

STATIC HOSTING :

1) On local server (XAMPP)

Step 1: Install XAMPP from <https://www.apachefriends.org/>

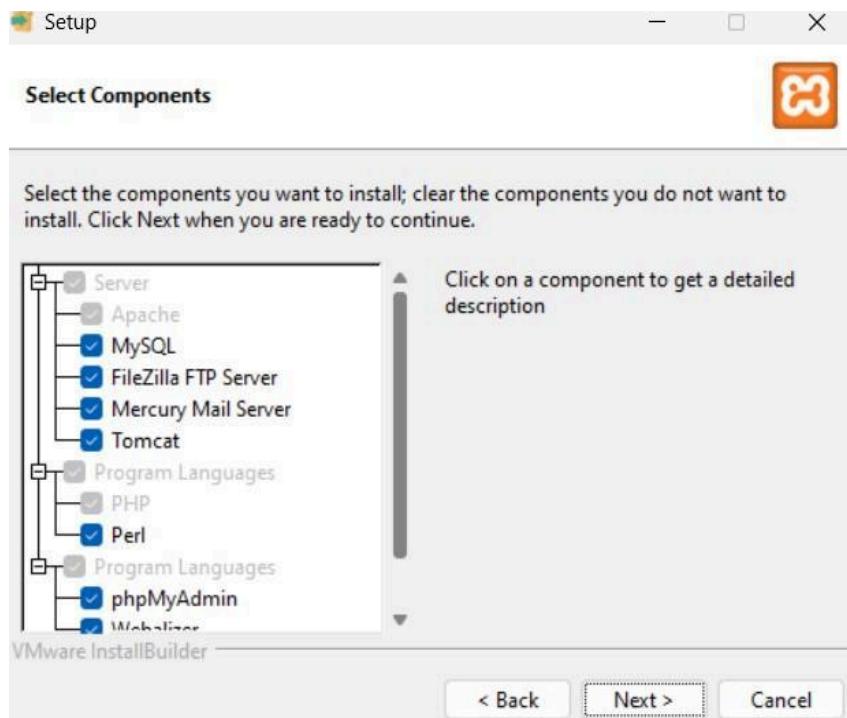
a) Download xampp



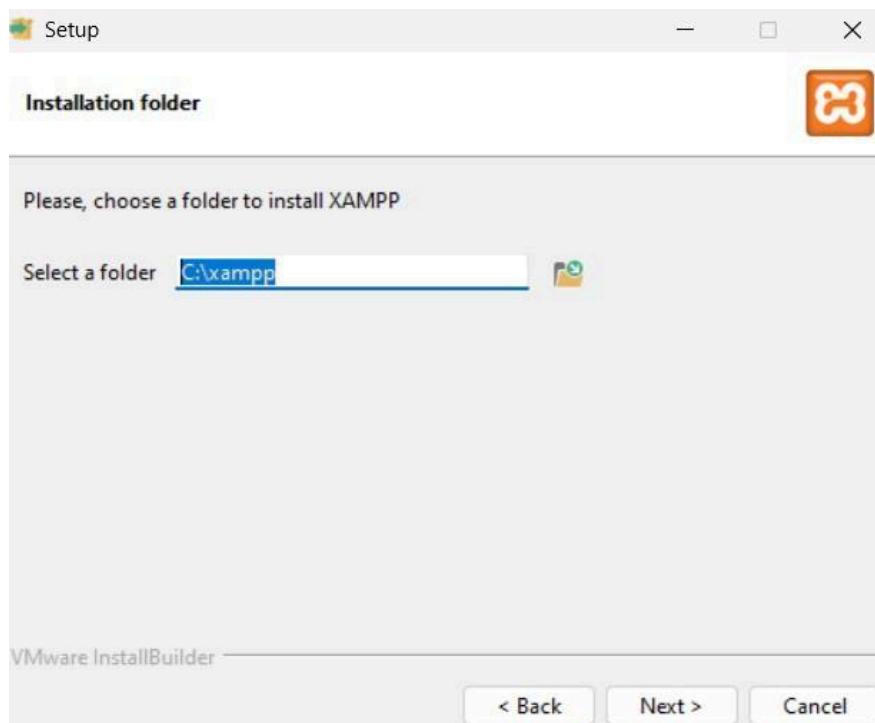
b) Open the setup file. Click on Next



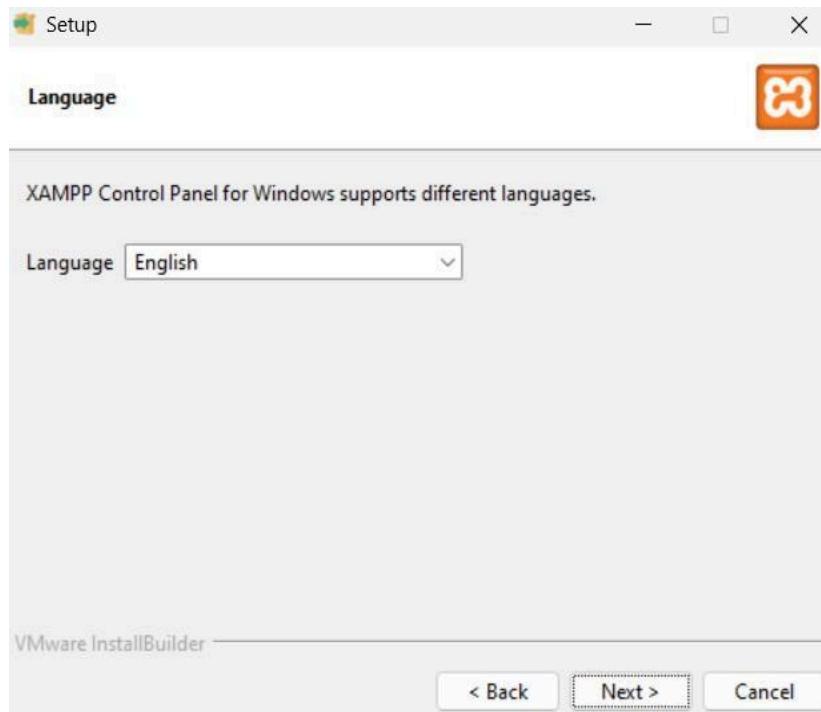
c) Select all the required components and click next



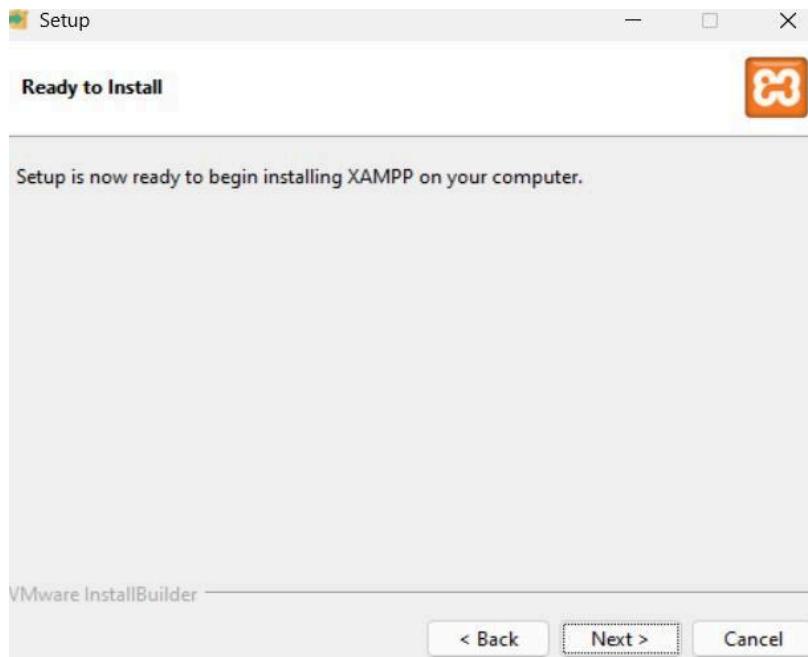
d) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



e) Select the language, click next. XAMPP starts to install



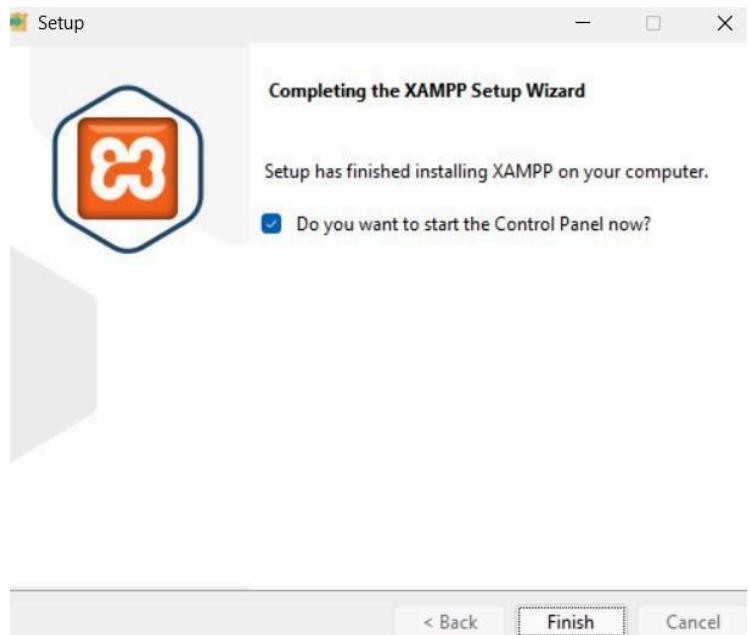
f) Click on Next



g) Wait until unpacking of all files is done



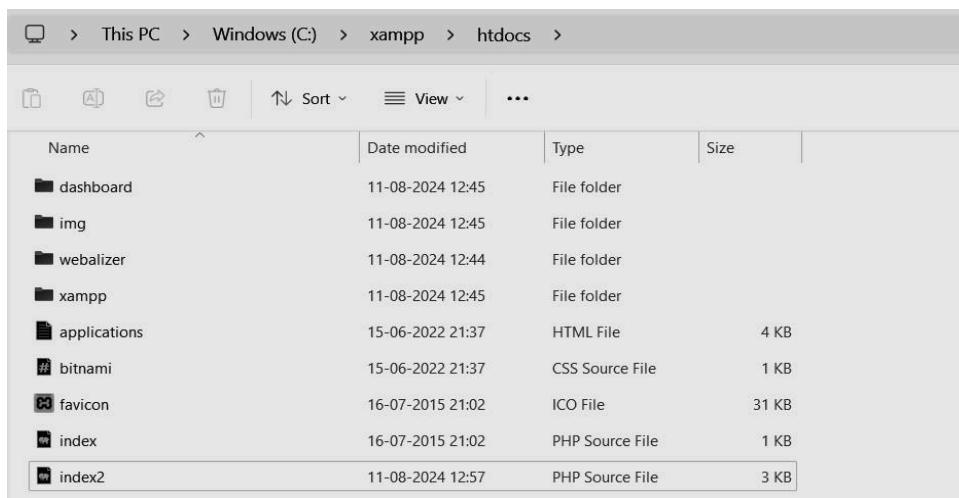
h) The installation is complete. Click Finish



Step 2: Setup a file that is to be hosted on the server. Make sure the file has extension .php

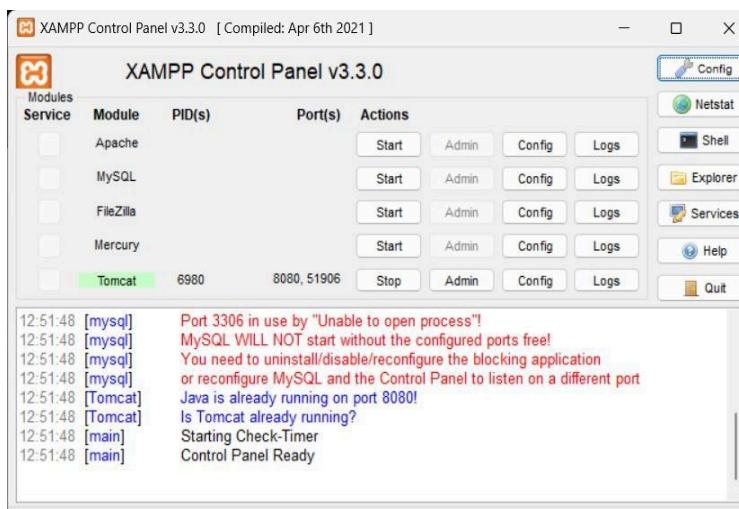
| Name | Date modified | Type | Size |
|--------|------------------|-----------------|------|
| index | 04-08-2024 18:02 | HTML File | 3 KB |
| index2 | 11-08-2024 12:57 | PHP Source File | 3 KB |

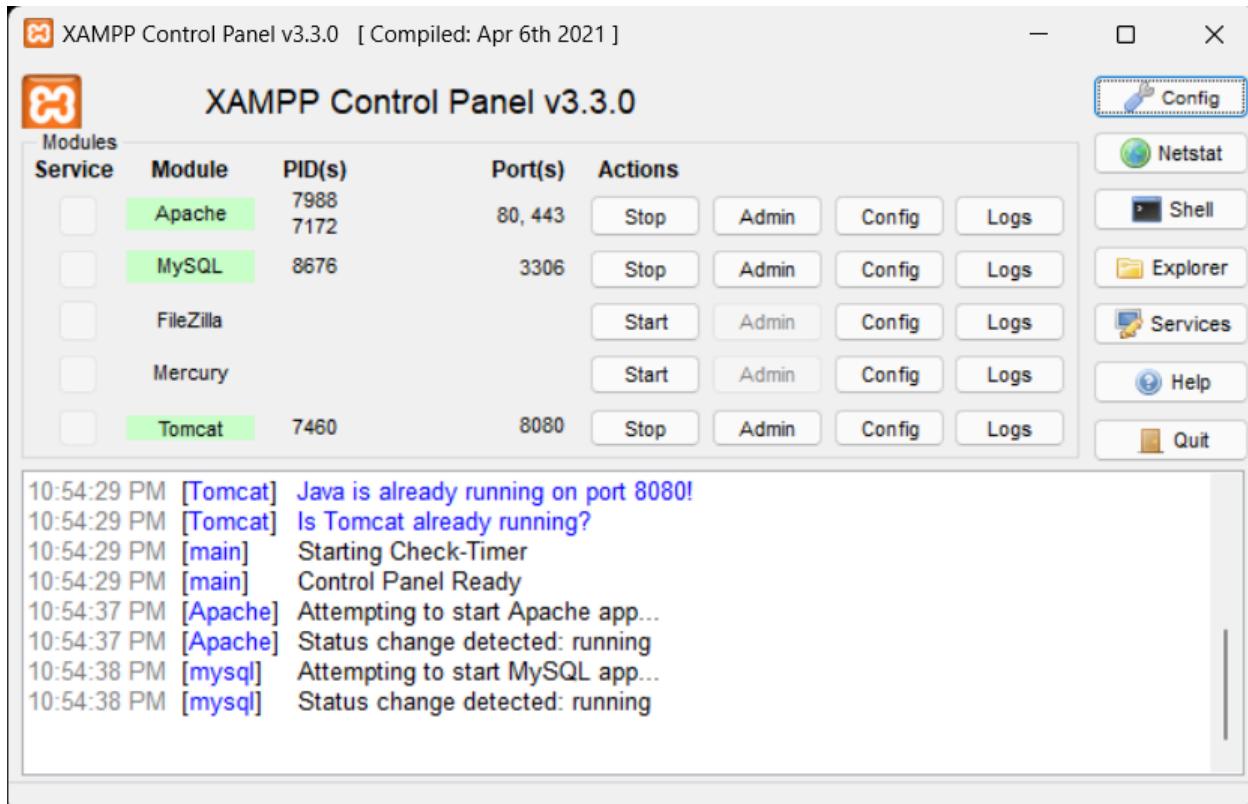
Step 3: Go to the directory where XAMPP was installed. Go to htdocs folder. Place your folder in this directory i.e Paste the index2.php here



| Name | Date modified | Type | Size |
|--------------|------------------|-----------------|-------|
| dashboard | 11-08-2024 12:45 | File folder | |
| img | 11-08-2024 12:45 | File folder | |
| webalizer | 11-08-2024 12:44 | File folder | |
| xampp | 11-08-2024 12:45 | File folder | |
| applications | 15-06-2022 21:37 | HTML File | 4 KB |
| bitnami | 15-06-2022 21:37 | CSS Source File | 1 KB |
| favicon | 16-07-2015 21:02 | ICO File | 31 KB |
| index | 16-07-2015 21:02 | PHP Source File | 1 KB |
| index2 | 11-08-2024 12:57 | PHP Source File | 3 KB |

Step 4: Open XAMPP Control Panel, start the Apache service (Required) and mySQL service (if needed)





Step 5: Open your web browser. Type localhost/demo. This will open your website on your browser.

localhost/demo/

i am Aditya Dubey d15c_10, Hello World

PHP Version 8.2.12

php

| | |
|---|---|
| System | Windows NT LAPTOP-824R819N 10.0 build 22621 (Windows 11) AMD64 |
| Build Date | Oct 24 2023 21:10:40 |
| Build System | Microsoft Windows Server 2019 Datacenter [10.0.17763] |
| Compiler | Visual C++ 2019 |
| Architecture | x64 |
| Configure Command | ./configure --enable-snapshot-build --enable-debug-pack --with-pdo-oci=.,.,.,instantclient/sdk/shared --with-oci8-19=.,.,.,instantclient/sdk/shared --enable-object-out-dir=../obj --enable-com-dotnet=shared --without-analyzer --with-pgo |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | enabled |
| Configuration File (php.ini) Path | no value |
| Loaded Configuration File | C:\xampp\php\php.ini |
| Scan this dir for additional .ini files | (none) |
| Additional .ini files parsed | (none) |
| PHP API | 20220829 |
| PHP Extension | 20220829 |
| Zend Extension | 420220829 |
| Zend Extension Build | API420220829_TS_VS16 |
| PHP Extension Build | API20220829_TS_VS16 |
| Debug Build | no |
| Thread Safety | enabled |
| Thread API | Windows Threads |
| Zend Signal Handling | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | provided by mbstring |
| Zend Max Execution Time | disabled |

1) On AWS S3

Step 1: Login to your AWS account. Go to services and open S3.

The screenshot shows the AWS Services console interface. The top navigation bar includes tabs for 'CloudShell' and 'Feedback'. The main menu on the left lists 'Console Home', 'myApplications', and 'All services'. The central area displays a grid of service icons and names under several categories: Storage (S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup, AWS Elastic Disaster Recovery), Database (RDS, ElastiCache, Neptune, Amazon QLDB, Amazon DocumentDB, Amazon Keyspaces, Amazon Timestream, DynamoDB, Amazon MemoryDB), Media Services (Kinesis Video Streams, MediaConvert, MediaLive, MediaPackage, MediaStore, MediaTailor), and Application Integration (AWS Amplify, AWS AppSync, Device Farm, Amazon Location Service). The bottom right corner shows copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 2: Click on Create Bucket

The screenshot shows the 'Amazon S3' service page. The left sidebar contains links for 'Buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', and 'Storage Lens' (with sub-links for 'Dashboards', 'Storage Lens groups', and 'AWS Organizations settings'). The main content area features an 'Account snapshot - updated every 24 hours' section with a 'View Storage Lens dashboard' button. Below this is a table titled 'General purpose buckets (2)' with columns for 'Name', 'AWS Region', 'IAM Access Analyzer', and 'Creation date'. The table shows one entry: 'elasticbeanstalk-eu-north-1' (Region: Europe (Stockholm), IAM: eu-north-1, Creation date: August 13, 2024, 23:34:46 (UTC+05:30)). At the bottom of the table is a 'Create bucket' button. The bottom right corner shows copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 3: Give a name to your bucket, keeping other options default, scroll down and click on Create Bucket

The screenshots show the AWS S3 'Create bucket' wizard process:

- Step 1: General configuration**
 - AWS Region: Europe (Stockholm) eu-north-1
 - Bucket type: General purpose (selected)
 - Bucket name: www.aditya1492025.com
 - Copy settings from existing bucket - optional: Only the bucket settings in the following configuration are copied.
- Step 2: Object Ownership**
 - Object Ownership: ACLs disabled (recommended) (selected)
 - 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
- Step 3: Block Public Access settings for this bucket**
 - Block Public Access settings: Turned off (not selected)
 - Description: 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](#)

Step 4: Click on the name of your bucket and goto Properties

The screenshot shows the AWS S3 bucket creation interface. In the 'Encryption type' section, 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' is selected. Under 'Bucket Key', 'Enable' is selected. A note states that using an S3 Bucket Key for SSE-KMS reduces costs but DSSE-KMS is not supported. The 'Advanced settings' section is collapsed. A note at the bottom says 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.' A large green success message at the top right says 'Successfully created bucket "www.aditya1492025.com"'.

| Name | AWS Region | IAM Access Analyzer | Creation date |
|--|---------------------------------|--|---------------------------------------|
| elasticbeanstalk-eu-north-1-025066268039 | Europe (Stockholm) eu-north-1 | View analyzer for eu-north-1 | August 13, 2024, 23:34:46 (UTC+05:30) |
| elasticbeanstalk-us-east-1-025066268039 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 7, 2024, 10:22:38 (UTC+05:30) |
| www.aditya1492025.com | Europe (Stockholm) eu-north-1 | View analyzer for eu-north-1 | August 16, 2024, 10:11:55 (UTC+05:30) |

Name:Aditya Dubey

Div:D15C

Roll no:10

AWS Region: Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN): arn:aws:s3:::www.aditya1492025.com

Creation date: August 16, 2024, 10:11:55 (UTC+05:30)

Bucket Versioning: Disabled

Step 5: Scroll down till you find Static website hosting, click on edit

Object Lock: Disabled

Requester pays: Disabled

Static website hosting: Disabled

Step 6: Click on Enable static website hosting

Step 7: Write the name of your document which you wanted to host on AWS from your local folder and in error document, give name as error..html. Save your changes.

Step 8: Save the Changes

Name:Aditya Dubey

Div:D15C

Roll no:10

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs like CloudShell, Feedback, and various AWS services. Below the navigation bar, the main content area displays the configuration for a static website hosting bucket. A prominent green success message at the top states "Successfully edited static website hosting." The configuration details include:

- 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).
- Bucket Key:** Info - When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. [Learn more](#).
- Enabled:** A checkbox indicating the feature is active.

Below this, there's a section for "Intelligent-Tiering Archive configurations (0)" with buttons for View details, Edit, Delete, and Create configuration. A search bar and a table header with columns Name, Status, Scope, Days until transition to Ar..., and Days until transition to De... are also visible.

Step 9: Go to Objects tab and click on upload file

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs like Services, Search, and a search bar. Below it, a sub-navigation bar has 'Objects' selected. The main area is titled 'Objects (0) Info'. It features a toolbar with actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar and a pagination control (page 1 of 1) are also present. A message states 'No objects' and 'You don't have any objects in this bucket.' An 'Upload' button is at the bottom.

Step 10: Click on Add files. Add all the files you want to upload. Then scroll down and click on Upload

The screenshot shows the AWS S3 'Upload' interface. The top navigation bar includes CloudShell, Feedback, and various AWS services. The main area is titled 'Upload info'. It says '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'. Below is a large dashed box for dragging files. A table lists 'Files and folders (2 Total, 2.1 KB)' with two entries: 'error.html' and 'index.html', both of which are 'text/html' type. At the bottom, a 'Destination' section is shown with a 'Destination' dropdown. The footer includes CloudShell, Feedback, and standard copyright information.

Name:Aditya Dubey

Div:D15C

Roll no:10

The screenshot shows two consecutive screenshots of the AWS S3 console.

Screenshot 1: The "Upload" screen. It displays a table titled "Files and folders (0)" with a note: "All files and folders in this table will be uploaded." Below the table is a search bar labeled "Find by name". A dropdown menu shows "Name" and "Folder" as filter options. At the bottom right are "Cancel" and "Upload" buttons.

Screenshot 2: The "Upload succeeded" confirmation screen. It shows a green header bar with the message "Upload succeeded" and "View details below." Below this is a "Summary" section with a table showing the destination bucket and upload results: "Succeeded" (2 files, 2.1 KB (100.00%)) and "Failed" (0 files, 0 B (0%)).

Details:

| Files and folders (2 Total, 2.1 KB) | | | | | |
|-------------------------------------|--------|-----------|---------|-----------|-------|
| Find by name | | | | | |
| Name | Folder | Type | Size | Status | Error |
| error.html | - | text/html | 982.0 B | SUCCEEDED | - |
| index.html | - | text/html | 1.2 KB | SUCCEEDED | - |

Step11: This will take you to the Objects screen. Switch to Properties, scroll down to Static web hosting. There you would find the link (Bucket website endpoint) to your website.

The screenshot shows the AWS S3 bucket configuration page for a bucket named 'aditya1492025'. Under the 'Static website hosting' section, 'Enabled' is selected for 'Static website hosting'. The 'Hosting type' is set to 'Bucket hosting'. The 'Bucket website endpoint' is displayed as <http://www.aditya1492025.com.s3-website.eu-north-1.amazonaws.com>.

Step12: Open the link. It will show a 403 forbidden error screen as the contents of the bucket are not available for the public users. To change this, go to Permissions tab, go to Block public access and click on edit

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 8TQ4EGP4TK06MVPB
- HostId: hF+ToadQUoCuDM8H+iFRsXdA28TGp+xikYbjb4CICS/t+3it4ihA/tvgA1Xr1xo+JL5AhkT6hJs=

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Step 13:

Uncheck the Block all public access checkbox and click on save changes

Name:Aditya Dubey

Div:D15C

Roll no:10

The screenshot shows the 'Edit Block public access (bucket settings)' page for the bucket 'www.aditya1492025.com'. The 'Block all public access' setting is currently off. The 'Edit' button is visible in the top right corner of the settings section.

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.

Permissions overview

Access finding

Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#).
[View analyzer for eu-north-1](#)

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

Off

Individual Block Public Access settings for this bucket

Step 14:Successfully Changed the Settings

Name:Aditya Dubey

Div:D15C

Roll no:10

The screenshot shows the AWS S3 Bucket Permissions settings for the bucket 'www.aditya1492025.com'. A green success message at the top states 'Successfully edited Block Public Access settings for this bucket.' Below this, the 'Permissions' tab is selected. Under 'Block public access (bucket settings)', the 'Block all public access' switch is set to 'Off'. There is a link to 'Individual Block Public Access settings for this bucket'. The bottom right corner shows the user's profile as 'Geeta House'.

Step 15: Scroll down to bucket policy and click edit and paste the code from given Github Link
<https://gist.github.com/Savjee/b4b3a21d143a30e7dc07>

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

Paste this code snippet in the policy textarea. Replace YOUR-BUCKET-NAME-HERE with the name you have given to your bucket. Save the changes

The screenshot shows the AWS S3 Bucket Policy configuration page. The main area displays the following JSON policy:

```
1▼ [{  
2    "Version": "2012-10-17",  
3    "Statement": [  
4        {  
5            "Sid": "PublicReadGetObject",  
6            "Effect": "Allow",  
7            "Principal": "*",  
8            "Action": "s3:GetObject",  
9            "Resource": "arn:aws:s3:::www.aditya1492025.com/*"  
10       }  
11    ]  
12}  
13]  
14]
```

To the right of the policy, there is a sidebar with the following sections:

- Edit statement**: A button to edit an existing statement.
- Select a statement**: A placeholder text indicating where to select an existing statement or add a new one.
- + Add new statement**: A button to add a new statement.

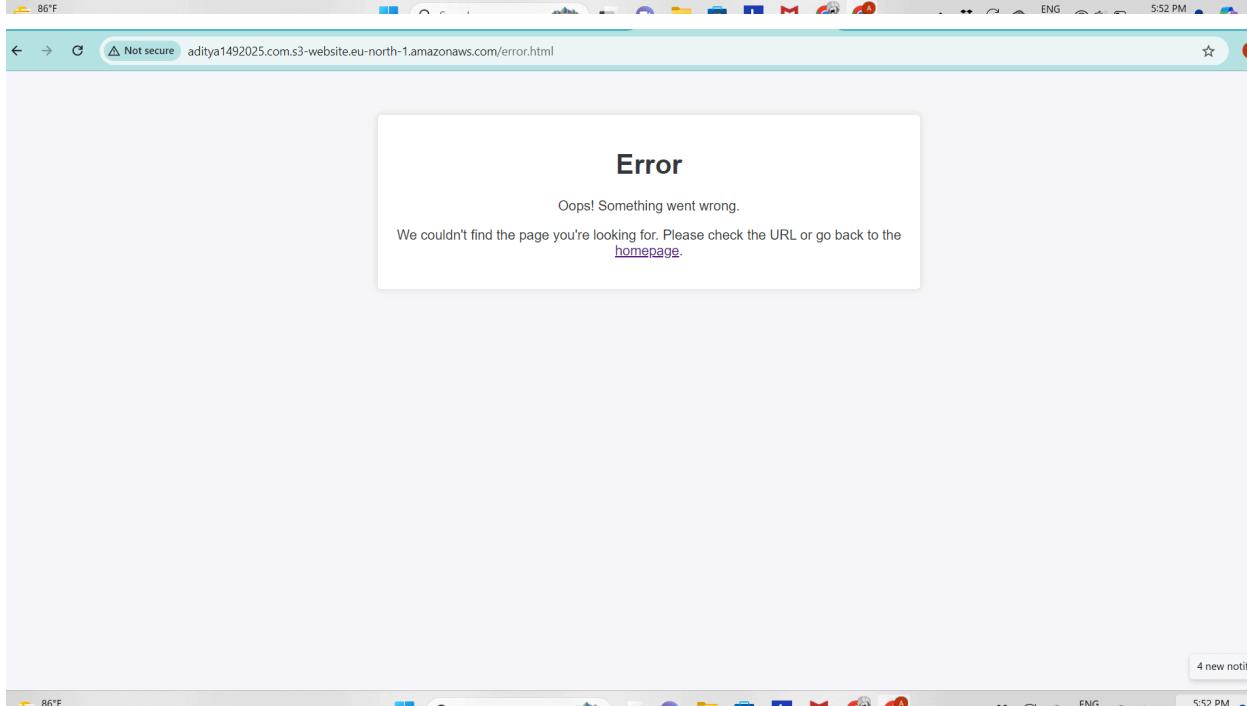
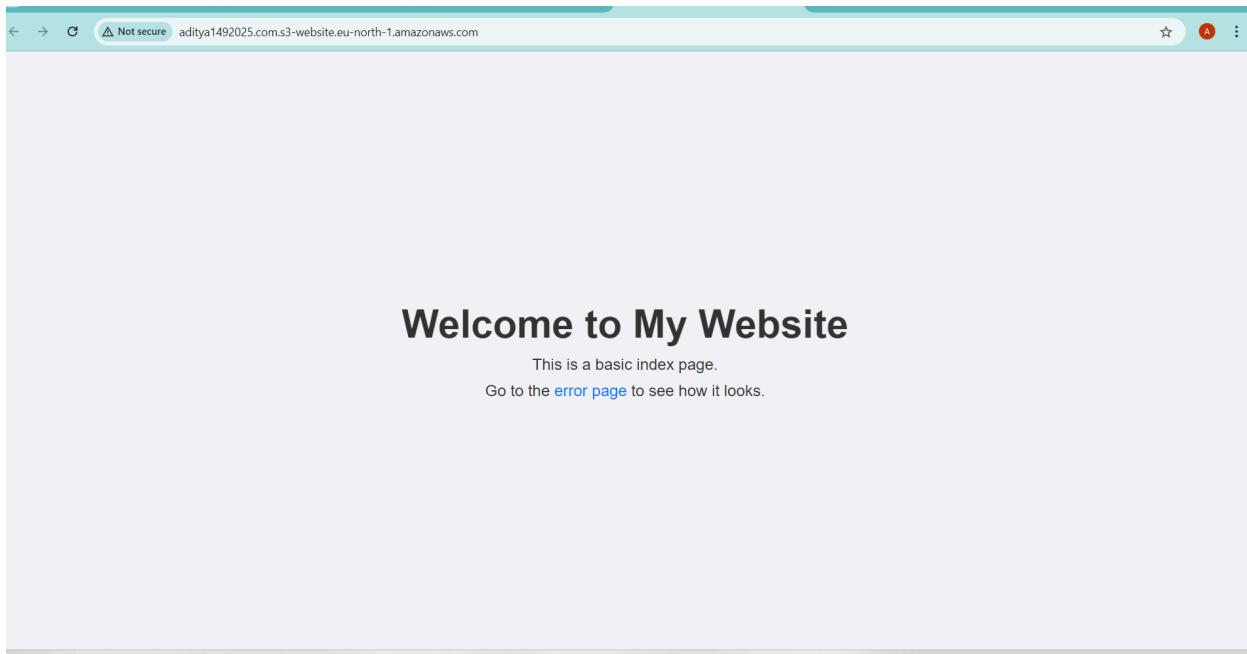
At the bottom of the page, there are links for CloudShell, Feedback, and a copyright notice: © 2024 Amazon Web Services, Inc. or its affiliates.

Step 16: Now reload the website. You can see your website

Name:Aditya Dubey

Div:D15C

Roll no:10



Experiment No: 1(B)

Step 1: Set up Cloud9 environment.

- 1) Go to Cloud9 services under developers tool in All services

The screenshot shows the AWS Management Console with the search bar set to 'Cloud9'. The results are categorized as follows:

- Developer Tools:** Cloud9, CloudShell, X-Ray, AWS FIS, CodeArtifact, Amazon CodeCatalyst, AWS AppConfig, Amazon Q Developer (Including Amazon CodeWhisperer), Application Composer, AWS App Studio.
- Analytics:** Athena, Amazon Redshift, CloudSearch, Amazon OpenSearch Service.
- Customer Enablement:** AWS IQ, Managed Services.
- End User Computing:** WorkSpaces, AppStream 2.0, WorkSpaces Secure Browser, WorkSpaces Thin Client.
- Internet of Things:** IoT Analytics, IoT Device Defender, IoT Device Management, IoT Greengrass, IoT SiteWise, IoT Core, IoT Events, AWS IoT FleetWise, IoT TwinMaker.
- Game Development:** Amazon GameLift.
- Others:** Direct Connect, AWS App Mesh, Global Accelerator, AWS Cloud Map, Route 53 Application Recovery Controller, AWS Private 5G, Amazon Kendra, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer, AWS DeepRacer, AWS Panorama, Amazon Monitron, AWS HealthLake, Amazon Lookout for Vision, Amazon Lookout for Equipment, Amazon Lookout for Metrics, Amazon Lex, Amazon Comprehend Medical, AWS Healthomics, Amazon Bedrock, AWS HealthImaging, Amazon Q, Amazon Q Business.

- 2) Click on create environment

The screenshot shows the AWS Cloud9 landing page. The main heading is "AWS Cloud9" with the subtext "A cloud IDE for writing, running, and debugging code". A prominent orange "Create environment" button is centered. Below it, there are two main sections:

- How it works:** A box containing text about creating a development environment on an EC2 instance or connecting to an existing Linux server via SSH, and how the AWS Cloud9 dashboard allows switching between multiple environments.
- Getting started:** A box with links to "Before you start (2 min read)", "Create an environment (2 min read)", "Working with environments (15 min read)", and "Working with the IDE (10 min read)".

At the bottom, there are links for "CloudShell" and "Feedback".

3) Give the name to your Environment ,keeping the other settings as default

The screenshot shows the 'Create environment' page in the AWS Cloud9 interface. The 'Details' tab is selected. In the 'Name' field, 'WebAppIDE' is entered. The 'Description - optional' field is empty. Under 'Environment type', the 'New EC2 instance' option is selected, which is highlighted with a blue border. The 'Existing compute' option is also available but not selected. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information.

4) Select the correct platform type as shown below and keep the others details as default

The screenshot shows the 'New EC2 instance' configuration page. Under 'Instance type', the 't2.micro (1 GiB RAM + 1 vCPU)' option is selected and highlighted with a blue border. Other options like 't3.small (2 GiB RAM + 2 vCPU)' and 'm5.large (8 GiB RAM + 2 vCPU)' are also listed. Below this, there's a section for 'Additional instance types' with a link to explore more. Under 'Platform', 'Amazon Linux 2023' is selected. The 'Timeout' setting is set to '30 minutes'. At the bottom, there are 'Network settings' and other configuration tabs. The footer includes links for 'CloudShell', 'Feedback', and copyright information.

Name:Aditya Dubey

Div:D15C

Roll No:10

5) Click on SSH under connection type in network settings and click on Create

The screenshot shows the AWS Cloud9 'Create Environment' wizard. In the 'Connection' step, 'AWS Systems Manager (SSM)' is selected. A note states: 'The following IAM resources will be created in your account'. It lists three items: 'AWS Service Role for AWS Cloud9', 'AWS Cloud9 SSM Access Role', and 'AWS Cloud9 SSM Instance Profile'. At the bottom are 'Cancel' and 'Create' buttons.

6) Successfully created the environment so now click on open

The screenshot shows the AWS Cloud9 'Environments' page. It displays a table with one environment row:

| Name | Cloud9 IDE | Environment type | Connection | Permission | Owner ARN |
|-----------|------------|------------------|--------------------|------------|---|
| WebAppIDE | Open | EC2 instance | Secure Shell (SSH) | Owner | arn:aws:sts::38055794473:assumed-role/voclabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR |

The 'Open' button for the 'WebAppIDE' environment is highlighted.

Step 2: Creating IAM user.

- 1) Search IAM on the services search bar and open it. Click on Create User

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with navigation links like Dashboard, Access management, and Access reports. The main area displays 'IAM resources' with counts: 0 User groups, 0 Users, 20 Roles, 4 Policies, and 0 Identity providers. Below this is a 'What's new' section with several recent updates. To the right, there are two panels: 'AWS Account' (Account ID: 380557944473, Sign-in URL: https://380557944473.signin.aws.amazon.co m/console) and 'Tools' (Policy simulator). The bottom of the screen includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

- 2) Click on the create user

The screenshot shows the 'Users' page under the IAM service. The sidebar is identical to the previous dashboard view. The main area shows a table titled 'Users (0) info' with a note: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar and a 'Create user' button are at the top of the table. The table has columns for User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. A message 'No resources to display' is shown below the table. The bottom of the screen includes the same footer links as the previous screenshot.

Name:Aditya Dubey

Div:D15C

Roll No:10

3) Write the name of the user you want to add and click on next

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. The 'User name' field contains 'apsit'. Below it, there's a note about character restrictions and an optional checkbox for console access. A callout box provides instructions for generating programmatic access keys. Navigation buttons 'Cancel' and 'Next' are at the bottom.

4) Click on the drop down menu of the set permissions boundary

The screenshot shows the 'Set permissions' step. It features a 'Permissions options' section with three choices: 'Add user to group' (selected), 'Copy permissions', and 'Attach policies directly'. A 'Get started with groups' callout suggests creating a group and attaching policies. A 'Set permissions boundary - optional' section is shown below. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

- 5) Click on the checkbox and search for cloud9 under permissions policies ,click on next

The screenshot shows the AWS IAM Permissions Policies page. At the top, there is a note about setting a permissions boundary. Below it, a checkbox is checked for "Use a permissions boundary to control the maximum permissions". A search bar at the top right contains the text "cloud9". The main table lists five AWS managed policies:

| Policy name | Type | Attached entities |
|-----------------------------|-------------|-------------------|
| AWSCloud9Administrator | AWS managed | 0 |
| AWSCloud9EnvironmentMember | AWS managed | 0 |
| AWSCloud9ServiceRolePolicy | AWS managed | 1 |
| AWSCloud9SSMInstanceProfile | AWS managed | 0 |
| AWSCloud9User | AWS managed | 0 |

At the bottom right, there are "Cancel", "Previous", and "Next" buttons. The "Next" button is highlighted in orange.

- 6) Scroll down and click on create user

The screenshot shows the AWS IAM Create User wizard, Step 1: Specify user details. The left sidebar shows steps: Step 1 (Specify user details), Step 2 (Set permissions), and Step 3 (Review and create). The main area is titled "Review and create" and contains the "User details" section. It shows a user name "apsit", a console password type "None", and a "Require password reset" option set to "No". Below this is the "Permissions summary" section, which shows "No resources". The "Tags - optional" section indicates "No tags associated with the resource".

Name:Aditya Dubey

Div:D15C

Roll No:10

The screenshot shows the 'Create user' page in the AWS IAM console. At the top, an error message is displayed: 'User was not created. User: arn:aws:sts::380557944473:assumed-role/voclabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR is not authorized to perform: iam>CreateUser on resource: arn:aws:iam::380557944473:user/apsit because no identity-based policy allows the iam>CreateUser action'. Below the error message, there is a search bar with 'apsit' and a dropdown menu set to 'None'. The main section is titled 'Permissions summary' and shows a table with one row: 'Name' (apsit), 'Type' (AWS Lambda function), and 'Used as' (No resources). Below the table, there is a section for 'Tags - optional' with a note: 'Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.' A button labeled 'Add new tag' is present, along with a note: 'You can add up to 50 more tags.' At the bottom right, there are 'Cancel', 'Previous', and 'Create user' buttons. The footer includes links for CloudShell, Feedback, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

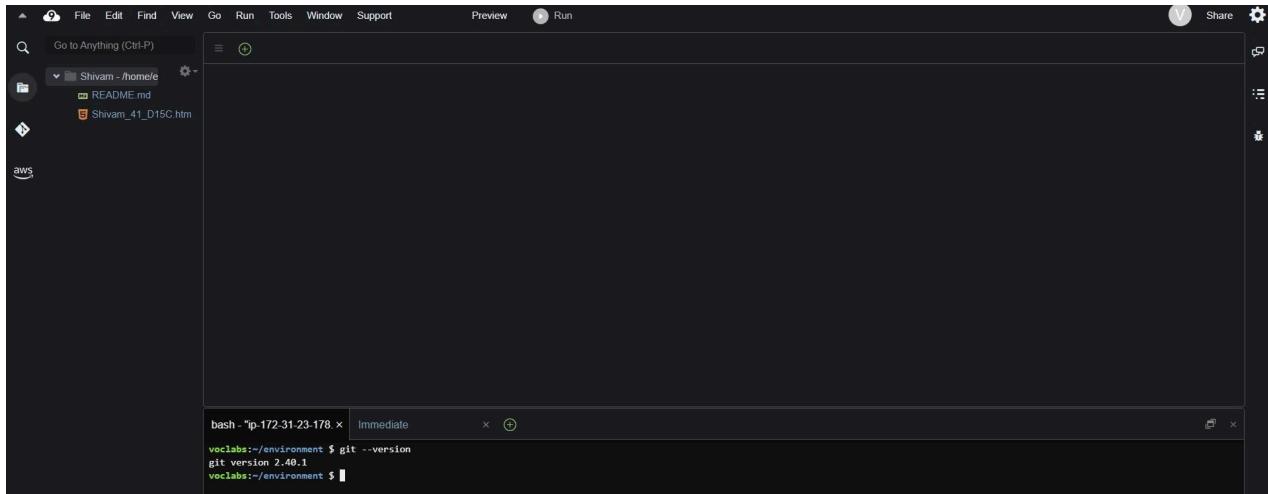
When we go to add user to a group, the AWS Academy account throws an error as we do not have the permissions to create a group. So we have to use our personal AWS account for this part.

Step 3: Working on Cloud9 IDE

1) Go to Cloud9 services. Click on Open under Cloud9 IDE

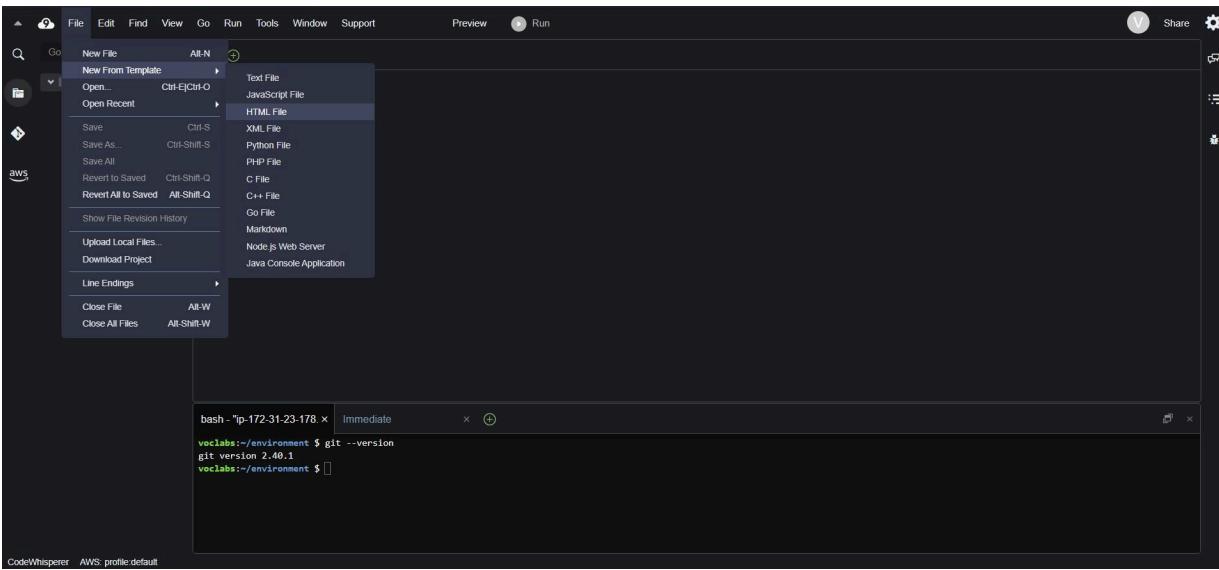
The screenshot shows the 'Environments' page in the AWS Cloud9 service. A green banner at the top indicates: 'Successfully created WebApplDE. To get the most out of your environment, see Best practices for using AWS Cloud9'. On the left, a sidebar lists 'My environments', 'Shared with me', 'All account environments', and 'Documentation'. The main area displays a table titled 'Environments (1)'. The table has columns: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. One environment is listed: 'WebApplDE' (Status: Open, Type: EC2 instance, Connection: Secure Shell (SSH), Permission: Owner, Owner ARN: arn:aws:sts::380557944473:assumed-role/voclabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR). At the bottom right of the table, there are 'Delete', 'View details', 'Open in Cloud9', and 'Create environment' buttons. The footer includes links for CloudShell, Feedback, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

- 2) This is the Cloud9 IDE interface. The major part of the screen is the coding IDE. There is a command console just below it. For example, the command git --version is run.



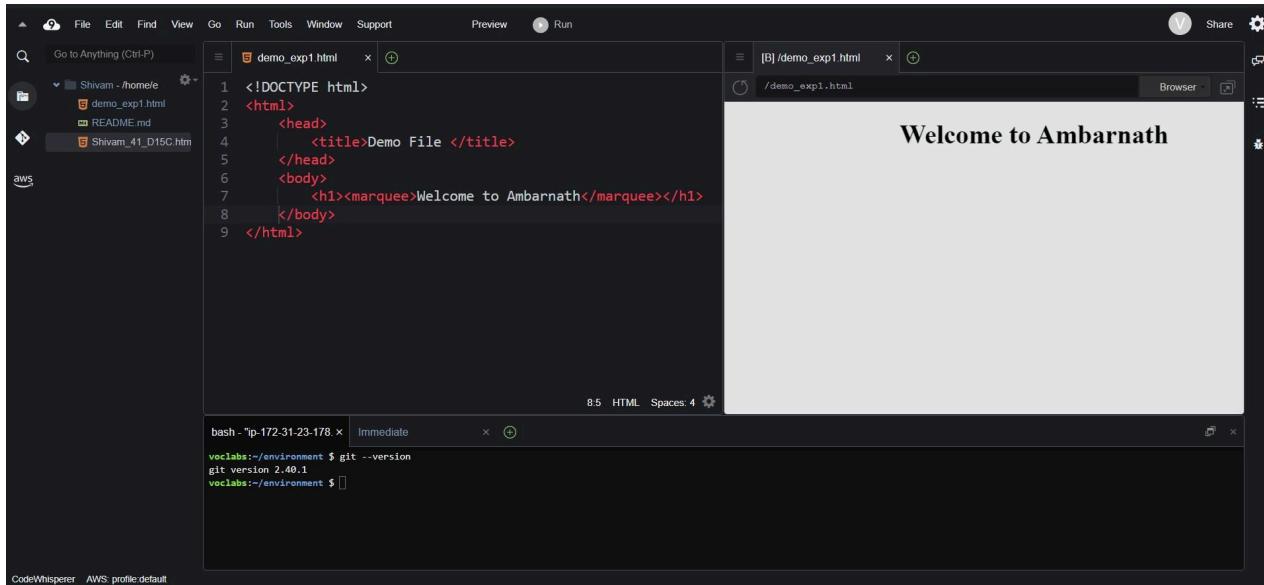
The screenshot shows the Cloud9 IDE interface. At the top is a navigation bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. Below the navigation bar is a file explorer sidebar showing a directory structure under 'Shivam - /home/vc'. The main workspace contains a terminal window titled 'bash - *ip-172-31-23-178.x' with the command 'git --version' run, displaying the output 'git version 2.40.1'. The bottom status bar indicates 'CodeWhisperer AWS profile default'.

- 3) To add a file, click on file. For this experiment, we are to add an HTML file. So go to File → New From Template → HTML file. This gives a basic HTML template on the coding IDE



The screenshot shows the Cloud9 IDE interface with the 'File' menu open. The 'File' menu includes options like New File, New From Template (which is currently selected), Open, Open Recent, Save, Save As, Save All, Revert to Saved, Show File Revision History, Upload Local Files, Download Project, Line Endings, Close File, and Close All Files. A sub-menu 'New From Template' is open, showing options for Text File, JavaScript File, HTML File, XML File, Python File, PHP File, C File, C++ File, Go File, Markdown, Node.js Web Server, and Java Console Application. The main workspace below shows a terminal window with the command 'git --version' run, displaying the output 'git version 2.40.1'. The bottom status bar indicates 'CodeWhisperer AWS profile default'.

4) Make a basic website on the HTML template and save it.



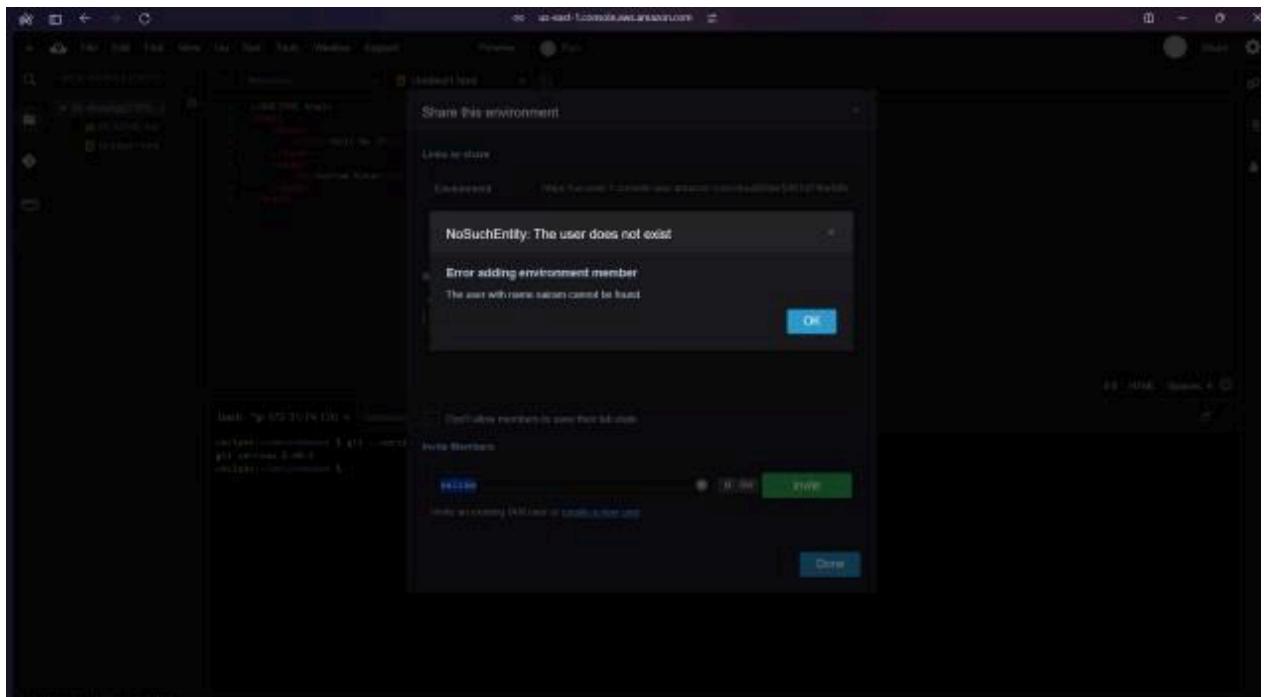
The screenshot shows the Cloud9 IDE interface. On the left, there's a file tree with files like demo_exp1.html, README.md, and Shivam_41_D15C.htm. The main editor window contains the following HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo File </title>
  </head>
  <body>
    <h1><marquee>Welcome to Ambarnath</marquee></h1>
  </body>
</html>
```

To the right of the editor is a browser preview window showing the output: "Welcome to Ambarnath". Below the editor, there's a terminal window with the following command history:

```
bash - *ip-172-31-23-178 ~ Immediate
vocabs:~/environment $ git --version
git version 2.40.1
vocabs:~/environment $
```

After saving, on the toolbar towards far right, click on Share. Then put the username that you had put during creating IAM user.



Here, it gives an error as Cloud9 was created on the academy account where creating an IAM group is not available, meanwhile on the personal account, the services of Cloud9 have been deprecated. So currently, it is not possible to integrate the cloud9 and IAM parts of the experiment.

Name:Aditya Dubey

Div:D15C

Roll No:10

Experiment No :2

Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.

The screenshot shows the AWS Services console with the search bar containing "Elastic Beanstalk". The "Compute" category is expanded, listing EC2, Lightsail, Lambda, Batch, and Elastic Beanstalk. Other categories like Quantum Technologies, Management & Governance, and Security, Identity, & Compliance are also visible.

Step 2: Go to Elastic Beanstalk and click on Create Application

The screenshot shows the Amazon Elastic Beanstalk landing page. It features a "Get started" section with a "Create application" button and a "Pricing" section stating "There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances."

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 3: Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on NEXT.

The screenshot shows the 'Configure environment' step of the AWS Elastic Beanstalk setup. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Configure environment', which is currently active. Step 2 is 'Configure service access', Step 3 is 'optional' (Set up networking, database, and tags), Step 4 is 'optional' (Configure instance traffic and scaling), Step 5 is 'optional' (Configure updates, monitoring, and logging), and Step 6 is 'Review'. The main panel shows the 'Environment tier' section with 'Web server environment' selected. Below it is the 'Application information' section where the 'Application name' is set to 'FirstWebApp'. There is also a section for 'Application tags (optional)'.

(Scroll Down)

The screenshot shows the 'Environment information' step of the AWS Elastic Beanstalk setup. The 'Environment name' field is filled with 'FirstWebApp-env'. The 'Domain' field contains 'Leave blank for autogenerated value' and '.us-east-1.elasticbeanstalk.com', with a 'Check availability' button next to it. The 'Environment description' field is empty. Below this, the 'Platform' section is visible, showing 'Platform type' with 'Managed platform' selected. The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

(Scroll Down)

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

The screenshot shows the 'Platform Info' configuration page. It includes fields for Platform type (set to Managed platform), Platform (set to PHP), Platform branch (set to PHP 8.3 running on 64bit Amazon Linux 2023), and Platform version (set to 4.3.1 (Recommended)). Below this is the 'Application code Info' section.

Platform type

- Managed platform
- Custom platform

Platforms published and maintained by Amazon Elastic Beanstalk. Learn more [\[link\]](#)

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

Application code Info

The screenshot shows the 'Application code Info' configuration page. It includes options for Sample application, Existing version (disabled), and Upload your code (disabled). Below this is the 'Presets Info' section.

Application code Info

- Sample application
- Existing version
- Upload your code

Presets Info

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- Single instance (free tier eligible)
- Single instance (using spot instance)
- High availability
- High availability (using spot and on-demand instances)
- Custom configuration

Cancel Next

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

(Click on Next)

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 4: Use an existing service role and choose whatever service role is present on your account

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk setup wizard. On the left sidebar, steps 1 through 6 are listed. Step 1 is 'Configure environment', Step 2 is 'Configure service access' (which is currently active), Step 3 is 'Set up networking, database, and tags' (optional), Step 4 is 'Configure instance traffic and scaling' (optional), Step 5 is 'Configure updates, monitoring, and logging' (optional), and Step 6 is 'Review'. The main content area is titled 'Configure service access' and contains a 'Service access' section. It explains that IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. A note says 'Learn more'. Below this, there's a 'Service role' section with two options: 'Create and use new service role' (radio button not selected) and 'Use an existing service role' (radio button selected). Under 'Existing service roles', it says 'Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.' A dropdown menu shows 'LabRole' selected. Below that is an 'EC2 key pair' section with a dropdown showing 'vokey' selected. At the bottom is an 'EC2 instance profile' section with a dropdown showing 'LabInstanceProfile' selected. At the very bottom of the page, there are links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 5: Click on Skip to Review

This screenshot shows the same 'Configure service access' step as the previous one, but with a different view. The main content area is identical, showing the 'Service role' section with 'Use an existing service role' selected and 'LabRole' chosen. Below the main content, there is a 'View permission details' button. At the bottom of the page, there are four buttons: 'Cancel', 'Skip to review' (highlighted in orange), 'Previous', and 'Next'.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 6: Review the settings that you have set up for your application and submit your application

The screenshot shows the AWS Elastic Beanstalk 'Review' step. On the left, a sidebar lists steps: 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), and Step 6 (Review). The main area is titled 'Step 1: Configure environment' with an 'Edit' button. It contains 'Environment information' with fields: Environment tier (Web server environment), Application name (FirstWebApp), Environment name (FirstWebApp-env), Application code (Sample application), and Platform (arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1). Below this is 'Step 2: Configure service access' with an 'Edit' button. It contains 'Service access' with an 'Info' link. At the bottom, there are links for CloudShell, Feedback, and navigation icons.

(Scroll Down)

The screenshot shows the AWS Elastic Beanstalk 'Review' step. The sidebar shows Step 1 (Configure environment) is selected. The main area displays environment properties. Under 'Environment properties', there is a table with columns 'Key' and 'Value'. The table shows one entry: 'No environment properties' with the note 'There are no environment properties defined'. At the bottom, there are 'Cancel', 'Previous', and 'Submit' buttons. The 'Submit' button is highlighted in orange.

(Click on the Submit)

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

The screenshot shows the AWS Elastic Beanstalk console. The main title bar says "Environment successfully launched." Below it, the navigation path is "Elastic Beanstalk > Environments > FirstWebApp-env". The central panel is titled "FirstWebApp-env" with a "Info" link. It displays an "Environment overview" table with two rows: "Health" (Ok) and "Domain" (FirstWebApp-env.eba-emitmfei.us-east-1.elasticbeanstalk.com). To the right, there's a "Platform" section showing "Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1" and "Platform state: Supported". At the bottom, tabs for "Events", "Health", "Logs", "Monitoring", "Alarms", "Managed updates", and "Tags" are visible. The left sidebar has sections for Applications, Environments, Change history, Application: FirstWebApp (Application versions, Saved configurations), and Environment: FirstWebApp-env (Go to environment, Configuration, Events, Health, Logs, Monitoring).

(Click on the link under domain it will redirect to a new page)

The screenshot shows a web browser displaying the deployed application. The title bar says "Not secure firstwebapp-env.eba-emitmfei.us-east-1.elasticbeanstalk.com". The main content features a large "Congratulations!" heading. Below it, text states: "Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud". It also mentions "You are running PHP version 8.3.7" and "This environment is launched with Elastic Beanstalk PHP Platform". On the right side, there are two sections: "What's Next?" and "AWS SDK for PHP". Both sections contain links to further resources.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 7 : Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>

The screenshot shows the GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0'. The repository is public and has 20 commits. The commit history includes updates to README.md, .github, dist, scripts, and various configuration files like CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE, and app-specification.yml. The repository has 425 forks and 4 stars. A prominent button at the top right says 'Fork your own copy of imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0'.

The screenshot shows the 'Create a new fork' form on GitHub. It asks for the owner (Kingmaker-2) and repository name (aws-codepipeline-s3-codedeploy-linux-2.0). A note says 'aws-codepipeline-s3-codedeploy-linux-2.0 is available.' Below this, it says 'By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.' There is a description field with placeholder text: 'Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.' A checked checkbox says 'Copy the master branch only' with a note: 'Contribute back to imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0 by adding your own branch.' At the bottom, there is a note: 'You are creating a fork in your personal account.' and a 'Create fork' button.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 8: Search CodePipeline in the services tab and click on it.

The screenshot shows the AWS Services Catalog interface. A search bar at the top contains the text 'CodePipeline'. Below the search bar, the results are displayed in a grid format. The 'Developer Tools' category is expanded, showing 'CodePipeline' as the first item. Other services listed in this category include CodeStar, CodeCommit, CodeBuild, CodeDeploy, Cloud9, CloudShell, X-Ray, AWS FIS, CodeArtifact, Amazon CodeCatalyst, AWS AppConfig, Amazon Q Developer (Including Amazon CodeWhisperer), Application Composer, AWS App Studio, and Customer Enablement. To the right of the developer tools, there are sections for End User Computing, Internet of Things, Analytics, and Game Development, each containing several AWS services. At the bottom of the page, there is a navigation bar with links for Privacy, Terms, and Cookie preferences.

Step 9: Click on Create Pipeline.

The screenshot shows the AWS CodePipeline Pipelines page. On the left, a sidebar menu for 'CodePipeline' is visible, with options like Source, Artifacts, Build, Deploy, Pipeline, Getting started, Pipelines, and Settings. The main content area displays a message about the new V2 pipeline type. Below this, a table titled 'Pipelines' is shown with columns for Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A search bar and a 'Create pipeline' button are located at the top of the table. At the bottom of the page, there are links for CloudShell and Feedback, along with standard AWS footer links for Privacy, Terms, and Cookie preferences.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 10: Give a name to your Pipeline. A new service role would be created with the name of the pipeline.

The screenshot shows the 'Create Pipeline' wizard on the AWS CodePipeline console. The current step is 'Set Execution mode and Service role'. The 'Execution mode' section is expanded, showing three options: 'Superseded' (radio button), 'Queued (Pipeline type V2 required)' (radio button, selected), and 'Parallel (Pipeline type V2 required)'. The 'Service role' section is also expanded, showing two options: 'New service role' (radio button, selected) and 'Existing service role'. Below these sections, the 'Role name' field contains 'AWSCodePipelineServiceRole-us-east-1-MyPipeline', and the 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' checkbox is checked. The 'Variables' section is collapsed, showing a note about adding variables at the pipeline level. At the bottom, there are 'CloudShell', 'Feedback', and 'Cookie preferences' buttons, along with copyright and legal links.

This screenshot continues from the previous one, showing the 'Create Pipeline' wizard on the AWS CodePipeline console. The 'Execution mode' and 'Service role' sections are collapsed. The 'Variables' section is expanded, showing a note about adding variables at the pipeline level and a warning that the first pipeline execution will fail if variables have no default values. The 'Advanced settings' section is collapsed. At the bottom, there are 'Cancel' and 'Next' buttons.

Step 11: Select a source provider (as GitHub Version (2)). Click on connect to Github (*This part have to be done in the personal account of aws as in academy account it wont allow you to create pipeline with github version 1 or 2*)

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

The screenshot shows the 'Add source stage' step in the AWS CodePipeline pipeline creation wizard. The 'Source' provider is set to 'GitHub (Version 2)'. A callout box highlights the 'New GitHub version 2 (app-based) action' feature, which allows users to create a connection using GitHub Apps. Below this, there are fields for 'Connection', 'Repository name', and 'Default branch'. The 'Connection' field has a search bar and a 'Connect to GitHub' button.

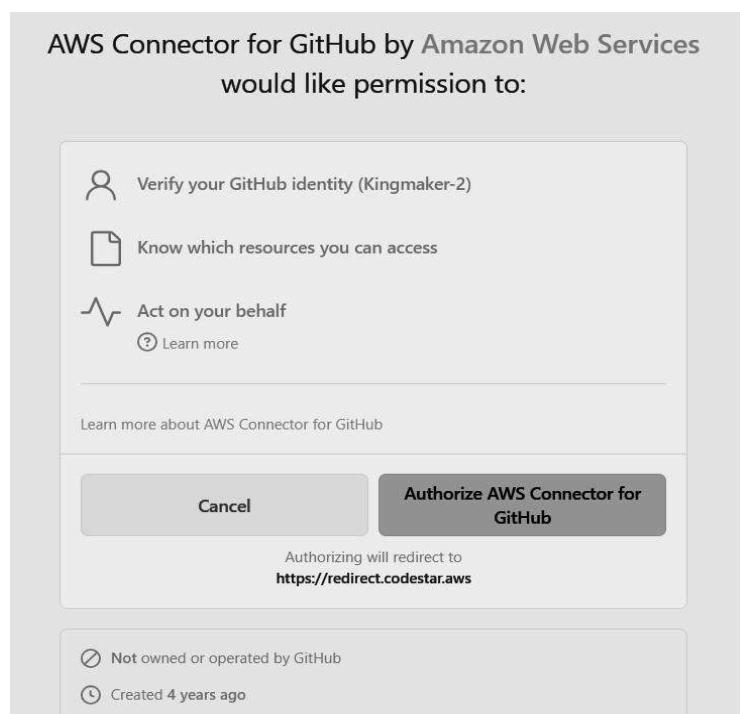
Step 12: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app.

The screenshot shows the 'Create a connection' page for GitHub. The 'Create GitHub App connection' section is active, with a 'Connection name' field containing 'MyGitHub'. A 'Tags - optional' section is present below. At the bottom right is a prominent orange 'Connect to GitHub' button. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice for 2024, Amazon Web Services, Inc. or its affiliates.

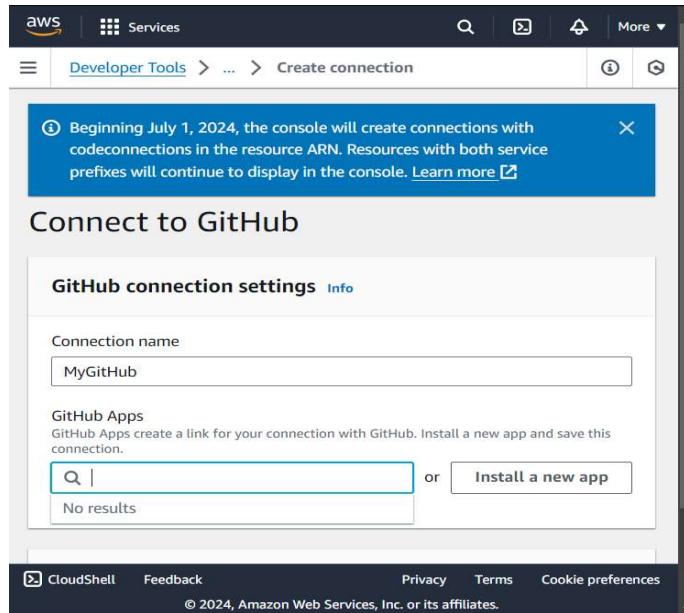
Name:Aditya Dubey

Div:D1
5C

Roll
No:10



(Click on Authorize)

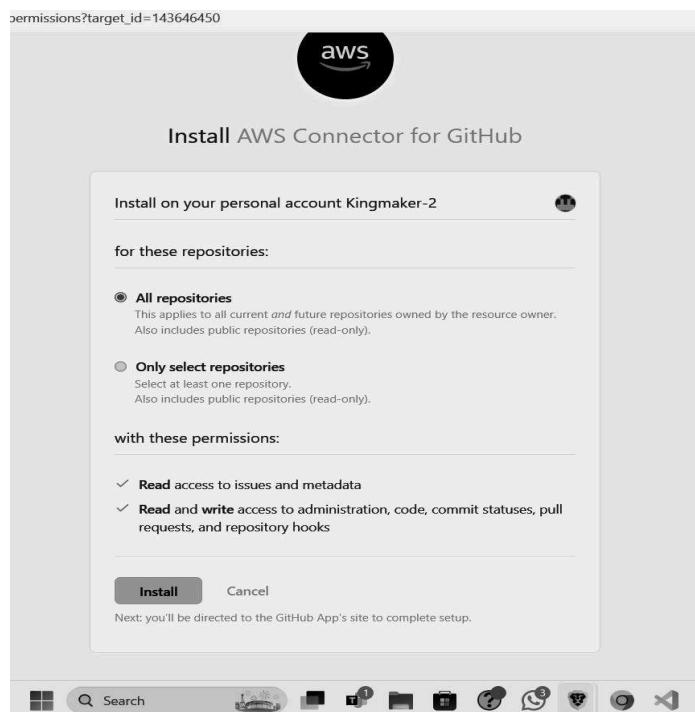


Name:Aditya Dubey

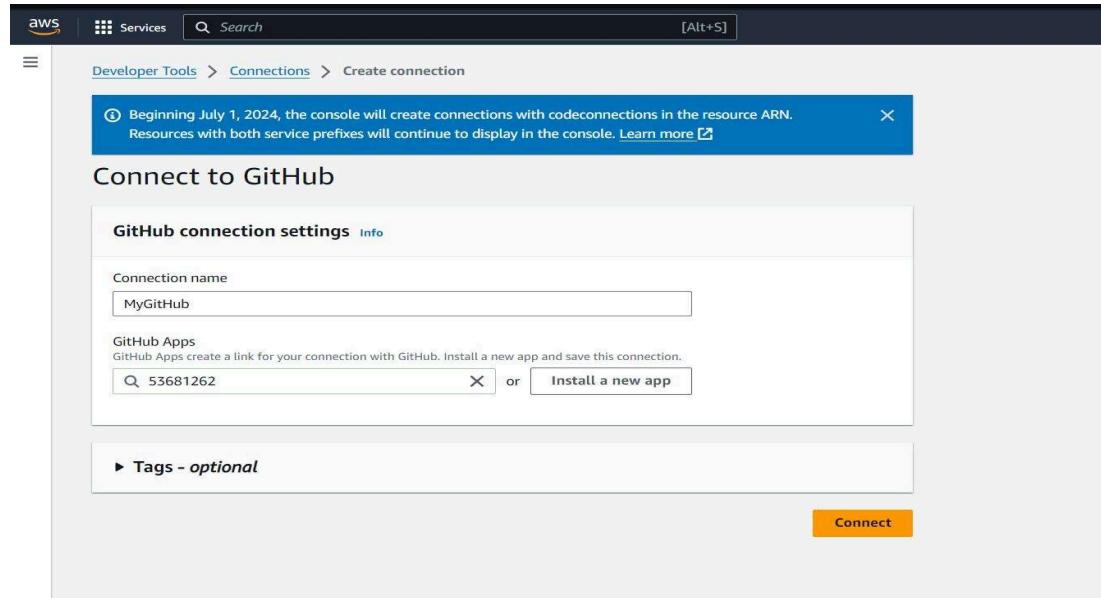
Div:D1
5C

Roll
No:10

Step 13: This will direct you to install AWS connector on your GitHub .Install it to your account and give it its permissions



Step 14: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use.



Name:Aditya Dubey

Div:D1
5C

Roll
No:10

The screenshot shows the 'Add source stage' step in the AWS CodePipeline 'Create new pipeline' wizard. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage, currently selected), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main panel is titled 'Source' and shows the 'Source provider' dropdown set to 'GitHub (Version 2)'. A callout box provides information about the new GitHub version 2 action. Below it, the 'Connection' section shows a connection named 'arn:aws:codeconnections:us-east-1:011528263337:connection/b7859e8a-5f' with a 'Connect to GitHub' button. A green box at the bottom indicates the GitHub connection is ready for use.

Step 15: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

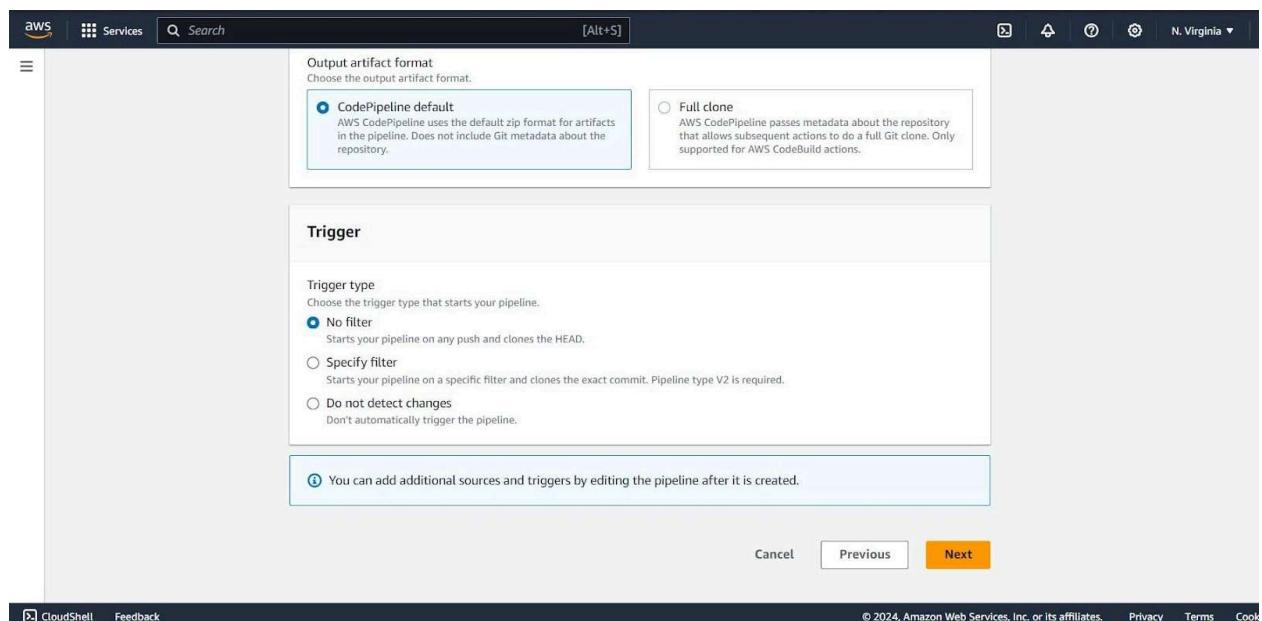
The screenshot shows the 'Repository name' configuration step. It includes fields for 'Repository name' (set to 'Kingmaker-2/aws-codepipeline-s3-codedeploy-linux-2.0') and 'Default branch' (set to 'master'). Under 'Output artifact format', the 'CodePipeline default' option is selected. The 'Trigger' section shows 'Trigger type' options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'.

Name:Aditya Dubey

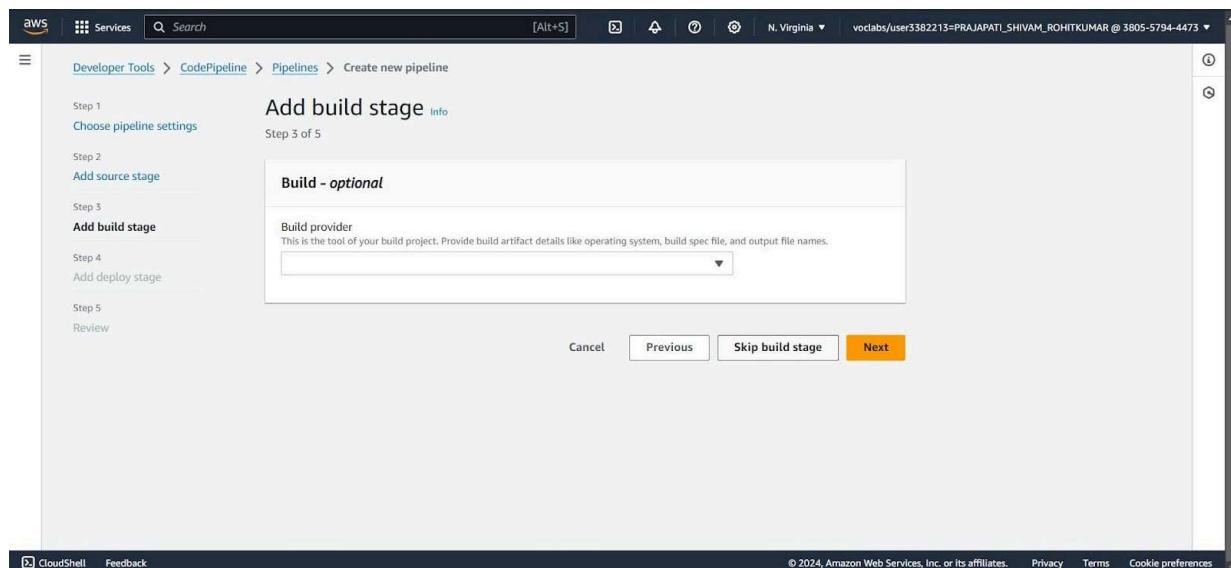
Div:D1
5C

Roll
No:10

Step 16: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.



Step 17: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.



Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Screenshot of the AWS CodePipeline 'Create new pipeline' wizard, Step 4 of 5: Add deploy stage.

The 'Deploy provider' dropdown is empty, and the 'Configure automatic rollback on stage failure' checkbox is unchecked.

Buttons at the bottom: Cancel, Previous, Next.

Screenshot of the AWS CodePipeline 'Create new pipeline' wizard, Step 4 of 5: Add deploy stage.

The 'Deploy provider' dropdown shows 'AWS Elastic Beanstalk' selected, and the 'Region' dropdown shows 'US East (N. Virginia)'.

Input artifacts dropdown shows 'SourceArtifact' selected.

Buttons at the bottom: Cancel, Previous, Next.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

The screenshot shows the AWS CloudFormation console interface for creating a pipeline. The current step is "Step 3: Add build stage". The configuration includes:

- Deployment provider: AWS Elastic Beanstalk
- Region: US East (N. Virginia)
- Input artifact: SourceArtifact
- Application name: FirstWebApp
- Environment name: FirstWebApp-env
- A checkbox for "Configure automatic rollback on stage failure" is checked.

At the bottom, there are "Cancel", "Previous", and "Next" buttons. The "Next" button is highlighted in orange.

Step 18: Check all the information and click on create Pipeline

The screenshot shows the AWS CloudFormation console interface for creating a pipeline. The current step is "Step 4: Add deploy stage". The configuration includes:

- Deploy action provider: AWS Elastic Beanstalk
- ApplicationName: FirstWebApp
- EnvironmentName: FirstWebApp-env
- A checkbox for "Configure automatic rollback on stage failure" is checked and set to "Enabled".

At the bottom, there are "Cancel", "Previous", and "Create pipeline" buttons. The "Create pipeline" button is highlighted in orange.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 19: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy

The screenshot shows the AWS CodePipeline console. At the top, a green banner says "Success Congratulations! The pipeline MyPipeline1 has been created." Below the banner, the pipeline name "MyPipeline1" is displayed with a status of "QUEUED". The pipeline type is "V2" and the execution mode is "QUEUED". On the left, a sidebar menu for "CodePipeline" includes options like Source, Artifacts, Build, Deploy, Pipeline, History, Settings, and Settings. The "Source" section is expanded, showing a step named "GitHub (Version 2)" with a status of "Succeeded - Just now" and a commit ID "8fd5da54". A "View details" button is present. A "Disable transition" button is located at the bottom of the pipeline card. To the right, there are two circular status indicators: a green one with a checkmark and a blue one with a question mark.

This screenshot shows the same AWS CodePipeline interface after the pipeline has moved to the "Deploy" stage. The pipeline card now displays the "Deploy" step, which is also in a "Succeeded" state. The commit ID "8fd5da54" is listed under the Deploy step. The "Start rollback" button is visible next to the Deploy step. The sidebar menu remains the same as in the previous screenshot.

Name:Aditya Dubey

Div:D1
5C

Roll
No:10

Step 20: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.

The screenshot shows the AWS CodePipeline console. On the left, there's a sidebar with 'Developer Tools' and 'CodePipeline' selected. Under 'CodePipeline', there are sections for Source (GitHub (Version 2) - Succeeded - Just now), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), and Settings. The main area displays a pipeline execution titled '8fd5da54' which has completed successfully. It includes tabs for 'Source' (GitHub (Version 2) - Succeeded - Just now), 'Deploy' (AWS Elastic Beanstalk - Succeeded - Just now), and 'View details'. A 'Start rollback' button is also visible.

Step 21: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain

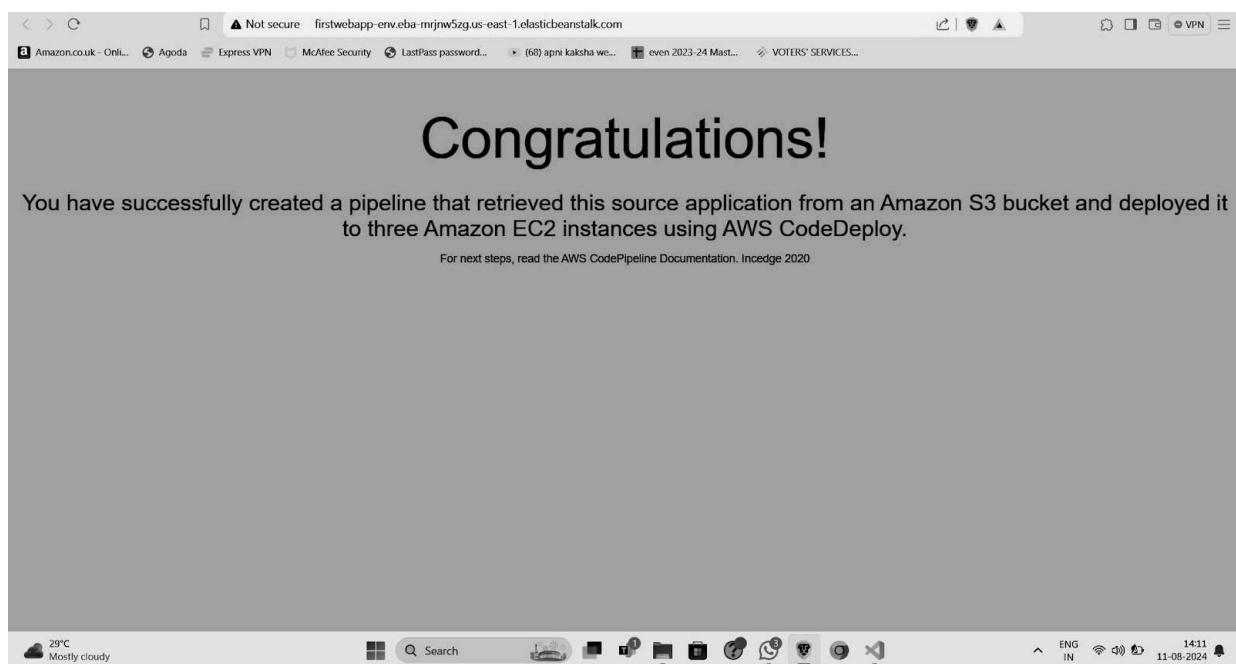
The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has 'Elastic Beanstalk' selected, with options for Applications, Environments, Change history, Application: FirstWebApp (selected), Application versions, Saved configurations, and Recent environments (FirstWebApp-env). The main area shows the 'Application FirstWebApp environments (1)' page. It lists one environment: 'FirstWebApp-env' (Health: Warning, Date created: August 11, 2024, Domain: FirstWebApp-env.eba-mrjn...). There are buttons for Actions and Create new environment. The URL in the browser bar is https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#.

Step 22: This will successfully show the sample website hosted.

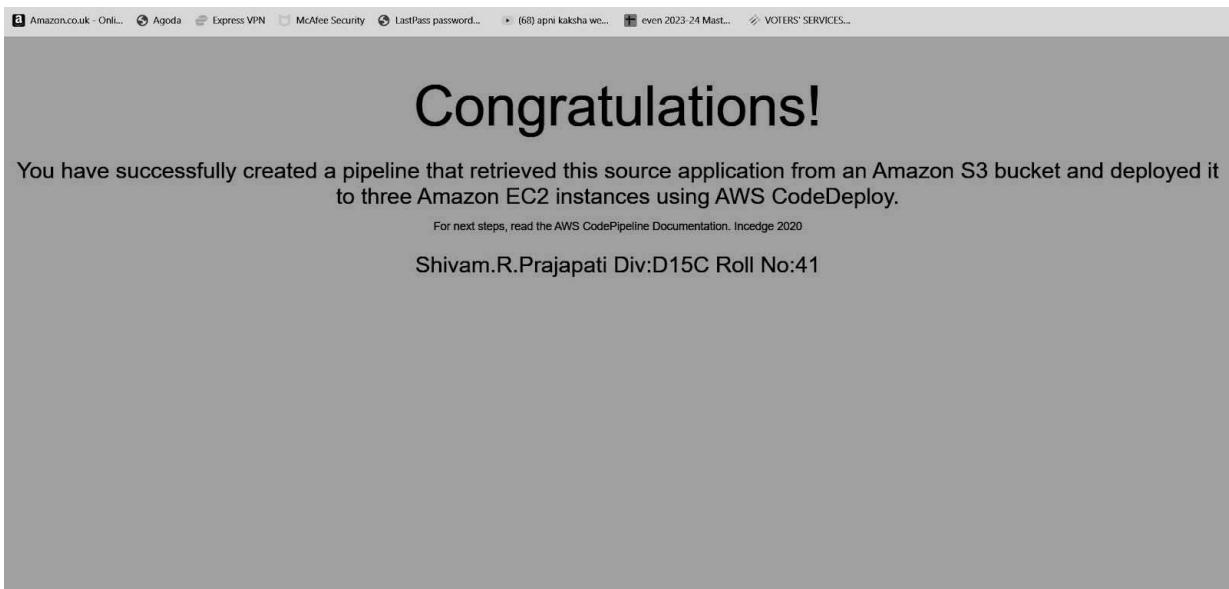
Name:Aditya Dubey

Div:D1
5C

Roll
No:10



Step 23: Now, we make some changes to the index.html file in the github. For eg: If you make some changes to the tag .Once the changes are committed ,when the website is refreshed ,the changes will be seen.



EXPERIMENT NO. 3

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

1. Create 3 EC-2 instances with all running on Amazon Linux as OS with inbound SSH allowed
To efficient run kubernetes cluster select instance type of at least t2.medium as kubernetes recommends at least 2 vCPU to run smoothly

| | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Z |
|--------------------------|----------|---------------------|--------------------------|---------------|------------------------------------|-------------------------------|----------------|
| <input type="checkbox"/> | worker-2 | i-03e737ce896f3e8ef | Running ⓘ ⓘ | t2.micro | 2/2 checks passed ⓘ ⓘ | View alarms + | us-east-1b |
| <input type="checkbox"/> | master | i-0650be926fe6acc7d | Running ⓘ ⓘ | t2.micro | 2/2 checks passed ⓘ ⓘ | View alarms + | us-east-1b |
| <input type="checkbox"/> | worker-1 | i-00f42f05b3b3762ed | Running ⓘ ⓘ | t2.micro | 2/2 checks passed ⓘ ⓘ | View alarms + | us-east-1b |

Set up Docker

Kubernetes requires a CRI-compliant container engine runtime such as [Docker](#), [containerd](#), or [CRI-O](#). This article shows you how to deploy Kubernetes using [Docker](#).

[Install Docker](#) on each server node by executing the steps below:

1. Update the package list:

```
sudo apt update
```

```
ubuntu@ip-172-31-92-255:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [351 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [123 kB]
```

2. Install Docker with the following command:

```
sudo apt install docker.io -y
```

```
root@ip-172-31-92-237:/home/ubuntu# sudo apt-get install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 130 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 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 bridge-utils amd64 1.7.1-1ubun
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubu
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.1
```

- cat <<EOF | sudo tee /etc/docker/daemon.json


```
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

```
root@ip-172-31-92-237:~# cd /etc/docker
root@ip-172-31-92-237:/etc/docker# cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
root@ip-172-31-92-237:/etc/docker#
```

- sudo systemctl enable docker
- sudo systemctl daemon-reload

```
root@ip-172-31-92-237:/home/ubuntu# sudo systemctl start docker
root@ip-172-31-92-237:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-92-237:/home/ubuntu#
```

Install Kubernetes

Setting up Kubernetes on an Ubuntu system involves adding the Kubernetes [repository](#) to the [APT](#) sources list and installing the relevant tools. Follow the steps below to install Kubernetes on all the nodes in your cluster.

Step 1: Add Kubernetes Signing Key

Since Kubernetes comes from a non-standard repository, download the signing key to ensure the software is authentic.

On each node, use the [curl command](#) to download the key and store it in a safe place (default is `/etc/apt/keyrings/`):

```
curl -fsSL
https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
root@ip-172-31-92-237:/etc/docker# curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/
ring.gpg
root@ip-172-31-92-237:/etc/docker# echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:
/etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/' |
root@ip-172-31-92-237:/etc/docker# sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb InRelease [1186 B]
Err:7 https://packages.cloud.google.com/apt kubernetes-xenial Release
  404 Not Found [IP: 172.253.122.100 443]
Get:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb Packages [9318 B]
Reading package lists... Done
E: The repository 'http://apt.kubernetes.io kubernetes-xenial' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-92-237:/etc/docker# sudo apt install kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
```

Step 2: Add Software Repositories

Kubernetes is not included in the default Ubuntu repositories. To add the Kubernetes repository to your list, enter this command on each node:

```
echo 'deb
[signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.30/deb/' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
root@master-node:/etc/docker# echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/' | sudo tee /etc/
pt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/
root@master-node:/etc/docker# sudo apt update
```

Ensure all packages are up to date:

```
sudo apt update
```

Step 3: Install Kubernetes Tools

Each Kubernetes deployment consists of three separate tools:

- Kubeadm. A tool that initializes a Kubernetes cluster by fast-tracking the setup using community-sourced [best practices](#).
- Kubelet. The work package that runs on every node and starts containers. The tool gives you command-line access to clusters.
- [Kubectl](#). The [command-line interface](#) for interacting with clusters.

Execute the following commands on each server node to install the [Kubernetes tools](#):

1. Run the install command:

```
sudo apt install kubeadm kubelet kubectl
root@ip-172-31-92-237:/etc/docker# sudo apt install kubeadm kubelet kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 130 not upgraded.
Need to get 93.5 MB of archives.
After this operation, 341 MB of additional disk space will be used.
Do you want to continue? [Y/n] v
```

2. Mark the packages as held back to prevent automatic installation, upgrade, or removal:

```
sudo apt-mark hold kubeadm kubelet kubectl
root@ip-172-31-92-237:/etc/docker# sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
```

Note: The process presented in this tutorial prevents APT from automatically updating Kubernetes. For instructions on how to update, please see the official [developers' instructions](#).

3. Verify the installation with:

```
kubeadm version
```

```
[root@ip-172-31-92-237 ~]# kubeadm version
root@master-node:/etc/docker# kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"30", GitVersion:"v1.30.5", GitCommit:"74e84a90c725047b1328ff3d589fdb1cb7a
"2024-09-12T00:17:07Z", GoVersion:"go1.22.6", Compiler:"gc", Platform:"linux/amd64"}
```

The output of the `version` command shows basic deployment information.

Note: BMC offers balanced and affordable [server instances](#) well suited for containerized services deployment. To simplify and streamline the process, deploy Kubernetes clusters on BMC using our [Rancher solution](#).

Deploy Kubernetes

With the necessary tools installed, proceed to deploy the cluster. Follow the steps below to make the necessary system adjustments, initialize the cluster, and join worker nodes.

Step 1: Prepare for Kubernetes Deployment

This section shows you how to prepare the servers for a Kubernetes deployment. Execute the steps below on each server node:

1. Disable all [swap spaces](#) with the `swapoff` command:

```
sudo swapoff -a
```

Then use the [sed command](#) below to make the necessary adjustments to the `/etc/fstab` file:

```
sudo sed -i '/ swap / s/^(\.*$)/#/g' /etc/fstab
```

```
root@ip-172-31-92-237:/home/ubuntu# sudo swapoff -a
root@ip-172-31-92-237:/home/ubuntu# sed -i '/ swap / s/^#//' /etc/fstab
```

2. Load the required containerd modules. Start by opening the containerd configuration file in a [text editor](#), such as `nano`:

```
sudo nano /etc/modules-load.d/containerd.conf
```

```
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/modules-load.d/containerd.conf
root@ip-172-31-92-237:/etc/docker#
```

3. Add the following two lines to the file:

```
overlay
br_netfilter
```

```
GNU nano 7.2
overlay
br_netfilter
```

Save the file and exit.

4. Next, use the [modprobe command](#) to add the modules:

```
sudo modprobe overlay
sudo modprobe br_netfilter
root@ip-172-31-92-237:/etc/docker# sudo modprobe overlay
root@ip-172-31-92-237:/etc/docker# sudo modprobe br_netfilter
```

5. Open the kubernetes.conf file to configure Kubernetes networking:

```
sudo nano /etc/sysctl.d/kubernetes.conf
root@ip-172-31-92-237:/etc/docker# sudo modprobe br_netfilter
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/sysctl.d/kubernetes.conf
root@ip-172-31-92-237:/etc/docker#
```

6. Add the following lines to the file:

```
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
```

```
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
net.ipv4.ip_forward = 1
```

Save the file and exit.

7. Reload the configuration by typing:

```
sudo sysctl --system  
root@ip-172-31-92-237:/etc/docker# sudo sysctl --system  
* Applying /usr/lib/sysctl.d/10-apparmor.conf ...  
* Applying /etc/sysctl.d/10-console-messages.conf ...  
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...  
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...  
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...  
* Applying /etc/sysctl.d/10-map-count.conf ...  
* Applying /etc/sysctl.d/10-network-security.conf ...  
* Applying /etc/sysctl.d/10-ptrace.conf ...  
* Applying /etc/sysctl.d/10-zero-page.conf ...  
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...  
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...  
* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...  
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...  
* Applying /etc/sysctl.d/99-sysctl.conf ...  
* Applying /etc/sysctl.d/kubernetes.conf ...  
* Applying /etc/sysctl.conf ...  
kernel.apparmor_restrict_unprivileged_userns = 1  
kernel.printk = 4 4 1 7  
net.ipv6.conf.all.use_tempaddr = 2  
net.ipv6.conf.default.use_tempaddr = 2  
kernel.kptr_restrict = 1  
kernel.sysrq = 176
```

Step 2: Assign Unique Hostname for Each Server Node

- Decide which server will be the master node. Then, enter the command on that node to name it accordingly:

```
sudo hostnamectl set-hostname master-node  
net.ipv4.ip_forward = 1  
root@ip-172-31-92-237:/etc/docker# sudo hostnamectl set-hostname master-node  
root@ip-172-31-92-237:/etc/docker#
```

- Next, [set the hostname](#) on the first worker node by entering the following command:

```
sudo hostnamectl set-hostname worker01 and worker02  
net.ipv4.ip_forward = 1  
root@ip-172-31-86-115:/etc/docker# sudo hostnamectl set-hostname worker01  
root@ip-172-31-86-115:/etc/docker#  
  
net.ipv4.ip_forward = 1  
root@ip-172-31-92-255:/etc/docker# sudo hostnamectl set-hostname worker02  
root@ip-172-31-92-255:/etc/docker#
```

- [Edit the hosts file](#) on each node by adding the [IP addresses](#) and hostnames of the servers that will be part of the cluster.

```
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/hosts  
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/hosts  
root@ip-172-31-92-237:/etc/docker#  
  
GNU nano 1.2  
127.0.0.1 localhost  
3.89.108.169 master-node  
44.202.47.176 worker01  
34.238.119.65 worker02  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
ff02::3 ip6-allhosts
```

- Restart the terminal application to apply the hostname change.

Step 3: Initialize Kubernetes on Master Node

Once you finish setting up hostnames on cluster nodes, switch to the master node and follow the steps to initialize Kubernetes on it:

1. Open the kubelet file in a text editor.

```
sudo nano /etc/default/kubelet
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/hosts
root@ip-172-31-92-237:/etc/docker# sudo nano /etc/default/kubelet
root@ip-172-31-92-237:/etc/docker#
```

2. Add the following line to the file:

```
KUBELET_EXTRA_ARGS="--cgroup-driver=cgroupfs"
```

```
GNU nano 7.2
KUBELET_EXTRA_ARGS="--cgroup-driver=cgroupfs" .
```

Save and exit.

3. Reload the configuration and restart the kubelet:

```
sudo systemctl daemon-reload && sudo systemctl restart kubelet
root@ip-172-31-92-237:/# sudo systemctl daemon-reload && sudo systemctl restart kubelet
root@ip-172-31-92-237:/#
```

6. Open the kubeadm configuration file:

```
sudo nano /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
root@ip-172-31-92-237:/# sudo nano /usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
root@ip-172-31-92-237:/#
```

7. Add the following line to the file:

```
Environment="KUBELET_EXTRA_ARGS=--fail-swap-on=false"
```

```

GNU nano 7.2
/usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf *
# Note: This dropin only works with kubeadm and kubelet v1.11+
[service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/manifests/kubelet.conf"
Environment="KUBELET_CONFIG_ARGS=--config=/var/lib/kubelet/config.yaml"
Environment="KUBELET_EXTRA_ARGS=--fail-swap-on=false"
# This is a file that "kubeadm init" and "kubeadm join" generates at runtime, populating the KUBELET_KUBEADM_ARGS variable dynamically
EnvironmentFile=/var/lib/kubelet/kubeadm-flags.env
# This is a file that the user can use for overrides of the kubelet args as a last resort. Preferably, the user should use
# the .NodeRegistration.KubeletExtraArgs object in the configuration files instead. KUBELET_EXTRA_ARGS should be sourced from
EnvironmentFile=/etc/default/kubelet
ExecStart=
ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS

```

Save the file and exit.

9. Reload the configuration and restart the kubelet:

```

sudo systemctl daemon-reload && sudo systemctl restart kubelet
root@ip-172-31-92-237:/# sudo systemctl daemon-reload && sudo systemctl restart kubelet
root@ip-172-31-92-237:/#

```

10. Finally, initialize the cluster by typing:

```

sudo kubeadm init --control-plane-endpoint=master-node
--upload-certs --ignore-preflight-errors=all
root@master-node:/etc/docker# sudo kubeadm init --control-plane-endpoint=master-node --upload-certs --ignore-preflight-errors=all
[INFO] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
[WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
[WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[WARNING Port-6443]: Port 6443 is in use
[WARNING Port-10259]: Port 10259 is in use
[WARNING Port-10257]: Port 10257 is in use
[WARNING FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists
[WARNING FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube-controller-manager.yaml already exists
[WARNING FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-scheduler.yaml already exists
[WARNING FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etcd.yaml already exists
[WARNING Port-10250]: Port 10250 is in use
[WARNING Port-2379]: Port 2379 is in use
[WARNING Port-2380]: Port 2380 is in use

```

Once the operation finishes, the output displays a `kubeadm join` command at the bottom. Make a note of this command, as you will use it to join the worker nodes to the cluster.

```

Then you can join any number of worker nodes by running the following on each as root:
kubeadm join master-node:6443 --token fjlof5.8fnxwt2begoiwzrf \
    --discovery-token-ca-cert-hash sha256:d79687f16bbb4c7d8c78a4c02995b1f6a906afa8aaefc31dc66695800c31aed6
root@master-node:/etc/docker# ^C
root@master-node:/etc/docker#

```

11. Create a [directory](#) for the Kubernetes cluster:

```
mkdir -p $HOME/.kube
root@master-node:/# mkdir -p $HOME/.kube
root@master-node:/# █
```

12. Copy the configuration file to the directory:

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

13. Change the ownership of the directory to the current user and group using the [chown command](#):

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

```
root@master-node:/etc/docker# cd ../..
root@master-node:/# mkdir -p $HOME/.kube
root@master-node:/# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@master-node:/# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@master-node:/# kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@master-node:/# kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/master-node untainted
root@master-node:/# █
```

Step 4: Deploy Pod Network to Cluster

A pod network is a way to allow communication between different nodes in the cluster. This tutorial uses the Flannel node network manager to create a pod network.

Apply the Flannel manager to the master node by executing the steps below:

1. Use `kubectl` to install Flannel:

```
kubectl apply -f
https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
root@master-node:/# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel unchanged
clusterrole.rbac.authorization.k8s.io/flannel unchanged
clusterrolebinding.rbac.authorization.k8s.io/flannel unchanged
serviceaccount/flannel unchanged
configmap/kube-flannel-cfg unchanged
daemonset.apps/kube-flannel-ds created
```

2. Untaint the node:

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

```
root@master-node:/# kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/master-node untainted
root@master-node:/#
```

Step 5: Join Worker Node to Cluster

Repeat the following steps on each worker node to create a cluster:

1. Stop and disable AppArmor:

```
sudo systemctl stop apparmor && sudo systemctl disable apparmor
```

2. Restart containerd:

```
sudo systemctl restart containerd.service
Removed "/etc/systemd/system/sysinit.target.wants/apparmor.service".
root@worker02:/etc/docker# sudo systemctl restart containerd.service
root@worker02:/etc/docker#
```

3. Apply the kubeadm join command from Step 3 on worker nodes to connect them to the master node. Prefix the command with `sudo`:

```
sudo kubeadm join [master-node-ip]:6443 --token [token]
--discovery-token-ca-cert-hash sha256:[hash]
```

```
root@worker01:/home/ubuntu# kubeadm join master-node:6443 --token fjlof5.8fnxwt2begoiwzrf
95b1f6a906afa8aaefc31dc66695800c31aed6
```

```
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.
```

Replace `[master-node-ip]`, `[token]`, and `[hash]` with the values from the `kubeadm join` command output.

4. After a few minutes, switch to the master server and enter the following command to check the status of the nodes:

```
kubectl get nodes
```

```
Every 2.0s: kubectl get nodes

NAME                      STATUS   ROLES      AGE     VERSION
ip-172-31-81-63.ec2.internal  Ready    control-plane  29m    v1.30.4
ip-172-31-87-137.ec2.internal  Ready    <none>    5m58s   v1.30.4
ip-172-31-92-18.ec2.internal  Ready    <none>    5m53s   v1.30.4
```

The system displays the master node and the worker nodes in the cluster.

Advanced DevOps Lab

Experiment 4

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

Theory:

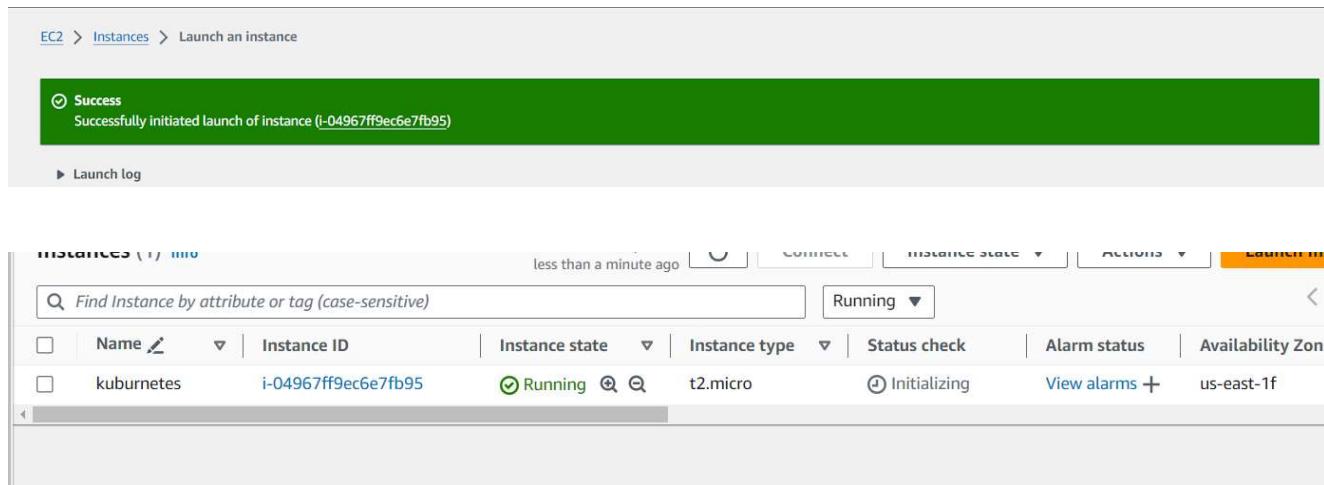
Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. In fact, Kubernetes has established itself as the defacto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), backed by key players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes Deployment

A Kubernetes Deployment is used to tell Kubernetes how to create or modify instances of the pods that hold a containerized application. Deployments can scale the number of replica pods, enable the rollout of updated code in a controlled manner, or roll back to an earlier deployment version if necessary.

Steps:

1. Create an EC2 Ubuntu Instance on AWS.



2. Edit the Security Group Inbound Rules to allow SSH

Inbound rules (1)

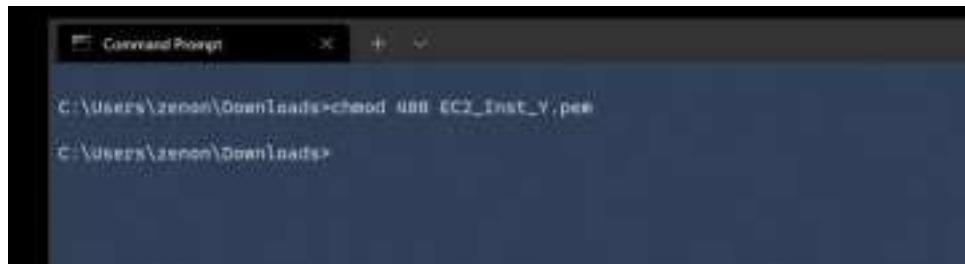
| Name | Security group rule ID | Type | Protocol | Port range | Source | Description - optional |
|-----------------------|------------------------|------|----------|------------|-------------------|------------------------|
| sgr-096cd156b62671... | SSH | TCP | 22 | My IP | 103.172.156.71/32 | |

Add rule

You have not made any changes.

Cancel Preview changes Save rules

3. SSH into the machine



```
ssh -i <keyname>.pem ubuntu@<public_ip_address>
```

```
c:\users\zenon\downloads>ssh -i EC2_Inst_Y.pem ubuntu@13.127.94.164
The authenticity of host '13.127.94.164 (13.127.94.164)' can't be established.
ECDSA key fingerprint is SHA256: [REDACTED]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

4. Install Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64]
```

```
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
```

```
ubuntu@ip-172-31-8-49:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
ubuntu@ip-172-31-8-49:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.
com/linux/ubuntu $(lsb_release -cs) stable"
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [97.7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [18.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
|
```

Then, configure cgroup in a daemon.json file.

```
cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

5. Install Kubernetes

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-8-49:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
> list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
> EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
ubuntu@ip-172-31-8-49:~$ cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
> deb https://apt.kubernetes.io/ kubernetes-xenial main
> EOF
deb https://apt.kubernetes.io/ kubernetes-xenial main
ubuntu@ip-172-31-8-49:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
0% [Waiting for headers] [Connecting to security.ubuntu.com (91.189.91.39)] {Waiting fo
```

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

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

6. Initialize the Kubecluster

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

```
ubuntu@ip-172-31-8-49:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16

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 172.31.1.1:6443 --token=43c92561ab1fb9f05855f2e8839b48ca183825d4e6c49218c8aef#tu
```

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
```

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

```
kubectl apply -f  
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/  
kube-flannel.yml  
ubuntu@ip-172-31-4-8:~/etc/docker$ kubectl apply -f https://raw.githubusercontent.com/coreos/fla  
nnel/master/Documentation/kube-flannel.yml
```

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-172-31-8-8:~/etc/kubernetes$ kubectl apply -f https://k8s.io/examples/application/deploye  
nt.yaml  
deployment.apps/nginx-deployment created
```

Use ‘kubectl get pods’ to verify if the deployment was properly created and the pod is working correctly.

Next up, create a name alias for this pod.

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

8. Lastly, port forward the deployment to your localhost so that you can view it.

```
kubectl port-forward $POD_NAME 8080:80
```

9. 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  
ubuntu@ip-172-31-4-8:~$ curl --head http://127.0.0.1:8080  
HTTP/1.1 200 OK  
Server: nginx/1.14.2  
Date: Sat, 02 Oct 2021 16:07:48 GMT  
Content-Type: text/html  
Content-Length: 612  
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT  
Connection: keep-alive  
ETag: "5c0692e1-264"  
Accept-Ranges: bytes
```

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

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

Name:Aditya Dubey

Div:D15C

Roll no:10

Step 1: Download terraform

website:<https://www.terraform.io/downloads.html>

The screenshot shows the Terraform website's 'Install Terraform' page. At the top, there's a search bar and a dropdown menu set to '1.9.4 (latest)'. Below the header, there's a navigation bar with links for 'Terraform Home', 'Install Terraform' (which is highlighted), 'Tutorials', 'Documentation', 'Registry', and 'Try Cloud'. On the left, a sidebar lists 'Operating Systems' including macOS, Windows (which is selected), Linux, FreeBSD, and OpenBSD. The main content area has a heading 'Binary download' and shows three options: 'AMD64 Version: 1.9.4' with a 'Download' button, 'ARM64 Version: 1.9.4' with a 'Download' button, and a 'Windows' section with a 'Binary download' link.

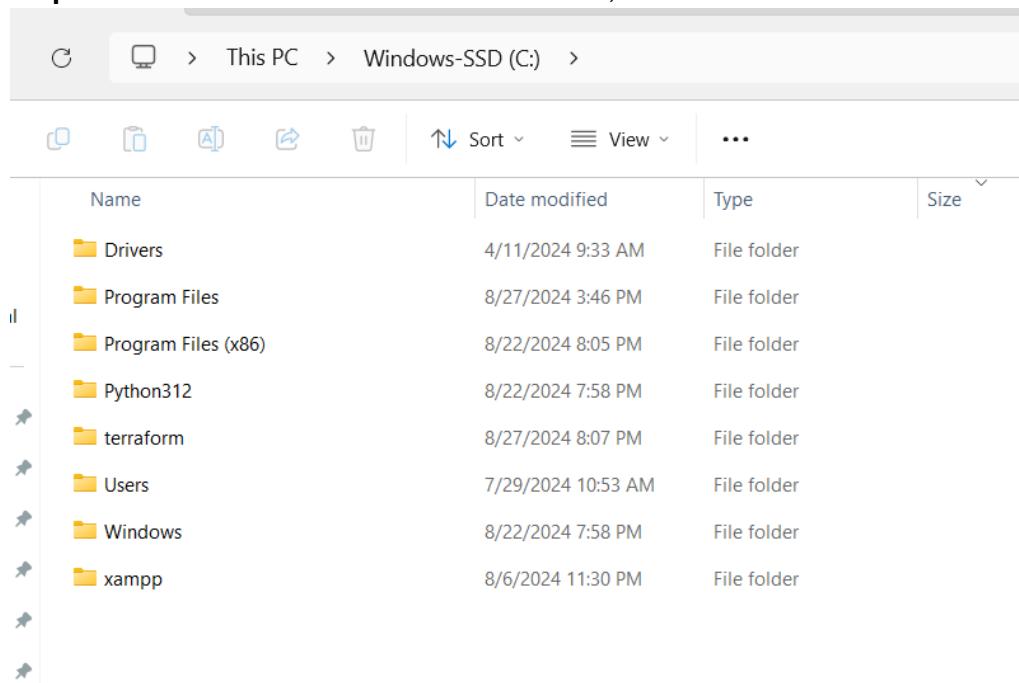
Step 2: Downlaod Windows Binary download 386

This screenshot shows the same 'Install Terraform' page but with 'Windows' selected in the sidebar. The main content area now shows a 'Binary download' section for Windows with two options: '386 Version: 1.9.4' with a 'Download' button and 'AMD64 Version: 1.9.4' with a 'Download' button.

Step 3: Extract the downloaded setup file Terraform.exe in C:\Terraform directory

A file extraction dialog box is shown. At the top, it says 'Extract Compressed (Zipped) Folders'. Below that, it asks 'Select a Destination and Extract Files'. A text input field shows the path 'C:\Users\ADITYA DUBEY\Downloads\terraform_1.9.5_windows_386'. To the right of the input field is a 'Browse...' button. Below the input field is a checked checkbox 'Show extracted files when complete'. At the bottom of the dialog are 'Extract' and 'Cancel' buttons.

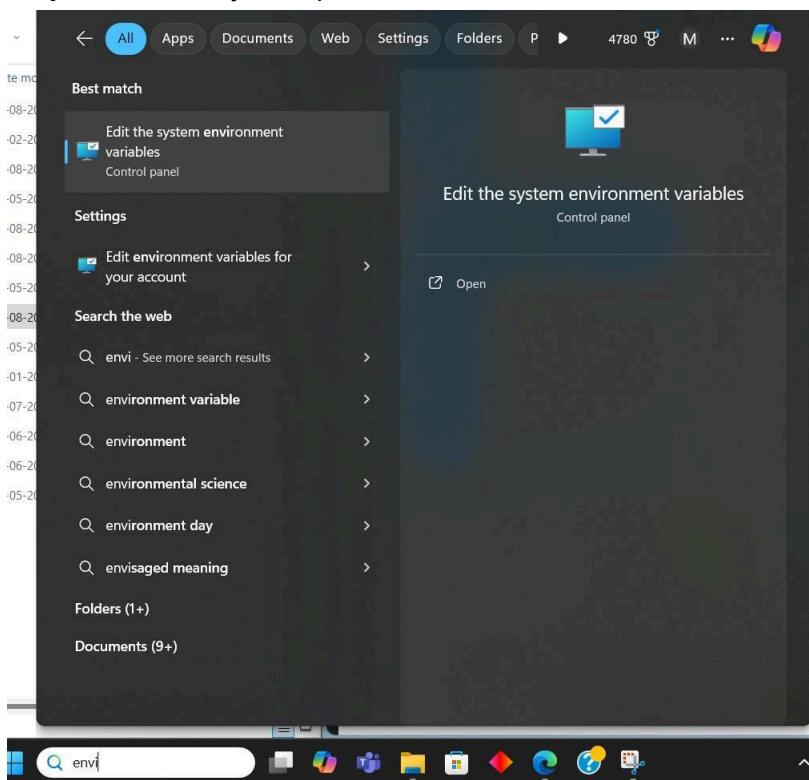
Step 4: Created folder in C: drive for terraform,

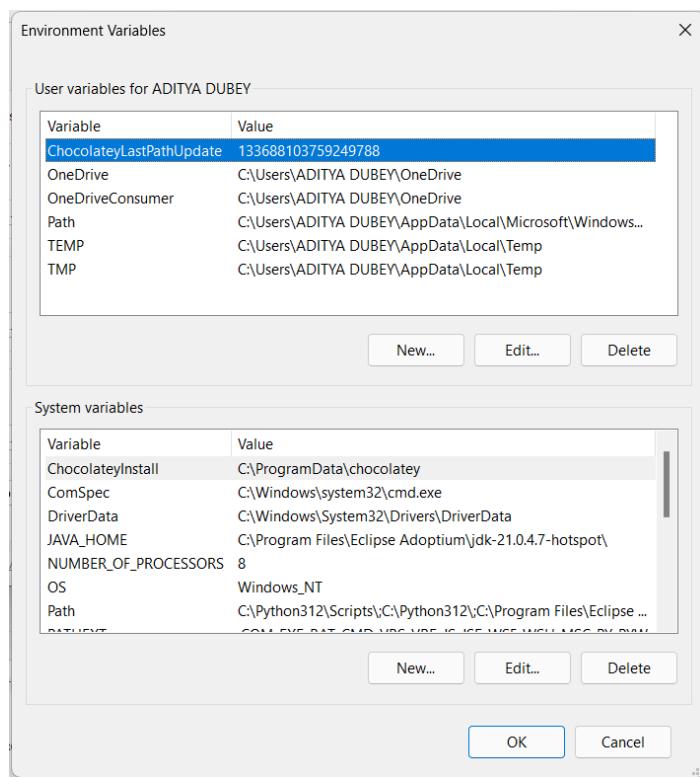
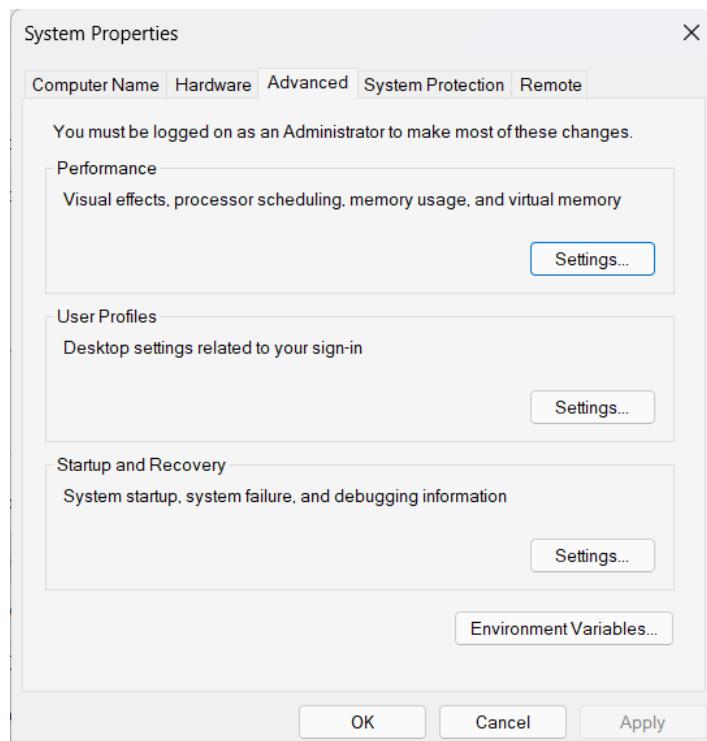


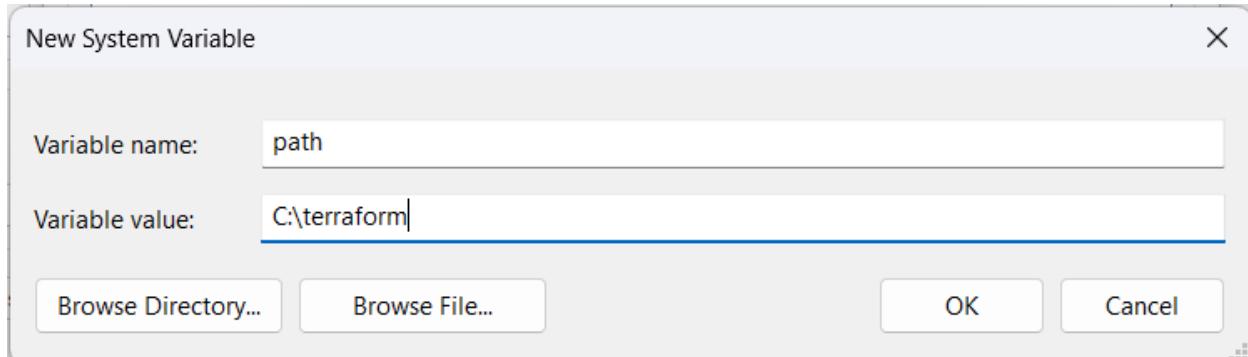
A screenshot of the Windows File Explorer interface. The address bar shows the path: C:\ > This PC > Windows-SSD (C:) >. Below the address bar is a toolbar with icons for back, forward, search, and file operations. To the right of the toolbar are buttons for 'Sort' and 'View'. A vertical sidebar on the left shows a tree view of the drive structure. The main area displays a table with columns: Name, Date modified, Type, and Size (with a dropdown arrow). The table lists several folders: Drivers, Program Files, Program Files (x86), Python312, terraform, Users, Windows, and xampp. All listed folders are of type 'File folder'.

| Name | Date modified | Type | Size |
|---------------------|--------------------|-------------|------|
| Drivers | 4/11/2024 9:33 AM | File folder | |
| Program Files | 8/27/2024 3:46 PM | File folder | |
| Program Files (x86) | 8/22/2024 8:05 PM | File folder | |
| Python312 | 8/22/2024 7:58 PM | File folder | |
| terraform | 8/27/2024 8:07 PM | File folder | |
| Users | 7/29/2024 10:53 AM | File folder | |
| Windows | 8/22/2024 7:58 PM | File folder | |
| xampp | 8/6/2024 11:30 PM | File folder | |

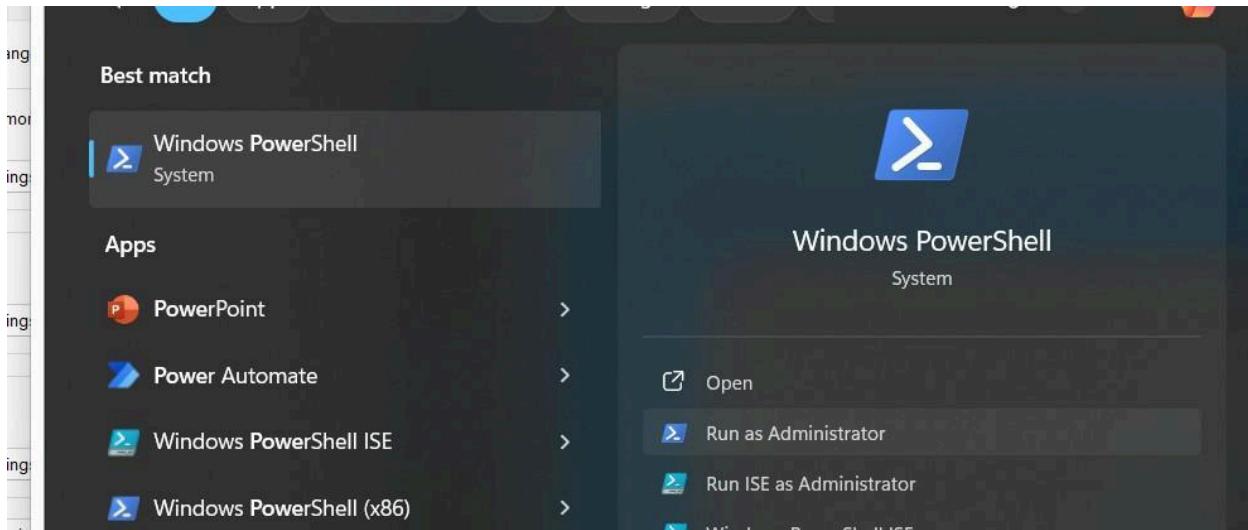
Step 5: Set the System path for Terraform in Environment Variable. Open Environment Variable.







Step 9: Open PowerShell with Admin Access



Step 10: Open Terraform in PowerShell and check its functionality

```

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.

```

A. Creating docker image using terraform

Prerequisite:

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

Step 1: Check the docker functionality

```
C:\Users\ADITYA DUBEY>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
```

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

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: Firstly create a new folder named 'Docker' in the 'TerraformScripts' folder. Then create a new docker.tf file using Atom editor and write the following contents into it to create a Ubuntu Linux container.

Script:

```
#docker.tf
```

```
#docker.tf
```

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

```

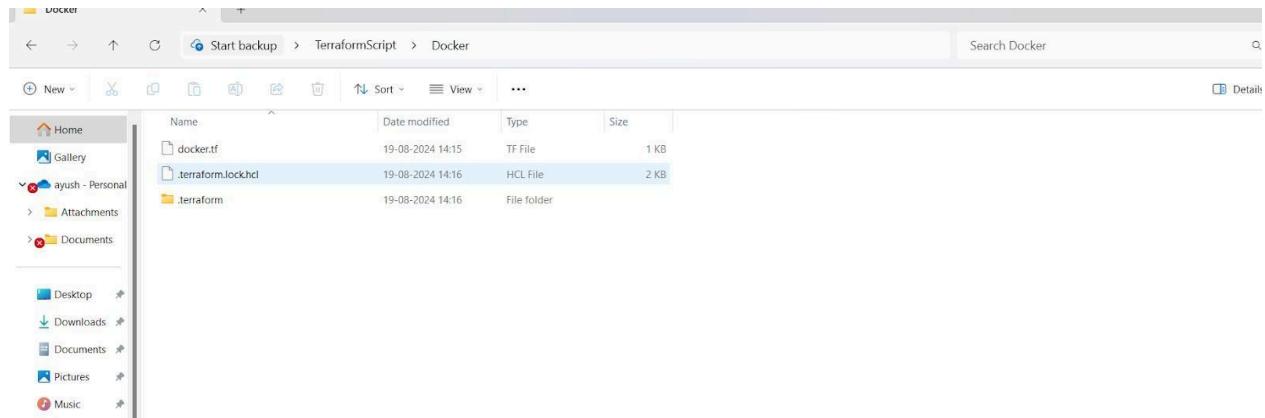
}
}

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

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

# Optional: Run a container from the image
resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id # Correct attribute
  name = "foo"
  command = ["sleep","infinity"]
}

```



```
Docker > docker.tf
1 #docker.tf
2
3 terraform {
4     required_providers {
5         docker = {
6             source  = "kreuzwerker/docker"
7             version = "2.21.0"
8         }
9     }
10 }
11
12 provider "docker" {
13     host = "npipe:///./pipe/docker_engine" # For Windows
14 }
15
16 # Define the Docker image
17 resource "docker_image" "ubuntu" {
18     name = "ubuntu:latest"
19 }
20
21 # Optional: Run a container from the image
22 resource "docker_container" "foo" [
23     image = docker_image.ubuntu.image_id # Correct attribute
24     name  = "foo"
25     command = ["sleep","infinity"]
26 ]
```

Step 3: Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>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          = (known after apply)
  + container_logs   = (known after apply)
  + entrypoint        = (known after apply)
  + env               = (known after apply)
  + exit_code         = (known after apply)
  + gateway           = (known after apply)
  + hostname          = (known after apply)
  + id                = (known after apply)
  + image              = (known after apply)
  + init               = (known after apply)
  + ip_address         = (known after apply)
  + ip_prefix_length  = (known after apply)
  + ipc_mode           = (known after apply)
  + log_driver         = (known after apply)
  + logs               = false
  + must_run           = true
  + name               = "foo"
  + network_data       = (known after apply)
  + read_only           = false
  + remove_volumes     = true
  + restart             = "no"
  + rm                 = false
  + runtime             = (known after apply)
  + security_opts       = (known after apply)
  + shm_size            = (known after apply)
  + start               = true
  + stdin_open          = false
  + stop_signal          = (known after apply)
  + stop_timeout         = (known after apply)
  + tty                 = false
  + healthcheck (known after apply)
}
```

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

```
C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>terraform plan
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

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

```
+ stdin_open      = false
+ stop_signal     = (known after apply)
+ stop_timeout    = (known after apply)
+ tty             = false
+ healthcheck (known after apply)
+ labels (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

docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=25e618f8f29715a205c36e0d31bfdd4a326dc83f513f2c7779aebd492ce9f602]

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

Docker images, After Executing Apply step:

```
C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   edbfe74c41f8  3 weeks ago  78.1MB
```

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

```
C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=e9d84fdf0bbada9cfa63df05a6ee74111495ea3805eb4afaeef53501a0f0a9e56]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach           = false -> null
    - command          = [
        - "sleep",
        - "infinity",
    ] -> null
    - cpu_shares       = 0 -> null
    - dns              = [] -> null
    - dns_opts         = [] -> null
    - dns_search       = [] -> null
    - entrypoint       = [] -> null
    - env              = [] -> null
    - gateway          = "172.17.0.1" -> null
    - group_add        = [] -> null
    - hostname         = "e9d84fdf0bbada9cfa63df05a6ee74111495ea3805eb4afaeef53501a0f0a9e56" -> null
    - id               = "e9d84fdf0bbada9cfa63df05a6ee74111495ea3805eb4afaeef53501a0f0a9e56" -> null
    - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
    - init              = false -> null
    - ip_address        = "172.17.0.2" -> null
    - ip_prefix_length = 16 -> null
    - ipc_mode          = "private" -> null
    - links             = [] -> null
    - log_driver        = "json-file" -> null
    - log_opts           = {} -> null
    - logs              = false -> null
    - max_retry_count   = 0 -> null
    - memory             = 0 -> null
    - memory_swap        = 0 -> null
    - must_run           = true -> null
    - name              = "foo" -> null
    - network_data       = [
        {
            - gateway          = "172.17.0.1"
            - global_ipv6_prefix_length = 0
            - ip_address        = "172.17.0.2"
            - ip_prefix_length  = 16
            - network_name      = "bridge"
            # (2 unchanged attributes hidden)
        },
    ] -> null
    - network_mode       = "bridge" -> null
    - privileged          = false -> null
    - publish_all_ports  = false -> null
    - read_only           = false -> null
}

# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
    - id               = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
    - image_id         = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - latest           = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - name              = "ubuntu:latest" -> null
    - repo_digest      = "ubuntu@sha256:8a37d68f4f73ebf3d4efabcf66379bf3728902a8038616808f04e34a9ab63ee" -> 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=601782417e17994f75628bb648b3e4cbbb989037e3a3d302303c4a675a39599b]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\ Docker>
```

Docker images After Executing Destroy step

```
C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE

C:\Users\ADITYA DUBEY\OneDrive\Desktop\terraformscripts\Docker>docker images
```

EXPERIMENT NO. 7

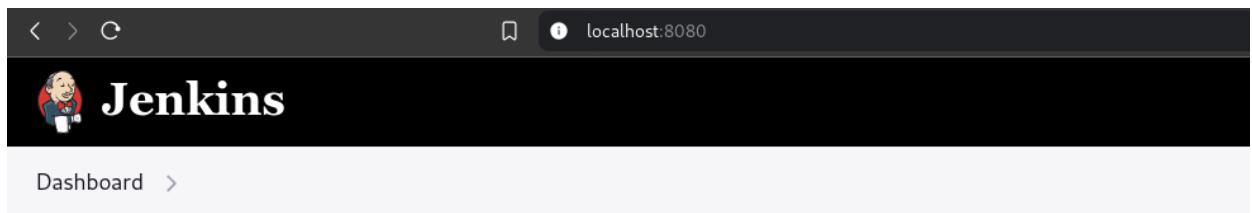
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Prerequisites:

- Jenkins installed (Java JDK required)
- Docker Installed (for SonarQube)

Steps to integrate Jenkins with SonarQube

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



Our jenkins is running on port 8080

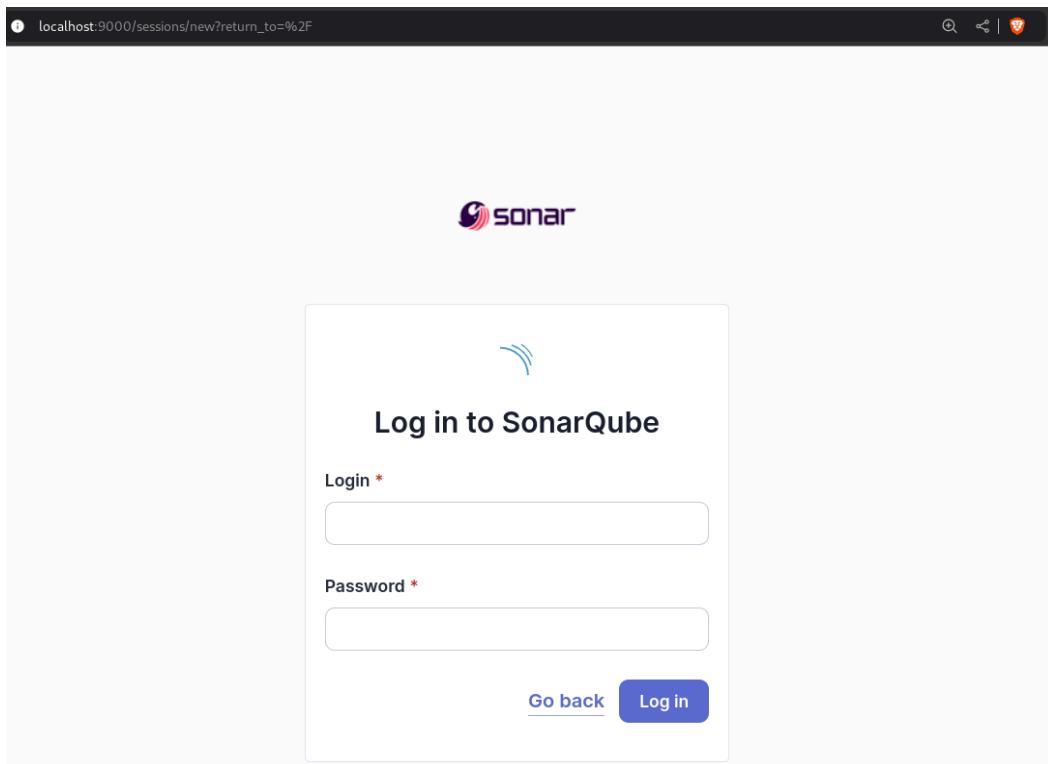
2. Run SonarQube in a Docker container using this command

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

Warning: run below command only once

```
quantum@machine ~ $ sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
[sudo] password for quantum:  
f9c595308e368210e19e099256a47ec1fe44affdc778eb58ccb53174163ce057
```

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



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

localhost:9000/account/reset_password

Update your password

⚠ This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

.....

New Password *

.....

Confirm Password *

.....|

Update

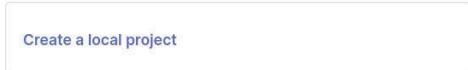
After logging, we have to change default password

5. Create a manual project in SonarQube with the name sonarqube Click on **create a local project** on dashboard

First, you need to set up a DevOps platform configuration.



Are you just testing or have an advanced use-case? Create a local project.



1 of 2

Create a local project

Project display name *

SONARQUBE_PROJECT



Project key *

SONARQUBE_PROJECT



The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Main branch name *

main

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

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

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

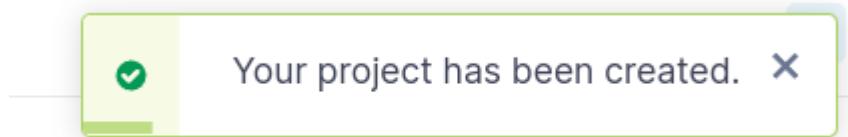
- Use the global setting

Previous version

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

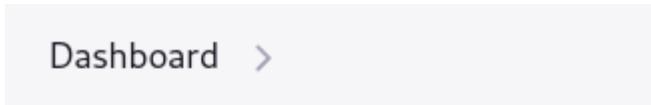
Recommended for projects following regular versions or releases.

We can either use new custom settings for the project or use global settings
Here I'm using global setting



On successful creation of project, we get a popup for the same

6. After setting project in sonarqube, go to **Jenkins Dashboard**



+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

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

System Configuration

| | | |
|--|---|--|
|  System Configure global settings and paths. |  Tools Configure tools, their locations and automatic installers. |  Plugins Add, remove, disable or enable plugins that can extend the functionality of Jenkins. <small>(32)</small> |
|  Nodes Add, remove, control and monitor the various nodes that Jenkins runs jobs on. |  Clouds Add, remove, and configure cloud instances to provision agents on-demand. |  Appearance Configure the look and feel of Jenkins |

The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text 'sonarqube'. Below the search bar, there are two tabs: 'Install' and 'Name ↴'. Under the 'Install' tab, a list of plugins is shown:

- SonarQube Scanner 2.17.2** (selected, checked icon):
 - External Site/Tool Integrations
 - Build Reports

This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.
- Sonar Gerrit 388.v9b_f1cb_e42306**:
 - External Site/Tool Integrations

This plugin allows to submit issues from [SonarQube](#) to [Gerrit](#) as comments directly.
- SonarQube Generic Coverage 1.0**:
 - TODO

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

Installing

Loading plugin extensions

Pending

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Our installation is in progress wait for it to download and install packages

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

 Success

Loading plugin extensions

 Success

Plugin installed successfully

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

System Configuration



System

Configure global settings and paths.



Tools

Configure tools, their locations and automatic installers.



Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

| | |
|-----------|---|
| Name | X |
| sonarqube | |

Server URL

Default is http://localhost:9000

| |
|-----------------------|
| http://localhost:9000 |
|-----------------------|

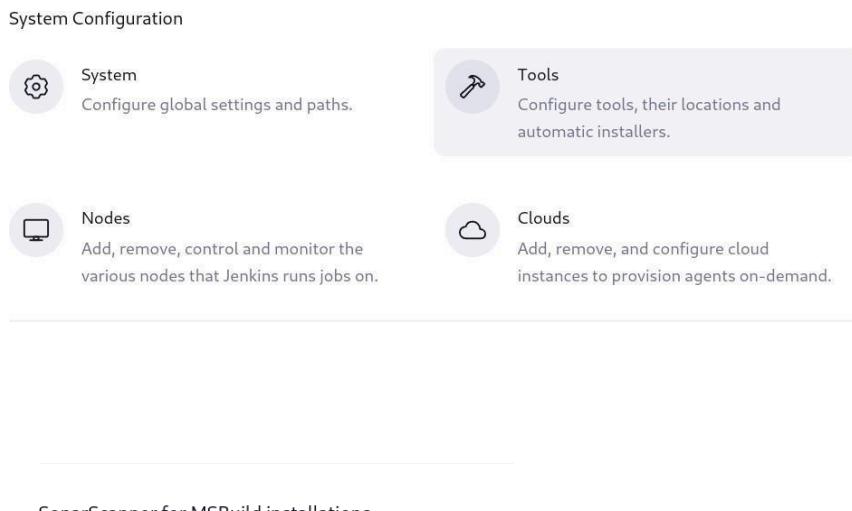
Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

| |
|----------|
| - none - |
| + Add ▾ |

Advanced ▾

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



SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Click on Add SonarQube Scanner

SonarQube Scanner installations

The screenshot shows the 'SonarQube Scanner' configuration page. It includes fields for 'Name' (set to 'sonarqube installer') and an option to 'Install automatically'. A sub-section titled 'Install from Maven Central' shows the selected version 'SonarQube Scanner 6.2.0.4584'. There is also a 'Add Installer' button.

Select Latest version and save configuration

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

New Item

Enter an item name

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.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



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.

OK

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 the integration

Source Code Management

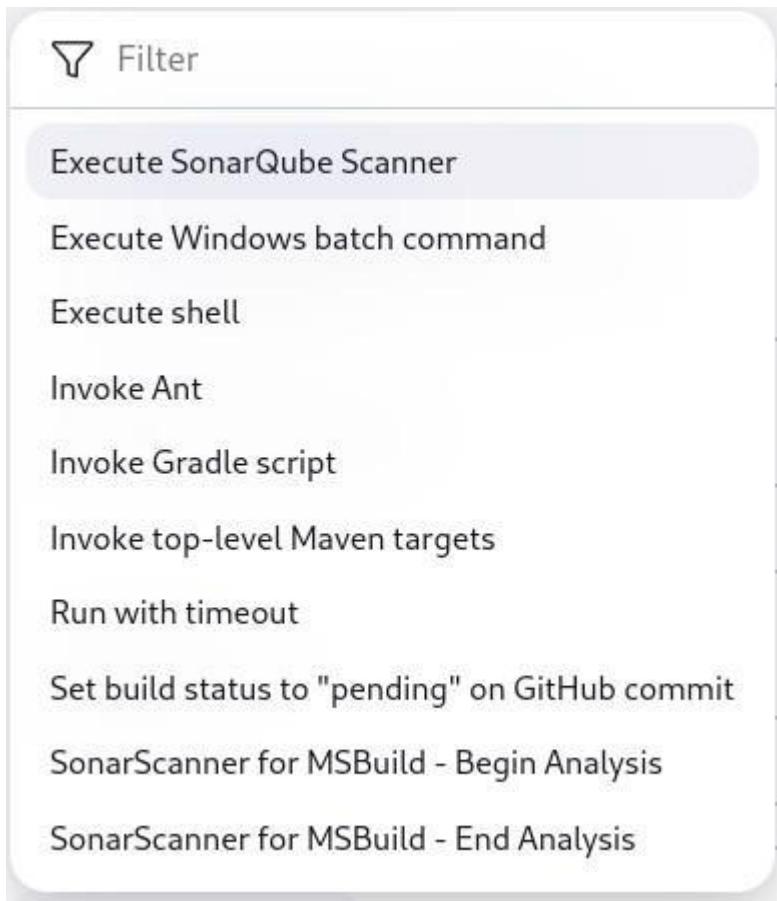


11. Under Build-> Execute SonarQube Scanner

Build Steps

Add build step ▾

Click on add build steps



Then click on **Execute SonarQube Scanner**

12. Mention the SonarQube Project Key, Login, Password, Source path and Host URL in Analysis properties

The image shows the configuration dialog for the "Execute SonarQube Scanner" step. The dialog has the following sections:

- JDK**: A dropdown menu currently set to "(Inherit From Job)".
- Path to project properties**: An empty input field.
- Analysis properties**: A text area containing the following SonarQube configuration:

```
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=admin1
sonar.host.url=http://localhost:9000
sonar.sources=.
```
- Additional arguments**: An empty input field.
- JVM Options**: An empty input field.

13. Go to http://localhost:9000/project_roles?id=<project_key> and allow Execute Permissions to the Admin user.

The screenshot shows a Jenkins interface for managing project roles. At the top, there are tabs for 'All', 'Users', and 'Groups', followed by a search bar. Below the search bar, there are four roles listed: 'Administrator Issues', 'Administrator Security Hotspots', 'Administrator', and 'Execute Analysis'. Under each role, there is a table with columns for 'Name', 'Role', and several checkboxes. The 'Execute Analysis' checkbox is checked for the 'Administrator' role. At the bottom of the page, it says '1 of 1 shown'.

14. Run The Build.

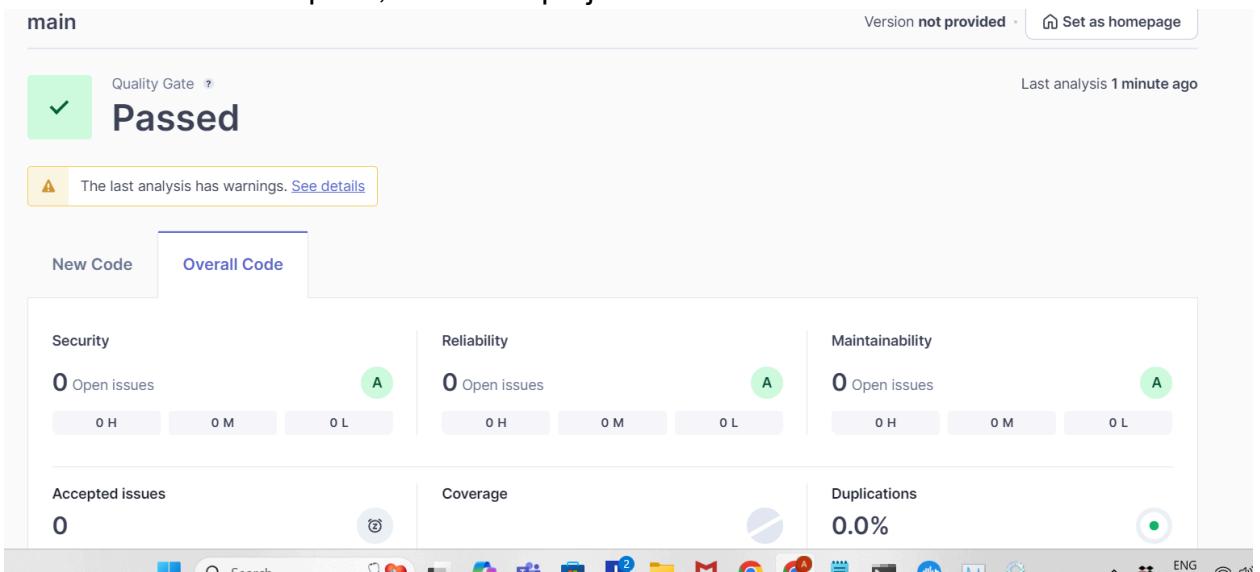
The screenshot shows a Jenkins project configuration page. On the left, there is a sidebar with icons for 'Status', 'Changes', 'Workspace', 'Build Now' (which is highlighted), 'Configure', 'Delete Project', 'SonarQube', and 'Rename'. The main area of the page is currently empty, showing a large white space.

15. Check the console output

```
Started by user Aditya Dubey
Running as SYSTEM
Building on the built-in node in workspace C:\Users\ADITYA DUBEY\.jenkins\workspace\Sonarqube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\Users\ADITYA DUBEY\.jenkins\workspace\Sonarqube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # tim
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[Sunarqube] $ "C:\Users\ADITYA DUBEY\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\insaller\bin\sonar-scanner.bat" -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=SONARQUBE_PROJECT -Dsonar.projectName=SONARQUBE_PROJECT -
Dsonar.host.url=http://localhost:9000 -Dsonar.login=admin -Dsonar.sources=. -Dsonar.password=a1b2d4e5i9j10 -Dsonar.projectBaseDir=C:\Users\ADITYA DUBEY\sonar-project
```

```
10:30:32.234 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:30:32.234 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:30:32.235 INFO More about the report processing at http://localhost:9000/api/ce/task?id=96795973-9667-4456-a15c-3311dbc9d067
10:30:32.244 INFO Analysis total time: 9.533 s
10:30:32.245 INFO SonarScanner Engine completed successfully
10:30:32.300 INFO EXECUTION SUCCESS
10:30:32.301 INFO Total time: 14.090s
Finished: SUCCESS
```

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



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

Conclusion: We began the experiment with installation of SonarQube Docker Image followed by setting up a new project in SonarQube. Then we installed the SonarQube scanner plugin and then created a new freestyle project in Jenkins with a Git repository for code analysis. Then we configured the jenkins with appropriate settings to work with sonarqube. Gave permissions to Jenkins to perform code analysis. It is essential to provide correct properties in **Analysis Properties** for Jenkins to run correctly. The jenkins project ran successfully with all tests passed in SonarQube

EXPERIMENT NO. 8

Aim: Create a Jenkins CI/CD 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.

Integrating Jenkins with SonarQube:

Prerequisites:

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

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

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



+ New Item

Build History

Project Relationship

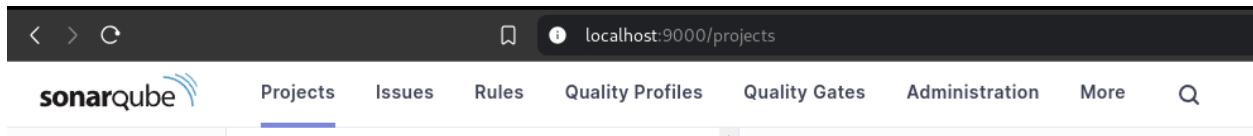
Check File Fingerprint

Manage Jenkins

2. Run SonarQube in a Docker container using this command

```
quantum@machine ~ $ sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

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



- Login to SonarQube using your username and password
- Create a manual project in SonarQube with the name sonarqube-test2

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](#)

database should be used for evaluation purposes only
This database will not scale, it will not support upgrading to newer versions of SonarQube.

Setup the project and come back to Jenkins Dashboard.

- Create a New Item in Jenkins, choose **Pipeline**.

New Item

Enter an item name

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.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



Pipeline

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



Multi-configuration project

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



Folder

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



Multibranch Pipeline

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



Organization Folder

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

7. Under Pipeline Script, enter the following -

```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            sh "<PATH_TO SONARQUBE FOLDER>/bin//sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=<Project_KEY> \
-D sonar.exclusions=vendor/**,resources/**,*/*.java \
-D sonar.host.url=http://127.0.0.1:9000/"
```

```
}
```

```
}
```

```
}
```

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

8. Install sonar-scanner

Now we need to install **sonar-scanner** binary to perform code analysis

To do so, go to

<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0-4584-linux-x64.zip> for linux OS

(<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0-4584-windows-x64.zip> for windows OS)

You will be prompted to download the zip file, download it and extract it and copy the path (absolute path) to <ROOT_DIRECTORY>/bin/sonar-scanner

For windows path would be

C:\\\\Users\\\\<USER_NAME>\\\\Downloads\\\\sonar-scanner-cli-6.2.0.4584-windows-x64\\\\sonar-scanner-cl... onar-scanner in my case its

“C:\\\\Users\\\\ADITYA DUBEY\\\\Downloads\\\\sonar-scanner-cl... 6.2.0.4584-windows-x64\\\\sonar-scanner-cl... //” for paths with space

Now we need to update the required details in the pipeline script as :

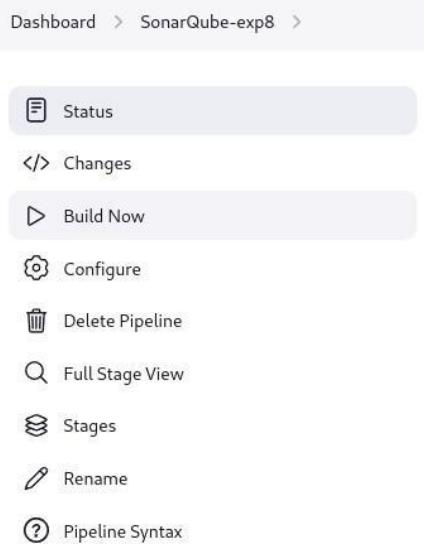
```
pipeline {
    agent any
    stages {
        stage('Cloning the GitHub Repo') {
            steps {
                git 'https://github.com/shazforiot/GOL.git'
            }
        }
        stage('SonarQube analysis') {
            steps {
                withSonarQubeEnv('sonarqube') {configuration
                    sh """
                    "C:\\\\Users\\\\ADITYA DUBEY\\\\Downloads\\\\sonar-scanner-cl... 6.2.0.4584-windows-x64\\\\sonar-scanner-cl... 6.2.0.4584-windows-x64\\\\bin\\\\sonar-scanner" ^
                    -D sonar.projectKey=sonarqube-test2 ^
                    -D sonar.sources= ^
                    -D sonar.exclusions=vendor/**,resources/**,*/*.java ^
                    """
                }
            }
        }
    }
}
```

```
-D sonar.host.url=http://127.0.0.1:9000 ^
-D sonar.login=admin ^
-D sonar.password=my pass
"""
}
}
}
}
```

Script ?

```
1 * pipeline {
2   agent any
3   stages {
4     stage('Cloning the GitHub Repo') {
5       steps {
6         git 'https://github.com/shazforiot/GOL.git'
7       }
8     }
9     stage('SonarQube Analysis') {
10    steps {
11      withSonarQubeEnv('sonarqube') {
12        bat """
13          C:/Users/ADITYA DUBEY/Downloads/sonar-scanner-cli-6.2.0.4584-windows-x64/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar
14          -D sonar.projectKey=SONARQUBE_PROJECT2 ^
15          -D sonar.sources=.^
16          -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
17          -D sonar.host.url=http://127.0.0.1:9000 ^
18          -D sonar.login=admin ^
19          -D sonar.password=a1b2d4e5f9j10
20        """
21      }
22    }
23  }
24 }
25 }
```

9. Run the build



Check the status of Build

SonarQube-exp8

SonarQube analysis for EXP 8

Stage View



As we can see the SonarQube analysis is completed

10. Check the console output once the build is complete.

Console Output

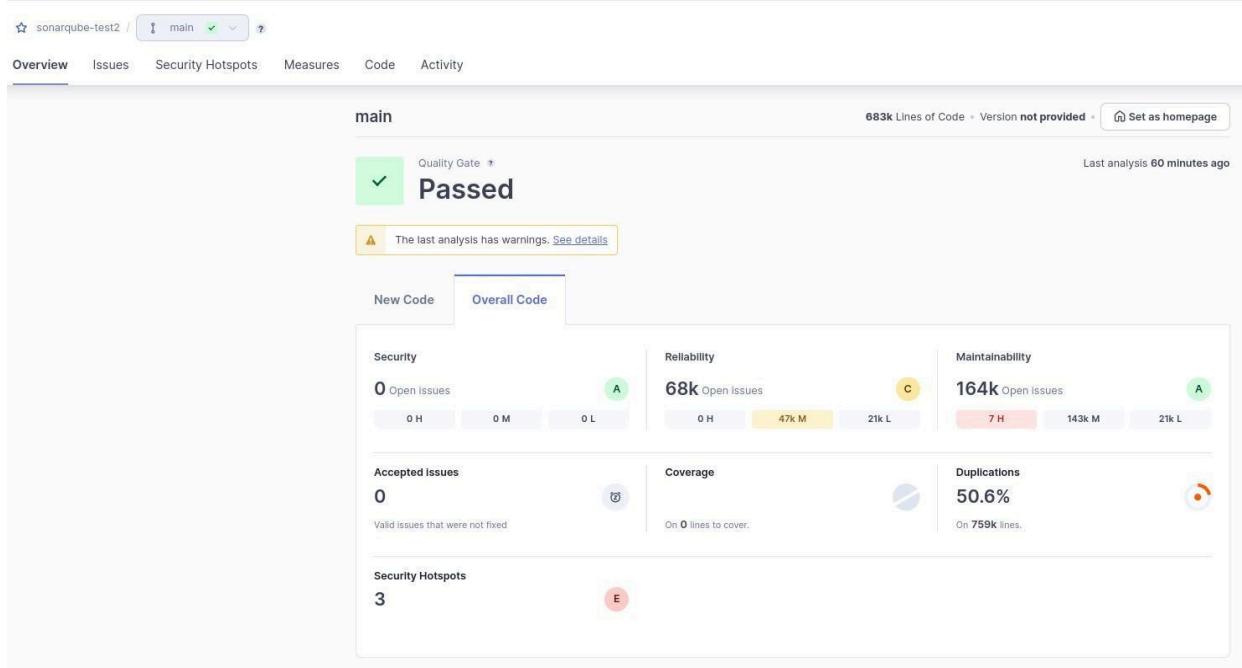
```
Skipping 4,226 KB.. Full Log
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 296. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 17. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 298. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 300. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 295. Keep only the first 100 references.
```

```

20:11:31.504 INFO CPD Executor CPD calculation finished (done) | time=104775ms
20:11:31.748 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:11:33.306 INFO Analysis report generated in 1507ms, dir size=127.2 MB
20:11:40.819 INFO Analysis report compressed in 7511ms, zip size=29.6 MB
20:11:42.147 INFO Analysis report uploaded in 1322ms
20:11:42.154 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test2
20:11:42.154 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:11:42.154 INFO More about the report processing at http://127.0.1:9000/api/ce/task?id=670aec3c-6c20-460b-ad52-ec6ce041fb1f
20:11:46.802 INFO Analysis total time: 3:42.216 s
20:11:46.835 INFO SonarScanner Engine completed successfully
20:11:48.173 INFO EXECUTION SUCCESS
20:11:48.296 INFO Total time: 3:47.113s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

11. After that, check the project in SonarQube.



Under different tabs, check all different issues with the code

12. Code Problems-

Issues / main

Filters

- Clean Code Attribute**
 - Consistency: 197k
 - Intentionality: 14k
 - Adaptability: 0
 - Responsibility: 0
- Software Quality**
 - Security: 0
 - Reliability: 54k
 - Maintainability: 164k
- Severity**
 - High: 0
 - Medium: 176k
 - Low: 21k

Bulk Change

Select issues | Navigate to issue | 196,662 issues | 3075d effort

gameoflife-core/build/reports/tests/all-tests.html

- Insert a <!DOCTYPE> declaration to before this <html> tag. **Consistency** **user-experience** **Reliability** **Open** Not assigned L1 × 5min effort × 4 years ago × Bug × Major
- Remove this deprecated "width" attribute. **Maintainability** **html5 obsolete** **Open** Not assigned L9 × 5min effort × 4 years ago × Code Smell × Major
- Remove this deprecated "align" attribute. **Maintainability** **html5 obsolete** **Open** Not assigned L11 × 5min effort × 4 years ago × Code Smell × Major
- Remove this deprecated "align" attribute. **Maintainability** **html5 obsolete** **Open** Not assigned L12 × 5min effort × 4 years ago × Code Smell × Major
- Remove this deprecated "size" attribute. **Consistency** **Open** Not assigned L13 × 5min effort × 4 years ago × Code Smell × Major

Code Smells

Issues / main

Filters

- Responsibility**: 0
- Software Quality**
 - Security: 0
 - Reliability: 21k
 - Maintainability: 164k
- Severity**
 - High: 7
 - Medium: 143k
 - Low: 21k
- Type**
 - Bug: 47k
 - Vulnerability: 0
 - Code Smell: 164k**
- Scope**
 - Main code: 164k
 - Test code: 0
- Status**

Bulk Change

Select issues | Navigate to issue | 164,034 issues | 1708d effort

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. **Maintainability** **Intentionality** **No tags** **Open** Not assigned L1 × 5min effort × 4 years ago × Code Smell × Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Maintainability** **Intentionality** **No tags** **Open** Not assigned L12 × 5min effort × 4 years ago × Code Smell × Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Maintainability** **Intentionality** **No tags** **Open** Not assigned L12 × 5min effort × 4 years ago × Code Smell × Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Maintainability** **Intentionality** **No tags** **Open** Not assigned L13 × 5min effort × 4 years ago × Code Smell × Major

gameoflife-core/build/reports/tests/all-tests.html

Bugs

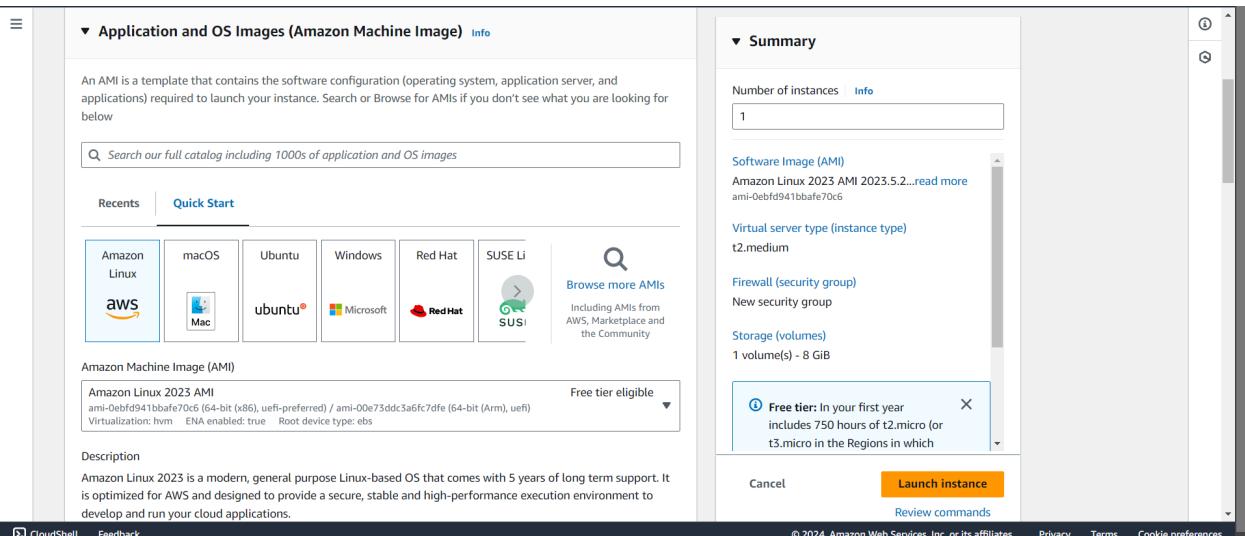
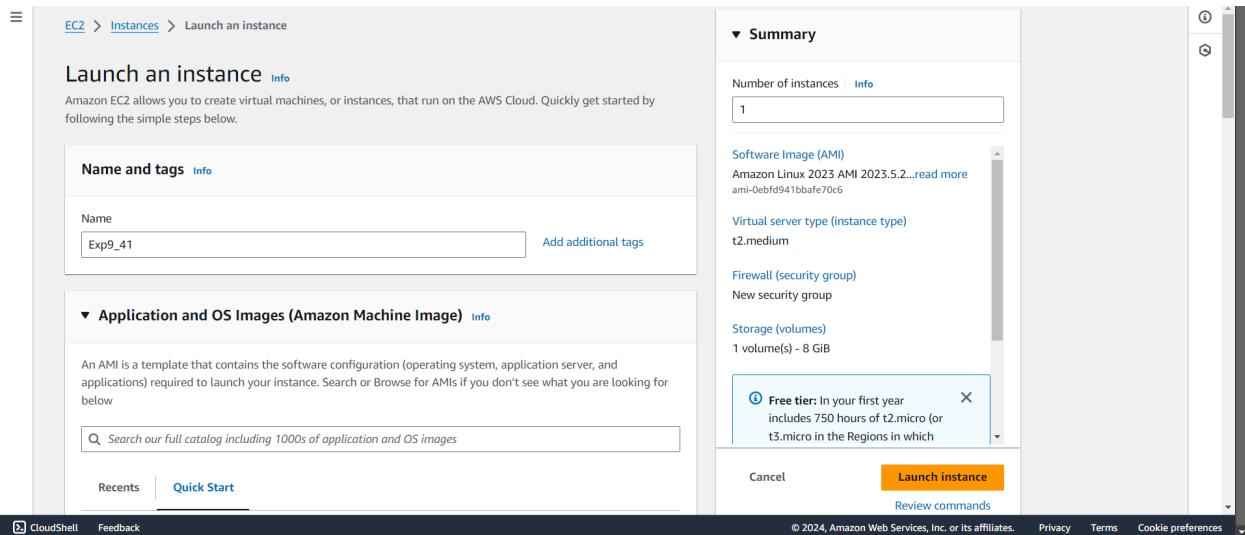
The screenshot shows the SonarQube 'Issues' page for the 'main' branch of the project 'sonarqube-test2'. The left sidebar displays various quality metrics and filters. The main area lists several issues, each with a title, severity (Reliability), status (Open), assignee (Not assigned), and creation details (L1 - 5min effort - 4 years ago). The issues are categorized by file path: 'gameoflife-core/build/reports/tests/all-tests.html', 'gameoflife-core/build/reports/tests/allclasses-frame.html', 'gameoflife-core/build/reports/tests/alltests-errors.html', and 'gameoflife-core/build/reports/tests/alltests-fails.html'. Each issue has a 'Bulk Change' button and a 'Select issues' dropdown.

Conclusion: We began the experiment with creating a new project in SonarQube and setting up a new Pipeline in Jenkins with proper configuration of pipeline script. Then we installed Sonar Scanner CLI so that jenkins can do code analysis of Git Repository. We can also configure the pipeline to use the installed Sonar Scanner plugin instead of locally installed Binary of Sonar Scanner. The pipeline ran successfully with all tests passed in SonarQube

Experiment No: 9

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

Step 1: Sign in to your AWS account. Look for EC2 in the services list. Open it and click on "Create Instance."



Select The OS Image as Amazon Linux.

Step 2: If you haven't created a private key or a .pem file yet, click on "Create a key pair." Otherwise, choose the key pair you created earlier. (Be sure to remember where the .pem file for that key is located on your system.) In my case I have created a new one.

The screenshot shows the AWS Launch Wizard interface for creating a new Amazon Linux 2023.5.2 AMI instance. The configuration steps shown are:

- Instance type:** t2.medium
- Key pair (login):** Lab9_41
- Network settings:** A dialog box is open for creating a new security group named 'launch-wizard-27'. It includes rules for SSH (Allow SSH traffic from Anywhere) and HTTPS (Allow HTTPS traffic from the internet). A warning message states: "Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."
- Summary:** Shows 1 instance, AMI (Amazon Linux 2023.5.2), Virtual server type (t2.medium), Firewall (New security group), and Storage (1 volume(s) - 8 GiB).
- Free tier:** A callout box indicates a free tier offer for the first year.
- Buttons:** Cancel, Launch instance, and Review commands.

AWS will create a security group for this instance. Keep the name of that instance saved.

Instance:

The screenshot shows the AWS CloudWatch Metrics interface for the instance 'Exp9_41' with the ARN 'i-03b653a01796c6519'. The status is listed as 'Running' with a green checkmark icon.

Step3: After you create the instance, click on "Security Groups" in the left sidebar. Look for the security group that corresponds to your instance, and then click on the security group ID for that group. (in my case launch-wizard-27 is the latest one.)

| Name | Security group ID | Security group name | VPC ID | Description |
|------|----------------------|---------------------|-----------------------|-------------|
| - | sg-0608ac064fada5668 | launch-wizard-27 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-0a21b9ee664c07c9b | launch-wizard-22 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-0a9c5a81f27ec0948 | launch-wizard-21 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-03d68f95bf6fb69d | launch-wizard-18 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-0b28ae9ba37abcf1 | launch-wizard-7 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-0db7e51255007eb3b | launch-wizard-4 | vpc-04fadfb026daa5d28 | launch-wiz |
| - | sg-0cc56e7d786d5b527 | Node | vpc-04fadfb026daa5d28 | Security gr |
| - | sg-0c352fb3519dde32 | launch-wizard-8 | vpc-04fadfb026daa5d28 | launch-wiz |

Click on Id

| Security group name | Security group ID | Description | VPC ID |
|---------------------|----------------------|---|-----------------------|
| launch-wizard-27 | sg-0608ac064fada5668 | launch-wizard-27 created 2024-09-28T07:46:12.556Z | vpc-04fadfb026daa5d28 |

Inbound rules (1)

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|------|------------------------|------------|------|----------|------------|
|------|------------------------|------------|------|----------|------------|

Click on the edit inbound rules

Name: Aditya Dubey

Div: D15C

Roll No: 10

EC2 > Security Groups > sg-0608ac064fada5668 - launch-wizard-27 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules [Info](#)

| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | Description - optional Info |
|------------------------|---------------------------|-------------------------------|---------------------------------|-----------------------------|---|
| sgr-0b3ded36f05264611 | SSH | TCP | 22 | Custom | 0.0.0.0/0 X |

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [X](#)

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

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

Next, click on "Add rules" and set up rules for the following protocols: HTTP, All ICMP (IPv6), HTTPS, All traffic, Custom TCP (Port 5666), and All ICMP (IPv4).

Inbound rules [Info](#)

| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | Description - optional Info |
|------------------------|---------------------------|-------------------------------|---------------------------------|-------------------------------|---|
| sgr-0b3ded36f05264611 | SSH | TCP | 22 | Custom | 0.0.0.0/0 X |
| - | HTTP | TCP | 80 | Anywhere... ▼ | 0.0.0.0/0 X |
| - | All ICMP - IPv6 | IPv6 ICMP | All | Anywhere... ▼ | 0.0.0.0/0 X |
| - | HTTPS | TCP | 443 | Anywhere... ▼ | 0.0.0.0/0 X |
| - | All traffic | All | All | Anywhere... ▼ | 0.0.0.0/0 X |
| - | Custom TCP | TCP | 5666 | Anywhere... ▼ | 0.0.0.0/0 X |
| - | All ICMP - IPv4 | ICMP | All | Anywhere... ▼ | 0.0.0.0/0 X |

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

Click on save. This will add all the inbound rules to the security group.

The screenshot shows the AWS EC2 Security Groups page. A success message at the top indicates that security group rules were successfully modified. The main section displays the details of the security group 'sg-0608ac064fada5668 - launch-wizard-27'. It shows the security group name, owner, security group ID, description, and VPC ID. Below this, the 'Inbound rules' tab is selected, showing a table with 7 entries. The table columns include Name, Security group rule..., IP version, Type, Protocol, Port range, and Source. The rules include various protocols like TCP, HTTP, ICMP, and SSH, with source ranges like 0.0.0.0/0 and specific IP addresses.

Step 4: Return to the instances screen and click on the instance ID of your instance. Then, click on "Connect."

The screenshot shows the AWS EC2 Instances page. A table lists 1 instance, with 7 more listed below it. The columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IP/DNS, and Price. One instance, 'Exp9_41', is selected and highlighted in blue. Its details are shown in a modal at the bottom: Name: Exp9_41, Instance ID: i-03b653a01796c6519, State: Running, Type: t2.medium, Status check: 2/2 checks passed, Alarm status: View alarms +, Availability Zone: us-east-1b, Public IP/DNS: ec2-52-91-78-149.com..., and Price: 52. The modal also shows the instance's ARN: i-03b653a01796c6519 (Exp9_41).

Click on "SSH client" and copy the example command provided.

The screenshot shows the 'Connect to instance' page in the AWS Management Console. The instance ID is i-03b653a01796c6519 (Exp9_41). The 'SSH client' tab is selected. The page provides instructions for connecting via SSH:

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Lab9_41.pem.
- Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "Lab9_41.pem"
- Connect to your instance using its Public DNS:
ec2-52-91-78-149.compute-1.amazonaws.com

Example command:

```
ssh -i "Lab9_41.pem" ec2-user@ec2-52-91-78-149.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

Step 5: Now, we need to connect our local terminal to the instance using SSH. Open the terminal where your private key file (.pem) is located by actually going to the folder which has the .pem file, paste the copied SSH command, and run it.

```
ec2-user@ip-172-31-84-149 ~ % 
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\praja\Desktop\Newfolder> ssh -i "Lab9_41.pem" ec2-user@ec2-52-91-78-149.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-78-149.compute-1.amazonaws.com (52.91.78.149)' can't be established.
ED25519 key fingerprint is SHA256:Yho9mkRy0vawd7JNbpHywWVnFJ7QIUtL0rohVayc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-78-149.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     /###_      Amazon Linux 2023
    /#####\
   /##|
  /#/
 V~' '--> https://aws.amazon.com/linux/amazon-linux-2023
  ~~' /
  ~~'-' /'
  ~~'-' /'
  ~~'-' /'
 [ec2-user@ip-172-31-84-149 ~]$ |
```

Step 6: Now we start working on this terminal. First run the command **sudo yum update**. This command will check for any updates for the YUM library to ensure that all libraries are up to date with the latest features and security fixes

```
[ec2-user@ip-172-31-84-149 ~]$ sudo yum update
Last metadata expiration check: 0:30:14 ago on Sat Sep 28 07:51:31 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 7: We are going to set up a web server software application called **Apache** and a programming language called **PHP** in this instance . To do this, run this command in your terminal

sudo yum install httpd php

```
[ec2-user@ip-172-31-84-149 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:31:34 ago on Sat Sep 28 07:51:31 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    48 k
php8_3           x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    10 k
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2  amazonlinux   129 k
apr-util         x86_64      1.6.3-1.amzn2023.0.1  amazonlinux   98 k
generic-logos-httdp noarch     18.0.0-12.amzn2023.0.3  amazonlinux   19 k
httpd-core       x86_64      2.4.62-1.amzn2023   amazonlinux   1.4 M
httpd-filesystem noarch     2.4.62-1.amzn2023   amazonlinux   14 k
httpd-tools      x86_64      2.4.62-1.amzn2023   amazonlinux   81 k
libbrotli        x86_64      1.0.9-4.amzn2023.0.2  amazonlinux   315 k
libsodium         x86_64      1.0.19-4.amzn2023   amazonlinux   176 k
libssl            x86_64      1.1.34-5.amzn2023.0.2  amazonlinux   241 k
mailcap          noarch     2.1.49-3.amzn2023.0.3  amazonlinux   33 k
nginx-filesystem noarch     1:1.24.0-1.amzn2023.0.4  amazonlinux   9.8 k
php8_3-cli       x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   3.7 M
php8_3-common    x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   737 k
php8_3-process   x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   45 k
php8_3-xml       x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   154 k
Installing weak dependencies:
apr-util-openssl x86_64      1.6.3-1.amzn2023.0.1  amazonlinux   17 k
mod_http2        x86_64      2.0.27-1.amzn2023.0.3  amazonlinux   166 k
mod_lua          x86_64      2.4.62-1.amzn2023   amazonlinux   61 k
=====
Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
libssl-1.1.34-5.amzn2023.0.2.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
nginx-filesystem=1:1.24.0-1.amzn2023.0.4.noarch
php8_3-cli-8.3.10-1.amzn2023.0.1.x86_64
php8_3-fpm-8.3.10-1.amzn2023.0.1.x86_64
php8_3-opcache-8.3.10-1.amzn2023.0.1.x86_64
php8_3-process-8.3.10-1.amzn2023.0.1.x86_64
php8_3-xml-8.3.10-1.amzn2023.0.1.x86_64
=====
april-util-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-tools-2.4.62-1.amzn2023.x86_64
libsodium-1.0.19-4.amzn2023.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_lua-2.4.62-1.amzn2023.x86_64
php8_3-8.3.10-1.amzn2023.0.1.x86_64
php8_3-common-8.3.10-1.amzn2023.0.1.x86_64
php8_3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
php8_3-pdo-8.3.10-1.amzn2023.0.1.x86_64
php8_3-sodium-8.3.10-1.amzn2023.0.1.x86_64
=====
Complete!
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 8: Now, we will **install the GCC compiler**, which is used for compiling and running C and C++ programs, along with the essential C libraries. To do this, enter the following command: **sudo yum install gcc glibc glibc-common**

```
[ec2-user@ip-172-31-84-149 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:34:20 ago on Sat Sep 28 07:51:31 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
      Package           Architecture   Version        Repository    Size
=====
Installing:
  gcc                  x86_64         11.4.1-2.amzn2023.0.2      amazonlinux  32 M
Installing dependencies:
  annobin-docs          noarch        10.93-1.amzn2023.0.1      amazonlinux  92 k
  annobin-plugin-gcc    x86_64         10.93-1.amzn2023.0.1      amazonlinux  887 k
  cpp                  x86_64         11.4.1-2.amzn2023.0.2      amazonlinux  10 M
  gc                   x86_64         8.0.4-5.amzn2023.0.2      amazonlinux  105 k
  glibc-devel           x86_64         2.34-52.amzn2023.0.11     amazonlinux  27 k
  glibc-headers-x86    noarch        2.34-52.amzn2023.0.11     amazonlinux  427 k
  guile22              x86_64         2.2.7-2.amzn2023.0.3      amazonlinux  6.4 M
  kernel-headers        x86_64         6.1.109-118.189.amzn2023 amazonlinux  1.4 M
  libmpc               x86_64         1.2.1-2.amzn2023.0.2      amazonlinux  62 k
  libtool-ltdl          x86_64         2.4.7-1.amzn2023.0.3      amazonlinux  38 k
  libxcrypt-devel       x86_64         4.4.33-7.amzn2023        amazonlinux  32 k
  make                 x86_64         1:4.3-5.amzn2023.0.2      amazonlinux  534 k
=====
Transaction Summary
=====
Install 13 Packages

Total download size: 52 M
```

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

Complete!
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 9: Next, we need to **install the GD library**, along with its development tools. This library helps with creating and manipulating images. For that, run this command **sudo yum install gd gd-devel**

```
[ec2-user@ip-172-31-84-149 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:36:28 ago on Sat Sep 28 07:51:31 2024.
Dependencies resolved.
=====
      Package           Architecture   Version        Repository    Size
=====
Installing:
  gd                  x86_64         2.3.3-5.amzn2023.0.3      amazonlinux  139 k
  gd-devel            x86_64         2.3.3-5.amzn2023.0.3      amazonlinux  38 k
Installing dependencies:
  brotli              x86_64         1.0.9-4.amzn2023.0.2      amazonlinux  314 k
  brotli-devel        x86_64         1.0.9-4.amzn2023.0.2      amazonlinux  31 k
  bzip2-devel          x86_64         1.0.8-6.amzn2023.0.2      amazonlinux  214 k
  cairo               x86_64         1.17.6-2.amzn2023.0.1      amazonlinux  684 k
  cmake-fs             x86_64         3.22.2-1.amzn2023.0.4      amazonlinux  16 k
  fontconfig          x86_64         2.13.94-2.amzn2023.0.2      amazonlinux  273 k
  fontconfig-devel    x86_64         2.13.94-2.amzn2023.0.2      amazonlinux  128 k
  fonts-fs             noarch        1:2.0.5-12.amzn2023.0.2      amazonlinux  9.5 k
  freetype             x86_64         2.13.2-5.amzn2023.0.1      amazonlinux  423 k
  freetype-devel       x86_64         2.13.2-5.amzn2023.0.1      amazonlinux  912 k
```

```
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
libsepol-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
```

```
Complete!
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 10: Now, we create a user called '**nagios**' and make sure that it has a home directory, and set up a password for it.

sudo adduser -m nagios

sudo passwd nagios

```
[ec2-user@ip-172-31-84-149 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-84-149 ~]$ |
```

Password :Shivam2@

Step 11: Now, we need to create a user group named **nagcmd**, which will be used to execute Nagios commands. To do this, run the following command: **sudo groupadd nagcmd**

```
[ec2-user@ip-172-31-84-149 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 12: Next, we'll add the users **apache** and **nagios** to the **nagcmd** group. This allows them to execute Nagios commands.

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-84-149 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 13: We'll create a directory called **downloads** to store the files related to the Nagios server that we download.

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-84-149 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-84-149 downloads]$ |
```

Step 14: Now we need to install the latest versions of nagios-core and nagios-plugins. Go to the respective websites and check whether a better version is available. If newer versions are available, then right click on the download button → Copy link address. Paste this link address in place of the current link in command. If not run these commands.

wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz>

```
[ec2-user@ip-172-31-84-149 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-28 08:37:22-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz      100%[=====]  1.97M  5.07MB/s   in 0.4s
2024-09-28 08:37:23 (5.07 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
[ec2-user@ip-172-31-84-149 downloads]$ |
```

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

```
[ec2-user@ip-172-31-84-149 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-28 08:38:31-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz 100%[=====]  2.62M  7.16MB/s   in 0.4s
2024-09-28 08:38:32 (7.16 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
[ec2-user@ip-172-31-84-149 downloads]$ |
```

Step 15: Now, we need to extract the Nagios Core file into the same directory. We can do this using the tar command.

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

The screenshot shows a terminal window with a dark background and white text. At the top, it says "ec2-user@ip-172-31-84-149:~". Below that, the command "tar zxvf nagios-4.5.5.tar.gz" is run, followed by a list of files being extracted. The list includes: 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/worker/Makefile.in, nagios-4.5.5/worker/ping/, nagios-4.5.5/worker/ping/.gitignore, nagios-4.5.5/worker/ping/Makefile.in, nagios-4.5.5/worker/ping/worker-ping.c, nagios-4.5.5/xdata/, nagios-4.5.5/xdata/.gitignore, nagios-4.5.5/xdata/Makefile.in, nagios-4.5.5/xdata/xcddefault.c, nagios-4.5.5/xdata/xcddefault.h, nagios-4.5.5/xdata/xodtemplate.c, nagios-4.5.5/xdata/xodtemplate.h, nagios-4.5.5/xdata/xpddefault.c, nagios-4.5.5/xdata/xpddefault.h, nagios-4.5.5/xdata/xrddefault.c, nagios-4.5.5/xdata/xrddefault.h, nagios-4.5.5/xdata/xsddefault.c, nagios-4.5.5/xdata/xsddefault.h. The command "[ec2-user@ip-172-31-84-149 ~]\$ |" is at the bottom right.

```
[ec2-user@ip-172-31-84-149 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/worker/Makefile.in
nagios-4.5.5/worker/ping/
nagios-4.5.5/worker/ping/.gitignore
nagios-4.5.5/worker/ping/Makefile.in
nagios-4.5.5/worker/ping/worker-ping.c
nagios-4.5.5/xdata/
nagios-4.5.5/xdata/.gitignore
nagios-4.5.5/xdata/Makefile.in
nagios-4.5.5/xdata/xcddefault.c
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step16: Now, we need to ensure that Nagios uses the nagcmd group for executing external commands.

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

```
nagios-4.5.5/xdata/xSudoRw.c: In function 'nagios_main':
[ec2-user@ip-172-31-84-149 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
[ec2-user@ip-172-31-84-149 downloads]$ |
```

An error was encountered here: `./configure: no such path or directory`. So Navigate to the nagios-4.5.5 folder in downloads. (version could vary)

ls :

```
[ec2-user@ip-172-31-84-149 ~]$ cd nagios-4.5.5
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

- `cd nagios-4.5.5` (use the version shown by your `ls` command)

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

Another error could be Cannot find SSL headers. To solve this, we need to install OpenSSL Dev Library : `sudo yum install openssl-devel`

```
[ec2-user@ip-172-31-83-157 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:21:59 ago on Sat Sep 28 03:46:46 2024.
Dependencies resolved.
=====
== Package                                         Repository          Architecture      Size   Version
=====
== Installing:
openssl-devel                                     amazonlinux        x86_64           3.0 M  1:3.0.8-1.amzn2023.0.14
=====
Transaction Summary
=====
Install 1 Package
Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm      18 MB/s | 3.0 MB  00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

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

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
```

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

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

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

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

[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

Step 17: Next, we need to compile all the components of the software based on the instructions in the Makefile. For that, use this command: **make all** Then, **sudo make install**

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o .
./common/shared.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o quer
y-handler.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ?
) ? slash : cmd_name);
|           ~~~~~~
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.
c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.
c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../com
mon/macros.c
gcc -Wall -I.. -I.. /lib -I.. /include -I.. /include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netutils.o netutils.
```

*** 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.

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /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
```

*** Config files installed ***

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

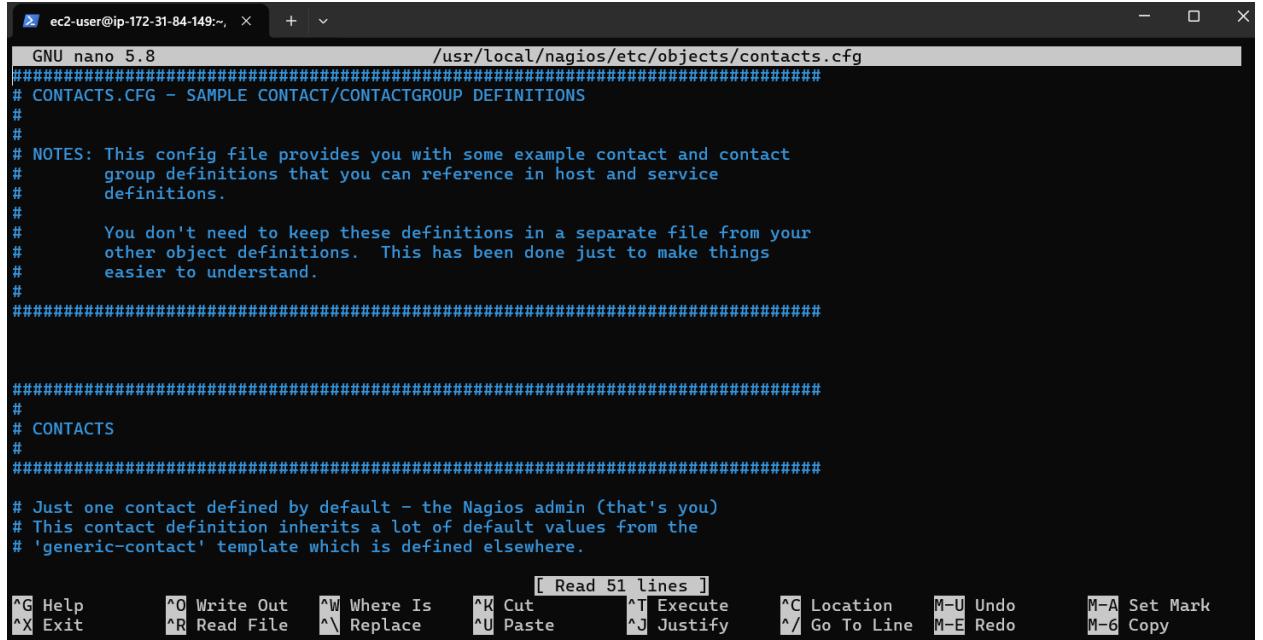
```
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
```

*** External command directory configured ***

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

Step 18: We need to update the email linked with this server to our email for it to send notifications (if any needed). **sudo nano /usr/local/nagios/etc/objects/contacts.cfg**

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg|
```



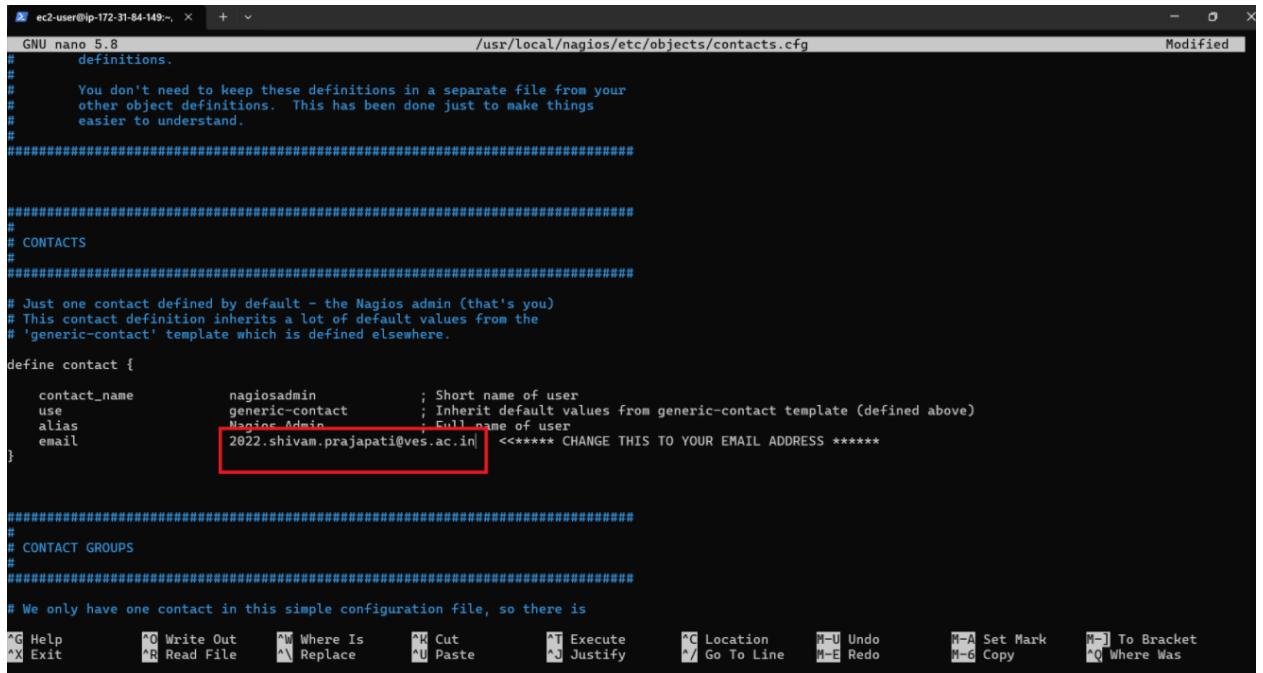
```

GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####
#
# CONTACTS
#
#####
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

[ Read 51 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^R Replace     ^U Paste     ^J Justify   ^L Go To Line M-E Redo   M-6 Copy

```

Here, change the email under ‘define contact{}’ to your email address



```

GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
Modified
#####
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####
#
# CONTACTS
#
#####
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name of user
    email            2022.shivam.prajapati@ves.ac.in <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
#
# CONTACT GROUPS
#
#####
# We only have one contact in this simple configuration file, so there is

[ Read 51 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^R Replace     ^U Paste     ^J Justify   ^L Go To Line M-E Redo   M-6 Copy
^M-J To Bracket
^Q Where Was

```

To save this use the following shortcut sequence CTRL+O→Enter→CTRL+X.

CTRL+O: Overwrite the existing file with edited file

CTRL+X: Exit nano editor

Step 19: We need to install the necessary configuration files for the Nagios web interface.
sudo make install-webconf

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ [ec2-user@ip-172-31-84-149 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

Step 20: Now we need to create a user to access the Nagios web interface. For that, run this command to create a user named '**nagiosadmin**'. Keep this username and password saved as it is needed to login to the web interface. **sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin**

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

Kingmaker is the password

Step 21: Now, we need to restart the Apache server to apply all the recent configurations. Use this command: **sudo service httpd restart**

```
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ |
```

Step 22: Now we go back to the downloads folder and extract the files of nagios plugin.
cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz (Version may vary)

```
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-84-149 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-84-149 downloads]$ |
```

```
[ec2-user@ip-172-31-84-149 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
```

Step 23: Again, we need to install the configurations for these files.

`cd nagios-plugins-2.4.11` (version may vary)

```
[ec2-user@ip-172-31-84-149 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ |
```

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

```
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
```

Step 24: We need to compile all the components of this software based on the instructions in the Makefile.

make**sudo make install**

```
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ make
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make all-recursive
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Nothing to be done for 'all-am'.
```

```
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ sudo make install
Making install in gl
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make install-recursive
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
if test yes = no; then \
  case 'linux-gnu' in \
    darwin[56]*) \
      need_charset_alias=true ;; \
    darwin* | cygwin* | mingw* | pw32* | cegcc*) \
      need_charset_alias=false ;; \
  *) \
    need_charset_alias=true ;; \
  esac ; \
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ |
```

Step 25: We need to register the Nagios service with the system to enable it to manage the server status

sudo chkconfig --add nagios**sudo chkconfig nagios on**

```
[ec2-user@ip-172-31-84-149 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.
```

Step 26: We need to verify the Nagios configuration for any syntax errors or issues before starting or restarting the Nagios service.

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

```
[ec2-user@ip-172-31-84-149 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...
```

sudo service nagios start

```
[ec2-user@ip-172-31-84-149 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-84-149 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 27: Check the status of the nagios.

```
[ec2-user@ip-172-31-84-149 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sat 2024-09-28 09:51:48 UTC; 43s ago
       Docs: https://www.nagios.org/documentation
    Process: 67663 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU)
   Process: 67664 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU)
 Main PID: 67665 (nagios)
   Tasks: 6 (limit: 4658)
  Memory: 5.8M
    CPU: 87ms
      CGroup: /system.slice/nagios.service
              ├─67665 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─67666 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67667 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67668 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67669 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─67670 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

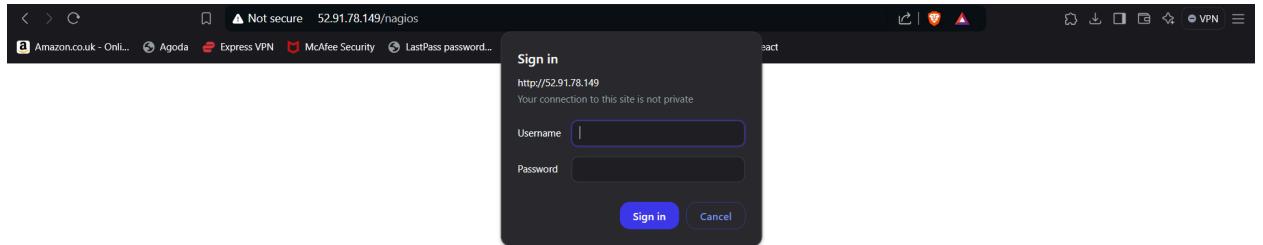
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successful
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: qh: core query handler registered
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: qh: echo service query handler registered
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: qh: help for the query handler registered
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: wproc: Successfully registered manager as @wproc with quer
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: wproc: Registry request: name=Core Worker 67669;pid=67669
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: wproc: Registry request: name=Core Worker 67666;pid=67666
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: wproc: Registry request: name=Core Worker 67667;pid=67667
Sep 28 09:51:48 ip-172-31-84-149.ec2.internal nagios[67665]: wproc: Registry request: name=Core Worker 67668;pid=67668
Sep 28 09:51:49 ip-172-31-84-149.ec2.internal nagios[67665]: Successfully launched command file worker with pid 67670
[ec2-user@ip-172-31-84-149 ~]$ |
```

Step 28: Go back to EC2 Console and copy the Public IP address of this instance. Open up your browser and look for **http://<your_public_ip_address>/nagios**

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, and various instance-related options. The main area displays an instance summary for an instance named i-03b653a01796c6519. The summary includes the following details:

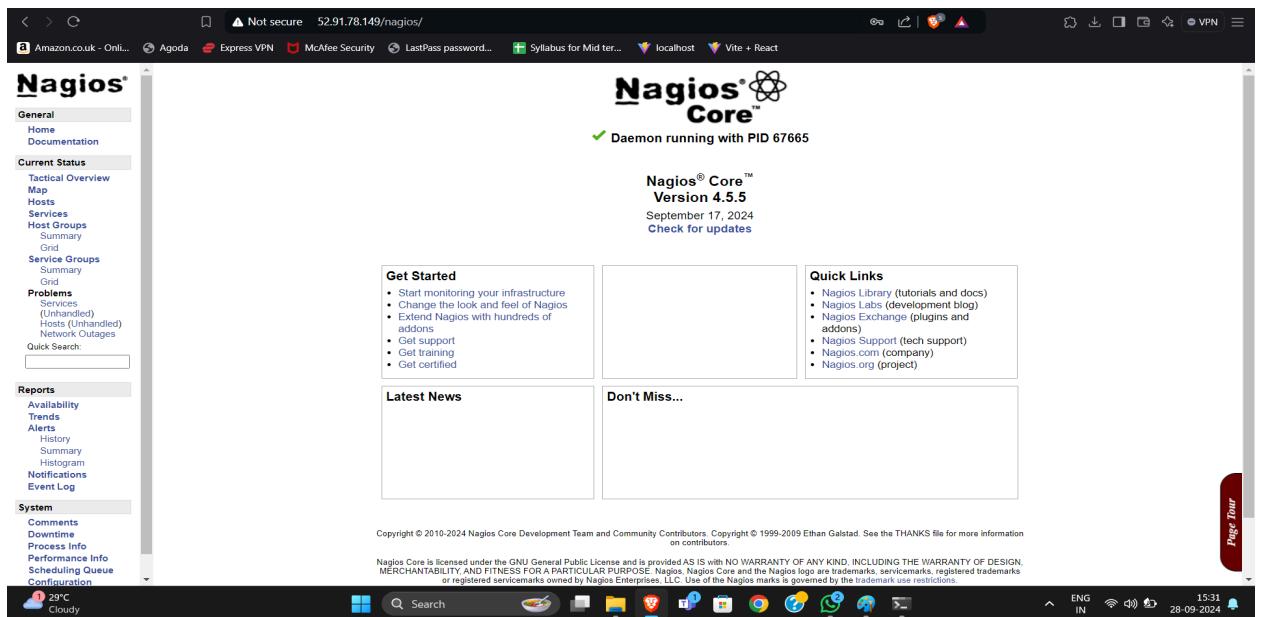
- Instance ID:** i-03b653a01796c6519 (Exp9_41)
- Public IPv4 address:** 52.91.78.149 [open address]
- Private IPv4 addresses:** 172.31.84.149
- Instance state:** Running
- Public IPv4 DNS:** ec2-52-91-78-149.compute-1.amazonaws.com [open address]
- Private IP DNS (IPv4 only):** ip-172-31-84-149.ec2.internal
- Instance type:** t2.medium
- Elastic IP addresses:** -
- VPC ID:** vpc-04fadfb026daa5d28
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations.
- Subnet ID:** subnet-0da4329f08fc9981a
- Auto Scaling Group name:** -
- Instance ARN:** arn:aws:ec2:us-east-1:380557944475:instance/i-03b653a01796c6519
- IAM Role:** -
- IMDSv2:** Required

[http://52.91.78.149/nagios.](http://52.91.78.149/nagios)



Enter **username as nagiosadmin** and **password as Kingmaker**.

Step 29: After entering the correct credentials, you will see this page



CONCLUSION:

In this experiment, we have learned how to install and configure Nagios Core, Nagios Plugins, and NRPE on a Linux machine. We used an Amazon Linux OS instance with the necessary security rules in place. It's important to ensure that the links for Nagios Core and Nagios Plugins are up to date (when using wget). After extracting and configuring these files, we should check for any issues before starting the server. Once everything is set up, we can start the Nagios server. By using the public IP address of the EC2 instance, we can access the Nagios dashboard by navigating to that IP followed by /nagios.

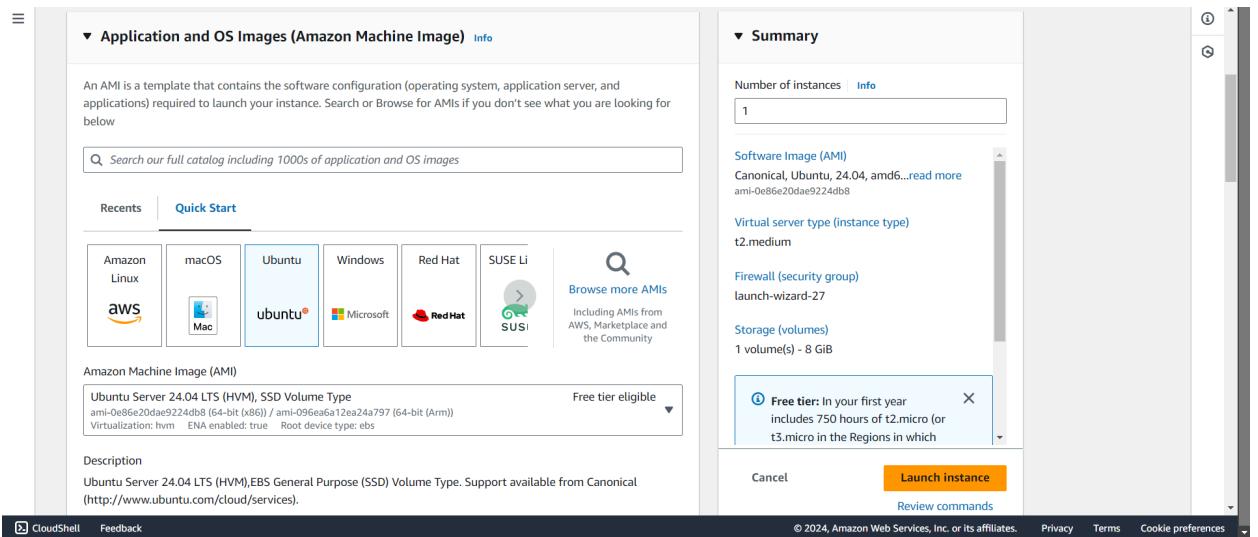
Experiment No:10

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

PREREQUISITES: We should have an Amazon Linux instance with nagios already set up.

Step 1: Set up ubuntu instance

- 1) Log in to your AWS account. Look for EC2 in the services menu. Open the interface and select Create Instance.



Select The OS Image as Ubuntu.

- 2) Ensure that you choose the same private key you created for the Amazon Linux instance. Additionally, select the same security group that you configured for the Linux instance.

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Key pair (login)

Key pair name - required: Lab9_41

Network settings

Network: vpc-04fadfb026daa5d28

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enabled

Configure storage

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64... ami-0e86e20dae9224db8

Virtual server type (instance type): t2.medium

Firewall (security group): launch-wizard-27

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

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6... ami-0e86e20dae9224db8

Virtual server type (instance type): t2.medium

Firewall (security group): launch-wizard-27

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

Launch Instance

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

Instance is:

nagios_client_41... i-0d74a72c6a0429781

Running

t2.medium

- 3) Now return to the instances screen. Click on the instance ID of your instance, then select Connect. Click on SSH client and copy the example command. Next, we need to connect our local OS terminal to the instance using SSH. To do this, open the terminal where the private key file (.pem) is stored. Paste the copied SSH command and execute it.

EC2 > Instances > i-0d74a72c6a0429781 > Connect to instance

Connect to instance Info

Connect to your instance i-0d74a72c6a0429781 (nagios_client_41Lab10) using any of these options

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

Instance ID
 [i-0d74a72c6a0429781](#) (nagios_client_41Lab10)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Lab9_41.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Lab9_41.pem"
4. Connect to your instance using its Public DNS:
 ec2-3-86-39-170.compute-1.amazonaws.com

Example:
 ssh -i "Lab9_41.pem" ubuntu@ec2-3-86-39-170.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.

Copy the example command

```
ubuntu@ip-172-31-86-92: ~ + 
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\praja\OneDrive\Desktop\NewFolder> ssh -i "Lab9_41.pem" ubuntu@ec2-3-86-39-170.compute-1.amazonaws.com
The authenticity of host 'ec2-3-86-39-170.compute-1.amazonaws.com (3.86.39.170)' can't be established.
ED25519 key fingerprint is SHA256:JPN0h3iHhSx0lnxMooo981B3xhjC8bD9+I1NuUgbyF4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-39-170.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 Sat Sep 28 11:36:50 UTC 2024

System load:  0.0          Processes:      112
Usage of /:   22.7% of 6.71GB  Users logged in:   0
Memory usage: 5%           IPv4 address for enX0: 172.31.86.92
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
```

Successfully connected the instance via SSH

Step 2: On Nagios Host machine (Linux) execute the following which we have already created as a prerequisites:

- 1) We need to verify whether the nagios service is running or not. For that, run this command : **ps -ef | grep nagios**

```
[ec2-user@ip-172-31-84-149 ~]$ ps -ef | grep nagios
nagios  67665      1  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  67666  67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67667  67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67668  67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67669  67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67670  67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
[ec2-user  70887  3276  0 10:48 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-84-149 ~]$ |
```

- 2) Next, switch to the root user and create a directory at the path '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts'.

sudo su

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

```
[ec2-user@ip-172-31-84-149 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]# |
```

- 3) We need to create a configuration file in this directory. To do this, copy the contents of the existing localhost configuration into the new file named 'linuxserver.cfg'.

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

```
[root@ip-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg': No such file or directory
```

So make the second directory again and run the cp command

```
[root@ip-172-31-84-149 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-84-149 ec2-user]# |
```

We need to make some changes in this config file. Open it using a nano editor.

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

```
[root@ip-172-31-84-149 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Change **hostname** and **alias** to **linuxserver**. Change address to **public ip address of client instance** (Ubuntu instance)

```
# Define a host for the local machine

define host {

    use          linux-server

    host_name    linuxserver
    alias        linuxserver
    address      3.86.39.170
}
```

Change hostgroup_name to **linux-servers1**

```
# 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
}

#####
#
# SERVICE DEFINITIONS
#
#####
```

Change the **occurrences of hostname** further in the document from **localhost** to **linuxserver**

Now, we need to edit the nagios configuration file to add this directory. Run this command

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

```
[root@ip-172-31-84-149 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

and add the following line **cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/**

```
GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg                         Modified
# 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/timerperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

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

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

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

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

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/monitors
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

OBJECT CACHE FILE
This option determines where object definitions are cached when

^G Help ^O Write Out ^W Where Is ^K Cut ^T 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 ^/ Go To Line M-E Redo M-C Copy ^Q Where Was

Now we verify the configuration files. **/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg**

```
[root@ip-172-31-84-149 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.
```

```

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-172-31-84-149 ec2-user]# |

```

Once the files are verified, we need to restart the server: **service nagios restart**

```

Things look okay - No serious problems were detected during
[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 ec2-user]# |

[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 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 Sat 2024-09-28 11:30:31 UTC; 3min 57s ago
       Docs: https://www.nagios.org/documentation
   Process: 73417 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 73418 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 73419 (nagios)
   Tasks: 6 (limit: 4658)
    Memory: 4.2M
      CPU: 113ms
     CGroup: /system.slice/nagios.service
             └─73419 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                 ├─73420 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                 ├─73421 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                 ├─73422 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                 ├─73423 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                 ├─73425 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73423;pid=73423
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73421;pid=73421
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73420;pid=73420
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73422;pid=73422
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: Successfully launched command file worker with pid 73425
Sep 28 11:32:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;1;connect to address 3.86.39.170 and port 80: Co
Sep 28 11:33:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;2;connect to address 3.86.39.170 and port 80: Co
Sep 28 11:34:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;3;connect to address 3.86.39.170 and port 80: Co
Lines 1-28/28 (END)

```

Step 3: Execute the following on Nagios Client machine (Ubuntu)

- 1) First, check for any available updates, and then proceed to install gcc, the Nagios NRPE server, and Nagios plugins.

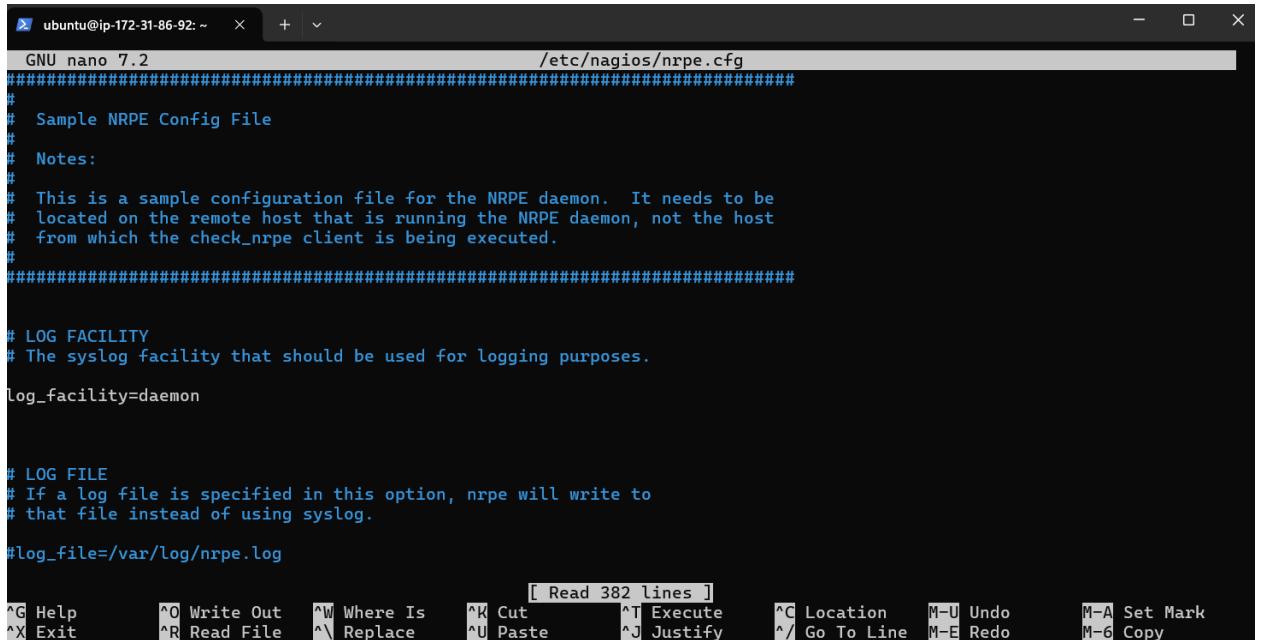
sudo apt update -y

```
sudo apt install gcc -y
```

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

```
[x] ubuntu@ip-172-31-86-92:~ x + ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-86-92:~$ sudo apt update -y  
sudo apt install gcc -y  
sudo apt install -y nagios-nrpe-server nagios-plugins  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]  
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]  
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]  
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]  
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]  
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]  
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]  
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]  
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [10.3 kB]  
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
```

- 2) We need to include the public IP address of our Nagios host machine (Linux) in the NRPE configuration file. **sudo nano /etc/nagios/nrpe.cfg**



```
ubuntu@ip-172-31-86-92: ~      + | 
GNU nano 7.2                      /etc/nagios/nrpe.cfg
#####
# Sample NRPE Config File
#
# Notes:
#
# This is a sample configuration file for the NRPE daemon. It needs to be
# located on the remote host that is running the NRPE daemon, not the host
# from which the check_nrpe client is being executed.
#
#####

# LOG FACILITY
# The syslog facility that should be used for logging purposes.

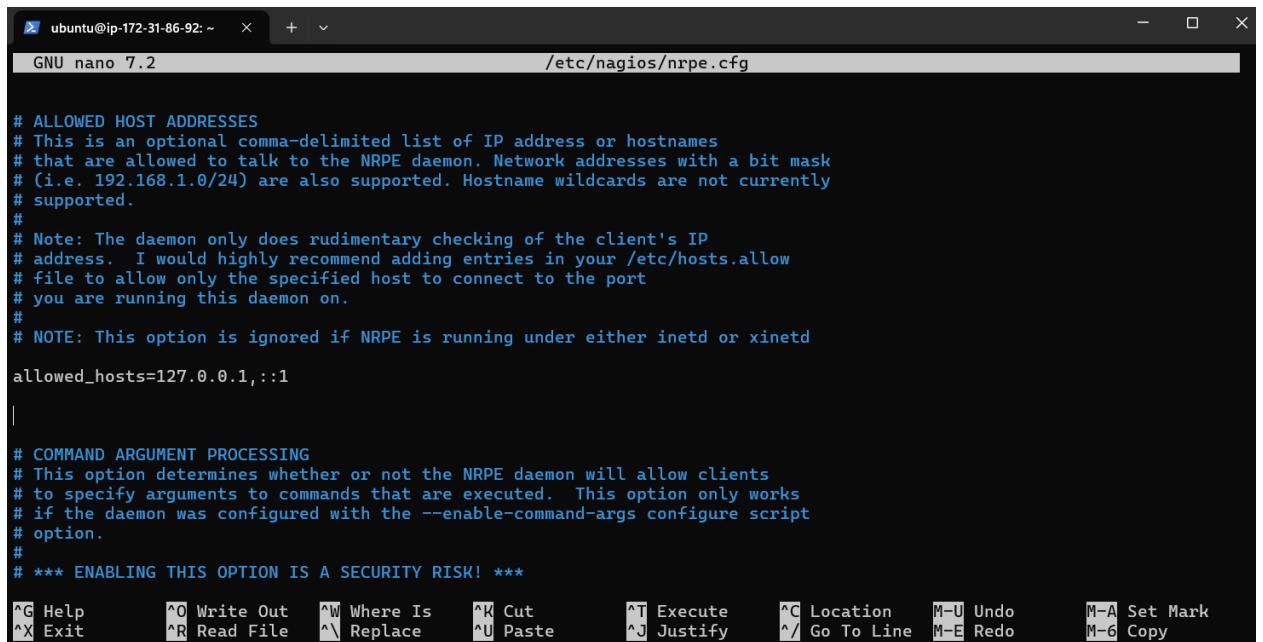
log_facility=daemon

# LOG FILE
# If a log file is specified in this option, nrpe will write to
# that file instead of using syslog.

#log_file=/var/log/nrpe.log

[ Read 382 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo    M-A Set Mark
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line M-E Redo    M-6 Copy
```

Under allowed_hosts, add the nagios host ip address (public)



```
ubuntu@ip-172-31-86-92: ~      + | 
GNU nano 7.2                      /etc/nagios/nrpe.cfg

# 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

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***

[ Read 382 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo    M-A Set Mark
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line M-E Redo    M-6 Copy
```

Step 4: Check the Nagios Dashboard. Go to Nagios dashboard, click on hosts. Here, we can see that the linuxserver is also added as a host.

The screenshot shows the Nagios web interface. In the top navigation bar, there are tabs for ChatGPT, Launch AWS Academy Learner, Instance details | EC2 | us-east, Instance details | EC2 | us-east, and Nagios: 54.210.81.106. The main content area displays the 'Host Status Details For All Host Groups' table. The table has columns: Host, Status, Last Check, Duration, and Status Information. It shows two hosts: 'linuxserver' (UP, UP) and 'localhost' (UP, UP). The 'Status Information' column indicates PING OK - Packet loss = 0%, RTA = 1.15 ms for both hosts.

| Host | Status | Last Check | Duration | Status Information |
|-------------|--------|---------------------|--------------|---|
| linuxserver | UP | 09-28-2024 04:42:16 | 0d 0h 3m 35s | PING OK - Packet loss = 0%, RTA = 1.15 ms |
| localhost | UP | 09-28-2024 04:38:21 | 0d 0h 24m 0s | PING OK - Packet loss = 0%, RTA = 0.03 ms |

Click on linuxserver. we can check all the information about linuxserver host.

The screenshot shows the Nagios host information page for 'linuxserver'. The left sidebar includes links for General, Current Status, Hosts, Services, Host Groups, Problems, Reports, and System. The main content area is divided into sections: Host Information, Host State Information, Host Commands, and Host Comments.

Host Information:

- Last Updated: Sat Sep 28 04:43:37 UTC 2024
- Nagios® Core™ 4.5.5 - www.nagios.org
- Logged in as nagiosadmin

Host State Information:

- Host Status: UP (for 0d 0h 3m 13s)
- Status Information: PING OK - Packet loss = 0%, RTA = 1.15 ms
- Performance Data: rta=1.151000ms;3000.000000;5000.000000;0.000000 pl=0% 80.100.0
- Current Attempt: 1/10 (HARD state)
- Last Check Time: 09-28-2024 04:42:16
- Check Type: ACTIVE
- Check Interval / Duration: 0.000 / 4.039 seconds
- Next Scheduled Check: 09-28-2024 04:47:16
- Last State Change: 09-28-2024 04:40:24
- Last Notification: N/A (notification 0)
- Is This Host Flapping?: NO (0.00% state change)
- In Scheduled Downtime?: NO
- Last Update: 09-28-2024 04:43:33 (0d 0h 0m 4s ago)

Host Commands:

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

Host Comments:

Add a new comment | Delete all comments

Entry Time Author Comment Comment ID Persistent Type Expires Actions

This host has no comments associated with it.

Click on services. Here we can see all the services that are being monitored by linuxserver.

The screenshot shows the Nagios monitoring interface with the URL <http://54.210.81.106/nagios/>. The dashboard displays current network status, host status totals (2 up, 0 down, 0 unreachable, 0 pending), and service status totals (10 OK, 1 Warning, 0 Unknown, 2 Critical, 3 Pending). The main table lists services for two hosts: 'linuxserver' and 'localhost'. For 'linuxserver', services include Current Load (OK), Current Users (OK), HTTP (CRITICAL), PING (OK), Root Partition (OK), SSH (PENDING), Swap Usage (PENDING), and Total Processes (PENDING). For 'localhost', services include Current Load (OK), Current Users (OK), HTTP (WARNING), PING (OK), Root Partition (OK), SSH (OK), Swap Usage (CRITICAL), and Total Processes (OK). A tooltip for the 'HTTP' service on 'localhost' indicates a 403 Forbidden error with a response time of 319 bytes over 0 seconds. The Nagios logo is visible at the bottom right of the interface.

CONCLUSION:

In this experiment, we learned to conduct port service monitoring and server monitoring using Nagios. To do this, we require a Linux instance to host the Nagios dashboard and a separate Ubuntu instance linked as a second host. We need to configure the Linux instance and include the IP address of the Ubuntu instance. Subsequently, we must replicate the initial setup from the Linux instance on the Ubuntu instance by adding the IP address of the Linux instance to the list of allowed hosts. After restarting the NRPE server, we should see the 'linuxserver' host listed.

Name: Aditya Dubey

Div: D15C

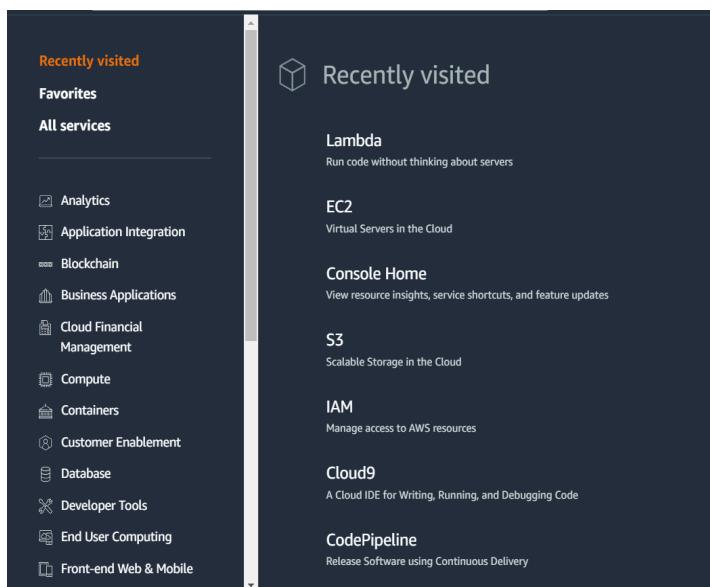
Roll No: 10

Experiment No: 11

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

CREATION OF LAMBDA FUNCTION:

Step1: Log in to your AWS Personal or Academy account. Navigate to Lambda, then select the 'Create Function' button

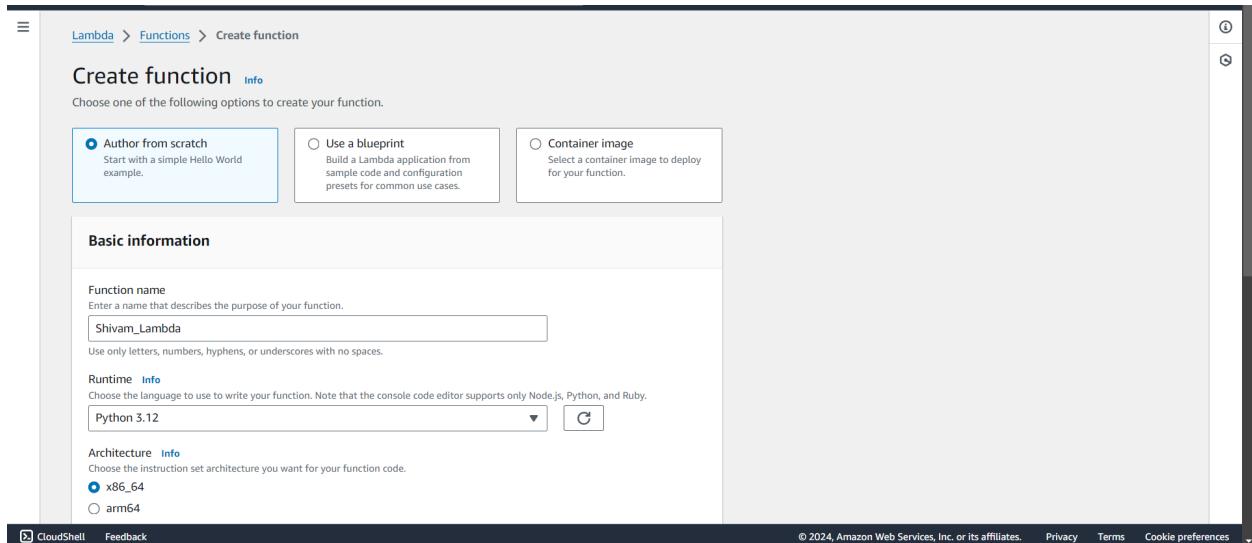


The screenshot shows the AWS Lambda Functions list page. The left sidebar has 'Lambda' selected, with 'Dashboard', 'Applications', and 'Functions' (which is underlined in blue). Below these are sections for 'Additional resources' (Code signing configurations, Event source mappings, Layers, Replicas) and 'Related AWS resources' (Step Functions state machines). The main area is titled 'Functions (5)' and shows a table with columns: Function name, Description, Package type, Runtime, and Last modified. The table contains five rows:

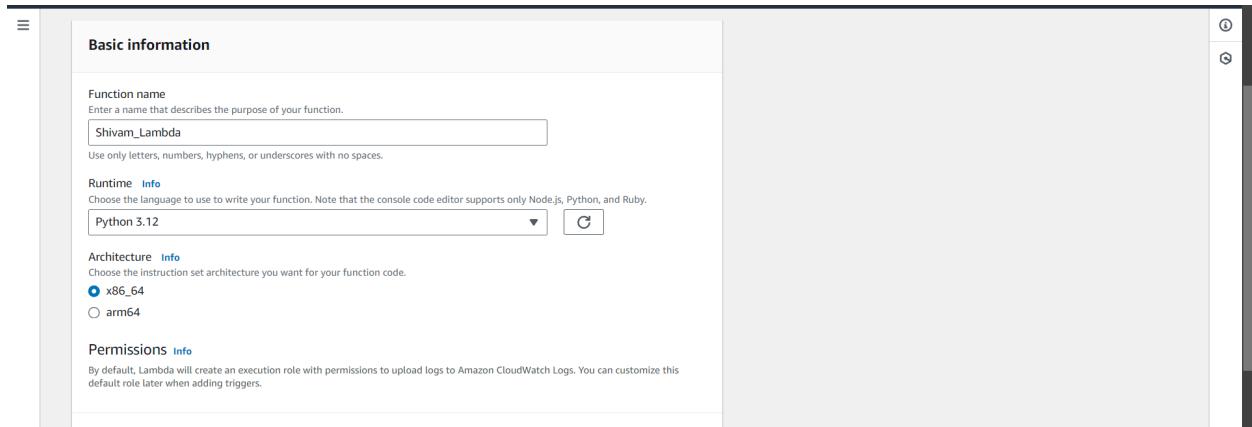
| Function name | Description | Package type | Runtime | Last modified |
|---------------------------|---|--------------|------------|---------------|
| ModLabRole | updates LabRole to allow it to assume itself | Zip | Python 3.8 | 2 months ago |
| RedshiftOverwatch | Deletes Redshift Cluster if the count is more than 2. | Zip | Python 3.8 | 2 months ago |
| RoleCreationFunction | Create SLR if absent | Zip | Python 3.8 | 2 months ago |
| MainMonitoringFunction | - | Zip | Python 3.8 | 2 months ago |
| RedshiftEventSubscription | Create Redshift event subscription to SNS Topic. | Zip | Python 3.8 | 2 months ago |

At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

Step 2: Give your Lambda function a name and choose a programming language. The code editor only supports Node.js, Python, and Ruby, so in my case I have chosen **Python 3.12**. Set the **architecture to x86**. For the execution role, select '**Use an existing role**', then pick '**Lab role**' from the dropdown menu under existing roles .
 (This is because the Lab role already has the permissions needed for Lambda to run properly, so you don't need to create a new role from scratch. It's a quicker and more convenient option)



Give the function name and select required language for lambda function

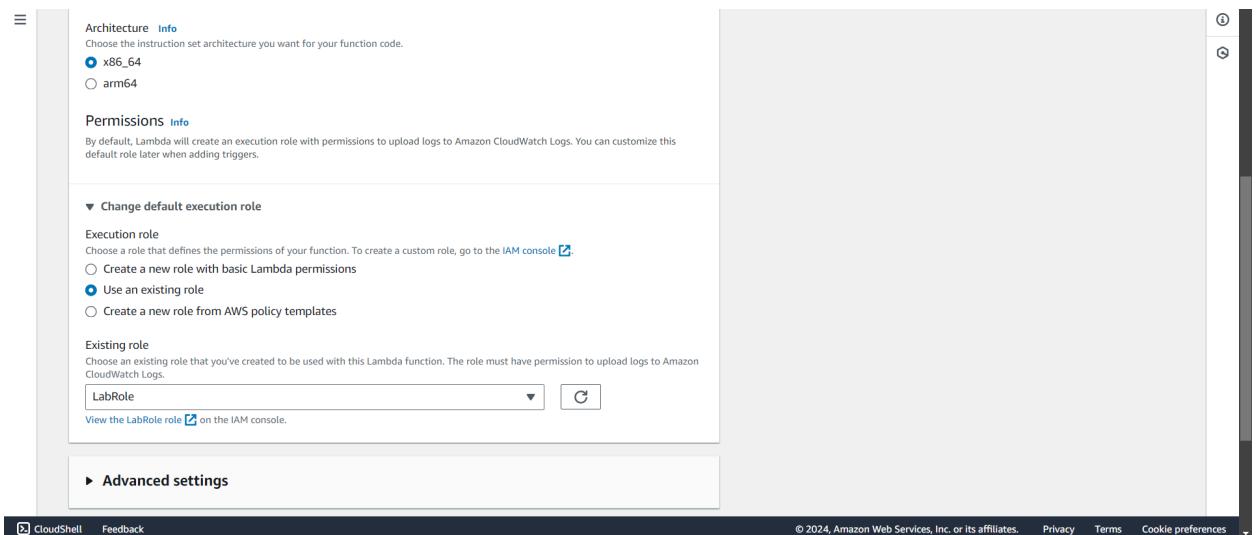


Architecture will be x86_64

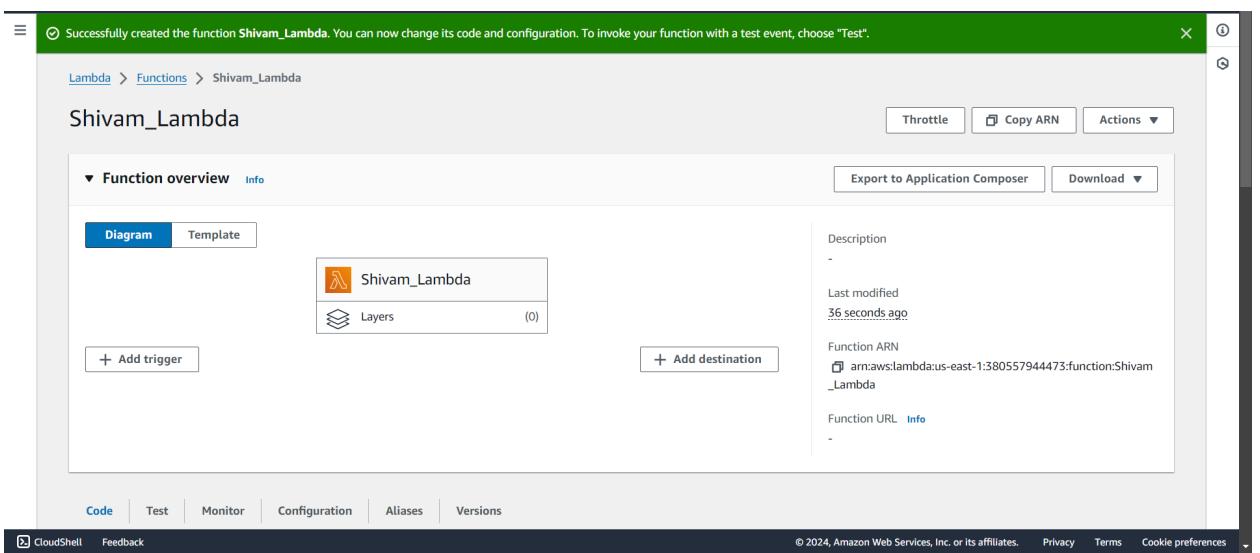
Name: Aditya Dubey

Div: D15C

Roll No: 10



Select proper Execution role



Successfully created Lambda function

Step 3: To view or change the basic settings, go to the 'Configuration' tab and click 'Edit' under 'General settings.' (THIS STEP IS OPTIONAL)

Name: Aditya Dubey

Div: D15C

Roll No: 10

The screenshot shows the AWS Lambda function configuration page for 'Shivam_Lambda'. The 'General configuration' tab is selected. On the left, a sidebar lists other configuration tabs: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, and Concurrency and recursion detection. The main panel displays the following configuration details:

| General configuration | | |
|-----------------------|-----------|-------------------|
| Description | Memory | Ephemeral storage |
| - | 128 MB | 512 MB |
| Timeout | SnapStart | |
| 0 min 3 sec | Info | |
| | None | |

At the bottom of the page, there is a URL bar with the address https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/Shivam_Lambda?tab=configure, and a footer with links to '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

You can add a description and adjust the memory and timeout settings. I've changed the timeout to 1 second, as that's enough for now.

The screenshot shows the 'Edit basic settings' page for the 'Shivam_Lambda' function. The top navigation bar shows the path: Lambda > Functions > Shivam_Lambda > Edit basic settings. The main section is titled 'Basic settings' and contains the following fields:

- Description - optional:** A text input field containing the value "Basic Setting of Shivam_Lambda".
- Memory:** A dropdown menu set to "128 MB". A note states: "Your function is allocated CPU proportional to the memory configured."
- Ephemeral storage:** A dropdown menu set to "512 MB". A note states: "You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)".
- SnapStart:** A dropdown menu set to "None". A note states: "Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#)".

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and a footer with links to '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Name: Aditya Dubey

Div: D15C

Roll No: 10

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).
None
Supported runtimes: Java 11, Java 17, Java 21.

Timeout
0 min 3 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
 Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role [on the IAM console](#).

Cancel **Save**

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

Click on Save

Step 4: Go to the 'Test' tab and click 'Create a new event.' Give the event a name, set 'Event Sharing' to private, and choose the 'hello-world' template.

We basically create a new event to test and verify your Lambda function; setting Event Sharing to private keeps it secure and choosing the "hello-world" template provides a simple structure for testing without complex inputs.

Code **Test** Monitor Configuration Aliases Versions

Test event [Info](#) Save Test

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
Event_41
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](#)

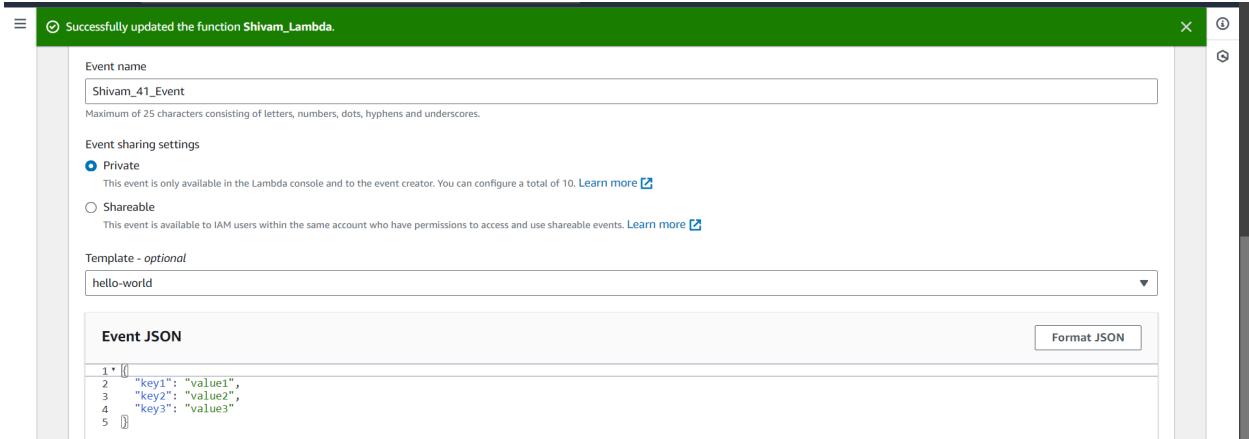
Template - optional

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

Name: Aditya Dubey

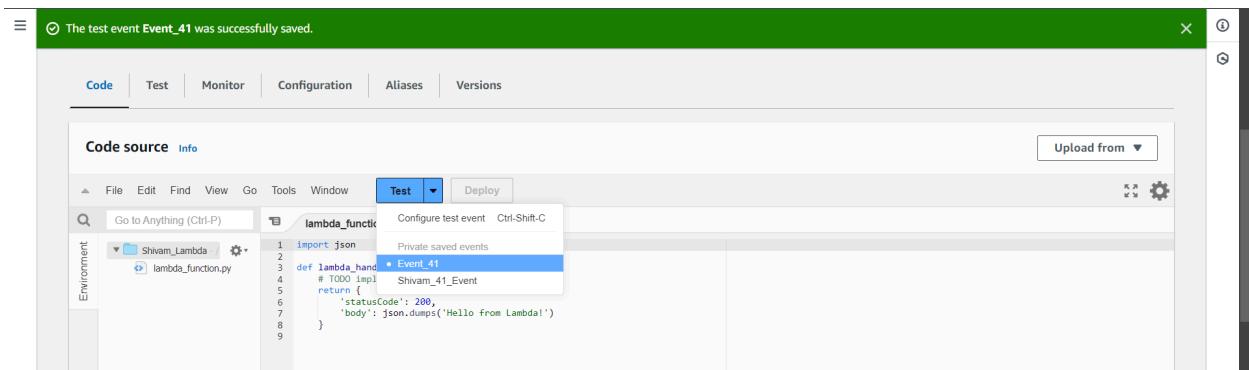
Div: D15C

Roll No: 10

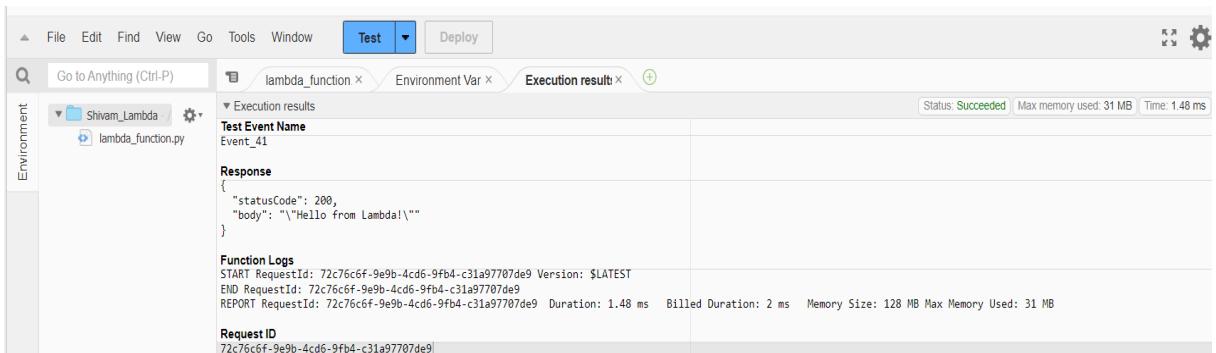


Click on save button above

Step 5: In the Code section, select the event you created from the dropdown menu under 'Test,' then click 'Test.' You should see the output below."

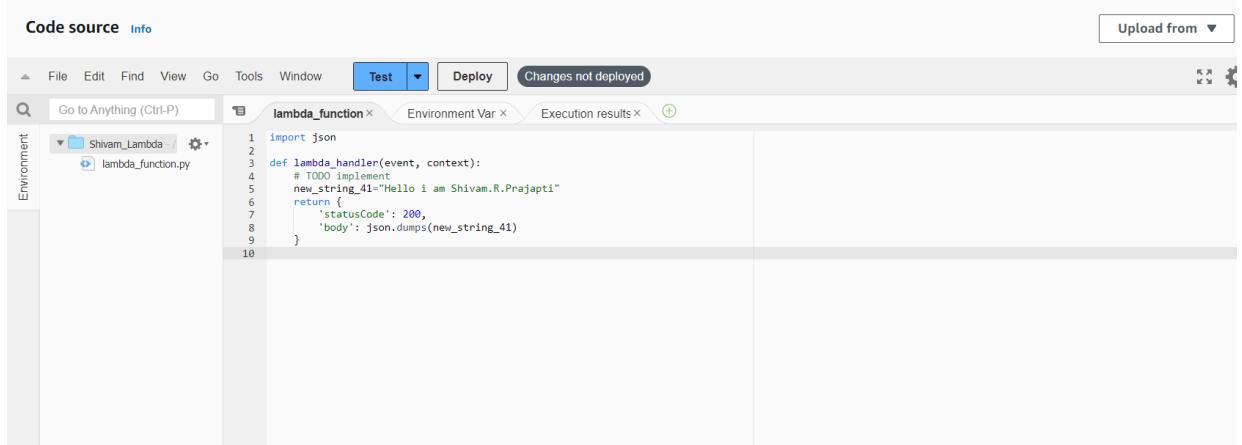


Output :

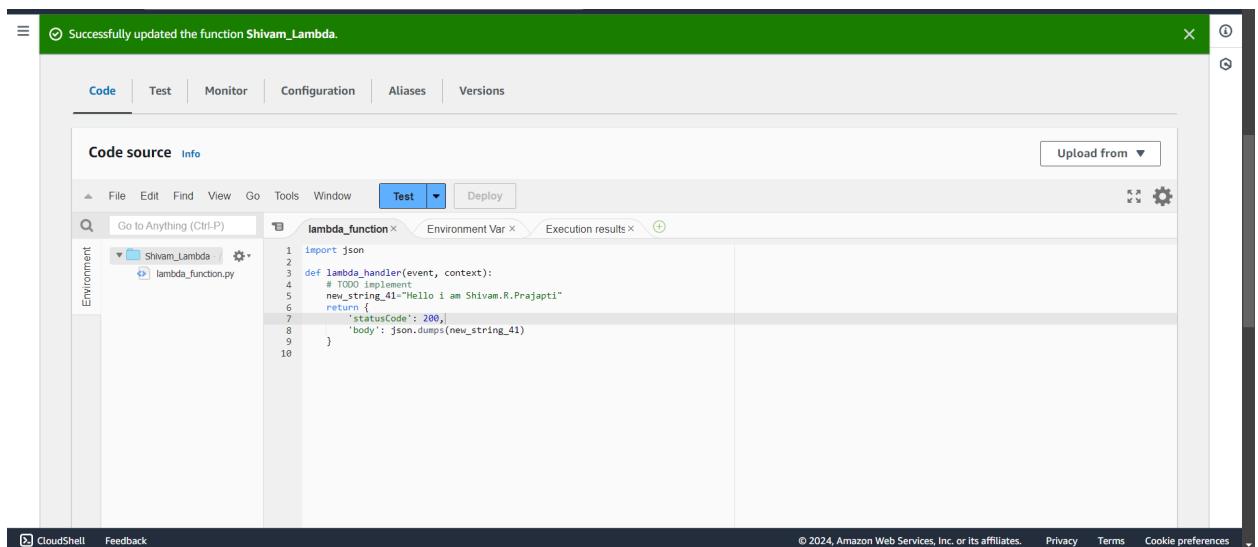


You select the created event to run the specific test you set up, and clicking 'Test' executes your Lambda function to check if it works as expected and produces the desired output.

Step 6: You can edit your lambda function code. I have changed the code to display the new String. After Changing save it by Control + S and click on Deploy . Make sure you have internet connectivity while deploying or else it will show failed deployment



```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string_41="Hello i am Shivam.R.Prajapti"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string_41)
9     }
10
```

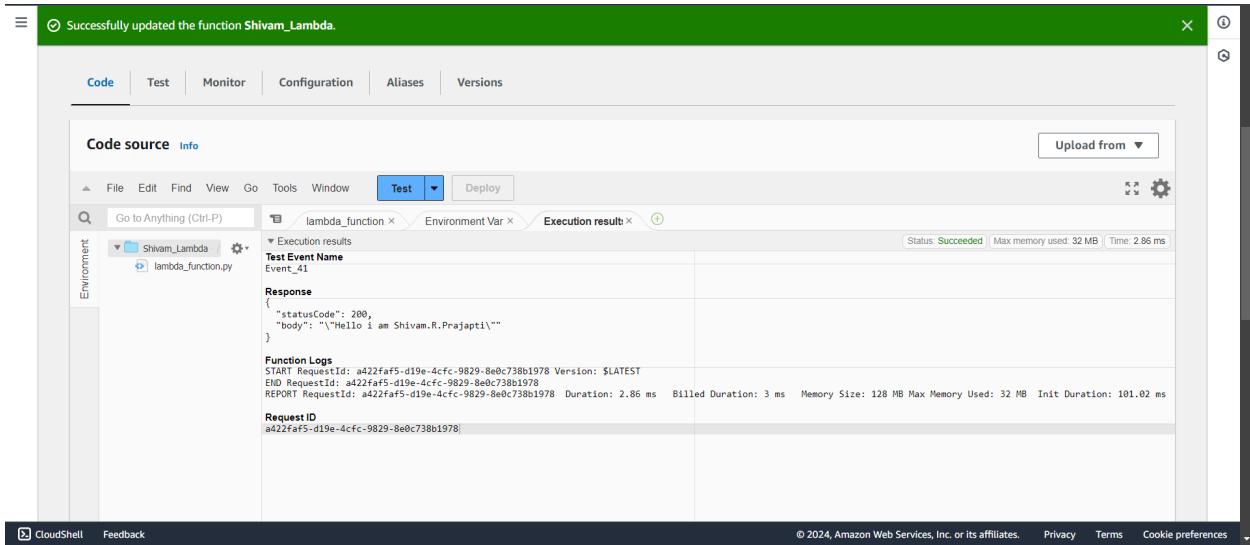


Successfully updated the function **Shivam_Lambda**.

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string_41="Hello i am Shivam.R.Prajapti"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string_41)
9     }
10
```

Successfully changed the function.

Step 7: Click on 'Test' to see the output. You'll get a status code of **200** which means "OK" and indicates that the request was successful, your string output, and the function logs, showing that it was deployed successfully.



CONCLUSION:

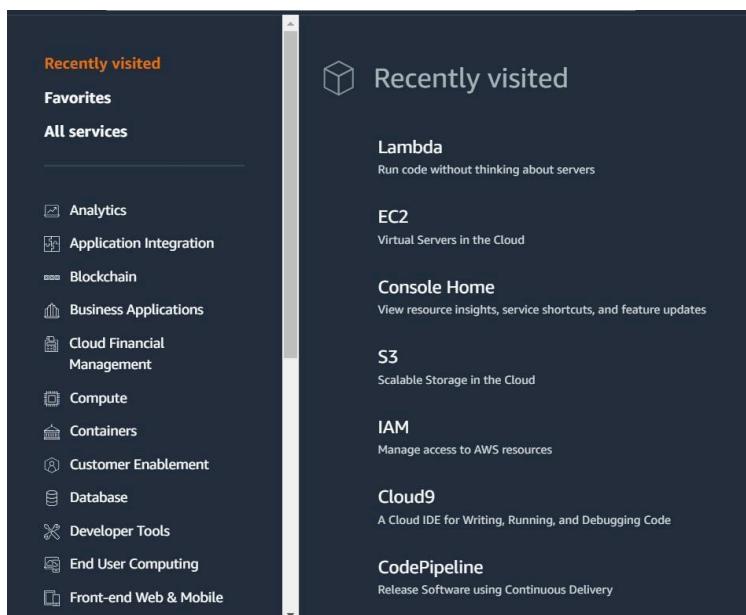
In this experiment, we successfully created an AWS Lambda function and followed the important steps involved. First, we set up the function using Python and adjusted the timeout setting to 1 second. Then, we created a test event to see how the function works and checked the output to ensure it was correct. We also modified the function's code and redeployed it to see the changes in real-time. So Lambda Function allows you to concentrate on writing code while AWS manages the infrastructure and automatically scales the service as needed. This makes it easier to develop and run applications without worrying about server management.

Experiment No :12

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

CREATING LAMBDA FUNCTION :

Step 1: Log in to your AWS Personal account. Then go to S3 in the services menu and click on "Create S3 Bucket."



The screenshot shows the 'Amazon S3' console. On the left, there is 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 for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, and Feature spotlight. The main area is titled 'Amazon S3' and shows an 'Account snapshot - updated every 24 hours' with a link to 'All AWS Regions'. Below this, there are tabs for 'General purpose buckets' (selected) and 'Directory buckets'. A search bar at the top says 'Find buckets by name'. The 'General purpose buckets' table lists four entries:

| Name | AWS Region | IAM Access Analyzer | Creation date |
|--|-----------------|---------------------|--|
| elasticbeanstalk-us-east-1-38055794473 | All AWS Regions | None | August 5, 2024, 14:37:53 (UTC+05:30) |
| my-video-stream | All AWS Regions | None | September 28, 2024, 20:18:57 (UTC+05:30) |
| www.ingmaker.com | All AWS Regions | None | August 4, 2024, 17:55:21 (UTC+05:30) |
| www.mywebsite.com | All AWS Regions | None | July 28, 2024, 18:09:18 (UTC+05:30) |

At the bottom, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. The footer includes links for CloudShell, Feedback, and various AWS policies.

Step 2: Give your bucket a name, select "General purpose project," then uncheck "Block public access." Keep the other settings as they are.

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type Info

- General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability zones.
- Directory

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability zone.

Bucket name Info

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

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

Given proper bucket Name

General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

Object Ownership Info

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

ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled

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

Object Ownership
Bucket owner enforced

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

Selected Appropriate object Ownership

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.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

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

Unchecked the block all public access checkbox and checked the lower checkbox.

Successfully created bucket "shivam41"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

▶ 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](#)

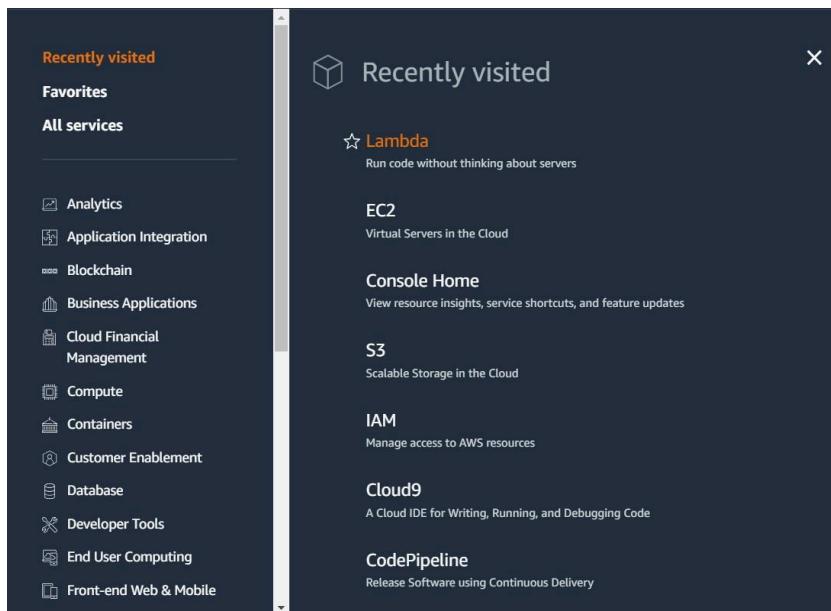
| General purpose buckets | | Directory buckets | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------------------------------|---|--|------------|---------------------|---------------|--|---------------------------------|---|--------------------------------------|---------------------------------|---------------------------------|---|--|--------------------------|---------------------------------|---|--|-----------------------------------|---------------------------------|---|--------------------------------------|-----------------------------------|---------------------------------|---|-------------------------------------|
| General purpose buckets (5) Info All AWS Regions Buckets are containers for data stored in S3. | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Find buckets by name <input type="text"/> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name ▲ AWS Region ▼ IAM Access Analyzer Creation date | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Name</th> <th>AWS Region</th> <th>IAM Access Analyzer</th> <th>Creation date</th> </tr> </thead> <tbody> <tr> <td>elasticbeanstalk-us-east-1-38055794473</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>August 5, 2024, 14:37:53 (UTC+05:30)</td> </tr> <tr> <td>my-video-stream</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>September 28, 2024, 20:18:57 (UTC+05:30)</td> </tr> <tr> <td>shivam41</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>September 30, 2024, 14:49:48 (UTC+05:30)</td> </tr> <tr> <td>www.kingmaker.com</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>August 4, 2024, 17:55:21 (UTC+05:30)</td> </tr> <tr> <td>www.mywebsite.com</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>July 28, 2024, 18:09:18 (UTC+05:30)</td> </tr> </tbody> </table> | | | Name | AWS Region | IAM Access Analyzer | Creation date | elasticbeanstalk-us-east-1-38055794473 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 5, 2024, 14:37:53 (UTC+05:30) | my-video-stream | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | September 28, 2024, 20:18:57 (UTC+05:30) | shivam41 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | September 30, 2024, 14:49:48 (UTC+05:30) | www.kingmaker.com | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 4, 2024, 17:55:21 (UTC+05:30) | www.mywebsite.com | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | July 28, 2024, 18:09:18 (UTC+05:30) |
| Name | AWS Region | IAM Access Analyzer | Creation date | | | | | | | | | | | | | | | | | | | | | | | |
| elasticbeanstalk-us-east-1-38055794473 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 5, 2024, 14:37:53 (UTC+05:30) | | | | | | | | | | | | | | | | | | | | | | | |
| my-video-stream | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | September 28, 2024, 20:18:57 (UTC+05:30) | | | | | | | | | | | | | | | | | | | | | | | |
| shivam41 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | September 30, 2024, 14:49:48 (UTC+05:30) | | | | | | | | | | | | | | | | | | | | | | | |
| www.kingmaker.com | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 4, 2024, 17:55:21 (UTC+05:30) | | | | | | | | | | | | | | | | | | | | | | | |
| www.mywebsite.com | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | July 28, 2024, 18:09:18 (UTC+05:30) | | | | | | | | | | | | | | | | | | | | | | | |

<https://us-east-1.console.aws.amazon.com/s3/access?region=us-east-1>

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

Successfully created the bucket

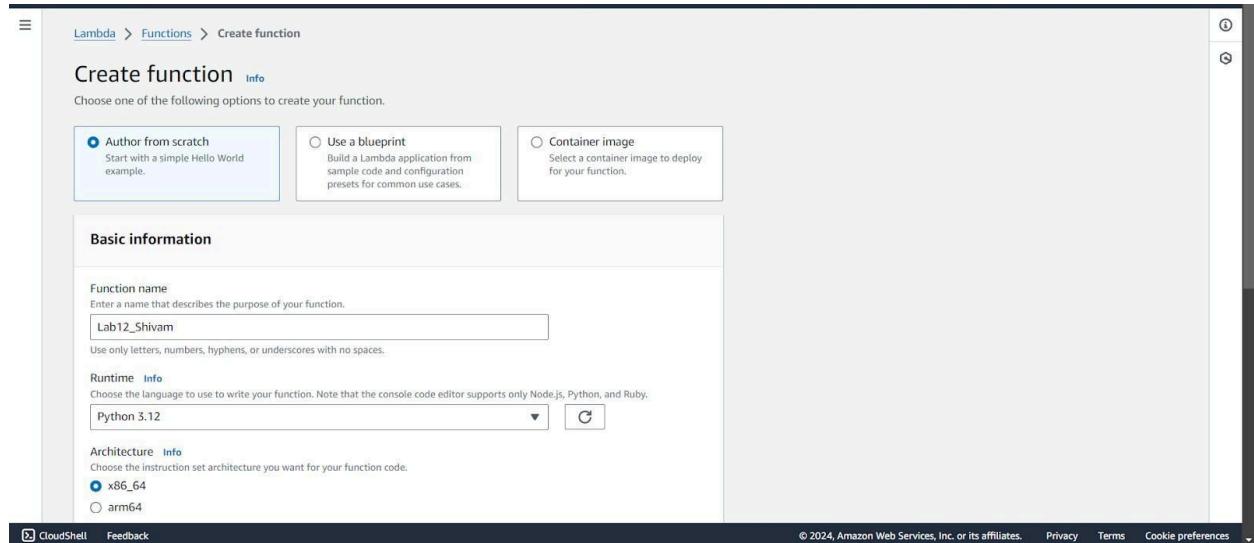
Step 3: Open lambda console and click on create function button.



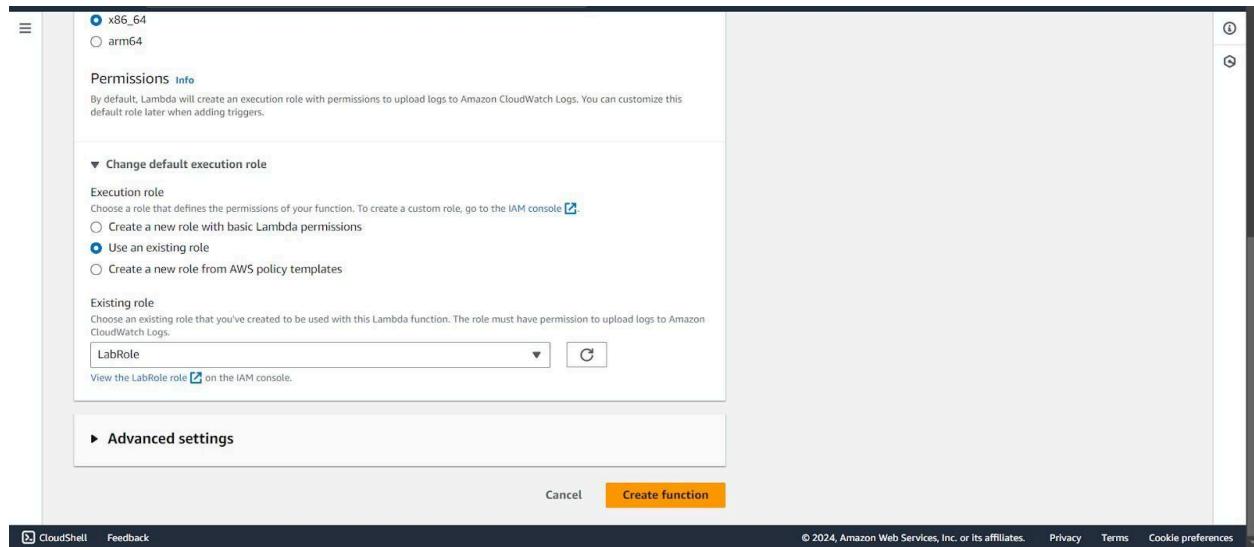
A screenshot of the AWS Lambda Functions page. The left sidebar shows 'Lambda' with 'Dashboard', 'Applications', and 'Functions' selected. Under 'Additional resources', there are links for 'Code signing configurations', 'Event source mappings', 'Layers', and 'Replicas'. Under 'Related AWS resources', there is a link for 'Step Functions state machines'. The main content area shows a table titled 'Functions (6)'. The table has columns for 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. The first function listed is 'Shivam_Lambda', which is described as 'Basic-Setting of Shivam_Lambda'. It was created with Zip package type, Python 3.12 runtime, and was modified 15 minutes ago. The second function is 'ModLabRole', which updates LabRole to allow it to assume itself. It was created with Zip package type, Python 3.8 runtime, and was modified 2 months ago. The third function is 'RedshiftOverwatch', which deletes Redshift Cluster if the count is more than 2. It was created with Zip package type, Python 3.8 runtime, and was modified 2 months ago. At the bottom of the table, there are buttons for 'Role creation' and 'Create SLR'. The top right of the page has a 'Create function' button. The footer includes links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

Step 4: Give your Lambda function a name and choose a programming language. The code editor only supports Node.js, Python, and Ruby, so in my case I have chosen **Python 3.12**. Set the **architecture to x86**. For the execution role, select '**Use an existing role**', then pick '**Lab role**' from the dropdown menu under existing roles .

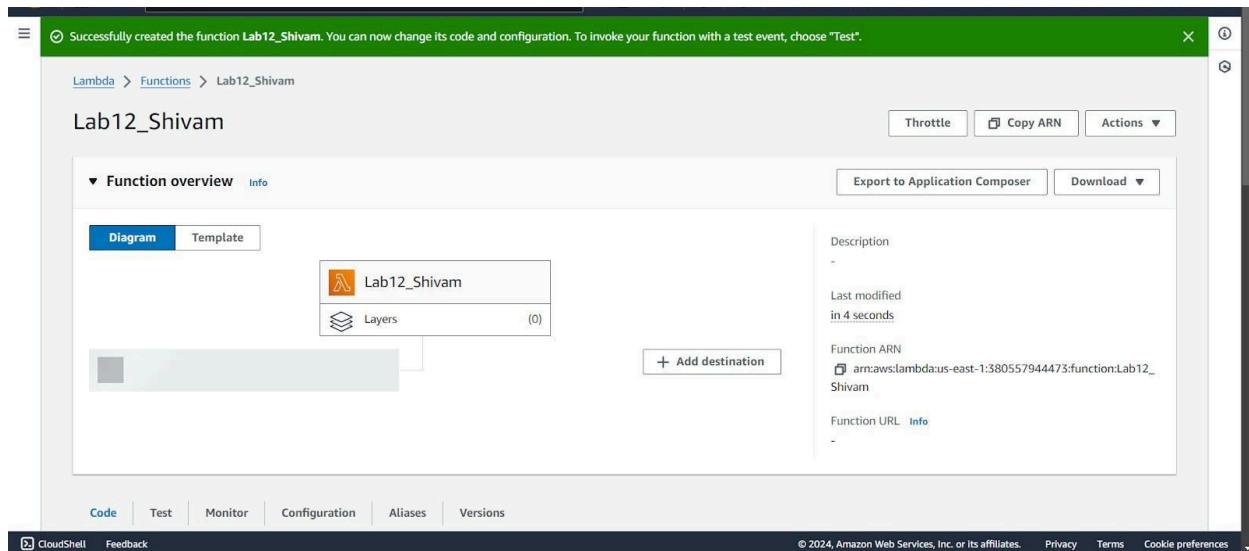
(This is because the Lab role already has the permissions needed for Lambda to run properly, so you don't need to create a new role from scratch. It's a quicker and more convenient option)



Given proper function name and selected language for Lambda function

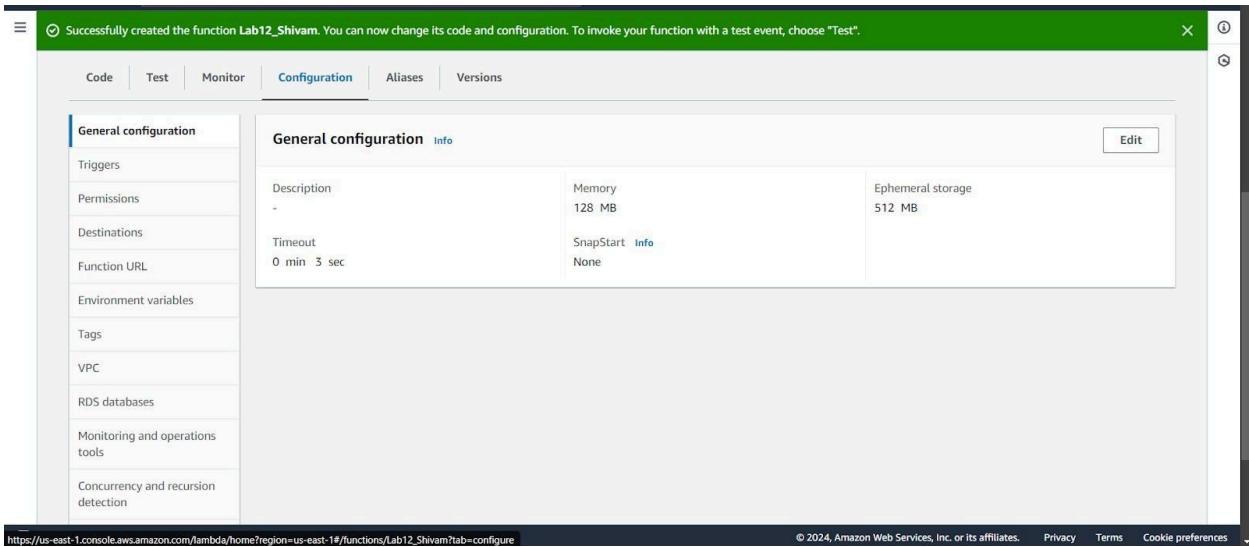


Selected Appropriate Execution role

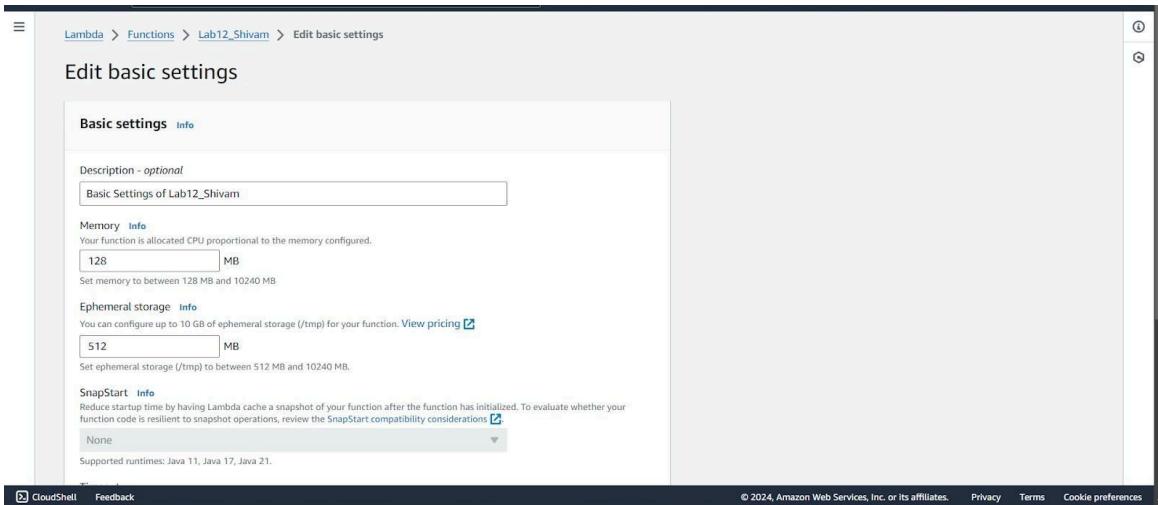


Successfully created the Lambda function.

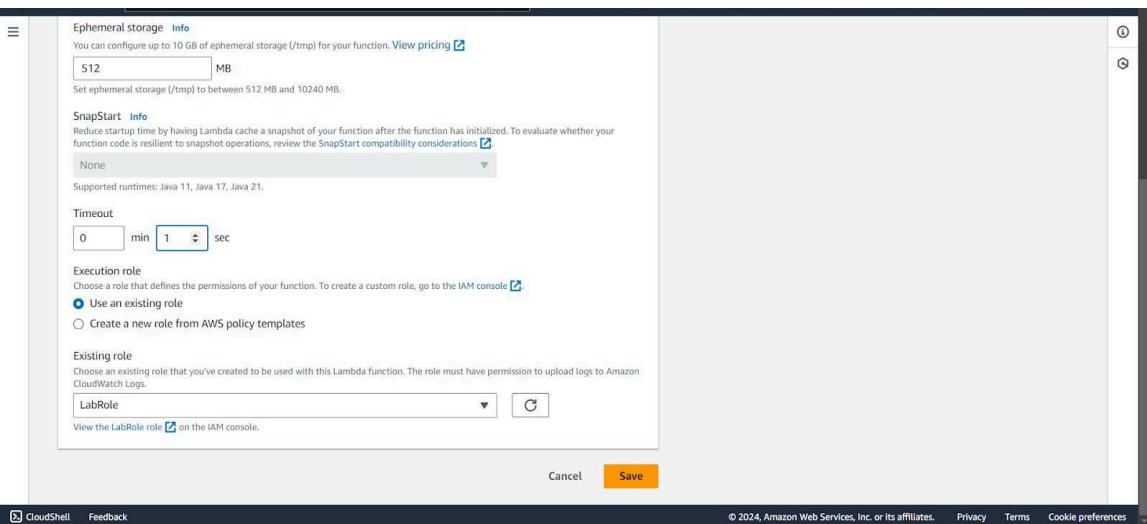
Step 5: To view or change the basic settings, go to the 'Configuration' tab and click 'Edit' under 'General settings.' (THIS STEP IS OPTIONAL)



You can add a description and adjust the memory and timeout settings. I've changed the timeout to 1 second, as that's enough for now.

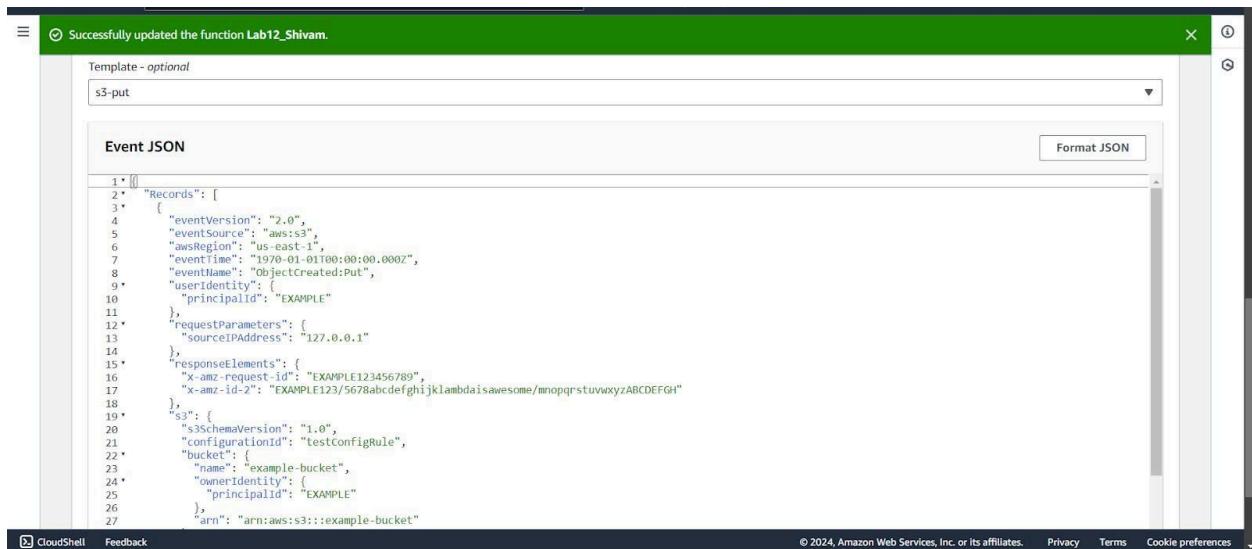
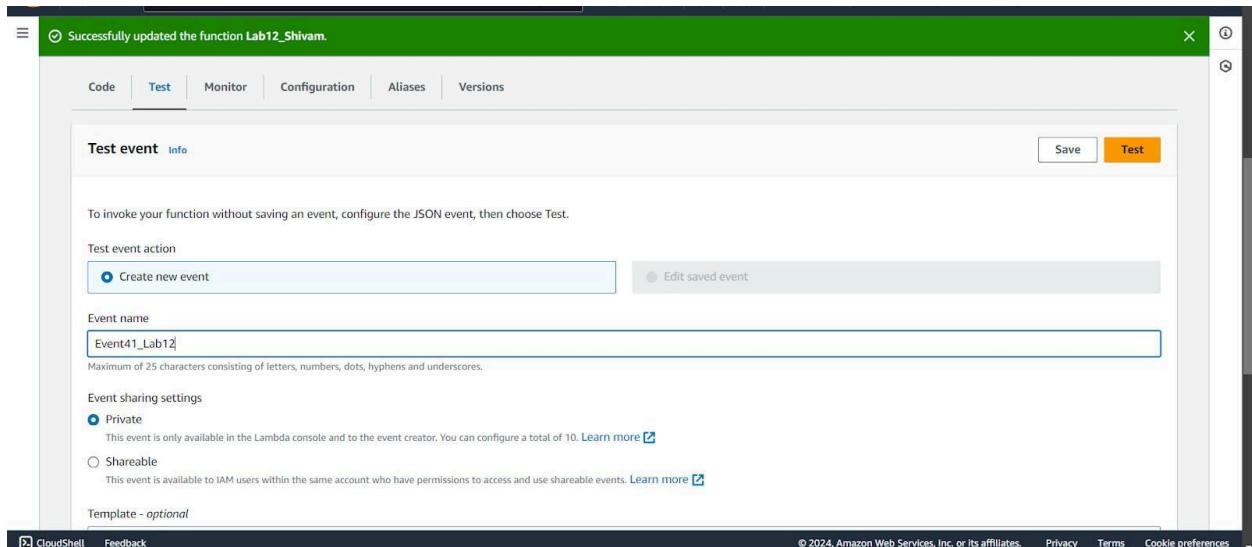


Given some description for your settings



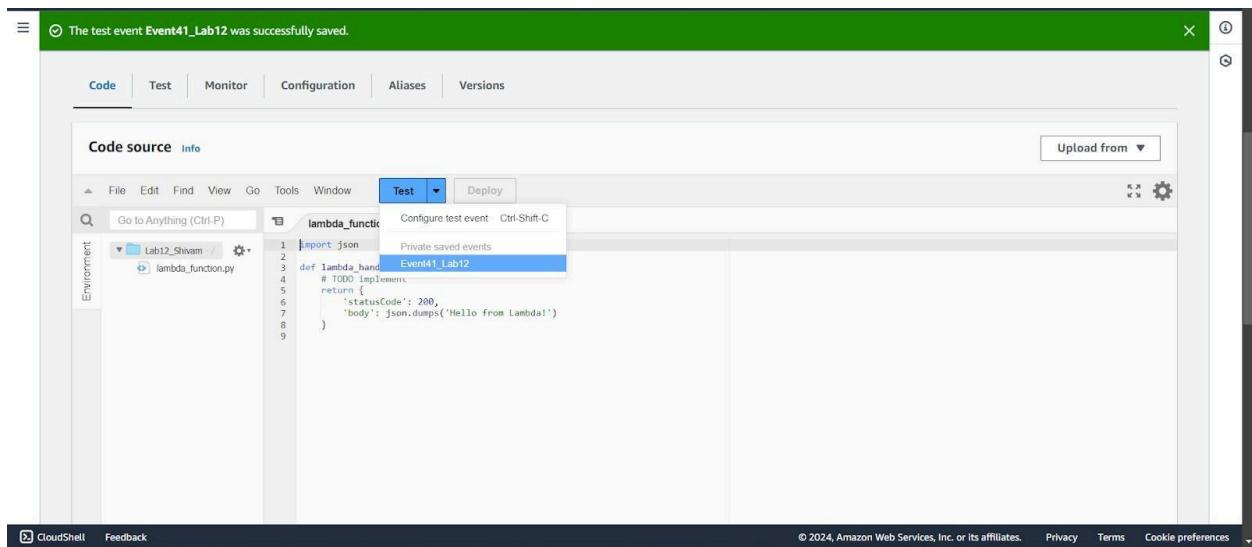
Click on Save.

Step 6: Click on the "Test" tab, then select "Create a new event." Give the event a name, set "Event Sharing" to private, and choose the "S3 Put" template.S3 (Simple Storage Service) template allows you to test your Lambda function specifically for events related to uploading files to an S3 bucket.

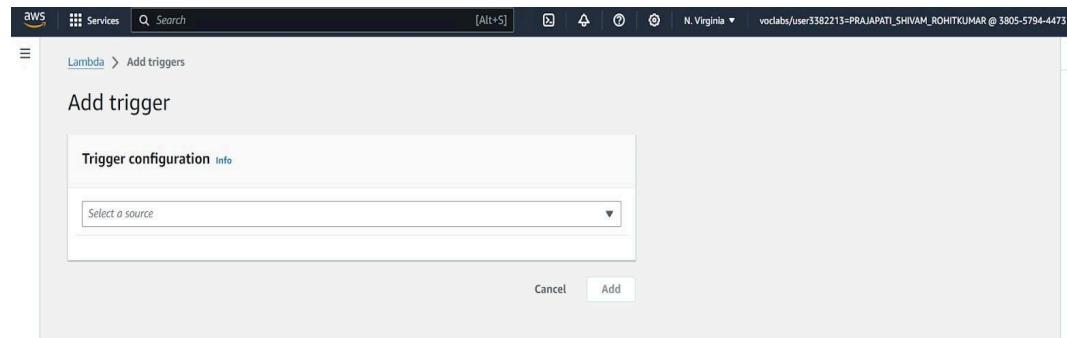


Event Jason code will be automatically generated once S3 -put is selected.

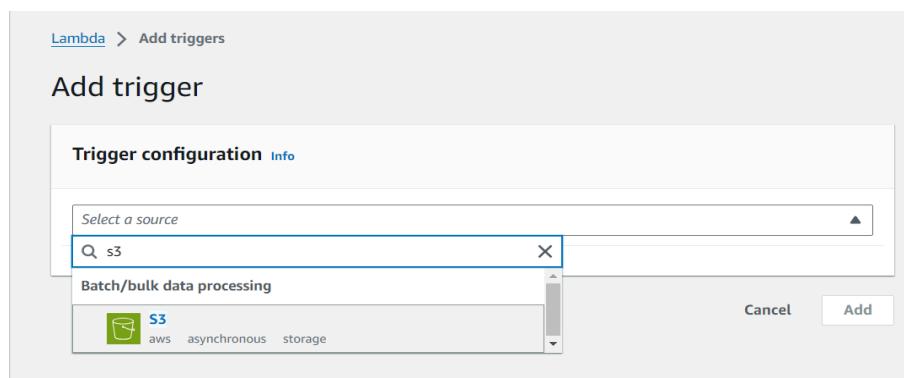
Step 7: Now In the Code section select the created event from the dropdown .



Step 8: In the Lambda function, click on "Add Trigger." Adding a trigger allows your Lambda function to automatically run in response to specific events such as uploads to an S3 bucket



Now select the source as S3, then choose the bucket name from the dropdown menu. Keep the other settings as default, and you can also add a prefix for the image if you want. A prefix for an image (or any file) in S3 is a string that you can use to organize or filter files within a bucket. It acts like a folder name, helping to categorize your files.

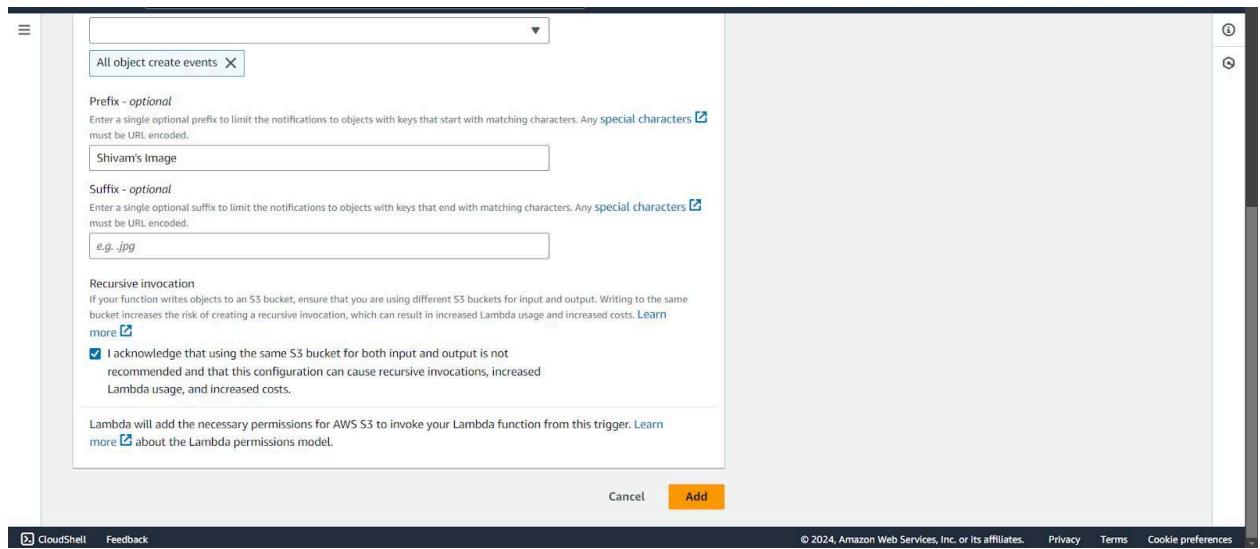


Name:Aditya Dubey

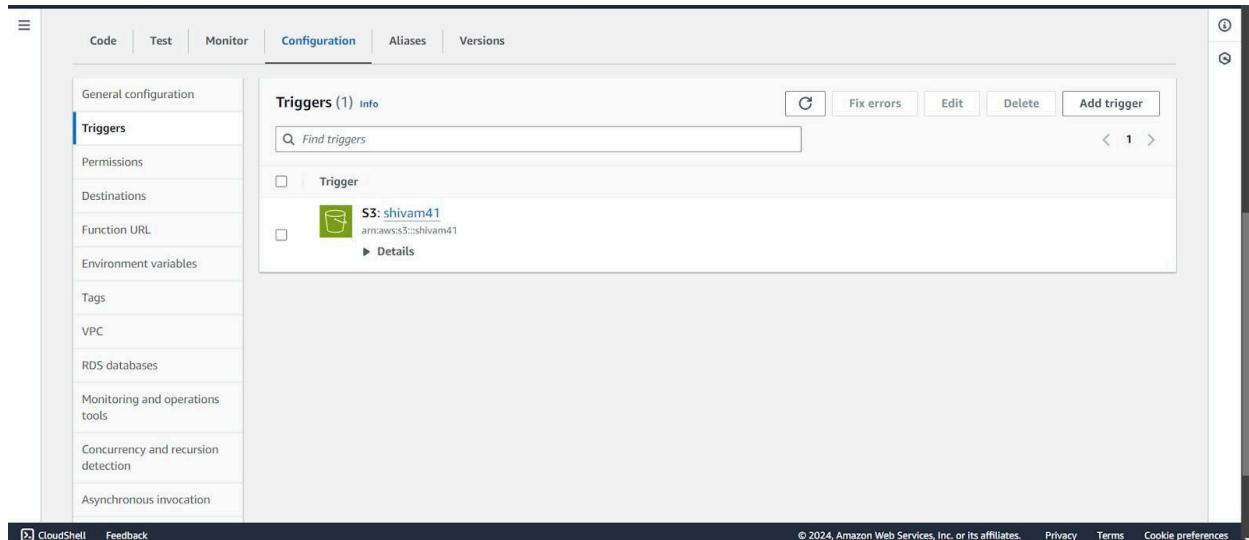
Div:D15C

Roll NO :10

The screenshot shows the 'Add trigger' configuration page for an AWS Lambda function. The top navigation bar includes 'Lambda > Add triggers' and a 'Trigger configuration' section with a 'Info' link. The main area is titled 'Add trigger' and contains a 'Trigger configuration' panel. In the 'Event source' dropdown, 'S3' is selected. Below it, a search bar shows 's3/shivam41' and a 'Bucket region' dropdown set to 'us-east-1'. Under 'Event types', 'All object create events' is selected. A 'Prefix - optional' field is present but empty. At the bottom of the panel are 'CloudShell', 'Feedback', and 'Changelog' links. The footer of the page includes copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.



Click on save



Step 9: 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 editor. The tab bar at the top has 'lambda_function' and 'Environment Var'. The code editor contains the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     print(f"An image has been added to the bucket {bucket_name}: {object_key}")
9
10    return {
11        'statusCode': 200,
12        'body': json.dumps('Log entry created successfully!')
13    }
```

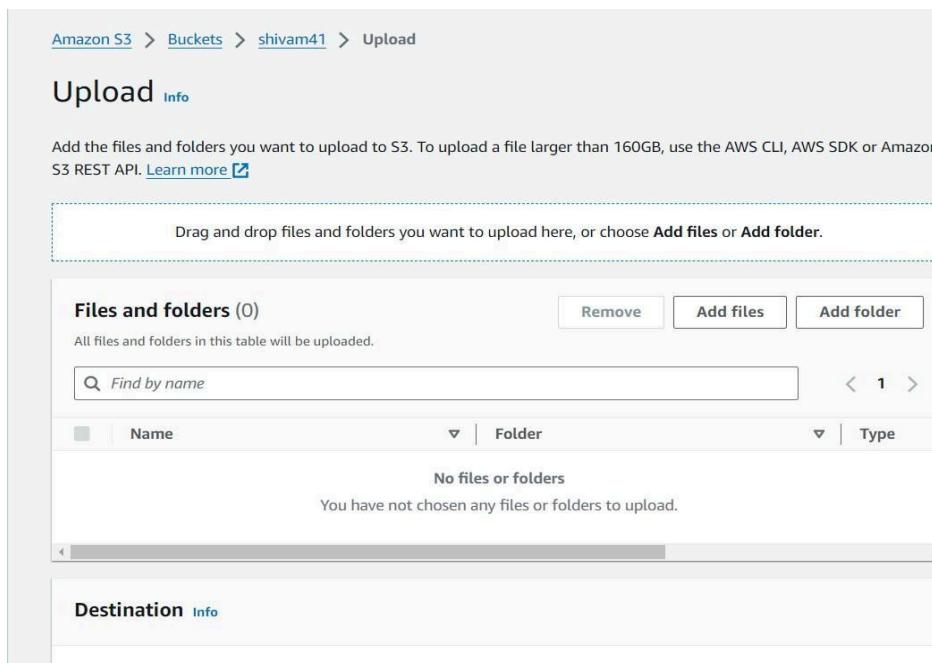
Save it by Control + S and deploy

The screenshot shows the AWS Lambda function configuration interface. At the top, a green banner displays the message "Successfully updated the function Lab12_Shivam.". Below this, the "Code source" tab is selected, showing the code for the lambda_function. The code is a Python script named lambda_function.py:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     print(f"An image has been added to the bucket {bucket_name}: {object_key}")
9
10    return [
11        {
12            'statusCode': 200,
13            'body': json.dumps('Log entry created successfully!')
14        }
15    ]
```

The interface includes tabs for "Test" and "Deploy", and a sidebar for "Environment". At the bottom, there are links for CloudShell, Feedback, and various AWS terms like Privacy, Terms, and Cookie preferences.

Step 10: Now we will upload any image to the bucket



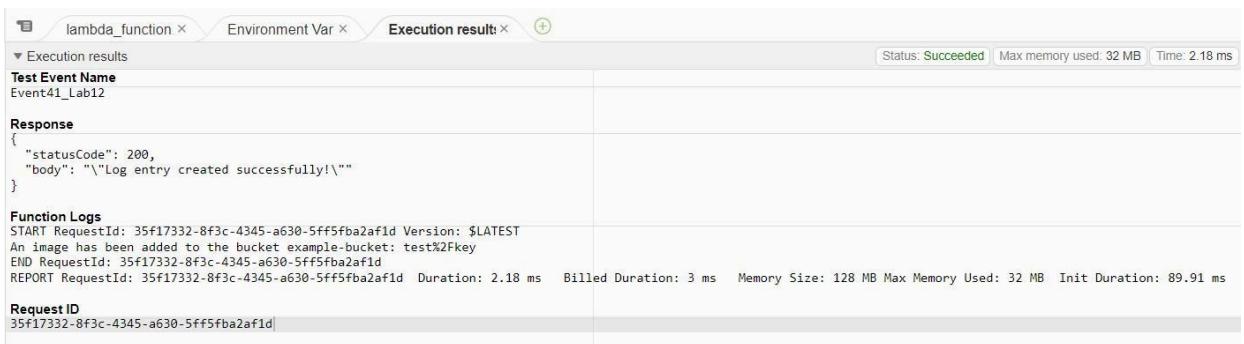
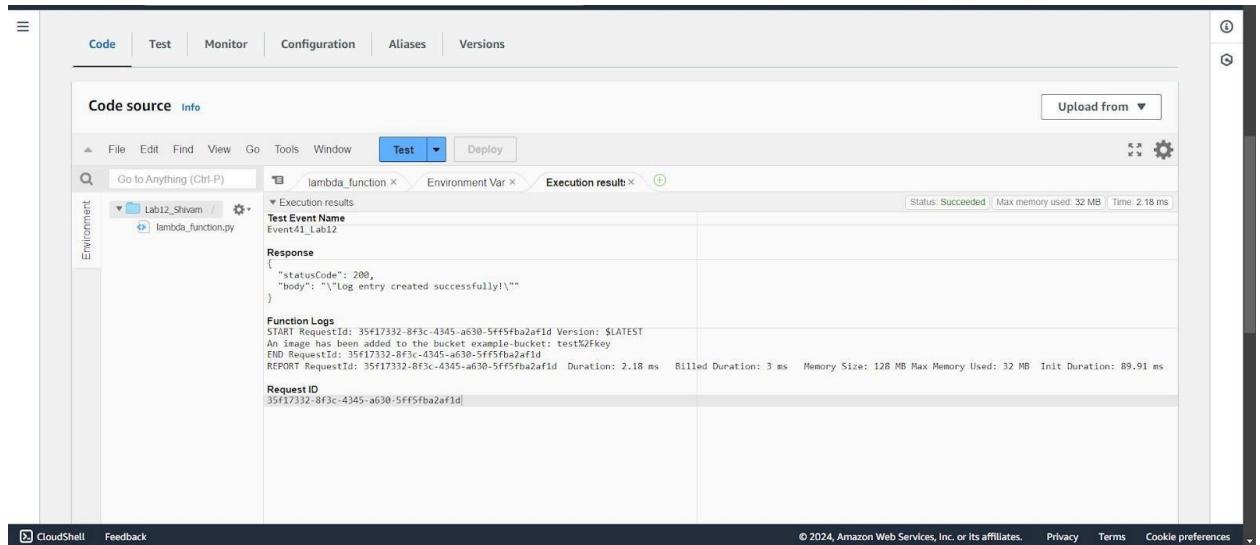
Click on add file where you can upload any image of your choice in your bucket

The screenshot shows the AWS S3 'Upload' interface. At the top, the path 'Amazon S3 > Buckets > shivam41 > Upload' is visible. Below it, the 'Upload' section has a note: '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' with a link icon. A central area says 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a table titled 'Files and folders (1 Total, 255.5 KB)'. It lists one item: 'back_ground.png' (image/png). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar 'Find by name' is present. The 'Destination' section shows 'Destination' set to 's3://shivam41'. The bottom navigation bar includes 'CloudShell', 'Feedback', and links to '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

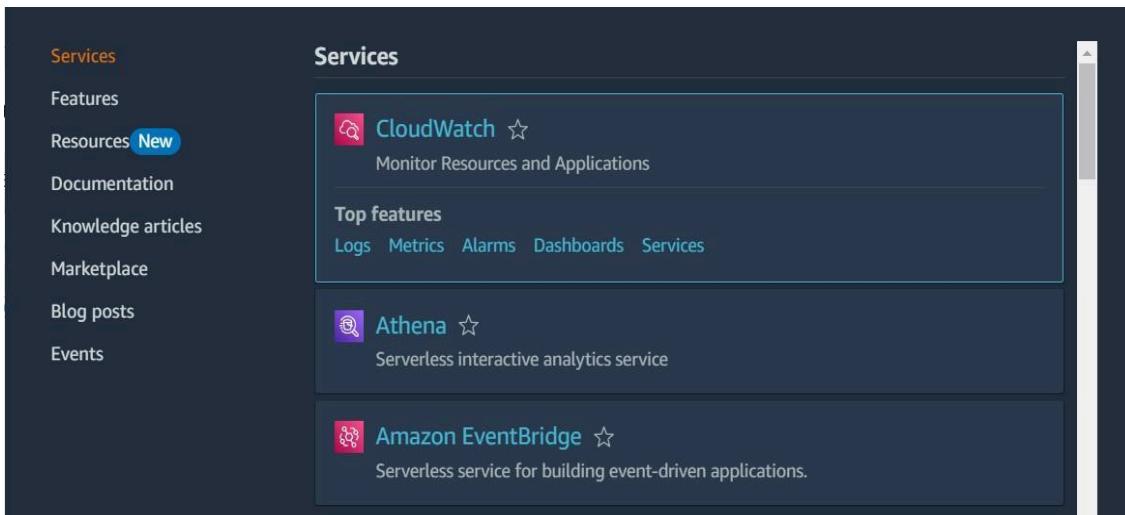
This screenshot shows the same AWS S3 upload interface as above, but with additional expanded sections. The 'Destination details' section is open, showing 'Bucket settings that impact new objects stored in the specified destination'. Below it, the 'Permissions' and 'Properties' sections are also expanded. The 'Upload' button is highlighted in orange at the bottom right. The rest of the interface is identical to the first screenshot.

Click on Upload

Step 11: Now click on "Test" in Lambda to see if it logs the activity when an image is added to S3.



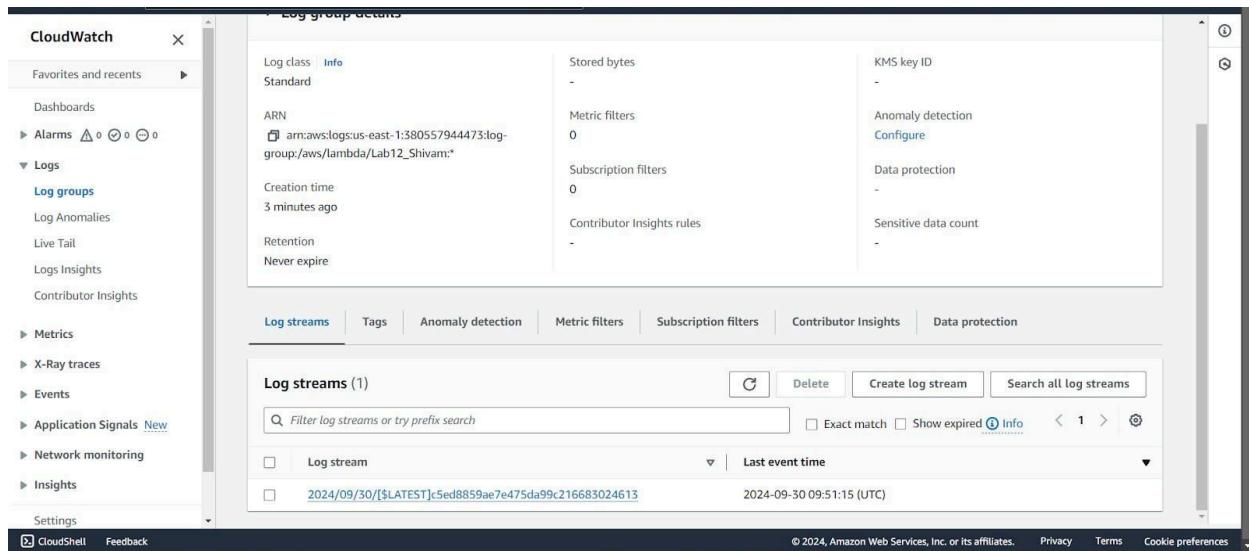
Step 12: Now let's check the logs on CloudWatch. Go to the "Monitor" section and click on "View CloudWatch Logs".



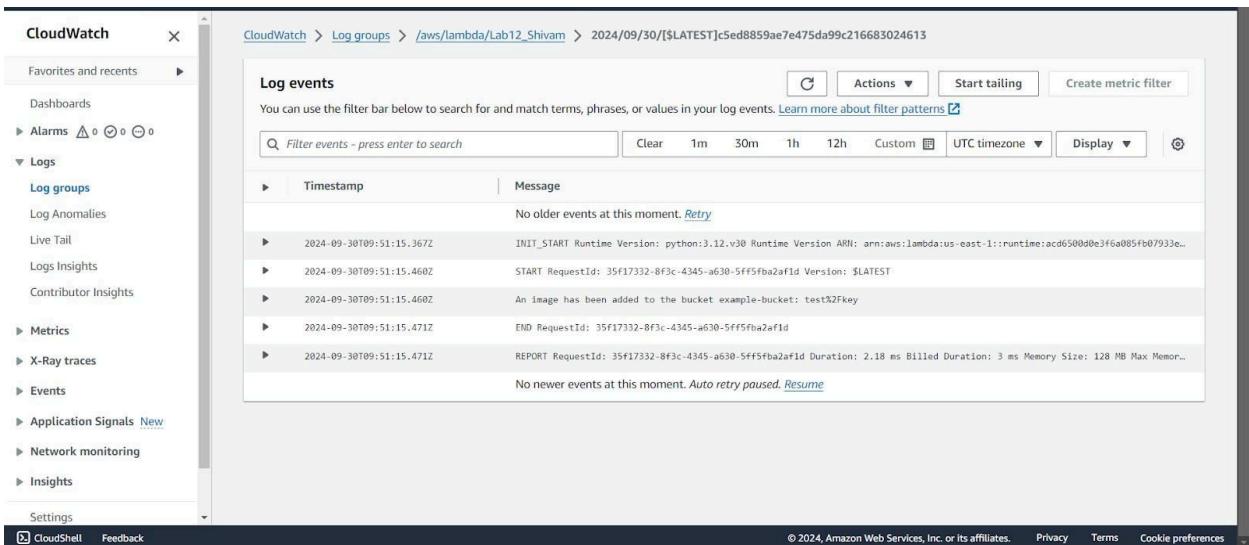
Click on the CloudWatch:

Click on the logs:

Click on the log group:



Scrolled down and click on the log stream:



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's important to choose the S3 Put template for the event; otherwise, the code will give an error. The function was triggered correctly when files were uploaded to S3, showing that Lambda's event-driven design works well. This experiment showed how Lambda can respond to S3 events and how to fix common problems with the event setup.