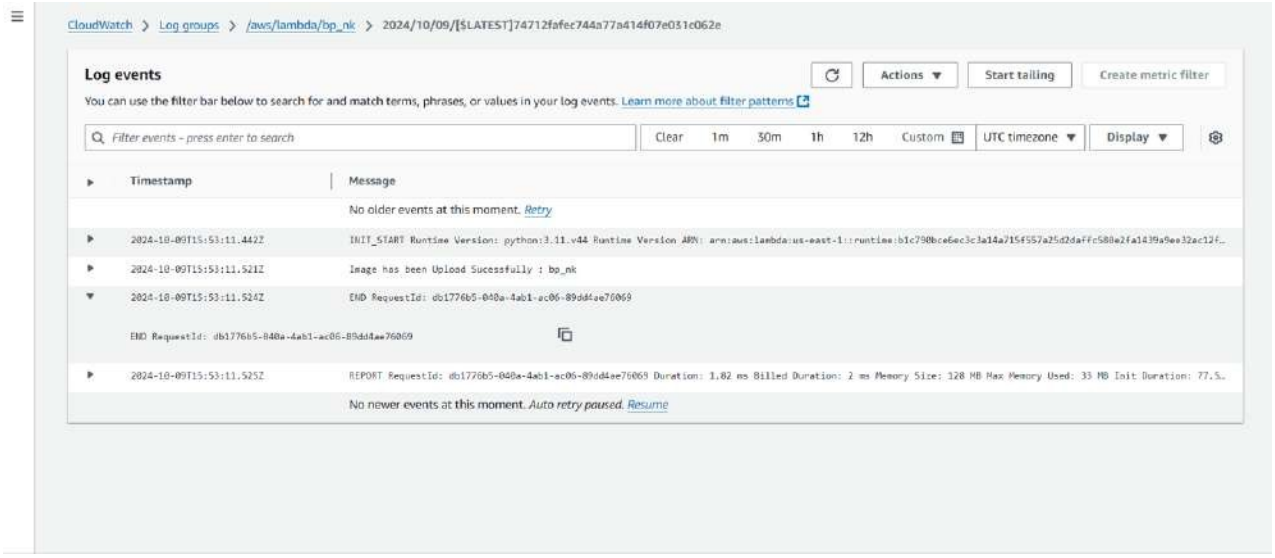
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



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



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



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