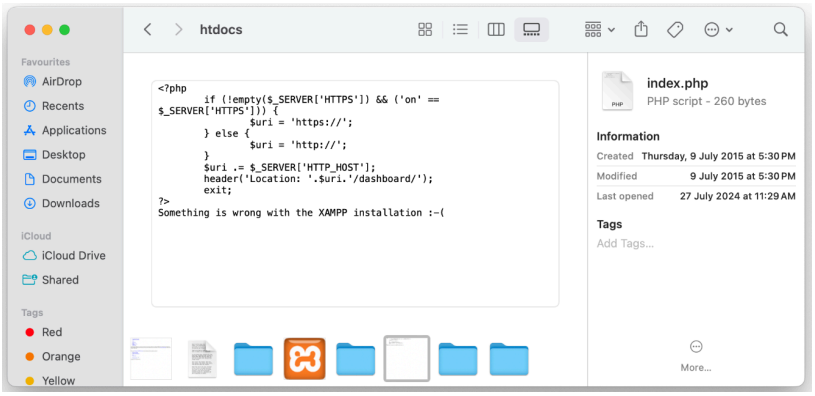
Experiment 1A

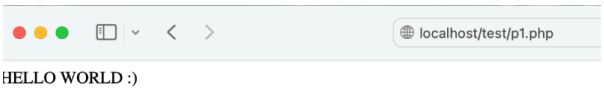
Hosting a static site using xampp

1. Download and open the xampp application

2. Go to the files “htdocs” and upload the php code there



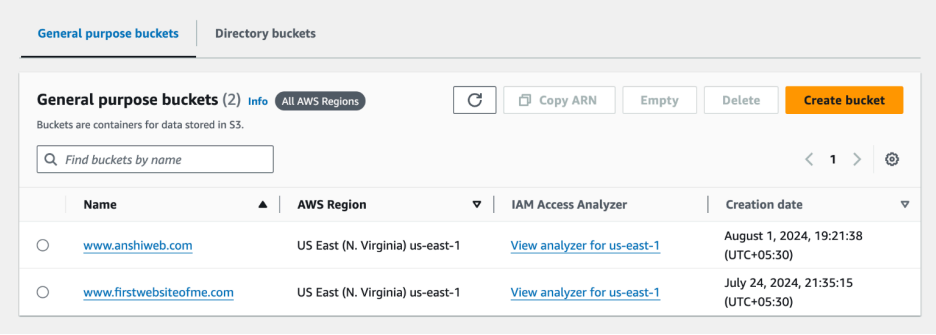
3. Go to xampp, enter the folder name in the top link and run the file
   You will see the website



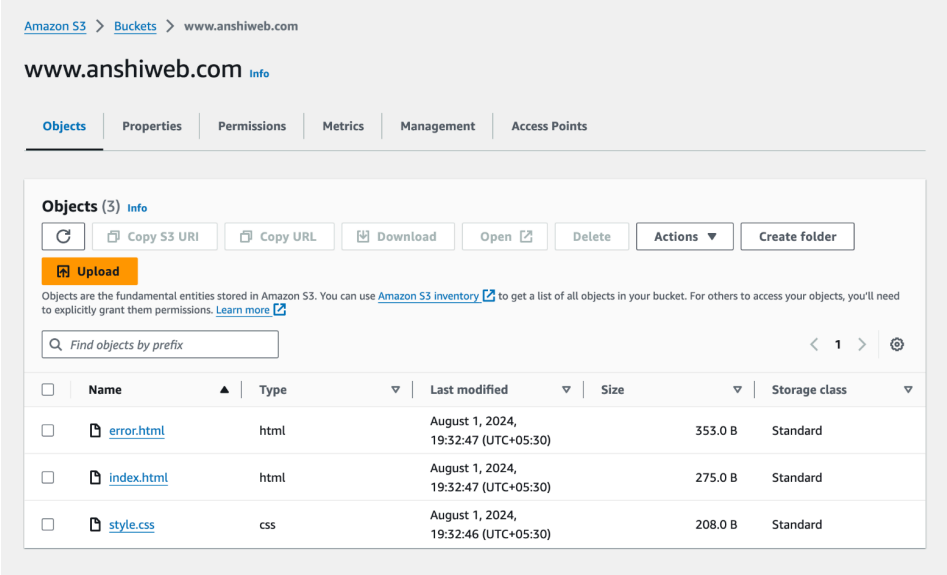
HELLO WORLD :)

Hosting a static website using AWS S3

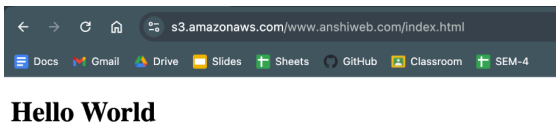
1. Create a bucket in AWS S3 interface



2. Upload all your files in the buckets inventory



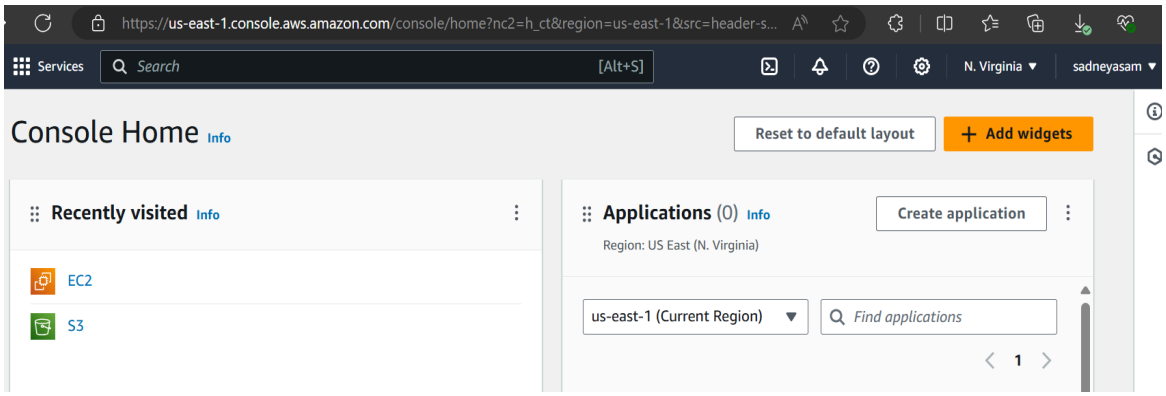
3. Click on the link of the bucket to go to the hosted site



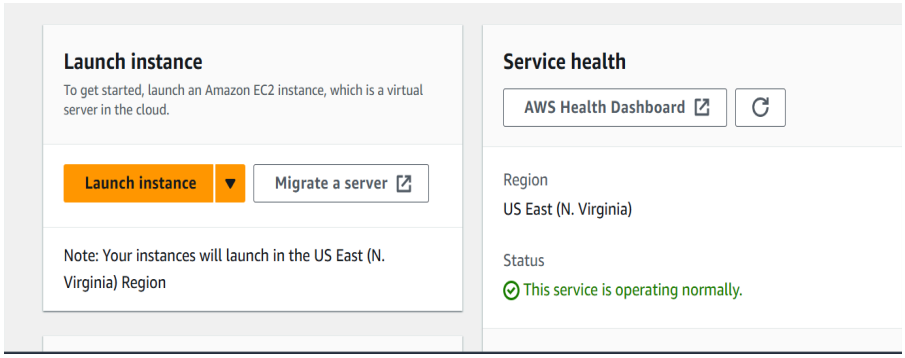
Hello World

Create instance using EC2

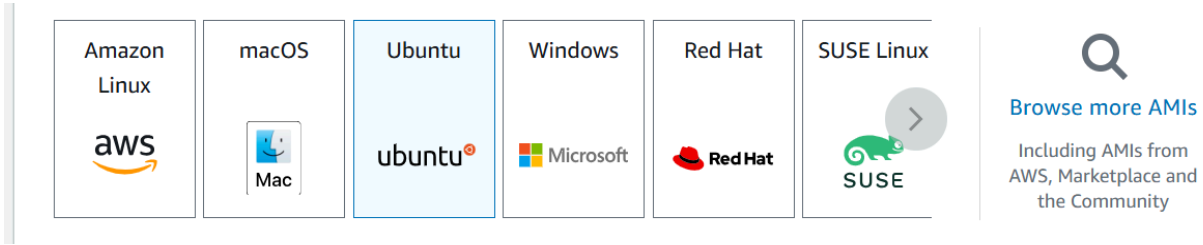
1. Open the aws console and choose EC2



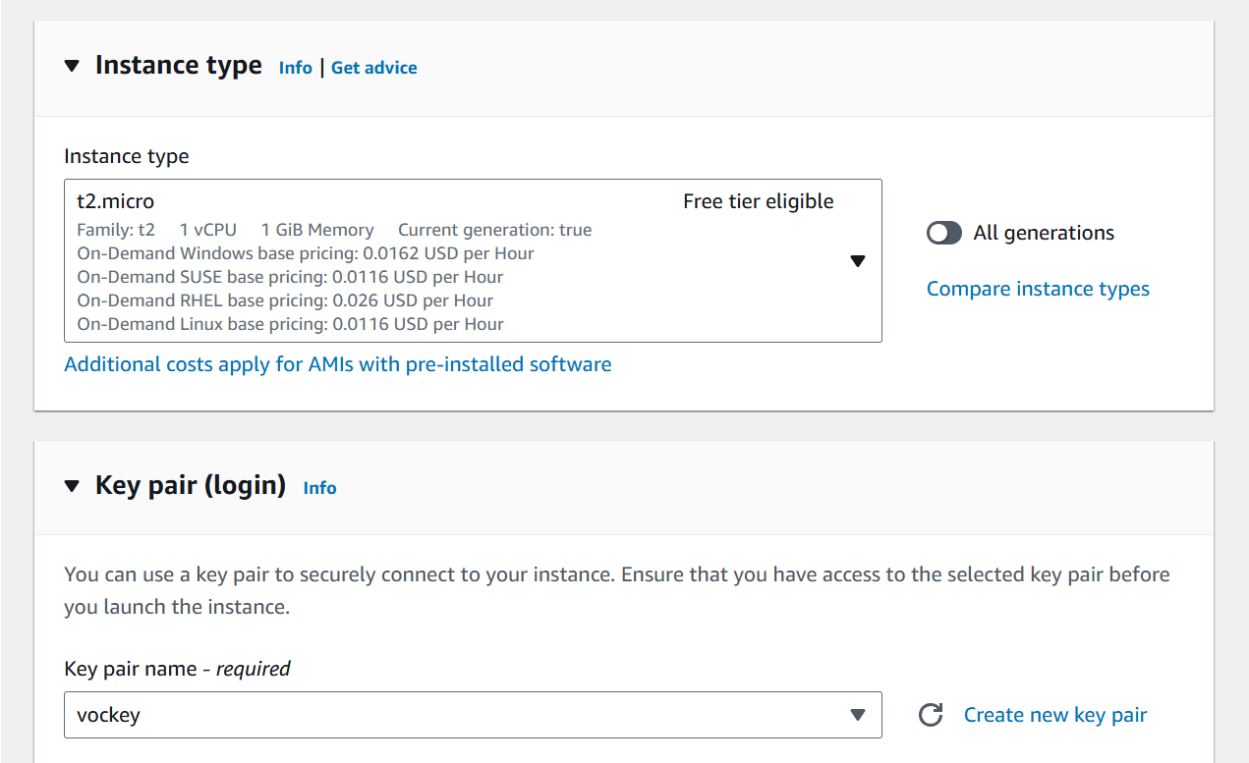
2. Launch instance



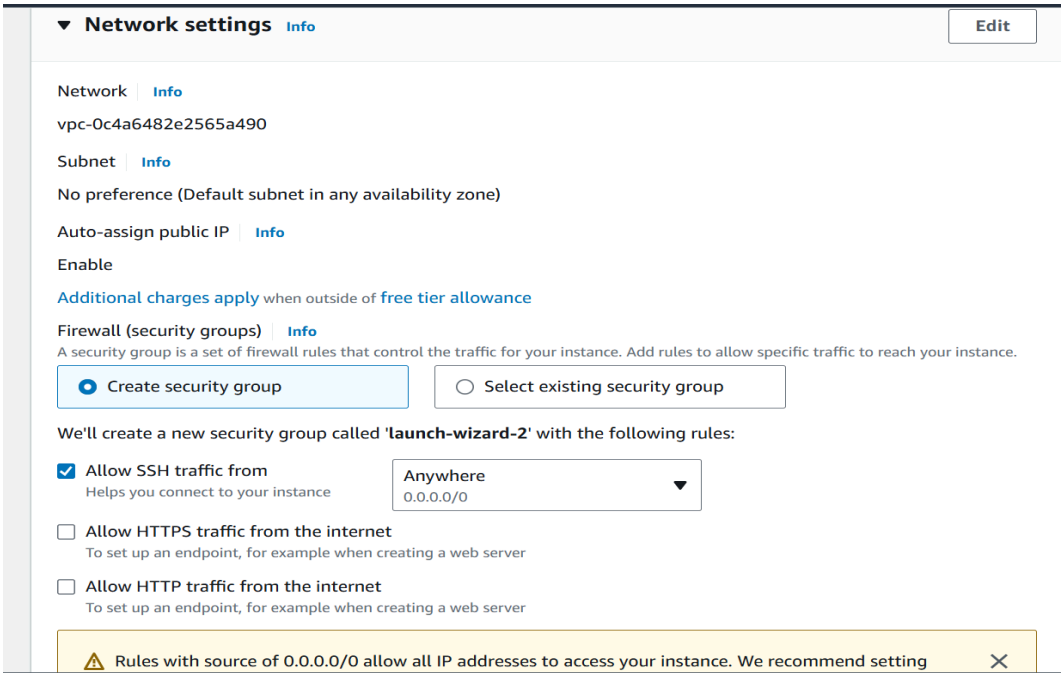
3. Choose ubuntu



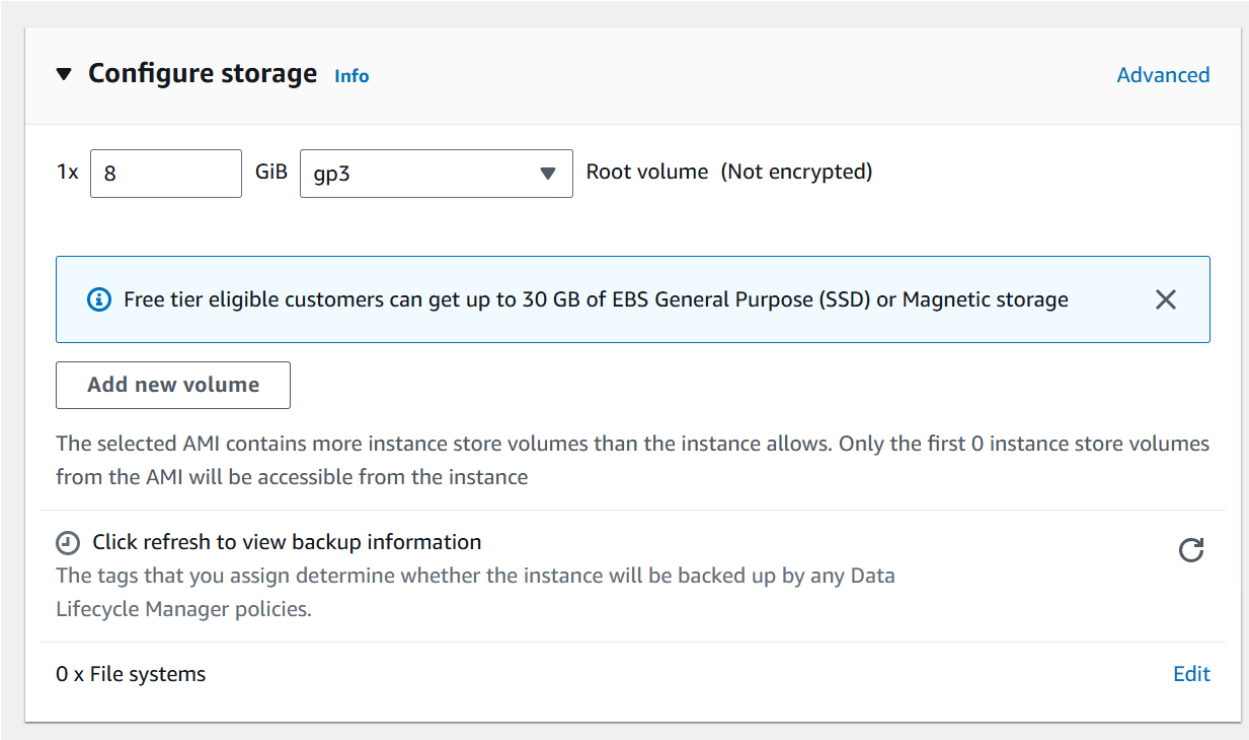
4. Choose instance and key



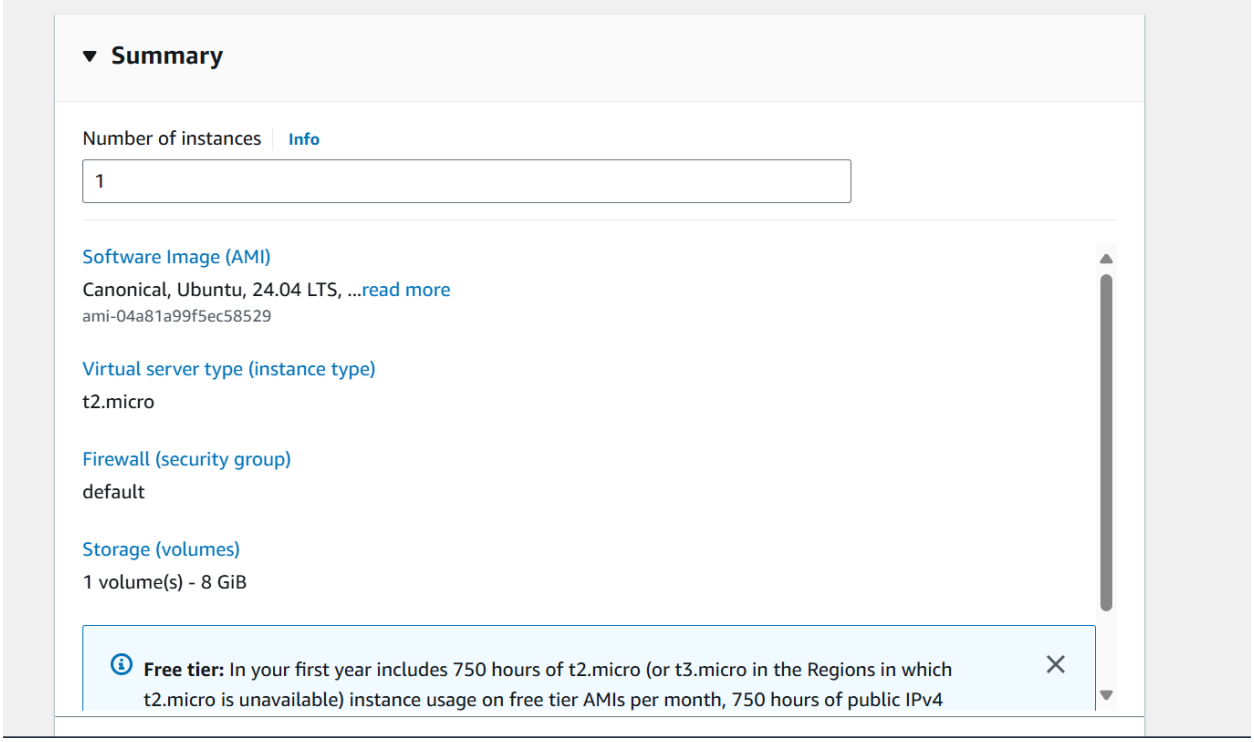
5. Configure network settings



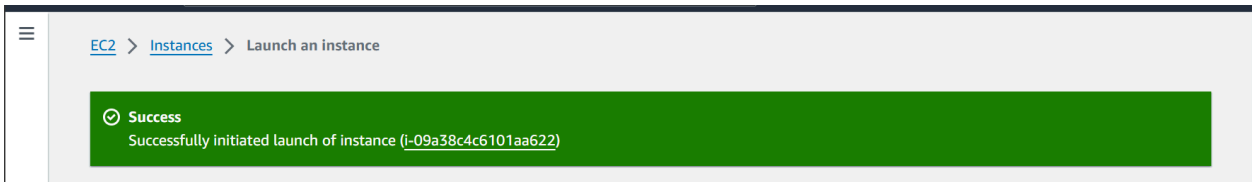
6. Configure storage settings



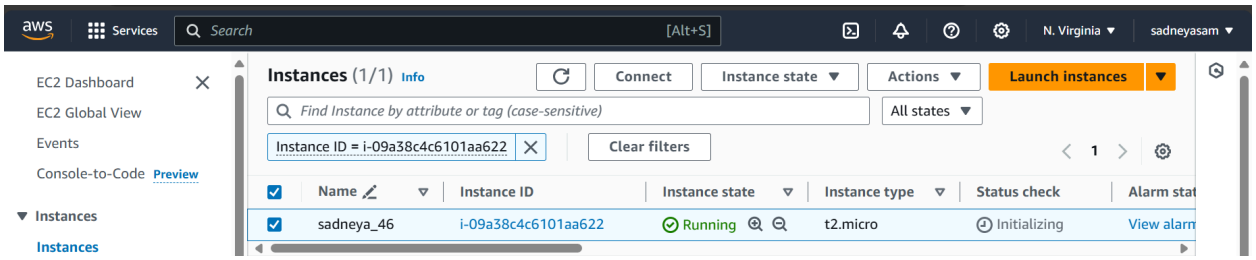
7. This is the instance summary



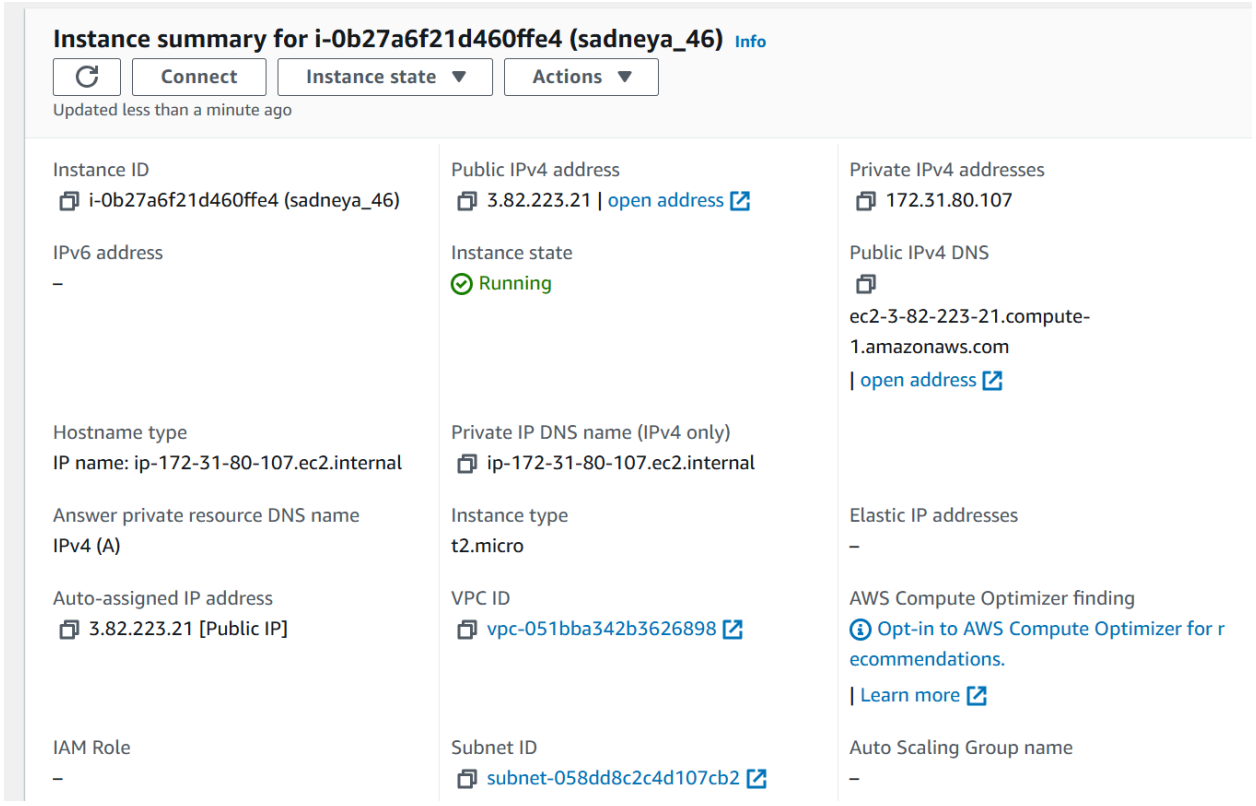
8. The process was successful



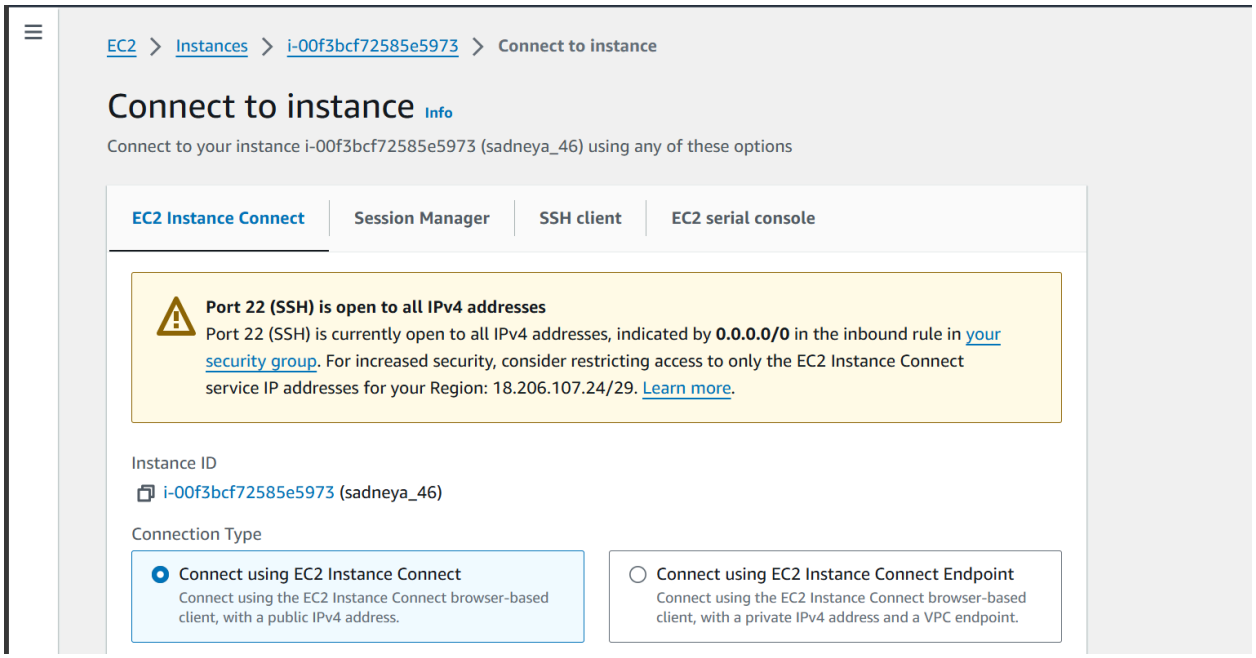
9. Select instance



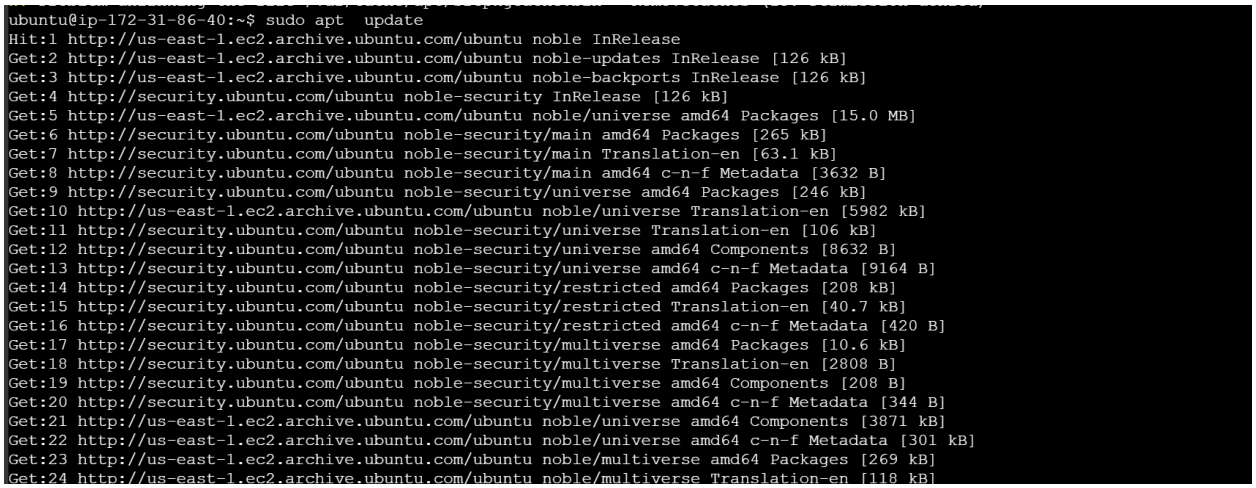
10. Instance summary



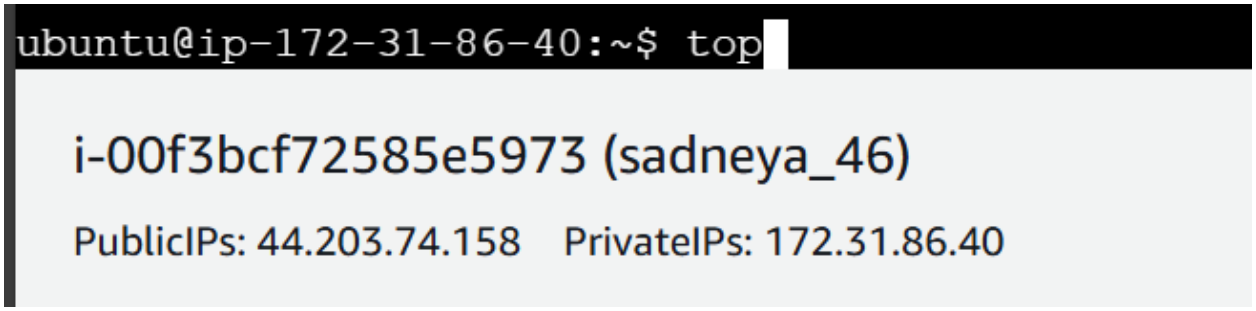
11. Connect to instance



12. Run sudo apt update in the ssh terminal



13. Run top



```
top - 04:37:13 up 10 min,  1 user,  load average: 0.17, 0.09, 0.07
Tasks: 105 total,   1 running, 104 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    957.4 total,    227.6 free,    354.7 used,    530.5 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    602.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0   22520  13536   9568 S   0.0   1.4   0:03.80 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
```

14. Run history

```
ubuntu@ip-172-31-86-40:~$ history
    1  sudo -l
    2  apt update
    3  sudo apt  update
    4  top
    5  history
```

15. run vmstat

```
ubuntu@ip-172-31-86-40:~$ vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- -------cpu-------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st gu
 2  0      0 233040  18712 524596    0    0   379   806  189    1  3  1 94  1  1  0
```

16. Run df

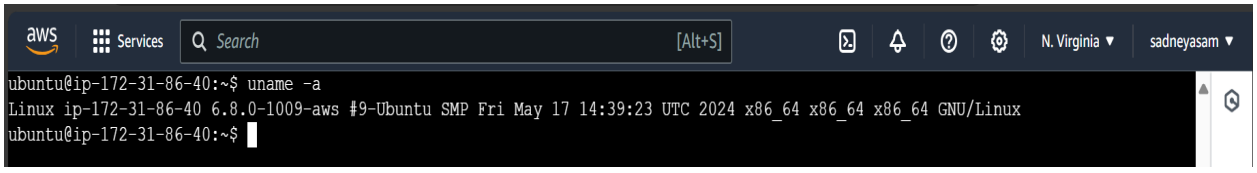
```
ubuntu@ip-172-31-86-40:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root        7034376 1814472   5203520  26% /
tmpfs             490208       0    490208   0% /dev/shm
tmpfs             196084     872    195212   1% /run
tmpfs               5120       0      5120   0% /run/lock
/dev/xvda16       901520   76972    761420  10% /boot
/dev/xvda15       106832    6246    100586   6% /boot/efi
tmpfs              98040      12     98028   1% /run/user/1000
```

17. Run df -kh, whatis df, df –help, uname -a

```
ubuntu@ip-172-31-86-40:~$ df -kh
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.8G  5.0G  26% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  872K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
```

```
ubuntu@ip-172-31-86-40:~$ whatis df
df (1)               - report file system space usage
```

```
ubuntu@ip-172-31-86-40:~$ df --help
Usage: df [OPTION]... [FILE]...
Show information about the file system on which each FILE resides,
or all file systems by default.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all             include pseudo, duplicate, inaccessible file systems
  -B, --block-size=SIZE  scale sizes by SIZE before printing them; e.g.,
                           '-BM' prints sizes in units of 1,048,576 bytes;
                           see SIZE format below
  -h, --human-readable  print sizes in powers of 1024 (e.g., 1023M)
  -H, --si              print sizes in powers of 1000 (e.g., 1.1G)
  -i, --inodes          list inode information instead of block usage
  -k                    like --block-size=1K
  -l, --local           limit listing to local file systems
      --no-sync         do not invoke sync before getting usage info (default)
      --output[=FIELD_LIST]  use the output format defined by FIELD_LIST,
                               or print all fields if FIELD_LIST is omitted.
  -P, --portability     use the POSIX output format
      --sync            invoke sync before getting usage info
      --total           elide all entries insignificant to available space,
                          and produce a grand total
  -t, --type=TYPE       limit listing to file systems of type TYPE
  -T, --print-type      print file system type
  -x, --exclude-type=TYPE   limit listing to file systems not of type TYPE
  -v                    (ignored)
```

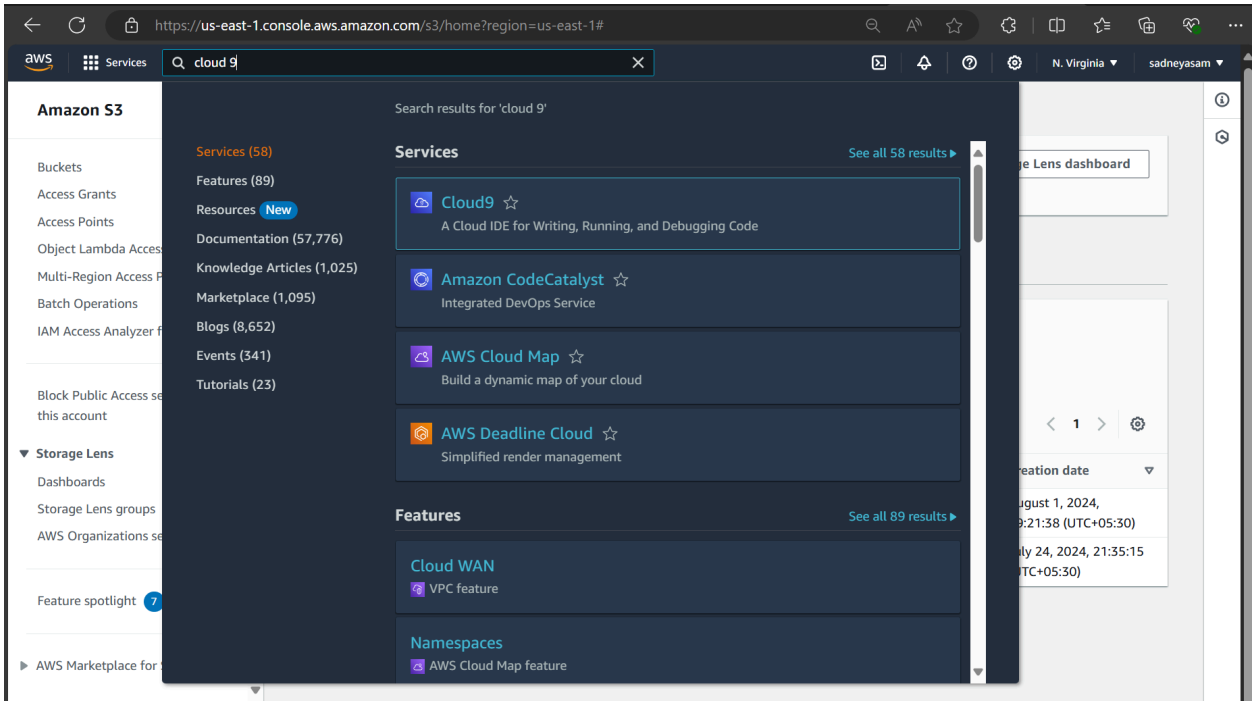


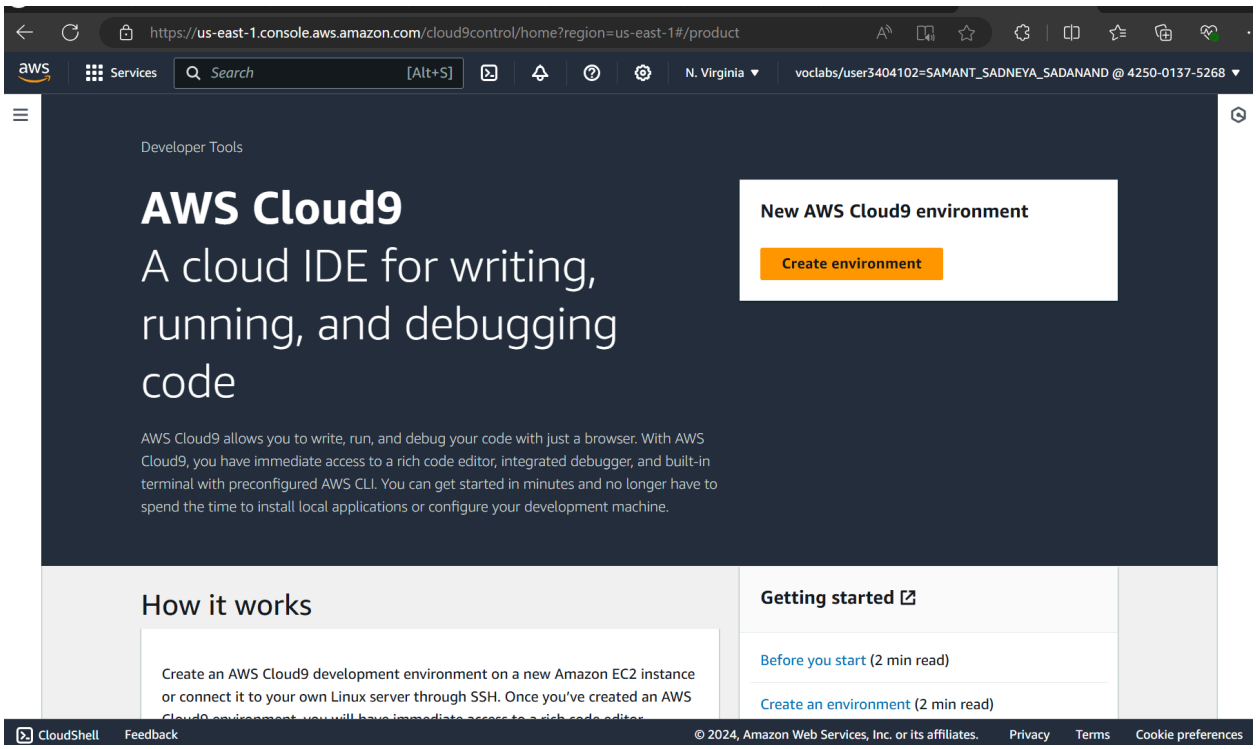
18. Create and do file operations

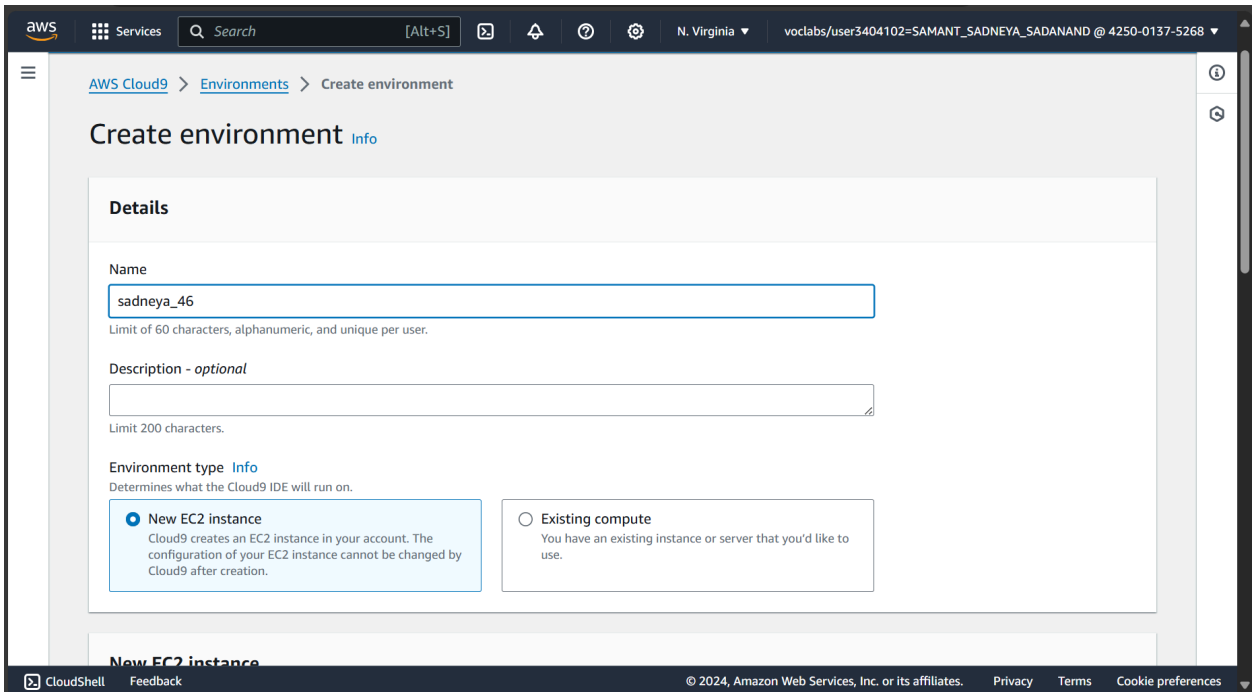
```
ubuntu@ip-172-31-80-107:~$ mkdir test
ubuntu@ip-172-31-80-107:~$ ls
test
ubuntu@ip-172-31-80-107:~$ cd test
ubuntu@ip-172-31-80-107:~/test$ touch file1
ubuntu@ip-172-31-80-107:~/test$ ls
file1
ubuntu@ip-172-31-80-107:~/test$ touch file2 file3
ubuntu@ip-172-31-80-107:~/test$ ls
file1  file2  file3
ubuntu@ip-172-31-80-107:~/test$ rm file*
ubuntu@ip-172-31-80-107:~/test$ ls
ubuntu@ip-172-31-80-107:~/test$ cd ..
ubuntu@ip-172-31-80-107:~$ rmdir test
ubuntu@ip-172-31-80-107:~$ cd ..
ubuntu@ip-172-31-80-107:/home$ ls
ubuntu
ubuntu@ip-172-31-80-107:/home$ cd ubuntu
ubuntu@ip-172-31-80-107:~$ mkdir test1 test2 test3
ubuntu@ip-172-31-80-107:~$ ls
test1  test2  test3
```

EXPERIMENT 1B

Abcd







New EC2 instance

Instance type Info

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

- t2.micro (1 GiB RAM + 1 vCPU) — Free-tier eligible. Ideal for educational users and exploration.
- t3.small (2 GiB RAM + 2 vCPU) — Recommended for small web projects.
- m5.large (8 GiB RAM + 2 vCPU) — Recommended for production and most general-purpose development.
- Additional instance types — Explore additional instances to fit your need.

Platform Info

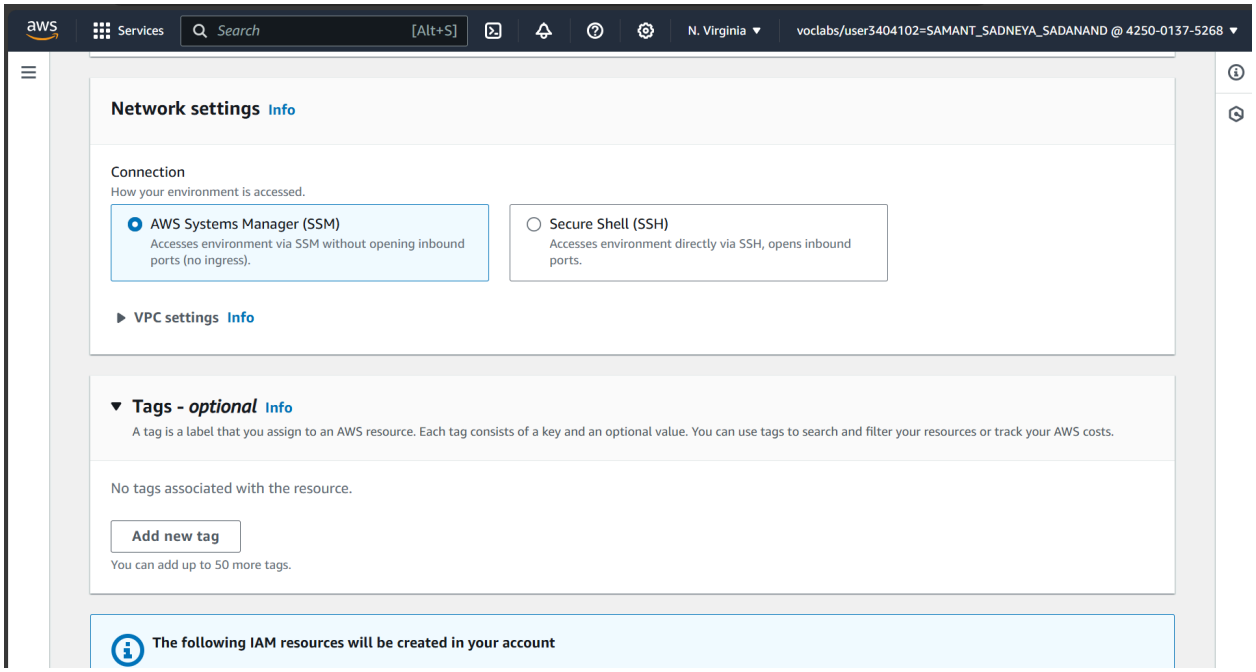
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

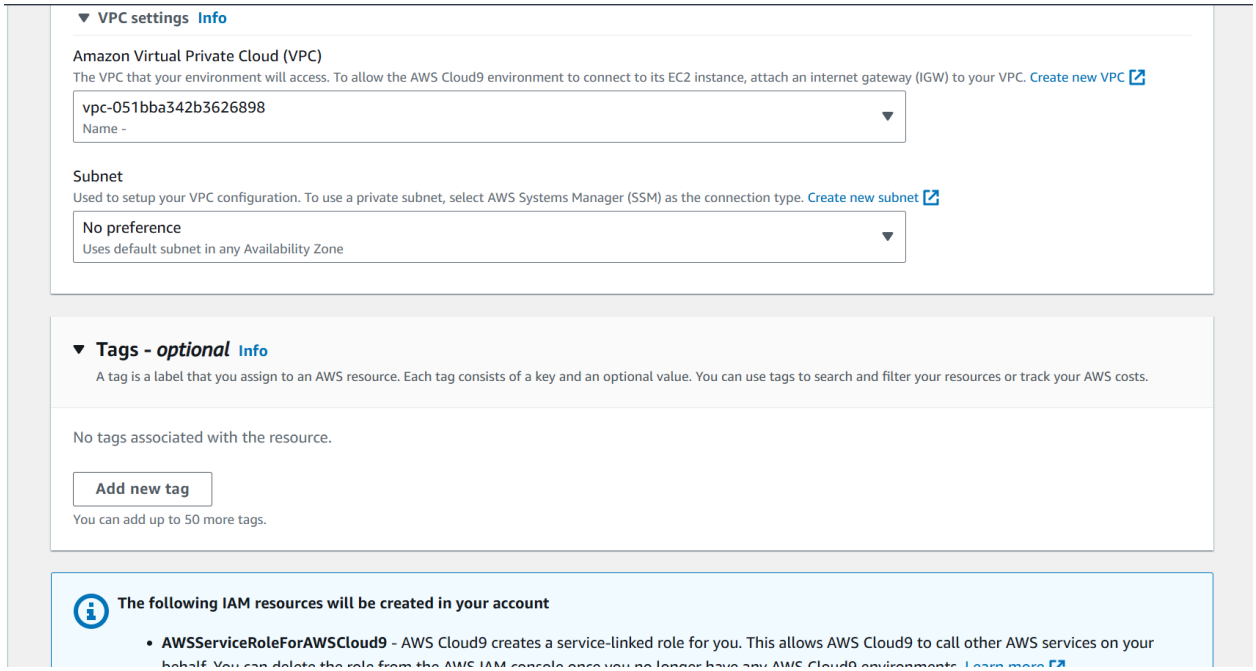
Amazon Linux 2023

Timeout

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
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.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
The following IAM resources will be created in your account


VPC settings Info
Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. Create new VPC
vpc-051bba342b3626898
Name -
Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. Create new subnet
No preference
Uses default subnet in any Availability Zone
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.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
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

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

Amazon Virtual Private Cloud (VPC)

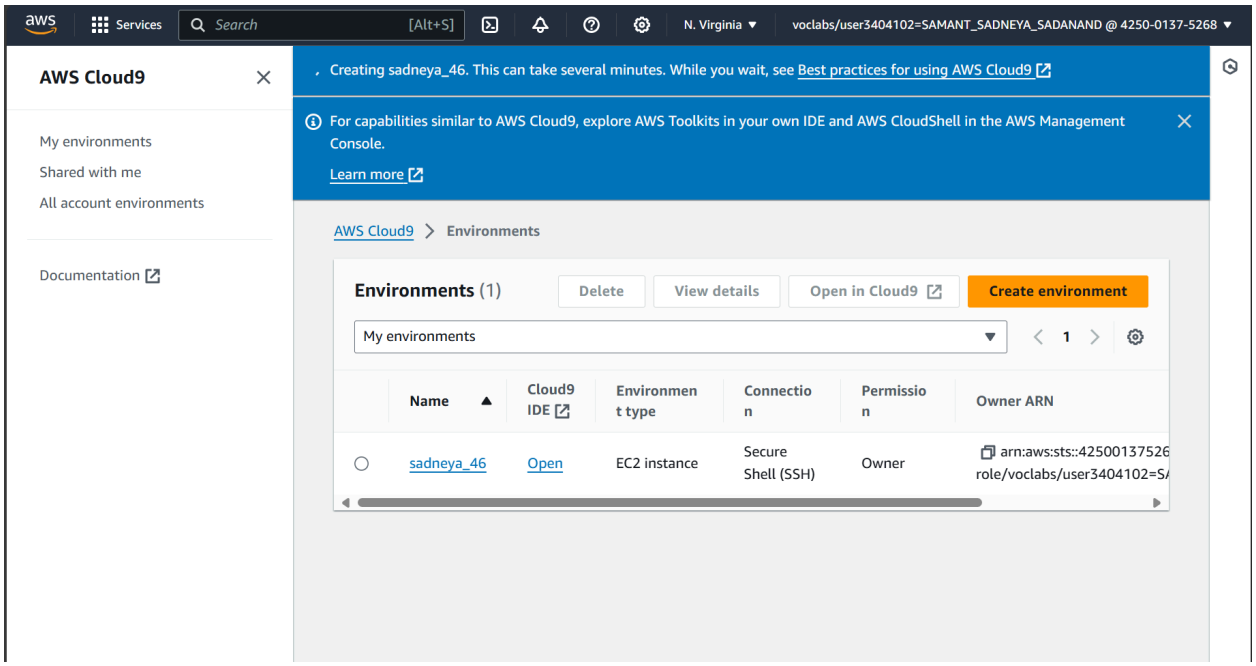
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. Create new VPC

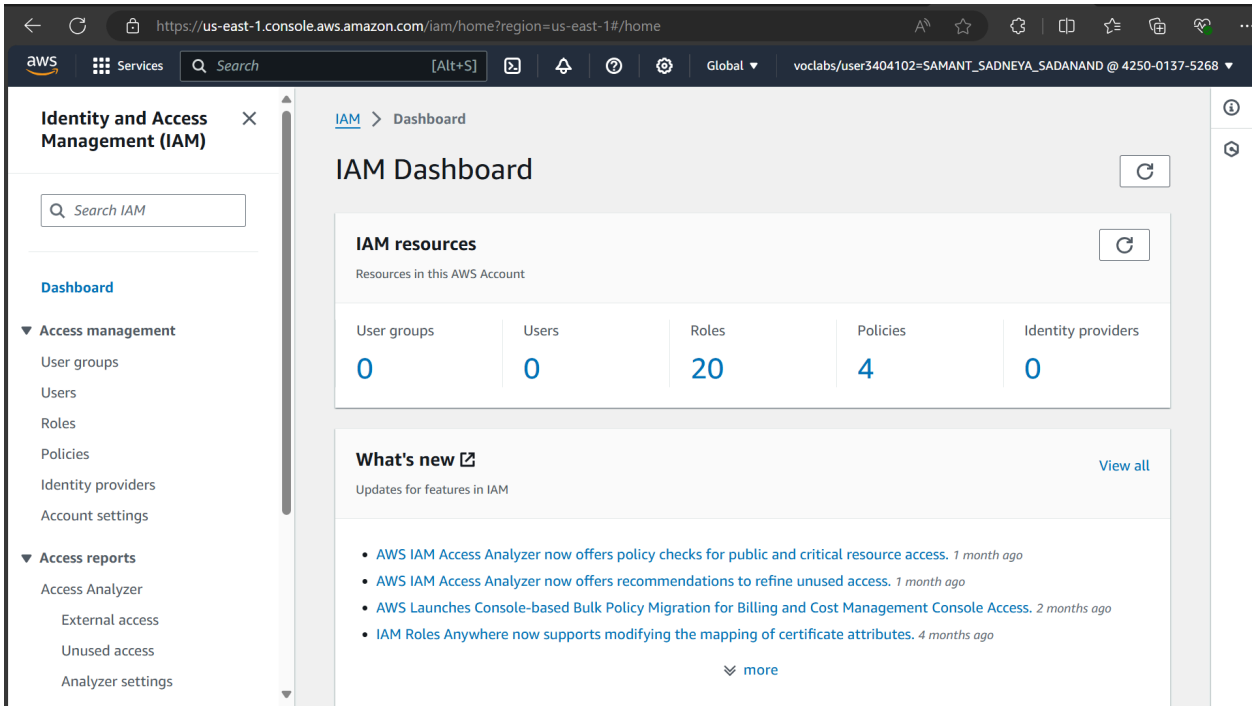
vpc-051bba342b3626898
Name -

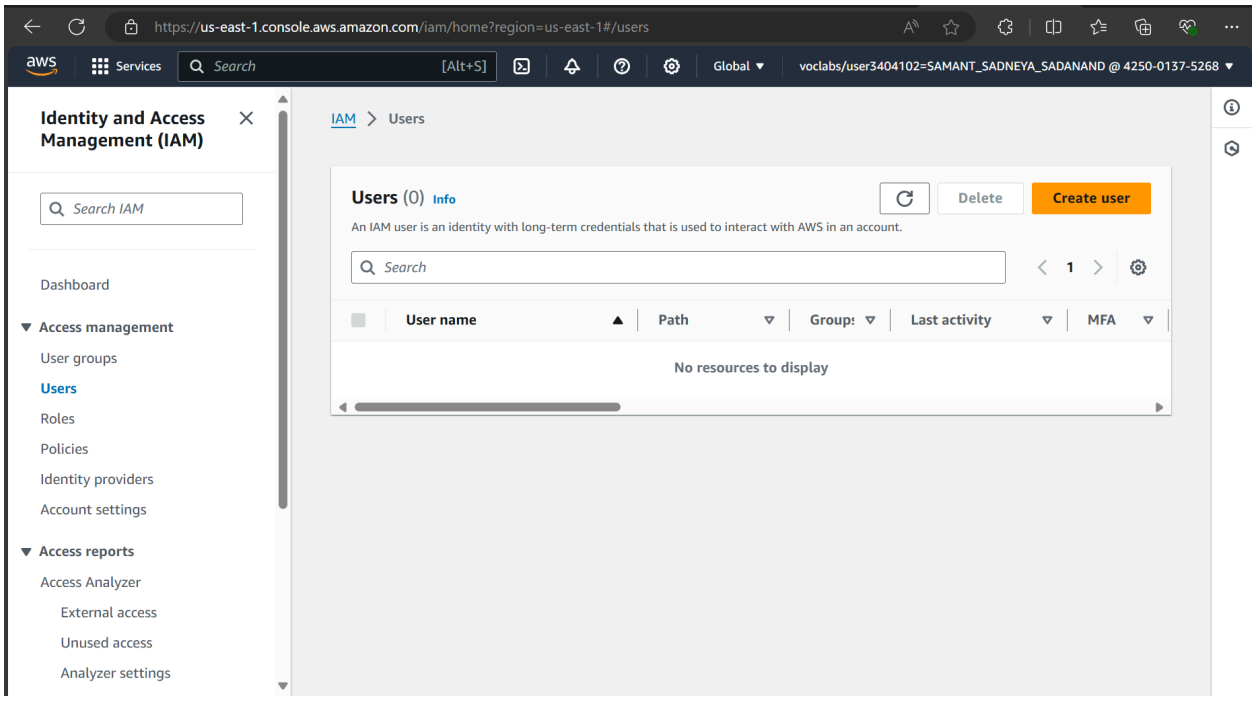
Subnet

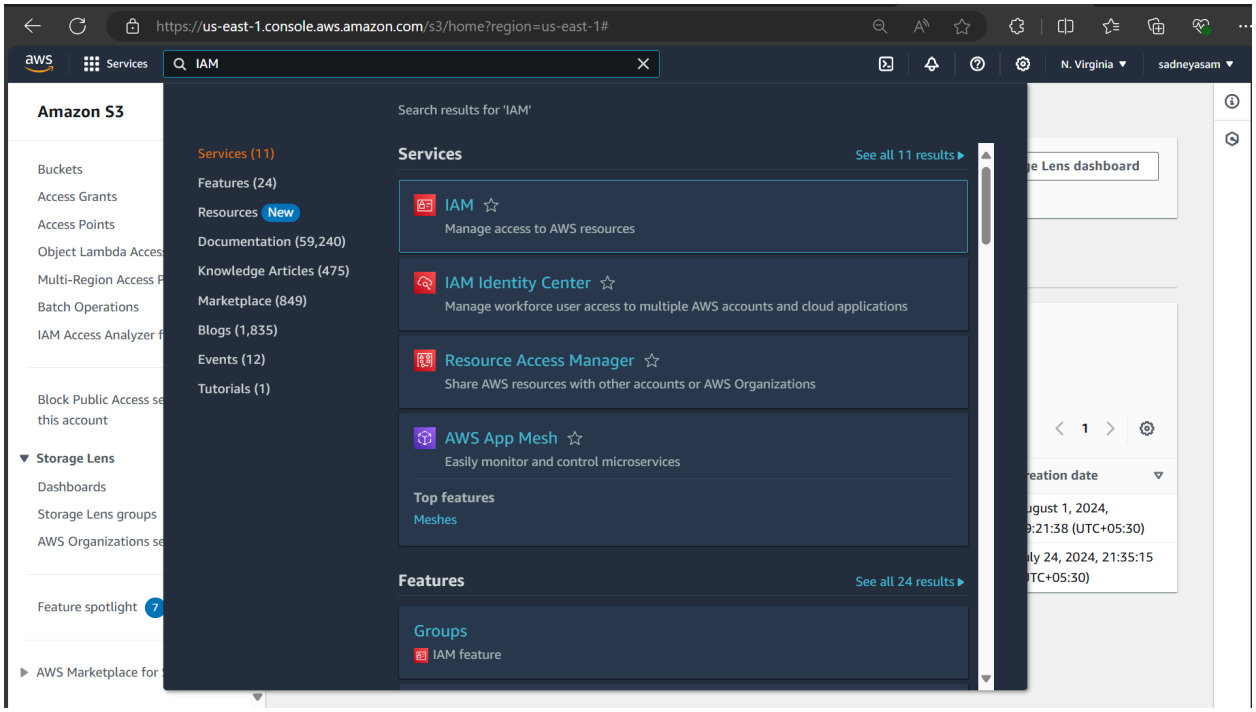
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. Create new subnet

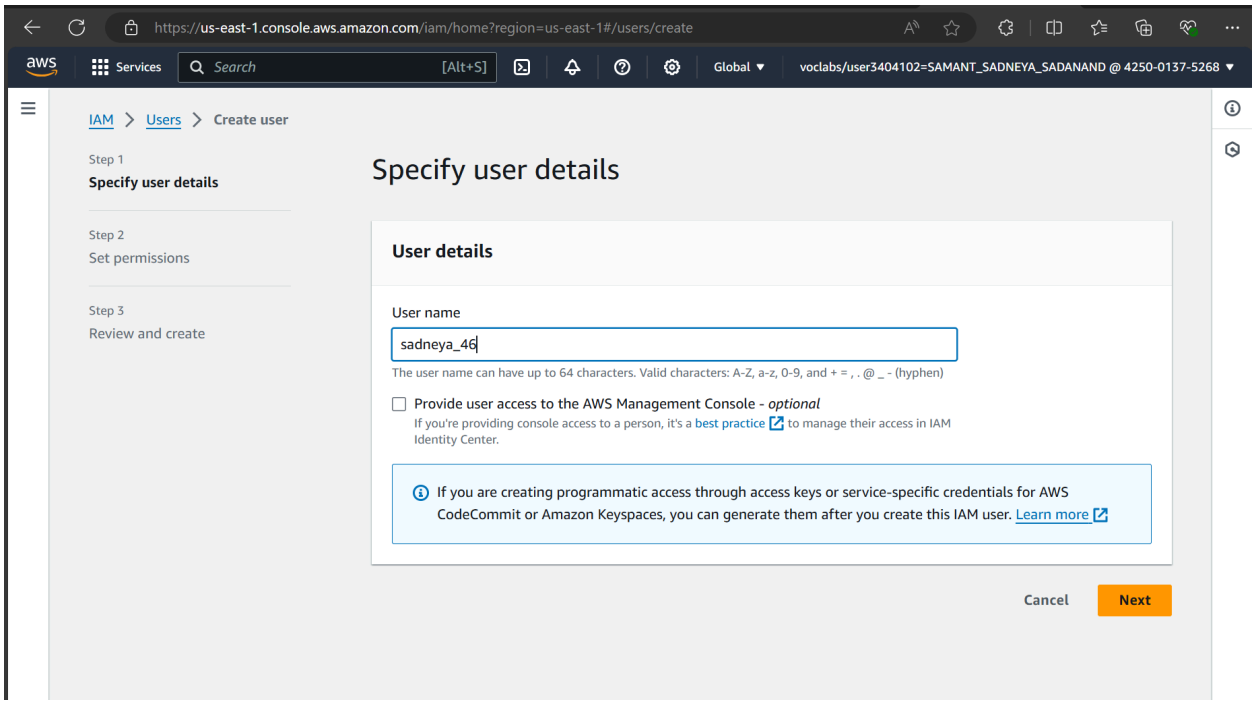
No preference
Uses default subnet in any Availability Zone

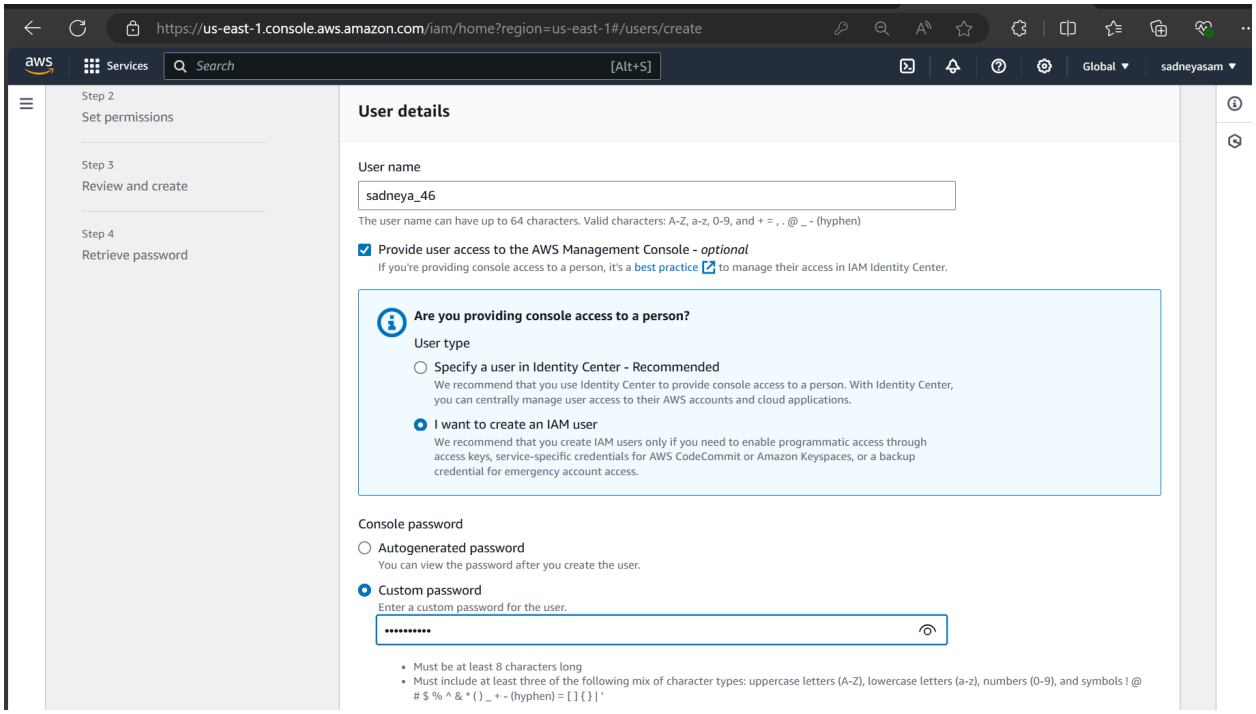


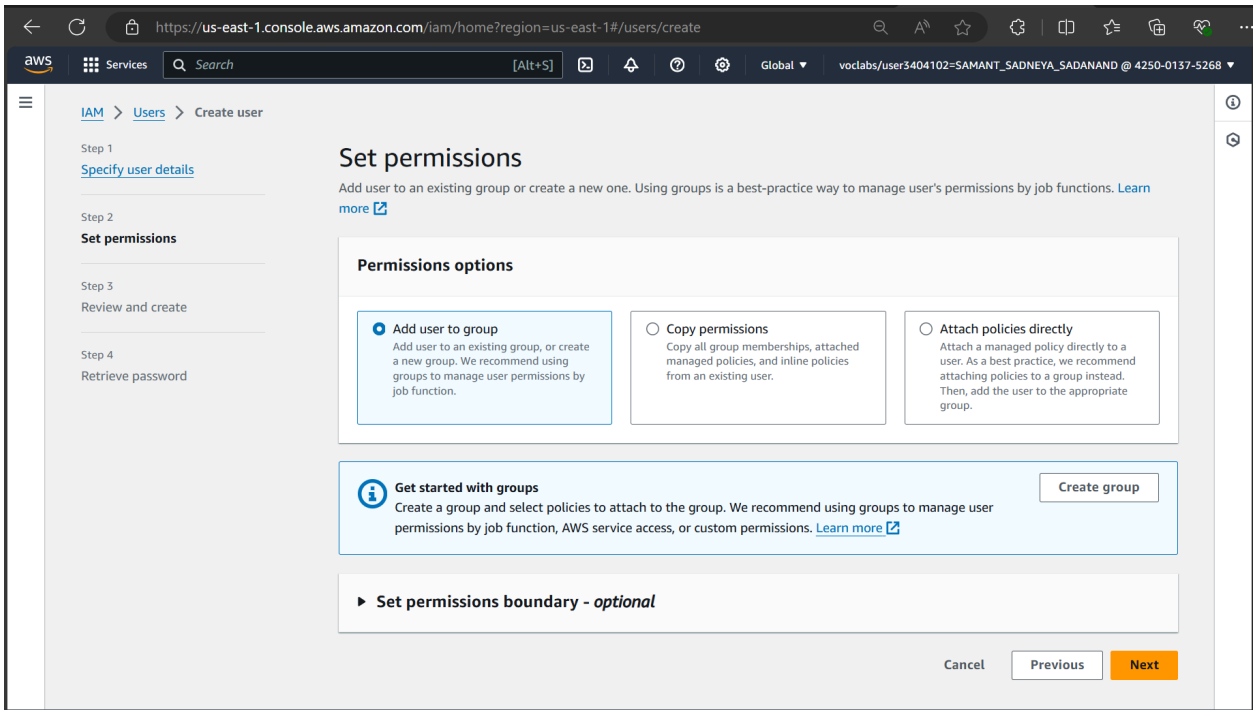

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/home
aws
Services
Search
[Alt+S]
Global
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
IAM > Dashboard
IAM Dashboard
IAM resources
Resources in this AWS Account
User groups
0
Users
0
Roles
20
Policies
4
Identity providers
0
What's new
Updates for features in IAM
View all
AWS IAM Access Analyzer now offers policy checks for public and critical resource access. 1 month ago
AWS IAM Access Analyzer now offers recommendations to refine unused access. 1 month ago
AWS Launches Console-based Bulk Policy Migration for Billing and Cost Management Console Access. 2 months ago
IAM Roles Anywhere now supports modifying the mapping of certificate attributes. 4 months ago
more

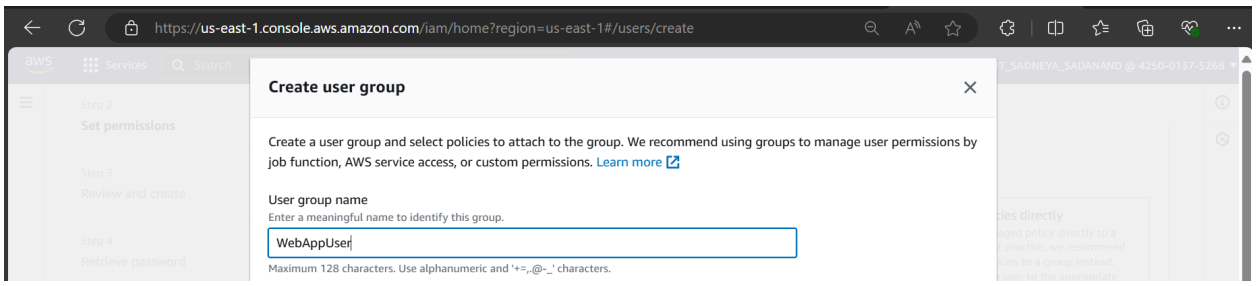

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users
aws
Services
Search
[Alt+S]
Global
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
IAM > Users
Users (0) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Group:
Last activity
MFA
No resources to display

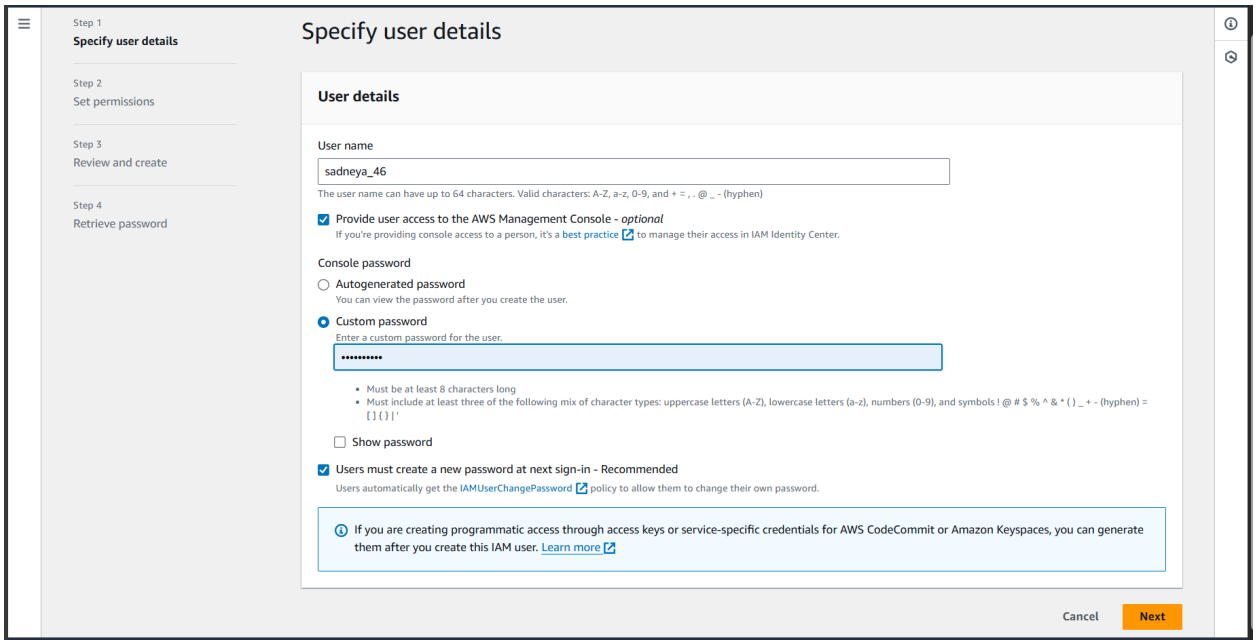

https://us-east-1.console.aws.amazon.com/s3/home?region=us-east-1#
Services
IAM
N. Virginia
sadneyasam
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Acces
Multi-Region Access P
Batch Operations
IAM Access Analyzer f
Block Public Access se
this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations se
Feature spotlight
AWS Marketplace for
Search results for 'IAM'
Services (11)
Features (24)
Resources New
Documentation (59,240)
Knowledge Articles (475)
Marketplace (849)
Blogs (1,835)
Events (12)
Tutorials (1)
Services
See all 11 results
IAM
Manage access to AWS resources
IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications
Resource Access Manager
Share AWS resources with other accounts or AWS Organizations
AWS App Mesh
Easily monitor and control microservices
Top features
Meshes
Features
See all 24 results
Groups
IAM feature
e Lens dashboard
1
reation date
ugust 1, 2024,
:21:38 (UTC+05:30)
ly 24, 2024, 21:35:15
TC+05:30)

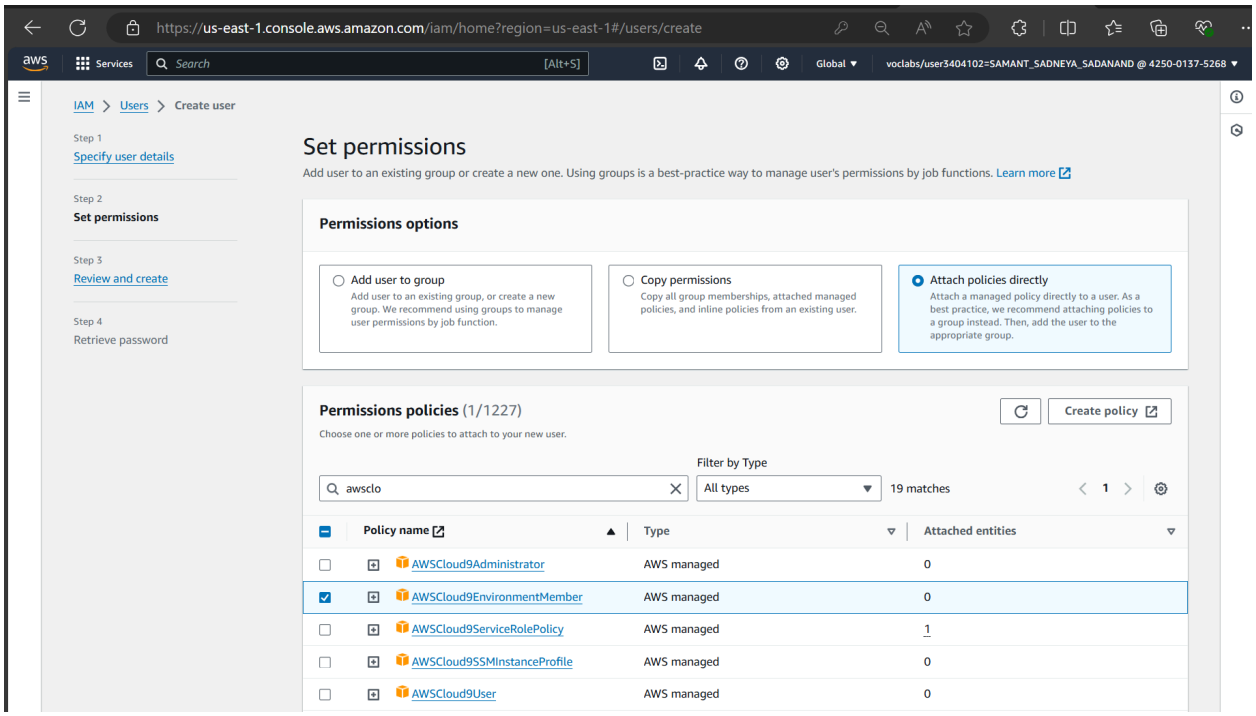

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Specify user details
User details
User name
sadneya_46
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next

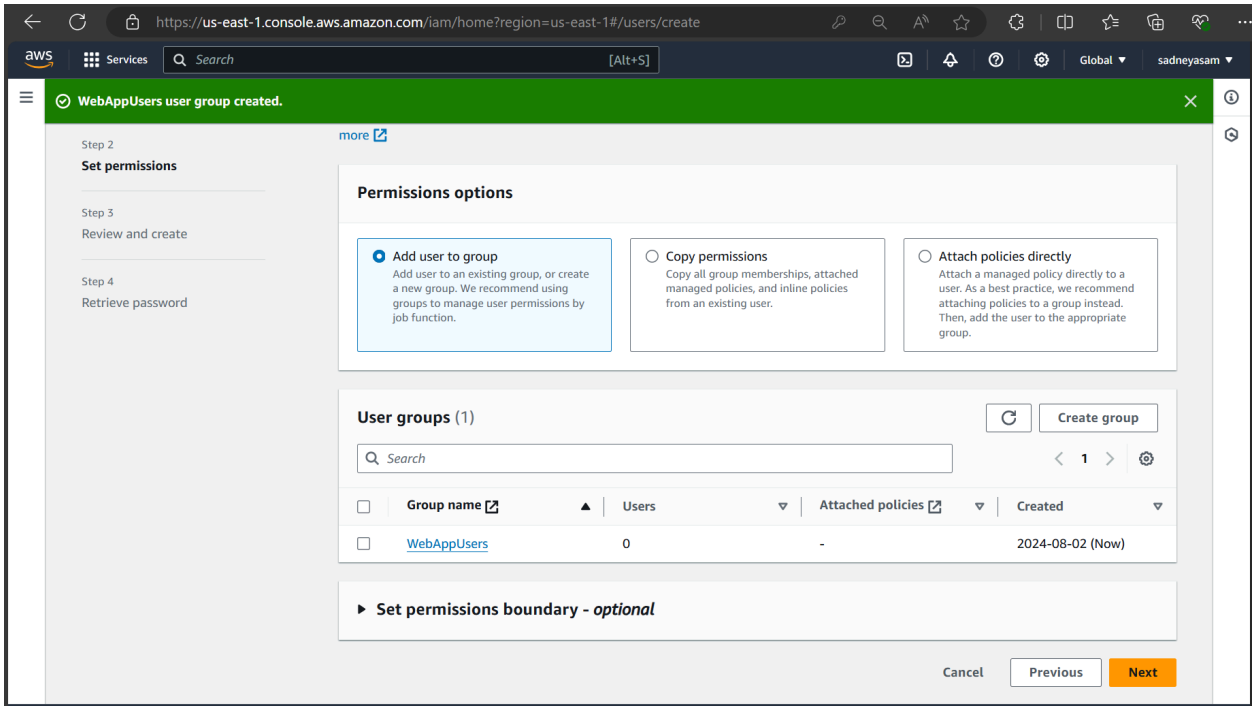

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
sadneyasam
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
User details
User name
sadneya_46
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
Are you providing console access to a person?
User type
Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '

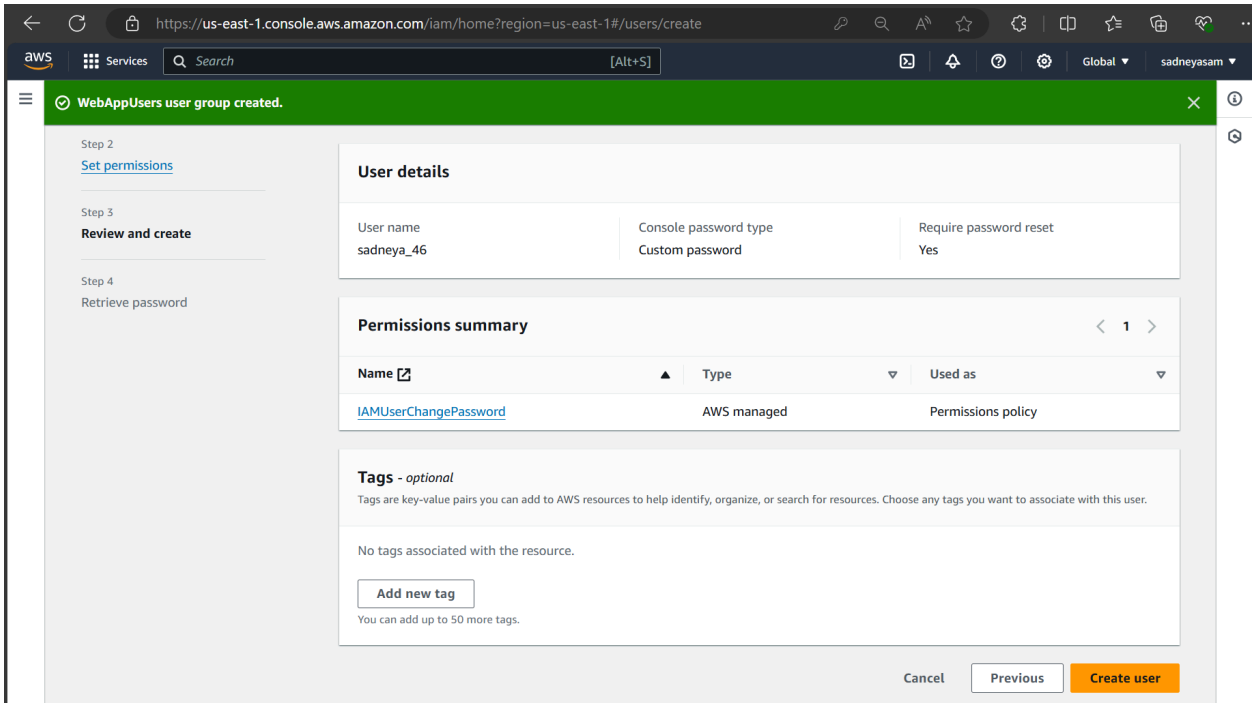

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
IAM
Users
Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more
Create group
Set permissions boundary - optional
Cancel
Previous
Next

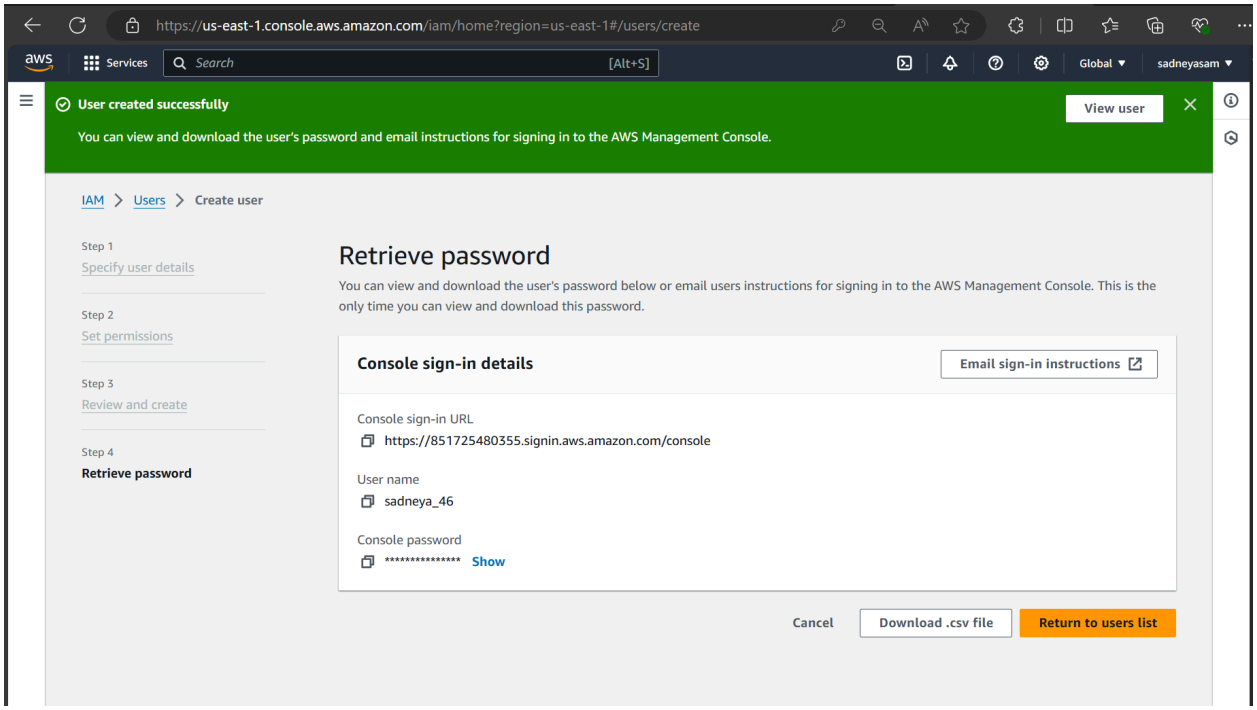

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Create user group
Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more
User group name
Enter a meaningful name to identify this group.
WebAppUser
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

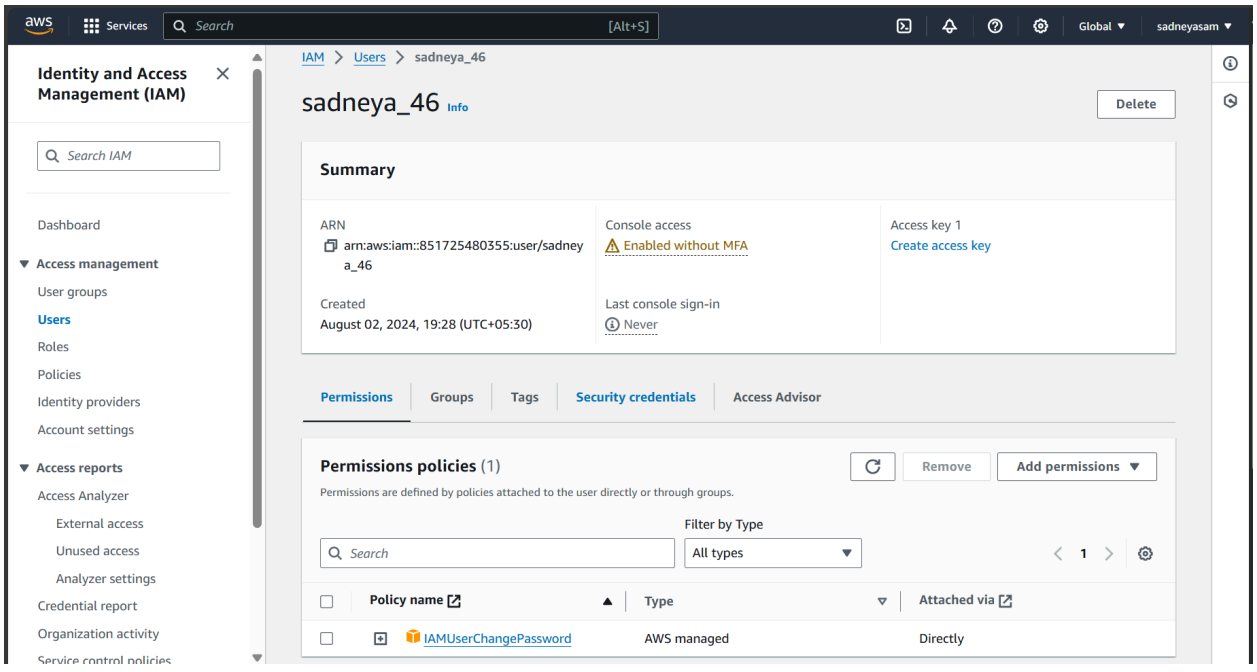

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Specify user details
User details
User name
sadneya_46
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '
Show password
Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next

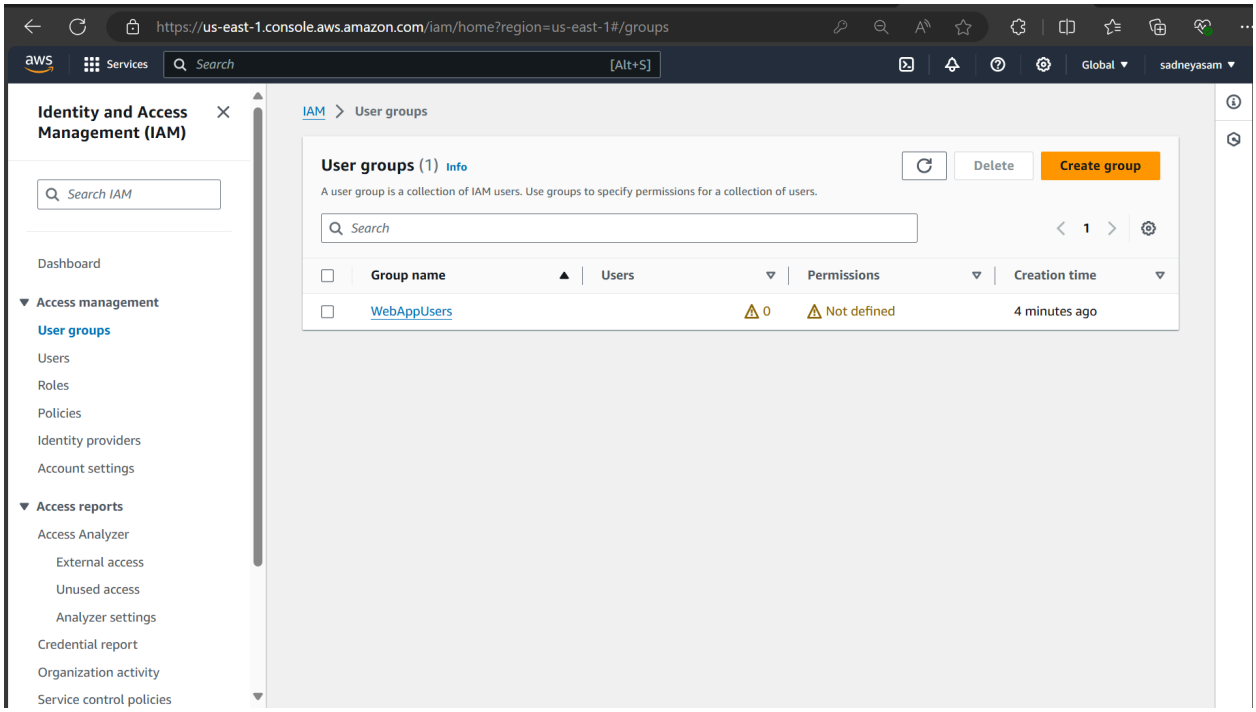

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268
IAM
Users
Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (1/1227)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
awsclo
All types
19 matches
1
Policy name
Type
Attached entities
AWSCloud9Administrator
AWS managed
0
AWSCloud9EnvironmentMember
AWS managed
0
AWSCloud9ServiceRolePolicy
AWS managed
1
AWSCloud9SSMInstanceProfile
AWS managed
0
AWSCloud9User
AWS managed
0

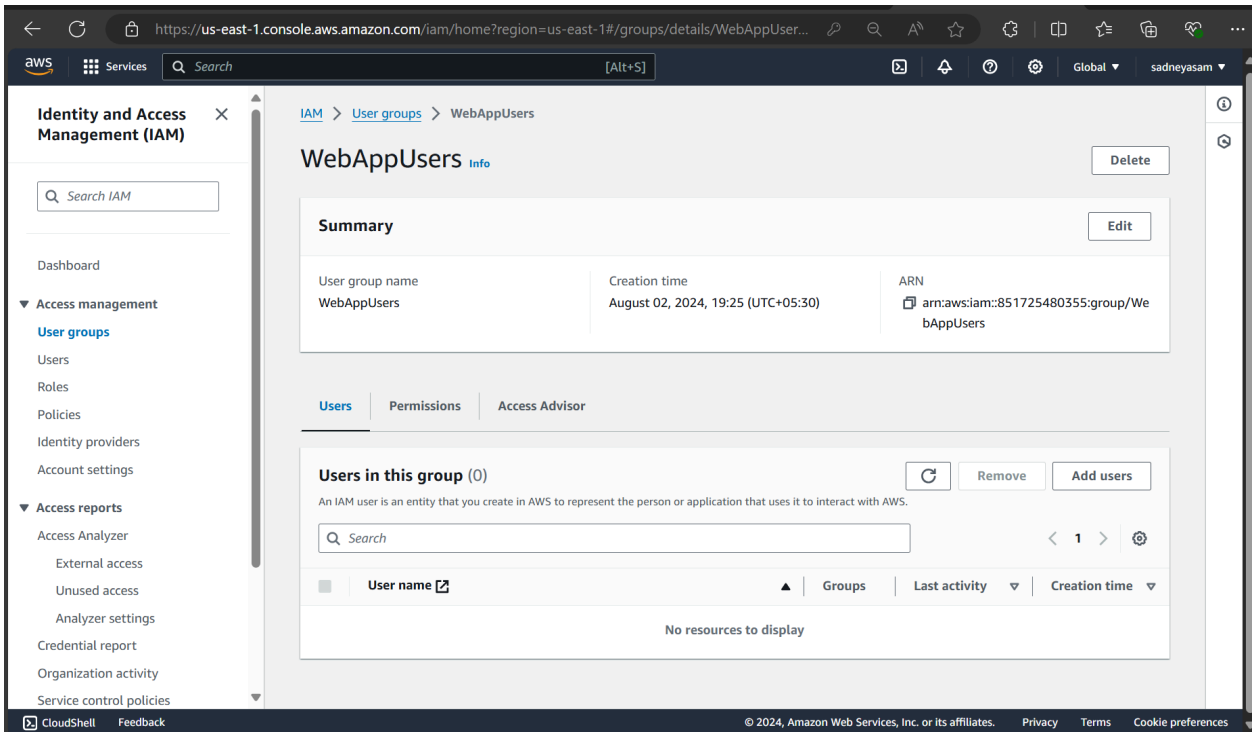

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
sadneyasam
WebAppUsers user group created.
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
User groups (1)
Create group
Search
1
Group name
Users
Attached policies
Created
WebAppUsers
0
-
2024-08-02 (Now)
Set permissions boundary - optional
Cancel
Previous
Next


https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
sadneyasam
WebAppUsers user group created.
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
User details
User name
sadneya_46
Console password type
Custom password
Require password reset
Yes
Permissions summary
1
Name
Type
Used as
IAMUserChangePassword
AWS managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user


https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create
Services
Search
[Alt+S]
Global
sadneyasam
User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.
Console sign-in details
Email sign-in instructions
Console sign-in URL
https://851725480355.signin.aws.amazon.com/console
User name
sadneya_46
Console password
********** Show
Cancel
Download .csv file
Return to users list

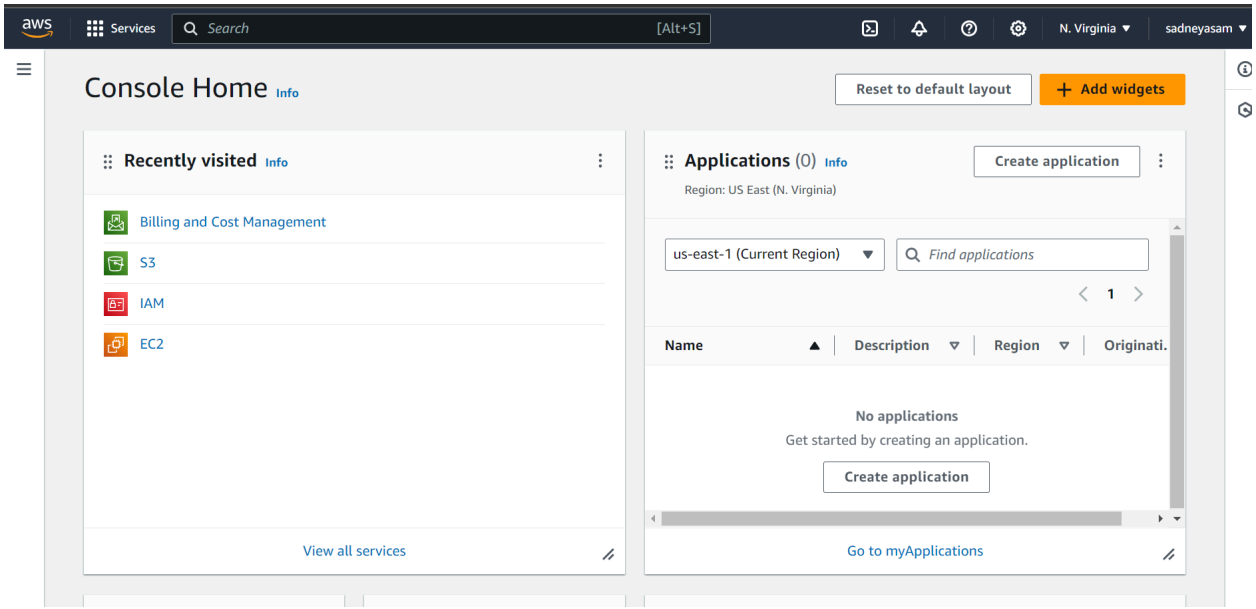

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
IAM > Users > sadneya_46
sadneya_46 Info
Delete
Summary
ARN
arn:aws:iam::851725480355:user/sadneya_46
Console access
Enabled without MFA
Access key 1
Create access key
Created
August 02, 2024, 19:28 (UTC+05:30)
Last console sign-in
Never
Permissions
Groups
Tags
Security credentials
Access Advisor
Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached via
IAMUserChangePassword
AWS managed
Directly


https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/groups
aws
Services
Search
[Alt+S]
Global
sadneyasam
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
IAM > User groups
User groups (1) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.
Delete
Create group
Search
1
Group name
Users
Permissions
Creation time
WebAppUsers
0
Not defined
4 minutes ago

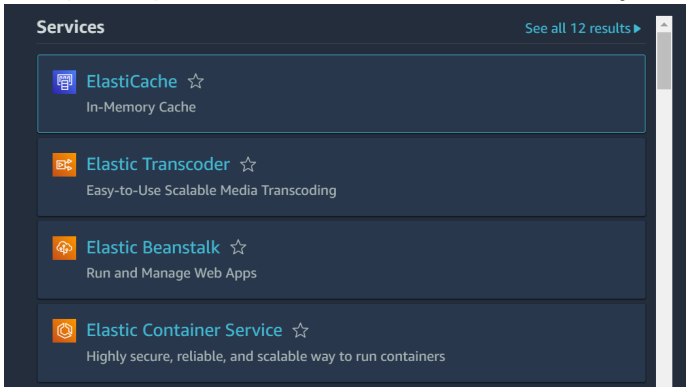

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/groups/details/WebAppUser...
aws
Services
Search
[Alt+S]
Global
sadneyasam
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
IAM > User groups > WebAppUsers
WebAppUsers Info
Delete
Summary
Edit
User group name
WebAppUsers
Creation time
August 02, 2024, 19:25 (UTC+05:30)
ARN
arn:aws:iam::851725480355:group/WebAppUsers
Users
Permissions
Access Advisor
Users in this group (0)
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
Remove
Add users
Search
1
User name
Groups
Last activity
Creation time
No resources to display
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Experiment 2

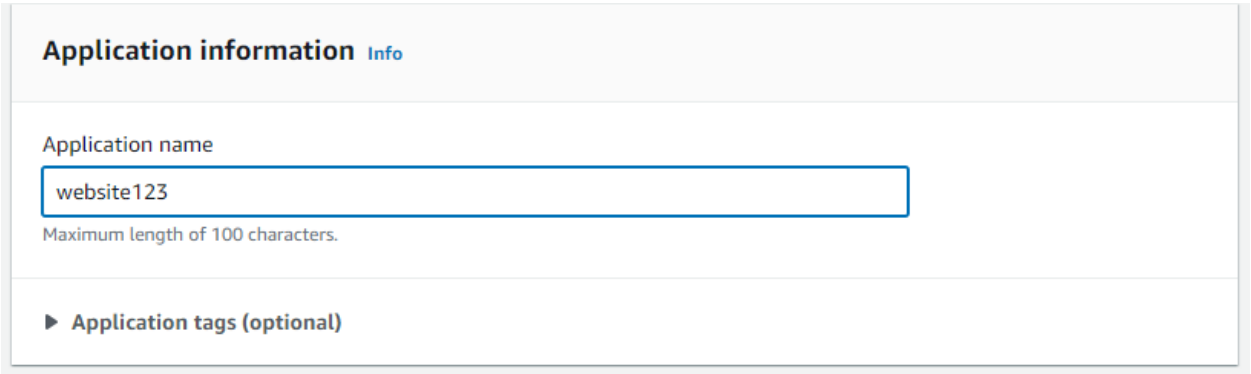
1. Go to the amazon console page

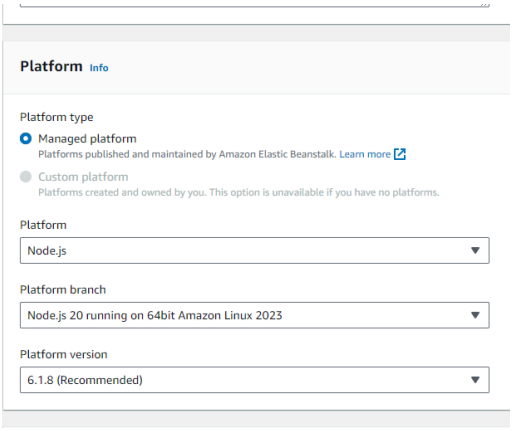


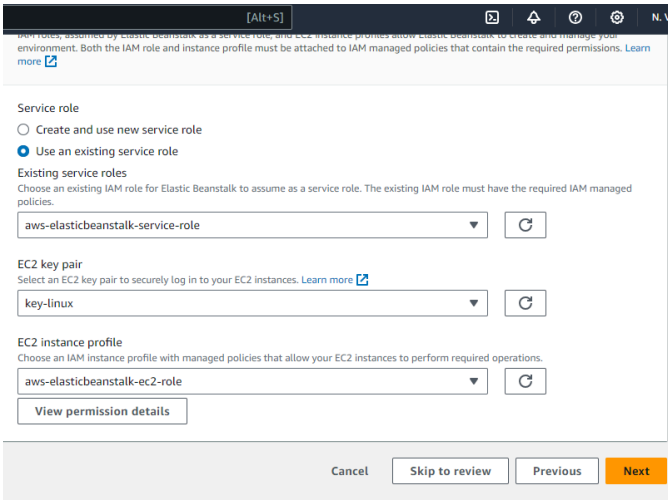
2. Open up Elastic Beanstalk and name your web app.



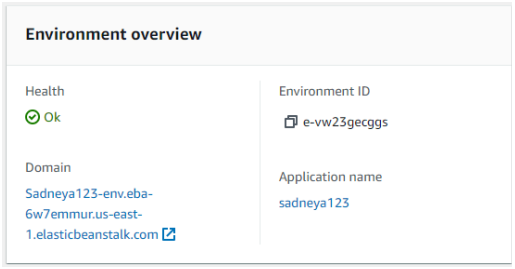
3. Enter your website and its information



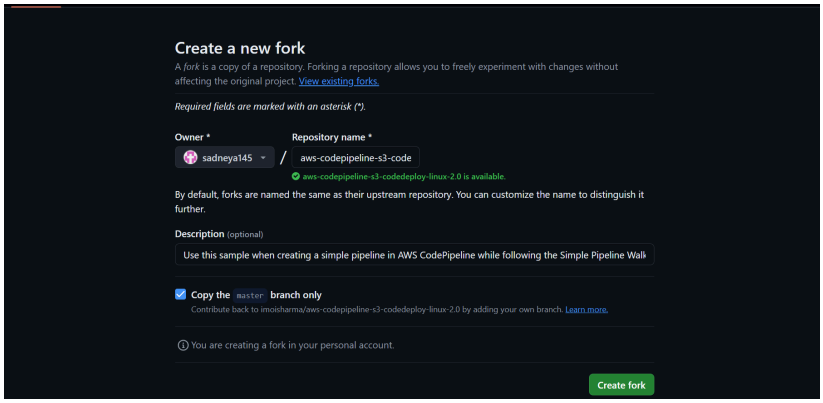




4. Your environment is created



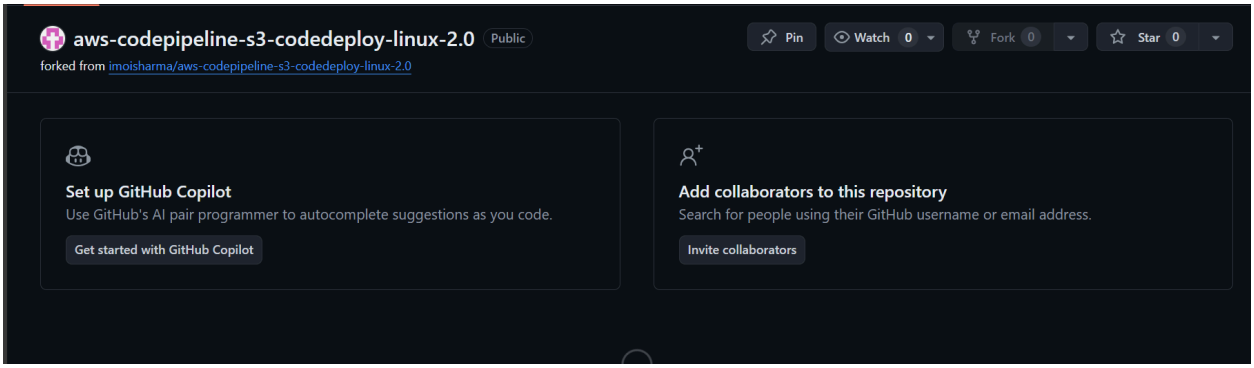
Step 2: Get a copy of your sample code



In this step, we will get the sample code from this GitHub Repository to host it later. The pipeline takes code from the source and then performs actions on it.

We will use this forked GitHub repository as a source for this experiment. We can alternatively also use Amazon S3 and AWS CodeCommit.

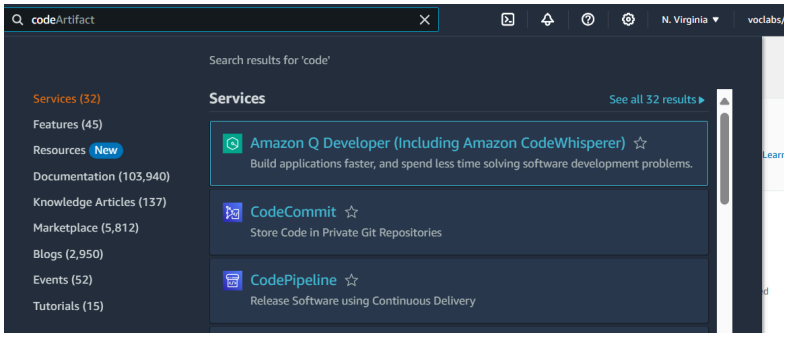
Go to the repository shared above and simply fork it.

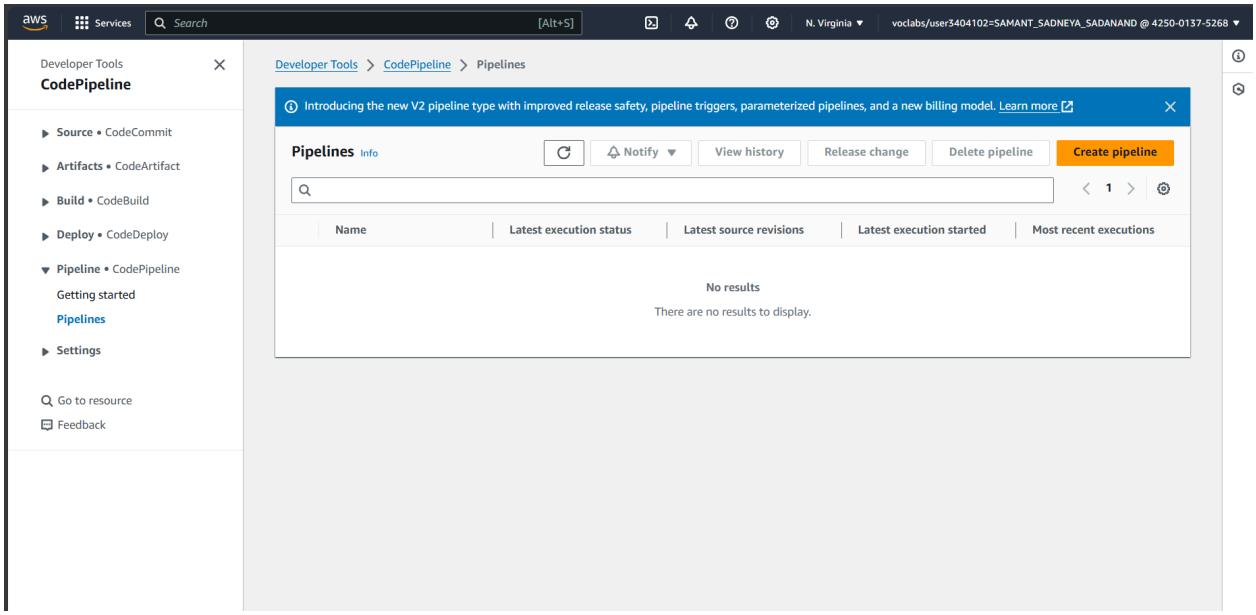


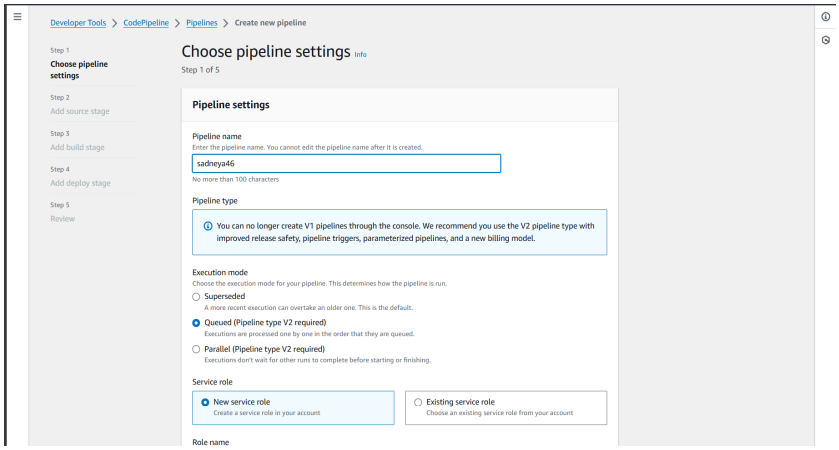
Step 3: Creating a CodePipeline

In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

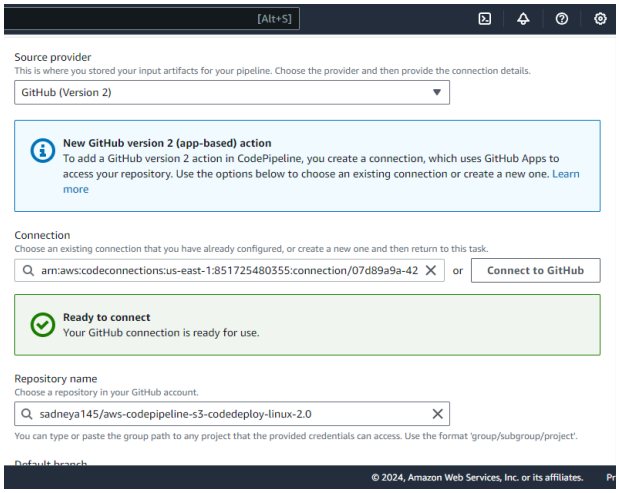
1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.







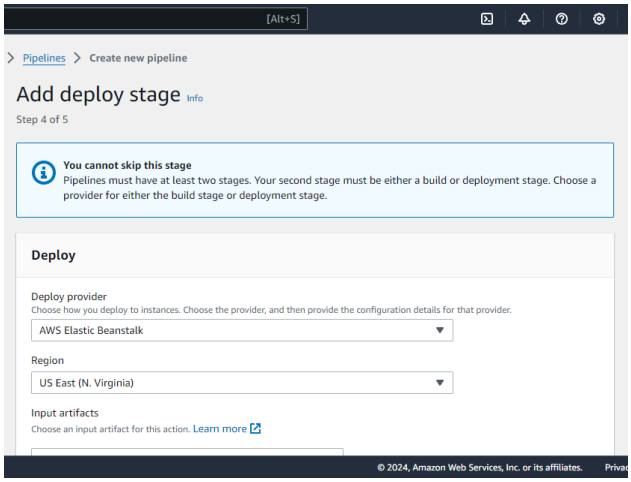
2. In the source stage, choose GitHub v1 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

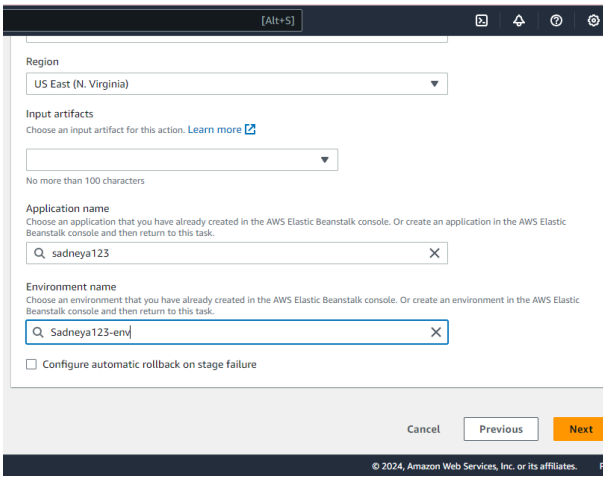


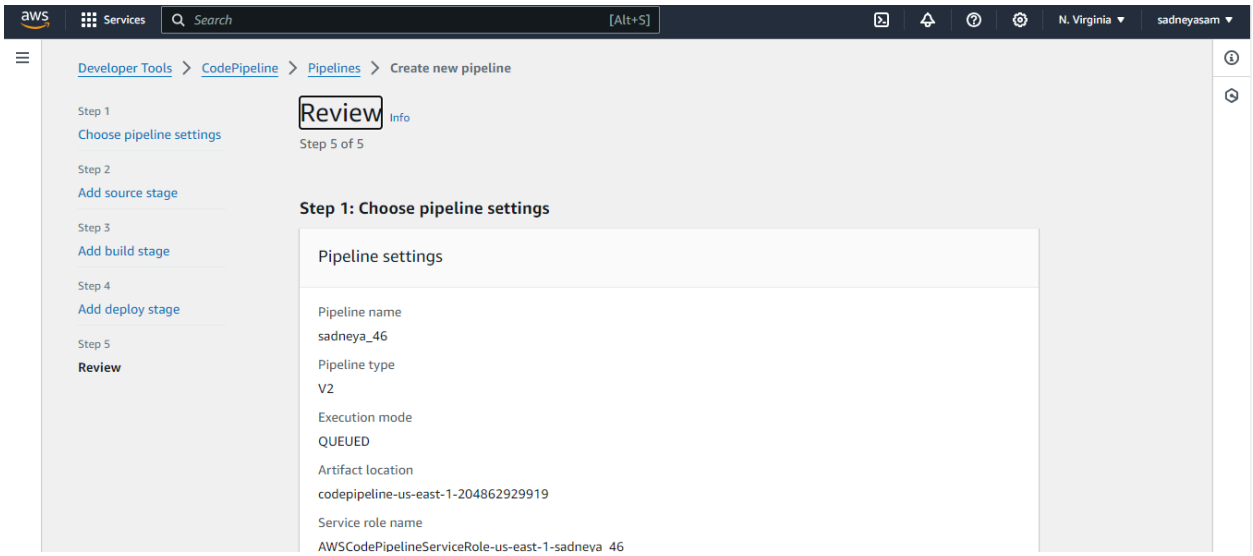
3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

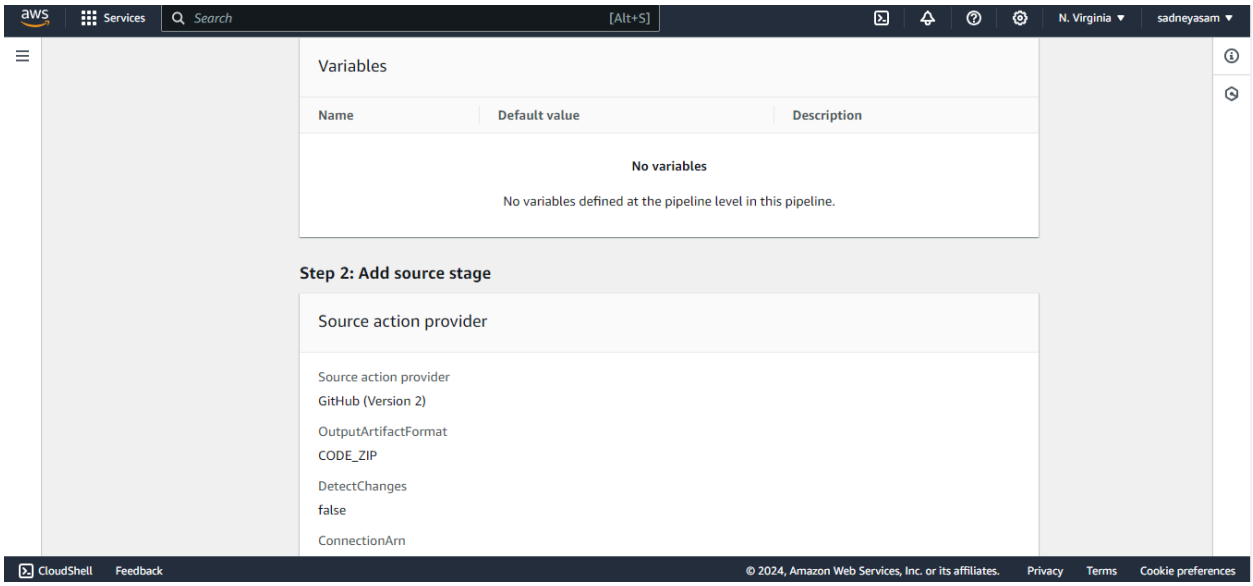
Step 4: Deployment

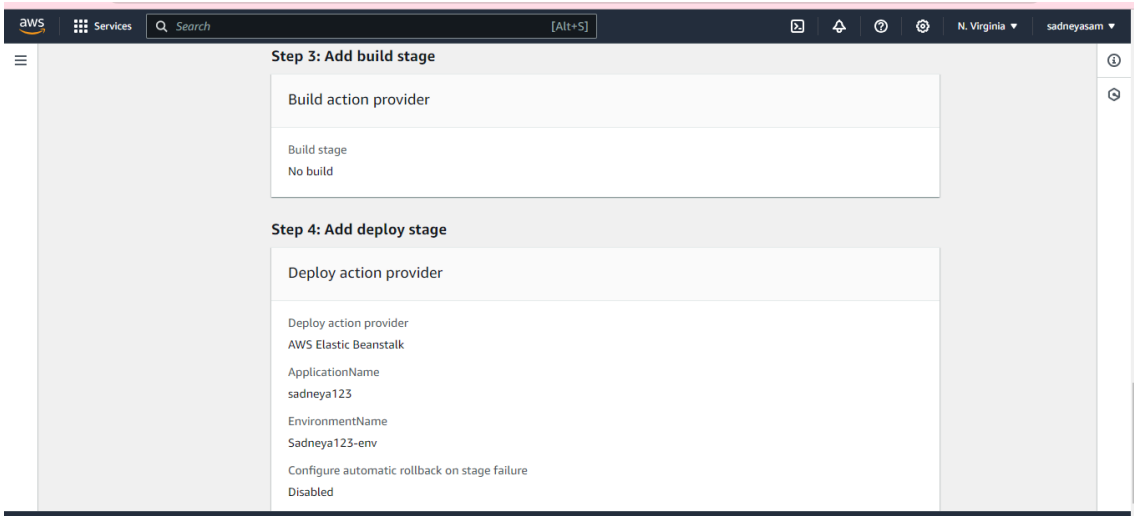
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.

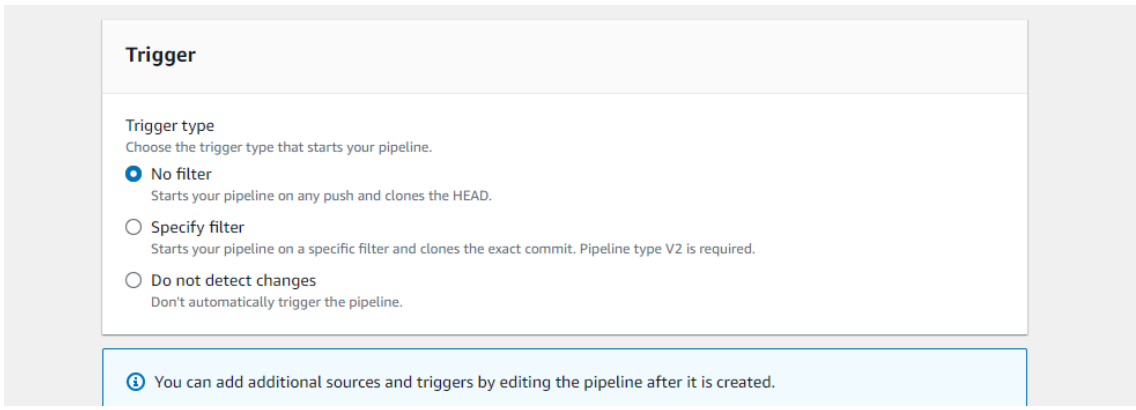

[Alt+S]
Pipelines > Create new pipeline
Add deploy stage Info
Step 4 of 5
You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.
Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk
Region
US East (N. Virginia)
Input artifacts
Choose an input artifact for this action. Learn more
© 2024, Amazon Web Services, Inc. or its affiliates.

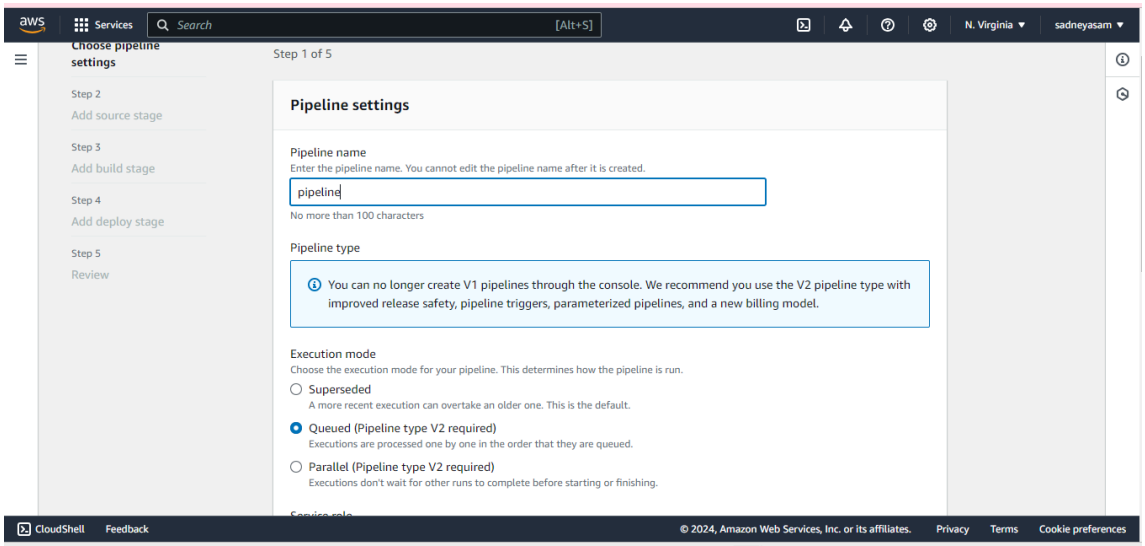

[Alt+S]
Region
US East (N. Virginia)
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
sadneya123
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.
Sadneya123-env
Configure automatic rollback on stage failure
Cancel
Previous
Next
© 2024, Amazon Web Services, Inc. or its affiliates.


aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Review Info
Step 5 of 5
Step 1: Choose pipeline settings
Pipeline settings
Pipeline name
sadneya_46
Pipeline type
V2
Execution mode
QUEUED
Artifact location
codepipeline-us-east-1-204862929919
Service role name
AWSCodePipelineServiceRole-us-east-1-sadneya_46

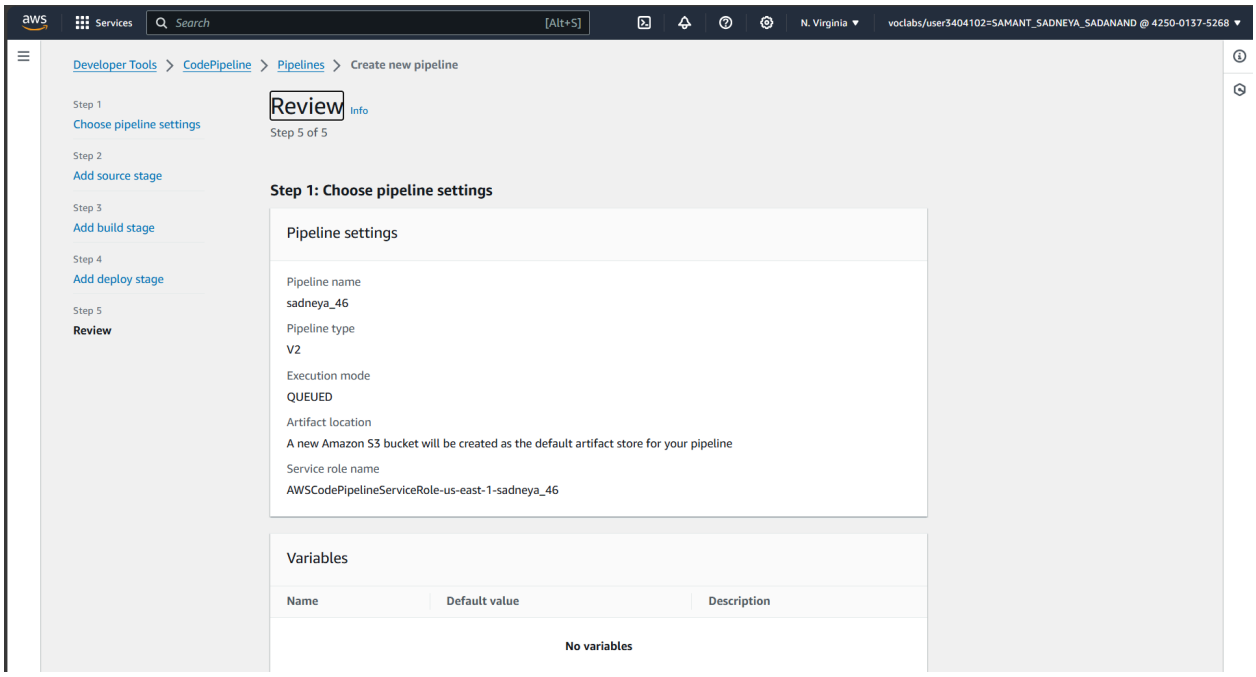

aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Variables
Name
Default value
Description
No variables
No variables defined at the pipeline level in this pipeline.
Step 2: Add source stage
Source action provider
Source action provider
GitHub (Version 2)
OutputArtifactFormat
CODE_ZIP
DetectChanges
false
ConnectionArn
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

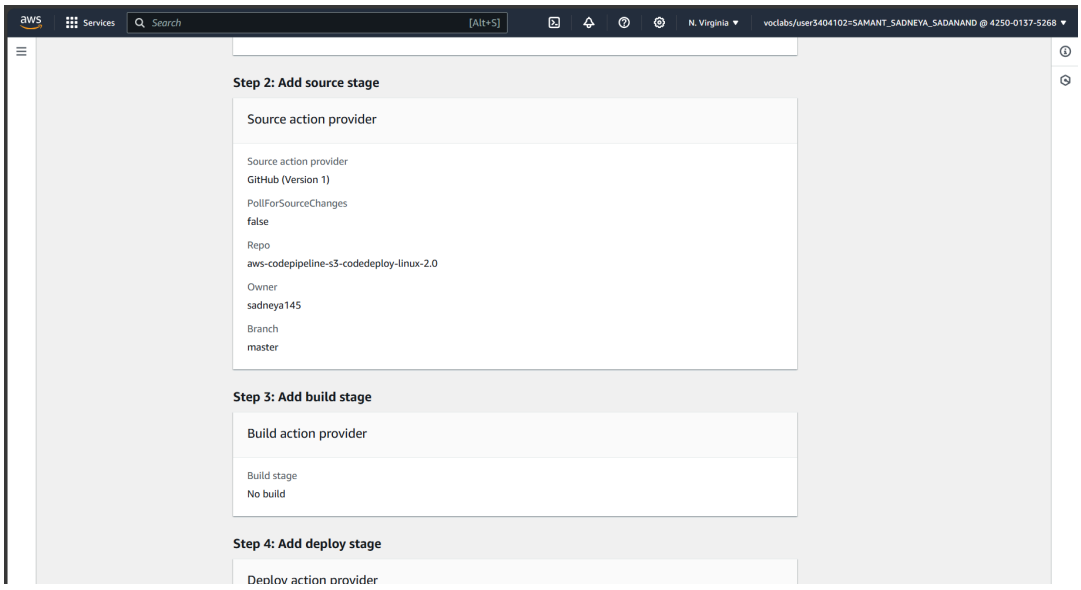

aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Step 3: Add build stage
Build action provider
Build stage
No build
Step 4: Add deploy stage
Deploy action provider
Deploy action provider
AWS Elastic Beanstalk
ApplicationName
sadneya123
EnvironmentName
Sadneya123-env
Configure automatic rollback on stage failure
Disabled

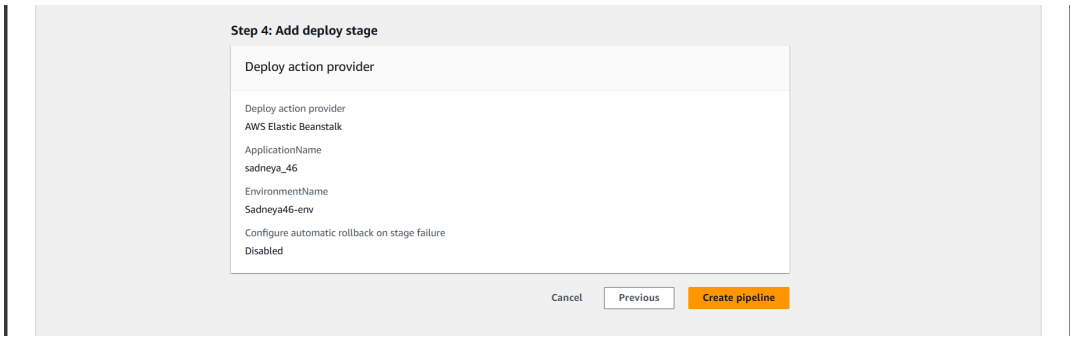

Trigger
Trigger type
Choose the trigger type that starts your pipeline.
No filter
Starts your pipeline on any push and clones the HEAD.
Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
Do not detect changes
Don't automatically trigger the pipeline.
You can add additional sources and triggers by editing the pipeline after it is created.


aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Step 1 of 5
Pipeline settings
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
pipeline
No more than 100 characters
Pipeline type
You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.
Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.
Superseded
A more recent execution can overtake an older one. This is the default.
Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.
Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

2. Review all the settings and click on create pipeline







3.in the end, you can see that the pipeline has been deployed successfully

Experiment No. 3

Objective: To understand the Kubernetes Cluster Architecture and to set up a Kubernetes Cluster on Linux Machines/Cloud.

Procedure:

1. **Instance Setup:**
   - Create three EC2 instances using Amazon Linux as the operating system.
   - Ensure that SSH traffic is allowed from any source.
   - For optimal performance, choose an instance type of at least `t2.medium`, as Kubernetes recommends a minimum of 2 vCPUs.

Name	Instance ID	Instance state	Instance type
kube-master	i-00aa79ac09d7462c0	Running	t2.medium
kube-worker1	i-0bab86cd3fbfcb40a	Running	t2.medium
kube-worker2	i-00dcfd302ffd80dda	Running	t2.medium

2. **SSH Access:**
   - SSH into each of the three machines using separate terminal windows: `ssh -i <keyname>.pem ubuntu@<public_ip_address>`

```
quantum@machine ~/Downloads ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-3-88-111-183.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-111-183.compute-1.amazonaws.com (3.88.111.183)' can't be established.
ED25519 key fingerprint is SHA256:pQu+xs9foYbY3de1twjZcVVAOzmGwGv6PHmVruF/Q1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-111-183.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

3. **Docker Installation and Configuration:**
   - On all three machines, install Docker with the command: `sudo yum install docker -y`

   - Configure Docker to use `systemd` as the cgroup driver by creating and editing the `daemon.json` file:
     - Change directory to `/etc/docker`

- Use the command `cat <<EOF | sudo tee /etc/docker/daemon.json` followed by the JSON configuration details and end with `EOF`

- Enable and restart Docker: `sudo systemctl enable docker sudo systemctl daemon-reload sudo systemctl restart docker docker -v`

```
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
================================================================================
 Package                                                    Architecture
================================================================================
Installing:
 docker                                                     x86_64
Installing dependencies:
 containerd                                                 x86_64
 iptables-libs                                              x86_64
 iptables-nft                                               x86_64
 libcgroup                                                  x86_64
 libnetfilter_conntrack                                     x86_64
 libnfnetlink                                               x86_64
 libnftnl                                                   x86_64
 pigz                                                       x86_64
 runc                                                       x86_64

Transaction Summary
```

4. **Kubernetes Installation:**
   - Disable SELinux before configuring `kubelet`: `sudo setenforce 0 sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config`

   - Add the Kubernetes repository and install Kubernetes components:
     - Use the command `cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo` followed by the repository configuration details and end with `EOF sudo yum update sudo yum install -y kubelet kubeadm kubectl 0--disableexcludes=kubernetes`

- Configure networking for bridging: `sudo swapoff -a echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf sudo sysctl -p`

```
[ec2-user@ip-172-31-81-63 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-81-63 docker]$
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
================================================================================================================================================
 Package                                                 Architecture                                 Version
================================================================================================================================================
Installing:
 kubeadm                                                 x86_64                                       1.30.4-150500.1.1
 kubectl                                                 x86_64                                       1.30.4-150500.1.1
 kubelet                                                 x86_64                                       1.30.4-150500.1.1
Installing dependencies:
 conntrack-tools                                         x86_64                                       1.4.6-2.amzn2023.0.2
 cri-tools                                               x86_64                                       1.30.1-150500.1.1
 kubernetes-cni                                          x86_64                                       1.4.0-150500.1.1
 libnetfilter_cthelper                                   x86_64                                       1.0.0-21.amzn2023.0.2
 libnetfilter_cttimeout                                  x86_64                                       1.0.0-19.amzn2023.0.2
 libnetfilter_queue                                      x86_64                                       1.0.5-2.amzn2023.0.2
 socat                                                   x86_64                                       1.7.4.2-1.amzn2023.0.2

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

5. **Master Node Setup:**
   - Initialize the Kubernetes master node (perform only on the master machine): `sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all`

   - After initialization, set up the Kubernetes configuration on the master node: `mkdir -p $HOME/.kube sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config`

- Save the generated join command from the output for worker nodes. This command is unique and specific to your cluster setup: `kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5\--discovery-token-ca-cert-hashsha256 :dd8426260174d673303aef17717f740772fcf7ee782245bc653eecf4a13 05da7`

```
ubuntu@ip-172-31-28-117:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0sqeukjai8sgfg3 \
        --discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[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-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.396793ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

- Deploy the Flannel networking plugin to enable pod communication: `kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml`

- Check the status of the pods to ensure they are running: `kubectl get pods --all-namespaces`

```
ubuntu@ip-172-31-27-176:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE     VERSION
ip-172-31-18-135   NotReady   <none>          88s     v1.31.1
ip-172-31-27-176   NotReady   control-plane   10m     v1.31.1
ip-172-31-28-117   NotReady   <none>          2m58s   v1.31.1
```

```
ubuntu@ip-172-31-27-176:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
```

sudo systemctl status kubelet

```
ubuntu@ip-172-31-27-176:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
    Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Mon 2024-09-16 15:40:01 UTC; 11min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 5989 (kubelet)
      Tasks: 10 (limit: 4676)
     Memory: 32.6M (peak: 33.2M)
        CPU: 10.705s
     CGroup: /system.slice/kubelet.service
             └─5989 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/

Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497458    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497516    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497569    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497620    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497669    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497719    5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s
Sep 16 15:51:31 ip-172-31-27-176 kubelet[5989]: E0916 15:51:31.605091    5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR
Sep 16 15:51:32 ip-172-31-27-176 kubelet[5989]: I0916 15:51:32.366237    5989 scope.go:117] "RemoveContainer" containerID="f44f06967c5b3e567e07841a7b4352ae
Sep 16 15:51:36 ip-172-31-27-176 kubelet[5989]: E0916 15:51:36.606675    5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR
Sep 16 15:51:41 ip-172-31-27-176 kubelet[5989]: E0916 15:51:41.608404    5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR
```

Now Run command kubectl get nodes -o wide we can see Status is ready.

```
ubuntu@ip-172-31-27-176:~$ ubuntu@ip-172-31-27-176:~$ kubectl get nodes -o wide
NAME              STATUS   ROLES           AGE     VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-18-135  Ready    <none>          6m19s   v1.31.1   172.31.18.135   <none>        Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-27-176  Ready    control-plane   15m     v1.31.1   172.31.27.176   <none>        Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-28-117  Ready    <none>          7m49s   v1.31.1   172.31.28.117   <none>        Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
```

6. **Worker Node Setup:**
    - On each worker node, install the required package and configure `kubelet`: `sudo yum install iproute-tc -y sudo systemctl enable kubelet sudo systemctl restart kubelet`

    - Join the worker nodes to the Kubernetes cluster using the join command from the master node: `kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5\--discovery-token-ca-cert-hashsha256:dd8426260174d673303aef17717f740772fcf7ee782245bc653eecf4a1305da7`

Node 1

```
ubuntu@ip-172-31-28-117:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0sqeukjai8sgfg3 \
        --discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[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-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.396793ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

ubuntu@ip-172-31-28-117:~$
```

Node 2

```
ubuntu@ip-172-31-18-135:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0sqeukjai8sgfg3 \
        --discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6
[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-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001003808s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

ubuntu@ip-172-31-18-135:~$
```

7. **Verify Node Status:**
    - On the master node, verify that the worker nodes have successfully joined the cluster by running: `watch kubectl get nodes`

Or run kubectl get nodes

```
ubuntu@ip-172-31-27-176:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-18-135   Ready    Node2           24m   v1.31.1
ip-172-31-27-176   Ready    control-plane   33m   v1.31.1
ip-172-31-28-117   Ready    Node1           25m   v1.31.1
ubuntu@ip-172-31-27-176:~$
```

Conclusion:

Setting up the Kubernetes cluster involved several challenges. Network configuration issues initially hindered the deployment of the Flannel plugin, requiring open ports and a functional Kubernetes API server. Disabling SELinux and adjusting firewall rules were essential for proper communication between

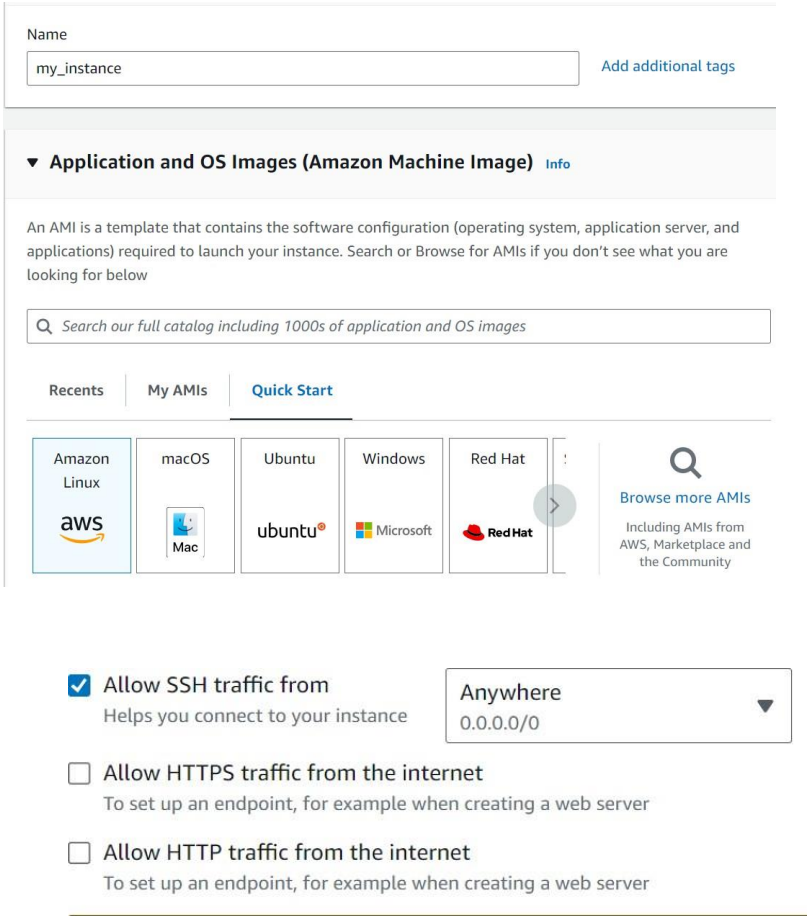
components. Worker nodes experienced difficulties with the kubelet service, which needed to be correctly configured and restarted. Additionally, accurate copying of the join command, including the token and discovery-token-ca-cert-hash, was crucial for integrating worker nodes into the cluster. These issues underscored the need for precise configuration and troubleshooting to achieve a stable Kubernetes setup.

Experiment 4

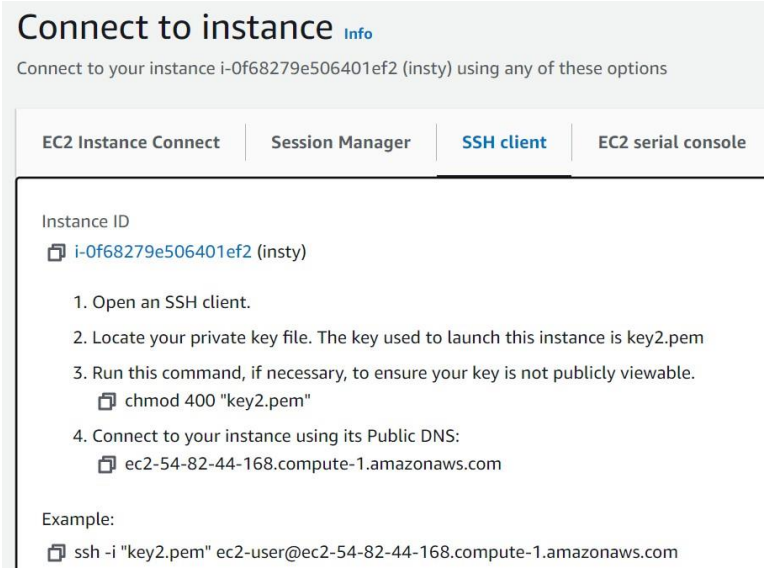
Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Procedure:

1. **Creation Of EC-2 instance**
   Create an EC2 AWS Linux instance on AWS .also edit the Security Group Inbound Rules to allow SSH. then select the t2.micro instance type



- Thus Kuber named -instance gets created.Then click on Id of that instance then click on connect button you will se this.



- Then go into SSH client where you will get this command
  Chmod 400 “keyname.pem”
  ssh -i <keyname>.pem ubuntu@<public_ip_address> copy it and then connect it and run the following command for establishing connection.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 “key name.pem”. Then use the given command for making connections).

```
Anshi@anshi MINGW64 ~
$ cd Downloads

Anshi@anshi MINGW64 ~/Downloads
$ chmod 400 "key2.pem"

Anshi@anshi MINGW64 ~/Downloads
$ ec2-3-85-239-227.compute-1.amazonaws.com
bash: ec2-3-85-239-227.compute-1.amazonaws.com: command not found

Anshi@anshi MINGW64 ~/Downloads
$ ssh -i "key2.pem" ec2-user@ec2-3-85-239-227.compute-1.amazonaws.com
The authenticity of host 'ec2-3-85-239-227.compute-1.amazonaws.com (3.85.239.227
)' can't be established.
ED25519 key fingerprint is SHA256:3ytsjVZbzSc5N7KSAwq0IAh/LRz+zWWqkIlf4gWKjfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-85-239-227.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

2. Installation of Docker

1. . For installation of Docker into the machines run the following command: sudo yum install docker -y

```
      _/m/
[ec2-user@ip-172-31-26-174 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:05:13 ago on Fri Sep 13 13:17:25 2024.
Dependencies resolved.
================================================================================================================================================
 Package                                 Architecture                 Version                                    Repository
================================================================================================================================================
Installing:
 docker                                  x86_64                       25.0.6-1.amzn2023.0.2                      amazonlinux
Installing dependencies:
 containerd                              x86_64                       1.7.20-1.amzn2023.0.1                      amazonlinux
 iptables-libs                           x86_64                       1.8.8-3.amzn2023.0.2                       amazonlinux
 iptables-nft                            x86_64                       1.8.8-3.amzn2023.0.2                       amazonlinux
 libcgroup                               x86_64                       3.0-1.amzn2023.0.1                         amazonlinux
 libnetfilter_conntrack                  x86_64                       1.0.8-2.amzn2023.0.2                       amazonlinux
 libnfnetlink                            x86_64                       1.0.1-19.amzn2023.0.2                      amazonlinux
 libnftnl                                x86_64                       1.2.2-2.amzn2023.0.2                       amazonlinux
 pigz                                    x86_64                       2.5-1.amzn2023.0.3                         amazonlinux
 runc                                    x86_64                       1.1.13-1.amzn2023.0.1                      amazonlinux

Transaction Summary
================================================================================================================================================
```

- Then, configure cgroup in a daemon.json file by using following commands cd /etc/docker
  cat <<EOF | sudo tee /etc/docker/daemon.json
  {
  "exec-opts":
  ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
  "max-size": "100m"
  },
  "storage-driver": "overlay2"
  }
  EOF

```
[ec2-user@ip-172-31-26-174 ~]$ cd /etc/docker
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.
  sudo systemctl enable docker
  sudo systemctl daemon-reload
  sudo systemctl restart docker

```
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/syste
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-26-174 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

- docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Then Install Kubernetes with the following command.

- SELinux needs to be disable before configuring kubelet thus run the following command
  sudo setenforce 0
  sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- Here We are adding kubernetes using the repository whose command is given below. cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
  [kubernetes]
  name=Kubernetes
  baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
  enabled=1
  gpgcheck=1
  gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
  exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
  EOF

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl:

  sudo yum update

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
Kubernetes                                                    100 kB/s |  17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.
==========================================================================================
 Package                   Architecture    Version                    Repository      Size
==========================================================================================
Installing:
 kubeadm                   x86_64          1.30.5-150500.1.1          kubernetes      10 M
 kubectl                   x86_64          1.30.5-150500.1.1          kubernetes      10 M
 kubelet                   x86_64          1.30.5-150500.1.1          kubernetes      17 M
Installing dependencies:
 conntrack-tools           x86_64          1.4.6-2.amzn2023.0.2       amazonlinux    208 k
 cri-tools                 x86_64          1.30.1-150500.1.1          kubernetes     8.6 M
 kubernetes-cni            x86_64          1.4.0-150500.1.1           kubernetes     6.7 M
 libnetfilter_cthelper     x86_64          1.0.0-21.amzn2023.0.2      amazonlinux     24 k
 libnetfilter_cttimeout    x86_64          1.0.0-19.amzn2023.0.2      amazonlinux     24 k
 libnetfilter_queue        x86_64          1.0.5-2.amzn2023.0.2       amazonlinux     30 k

Transaction Summary
==========================================================================================
Install  9 Packages
```

```
Total                                                          64 MB/s |  53 MB     00:00
Kubernetes                                                     20 kB/s | 1.7 kB     00:00
Importing GPG key 0x9A296436:
 Userid     : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
 Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
 From       : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                              1/1
  Installing       : kubernetes-cni-1.4.0-150500.1.1.x86_64                       1/9
  Installing       : cri-tools-1.30.1-150500.1.1.x86_64                           2/9
  Installing       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64               3/9
  Installing       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64          4/9
  Installing       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64           5/9
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                  6/9
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                  6/9
  Installing       : kubelet-1.30.5-150500.1.1.x86_64                             7/9
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64                             7/9
  Installing       : kubeadm-1.30.5-150500.1.1.x86_64                             8/9
  Installing       : kubectl-1.30.5-150500.1.1.x86_64                             9/9
  Running scriptlet: kubectl-1.30.5-150500.1.1.x86_64                             9/9
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                  1/9
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64           2/9
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64          3/9
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64               4/9
  Verifying        : cri-tools-1.30.1-150500.1.1.x86_64                           5/9
  Verifying        : kubeadm-1.30.5-150500.1.1.x86_64                             6/9
  Verifying        : kubectl-1.30.5-150500.1.1.x86_64                             7/9
  Verifying        : kubelet-1.30.5-150500.1.1.x86_64                             8/9
  Verifying        : kubernetes-cni-1.4.0-150500.1.1.x86_64                       9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64     cri-tools-1.30.1-150500.1.1.x86_64     kubeadm-1.30.5-150500.1.1.x86_64
  kubectl-1.30.5-150500.1.1.x86_64     kubelet-1.30.5-150500.1.1.x86_64     kubernetes-cni-1.4.0-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64     libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64     libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
```

- After installing Kubernetes, we need to configure internet options to allow bridging.

1. sudo swapoff -a
2. echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
3. sudo sysctl -p

```
[ec2-user@ip-172-31-26-174 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

4. Initialize the Kubecluster

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0913 10:32:44.629146   26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks

Your Kubernetes control-plane has initialized successfully!

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

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

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

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8 \
        --discovery-token-ca-cert-hash sha256:8293b2f6d29de466bd859007f5adbcdb3a
ecb0c446ba09033d32a5846b3d434f
```

- copy the token and save for future use .
  kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8
  \--discovery-token-ca-cert-hash
  sha256:8293b2f6d29de466bd859007f5adbcdb3aecb0c446ba09033d32a5846b3d434f
- Copy the mkdir and chown commands from the top and execute them
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
[ec2-user@ip-172-31-80-126 docker]$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- Then, add a common networking plugin called flannel as mentioned
  in the code.

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

```
[ec2-user@ip-172-31-26-174 docker]$ 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
```

5. Now that the cluster is up and running,we can deploy our nginx

server on this cluster. Apply deployment using this following command:

kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://k8s.io/examples/pods/s
imple-pod.yaml
pod/nginx created
```

Then use **kubectl get pods** to check whether the pod gets created or not.

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          12s
```

To convert state from pending to running use following command:

kubectl describe pod nginx This command will help to describe the pods

it gives reason for failure as it shows the untolerated taints which need to

be untainted.

- kubectl describe pod nginx

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k4lj6 (ro)
```

```
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-k4lj6:
    Type:                    Projected (a volume that contains injected data from m
ultiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 3
00s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for
 300s
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  7s    default-scheduler  0/1 nodes are available: 1 no
de(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption:
0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

- kubectl taint nodes --all node-role.kubernetes.io/control-plane-

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl taint nodes --all node-role.kubernetes.io
/control-plane-
node/ip-172-31-26-174.ec2.internal untainted
```

6. Now check pod status is is running perform **kubectl get pods** this command.

```
[ec2-user@ip-172-31-28-70 docker]$  kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          39s
[ec2-user@ip-172-31-28-70 docker]$  kubectl get pods
NAME    READY   STATUS    RESTARTS      AGE
nginx   1/1     Running   1 (45s ago)   70s
```

7. Lastly, mention the port you want to host. Here i have used localhost 8081 then check it.

kubectl port-forward nginx 8081:80

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

8. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8081

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\bhush\one drive 2\OneDrive\Desktop\New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sun Sep 15 07:58:53 UTC 2024

  System load:  0.15              Processes:             152
  Usage of /:   55.3% of 6.71GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.20.171
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

132 updates can be applied immediately.
38 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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


Last login: Sun Sep 15 07:27:23 2024 from 152.58.2.47
```

HTTP/1.1 200 OK

```
ubuntu@ip-172-31-20-171:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 07:59:03 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

If the response is 200 OK and you can see the Nginx server name, your deployment was successful. We have successfully deployed our Nginx server on our EC2 instance.

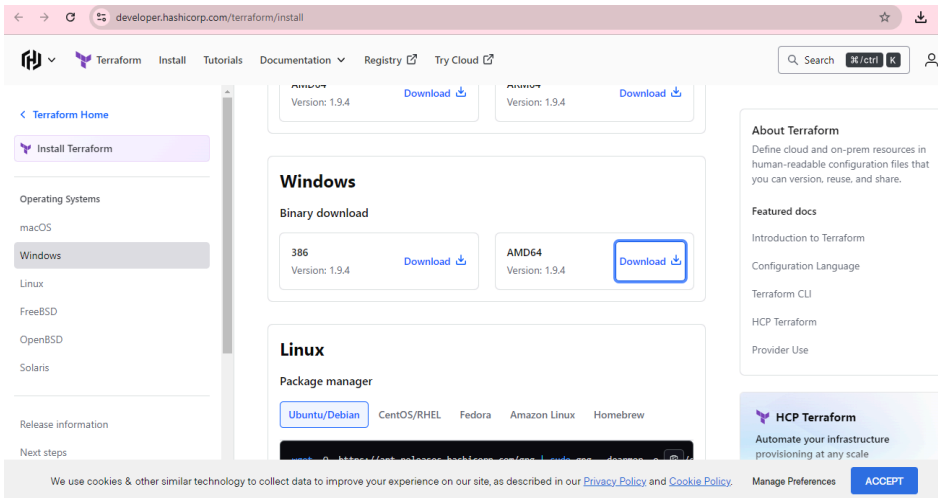
Conclusion: Firstly I created an EC2 AWS Linux instance successfully.then installed docker and kubernetes successfully.then initialized kubernetes which given me token and chown and mkdir command. Then I execute mkdir and chown the command successfully. Then I installed a networking plugin called flannel successfully. Then I tried to deploy nginx which initially gave an error. Then I deployed (simple-pod.yml) nginx successfully and also checked by using the get pods command.then hosted it on localhost 8081 ie http://localhost:8081 successfully.

Adve_devops_5 Name: Anshi Tiwari Roll no:56

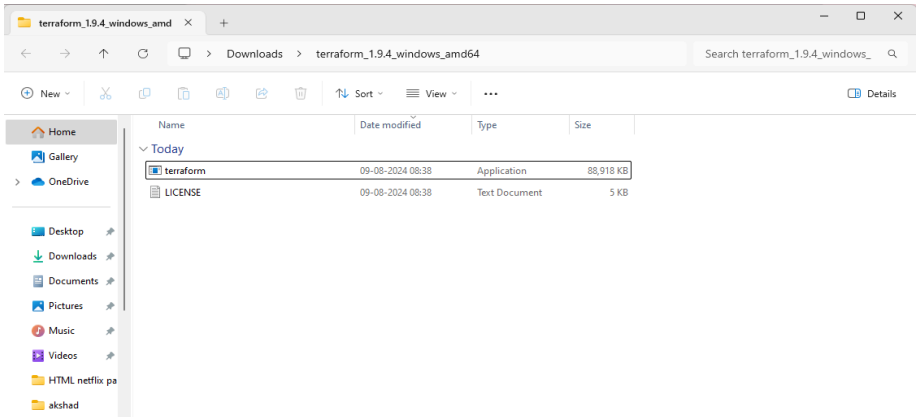
1. Download terraform

   To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

   Website: https://www.terraform.io/downloads.html

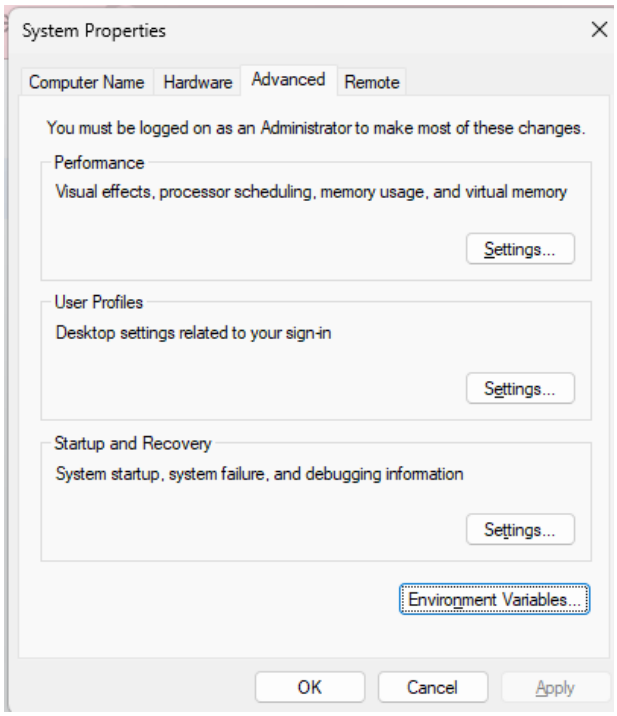


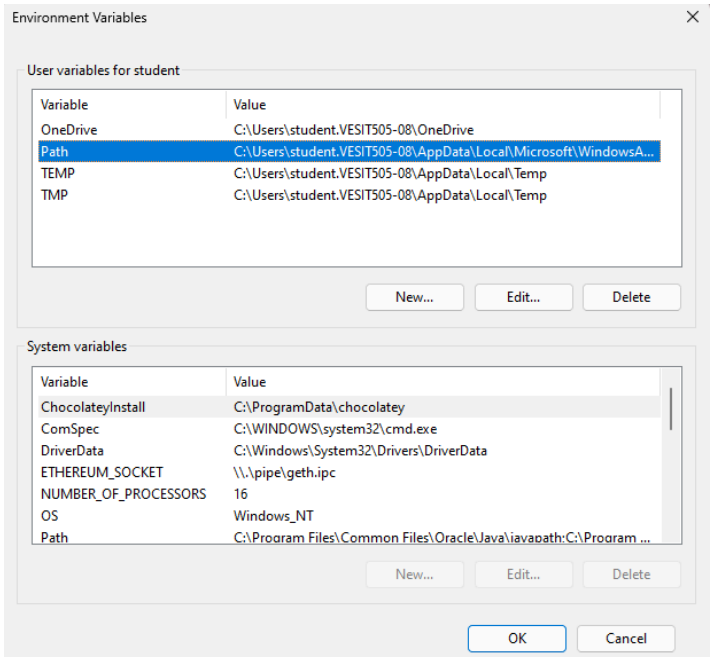
2. Extract the downloaded setup file Terraform.exe in C:\Terraform directory

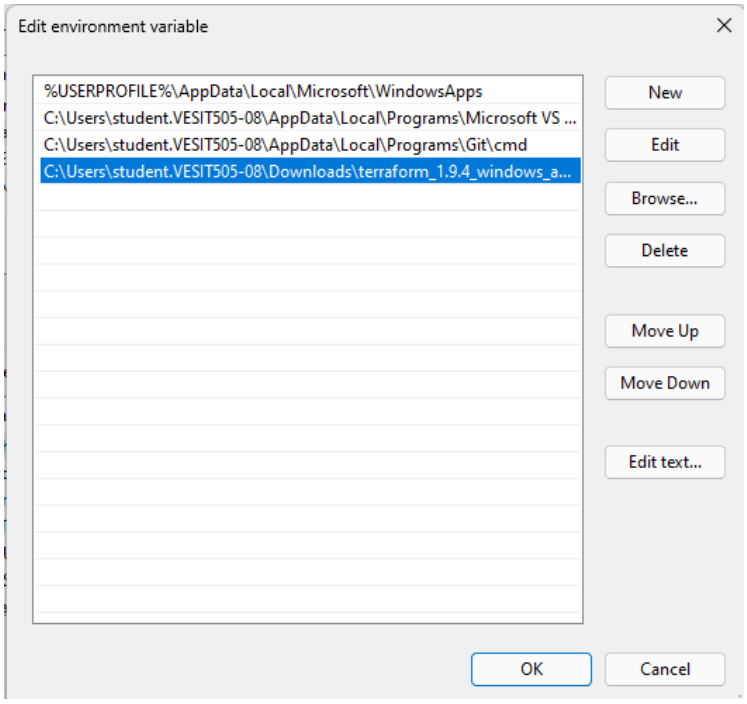


Adve_devops_5 Name: Anshi Tiwari Roll no:56

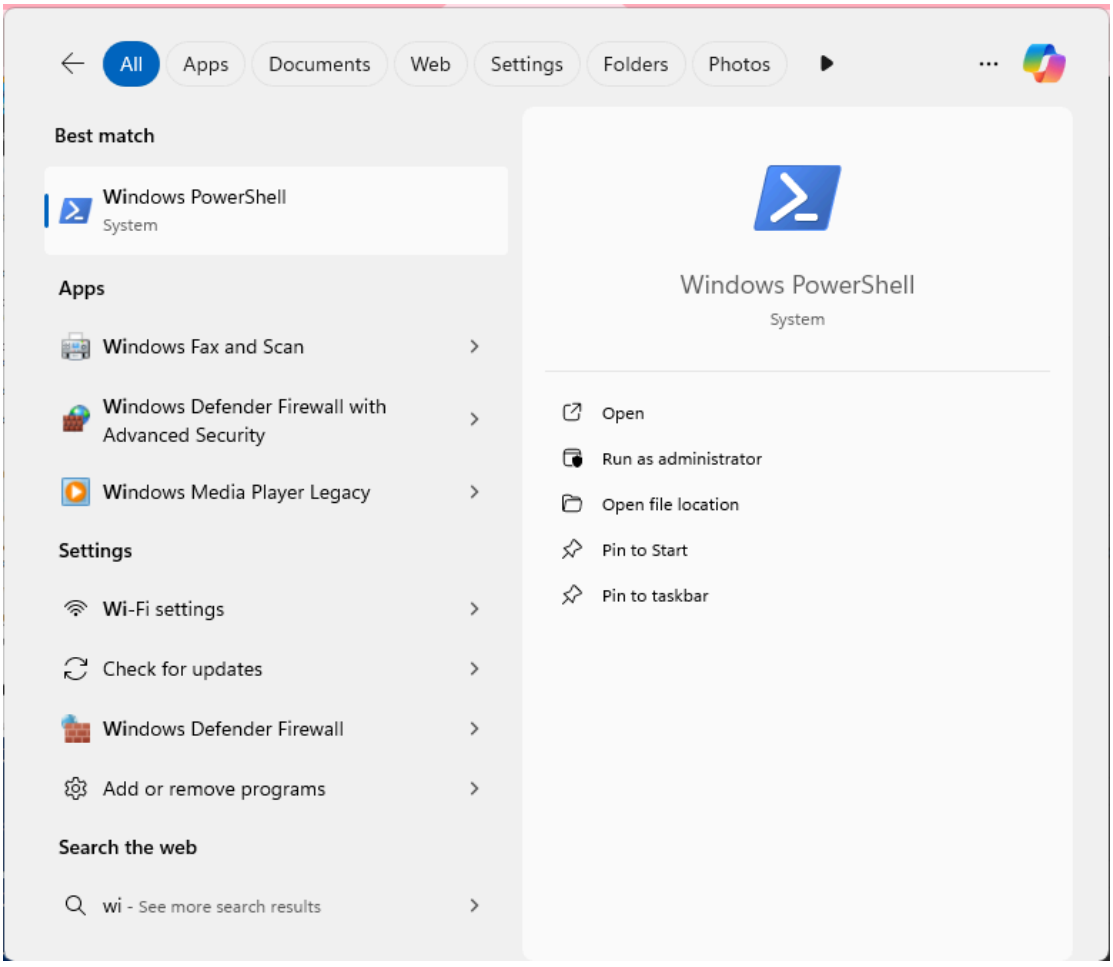
3. Set the System path for Terraform in Environment Variables







4. Open PowerShell with Admin Access



5. Open Terraform in PowerShell and check its functionality

Experiment 6:

1. Run some basic commands to check the version and confirm installation of docker

```
C:\Users\INFT505-17>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx
  compose*    Docker Compose
  container   Manage containers
  context     Manage contexts
```

```
C:\Users\INFT505-17>docker --version
Docker version 27.1.1, build 6312585
```

2. Initialise terraform

```
C:\Users\INFT505-17>terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.

C:\Users\INFT505-17>cd Desktop

C:\Users\INFT505-17\Desktop>
C:\Users\INFT505-17\Desktop>cd Terraform Scripts

C:\Users\INFT505-17\Desktop\Terraform Scripts>cd Docker

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

3. Terraform plan

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform plan

Terraform used the selected providers to generate the following exec
following symbols:
  + create

Terraform will perform the following actions:

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

4. Terraform apply

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

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

  Enter a value: yes

docker_container.foo: Creating...
docker_container.foo: Creation complete after 3s [id=1adc9dfc498825398e014939d7966749d843181417953d0b9f462a55ae7c1492]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    edbfe74c41f8   3 weeks ago   78.1MB
```

5. Terraform destroy

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=ad11c2cdc9ccae208b906f7e6be260805512a66568a65f6b3bffb3682cf30cb4]

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

Terraform will perform the following actions:

  # docker_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach            = false -> null
      - command           = [
          - "/bin/bash",
          - "-c",
          - "while true; do sleep 3600; done",
        ] -> null
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      - entrypoint        = [] -> null
```

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

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

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

  Enter a value: yes

docker_container.foo: Destroying... [id=ad11c2cdc9ccae208b906f7e6be260805512a66568a65f6b3bffb3682cf30cb4]
docker_container.foo: Destruction complete after 2s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>
```

6. Terraform Validate

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform validate
Success! The configuration is valid.
```

7. Terraform refresh

```
:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform refresh
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubu
:u:latest]
docker_container.foo: Refreshing state... [id=4adda4f9a5c585809eebe921da757560d333f9a0101a8cd25784bd238355aba5]
```

8. Terraform state list

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform state list
docker_container.foo
docker_image.ubuntu
```

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

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

What problems does SAST solve?

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

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

Why is SAST important?

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

What are the key steps to run SAST effectively?

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

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

2. **Create the scanning infrastructure, and deploy the tool.** This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.

3. **Customize the tool.** Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.

4. **Prioritize and onboard applications.** Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.

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

6. **Provide governance and training.** Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

Integrating Jenkins with SonarQube:

Windows installation
Step 1 Install JDK 1.8
Step 2 download and install jenkins
https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows

Ubuntu installation

https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt on-ubuntu-20-04#installing-the-default-jre-jdk

Step 1 Install JDK 1.8
sudo apt-get install openjdk-8-jre

sudo apt install default-jre

https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu 20-04

Open SSH

Prerequisites:

- Jenkins installed
- Docker Installed (for SonarQube)

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

- SonarQube Docker Image

Steps to integrate Jenkins with SonarQube

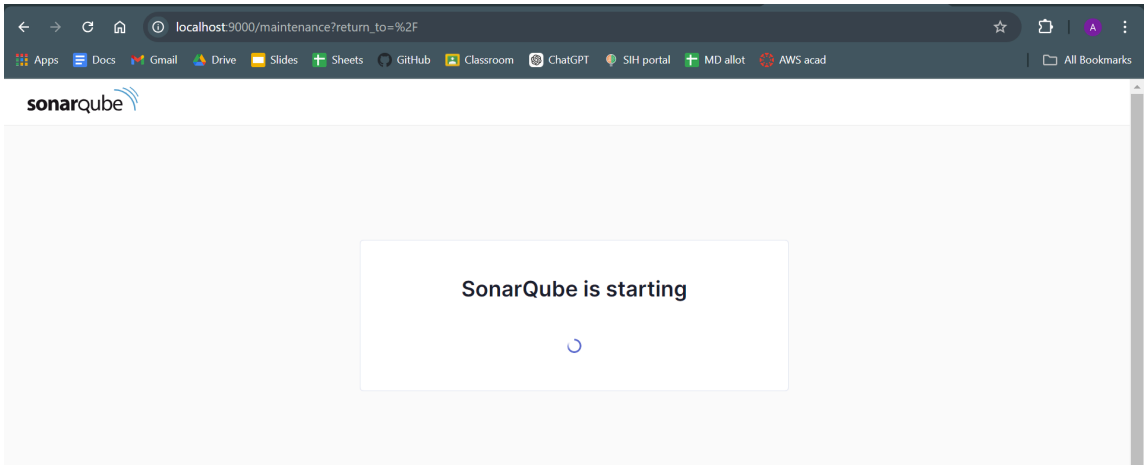
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -

```
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
b37288b0b410d9fab6bebfd8d6b87a0ef7f75387b6070308da2f24c8548481cc
```

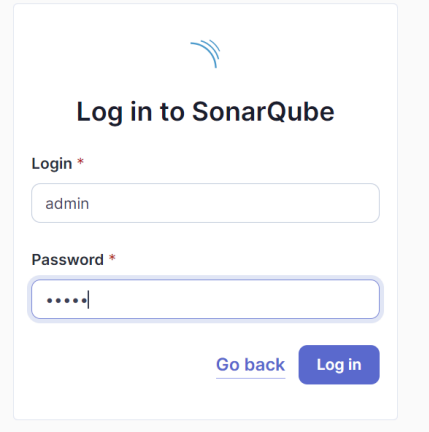
Warning: run below command only once

docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

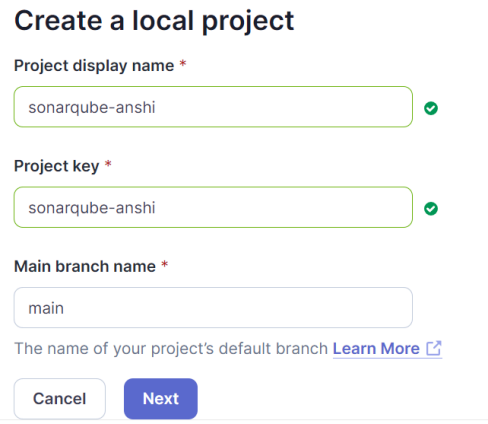
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



4. Login to SonarQube using username *admin* and password *admin*.

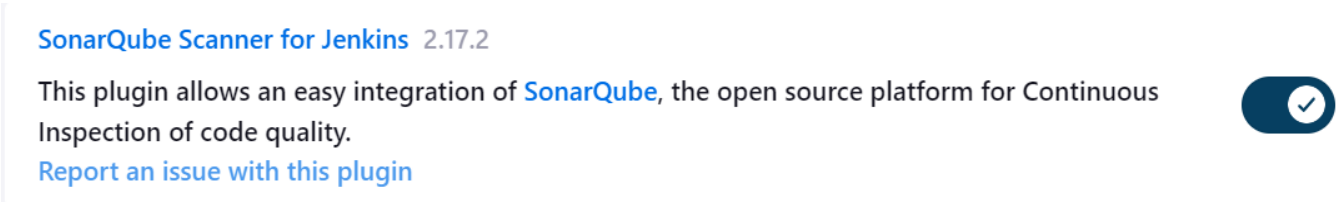


5. Create a manual project in SonarQube with the name **sonarqube**



Setup the project and come back to Jenkins Dashboard.

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



6. Under Jenkins 'Configure System', look for SonarQube Servers and enter the details.

Enter the Server Authentication token if needed.

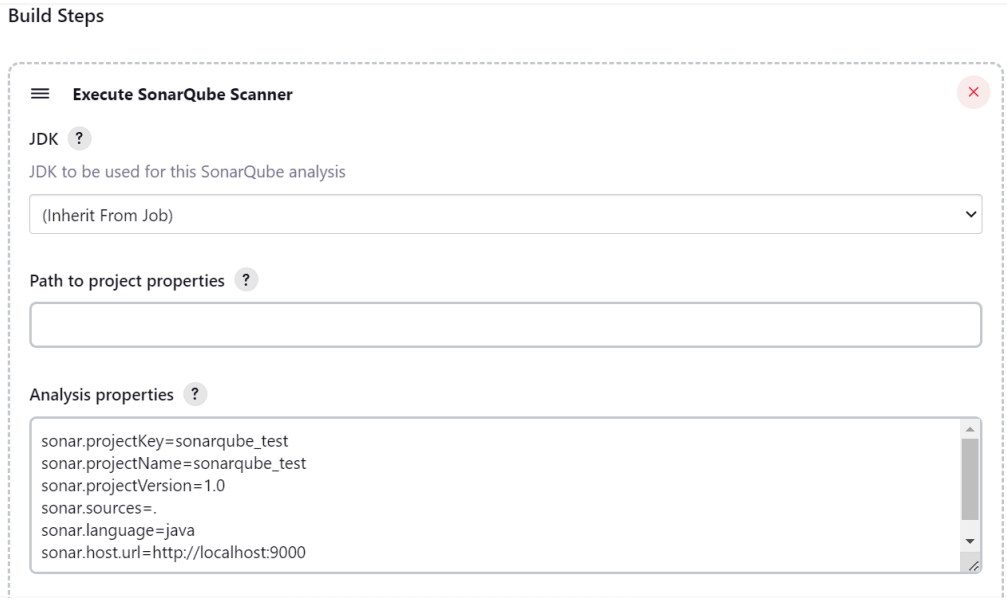
7. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

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

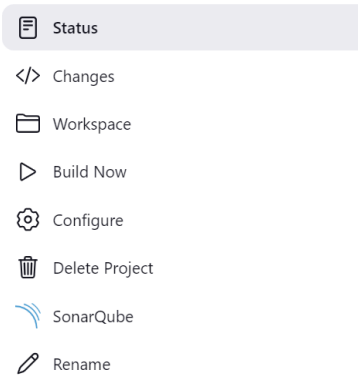
9. Choose this GitHub repository in Source Code Management.
https://github.com/shazforiot/MSBuild_firstproject.git
It is a sample hello-world project with no vulnerabilities and issues, just to test

the integration.

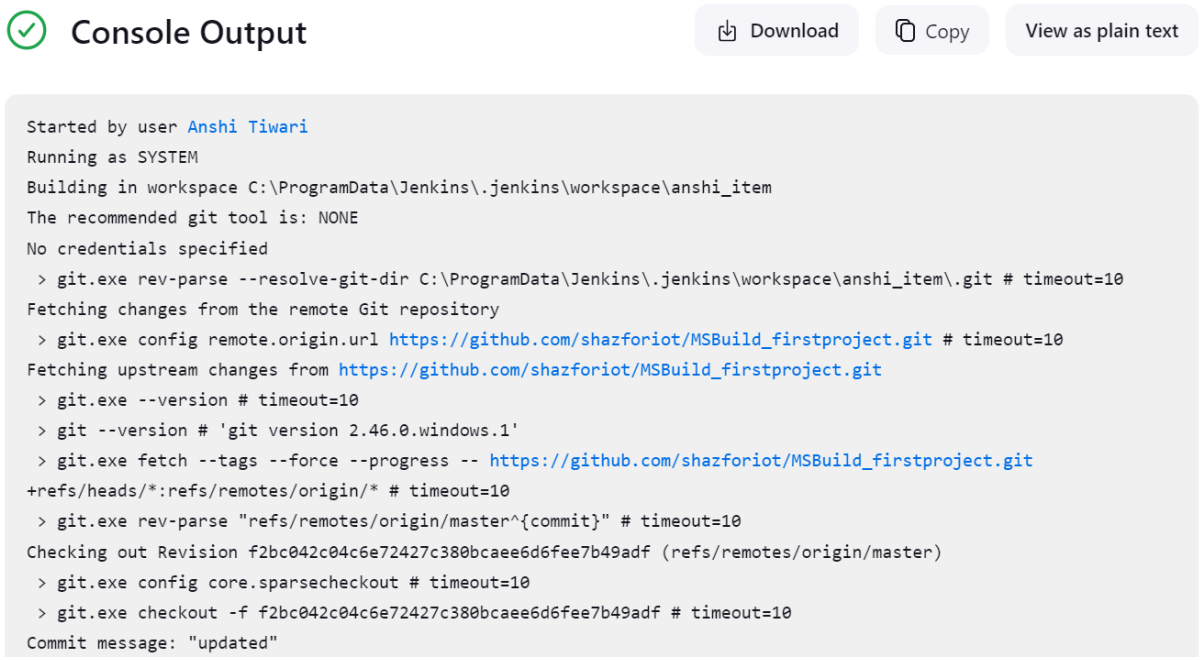
10. Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL. 11. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.



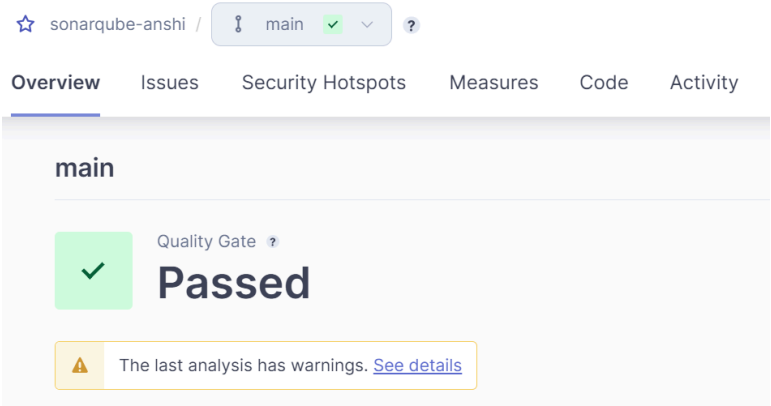
12. Run The Build.



Check the console output.



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



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

Conclusion of the Experiment

This experiment focused on integrating Jenkins with SonarQube for Static Application Security Testing (SAST). Throughout the process, several challenges emerged:

1. **Token Authentication Issues**: Encountering a 401 Unauthorized error emphasized the importance of correctly managing authentication tokens and user permissions within SonarQube.

2. **Project Configuration**: Initial confusion regarding the configuration parameters in Jenkins highlighted the need for attention to detail. Ensuring the sonar.login property was set correctly was crucial for successful integration.
3. **Maven Misunderstanding**: Misunderstanding the role of Maven led to unnecessary command-line attempts. Recognizing that Jenkins handles the build process streamlined the workflow.
4. **Error Navigation**: Reading console logs proved vital in troubleshooting and identifying issues, enhancing my problem-solving skills.

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.

Theory:

What is SAST?

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

What problems does SAST solve?

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

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

Why is SAST important?

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

What is a CI/CD Pipeline?

CI/CD pipeline refers to the Continuous Integration/Continuous Delivery pipeline. Before we dive deep into this segment, let's first understand what is meant by the term ‘pipeline’?

A pipeline is a concept that introduces a series of events or tasks that are connected in a sequence to make quick software releases. For example, there is a task, that task has got five different stages, and each stage has got some steps. All the steps in phase one have to be completed, to mark the latter stage to be complete.

Now, consider the CI/CD pipeline as the backbone of the DevOps approach. This Pipeline is responsible for building codes, running tests, and deploying new software versions. The Pipeline executes the job in a defined manner by first coding it and then structuring it inside several blocks that may include several steps or tasks.

What is SonarQube?

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code

smells, vulnerabilities, code duplications.
It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

Benefits of SonarQube

● **Sustainability -** Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.

● **Increase productivity -** Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code

● **Quality code -** Code quality control is an inseparable part of the process of software development.

● **Detect Errors -** Detects errors in the code and alerts developers to fix them automatically before submitting them for output.

● **Increase consistency -** Determines where the code criteria are breached and enhances the quality

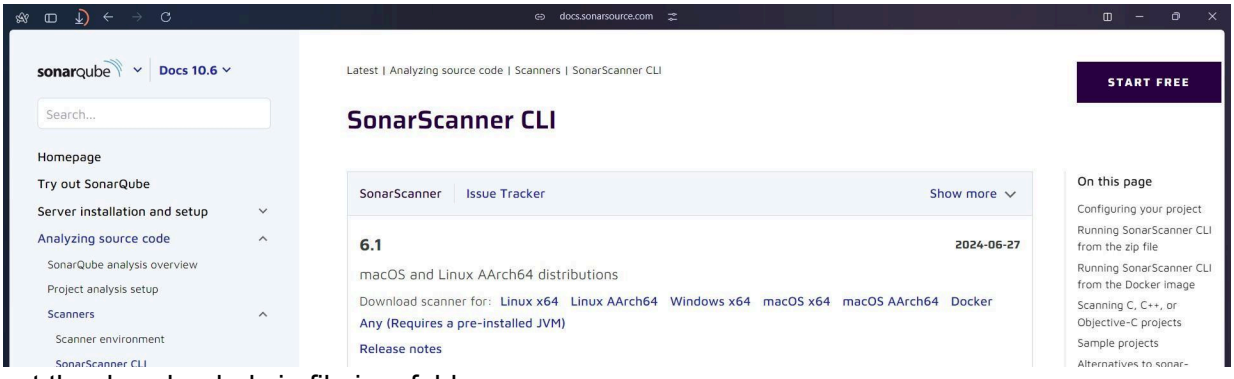
● **Business scaling -** No restriction on the number of projects to be evaluated

● **Enhance developer skills -** Regular feedback on quality problems helps developers to improve their coding skills

Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

Step 1: Download sonar scanner

https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/ Visit this link and download the sonarqube scanner CLI.



Extract the downloaded zip file in a folder.

1) Docker

Run docker -v command.

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

2) Install sonarqube image

Command: docker pull sonarqube

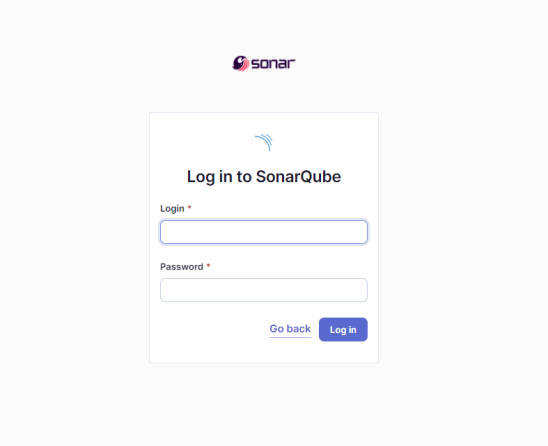
```
C:\Users\Anshi>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

3) Run sonarqube image

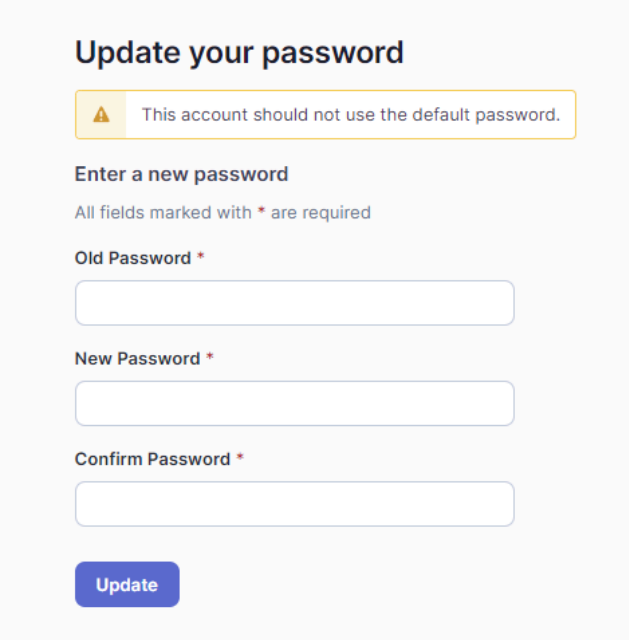
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Users\saira\OneDrive\Desktop\AdvDevOps\lab7> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=t
rue -p 9000:9000 sonarqube:latest
36ff8a656bd28857ba9a28bf2bb0174099ae3232a9fc9ba2766d46f0c14d08a6
```

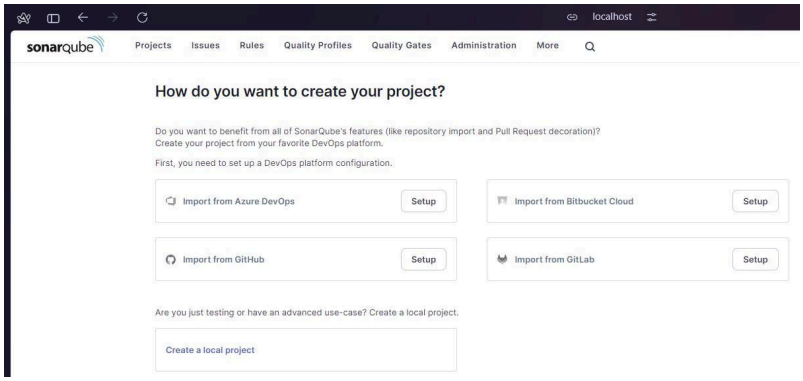
4) Run localhost:9000

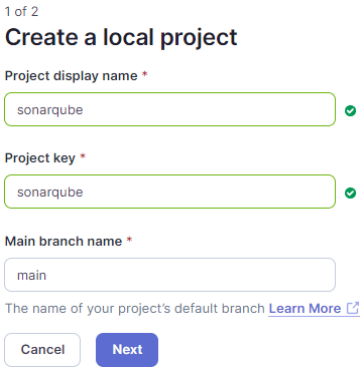


5) Login using username=”admin”, password=”admin”. It will promt you to set a new password.

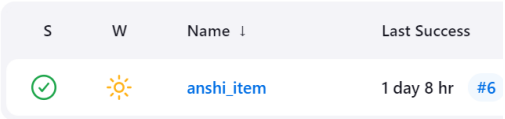


6) This is the interface. Create a local project with the name sonarqube.

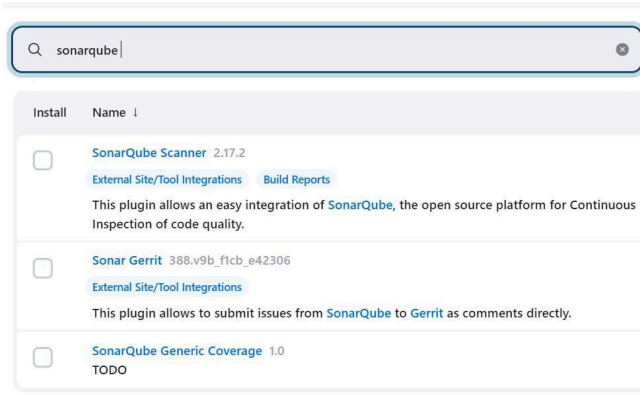




7) Open jenkins dashboard using localhost on whichever port it is hosted.

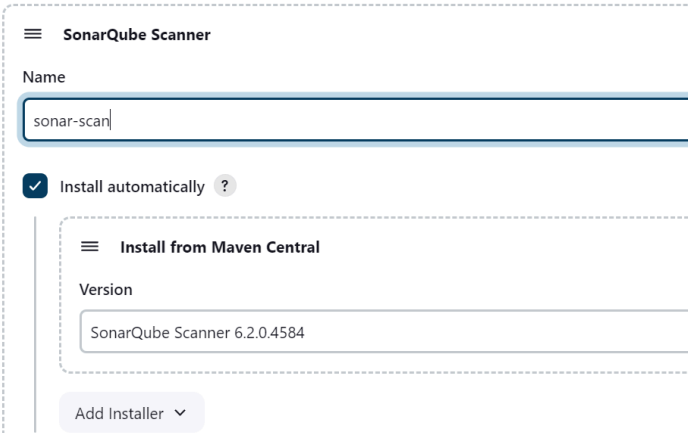


8) Go to manage jenkins → Search for Sonarqube Scanner for Jenkins and install it.

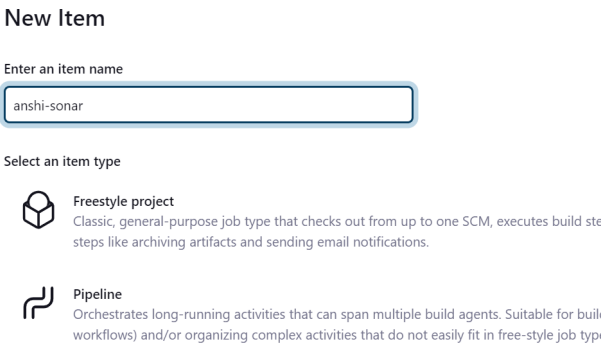


9) Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.

10) Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose install automatically.



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



12) 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 Jenkins>') { 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=<Link to hosted SonarQube>(default: http://localhost:9000/) """
}
}
}
```

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

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=<admin123> \
-D sonar.projectKey=<sonarqube-anshi-2> \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=<Link to hosted SonarQube>(default: http://localhost:9000/) """
}
}
}
```

13) Check the console output once

Console Output

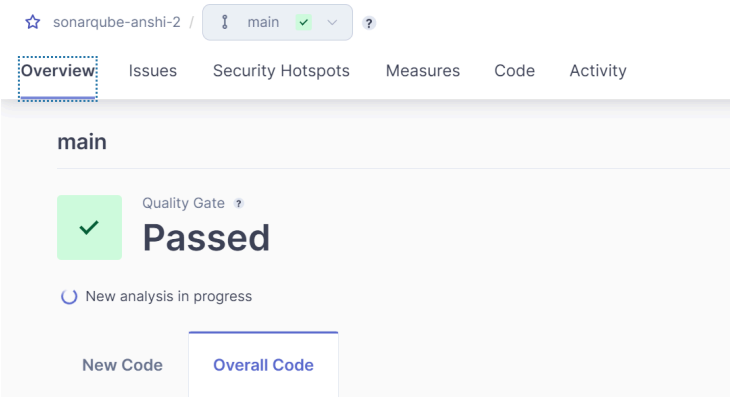
Download Copy View as plain text

```
Started by user Anshi Tiwari
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\anshi-sonar
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
```

```
21:10:28.890 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
21:10:34.239 INFO  Analysis report generated in 3621ms, dir size=127.2 MB
21:10:51.328 INFO  Analysis report compressed in 17087ms, zip size=29.6 MB
21:10:51.873 INFO  Analysis report uploaded in 546ms
21:10:51.874 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-anshi-2
21:10:51.874 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:10:51.874 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=b1d670e7-bff2-41c8-8e0a-b5ab1d303aac
21:11:03.958 INFO  Analysis total time: 7:46.716 s
21:11:03.960 INFO  SonarScanner Engine completed successfully
21:11:04.665 INFO  EXECUTION SUCCESS
21:11:04.666 INFO  Total time: 7:51.358s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

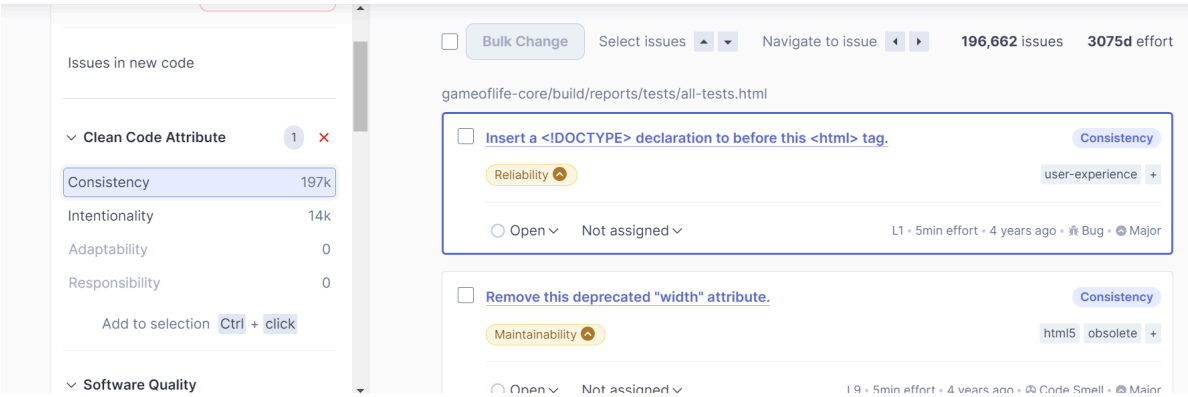
14) Now, check the project in SonarQube

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

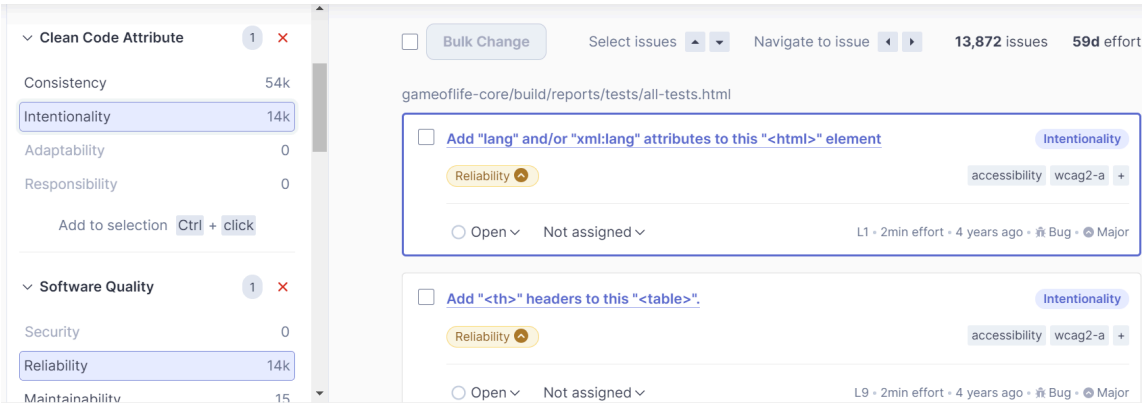


15) Code Problems

- Consistency



- Intentionality



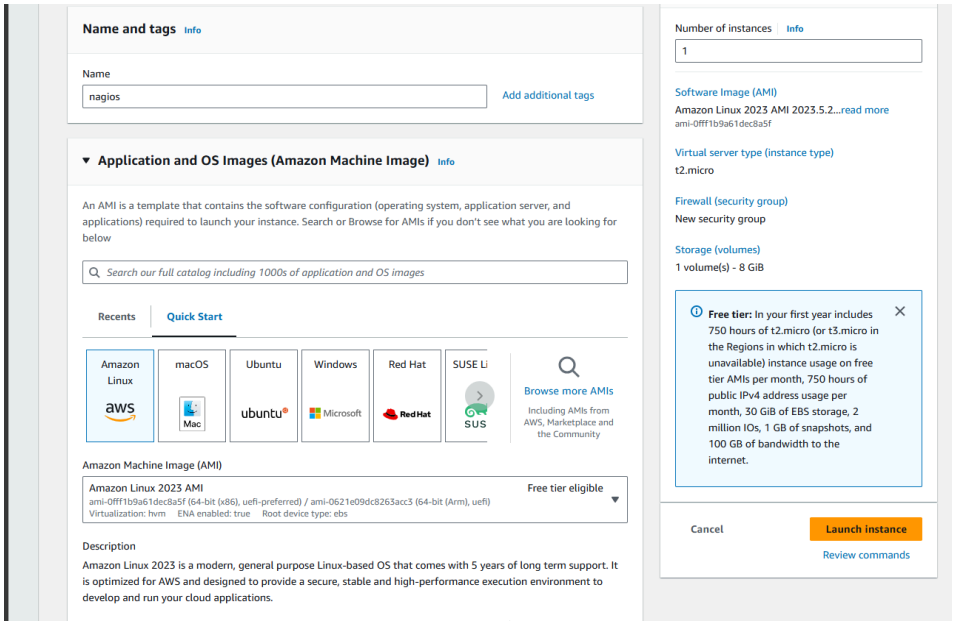
Conclusion

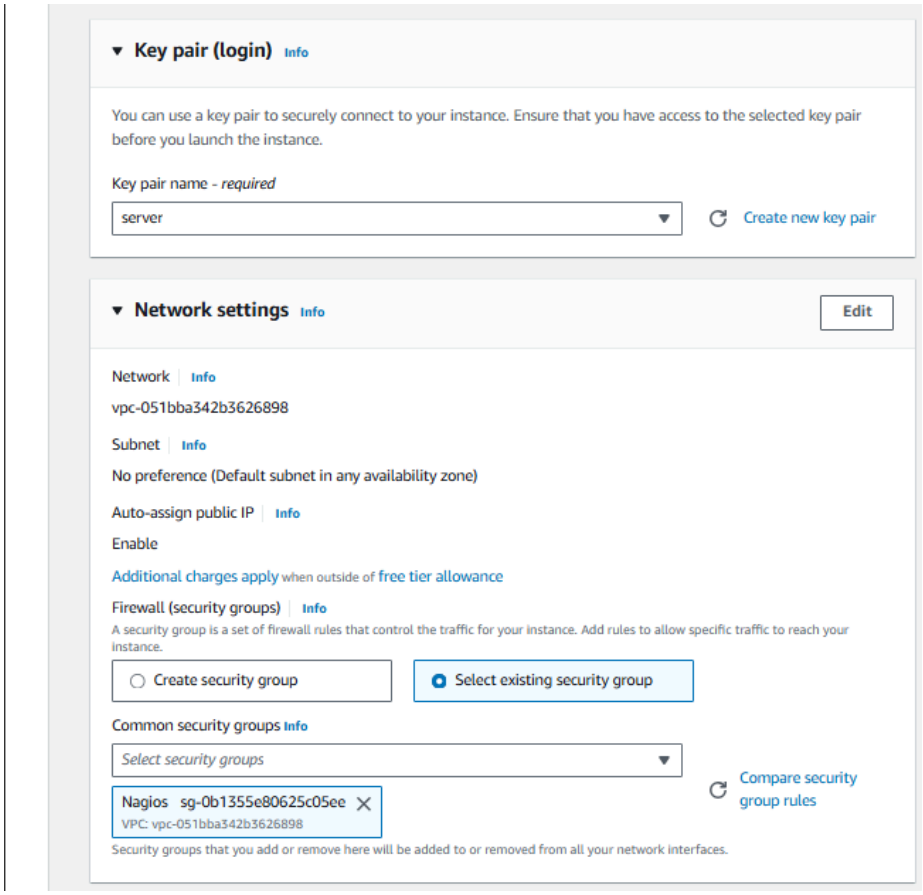
The experiment successfully integrated SonarQube analysis into a Jenkins pipeline for a GitHub project, allowing for automated code quality checks. While the cloning of the repository worked well, several challenges arose, particularly with configuring the SonarScanner path and handling command execution errors in the Windows environment. Issues with authentication and the need for a user token added complexity to the setup. Despite these hurdles, the integration improved the project's continuous integration process, and lessons learned can inform future enhancements, such as better error handling and notifications for analysis results.

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

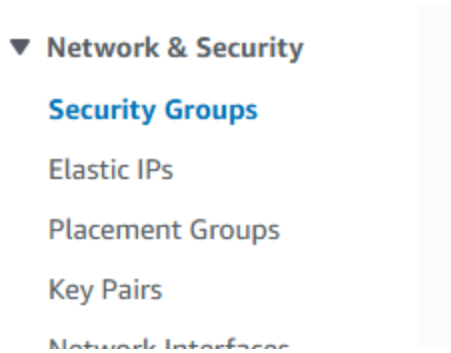
Steps:

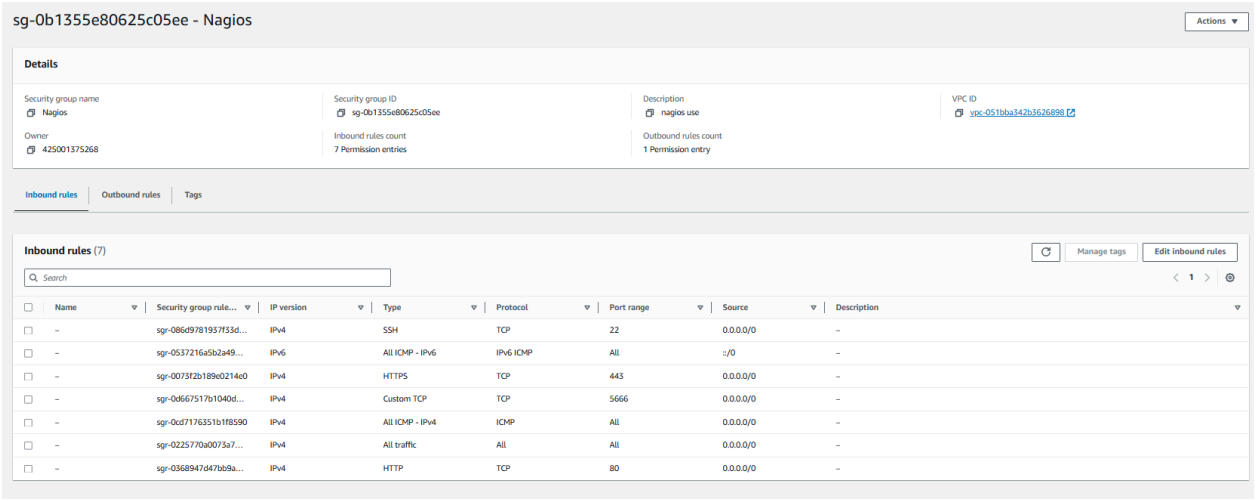
1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host





2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.





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-172-31-39-90 ~]$ sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
Last metadata expiration check: 0:02:27 ago on Fri Oct  4 03:35:02 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

```
Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64                    brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
  bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64               cairo-1.17.6-2.amzn2023.0.1.x86_64
  cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64         fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64        fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
  freetype-2.13.2-5.amzn2023.0.1.x86_64                 freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
  gd-2.3.3-5.amzn2023.0.3.x86_64                        gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64            google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch  graphite2-1.3.14-7.amzn2023.0.2.x86_64
  graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64          harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64            harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64               langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
  libICE-1.0.10-6.amzn2023.0.2.x86_64                   libSM-1.2.3-8.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64                    libX11-common-1.7.2-3.amzn2023.0.4.noarch
  libX11-devel-1.7.2-3.amzn2023.0.4.x86_64              libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64                    libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64                   libXpm-3.5.15-2.amzn2023.0.3.x86_64
  libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64             libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64                     libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64              libicu-67.1-7.amzn2023.0.3.x86_64
  libicu-devel-67.1-7.amzn2023.0.3.x86_64               libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64       libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64                libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
  libselinux-devel-3.4-5.amzn2023.0.2.x86_64            libsepol-devel-3.4-3.amzn2023.0.3.x86_64
  libtiff-4.4.0-4.amzn2023.0.18.x86_64                  libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64                   libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64                   libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64            pcre2-devel-10.40-1.amzn2023.0.3.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64               pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
  pixman-0.40.0-3.amzn2023.0.3.x86_64                   sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch               xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
```

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-172-31-39-90 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
```

6. Create a new user group

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo groupadd nagcmd
```

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-172-31-39-90 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

8. Create a new directory for Nagios downloads

```
mkdir ~/downloads
cd ~/downloads
```

```
[ec2-user@ip-172-31-39-90 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Use wget to download the source zip files.

```
wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
```

```
[ec2-user@ip-172-31-39-90 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-04 03:45:50--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.92.120.28, 34.237.219.119, 52.54.96.194, ...
Connecting to go.nagios.org (go.nagios.org)|3.92.120.28|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Dow
utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51--  http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&
=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Do
&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org
t=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx                          100%[=====================================================>]   1.97M  7.32MB/s    i

2024-10-04 03:45:51 (7.32 MB/s) - '6kqcx' saved [2065473/2065473]

[ec2-user@ip-172-31-39-90 downloads]$
```

```
Wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
```

```
[ec2-user@ip-172-31-39-90 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 03:47:43--  http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.4.11.tar.gz’

nagios-plugins-2.4.11.tar.gz      100%[=============================================================>]   2.62M

2024-10-04 03:47:44 (6.85 MB/s) - ‘nagios-plugins-2.4.11.tar.gz’ saved [2753049/2753049]
```

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

```
tar zxvf 6kqcx
```

```
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
```

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-4.5.5
```

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

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

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ ./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 the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
```

error:

```
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:16:59 ago on Fri Oct  4 03:35:02 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture           Version                    Repository                  Size
================================================================================
Installing:
 openssl-devel           x86_64                 1:3.0.8-1.amzn2023.0.14    amazonlinux                3.0 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm                  19 MB/s | 3.0 MB     00:00
--------------------------------------------------------------------------------
Total                                                           15 MB/s | 3.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64           1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64           1/1
  Verifying        : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64           1/1

Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64

Complete!
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ ./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 the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for getopt.h... yes
checking for grp.h... yes
```

```
Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

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

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


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

12. Compile the source code.

```
make all
```

```
*** Compile finished ***

If the main program and CGIs compiled without any errors, you
can continue with testing or installing Nagios as follows (type
'make' without any arguments for a list of all possible options):

  make test
     - This runs the test suite

  make install
     - This installs the main program, CGIs, and HTML files

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-daemoninit
     - This will initialize the init script
       in /lib/systemd/system
```

```
*** 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
```

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

```
sudo nano /usr/local/nagios/etc/objects
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
       /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
```

```
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc
```

```
*** 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 ***
```

```
cd nagios-plugins-2.4.11
```

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ ./configure
      --with-nagios-user=nagios --with-nagios-group=nagios
```

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
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 gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
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
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
```

15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ 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 ***
```

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-172-31-39-90 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

17. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

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

```
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
nagios-plugins-2.4.11/gl/m4/
```

20. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
```

(error)

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
```

```
sudo chkconfig nagios on
```

Verify the sample configuration files

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

```
ec2-user@ip-172-31-39-90:~
[ec2-user@ip-172-31-39-90 ~]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
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.
        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 1 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
```

```
ec2-user@ip-172-31-39-90:~/
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
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.
        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 1 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
```

```
sudo service nagios start
```

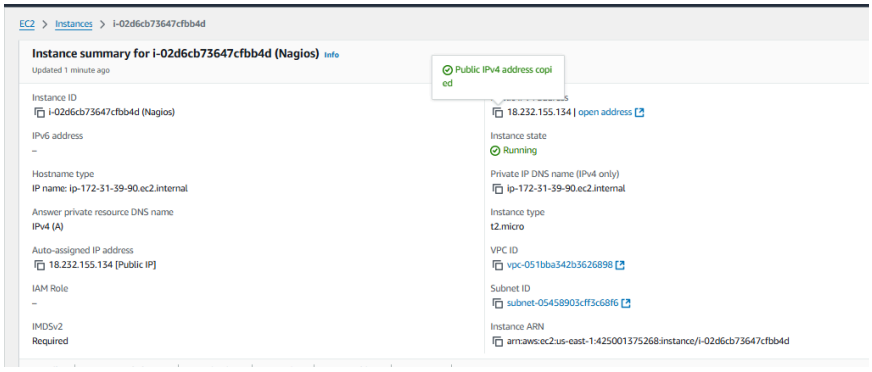
```
[ec2-user@ip-172-31-39-90 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

21. Check the status of Nagios

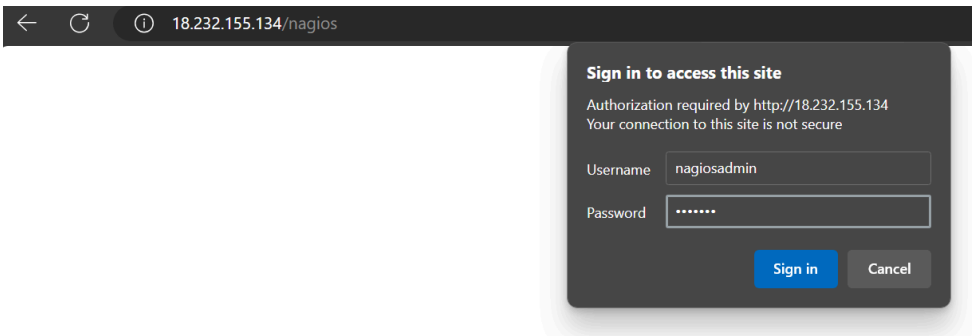
```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 1min 41s ago
       Docs: https://www.nagios.org/documentation
    Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 75300 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.6M
        CPU: 89ms
     CGroup: /system.slice/nagios.service
             ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Successfully registered manager as @wproc with query handler
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75301;pid=75301
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75302;pid=75302
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75304;pid=75304
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75303;pid=75303
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: Successfully launched command file worker with pid 75305
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(/v
Oct 04 04:15:06 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;1;(No output on stdo
Oct 04 04:15:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(/v
Oct 04 04:15:44 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Users;CRITICAL;HARD;1;(No output on std
[ec2-user@ip-172-31-39-90 ~]$
```

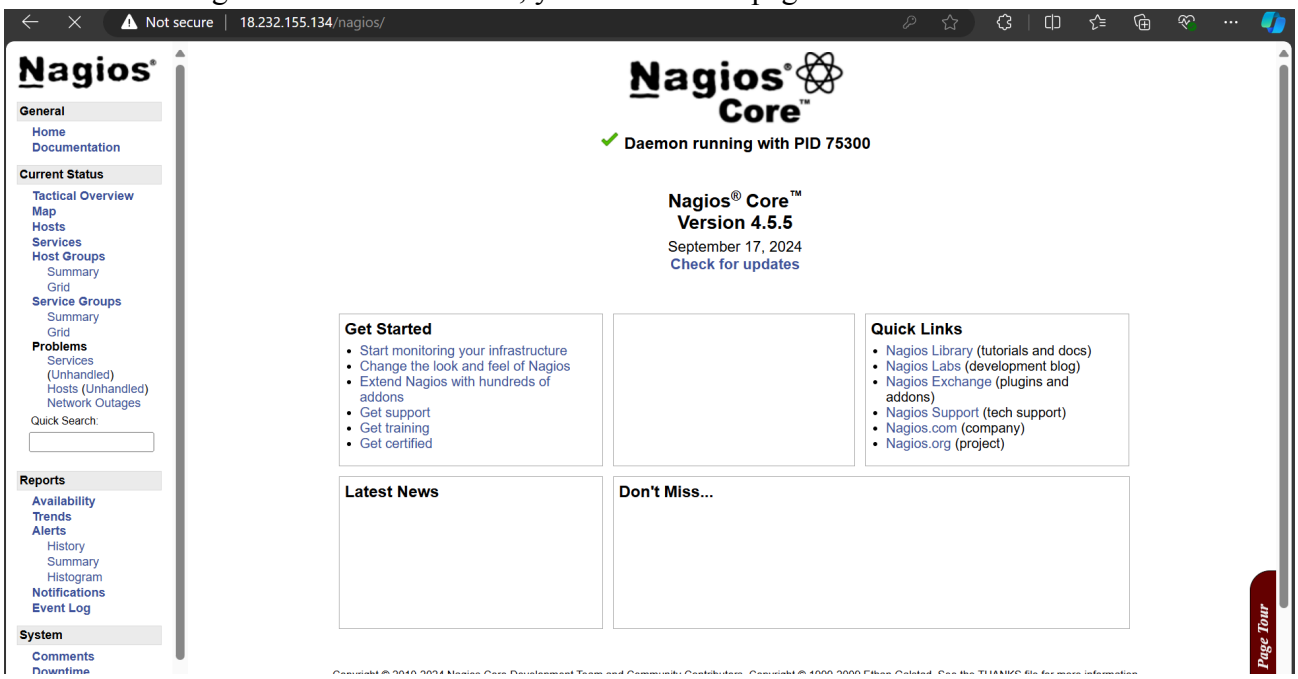
22. Go back to EC2 Console and copy the Public IP address of this instance



23. Open up your browser and look for **http://<your_public_ip_address>/nagios**
    Enter username as nagiosadmin and password which you set in Step 16.



24. After entering the correct credentials, you will see this page.



This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion:

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

Theory:

Port and Service Monitoring
Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

Windows/Linux Server Monitoring
Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

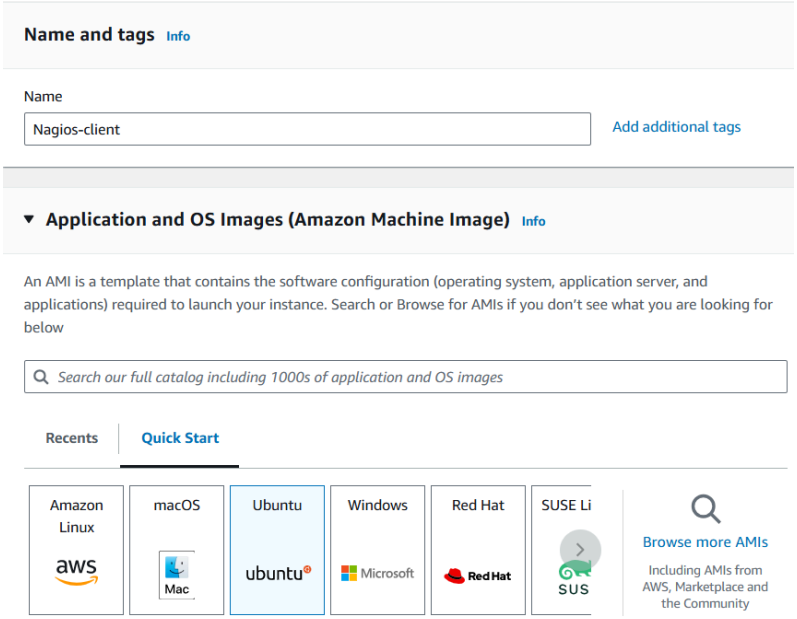
Prerequisites:
AWS Academy or Personal account.
Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring Using Nagios:

Step 1: To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host).
sudo systemctl status nagios

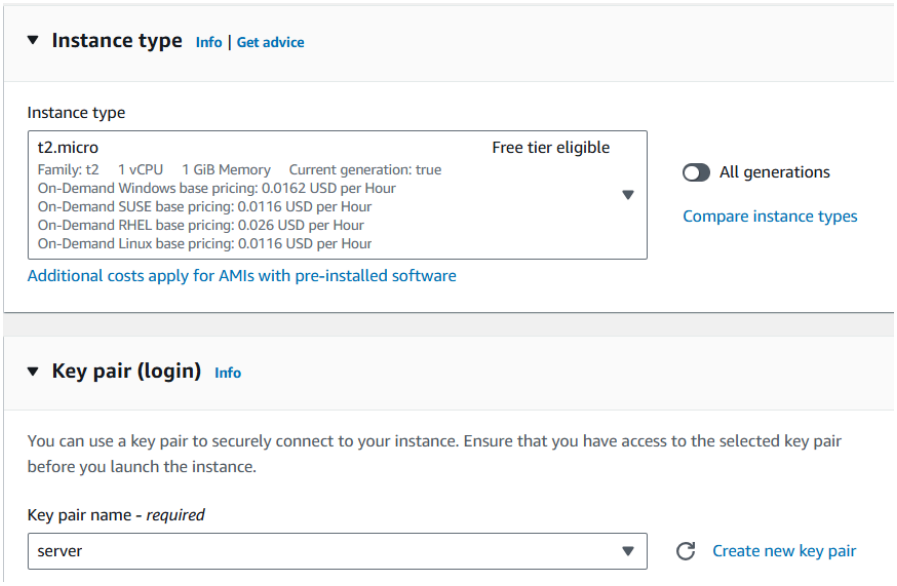
```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 9min ago
       Docs: https://www.nagios.org/documentation
    Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 75300 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.6M
        CPU: 164ms
     CGroup: /system.slice/nagios.service
             ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:17:36 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Root Partition;CRITICAL;HARD;1;(No output on st>
Oct 04 04:18:14 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;SSH;CRITICAL;HARD;1;(No output on stdout) stder>
Oct 04 04:18:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;5;(No output on stdout) stderr: execvp(/>
Oct 04 04:18:51 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;1;(No output on stdout>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Total Processes;CRITICAL;HARD;1;(No output on s>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(/>
Oct 04 04:20:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;7;(No output on stdout) stderr: execvp(/>
Oct 04 04:21:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(/>
Oct 04 04:22:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(/>
Oct 04 04:23:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(>
lines 1-28/28 (END)
```

You can now proceed if you get the above message/output.
Step 2: Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

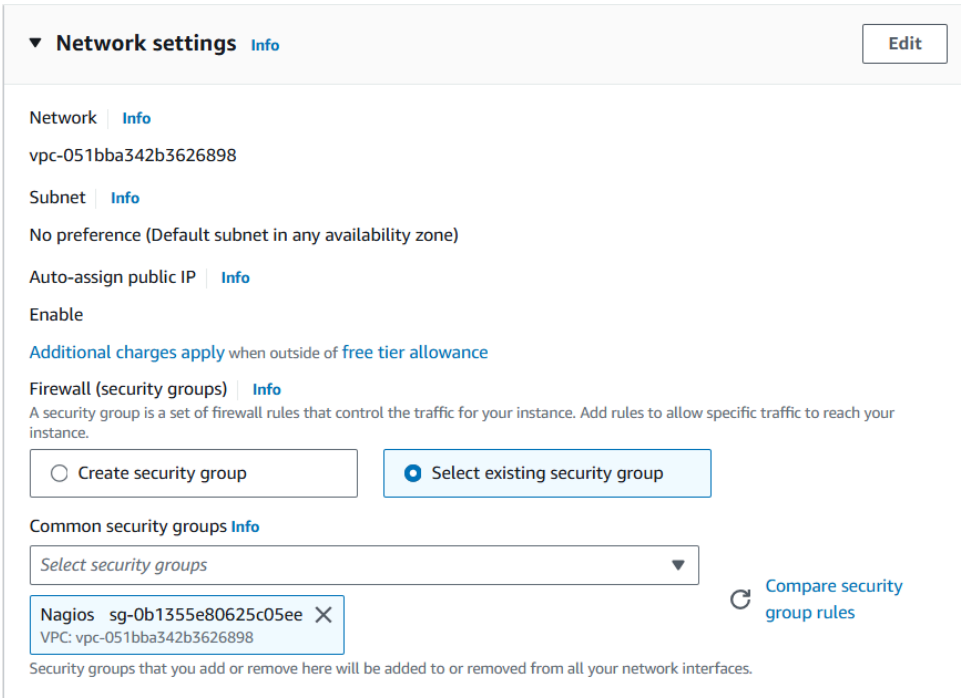


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

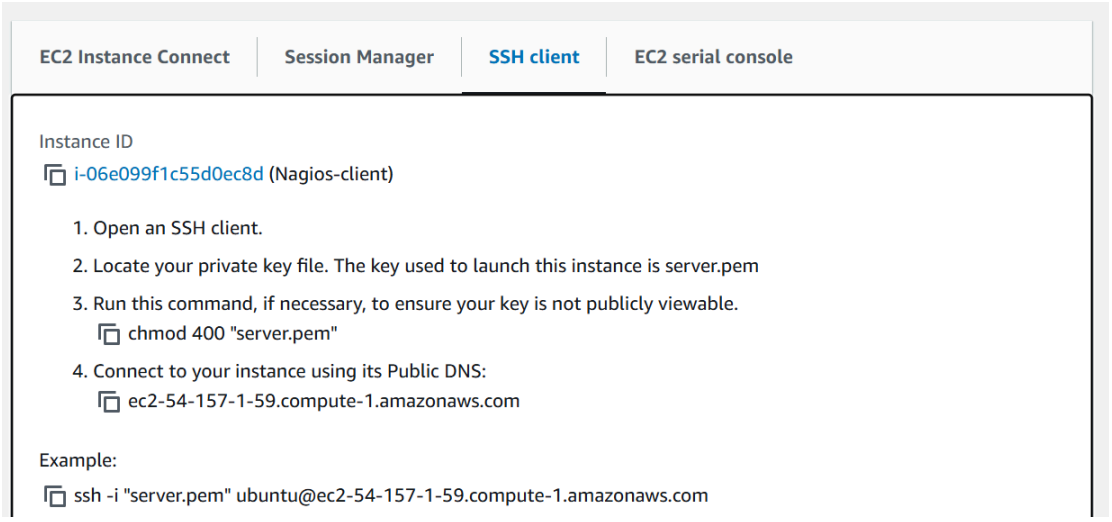
Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .
Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.



Successfully connected to the instance.

```
PS C:\Users\Sadneya\downloads> ssh -i "server.pem" ec2-user@ec2-3-81-91-101.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-91-101.compute-1.amazonaws.com (3.81.91.101)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tozb5yML0ekn5Xzfrw.
This host key is known by the following other names/addresses:
    C:\Users\Sadneya/.ssh/known_hosts:68: ec2-18-232-155-134.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-91-101.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Oct  4 03:36:10 2024 from 125.99.93.18
```

Now perform all the commands on the Nagios-host till step 10

Step 4: Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-31-39-90 ~]$ ps -ef | grep nagios
ec2-user    2377    2350  0 09:15 pts/0    00:00:00 grep --color=auto nagios
```

Step 5: Now Become root user and create root directories.

sudo su

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

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

```
[ec2-user@ip-172-31-39-90 ~]$ sudo su
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

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

```
[root@ip-172-31-39-90 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linux
hosts/linuxserver.cfg
```

Step 7: Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-39-90 ec2-user]#
```

Change `hostname` to **linuxserver**.

Change `address` to the public IP of your Linux client.

Set `hostgroup_name` to **linux-servers1**.

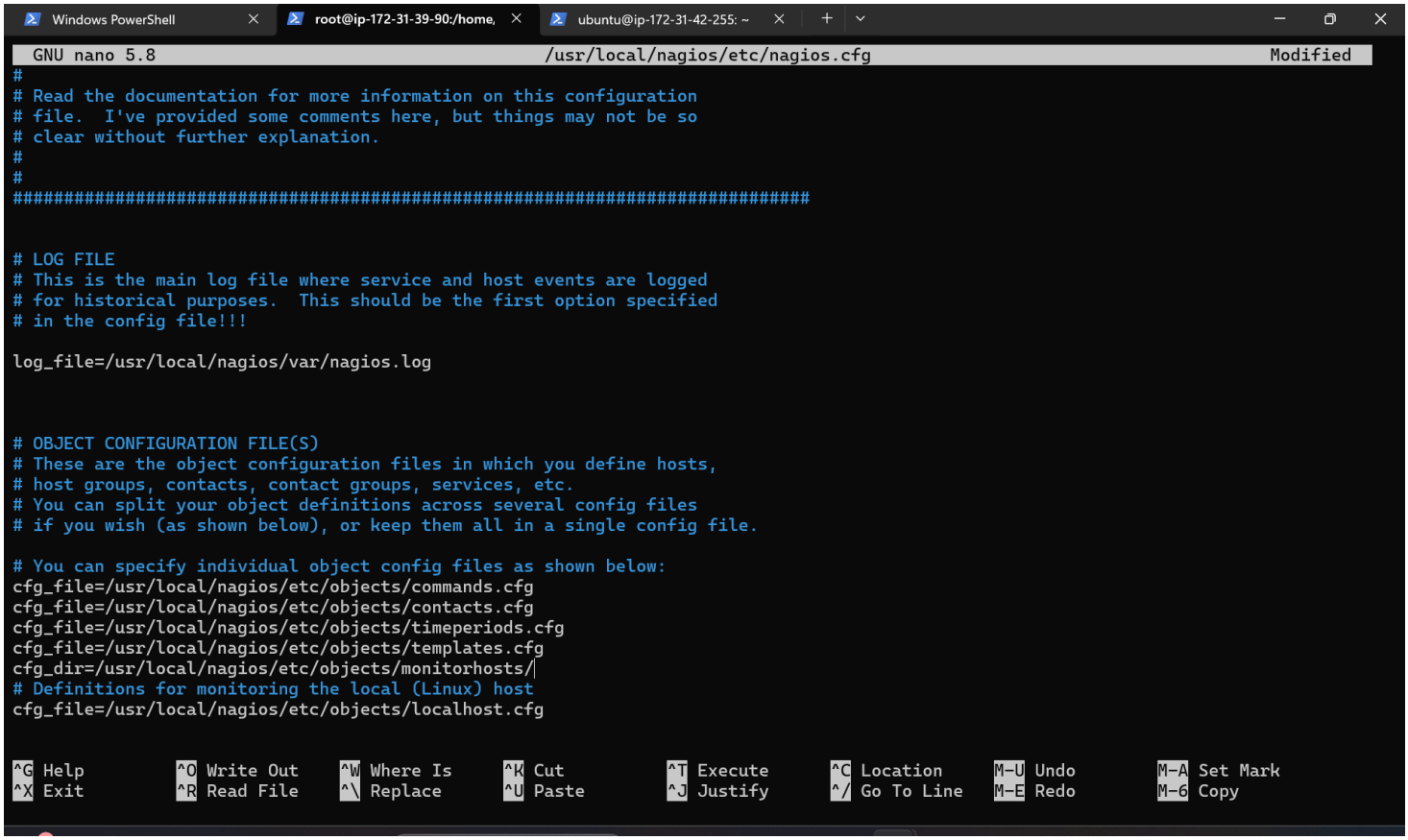
nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

Windows PowerShell | root@ip-172-31-39-90:/home, | ubuntu@ip-172-31-42-255: ~

```
  GNU nano 5.8                  /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg                Modified
##############################################################################

# Define a host for the local machine

define host {

    use                     linux-server            ; Name of host template to use
                                                    ; This host definition will inherit all variables that are defined
                                                    ; in (or inherited by) the linux-server host template definition.
    host_name               linuxserver
    alias                   localhost
    address                 98.83.6.103
}



##############################################################################
#
# HOST GROUP DEFINITION
#
##############################################################################

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name          linux-servers1          ; The name of the hostgroup
    alias                   Linux Servers           ; Long name of the group
    members                 localhost               ; Comma separated list of hosts that belong to this group
}



^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

Step 8: Now update the Nagios config file .Add the following line in the file.

Line to add : cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
Run the command : **nano /usr/local/nagios/etc/nagios.cfg**



Step 9: Now Verify the configuration files by running the following commands.
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-39-90 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 152)
Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 138)
Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 125)
Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 112)
Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 100)
Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 86)
Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 72)
Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 58)
   Read object config files okay...

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-39-90 ec2-user]#
```

Step 10: Now restart the services of nagios by running the following command.
service nagios restart

```
[root@ip-172-31-39-90 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.
sudo apt update -y

```
ubuntu@ip-172-31-42-255:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

```
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 6s (4883 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo apt install gcc -y

```
ubuntu@ip-172-31-42-255:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
  fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1
  libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
  libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
  libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
  libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
  binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc
  gcc-13-multilib gcc-13-doc gdb-x86-64-linux-gnu glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec
  libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e
  libheif-plugin-svtenc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
  fonts-dejavu-core fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8
  libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
  libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
  libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
  libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libsframe1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbinutils amd64 2.42-4ubuntu2 [572 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]
```

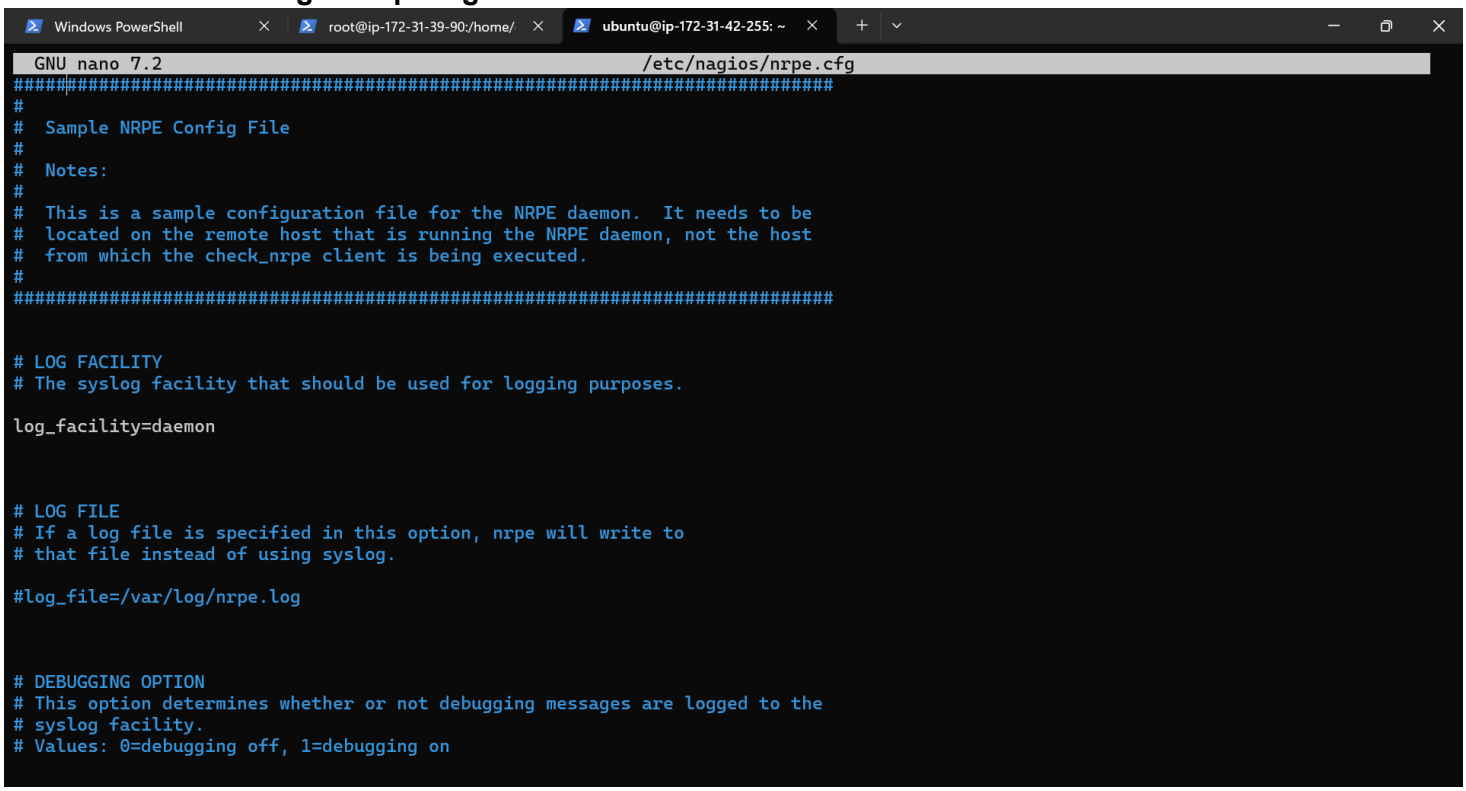
```
Setting up gcc (4:13.2.0-7ubuntu1) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.15.0-1.1ubuntu2) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~$
```

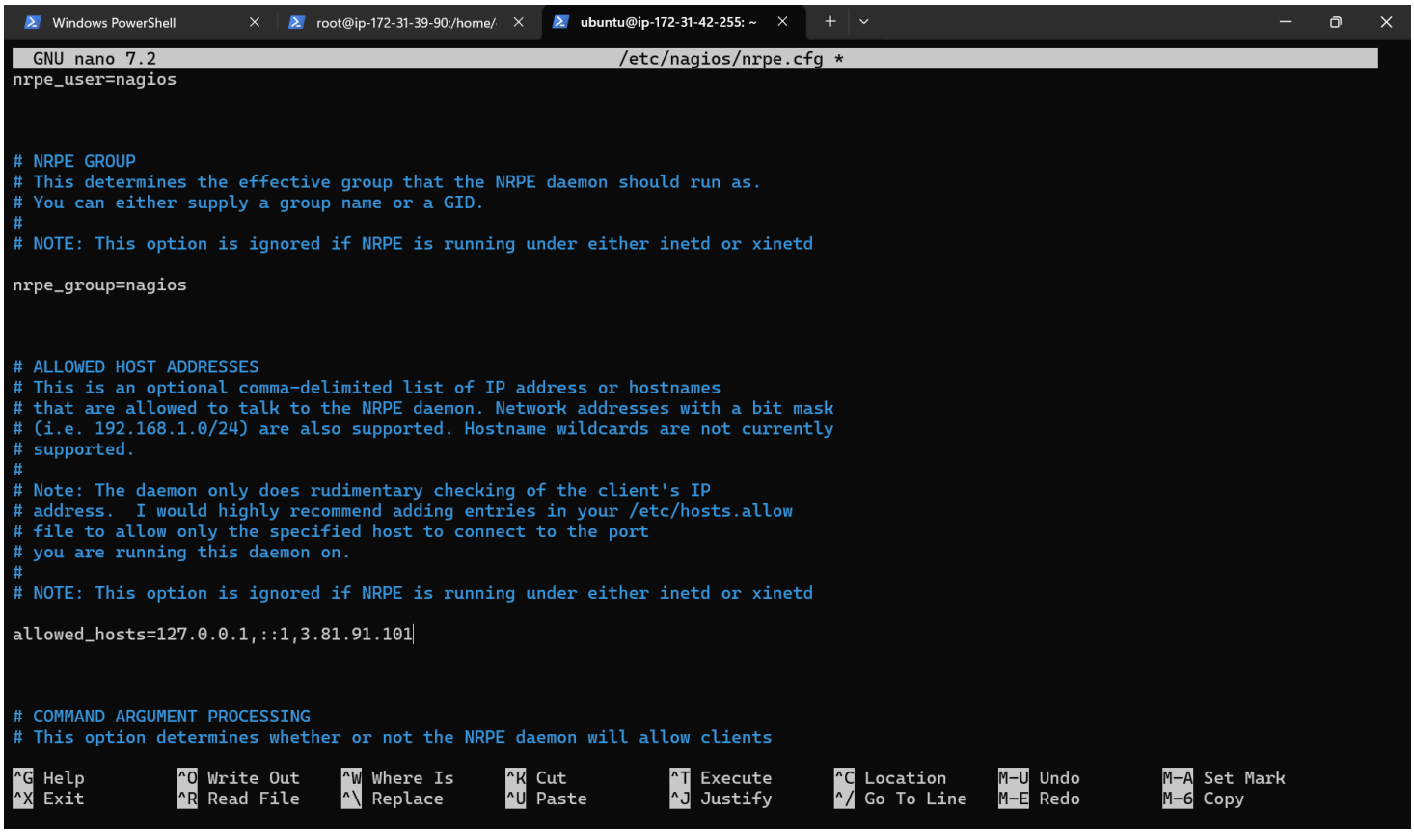
sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-42-255:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmbclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common python3-gpg python3-ldb
  python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs
  smbclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib
  fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients
  python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmbclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg
  python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules
  samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-client3 amd64 0.8-13ubuntu6 [26.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcups2t64 amd64 2.4.7-1.2ubuntu7.3 [272 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdbi1t64 amd64 0.9.0-6.1build1 [25.7 kB]
```

```
Creating config file /etc/nagios-plugins/config/netware.cfg with new version

Creating config file /etc/nagios-plugins/config/nt.cfg with new version

Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version

Creating config file /etc/nagios-plugins/config/radius.cfg with new version

Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version

Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~$
```

Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address.
sudo nano /etc/nagios/nrpe.cfg





Step 13: Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server

```
ubuntu@ip-172-31-42-255:~$ sudo systemctl restart nagios-nrpe-server
```

Step 14: Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

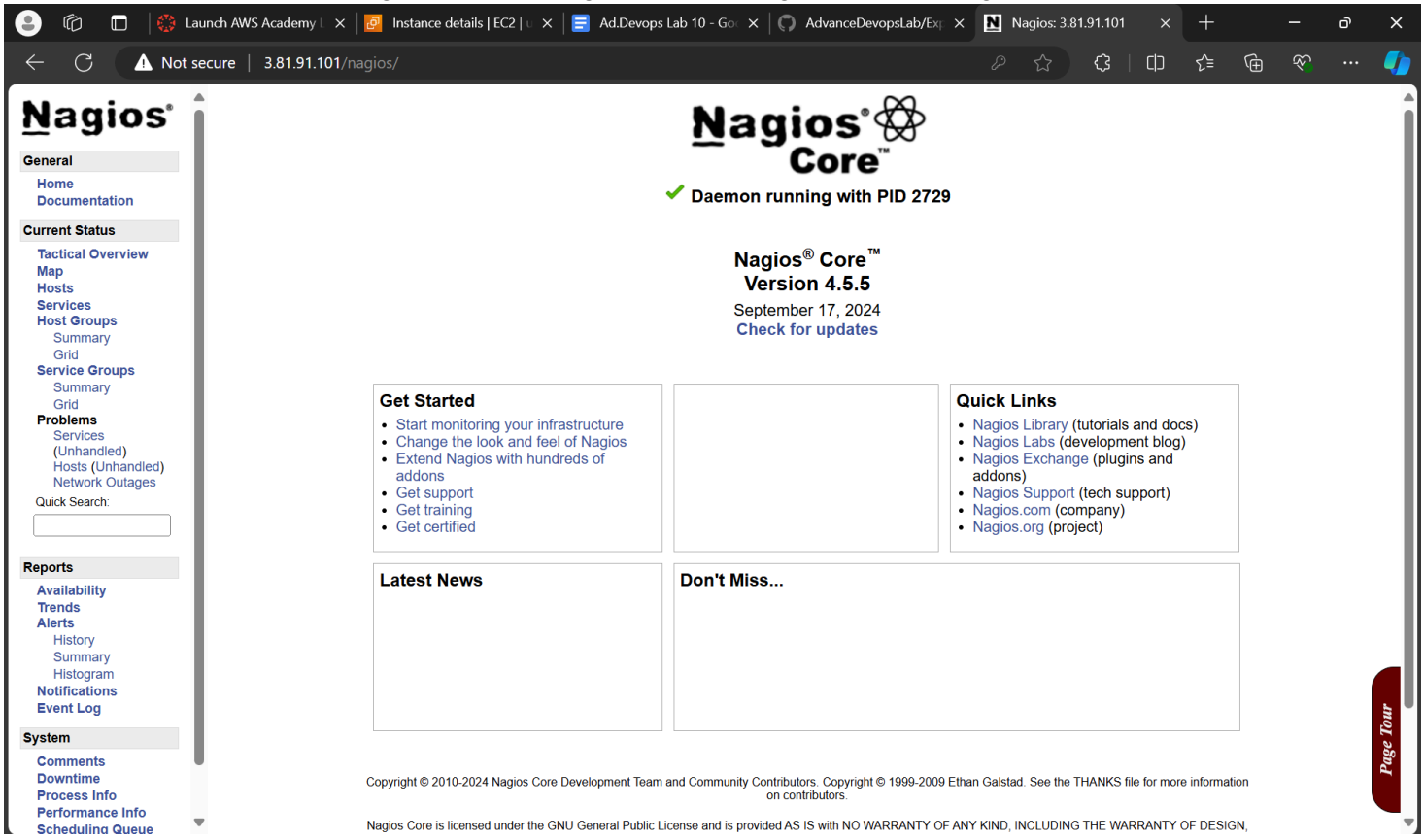
sudo systemctl status nagios

```
[root@ip-172-31-39-90 ec2-user]#
sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 09:24:28 UTC; 9min ago
       Docs: https://www.nagios.org/documentation
    Process: 2725 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCC>
    Process: 2727 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 2729 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.2M
        CPU: 114ms
     CGroup: /system.slice/nagios.service
             ├─2729 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─2730 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2731 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2732 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2733 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─2742 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 09:30:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;7;(No output on stdout) stderr>
Oct 04 09:31:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;8;(No output on stdout) stderr>
Oct 04 09:32:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;9;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST NOTIFICATION: nagiosadmin;linuxserver;DOWN;notify-host-by-e>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;HARD;10;(No output on stdout) stder>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: NOTIFY job 6 from worker Core Worker 2732 is a non-check >
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   host=linuxserver; service=(none); contact=nagiosadmin
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   stderr line 01: /bin/sh: line 1: /bin/mail: No such fil>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pi>
[root@ip-172-31-39-90 ec2-user]#
```

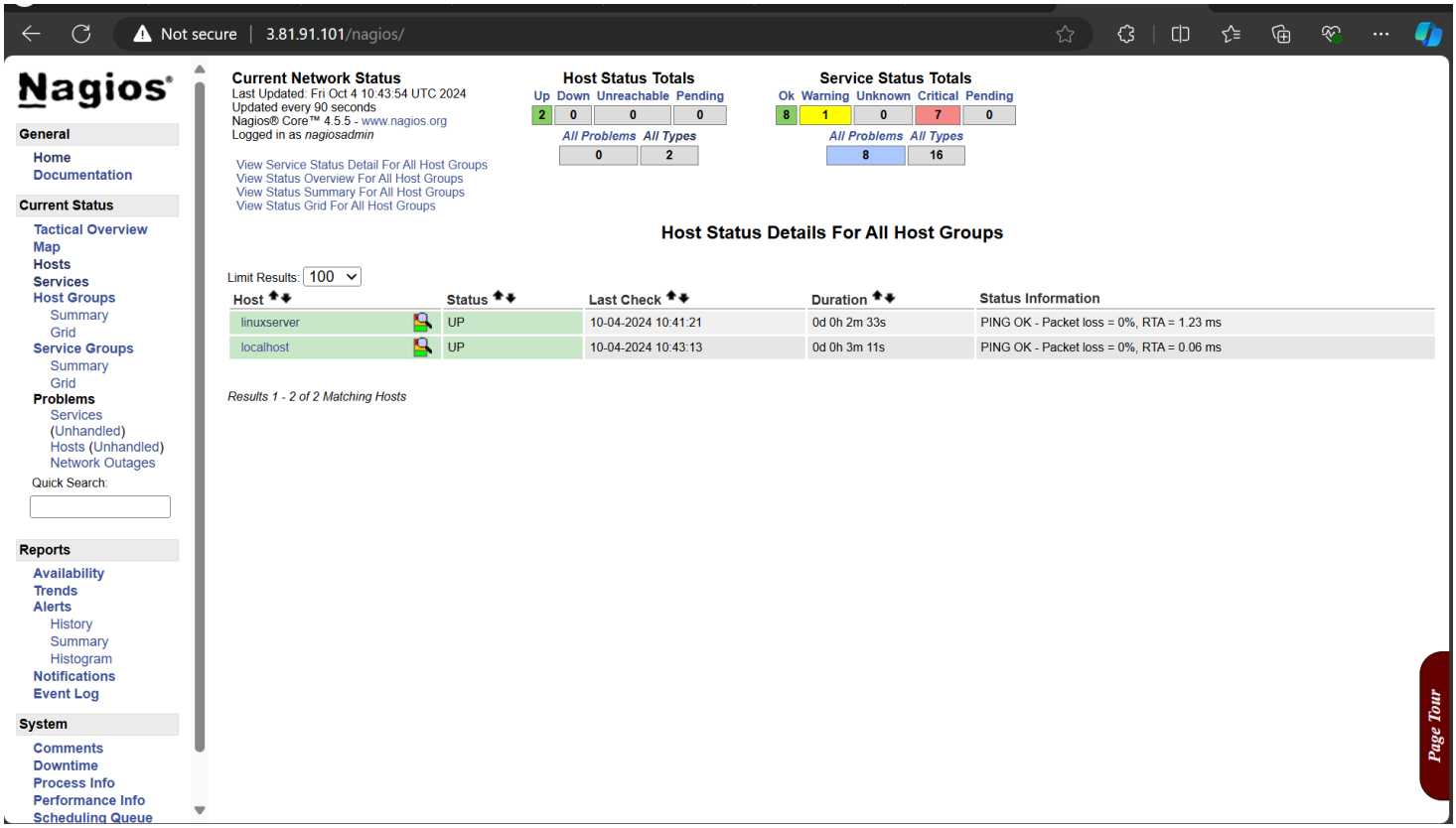
sudo systemctl status httpd
sudo systemctl start httpd
sudo systemctl enable httpd

```
[root@ip-172-31-39-90 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2024-10-04 09:45:14 UTC; 33min ago
       Docs: man:httpd.service(8)
   Main PID: 4146 (httpd)
     Status: "Total requests: 19; Idle/Busy workers 100/0;Requests/sec: 0.0095; Bytes served/sec:  70 B/sec"
      Tasks: 230 (limit: 1112)
     Memory: 17.3M
        CPU: 1.428s
     CGroup: /system.slice/httpd.service
             ├─4146 /usr/sbin/httpd -DFOREGROUND
             ├─4148 /usr/sbin/httpd -DFOREGROUND
             ├─4149 /usr/sbin/httpd -DFOREGROUND
             ├─4150 /usr/sbin/httpd -DFOREGROUND
             ├─4151 /usr/sbin/httpd -DFOREGROUND
             └─4533 /usr/sbin/httpd -DFOREGROUND

Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal httpd[4146]: Server configured, listening on: port 80
[root@ip-172-31-39-90 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-39-90 ec2-user]# sudo systemctl enable httpd
```

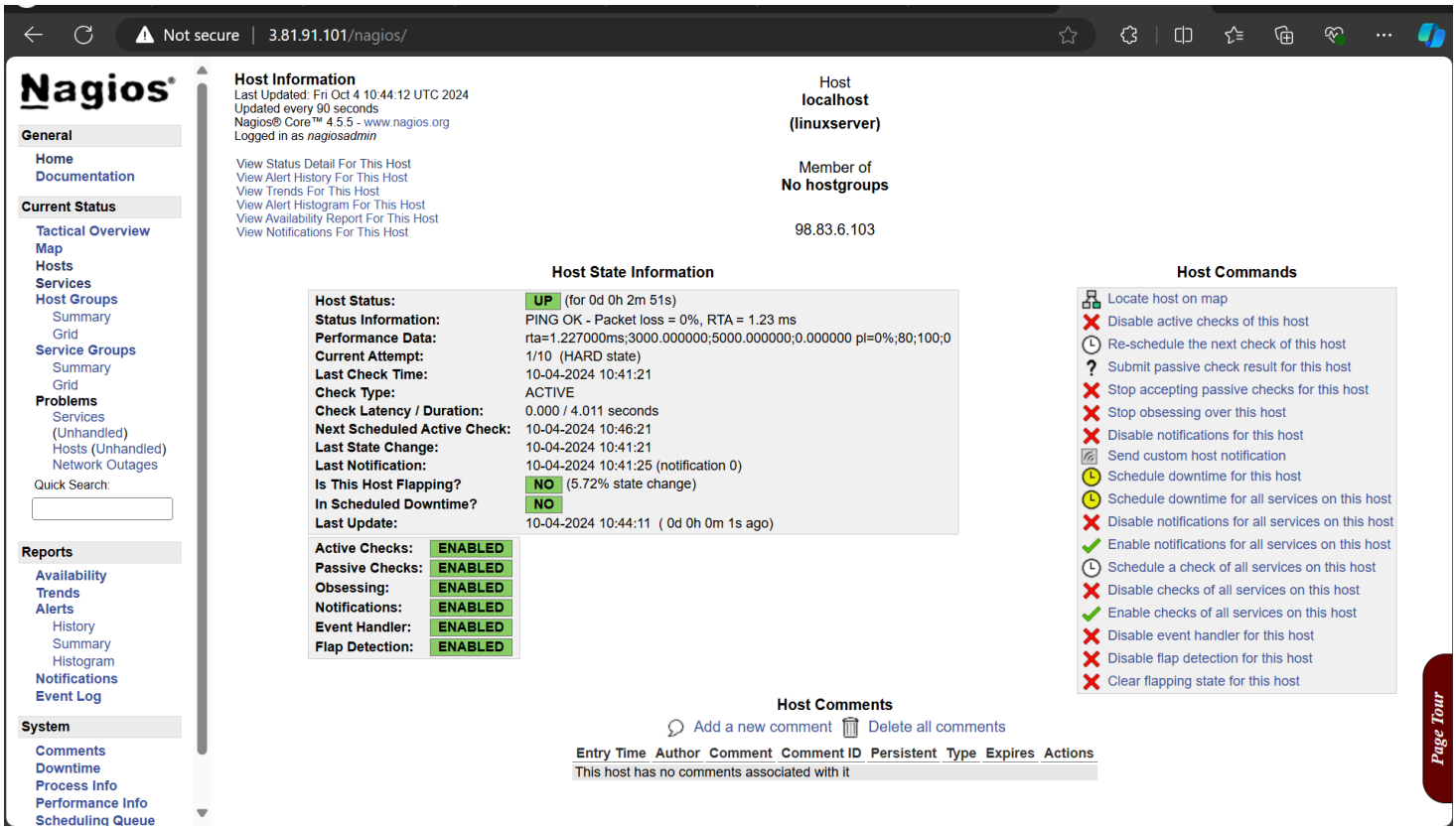
Step 15: Now to check Nagios dashboard go to http://<Nagios-host ip>/nagios .



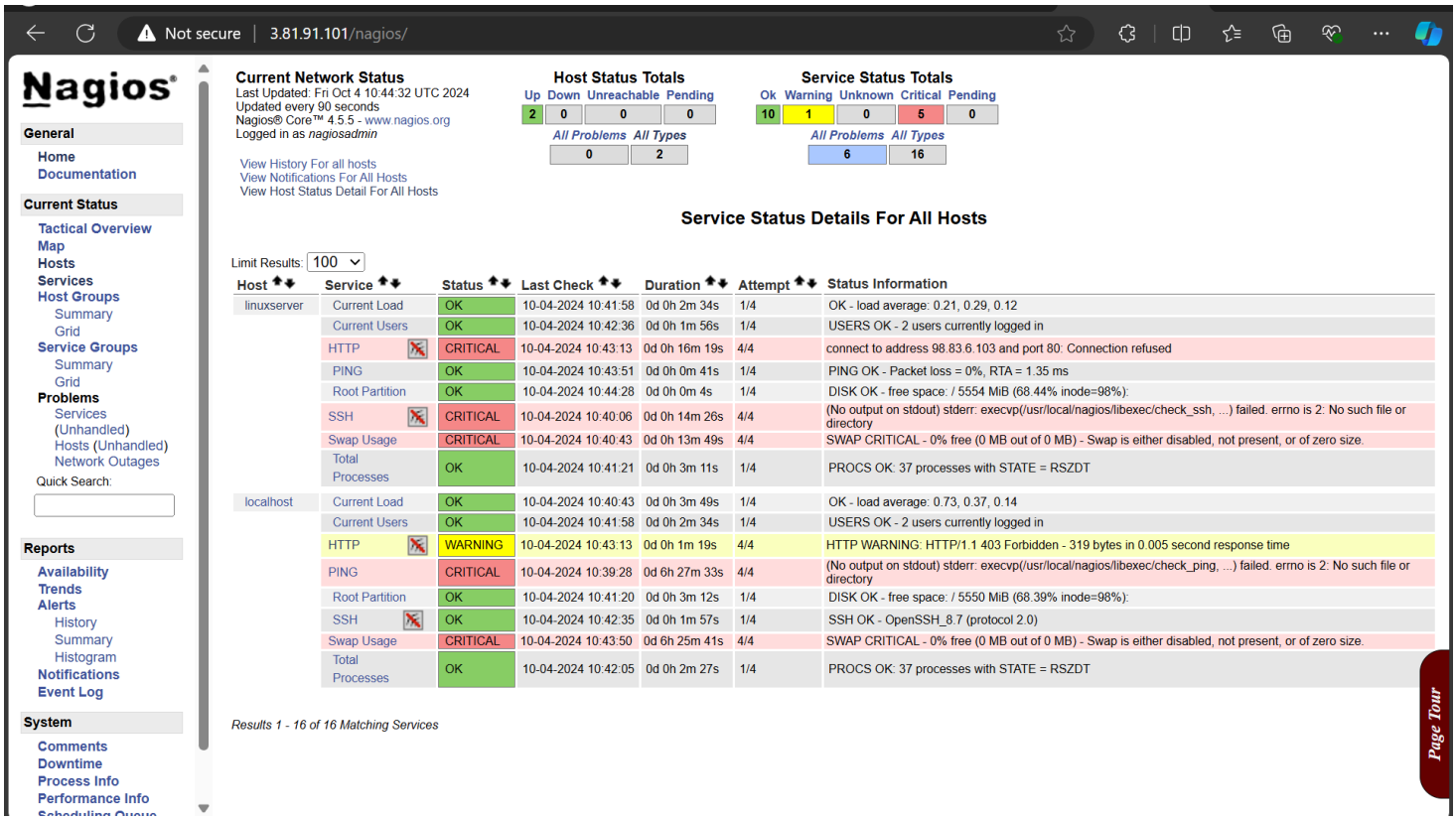
Now Click on Hosts from left side panel


Not secure | 3.81.91.101/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends
Alerts
History
Summary
Histogram
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Current Network Status
Last Updated: Fri Oct 4 10:43:54 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
8 1 0 7 0
All Problems All Types
8 16
Host Status Details For All Host Groups
Limit Results: 100
Host Status Last Check Duration Status Information
linuxserver UP 10-04-2024 10:41:21 0d 0h 2m 33s PING OK - Packet loss = 0%, RTA = 1.23 ms
localhost UP 10-04-2024 10:43:13 0d 0h 3m 11s PING OK - Packet loss = 0%, RTA = 0.06 ms
Results 1 - 2 of 2 Matching Hosts
Page Tour

We can see our linuxserver now click on it we can see the host information.



Current Network Status



Conclusion: In conclusion, the experiment focused on monitoring ports, services, and a Linux server using Nagios. Through the step-by-step process, we successfully configured Nagios to monitor

essential network services on the Linux server. By setting up both the Nagios host and client, we were able to track system performance, ensure service availability, and monitor key metrics like CPU and memory usage.

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

Theory:

AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

Lambda Workflow

- **Create a Function**: Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources**: Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution**: When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency**: Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring **reserved concurrency** to manage traffic.
- **Monitoring and Logging**: Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

AWS Lambda Functions

- **Python**: Great for quick development with its rich standard library and support for lightweight tasks.
- **Java**: Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js**: Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

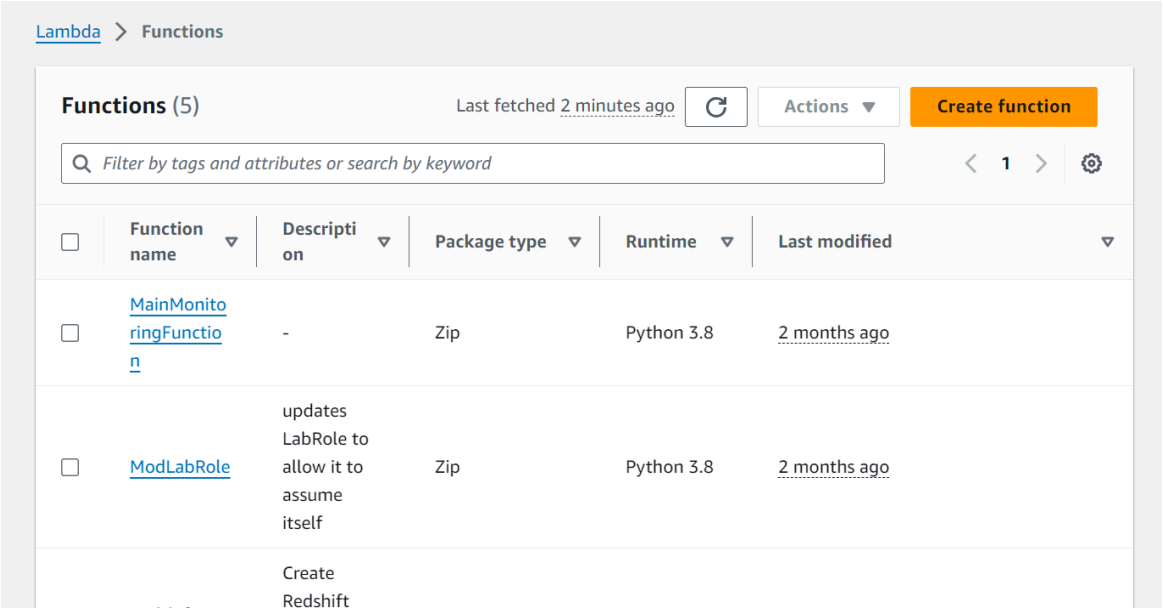
Prerequisites: AWS Personal/Academy Account

Name:Bhushan Mukund Kor **Academic Year:2024-2025**

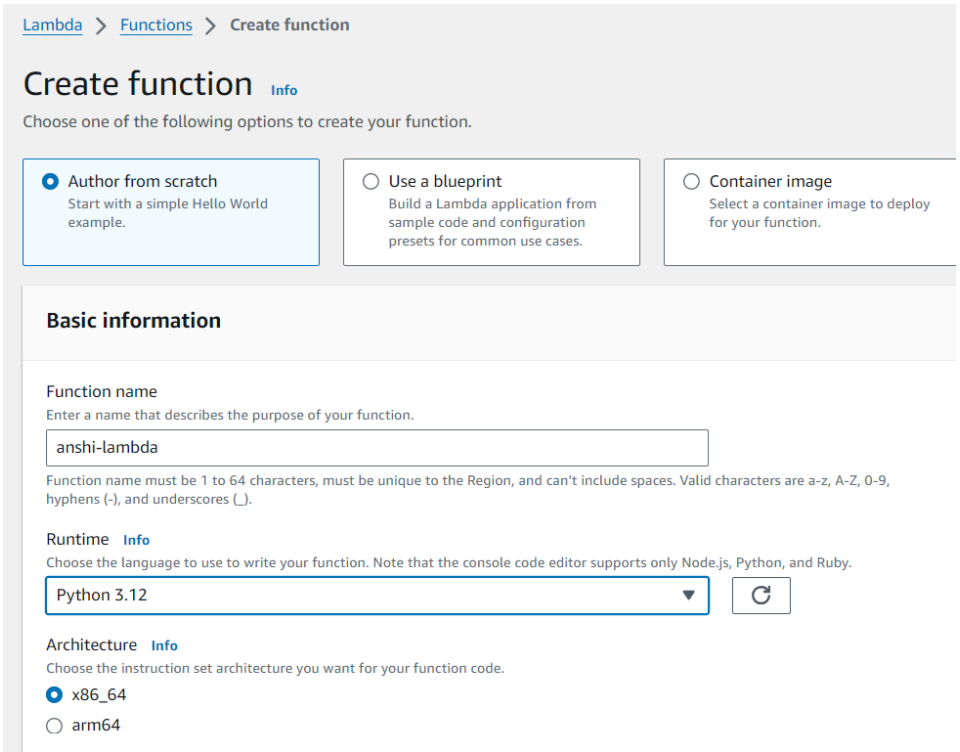
Division: D15C **Roll No: 28**

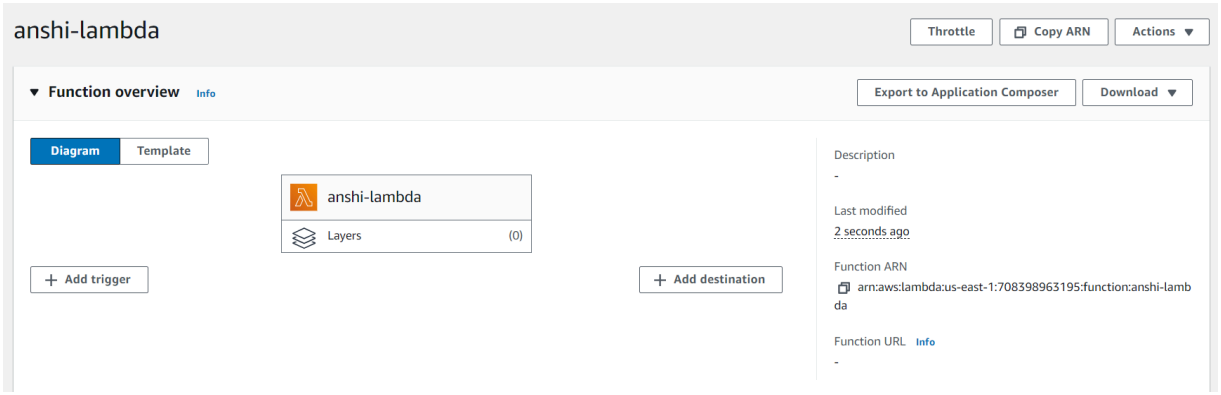
Steps To create the lambda function:

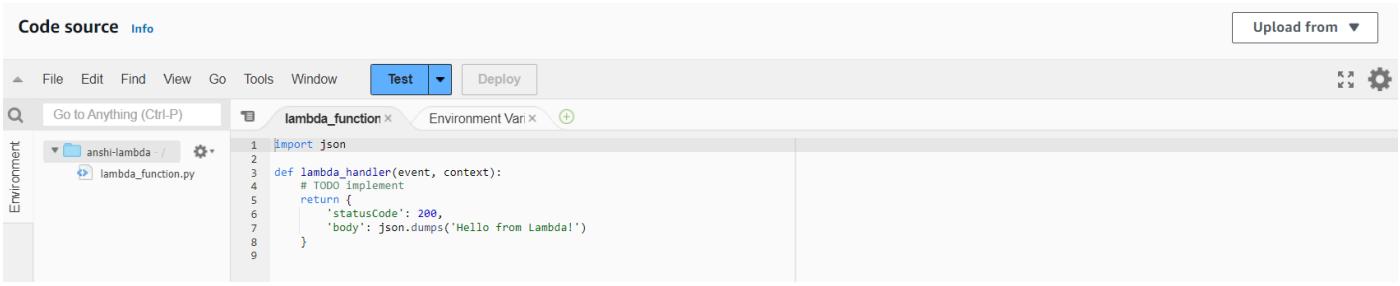
Step 1: Login to your AWS Personal/Academy Accout.Open lambda and click on create function button.



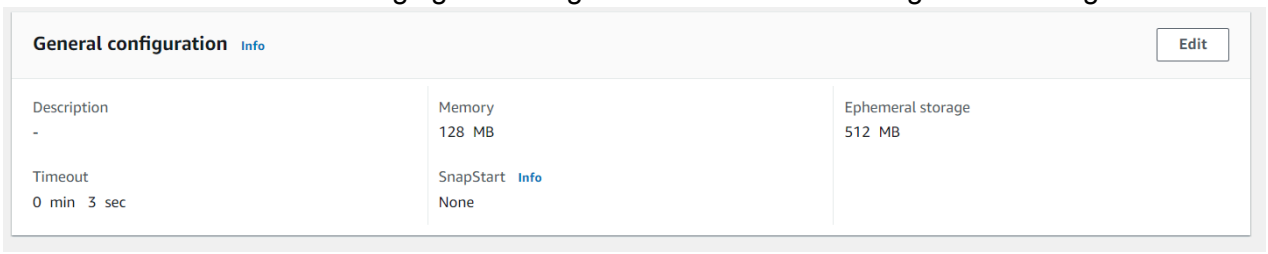
Step 2: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Exceution role to Create a new role with basic Lambda permissions.



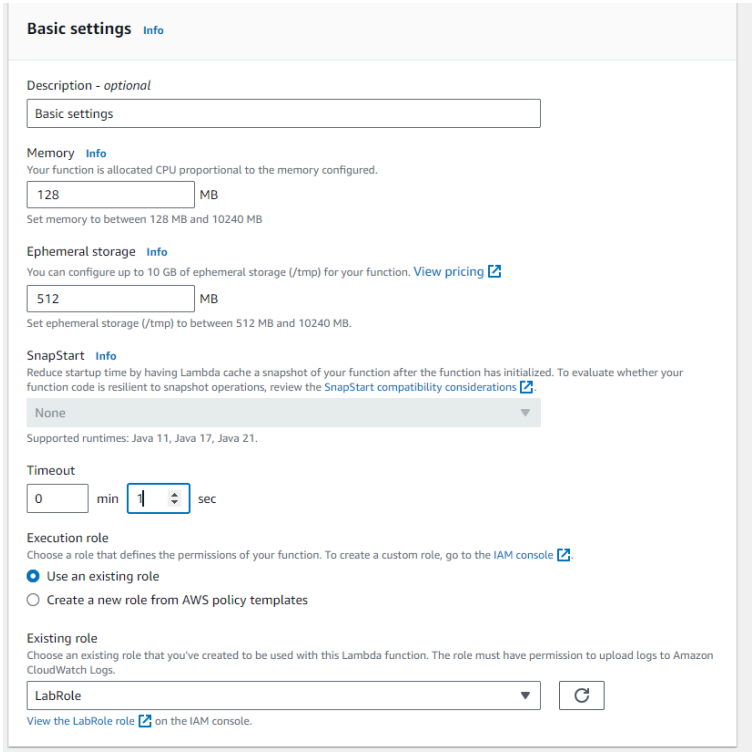




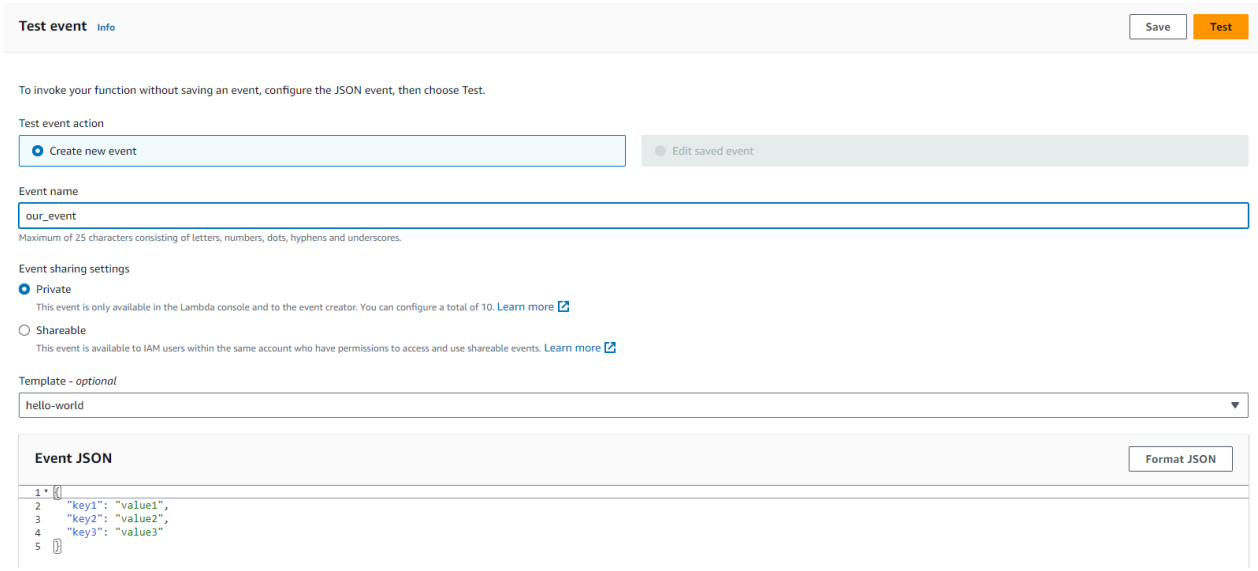
So See or Edit the basic settings go to configuration then click on edit general setting.



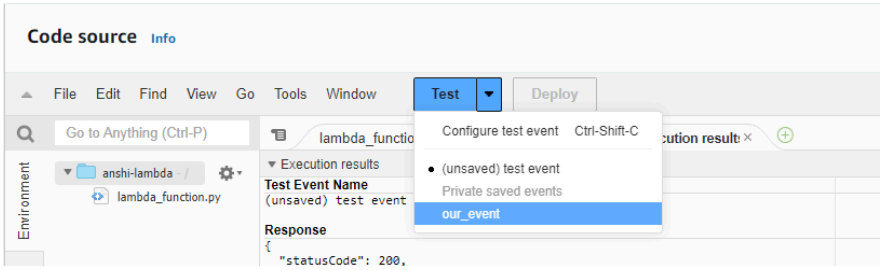
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

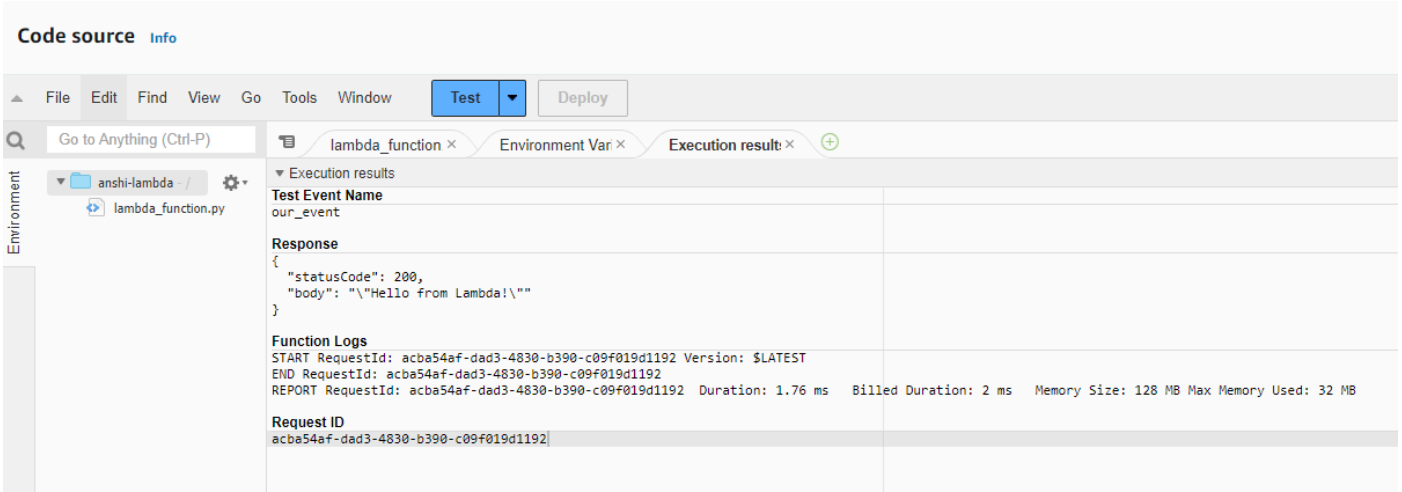


Step 3: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.

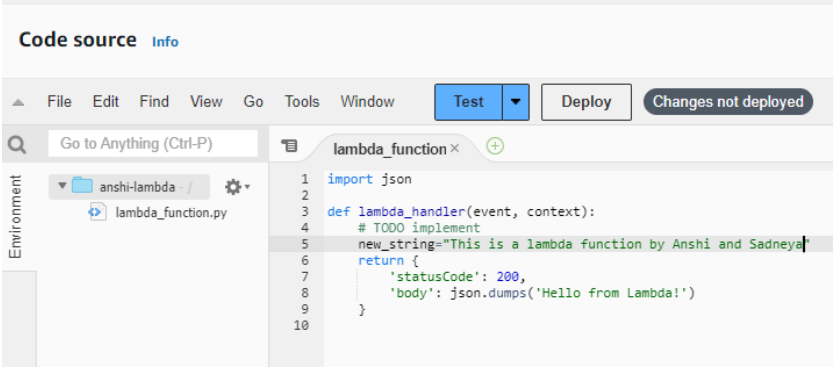


Step 4: Now In Code section select the created event from the dropdown of test then click on test . You will see the below output.

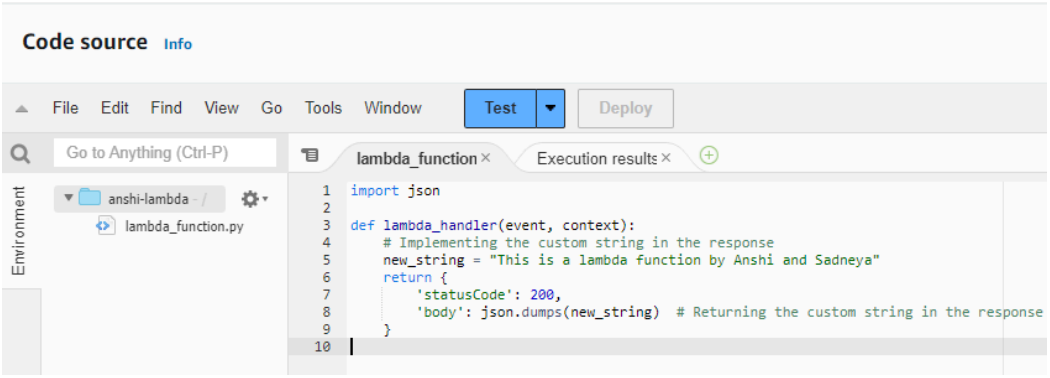




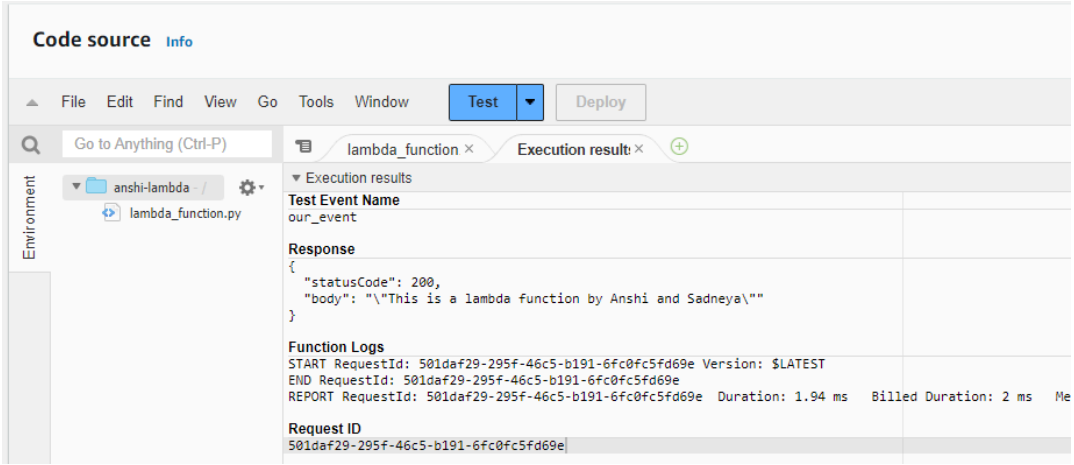
Step 5: You can edit your lambda function code. I have changed the code to display the new String.



Now ctrl+s to save and click on deploy to deploy the changes.



Step 6: Now click on the test and observe the output. We can see the status code 200 and your string output and function logs. On successful deployment.



Conclusion: In this experiment, we successfully created an AWS Lambda function and walked through its essential steps. After setting up the function with Python, we configured the basic settings, including adjusting the timeout to 1 second. We then created a test event, deployed the function, and validated the output. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time.

This practical experience demonstrated the simplicity and flexibility of AWS Lambda in creating serverless applications, allowing you to focus on code while AWS manages the infrastructure and scaling.

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

Theory:

AWS Lambda and S3 Integration:

AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:

1. Create an S3 Bucket:

○ First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

2. Create the Lambda Function:

○ Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.

○ Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

○ Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

4. Configure S3 Trigger:

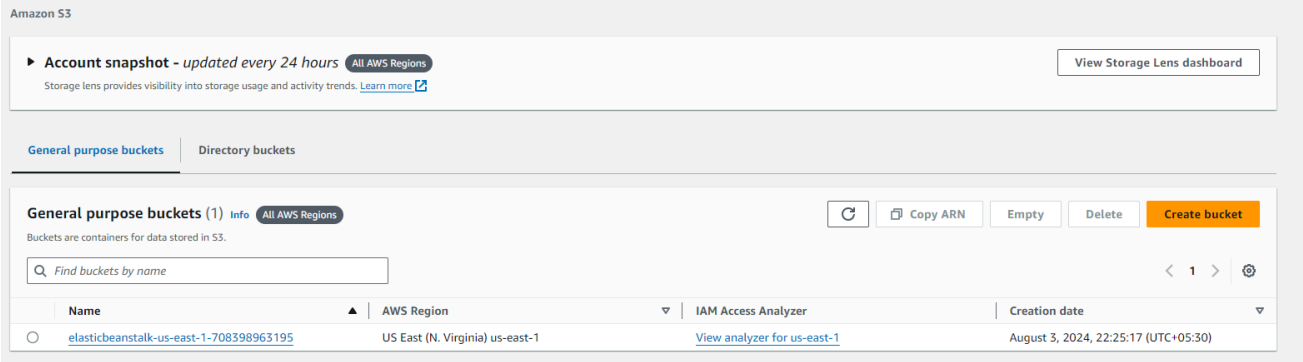
○ Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

5. Test the Setup:

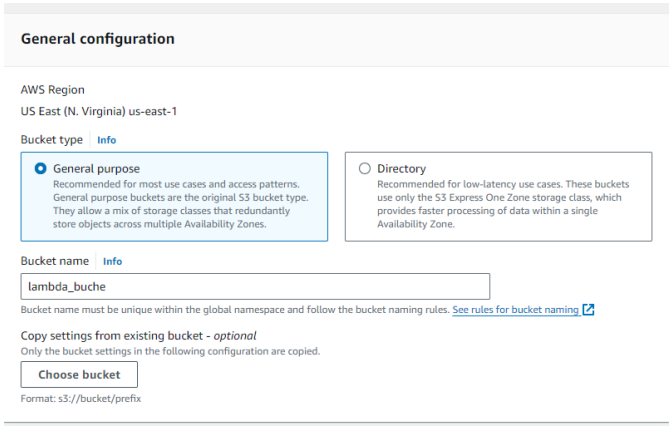
○ Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

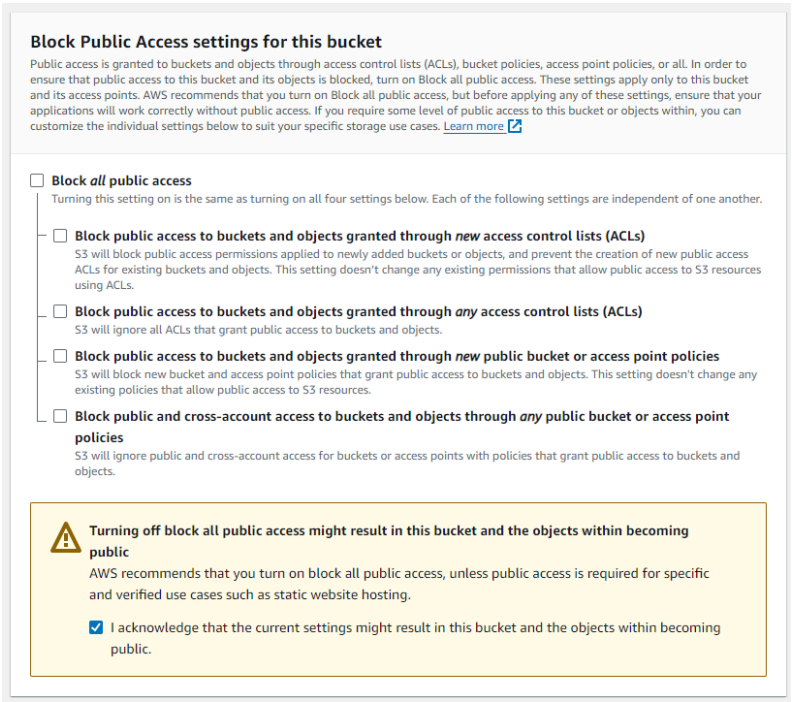
Steps To create the lambda function:

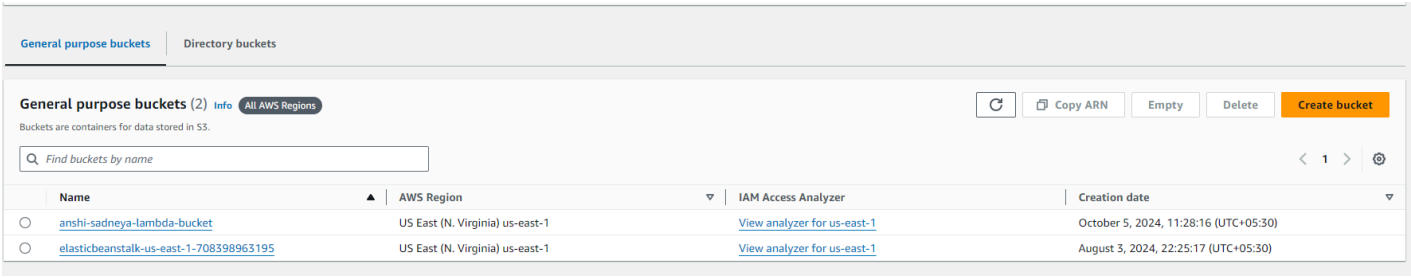
Step 1: Login to your AWS Personal account. Now open S3 from services and click on create S3 bucket.



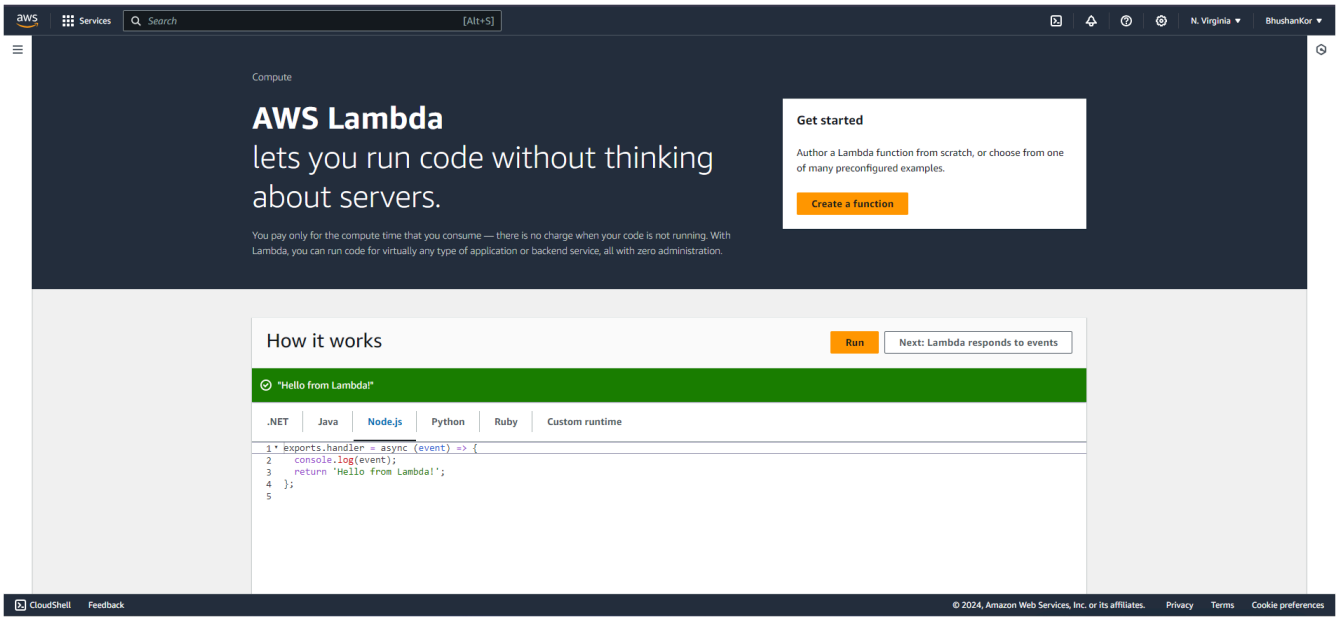
Step 2: Now Give a name to the Bucket, select general purpose project and deselect the Block public access and keep other this to default.



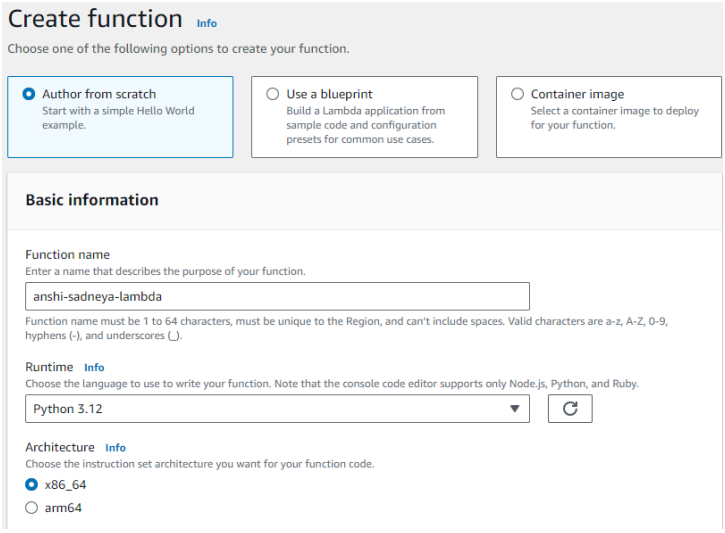


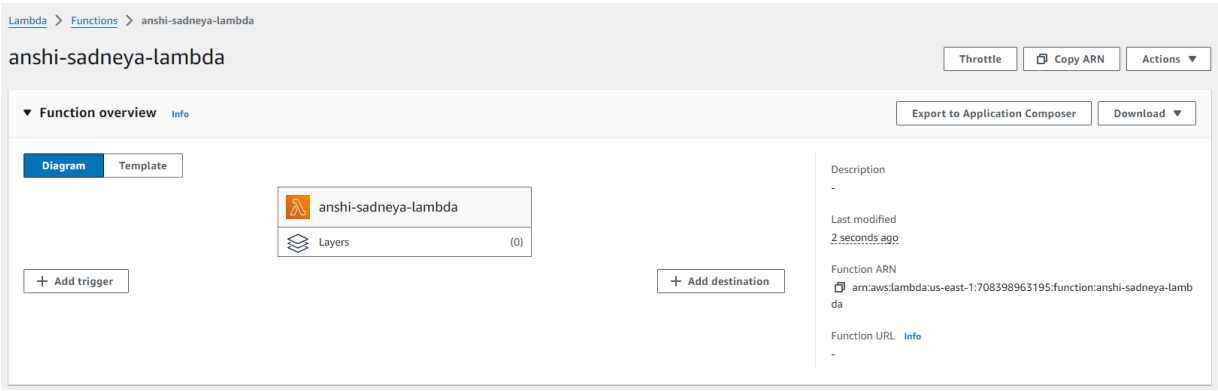


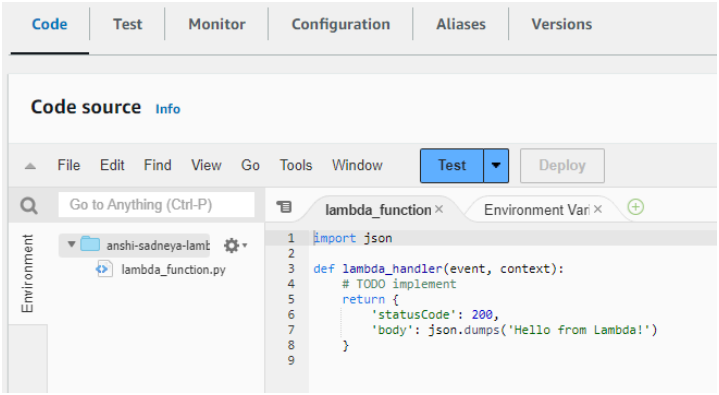
Step 3: Open lambda console and click on create function button.



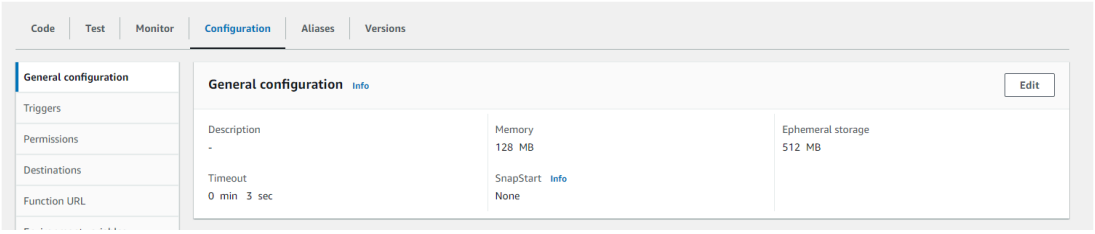
Step 4: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Exceution role to Create a new role with basic Lambda permissions.



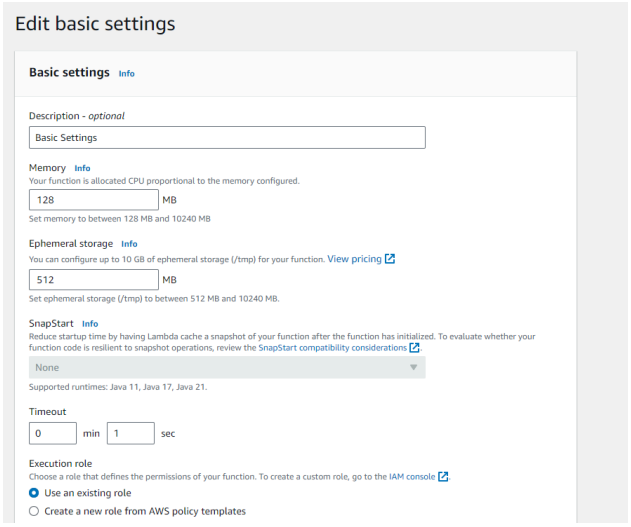




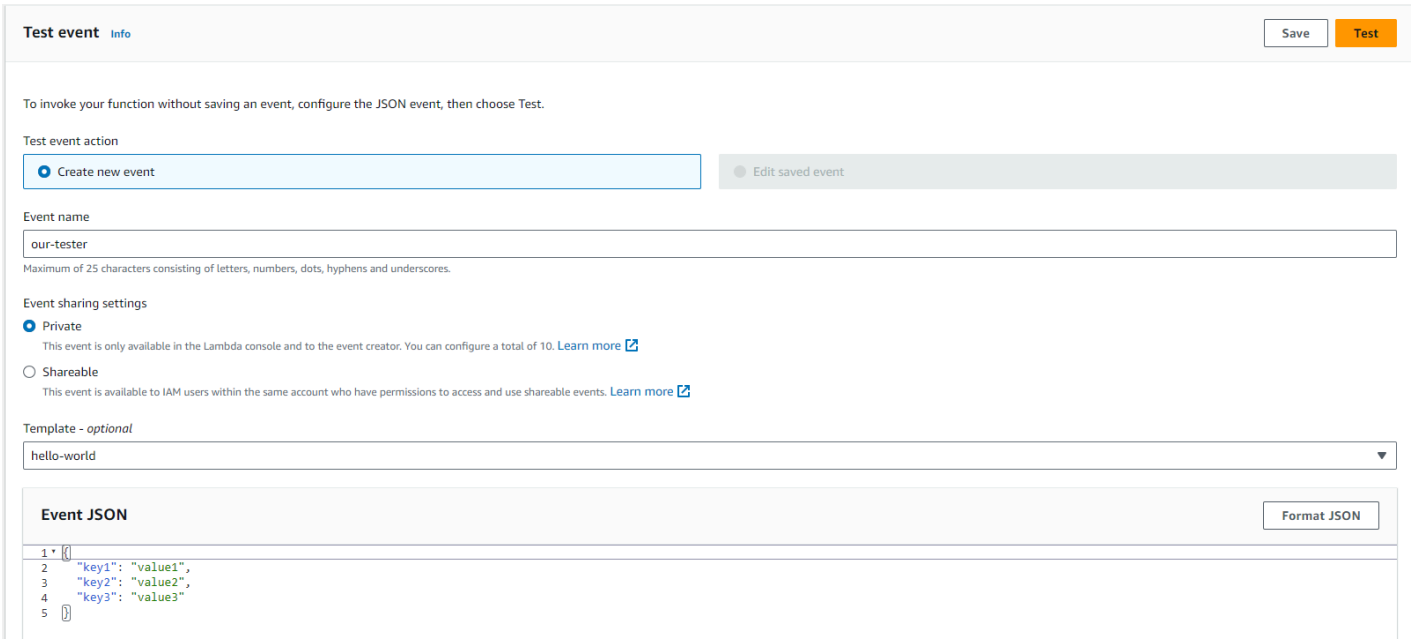
So See or Edit the basic settings go to configuration then click on edit general setting.



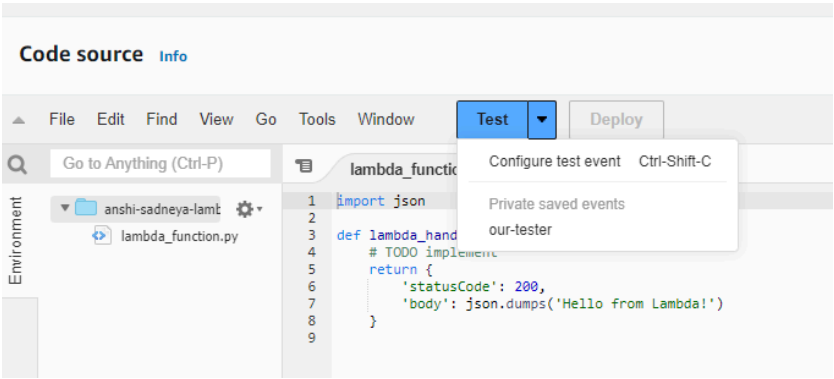
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



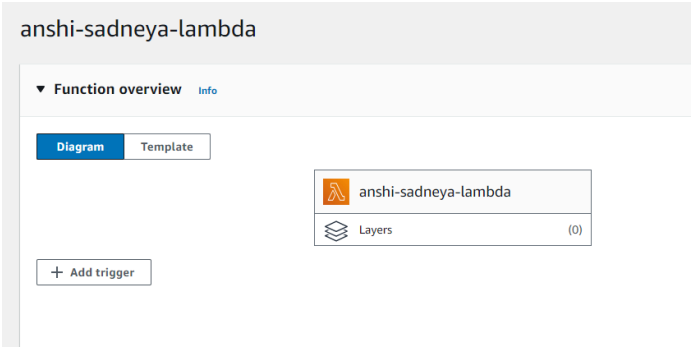
Step 5: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.



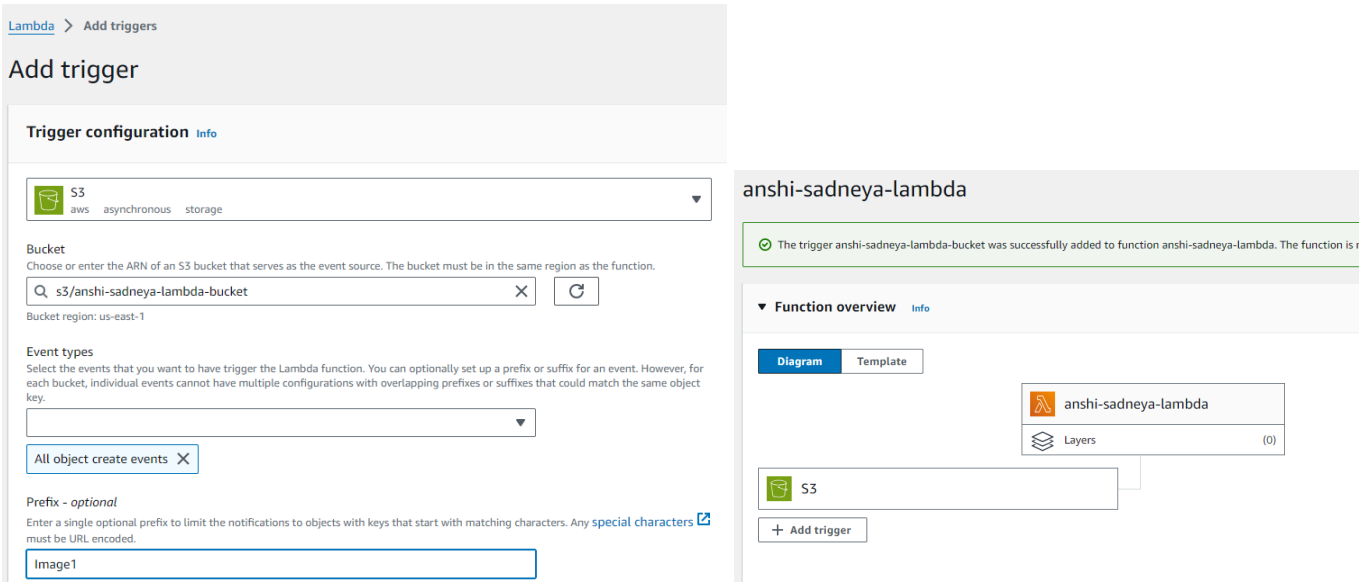
Step 6: Now In Code section select the created event from the dropdown .

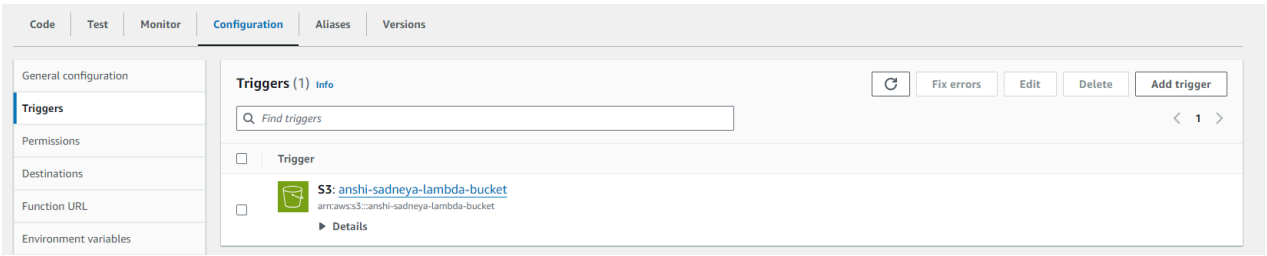


Step 7: Now In the Lambda function click on add tigger.

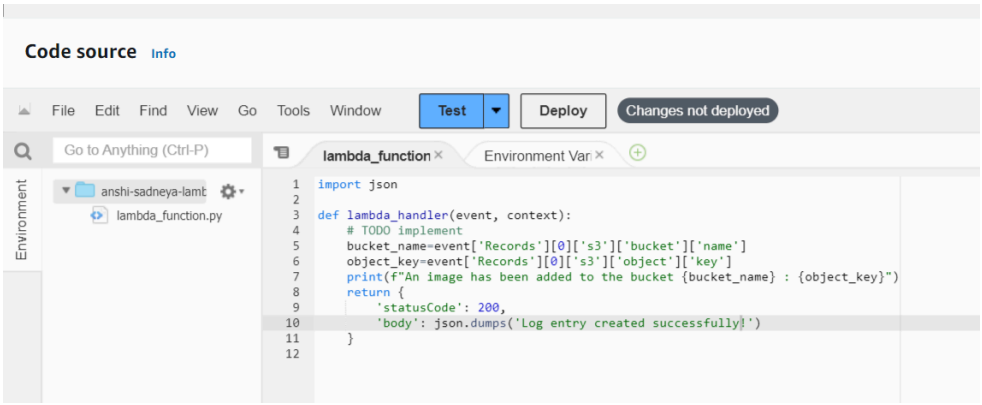


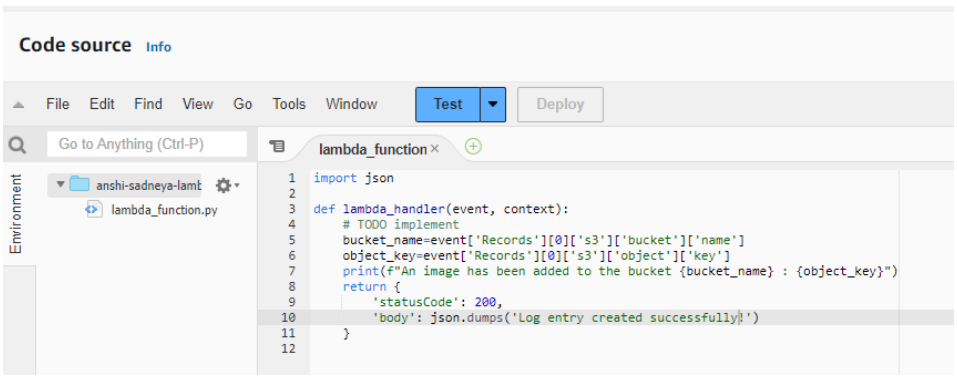
Now select the source as S3 then select the bucket name from the dropdown, keep other things to default and also you can add prefix to image.



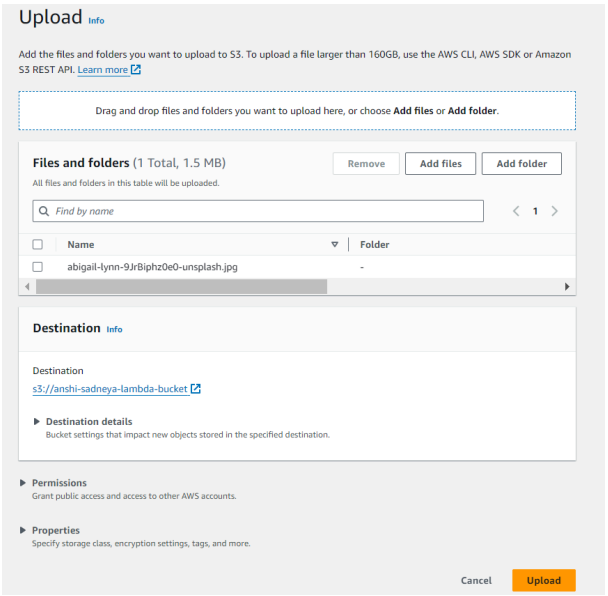


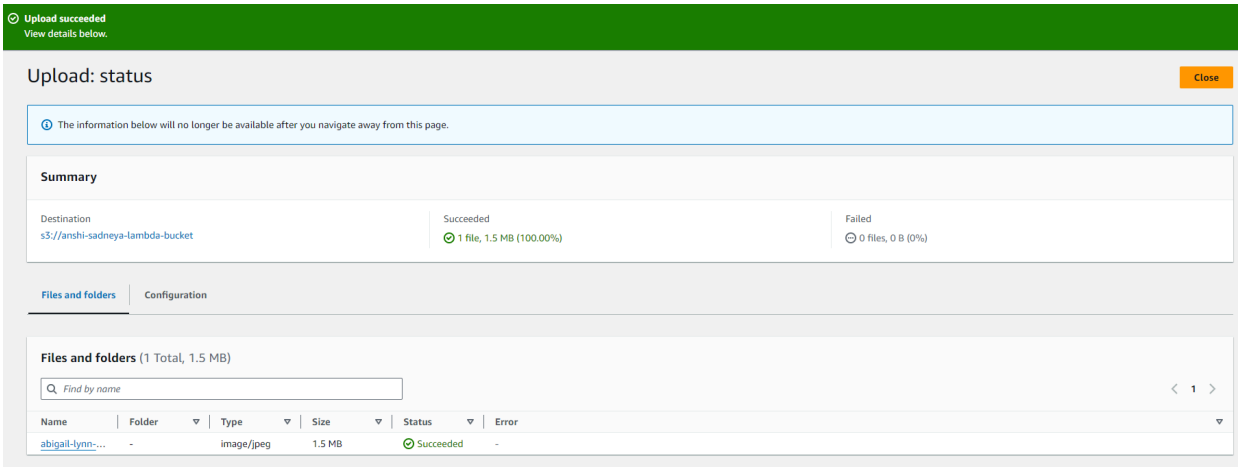
Step 8: Now Write code that logs a message like “An Image has been added” when triggered. Save the file and click on deploy.



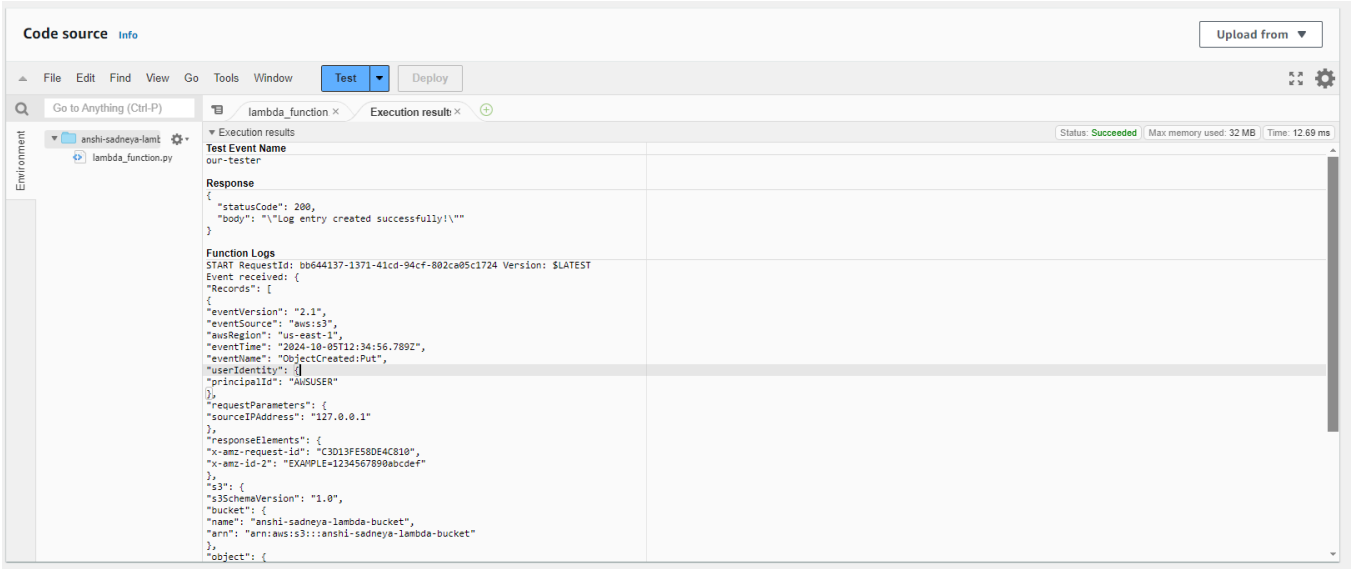


Step 9: Now upload any image to the bucket.

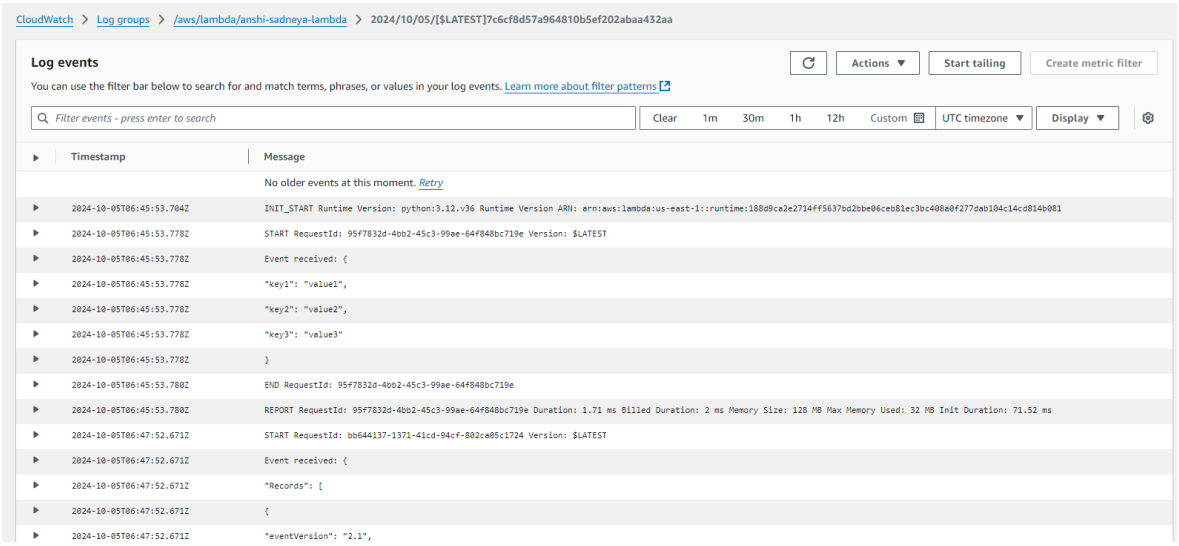




Step 10: Now to click on test in lambda to check whether it is giving log when image is added to S3.



Step 11: Now Lets see the log on Cloud watch.To see it go to monitor section and then click on view cloudwatch logs.



Conclusion: In this experiment, we successfully created an AWS Lambda function that logs a message whenever an image is uploaded to an S3 bucket. One key takeaway is that selecting the correct event template, specifically the S3 Put event, is essential. Using an incorrect event format initially caused errors due to the Lambda function not receiving the expected event structure.

After resolving these issues, the function was successfully triggered by S3 object uploads, confirming the effectiveness of Lambda's event-driven architecture. This experiment not only showcased Lambda's ability to respond to S3 events efficiently but also highlighted the importance of understanding and troubleshooting common event structure issues.