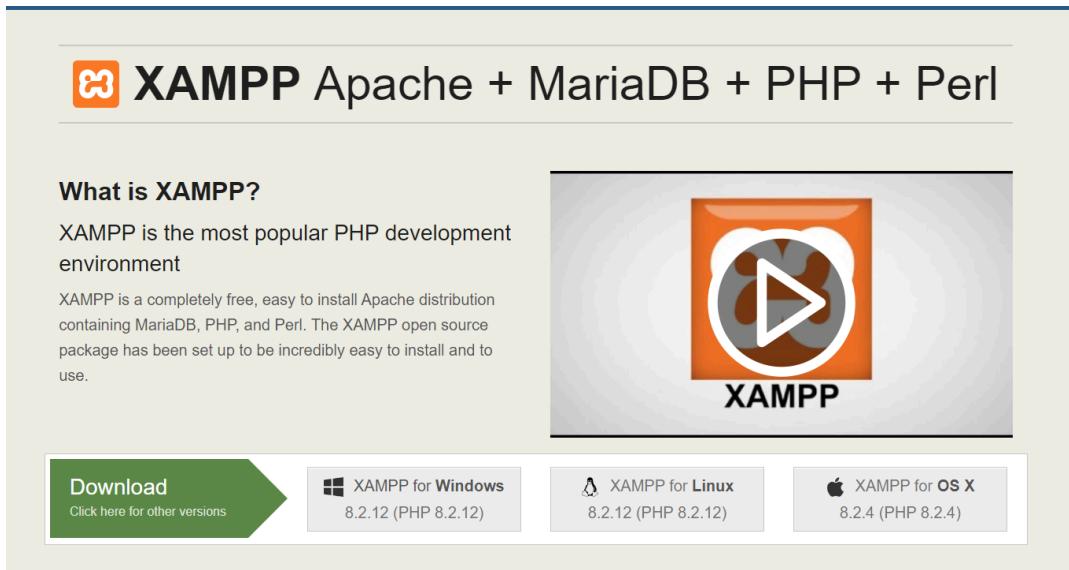


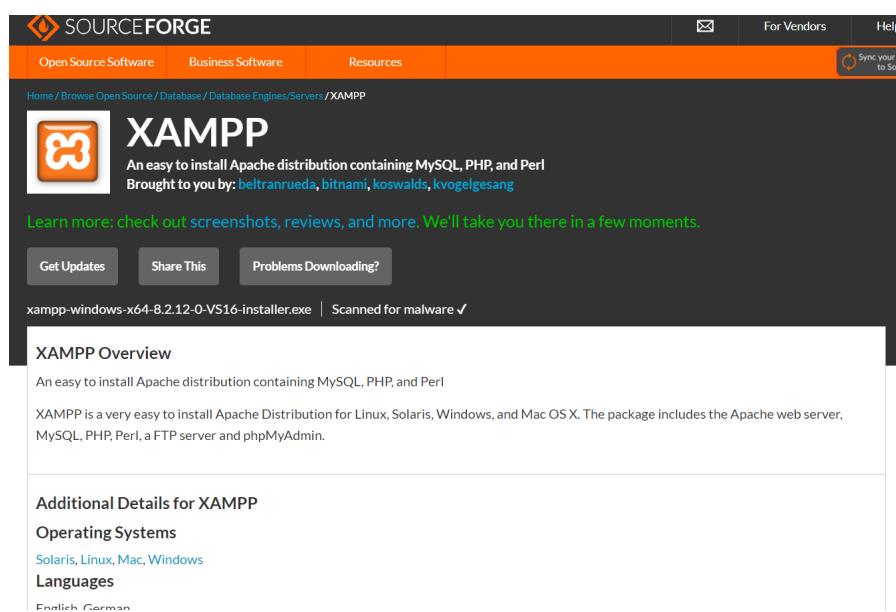
Exp :1A

**Aim : AWS (EC2) Installation steps for Linux instance
Hosting a website on Local Virtual Machine using Xampp**

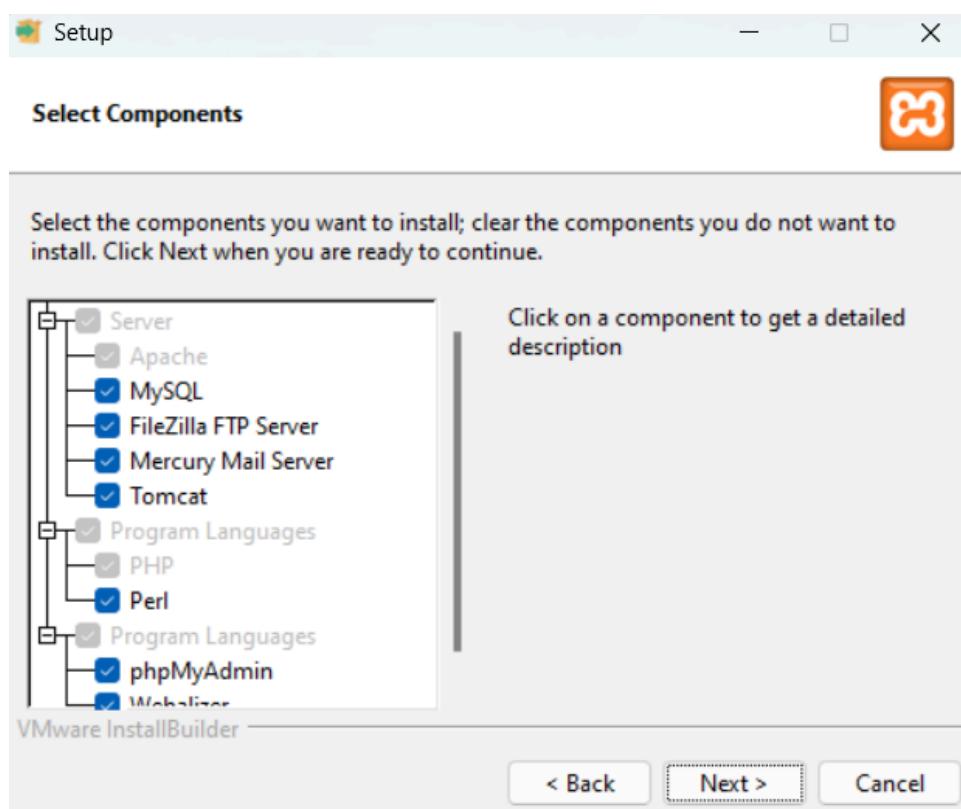
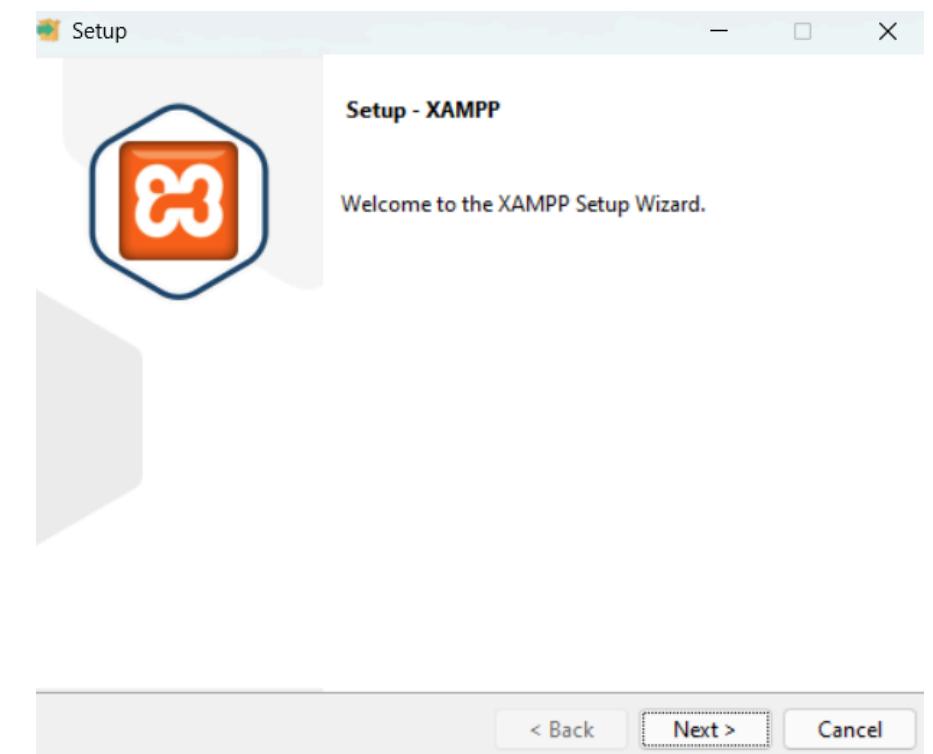
- 1) Go to official website of xampp



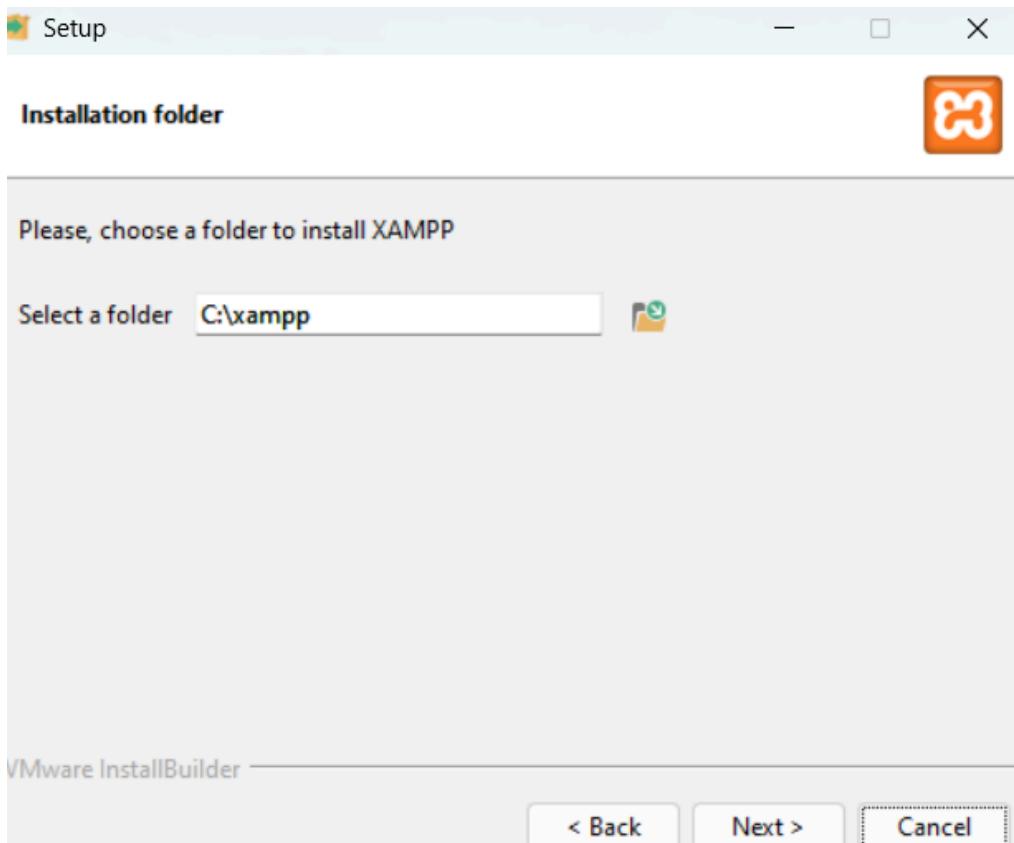
- 2) click on download and it will automatically get downloaded



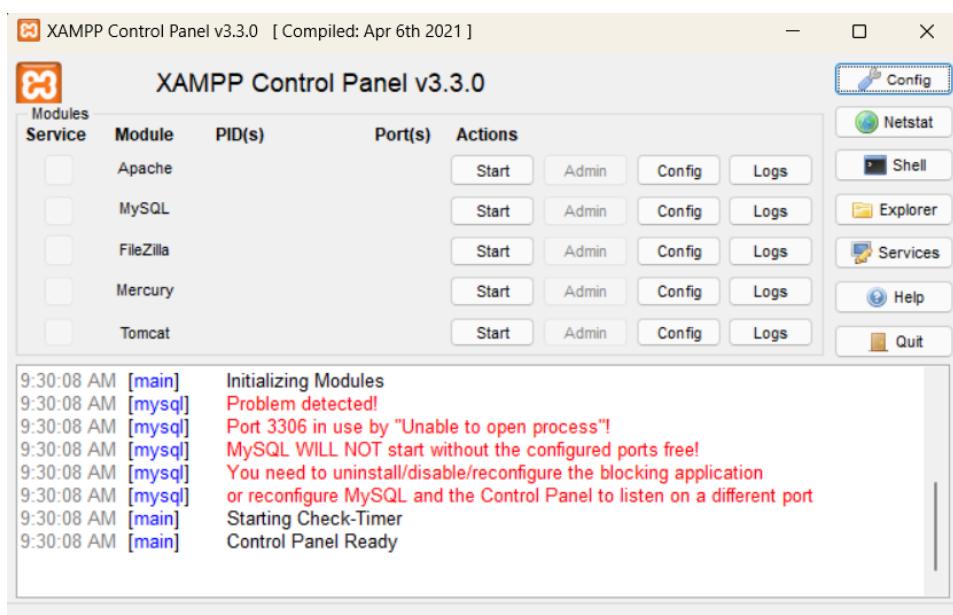
3)click next



4)click on next till the setup gets complete



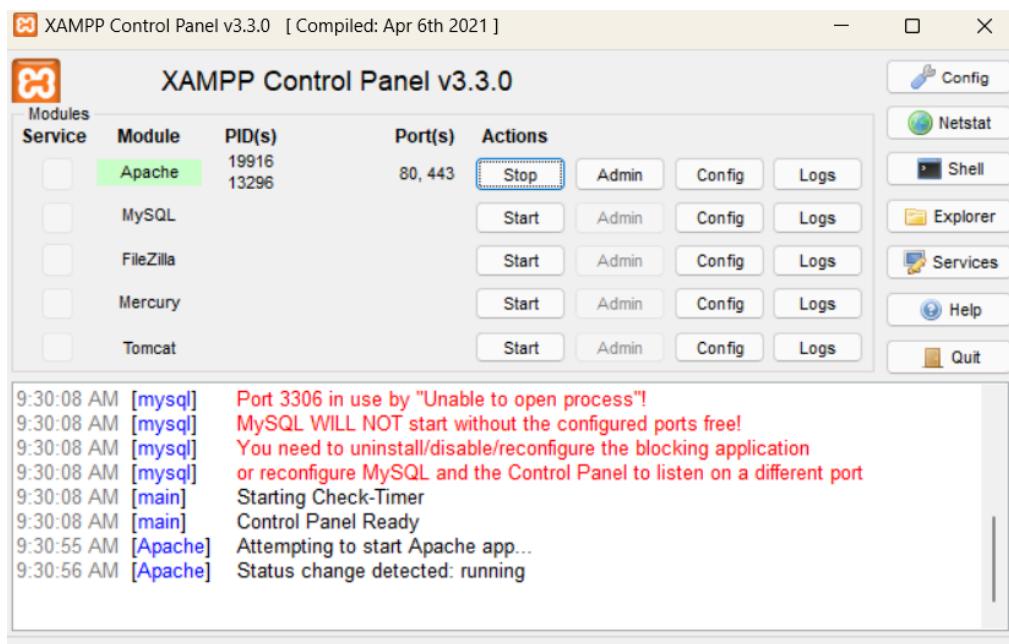
5)Open Xampp



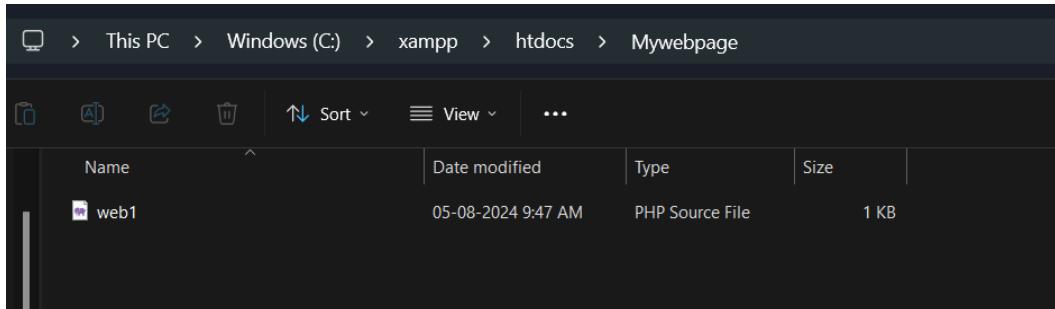
6)Write a php code

```
<?php  
echo "Hello, My Name is Atharv Nikam";  
  
echo "<br>";  
echo "My roll no is 36";  
echo "<br>";  
echo "Welcome to Adv Devops Lab";  
?>
```

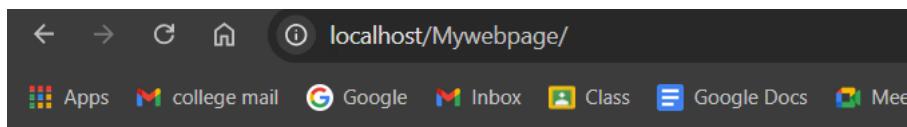
7)Starting Xampp



8)put your php file in the xampp ->htdocs



9)Open this

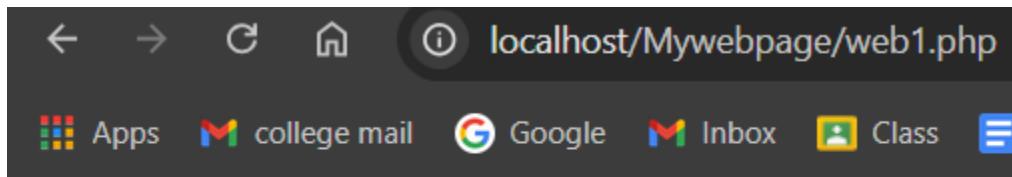


Index of /Mywebpage

Name	Last modified	Size	Description
Parent Directory		-	
web1.php	2024-08-05 09:47	140	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

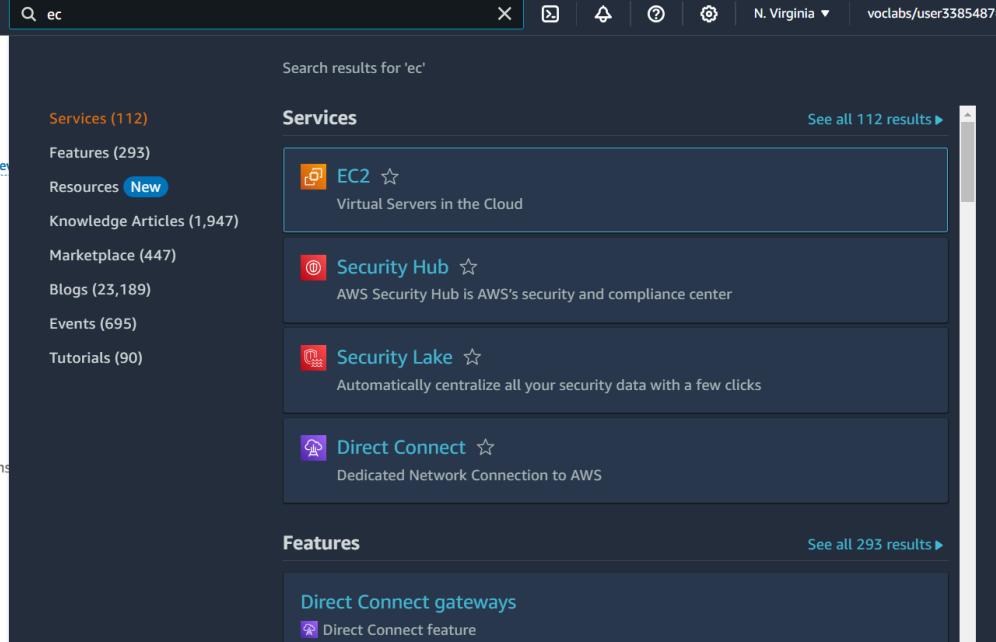
Click on your file and your website will open



Hello, My Name is Atharv Nikam
My roll no is 36
Welcome to Adv Devops Lab

Name:Atharv Nikam Div D15C Roll No:36
Aim : AWS (EC2) Installation steps for Linux instance

1)Go to aws homepage and click on ec2



Search results for 'ec'

Services (112)

- EC2
- Security Hub
- Security Lake
- Direct Connect

See all 112 results ►

Features (293)

- Direct Connect gateways

See all 293 results ►

Resources (New)

Knowledge Articles (1,947)

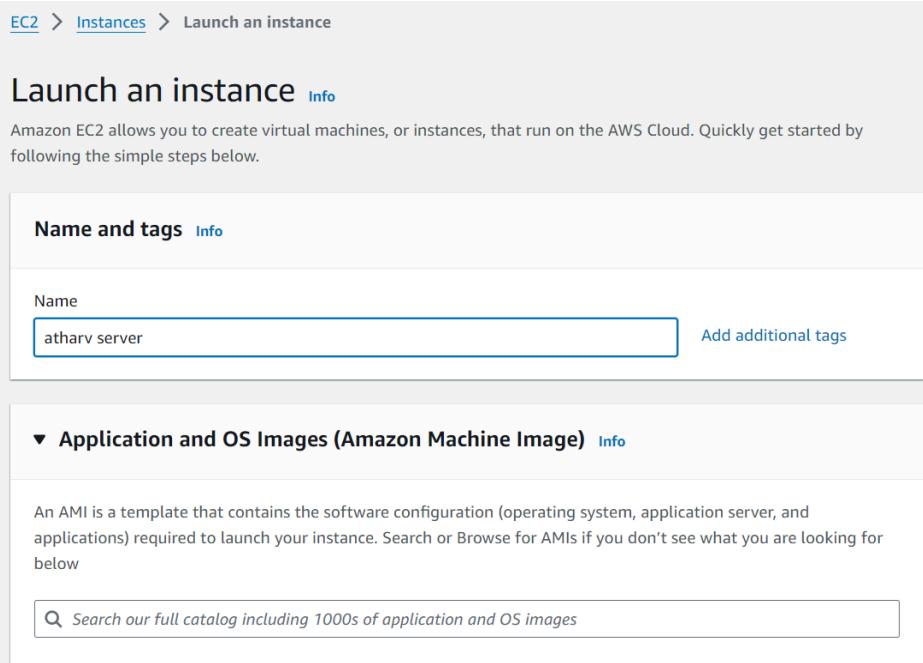
Marketplace (447)

Blogs (23,189)

Events (695)

Tutorials (90)

2)click on ec and give a name



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

atharv server

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

3)select on ubuntu

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 *Search our full catalog including 1000s of application and OS images*

Recents

Quick Start



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM) FRS General Purpose (SSD) Volume Type. Support available from Canonical.

4)select instance type t2

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

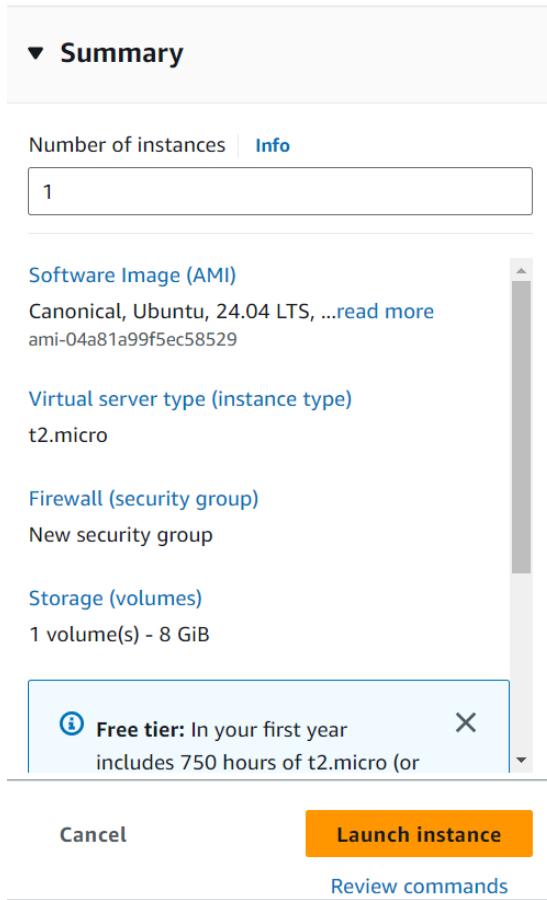
Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: required

5) see the summary and launch the instance



6) Successfully instance created



7)see your running instances

Instances (2) Info		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼	
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states ▾	< 1 >			
<input type="checkbox"/>	Name	▲	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Atharv Server		i-074ec2b12248b84a0	Running	t2.micro	Initializing	View alarms	us-east-1c

8)click on connect

Instances (1/2) Info		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states ▾	< 1 >		
<input type="checkbox"/>	Name	▼	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status
<input checked="" type="checkbox"/>	Atharv Server		i-074ec2b12248b84a0	Running	t2.micro	2/2 checks passed View alarms	
<input type="checkbox"/>	aws-cloud9-AtharvCloud9-020b82...		i-0709c00c32c38714b	Running	t2.micro	2/2 checks passed View alarms	

9)you will see this page

Connect to your instance i-074ec2b12248b84a0 (Atharv Server) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

Port 22 (SSH) is open to all IPv4 addresses
Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#).

Instance ID
 i-074ec2b12248b84a0 (Atharv Server)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
 44.206.244.123

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

10)this console will open

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

System information as of Wed Aug 14 07:02:38 UTC 2024

System load: 0.08          Processes: 106
Usage of /: 29.7% of 6.71GB  Users logged in: 0
Memory usage: 20%          IPv4 address for enX0: 172.31.90.246
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.

0 updates can be applied immediately.

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

Last login: Sun Aug 11 13:48:16 2024 from 18.206.107.28
ubuntu@ip-172-31-90-246:~$
```

i-074ec2b12248b84a0 (Atharv Server)
PublicIPs: 44.206.244.123 PrivateIPs: 172.31.90.246

11)Run all the commands

```
Memory usage: 22%          IPv4 address for enX0: 172.31.51.5
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-51-5:~$
```

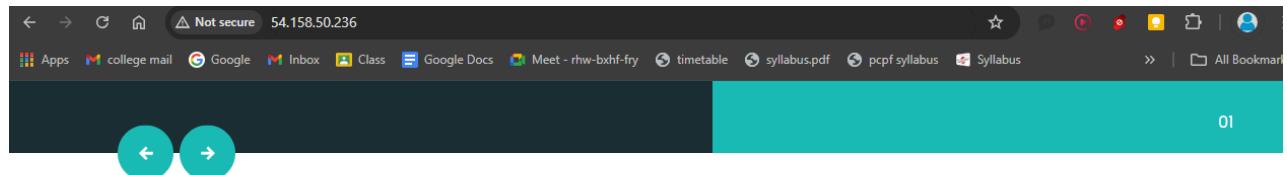
i-0a12db377565d1313 (Atharv Server)
PublicIPs: 54.209.65.33 PrivateIPs: 172.31.51.5

```
root@ip-172-31-90-246:~/temp# ls
spering-html  spering.zip
root@ip-172-31-90-246:~/temp# ls -lrt
total 552
drwxr-xr-x 5 root root 4096 Sep 16 2020 spering-html
-rw-r--r-- 1 root root 557415 Aug 20 2021 spering.zip
root@ip-172-31-90-246:~/temp# cd spering-html
root@ip-172-31-90-246:~/temp/sporing-html# ls -lrt
total 72
-rw-r--r-- 1 root root 23212 Jul 28 2020 index.html
-rw-r--r-- 1 root root 10108 Jul 28 2020 about.html
-rw-r--r-- 1 root root 9824 Jul 28 2020 category.html
-rw-r--r-- 1 root root 11825 Jul 28 2020 work.html
drwxr-xr-x 2 root root 4096 Sep 16 2020 js
drwxr-xr-x 2 root root 4096 Sep 16 2020 images
drwxr-xr-x 2 root root 4096 Sep 16 2020 css
root@ip-172-31-90-246:~/temp/sporing-html# mv * /var/www/html/
root@ip-172-31-90-246:~/temp/sporing-html# cd /var/www/html/
root@ip-172-31-90-246:/var/www/html# ls -lrt
total 72
-rw-r--r-- 1 root root 23212 Jul 28 2020 index.html
-rw-r--r-- 1 root root 10108 Jul 28 2020 about.html
-rw-r--r-- 1 root root 9824 Jul 28 2020 category.html
-rw-r--r-- 1 root root 11825 Jul 28 2020 work.html
drwxr-xr-x 2 root root 4096 Sep 16 2020 js
drwxr-xr-x 2 root root 4096 Sep 16 2020 images
drwxr-xr-x 2 root root 4096 Sep 16 2020 css
root@ip-172-31-90-246:/var/www/html# █
```

12)Enter the public domain from here

Name	Instance ID	Instanc...	Instanc...	Status check	Alarm status	Availabi...	Public I...	Public IPv4 ...
Atharv Server	i-074ec2b1...	Runn...	t2.micro	2/2 checks p:	View alarms +	us-east-1c	ec2-44-20...	44.206.244.123

13)Enter the domain and open it on your browser and you will see the website



BEST EXPERINCED FREELANCER HERE

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as

[Read More](#)

[Hire](#)

CATEGORY

← → ⌛ Not secure 54.158.50.236

Apps college mail Google Inbox Class Google Docs Meet - rhw-bxhf-fry timetable syllabus.pdf pcf syllabus Syllabus

All Bookmarks

 Spering

**YOU CAN
HIRE FREELANCER
HERE**

It is a long established fact that a reader will be distracted by the readable content of a page

About Us Get A Quote

03



← →

A screenshot of a web browser showing a landing page for "Spering". The page has a dark background on the left and a teal background on the right. On the left, there's a large heading "YOU CAN HIRE FREELANCER HERE" and a subtext about readability. Below that are two buttons: "About Us" (red) and "Get A Quote" (white). On the right, there's an illustration of a woman sitting at a desk with a laptop, and a red cat lying next to her. The URL in the address bar is "54.158.50.236" and the status bar says "Not secure".

Experiment 1B

Aim : AWS (EC2) Installation steps for Linux instance

1. Open your AWS account and search for Cloud9 service inside Developer tools. Create a new Cloud9 environment by filling in the required details. Make sure you use an EC2 instance to create your environment.

Create environment Info

Details

Name
 Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional*
 Limit 200 characters.

Environment type Info
Determines what the Cloud9 IDE will run on.

New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute
You have an existing instance or server that you'd like to use.

2)Select T2 Micro

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.

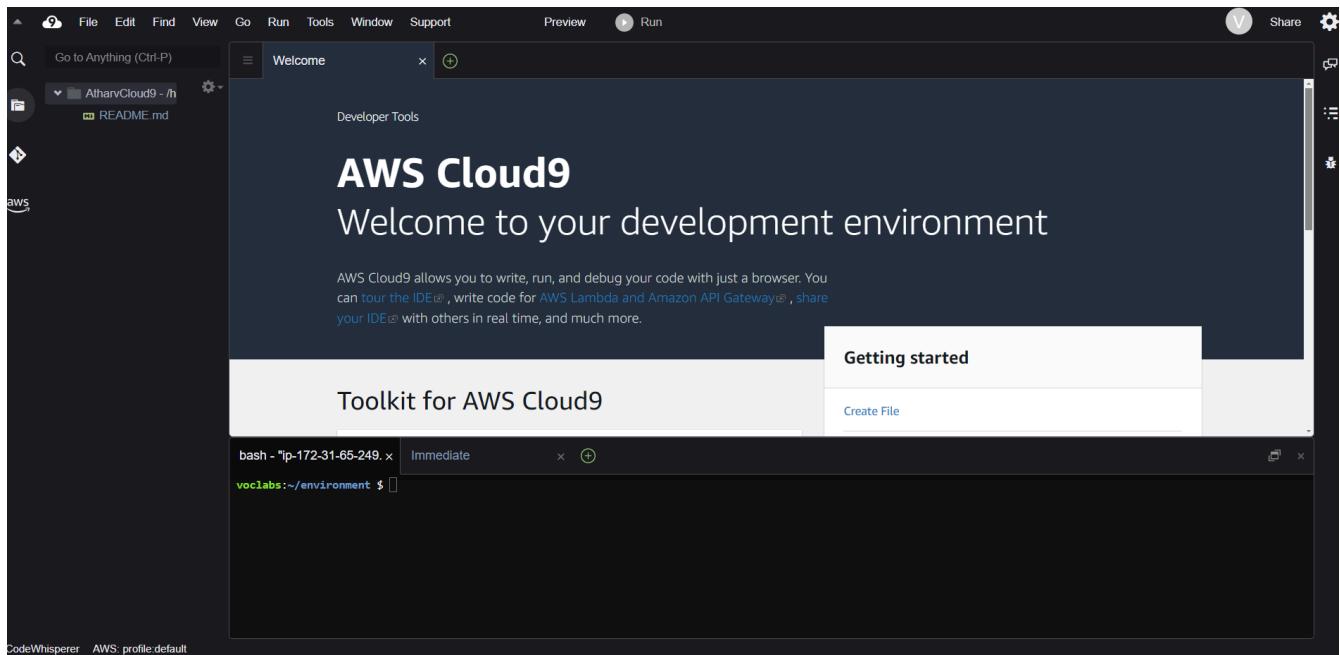
Timeout
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

3) See your summary

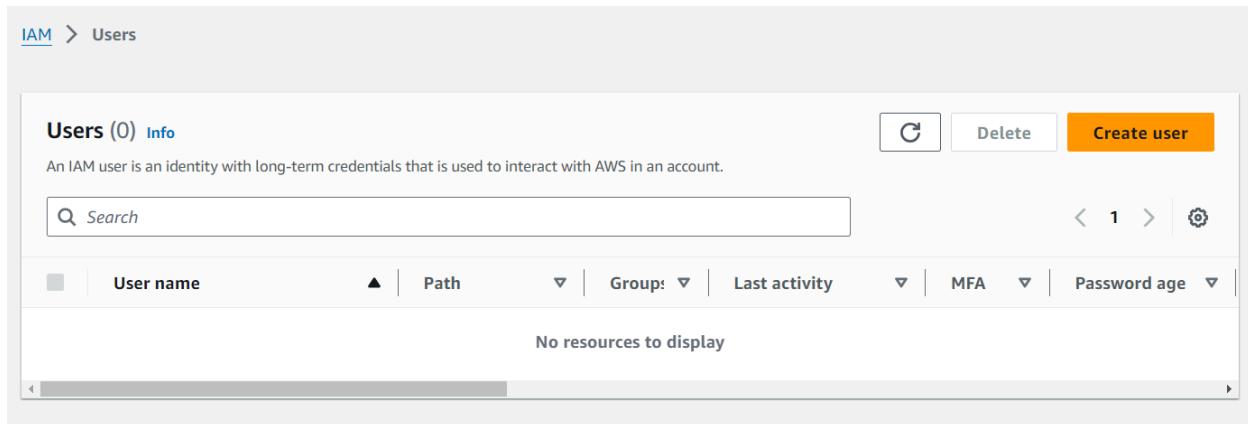
AtharvCloud9		Delete	Open in Cloud9
Details		Edit	
Name	AtharvCloud9	Owner ARN	Status
Description	CLoud9 installation	 arn:aws:sts::742555988891:assumed-role/voclabs/user3385487=NIKAM_ATHARV_SANJAY	 Ready
Environment type	EC2 instance	Number of members	Lifecycle status
		1	 Created

AWS Cloud9 > Environments						
Environments (1)						Create environment
My environments						
Name		Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
AtharvCloud9		EC2 instance	Secure Shell (SSH)	Owner	 arn:aws:sts::742555988891:assumed-role/voclabs/user3385487=NIKAM_ATHARV_SANJAY	

4)Your Aws Cloud9 Console will open



5)Click On IAM and create a new user



6)Enter your userName

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

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 Keypaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

7)Enter a Password

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 Keypaces, 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.

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 Keypaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

8)Now similarly Create a group

The screenshot shows the 'User groups' page in the AWS IAM console. At the top, there is a header with 'User groups (0)' and a 'Create group' button. Below the header, a search bar and a table with columns for 'Group name', 'Users', 'Permissions', and 'Creation time'. A message at the bottom states 'No resources to display'.

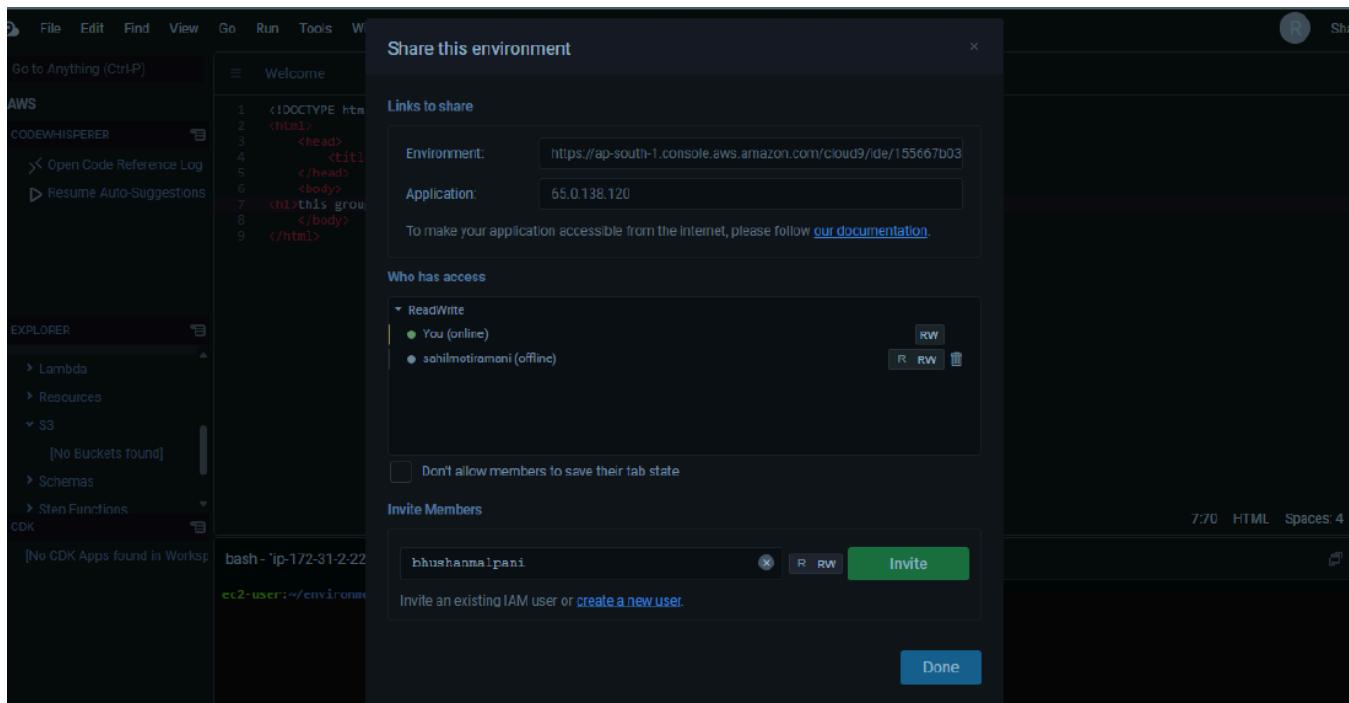
9)give a name

The screenshot shows the 'Create user group' wizard. The first step is 'Name the group'. It has a field for 'User group name' containing 'MSBCLOUD9'. There is also a note about character limits and allowed characters.

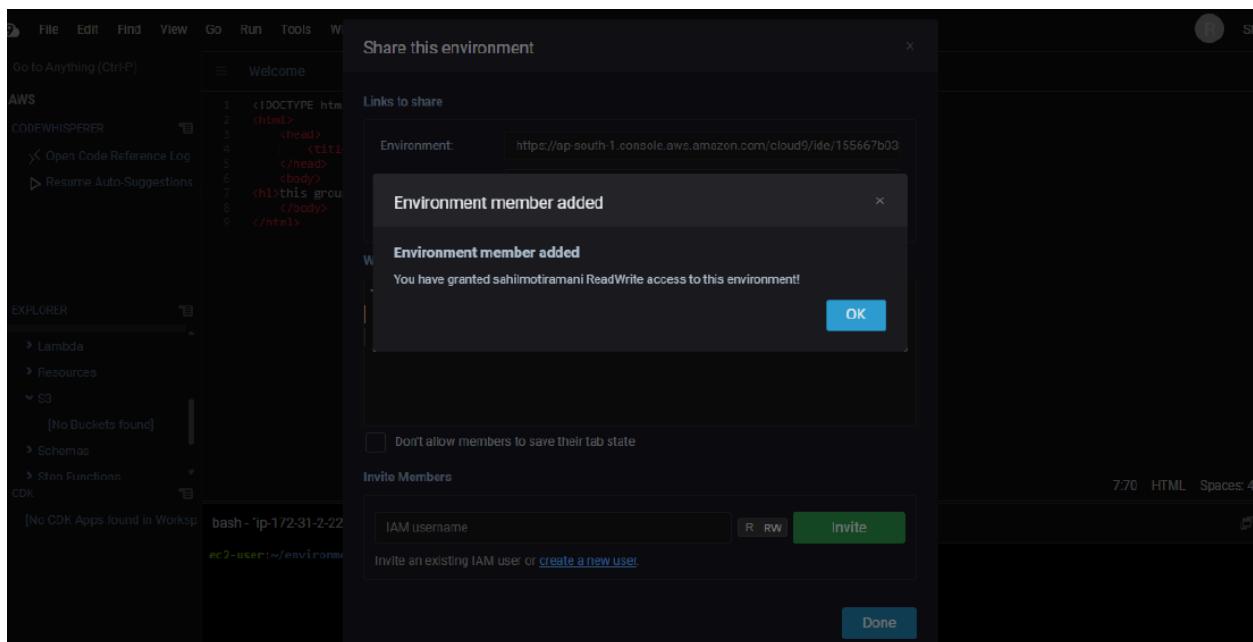
10)The user has successfully been created i.e There is a custom made username and a password for the IAM user.

The screenshot shows a confirmation dialog titled 'JD9 user group created.' It displays 'User details' for a user named 'sahilmotiramani' with a 'Custom password' type and 'Yes' for 'Require password reset'. Below this is a 'Permissions summary' section showing two entries: 'IAMUserChangePassword' (AWS managed, used as Permissions policy) and 'MSBCLOUD9' (Group, used as Permissions group). At the bottom, there is a 'Tags - optional' section which is currently empty.

11)Now you can share your environment now you can add collaborators



12)New Member added



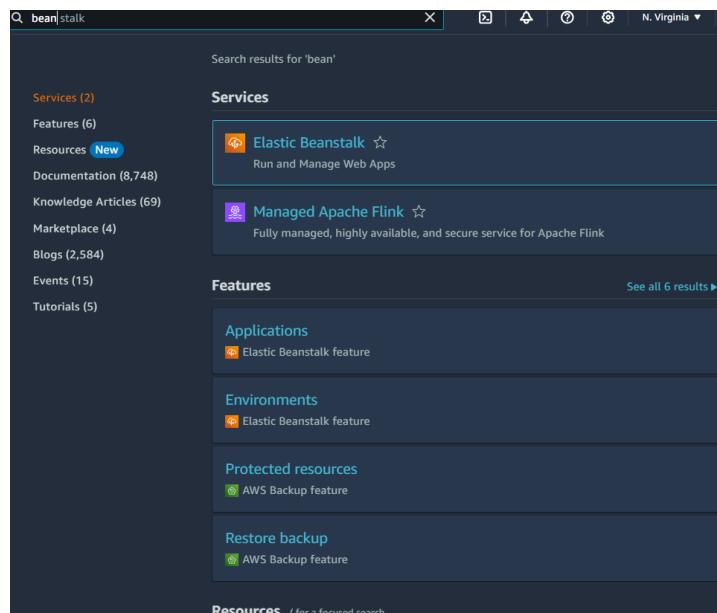
We were required to log in from another browser using the IAM user's credentials to gain access to the shared Cloud9 environment. Unfortunately, we were unable to complete these steps because the Cloud9 services were disrupted, which also blocked remote access to the IAM user account. This disruption has prevented us from performing the necessary actions, leaving us unable to access the shared environment as intended.

Exp 2

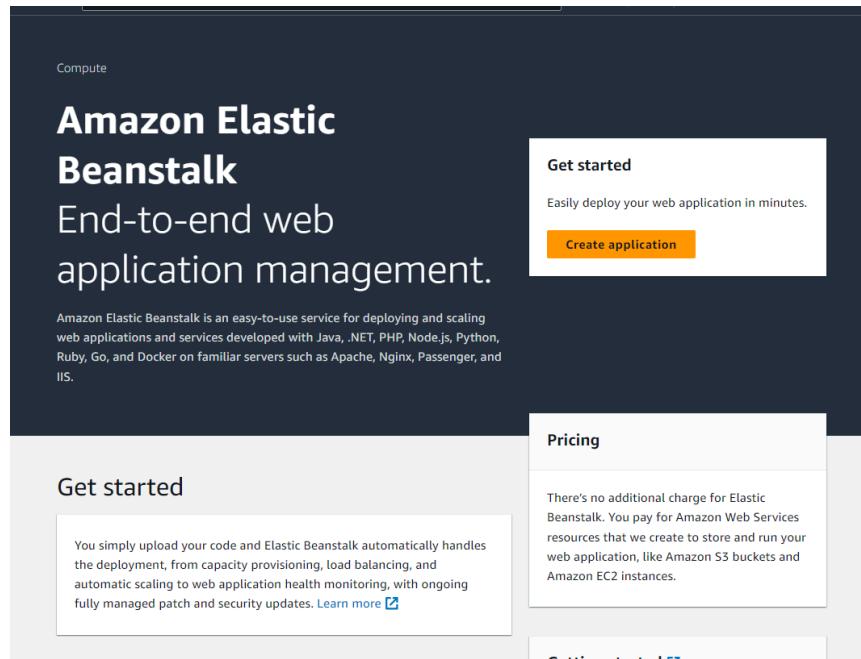
Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using

AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy

1)Login to your account and search for Elastic Beanstalk



2)Click on it and select create application

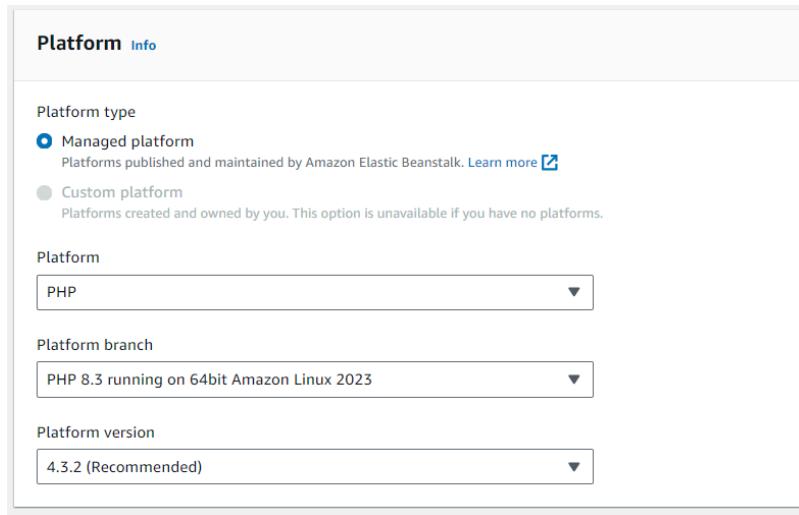


3) Give a appropriate name to your application

The screenshot shows the "Configure environment" step in the AWS Elastic Beanstalk setup wizard. It includes sections for "Environment tier", "Application information", and "Environment information".

- Environment tier:** Set to "Web server environment".
 - Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Application information:** Application name is "atharvbeanstalk".
 - Maximum length of 100 characters.
- Environment information:** Environment name is "Atharvbeanstalk-env".
 - Choose the name, subdomain and description for your environment. These cannot be changed later.

4) Select php and other will automatically get filled



5)Now, while creating the environment, we are asked to provide an IAM role with the necessary

EC2 permissions. We are supposed to make sure that we have made an existing IAM role with the following set of permissions:

1. AWSElasticBeanStalkWebTier
2. AWSElasticBeanStalkWorkerTier
3. AWSElasticBeanStalkMulticontainerDocker

Select iam and then create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Select trusted entity Info

Trusted entity type

AWS service

Allow AWS services like EC2, Lambda, or others to perform actions in this account.

AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity

Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

5)give a name and then select ec2

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+=, @-_` characters.

Description

Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=., @-^{\}!#\$%^&()~`"

Step 1: Select trusted entities

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case



Choose a use case for the specified service.

Use case

EC2

Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager

Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

EC2 Spot Fleet Role

Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

EC2 - Spot Fleet Auto Scaling

Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

EC2 - Spot Fleet Tagging

Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

Add roles

Add permissions Info

Permissions policies (3/949) Info

Choose one or more policies to attach to your new role.

Filter by Type

Policy name	Type	Description
<input type="checkbox"/> AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk op...
<input checked="" type="checkbox"/> AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instance...

▶ Set permissions boundary - *optional*

These 3 roles should be added

Step 2: Add permissions Edit

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

6) Now create an Application

Click on create application

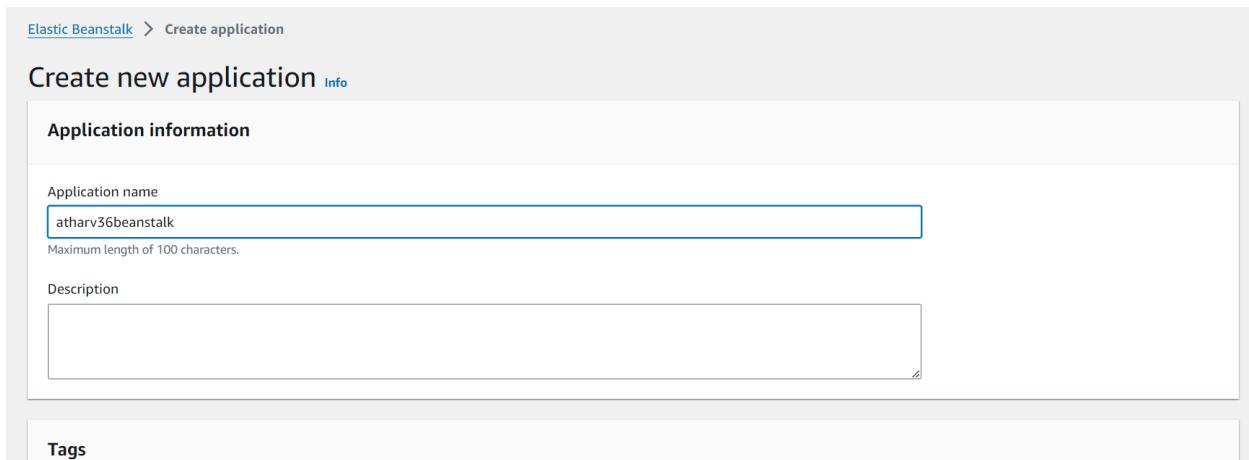


Elastic Beanstalk > Applications

Applications (8) [Info](#)

Filter results matching the display value

Create application



Elastic Beanstalk > Create application

Create new application [Info](#)

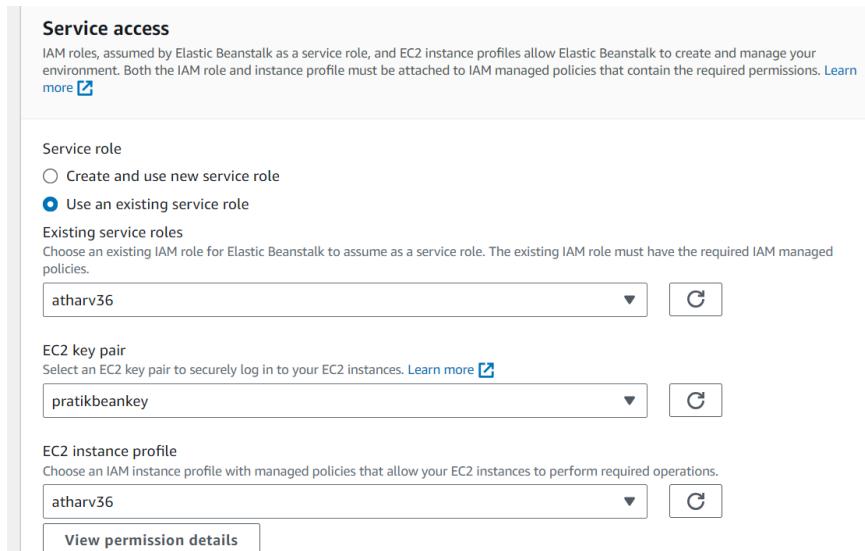
Application information

Application name
atharv36beanstalk

Description

Tags

7) in services access give the role we just created



Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

atharv36

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

pratikbeankey

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

atharv36

[View permission details](#)

8) See the summary and click on create application

Step 1: Configure environment

Edit

Environment information

Environment tier	Application name
Web server environment	atharv36beanstalk
Environment name	Application code
Atharv36beanstalk-env	Sample application
Platform	arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2

9)then you will get a message that your environment is created

Environment successfully launched. X

[Elastic Beanstalk](#) > [Environments](#) > Atharv36beanstalk-env-1

Atharv36beanstalk-env-1 [Info](#) C Actions ▾ Upload and deploy

Environment overview

Health	Environment ID
⚠ Warning	e-f2sk336vqk
Domain	Application name
Atharv36beanstalk-env-1.eba- 4ivtm3r.us-east- 1.elasticbeanstalk.com	atharv36beanstalk

Platform Change version

Platform	PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2
Running version	-
Platform state	✔ Supported

[Events](#) | [Health](#) | [Logs](#) | [Monitoring](#) | [Alarms](#) | [Managed updates](#) | [Tags](#)

10)Create a pipeline

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type
Info 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.

Service role
 New service role
Create a service role in your account
 Existing service role
Choose an existing service role from your account

11)select source and select Github version 2

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

Info **New GitHub version 2 (app-based) action**
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

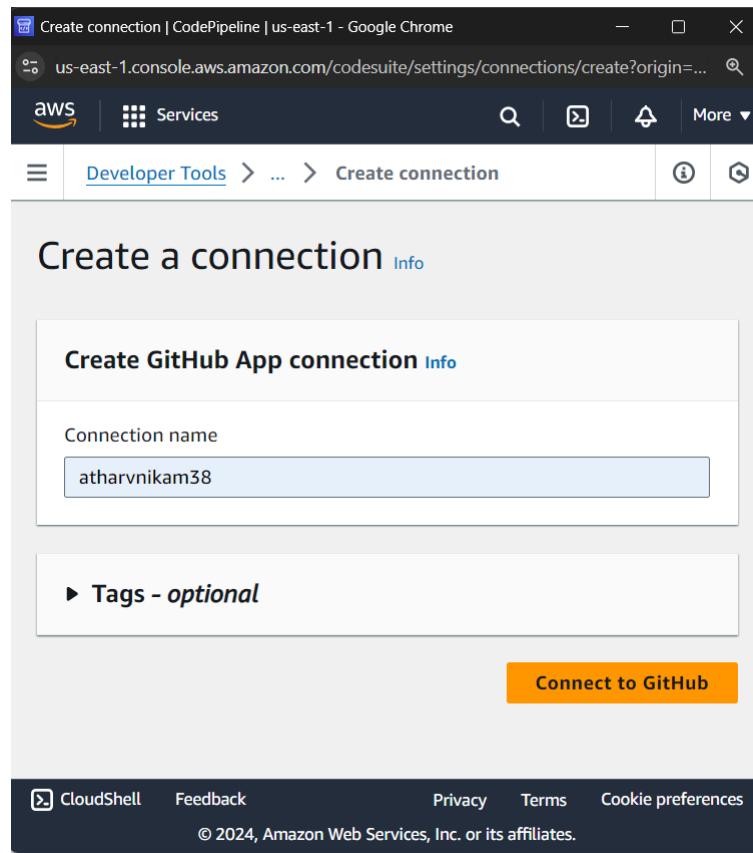
12)First you have to fork a github repo

The screenshot shows a GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0'. The repository is public and has 20 commits. The commit history includes updates to README.md, adding a template, creating a dist folder, setting up S3, and updating CONTRIBUTING files. The repository has 436 forks and 4 stars. The sidebar on the right provides links to the README, Apache-2.0 license, Code of conduct, Activity, and Report repository.

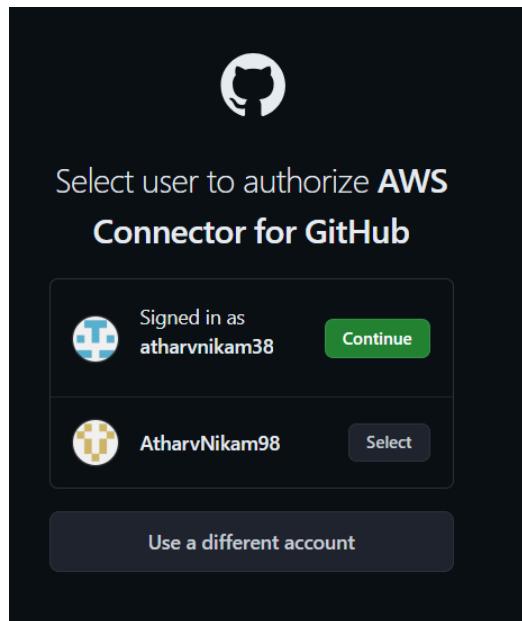
13)forked this image on the your account

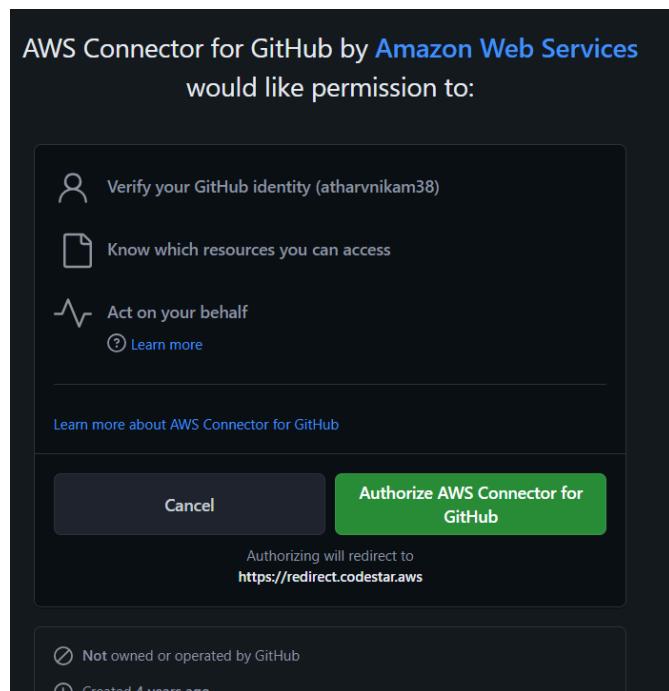
The screenshot shows a forked GitHub repository for 'atharvnikam38'. The repository is a fork of 'imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0'. It has 1 branch and 20 commits. The commit history is identical to the original repository. The repository has 0 forks and 0 stars. The sidebar on the right provides links to the README, Apache-2.0 license, Activity, and Packages.

14)Now connect your github repo here



15)Authorize it





Connect to GitHub

GitHub connection settings [Info](#)

Connection name
atharvnikam38

GitHub Apps
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.
53763427 [X](#) or [Install a new app](#)

▶ Tags - *optional*

[Connect](#)

16)now you have connect the repo

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2)

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:us-east-1:069450516824:connection/135ebd03-0a or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

atharvnikam38/aws-codepipeline-s3-codedeploy-linux-2.0

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

17)click next and click skip build test

Build - optional
This is the stage where your pipeline will build your code before it runs tests or deploys it.

Skip build stage

Your pipeline will not include a build stage. Are you sure you want to skip this stage?

Cancel Skip

ge Next

18)enter final info

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](#)

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.

Q atharvbeanstalk X

Environment name
AtharvBeanstalk-env

Q AtharvBeanstalk-env X

Configure automatic rollback on stage failure

19)you have successfully created the pipeline

Success
Congratulations! The pipeline atharvpipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > atharvpipeline

atharvpipeline

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: [2ae6f2da-083a-48b0-8c79-5a4aa7f2db88](#)

Source
[GitHub \(Version 2\)](#)
Succeeded - 1 minute ago
[cded2323](#)
[View details](#)

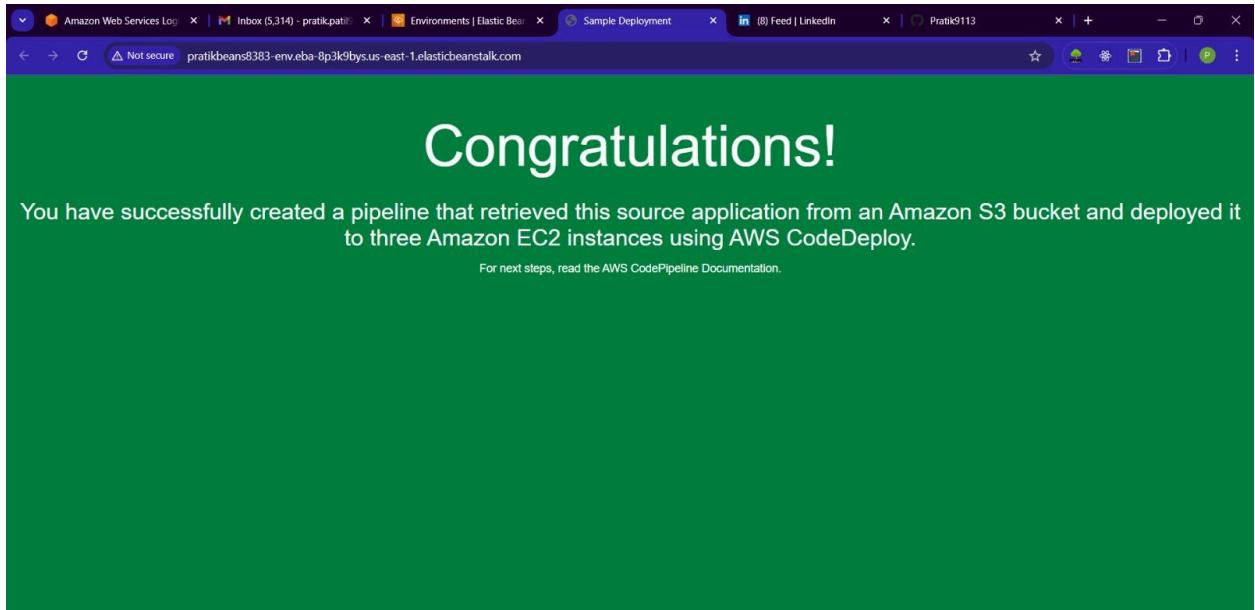
cded2323 [Source: Update index.html](#)

[Disable transition](#)

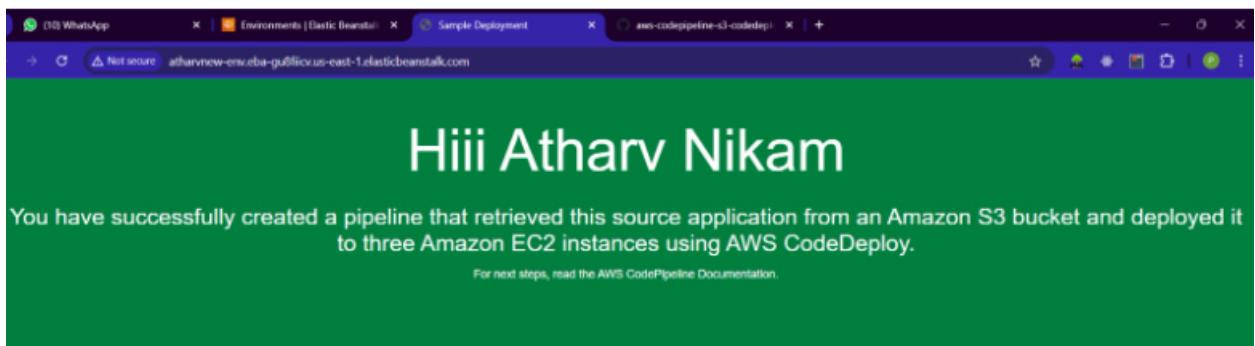
Deploy Succeeded
Pipeline execution ID: [2ae6f2da-083a-48b0-8c79-5a4aa7f2db88](#)

[Start rollback](#)

20)you will get this message



21)make changes to your repo and rerun the pipeline



Experiment 3

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

1)Launch an EC2 Instance:

Choose Amazon Linux as the operating system for your instance.

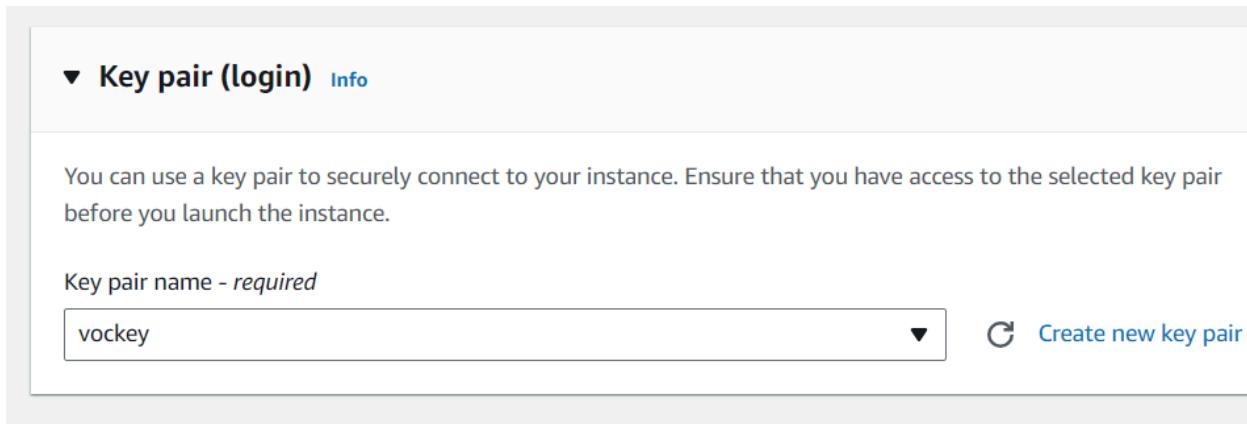
The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a 'Create Function' button. Below it, a section titled 'Lambda@Edge' has a 'Create' button. The main area is titled 'My Functions' and shows a single function named 'HelloWorldFunction'. This function has a status of 'Active' and was created '1 day ago'. It includes a 'Edit' button and a 'Delete' button. A 'Logs' tab is also visible.

Note: The AWS free tier provides a t2.micro instance (1 CPU, 1 GiB RAM), but Kubernetes requires a minimum of 2 CPUs and 2 GiB RAM. So, make sure to select **t2.medium**

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a 'Create Function' button. Below it, a section titled 'Lambda@Edge' has a 'Create' button. The main area is titled 'My Functions' and shows a single function named 'HelloWorldFunction'. This function has a status of 'Active' and was created '1 day ago'. It includes a 'Edit' button and a 'Delete' button. A 'Logs' tab is also visible.

2)select a Key Pair:

You can either use the default key pair provided by AWS or create a new one for better security. Click on **Create**.



All your instances will appear here

Instances (3) Info										
Last updated less than a minute ago										
Actions Launch instances										
<input type="text"/> Find Instance by attribute or tag (case-sensitive)										Running
Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availabi...	Public I...	Pu...		
<input type="checkbox"/> Master	i-0d3f7911...	Running	t2.medium	2/2 checks p:	View alarms +	us-east-1d	ec2-54-24...	54		
<input type="checkbox"/> Computer2	i-09a31ba0...	Running	t2.medium	Initializing	View alarms +	us-east-1d	ec2-34-20...	34		
<input type="checkbox"/> Computer1	i-041bb34...	Running	t2.medium	2/2 checks p:	View alarms +	us-east-1d	ec2-54-17...	54		

The screenshot shows the 'Summary' section of the launch instance wizard. It displays the number of instances (1), software image (Amazon Linux 2023 AMI 2023.5.2...), virtual server type (t2.medium), firewall (New security group), and storage (1 volume(s) - 8 GiB). At the bottom are 'Cancel', 'Launch instance' (highlighted in orange), and 'Review commands' buttons.

3)Connect to the Instance:

Navigate to the instances page, locate your instance, and click on its ID. Then, select **Connect** and keep the default connection settings. Finally, click **Connect** to start your session.

The screenshot shows the AWS EC2 Instances page. The instance summary for 'i-0d3f7911e0aabcc35 (Master)' is displayed. Key details include:

- Instance ID:** i-0d3f7911e0aabcc35 (Master)
- Public IPv4 address:** 54.242.215.34 [open address]
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-22-81.ec2.internal
- Instance type:** t2.medium
- VPC ID:** vpc-0f7970ea32a533bcc [open]
- Subnet ID:** subnet-09ba566295275f771 [open]
- Instance ARN:** arnaws:ec2:us-east-1:74255598891:instance/i-0d3f7911e0aabcc35

At the bottom, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Details tab is selected.

The screenshot shows the 'EC2 Instance Connect' dialog box. It includes the following sections:

- EC2 Instance Connect** (selected), **Session Manager**, **SSH client**, **EC2 serial console**
- Port 22 (SSH) is open to all IPv4 addresses:** A warning message states that Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). It advises increasing security by restricting access to only the EC2 Instance Connect service IP addresses for your Region: **18.206.107.24/29**. [Learn more](#).
- Connection Type:**
 - Connect using EC2 Instance Connect: Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
 - Connect using EC2 Instance Connect Endpoint: Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
- Public IPv4 address:** 54.242.215.34
- Username:** ec2-user (input field)
- Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
- Buttons:** Cancel, Connect

Name:Atharv Sanjay Nikam

Div:D15C

Roll:36

Instance summary for i-041bb34892373b3d7 (Computer1) Info		
Updated less than a minute ago		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-041bb34892373b3d7 (Computer1)	54.174.132.164 open address	172.31.28.157
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-174-132-164.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-28-157.ec2.internal	ip-172-31-28-157.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.medium	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address	VPC ID	Auto Scaling Group name
54.174.132.164 [Public IP]	vpc-0f7970ea32a533bcc	-
IAM Role	Subnet ID	
-	subnet-09ba566295275f771	
IMDSv2	Instance ARN	
Required	arn:aws:ec2:us-east-1:742555988891:instance/i-041bb34892373b3d7	

Instance summary for i-041bb34892373b3d7 (Computer1) Info		
Updated less than a minute ago		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-041bb34892373b3d7 (Computer1)	54.174.132.164 open address	172.31.28.157
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-174-132-164.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-28-157.ec2.internal	ip-172-31-28-157.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.medium	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address	VPC ID	Auto Scaling Group name
54.174.132.164 [Public IP]	vpc-0f7970ea32a533bcc	-
IAM Role	Subnet ID	
-	subnet-09ba566295275f771	
IMDSv2	Instance ARN	
Required	arn:aws:ec2:us-east-1:742555988891:instance/i-041bb34892373b3d7	

All the three server

```
  _\ _###_      Amazon Linux 2023
~~ \###\_
~~ \###|
~~ \|/   https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'__->
~~ /_/
~~ ./_/_
/_m/,_/_/
[ec2-user@ip-172-31-28-157 ~]$
```

i-041bb34892373b3d7 (Computer1)
PublicIPs: 54.174.132.164 PrivateIPs: 172.31.28.157

```
  _\ _###_      Amazon Linux 2023
~~ \###\_
~~ \###|
~~ \|/   https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'__->
~~ /_/
~~ ./_/_
/_m/,_/_/
[ec2-user@ip-172-31-22-81 ~]$
```

i-0d3f7911e0aabcc35 (Master)
PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81

```
  _\ _###_      Amazon Linux 2023
~~ \###\_
~~ \###|
~~ \|/   https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'__->
~~ /_/
~~ ./_/_
/_m/,_/_/
[ec2-user@ip-172-31-23-208 ~]$
```

i-09a31ba0572c10bdb (Computer2)
PublicIPs: 34.204.51.153 PrivateIPs: 172.31.23.208

Step 2: Installation of Docker

1)Switch to Root User:

Run `sudo su` to get root-level access in the terminal.

```
/m/
[ec2-user@ip-172-31-22-81 ~]$ sudo su
[root@ip-172-31-22-81 ec2-user]#
```

2)Install Docker:

Use the YUM package manager to install Docker by running:

`yum install docker -y`

```
[root@ip-172-31-22-81 ec2-user]# yum install docker -y
Last metadata expiration check: 0:11:19 ago on Sun Sep 15 06:50:06 2024.
Dependencies resolved.
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-lib-1.8.8-3.amzn2023.0.2.x86_64.rpm           6.3 MB/s | 401 kB   00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm          6.0 MB/s | 183 kB   00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm             2.8 MB/s | 75 kB   00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.6 MB/s | 58 kB   00:00
(5/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.4 MB/s | 30 kB   00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.9 MB/s | 84 kB   00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                1.1 MB/s | 83 kB   00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm              22 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm        41 MB/s | 35 MB   00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm           38 MB/s | 44 MB   00:01

Total download size: 84 M
Installed size: 317 M
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-lib-1.8.8-3.amzn2023.0.2.x86_64.rpm           6.3 MB/s | 401 kB   00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm          6.0 MB/s | 183 kB   00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm             2.8 MB/s | 75 kB   00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.6 MB/s | 58 kB   00:00
(5/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.4 MB/s | 30 kB   00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.9 MB/s | 84 kB   00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                1.1 MB/s | 83 kB   00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm              22 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm        41 MB/s | 35 MB   00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm           38 MB/s | 44 MB   00:01
```

```
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-lib-1.8.8-3.amzn2023.0.2.x86_64.rpm           6.3 MB/s | 401 kB   00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm          6.0 MB/s | 183 kB   00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm             2.8 MB/s | 75 kB   00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.6 MB/s | 58 kB   00:00
(5/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.4 MB/s | 30 kB   00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.9 MB/s | 84 kB   00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                1.1 MB/s | 83 kB   00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm              22 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm        41 MB/s | 35 MB   00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm           38 MB/s | 44 MB   00:01

Total download size: 84 M
Installed size: 317 M
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing:
    Installing : runc-1.1.13-1.amzn2023.0.1.x86_64
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  Installing : libnftnl-1.0.1-19.amzn2023.0.2.x86_64
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  Installing : iptables-lib-1.8.8-3.amzn2023.0.2.x86_64
  Running scriptlet: iptables-lib-1.8.8-3.amzn2023.0.2.x86_64
  Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
  Installing : docker-25.0.6-1.amzn2023.0.2.x86_64
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
```

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64  docker-25.0.6-1.amzn2023.0.2.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnftnl-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-22-81 ec2-user]# 
```

3)Install in all devices

Computer 1

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64  docker-25.0.6-1.amzn2023.0.2.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnftnl-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-29-157 ~]$ 
```

Computer2

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64  docker-25.0.6-1.amzn2023.0.2.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnftnl-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-23-208 ~]$ 
```

i-09a31ba0572c10bdb (Computer2)
PublicIPs: 34.204.51.153 PrivateIPs: 172.31.23.208

4)Configure Docker Daemon:

You need to configure Docker to use the systemd cgroup driver.

Run the following commands:`cd /etc/docker`

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": [ "native.cgroupdriver=systemd" ],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
```

```
[root@ip-172-31-22-81 docker]# cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

5)Enable and Start Docker:

Start Docker by running:

```
bash
Copy code
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
[root@ip-172-31-22-81 docker]# sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-22-81 docker]# ]
```

Step 3: Installing Kubernetes

For installing kubernetes, we will be using kubeadm, a framework used for creating kubernetes clusters using command line.

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm>
The following will be visible when you visit the website.

The screenshot shows the official Kubernetes documentation website. The left sidebar has a search bar and a navigation menu with sections like Documentation, Getting started, Production environment, and a detailed 'Installing Kubernetes with deployment tools' section which is currently selected. The main content area has a note about package repositories for minor versions, links for Debian-based and Red Hat-based distributions, and instructions for using kubeadm without a package manager. It includes a code snippet for setting SELinux to permissive mode and a cautionary note about the implications of doing so.

1) Prepare the System:

Configure SELinux to run in permissive mode to avoid permission issues during the Kubernetes setup:

```
bash
```

Copy code

```
sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
```

```
/etc/selinux/config
```

```
[root@ip-172-31-22-81 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-22-81 ec2-user]# 
```

Computer 1

```
[ec2-user@ip-172-31-28-157 ~]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-28-157 ~]$ 
```

Computer 2

```
[ec2-user@ip-172-31-23-208 ~]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-23-208 ~]$ 
```

2)Add the Kubernetes Repo:

Create a Kubernetes repository by running:

```
bash
Copy code
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

```
[root@ip-172-31-22-81 ec2-user]# 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/repo/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/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-22-81 ec2-user]# ]
```

i-0d3f7911e0aabcc35 (Master)

PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81

```
[root@ip-172-31-22-81 ec2-user]# sudo yum update
[====] ----- B/s | 0 B      --:-- ETA
90 kB/s | 17 kB      00:00
Kubernetes
Kubernetes
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-22-81 ec2-user]# ]
```

yum repolist

This command shows the repositories created on the machine.

```
[root@ip-172-31-22-81 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
kubernetes                               Kubernetes
[root@ip-172-31-22-81 ec2-user]# ]
```

3)Install Kubernetes Components:

Now, install the required Kubernetes tools by executing:

bash

Copy code

```
sudo yum install -y kubelet kubeadm kubectl
--disablescludes=kubernetes
```

```
[root@ip-172-31-22-81 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:06 ago on Sun Sep 15 07:47:07 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size
|=====|
| Installing:
|   kubelet          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubeadm         | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubectl          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 17 M
| Installing dependencies:
|   contrack-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 download size: 53 M
| Installed size: 292 M
| Downloading Packages:
| (1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 432 kB/s | 24 kB 00:00
| (2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm 374 kB/s | 24 kB 00:00
```

```
[ec2-user@ip-172-31-28-157 ~]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:08 ago on Sun Sep 15 07:47:52 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size
|=====|
| Installing:
|   kubelet          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubeadm         | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubectl          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 17 M
| Installing dependencies:
|   contrack-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 download size: 53 M
| Installed size: 292 M
| Downloading Packages:
| (1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 426 kB/s | 24 kB 00:00
```

```
[ec2-user@ip-172-31-23-208 ~]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:03:07 ago on Sun Sep 15 07:47:58 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size
|=====|
| Installing:
|   kubelet          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubeadm         | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubectl          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 17 M
| Installing dependencies:
|   contrack-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 download size: 53 M
| Installed size: 292 M
| Downloading Packages:
| (1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 455 kB/s | 24 kB 00:00
```

Configure System Settings:

Run the following to configure a network bridge:

`bash`

`Copy code`

`sudo swapoff -a`

`echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a`

`/etc/sysctl.conf`

`sudo sysctl -p`

```
[root@ip-172-31-22-81 ec2-user]# 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
[root@ip-172-31-22-81 ec2-user]#
```

i-0d3f7911e0aabcc35 (Master)

PublicIPs: 54.242.215.34 PrivateIPs: 172.31.22.81

PERFORM THE FOLLOWING ON ONLY THE MASTER MACHINE

Initialize Kubernetes:

Initialize your Kubernetes cluster with the following command:

bash

Copy code

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=NumCPU,Mem
```

```
[root@ip-172-31-22-81 ec2-user]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=all
I0915 07:56:11.461747    30220 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
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0915 07:56:11.697643    30220 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container
nended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificatebir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal kubernetes kubernetes.default kub
[10.96.0.1 172.31.22.81]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal localhost] and IPs [172.31.22.8
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-22-81.ec2.internal localhost] and IPs [172.31.22.81
[certs] Generating "etcd/healthcheck-client" certificate and key
```

Configure kubectl Access:

To set up `kubectl` for your non-root user, run:

bash

Copy code

```
mkdir -p $HOME/.kube
```

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

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

```
[root@ip-172-31-22-81 ec2-user]# mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-22-81 ec2-user]#
```

To check whether nodes are connected, run the command

- `kubectl get nodes`

This output shows only master is connected right now.

Kubtectl get nodes

```
[root@ip-172-31-22-81 ec2-user]# kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-22-81.ec2.internal   NotReady  control-plane  4m52s   v1.30.5
```

Perform this Only on node Machines

```
[root@ip-172-31-22-81 ec2-user]# sudo yum install socat -y
Last metadata expiration check: 0:19:33 ago on Sun Sep 15 07:47:07 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version   | Repository | Size
=====
| Installing:      |             |           |            |       |
| socat          | x86_64      | 1.7.4.2-1.amzn2023.0.2 | amazonlinux | 303
| transaction Summary |           |           |            |       |
| Install 1 Package |           |           |            |       |
| Total download size: 303 k |           |           |            |       |
| Installed size: 1.1 M |           |           |            |       |
| Downloading Packages: |           |           |            |       |
| socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm |           |           |            |       |
|                                         2.4 MB/s | 303 kB | 00:00
```

This is the token which we got

```
172.31.22.81:6443 --token 9wf1pg.1xgp88wof0wpslc9 \
--discovery-token-ca-cert-hash
sha256:b4efc86172e4999d3d1e530147cc26705f6543cd48689416874811c98b2
a325f
```

Put this command on the node machine to connect

Now go Back to the master machine and write ‘kubectl get nodes’

```
[root@ip-172-31-22-81 docker]# kubectl get nodes
NAME                 STATUS   ROLES      AGE     VERSION
ip-172-31-22-81.ec2.internal   Ready    control-plane   4m13s   v1.30.5
ip-172-31-28-157.ec2.internal Not Ready   computer 1   2m24s   v1.30.5
ip-172-31-23-208.ec2.internal Not Ready   computer 2   2m17s   v1.30.5
```

We will see nodes are <NOT READY> to change it install a CNI plugin on Master Machine

kubectl apply -f

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

Now ‘Run kubectl get nodes’

```
[root@ip-172-31-22-81 docker]# kubectl get nodes
NAME                 STATUS   ROLES      AGE     VERSION
ip-172-31-22-81.ec2.internal   Ready    control-plane   9m33s   v1.30.5
ip-172-31-28-157.ec2.internal Ready    computer 1   7m24s   v1.30.5
ip-172-31-23-208.ec2.internal Ready    computer 2   7m17s   v1.30.5
```

Conclusion

In this experiment, we explored the architecture of a Kubernetes cluster and successfully deployed it on AWS EC2 instances, consisting of a master node and two worker nodes. After setting up Docker, Kubernetes components (kubelet, kubeadm, kubectl), and containerd across all nodes, the cluster was initialized by configuring the master node and joining the worker nodes. Initially, the nodes showed a "NotReady" status, which was corrected by implementing the Calico networking solution. Additionally, nodes were labeled appropriately based on their roles. By the end of the process, all nodes were fully operational, confirming that the Kubernetes cluster was configured correctly .

Experiment 4

Aim: To set up Kubectl for managing a Kubernetes cluster and deploy a basic application.

Theory: Kubernetes, which originated from Google, is a popular open-source platform for managing containerized applications. It streamlines the scaling, deployment, and maintenance of containers, ensuring resilience and flexibility. It's become a standard in the industry for orchestrating containers, with contributions from top technology companies through the Cloud Native Computing Foundation (CNCF).

Kubernetes Deployment:

This refers to a resource in Kubernetes that allows for rolling updates and rollbacks of applications. It ensures that the correct number of pods are running, maintaining desired configurations at all times.

Requirements:

- **EC2 Instance:** A t2.medium instance with at least 2 CPUs is necessary to accommodate Kubernetes' resource needs.
 - **Minimum configuration:**
 - Instance Type: t2.medium
 - CPUs: 2
 - Memory: Suitable for container operations

Step 1:

Log in to your AWS account and launch an EC2 instance. Choose Ubuntu as the AMI and set the instance type to t2.medium. Generate an RSA key in `.pem` format, and move it to a secure folder on your system.

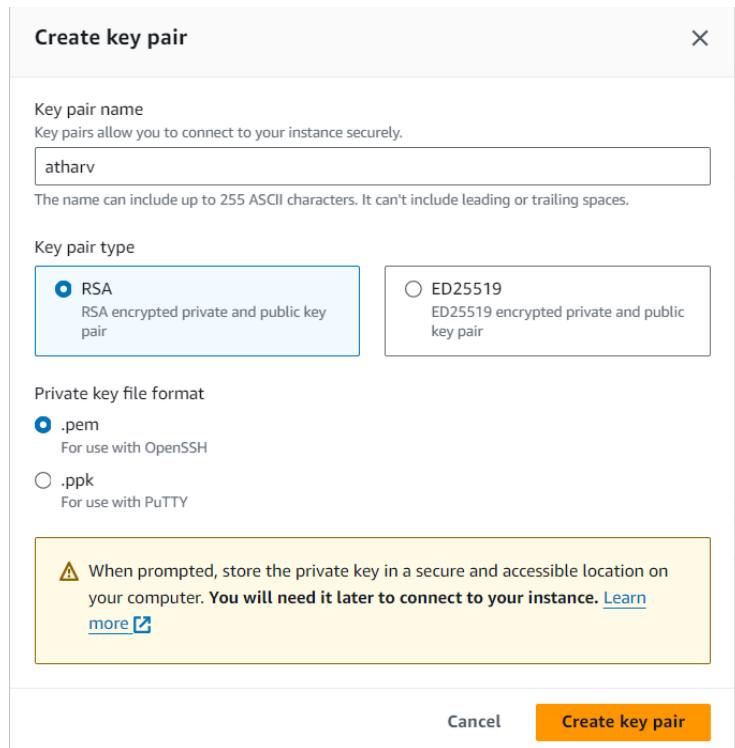
Note: Kubernetes requires at least 2 CPUs, so ensure you select the t2.medium instance. Be mindful to terminate the instance after the task is complete, as it isn't covered under the free tier.

The screenshot shows the 'Launch an instance' step of the AWS EC2 wizard. In the 'Name and tags' section, the name 'AtharvNikam' is entered. In the 'Application and OS Images (Amazon Machine Image)' section, the 'Quick Start' tab is selected, showing recent AMIs like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar is available to browse more AMIs from AWS Marketplace.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone	Public IP	Public IPv4	Elastic IP
AtharvNikam	i-0b6c3b26...	Running	t2.micro	Initializing	View alarms	us-east-1d	ec2-54-89-...	54.89.11.80	-

Step 2:

Connect to your EC2 instance via SSH. Open the terminal in the directory where the `.pem` key is stored, and run the SSH command to access the instance.



```
PS D:\Key> ssh -i "atharv.pem" ubuntu@ec2-54-83-70-136.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 Mon Sep 16 05:30:31 UTC 2024

 System load:  0.24           Processes:          159
 Usage of /:   55.6% of 6.71GB  Users logged in:     1
 Memory usage: 20%            IPv4 address for enX0: 172.31.30.208
 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.

135 updates can be applied immediately.
41 of these updates are standard security updates.
```

Name:Atharv Nikam

Div:D15C

Roll:36

Step 3:

Install Docker on your EC2 instance using the following commands:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
```

```
Ubuntu@ip-172-31-30-208:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-30-208:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-sr entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:3 https://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports 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 InRelease [126 kB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [123 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8264 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [365 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [150 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.3 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [317 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [61.5 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
```

```
Ubuntu@ip-172-31-30-208:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Reading package lists... Done
w: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
w: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroups-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 MB of additional disk space will be used.
get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 kB]
get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 kB]
get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.1-1~ubuntu.24.04~noble [15.0 kB]
get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.1-1~ubuntu.24.04~noble [25.6 kB]
get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.1-1~ubuntu.24.04~noble [9571 kB]
get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 kB]
Fetched 122 MB in 2s (66.5 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
```

```
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4ubuntu2) ...
Processing triggers for libc-bin (2.39-0ubuntu0.2) ...
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-172-31-30-208:~$
```

Configure Docker to use the systemd cgroup driver:

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"]
} EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-30-208:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"] } EOF
{
"exec-opts": ["native.cgroupdriver=systemd"] }ubuntu@ip-172-31-30-208:~$
```

Enable and restart Docker:

```
sudo systemctl enable docker
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-30-208:~$
```

Install Kubernetes using the following commands:

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

Name:Atharv Nikam

Div:D15C

Roll:36

```
ubuntu@ip-172-31-30-208:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keys/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-30-208:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-30-208:~$
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get update
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]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 132 kB in 1s (218 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-30-208:~$
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 133 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (81.3 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68007 files and directories currently installed.)
ubuntu@ip-172-31-30-208:~$
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-30-208:~$
```

Step 5:

To initialize the Kubernetes cluster, run:

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

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-30-208:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0916 04:01:46.391802    6909 checks.go:1080] [preflight] WARNING: Couldn't create the interface
validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error
    [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime
e runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-30-208:~$
```

If you encounter any errors, such as missing container runtimes, install containerd:

```
sudo apt-get install -y containerd
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd
```

```
ubuntu@ip-172-31-30-208:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 133 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (82.5 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4.1) ...
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam

Div:D15C

Roll:36

```
ubuntu@ip-172-31-30-208:~$ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

[metrics]
  address = ""
  grpc_histogram = false

[plugins]

  [plugins."io.containerd.gc.v1.scheduler"]
    deletion_threshold = 0
    mutation_threshold = 100
    pause_threshold = 0.02
    schedule_delay = "0s"
    startup_delay = "100ms"

  [plugins."io.containerd.grpc.v1.cri"]
    cdi_spec_dirs = ["/etc/cdi", "/var/run/cdi"]
```

```
ubuntu@ip-172-31-30-208:~$ sudo systemctl restart containerd
ubuntu@ip-172-31-30-208:~$ sudo systemctl enable containerd
ubuntu@ip-172-31-30-208:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
  Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-09-16 04:11:07 UTC; 22s ago
    Docs: https://containerd.io
  Main PID: 7659 (containerd)
    Tasks: 7
      Memory: 14.0M (peak: 14.7M)
        CPU: 107ms
      CGroup: /system.slice/containerd.service
              └─7659 /usr/bin/containerd

Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807160348Z" level=info msg="Start subscribing containerd event"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807213129Z" level=info msg="Start recovering state"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807163270Z" level=info msg=serving... address=/run/containerd/containerd.sock.ttrpc
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807351018Z" level=info msg=serving... address=/run/containerd/containerd.sock
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807469775Z" level=info msg="Start event monitor"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807526625Z" level=info msg="Start snapshots syncer"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807580042Z" level=info msg="Start cnetwork conf syncer for default"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807593159Z" level=info msg="Start streaming server"
Sep 16 04:11:07 ip-172-31-30-208 containerd[7659]: time="2024-09-16T04:11:07.807763589Z" level=info msg="containerd successfully booted in 0.028456s"
Sep 16 04:11:07 ip-172-31-30-208 systemd[1]: Started containerd.service - containerd container runtime.
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam

Div:D15C

Roll:36

Set up your Kubernetes configuration:

```
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-172-31-25-17 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
[ec2-user@ip-172-31-25-17 docker]$ █
```

Install a networking plugin:

```
kubectl apply -f
```

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
ubuntu@ip-172-31-30-208:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-30-208:~$
```

Step 7:

To deploy an Nginx server, use the following commands:

```
ubuntu@ip-172-31-30-208:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-30-208:~$
```

Name:Atharv Nikam

Div:D15C

Roll:36

```
ubuntu@ip-172-31-30-208:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fzkg4   0/1     Pending   0          30s
nginx-deployment-d556bf558-qvghq   0/1     Pending   0          30s
ubuntu@ip-172-31-30-208:~$
```

Forward the port to access the server:

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
```

```
ubuntu@ip-172-31-30-208:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-30-208:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-30-208:~$
```

If the pod is in a pending state, remove the control-plane taint:

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

```
ubuntu@ip-172-31-30-208:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
```

```
ubuntu@ip-172-31-30-208:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-30-208   Ready   control-plane   39m   v1.31.1
ubuntu@ip-172-31-30-208:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fzkg4   1/1     Running   0          23m
nginx-deployment-d556bf558-qvghq   1/1     Running   0          23m
ubuntu@ip-172-31-30-208:~$
```

Step 8:

Finally, verify the Nginx server is running:

```
ubuntu@ip-172-31-30-208:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Mon, 16 Sep 2024 05:04:04 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**, your Nginx deployment is successful.

Conclusion:

Through this experiment, Kubernetes was successfully installed on an EC2 instance, and an Nginx server was deployed. The process included troubleshooting common issues like pod states and container runtimes, ensuring a stable deployment environment.

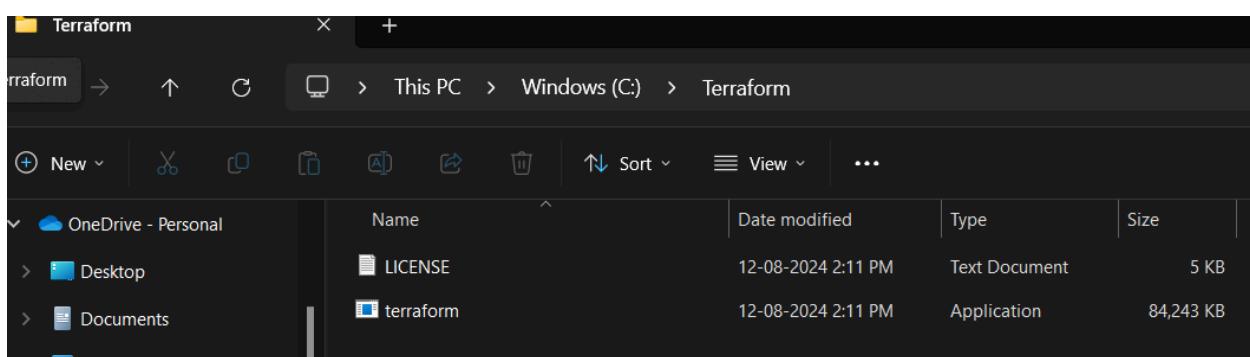
Experiment 5

1) Downloading Terraform from official website

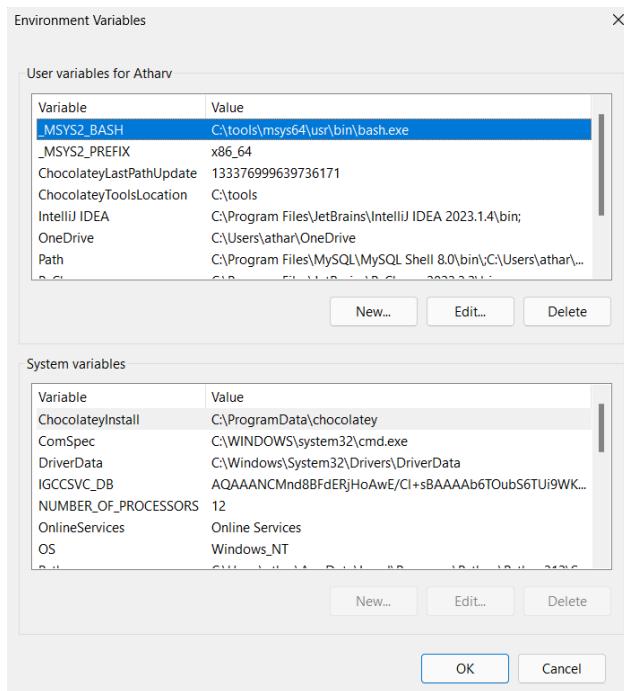
<https://developer.hashicorp.com/terraform/install>

The screenshot shows the Terraform official website's "Install Terraform" section. On the left, a sidebar lists "Operating Systems" (macOS, Windows, Linux, FreeBSD, OpenBSD, Solaris), "Release information", and "Next steps". The main content area is divided into two sections: "macOS" and "Windows".
macOS: Contains a "Package manager" section with the command "brew tap hashicorp/tap" and "brew install hashicorp/tap/terraform", and a "Binary download" section for "AMD64" (Version 1.9.4) and "ARM64" (Version 1.9.4).
Windows: Contains a "Binary download" section for "386" (Version 1.9.4) and "AMD64" (Version 1.9.4).

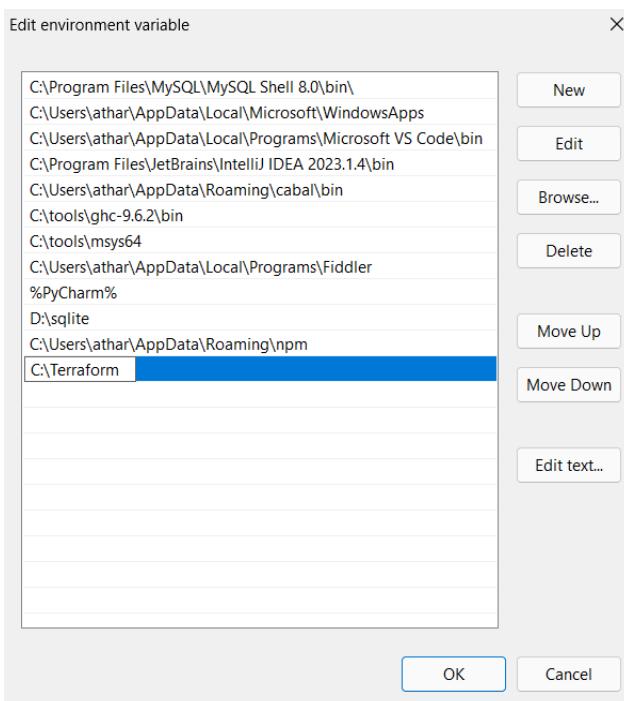
2) Save it into Windows C



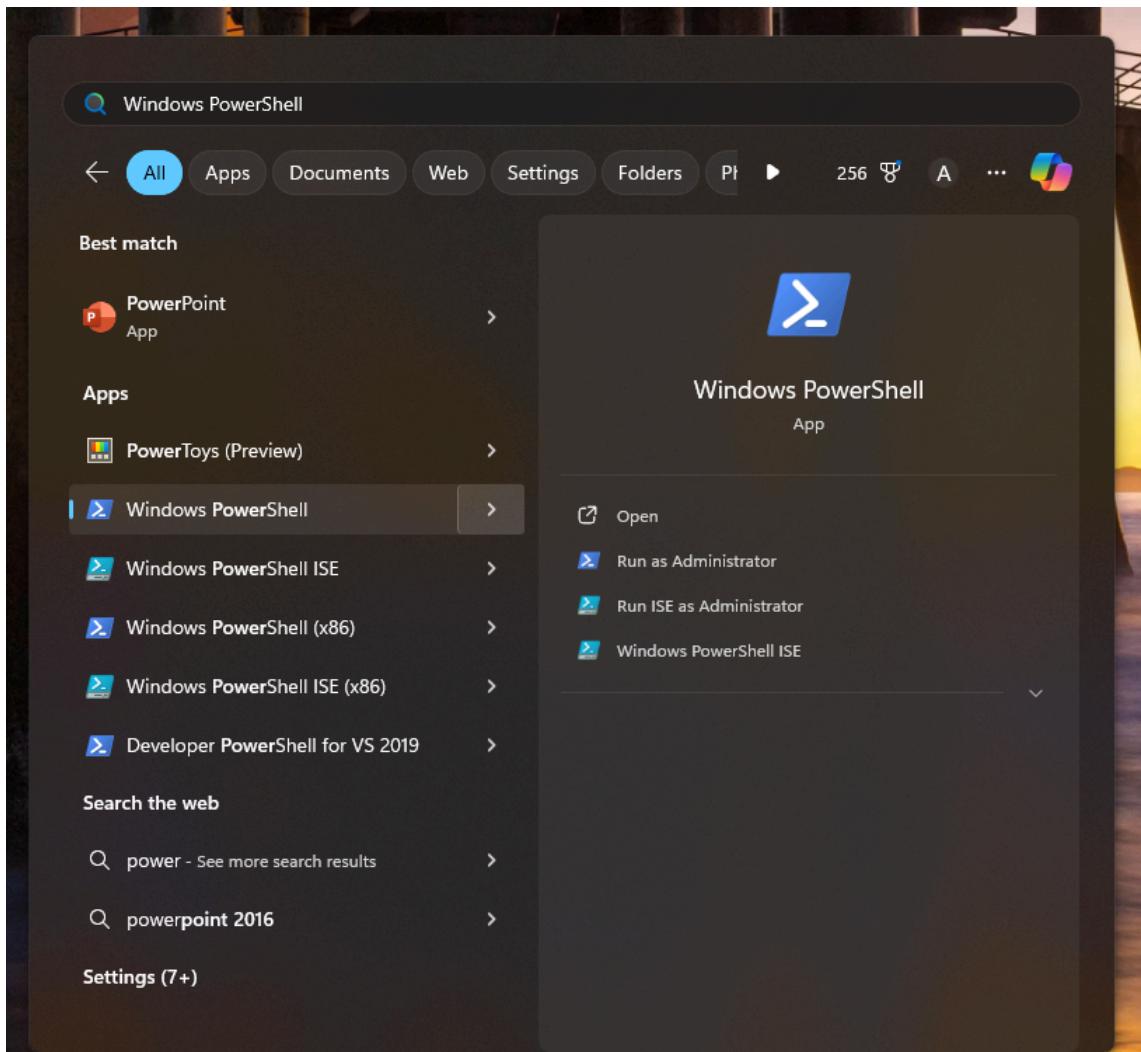
3)Change Environment Variables



4) Click on path and add new C:\Terraform



5)search for powershell and open it



5)To Check if terraform is installed open powershell and type Terraform --version

```
PS C:\Users\athar> Terraform --version
Terraform v1.9.4
on windows_386
PS C:\Users\athar>
```

6)To see all terraform commands type Terraform –help

```
PS C:\Users\athar> Terraform --help
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint      Mark a resource instance as not fully functional
  test       Execute integration tests for Terraform modules
  untaint   Remove the 'tainted' state from a resource instance
  version    Show the current Terraform version
  workspace  Workspace management

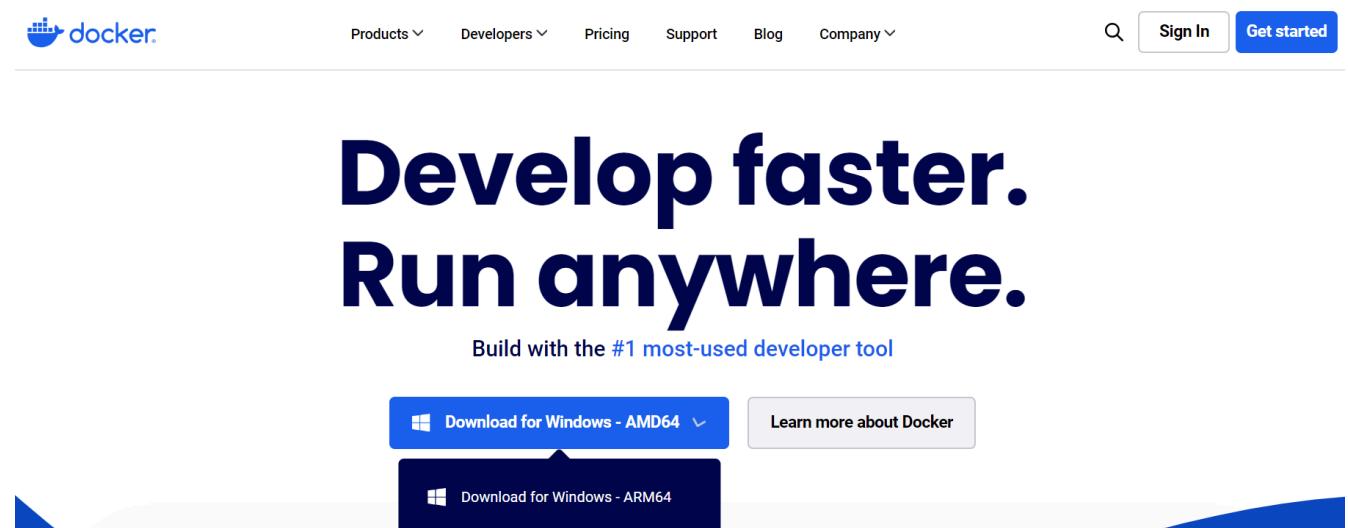
Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
              given subcommand.
  -help       Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.
PS C:\Users\athar> |
```

Exp 6

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

Step1: check if Docker is installed in your system to check open your powershell and type docker

If not then install docker from <https://www.docker.com/> and download it



Check if docker has installed by opening command prompt and typing
docker —version

```
Microsoft Windows [Version 10.0.22621.4037]
(c) Microsoft Corporation. All rights reserved.

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

C:\Users\athar>
```

Type docker to see all the commands

```
C:\Users\athar>docker

Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

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

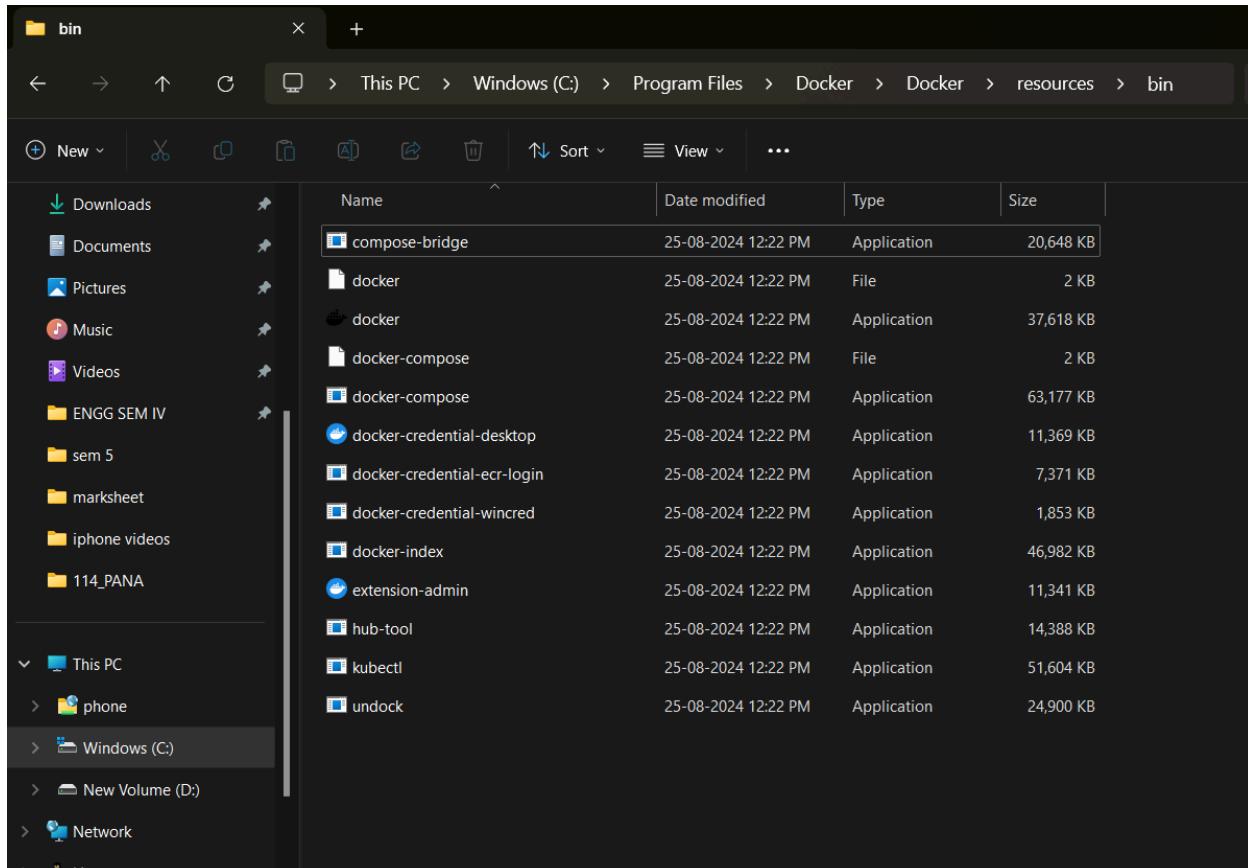
Management Commands:
  builder   Manage builds
  buildx*   Docker Buildx
  checkpoint Manage checkpoints
  compose*  Docker Compose
  container  Manage containers
  context    Manage contexts
  debug*    Get a shell into any image or container
  desktop*  Docker Desktop commands (Alpha)
  dev*      Docker Dev Environments
  extension* Manages Docker extensions
  feedback* Provide feedback, right in your terminal!
  image     Manage images
  init*    Creates Docker-related starter files for your project
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks
  plugin    Manage plugins
  sbom*    View the packaged-based Software Bill Of Materials (SBOM) for an image
  scout*   Docker Scout
  system    Manage Docker
  trust     Manage trust on Docker images
  volume    Manage volumes

Swarm Commands:
  config   Manage Swarm configs
  node     Manage Swarm nodes
  secret   Manage Swarm secrets
  service  Manage Swarm services
  stack    Manage Swarm stacks
  swarm    Manage Swarm

Commands:
  attach    Attach local standard input, output, and error streams to a running container
  commit   Create a new image from a container's changes
  cp       Copy files/folders between a container and the local filesystem
  create   Create a new container
  diff     Inspect changes to files or directories on a container's filesystem
  events   Get real time events from the server
  export   Export a container's filesystem as a tar archive
  history  Show the history of an image
  import   Import the contents from a tarball to create a filesystem image
  inspect  Return low-level information on Docker objects
  kill     Kill one or more running containers
  load    Load an image from a tar archive or STDIN
  logs    Fetch the logs of a container
```

If you are not getting the version try these steps

Go to your file explorer and go to bin folder of your docker file or just paste this
C:\Program Files\Docker\Docker\resources\bin



Access the 'Edit the System Environment Variables' option on your computer.
Then, select Environment Variables.

Look for a 'Path' variable under System variables. If it's there, select it and click on Edit.

If it's not there, click on New and create a 'Path' variable.

If the variable already exists, click on Edit and then on New to open a text box.
Paste the path you copied into the box and click OK to close all the windows.

Now do step 1 again to check the docker version

Step 2: Create a folder Terraform scripts and then inside that create docker file and at last create a docker.ts file

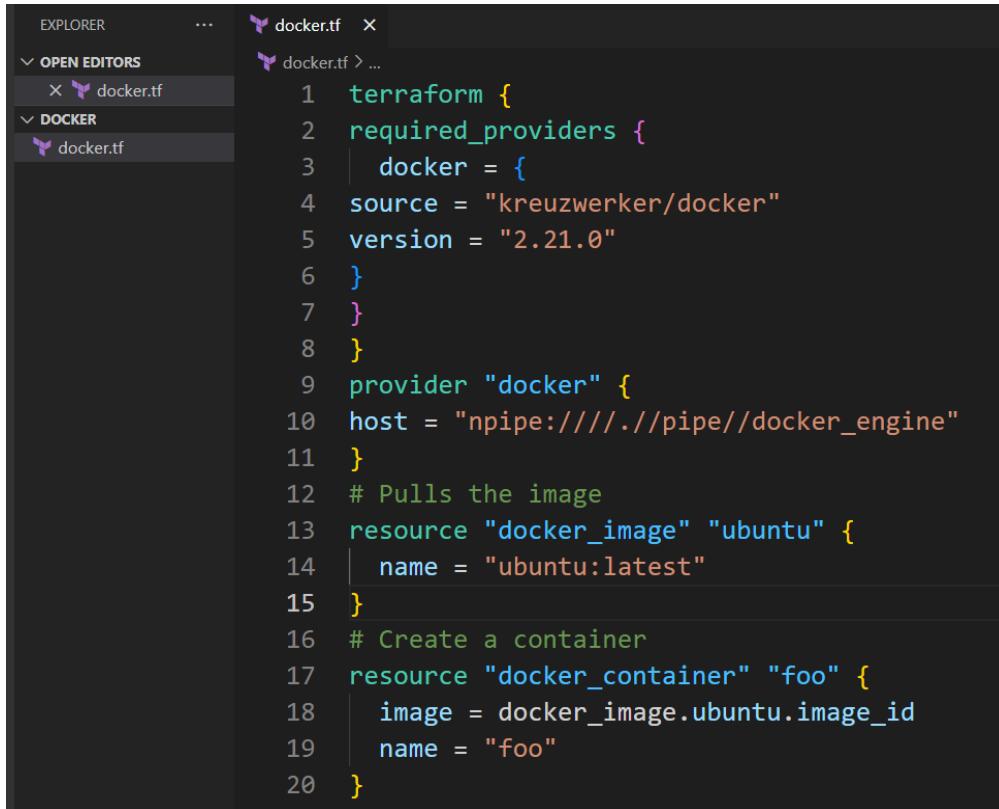
After that open that folder in visual studio and paste this code

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

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

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

# Create a container
resource "docker_container" "foo"
  { image =
    docker_image.ubuntu.image_idname =
    "foo"
}
```



The screenshot shows the VS Code interface with the Explorer sidebar on the left. Under 'OPEN EDITORS', there is a file named 'docker.tf'. Under 'DOCKER', there is also a file named 'docker.tf'. The main editor area displays the following Terraform code:

```

1  terraform {
2    required_providers {
3      docker = {
4        source = "kreuzwerker/docker"
5        version = "2.21.0"
6      }
7    }
8  }
9  provider "docker" {
10   host = "npipe://./pipe/docker_engine"
11 }
12 # Pulls the image
13 resource "docker_image" "ubuntu" {
14   name = "ubuntu:latest"
15 }
16 # Create a container
17 resource "docker_container" "foo" {
18   image = docker_image.ubuntu.image_id
19   name = "foo"
20 }

```

Step 3: Open the terminal and go to folder where docker.tf is present

```

PS D:\all code\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.

```
PS D:\all code\Terraform Scripts\Docker>
```

Step 4:Run the Command ‘terraform plan’ This would create a execution plan and you could see an overview of your plan

```
PS D:\all code\Terraform Scripts\Docker> terraform plan

Planning failed. Terraform encountered an error while generating this plan.

Error: Error pinging Docker server: error during connect: Get "http://%2F%2F.%2F%2Fpipe%2Fdocker_engine/_ping": open //./pipe//docker_engine: The system cannot find the file specified.

with provider["registry.terraform.io/kreuzwerker/docker"],
on docker.tf line 9, in provider "docker":
  9: provider "docker" {
```

Sometimes the docker engine os not working so on the docker Docker Desktops and try again

Step 5 :Run terraform plan again

```
PS D:\all code\Terraform Scripts\Docker> terraform plan

Terraform used the selected providers to generate the following execution plan.
+ 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)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
```

Step 6:run command terraform apply

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Creation complete after 12s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...

Error: container exited immediately

with docker_container.foo,
on docker.tf line 17, in resource "docker_container" "foo":
17: resource "docker_container" "foo" {
```

The script will give an error because this script took very less time to execute to resolve this issue we have to add this command

'command=[“sleep”, “infinity”]'

Now we will get this

```
PS D:\all code\Terraform Scripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "infinity",
    ]
    + 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           = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data    = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime          = (known after apply)
    + security_opts   = (known after apply)
    + shm_size         = (known after apply)
    + start            = true
    + stdio            = false
}
```

```
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 0s [id=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]
```

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

```
PS D:\all code\Terraform Scripts\Docker> █
```

Docker images before terraform apply

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
PS D:\all code\Terraform Scripts\Docker> █
```

Docker images after terraform apply

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       edbfe74c41f8  3 weeks ago   78.1MB
PS D:\all code\Terraform Scripts\Docker> █
```

Step 7: Now the image is created to destroy that we have to use the command Terraform destroy

```
PS D:\all code\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach           = false -> null
    - command         = [
        - "sleep",
        - "infinity",
    ] -> null
    - cpu_shares      = 0 -> null
    - dns              = [] -> null
    - dns_opts         = [] -> null
    - dns_search       = [] -> null
    - entrypoint       = [] -> null
    - env              = [] -> null
    - gateway          = "172.17.0.1" -> null
    - group_add        = [] -> null
    - hostname         = "305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad"
    - id               = "305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad" -> null
    - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - init              = false -> null
    - ip_address        = "172.17.0.2" -> null
    - ip_prefix_length = 16 -> null
    - ipc_mode          = "private" -> null
    - links             = [] -> null
    - log_driver         = "json-file" -> null
    - log_opts           = {} -> null
    - logs              = false -> null
    - max_retry_count   = 0 -> null
    - memory             = 0 -> null
}
```

Enter yes

```
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=305bdd50733837ad82692315d99bfd8178934f2e5ce0c8407d4744644d11d8ad]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
PS D:\all code\Terraform Scripts\Docker>
```

To check if the images are destroyed check run docker images

```
PS D:\all code\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED     SIZE
PS D:\all code\Terraform Scripts\Docker>
```

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

Prerequisites:**1. Docker:**

Confirm Docker is installed and functioning by running the command:

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

2 . SonarQube Image Installation:

Use the following command to download the SonarQube image via Docker:

```
PS C:\Users\athar> 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

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

3.Jenkins Installation:

Ensure Jenkins is already installed and properly configured on your system

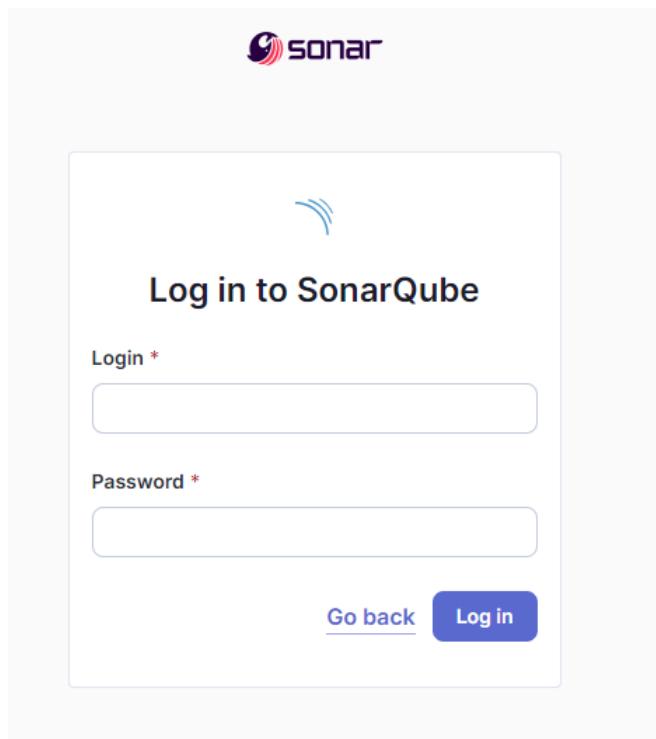
Experiment Steps:**Step 1:**

Run the SonarQube Docker container using the command:

```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar> |
```

Step 2:

After SonarQube starts, navigate to <http://localhost:9000> in your browser.



Step 3:

Log in using the default credentials (**admin/admin**). After logging in, you will be prompted to change the password. Ensure to note the new password.

Update your password

⚠ This account should not use the default password.

Enter a new password
All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

Step 4:

Once logged in, create a new project by clicking on "Create a Local Project." Provide a project name and a project key, then proceed.

The screenshot shows the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, a section titled "How do you want to create your project?" is displayed. It includes a note about benefiting from SonarQube's features and creating a project from a DevOps platform. There are five buttons for importing from different sources: "Import from Azure DevOps" (Setup), "Import from Bitbucket Cloud" (Setup), "Import from Bitbucket Server" (Setup), "Import from GitHub" (Setup), and "Import from GitLab" (Setup). Below these buttons, a note says "Are you just testing or have an advanced use-case? Create a local project." followed by a "Create a local project" button. At the bottom of the page, a yellow warning box states: "⚠ Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

The name of your project's default branch [Learn More](#) 

CancelNext

2 of 2

Set up project for Clean as You Code



The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#) 

Choose the baseline for new code for this project

Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.

Recommended for projects following continuous delivery.

Reference branch

Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

[Back](#)[Create project](#)

Your project has been created. 

Name:Atharv Nikam

Div:D15C

Roll:36

Step 5:

Open Jenkins on its assigned port (http://localhost:<port_number>).

The screenshot shows the Jenkins dashboard with the following details:

- Left sidebar:** Includes links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views.
- Build Queue:** Shows "No builds in the queue."
- Build Executor Status:** Shows 1 Idle and 2 Idle nodes, with one node named "FirstNode" listed as (offline).
- Central Table:** Displays three Jenkins jobs:
 - FirstJob:** Status: N/A, Last Success: N/A, Last Failure: N/A, Last Duration: N/A.
 - MyFirstPipeline:** Status: N/A, Last Success: N/A, Last Failure: N/A, Last Duration: N/A.
 - TomcatProject:** Status: N/A, Last Success: 27 days #1, Last Failure: N/A, Last Duration: 2.1 sec.
- Top right:** Search bar, notifications, user profile (Atharv Nikam), and log out link.

Step 6:

In Jenkins, go to **Manage Jenkins**, search for **SonarQube Scanner for Jenkins**, and install the plugin.

The screenshot shows the Jenkins Manage Jenkins page with the following sections:

- Left sidebar:** Includes links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (selected), My Views, Build Queue, and Build Executor Status.
- Central Content:**
 - System Configuration:** Includes links for System, Tools, Plugins, Nodes, Clouds, Appearance, Security, Credentials, Credential Providers, and Users.
 - Status Information:** Includes links for System Information, System Log, Load Statistics, and About Jenkins.
- Top right:** Search bar, user profile (Atharv Nikam), and log out link.



Step 7:

After installation, navigate to **Manage Jenkins → System**. Under **SonarQube servers**, add your SonarQube server and configure it with the necessary authentication token.

The screenshot shows the Jenkins configuration interface for SonarQube servers. It includes fields for "Name" (set to "atharvsonarqube"), "Server URL" (set to "http://localhost:9000"), and "Server authentication token" (set to "- none -"). There is also an "Advanced" dropdown and a "Add SonarQube" button.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name

atharvsonarqube

Server URL

Default is <http://localhost:9000>

<http://localhost:9000>

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

Advanced ▾

Add SonarQube

Step 8:

Next, go to **Manage Jenkins → Tools**. Under **SonarQube Scanner**, choose the latest configuration and enable automatic installation.

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. The 'SonarQube Scanner' configuration page is displayed. A new configuration is being added with the name 'atharvsonarqube'. The 'Install automatically' checkbox is checked. The 'Install from Maven Central' section is expanded, showing the version 'SonarQube Scanner 6.2.0.4584'. Below this, there is an 'Add Installer' button. At the bottom of the configuration form, there is a 'Save' button. Above the configuration form, there is a separate 'Ant installations' section with a 'Save' and 'Apply' button.

Step 9:

Create a new item in Jenkins, selecting a **Freestyle Project**.

The screenshot shows the Jenkins 'New Item' dialog. In the 'Select an item type' section, the 'Freestyle project' option is selected. It is described as a 'Classic general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.' Other options shown include 'Pipeline', 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. At the bottom of the dialog, there is an 'OK' button.

Step 10:

In the Source Code Management section, use this GitHub repository:

https://github.com/shazforiot/MSBuild_firstproject

This repository contains a sample project to test.

Step 11:

In the Build section, add the SonarQube Scanner. Enter the required SonarQube project details such as project key, login credentials, source path, and the server URL.

sonar.projectKey=atharvsonarqube

sonar.login=admin

sonar.password=atharv@123

sonar.sources=.

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

Step 12:

Grant the local user (e.g., admin) permission to execute analysis on SonarQube by navigating to:

http://localhost:<port_number>/admin/permissions

Check the "Execute Analysis" box.

Global Permissions

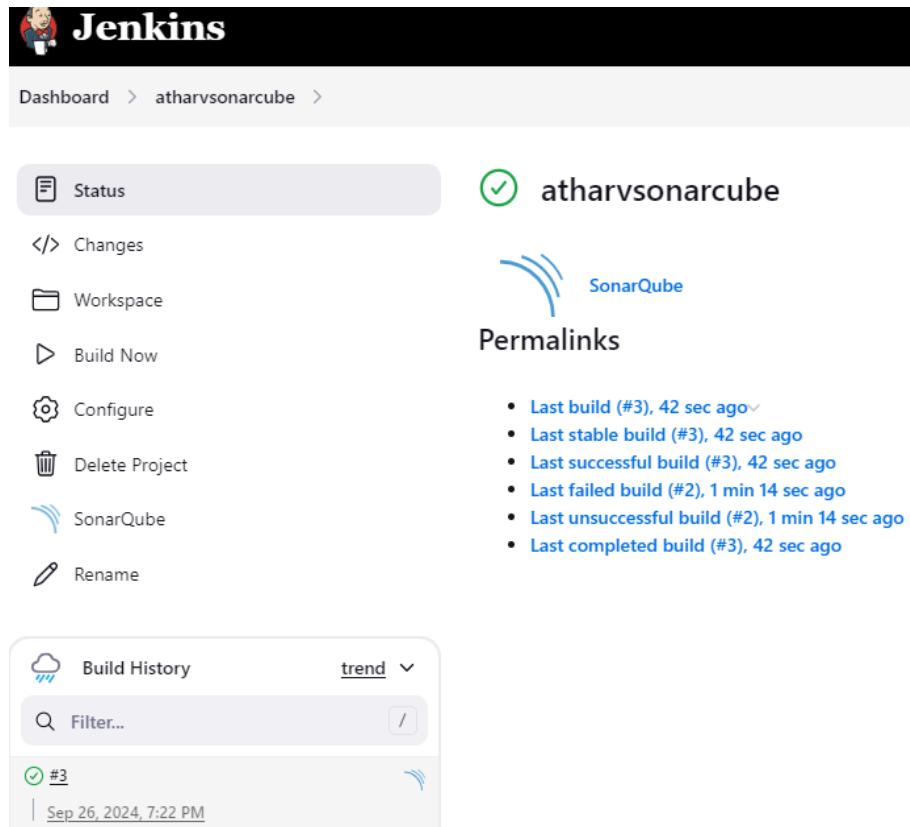
Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

All Users Groups Search for users or groups...

		Administer System	Administer	Execute Analysis	Create
 sonar-administrators	System administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
 sonar-users	Every authenticated user automatically belongs to this group	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
 Anyone <small>DEPRECATED</small>	Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
 A Administrator admin		<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects

Step 13:

Go back to Jenkins and trigger a build by selecting **Build Now** for the project you created.



The screenshot shows the Jenkins dashboard for the project "atharvsonarcube". The left sidebar contains project management options: Status (selected), Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area displays the project name "atharvsonarcube" with a green checkmark icon. Below it is the SonarQube logo and the word "Permalinks". A list of recent builds is shown:

- Last build (#3), 42 sec ago
- Last stable build (#3), 42 sec ago
- Last successful build (#3), 42 sec ago
- Last failed build (#2), 1 min 14 sec ago
- Last unsuccessful build (#2), 1 min 14 sec ago
- Last completed build (#3), 42 sec ago

At the bottom, there is a "Build History" section with a "trend" dropdown set to "trend", a "Filter..." search bar, and a list of builds. The first build is highlighted with a green checkmark and labeled "#3". The timestamp for this build is "Sep 26, 2024, 7:22 PM".

Console Output

```

Started by user Atharv Nikam
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[atharvsonarcube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarcube\bin\sonar-scanner.bat -Dsonar.hos
Dsonar.projectKey=atharvsonarcube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=atharv@123 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
19:23:00.321 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
19:23:00.321 INFO Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarcube\bin\..
19:23:00.321 INFO Project root configuration file: NONE
19:23:00.339 INFO SonarScanner CLI 6.2.0.4584
19:23:00.339 INFO Java 20 Oracle Corporation (64-bit)
19:23:00.339 INFO Windows 11 10.0 amd64
19:23:00.354 INFO User cache: C:\WINDOWS\system32\config\systemprofile\.sonar\cache
19:23:01.038 INFO JRE provisioning: os[windows], arch[amd64]
19:23:04.310 INFO Communicating with SonarQube Server 10.6.0.92116
19:23:04.678 INFO Starting SonarScanner Engine...
19:23:04.678 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
19:23:05.407 INFO Load global settings

```

```

19:23:21.882 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
19:23:21.882 INFO Sensor C# File Caching Sensor [csharp] (done) | time=0ms
19:23:21.882 INFO Sensor Zero Coverage Sensor
19:23:21.882 INFO Sensor Zero Coverage Sensor (done) | time=0ms
19:23:21.882 INFO SCM Publisher SCM provider for this project is: git
19:23:21.882 INFO SCM Publisher 4 source files to be analyzed
19:23:22.262 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=380ms
19:23:22.262 INFO CPD Executor Calculating CPD for 0 files
19:23:22.262 INFO CPD Executor CPD calculation finished (done) | time=0ms
19:23:22.269 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
19:23:22.409 INFO Analysis report generated in 53ms, dir size=201.0 kB
19:23:22.441 INFO Analysis report compressed in 16ms, zip size=22.5 kB
19:23:22.615 INFO Analysis report uploaded in 171ms
19:23:22.615 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarcube
19:23:22.615 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
19:23:22.615 INFO More about the report processing at http://localhost:9000/api/ce/task?id=3cfcbc0c-0055-497d-a657-1cad4d6e2a05
19:23:22.615 INFO Analysis total time: 14.707 s
19:23:22.615 INFO SonarScanner Engine completed successfully
19:23:22.661 INFO EXECUTION SUCCESS
19:23:22.661 INFO Total time: 22.340s
Finished: SUCCESS

```

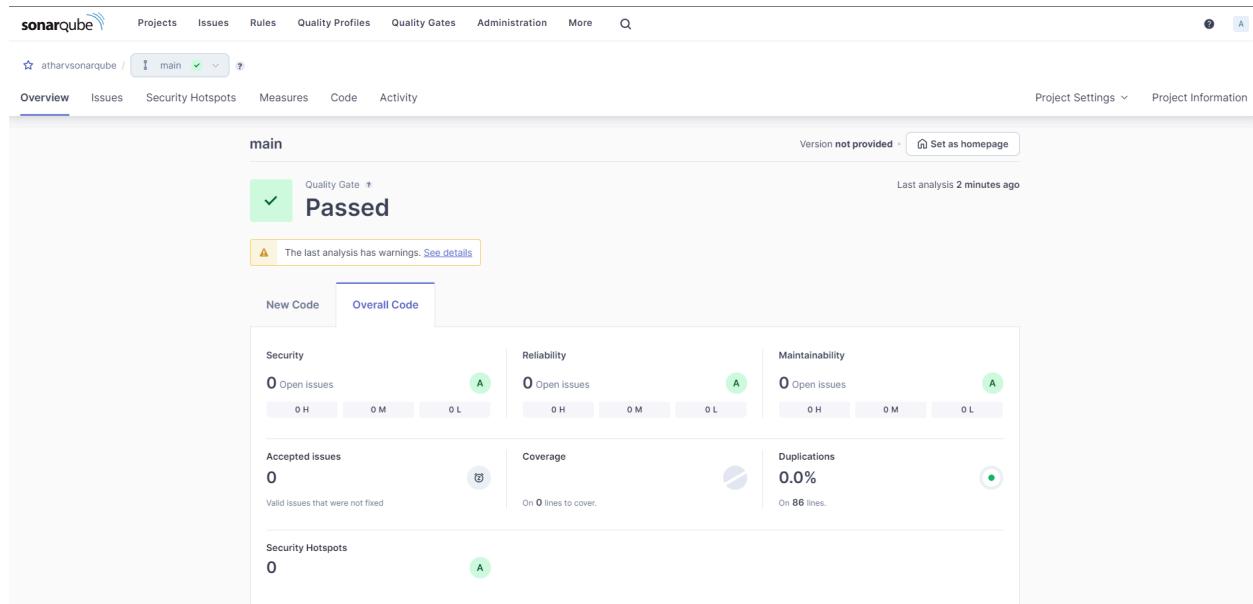
Name:Atharv Nikam

Div:D15C

Roll:36

Step 14:

Once the build process is complete, head back to SonarQube and verify the analysis results linked to your project.



Conclusion:

Through this experiment, we explored how to set up and perform Static Application Security Testing (SAST) using Jenkins in combination with SonarQube. By leveraging Docker, we utilized a SonarQube container without needing a local installation. Following the setup of Jenkins and SonarQube, we analyzed a sample project from GitHub for potential vulnerabilities. Once the project was built, the analysis confirmed the security status, providing valuable insight into the effectiveness of SAST in the CI/CD pipeline.

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.

Prerequisites:

1. Download Sonar Scanner:

Access the SonarQube documentation and download the SonarQube scanner CLI from this link:

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>

The screenshot shows the SonarScanner CLI page on the SonarQube documentation site. The left sidebar has a search bar and links to 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code' (selected), 'Project analysis setup', 'Scanners' (selected), 'Scanner environment', 'SonarScanner CLI' (selected), 'SonarQube extension for Azure DevOps', 'SonarQube extension for Jenkins', 'SonarScanner for .NET', 'SonarScanner for Maven', 'SonarScanner for Gradle', 'SonarScanner for NPM' (selected), 'SonarScanner for Ant (Deprecated)', 'SonarScanner for Python (Beta)', 'Analysis parameters', and 'Languages'. The main content area has a 'SonarScanner CLI' heading, a table showing version 6.2 (2024-09-17) with support for PKCS12 truststore generated with OpenSSL, download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM), and release notes. Below the table, it says the SonarScanner CLI is the scanner to use when there is no specific scanner for your build system. It also mentions that the SonarScanner does not yet officially support ARM architecture. A note says SonarScanners run on code that is checked out. On the right, there's a 'START FREE' button and a 'On this page' sidebar with links to 'Configuring your project', 'Running SonarScanner CLI from the zip file', 'Running SonarScanner CLI from the Docker image', 'Scanning C, C++, or Objective-C projects', 'Sample projects', 'Alternatives to sonar-project.properties', 'Alternate analysis directory', 'Advanced configuration', and 'Troubleshooting'.

2. After downloading, extract the zip file into a designated folder.

Install Docker:

Run the following command to verify Docker is installed:

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

3 .Pull SonarQube Docker Image:

Install the SonarQube image by executing:

Copy code

```
docker pull sonarqube
```

```
PS C:\Users\athar> 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

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

4. Ensure Jenkins is installed:

Confirm that Jenkins is installed and configured on your system.

Experiment Steps:

Step 1:

Run the SonarQube Docker container by entering the command below:

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

```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p
9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar> |
```

Step 2:

After SonarQube is running, open your browser and go to <http://localhost:9000>.

Step 3:

Log in to SonarQube using the default credentials:

Username: admin Password: admin

You will be asked to reset the password after logging in for the first time. Set a new password and remember it.

Name:Atharv Nikam

Div:D15C

Roll :36



Log in to SonarQube

Login *

Password *

[Go back](#) [Log in](#)

Update your password

⚠ This account should not use the default password.

Enter a new password
All fields marked with * are required

Old Password *

New Password *

Confirm Password *

[Update](#)

Step 4:

On the SonarQube dashboard, click **Create a Local Project**. Provide a project name and a unique project key.

The screenshot shows the SonarQube dashboard with the 'Create a local project' option selected. The interface includes sections for importing from various platforms (Azure DevOps, Bitbucket Cloud, Bitbucket Server, GitHub, GitLab) and a 'Create a local project' button. A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

1 of 2

Create a local project

Project display name *

 (Valid)

Project key *

 (Valid)

Main branch name *

The name of your project's default branch [Learn More](#)

[Cancel](#) [Next](#)

Your project has been created. (Close)

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

Use the global setting

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

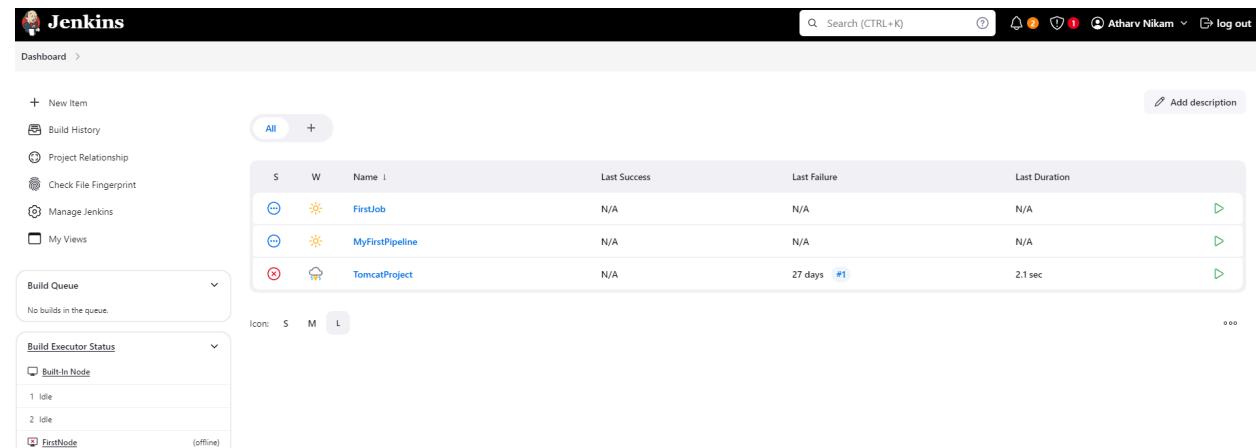
Reference branch
Choose a branch as the baseline for the new code.
Recommended for projects using feature branches.

[Back](#) [Create project](#)

Step 5:

Open Jenkins by navigating to the port on which it is installed:

http://localhost:<port_number>



The screenshot shows the Jenkins dashboard with the following details:

- Header:** Jenkins logo, Search (CTRL+K), Notifications, User Atharv Nikam, Log out.
- Left Sidebar:**
 - New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
 - My Views
- Build Queue:** No builds in the queue.
- Build Executor Status:**
 - Built-In Node
 - 1 idle
 - 2 idle
 - FirstNode (offline)
- Central Area:** A table showing project statistics:

S	W	Name	Last Success	Last Failure	Last Duration
:(☀️	FirstJob	N/A	N/A	N/A
:(☀️	MyFirstPipeline	N/A	N/A	N/A
(X)	☁️	TomcatProject	N/A	27 days #1	2.1 sec

Step 6:

In Jenkins, go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner for Jenkins**. Install the plugin.

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like '+ New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is selected and highlighted in blue), 'My Views', 'Build Queue' (empty), 'Build Executor Status' (with one idle node), and 'FirstNode' (offline). The main content area is titled 'Manage Jenkins' and contains several sections: 'System Configuration' (with 'System', 'Tools', 'Clouds', 'Appearance' sub-links), 'Security' (with 'Security', 'Credentials', 'Credential Providers' sub-links), and 'Status Information' (with 'System Information', 'System Log', 'Load Statistics', 'About Jenkins' sub-links). A prominent orange bar at the top says 'Restore the previous version of Jenkins' with a 'Downgrade to 2.462.1' button. A search bar at the top right says 'Search (CTRL+K)'.

Step 7:

Once installed, head to **Manage Jenkins → System**. Under **SonarQube Servers**, add your SonarQube server, and provide any necessary authentication tokens.

The screenshot shows the Jenkins Manage Jenkins → Plugins page. A search bar at the top left shows 'sonarqube S'. A large blue 'Install' button is at the top right. Below it, a table lists the 'SonarQube Scanner' plugin: it's version 2.17.2, released 7 months and 10 days ago, and is categorized under 'External Site/Tool Integrations'. The 'Install' checkbox is checked. A note below the table states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

Step 8:

Next, under **Manage Jenkins → Tools**, navigate to **SonarQube Scanner** and configure it to automatically install the latest version.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name	atharvsonarqube	X
Server URL	Default is http://localhost:9000 http://localhost:9000	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. - none - + Add	
Advanced		

Add SonarQube

Step 9:

Create a new pipeline item in Jenkins

New Item

Enter an item name

atharvsonarqubetest

Select an item type

**Freestyle project**

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

OK

Step 10:

In the pipeline script section, input the following:

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

    stage('SonarQube Analysis') {
        withSonarQubeEnv('atharvsonarqube') {
            bat """
                <PATH_TO SONARSCANNER_FOLDER>\bin\sonar-scanner.bat ^
                -D sonar.login=<SONARQUBE_LOGIN> ^
                -D sonar.password=<SONARQUBE_PASSWORD> ^
                -D sonar.projectKey=<PROJECT_KEY> ^
                -D sonar.exclusions=vendor/**,resources/**,*/*.java ^
                -D sonar.host.url=http://localhost:9000/
            """
        }
    }
}
```

Definition

Pipeline script

Script ?

```
2 stage('Cloning the GitHub Repo')
3 {
4     git 'https://github.com/shazforiot/GOL.git'
5 }
6 stage('SonarQube analysis') {
7     withSonarQubeEnv('sonarqube') {
8         bat """
9             D:\sonar\sonar-scanner-cli-6.2.0.4584-windows-x64\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat ^
10            -D sonar.login=admin ^
11            -D sonar.password=atharv@123 ^
12            -D sonar.projectKey=atharvsonarqubetest ^
13            -D sonar.exclusions=vendor/**,resources/**,*/*.java ^
14            -D sonar.host.url=http://localhost:9000/
15        """
16    }
17 }
18 }
```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

This script clones a sample Java project from GitHub, which has several issues that SonarQube will detect.

Step 11:

Go back to Jenkins, select the job you just created, and click **Build Now** to run the pipeline.



Permalinks

- [Last build \(#11\), 6 min 16 sec ago](#)
- [Last stable build \(#11\), 6 min 16 sec ago](#)
- [Last successful build \(#11\), 6 min 16 sec ago](#)
- [Last failed build \(#10\), 8 min 31 sec ago](#)
- [Last unsuccessful build \(#10\), 8 min 31 sec ago](#)
- [Last completed build \(#11\), 6 min 16 sec ago](#)

```
20:29:07.725 INFO CPD Executor CPD calculation finished (done) | time=71153ms
20:29:07.910 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:30:07.200 INFO Analysis report generated in 3510ms, dir size=126.4 MB
20:30:16.243 INFO Analysis report compressed in 9029ms, zip size=29.5 MB
20:30:18.716 INFO Analysis report uploaded in 2466ms
20:30:18.723 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarqubetest1
20:30:18.723 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:30:18.723 INFO More about the report processing at http://localhost:9000/api/ce/task?id=e3321ac6-d5a9-4e20-be13-d9d71fb2c392
20:30:31.564 INFO Analysis total time: 5:54.794 s
20:30:31.583 INFO SonarScanner Engine completed successfully
20:30:32.216 INFO EXECUTION SUCCESS
20:30:32.278 INFO Total time: 5:59.482s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Step 12:

Once the build is complete, return to SonarQube to view the analysis of your project. Check for bugs, code smells, duplications, and other metrics related to the quality of your code.

The screenshot shows the SonarQube interface for the project 'atharvsonarqubetest1'. The top navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search bar. Below the navigation is a breadcrumb trail: 'atharvsonarqubetest1 / main'. The main content area has tabs for 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. A prominent green 'Passed' status is displayed, indicating the last analysis was 10 minutes ago. The 'Measures' tab is selected. Key metrics shown are: Security (0 Open issues, A grade), Reliability (68k Open issues, C grade), Maintainability (164k Open issues, A grade), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6% on 759k lines).

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

- **Code Problems**

The screenshot shows the SonarQube 'Measures' page for the same project. The left sidebar under 'Issues' shows categories: New Code (Open Issues: 0), Overall Code (Open Issues: 210,549), and Activity (Confirmed Issues: 0, Accepted Issues: 0, False Positive Issues: 0). The main content area shows the 'New Open Issues' section with 0 results, and a note that new code was added since September 26, 2024. A list of code components is provided, all showing 0 issues: gameoflife-acceptance-tests, gameoflife-build, gameoflife-core, gameoflife-deploy, gameoflife-web, and pom.xml. A footer indicates 6 of 6 items are shown.

● Consistency

The screenshot shows the SonarQube Issues page for the project 'atharvsonarqubetest1'. The 'Issues' tab is selected. On the left, a sidebar shows filters for 'Clean Code Attribute' (Consistency: 197k, Intentionality: 14k, Adaptability: 0, Responsibility: 0) and 'Software Quality' (Security: 0, Reliability: 54k). The main panel displays three issues under the 'gameoflife-core/build/reports/tests/all-tests.html' file:

- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency, user-experience) - Reliability: Open, Not assigned. L1 + 5min effort • 4 years ago • Bug • Major
- Remove this deprecated "width" attribute.** (Consistency, html5) - Maintainability: Open, Not assigned. L9 + 5min effort • 4 years ago • Code Smell • Major
- Remove this deprecated "align" attribute.** (Consistency, html5) - Maintainability: Open, Not assigned. L11 + 5min effort • 4 years ago • Code Smell • Major

● Intentionality

The screenshot shows the SonarQube Issues page for the project 'atharvsonarqubetest1'. The 'Issues' tab is selected. On the left, a sidebar shows filters for 'Clean Code Attribute' (Consistency: 197k, Intentionality: 14k, Adaptability: 0, Responsibility: 0) and 'Software Quality' (Security: 0, Reliability: 14k). The main panel displays three issues under the 'gameoflife-acceptance-tests/Dockerfile' file:

- Use a specific version tag for the image.** (Intentionality) - Maintainability: Open, Not assigned. L1 + 5min effort • 4 years ago • Code Smell • Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Intentionality) - Maintainability: Open, Not assigned. L12 + 5min effort • 4 years ago • Code Smell • Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Intentionality) - Maintainability: Open, Not assigned. L12 + 5min effort • 4 years ago • Code Smell • Major

● Bugs

SonarQube Issues page for project `atharvsonarqubetest1`. The main navigation bar includes Overview, Issues (selected), Security Hotspots, Measures, Code, and Activity. The top right shows Project Settings and Project Information with 46,515 issues and 1426d effort.

The left sidebar filters include:

- Maintainability: 0
- Severity: 0
- Type: Bug (47k), Vulnerability (0), Code Smell (164k). An "Add to selection" button is available.
- Scope: 0
- Status: 0
- Security Category: 0
- Creation Date: 0

The right panel lists three bugs under the `gameoflife-core/build/reports/tests/all-tests.html` file:

- Bug**: Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability) Status: Open, Not assigned. L1 - 5min effort - 4 years ago. Type: Bug, Major.
- Bug**: Add "lang" and/or "xml:lang" attributes to this "<html>" element. (Reliability) Status: Open, Not assigned. L1 - 2min effort - 4 years ago. Type: Bug, Major.
- Bug**: Add "<th>" headers to this "<table>". (Reliability) Status: Open, Not assigned. L9 - 2min effort - 4 years ago. Type: Bug, Major.

A note at the bottom states: "Embedded database should be used for evaluation purposes only."

● Code Smells

SonarQube Issues page for project `atharvsonarqubetest1`. The main navigation bar includes Overview, Issues (selected), Security Hotspots, Measures, Code, and Activity. The top right shows Project Settings and Project Information with 164,034 issues and 1708d effort.

The left sidebar filters include:

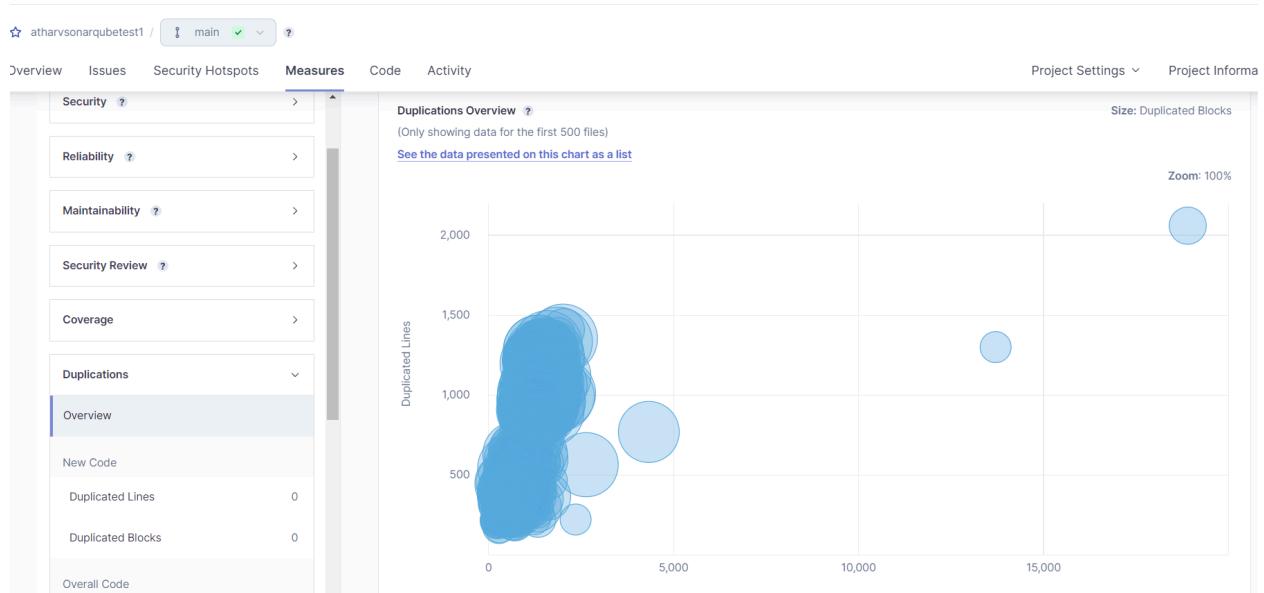
- Maintainability: 164k
- Severity: 0
- Type: Bug (0), Vulnerability (0), Code Smell (164k). An "Add to selection" button is available.
- Scope: 0
- Status: 0
- Security Category: 0
- Creation Date: 0

The right panel lists three code smells under the `gameoflife-acceptance-tests/Dockerfile` file:

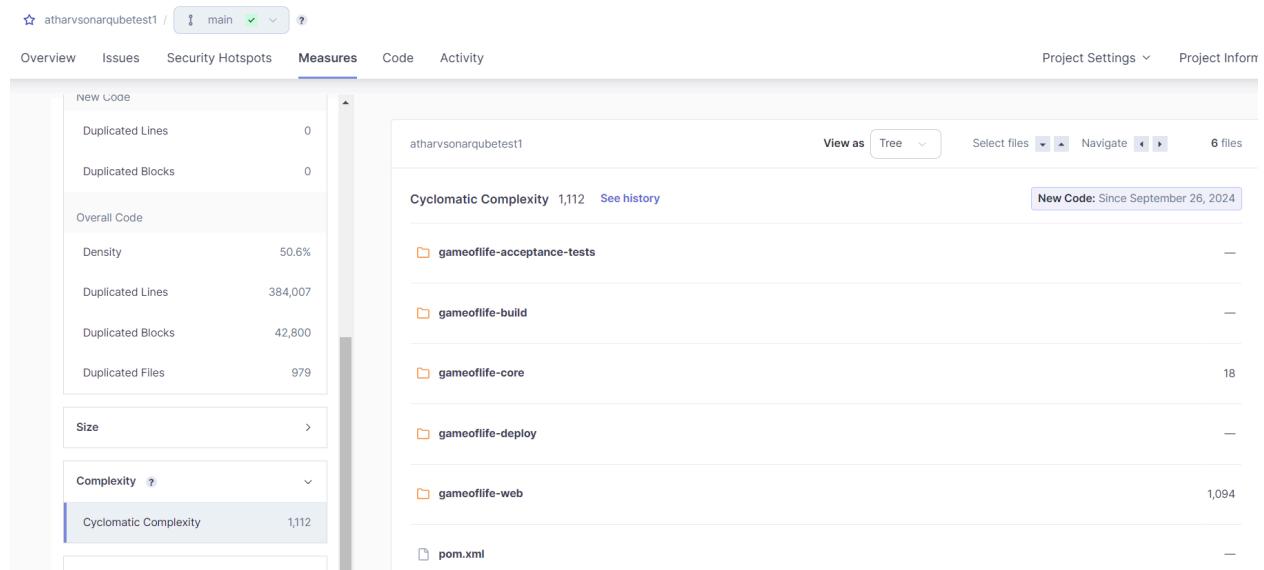
- Code Smell**: Use a specific version tag for the image. (Maintainability) Status: Open, Not assigned. L1 - 5min effort - 4 years ago. Type: Code Smell, Major.
- Code Smell**: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Status: Open, Not assigned. L12 - 5min effort - 4 years ago. Type: Code Smell, Major.
- Code Smell**: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Status: Open, Not assigned. L12 - 5min effort - 4 years ago. Type: Code Smell, Major.

A note at the bottom states: "Embedded database should be used for evaluation purposes only."

- Duplications



- Cyclomatic Complexities



Conclusion:

This experiment allowed us to integrate Jenkins and SonarQube to set up a CI/CD pipeline capable of performing static analysis on Java code. Through this process, we automated the detection of common code issues such as bugs, code smells, and duplications. By leveraging Docker for SonarQube and the Jenkins pipeline, we streamlined the code scanning process, ensuring any issues were highlighted during the build phase. This integration demonstrates the importance of automated code quality checks in a continuous delivery environment.

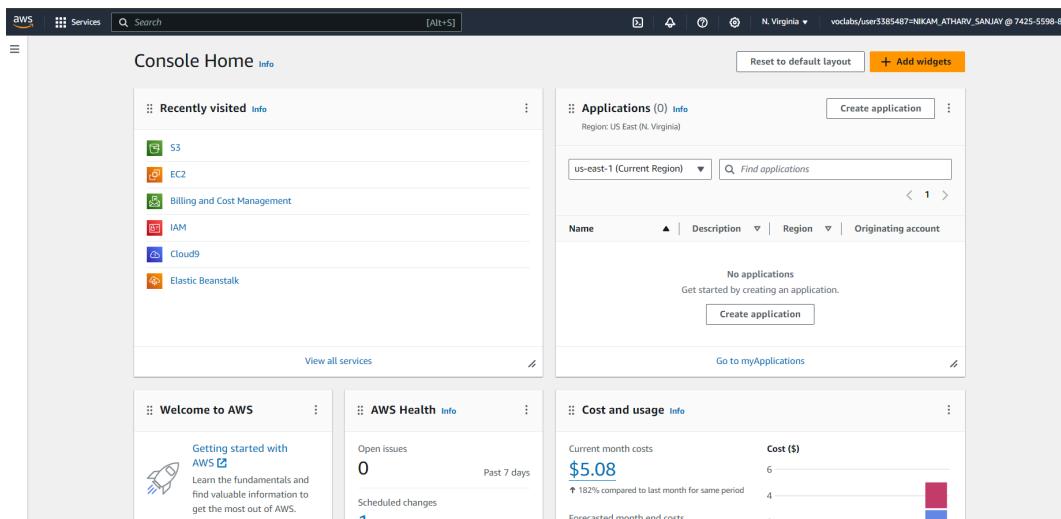
Experiment 9

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

STEPS

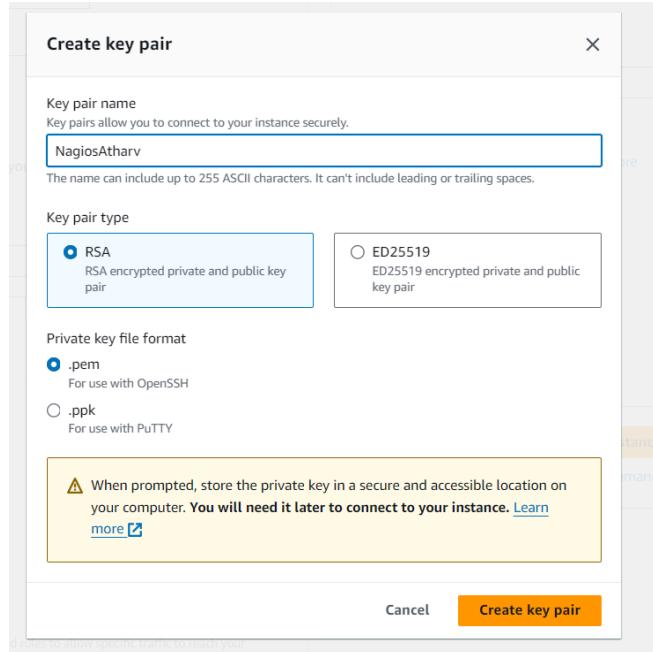
Step 1: Accessing AWS EC2

Log in to your AWS account, search for EC2 in the services section, and open the interface. Click on "Create Instance" and select **Amazon Linux** as the OS image for the instance.



Step 2: Setting Up Key Pair

If you don't already have a key pair or `.pem` file, generate a new key pair. If one exists, select it. Ensure that the `.pem` file is saved securely on your system for future use.



Step 3: Configuring Security Group

Navigate to the **Security Groups** section from the left panel. Locate the security group linked to your instance, click on the instance ID, and edit the **Inbound Rules**. Add the following rules: HTTP, HTTPS, Custom TCP (port 5666), All ICMP (IPv4 and IPv6), and All traffic. Save the rules.

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Security Groups (1) Info					
Actions Export security groups to CSV Create security group					
	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-05a60d807bfc11237	default	vpc-0f7970ea32a533bcc	default VPC security group

Name:Atharv Nikam

Div :D15C

Roll: 36

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details	
Security group name <small>Info</small>	Nagios
Name cannot be edited after creation.	
Description <small>Info</small>	Allow ssh acces to developers
VPC Info	vpc-0f7970ea32a533bcc

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <small>Info</small>					
Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
-	SSH	TCP	22	Anywhere... ▾	<input type="text" value="0.0.0.0"/> X
-	HTTP	TCP	80	Anywhere... ▾	<input type="text" value="0.0.0.0"/> X
-	HTTPS	TCP	443	Anywhere... ▾	<input type="text" value="0.0.0.0"/> X
-	All ICMP - IPv4	ICMP	All	Anywhere... ▾	<input type="text" value="0.0.0.0"/> X

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

⌚ Security group (sg-0b59e140edaa1f431 | Nagios) was created successfully

► Details

[EC2](#) > [Security Groups](#) > sg-0b59e140edaa1f431 - Nagios

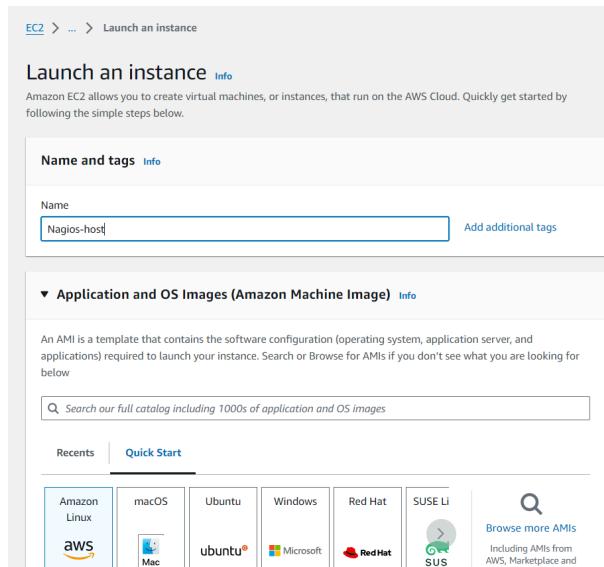
sg-0b59e140edaa1f431 - Nagios [Actions ▾](#)

Details			
Security group name Nagios	Security group ID sg-0b59e140edaa1f431	Description Allow ssh acces to developers	VPC ID vpc-0f7970ea32a533bcc
Owner 742555988891	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) [Outbound rules](#) [Tags](#)

Step 4: Connecting to EC2 Instance

Back in the EC2 dashboard, click on the instance ID and choose **Connect**. Under **SSH Client**, copy the provided command. Open your terminal where the **.pem** file is located, paste the SSH command, and execute it to connect.



EC2 > Instances > i-084ba8cc8adfcce2a > Connect to instance

Connect to instance Info

Connect to your instance i-084ba8cc8adfcce2a (Nagios-host) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID
 i-084ba8cc8adfcce2a (Nagios-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is NagiosAtharv.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "NagiosAtharv.pem"
4. Connect to your instance using its Public DNS:
 ec2-44-202-137-247.compute-1.amazonaws.com

Command copied

ssh -i "NagiosAtharv.pem" ec2-user@ec2-44-202-137-247.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

```
PS D:\Advancededevops key> ssh -i "NagiosAtharv.pem" ec2-user@ec2-44-201-253-108.compute-1.amazonaws.com
A newer release of "Amazon Linux" is available.
Version 2023.5.20241001:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,
#_
~\_ ####_ Amazon Linux 2023
~~ \_#####\
~~ \###|
~~ #/ ____ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' '->
~~ .-
/-
/`/`/
/m/`/
[ec2-user@ip-172-31-86-175 ~]$
```

Step 5: Updating YUM Packages

Run `sudo yum update` to update the system's YUM package manager. This ensures the latest updates and security patches are installed

```
=====
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-86-175 ~]$
```

Step 6: Installing Apache and PHP

Install the necessary web server components, Apache and PHP, by running:

`sudo yum install httpd php`

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install httpd php
Last metadata expiration check: 1 day, 13:28:04 ago on Fri Oct  4 03:32:13 2024.
Dependencies resolved.
=====
Architecture          Version           Repository      Size
=====
httpd                x86_64            2.4.62-1.amzn2023   amazonlinux    48 k
php8.3               x86_64            8.3.10-1.amzn2023.0.1  amazonlinux    10 k
=====
Installing:
=====
httpd                x86_64            2.4.62-1.amzn2023   amazonlinux    48 k
php8.3               x86_64            8.3.10-1.amzn2023.0.1  amazonlinux    10 k
=====
Installing dependencies:
apr                  x86_64            1.7.2-2.amzn2023.0.2   amazonlinux    129 k
apr-util              x86_64            1.6.3-1.amzn2023.0.1   amazonlinux    98 k
generic-logos-httpd  noarch            18.0.0-12.amzn2023.0.3  amazonlinux    19 k
httpd-core            x86_64            2.4.62-1.amzn2023   amazonlinux    1.4 M
httpd-filesystem      noarch            2.4.62-1.amzn2023   amazonlinux    14 k
httpd-tools            x86_64            2.4.62-1.amzn2023   amazonlinux    81 k
libbrotli             x86_64            1.0.9-4.amzn2023.0.2   amazonlinux    315 k
libsodium              x86_64            1.0.19-4.amzn2023   amazonlinux    176 k
libxmlsl              x86_64            1.1.34-5.amzn2023.0.2  amazonlinux    241 k
mailcap               noarch            2.1.49-3.amzn2023.0.3  amazonlinux    33 k
nginx-filesystem      noarch            1:1.24.0-1.amzn2023.0.4  amazonlinux    9.8 k
php8.3-cli             x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    3.7 M
php8.3-common          x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    737 k
php8.3-process         x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    45 k
php8.3-xml              x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    154 k
=====
Installing weak dependencies:
apr-util-openssl       x86_64            1.6.3-1.amzn2023.0.1   amazonlinux    17 k
mod_http2              x86_64            2.0.27-1.amzn2023.0.3   amazonlinux    166 k
mod_lua                x86_64            2.4.62-1.amzn2023   amazonlinux    61 k
php8.3-fpm              x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    1.9 M
php8.3-mbstring         x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    528 k
php8.3-opcache          x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    379 k
php8.3-pdo              x86_64            8.3.10-1.amzn2023.0.1   amazonlinux    89 k
```

```
Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
libbsl-1.1.34-5.amzn2023.0.2.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64
php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
php8.3-process-8.3.10-1.amzn2023.0.1.x86_64
php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

apr-util-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-tools-2.4.62-1.amzn2023.x86_64
libsodium-1.0.19-4.amzn2023.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_lua-2.4.62-1.amzn2023.x86_64
php8.3-8.3.10-1.amzn2023.0.1.x86_64
php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64
php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64

[Complete!]
```

Step 7: Installing C/C++ Compiler

Install the GCC compiler along with C libraries using:

```
sudo yum install gcc glibc glibc-common.
```

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 1 day, 13:29:48 ago on Fri Oct  4 03:32:13 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package          Architecture Version       Repository  Size
=====
Installing:
  gcc            x86_64      11.4.1-2.amzn2023.0.2   amazonlinux 32 M
Installing dependencies:
  annobin-docs    noarch     10.93-1.amzn2023.0.1   amazonlinux 92 k
  annobin-plugin-gcc x86_64    10.93-1.amzn2023.0.1   amazonlinux 887 k
  cpp            x86_64      11.4.1-2.amzn2023.0.2   amazonlinux 10 M
  gc              x86_64      8.0.4-5.amzn2023.0.2   amazonlinux 105 k
  glibc-devel     x86_64      2.34-52.amzn2023.0.11  amazonlinux 27 k
  glibc-headers-x86 noarch     2.34-52.amzn2023.0.11  amazonlinux 427 k
  guile22        x86_64      2.2.7-2.amzn2023.0.3   amazonlinux 6.4 M
  kernel-headers x86_64      6.1.109-118.189.amzn2023 amazonlinux 1.4 M
  libmpc          x86_64      1.2.1-2.amzn2023.0.2   amazonlinux 62 k
  libtool-ltdl    x86_64      2.4.7-1.amzn2023.0.3   amazonlinux 38 k
  libxcrypt-devel x86_64      4.4.33-7.amzn2023      amazonlinux 32 k
  make            x86_64      1:4.3-5.amzn2023.0.2   amazonlinux 534 k

Transaction Summary
=====
Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]:
```

```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch
  cpp-11.4.1-2.amzn2023.0.2.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
  kernel-headers-6.1.109-118.189.amzn2023.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64

  annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  gc-8.0.4-5.amzn2023.0.2.x86_64
  glibc-devel-2.34-52.amzn2023.0.11.x86_64
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libxcrypt-devel-4.4.33-7.amzn2023.x86_64
```

Step 8: Installing GD Library

For rendering images in Nagios, install the GD library with:

`sudo yum install gd gd-devel.`

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 1 day, 13:31:19 ago on Fri Oct  4 03:32:13 2024.
Dependencies resolved.
=====
| Package           | Architecture | Version   | Repository | Size |
|=====|
| Installing:      |             |           |            |        |
| gd               | x86_64      | 2.3.3-5.amzn2023.0.3 | amazonlinux | 139 k |
| gd-devel         | x86_64      | 2.3.3-5.amzn2023.0.3 | amazonlinux | 38 k  |
|=====|
| Installing dependencies: |
| brotli           | x86_64      | 1.0.9-4.amzn2023.0.2 | amazonlinux | 314 k |
| brotli-devel     | x86_64      | 1.0.9-4.amzn2023.0.2 | amazonlinux | 31 k  |
| bzip2-devel      | x86_64      | 1.0.8-6.amzn2023.0.2 | amazonlinux | 214 k |
| cairo             | x86_64      | 1.17.6-2.amzn2023.0.1 | amazonlinux | 684 k |
| cmake-filesystem | x86_64      | 3.22.2-1.amzn2023.0.4 | amazonlinux | 16 k  |
| fontconfig        | x86_64      | 2.13.94-2.amzn2023.0.2 | amazonlinux | 273 k |
| fontconfig-devel  | x86_64      | 2.13.94-2.amzn2023.0.2 | amazonlinux | 128 k |
| fonts-filesystem | noarch      | 1:2.0.5-12.amzn2023.0.2 | amazonlinux | 9.5 k |
| freetype          | x86_64      | 2.13.2-5.amzn2023.0.1 | amazonlinux | 423 k |
| freetype-devel    | x86_64      | 2.13.2-5.amzn2023.0.1 | amazonlinux | 912 k |
| glib2-devel       | x86_64      | 2.74.7-689.amzn2023.0.2 | amazonlinux | 486 k |
| google-noto-fonts-common | noarch      | 20201206-2.amzn2023.0.2 | amazonlinux | 15 k  |
| google-noto-sans-vf-fonts | noarch      | 20201206-2.amzn2023.0.2 | amazonlinux | 492 k |
| graphite2         | x86_64      | 1.3.14-7.amzn2023.0.2 | amazonlinux | 97 k  |
| graphite2-devel   | x86_64      | 1.3.14-7.amzn2023.0.2 | amazonlinux | 21 k  |
| harfbuzz          | x86_64      | 7.0.0-2.amzn2023.0.1 | amazonlinux | 868 k |
|=====|
```

```
Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-2.3.3-5.amzn2023.0.3.x86_64
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libICE-1.0.10-6.amzn2023.0.2.x86_64
libX11-1.7.2-3.amzn2023.0.4.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-86-175 ~]$
```

Step 9: Creating Nagios User

Create a user called **nagios**, provide it with a home directory, and set a password:

```
sudo adduser -m nagios
sudo passwd nagios.
```

```
Complete!
[ec2-user@ip-172-31-86-175 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-86-175 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-86-175 ~]$ _
```

Step 10: Creating Nagcmd Group

Create a group called **nagcmd** to manage Nagios command executions:

```
sudo groupadd nagcmd
```

Step 11: Adding Users to Nagcmd Group

Add both **nagios** and **apache** users to the **nagcmd** group:

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

```
[ec2-user@ip-172-31-86-175 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-86-175 ~]$ sudo usermod -a -G nagcmd apache
```

Step 12: Creating Download Directory

Create a directory to store Nagios files:

```
mkdir ~/downloads
```

Navigate to the directory using:

```
cd ~/downloads.
```

```
[ec2-user@ip-172-31-86-175 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-86-175 ~]$ cd ~/downloads
[ec2-user@ip-172-31-86-175 downloads]$ _
```

Step 13: Downloading Nagios Core and Plugins

Download the latest versions of Nagios Core and Plugins. If no newer versions are available, use:

```
wget
```

```
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
```

```
wget
```

```
https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-05 17:11:46-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00:f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz      100%[=====]  1.97M  5.10MB/s   in 0.4s
2024-10-05 17:11:46 (5.10 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-86-175 downloads]$ _
```

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-05 17:12:19-- https://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|:443... 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  6.37MB/s   in 0.4s
2024-10-05 17:12:19 (6.37 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-86-175 downloads]$ _
```

Step 14: Extracting Nagios Core

Extract the Nagios Core files in the same directory using:

```
tar zxvf nagios-4.5.5.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 downloads]$ tar zxvf nagios-4.5.5.tar.gz
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
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
```

```
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-86-175 downloads]$
```

Step 15: Configuring Nagios Core

```
[ec2-user@ip-172-31-86-175 downloads]$ ls  
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz  
[ec2-user@ip-172-31-86-175 downloads]$
```

If there's an error regarding SSL headers, install OpenSSL using:

`sudo yum install openssl-devel`.

```
[ec2-user@ip-172-31-86-175 downloads]$ sudo yum install openssl-devel  
Last metadata expiration check: 1 day, 13:46:43 ago on Fri Oct 4 03:32:13 2024.  
Dependencies resolved.  
=====  
 Package          Architecture      Version       Repos  
=====  
Installing:  
 openssl-devel    x86_64          1:3.0.8-1.amzn2023.0.14  amazo  
  
Transaction Summary  
=====  
Install 1 Package  
  
Total download size: 3.0 M  
Installed size: 4.7 M
```

To ensure Nagios uses the `nagcmd` group for external commands, run:

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

```
[ec2-user@ip-172-31-86-175 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
```

Step 16: Compiling and Installing Nagios

Compile Nagios by running:

```
make all
```

Install with:

```
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
```

```
sudo make install
```

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install
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'
```

Name:Atharv Nikam

Div :D15C

Roll: 36

sudo make install-init

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
```

sudo make install-commandmode.

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-commandmode
/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 ***

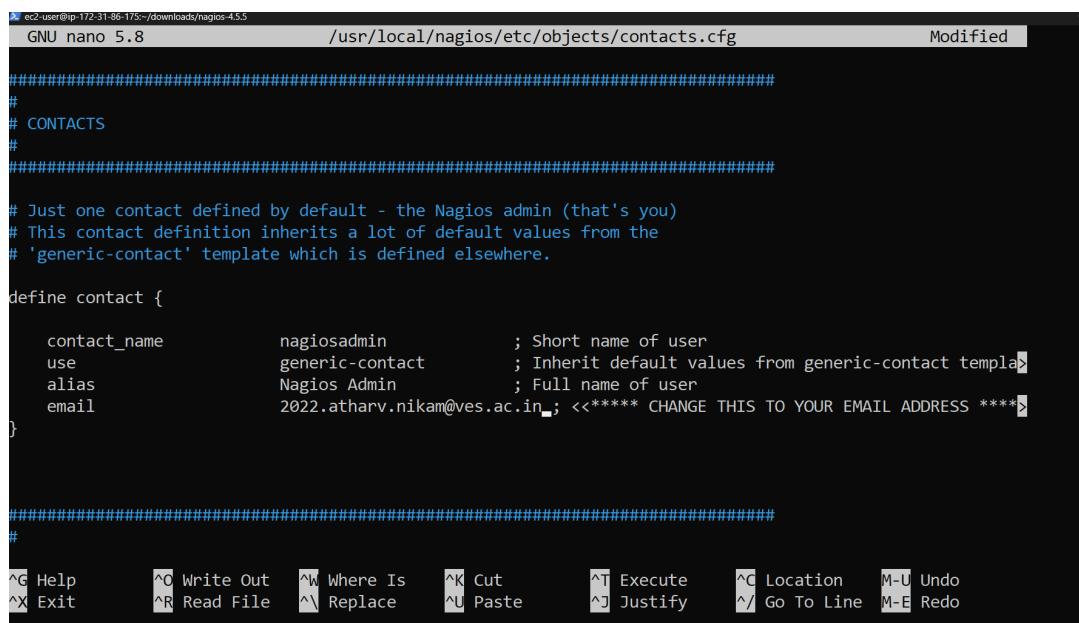
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 17: Configuring Email for Notifications

To configure notifications, open the `contacts.cfg` file and set your email address under `define contact{}` using:

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

Save your changes with **CTRL+O** and exit with **CTRL+X**.



```
ec2-user@ip-172-31-86-175:~/downloads/nagios-4.5.5$ nano /usr/local/nagios/etc/objects/contacts.cfg
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg                         Modified

#####
#
# CONTACTS
#
#####

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template
    alias             Nagios Admin         ; Full name of user
    email             2022.atharv.nikam@ves.ac.in; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>

}

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

Step 18: Installing Nagios Web Interface

Install the web interface configuration by running:

```
sudo make install-webconf.
```

```
[ec2-user@ip-172-31-86-175 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 [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

Step 19: Creating Web Interface User

Create a user called **nagiosadmin** to access the Nagios web interface:

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

```
[ec2-user@ip-172-31-86-175 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
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 20: Restarting Apache Server

Restart the Apache server to apply the new configurations:

```
sudo service httpd restart.
```

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 21: Extracting Nagios Plugins

Extract the Nagios plugins in the `downloads` directory using:

```
tar zxvf nagios-plugins-2.4.11.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 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/litmain.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/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-86-175 downloads]$ ■
```

```
[ec2-user@ip-172-31-86-175 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/ltmain.sh
```

Step 22: Configuring Nagios Plugins

Navigate to the extracted folder and configure Nagios plugins with:

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

```
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
```

```
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

```
gins/utils.o -L. ../lib/libnagiosplug.a ../gl/libgnu.a -lresolv -lssl -lcrypto -lpthread
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/plugins-root'
Making all in po
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 23: Compiling and Installing Plugins

Compile and install the plugins using:

```
make
```

```
sudo make install.
```

```
fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 24: Registering Nagios as a Service

To make Nagios manageable as a system service, run:

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

```
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

```
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...
```

```
Checking misc settings...
```

```
Total Errors: 0
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

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

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Check its status with:

```
sudo systemctl status nagios.
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-05 19:18:16 UTC; 14s ago
     Docs: https://www.nagios.org/documentation
 Process: 68151 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0)
 Process: 68152 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0)
 Main PID: 68153 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 5.5M
      CPU: 79ms
 CGroup: /system.slice/nagios.service
         ├─68153 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
         ├─68154 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
         ├─68155 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
         ├─68156 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
         ├─68157 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh

Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
```

```
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios'>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6>
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: Successfully launched command file worker w>
lines 3-28/28 (END)
```

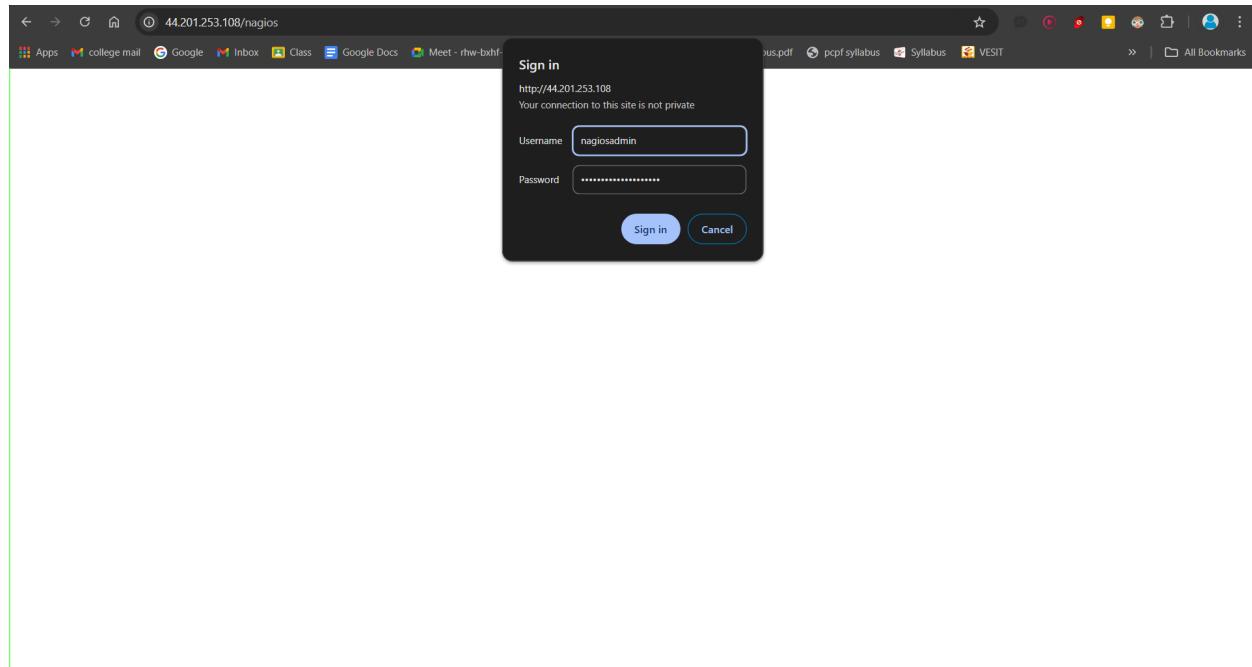
Step 27: Accessing Nagios Dashboard

Copy the public IP address of your EC2 instance and type it in your browser, appending `/nagios` to access the dashboard:

`http://<public-ip-address>/nagios`.

Username:nagiosadmin

password:(the password you have set)



The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A message indicates "Daemon running with PID 68153". On the left, a sidebar menu includes sections for General, Current Status (with links to Tactical Overview, Map, Hosts, Services, Host Groups, Grid, Service Groups, and Problems), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). The main content area features several panels: "Get Started" with a list of links; "Latest News" and "Don't Miss..." which are currently empty; and a "Quick Links" panel containing links to Nagios Library, Labs, Exchange, Support, and the official website. A footer note states: "Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors." Another note below it says: "Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions." A "Page Tour" button is located on the right side.

Conclusion:

In this experiment, I successfully explored the process of installing and configuring Nagios Core, Plugins, and NRPE on a Linux-based system. By carefully configuring the necessary security settings, installing dependencies, and resolving common issues like permission errors and missing directories, I ensured a smooth setup. This setup allows for efficient monitoring using Nagios, and I accessed the monitoring interface through the EC2 instance's public IP, demonstrating the effectiveness of continuous monitoring in a cloud environment.

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

Step 1: Confirming Nagios on the Server

First, ensure that Nagios is running on your server by executing the following command on your Amazon Linux machine (Nagios-host):

```
sudo systemctl status nagios.
```

```
[ec2-user@ip-172-31-86-175 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-10-06 06:42:44 UTC; 11s ago
       Docs: https://www.nagios.org/documentation
    Process: 2847 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 2848 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 2849 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.0M
    CPU: 17ms
   CGroup: /system.slice/nagios.service
           ├─2849 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─2850 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2851 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2852 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2853 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─2854 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: core query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: echo service query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: help for the query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Successfully registered manager as @wproc with query handler
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2852;pid=2852
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2853;pid=2853
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2851;pid=2851
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2850;pid=2850
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: Successfully launched command file worker with pid 2854
[ec2-user@ip-172-31-86-175 ~]$
```

Step 2: Creating EC2 Instance

Next, create a new EC2 instance named **Nagios-client** with the Ubuntu AMI and **t2.micro** instance type. Generate an RSA key pair (.pem file), or use an existing one if available. Make sure to select the security group used in your previous Nagios-host setup.

Name:Atharv Nikam

Div:D15C

Roll:36

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar labeled "Search our full catalog including 1000s of application and OS images". Below it, there are two tabs: "Recents" and "Quick Start", with "Quick Start" being the active tab. Under "Quick Start", there are several pre-built Lambda functions represented by cards: Amazon Linux (aws logo), macOS (Mac logo), Ubuntu (Ubuntu logo), Windows (Microsoft logo), Red Hat (Red Hat logo), and SUSE Linux (SUSE logo). To the right of these cards is a search icon and a link "Browse more AMIs". A note below the cards states: "An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below".

Step 3: Connecting to the Instance

After creating the EC2 instance, connect to it. Navigate to the folder where the key (.pem) is stored on your local machine. Copy the provided SSH command from the instance's **SSH Client** section and paste it into your terminal.

The screenshot shows the AWS EC2 instance configuration page. The first section is titled "Key pair (login)" with a note: "You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance." It includes a dropdown menu for "Key pair name - required" containing "Nagiosatharvexp10" and a "Create new key pair" button. The second section is titled "Network settings" with an "Edit" button. It displays network information: "Network" (vpc-0f7970ea32a533bcc), "Subnet" (No preference (Default subnet in any availability zone)), and "Auto-assign public IP" (Enable). There's a note about "Additional charges apply when outside of free tier allowance". The "Firewall (security groups)" section includes a note: "A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance." It has two options: "Create security group" (radio button unselected) and "Select existing security group" (radio button selected). The "Common security groups" section shows a dropdown menu with "Nagios sg-0b59e140edaa1f431" selected, along with a note: "VPC: vpc-0f7970ea32a533bcc". There's also a "Compare security group rules" button.

Name:Atharv Nikam

Div:D15C

Roll:36

```
PS D:\Advanceddevops key\exp10> ssh -i "Nagiosatharvexp10.pem" ubuntu@ec2-3-80-172-58.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

System information as of Sun Oct  6 06:55:47 UTC 2024

System load:  0.08           Processes:          106
Usage of /:   22.9% of 6.71GB  Users logged in:    0
Memory usage: 20%            IPv4 address for enX0: 172.31.46.49
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-49:~$ |
```

Step 4: Checking Nagios Status

To confirm Nagios is running correctly on the Nagios-host, execute the following:

```
ps -ef | grep nagios.
```

```
[ec2-user@ip-172-31-86-175 ~]$ ps -ef | grep nagios
nagios  2849      1  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  2850  2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios  2851  2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios  2852  2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios  2853  2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios  2854  2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  3397  2384  0 06:56 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-86-175 ~]$ |
```

Step 5: Creating Root Directories

Switch to the root user and create necessary directories for monitoring hosts:

```
sudo su
```

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

```
[ec2-user@ip-172-31-86-175 ~]$ sudo su  
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts  
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
[root@ip-172-31-86-175 ec2-user]# |
```

Step 6: Configuring Monitoring for Linux Server

Copy the sample Nagios configuration file for localhost and create a new configuration file for the Linux server:

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

```
[root@ip-172-31-86-175 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-86-175 ec2-user]# |
```

Step 7: Editing Linux Server Configuration

Open the `linuxserver.cfg` file and modify the hostname, IP address, and hostgroup as follows:

- **hostname:** linuxserver
- **address:** Public IP of the Linux client
- **hostgroup_name:** linux-servers1

Use the command:

```
nano
```

```
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/l  
inuxserver.cfg.
```

```

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     172.31.46.49
}

#####
# 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
}

```

Step 8: Updating Nagios Configuration

Add the following line to Nagios' main configuration file to include the monitoring hosts directory:

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

Edit the file using:

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

```

#
# Read the documentation for more information on this configuration
#
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!
log_file=/usr/local/nagios/var/nagios.log

# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg

```

Step 9: Verifying Nagios Configuration

To check for any syntax errors in your configuration, run:

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

```
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
```

Step 10: Restarting Nagios

Restart the Nagios service to apply the configuration changes:

```
sudo service nagios restart
```

```
[root@ip-172-31-86-175 ec2-user]# service nagios restart  
Redirecting to /bin/systemctl restart nagios.service  
[root@ip-172-31-86-175 ec2-user]# |
```

Step 11: Installing NRPE on Nagios Client

Connect to the Nagios-client instance and update the system. Then install NRPE and necessary Nagios plugins using:

```
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
```

```
ubuntu@ip-172-31-46-49:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:24 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
```

```
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 libbavahi-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-172-31-46-49:~$
```

Step 12: Configuring NRPE

Edit the NRPE configuration file to allow the Nagios-host to communicate with the client. Add the Nagios-host's IP address under `allowed_hosts`:

```
sudo nano /etc/nagios/nrpe.cfg.
```

```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_group=nagios

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,34.238.152.163

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

Step 13: Restarting NRPE

After editing the NRPE configuration, restart the NRPE server:

```
sudo systemctl restart nagios-nrpe-server.
```

```
ubuntu@ip-172-31-46-49:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-46-49:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-46-49:~$ |
```

Step 14: Checking Nagios and HTTPD Services

On the Nagios-host, check the status of Nagios and ensure that the HTTPD service is active:

```
sudo systemctl status nagios
sudo systemctl status httpd.
```

```
[root@ip-172-31-86-175 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
    Active: inactive (dead)
      Docs: man:httpd.service(8)
[root@ip-172-31-86-175 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-86-175 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-86-175 ec2-user]# |
```

If HTTPD is not active, start and enable it:

```
sudo systemctl start httpd
sudo systemctl enable httpd.
```

Step 15: Accessing Nagios Dashboard

To view the Nagios dashboard, open your browser and go to:

<http://<Nagios-host-ip>/nagios>.

Click on **Hosts** from the left panel to view the status of your Linux server.

Nagios® Core™

✓ Daemon running with PID 3969

Nagios® Core™
Version 4.5.5
September 17, 2024
[Check for updates](#)

Get Started <ul style="list-style-type: none"> Start monitoring your infrastructure Change the look and feel of Nagios Extend Nagios with hundreds of addons Get support Get training Get certified 	Quick Links <ul style="list-style-type: none"> Nagios Library (tutorials and docs) Nagios Labs (development blog) Nagios Exchange (plugins and addons) Nagios Support (tech support) Nagios.com (company) Nagios.org (project)
Latest News	Don't Miss...

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

MONITORED BY
Nagios

Not secure 34.238.152.163/nagios/

Apps college mail Google Inbox Class Google Docs Meet - rhw-bxvf-fy timetable Console Home | Co... syllabus.pdf pcpf syllabus Syllabus VESIT All Bookmarks

Nagios®

Current Network Status
Last Updated: Sun Oct 6 07:33:16 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

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:

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
6 1 0 1 0
All Problems All Types
2 8

Host Status Details For All Host Groups
Limit Results: 100 Host Status Last Check Duration Status Information
linuxserver UP 10-06-2024 07:30:19 0d 0h 22m 57s PING OK - Packet loss = 0%, RTA = 1.68 ms
localhost UP 10-06-2024 07:30:41 0d 12h 15m 0s PING OK - Packet loss = 0%, RTA = 0.24 ms

Results 1 - 2 of 2 Matching Hosts

Reports Availability Trends Alerts History Summary Histogram Notifications Event Log

System Comments Downtime Process Info Performance Info Scheduling Queue Configuration

Page Tour

Name:Atharv Nikam

Div:D15C

Roll:36

Host Information

Last Updated: Sun Oct 6 07:34:10 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View Status Detail For This Host](#)
[View Alert History For This Host](#)
[View Trends For This Host](#)
[View Alert Histogram For This Host](#)
[View Availability Report For This Host](#)
[View Notifications For This Host](#)

Host
localhost
(linuxserver)

Member of
No hostgroups

172.31.46.49

Host State Information

Host Status:	UP (for 0d 0h 23m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.68 ms
Performance Data:	rtt=1.679000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-06-2024 07:30:19
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.011 seconds
Next Scheduled Active Check:	10-06-2024 07:35:19
Last State Change:	10-06-2024 07:10:19
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-06-2024 07:34:09 (0d 0h 0m 1s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

Host Comments

[Add a new comment](#) [Delete all comments](#)

Entry	Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it								

Not secure 34.238.152.163/nagios/

Apps college mail Google Inbox Class Google Docs Meet - rhw-bxhf-fry timetable Console Home | Co... syllabus.pdf pcpf syllabus Syllabus VESIT All Bookmarks

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:

Current Network Status

Last Updated: Sun Oct 6 07:35:35 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View History For all hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems	All Types		
0	2		

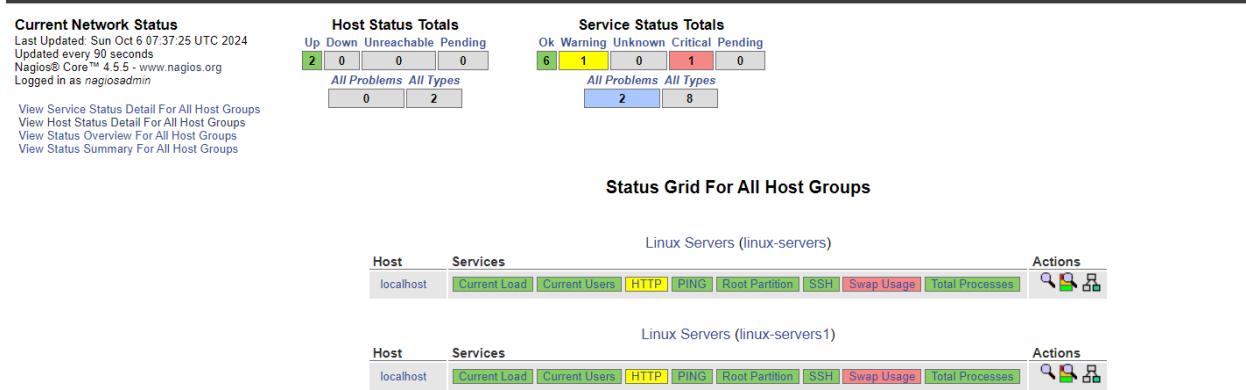
Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0
All Problems	All Types			
2	8			

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-06-2024 07:34:26	0d 12h 16m 42s	1/4	OK - load average: 0.00, 0.00, 0.00
localhost	Current Users	OK	10-06-2024 07:35:04	0d 12h 16m 4s	1/4	USERS OK - 2 users currently logged in
HTTP		WARNING	10-06-2024 07:30:41	0d 0h 14m 54s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
localhost	PING	OK	10-06-2024 07:31:19	0d 12h 14m 49s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
localhost	Root Partition	OK	10-06-2024 07:31:56	0d 12h 14m 12s	1/4	DISK OK - free space: / 6106 MIB (75.24% inode=98%).
localhost	SSH	OK	10-06-2024 07:32:34	0d 12h 13m 34s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	Critical	10-06-2024 07:33:11	0d 12h 9m 57s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	10-06-2024 07:33:49	0d 12h 12m 19s	1/4	PROCS OK: 37 processes with STATE = RSZDT

Display 1 of 8 Matching Services



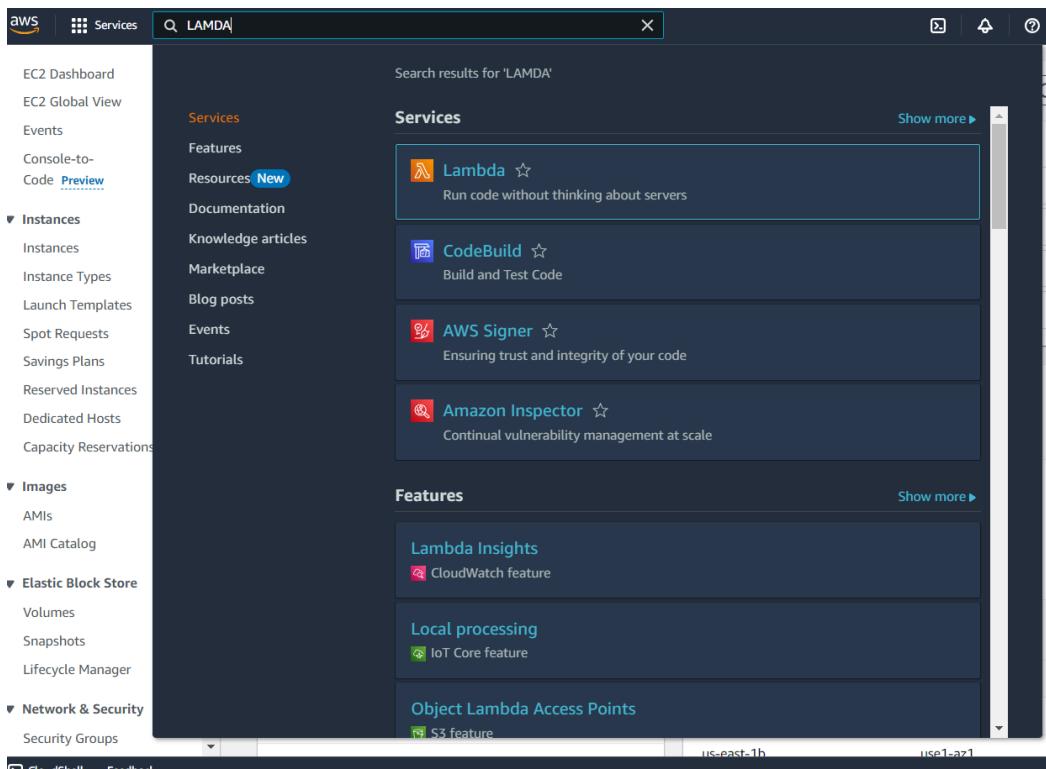
Conclusion:

This experiment was designed to set up monitoring for ports, services, and a Linux server using Nagios. By carefully configuring both the Nagios host and client, we were able to monitor essential network services and assess server performance. The experiment demonstrated how Nagios can be used effectively to track system metrics such as CPU usage and memory consumption. This hands-on experience highlights the importance of proactive monitoring in maintaining server health and ensuring the availability of critical services across Linux and Windows platforms.

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

Step 1: Accessing AWS

Log in to your AWS Personal/Academy account. Navigate to the Lambda service by searching for "Lambda" in the AWS Management Console.



Step 2: Creating a New Lambda Function

Click on the "Create function" button. Provide a name for your Lambda function and select the language you wish to use, such as Python 3.12. For architecture, choose x86, and for execution role, opt to create a new role with basic Lambda g permissions.

The screenshot shows the AWS Lambda landing page. At the top, there's a dark header with the word "Compute". Below it, a large heading says "AWS Lambda lets you run code without thinking about servers." A subtext explains: "You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration." To the right, there's a "Get started" box with a "Create a function" button.

How it works

Run Next: Lambda responds to events

.NET Java Node.js Python Ruby Custom runtime

```
1 * exports.handler = async (event) => {  
2     console.log(event);  
3     return 'Hello from Lambda!';  
4 };  
5
```

Step 3: Configuring Basic Settings

To modify the basic settings, navigate to the "Configuration" tab and click on "Edit" under General Settings. Here, you can add a description and adjust the memory and timeout settings. For this experiment, I set the timeout to 1 second, which is sufficient for testing.

The screenshot shows the "Basic information" section of the Lambda configuration. It includes fields for "Function name" (set to "lamda_demo"), "Runtime" (set to "Python 3.12"), "Architecture" (set to "x86_64"), and "Permissions" (with a note about default execution role creation).

Basic information

Function name

Enter a name that describes the purpose of your function.
lamda_demo

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12

Architecture Info

Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named ATHARV_LAMDA-role-0u7c9ooi, with permission to upload logs to Amazon CloudWatch Logs.

▶ Additional Configurations

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

[Cancel](#) [Create function](#)

aws Services Search [Alt+S]

Successfully created the function lamda_demo. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamda_demo

lamda_demo

▼ Function overview [Info](#)

Throttle [Copy ARN](#) Actions ▾

Diagram Template

lambda_demo

+ Add trigger + Add destination

Description -

Last modified 26 seconds ago

Function ARN [arn:aws:lambda:eu-north-1:010928207735:function:lambda_demo](#)

Function URL [Info](#) -

Export to Application Composer Download ▾

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#) [Start tutorial](#)

Code Test Monitor Configuration Aliases Versions

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully created the function **lambda_demo**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

Step 4: Testing the Function

Click on the "Test" tab and select "Create a new event." Name your event, set the event sharing to private, and choose the "hello-world" template.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event Edit saved event

Event name

MyEventName

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

```
1 [{"key1": "value1", "key2": "value2", "key3": "value3"}]
```

Format JSON

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_demo /

lambda_function.py

lambda_function

Configure test event Ctrl-Shift-C

Private saved events

● demo1

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
```

Code **Test** **Monitor** **Configuration** **Aliases** **Versions**

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_demo /

lambda_function.py

lambda_function

Execution result:

Execution results

Test Event Name: demo1

Response:

```
{ "statusCode": 200,
  "body": "\\"Hello from Lambda\\\""
}
```

Function Logs

```
START RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Version: $LATEST
END RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa
REPORT RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Duration: 1.35 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID

```
86829fa5-e154-46b3-8ff5-6f12ba6c3efa
```

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is currently selected), 'Deploy', and a status message 'Changes not deployed'. On the left, there's a sidebar with 'Environment' and a search bar 'Go to Anything (Ctrl-P)'. The main area displays the code for 'lambda_function.py':

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     my_string="Hello this is Exp 11"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(my_string);
9     }
10
```

Step 5: Running the Test

In the Code section, select the newly created event from the dropdown menu and click on "Test." You should see the output displayed below.

The screenshot shows the AWS Lambda function editor interface with the 'Execution result' tab selected. The top navigation bar is identical to the previous screenshot. The main area displays the execution results for the 'demo1' test event:

Status: Succeeded | Max memory used: 32 MB | Time: 1.62 ms

Test Event Name
demo1

Response

```
{
    "statusCode": 200,
    "body": "\\"Hello this is Exp 11\\\""
}
```

Function Logs

```
START RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Version: $LATEST
END RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec
REPORT RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Duration: 1.62 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Du
Request ID
298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec
```

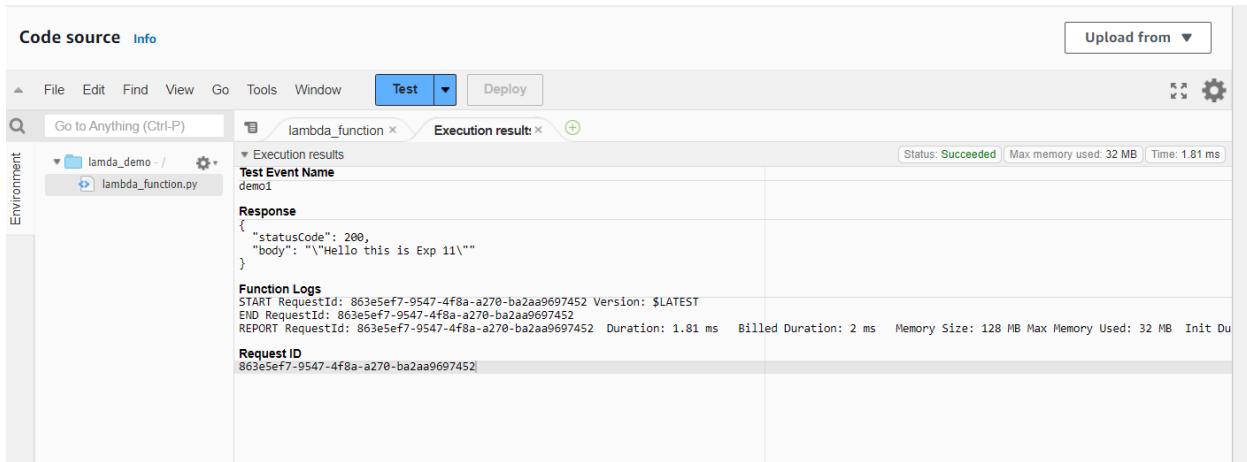
Step 6: Editing and Deploying the Code

You can modify your Lambda function's code as needed. I updated the code to display a new string. After making changes, press `Ctrl + S` to save and then click on "Deploy" to apply the updates.



Step 7: Final Testing

Return to the "Test" tab and execute the test again to observe the output. You should see a status code of 200 along with your string output and function logs confirming a successful deployment.



The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, a 'Test' button (which is highlighted in blue), and a 'Deploy' button. To the right of the 'Test' button is an 'Upload from' dropdown. On the left, there's a sidebar labeled 'Environment' with a dropdown menu. The main area has a search bar with 'Go to Anything (Ctrl-P)' and a file tree showing 'lambda_function' and 'lambda_demo / lambda_function.py'. Below the file tree is a section titled 'Execution results' with a 'Test Event Name' dropdown set to 'demo1'. To the right of this is a status bar showing 'Status: Succeeded', 'Max memory used: 32 MB', and 'Time: 1.81 ms'. Under 'Execution results', there are sections for 'Response' and 'Function Logs'. The 'Response' section contains a JSON object: { "statusCode": 200, "body": "Hello this is Exp 11\\\" }'. The 'Function Logs' section displays log entries: START RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Version: \$LATEST, END RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 REPORT RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Duration: 1.81 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 0 ms. A 'Request ID' section also lists the request ID: 863e5ef7-9547-4f8a-a270-ba2aa9697452.

Conclusion

In this experiment, I successfully navigated the process of creating an AWS Lambda function. After configuring the function with Python, I adjusted the settings to optimize its performance. I created a test event, deployed the function, and verified the output, which confirmed the expected behavior. This hands-on experience highlighted the user-friendly nature of AWS Lambda, illustrating how it enables developers to focus on coding while AWS efficiently handles the underlying infrastructure and scaling. This project not only deepened my understanding of serverless computing but also reinforced the practical application of cloud services in modern software development.

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.

Step 1: Create a s3 bucket.

- 1) Search for S3 bucket in the services search. Then click on create bucket.

The screenshot shows the AWS S3 service page. On the left, there's a sidebar with links like 'Buckets', 'Access Grants', 'Access Points', etc. The main content area features the 'Amazon S3' logo and the tagline 'Store and retrieve any amount of data from anywhere'. Below this, a section titled 'How it works' contains a video thumbnail for 'Introduction to Amazon S3'. To the right, there are sections for 'Create a bucket', 'Pricing', and 'Resources'.

- 2) Keep the bucket as a general purpose bucket. Give a name to your bucket.

The screenshot shows the 'Create bucket' configuration page. It includes sections for 'General configuration' (AWS Region set to US East (N. Virginia) us-east-1, Bucket type set to General purpose), 'Object Ownership' (ACLs disabled recommended), and other optional settings like 'Copy settings from existing bucket'.

3) Uncheck block all public access

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

4) Keeping all other options same, click on create. This would create your bucket. Now click on the name of the bucket

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-01092807735	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 14, 2024, 22:12:26 (UTC+05:30)
s3lamdaexp11	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 7, 2024, 09:40:50 (UTC+05:30)

5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload

Amazon S3 > Buckets > s3lamdaexp11

s3lamdaexp11 [Info](#)

Objects (0) [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name Type Last modified Size Storage class

No objects
You don't have any objects in this bucket.

[Upload](#)

Amazon S3 > Buckets > s3lamdaexp11 > Upload

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder.

Files and folders (1 Total, 990.9 KB)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input type="text"/> Find by name		< 1 >		
<input type="checkbox"/> Name	Folder			
<input type="checkbox"/> football.jpg	-			

Destination [Info](#)

Destination
<s3://s3lamdaexp11>

▶ **Destination details**
Bucket settings that impact new objects stored in the specified destination.

▶ **Permissions**
Grant public access and access to other AWS accounts.

▶ **Properties**
Specify storage class, encryption settings, tags, and more.

6) The image has been uploaded to the bucket

The screenshot shows the AWS S3 'Upload: status' page. At the top, a green header bar indicates 'upload succeeded' with a link to 'View details below.' Below this, a summary table shows the destination 's3://s3lambdaexp11' and the upload status: 'Succeeded' with '1 file, 990.9 KB (100.00%)'. A 'Failed' section shows '0 files, 0 B (0%)'. Below the summary is a table titled 'Files and folders (1 Total, 990.9 KB)' containing one item: 'football.jpg' (image/jpeg, 990.9 KB, Succeeded). There is also a 'Find by name' search bar and navigation controls.

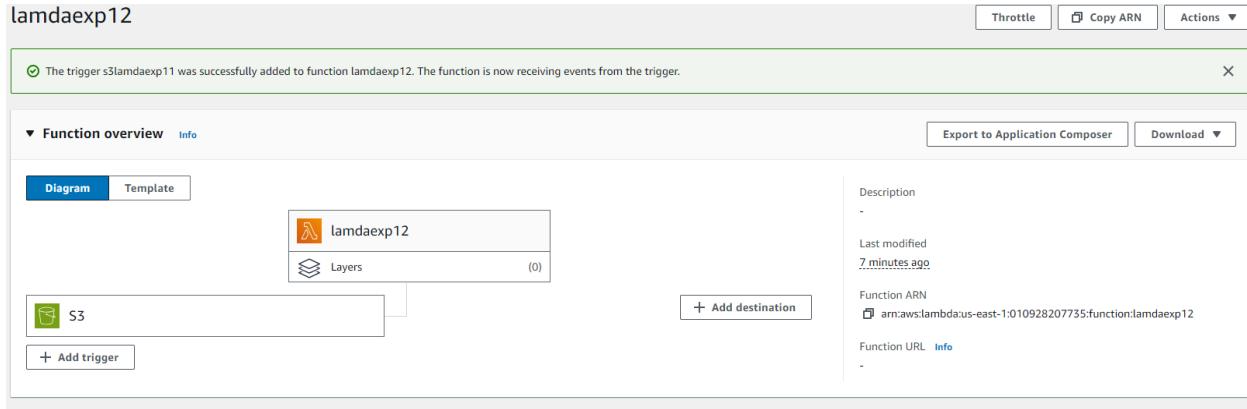
Step 2: Configure Lambda function

- 1) Go to the lambda function you had created before. (Services → Lambda → Click on name of function). Here, click on add trigger

The screenshot shows the 'Create function' wizard. It starts with a choice between 'Author from scratch', 'Use a blueprint', and 'Container image'. The 'Author from scratch' option is selected. The next step is 'Basic information', where the function name is set to 'lamdaexp12'. The 'Runtime' is chosen as 'Node.js 20.x', and the 'Architecture' is 'x86_64'. In the 'Permissions' section, there is a link to 'Change default execution role'. The wizard is currently at the 'Basic information' step.

2) Under trigger configuration, search for S3 and select it.

3) Here, select teh S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function



4) Scroll down to the code section of the function. Add the following javascript code to the code area by replacing the existing code

```
export const handler = async (event) => {
if (!event.Records || event.Records.length === 0) {
console.error("No records found in the event.");
return {
statusCode: 400,
body: JSON.stringify('No records found in the event')
};
}

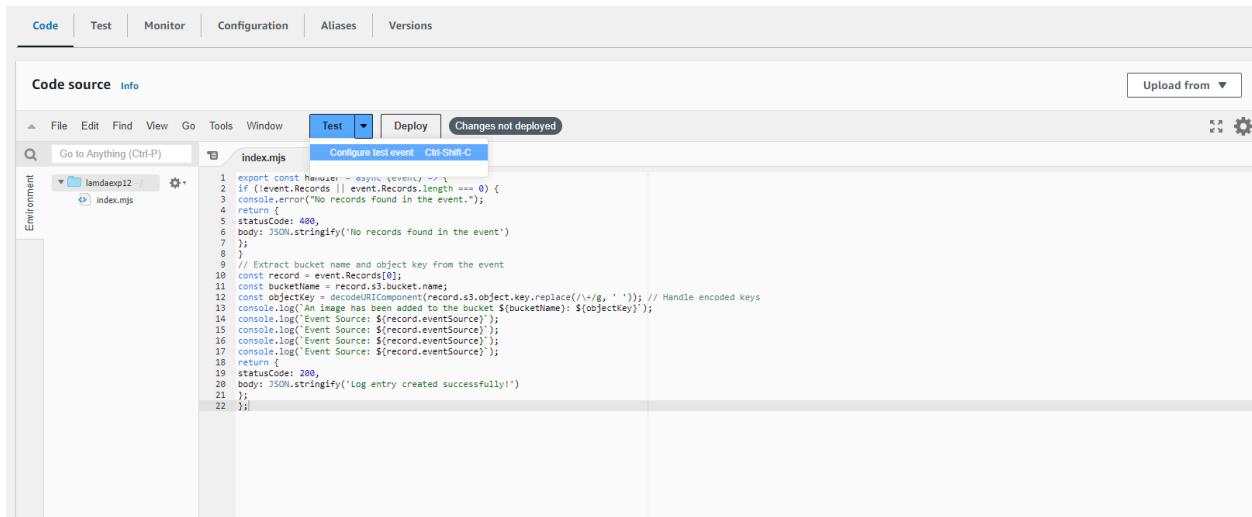
// Extract bucket name and object key from the event
const record = event.Records[0];
const bucketName = record.s3.bucket.name;
const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
return {
statusCode: 200,
body: JSON.stringify('Log entry created successfully!')
};
};
```

This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.



The screenshot shows the AWS Lambda function editor with the 'Code source' tab selected. The code is written in JavaScript and handles an S3 event. It logs the event records, extracts the bucket name and object key, and then logs the event source. The code is as follows:

```
1 export const handler = async (event) => {
2   if (!event.Records || event.Records.length === 0) {
3     console.error("No records found in the event.");
4     return {
5       statusCode: 400,
6       body: JSON.stringify('No records found in the event')
7     };
8   }
9   // Extract bucket name and object key from the event
10  const record = event.Records[0];
11  const bucketName = record.s3.bucket.name;
12  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\//g, ' ')); // Handle encoded keys
13  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
14  console.log(`Event Source: ${record.eventSource}`);
15  console.log(`Event Source: ${record.eventSource}`);
16  console.log(`Event Source: ${record.eventSource}`);
17  console.log(`Event Source: ${record.eventSource}`);
18  return {
19    statusCode: 200,
20    body: JSON.stringify('Log entry created successfully!')
21  };
22};
```



The screenshot shows the AWS Lambda function editor with the 'Code source' tab selected. A 'Configure test event' button is visible above the code area. The code is identical to the one in the previous screenshot. The configuration for the test event is set to 'aws:s3:ObjectCreated:Put'.

Configure test event CM-SHIFT-C

```
1 export const handler = async (event) => {
2   if (!event.Records || event.Records.length === 0) {
3     console.error("No records found in the event.");
4     return {
5       statusCode: 400,
6       body: JSON.stringify('No records found in the event')
7     };
8   }
9   // Extract bucket name and object key from the event
10  const record = event.Records[0];
11  const bucketName = record.s3.bucket.name;
12  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\//g, ' ')); // Handle encoded keys
13  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
14  console.log(`Event Source: ${record.eventSource}`);
15  console.log(`Event Source: ${record.eventSource}`);
16  console.log(`Event Source: ${record.eventSource}`);
17  console.log(`Event Source: ${record.eventSource}`);
18  return {
19    statusCode: 200,
20    body: JSON.stringify('Log entry created successfully!')
21  };
22};
```

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

 Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

[Format JSON](#)

```
1 [ ]  
2 "Records": [  
3 {  
4     "eventVersion": "2.0",  
5     "eventSource": "aws:s3",  
6     "awsRegion": "us-east-1",  
7     "eventTime": "1970-01-01T00:00:00.000Z",  
8     "eventName": "ObjectCreated:Put",  
9     "userIdentity": {  
10         "principalId": "EXAMPLE"  
11     },  
12     "requestParameters": {  
13         "sourceIPAddress": "127.0.0.1"  
14     },  
15     "responseElements": {  
16         "x-amz-request-id": "EXAMPLE123456789",  
17         "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmabaisawesome/mnopqrstuvwxyzABCDEFGHI"  
18     },  
19     "s3": {  
20         "s3SchemaVersion": "1.0",  
21         "configurationId": "testConfigRule",  
22         "bucket": {  
23             "name": "example-bucket",  
24             "ownerIdentity": {  
25                 "principalId": "EXAMPLE"  
26             },  
27             "arn": "arn:aws:s3:::example-bucket"  
28         },  
29         "object": {  
30             "key": "test%2Fkey",  
31         }  
32     }  
33 }
```

1:1 JSON Spaces: 2

[Cancel](#)

[Invoke](#)

[Save](#)

The test event **myevent1** was successfully saved.

Function URL [Info](#)

Step 3: Check the logs

- 1) To check the logs explicitly, search for CloudWatch on services and open it in a new tab

The screenshot shows the AWS search interface with the query 'cloud watch' entered in the search bar. The results are categorized into 'Services' and 'Features'.

Services

- CloudWatch (Monitor Resources and Applications)
- Athena (Serverless interactive analytics service)
- Amazon EventBridge (Serverless service for building event-driven applications)
- S3 (Scalable Storage in the Cloud)

Features

- CloudWatch dashboard (Systems Manager feature)
- Data sources (Athena feature)
- Create a SFTP server (AWS Transfer Family feature)
- Event buses

- 2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.

The screenshot shows the AWS CloudWatch Log groups interface. At the top, there's a search bar labeled "Filter log groups or try prefix search" and a checkbox for "Exact match". Below the search bar is a table with one row. The first column is "Log group" with the value "/aws/lambda/lamdaexp12". The second column is "Log class" with the value "Standard". The third column is "Retention" with the value "Never expire". There are also "Configure", "Actions", "View in Logs Insights", and "Start tailing" buttons.

3) Here, under Log streams, select the log stream you want to check.

The screenshot shows the "Log group details" page for the log group "/aws/lambda/lamdaexp12". It includes sections for Log class (Info), ARN, Creation time (3 minutes ago), and Retention (Never expire). Below this is a "Log streams" section with a table showing one log stream: "2024/10/07[\$LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00" with a last event time of "2024-10-07 04:34:00 (UTC)". There are tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, and Data protection.

4) Here again, we can see that 'An image has been added to the bucket'.

The screenshot shows the "Log events" page for the log stream "2024/10/07[\$LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00". It lists several log events, including:

- 2024-10-07T04:45:01.475Z INIT_START Runtime Version: nodejs:20.v39 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:ad902ae231dfc4c3325e183024ccb4d9de1aa14796d98295f898140041242f7
- 2024-10-07T04:45:01.612Z START RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409dd7c Version: \$LATEST
- 2024-10-07T04:45:01.632Z END RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409dd7c
- 2024-10-07T04:45:01.632Z REPORT RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409dd7c Duration: 18.74 ms Billed Duration: 19 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 135.57 ms
- 2024-10-07T04:46:20.191Z START RequestId: 3cc0dc18-4346-4ea5-8c87-eabb2930f86a Version: \$LATEST
- 2024-10-07T04:46:20.194Z END RequestId: 3cc0dc18-4346-4ea5-8c87-eabb2930f86a
- 2024-09-28T09:24:40.324Z 2024-09-28T09:24:40.324Z 01723939-728b-421b-8a58-421085754b5 INFO An image has been added to the bucket example-bucket: test/key
- 2024-10-07T04:46:54.248Z START RequestId: a7de0c16-28c0-4c19-9ea0-5a616357447b Version: \$LATEST
- 2024-10-07T04:46:54.250Z END RequestId: a7de0c16-28c0-4c19-9ea0-5a616357447b
- 2024-10-07T04:46:54.250Z REPORT RequestId: a7de0c16-28c0-4c19-9ea0-5a616357447b Duration: 1.48 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB

At the bottom, it says "No newer events at this moment. Auto retry paused. Resume".

Conclusion: In this experiment, we successfully implemented a Lambda function that automatically logs a message whenever an image is uploaded to a designated S3 bucket. By setting up an S3 bucket trigger, we showcased how AWS services can efficiently interact to automate workflows. The Lambda function captured essential event details, such as the bucket name and object key. After deploying the function, we tested it with a sample image upload and confirmed through CloudWatch logs that the function correctly identified and logged the event. This experiment demonstrated the smooth integration of AWS Lambda with S3 for handling event-driven automation.