

Aim: To develop a website and host it on
a)Your Local Machine/VM
b)Amazon S3
c)Netlify

A)Static Hosting on Local Machine Using XAMPP:

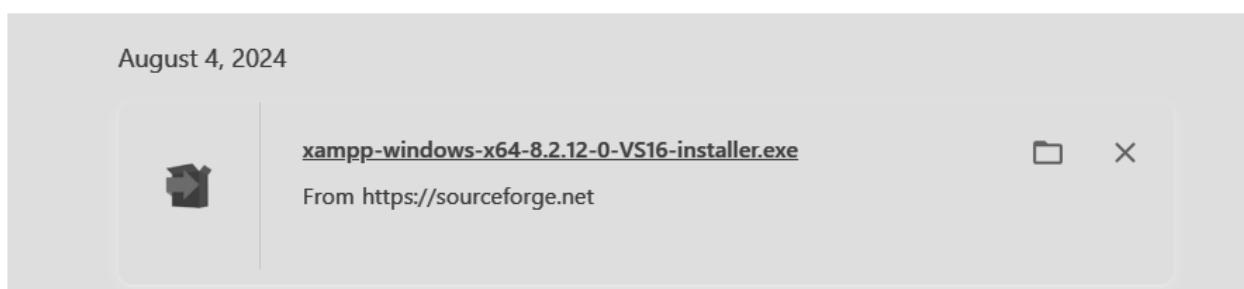
XAMPP :

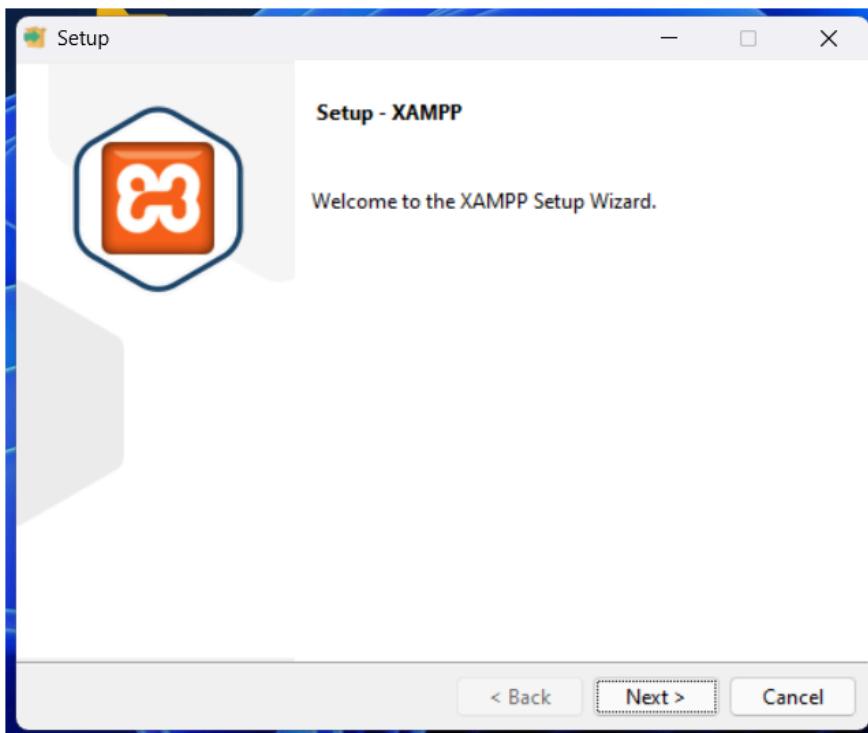
XAMPP is an open-source web server solution stack that provides a local server environment to run and test PHP, MySQL, and Perl-based web applications. It includes Apache, MySQL (MariaDB), PHP, and Perl, packaged together for easy installation. XAMPP is widely used for web development and testing before deploying websites live.

Step 1: Go to the XAMPP Website and Download it according to your Operating System. Here We have downloaded it for Windows.

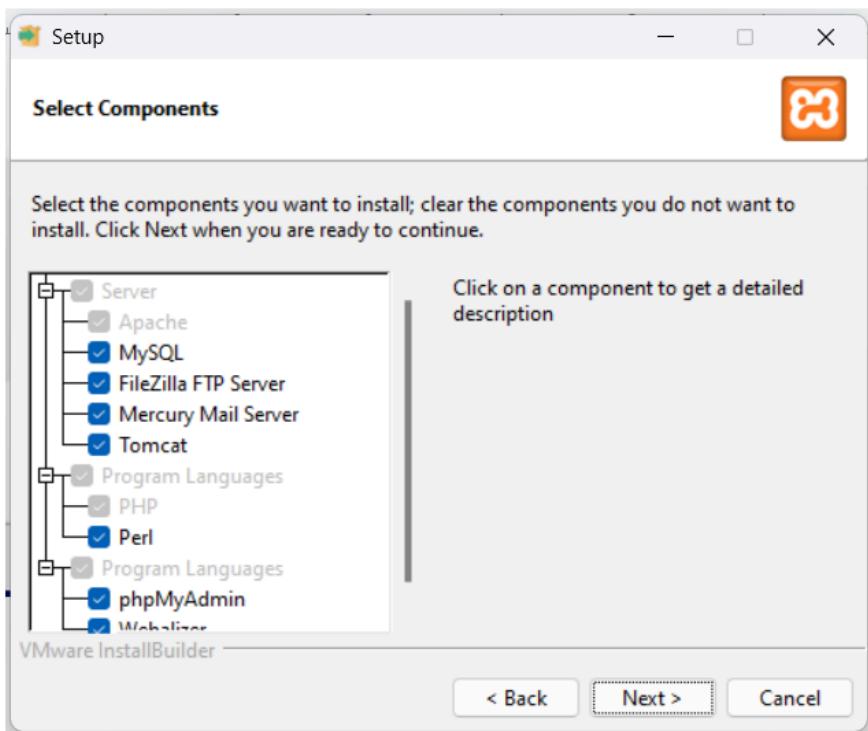


Step 2: Run .exe file.





Step 3:Select Components as per your need Here we have selected all the components.



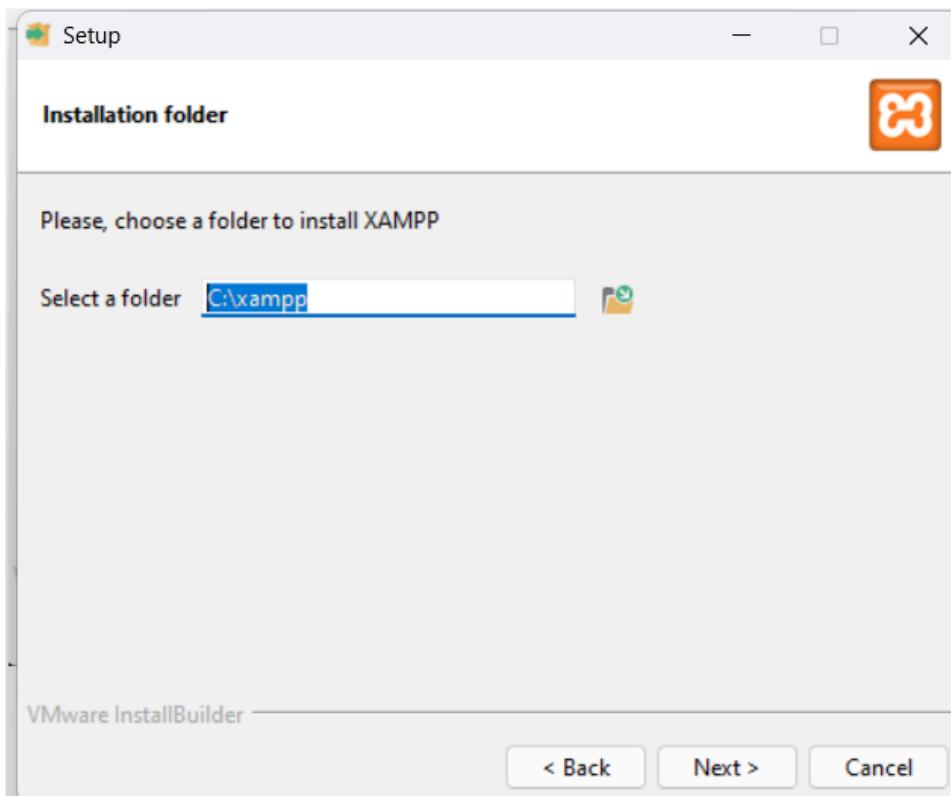
Name:Bhushan Mukund Kor

Academic Year:2024-2025

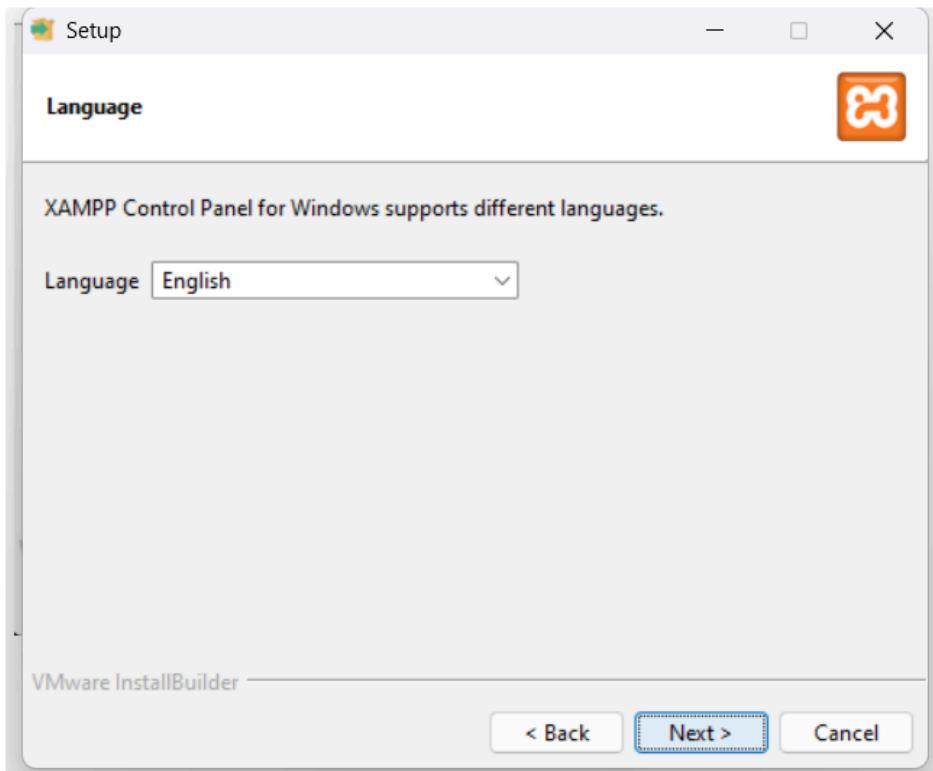
Division: D15C

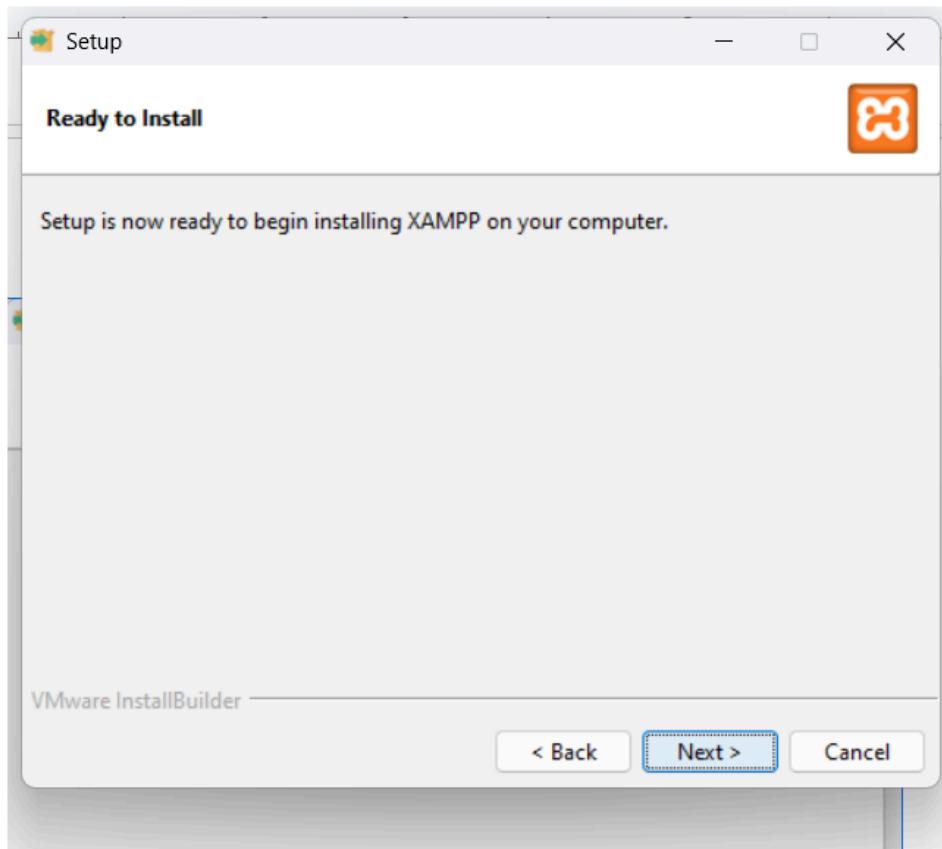
Roll No: 28

Step 4:Set path to C:\xampp

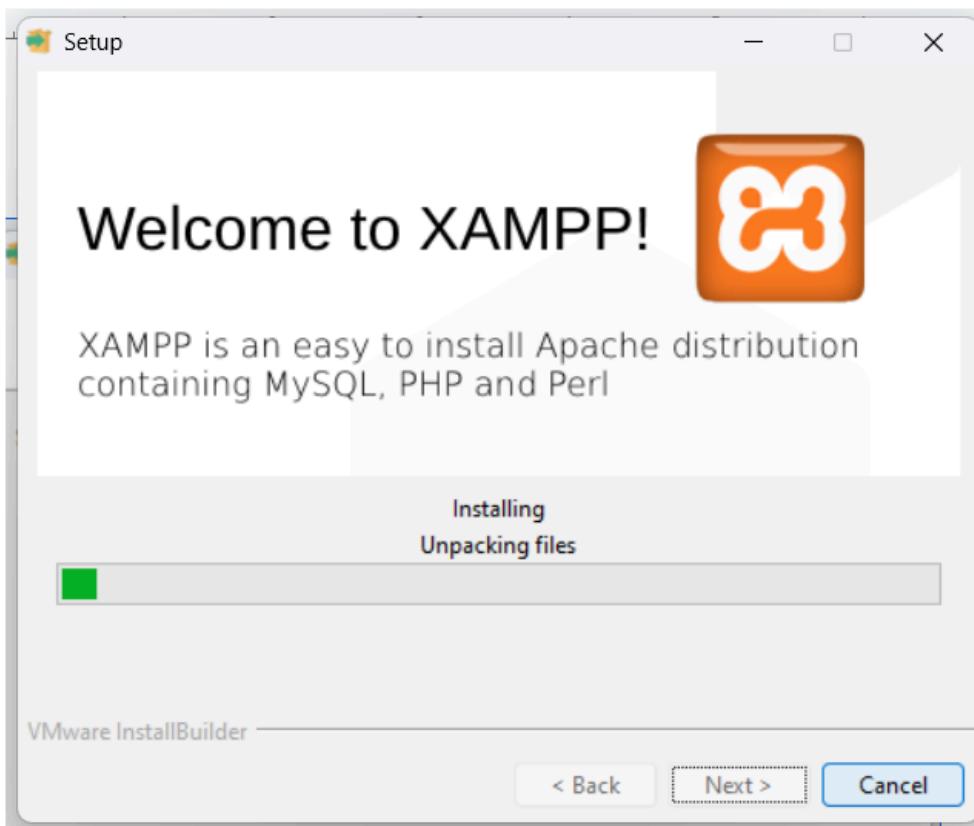


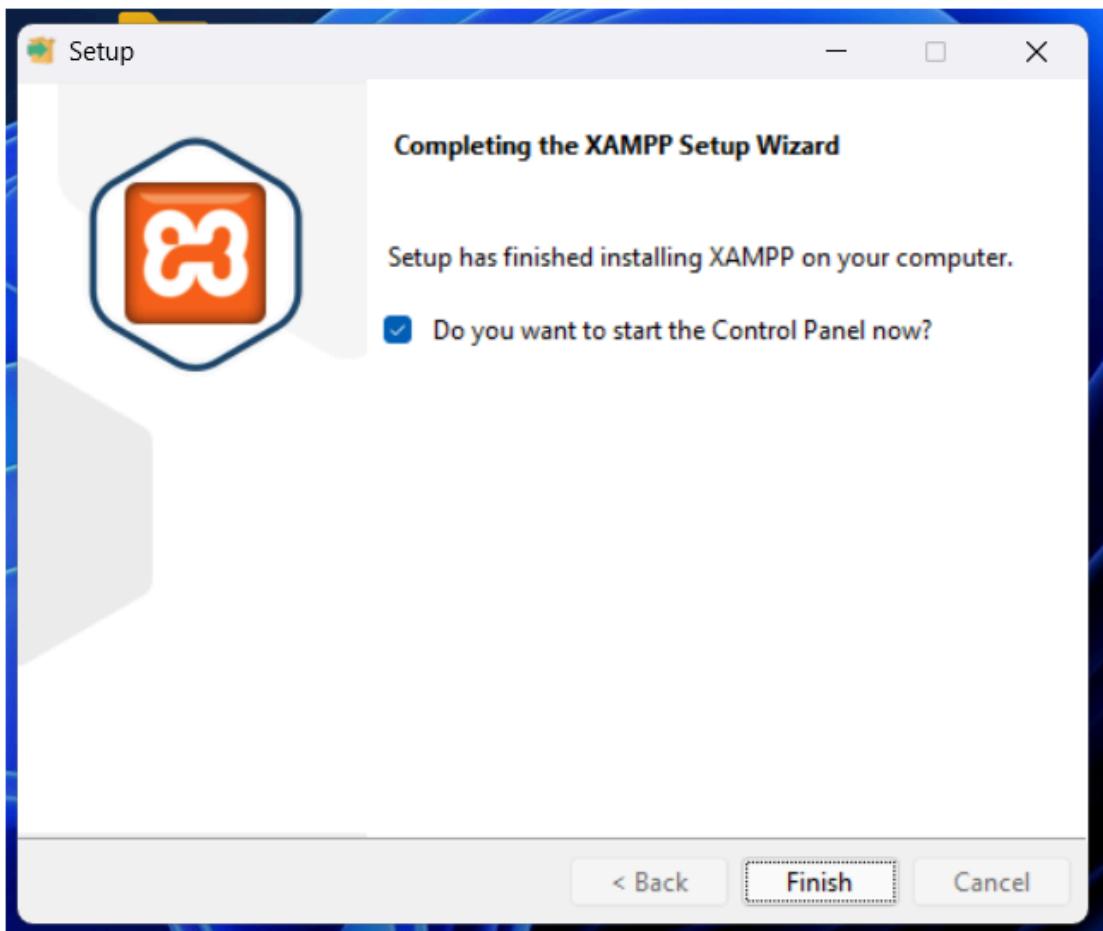
Step 5:Select language and Steup is ready to install click next.



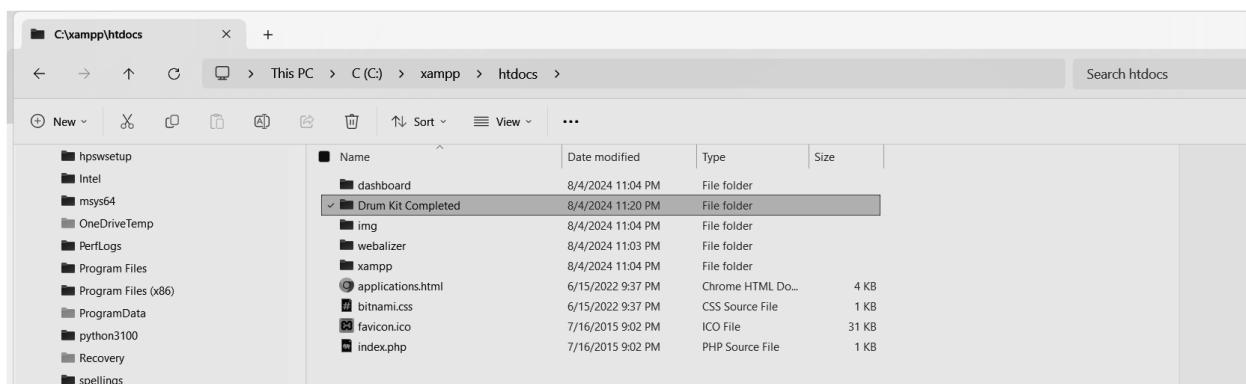


Step 6: Wait till completion of Installation then click on finish and Launch the XAMPP Control Panel.





Step 7:Add your files in htdocs which present in location: C:\xampp\htdocs\



Step 8:If you have index.html change it into index.php and if you have to add some php feature then you can also add it in index.php file by adding php tag.

This PC > C (C:) > xampp > htdocs > Drum Kit Completed >				
	Name	Date modified	Type	Size
	images	8/4/2024 11:20 PM	File folder	
	sounds	8/4/2024 11:20 PM	File folder	
	.DS_Store	6/11/2024 12:08 PM	DS_STORE File	7 KB
	index.js	6/11/2024 12:08 PM	JSFile	2 KB
✓	index.php	8/15/2024 11:28 AM	PHP Source File	1 KB
	index.txt	8/15/2024 11:16 AM	Text Document	1 KB
	styles.css	6/11/2024 12:08 PM	CSS Source File	2 KB

```

<!DOCTYPE html>
<html lang="en" dir="ltr">

<head>
  <meta charset="utf-8">
  <title>Drum Kit</title>
  <link rel="stylesheet" href="styles.css">
  <link href="https://fonts.googleapis.com/css?family=Arvo" rel="stylesheet">
</head>

<body>
  <h1 id="title">Drum Kit</h1>
  <div class="set">
    <button class="w drum">w</button>
    <button class="a drum">a</button>
    <button class="s drum">s</button>
    <button class="d drum">d</button>
    <button class="j drum">j</button>
    <button class="k drum">k</button>
    <button class="l drum">l</button>
  </div>

  <script src="index.js" charset="utf-8"></script>

  <footer>
    Made with ❤ in India.
  </footer>
</body>
</html>

<?php
// Sample PHP code
$title = "This Line is Coming from Echo in PHP part.";
echo "<h2 style='color:white' id='title'>$title</h1>";
?>

```

Name:Bhushan Mukund Kor

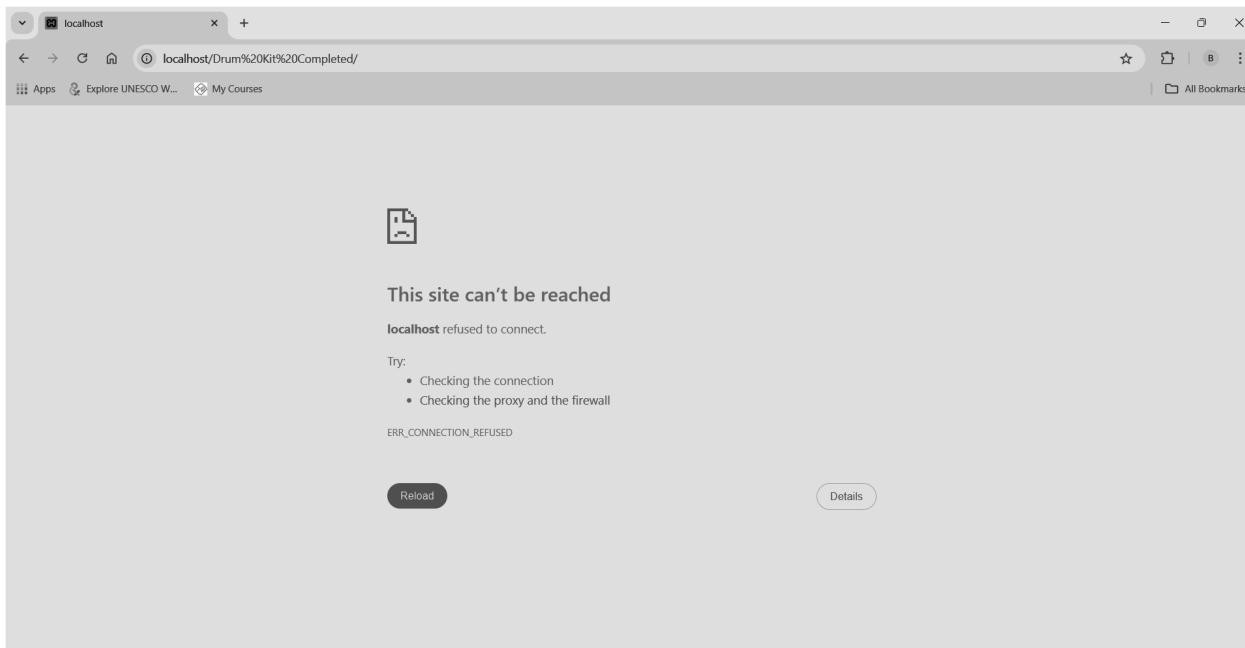
Academic Year:2024-2025

Division: D15C

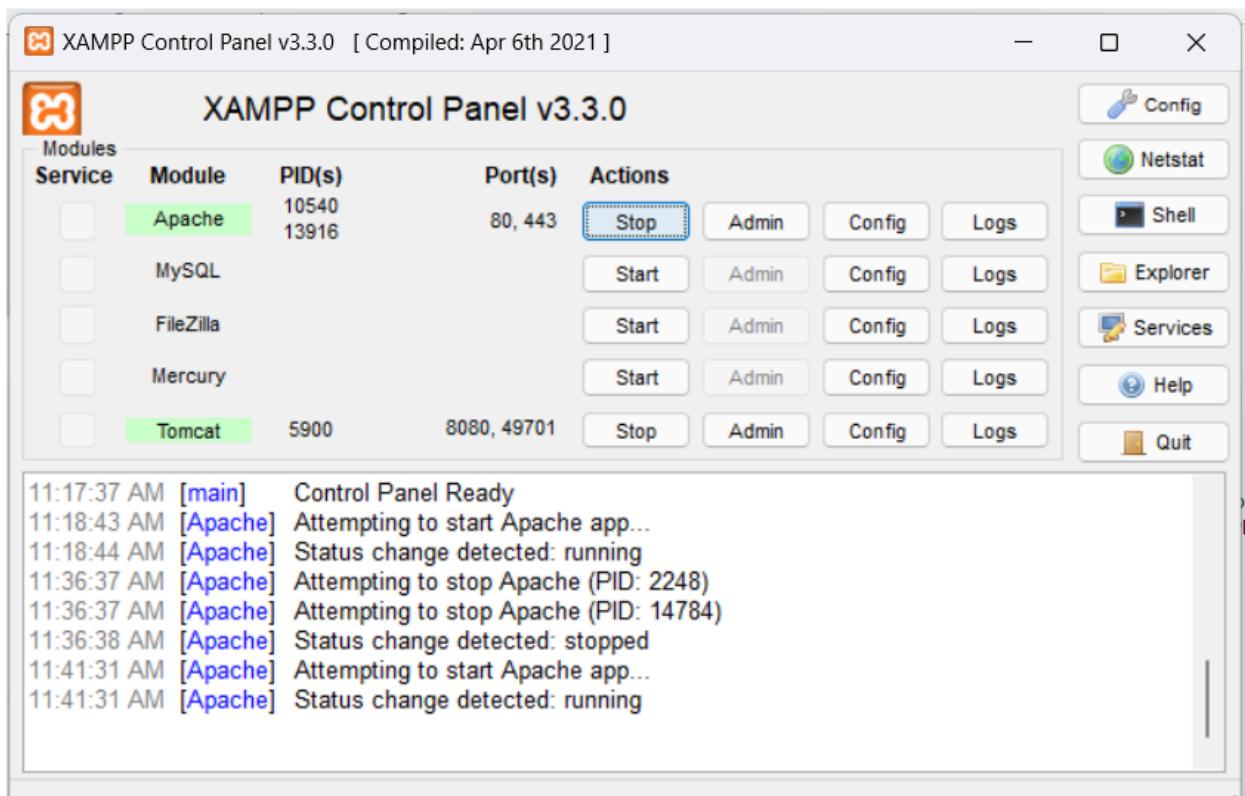
Roll No: 28

Step 9: Start the Server and type “<http://localhost/Drum Kit Completed/>” in the browser or localhost/your filename

Before Starting Server



Start the Server



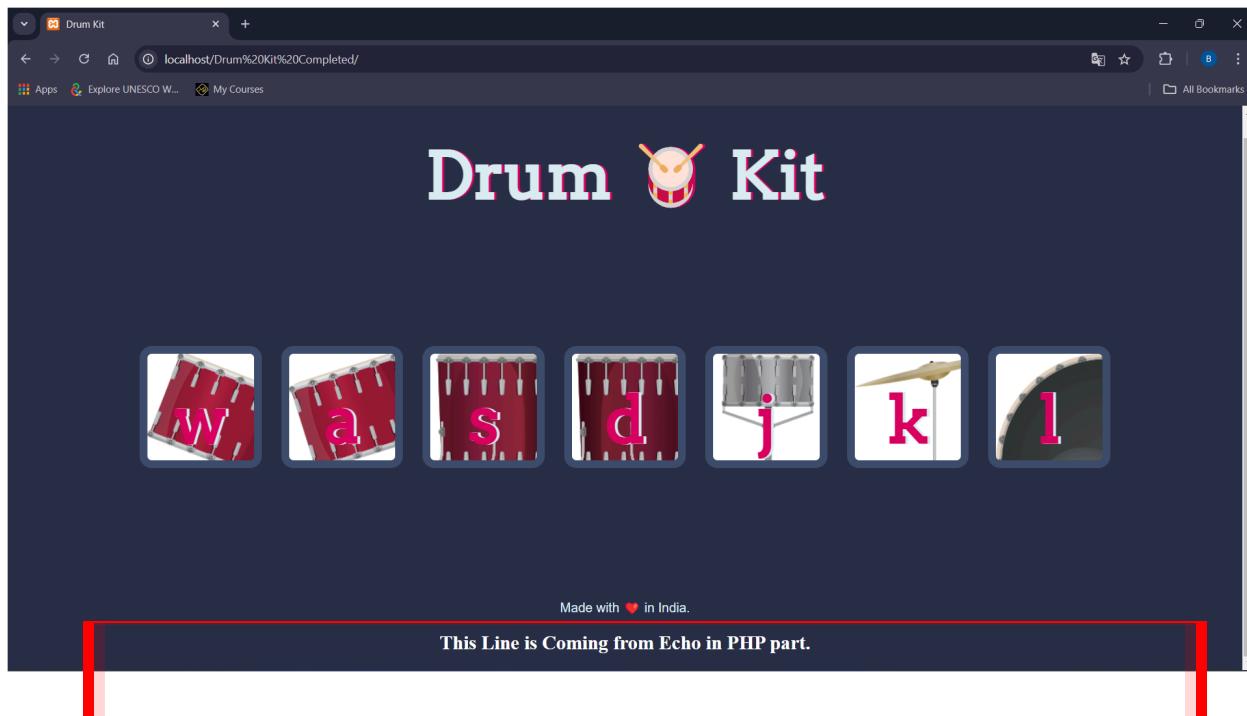
Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

After Starting the server refresh it we can see html and at the end php part



B)Stating hosting Using Amazon S3 Bucket

Amazon S3:Amazon S3 (Simple Storage Service) is a scalable cloud storage service provided by AWS, designed for storing and retrieving any amount of data at any time. It offers high availability, durability, and security, making it ideal for a wide range of use cases such as backup, archiving, and big data analytics. S3 also supports a pay-as-you-go pricing model, allowing businesses to optimize costs.

Step 1:Open Your AWS academy account or Personal AWS account .Here we have used AWS academy account. Then **Search for S3 in Services and click on it.**

The screenshot shows the AWS search interface with the query 's3' entered in the search bar. The results are categorized into 'Services' and 'Features'. The 'Services' section is expanded, showing four items: S3 (Scalable Storage in the Cloud), S3 Glacier (Archive Storage in the Cloud), AWS Snow Family (Large Scale Data Transport), and Storage Gateway (Hybrid Storage Integration). Each service item includes a star icon indicating popularity. The 'Features' section is partially visible below, showing 'Imports from S3' and 'DynamoDB feature'. A 'See all 8 results' link is located above the services list, and a 'See all 39 results' link is located above the features list.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 2:Click on Create Bucket Button.

The screenshot shows the Amazon S3 homepage. At the top right, there is a prominent 'Create a bucket' button. To its left, a text box explains that every object in S3 is stored in a bucket and provides instructions for uploading files and folders. Below this, there is a section titled 'How it works' featuring a video thumbnail for 'Introduction to Amazon S3' and a 'Copy link' button. On the right side of the page, there is a 'Pricing' section stating that there are no minimum fees and providing links for monthly bill estimation and viewing details. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Step 3:Select **Bucket type: General Purpose** then give Bucket name.

The screenshot shows the 'Create bucket' configuration page. The 'General configuration' tab is selected. In the 'AWS Region' section, 'US East (N. Virginia) us-east-1' is chosen. The 'Bucket type' section shows two options: 'General purpose' (selected) and 'Directory - New'. The 'Bucket name' field contains 'staticwebhosting28'. Below the name, a note states that the name must be unique and follow naming rules, with a link to the rules. The 'Copy settings from existing bucket - optional' section indicates that only bucket settings are copied and includes a 'Choose bucket' button and a placeholder for the format 'Format: s3://bucket/prefix'.

Step 4: Keep all the other things By Default Refer to screenshots. And Click on Create Bucket.

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

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

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources

dShell Feedback

- 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.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

dShell Feedback

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the **Storage** tab of the [Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable
 Enable

Advanced settings

Info After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Create bucket

Step 5:Bucket is created Successfully.

AWS Services Search [Alt+S] N. Virginia v vocabs/user3385470=KOR_BHUSHAN_MUKUND @ 8473-0277-0411 ▾

Amazon S3 > Buckets

Account snapshot - updated every 24 hours [All AWS Regions](#) [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets [Directory buckets](#)

General purpose buckets (1) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Find buckets by name [C](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Name	AWS Region	IAM Access Analyzer	Creation date
staticwebhosting28	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 4, 2024, 23:33:54 (UTC+05:30)

© 2024 Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshot shows the AWS S3 Bucket configuration for 'staticwebhosting28'. In the 'Bucket overview' section, it displays the AWS Region as 'US East (N. Virginia) us-east-1', the Amazon Resource Name (ARN) as 'arn:aws:s3:::staticwebhosting28', and the Creation date as 'August 4, 2024, 23:33:54 (UTC+05:30)'. In the 'Bucket Versioning' section, it shows that Versioning is disabled, Multi-factor authentication (MFA) delete is disabled, and there is no additional layer of security for changing Bucket Versioning settings.

Step 6: By Default Static hosting is disabled so enable it.Also give name of index document here it is index.html

The screenshot shows the 'Static website hosting' section for the 'staticwebhosting28' bucket. It indicates that the bucket is currently used to host a website or redirect requests, with a note about using the bucket endpoint as the web address. The 'Static website hosting' status is set to 'Disabled'.

The screenshot shows the 'Edit static website hosting' configuration page for the 'staticwebhosting28' bucket. Under 'Static website hosting', the 'Enable' option is selected. Under 'Hosting type', the 'Host a static website' option is selected, with a note explaining that the bucket endpoint will be used as the web address. A callout box provides information about making content publicly readable using S3 Block Public Access. Under 'Index document', the value 'index.html' is specified as the home or default page of the website.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 7: Click on the Bucket name and then Upload your files in Bucket by clicking on the upload button refer screenshots.

Amazon S3 > Buckets > staticwebhosting28

staticwebhosting28 [Info](#)

Objects (0) [Info](#)

C [Copy S3 URI](#) [Copy URL](#) [Download](#) Open Delete Actions Create folder [Upload](#)

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

Find objects by prefix

< 1 > | [@](#)

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

[Upload](#)

Amazon S3 > Buckets > staticwebhosting28 > **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.

Name	Folder	Type
Files and folders (0)		

Remove [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

Find by name

< 1 >

Name	Folder	Type
No files or folders You have not chosen any files or folders to upload.		

Destination [Info](#)

Amazon S3 > Buckets > staticwebhosting28 > 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 (18 Total, 374.7 KB)			Remove	Add files	Add folder
All files and folders in this table will be uploaded.					
<input type="text"/> Find by name			<	1	2 >
<input type="checkbox"/>	Name	Folder	▼	Type	
<input type="checkbox"/>	.DS_Store	Drum Kit Completed/	-		
<input type="checkbox"/>	index.html	Drum Kit Completed/	text/html		
<input type="checkbox"/>	index.js	Drum Kit Completed/	text/javascript		
<input type="checkbox"/>	styles.css	Drum Kit Completed/	text/css		
<input type="checkbox"/>	crash.mp3	Drum Kit Completed/sounds/	audio/mpeg		
<input type="checkbox"/>	kick-bass.mp3	Drum Kit Completed/sounds/	audio/mpeg		

Files and folders (18 Total, 374.7 KB)			Remove	Add files	Add folder
All files and folders in this table will be uploaded.					
<input type="text"/> Find by name			<	1	2 >
<input type="checkbox"/>	Name	Folder	▼	Type	

Destination Info

Destination
s3://staticwebhosting28

► **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.

Cancel **Upload**

Step 8: Now click on the link. Then you will get to know access is block so change the access settings.

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://staticwebhosting28.s3-website-us-east-1.amazonaws.com>

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: BG0328HP96F83TQ1
- HostId: OfGAgKa7cFKwwCAVutc37YPwLicC9TNEfOWsgROBYudamAXB79gJbcBFAMbT16vPX6bO8A49wcE=

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Edit Block public access (bucket settings)

[Info](#)

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

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

Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access 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)

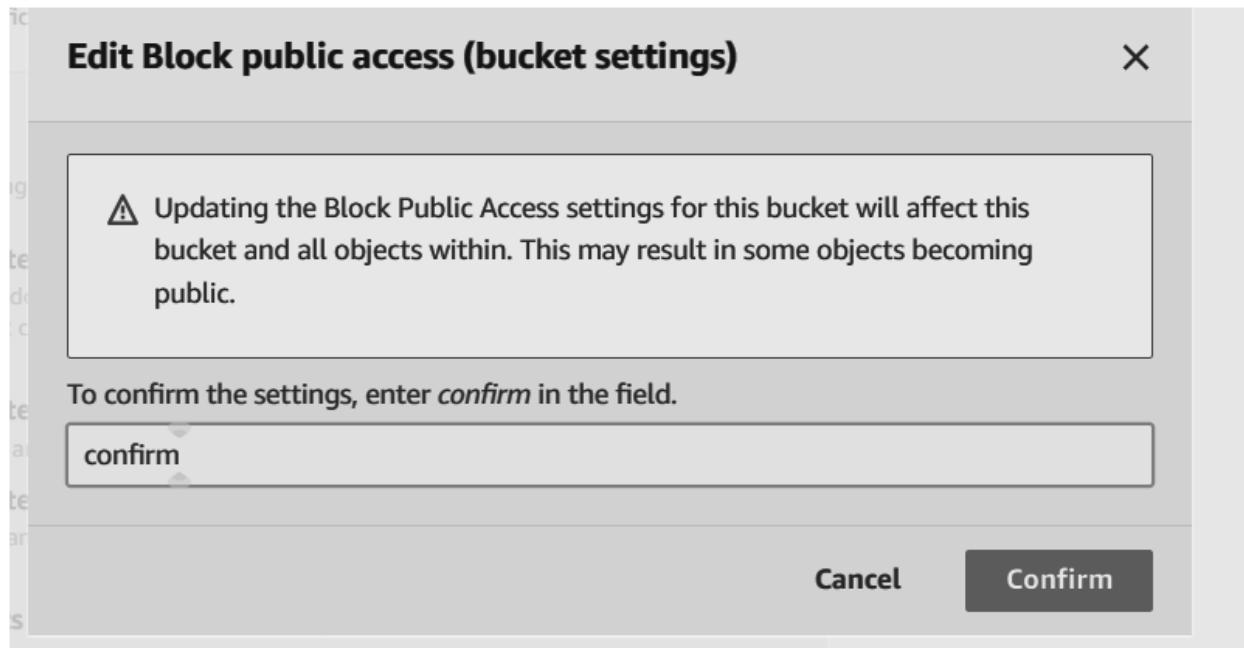
S3 will ignore all ACLs that grant public access to buckets and objects.

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

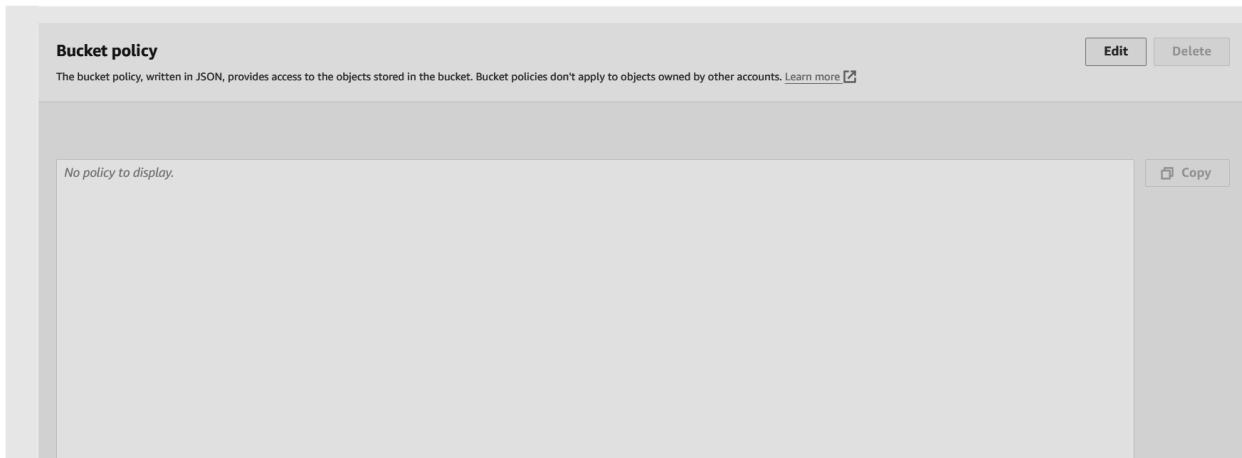
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Step 9: The website is still not working because there is one more step that you have to add some policies so click on edit policy. And Paste the below policy in the box and save it.

Do not forget to replace your Bucket Name.

```
{  
  "Version": "2012-10-17",  
  "Statement" : [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "*"  
      },  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::YOUR-BUCKET-NAME-HERE/*"  
    }  
  ]  
}
```

A screenshot of the AWS S3 Bucket ARN and Policy pages. The ARN is listed as "arn:aws:s3:::staticwebhosting28". The Policy section shows the following JSON code:

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "PublicReadGetObject",  
6       "Effect": "Allow",  
7       "Principal": {  
8         "AWS": "*"  
9       },  
10      "Action": "s3:GetObject",  
11      "Resource": "arn:aws:s3:::staticwebhosting28/*"  
12    }  
13  ]  
14}  
15
```

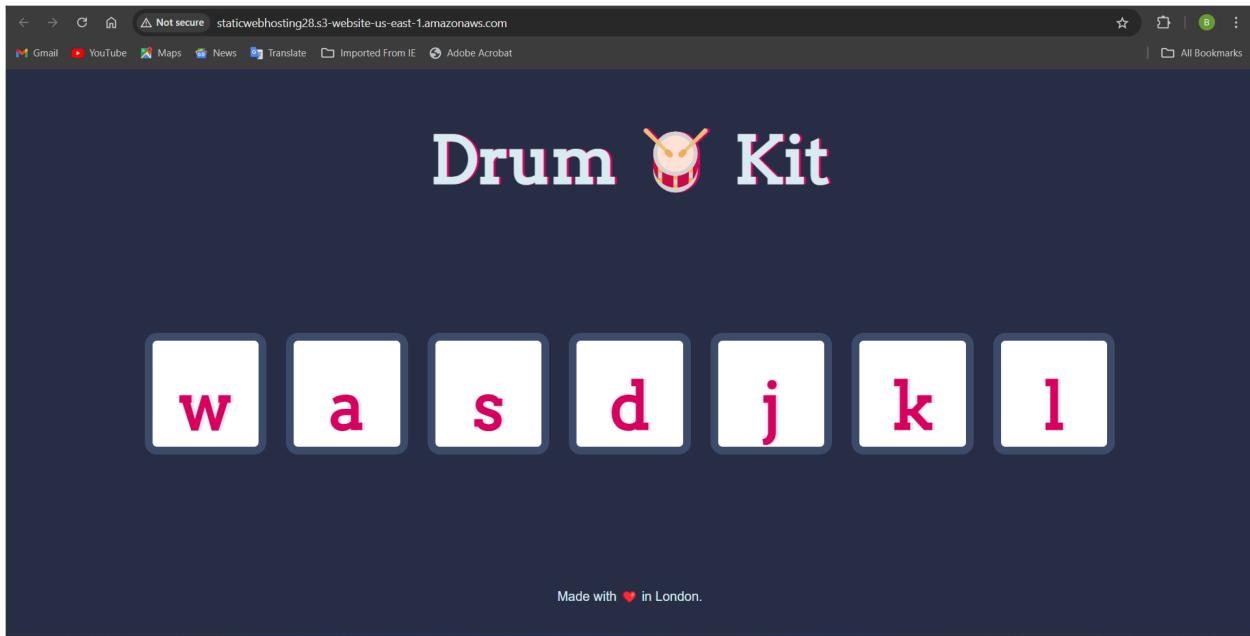
Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 10: Done Now go to the link where you can see your website.



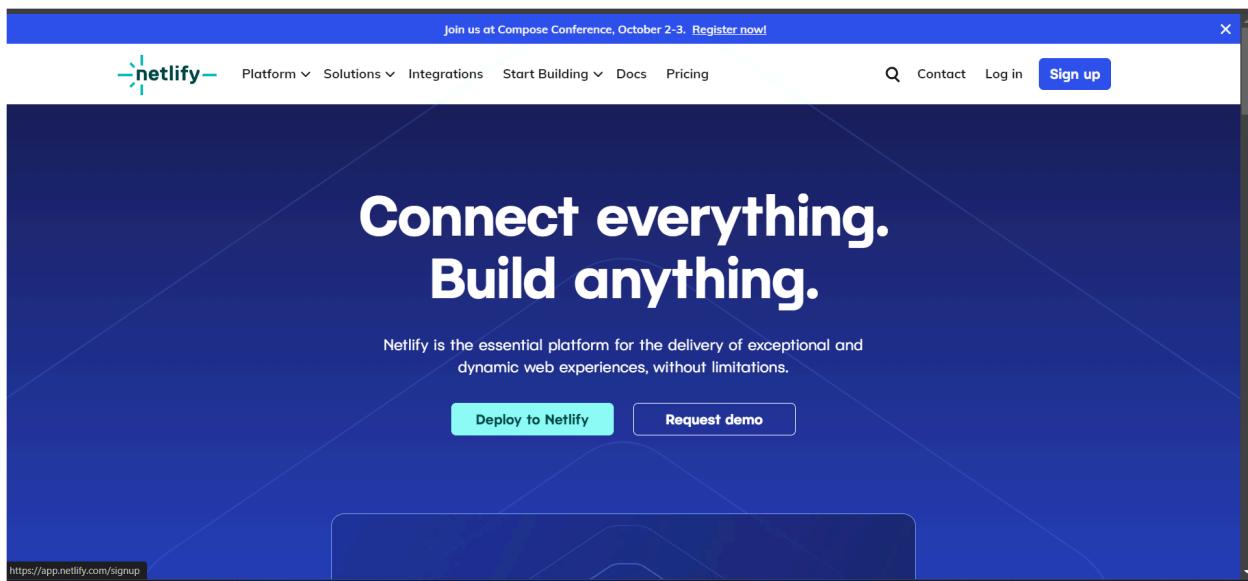
C)Static Hosting Using Netlify

Netlify

Netlify is a cloud platform that automates the deployment, management, and hosting of modern web projects. It provides developers with tools for continuous integration, automatic builds, and seamless deployment from Git repositories. Netlify supports static sites and serverless functions, making it ideal for JAMstack applications.

It is the easiest and fastest way to host a website.

Step1: Create your account or Sign up with a Google account on Netlify.



Step 2: Give Some Basic Information Regarding your web application. Add your files or connect to a GitHub account.

—netlify—

Nice to meet you! Let's get acquainted.

How are you planning to use Netlify?

Personal School Client Work

ⓘ Required field

What kind of site do you want to build first?

Company site Web app eCommerce Games
 Web 3.0 Documentation Personal site Blog
 Something else

What are you building?

Something else

What are you building?

Web 2.0

What best describes your role?

Freelancer Hobby Developer Other

What is the name of your team?

Name your team

Drum Kit

💡 People who use Netlify for personal work often use a project name or their own name.

Continue to deploy

ⓘ By submitting this form, you agree to Netlify's [Privacy Policy](#) and that Netlify can sen

Step 3: Click on the link where you can access your web application and you can also share this link with your friends.

The screenshot shows the Netlify site overview for a project named 'rainbow-biscuit-058458'. The left sidebar has sections for Site overview, Site configuration, Deployments (selected), Logs, Integrations, Metrics, Domain management, Forms, and Blobs. A prominent 'Upgrades' button is at the bottom. The main area features a 'Get started with Netlify' guide with steps 1/7 ('Let's build your first site!') and 2/7 ('Customize site name'). Below it is a 'Published deploy for rainbow-biscuit-058458' card showing deployment details: Today at 12:21 AM, Production, Download, Open production deploy, Lock to stop auto publishing, and Options. At the bottom is a 'Deploy summary' section indicating all files are already uploaded.

The screenshot shows the public website 'Drum Kit' at the URL <https://66afcda365041dea79dccce0--rainbow-biscuit-058458.netlify.app/>. The page has a dark blue background. At the top center is the title 'Drum 🥁 Kit'. Below it is a row of seven images, each containing a letter from 'w' to 'l', representing different drum components. At the bottom center is the text 'Made with ❤️ in London.'

Step 4: Done. This is the Public link.
Link: <https://66afcda365041dea79dccce0--rainbow-biscuit-058458.netlify.app/>

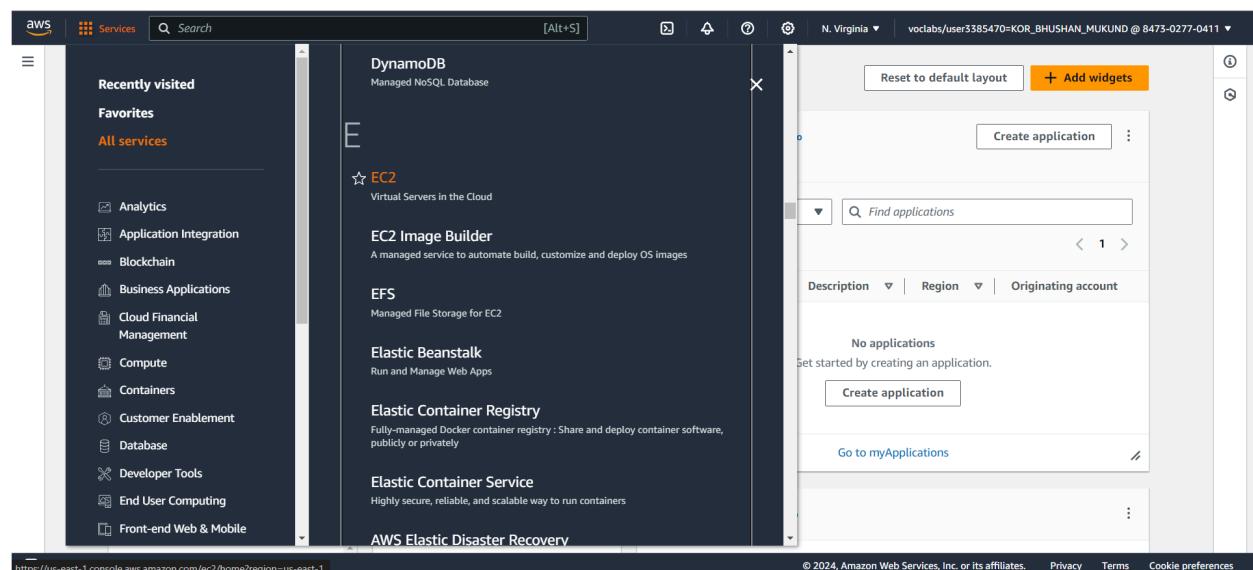
Aim: To understand the benefits of Cloud Infrastructure, Setup EC2 and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE, and Perform Collaboration Demonstration.

A)EC2

EC2

Amazon EC2 (Elastic Compute Cloud) is a web service that provides resizable compute capacity in the cloud. It allows users to run virtual servers, known as instances, on demand. EC2 offers flexibility in scaling resources up or down based on your needs, making it ideal for hosting applications, running batch jobs, and more, with only the cost of the resources you actually use.

Step 1: Open Your AWS Academy or Personal AWS account and Search for EC2 in services.



Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 2: After Opening EC2 click on Launch instance.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, there are several sections: EC2 Global View, Events, Console-to-Code (with a 'Preview' link), Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area has a 'Resources' summary table and a 'Launch instance' section. The 'Launch instance' section contains a large orange 'Launch instance' button and a smaller 'Migrate a server' button. Below these buttons is a note: 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right of the main content are 'Account attributes' (Default VPC, Settings, Explore AWS), 'Service health' (AWS Health Dashboard), and promotional sections for cost reduction and best price-performance.

Step 3: Give a name to your instance.

The screenshot shows the 'Launch an instance' wizard. The first step is 'Name and tags'. It has a 'Name' field containing 'MyEC2' and an 'Add additional tags' link. Below this is a search bar for 'Application and OS Images (Amazon Machine Image)'. The second step is 'Summary', which shows the configuration: 1 instance, Canonical, Ubuntu, 24.04 LTS AMI, t4g.nano instance type, New security group, and 1 volume(s) - 8 GiB. A tooltip for the Free tier is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t2.micro in the Regions in which you launch this instance)'. At the bottom are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

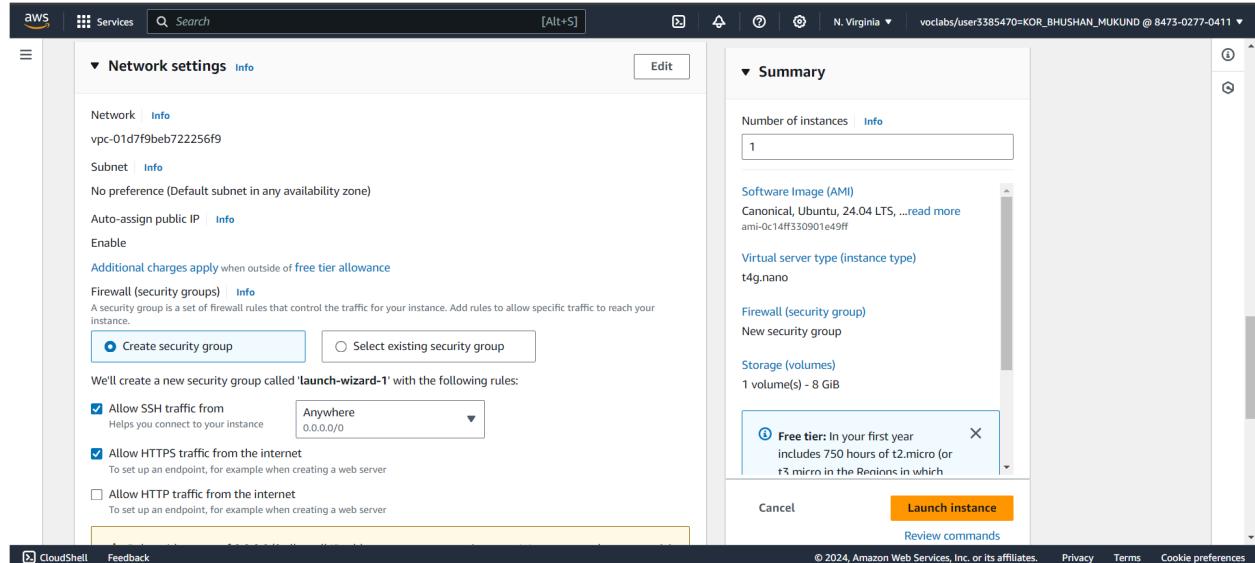
Step 4: Select the server as Ubuntu and you can select Architecture x86 or ARM .

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the name is set to 'MyEC2'. Under 'Application and OS Images (Amazon Machine Image)', the software image is set to Canonical, Ubuntu, 24.04 LTS, and the virtual server type is t4g.nano. A tooltip for the Free tier indicates it includes 750 hours of t2.micro or t3.micro in the Regions in which the instance runs. The 'Launch instance' button is highlighted.

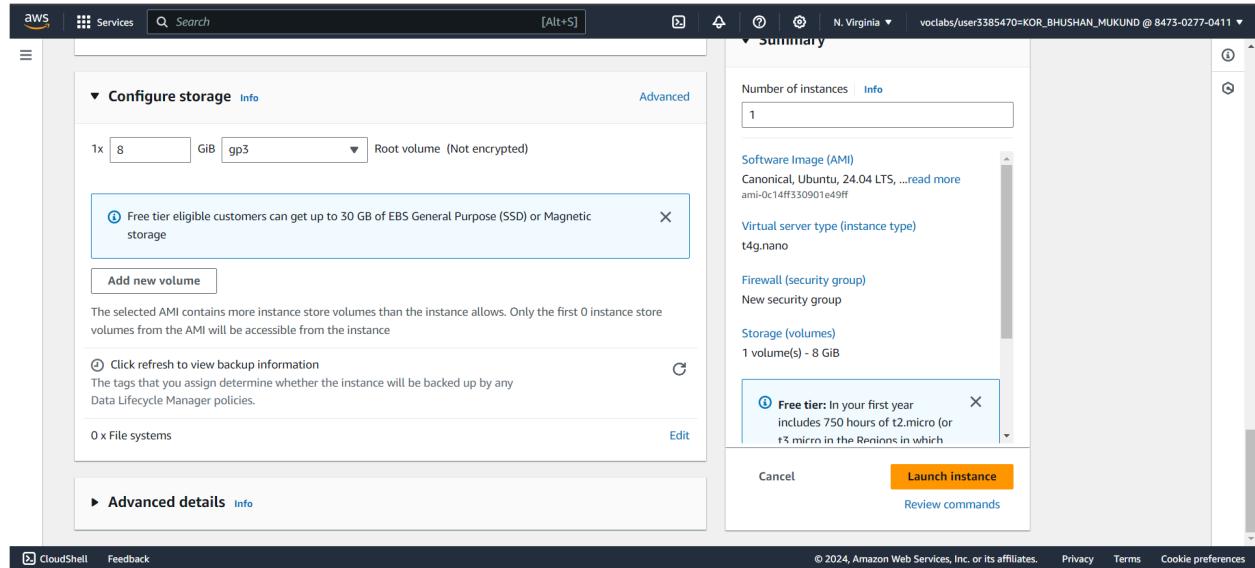
Step 5: Keep instance type by default and key pair as default (vockey).

The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Instance type' section is set to t4g.nano. The 'Key pair (login)' section has 'vockey' selected. Under 'Network settings', the network interface is listed as 'vpc_01d7f0beb722256f0'. A tooltip for the Free tier indicates it includes 750 hours of t2.micro or t3.micro in the Regions in which the instance runs. The 'Launch instance' button is highlighted.

Step 6:Select the Create Security Group option and allow 2 permissions refer to screenshot.
And if any error occurs allow only 1st permission.



Step 7: Configure storage to 8 GiB and gp3 if any error occurs make it to 10 GiB and gp3.



Step 10: Done Your EC2 instance is successfully created.

The screenshot shows the AWS EC2 Instances page. At the top, there's a green success message: "Successfully initiated launch of instance (i-0c35b5bbffdd03e5c)". Below this, under "Next Steps", there are four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". Each card has a "Learn more" link. At the bottom, it shows the instance details: MyEC2 (i-0c35b5bbffdd03e5c), Running, t4g.nano, 2/2 checks passed, us-east-1a, and ec2-23-20-172-31-26-195-166.compute-1.amazonaws.com.

Step 11: Click on the instance ID then click on connect to connect to environment.

The screenshot shows the AWS EC2 Instances page with the instance summary for i-0c35b5bbffdd03e5c (MyEC2). The summary includes fields like Instance ID, Public IPv4 address, Private IP DNS name, Instance state, Instance type, VPC ID, Subnet ID, and Instance ARN. On the left sidebar, the "Instances" section is expanded, showing options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store.

Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws aarch64)

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

System information as of Thu Aug 8 15:14:08 UTC 2024

System load: 0.08	Temperature: -273.1 C
Usage of /: 23.0% of 6.71GB	Processes: 142
Memory usage: 70%	Users logged in: 0
Swap usage: 0%	IPv4 address for ens5: 172.31.26.195

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>".

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

Step 12: Run the following command in Ubuntu (On EC2 Instance).

1. sudo -l
2. sudo apt update
3. uname -a
4. df --help and df
5. ls
6. mkdir test
7. ls
8. cd test
9. touch file1
10. ls
11. touch file2 file3
12. ls
13. rm file1
14. ls
15. rm file*
16. ls
17. cd
18. ls
19. rmdir test
20. ls
21. mkdir test1 test2 test3
22. ls
23. rmdir test*
24. ls

25. History

26. top

27. vmstat

```
ubuntu@ip-172-31-26-195:~$ sudo -l
Matching Defaults entries for ubuntu on ip-172-31-26-195:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User ubuntu may run the following commands on ip-172-31-26-195:
(ALL : ALL) ALL
(ALL : NOPASSWD: ALL)

ubuntu@ip-172-31-26-195:~$ sudo apt update
Hit:1 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble InRelease
Get:2 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates InRelease [126 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease [126 kB]
Get:4 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports InRelease [126 kB]
Get:5 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/universe arm64 Packages [15.3 MB]
Get:6 http://ports.ubuntu.com/ubuntu-ports noble-security/main arm64 Packages [262 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports noble-security/main Translation-en [64.4 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports noble-security/main arm64 Components [5428 B]
Get:9 http://ports.ubuntu.com/ubuntu-ports noble-security/main arm64 c-n-f Metadata [3696 B]
Get:10 http://ports.ubuntu.com/ubuntu-ports noble-security/universe arm64 Packages [244 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports noble-security/universe Translation-en [108 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports noble-security/universe arm64 Components [8632 B]
Get:13 http://ports.ubuntu.com/ubuntu-ports noble-security/universe arm64 c-n-f Metadata [9356 B]
Get:14 http://ports.ubuntu.com/ubuntu-ports noble-security/restricted arm64 Packages [206 kB]
Get:15 http://ports.ubuntu.com/ubuntu-ports noble-security/restricted Translation-en [40.7 kB]
Get:16 http://ports.ubuntu.com/ubuntu-ports noble-security/restricted arm64 Components [212 B]
Get:17 http://ports.ubuntu.com/ubuntu-ports noble-security/restricted arm64 c-n-f Metadata [372 B]
Get:18 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse arm64 Packages [10.1 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse Translation-en [2808 B]
Get:20 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse arm64 Components [212 B]
Get:21 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse arm64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/universe Translation-en [5982 kB]
Get:23 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/universe arm64 Components [3573 kB]
Get:24 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/universe arm64 c-n-f Metadata [295 kB]
Get:25 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 Packages [223 kB]
Get:26 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/multiverse Translation-en [118 kB]
Get:27 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 Components [31.6 kB]
Get:28 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble/multiverse arm64 c-n-f Metadata [7152 B]
Get:29 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 Packages [334 kB]
Get:30 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/main Translation-en [86.2 kB]
Get:31 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 c-n-f Metadata [5720 B]
Get:32 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 Packages [215 kB]
Get:33 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/universe Translation-en [135 kB]
Get:34 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 Components [45.0 kB]
Get:35 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 c-n-f Metadata [12.5 kB]
Get:36 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/restricted arm64 Packages [237 kB]
Get:37 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/restricted Translation-en [46.4 kB]
Get:38 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/restricted arm64 c-n-f Metadata [368 B]
Get:39 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/multiverse arm64 Packages [10.1 kB]
Get:40 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/multiverse Translation-en [3608 B]
Get:41 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/multiverse arm64 Components [212 B]
Get:42 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-updates/multiverse arm64 c-n-f Metadata [340 B]
Get:43 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/main arm64 Components [208 B]
Get:44 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/main arm64 c-n-f Metadata [112 B]
Get:45 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/universe arm64 Packages [10.3 kB]
Get:46 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/universe Translation-en [10.5 kB]
Get:47 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/universe arm64 Components [17.7 kB]
Get:48 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/universe arm64 c-n-f Metadata [1020 B]
Get:49 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/restricted arm64 Components [216 B]
Get:50 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/restricted arm64 c-n-f Metadata [116 B]
Get:51 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/multiverse arm64 Components [212 B]
Get:52 http://us-east-1.ec2.ports.ubuntu.com/ubuntu-ports noble-backports/multiverse arm64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 13s (2122 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
ubuntu@ip-172-31-26-195:~$ uname -a
Linux ip-172-31-26-195 6.8.0-1009-aws #9-Ubuntu SMP Fri May 17 20:15:49 UTC 2024 aarch64 aarch64 aarch64 GNU/Linux
ubuntu@ip-172-31-26-195:~$
```

```
ubuntu@ip-172-31-26-195:~$ df --help
Usage: df [OPTION]... [FILE]...
Show information about the file system on which each FILE resides,
or all file systems by default.

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

Display values are in units of the first available SIZE from --block-size,
and the DF_BLOCK_SIZE, BLOCK_SIZE and BLOCKSIZE environment variables.
Otherwise, units default to 1024 bytes (or 512 if POSIXLY_CORRECT is set).

Display values are in units of the first available SIZE from --block-size,
and the DF_BLOCK_SIZE, BLOCK_SIZE and BLOCKSIZE environment variables.
Otherwise, units default to 1024 bytes (or 512 if POSIXLY_CORRECT is set).

The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y,R,Q (powers of 1024) or KB,MB,... (powers of 1000).
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

FIELD_LIST is a comma-separated list of columns to be included. Valid
field names are: 'source', 'fstype', 'itotal', 'iused', 'iavail', 'ipcent',
'size', 'used', 'avail', 'pcent', 'file' and 'target' (see info page).

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>
Report any translation bugs to <<https://translationproject.org/team/>>
Full documentation <<https://www.gnu.org/software/coreutils/df>>
or available locally via: info '(coreutils) df invocation'

```
ubuntu@ip-172-31-26-195:~$ mkdir test
ubuntu@ip-172-31-26-195:~$ ls
test
ubuntu@ip-172-31-26-195:~$ cd test
ubuntu@ip-172-31-26-195:~/test$ touch file1
ubuntu@ip-172-31-26-195:~/test$ ls
file1
ubuntu@ip-172-31-26-195:~/test$ touch file2 file3
ubuntu@ip-172-31-26-195:~/test$ ls
file1 file2 file3
ubuntu@ip-172-31-26-195:~/test$ rm file1
ubuntu@ip-172-31-26-195:~/test$ ls
file2 file3
ubuntu@ip-172-31-26-195:~/test$ rm file*
ubuntu@ip-172-31-26-195:~/test$ ls
ubuntu@ip-172-31-26-195:~/test$ cd
```

```
ubuntu@ip-172-31-26-195:~$ ls
test
ubuntu@ip-172-31-26-195:~$ rmdir test
ubuntu@ip-172-31-26-195:~$ ls
ubuntu@ip-172-31-26-195:~$ mkdir test1 test2 test3
ubuntu@ip-172-31-26-195:~$ ls
test1 test2 test3
ubuntu@ip-172-31-26-195:~$ rmdir test*
ubuntu@ip-172-31-26-195:~$ ls
ubuntu@ip-172-31-26-195:~$ history
```

```
ubuntu@ip-172-31-26-195:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        7034376  1831888   5186104  27% /
tmpfs            212128       0   212128   0% /dev/shm
tmpfs            84852     1040    83812   2% /run
tmpfs            5120       0     5120   0% /run/lock
efivarfs          128       4     125   3% /sys/firmware/efi/efivars
/dev/nvme0n1p16   911580   57648    790292   7% /boot
/dev/nvme0n1p15   99791    6475    93317   7% /boot/efi
tmpfs            42424      12    42412   1% /run/user/1000
```

```
ubuntu@ip-172-31-26-195:~$ history
 1 sudo -l
 2 apt update
 3 sudo apt update
 4 mkdir test
 5 ls
 6 cd test
 7 touch file1
 8 ls
 9 touch file2 file3
10 ls
11 rm file1
12 ls
13 rm file*
14 ls
15 cd
16 ls
17 rmdir test
18 mkdir test1 test2 test3
19 ls
20 rmdir test*
21 ls
22 history
23 top
24 vmstat
25 df
26 whatis df
27 df --help
28 uname -a
29 ls
30 mkdir test
31 ls
32 rmdir test
33 ls
34 mkdir test1 test2 test3
35 ls
36 rmdir test*
37 ls
38 history
ubuntu@ip-172-31-26-195:~$
```

```
ubuntu@ip-172-31-26-195:~$ vmstat
procs -----memory----- ---swap-- ----io---- -system-- -----cpu-----
r b    swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st gu
2 0      0 36688   3180 128208    0    0   193   137   43    0  0  0 99  0  0  0
```

```
aws | Services Q Search [Alt+S] N. Virginia ▾ vocabs/user3385470=KOR_BHUSHAN_MUKUND @ 8473-0277-0411 ▾
top - 15:21:51 up 1:09, 1 user, load average: 0.00, 0.02, 0.00
Tasks: 139 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MIB Mem : 414.3 total, 33.1 free, 272.6 used, 127.2 buff/cache
MIB Swap: 0.0 total, 0.0 free, 0.0 used. 141.8 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 22456 9668 5572 S 0.0 2.3 0:01.64 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthread
3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 pool_workqueue_release
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rCU_g
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rCU_p
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-slab_
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-netns
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0-kblockd
12 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-mm_p
13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
14 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
15 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
16 root 20 0 0 0 0 I 0.0 0.0 0:00.09 rcu_sched
17 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
18 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
20 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
21 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/1
22 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/1
23 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/1

ubuntu@irn-172-31-26-105:~$ iometest
```

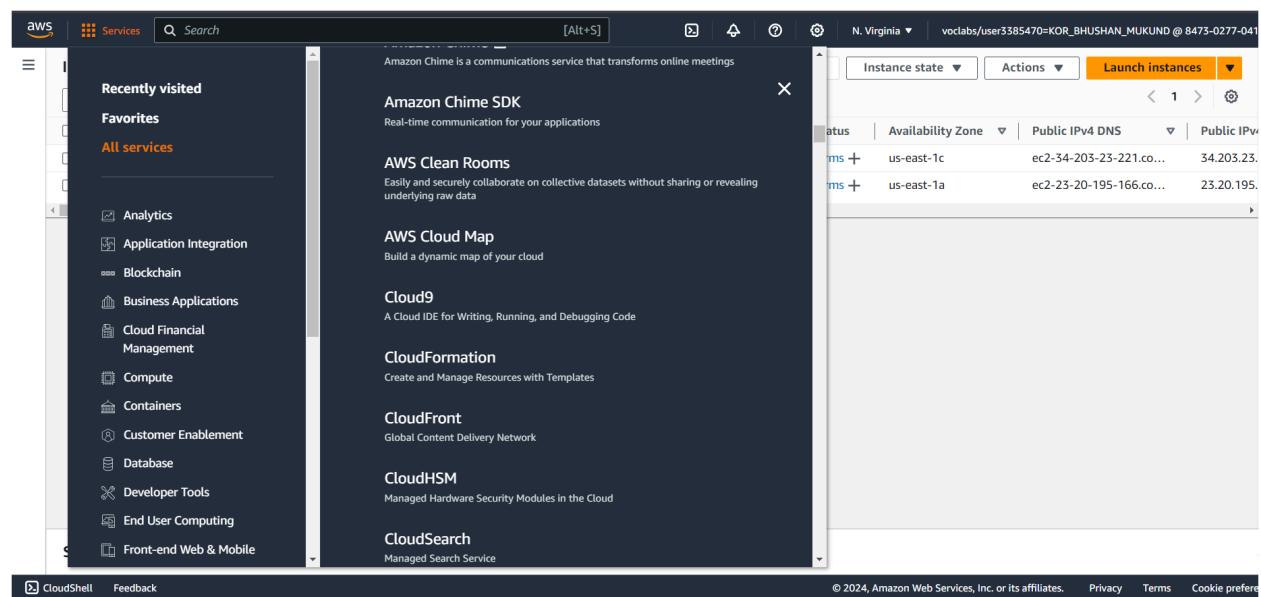
B)Cloud 9 and Cloud 9 IDE

Cloud 9 and Cloud 9 IDE

Cloud 9 is a popular term in English meaning extreme happiness or euphoria. It has no technical association with development tools.

Cloud 9 IDE, on the other hand, is an online integrated development environment (IDE) that allows developers to write, run, and debug code in various programming languages directly in the cloud. It was originally an independent product but is now part of Amazon Web Services (AWS).

Step 1:Select Cloud 9 from services in AWS.



Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 2: Now Click on Create Environment and give name to your environment and select new EC2 instance option.

Create environment [Info](#)

Details

Name
Bhushan_Cloud_9
Limit of 60 characters, alphanumeric, and unique per user.

Description - optional
Cloud 9 for Lab 1 Experiment and learning.
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.

New EC2 instance

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

Step 3: Keep other things by default then select platform as Amazon Linux 2023 or latest and time 30 minutes.

New EC2 instance

Instance type: [Info](#)
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

Additional instance types
Explore additional instances to fit your need.

Platform: [Info](#)
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

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

30 minutes

Network settings: [Info](#)

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

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 4: Select Secure Shell (SSH) and click on Create.

The screenshot shows the 'Network settings' configuration page for AWS Cloud9. Under the 'Connection' section, the 'Secure Shell (SSH)' option is selected, highlighted with a blue border. A note below it states: 'Accesses environment directly via SSH, opens inbound ports.' The 'AWS Systems Manager (SSM)' option is also shown. Below this, there's a 'Tags - optional' section and a note about IAM resource creation:

The following IAM resources will be created in your account

- AWSServiceRoleForAWSCloud9 - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

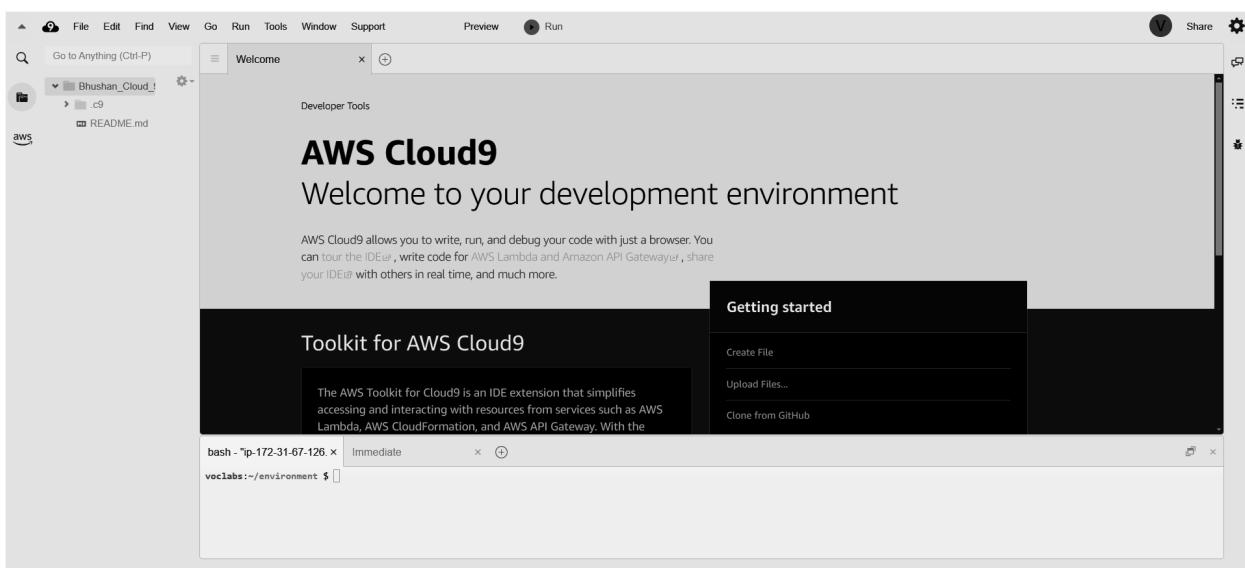
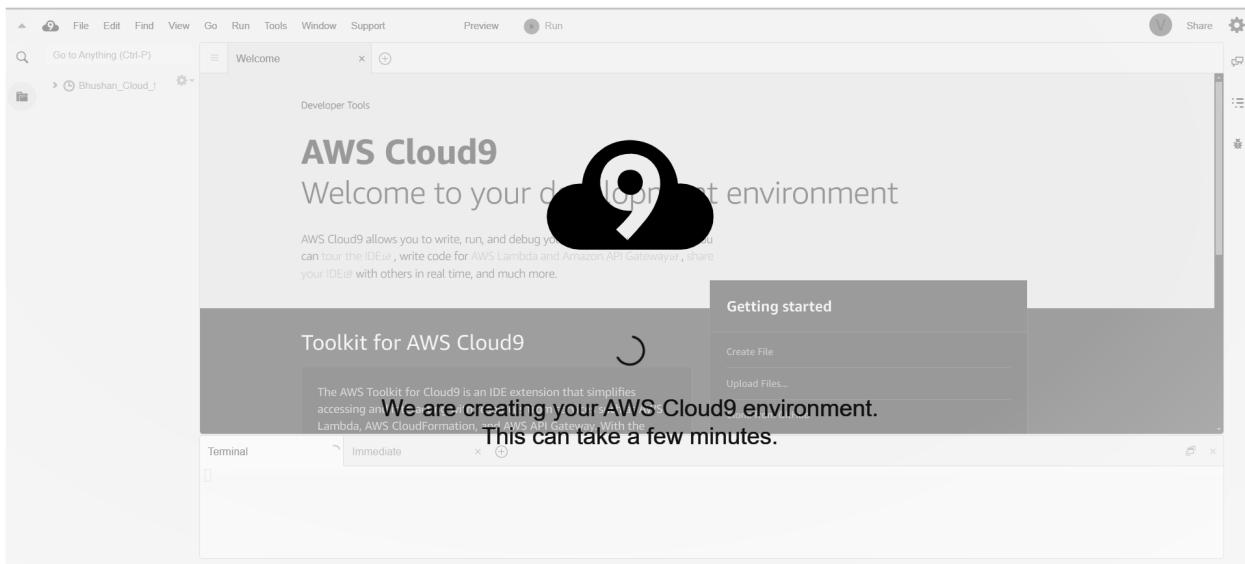
At the bottom right, there are 'Cancel' and 'Create' buttons. The 'Create' button is highlighted with a yellow background.

Step 5: Now It will create the environment after that open it.

The screenshot shows the 'Environments' list page for AWS Cloud9. On the left, there's a sidebar with 'My environments', 'Shared with me', and 'All account environments' options. The main area displays a table of environments:

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Bhushan	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::847302770411:assumed-role/voclabs/user3385470=KOR_BHUSHAN_MUKUND
Bhushan_Cloud_9	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::847302770411:assumed-role/voclabs/user3385470=KOR_BHUSHAN_MUKUND

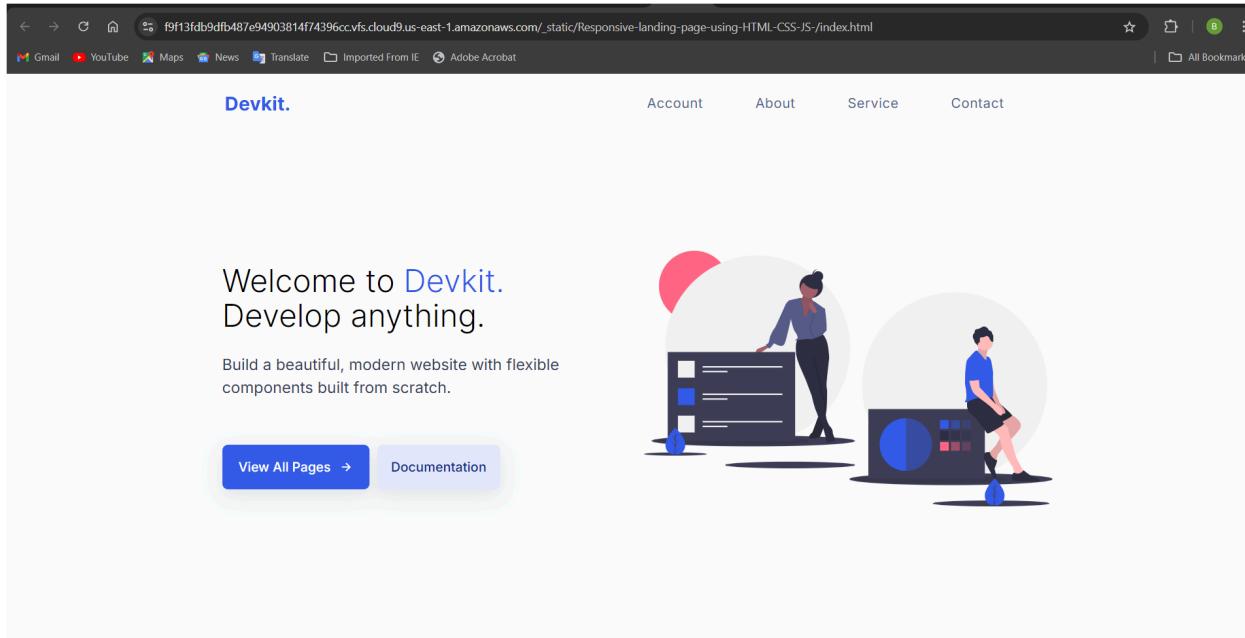
At the top right of the main area, there are buttons for 'Delete', 'View details', 'Open in Cloud9', and 'Create environment'. The 'Create environment' button is highlighted with a yellow background.



Step 6:Add some files or create some files and see the preview of it. Here we have created an index.html file.

The screenshot shows a code editor interface with two tabs open. The top tab is titled 'index.html' and contains the source code for a responsive landing page. The code includes DOCTYPE, meta tags for charset and viewport, a title, and various CSS and JavaScript imports. The bottom tab is titled 'bash - p-172-31-67-126' and shows a terminal window with the command 'voclabs:~/environment \$'. Below the code editor is a terminal window showing the command 'bash - p-172-31-67-126' and the prompt 'voclabs:~/environment \$'. The status bar at the bottom right indicates '27:19 HTML Spaces: 2'.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Responsive Landing Page using HTML, CSS & Javascript</title>
<!-- ===== STYLE.CSS ===== -->
<link rel="stylesheet" href="./css/style.css" />
<!-- ===== REMIXICON CDN ===== -->
<link href="https://cdn.jsdelivr.net/npm/remixicon@2.5.0/fonts/remixicon.css" rel="stylesheet"
/>
<!-- ===== ANIMATE ON SCROLL CSS CDN ===== -->
<link href="https://unpkg.com/aos@2.3.1/dist-aos.css" rel="stylesheet" />
</head>
<body>
<!-- ===== HEADER ===== -->
<header class="header container header">
<div class="header__inner">
<div class="nav">
<div class="logo">
<h2>Devkit.</h2>
</div>
<div class="nav_menu" id="nav_menu">
<button class="close-btn" id="close-btn">
```



Step 7: Now lets share the file with IAM users.

Share this environment

Links to share

Environment: <https://us-east-1.console.aws.amazon.com/cloud9/ide/f9f13fdb9dfb48>

Application: 98.80.97.50

To make your application accessible from the internet, please follow [our documentation](#).

Who has access

▼ ReadWrite

You (online) RW

Don't allow members to save their tab state

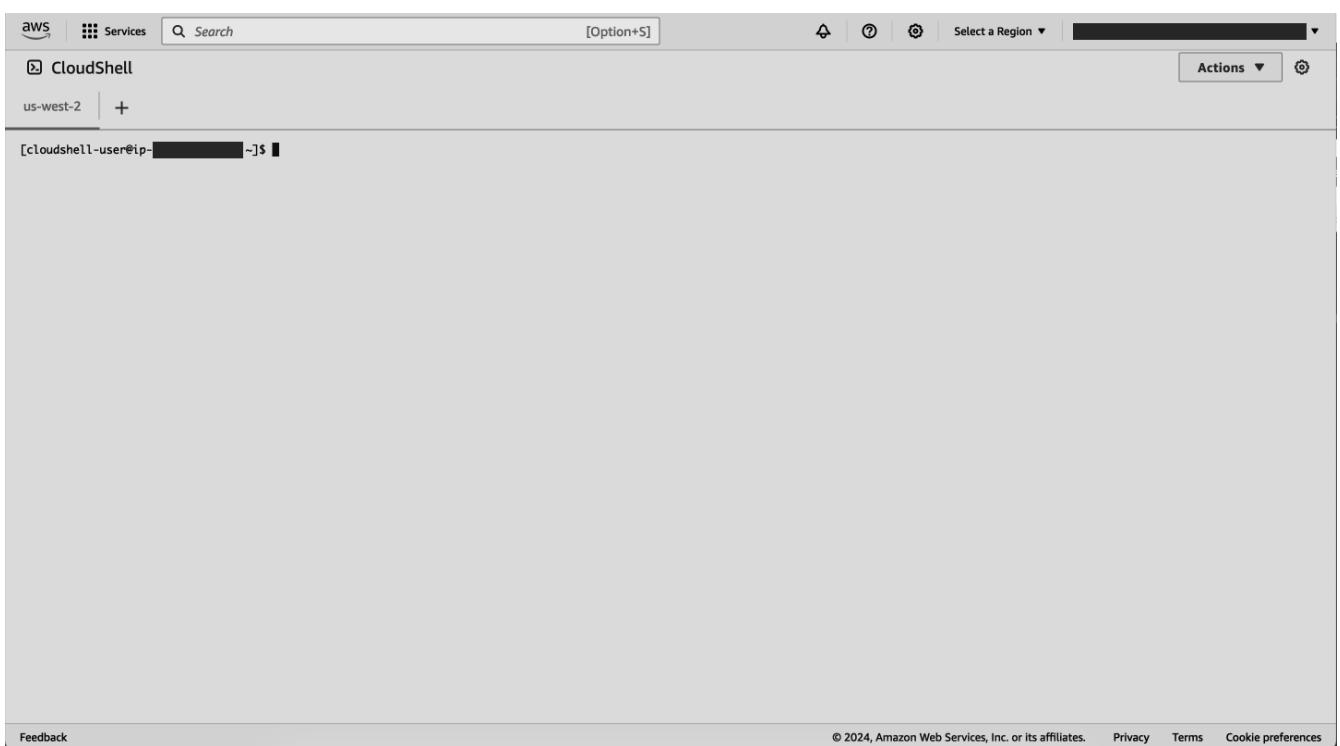
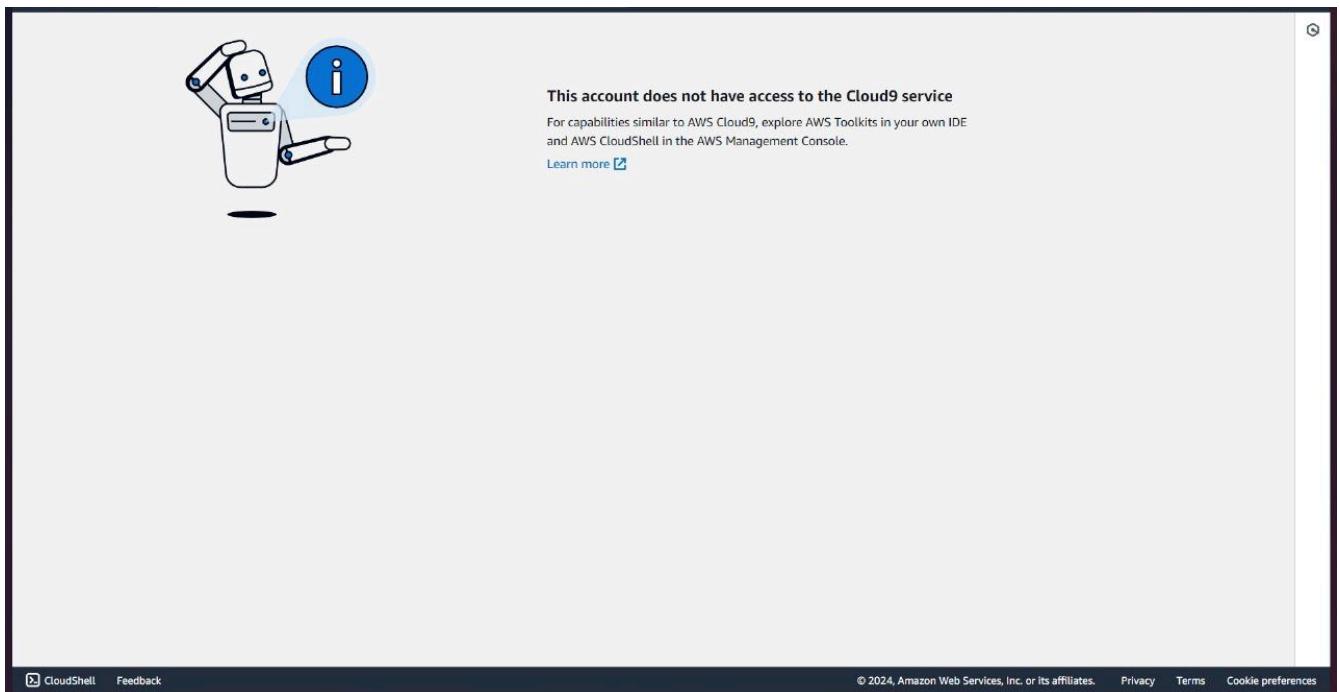
Invite Members

IAM username R RW Invite

Invite an existing IAM user or [create a new user](#).

Done

Step 8: Note That sharing is not possible as we are using AWS academy account in which we do not have access to the IAM policies. And Also 2nd Important thing to note that is AWS has closed Cloud 9 for those who have created their AWS account after 25 Jul 2024. Instead of Cloud 9 AWS has alternative as AWS IDE Toolkits and AWS cloudbshell.



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

Elastic Beanstalk:

Amazon Elastic Beanstalk is a fully managed AWS service that streamlines the deployment and scaling of web applications. Developers simply upload their code, and Elastic Beanstalk automatically handles tasks like resource provisioning, load balancing, auto-scaling, and application health monitoring. It supports multiple programming languages and frameworks, offering flexibility and control over the underlying AWS resources if needed. This enables rapid application deployment without the complexities of managing infrastructure.

CodePipeline:

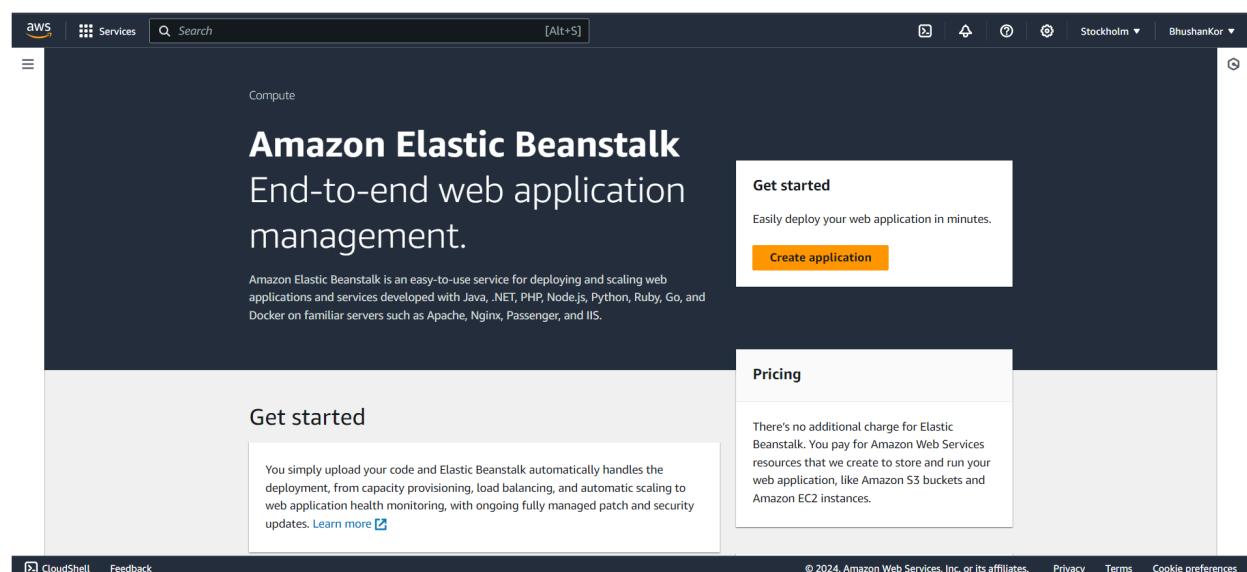
A CodePipeline is an AWS service that automates the build, test, and deploy phases of your release process. It enables continuous integration and continuous delivery (CI/CD) by defining the workflow of code changes from source to production. CodePipeline integrates with various AWS services and third-party tools to streamline and accelerate the release of new features.

Prerequisites: Before you start go to the end of this document and do the steps of New learning because it is necessary for step 5 and codePipeline.

A)Elastic Beanstalk

Step 1:Open Your **personal AWS** Account Because Elastic Beanstalk can be created in AWS Academy but **connecting GitHub with the code pipeline requires an IAM policy that is not present in the AWS Academy account.**

In services search for Elastic Beanstalk and click on it.



Step 2: Click on Create a new application and give a name to your application.

Bhushan_App_4 application is being deleted (2)

Elastic Beanstalk > Create application

Create new application [Info](#)

Application information

Application name Maximum length of 100 characters.

Description

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

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

Step 3: After creation of the application click on Create Environment and select Web Server Environment.

Step 1 Configure environment

Step 2 Configure service access

Step 3 - optional Set up networking, database, and tags

Step 4 - optional Configure instance traffic and scaling

Step 5 - optional Configure updates, monitoring, and logging

Step 6 Review

Configure environment [Info](#)

Environment tier [Info](#)
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name Maximum length of 100 characters.

► Application tags (optional)

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

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

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 4: Select the platform as PHP and application code to sample or if you have you can upload and keep others to default.

The screenshot shows the 'Platform Info' step of the AWS Elastic Beanstalk environment creation wizard. It includes fields for Environment name (BhushanApp4-env), Domain (.us-east-1.elasticbeanstalk.com), and Environment description. Under 'Platform type', 'Managed platform' is selected. The 'Platform' dropdown shows 'PHP'. The 'Platform branch' dropdown shows 'PHP 8.3 running on 64bit Amazon Linux 2023'. The 'Platform version' dropdown shows '4.3.1 (Recommended)'. A 'Check availability' button is also present.

The screenshot shows the 'Application code' step of the wizard. It has three options: 'Sample application' (selected), 'Existing version', and 'Upload your code'. Below this is the 'Presets' section, which includes configuration presets like 'Single instance (free tier eligible)' (selected). At the bottom right are 'Cancel' and 'Next' buttons.

Step 5: Select the use an existing service role option because create and use new service role might not work according to the new policy of AWS (Refer to the New learnings of this document).

Select the existing roles and instances and keys.

Step 6: Review and click on submit.

Screenshot of the AWS CloudFormation Step Functions interface showing the "Step 3: Set up networking, database, and tags" and "Step 4: Configure instance traffic and scaling" steps.

Step 3: Set up networking, database, and tags

Networking, database, and tags [Info](#)
Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

No options configured

Tags

Key	Value
No tags	

There are no tags defined

Step 4: Configure instance traffic and scaling

Instance traffic and scaling [Info](#)
Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

IMDSv1
Deactivated

Capacity

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 9:07 PM 8/11/2024

Screenshot of the AWS CloudFormation Step Functions interface showing the "Step 4: Configure instance traffic and scaling" step.

Step 4: Configure instance traffic and scaling

Instance traffic and scaling [Info](#)
Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

IMDSv1
Deactivated

Capacity

Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360
Processor type	Instance types	AMI ID
x86_64	t3.micro,t3.small	ami-083f545ce1a73bf03
Availability Zones	Metric	Statistic
Any	NetworkOut	Average
Unit	Period	Breach duration
Bytes	5	5
Upper threshold	Scale up increment	Lower threshold
6000000	1	2000000

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

Screenshot of the AWS CloudFormation Step Functions interface showing the "Step 5: Configure updates, monitoring, and logging" step.

Step 5: Configure updates, monitoring, and logging

Updates, monitoring, and logging [Info](#)
Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—

Log streaming

Retention	Lifecycle
7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok

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

The screenshot shows the AWS Elastic Beanstalk environment configuration page. It displays several configuration sections:

- Health**: Instance replacement is set to "Ignore health check" (false) and "AllAtOnce".
- Platform software**: Includes settings for Lifecycle (false), Log streaming (Deactivated), Allow URL fopen (On), Display errors (Off), Document root (-), Max execution time (60), Memory limit (256M), Zlib output compression (Off), Proxy server (nginx), Logs retention (7), Rotate logs (Deactivated), Update level (minor), and X-Ray enabled (Deactivated).
- Environment properties**: A table showing "No environment properties".

At the bottom, there are "Cancel", "Previous", and "Submit" buttons.

Step 7: Done Elastic BeanStalk environment is created successfully, click on the link to view the preview of your sample code or uploaded code.

The screenshot shows the AWS Elastic Beanstalk environment overview page for "BhushanApp4-env". The top bar indicates "Environment successfully launched".

Environment overview section:

- Health**: Shows a warning icon.
- Platform**: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1.
- Events** (11):
 - August 11, 2024 21:11:41 (UTC+5:30): INFO - Successfully launched environment: BhushanApp4-env
 - August 11, 2024 21:11:38 (UTC+5:30): INFO - Added instance [i-049e1dd5eb5428d26] to your environment.
 - August 11, 2024 21:11:38 (UTC+5:30): WARN - Environment health has transitioned from Pending to Warning. Initialization completed 16 seconds ago and took 2 minutes. Unable to assume role "arn:aws:iam::010928205712:role/Bhushan_EC2_new". Verify that the role exists and is configured correctly.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

The screenshot shows a web browser window with the following details:

- Address Bar:** bhushanapp4-env.eba-2mcvqpx.us-east-1.elasticbeanstalk.com
- Page Content (Left Side):**
 - # Congratulations!
 - Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud.
 - You are running PHP version 8.3.7
 - This environment is launched with Elastic Beanstalk PHP Platform
- Page Content (Right Side):**
 - What's Next?**
 - AWS Elastic Beanstalk overview
 - Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
 - Using Amazon RDS with PHP
 - Customizing the Software on EC2 Instances
 - Customizing Environment Resources
 - AWS SDK for PHP**
 - AWS SDK for PHP home
 - PHP developer center
 - AWS SDK for PHP on GitHub

B)CodePipeline**Step 1:**Search for codePipeline in services and click on create pipeline.

The screenshot shows the AWS CodePipeline landing page. At the top, there's a search bar and a 'Create pipeline' button. Below the header, a main section titled 'AWS CodePipeline' with the subtitle 'Visualize and automate the different stages of your software release process'. It includes a brief description of what CodePipeline does and a 'Create pipeline' button. To the right, there's a 'Pricing (US)' section showing a cost of \$1/month per active pipeline, with a note that pipelines are free for the first 30 days. Below that is a 'Getting started' section with links to 'What is AWS CodePipeline?', 'Getting started with AWS CodePipeline?', 'Working with AWS CodePipeline?', and 'Documentation'. A large central image illustrates the concept of a CI/CD pipeline with multiple computer monitors showing code and yellow stars indicating successful stages.

The screenshot shows the 'Pipelines' list page within the AWS CodePipeline service. The left sidebar has a navigation menu with options like Source, Artifacts, Build, Deploy, Pipeline, Pipelines, Settings, Go to resource, and Feedback. The main area shows a table with columns for Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A message at the top states 'Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more'. Below the table, there's a note saying 'No results' and 'There are no results to display.' At the top of the main area, there are buttons for Create pipeline, Notify, View history, Release change, and Delete pipeline.

Step 2: Give the name to the pipeline, select a new service role, and keep the rest all to default.

Choose pipeline settings [Alt+S]

Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
Bhushan_PipeLines
No more than 100 characters

Pipeline type
You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.
 Superseded
A more recent execution can overtake an older one. This is the default.
 Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.
 Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

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

Role name
AWSCodePipelineServiceRole-us-east-1-Bhushan_PipeLines

Type your service role name
 Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Execution mode
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

Role name
AWSCodePipelineServiceRole-us-east-1-Bhushan_PipeLine

Type your service role name
 Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Variables
You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

Add variable
You can add up to 50 variables.

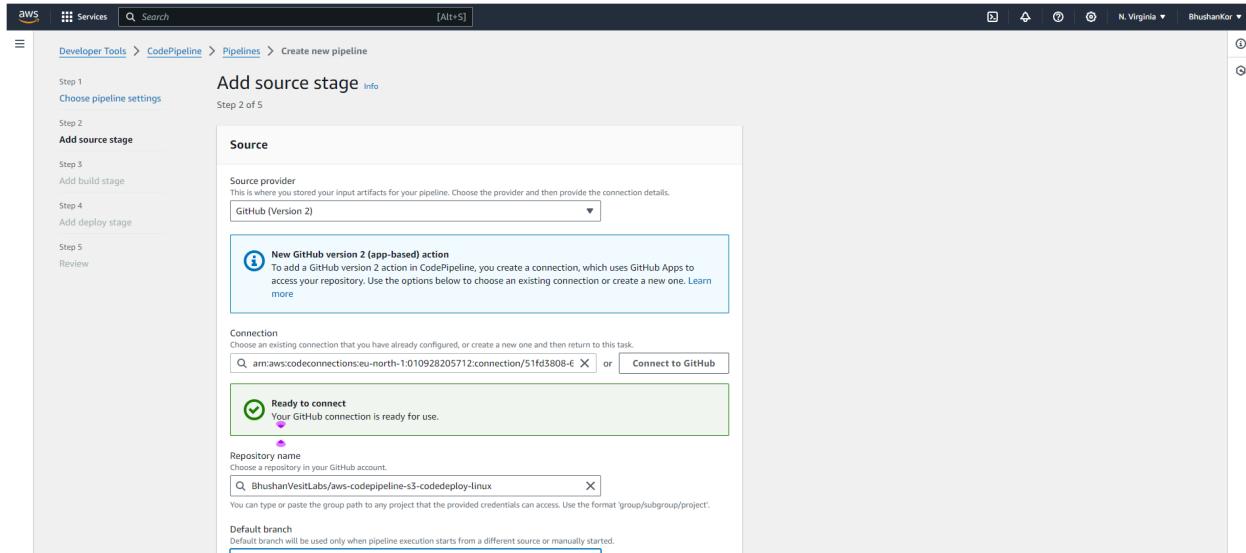
The first pipeline execution will fail if variables have no default values.

Advanced settings

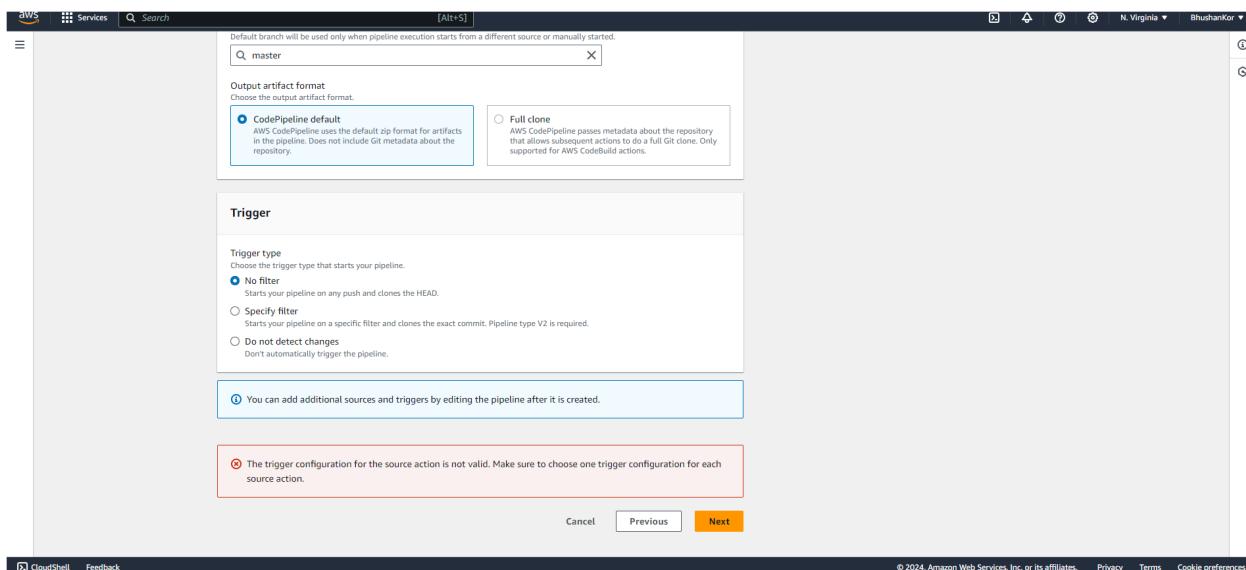
Cancel Next Step

Before Step 3 Fork or clone the below repository to your GitHub account.
<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux.git>

Step 3: Now in the Add source stage Select GitHub (Version 2) [This will work only in personal AWS academy], then connect your GitHub account and select repository name, branch to main/master.



Step 4: Select Trigger to No filter otherwise it will give an error.



Step 5: In the Deploy stage select AWS ElasticBeanstalk as the deploy provider, Select a region to US East (N.Virginia) or any with N.Virginia. Input artifacts to default and the Application name and environment name of your Elastic Beanstalk.

The screenshot shows the AWS CodePipeline interface at Step 4 of 5, titled 'Add deploy stage'. A prominent error message box states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' Below this, the 'Deploy' section is visible, containing fields for 'Deploy provider' (set to 'AWS Elastic Beanstalk'), 'Region' (set to 'US East (N. Virginia)'), 'Input artifacts' (set to 'SourceArtifact'), 'Application name' (set to 'Bhushan_App_4'), and 'Environment name' (set to 'BhushanApp4-env').

This screenshot is identical to the one above, but it includes a checked checkbox labeled 'Configure automatic rollback on stage failure' located at the bottom of the 'Deploy' configuration section. The rest of the pipeline setup remains the same, including the error message about having at least two stages.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

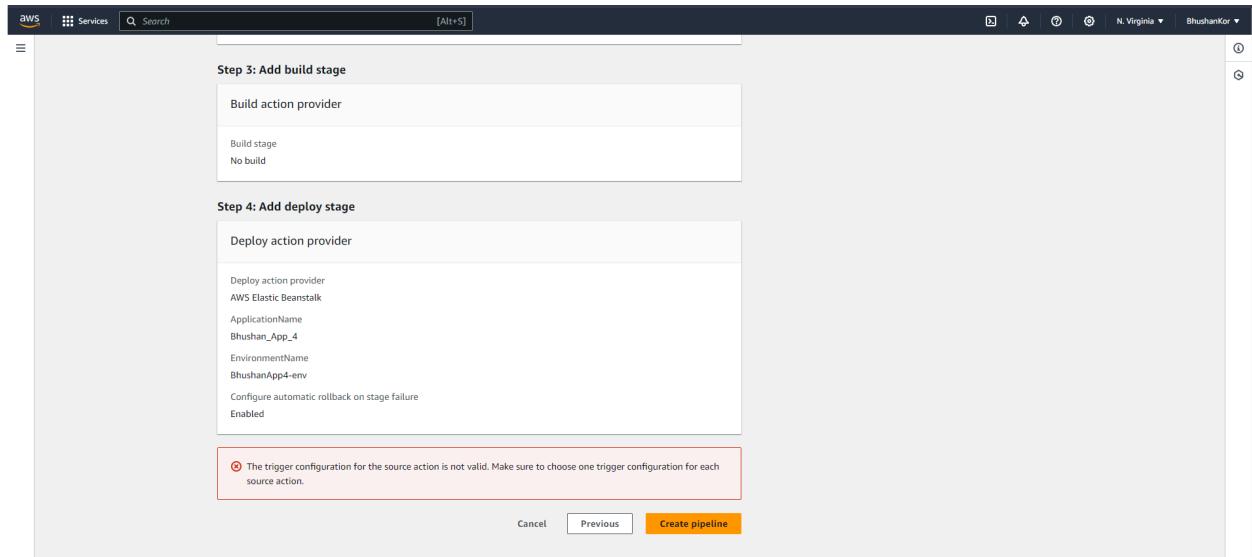
Division: D15C

Roll No: 28

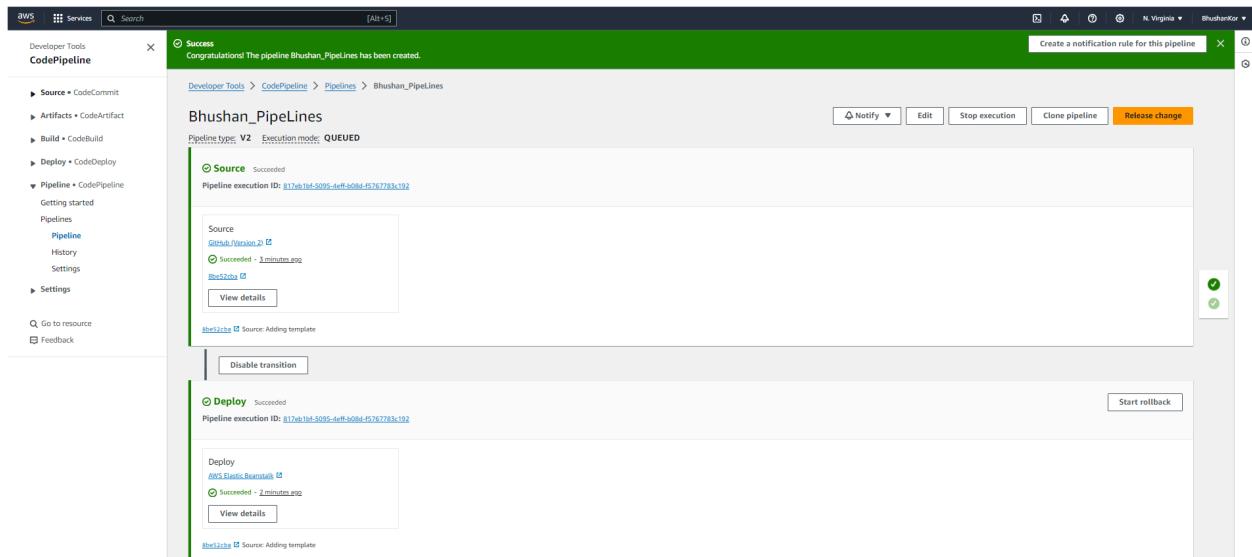
Step 6: Review and Click on Create Pipeline.

The screenshot shows the 'Review' step of creating a new pipeline. The pipeline name is set to 'Bhushan_PipeLines'. The pipeline type is 'V2' and the execution mode is 'QUEUED'. The artifact location is 'codepipeline-us-east-1-934567252759' and the service role name is 'AWSCodePipelineServiceRole-us-east-1-Bhushan_PipeLines'. There are no variables defined at the pipeline level. The sidebar on the left lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review).

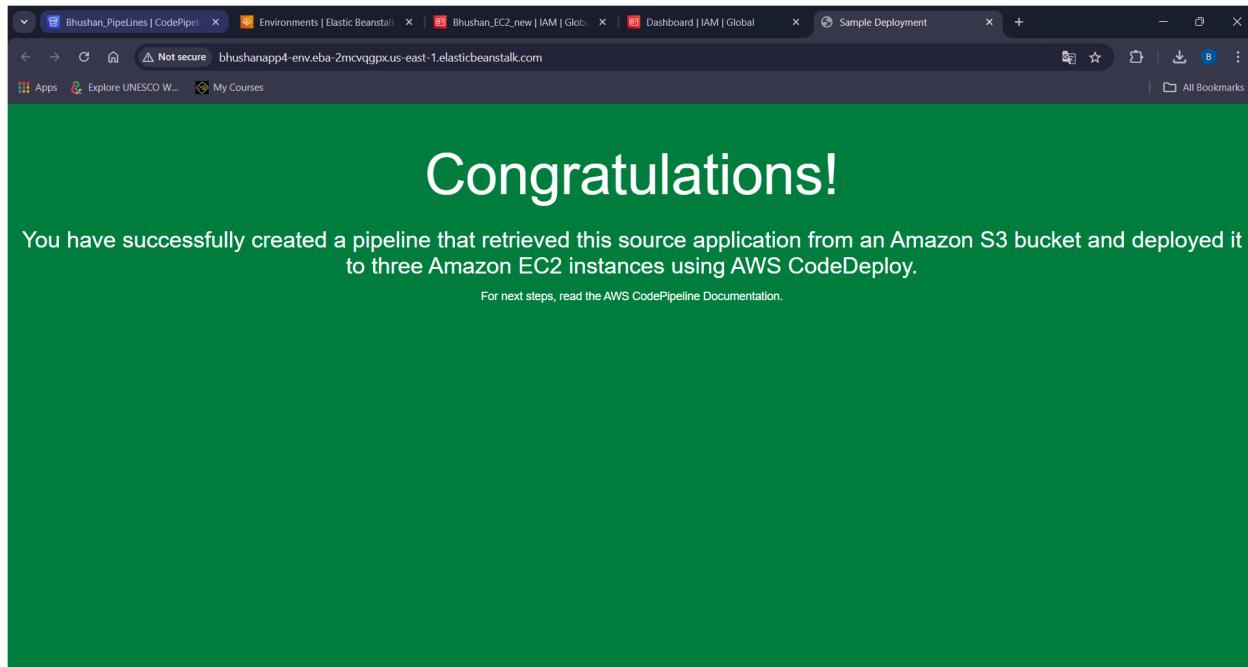
The screenshot shows the 'Step 2: Add source stage' interface. Under 'Source action provider', it lists GitHub (Version 2) as the selected provider. Other options include OutputArtifactFormat (CODE_ZIP), DetectChanges (true), ConnectionArn (am:aws:codeconnections:eu-north-1:010928205712:connection/51fd3808-6436-478b-b88b-030b391fb258), FullRepositoryId (BhushanVestLabs/aws-codepipeline-s3-codedeploy-linux), and Default branch (master). Under 'Trigger configuration', it says 'You can add additional pipeline triggers after the pipeline is created.' and shows 'Trigger type' as 'No filter'. The sidebar on the left lists Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review).



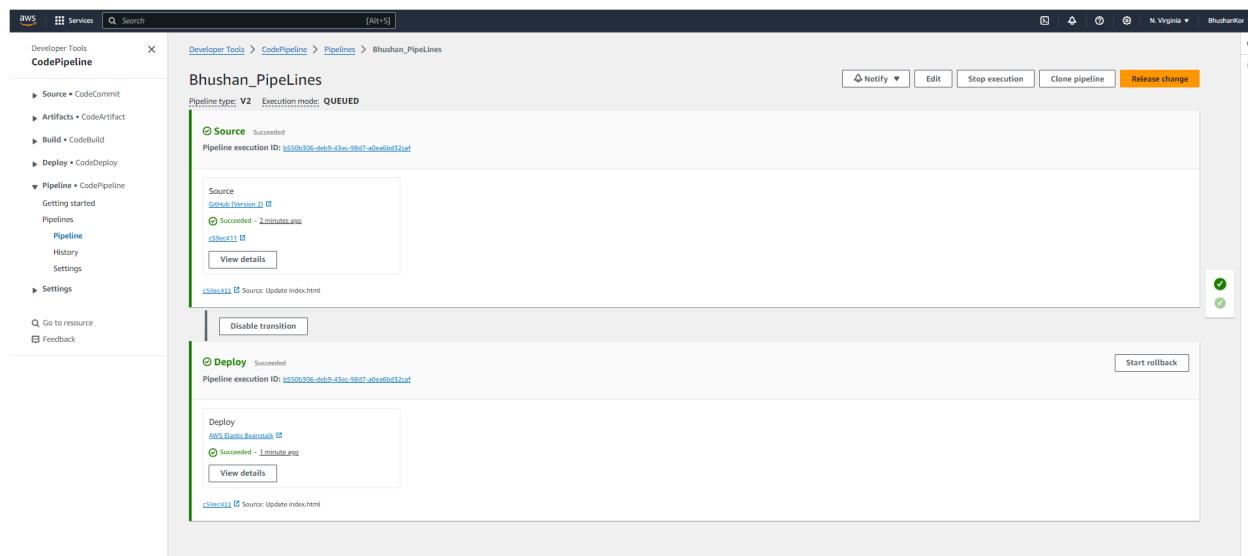
Step 7: Your Pipeline is Created Successfully.Also it is successfully connected to the source and successfully Deployed.



Step 8: Click on the given link and you will see the Result.



Step 9: Make Some changes in the Source code file on GitHub and commit changes and then deploy the changes in pipeline.



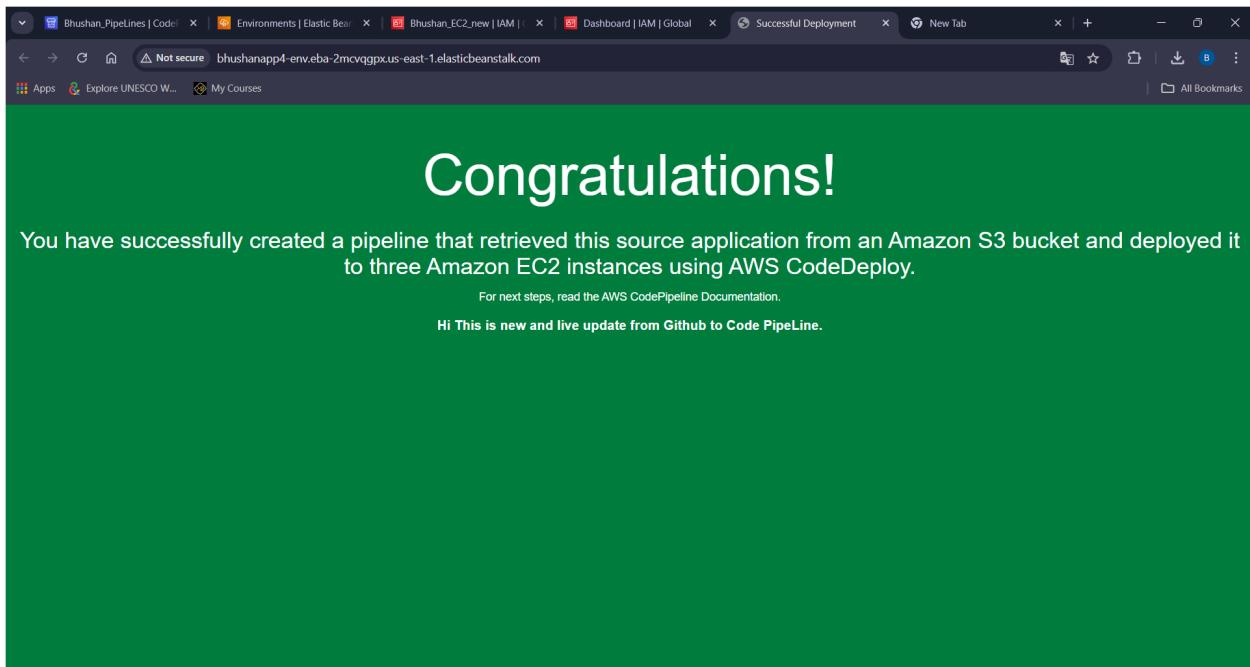
Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

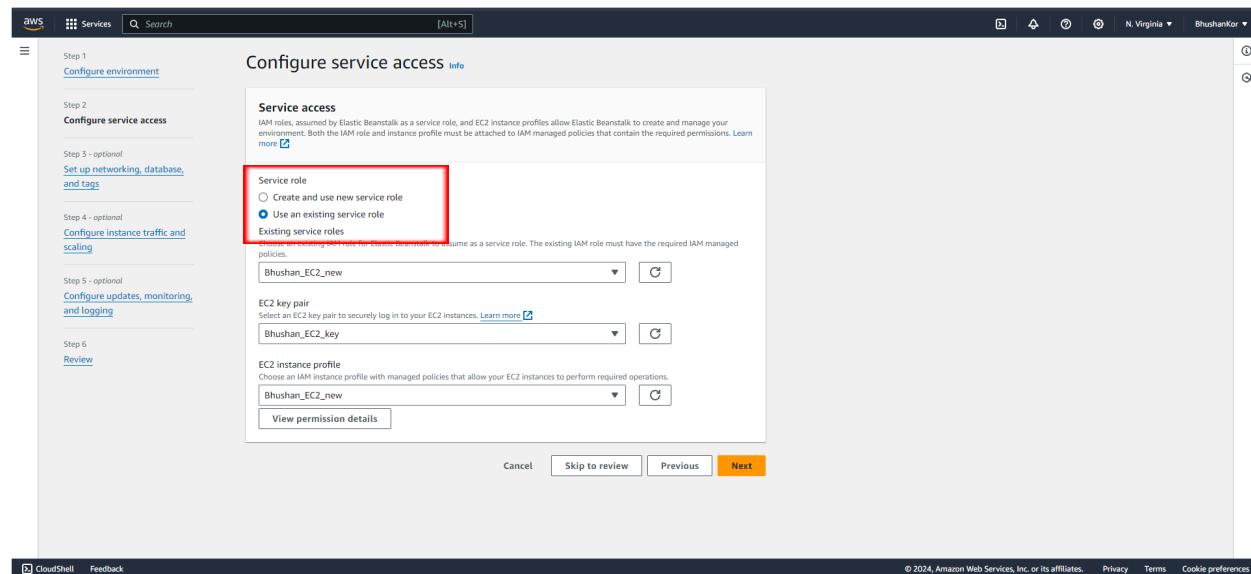
Step 10: Name of website is changed also one we line is added at the end.



New Learnings in this experiment:**Note:**

Previously Elastic Beanstalk created a default EC2 instance profile named `aws-elasticbeanstalk-ec2-role` the first time an AWS account created an environment. This instance profile included default managed policies. If your account already has this instance profile, it will remain available for you to assign to your environments.

However, recent AWS security guidelines don't allow an AWS service to automatically create roles with trust policies to other AWS services, EC2 in this case. Because of these security guidelines, Elastic Beanstalk no longer creates a default `aws-elasticbeanstalk-ec2-role` instance profile.



Because of the above problem, we have chosen the existing role and for that, we have to create a new role with certain policies.

To create an instance profile

1. Open the Roles page in the IAM console.
2. Choose Create role.
3. Under the Trusted entity type, choose AWS service.
4. Under Use case, choose EC2.
5. Choose Next.
6. Attach the appropriate managed policies provided by Elastic Beanstalk and any additional policies that provide permissions that your application needs.
7. Choose Next.
8. Enter a name for the role.
9. (Optional) Add tags to the role.
10. Choose Create role.

For Point 6 Do the below Steps**To add managed policies to the role attached to the default instance profile**

1. Type `AWSElasticBeanstalk` to filter the policies.
2. Select the following policies, and then choose Attach policy:
 - `AWSElasticBeanstalkWebTier`
 - `AWSElasticBeanstalkWorkerTier`
 - `AWSElasticBeanstalkMulticontainerDocker`
3. Also, add `AmazonS3FullAccess` or `AmazonDynamoDBFullAccess`.
4. Choose Attach policy.

Add the Trust relationship policy for EC2

To allow the EC2 instances in your environment to assume the required role, the instance profile must specify Amazon EC2 as a trusted entity in the trust relationship policy, as follows.

```
{  
  "Version": "2008-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {
```

```

        "Service": "ec2.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
]
}

```

To customize permissions, you can add policies to the role attached to the default instance profile or create your own instance profile with a restricted set of permissions.

Screenshot for reference

Policies:

The screenshot shows the AWS IAM 'Permissions' tab. Under the 'Permissions policies' section, there are 10 managed policies listed:

Policy name	Type	Attached entities
AdministratorAccess-AWSElasticBeanstalk	AWS managed	1
AmazonS3FullAccess	AWS managed	1
AWSCloud9EnvironmentMember	AWS managed	1
AWSCloud9SSMInstanceProfile	AWS managed	2
AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	1
AWSElasticBeanstalkEnhancedHealth	AWS managed	1
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	1
AWSElasticBeanstalkMulticontainerDocker	AWS managed	1
AWSElasticBeanstalkWebTier	AWS managed	1
AWSElasticBeanstalkWorkerTier	AWS managed	1

Below the table, there is a note: '▶ Permissions boundary (not set)'

Trust Relationship:

The screenshot shows the AWS IAM 'Trust relationships' tab. Under the 'Trusted entities' section, it says: 'Entities that can assume this role under specified conditions.' A 'Edit trust policy' button is available.

```

1 * []
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "ec2.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10      }
11    ]
12  []

```

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

IAM Role:

The screenshot shows the AWS IAM Roles page. The URL is IAM > Roles > Bhushan_EC2_new. The role name is Bhushan_EC2_new. A brief description states: "Allows EC2 instances to call AWS services on your behalf." There are two tabs: "Summary" and "Edit". Under "Summary", there are three columns: "Creation date" (August 11, 2024, 21:00 (UTC+05:30)), "ARN" (arn:aws:iam::010928205712:role/Bhushan_EC2_new), and "Instance profile ARN" (arn:aws:iam::010928205712:instance-profile/Bhushan_EC2_new). The "Last activity" section shows "3 days ago". The "Maximum session duration" is set to "1 hour".

EC2 Instance with attached IAM Role:

The screenshot shows the AWS EC2 Instances page. The title bar says "Successfully attached Bhushan_EC2_new to instance i-0e721f56644c4f090". The main table shows one instance: Bhushan_EC2 (i-0e721f56644c4f090). The instance is stopped, t2.micro type, us-east-1c availability zone, and has a Public IPv4 DNS of 172.31.84.117. On the left sidebar, under "Instances", "Details" is selected. The "Details" tab for the instance shows the attached IAM role: Bhushan_EC2_new.

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

Theory:

Container-based microservices architectures have revolutionized how development and operations teams test and deploy modern software. Containers allow companies to scale and deploy applications more efficiently, but they also introduce new challenges, adding complexity by creating a whole new infrastructure ecosystem.

Today, both large and small software companies are deploying thousands of container instances daily. Managing this level of complexity at scale requires advanced tools. Enter Kubernetes.

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their applications while Kubernetes takes care of key tasks, including:

- Managing resource consumption by applications or teams
- Distributing application load evenly across the infrastructure
- Automatically load balancing requests across multiple instances of an application
- Monitoring resource usage to prevent applications from exceeding resource limits and automatically restarting them if needed
- Moving application instances between hosts when resources are low or if a host fails
- Automatically utilizing additional resources when new hosts are added to the cluster
- Facilitating canary deployments and rollbacks with ease

Necessary Requirements:

- **EC2 Instance:** The experiment required launching a t2.medium EC2 instance with 2 CPUs, as Kubernetes demands sufficient resources for effective functioning.

- **Minimum Requirements:**

- **Instance Type:** t2.medium
- **CPUs:** 2
- **Memory:** Adequate for container orchestration.

This ensured that the Kubernetes cluster had the necessary resources to function smoothly

Note:

AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account.). If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Prerequisites :

Create 2 Security Groups for Master and Nodes and add the following rules inbound rules in those Groups.

Master:

Inbound rules Info						
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-017c1a22a7cb5e5	HTTP	TCP	80	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-0d3f86194443b29f1	All traffic	All	All	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-010d128b1484ff322	Custom TCP	TCP	6443	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-05bb413f0626b9c3b	Custom TCP	TCP	10251	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-04bd098c8f409420d	Custom TCP	TCP	10250	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-01438a40425cf867c	All TCP	TCP	0 - 65535	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-05dc20e8c2b541402	Custom TCP	TCP	10252	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-08d45afafe6c06c26	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 X	Delete

Node :

Edit inbound rules Info						
Inbound rules Info						
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-00d83454961e5d3e9	All traffic	All	All	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-0402e9e84cd3dea45	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-05770af1e4c56697f	Custom TCP	TCP	10250	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-0b3fb7516970bc90	All TCP	TCP	0 - 65535	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-07384bc31bec899e9	Custom TCP	TCP	30000 - 32767	Custom	<input type="text"/> 0.0.0.0/0 X	Delete
sgr-05188e46d7e21828d	HTTP	TCP	80	Custom	<input type="text"/> 0.0.0.0/0 X	Delete

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances.

Select Ubuntu as AMI and **t2.medium** as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.We can use 3 Different keys or 1 common key also.

Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

Also Select Security groups from existing.

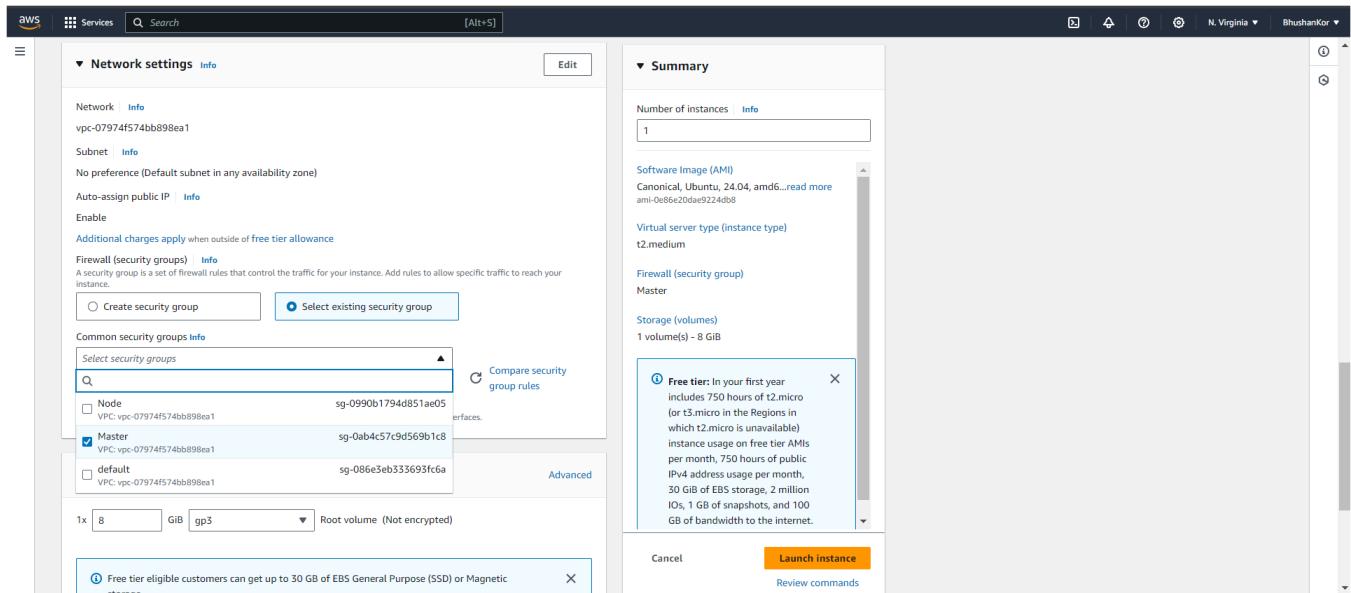
Master:

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28



Do Same for 2 Nodes and use security groups of Node for that.

Step 2: After creating the instances click on Connect & connect all 3 instances and navigate to SSH Client.

Instances (3) Info											
Find Instance by attribute or tag (case-sensitive)											
Last updated less than a minute ago											
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	Actions ▾
<input type="checkbox"/>	Master	i-0cfb6b3b53bc03ab1	Running View Logs	t2.medium	2/2 checks passed View alarms +	View alarms +	us-east-1d	ec2-52-90-3-215.comp...	52.90.3.215	-	Launch instances ▾
<input type="checkbox"/>	Node 1	i-0b64c605d31b0bd44	Running View Logs	t2.medium	0/2 initializing View alarms +	View alarms +	us-east-1d	ec2-3-80-56-65.comput...	3.80.56.65	-	Launch instances ▾
<input type="checkbox"/>	Node 2	i-0af54010ae84808d2	Running View Logs	t2.medium	0/2 initializing View alarms +	View alarms +	us-east-1d	ec2-34-224-169-38.co...	34.224.169.38	-	Launch instances ▾

(Downloaded Key

C:\ Bhushan - Personal > Desktop > New folder (4)					
Sort ▾ View ▾ ...					
Name	Status	Date modified	Type	Size	
Master_Ec2_Key.pem	View Logs	9/14/2024 4:32 PM	PEM File	2 KB	

)

Step 3: Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)

Master:

EC2 > Instances > i-0cfb6b3b53bc03ab1 > Connect to instance

Connect to instance Info

Connect to your instance i-0cfb6b3b53bc03ab1 (Master) using any of these options

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

Instance ID
i-0cfb6b3b53bc03ab1 (Master)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Master_Ec2_Key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Master_Ec2_Key.pem"
4. Connect to your instance using its Public DNS:
 ec2-52-90-3-215.compute-1.amazonaws.com

Example:
 ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-52-90-3-215.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

Node 1:

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

Instance ID
i-0b64c605d31b0bd44 (Node 1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Node1.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Node1.pem"
4. Connect to your instance using its Public DNS:
 ec2-3-80-56-65.compute-1.amazonaws.com

Example:
 ssh -i "Node1.pem" ubuntu@ec2-3-80-56-65.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.

Node 2:

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

Instance ID
 [i-0af54010ae84808d2 \(Node 2\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is [Node1.pem](#)
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "Node1.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-34-224-169-38.compute-1.amazonaws.com`

Example:
 `ssh -i "Node1.pem" ubuntu@ec2-34-224-169-38.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.

Here I have use 2 keys 1 for master and 1 for 2 node so I have to run open 3 terminals.In master key folder 1 terminal and 2 terminals in node 1 key folder.

If you use 1 Key only, you can open 3 terminal in one folder only.

Successful Connection:

```
ubuntu@ip-172-31-27-176:~ % + ^
E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-90-3-215.compute-1.amazonaws.com' (ED2551
9) to the list of known hosts.
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 15:13:30 UTC 2024

System load: 0.08 Processes: 115
Usage of /: 22.9% of 6.71GB Users logged in: 0
Memory usage: 5% IPv4 address for enX0: 172.31.27.176
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-172-31-28-117:~$ |
```

```
ubuntu@ip-172-31-28-117:~ % + ^
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-172-31-28-117:~$ |
```

```
ubuntu@ip-172-31-18-135:~ % + ^
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-172-31-18-135:~$ |
```

Step 4: Run on Master,Node 1, and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee  
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-27-176:~$  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/  
trusted.gpg.d/docker.gpg > /dev/null  
  
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/  
ubuntu $(lsb_release -cs) stable"  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead  
(see apt-key(8)).  
OK  
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_docke  
r_com_linux_ubuntu-noble.list  
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_d  
ownload_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 http://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 InRelea  
se [126 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Pa  
ckages [15.0 MB]  
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [  
13.8 kB]  
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [  
354 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [  
79.4 kB]
```

```
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:51 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:52 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.9 MB in 4s (6976 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 h
as an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
ubuntu@ip-172-31-27-176:~$ |
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-27-176:~$ sudo apt-get update
sudo apt-get install -y docker-ce
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 InRelea
se
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security 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 h
as an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
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 cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  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 pi
gz amd64 2.8-1 [65.6 kB]
```

```
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 → /us
r/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
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-172-31-27-176:~$ |
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
```

```
ubuntu@ip-172-31-27-176:~$ 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-172-31-27-176:~$
```

```
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-27-176:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart 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-172-31-27-176:~$
```

Step 5: Run the below command to install Kubernets.

```
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
```

```
ubuntu@ip-172-31-27-176:~$ 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
```

```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-27-176:~$
```

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-27-176:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
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 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security 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 6051 B in 1s (11.1 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.
Reading package lists... Done
Building dependency tree... Done
```

```
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.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-27-176:~$ |
```

```
sudo systemctl enable --now kubelet
```

```
sudo apt-get install -y containerd
```

```
-----  
ubuntu@ip-172-31-27-176:~$ sudo systemctl enable --now kubelet  
ubuntu@ip-172-31-27-176:~$ 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.
```

```
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.
```

```
sudo mkdir -p /etc/containerd
```

```
sudo containerd config default | sudo tee /etc/containerd/config.toml
```

```
ubuntu@ip-172-31-27-176:~$ sudo mkdir -p /etc/containerd
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
```

```
[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"
```

```
[ttrpc]
  address = ""
  gid = 0
  uid = 0
```

```
sudo systemctl restart containerd
```

```
sudo systemctl enable containerd
```

```
sudo systemctl status containerd
```

```
ubuntu@ip-172-31-27-176:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
    Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; p>
      Active: active (running) since Mon 2024-09-16 15:31:58 UTC; 210ms ago
        Docs: https://containerd.io
   Main PID: 4763 (containerd)
     Tasks: 7
    Memory: 13.9M (peak: 14.4M)
      CPU: 50ms
     CGroup: /system.slice/containerd.service
             └─4763 /usr/bin/containerd

Sep 16 15:31:58 ip-172-31-27-176 containerd[4763]: time="2024-09-16T15:31:5>
Sep 16 15:31:58 ip-172-31-27-176 systemd[1]: Started containerd.service - c>
```

sudo apt-get install -y socat

```
ubuntu@ip-172-31-27-176:~$ sudo apt-get install -y socat
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 NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 133 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat
  amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (11.2 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
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-172-31-27-176:~$ |
```

Step 6: Initialize the Kubecluster .Now Perform this Command only for Master.

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

```
ubuntu@ip-172-31-27-176:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[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'
W0916 15:39:33.685919      5313 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-27-176 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.27.176]
[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-172-31-27-176 localhost] and IPs [172.31.27.176 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-27-176 localhost] and IPs [172.31.27.176 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001059178s
```

```
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 6.500965245s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-27-176 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-27-176 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: ttay2x.n0squeukjai8sgfg3
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csraprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

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

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

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

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

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

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

```
kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acf9da9184ecc028d77157608fb6
ubuntu@ip-172-31-27-176:~$
```

Run this command on master and also copy and save the Join command from above.

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

```
ubuntu@ip-172-31-27-176:~$ mkdir -p $HOME/.kube
    sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
    sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-27-176:~$
```

Step 7: Now Run the command kubectl get nodes to see the nodes before executing Join command on nodes.

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-27-176	NotReady	control-plane	5m38s	v1.31.1

Step 8: Now Run the following command on Node 1 and Node 2 to Join to master.

```
sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash
sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6
```

Node 1:

```
ubuntu@ip-172-31-28-117:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.396793ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

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

Node 2:

```
ubuntu@ip-172-31-18-135:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001003808s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

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

Step 9: Now Run the command kubectl get nodes to see the nodes after executing Join command on nodes.

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

Step 10: Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

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

sudo systemctl status kubelet

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

```
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497458 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497516 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497569 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497620 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497669 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497719 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume s>
Sep 16 15:51:31 ip-172-31-27-176 kubelet[5989]: E0916 15:51:31.605091 5989 kubelet.go:2982] "Container runtime network not ready" networkReady="NetworkR>
Sep 16 15:51:32 ip-172-31-27-176 kubelet[5989]: I0916 15:51:32.366237 5989 scope.go:1117] "RemoveContainer" containerID="f44f06967c5b3e567e07841a7b4352ae>
Sep 16 15:51:36 ip-172-31-27-176 kubelet[5989]: E0916 15:51:36.606675 5989 kubelet.go:2982] "Container runtime network not ready" networkReady="NetworkR>
Sep 16 15:51:41 ip-172-31-27-176 kubelet[5989]: E0916 15:51:41.608404 5989 kubelet.go:2982] "Container runtime network not ready" networkReady="NetworkR>
```

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

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

Now to Rename run this command

```
kubectl label node ip-172-31-18-135 kubernetes.io/role=worker
```

Rename to Node 1: kubectl label node ip-172-31-28-117 kubernetes.io/role=Node1

Rename to Node 2: kubectl label node ip-172-31-18-135 kubernetes.io/role=Node2

```
ubuntu@ip-172-31-27-176:~$ kubectl label node ip-172-31-28-117 kubernetes.io/role=Node1
node/ip-172-31-28-117 labeled
ubuntu@ip-172-31-27-176:~$ kubectl label node ip-172-31-18-135 kubernetes.io/role=Node2
node/ip-172-31-18-135 labeled
```

Step 11: Run command kubectl get nodes -o wide . And Hence we can see we have Successfully connected Node 1 and Node 2 to the Master.

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
ip-172-31-18-135	Ready	Node2	12m	v1.31.1	172.31.18.135	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12
ip-172-31-27-176	Ready	control-plane	21m	v1.31.1	172.31.27.176	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12
ip-172-31-28-117	Ready	Node1	13m	v1.31.1	172.31.28.117	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12

Or run kubectl get nodes

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

Conclusion: In this experiment, we successfully set up a Kubernetes cluster with one master and two worker nodes on AWS EC2 instances. After installing Docker, Kubernetes tools (kubelet, kubeadm, kubectl), and containerd on all nodes, the master node was initialized and the worker nodes were joined to the cluster. Initially, the nodes were in the NotReady state, which was resolved by installing the Calico network plugin. We also labeled the nodes with appropriate roles (control-plane and worker). The cluster became fully functional with all nodes in the Ready state, demonstrating the successful configuration and orchestration of Kubernetes.

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

Theory:

Kubernetes, originally developed by Google, is an open-source container orchestration platform. It automates the deployment, scaling, and management of containerized applications, ensuring high availability and fault tolerance. Kubernetes is now the industry standard for container orchestration and is governed by the **Cloud Native Computing Foundation (CNCF)**, with contributions from major cloud and software providers like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes Deployment: Is a resource in Kubernetes that provides declarative updates for Pods and ReplicaSets. With a Deployment, you can define how many replicas of a pod should run, roll out new versions of an application, and roll back to previous versions if necessary. It ensures that the desired number of pod replicas are running at all times.

Necessary Requirements:

- **EC2 Instance:** The experiment required launching a t2.medium EC2 instance with 2 CPUs, as Kubernetes demands sufficient resources for effective functioning.
- **Minimum Requirements:**
 - **Instance Type:** t2.medium
 - **CPUs:** 2
 - **Memory:** Adequate for container orchestration.

This ensured that the Kubernetes cluster had the necessary resources to function smoothly.

Note:

AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account.).

If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance.

Select Ubuntu as AMI and **t2.medium** as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.

Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a 'Create New Function' button. Below that, a table lists existing functions: 'HelloWorld' (Node.js, 100ms, 128 MB), 'Lambda@Edge' (Node.js, 100ms, 128 MB), and 'Lambda@Edge' (Node.js, 100ms, 128 MB). The 'HelloWorld' function is selected. The main area shows the code editor with the following code:

```
function handler(event, context) {
  // ...
}
```

Below the code editor, there are tabs for 'Overview', 'Configuration', 'Logs', and 'Test'. On the right side, there are sections for 'Environment Variables', 'Tracing', and 'Monitoring'.

Step 2: After creating the instance click on Connect the instance and navigate to SSH Client.

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a 'Launch instances' button. Below that, a table lists instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
Experiment 4	i-09f3752831db50f7d	Running	t2.medium	Initializing		us-east-1d	ec2-54-165-99-170.co...	54.165.99.170	-

EC2 > Instances > i-09f3752831db50f7d > Connect to instance

Connect to instance Info

Connect to your instance i-09f3752831db50f7d (Experiment 4) using any of these options

EC2 Instance Connect | **Session Manager** | **SSH client** (selected) | **EC2 serial console**

Instance ID

i-09f3752831db50f7d (Experiment 4)

1. Open an SSH client
2. Locate your private key file. The key used to launch this instance is `Master_Ec2_Key.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "Master_Ec2_Key.pem"
4. Connect to your instance using its Private IP:
172.31.20.171

Example:

```
ssh -i "Master_Ec2_Key.pem" ubuntu@172.31.20.171
```

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

Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)

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

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

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

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

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

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

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

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

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

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

Step 4: Run the below commands to install and setup Docker.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-20-171:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-20-171:~$ 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.
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Found existing deb-src entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src 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
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
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 0s (128 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 a
n 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 s
ection in apt-key(8) for details.
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y docker-ce
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 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security 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 a
n 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 s
ection in apt-key(8) for details.
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 cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
 containerd.io docker-buildx-plugin docker-ce docker-ce-cli
 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 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
```

```

Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 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 MB]
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 MB]
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 MB]
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 MB]
Fetched 122 MB in 2s (71.3 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 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.

Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-ce-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 (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0Ubuntu8.2) ...
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.

```

```

sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF

```

```
ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-20-171:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
```

**sudo systemctl enable docker
**sudo systemctl daemon-reload
sudo systemctl restart docker****

```
ubuntu@ip-172-31-20-171:~$ 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
```

Step 5: Run the below command to install Kubernets.

```
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
```

```
ubuntu@ip-172-31-20-171:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-20-171:~$ 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/ /
```

**sudo apt-get update
**sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl****

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
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
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 6051 B in 0s (12.9 kB/s)
Reading package lists... Done
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 130 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 noble/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 (77.1 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68011 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
```

```

Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.

sudo systemctl enable --now kubelet

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

```

ubuntu@ip-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0915 07:47:37.419191    7952 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create
new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: cod
e = Unimplemented desc = unknown service runtime.v1.RuntimeService
[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 API for endpoint "unix:///var/run/containerd/containerd.sock"
: rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal
with '--ignore-preflight-errors='...
To see the stack trace of this error execute with --v=5 or higher

```

Now We have got an error.**So we have to perform some additional commands as follow.****sudo apt-get install -y containerd**

```
To see the stack trace of this error execute with --v=5 or higher    ubuntu@ip-172-31-20-171:~$ 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 130 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 (74.5 MB/s)
(Reading database ... 68068 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 ... 68048 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.
```

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.**

sudo mkdir -p /etc/containerd**sudo containerd config default | sudo tee /etc/containerd/config.toml**

```
ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/containerd
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
```

...

sudo systemctl restart containerd**sudo systemctl enable containerd****sudo systemctl status containerd**

```
ubuntu@ip-172-31-20-171:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
ubuntu@ip-172-31-20-171:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
  Loaded: loaded (/usr/lib/systemd/system/containerd.service; en>
    Active: active (running) since Sun 2024-09-15 07:49:23 UTC; 5s>
      Docs: https://containerd.io
     Main PID: 8398 (containerd)
        Tasks: 7
       Memory: 13.5M (peak: 14.0M)
         CPU: 70ms
        CGroup: /system.slice/containerd.service
                  └─8398 /usr/bin/containerd

Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 systemd[1]: Started containerd.ser>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
```

sudo apt-get install -y socat

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get install -y socat
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 libsslip0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (12.1 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68112 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
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.
```

Step 6: Initialize the Kubecluster**sudo kubeadm init --pod-network-cidr=10.244.0.0/16**

```
ubuntu@ip-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[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'
W0915 07:49:42.979851   8570 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver cert is signed for DNS names [ip-172-31-20-171 kubernetes.kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.20.171]
[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 "/etc/certs/ca" certificate and key
[certs] Generating "/etc/certs/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "/etc/certs/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "/etc/certs/healthcheck-client" certificate and key
[certs] Generating "/apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 502.777379ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 4.501245501s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
```

```
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

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

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

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

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

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

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

```
kubeadm join 172.31.20.171:6443 --token 7acddu.inheshzwxti0372v \
--discovery-token-ca-cert-hash sha256:aed5faf97bac361d1bb7f33a89fb05d2bb28c7fc065024eac2302a734c330a36
```

Copy the mkdir and chown commands from the top and execute them.

mkdir -p \$HOME/.kube

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

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

```
ubuntu@ip-172-31-20-171:~$ mkdir -p $HOME/.kube
      sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
      sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Add a common networking plugin called flannel as mentioned in the code.

kubectl apply -f

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

```
ubuntu@ip-172-31-20-171:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment

kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

```
ubuntu@ip-172-31-20-171:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

kubectl get pods

```
ubuntu@ip-172-31-20-171:~$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-vz8rv   0/1     Pending   0          8s
nginx-deployment-d556bf558-wz5wc   0/1     Pending   0          8s
```

POD_NAME=\$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")

kubectl port-forward \$POD_NAME 8080:80

```
ubuntu@ip-172-31-20-171:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-20-171:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
```

Note : We have faced an error as pod status is pending so make it running run below commands then again run above 2 commands.

kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted

kubectl get nodes

```
ubuntu@ip-172-31-20-171:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-20-171 untainted
ubuntu@ip-172-31-20-171:~$ kubectl get nodes
NAME            STATUS   ROLES      AGE     VERSION
ip-172-31-20-171   Ready   control-plane   5m23s   v1.31.1
```

kubectl get pods

```
ubuntu@ip-172-31-20-171:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-vz8rv   1/1     Running   0          3m4s
nginx-deployment-d556bf558-wz5wc   1/1     Running   0          3m4s
```

POD_NAME=\$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")

kubectl port-forward \$POD_NAME 8080:80

```
ubuntu@ip-172-31-20-171:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

Step 8: Verify your deployment

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

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

curl --head http://127.0.0.1:8080

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

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

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

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

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

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

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

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

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

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

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

ubuntu@ip-172-31-20-171:~$
```

If the response is 200 OK and you can see the Nginx server name, your deployment was successful.

We have successfully deployed our Nginx server on our EC2 instance.

Conclusion:

In this experiment, we successfully installed Kubernetes on an EC2 instance and deployed an Nginx server using Kubectl commands. During the process, we encountered two main errors: the Kubernetes pod was initially in a pending state, which was resolved by removing the control-plane taint using `kubectl taint nodes --all`, and we also faced an issue with the missing `containerd` runtime, which was fixed by installing and starting containerd. We used a **t2.medium EC2 instance with 2 CPUs** to meet the necessary resource requirements for the Kubernetes setup and deployment.

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

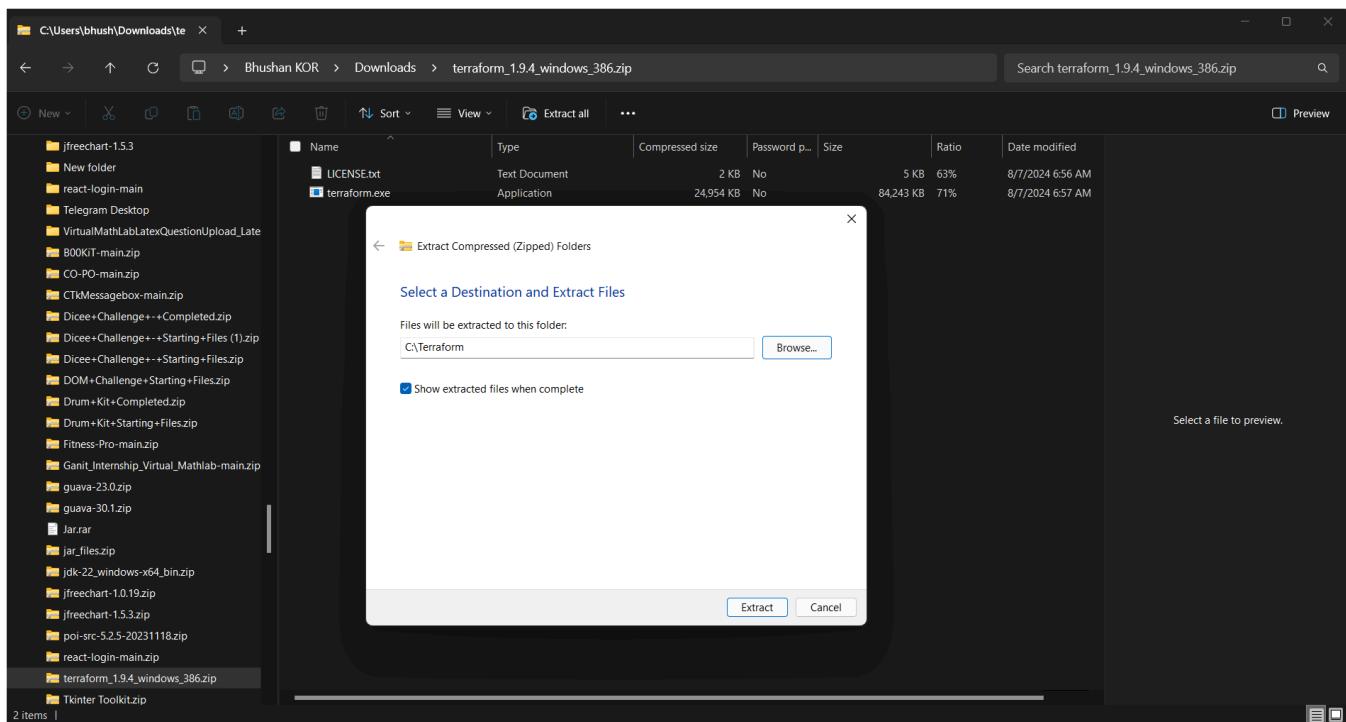
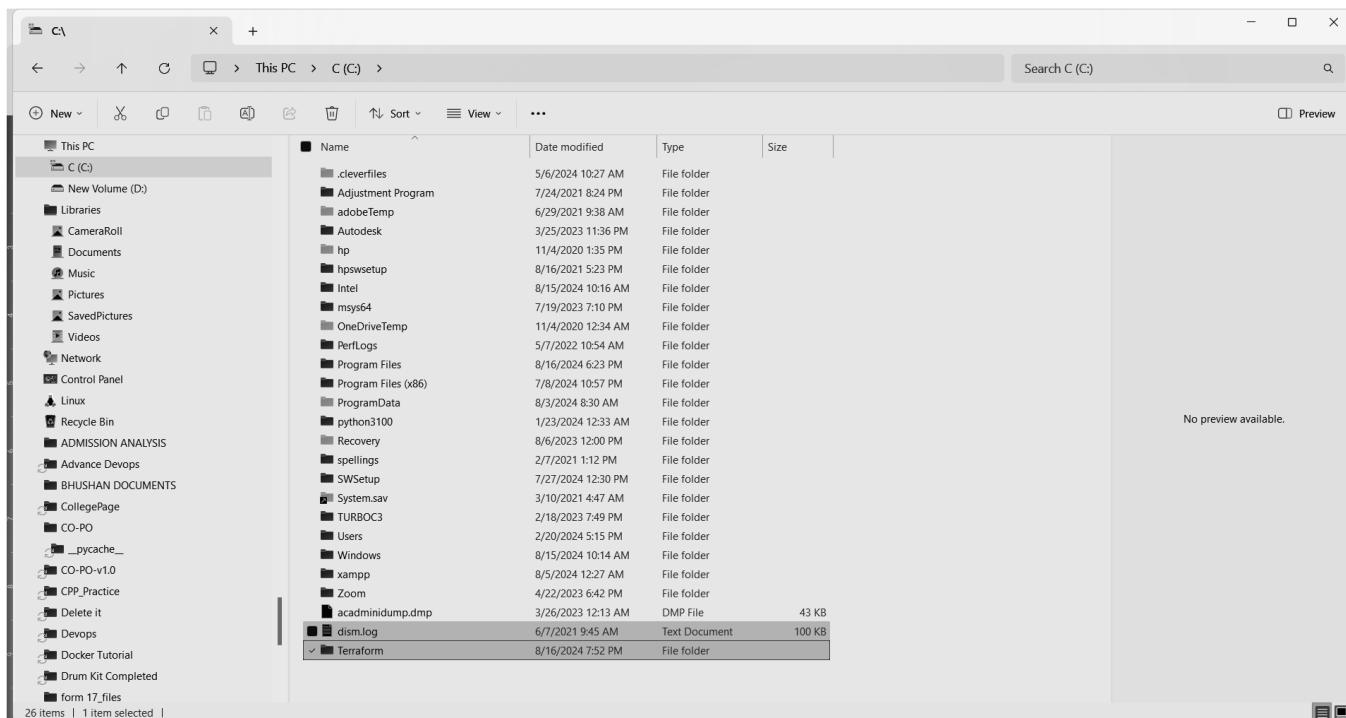
Terraform: Terraform is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It allows you to define, provision, and manage infrastructure across various cloud providers using a simple declarative configuration language. Terraform's key features include automation of infrastructure deployments, version control for infrastructure changes, and the ability to manage both cloud and on-premises resources in a consistent manner.

Step 1:Visit the website:

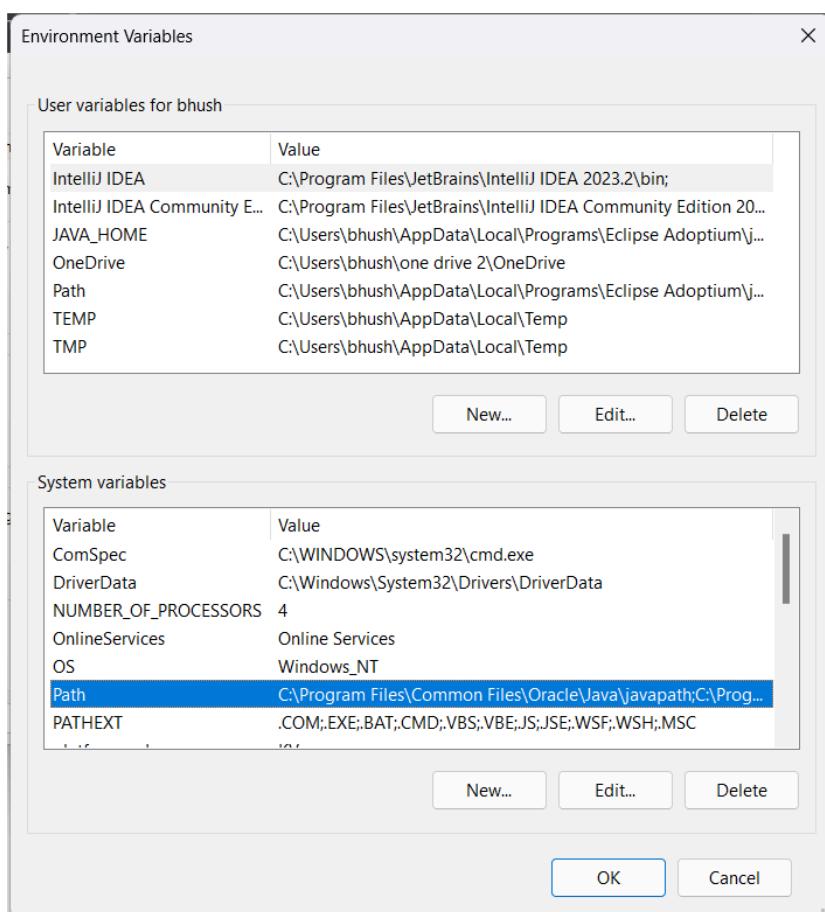
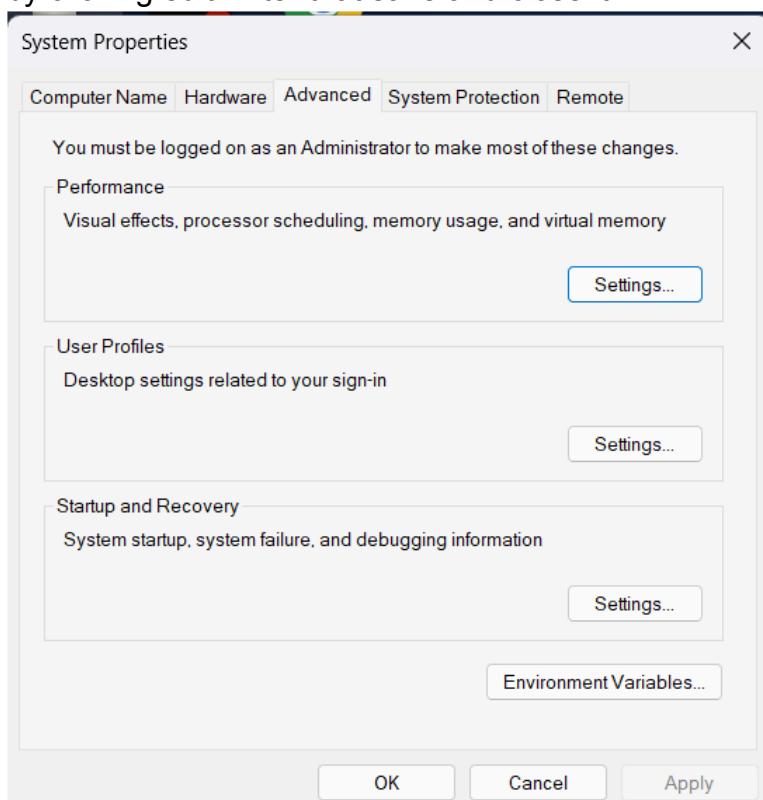
https://developer.hashicorp.com/terraform/install?product_intent=terraform and install the terraform according to your system. Here we are downloading it for Windows.

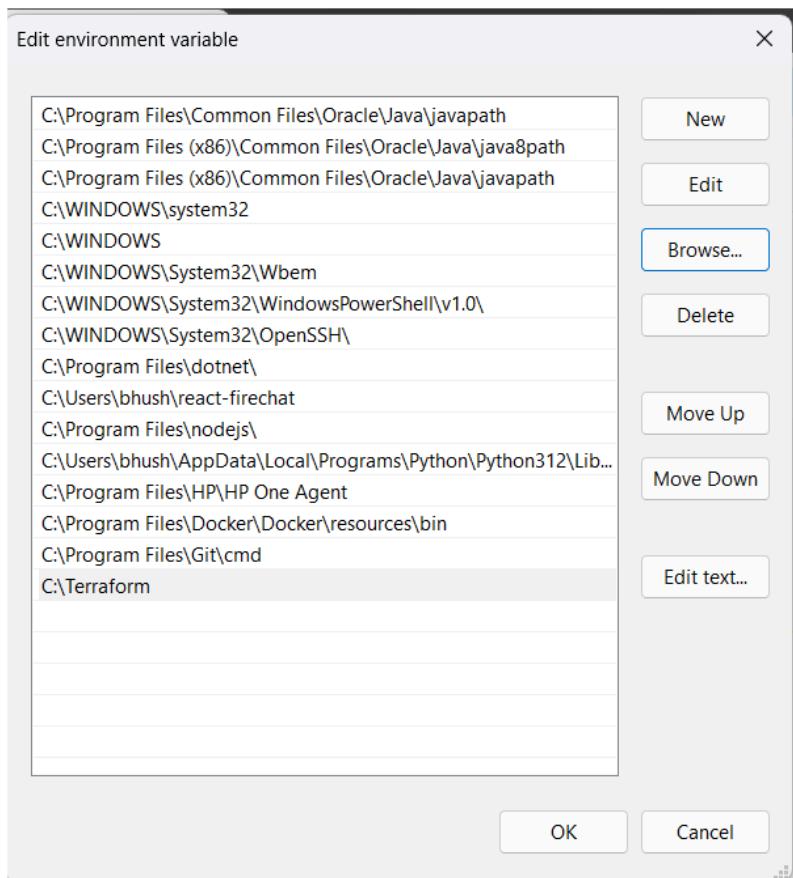
The screenshot shows the Terraform website's 'Install' page. On the left, there's a sidebar with links for 'Operating Systems' (macOS, Windows, Linux, FreeBSD, OpenBSD, Solaris), 'Release information', and 'Next steps'. The main content area has tabs for 'Windows' and 'Linux'. Under 'Windows', there are download links for '386' (Version: 1.9.4) and 'AMD64' (Version: 1.9.4). Under 'Linux', there's a 'Package manager' section with links for 'Ubuntu/Debian', 'CentOS/RHEL', 'Fedora', 'Amazon Linux', and 'Homebrew'. A terminal window at the bottom shows the command: 'wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg' followed by 'echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com/ \$(lsb_release -cs) main"'.

Step 2: Done Downloaded Successfully Now create the folder named Terraform in C: directory and extract the file in C:\Terraform.

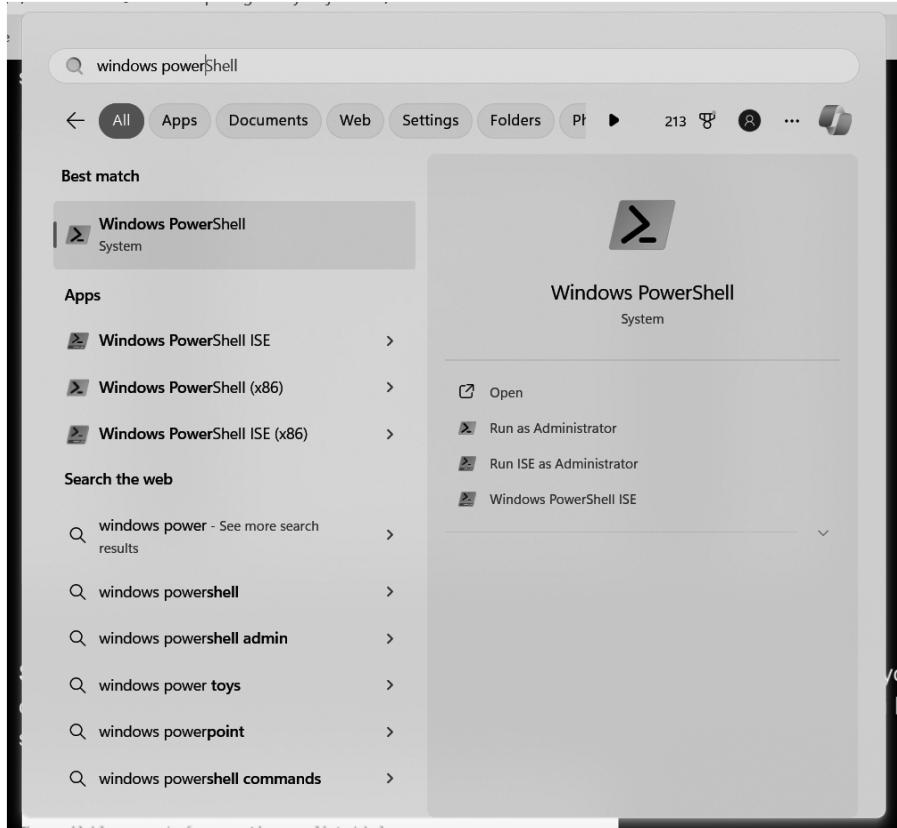


Step 3: Edit the environment variable. In the path of system variables add the path of terraform by clicking edit. After that save and close it.





Step 4: Search for Windows power shell and Run it as administrator.



Step 5: Now run the command terraform in Windoes Powershell.

```
Administrator: Windows PowerShell (x86)
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\WINDOWS\system32> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  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:\WINDOWS\system32> _
```

Step 6: Congratulations You have successfully installed Terraform in your system.

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

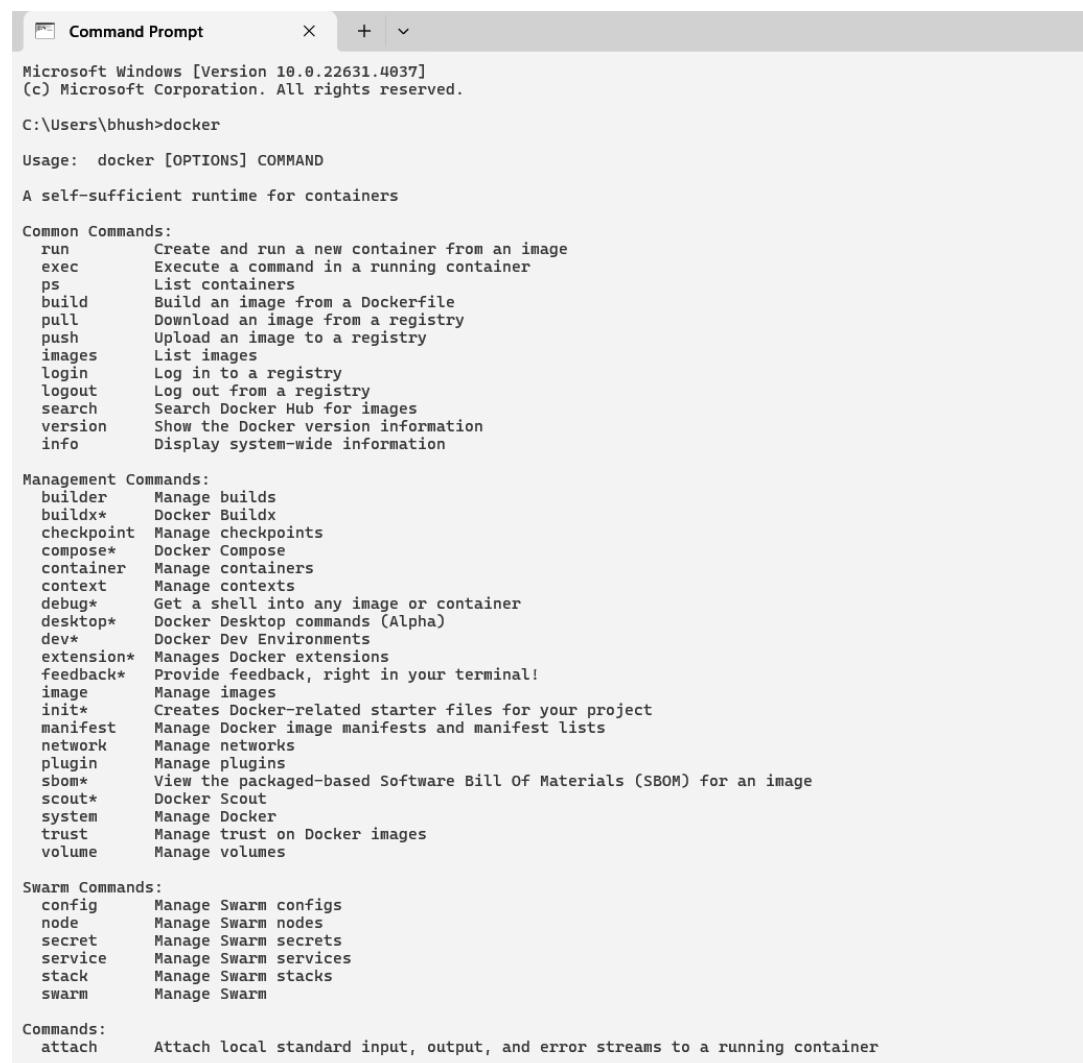
Terraform: Terraform is an open-source infrastructure as code (IaC) tool that allows you to define, provision, and manage cloud resources across various providers using a declarative configuration language. It enables consistent and repeatable infrastructure deployments, supports multi-cloud environments, and maintains state files to track resource changes. Terraform automates the creation and management of infrastructure, making it easier to scale and modify resources.

Creating a docker image using Terraform :

Prerequisite:

Download and Install Docker Desktop from <https://www.docker.com/>

Step 1: Run docker command in cmd to check the functionality of docker and also run docker --version to check which docker version is installed on your system.



```
Command Prompt      X  +  ▾
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhush>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
```



```

Command Prompt      + | v

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
pause            Pause all processes within one or more containers
port             List port mappings or a specific mapping for the container
rename           Rename a container
restart          Restart one or more containers
rm               Remove one or more containers
rmi              Remove one or more images
save             Save one or more images to a tar archive (streamed to STDOUT by default)
start            Start one or more stopped containers
stats            Display a live stream of container(s) resource usage statistics
stop             Stop one or more running containers
tag              Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top              Display the running processes of a container
unpause          Unpause all processes within one or more containers
update           Update configuration of one or more containers
wait             Block until one or more containers stop, then print their exit codes

Global Options:
--config string   Location of client config files (default "C:\\\\Users\\\\bhush\\\\.docker")
--c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug        Enable debug mode
-H, --host list    Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls             Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "C:\\\\Users\\\\bhush\\\\.docker\\\\ca.pem")
--tlscert string   Path to TLS certificate file (default "C:\\\\Users\\\\bhush\\\\.docker\\\\cert.pem")
--tlskey string    Path to TLS key file (default "C:\\\\Users\\\\bhush\\\\.docker\\\\key.pem")
--tlsverify       Use TLS and verify the remote
-v, --version      Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

C:\\\\Users\\\\bhush>

```

```
C:\\\\Users\\\\bhush>docker --version
Docker version 27.0.3, build 7d4bcd8
```

```
C:\\\\Users\\\\bhush>
```

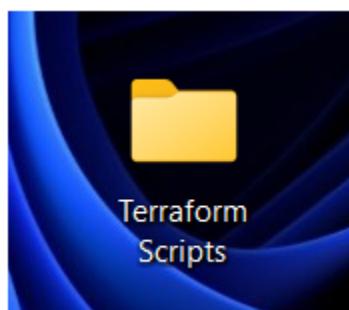
Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

Step 2: Now create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using Atom editor or Vscode and write the following contents into it to create a Ubuntu Linux container.

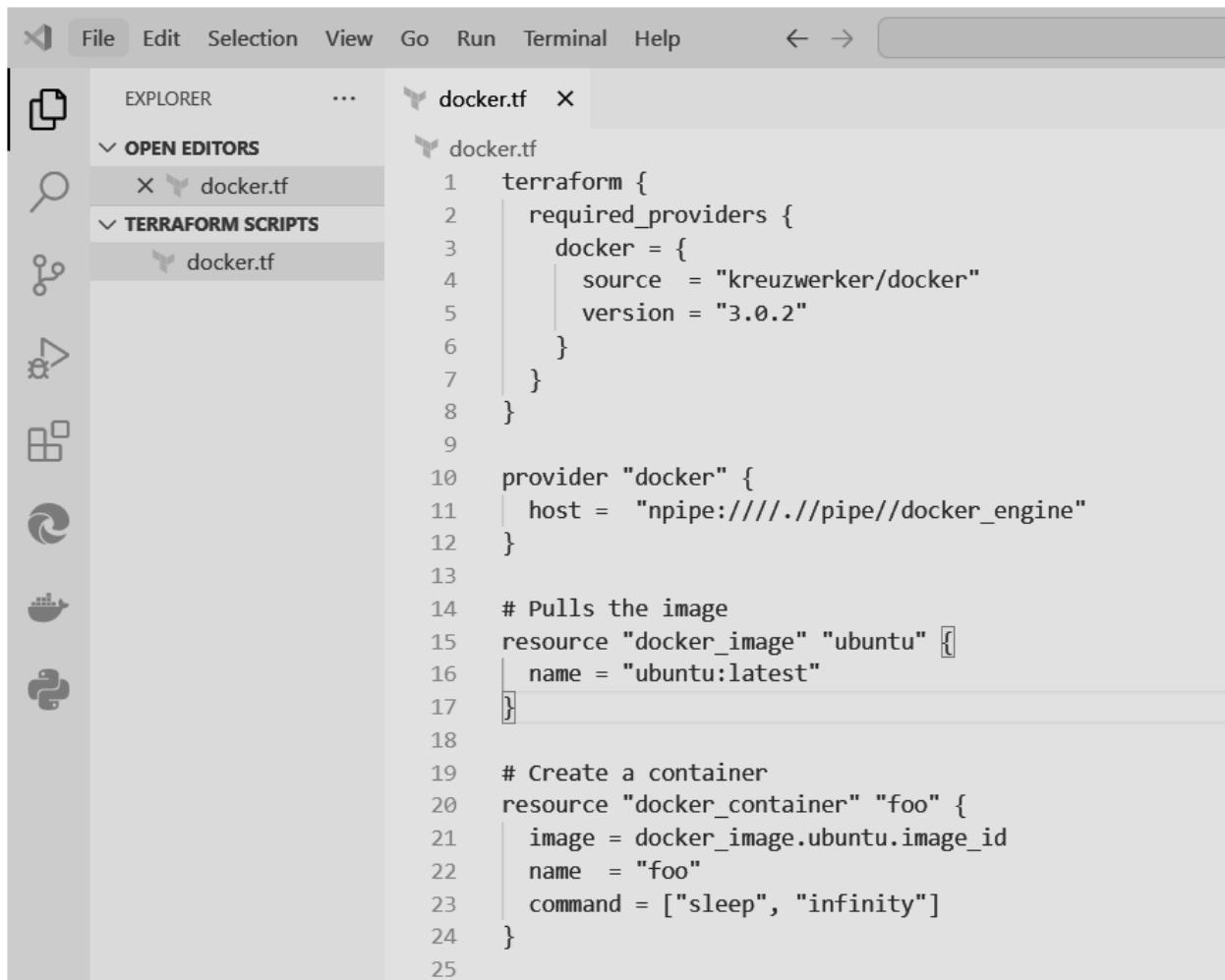
(Note part of the script highlighted is a must to get the image otherwise it will give an error.)

Script:

```
terraform {  
    required_providers {  
        docker = {  
            source  = "kreuzwerker/docker"  
            version = "3.0.2"  
        }  
    }  
}  
  
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_id  
    name  = "foo"  
    command = ["sleep", "infinity"]  
}
```



Name	Status	Date modified	Type	Size
docker.tf	🕒	8/23/2024 5:54 PM	TF File	1 KB

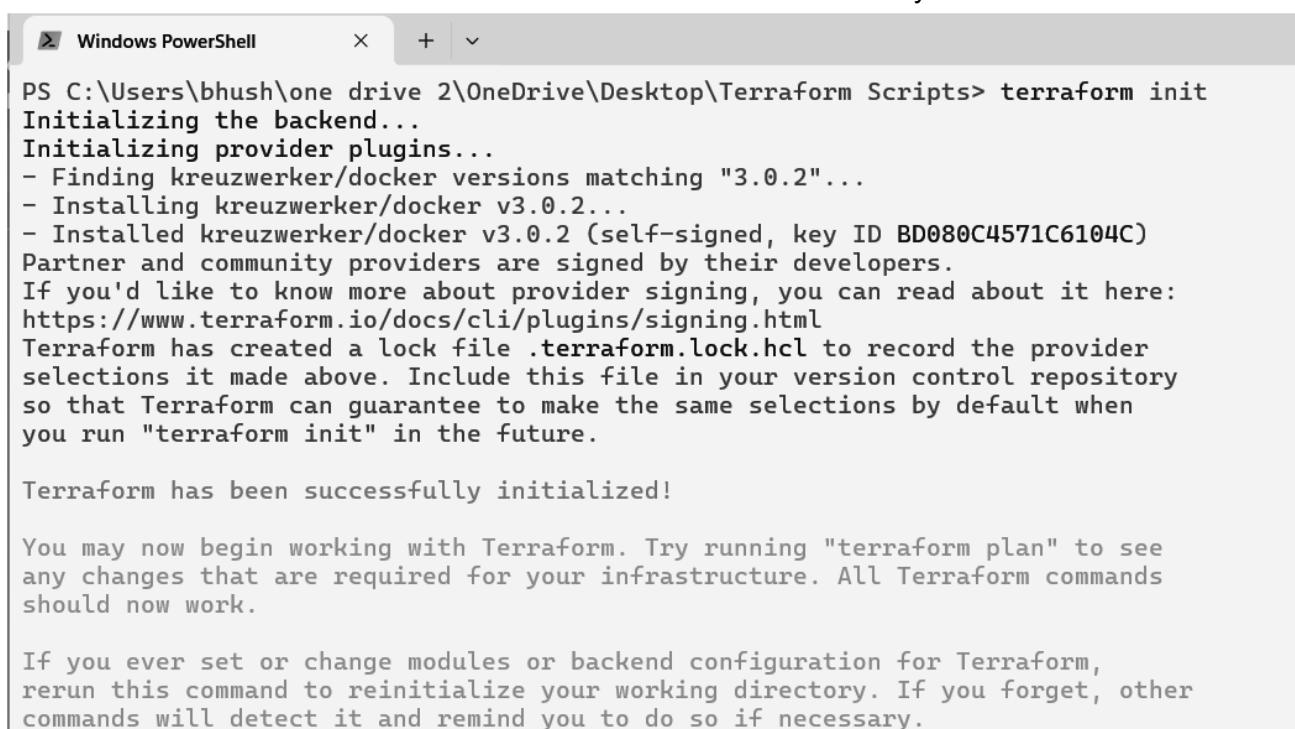


```

1 terraform {
2   required_providers {
3     docker = {
4       source  = "kreuzwerker/docker"
5       version = "3.0.2"
6     }
7   }
8 }
9
10 provider "docker" {
11   host = "npipe://./pipe//docker_engine"
12 }
13
14 # Pulls the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "infinity"]
24 }
25

```

Step 3: Now open the terminal in the Terraform Scripts folder and Execute the terraform init command to initialize resources. This will initialize terraform in directory.



```

PS C:\Users\bhush\OneDrive\Desktop\Terraform Scripts> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "3.0.2"...
- Installing kreuzwerker/docker v3.0.2...
- Installed kreuzwerker/docker v3.0.2 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
  https://www.terraform.io/docs/cli/plugins/signing.html
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

```

Step 4: Run the command terraform plan. This will create an execution plan and let you overview changes that are going to happen in your infrastructure.

```
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> terraform plan

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach                               = false
    + bridge                               = (known after apply)
    + command
        + "sleep",
        + "infinity",
    ]
    + container_logs                      = (known after apply)
    + container_read_refresh_timeout_milliseconds = 15000
    + entrypoint                          = (known after apply)
    + env                                 = (known after apply)
    + exit_code                           = (known after apply)
    + hostname                           = (known after apply)
    + id                                 = (known after apply)
    + image                               = (known after apply)
    + init                               = (known after apply)
    + ipc_mode                           = (known after apply)
    + log_driver                         = (known after apply)
    + logs                               = false
    + must_run                           = true
    + name                               = "foo"
    + network_data                      = (known after apply)
    + read_only                          = false
    + remove_volumes                    = true
    + restart                            = "no"
    + rm                                = false
    + runtime                            = (known after apply)
    + security_opts                     = (known after apply)
    + shm_size                           = (known after apply)
    + start                             = true
    + stdin_open                         = false
    + stop_signal                        = (known after apply)
    + stop_timeout                       = (known after apply)
    + tty                               = false
    + wait                             = false
    + wait_timeout                      = 60

    + healthcheck (known after apply)
    + labels (known after apply)
}

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

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

Step 5: Now, run the command terraform apply to carry out the changes that

We have made when terrafrom plan command was executed.After running the command it will ask for a value for confirmation that time type yes.(Before step 5 run command docker images for next step)

```
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach                               = false
    + bridge                               = (known after apply)
    + command
        + "sleep",
        + "infinity",
    ]
    + container_logs                      = (known after apply)
    + container_read_refresh_timeout_milliseconds = 15000
    + entrypoint                           = (known after apply)
    + env                                  = (known after apply)
    + exit_code                            = (known after apply)
    + hostname                            = (known after apply)
    + id                                   = (known after apply)
    + image                                = (known after apply)
    + init                                 = (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_open                           = false
    + stop_signal                         = (known after apply)
    + stop_timeout                        = (known after apply)
    + tty                                 = false
    + wait                                = false
    + wait_timeout                        = 60

    + healthcheck (known after apply)
    + labels (known after apply)
}

# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
    + id          = (known after apply)
    + image_id   = (known after apply)

    + name       = "ubuntu:latest"
    + repo_digest = (known after apply)
}

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Still creating... [40s elapsed]
docker_image.ubuntu: Creation complete after 42s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 3s [id=2157b0fa12aed015eaaf4b3686ce28eca721f409d1256450da2512f0830374c3]

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

Step 6: Now run the command docker images before terraform apply command and after terraform apply command.And see the changes.

Before Terraform apply Command :

```
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> |
```

After Terraform apply Command :

```
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   edbfe74c41f8  3 weeks ago  78.1MB
```

Step 7: From above command we can clearly see that the ubuntu image is created.Now we have to destroy it, so we will use terraform destroy command.After running the command it will ask for a value for confirmation that time type yes.

```
PS C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=2157b0fa12aed015eaaf4b3686ce28eca721f409d1256450da2512f0830374c3]

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
  - command
    - "sleep",
    - "infinity",
  ] -> null
  - container_read_refresh_timeout_milliseconds = 15000 -> null
  - cpu_shares
  - dns
  - dns_opts
  - dns_search
  - entrypoint
  - env
  - group_add
  - hostname
  - id
-> null
  - image
  - init
  - ipc_mode
  - log_driver
  - log_opts
  - logs
  - max_retry_count
  - memory
  - memory_swap
  - must_run
  - name
  - network_data
    - {
      - gateway
        = "172.17.0.1"
      - global_ipv6_prefix_length = 0
      - ip_address
        = "172.17.0.2"
      - ip_prefix_length
        = 16
      - mac_address
        = "02:42:ac:11:00:02"
      - network_name
        = "bridge"
      # (2 unchanged attributes hidden)
    },
] -> null
  - network_mode
  - privileged
  - publish_all_ports
  - read_only
  - remove_volumes
  - restart
}
```

```

- remove_volumes          = true -> null
- restart                 = "no" -> null
- rm                      = false -> null
- runtime                 = "runc" -> null
- security_opts           = [] -> null
- shm_size                = 64 -> null
- start                   = true -> null
- stdin_open               = false -> null
- stop_timeout             = 0 -> null
- storage_opts             = {} -> null
- sysctls                  = {} -> null
- tmpfs                    = {} -> null
- tty                      = false -> null
- wait                     = false -> null
- wait_timeout              = 60 -> null
  # (8 unchanged attributes hidden)
}

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=2157b0fa12aed015eaaf4b3686ce28eca721f409d1256450da2512f0830374c3]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.

```

Again Run docker image command to verify image is deleted or not.

```

PS C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
PS C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts>

```

Step 7: Done You have successfully created a docker image of Ubuntu using Terraform and also destroyed it.

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

Terraform: Terraform is an open-source infrastructure as code (IaC) tool that allows you to define, provision, and manage cloud resources across various providers using a declarative configuration language. It enables consistent and repeatable infrastructure deployments, supports multi-cloud environments, and maintains state files to track resource changes. Terraform automates the creation and management of infrastructure, making it easier to scale and modify resources.

Prerequisites:

- Install Terraform .
 - Install Vscode.
 - Hashicorp extension in Vscode.
- AWS Academy Account

Step 1: Open your AWS Academy account . Then Start the lab from modules. After starting the lab click on the AWS details and click on show button after the AWS CLI to get Access keys and other details now copy full credentials.

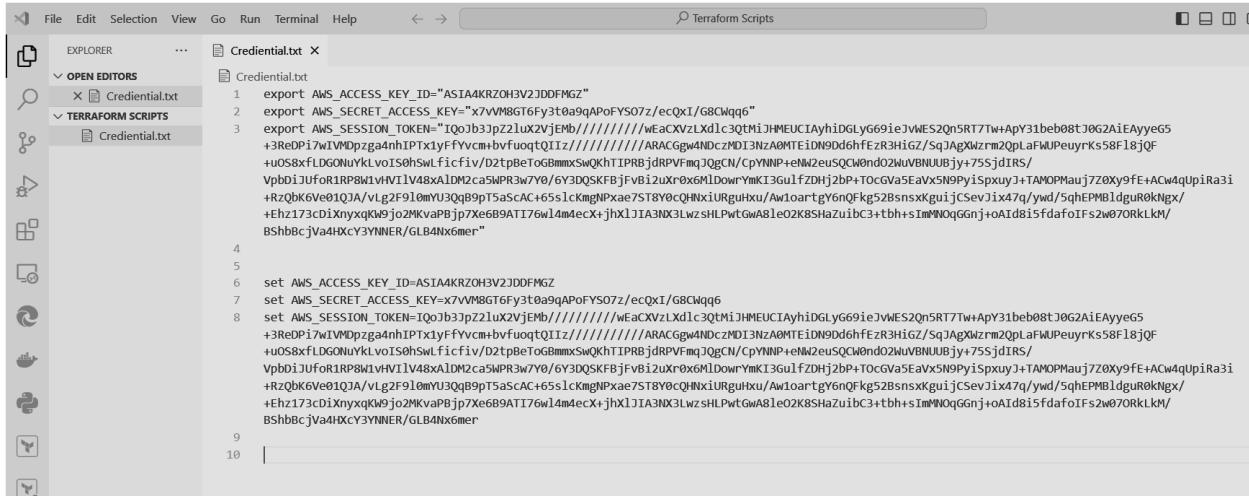
The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with icons for Home, Modules (selected), Discussions, Grades, and Lucid. The main area has a breadcrumb navigation: ALLv2EN-US... > Modules > AWS Acad... > Launch AWS Academy Learner Lab. A progress bar indicates 'Used \$16.5 of \$50' and a timer shows '03:59'. At the top right are buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', and 'Reset'. Below the navigation is a terminal window titled 'eee_W_3387987@runweb131836:~\$'. To the right of the terminal is a large panel titled 'AWS CLI:' containing the copied AWS CLI credentials. At the bottom are navigation buttons for 'Previous' and 'Next'.

Step 2: Now Create a folder Named As “Terraform Scripts “ and Open it on Vscode or any code editor. Now create one file Credential.txt and paste the copied credentials 2 times .

```
aws_access_key_id=ASIA4KRZ0H3V2JDDFMGZ
aws_secret_access_key=x7VM8GT6Fy3t0a9qAPoFYSo7z/ecQxI/G8Cwqq6
aws_session_token=IQoJb3jpZ2luX2VjEMb//////////wEaCXvzLxdlc3QtMiJHMEUCIAyhIDGLyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ0G2AiEAyyeG5
+3ReDPi7wIVMDpzga4nhIPTxlyFFvcm+bvfuqtQITz//////////ARACGgw4NDczMDI3NzA0MTEiD99d6hfEZr3HIGz/SqJAgXwzrm2QpLaFWUPeuyrKs58F18jqF
+u0S8xfLDGONuYkLvoIS0hSwlficfiv/D2tpBeToBmmxSwQkhTIPRBjdrPvFmqJQgCN/CpYNP+nEW2euScQw0nd02wUVBNNUUBjy+75SjdIRS/
VpbDiJuf0R1RP8W1vHVI1V48xAlDM2ca5WPR3w7Y0/6Y3DQSFKBjFvBi2uXr0x6MlDowrYmKI3Gu1fZDHj2bP+TocGva5Eavx5N9PyiSpuxyJ+TAMOPMauj7Z0xy9fE+Acw4qUpiRa3i
+RzQbK6Ve01QjA/vlg2F9l0myU3qQbpT5aScAC+65s1ckmgNPxae7ST8Y0cQHNxiURguhxu/Aw1oartgY6nfkg52BsnsxkguijCSevJix47q/ywd/5qhEPMBldguR0kNgx/
+Ehz173cdiXnyxqKw9jo2MKvapBjp7xe6B9ATI76w14m4ecX+jhXlJIA3NX3LwzsHLPwtGwA8leO2K8SHaZuibC3+tbh+sIMMNoqGGnj+oAId8i5fdfaoIFs2w070RkLkm/BShhbCjVa4Hx
BShhbCjVa4HxycY3YNNER/GLB4Nx6mer

aws_access_key_id=ASIA4KRZ0H3V2JDDFMGZ
aws_secret_access_key=x7VM8GT6Fy3t0a9qAPoFYSo7z/ecQxI/G8Cwqq6
aws_session_token=IQoJb3jpZ2luX2VjEMb//////////wEaCXvzLxdlc3QtMiJHMEUCIAyhIDGLyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ0G2AiEAyyeG5
+3ReDPi7wIVMDpzga4nhIPTxlyFFvcm+bvfuqtQITz//////////ARACGgw4NDczMDI3NzA0MTEiD99d6hfEZr3HIGz/SqJAgXwzrm2QpLaFWUPeuyrKs58F18jqF
+u0S8xfLDGONuYkLvoIS0hSwlficfiv/D2tpBeToBmmxSwQkhTIPRBjdrPvFmqJQgCN/CpYNP+nEW2euScQw0nd02wUVBNNUUBjy+75SjdIRS/
VpbDiJuf0R1RP8W1vHVI1V48xAlDM2ca5WPR3w7Y0/6Y3DQSFKBjFvBi2uXr0x6MlDowrYmKI3Gu1fZDHj2bP+TocGva5Eavx5N9PyiSpuxyJ+TAMOPMauj7Z0xy9fE+Acw4qUpiRa3i
+RzQbK6Ve01QjA/vlg2F9l0myU3qQbpT5aScAC+65s1ckmgNPxae7ST8Y0cQHNxiURguhxu/Aw1oartgY6nfkg52BsnsxkguijCSevJix47q/ywd/5qhEPMBldguR0kNgx/
+Ehz173cdiXnyxqKw9jo2MKvapBjp7xe6B9ATI76w14m4ecX+jhXlJIA3NX3LwzsHLPwtGwA8leO2K8SHaZuibC3+tbh+sIMMNoqGGnj+oAId8i5fdfaoIFs2w070RkLkm/BShhbCjVa4Hx
BShhbCjVa4HxycY3YNNER/GLB4Nx6mer
```

Now Make RHS of both copied keys to upper case and put the LHS of 1st copied keys inside the Double quotes “”
And for 1st copied give prefix export and 2nd copied give prefix set.

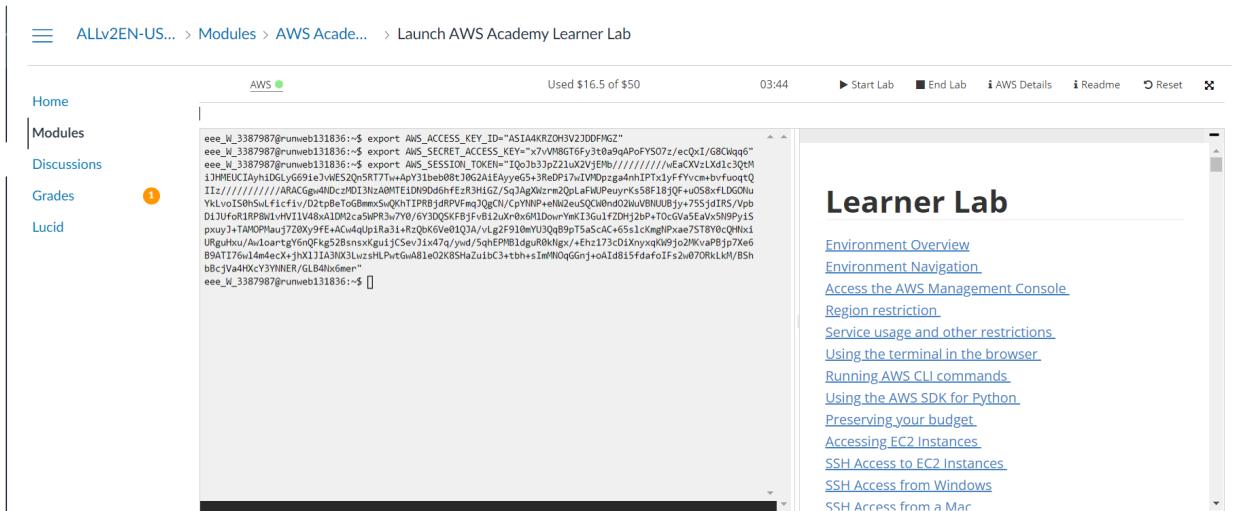


```

File Edit Selection View Go Run Terminal Help ↵ → Terraform Scripts
OPEN EDITORS Credential.txt
Credential.txt
1 export AWS_ACCESS_KEY_ID="ASIA4KRZOH3V2JDDFMGZ"
2 export AWS_SECRET_ACCESS_KEY="x7vVM8Gt6Fy3t0aq9APoFYS07z/ecQxI/G8CwQqg6"
3 export AWS_SESSION_TOKEN="IQoJb3Jp2lu2lx2VjEmB//////////wEAxVzLxd1c3oMjJHMEUCIAyhidiDGlyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ062AiEAYyeG5
+3ReP17wIVMDpzga4nhPTx1yfFvcm+bvfuoqtQIIz//////////ARACGgw4HDCzMD13nza0MTEiDn9d6hfEr3HiGz/SqJAgXzrmQ2pLaFwUPeuryk58F18jQF
+uOSRxFLDGNuVkyLvoISohSwLficfiv/D2tpbEt0GBmmxSwQKHTPRBjdrPVfmqJ0gCh/CpYNNP+eNwZeusQCM0ndo2wuvBNNUUbj+y75sjdIRS/
VpbD1JUf0RIRP8w1MV1lV48xALDm2a5WP3rW7Y0/6Y3DQSKFBjfvb12uxr0x61DownrYmkI3Gu1fZDHj2bP+TOCGVa5eavxSN9PyiSpuxyJ+TAMOPMauj7Z0Xy9fE+AcW4qUpiRa3i
+RzQbK6Ve01QJA/vlgF910mY3Q0qB9pt5aScAC+65s1cKmgNPxae7ST8YoCQhNxuiRguixu/AwLoartyg6nQfk52BsnsxkguijCsevJix47q/ywd/5qhEPMBldguR0KNgx/
+Ehz173cdixnyxqkW9jo2MKMvaPBjpx7xe6B9AT76w14meecX+jhx1JIA3NX3lwzShl.PvtGwA81e02K8SHaZu1bC3+tbh+sIMM0qGGnj+oAId815fdfa0Ifs2w070Rklkm/BShbcbjVa4HxycY3YNNEr/GLB4Nx6mer"
4
5
6 set AWS_ACCESS_KEY_ID=ASIA4KRZOH3V2JDDFMGZ
7 set AWS_SECRET_ACCESS_KEY=x7vVM8Gt6Fy3t0aq9APoFYS07z/ecQxI/G8CwQqg6
8 set AWS_SESSION_TOKEN=IQoJb3Jp2lu2lx2VjEmB//////////wEAxVzLxd1c3oMjJHMEUCIAyhidiDGlyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ062AiEAYyeG5
+3ReP17wIVMDpzga4nhPTx1yfFvcm+bvfuoqtQIIz//////////ARACGgw4HDCzMD13nza0MTEiDn9d6hfEr3HiGz/SqJAgXzrmQ2pLaFwUPeuryk58F18jQF
+uOSRxFLDGNuVkyLvoISohSwLficfiv/D2tpbEt0GBmmxSwQKHTPRBjdrPVfmqJ0gCh/CpYNNP+eNwZeusQCM0ndo2wuvBNNUUbj+y75sjdIRS/
VpbD1JUf0RIRP8w1MV1lV48xALDm2a5WP3rW7Y0/6Y3DQSKFBjfvb12uxr0x61DownrYmkI3Gu1fZDHj2bP+TOCGVa5eavxSN9PyiSpuxyJ+TAMOPMauj7Z0Xy9fE+AcW4qUpiRa3i
+RzQbK6Ve01QJA/vlgF910mY3Q0qB9pt5aScAC+65s1cKmgNPxae7ST8YoCQhNxuiRguixu/AwLoartyg6nQfk52BsnsxkguijCsevJix47q/ywd/5qhEPMBldguR0KNgx/
+Ehz173cdixnyxqkW9jo2MKMvaPBjpx7xe6B9AT76w14meecX+jhx1JIA3NX3lwzShl.PvtGwA81e02K8SHaZu1bC3+tbh+sIMM0qGGnj+oAId815fdfa0Ifs2w070Rklkm/BShbcbjVa4HxycY3YNNEr/GLB4Nx6mer"
9
10

```

Step 3: Copy paste the 1st paragraph or 1st copied in learners lab terminal and hit enter.



ALLv2EN-US... > Modules > AWS Academ... > Launch AWS Academy Learner Lab

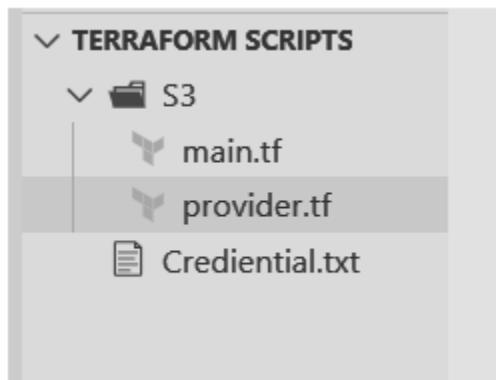
AWS Used \$16.5 of \$50 03:44 Start Lab End Lab AWS Details Readme Reset

eee_W_3387987@runweb131836:~\$ export AWS_ACCESS_KEY_ID="ASIA4KRZOH3V2JDDFMGZ"
eee_W_3387987@runweb131836:~\$ export AWS_SECRET_ACCESS_KEY="x7vVM8Gt6Fy3t0aq9APoFYS07z/ecQxI/G8CwQqg6"
eee_W_3387987@runweb131836:~\$ export AWS_SESSION_TOKEN="IQoJb3Jp2lu2lx2VjEmB//////////wEAxVzLxd1c3oMjJHMEUCIAyhidiDGlyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ062AiEAYyeG5
+3ReP17wIVMDpzga4nhPTx1yfFvcm+bvfuoqtQIIz//////////ARACGgw4HDCzMD13nza0MTEiDn9d6hfEr3HiGz/SqJAgXzrmQ2pLaFwUPeuryk58F18jQF
+uOSRxFLDGNuVkyLvoISohSwLficfiv/D2tpbEt0GBmmxSwQKHTPRBjdrPVfmqJ0gCh/CpYNNP+eNwZeusQCM0ndo2wuvBNNUUbj+y75sjdIRS/
VpbD1JUf0RIRP8w1MV1lV48xALDm2a5WP3rW7Y0/6Y3DQSKFBjfvb12uxr0x61DownrYmkI3Gu1fZDHj2bP+TOCGVa5eavxSN9PyiSpuxyJ+TAMOPMauj7Z0Xy9fE+AcW4qUpiRa3i
+RzQbK6Ve01QJA/vlgF910mY3Q0qB9pt5aScAC+65s1cKmgNPxae7ST8YoCQhNxuiRguixu/AwLoartyg6nQfk52BsnsxkguijCsevJix47q/ywd/5qhEPMBldguR0KNgx/
+Ehz173cdixnyxqkW9jo2MKMvaPBjpx7xe6B9AT76w14meecX+jhx1JIA3NX3lwzShl.PvtGwA81e02K8SHaZu1bC3+tbh+sIMM0qGGnj+oAId815fdfa0Ifs2w070Rklkm/BShbcbjVa4HxycY3YNNEr/GLB4Nx6mer"
eee_W_3387987@runweb131836:~\$

Learner Lab

Environment Overview Environment Navigation Access the AWS Management Console Region restriction Service usage and other restrictions Using the terminal in the browser Running AWS CLI commands Using the AWS SDK for Python Preserving your budget Accessing EC2 Instances SSH Access to EC2 Instances SSH Access from Windows SSH Access from a Mac

Step 4: Now create S3 folder inside the same folder and create provider.tf and main.tf file in S3 folder.



Step 5: In provider.tf add the following script and do not forget to add your key values and region inside the “ ” quotes all details you will get in AWS details in lab .

```
provider "aws" {
  access_key=""
  secret_key=""
  token =""
  region=""
}
```

```
S3 > provider.tf > provider "aws"
1 provider "aws" []
2 access_key="ASIA4KRZOH3V2JDDFMGZ"
3 secret_key="x7VM8GT6y3taoqAPoFYS07z/ecQXI/G8CWqq6"
4 token ="IQoJb3jp221uXv2jEmb//////////wEaCXvLXd1c3qtMiJHEUCIAyhIDGLyG69ieJvWES2Qn5RT7Tw+ApY31beb08tJ0G2AiEayyG5+3ReDPi7wIVMDpzga4nhIPTx1yFfYvcm
+bvfuoqtQIIz//////////ARACGgw4NDcZMDI3NzA0MTEidN9d6hfEzR3HiGz/SqJAgXwrm2QpLaFWUPeuyrKs5F18jqF+u0SxfLDG0NuYkLvoIS0hSwLfifciv/
D2tpBeToG8mmxSwQchTPRBjdrPVfmqJogCN/CpYNnP+eNW2euSQCW0ndo2wvBNUUBjy+75sjdIRS/vpdDiJUfOr1RPBW1vHVI1V48xAlM2ca5wPR3wY0/
6Y3DQSKEFBjfvb2iuXr0x6MldowrYmk13GuLfZDHj2P+TcGVa5EavX5N9PyiSpuxyJ+TAMOPMauj7Z0xy9fE+ACw4qUpiRa3i+Rzbk6Ve01QJA/vLg2F9l0mYU3QqB9pt5aScAC
+65s1cKmgNPxae7ST8Y0cQHNxiURguhx/AwIoartgy6nQFkg52BsnsxKgujCSevJix47q/ywd/SqHEPMB1dgur0kNgx/+Ehz173cd1XnyxqKw9jo2MKvaPBjp7xe6B9ATI76w14m4ec
+jhx1lIA3Nx3LwshLwptGwA81eO2K8SHaZuibC3+tbh+sIMMNQqGGnj+oAIId8i5fdf0fIFS2w070RKLM/BShb8cjVa4HxCy3YNNER/GLB4Nx6mer"
5 region="us-east-1"
6 }
```

Step 6 : Visit the website Terraform By Hashicorp go inside registry then click on Browse provider then click on or search AWS then click on use provider and copy the code exact provider part. Because provider part we have already cover.Paste the code in main.tf.

The screenshot shows the Hashicorp Registry page for the AWS provider. The provider is listed as 'aws' with version 5.64.0 published 2 days ago. The provider is maintained by Hashicorp and is part of the Public Cloud category. The 'How to use this provider' section contains the following Terraform configuration code:

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.64.0"
    }
  }
}

provider "aws" {
  # Configuration options
}
```

```
terraform {  
    required_providers {  
        aws = {  
            source = "hashicorp/aws"  
            version = "5.64.0"  
        }  
    }  
}
```



The screenshot shows a code editor with three tabs: Credential.txt, provider.tf, and main.tf. The main.tf tab is active, displaying the following Terraform configuration:

```
S3 > main.tf > terraform  
1   terraform {  
2       required_providers {  
3           aws = {  
4               source = "hashicorp/aws"  
5               version = "5.64.0"  
6           }  
7       }  
8   }
```

Step 7: Now write the following code in main.tf to create the bucket.

```
resource "aws_s3_bucket" "any_name" {  
  
    bucket = "Bucket_Name_It_sholud_be_Uneque_and_all_in_lowercase_follow_Documentation"  
    tags = {  
        Name="Any_name"  
    }  
}
```

The screenshot shows a code editor with three tabs: 'Credential.txt', 'provider.tf', and 'main.tf'. The 'main.tf' tab is active and displays the following Terraform code:

```
S3 > main.tf > resource "aws_s3_bucket" "my-bucket-us-east-1" > bucket
1  terraform {
2    required_providers {
3      aws = {
4        source = "hashicorp/aws"
5        version = "5.64.0"
6      }
7    }
8  }
9
10 resource "aws_s3_bucket" "my-bucket-us-east-1" {
11
12   bucket = "bhushan-kor-aws-bucket-terraform"
13   tags = {
14     Name="Sample Bucket"
15   }
16 }
```

Step 8: Now open new terminal and cd to S3 And perform the following commands

- 1)terraform init
- 2)terraform plan
- 3)terraform apply

The screenshot shows a terminal window with the following content:

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

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

C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts>cd S3
```

```
C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts\S3>terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding hashicorp/aws versions matching "5.64.0"...

- Installing hashicorp/aws v5.64.0...

- Installed hashicorp/aws v5.64.0 (signed by HashiCorp)

Terraform has created a lock file `.terraform.lock.hcl` to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

```
C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts\S3>terraform plan
```

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

Terraform will perform the following actions:

```
# aws_s3_bucket.my-bucket-us-east-1 will be created
+ resource "aws_s3_bucket" "my-bucket-us-east-1" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "bhushan-Kor-aws-bucket-terraform"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy            = false
    + hosted_zone_id           = (known after apply)
    + id                       = (known after apply)
    + object_lock_enabled       = (known after apply)
    + policy                   = (known after apply)
    + region                   = (known after apply)
    + request_payer             = (known after apply)
    + tags                     = {
        + "Name" = "Sample Bucket"
    }
    + tags_all                 = {
        + "Name" = "Sample Bucket"
    }
    + website_domain           = (known after apply)
    + website_endpoint          = (known after apply)

    + cors_rule (known after apply)
    + grant (known after apply)
    + lifecycle_rule (known after apply)
}
```

```
+ lifecycle_rule (known after apply)
+ logging (known after apply)
+ object_lock_configuration (known after apply)
+ replication_configuration (known after apply)
+ server_side_encryption_configuration (known after apply)
+ versioning (known after apply)
+ website (known after apply)
}
```

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

```
C:\Users\bhush\OneDrive\Desktop\Terraform Scripts\s3>terraform apply
```

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

Terraform will perform the following actions:

```
# aws_s3_bucket.my-bucket-us-east-1 will be created
+ resource "aws_s3_bucket" "my-bucket-us-east-1" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "bhushan-kor-aws-bucket-terraform"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy              = false
    + hosted_zone_id            = (known after apply)
    + id                        = (known after apply)
    + object_lock_enabled        = (known after apply)
    + policy                     = (known after apply)
    + region                     = (known after apply)
    + request_payer              = (known after apply)
    + tags                      = {
        + "Name" = "Sample Bucket"
    }
    + tags_all                  = {
        + "Name" = "Sample Bucket"
    }
    + website_domain            = (known after apply)
    + website_endpoint           = (known after apply)

    + cors_rule (known after apply)
    + grant (known after apply)
```

```
+ cors_rule (known after apply)
+ grant (known after apply)
+ lifecycle_rule (known after apply)
+ logging (known after apply)
+ object_lock_configuration (known after apply)
+ replication_configuration (known after apply)
+ server_side_encryption_configuration (known after apply)
+ versioning (known after apply)
+ website (known after apply)
+ website (known after apply)
}
+ website (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
aws_s3_bucket.my-bucket-us-east-1: Creating...
aws_s3_bucket.my-bucket-us-east-1: Creation complete after 8s [id=bhushan-kor-aws-bucket-terraform]
```

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

```
+ website (known after apply)
}
```

```
+ website (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
aws_s3_bucket.my-bucket-us-east-1: Creating...
aws_s3_bucket.my-bucket-us-east-1: Creation complete after 8s [id=bhushan-kor-aws-bucket-terraform]
```

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

```
C:\Users\bhush\OneDrive\Desktop\Terraform Scripts\S3>
+ website (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
aws_s3_bucket.my-bucket-us-east-1: Creating...
aws_s3_bucket.my-bucket-us-east-1: Creation complete after 8s [id=bhushan-kor-aws-bucket-terraform]
```

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

```
C:\Users\bhush\OneDrive\Desktop\Terraform Scripts\S3>
```

```
+ website (known after apply)
```

```
+ website (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_s3_bucket.my-bucket-us-east-1: Creating...

aws_s3_bucket.my-bucket-us-east-1: Creation complete after 8s [id=bhushan-kor-aws-bucket-terraform]

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

C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts\S3>

```
}
```

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

aws_s3_bucket.my-bucket-us-east-1: Creating...

aws_s3_bucket.my-bucket-us-east-1: Creation complete after 8s [id=bhushan-kor-aws-bucket-terraform]

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

C:\Users\bhush\one drive 2\OneDrive\Desktop\Terraform Scripts\S3>

Now If you Click on AWS on lab it will open your AWS account and in S3 you can see you bucket.

Before terrafrom Apply:

The screenshot shows the AWS S3 console interface. On the left, there's a navigation sidebar with links like 'Amazon S3', 'Buckets', 'Access Grants', 'Access Points', etc. The main area is titled 'Amazon S3' and shows a 'General purpose buckets' section. It lists five buckets: 'bhushan2', 'elasticbeanstalk-us-east-1-847302770411', 'exp6bucket', 's3bucketexp6', and 'staticwebhosting28'. Each entry includes details like 'AWS Region' (US East (N. Virginia) us-east-1), 'IAM Access Analyzer' (with a link to view it), and 'Creation date'. A 'Create bucket' button is also visible at the top of this list. At the bottom of the page, there's a 'Feature spotlight' section and a note about the AWS Marketplace for S3.

Name	AWS Region	IAM Access Analyzer	Creation date
bhushan2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:17:09 (UTC+05:30)
elasticbeanstalk-us-east-1-847302770411	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 5, 2024, 14:00:50 (UTC+05:30)
exp6bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 19:57:25 (UTC+05:30)
s3bucketexp6	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:08:51 (UTC+05:30)
staticwebhosting28	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 4, 2024, 23:53:54 (UTC+05:30)

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

After terraform apply:

The screenshot shows the AWS S3 console with the 'General purpose buckets' tab selected. There are six buckets listed:

Name	AWS Region	IAM Access Analyzer	Creation date
bhushan-kor-aws-bucket-terraform	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 25, 2024, 12:12:05 (UTC+05:30)
bhushan2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:17:09 (UTC+05:30)
elasticeanstalk-us-east-1-847302770411	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 5, 2024, 14:00:50 (UTC+05:30)
exp6bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 19:57:25 (UTC+05:30)
s3bucketexp6	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:08:51 (UTC+05:30)
staticwebhosting28	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 4, 2024, 23:33:54 (UTC+05:30)

Step 9: Now to destroy the bucket run command terraform destroy.

```
C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts\S3>terraform destroy
aws_s3_bucket.my-bucket-us-east-1: Refreshing state... [id=bhushan-kor-aws-bucket-terraform]

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:

# aws_s3_bucket.my-bucket-us-east-1 will be destroyed
- resource "aws_s3_bucket" "my-bucket-us-east-1" {
    - arn = "arn:aws:s3:::bhushan-kor-aws-bucket-terraform" -> null
    - bucket = "bhushan-kor-aws-bucket-terraform" -> null
    - bucket_domain_name = "bhushan-kor-aws-bucket-terraform.s3.amazonaws.com" -> null
    - bucketRegionalDomainName = "bhushan-kor-aws-bucket-terraform.s3.us-east-1.amazonaws.com" -> null
    - force_destroy = false -> null
    - hostedZoneId = "Z3AQ0B5TGFYJSTF" -> null
    - id = "bhushan-kor-aws-bucket-terraform" -> null
    - objectLockEnabled = false -> null
    - region = "us-east-1" -> null
    - requestPayer = "BucketOwner" -> null
    - tags = {
        - "Name" = "Sample Bucket"
    } -> null
    - tags_all = {
        - "Name" = "Sample Bucket"
    } -> null
    # (3 unchanged attributes hidden)

    - grant {
        - id = "778043bd0e67860760caebd7a1a61d745d8798fa35ab31144e54d7003ee08ae8" -> null
        - permissions = [
            - "FULL_CONTROL",
        ] -> null
        - type = "CanonicalUser" -> null
        # (1 unchanged attribute hidden)
    }
}
```

```

    - "FULL_CONTROL",
  ] -> null
- type      = "CanonicalUser" -> null
# (1 unchanged attribute hidden)
}

- server_side_encryption_configuration {
  - rule {
    - bucket_key_enabled = false -> null

    - apply_server_side_encryption_by_default {
      - sse_algorithm      = "AES256" -> null
      # (1 unchanged attribute hidden)
    }
  }
}

- versioning {
  - enabled      = false -> null
  - mfa_delete   = false -> null
}
}

```

Plan: 0 to add, 0 to change, 1 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

```
aws_s3_bucket.my-bucket-us-east-1: Destroying... [id=bhushan-kor-aws-bucket-terraform]
aws_s3_bucket.my-bucket-us-east-1: Destruction complete after 2s
```

Destroy complete! Resources: 1 destroyed.

```
C:\Users\bhush\OneDrive\OneDrive\Desktop\Terraform Scripts\S3>
```

After terraform destroy :

Name	AWS Region	IAM Access Analyzer	Creation date
bhushan2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:17:09 (UTC+05:30)
elasticbeanstalk-us-east-1-847302770411	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 5, 2024, 14:00:50 (UTC+05:30)
exp6bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 19:57:25 (UTC+05:30)
s3bucketexp6	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 20:08:51 (UTC+05:30)
staticwebhosting28	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 4, 2024, 23:33:54 (UTC+05:30)

Step 10: Congratulations we are done with creating and destroying the S3 bucket on AWS using Terraform.

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

Theory :**1. Static Application Security Testing (SAST)**

- **Definition:** SAST is a method of debugging by examining source code before a program is run. It identifies vulnerabilities early in the development lifecycle.
- **Key Features:**
 - **Early Detection:** Finds vulnerabilities during the coding phase, reducing remediation costs.
 - **Code Quality Analysis:** Beyond security, it also assesses code quality, maintainability, and adherence to coding standards.
 - **Integration:** Can be integrated into CI/CD pipelines for continuous security assessment.

2. SonarQube

- **Definition:** SonarQube is an open-source platform for continuous inspection of code quality, which includes detecting bugs, vulnerabilities, and code smells.
- **SAST Integration:** Supports SAST tools to analyze code and provide metrics and reports within the SonarQube dashboard.
- **Quality Gates:** Allows setting thresholds (quality gates) that must be met before code can proceed in the CI/CD pipeline.

3. Reporting and Remediation

- **Results Analysis:** After SAST scans, results are usually presented in a detailed report highlighting vulnerabilities, their severity, and remediation advice.
- **Feedback Loop:** Integrating SAST results into the development workflow helps create a feedback loop, encouraging developers to address vulnerabilities proactively.

4. Best Practices

- **Regular Updates:** Keep SAST tools and configurations updated to recognize the latest vulnerabilities.
- **Customization:** Tailor SAST rules and configurations to suit the specific needs of the project and team.
- **Training:** Ensure developers are trained on security best practices to understand and mitigate vulnerabilities effectively.

Prerequisites:

- Jenkins installed
- Docker Installed (for SonarQube)

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

The screenshot shows the Jenkins dashboard with the following details:

- Left Sidebar:**
 - New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
- Build Queue:** No builds in the queue.
- Build Executor Status:**
 - Built-In Node: 1 Idle, 2 Idle
 - Bhushan Exp_7 Node: (offline)
 - My Node: (offline)
- Central View:** A table showing Jenkins jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Bhushan_EXP_6_Pipeline	24 days #1	N/A	11 sec
✓	☀️	BhushanKor	24 days #5	N/A	1.5 sec
✓	☀️	Devops	1 mo 13 days #7	N/A	15 sec
...	☀️	Exp_6_Job	N/A	N/A	N/A
✗	☁️	Exp_6_Job_Maven	N/A	23 days #6	21 sec
✗	☁️	test	N/A	24 days #2	0.25 sec
✓	☁️	Test2	24 days #3	24 days #2	0.27 sec
- Bottom:** Icons for S, M, L and a three-dot menu.

Step 2: Run SonarQube in a Docker container using this command

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

```
C:\Users\bhush\OneDrive\2\OneDrive\Desktop\Docker>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59bed36c86
Status: Downloaded newer image for sonarqube:latest
dc00d2f31a229b2076f15c273ad750e45b74f871dd53b6d177b6f860d4fc8f0e

C:\Users\bhush\OneDrive\2\OneDrive\Desktop\Docker>
```

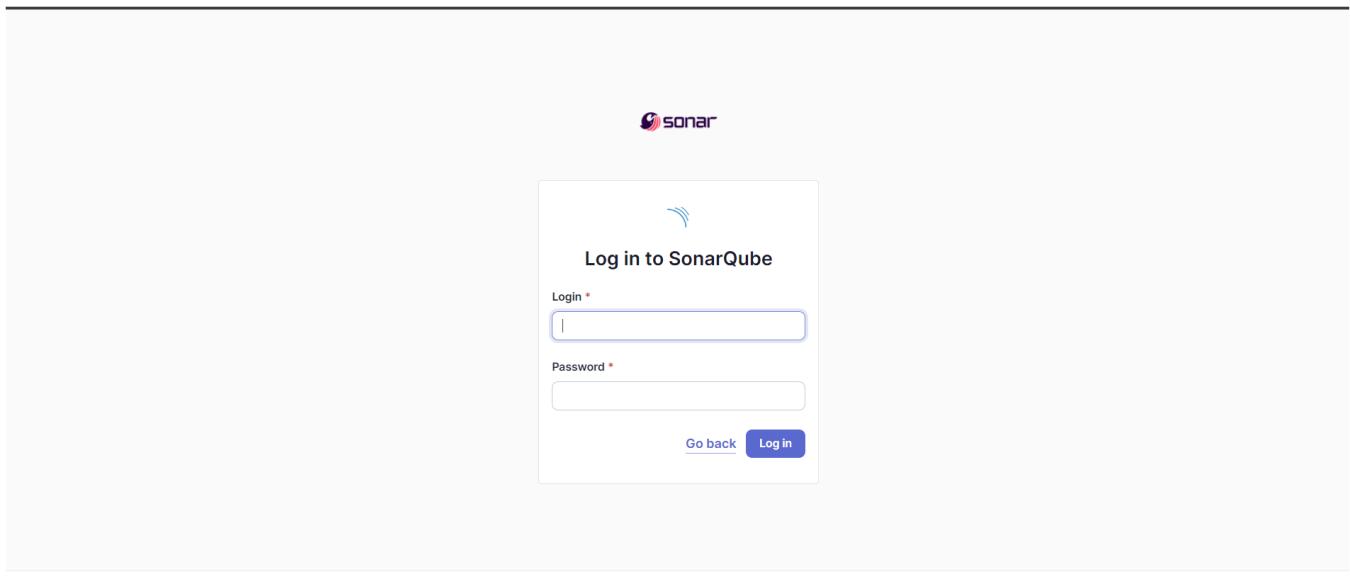
Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

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



SonarQube™ technology is powered by [SonarSource SA](#)

LGPL v3 [Community](#) [Documentation](#) [Plugins](#)

Start 4: Login to SonarQube using username admin and password admin.

A screenshot of the SonarQube "Create a Project" page. The top navigation bar includes links for "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", "More", and a search icon. A sidebar on the left lists "Recent Projects" and "My Projects". The main content area is titled "How do you want to create your project?". It features several import options: "Import from Azure DevOps", "Import from Bitbucket Cloud", "Import from Bitbucket Server", "Import from GitHub", and "Import from GitLab", each with a "Setup" button. Below these is a section for local projects with a "Create a local project" button. A note at the bottom states: "Are you just testing or have an advanced use-case? Create a local project." A yellow warning box at the bottom left says: "⚠️ 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."

SonarQube™ technology is powered by [SonarSource SA](#)

Community Edition v10.6 (92116) ACTIVE [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 5: Create a manual project in SonarQube with any Name

The screenshot shows the first step of creating a local project in SonarQube. The page title is "Create a local project". It has fields for "Project display name" (Bhushan's SonarQube), "Project key" (Bhushan-s-SonarQube), and "Main branch name" (main). A note says "The name of your project's default branch [Learn More](#)". Buttons for "Cancel" and "Next" are at the bottom.

1 of 2
Create a local project
Project display name *
Bhushan's SonarQube
Project key *
Bhushan-s-SonarQube
Main branch name *
main
The name of your project's default branch [Learn More](#)
Cancel Next

The screenshot shows the second step of setting up the project for "Clean as You Code". It asks to choose a baseline for new code. The "Use the global setting" option is selected. Other options include "Previous version" (recommended for regular releases), "Number of days" (recommended for continuous delivery), and "Reference branch" (recommended for feature branches). A note says "Any code that has changed since the previous version is considered new code." and "Any code that has changed in the last x days is considered new code." Buttons for "Back" and "Create project" are at the bottom.

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.
SonarQube™ technology is powered by SonarSource SA [Community Edition v10.6 \(92116\) ACTIVE](#) [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

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 6: Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "sonar". Below the search bar, there are tabs for "Updates" (17), "Available plugins", "Installed plugins" (selected), and "Advanced settings". A search result for "SonarQube Scanner for Jenkins 2.17.2" is displayed, with a description stating: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." An "Enabled" switch is turned on, and a red "Uninstall" button is visible.

Step 7: Under Jenkins 'Configure System', look for SonarQube Servers and enter the details. Enter the Server Authentication token if needed.

The screenshot shows the Jenkins System configuration page under "SonarQube servers". It includes fields for "Name" (Bhushan's Server), "Server URL" (http://localhost:9000), and "Server authentication token" (set to "- none -"). There is also an "Advanced" dropdown. At the bottom are "Save" and "Apply" buttons.

Step 8: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

The screenshot shows the Jenkins Tools configuration page under "SonarQube Scanner installations". It shows a configuration for "Bhushan's Scanner" with the "Install automatically" checkbox checked. A sub-section for "Install from Maven Central" shows the version "SonarQube Scanner 6.1.0.4477". At the bottom, there is an "Add Installer" button and a "Saved" confirmation message.

Step 9: After the configuration, create a New Item in Jenkins, and choose a freestyle project.

New Item

Enter an item name
Bhushan's SonarQube

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.

OK

Step 10: Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git. It is a sample hello-world project with no vulnerabilities and issues, just to test integration.

Configure

Source Code Management

General

Source Code Management

Git

Repositories

Repository URL
https://github.com/BhushanVestLabs/MSBuild_firstproject.git

Credentials
- none -

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')
*/master

Save Apply

Step 11: Under Build select Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

The screenshot shows the Jenkins configuration interface for a job named "Bhushan's SonarQube".

Configure Section:

- General:** Poll SCM
- Source Code Management:** None
- Build Triggers:** None
- Build Environment:** Selected
- Build Steps:** None
- Post-build Actions:** None

Build Environment Configuration:

A modal window titled "Configure" is open, showing the "Build Environment" section. It lists various build steps, with "Execute SonarQube Scanner" highlighted. Other options include "Execute Windows batch command", "Execute shell", "Invoke Ant", "Invoke Gradle script", "Invoke top-level Maven targets", "Run with timeout", "Set build status to "pending" on GitHub commit", "SonarScanner for MSBuild - Begin Analysis", and "SonarScanner for MSBuild - End Analysis".

Post-build Actions:

An "Add post-build action" dropdown menu is visible.

Save and Apply Buttons:

Save and Apply buttons are located at the bottom of the configuration screen.

Build Steps Section:

The "Build Steps" section is expanded, showing the "Configure" tab. A single "Execute SonarQube Scanner" step is listed.

Execute SonarQube Scanner Step Configuration:

- JDK:** (Inherit From Job)
- Path to project properties:** None
- Analysis properties:**

```
sonar.projectKey=Bhushan-s-SonarQube
sonar.login=admin
sonar.sources=
sonar.host.url=http://localhost:9000
```
- Additional arguments:** None
- JVM Options:** None

Save and Apply Buttons:

Save and Apply buttons are located at the bottom of the build steps configuration screen.

```
sonar.projectKey=<Your Project Key>
sonar.login=<User Name>
sonar.password=<Password>
sonar.sources=.
sonar.host.url=http://localhost:9000
```

Step 12: In the SonarQube go to Security then for Administrator allow Administer system and Execute analysis.

	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 DEPRECATED 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
Administrator admin	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects

4 of 4 shown

Step 13: See the Console Output.

```

Started by user unknown or anonymous
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\Bhushan's SonarQube
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\Bhushan's SonarQube.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/BhushanVesitLabs/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/BhushanVesitLabs/MSBuild_firstproject.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/BhushanVesitLabs/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse --refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c04cd672427c380bcacee6d6fee7b49adf (refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f f2bc042c04cd672427c380bcacee6d6fee7b49adf # timeout=10
Commit message: "updated"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk f2bc042c04cd672427c380bcacee6d6fee7b49adf # timeout=10
[Bhushan's SonarQube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\Bhushan_s_Scanner\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=Bhushan-SonarQube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=Bhushan -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\Bhushan's SonarQube
21:01:37.182 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
21:01:37.224 INFO Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\Bhushan_s_Scanner\bin..\conf\sonar-scanner.properties
21:01:37.231 INFO Project root configuration file: NONE
21:01:37.316 INFO SonarScanner CLI 6.1.0.4427
21:01:37.316 INFO Java 22.0.1 Oracle Corporation (64-bit)
21:01:37.333 INFO Windows 11 10.0 amd64
21:01:37.447 INFO User cache: C:\WINDOWS\system32\config\systemprofile\.sonar\cache
21:01:42.365 INFO JRE provisioning: os\windows, arch\amd64
21:03:06.440 INFO Communicating with SonarQube Server 10.6.0.92116
21:03:08.434 INFO Starting SonarScanner Engine...
21:03:08.434 INFO Java 17.0.11 Eclipse Adoption (64-bit)
21:03:11.123 INFO Load global settings
21:03:16.628 INFO Load global settings (done) | time=5473ms
21:03:16.634 INFO Server id: 1478411E-AZIKhptfueCLG3yP00pN
21:03:16.640 INFO Loading required plugins

```

Dashboard > Bhushan's SonarQube > #2 > Console Output

```
* Any directory in the file path has a name ending in "test" or "tests"

21:09:00.357 INFO Using git CLI to retrieve untracked files
21:09:00.653 INFO Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
21:09:00.889 INFO 14 source files to be analyzed
21:09:03.146 INFO 14/14 source files have been analyzed
21:09:03.151 INFO Sensor TextAndSecretsSensor [text] (done) | time=5027ms
21:09:03.165 INFO -----
21:09:03.165 INFO Sensor C# [csharp]
21:09:03.370 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
21:09:03.370 INFO Sensor C# [csharp] (done) | time=0ms
21:09:03.370 INFO Sensor Analysis Warnings Import [csharp]
21:09:03.370 INFO Sensor Analysis Warnings Import [csharp] (done) | time=9ms
21:09:03.370 INFO Sensor C# File Caching Sensor [csharp]
21:09:03.376 INFO Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
21:09:03.376 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
21:09:03.376 INFO Sensor Zero Coverage Sensor
21:09:03.400 INFO Sensor Zero Coverage Sensor (done) | time=26ms
21:09:03.406 INFO SCM Publisher SCM provider for this project is: git
21:09:03.408 INFO SCM Publisher 4 source files to be analyzed
21:09:06.271 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=2859ms
21:09:06.271 INFO CPD Executor Calculating CPD for 0 files
21:09:06.271 INFO CPD Executor CPD calculation finished (done) | time=0ms
21:09:06.308 INFO SCM revision ID: "f2bc042c04ce72427c380bcae6df6fe7049adf"
21:09:12.497 INFO Analysis report generated in 494ms, dir size=201.1 kB
21:09:12.727 INFO Analysis report compressed in 60ms, zip size=22.3 kB
21:09:23.622 INFO Analysis report uploaded in 10889ms
21:09:23.629 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=Bhushan-s-SonarQube
21:09:23.631 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:09:23.631 INFO More about the report processing at http://localhost:9000/api/ce/task?id=a6525be8-4189-4c6d-8aca-a6d0b9bb7551
21:09:23.663 INFO Analysis total time: 5:02.251 s
21:09:23.676 INFO SonarScanner Engine completed successfully
21:09:23.925 INFO EXECUTION SUCCESS
21:09:24.066 INFO Total time: 7:46.713s
Finished: SUCCESS
```

Jenkins

Dashboard > Bhushan's SonarQube >

Status: ✓ Bhushan's SonarQube

✓ SonarQube

Permalink: [SonarQube](#)

Changes: ✓ Status

Build Now: ✓ Changes

Configure: ✓ Workspace

Delete Project: ✓ Build Now

SonarQube: ✓ Configure

Rename: ✓ Delete Project

Add description: ✓ SonarQube

Build History: ✓ Rename

Build History: trend ▾

Filter... /

Build #2: Sep 19, 2024, 9:01PM

Build #1: Sep 19, 2024, 8:52PM

Atom feed for all Atom feed for failures

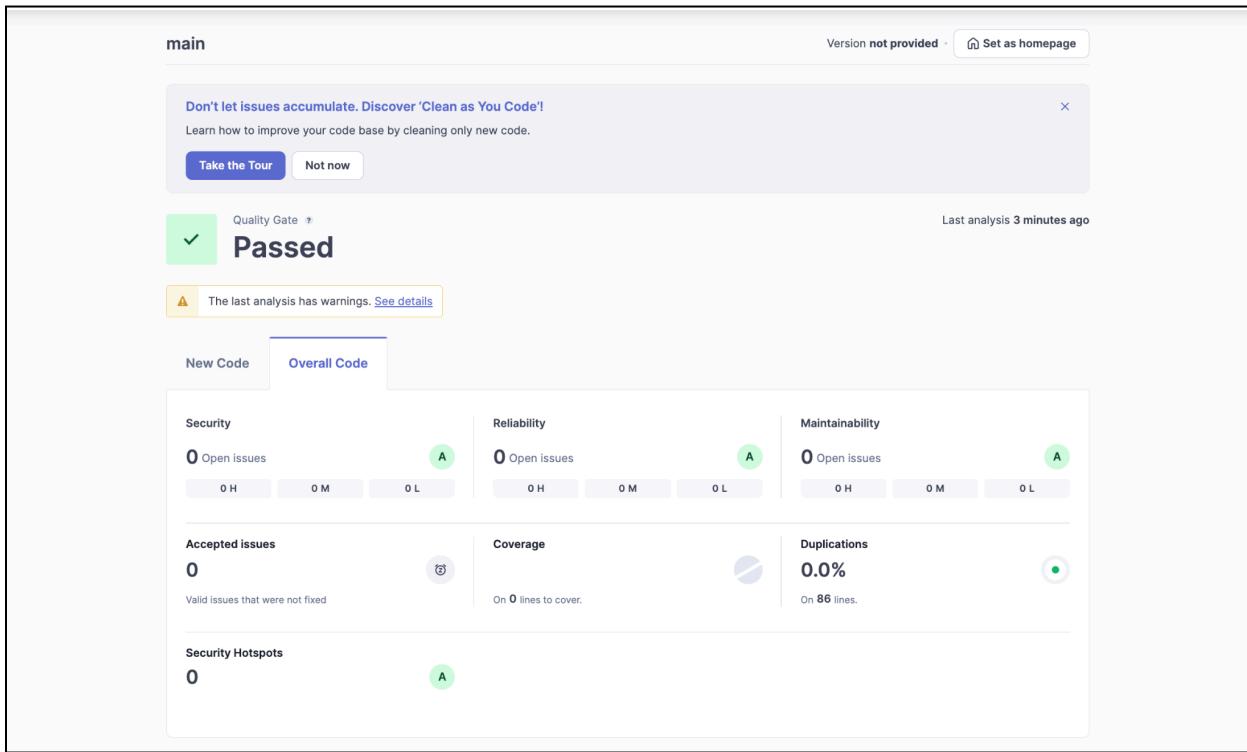
Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 14: Now See the SonarQube Project.



Conclusion :

In this experiment, we have learned how to perform Jenkins SAST using SonarQube. For this, we used a docker image of SonarQube to not install it locally on our system. After installing the required configurations on Jenkins, using a code from a GitHub repository, we analyze its code using SonarQube. Once we build the project, we can see that the SonarQube project displays that the code has no errors.

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

Theory:

What is SAST?

Static Application Security Testing (SAST) is a methodology that analyzes source code to identify security vulnerabilities before compilation. It is often referred to as white box testing and helps developers detect issues early in the software development lifecycle (SDLC).

Problems SAST Solves

1. **Early Detection:** Identifies vulnerabilities in the initial development stages, reducing later risks.
2. **Real-Time Feedback:** Provides immediate insights, allowing developers to fix issues before moving forward.
3. **Code Navigation:** Offers visual representations of vulnerabilities for easier code understanding.
4. **Guidance on Fixes:** Suggests specific remediation steps without requiring deep security expertise.
5. **Comprehensive Coverage:** Analyzes the entire codebase quickly, outperforming manual reviews.
6. **Regular Scanning:** Ensures continuous security assessment through scheduled scans during builds or releases.

Importance of SAST

- **Resource Efficiency:** Automates code reviews, addressing the resource gap between developers and security staff.
- **Speed:** Processes millions of lines of code in minutes, identifying critical vulnerabilities.
- **Proactive Security:** Integrates security into the development process, preventing vulnerabilities from being overlooked.

What is a CI/CD Pipeline?

A **CI/CD Pipeline** refers to Continuous Integration and Continuous Delivery, automating software development tasks. It includes stages such as coding, building, testing, and deploying, ensuring each step is completed sequentially for efficient releases.

What is SonarQube?

SonarQube is an open-source platform for continuous code quality inspection. It performs static code analysis to generate reports on bugs, vulnerabilities, and code duplications across various programming languages.

Benefits of SonarQube

- **Sustainability:** Optimizes application lifecycle by reducing complexity and vulnerabilities.
- **Increased Productivity:** Minimizes maintenance efforts and costs.
- **Quality Control:** Integrates code quality checks into development.
- **Error Detection:** Alerts developers to fix issues before release.
- **Scalability:** Supports multiple projects without restrictions.
- **Skill Enhancement:** Provides regular feedback to improve developer skills.

Prerequisites:

- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

Download The SonarQube CLI according to your system :

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

The screenshot shows the SonarScanner CLI documentation page. At the top, there's a navigation bar with the SonarQube logo, "Docs 10.6", and a search bar. Below the navigation, there's a sidebar with links like "Homepage", "Try out SonarQube", "Server installation and setup", "Analyzing source code", "Scanners", "SonarScanner CLI", and "SonarScanner for NPM". The main content area has a title "SonarScanner CLI" and a sub-section "6.2". It includes a table with "SonarScanner" and "Issue Tracker" tabs, showing a release date of "2024-09-17" and a note about OpenSSL support. Below the table, there's a paragraph about the SonarScanner CLI being the scanner to use when no specific scanner is available. A callout box provides a tip about verifying code checkout. On the right side, there's a sidebar titled "On this page" with links to various configuration and troubleshooting topics.

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

The screenshot shows the Jenkins dashboard with the following details:

- Left Sidebar:**
 - New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
- Build Queue:** No builds in the queue.
- Build Executor Status:**
 - Built-In Node: 1 Idle, 2 Idle
 - Bhushan_Exp_7_Node: (offline)
 - My_Node: (offline)
- Central View:** A table listing Jenkins jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Bhushan_EXP_6_Pipeline	24 days #1	N/A	11 sec
✓	☀️	BhushanKor	24 days #5	N/A	1.5 sec
✓	☀️	Devops	1 mo 13 days #7	N/A	15 sec
...	☀️	Exp_6_Job	N/A	N/A	N/A
✗	☁️	Exp_6_Job_Maven	N/A	23 days #6	21 sec
✗	☁️	test	N/A	24 days #2	0.25 sec
✓	☁️	Test2	24 days #3	24 days #2	0.27 sec
- Bottom:** Icons for S, M, L and a three-dot menu.

Step 2: Run SonarQube in a Docker container using this command

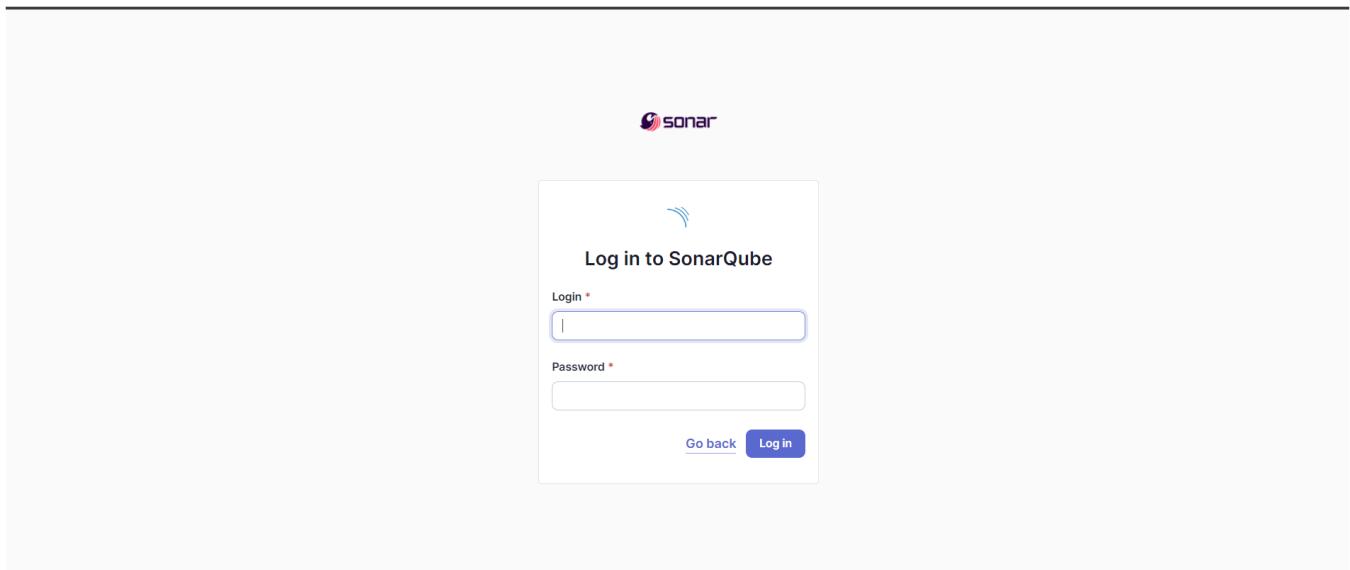
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

```
Microsoft Windows [Version 10.0.22621.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\INFT505-11>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
d72b183b1866cea7ecdb976a63dfe521172c307eb45eace7b769f726f0bbf989

C:\Users\INFT505-11>
```

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



Start 4: Login to SonarQube using username admin and password admin.

A screenshot of the SonarQube interface showing the "Create a new project" section. At the top, there is a navigation bar with tabs: "sonarqube", "Projects" (which is selected), "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", "More", and a search icon. Below the navigation bar, the main content area has a heading "How do you want to create your project?". It asks if the user wants to benefit from SonarQube's features like repository import and Pull Request decoration, and suggests creating a project from a favorite DevOps platform. It then asks if the user needs to set up a DevOps platform configuration. There are six "Import" buttons arranged in a grid: "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, there is a note about testing or advanced use-cases, followed by a "Create a local project" button. At the bottom of the page, there is a yellow warning box with the text "Embedded database should be used for evaluation purposes only" and a note that it will not scale, support newer versions, or allow migration to a different database engine. The footer includes links to "SonarQube™ technology is powered by SonarSource SA", "Community Edition v10.6 (92116) ACTIVE", "LGPL v3", "Community", "Documentation", "Plugins", and "Web API".

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 5: Create a manual project in SonarQube with any Name

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](#)

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

⚠️ 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.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

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 6: Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "sonar". Below the search bar, there are four tabs: "Updates" (17), "Available plugins", "Installed plugins", and "Advanced settings". The "Installed plugins" tab is selected. A list of installed plugins is displayed, with "SonarQube Scanner for Jenkins 2.17.2" being the first item. This plugin is described as allowing an easy integration of SonarQube, an open source platform for Continuous Inspection of code quality. The "Enabled" switch next to the plugin is turned on, indicated by a blue circle with a white checkmark. The status of the plugin is "Enabled".

Step 7: Under Jenkins 'Configure System', look for SonarQube Servers and enter the details. Enter the Server Authentication token if needed.

The screenshot shows the Jenkins System configuration page under "Manage Jenkins > System". The section is titled "SonarQube servers". It includes fields for "Environment variables" (unchecked) and "SonarQube installations". A "Name" field contains "Bhushan's Server", a "Server URL" field contains "http://localhost:9000", and a "Server authentication token" dropdown menu is set to "- none -". An "Advanced" button is visible. At the bottom are "Save" and "Apply" buttons.

Step 8: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

The screenshot shows the Jenkins Tools configuration page under "Manage Jenkins > Tools". The section is titled "SonarQube Scanner installations". A "Name" field contains "Bhushan's Scanner". The "Install automatically" checkbox is checked. A "Version" dropdown menu is set to "SonarQube Scanner 6.1.0.4477". An "Add Installer" button is visible. At the bottom are "Add SonarQube Scanner" and "Ant installations" buttons. A green success message "Saved" is displayed at the bottom left.

Step 9: Now click on the new item and select the pipeline project and give name.

New Item

Enter an item name

Bhushan's Pipeline

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

OK

Step 10: Under the scripts add the following script. **(Do not forget to replace details with your details.)**

```
node {
    stage('Cloning the GitHub Repo'){
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat "<your bin file location of SonarQube CLI>"+
                "-D sonar.login=<your user name>"+
                "-D sonar.password=<your password>"+
                "-D sonar.projectkey=<your projectkey>"+
                "-D sonar.exclusions=vendor/**,resources/**,*/*.java "+
                "-D sonar.host.url=http://localhost:9000/"+
        }
    }
}
```

Dashboard > Bhushan's Pipeline > Configuration

Configure

General

Advanced Project Options

Pipeline

Definition

Pipeline script

Script

```
1> node {
2>     stage('Cloning the GitHub Repo'){
3>         git 'https://github.com/shazforiot/GOL.git'
4>     }
5>     stage('SonarQube analysis') {
6>         withSonarQubeEnv('sonarqube') {
7>             bat "<your bin file location of SonarQube CLI>"+
8>                 "-D sonar.login=<your user name>"+
9>                 "-D sonar.password=<your password>"+
10>                 "-D sonar.projectkey=<your projectkey>"+
11>                 "-D sonar.exclusions=vendor/**,resources/**,*/*.java "+
12>                 "-D sonar.host.url=http://localhost:9000/"+
13>             }
14>         }
15>     }
16> }
```

Use Groovy Sandbox

Pipeline Syntax

Save Apply

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

Step 11: Go back to jenkins. Go to the job you had just built and click on Build Now.

The screenshot shows the Jenkins dashboard for the 'Bhushan's Pipeline' project. The pipeline status is green with a checkmark, indicating it is healthy. It lists seven builds, with the most recent being build #2, which was successful 7 days and 23 hours ago. On the left, there is a sidebar with various project management options like Status, Changes, Workspace, and Build Now. On the right, there is a 'Permalinks' section with a SonarQube integration icon and a list of build history items.

Step 12: Check the console output.

The screenshot shows the Jenkins console output for build #11 of the 'Bhushan's Pipeline' project. The output log shows the following steps:

```

Started by user Bhushan
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Bhushan's Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] {
(Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Bhushan's Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a461232b0412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a461232b0412c5e6e1e5e4 # timeout=10

```

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 13: Once the build is complete, go back to SonarQube and check the project linked.

The screenshot shows the SonarQube interface with the 'Projects' tab selected. A sidebar on the left contains filters for 'Quality Gate' (Passed: 1, Failed: 0), 'Reliability' (Security: 0, Reliability: 0, Maintainability: 1), and 'Security' (0). The main area displays the project 'Bhushan's Pipeline' with a status of 'Passed'. Key metrics shown are: Security (0), Reliability (68k), Maintainability (164k), Hotspots Reviewed (0.0%), Coverage (50.6%), and Duplications (50.6%). A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale. It will not support connections to a remote instance of SonarQube, and there is no support for migrating your data out of it into a different database vendor.'

The screenshot shows the SonarQube interface with the 'main' project selected. The top navigation bar includes 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. The 'Overview' tab is active. The main content area displays the project 'main' with a 'Quality Gate' status of 'Passed'. It shows 683k Lines of Code, a new analysis in progress, and various metrics: Security (0 Open Issues), Reliability (68k Open Issues), Maintainability (164k Open Issues), Accepted issues (0), Coverage (0%), and Duplications (50.6%). A note at the bottom states: 'Last analysis 10 minutes ago. Set as homepage.'

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Click on Issues and see the different Issues.

Codesmell:

The screenshot shows the SonarQube interface for the 'Bhushan's Pipeline' project. The 'Issues' tab is selected. On the left, a sidebar displays project statistics: Security (0), Reliability (21k), and Maintainability (164k). Under 'Type', 'Code Smell' is selected, showing 164k issues. The main panel lists three specific code smell issues found in 'gameoflife-acceptance-tests/Dockerfile': 1) 'Use a specific version tag for the image.' (Severity: Not assigned, Intentionality: Maintainability, L1 effort: 5min, 4 years ago, Major). 2) 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (Severity: Not assigned, Intentionality: Maintainability, L1 effort: 5min, 4 years ago, Major). 3) 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (Severity: Not assigned, Intentionality: Maintainability, L1 effort: 5min, 4 years ago, Major). A footer note states: 'localhost:9000/security_hotspots?id=Bhushan's+Pipeline for evaluation purposes only'.

Bugs:

The screenshot shows the SonarQube interface for the 'Bhushan's Pipeline' project. The 'Issues' tab is selected. On the left, a sidebar displays project statistics: Security (0), Reliability (47k), and Maintainability (0). Under 'Type', 'Bug' is selected, showing 47k issues. The main panel lists three specific bug issues found in 'gameoflife-core/build/reports/tests/all-tests.html': 1) 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Severity: Not assigned, Intentionality: Reliability, Consistency: user-experience, L1 effort: 5min, 4 years ago, Major). 2) 'Add "lang" and/or "xml:lang" attributes to this "html" element' (Severity: Not assigned, Intentionality: Reliability, accessibility: wcag2-a, L1 effort: 2min, 4 years ago, Major). 3) 'Add "<th>" headers to this "<table>".' (Severity: Not assigned, Intentionality: Reliability, accessibility: wcag2-a, L9 effort: 2min, 4 years ago, Major). A footer note states: 'Embedded database should be used for evaluation purposes only'.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

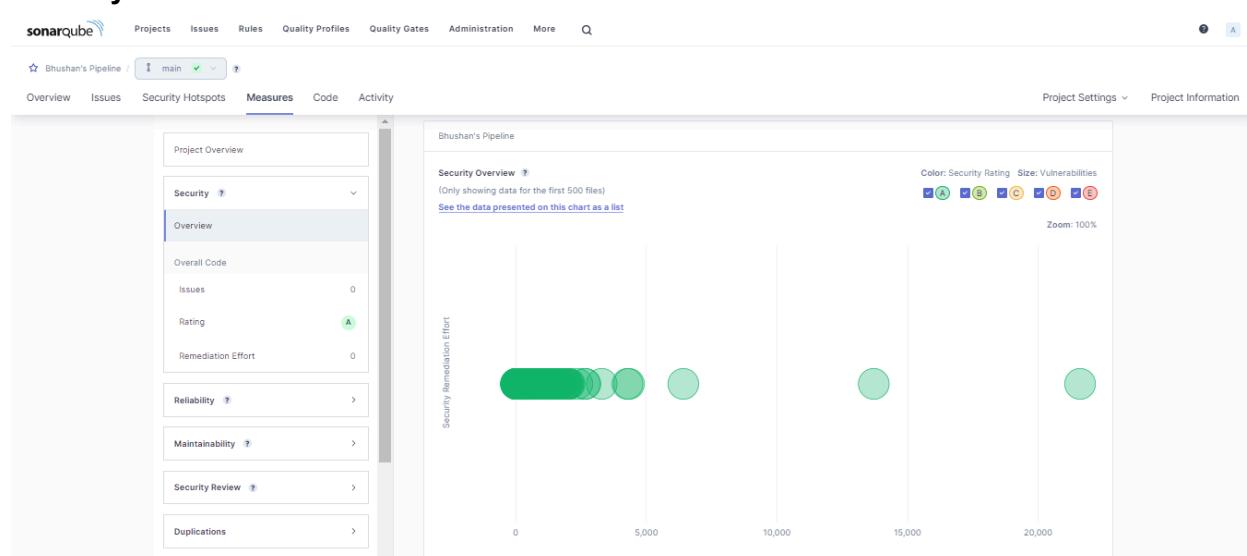
Division: D15C

Roll No: 28

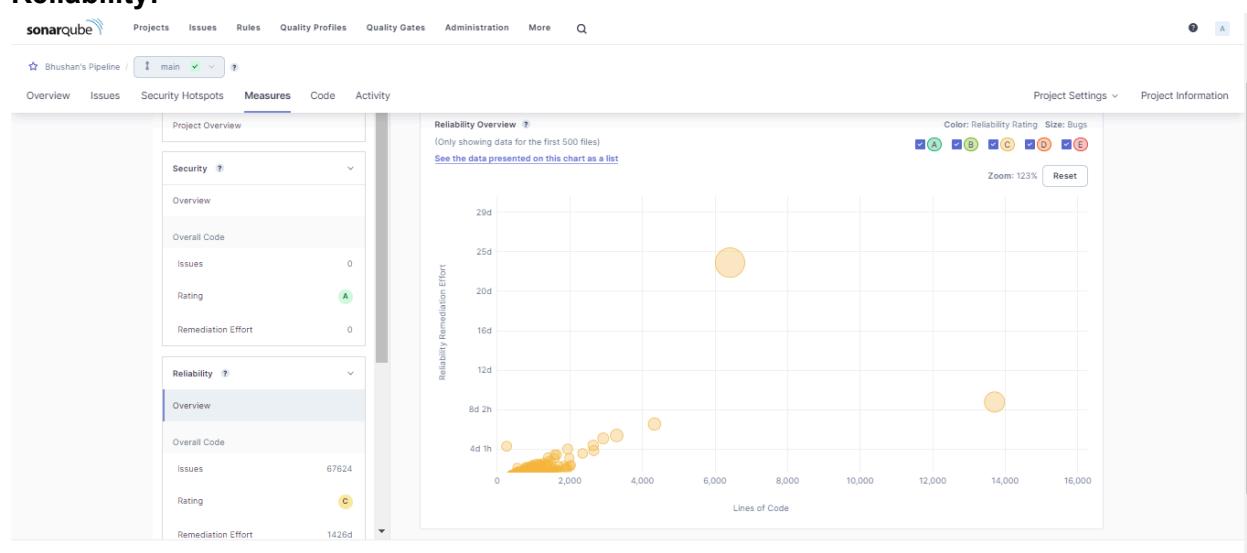
Click on Security hotspots and see the different Security hotspots.

The screenshot shows the SonarQube interface for 'Bhushan's Pipeline' project. The 'Security Hotspots' tab is selected. A single hotspot is listed: 'The tomcat image runs with root as the default user. Make sure it is safe here.' with a review priority of 'Medium'. The hotspot details include its status as 'To review', a description of the risk, and a 'Review' button. Below the main list, there is a code snippet from a Dockerfile with a highlighted line: 'FROM tomcat:8-jre8'. The code snippet contains the same security warning message.

Click on Measures and see the Measures in the form of Graphs.
Security:



Reliability:



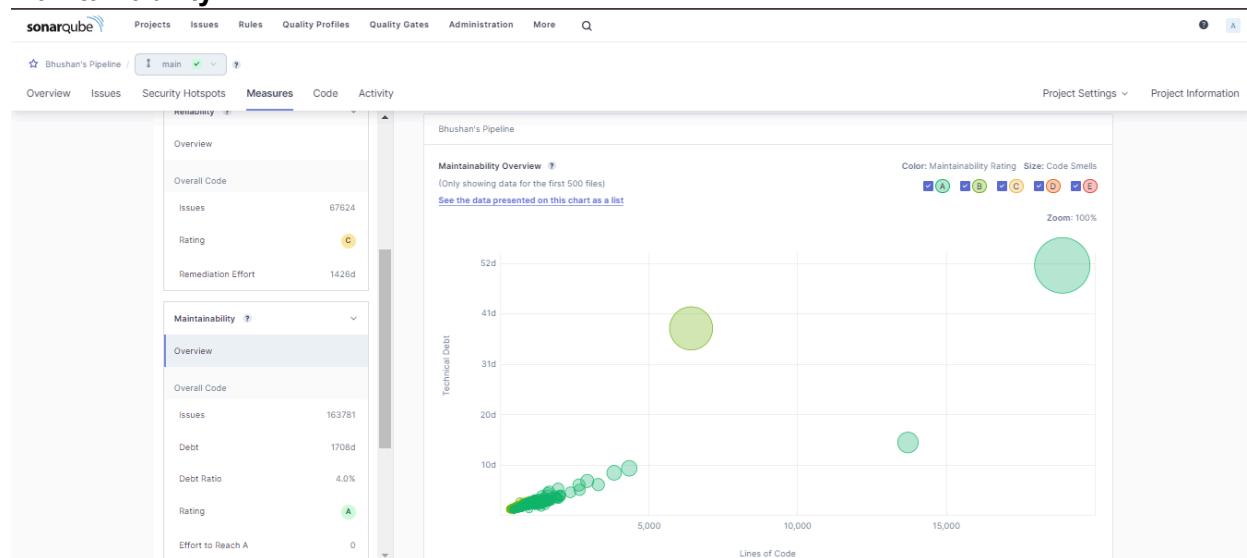
Name:Bhushan Mukund Kor

Academic Year:2024-2025

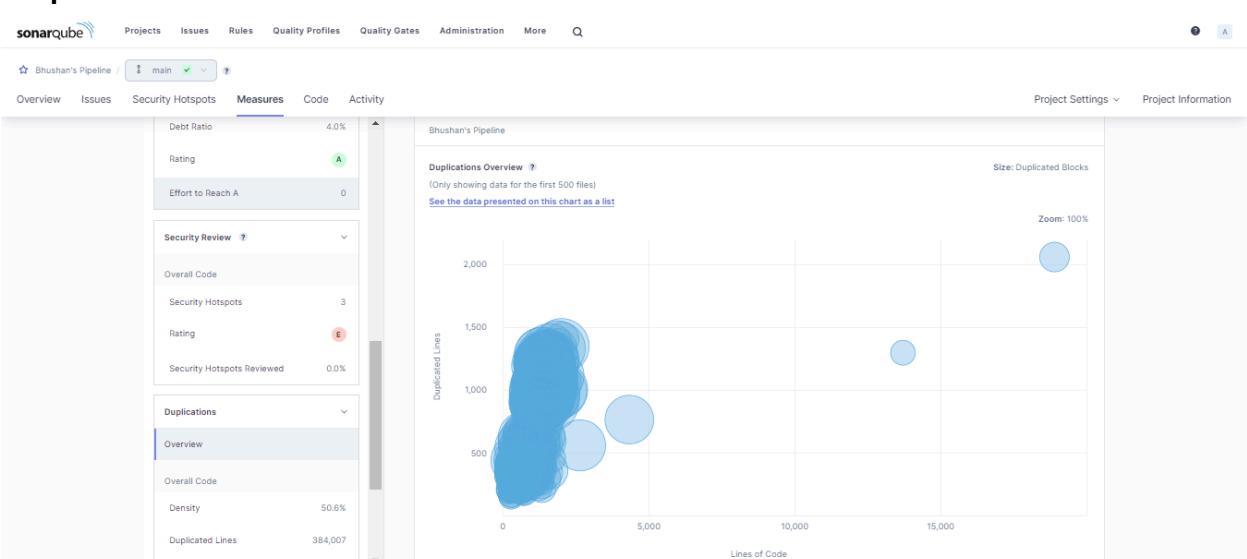
Division: D15C

Roll No: 28

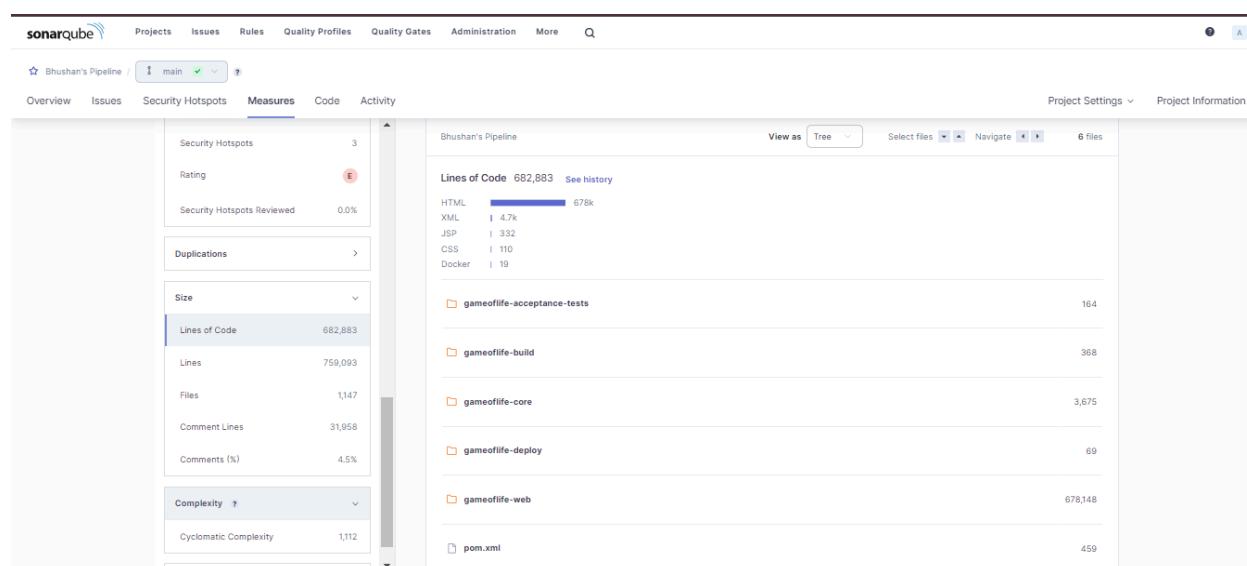
Maintainability:



Duplication:



Sizes:



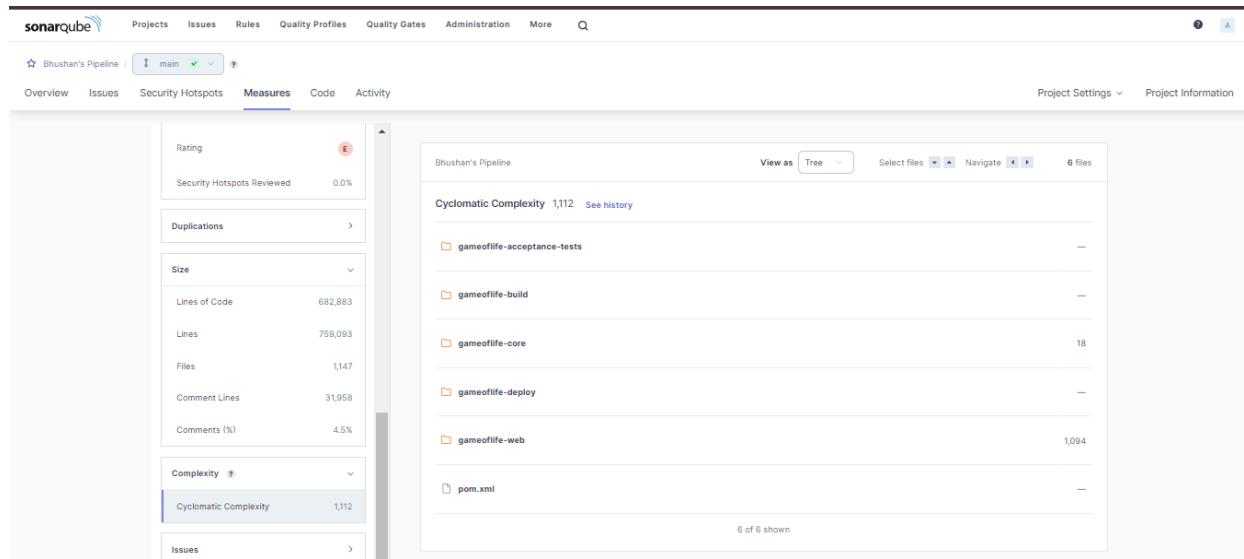
Name:Bhushan Mukund Kor

Academic Year:2024-2025

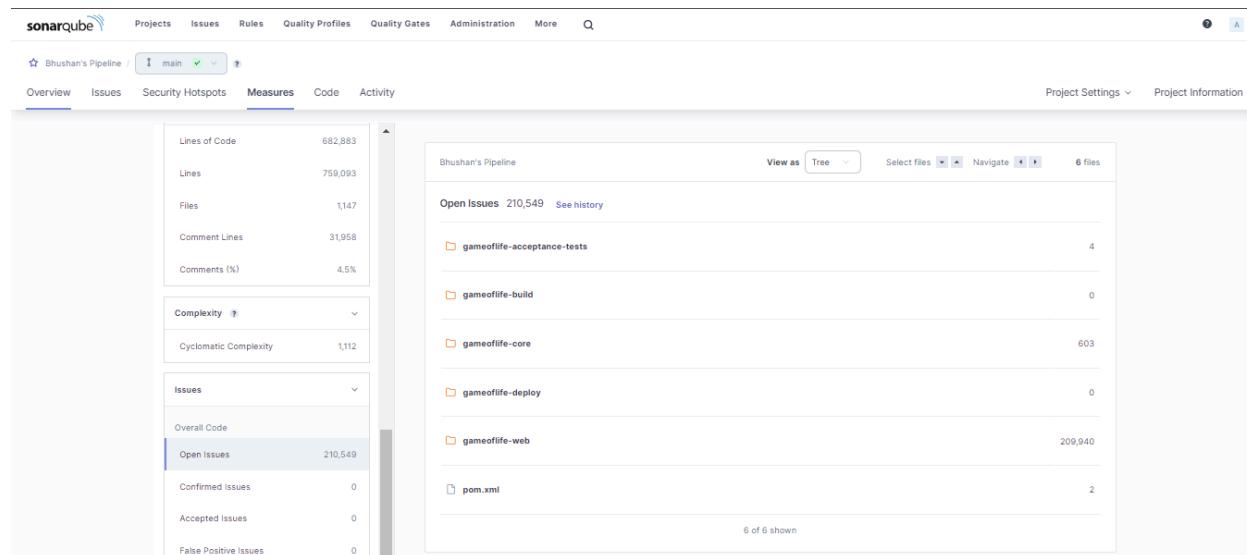
Division: D15C

Roll No: 28

Complexity:



Issues:



Conclusion: In this experiment, we have learned how to perform static analysis of a code using Jenkins CI/CD Pipeline with SonarQube analysis. A pipeline project is to be created which is given a pipeline script. This script contains all the information needed for the project to run the SonarQube analysis. After the necessary configurations are made on Jenkins, the Jenkins project is built. The code provided in this experiment contains lots of errors, bugs, duplications which can be checked on the SonarQube project linked with this build.

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

Theory:

What is Nagios?

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately. Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

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

Features of Nagios

Following are the important features of Nagios monitoring tool:

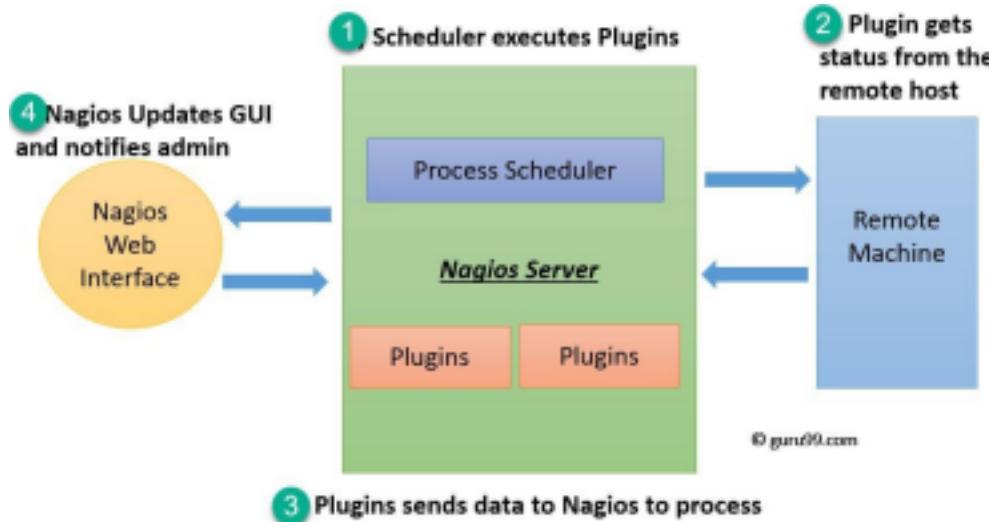
- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts

- Ability to define event handlers that runs during service or host events for proactive problem resolution

- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.



1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Prerequisites: AWS Personal or Academy Account.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Step 1: Login to your AWS account Personal / Academy. Click on EC2 instance then click on Create Security Group. Give the name as Nagios and any description and add the following inbounds rules.

The screenshot shows the AWS Console Home page. At the top, there's a navigation bar with 'aws' and 'Services' buttons, a search bar, and account information for 'N. Virginia' and 'BhushanKor'. Below the navigation bar, there are several sections: 'Recently visited' (EC2, Elastic Beanstalk, Billing and Cost Management, Amazon EventBridge, AWS Health Dashboard, Support, CloudFormation, Resource Groups & Tag Editor), 'Applications (0)' (Create application), 'Welcome to AWS' (Getting started with AWS, Training and certification), and 'Cost and usage' (Current month costs: \$2.36, Forecasted month end costs: \$2.43).

This screenshot shows the 'Network & Security' section of the AWS console. On the left, there's a sidebar with 'Lifecycle Manager', '▼ Network & Security' (expanded), 'Security Groups', 'Elastic IPs', 'Placement Groups', 'Key Pairs', 'Network Interfaces', and '▼ Load Balancing'. To the right, there are three main boxes: 'Placement groups', 'Snapshots', and 'Launch instances' (with a large orange 'Launch instance' button). A vertical grey bar separates the sidebar from the main content area.

The screenshot shows the 'Security Groups' page. At the top, there's a header with 'Security Groups (1) Info' and a search bar. Below the header is a table with one row, showing a single security group named 'sg-086e5eb333693fc6a' with a 'default' name and 'vpc-07974f574bb898ea1' VPC ID. There are buttons for 'Actions', 'Export security groups to CSV', and 'Create security group'.

Details

Security group name Nagios	Security group ID sg-07958b65d9d9f85b6	Description Nagios	VPC ID vpc-07974f574bb898ea1
Owner 01092820512	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-09bb0926ee6fc8361	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
-	sgr-01e7e94e7ae26e172	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-0769c7cd62438fe8	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0c97e92ada858080a	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-
-	sgr-0add40b87d65e...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-0cacbe20d4b601d55	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
-	sgr-0d9224b38f219c135	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-

Step 2: Now Create a new EC2 instance. Name: Nagios-host ,AMI: Amazon Linux, Instance Type: t2.micro.

EC2 > Instances > Launch an instance

Launch an instance

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

Name: Nagios-host

Application and OS Images (Amazon Machine Image)

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Browse more AMIs

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...
ami-debf9d41bbaf570d6

Virtual server type (instance type): t2.micro

Firewall (security group): Nagios

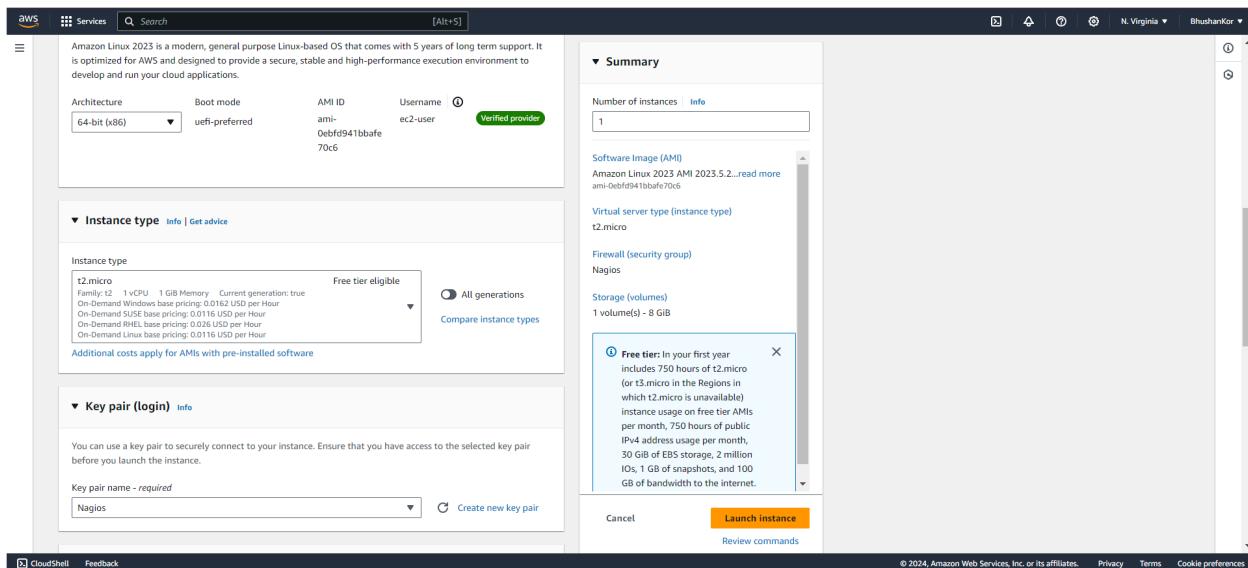
Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

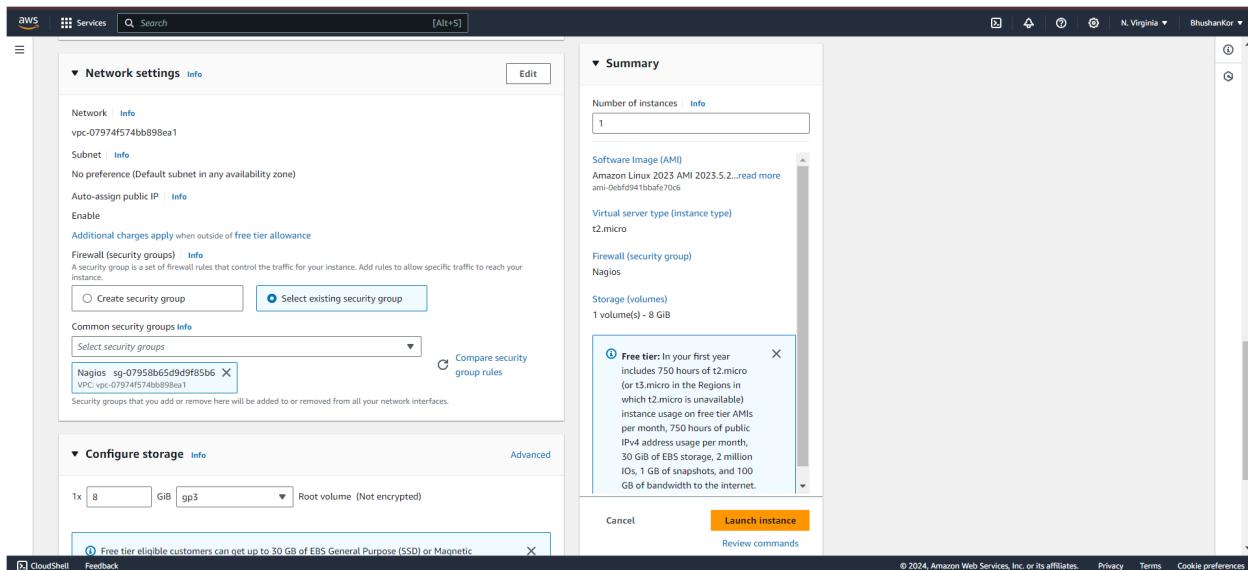
Cancel Launch instance

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

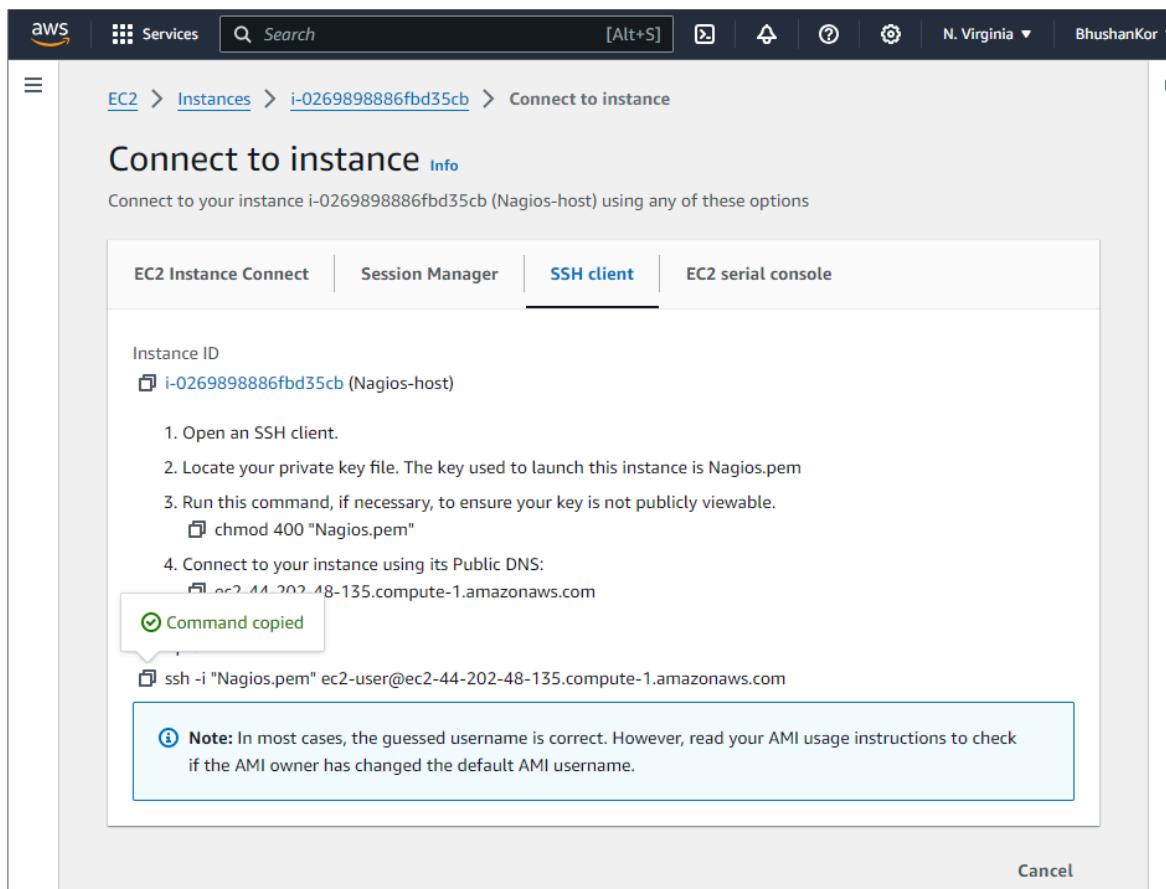
Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



Select the Existing Security Group and select the Security Group we have created in Step 1.



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .



Now open the terminal in the folder where your key(RSA key with .pem) is located.and paste that copied command.



Successfully connected to the instance.

```

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

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

PS C:\Users\bhush\OneDrive\Desktop\New folder (5)> ssh -i "Nagios.pem" ec2-user@ec2-44-202-48-135.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-48-135.compute-1.amazonaws.com (64:ff9b::2cca:3087)' can't be established.
ED25519 key fingerprint is SHA256:eh59Qe0hAQ5xtHiKD4/Z6g5P393uJ661HQ3kGcUr500.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-48-135.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ #####_      Amazon Linux 2023
~~ \_#####\_
~~ \###]
~~ \#_      https://aws.amazon.com/linux/amazon-linux-2023
~~ \#_      V~' '-->
~~ \_      /
~~ \_      /_
~~ \_      /_
/m/ _/_
/_m/ _/_

[ec2-user@ip-172-31-81-4 ~]$ 

```

Step 4: Now Run the following command to make a new user.

sudo adduser -m nagios

sudo passwd nagios

```

[ec2-user@ip-172-31-81-4 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-81-4 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

```

Step 5: Now Run the following command to make a new user group.

sudo groupadd nagcmd

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```

[ec2-user@ip-172-31-81-4 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-81-4 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache

```

Step 6: Now make a new directory and go to that directory.

mkdir ~/downloads

cd ~/downloads

```

sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-81-4 ~]$ mkdir ~/downloads
cd ~/downloads

```

Step 7: Now to download the Nagios 4.5.5 and Nagios-plugins 2.4.11 run the following commands respectively.

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

```
[ec2-user@ip-172-31-81-4 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-09-23 15:43:53-- https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 34.237.219.119, 52.54.96.194, 18.208.125.13, ...
Connecting to go.nagios.org (go.nagios.org)|34.237.219.119|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-23 15:43:53-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3:0:0::f03c:92ff:fe:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-23 15:43:53-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx      100%[=====] 1.97M 7.19MB/s   in 0.3s

2024-09-23 15:43:54 (7.19 MB/s) - '6kqcx' saved [2065473/2065473]
```

wget <https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz>

```
[ec2-user@ip-172-31-81-4 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-23 15:44:01-- 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-2.4 100%[=====] 2.62M 6.70MB/s   in 0.4s

2024-09-23 15:44:01 (6.70 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

Step 8: Now to extract the files from the downloaded Nagios 4.5.5 run the following command.

tar zxvf 6kqcx (Replace 6kqcx with your saved file name of Nagios 4.5.5 refer above screenshot(1st))

```
[ec2-user@ip-172-31-81-4 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autogen.sh
nagios-4.5.5/CHANGES
```

Step 9: Now change the directory to nagios-4.5.5 (Or which version you have downloaded)

```
[ec2-user@ip-172-31-81-4 downloads]$ cd nagios-4.5.5
```

Step 10: Now run the following command to configure.

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

```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
```

At the end we have found the error of cannot find ssl header .

```
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

So run following command to install ssl.

```
sudo yum install openssl-devel
```

```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:10:45 ago on Mon Sep 23 15:36:08 2024.
Dependencies resolved.
=====
 Package           Arch      Version       Repository      Size
 =====
 Installing:
 openssl-devel    x86_64    1:3.0.8-1.amzn2023.0.14    amazonlinux    3.0 M

 Transaction Summary
 =====
 Install 1 Package

 Total download size: 3.0 M
 Installed size: 4.7 M
 Is this ok [y/N]: y
 Downloading Packages:
 openssl-devel-3.0.8-1.amzn2023.0.14.x86_64. 35 MB/s | 3.0 MB   00:00
 -----
 Total                                         24 MB/s | 3.0 MB   00:00
 Running transaction check
 Transaction check succeeded.
 Running transaction test
 Transaction test succeeded.
 Running transaction
 Preparing : 1/1
 Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
 Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
 Verifying   : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1

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

 Complete!
```

Now rerun the command `./configure --with-command-group=nagcmd`

```
[ec2-user@ip-172-31-81-4 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...
-----
```

- This installs the Exfoliation theme for the Nagios web interface

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

*** Support Notes *****

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

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

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

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

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

Enjoy.

Step 11: Now run the following commands to steup the Nagios.

sudo make install

```
[ec2-user@ip-172-31-81-4 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'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
rm -f /usr/local/nagios/share/map.html
```

```
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/spool/checkresults
chmod g+s /usr/local/nagios/var/spool/checkresults

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

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

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

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

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

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
```

sudo make install-init

```
[ec2-user@ip-172-31-81-4 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-172-31-81-4 nagios-4.5.5]$ sudo make install-config
```

sudo make install-config

```
[ec2-user@ip-172-31-81-4 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 -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
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switchover.cfg /usr/local/nagios/etc/objects/switchover.cfg

*** Config files installed ***
```

sudo make install-webconf

```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin (To set the password)

```
[ec2-user@ip-172-31-81-4 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
```

Now to restart the httpd service run the following command.

sudo service httpd restart

```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

Step 12: Now to extract the files from the downloaded Nagios plugin 2.4.11 run the following command first change the directory.

cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz (According to your version)

```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-81-4 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/doncsm
```

Step 13: Now change the directory to nagios-plugins-2.4.11 and run the config command to configure.
cd nagios-plugins-2.4.11

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

```
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-81-4 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build custom type... x86_64-unknown-linux-gnu
```

Step 14: Run the following commands to check nagios and start it.

sudo chkconfig --add nagios

```
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-172-31-81-4 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

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

```
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

cd

sudo service nagios start

```
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-81-4 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

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

```
[ec2-user@ip-172-31-81-4 ~]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-81-4 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

sudo systemctl status nagios

```
[ec2-user@ip-172-31-81-4 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; presen
   Active: active (running) since Mon 2024-09-23 16:00:54 UTC; 1min 18s ago
     Docs: https://www.nagios.org/documentation
   Process: 64801 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nag
   Process: 64802 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagio
 Main PID: 64803 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.9M
    CPU: 95ms
   CGroup: /system.slice/nagios.service
           ├─64803 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/
           ├─64884 /usr/local/nagios/bin/nagios --worker /usr/local/nagios
           ├─64885 /usr/local/nagios/bin/nagios --worker /usr/local/nagios
           ├─64886 /usr/local/nagios/bin/nagios --worker /usr/local/nagios
           ├─64887 /usr/local/nagios/bin/nagios --worker /usr/local/nagios
           └─64888 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: Socket '/usr
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: core query h
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: echo service
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: help for the
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Successfu
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: Successfully lau
Lines 1-28 (END)
* nagios.service - Nagios Core 4.5.5
```

```

● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-09-23 16:00:54 UTC; 1min 18s ago
       Docs: https://www.nagios.org/documentation
    Process: 64801 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 64802 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 64803 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.9M
    CPU: 95ms
   CGroup: /system.slice/nagios.service
           └─64803 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─64804 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64805 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64806 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64807 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64808 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: core query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: echo service query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: help for the query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Successfully registered manager as @wproc with query handler
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64807;pid=64807
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64806;pid=64806
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64805;pid=64805
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64804;pid=64804
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: Successfully launched command file worker with pid 64808
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Step 15: We can see we have successfully launched the Nagios now . Open <http://<instance public ip >/nagios/> here it is <http://44.202.48.135/nagios> we can see the running web page of nagios.

The screenshot shows the Nagios Core 4.5.5 web interface. At the top, there's a navigation bar with links for Gmail, YouTube, Maps, News, Translate, Imported From IE, and Adobe Acrobat. Below the bar, the URL is 44.202.48.135/nagios/. The main content area has a header "Nagios® Core™ Version 4.5.5" dated September 17, 2024, with a "Check for updates" link. On the left, a sidebar menu includes sections for General, Current Status, Reports, and System. The "Current Status" section is active, displaying a "Tactical Overview" with tabs for Map, Hosts, Services, Host Groups, Service Groups, and Problems. It also shows a "Daemon running with PID 64803". The "Get Started" section lists items like "Start monitoring your infrastructure", "Change the look and feel of Nagios", and "Extend Nagios with hundreds of addons". The "Quick Links" section provides links to Nagios Library, Labs, Exchange, Support, and the official website. The "Latest News" and "Don't Miss..." sections are currently empty. At the bottom, there are copyright notices and a "Page Tour" button.

Conclusion: In this experiment, we have setup the Nagios core with plugins on Amazon Linux. Which will help us to understand Continuous monitoring and Installation. It is important to note that whatever set of rules we have added in step 1 are very important for this experiment.

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

Theory:

Port and Service Monitoring

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

Windows/Linux Server Monitoring

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

Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring Using Nagios:

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

sudo systemctl status nagios

```

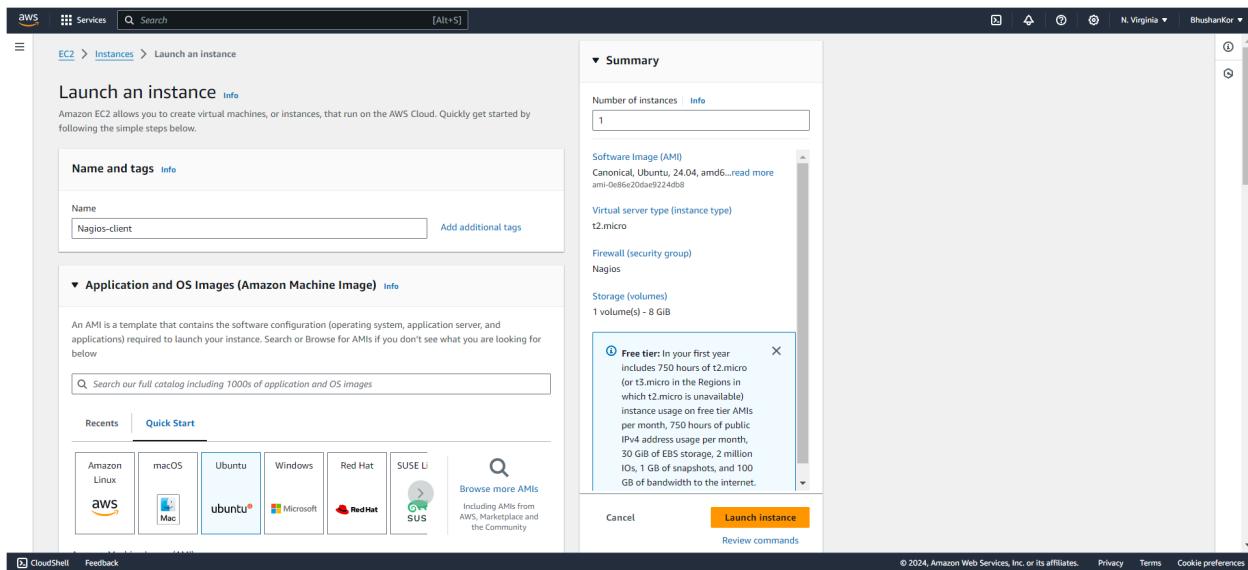
ec2-user@ip-172-31-81-4:~ % sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-09-23 16:32:36 UTC; 18min ago
     Docs: https://www.nagios.org/documentation
   Process: 1969 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 1971 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1972 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 6.8M
    CPU: 320ms
   CGroup: /system.slice/nagios.service
           └─1972 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─1974 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1975 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1976 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1977 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─1983 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 23 16:32:36 ip-172-31-81-4.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Sep 23 16:33:20 ip-172-31-81-4.ec2.internal nagios[1972]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;SOFT;3;SWAP CRITICAL - 0% free (0 MB out of 0 MB) ->
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRIT>
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;4;SWAP CRITICAL - 0% free (0 MB out of 0 MB) ->
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: wproc: NOTIFY job 1 from worker Core Worker 1975 is a non-check helper but exited with return cod>
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 23 16:34:20 ip-172-31-81-4.ec2.internal nagios[1972]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Sep 23 16:34:28 ip-172-31-81-4.ec2.internal nagios[1972]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;4;connect to address 127.0.0.1 and port 80: Connection<
[ec2-user@ip-172-31-81-4 ~]$ 

```

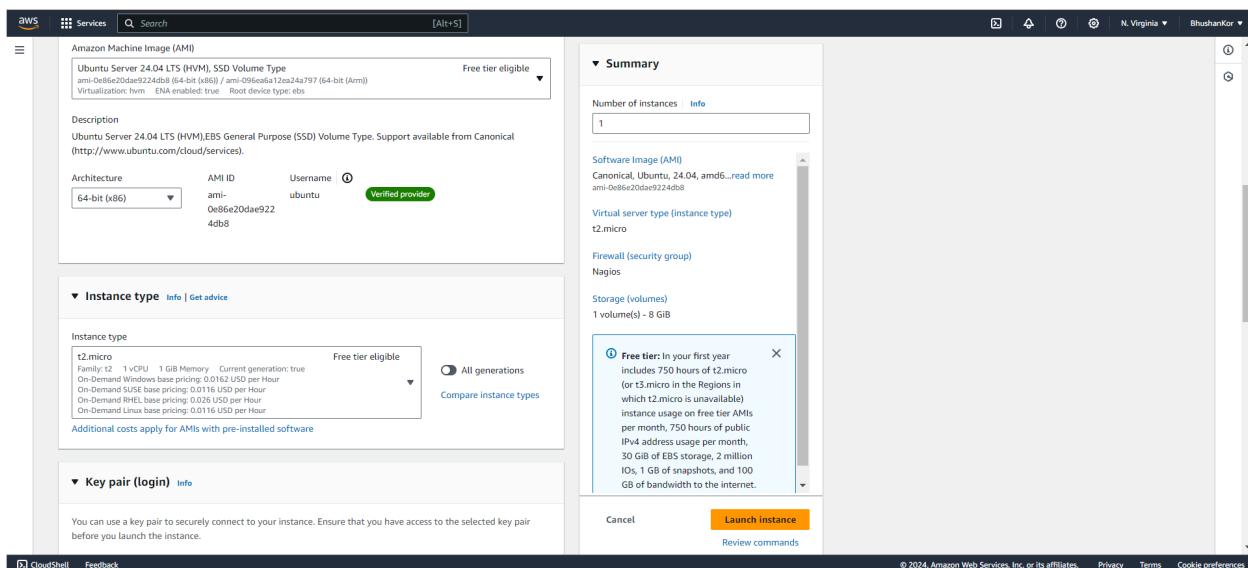
You can now proceed if you get the above message/output.

Step 2: Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

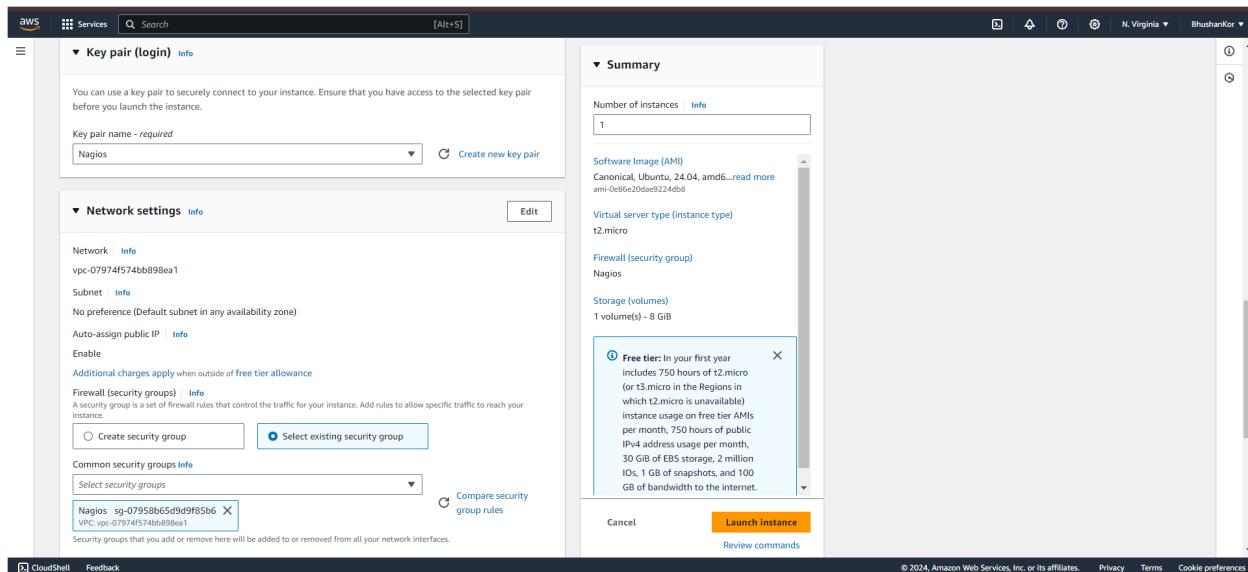


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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

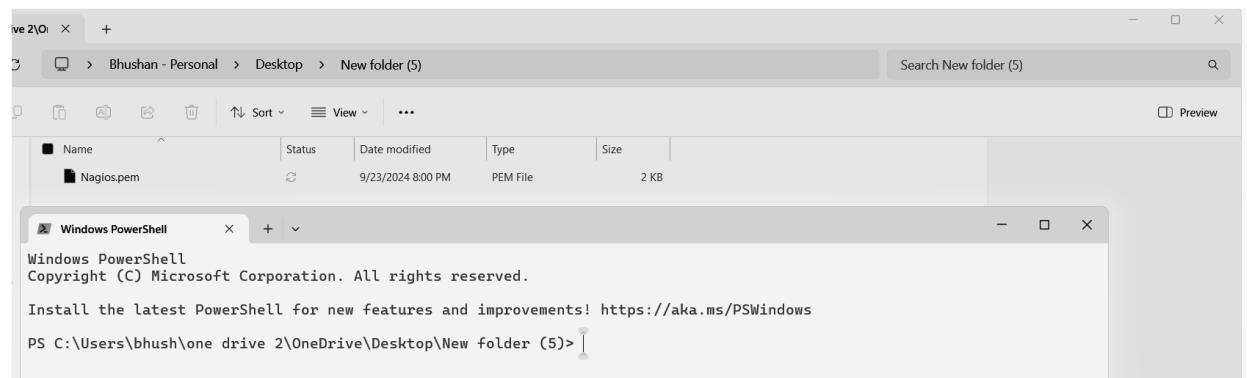


Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.



Successfully connected to the instance.

```
ubuntu@ip-172-31-83-152:~ x + ~
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bhush\OneDrive\Desktop>New folder (5)> ssh -i "Nagios.pem" ubuntu@ec2-3-88-57-181.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-57-181.compute-1.amazonaws.com (64:ff9b::358:39b5)' can't be established.
ED25519 key fingerprint is SHA256:r4BlMsLkuQqtE3zfW3Vqd7akVF0k5t+Zkh1k6l3cDA4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-57-181.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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 23 16:55:50 UTC 2024

System load: 0.81      Processes:           119
Usage of /:   22.8% of 6.71GB  Users logged in:     0
Memory usage: 21%        IPv4 address for enX0: 172.31.83.152
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-172-31-83-152:~$
```

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

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

ps -ef | grep nagios

```
[ec2-user@ip-172-31-81-4 ~]$ ps -ef | grep nagios
nagios 1971 1 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1972 1971 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1973 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1974 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1975 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1976 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1977 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1978 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1979 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1980 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1981 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1982 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1983 1972 0 16:32 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 3135 2898 0 16:57 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-81-4 ~]$
```

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

sudo su

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

```
[root@ip-172-31-81-4 ec2-user]# sudo su
```

```
[root@ip-172-31-81-4 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-81-4 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-81-4 ec2-user]#
```

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

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

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

```
[root@ip-172-31-81-4 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-81-4 ec2-user]#
```

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

Change `hostname` to `linuxserver`.

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

Set `hostgroup_name` to `linux-servers1`.

```
nano /usr/local/naqios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
root@ip-172-31-81-4:/home/e + - X
GNU nano 5.8 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg Modified

#####
#
# HOST DEFINITION
#
#####

# Define a host for the local machine

define host {

    use          linux-server ; Name of host template to use
                  ; This host definition will inherit all variables that are defined
                  ; in (or inherited by) the linux-server host template definition.

    host_name    linuxserver
    alias        localhost
    address      172.31.83.152
}

#####

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

Line to add : cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

Run the command : nano /usr/local/nagios/etc/nagios.cfg



```
root@ip-172-31-81-4:~# nano /usr/local/nagios/etc/nagios.cfg
#####
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# 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

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

Step 9: Now Verify the configuration files by running the following commands.

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



```
[root@ip-172-31-81-4 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-81-4 ec2-user]#
```

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

```
[root@ip-172-31-81-4 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-81-4 ec2-user]# sudo systemctl status nagi
```

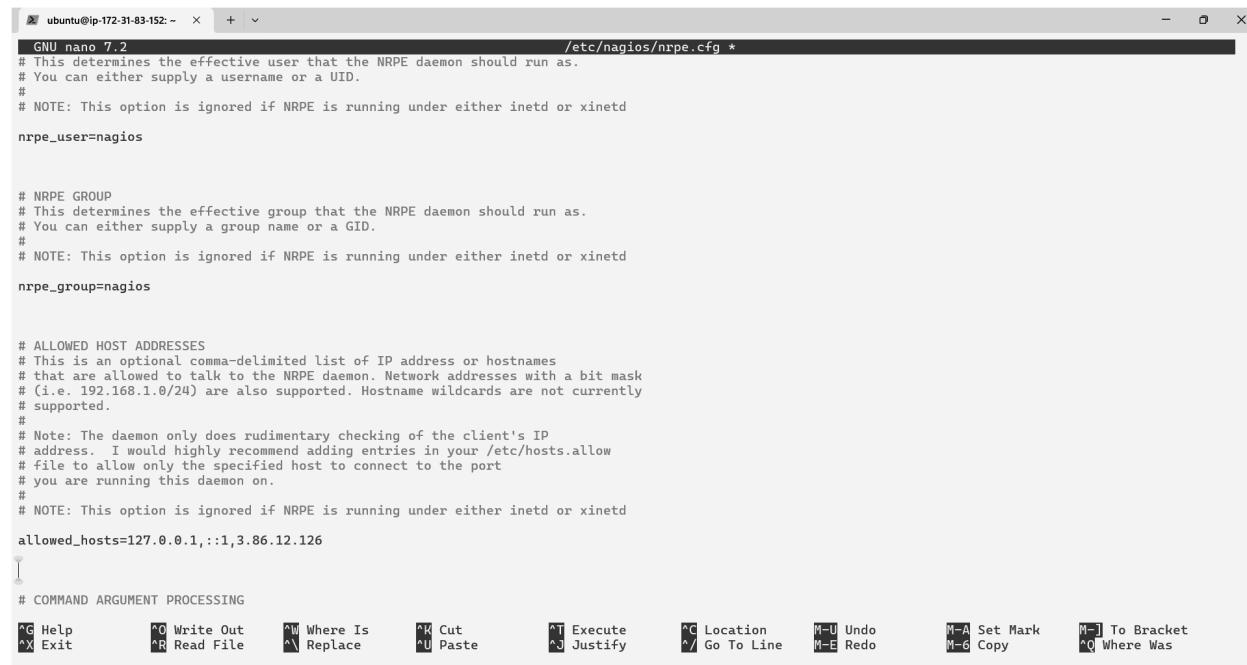
Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

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

```
ubuntu@ip-172-31-83-152:~$ 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
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
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
135 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libasan8 libatomic1 libcc1-0
  libgcc-13-dev libomp1 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libquadmath0 libtsan2 libubsan1
Suggested packages:
  cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gcc-13-doc gdb-x86-64-linux-gnu
The following NEW packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu gcc gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libasan8 libatomic1 libcc1-0
  libgcc-13-dev libomp1 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libquadmath0 libtsan2 libubsan1
0 upgraded, 22 newly installed, 0 to remove and 135 not upgraded.
Need to get 47.3 MB of archives.
After this operation, 163 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 gcc-13-base amd64 13.2.0-23ubuntu4 [49.0 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libis123 amd64 0.26-3build1 [680 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libmpc3 amd64 1.3.1-1build1 [54.5 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 cpp-13-x86-64-linux-gnu amd64 13.2.0-23ubuntu4 [11.2 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 cpp-13 amd64 13.2.0-23ubuntu4 [1032 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 cpp-x86-64-linux-gnu amd64 4:13.2.0-7ubuntu1 [5326 B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 cpp amd64 4:13.2.0-7ubuntu1 [22.4 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libcc1-0 amd64 14-20240412-0ubuntu1 [47.7 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libomp1 amd64 14-20240412-0ubuntu1 [147 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libitm1 amd64 14-20240412-0ubuntu1 [28.9 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libatomic1 amd64 14-20240412-0ubuntu1 [10.4 kB]
```

```
ubuntu@ip-172-31-83-152:~ x + v
monitoring-plugins is already the newest version (2.3.5-1ubuntu3).
Suggested packages:
  xinetd | inetd
The following NEW packages will be installed:
  nagios-nrpe-server
0 upgraded, 1 newly installed, 0 to remove and 135 not upgraded.
Need to get 356 kB of archives.
After this operation, 469 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Fetched 356 kB in 0s (15.6 MB/s)
Selecting previously unselected package nagios-nrpe-server.
(Reading database ... 73771 files and directories currently installed.)
Preparing to unpack .../nagios-nrpe-server_4.1.0-1ubuntu3_amd64.deb ...
Unpacking nagios-nrpe-server (4.1.0-1ubuntu3) ...
Setting up nagios-nrpe-server (4.1.0-1ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...
Running kernel seems to be up-to-date.
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
No containers need to be restarted.
User sessions running outdated binaries:
ubuntu @ session #2: sshd[1001,1111]
ubuntu @ session #8: sshd[1205,1263]
ubuntu @ user manager service: systemd[1006]
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-83-152:~$
```

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



```

GNU nano 7.2
/etc/nagios/nrpe.cfg *
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

# 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,::1,3.86.12.126

# COMMAND ARGUMENT PROCESSING
^G Help      ^O Write Out    ^W Where Is      ^X Cut          ^T Execute      ^C Location     ^U Undo        M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File    ^A Replace      ^U Paste        ^J Justify      ^G Go To Line   M-E Redo        M-C Copy       ^Q Where Was

```

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

sudo systemctl restart nagios-nrpe-server

```

ubuntu@ip-172-31-83-152:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-83-152:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-83-152:~$ 

```

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

sudo systemctl status nagios

```

[root@ip-172-31-81-4 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-81-4 ec2-user]# 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 Mon 2024-09-23 17:26:10 UTC; 1min 39s ago
     Docs: https://www.nagios.org/documentation
 Process: 4227 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 4228 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 4234 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.2M
    CPU: 60ms
   CGroup: /system.slice/nagios.service
           └─4234 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─4236 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4237 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4238 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4239 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─4242 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: qh: echo service query handler registered
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: qh: help for the query handler registered
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: wproc: Successfully registered manager as @wproc with query handler
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: wproc: Registry request: name=Core Worker 4238;pid=4238
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: wproc: Registry request: name=Core Worker 4239;pid=4239
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: wproc: Registry request: name=Core Worker 4237;pid=4237
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: wproc: Registry request: name=Core Worker 4236;pid=4236
Sep 23 17:26:10 ip-172-31-81-4.ec2.internal nagios[4234]: Successfully launched command file worker with pid 4242
Sep 23 17:27:00 ip-172-31-81-4.ec2.internal nagios[4234]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;3;connect to address 172.31.83.152 and port 80: Con
Sep 23 17:27:30 ip-172-31-81-4.ec2.internal nagios[4234]: SERVICE ALERT: linuxserver;Swap Usage;CRITICAL;SOFT;1;SWAP CRITICAL - 0% free (0 MB out of 0 MB)

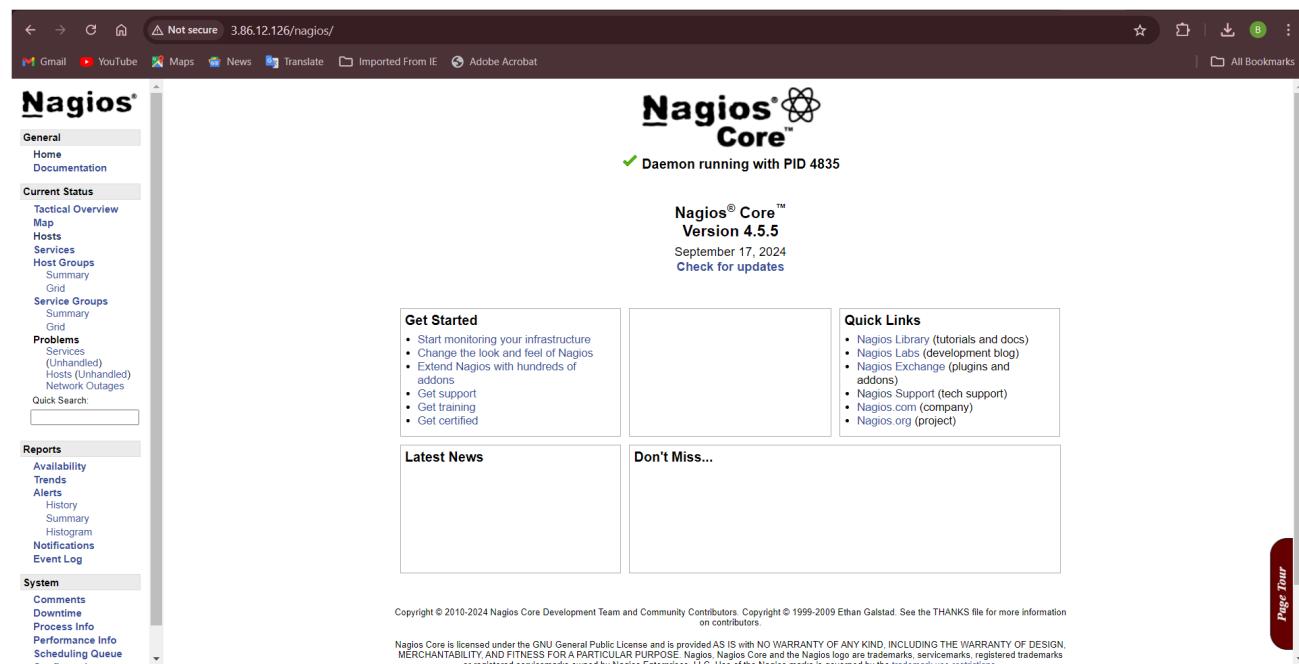
[root@ip-172-31-81-4 ec2-user]# 

```

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

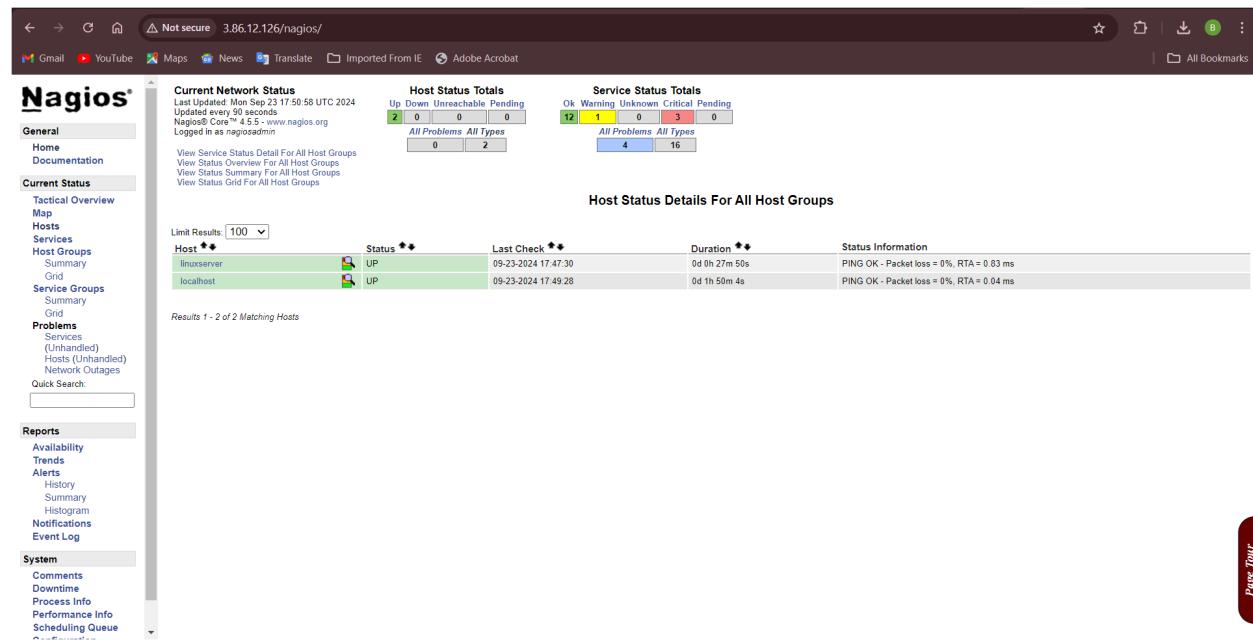
```
[root@ip-172-31-81-4 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-81-4 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-81-4 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-172-31-81-4 ec2-user]#
```

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



The screenshot shows the Nagios Core dashboard at the URL <http://3.86.12.126/nagios/>. The top navigation bar includes links for Gmail, YouTube, Maps, News, Translate, Imported From IE, Adobe Acrobat, and All Bookmarks. The main header features the Nagios Core logo with a green checkmark and the text "Daemon running with PID 4835". On the left, a sidebar menu lists General, Home, Documentation, Current Status (with options like Tactical Overview, Map, Hosts, Services, Host Groups, Grid, Service Groups, Problems, Reports, System), and a System section. The central content area contains several panels: "Get Started" with a list of monitoring steps, "Latest News" (empty), "Don't Miss..." (empty), and "Quick Links" with links to Nagios Library, Labs, Exchange, Support, and the official website. At the bottom, there are copyright notices and a "Page Tour" link.

Now Click on Hosts from left side panel



The screenshot shows the Nagios Core dashboard at the URL <http://3.86.12.126/nagios/>, specifically the "Hosts" section. The left sidebar remains the same as the previous screenshot. The main content area displays a table titled "Host Status Details For All Host Groups" with the following data:

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-23-2024 17:47:30	0d 0h 27m 50s	PING OK - Packet loss = 0%, RTA = 0.83 ms
localhost	UP	09-23-2024 17:49:28	0d 1h 50m 4s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Below the table, it says "Results 1 - 2 of 2 Matching Hosts". A "Page Tour" link is also present on the right side.

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

The screenshot shows the Nagios web interface at the URL 3.86.12.126/nagios/. The left sidebar contains navigation links for General, Documentation, Current Status, Reports, and System. The main content area displays 'Host Information' for the host 'localhost (linuxserver)'. It shows the host status as 'UP' (for 0d 0h 28m 21s), performance data (rtt=0.83100ms, 3000 000000, 5000 000000, 0.000000, p1=0%, 80, 100), and various check details. On the right, there's a 'Host Commands' section with a list of actions like 'Locate host on map' and 'Disable active checks of this host'. Below that is a 'Host Comments' section with a comment input field and a link to 'Delete all comments'. At the bottom, there's a table for 'Host Status Totals' and a 'Service Status Details For All Hosts' table.

Current Network Status

The screenshot shows the Nagios web interface at the URL 3.86.12.126/nagios/. The left sidebar contains navigation links for General, Documentation, Current Status, Reports, and System. The main content area displays 'Current Network Status' for the host 'localhost'. It shows the host status as 'UP' (for 0d 0h 28m 21s), performance data (rtt=0.83100ms, 3000 000000, 5000 000000, 0.000000, p1=0%, 80, 100), and various check details. On the right, there's a 'Service Status Details For All Hosts' table showing service status for both 'localhost' and 'linuxserver'. The table includes columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. Services listed include Current Load, Current Users, HTTP, PING, Root Partition, SSH, Swap Usage, Total Processes, and more. The table highlights critical errors for swap usage on both hosts.

Conclusion: In conclusion, the experiment focused on monitoring ports, services, and a Linux server using Nagios. Through the step-by-step process, we successfully configured Nagios to monitor essential network services on the Linux server. By setting up both the Nagios host and client, we were able to track system performance, ensure service availability, and monitor key metrics like CPU and memory usage.

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

Theory:

AWS Lambda

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

Lambda Workflow

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

AWS Lambda Functions

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

Prerequisites: AWS Personal/Academy Account

Name:Bhushan Mukund Kor

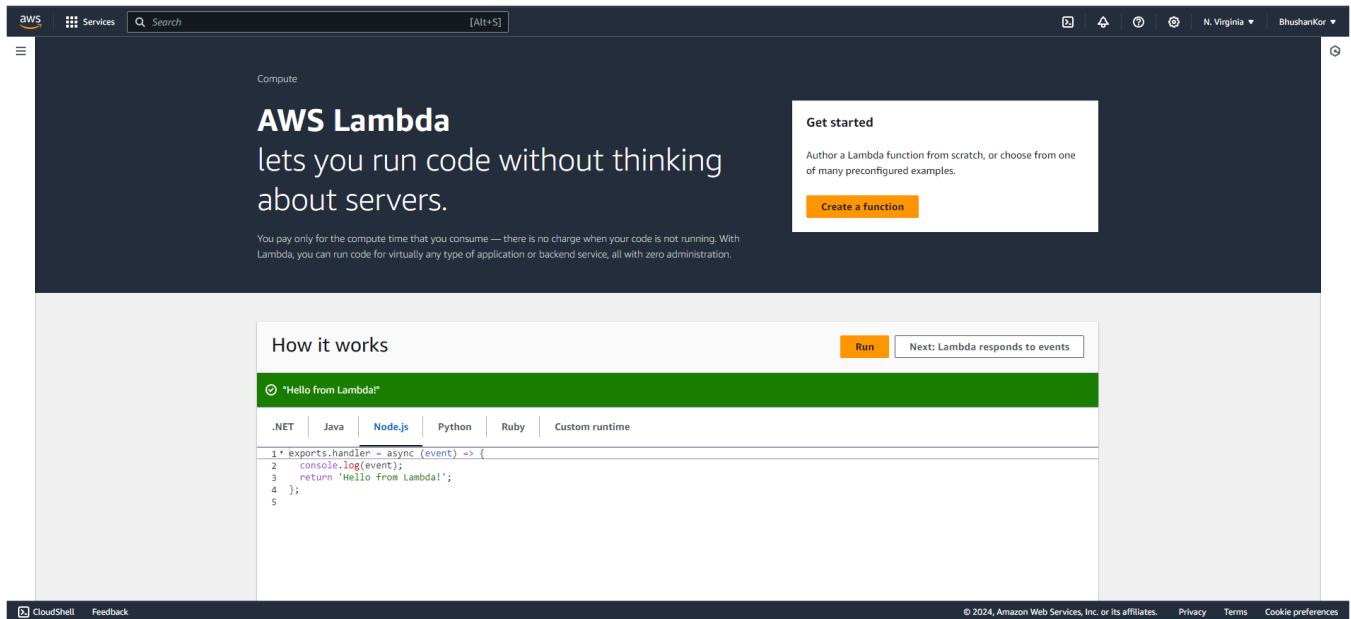
Academic Year:2024-2025

Division: D15C

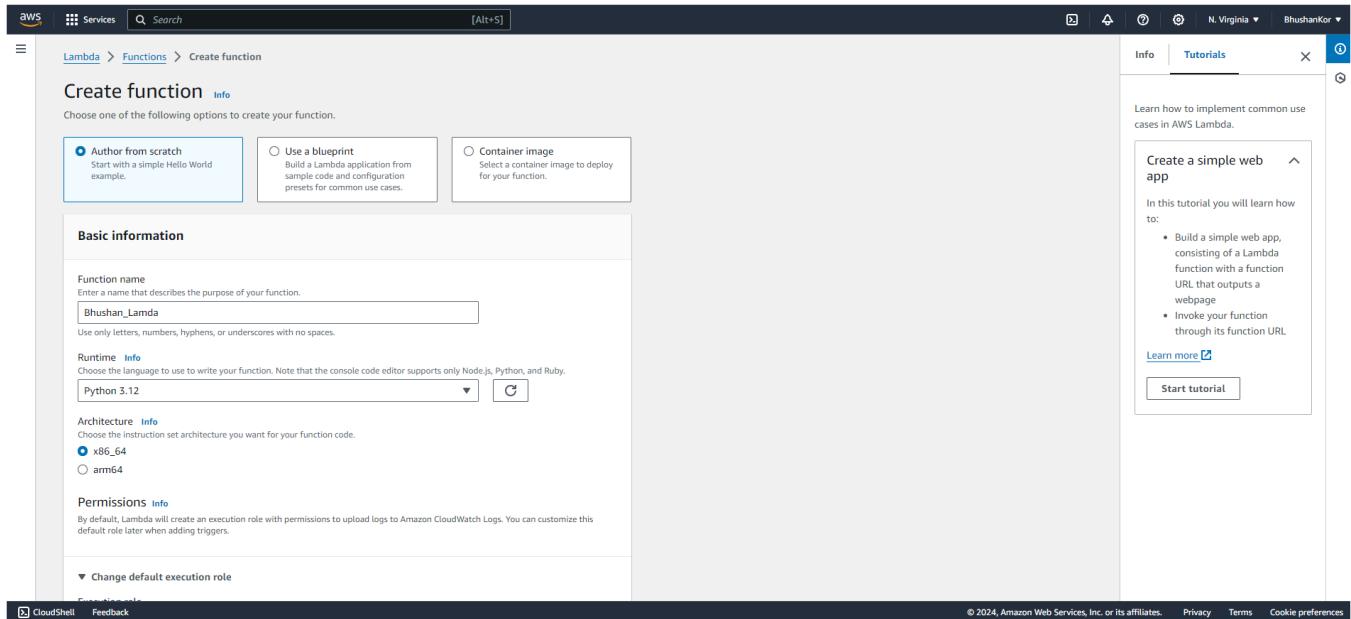
Roll No: 28

Steps To create the lambda function:

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



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



Step 1: AWS Lambda Function Creation

Step 2: Lambda Function Overview

```

import json

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

```

Step 3: Lambda Function Test

```

{
  "statusCode": 200,
  "body": "Hello from Lambda!"
}

```

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

The screenshot shows the AWS Lambda Configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area displays the General configuration settings:

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart Info	
0 min 3 sec	None	

An 'Edit' button is located in the top right corner of the configuration table.

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

The screenshot shows the AWS Lambda Edit basic settings interface. The top navigation bar includes AWS, Services, Search, and [Alt+S]. The left sidebar shows the path: Lambda > Functions > Bhushan_Lambda > Edit basic settings. The main content area is titled "Edit basic settings" and contains the following configuration fields:

- Basic settings**: Description - optional: Basic Settings; Memory: 128 MB; Ephemeral storage: 512 MB; SnapStart: None; Timeout: 0 min 1 sec; Execution role: Use an existing role (selected).
- Tutorials**: A sidebar titled "Create a simple web app" provides instructions on how to build a simple web application using Lambda.

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

The test event Bhushan_Event was successfully saved.

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

Test event Info

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

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

Event sharing settings

- Private
- Shareable

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

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

Template - optional

Event JSON

```

1  [
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 ]

```

Format JSON

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

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

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window

Test Deploy

Go to Anything (Ctrl-P)

lambda_function

Execution results

Test Event Name

Bhushan_Event

Response

The test event Bhushan_Event was successfully saved.

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window

Test Deploy

Go to Anything (Ctrl-P)

lambda_function Environment Var Execution result

Execution results

Test Event Name

Bhushan_Event

Response

```

{
  "statusCode": 200,
  "body": "Hello from Lambda!"
}

```

Status: Succeeded | Max memory used: 32 MB | Time: 2.05 ms

Function Logs

```

START RequestId: 8eb17f92-b164-401a-917f-900e4219f282 Version: $LATEST
END RequestId: 8eb17f92-b164-401a-917f-900e4219f282
REPORT RequestId: 8eb17f92-b164-401a-917f-900e4219f282 Duration: 2.05 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 89.00 ms
8eb17f92-b164-401a-917f-900e4219f282
8eb17f92-b164-401a-917f-900e4219f282

```

Info | Tutorials

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](#)

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

The screenshot shows the AWS Lambda function editor interface. The code editor window displays the following Python code:

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string="Hello ! I am Bhushan Kor From VESIT"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10

```

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

The screenshot shows the AWS Lambda function editor interface with a green header bar indicating "Successfully updated the function Bhushan_Lambda". The code editor window is identical to the previous one.

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

The screenshot shows the AWS Lambda function editor interface with the "Test" tab selected. The "Execution results" tab is active, displaying the following information:

- Test Event Name:** Bhushan_Event
- Status:** Succeeded | Max memory used: 32 MB | Time: 2.07 ms
- Response:**

```
{
    "statusCode": 200,
    "body": "\Hello ! I am Bhushan Kor From VESIT\""
}
```
- Function Logs:**

```
START RequestId: ce2370c2-9faa-4cf3-8618-728f972cac6a Version: $LATEST
END RequestId: ce2370c2-9faa-4cf3-8618-728f972cac6a
REPORT RequestId: ce2370c2-9faa-4cf3-8618-728f972cac6a Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 95.19 ms
```
- Request ID:** ce2370c2-9faa-4cf3-8618-728f972cac6a

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

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

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

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

Theory:**AWS Lambda and S3 Integration:**

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

Workflow:**1. Create an S3 Bucket:**

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

2. Create the Lambda Function:

- Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

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

4. Configure S3 Trigger:

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

5. Test the Setup:

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

Prerequisites: AWS Personal Account

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Steps To create the lambda function:

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

The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area is titled "Amazon S3" and "Amazon S3". It features an "Account snapshot - updated every 24 hours" section with a link to "All AWS Regions". Below this is a "General purpose buckets" section with a sub-section for "General purpose buckets (4) All AWS Regions". A search bar says "Find buckets by name". The table lists four buckets: "codepipeline-eu-north-1-823007647292" (Europe (Stockholm) eu-north-1), "codepipeline-us-east-1-934567252759" (US East (N. Virginia) us-east-1), "elasticbeanstalk-eu-north-1-010928205712" (Europe (Stockholm) eu-north-1), and "elasticbeanstalk-us-east-1-10928205712" (US East (N. Virginia) us-east-1). Each entry has a "View analyzer for" link and a creation date. At the bottom right of the table are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete". The footer includes links for CloudShell, Feedback, and various AWS policies.

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

The screenshot shows the "Create bucket" wizard. The first step is "General configuration". It shows the "AWS Region" set to "US East (N. Virginia) us-east-1". Under "Bucket type", "General purpose" is selected (radio button is checked). A tooltip explains it's recommended for most use cases and access patterns. The "Bucket name" field contains "bhushanbucket2". A note says the name must be unique. The "Copy settings from existing bucket - optional" section shows a "Choose bucket" button with "Format: s3://bucket/prefix". The second step is "Object Ownership". It shows "ACLs disabled (recommended)" (radio button is checked) and "ACLs enabled". A note says object ownership determines who can specify access to objects. The footer includes links for CloudShell, Feedback, and various AWS policies.

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforces

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

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

Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access 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.

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Successfully created bucket "bhushanbucket2"

To upload files and folders, or to configure additional bucket settings, choose [View details](#).

[Amazon S3](#) > [Buckets](#)

Account snapshot - updated every 24 hours All AWS Regions

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (5) Info [All AWS Regions](#)

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
bhushanbucket2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 28, 2024, 21:40:20 (UTC+05:30)
codepipeline-eu-north-1-823007647292	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 8, 2024, 23:54:38 (UTC+05:30)
codepipeline-us-east-1-934567252759	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 22:46:59 (UTC+05:30)
elasticbeanstalk-eu-north-1-010928205712	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 9, 2024, 00:05:29 (UTC+05:30)
elasticbeanstalk-us-east-1-010928205712	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 20:15:18 (UTC+05:30)

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

The screenshot shows the AWS Lambda console interface. At the top right, there is a 'Create function' button. Below it, a green bar displays the code for a 'Hello from Lambda!' function written in Node.js. The code is as follows:

```

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

```

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

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is selected. The 'Function name' field contains 'Bhushan_Lambda'. The 'Runtime' dropdown is set to 'Python 3.12'. The 'Architecture' dropdown is set to 'x86_64'. On the right side, there is a 'Tutorials' panel titled 'Create a simple web app' which provides instructions on how to build a Lambda function for a simple web application.

Step 1: AWS Lambda Function Creation

Step 2: Lambda Function Overview

```

import json

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

```

Step 3: Lambda Function Test

```

{
  "statusCode": 200,
  "body": "Hello from Lambda!"
}

```

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

The screenshot shows the AWS Lambda Configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area displays the General configuration settings:

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart Info	
0 min 3 sec	None	

An 'Edit' button is located in the top right corner of the configuration table.

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

The screenshot shows the AWS Lambda Edit basic settings interface. The top navigation bar includes AWS, Services, Search, and [Alt+S]. The left sidebar shows the path: Lambda > Functions > Bhushan_Lambda > Edit basic settings. The main content area is titled "Edit basic settings" and contains the following configuration fields:

- Basic settings**: Description - optional: Basic Settings; Memory: 128 MB; Ephemeral storage: 512 MB; SnapStart: None; Timeout: 0 min 1 sec; Execution role: Use an existing role (selected).
- Tutorials**: A sidebar titled "Create a simple web app" provides instructions on how to build a simple web application using Lambda.

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

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

Test event [Info](#)

Save [Test](#)

Event name

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

Event JSON [Format JSON](#)

```

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": "EXAMPLE123/5678abcdefghijklmabdasawesome/mnopqrstuvwxyzABCDEFGH"
18 *     },
19 *     "s3": {
20 *       "s3SchemaVersion": "1.0",
21 *       "configurationId": "testConfigRule",
22 *       "bucket": {
23 *         "name": "example-bucket",
24 *         "ownerIdentity": {
25 *           "principalId": "EXAMPLE"
26 *         }
27 *       },
28 *       "object": {
29 *         "key": "test%2Fkey",
30 *         "size": 1024,
31 *       }
32 *     }
33 *   }
34 * ]
  
```

https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Event JSON [Format JSON](#)

```

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": "EXAMPLE123/5678abcdefghijklmabdasawesome/mnopqrstuvwxyzABCDEFGH"
18 *     },
19 *     "s3": {
20 *       "s3SchemaVersion": "1.0",
21 *       "configurationId": "testConfigRule",
22 *       "bucket": {
23 *         "name": "example-bucket",
24 *         "ownerIdentity": {
25 *           "principalId": "EXAMPLE"
26 *         }
27 *       },
28 *       "object": {
29 *         "key": "test%2Fkey",
30 *         "size": 1024,
31 *       }
32 *     }
33 *   }
34 * ]
  
```

CloudShell [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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

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

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

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

The screenshot shows the AWS Lambda console. In the left sidebar, under 'Triggers', there is one entry: 'S3: bhushanbucket2'. This indicates that the Lambda function 'Bhushan_Lamda' has been triggered by an S3 event from the bucket 'bhushanbucket2'. The main panel displays the function overview, configuration details, and triggers.

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

```
import json
def lambda_handler(event, context):
    # TODO implement
    bucket_name= event['Records'][0]['s3']['bucket']['name']
    object_key= event['Records'][0]['s3']['object']['key']
    print(f"An Image has been added to the bucket {bucket_name} : {object_key}")
    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }
```

The screenshot shows the AWS Lambda function configuration interface. The 'Code' tab is selected, displaying the function's code in a code editor. The code is a Python script named 'lambda_function' that prints a log message when an image is uploaded to a specified bucket and returns a success response. The 'Test' tab is also visible.

Step 9: Now upload any image to the bucket.

The screenshot shows the AWS S3 console. A file named 'Screenshot 2024-09-28 225153.png' is being uploaded to the 'bhushanbucket2' bucket. The 'Upload' step is shown, where the file is selected and the destination is set to 's3://bhushanbucket2'. The status bar at the bottom indicates 'Upload succeeded'.

The screenshot shows the AWS S3 console after the upload is completed. The summary page displays the upload status, showing '1 file, 33.1 KB (100.0%)' successfully uploaded. The file 'Screenshot 2024-09-28 225153.png' is listed in the files and folders table.

Name	Type	Size	Status	Error
Screenshot 2...	image/png	33.1 KB	Succeeded	-

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

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

The screenshot shows the AWS Lambda console's 'Test' tab for a function named 'lambda_function'. The environment is set to 'Bhushan-Bucket'. The 'Execution results' section displays the following log output:

```
START RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c Version: $LATEST
An Image has been added to the bucket example-bucket : test%2Fkey
END RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c
REPORT RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c Duration: 1.92 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 87.14 ms
Request ID
196fbe32-4fb6-4ef5-9263-aa8f46d7120c
```

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

The screenshot shows the AWS CloudWatch Logs interface for the 'Log groups' section. The log group path is 'CloudWatch > Log groups > /aws/lambda/Bhushan_Lambda > 2024/09/28[\$LATEST]cd1c3fb6cd649cfaba3222dc1e6e723'. The 'Log events' table lists the following log entries:

Timestamp	Message
2024-09-28T17:44:48.126Z	INIT_START Runtime Version: python:3.12.v30 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:acd650000e3f6e085fb07933e3472ed0e58360d19ec5dd91bc7c7e8ad119de42
2024-09-28T17:44:48.215Z	START RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c Version: \$LATEST
2024-09-28T17:44:48.216Z	An Image has been added to the bucket example-bucket : test%2Fkey
2024-09-28T17:44:48.233Z	END RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c
2024-09-28T17:44:48.233Z	REPORT RequestId: 196fbe32-4fb6-4ef5-9263-aa8f46d7120c Duration: 1.92 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 87.14 ms

Conclusion: In this experiment, we successfully created an AWS Lambda function that logs a message when an image is uploaded to an S3 bucket. It is important to note that we have to select S3-put template in event other wise code will give an error. The function was successfully triggered by S3 object uploads, validating the functionality of Lambda's event-driven architecture. This experiment demonstrated how Lambda can efficiently respond to S3 events and how to troubleshoot common issues with event structure.