

## Experiment No. 1A

### Static Hosting

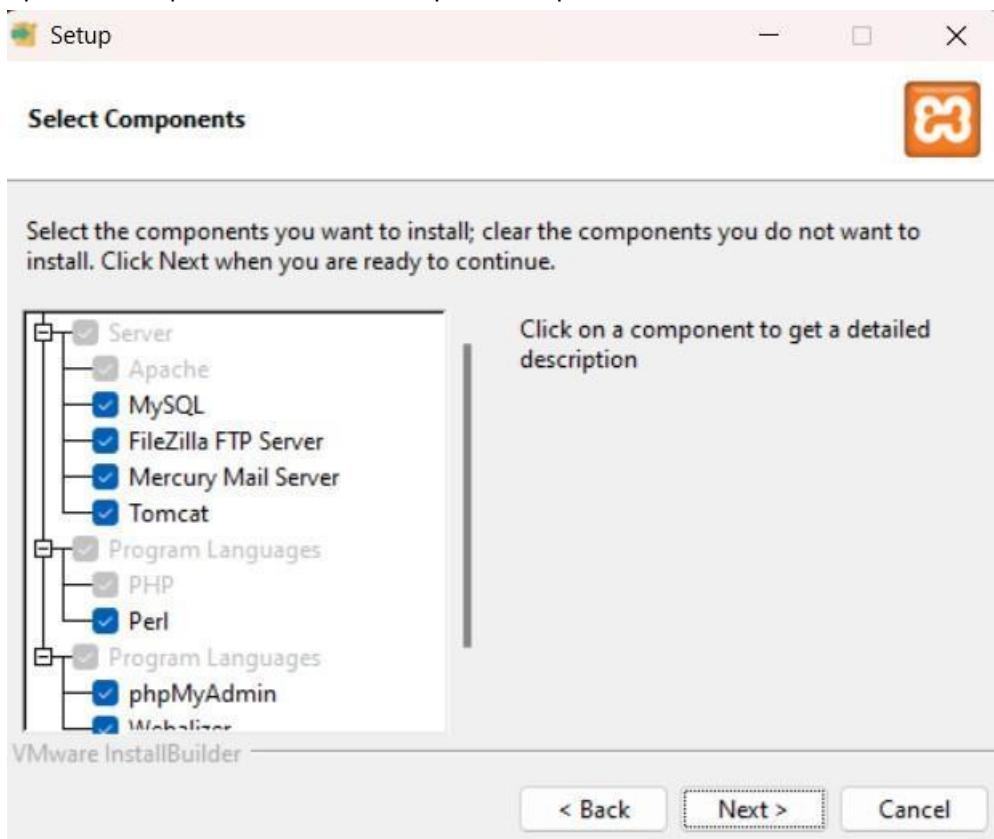
#### On local server (XAMPP)

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

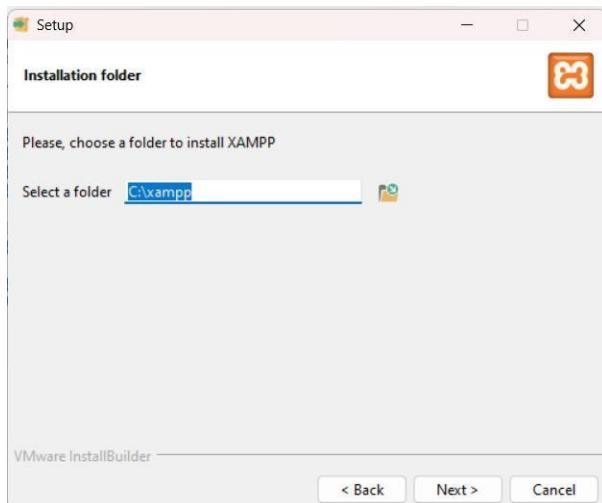
- 1) Select your OS. It will automatically start downloading.



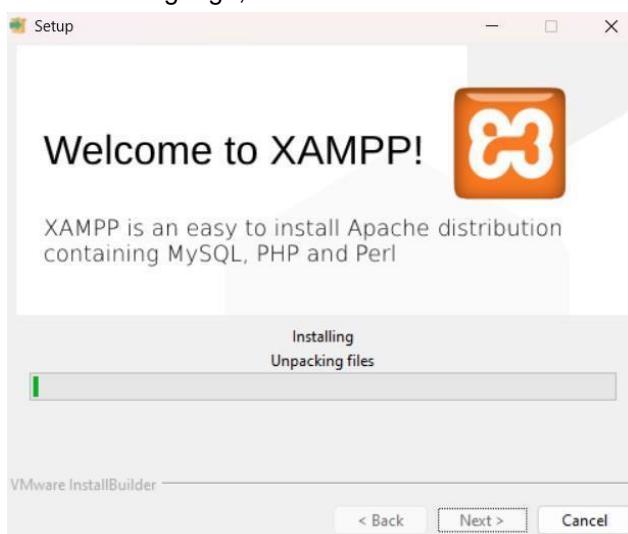
- 2) Open the setup file. Select all the required components and click next



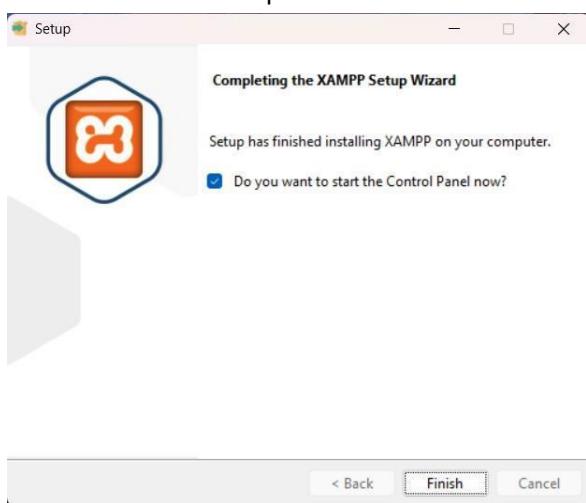
- 3) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



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



- 5) The installation is complete. Click Finish



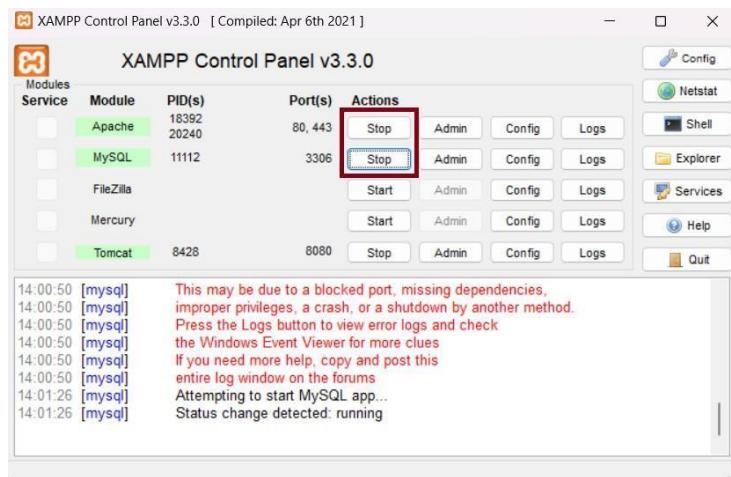
**Step 2:** Setup a file that is to be hosted on the server. Make sure the file has extension .php



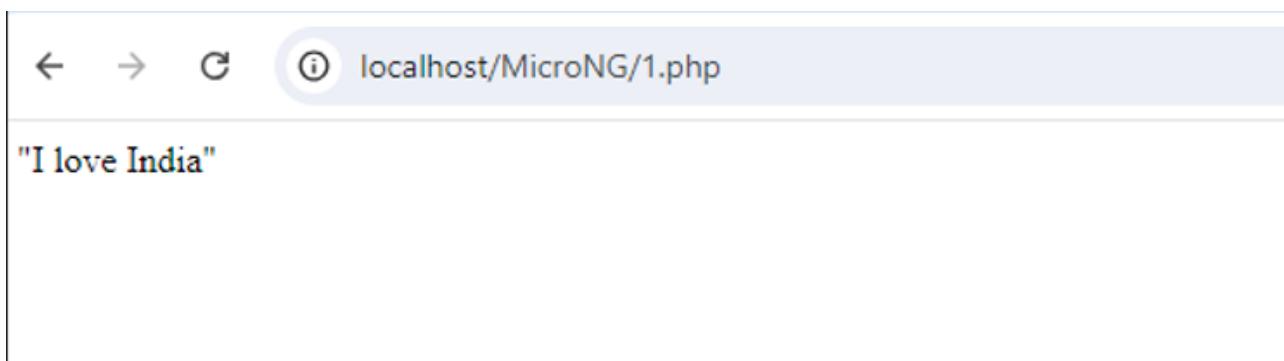
**Step 3:** Go to the directory where XAMPP was installed. Go to **htdocs** folder. Place your folder in this directory.

| Name         | Date modified    | Type              | Size  |
|--------------|------------------|-------------------|-------|
| dashboard    | 06-08-2024 20:42 | File folder       |       |
| img          | 06-08-2024 20:42 | File folder       |       |
| webalizer    | 06-08-2024 20:42 | File folder       |       |
| xampp        | 06-08-2024 22:44 | File folder       |       |
| applications | 15-06-2022 21:37 | Chrome HTML Do... | 4 KB  |
| bitnami      | 15-06-2022 21:37 | CSS Source File   | 1 KB  |
| favicon.ico  | 16-07-2015 21:02 | ICO File          | 31 KB |
| index        | 16-07-2015 21:02 | PHP Source File   | 1 KB  |
| 1            | 06-08-2024 22:48 | PHP Source File   | 1 KB  |
| text         | 06-08-2024 22:23 | PHP Source File   | 1 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/YOUR_FILENAME.php`. This will open your website on your browser.



## AWS S3

The screenshot shows the AWS Management Console with the 'Services' tab selected. The left sidebar has a tree view with categories like Lambda, Containers, Storage, and Database. Under Storage, 'S3' is selected and expanded, showing options like S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup, and AWS Elastic Disaster Recovery. The main content area is titled 'AWS S3' and contains sections for 'Create a bucket', 'How it works', and 'Pricing'. The top navigation bar includes a search bar, a help icon, and account information for 'N. Virginia'.

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

The screenshot shows the 'Amazon S3' service landing page. The title is 'Amazon S3' and the subtitle is 'Store and retrieve any amount of data from anywhere'. A subtext explains that Amazon S3 is an object storage service. On the right, there's a 'Create a bucket' button and a 'Pricing' section. Below the main title, there's a 'How it works' section with a video thumbnail titled 'Introduction to Amazon S3' and a 'Copy link' button.

## Step 2: Click on Create Bucket

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. The 'General configuration' section is visible, showing the AWS Region set to 'US East (N. Virginia) us-east-1'. The 'Bucket type' dropdown is set to 'General purpose'. A bucket name 'myawsbucket' is entered in the 'Bucket name' field. The 'Copy settings from existing bucket - optional' section is present but empty.

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

The screenshot shows the 'Create bucket' configuration page again, but with a different bucket name. The 'Bucket name' field now contains 'abhinav215'. The rest of the configuration remains the same as in Step 2.

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

The screenshot shows the AWS S3 console with the 'Objects' tab selected for the 'abhinav215' bucket. The interface includes a toolbar with actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar at the top allows finding objects by prefix. Below the toolbar is a table header with columns: Name, Type, Last modified, Size, and Storage class. A message indicates 'No objects' found in the bucket.

The screenshot shows the AWS S3 console with the 'Properties' tab selected for the 'abhinav215' bucket. The 'Bucket overview' section displays basic information: AWS Region (US East (N. Virginia) us-east-1), Amazon Resource Name (ARN) (arn:aws:s3:::abhinav215), and Creation date (August 7, 2024, 09:22:15 (UTC+05:30)). The 'Bucket Versioning' section is shown below, with a note about enabling versioning for multiple variants of an object. The bottom navigation bar includes CloudShell, Feedback, and links to AWS terms and conditions.

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

The screenshot shows the AWS S3 bucket properties page for a bucket named 'statichosting27'. The 'Static website hosting' section is highlighted, showing the status as 'Disabled'. There is an 'Edit' button next to the status.

**Step 6:** Enable static website hosting, in Index document, write the name of your document and in error document, give name as 404.html. Save your changes.

The screenshot shows the 'Edit static website hosting' configuration page for the same bucket. The 'Static website hosting' section is enabled. Under 'Hosting type', 'Host a static website' is selected. The 'Index document' field contains 'index.html' and the 'Error document - optional' field contains '404.html'. A note about making content publicly readable is visible in the background.

**Step 7:** Go to Objects tab and click on upload file.

The screenshot shows the AWS S3 console with the 'Objects' tab selected. At the top, there are several buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. The 'Upload' button is highlighted with a red box. Below these buttons is a search bar labeled 'Find objects by prefix'. Underneath is a table header with columns: Name, Type, Last modified, Size, and Storage class. A message 'No objects' is displayed, followed by a note: 'You don't have any objects in this bucket.' At the bottom right of the table area, the 'Upload' button is again highlighted with a red box.

**Step 8:** 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 console with the 'Add files' step. In the 'Files and folders' section, there is one item: 'index.html' (Total, 285.0 B). Below this is a 'Destination' section with the destination set to 's3://abhinav215'. At the bottom of the page, there is a large 'Upload' button.

**Step 9:** 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 console with the 'Properties' tab selected. In the 'Static website hosting' section, the 'Enabled' status is shown. Below it, the 'Bucket website endpoint' is listed as 'http://abhinav215.s3-website-us-east-1.amazonaws.com'. An 'Edit' button is located to the right of the 'Static website hosting' section.

**Step 10:** 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

The screenshot shows a browser window with the URL `abhinav215.s3-website-us-east-1.amazonaws.com`. The page displays a 403 Forbidden error. The error details are as follows:

- Code: AccessDenied
- Message: Access Denied
- RequestId: 71H7WZK6HHA1JVQC
- HostId: ZiWmGRnGSh+AZp7nuQk2wsUAxzeQTiokjTXcmcFkysnVK3RXuO1nkTccqRIG14dXn1sm6JmPRfM=

**Step 11:** Uncheck the Block all public access checkbox and click on save changes

The screenshot shows the AWS S3 console under the 'Edit Block public access (bucket settings)' section. The 'Block all public access' checkbox is currently unchecked. Below it, there are four detailed options, each with its own checkbox:

- Block public access to buckets and objects granted through new access control lists (ACLs)**: Describes blocking new ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**: Describes ignoring all existing ACLs.
- Block public access to buckets and objects granted through new public bucket or access point policies**: Describes blocking new policies.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: Describes blocking cross-account access.

**Step 12:** Scroll down to bucket policy and click edit

The screenshot shows the 'Bucket policy' section of the AWS S3 console. A red box highlights the 'Edit' button in the top right corner. The main area displays the message: 'No policy to display.'

### Step 13:

```
{  
  "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 interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and account information ('N. Virginia' and 'voclabs/user3400217=\_ABHINAV\_SWAMINATHAN @ 9516-5874-3623'). Below the navigation is a main content area with a title 'Bucket policy' and a sub-section 'Policy'. The policy JSON code is pasted into the textarea, starting with line 1. To the right of the policy editor, there's a sidebar with a heading 'Edit statement' and a button 'Select a statement'. Below that, it says 'Select an existing statement in the policy or add a new statement.' and a button '+ Add new statement'. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information ('© 2024, Amazon Web Services, Inc. or its affiliates.'), along with 'Privacy', 'Terms', and 'Cookie preferences'.

### Step 14: Now reload the website. You can see your website

The screenshot shows a web browser window displaying a simple HTML page. The address bar shows the URL 'abhinav215.s3-website-us-east-1.amazonaws.com'. The page content consists of a large bold heading 'Hello, World!' and a smaller text 'Welcome to my first HTML page.'

Name: Abhinav Swaminathan

D15C

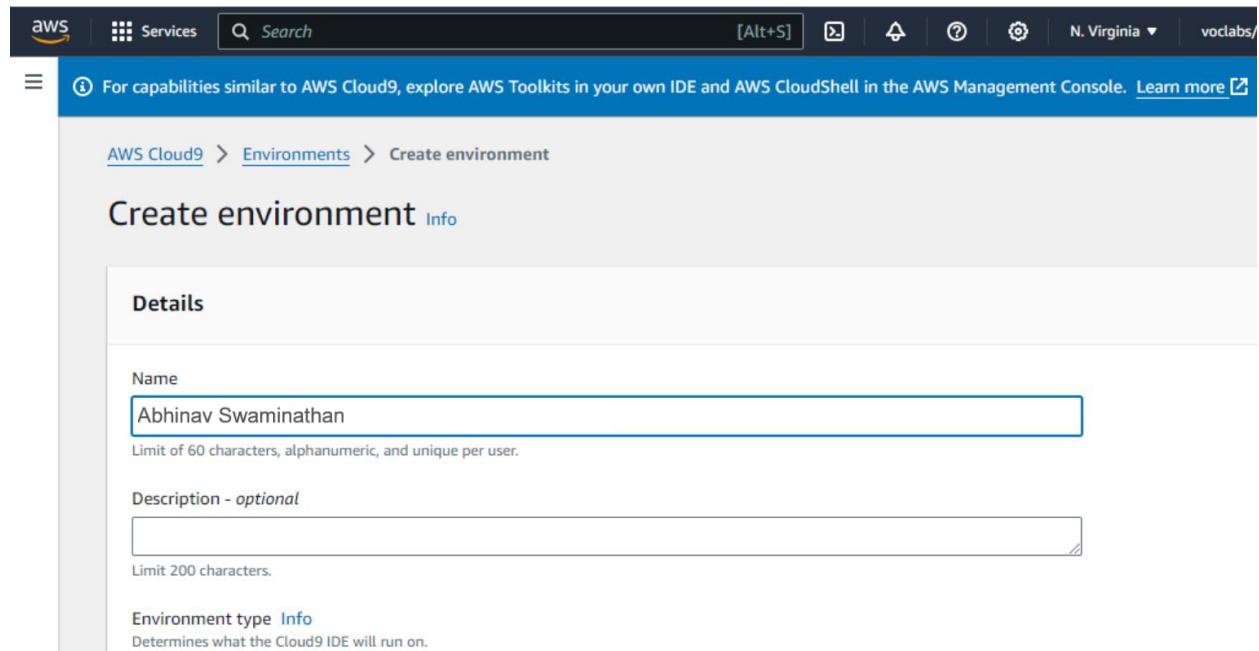
Roll No:01

## Experiment No. 1B

Open the AWS account and search for Cloud9. Click on create environment.



Enter the name and other required configuration for creating an environment. In network settings, using the AWS system manager gives an error while creating the environment. It states there was an error creating IAM resources needed for SSM.



## Use the Secure Shell option in Network settings

The screenshot shows the 'Network settings' section of the AWS Cloud9 configuration. It includes options for 'AWS Systems Manager (SSM)' and 'Secure Shell (SSH)'. The 'Secure Shell (SSH)' option is selected. Below this, there's a 'Tags - optional' section and a note about IAM resource creation.

**Connection**  
How your environment is accessed.

AWS Systems Manager (SSM)  
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

► VPC settings Info

► Tags - optional Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

Once the configuration is complete, click on create environment to create a Cloud9 environment.

The screenshot shows the 'Environments' list in the AWS Cloud9 interface. It displays one environment named 'Abhinav Swaminathan' which is an EC2 instance using Secure Shell (SSH) connection. The environment was successfully created by Shiven Bansal.

AWS Cloud9

Successfully created Shiven Bansal. To get the most out of your environment, see Best practices for using AWS Cloud9

For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. Learn more

My environments

Shared with me

All account environments

Documentation

Environments (1)

Name Cloud9 IDE Environment type Connection Permission Owner ARN

| Name                | Cloud9 IDE | Environment type | Connection         | Permission | Owner ARN  |
|---------------------|------------|------------------|--------------------|------------|--|
| Abhinav Swaminathan | Open       | EC2 instance     | Secure Shell (SSH) | Owner      | arn:aws:sts::583784900342:assumed-role/voclabs/user3397511=Abhinav Swaminathan |

Click on the environment name to open the created Cloud9 Environment.

The screenshot shows the AWS Cloud9 development environment. It features a terminal window with a bash prompt, a code editor with a 'README.md' file, and a 'Getting started' sidebar. The main area displays the 'AWS Cloud9' welcome message: 'Welcome to your development environment'.

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl+P)

AbhinavCLD9 - /

Welcome

Developer Tools

AWS Cloud9

Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

Toolkit for AWS Cloud9

Getting started

Create File

Upload Files...

bash - ip-172-31-33-21.ex Immediate voclabs:/environment \$

Open the aws account and search for IAM service. Then go to users tab and click on create user to create a new user.

The screenshot shows the AWS IAM 'Users' page. At the top, there is a breadcrumb navigation: IAM > Users. Below the header, it says 'Users (0) Info'. A note states: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' There is a 'Create user' button in the top right corner. A search bar is present. The main table has columns: User name, Path, Group, Last activity, MFA, and Password age. The table displays the message 'No resources to display'.

Give the user name, select the “provide user access” checkbox. Also select the option “I want to create an IAM user”. Otherwise we will have to enable the Identity center and specify a user there.

The screenshot shows the 'Specify user details' step in the 'Create user' wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Specify user details' and contains a 'User details' section. It shows a 'User name' field with 'Abhinav' entered. A note below says: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)' and includes a link to 'AWS Management Console'. A checked checkbox 'Provide user access to the AWS Management Console - optional' is followed by a note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A large callout box at the bottom asks 'Are you providing console access to a person?' with two options: 'Specify a user in Identity Center - Recommended' (selected) and 'Specify a user in IAM - Recommended'.

- Provide user access to the AWS Management Console - *optional*  
 If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).

### Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password

Autogenerated password

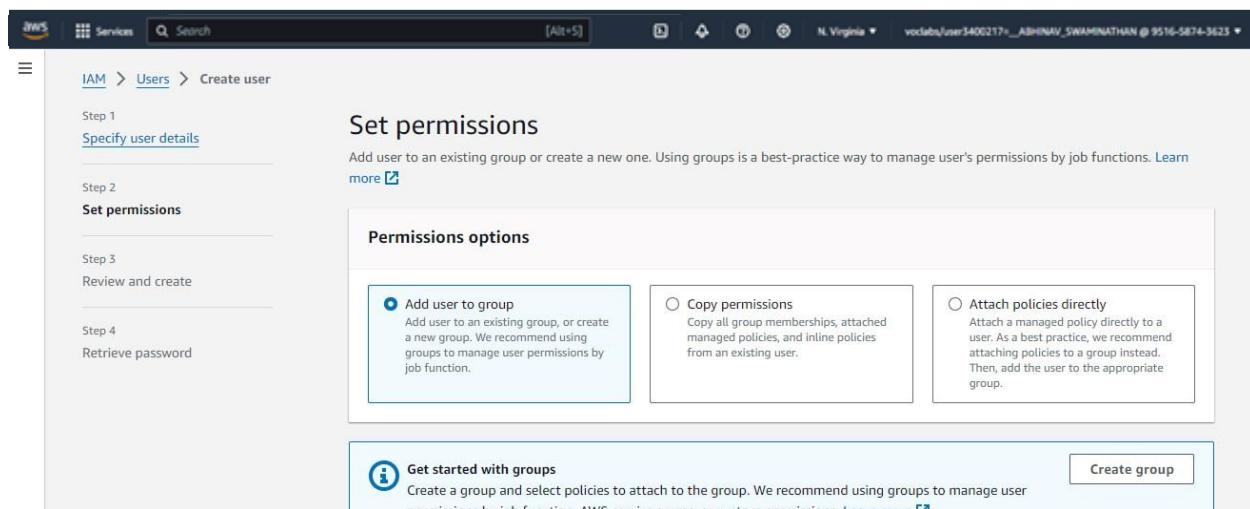
You can view the password after you create the user.

Custom password

Enter a custom password for the user.

\*\*\*\*\*

- Must be at least 8 characters long



The screenshot shows the AWS IAM 'Create user' wizard at the 'Set permissions' step. The left sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions, currently selected), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area has a title 'Set permissions' with a sub-instruction: 'Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions.' Below this is a 'Permissions options' section with three choices:

- Add user to group: 'Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.'
- Copy permissions: 'Copy all group memberships, attached managed policies, and inline policies from an existing user.'
- Attach policies directly: 'Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.'

At the bottom, there is a 'Get started with groups' callout: 'Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions.' A 'Create group' button is also visible.

Next click on add user to group. If you do not have an existing group, select create group. Then Give the group name and policies if required and create a group.

### Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name  
Enter a meaningful name to identify this group.  
  
Maximum 128 characters. Use alphanumeric and '+,-,@,\_' characters.

#### Permissions policies (947)

Filter by Type  
 All ty... ▾

| <input type="checkbox"/> | Policy name                         | Type            | Used as | Description                               |
|--------------------------|-------------------------------------|-----------------|---------|---|
| <input type="checkbox"/> | <a href="#">AdministratorAccess</a> | AWS managed ... | None    | Provides full access to AWS services.     |
| <input type="checkbox"/> | <a href="#">AdministratorAcc...</a> | AWS managed     | None    | Grants account administrative permission. |
| <input type="checkbox"/> | <a href="#">AdministratorAcc...</a> | AWS managed     | None    | Grants account administrative permission. |

[Cancel](#) [Create user group](#)

Once the group is created, select the group in which the user should be added.

Step 1  
[Specify user details](#)

Step 2  
[Set permissions](#)

Step 3  
**Review and create**

Step 4  
[Retrieve password](#)

### Review and create

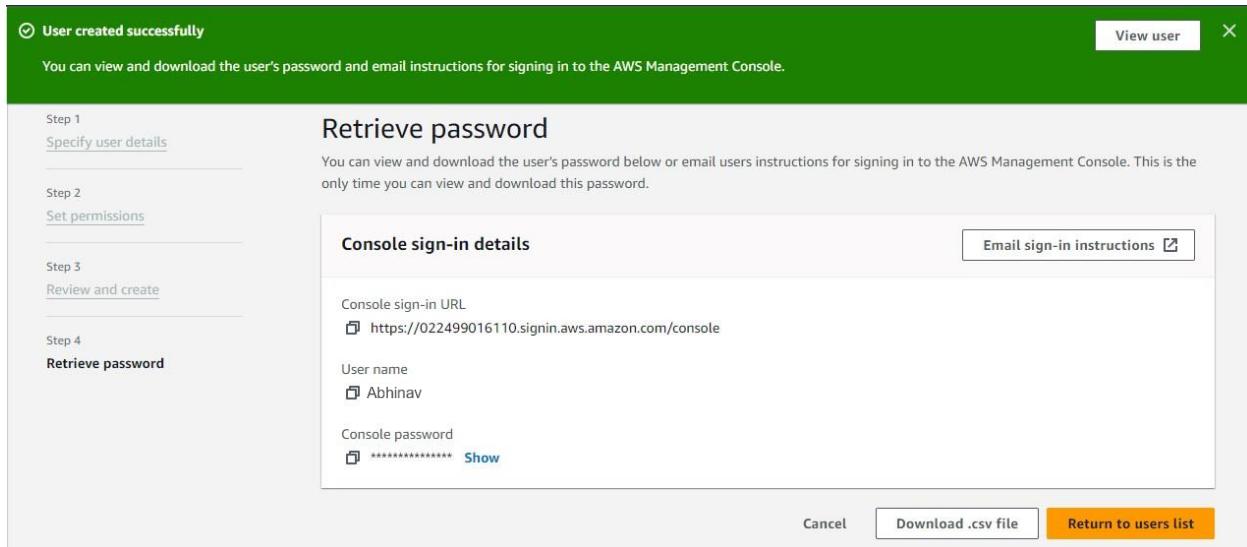
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

| User details         |  |                              |
|----------------------|--|------------------------------|
| User name<br>Abhinav | Console password type<br>Custom password | Require password reset<br>No |

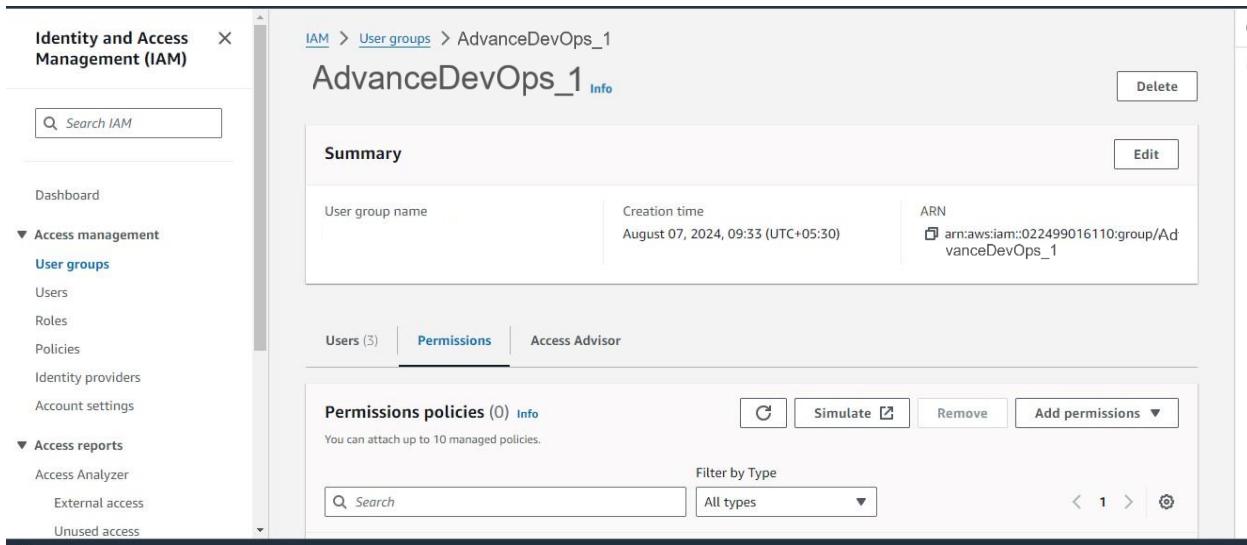
### Permissions summary

| Name                            | Type  | Used as           |
|---------------------------------|-------|-------------------|
| <a href="#">AdvanceDevOps_1</a> | Group | Permissions group |
| <a href="#">AdvanceDevOps_2</a> | Group | Permissions group |
| <a href="#">AdvDevOpsLab_3</a>  | Group | Permissions group |

Recheck all the configuration and details of the user and click on create user. Then you will this page.



After creation of the user, go to the user groups tab and select the group in which the user has been added. Navigate to the permissions tab. Click on add permissions and attach policy.



Search for the “AWSCloud9EnvironmentMember” policy and attach it.

The screenshot shows the 'Add permissions' dialog for the 'AdvanceDevOps\_1' user group. It displays two sections: 'Current permissions policies (0)' and 'Other permission policies (945)'. A search bar at the top right is set to 'cloud9'. In the 'Other permission policies' section, the 'AWSCloud9EnvironmentMember' policy is selected and highlighted with a blue border. At the bottom right of the dialog are 'Cancel' and 'Attach policies' buttons.

Attach permission policies to AdvanceDevOps\_1

▶ Current permissions policies (0)

Other permission policies (945)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter by Type

| Policy name                                | Type                       | Used as | Description                                |
|--|----------------------------|---------|--|
| <a href="#">AdministratorAccess</a>        | AWS managed - job function | None    | Provides full access to AWS services an... |
| <a href="#">AdministratorAccess-Amp...</a> | AWS managed                | None    | Grants account administrative permis...    |
| <a href="#">AdministratorAccess-AWS...</a> | AWS managed                | None    | Grants account administrative permis...    |
| <a href="#">AlexaForBusinessDeviceS...</a> | AWS managed                | None    | Provide device setup access to AlexaFo...  |

▶ Current permissions policies (0)

Other permission policies (1/945)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter by Type

| Policy name                              | Type        | Used as | Description                                 |
|--|-------------|---------|---|
| <a href="#">AWSCloud9Administrator</a>   | AWS managed | None    | Provides administrator access to AWS ...    |
| <a href="#">AWSCloud9Environment...</a>  | AWS managed | None    | Provides the ability to be invited into ... |
| <a href="#">AWSCloud9SSMInstanceP...</a> | AWS managed | None    | This policy will be used to attach a rol... |
| <a href="#">AWSCloud9User</a>            | AWS managed | None    | Provides permission to create AWS Clo...    |

Cancel Attach policies

The screenshot shows the 'Policies attached to this user group' summary page for the 'AdvanceDevOps\_3\_21\_9' user group. It includes a 'Summary' card with details like User group name, Creation time, and ARN. Below this is a 'Permissions' tab showing one attached policy: 'AWSCloud9EnvironmentMember'. The 'Add permissions' button is visible at the top right of this section.

Identity and Access Management (IAM)

Policies attached to this user group.

Summary

|                      |                                    |   |
|----------------------|------------------------------------|---|
| User group name      | Creation time                      | ARN   |
| AdvanceDevOps_3_21_9 | August 07, 2024, 09:33 (UTC+05:30) | arn:aws:iam:022499016110:group/AdvanceDevOps_3_21_9 |

Users (3) Permissions Access Advisor

Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type

| Policy name                               | Type        | Attached entities |
|---|-------------|-------------------|
| <a href="#">AWSCloud9EnvironmentMe...</a> | AWS managed | 3                 |

Add permissions

Name: Abhinav Swaminathan

D15C

Roll No: 01

## Experiment No: 2

### Step1 :- Creation of role:

1. Login to your AWS account and search for IAM

The screenshot shows the AWS CloudSearch interface with a search bar at the top containing 'IAM'. Below the search bar, there are two sections: 'Services' and 'Features'. The 'Services' section contains 173 items, with 'IAM' highlighted in blue and showing a star icon. Other services listed include IAM Identity Center, AWS App Mesh, and Amazon Machine Learning. The 'Features' section contains 446 items, with 'See all 446 results' link.

2. Then go into the role section and click on create role.

The screenshot shows the AWS IAM Roles list page. On the left, there is a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main area displays a table of roles with 14 entries. The columns are 'Role name', 'Trusted entities', and 'Last activity'. The roles listed include various AWS services like ec2, elasticbeanstalk, codepipeline, autoscaling, support, and trustedadvisor, along with some custom roles named after the user's name.

| Role name                                       | Trusted entities                             | Last activity  |
|---|--|----------------|
| aws-elasticbeanstalk-ec2-role                   | AWS Service: ec2                             | 17 minutes ago |
| aws-elasticbeanstalk-service-role               | AWS Service: elasticbeanstalk                | 20 minutes ago |
| AWSCodePipelineServiceRole-us-east-1-pipeline   | AWS Service: codepipeline                    | Yesterday      |
| AWSCodePipelineServiceRole-us-east-1-sadneya    | AWS Service: codepipeline                    | 2 days ago     |
| AWSCodePipelineServiceRole-us-east-1-sadneya_46 | AWS Service: codepipeline                    | Yesterday      |
| AWSCodePipelineServiceRole-us-east-1-sadneya46  | AWS Service: codepipeline                    | -              |
| AWSCodePipelineServiceRole-us-east-1-sadneya46  | AWS Service: autoscaling (Service-Linker)    | 23 minutes ago |
| AWSServiceRoleForAutoScaling                    | AWS Service: support (Service-Linker)        | -              |
| AWSServiceRoleForSupport                        | AWS Service: trustedadvisor (Service-Linker) | -              |
| AWSServiceRoleForTrustedAdvisor                 | AWS Service: codebuild                       | -              |
| codebuild-sadneya46-service-role                | AWS Service: codebuild                       | -              |
| codebuild-sampleweb-service-role                | AWS Service: codebuild                       | -              |

3. Then select a trusted entity as AWS service.

The screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. On the left, a sidebar lists 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The main area is titled 'Trusted entity type' and contains five options:

- AWS service**: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**: Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**: Create a custom trust policy to enable others to perform actions in this account.

4. Select use case as EC2.

The screenshot shows the 'Use case' step of the IAM Role creation wizard. It displays a single option under 'Service or use case': 'EC2'. Below it, a list of use cases is shown, with 'EC2' selected:

- EC2**: Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**: Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**: Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**: Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**: Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**: Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**: Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.

The screenshot shows the 'Permissions policy summary' and 'Step 3: Add tags' steps of the IAM Role creation wizard.

**Permissions policy summary:**

| Policy name                                     | Type        | Attached as        |
|---|-------------|--------------------|
| <a href="#">AWS-ElasticBeanstalk-WebTier</a>    | AWS managed | Permissions policy |
| <a href="#">AWS-ElasticBeanstalk-WorkerTier</a> | AWS managed | Permissions policy |

**Step 3: Add tags:**

Add tags - optional [Info](#)  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add new tag](#)  
You can add up to 50 more tags.

Cancel Previous **Create role**

## 6. Give a name to Role.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. On the left, a sidebar lists steps: 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The main area is titled 'Role details' and contains fields for 'Role name' (set to 'elasticbeanstalk-abhinav') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf'). Below these fields are 'Step 1: Select trusted entities' and 'Trust policy' tabs.

## 7. Then the role gets created

The screenshot shows the 'elasticbeanstalk-abhinav' role in the IAM console. The left sidebar shows navigation options like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access Analyzer, External access, Unused access). The main panel displays the 'Summary' of the role, including its ARN (arn:aws:iam::851725480355:role/aws-elasticbeanstalk-abhinav) and Instance profile ARN (arn:aws:iam::851725480355:instance-profile/aws-elasticbeanstalk-abhinav). Below the summary are tabs for 'Permissions', 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. Under 'Permissions', it shows 'Permissions policies (2)' and a 'Add permissions' button.

## Step 2 :- Creation Elastic Beanstalk Environment

1. search for Elastic Beanstalk in the search box.

The screenshot shows the AWS search interface with the query 'elastiCache'. The results are categorized into 'Services' and 'Features'. Under 'Services', 'ElastiCache' is listed as 'In-Memory Cache'. Below it are 'Elastic Transcoder' (Easy-to-Use Scalable Media Transcoding) and 'Elastic Beanstalk' (Run and Manage Web Apps). Under 'Features', 'Elastic Container Service' is listed as 'Highly secure, reliable, and scalable way to run containers'. A link 'See all 12 results' is also visible.

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

**Application information** Info

Application name  
abhinav

Maximum length of 100 characters.

► Application tags (optional)

**Environment information** Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name  
abhinav-env

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

3. Select platform as PHP.

**Platform Info**

**Platform type**

**Managed platform**  
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#) 

**Custom platform**  
Platforms created and owned by you. This option is unavailable if you have no platforms.

**Platform**

PHP 

**Platform branch**

PHP 8.3 running on 64bit Amazon Linux 2023 

**Platform version**

4.3.1 (Recommended) 

4. After clicking on next you need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created in 1st part. Select role, then select key you have created then profile will be automatically selected according to role. then click on create application by keeping all the remaining settings as it is.

Step 3 - optional  
[Set up networking, database, and tags](#)

Step 4 - optional  
[Configure instance traffic and scaling](#)

Step 5 - optional  
[Configure updates, monitoring, and logging](#)

Step 6  
[Review](#)

**Service role**

Create and use new service role  
 Use an existing service role  
Existing service roles  
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.  
aws-elasticbeanstalk-service-role  

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

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.  
aws-elasticbeanstalk-abhinav  

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) **Next**

Keep Set up networking, database and tags, Configure instance traffic and scaling, Configure updates, monitoring and logging all these default.

5. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with 'Applications', 'Environments', and 'Change history'. Under 'Application: sadneya123', there are 'Application versions' and 'Saved configurations'. Under 'Environment: Sadneya123-env', there are 'Go to environment', 'Configuration', 'Events', 'Health', 'Logs', and 'Monitoring'. The main area displays the 'Environment overview' for 'abhinav-env'. It shows 'Health' as 'Ok', 'Domain' as 'abhinav215-env.eba-6w7emmur.us-east-1.elasticbeanstalk.com', 'Environment ID' as 'e-vw23gecggs', and 'Application name' as 'Abhinav'. To the right, there's a 'Platform' section showing 'Platform' as 'Node.js 20 running on 64bit Amazon Linux 2023/6.1.8', 'Running version' as '—', and 'Platform state' as 'Supported'. At the bottom, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. A green banner at the top says 'Environment successfully launched.'

### Step 3: Get a copy of your sample code

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

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

Go to the repository shared above and simply fork it.



The screenshot shows the GitHub repository 'aws-codepipeline-s3-codedeploy-linux-2.0'. The repository was forked from 'imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0'. It has 1 branch ('master') and 1 tag. The repository has 20 commits. The latest commit was made by 'imoisharma' 3 years ago, updating 'README.md'. Other commits include adding a template, dist folder, and s3 setup scripts. The repository has 0 stars, 0 forks, and 0 releases. It uses the Apache-2.0 license.

## Step 4: Creating a CodePipeline

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

1. Search CodePipeline in the search bar and click on create a new Pipeline.

The screenshot shows the AWS search interface with the query 'code' entered in the search bar. The results are filtered under the 'Services' category, which contains 32 items. The 'CodePipeline' service is listed third, featuring a blue icon of a pipeline, the text 'CodePipeline', and the description 'Release Software using Continuous Delivery'. Other services like Amazon Q Developer, CodeCommit, and AWS Signer are also visible below it.

2. Give a name to your pipeline.

The screenshot shows the 'Pipeline settings' configuration page. It includes fields for 'Pipeline name' (set to 'pipeline-abhinav'), 'Pipeline type' (V2), 'Execution mode' (Queued), 'Service role' (New service role), and 'Role name' (AWSCodePipelineServiceRole-ap-south-1-pipeline-abhinav). A note at the bottom indicates that V1 pipelines are no longer supported.

**Pipeline settings**

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

**Pipeline type**

**Execution mode**  
Choose the execution mode for your pipeline. This determines how the pipeline is run.

**Superseded**  
A more recent execution can overtake an older one. This is the default.

**Queued (Pipeline type V2 required)**  
Executions are processed one by one in the order that they are queued.

**Parallel (Pipeline type V2 required)**  
Executions don't wait for other runs to complete before starting or finishing.

**Service role**

**New service role**  
Create a service role in your account

**Existing service role**  
Choose an existing service role from your account

**Role name**

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

3. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar includes 'Actions', 'Functions', 'Triggers', 'Logs', 'Metrics', and 'Configuration'. The main area displays the function details:

- Name:** lambda-function
- Description:** A function to trigger scheduled events
- Runtime:** Python 3.9
- Memory:** 128 MB
- Timeout:** 3.5 seconds
- Environment:** No environment variables are defined.
- Code** (selected tab):
  - Code source:** AWS Lambda@V2
  - Code location:** GitHub (Version 2)
  - Repository:** https://github.com/w4lyf/aws-codepipeline-s3-codedeploy-linux-2.0
  - Branch:** master
  - Artifact format:** CodePipeline default
  - Deployment group:** No deployment group is selected.
- Test** (tab): Contains a sample event and a 'Test' button.

4. Then select trigger type none.

Trigger

Trigger type  
Choose the trigger type that starts your pipeline.

No filter  
Starts your pipeline on any push and clones the HEAD.

Specify filter  
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

Do not detect changes  
Don't automatically trigger the pipeline.

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

## Step 5: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

### Deploy

#### Deploy provider

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk

#### Region

Asia Pacific (Mumbai)

#### Input artifacts

Choose an input artifact for this action. [Learn more](#)

SourceArtifact

No more than 100 characters

#### Application name

Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

Q abhinav



#### Environment name

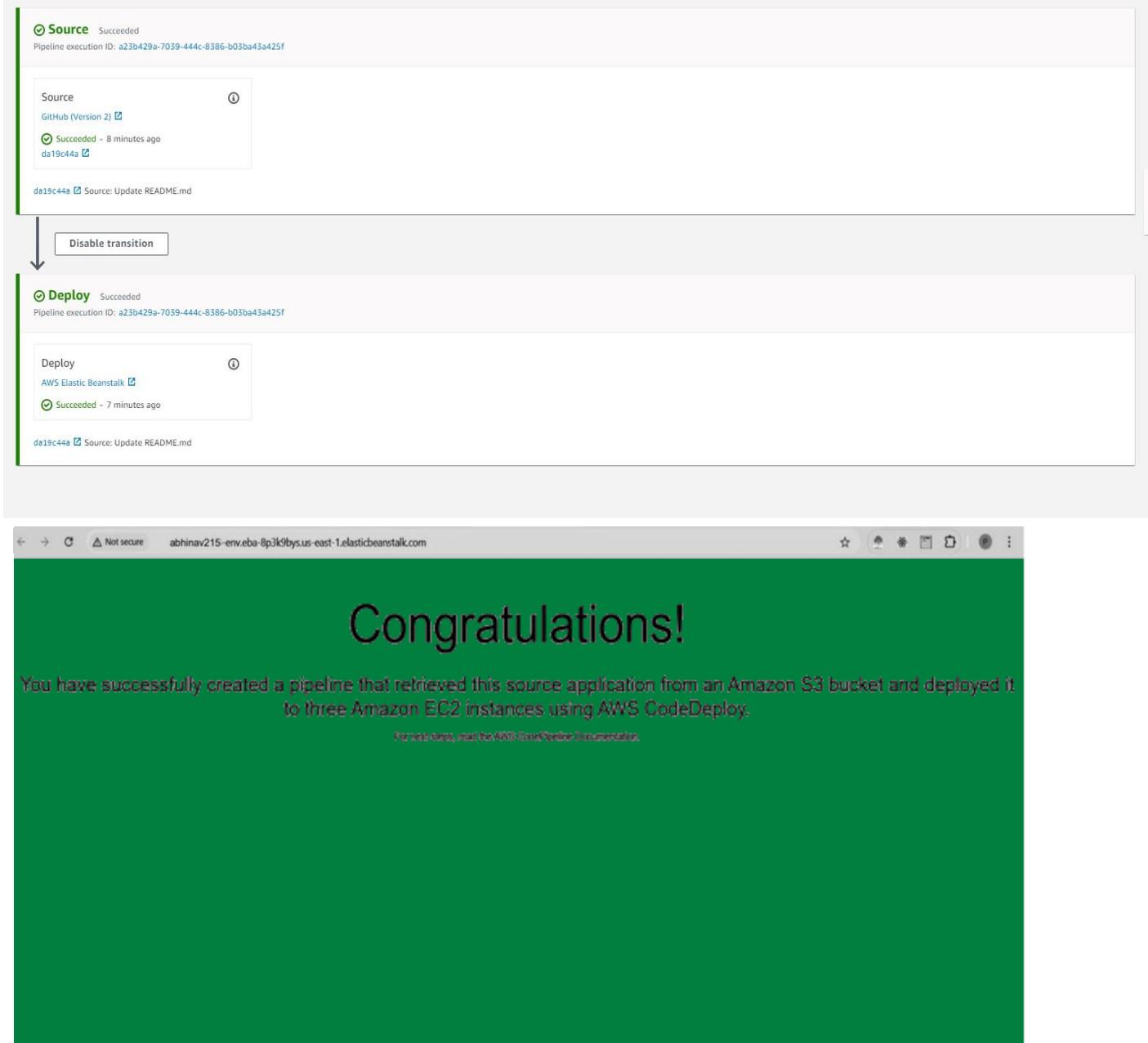
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

Q abhinav-env



Configure automatic rollback on stage failure

2. Then it will give you this result on screen. i.e. deployed successfully.



If you can see this, that means that you successfully created an automated software using CodePipeline.

### Step 6: Committing changes to update app

1. In this we make some changes in the file. Open [github.com](https://github.com) then open the forked repository. Then update the changes in the index.html file and finally commit those changes.

The screenshot shows a GitHub repository page for a sample AWS CodePipeline pipeline. The repository has 1 branch and 0 tags. The master branch is up-to-date with the remote. The commit history shows 20 commits from user imoisharma, starting with adding a template and a dist folder, followed by s3 setup scripts, CONTRIBUTING files, and finally updating index.html. The repository has 1 fork and 0 stars.

**Commit changes**

**Commit message**  
Update index.html

**Extended description**  
Add an optional extended description..

Commit directly to the `master` branch  
 Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

**Cancel** **Commit changes**

2. Then again start the deployment of the pipeline. And check the changes in the website



## Experiment No. 3

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

**Steps:**

1. Create 3 EC2 Ubuntu Instances on AWS. (Name 1 as Master, the other 2 as worker-1 and worker-2)

The screenshot shows the AWS EC2 'Launch an instance' wizard. The first step, 'Name and tags', has 'master' entered in the Name field. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a grid of OS icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. The third step, 'Key pair (login)', shows a dropdown with 'serverkey-01' and a 'Create new key pair' button.

EC2 > Instances > Launch an instance

### Launch an instance Info

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

**Name and tags Info**

Name  
master Add additional tags

**▼ Application and OS Images (Amazon Machine Image) Info**

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li SUS  Browse more AMIs Including AMIs from AWS, Marketplace and

**▼ Key pair (login) Info**

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

Key pair name - *required*  
serverkey-01  Create new key pair

**▼ Network settings** [Info](#)

[Edit](#)

Network | [Info](#)  
vpc-0404d393731afabc3

Subnet | [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)  
Enable  
  
Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

We'll create a new security group called '**launch-wizard-4**' with the following rules:

|   |   |
|---|---|
| <input checked="" type="checkbox"/> Allow SSH traffic from                | Anywhere<br>0.0.0.0/0   |
| <input checked="" type="checkbox"/> Allow HTTPS traffic from the internet | To set up an endpoint, for example when creating a web server |
| <input checked="" type="checkbox"/> Allow HTTP traffic from the internet  | To set up an endpoint, for example when creating a web server |

**⚠** 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. X

| Instances (3) <a href="#">Info</a>   |   |                     |  |  |  |                               |  |  |                 |  |
|--|---|---------------------|--|--|--|-------------------------------|--|--|-----------------|--|
| <a href="#">Launch instances</a> <span style="float: right;">▼</span>  |   |                     |  |  |  |                               |  |  |                 |  |
| <a href="#">Find Instance by attribute or tag (case-sensitive)</a> <span style="float: right;">Running <span style="border: 1px solid #ccc; padding: 0 5px;">▼</span></span> |   |                     |  |  |  |                               |  |  |                 |  |
|  | Name <span style="float: right;">▼</span> | Instance ID         | Instance state <span style="float: right;">▲</span>  | Instance type <span style="float: right;">▼</span> | Status check   | Alarm status                  | Availability Zone <span style="float: right;">▼</span> | Public IPv4 DNS <span style="float: right;">▼</span> | Public IPv4 ... |  |
| <input type="checkbox"/>   | master                                    | i-0a57da166d1f22307 | <span style="color: green;">Running</span> <span style="color: green;">Q</span> <span style="color: green;">Q</span> | t2.micro   | <span style="color: green;">Initializing</span>      | <a href="#">View alarms</a> + | us-east-1c   | ec2-54-165-196-241.co...                             | 54.165.196.24   |  |
| <input type="checkbox"/>   | worker-2                                  | i-09ef6bb0a00befc3c | <span style="color: green;">Running</span> <span style="color: green;">Q</span> <span style="color: green;">Q</span> | t2.micro   | <span style="color: green;">2/2 checks passed</span> | <a href="#">View alarms</a> + | us-east-1c   | ec2-184-72-206-70.co...                              | 184.72.206.70   |  |
| <input type="checkbox"/>   | worker-1                                  | i-04e9c3add3858c21c | <span style="color: green;">Running</span> <span style="color: green;">Q</span> <span style="color: green;">Q</span> | t2.micro   | <span style="color: green;">Initializing</span>      | <a href="#">View alarms</a> + | us-east-1c   | ec2-54-211-147-10.co...                              | 54.211.147.10   |  |

## 2. SSH into all 3 machines

- Give permissions to the current user to the downloaded pem file using -  
**chmod 400 <security\_filename.pem>**

```
abhin@Abhinavz-Acer MINGW64 ~ (master)
$ cd Downloads/

abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ chmod 400 "serverkey-01.pem"

abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ |
```

- ssh into all the three machines using –  
**ssh -i (keyname).pem (username)@(public ipv4 dns address)**  
 where keyname is name of the key you created. (server-01.pem). Other details can be found on the Instance dashboard.

```
abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ ssh -i "serverkey-01.pem" ec2-user@ec2-54-165-196-241.compute-1.amazonaws.com
,
#_
~\_ #####
~~ \##### Amazon Linux 2023
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~,-->
~~ / \
~~ / \
~~ /m/
[ec2-user@ip-172-31-90-103 ~]$
```

### 3. Installation Of Docker on three machines

- **sudo yum install docker -y**

```
[ec2-user@ip-172-31-90-103 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:13:41 ago on Sat Sep 14 03:42:27 2024.
Dependencies resolved.
=====
 Package           Arch      Version          Repository      size
=====
Installing:
 docker            x86_64   25.0.6-1.amzn2023.0.2  amazonlinux   44 M
Installing dependencies:
 containerd         x86_64   1.7.20-1.amzn2023.0.1   amazonlinux   35 M
 iptables-libs     x86_64   1.8.8-3.amzn2023.0.2   amazonlinux   401 k
 iptables-nft      x86_64   1.8.8-3.amzn2023.0.2   amazonlinux   183 k
 libcgroup         x86_64   3.0-1.amzn2023.0.1   amazonlinux   75 k
 libnetfilter_conntrack x86_64   1.0.8-2.amzn2023.0.2  amazonlinux   58 k
 libnfnetworklink x86_64   1.0.1-19.amzn2023.0.2  amazonlinux   30 k
 libnftnl          x86_64   1.2.2-2.amzn2023.0.2  amazonlinux   84 k
 pigz              x86_64   2.5-1.amzn2023.0.3   amazonlinux   83 k
 runc              x86_64   1.1.13-1.amzn2023.0.1  amazonlinux   3.2 M

Transaction summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_2.2 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_6 2.5 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.1.3 MB/s | 58 kB  00:00
(5/10): libnfnetworklink-1.0.1-19.amzn2023.0.2.x86_938 kB/s | 30 kB  00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 1.7 MB/s | 83 kB  00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm 21 kB/s | 3.2 MB  00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64 31 kB/s | 35 MB  00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm 28 kB/s | 44 MB  00:01

Total                                         52 MB/s | 84 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing       :
  Installing     : runc-1.1.13-1.amzn2023.0.1.x86_64
  Installing     : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
  Installing     : pigz-2.5-1.amzn2023.0.3.x86_64
  Installing     : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  Installing     : libnfnetworklink-1.0.1-19.amzn2023.0.2.x86_64
  Installing     : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  Installing     : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  Installing     : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
```

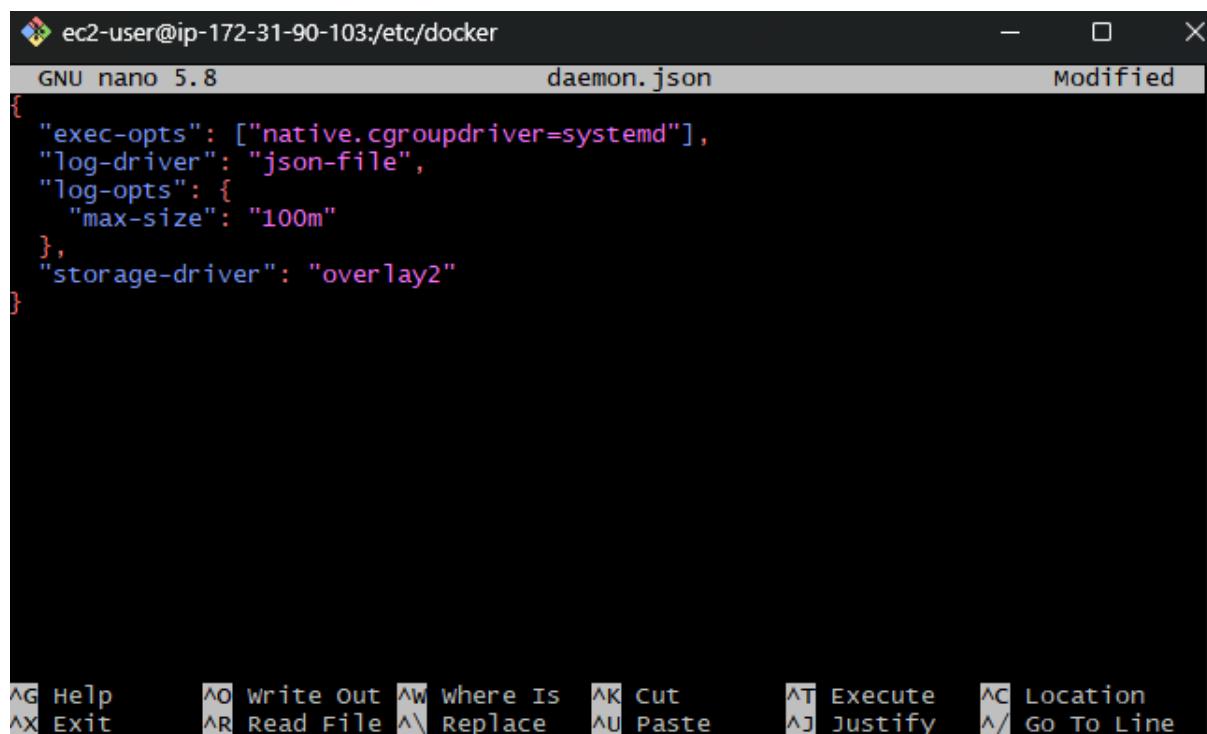
```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64           docker-25.0.6-1.amzn2023.0.2.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-90-103 ~]$
```

- Configure cgroup in a daemon.json  
(this can be done by creating the file and using **nano** text editor)

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

```
[ec2-user@ip-172-31-90-103 docker]$ sudo nano daemon.json
[ec2-user@ip-172-31-90-103 docker]$ |
```



- Enable and start docker and also load the daemon.json

```
sudo systemctl enable docker
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl enable docker
created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-90-103 docker]$
sudo systemctl restart docker
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Check if docker is installed

```
[ec2-user@ip-172-31-90-103 docker]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-90-103 docker]$ |
```

#### 4. Install Kubernetes on all 3 machines

- SELinux needs to be disabled before configuring kubelet

```
sudo setenforce 0
```

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

```
[ec2-user@ip-172-31-90-103 docker]$ sudo setenforce 0
[sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Add Kubernetes using the repo

(this is done by creating **kubernetes.repo** file in **/etc/yum.repos.d** and configuring it using **nano** editor)

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
[ec2-user@ip-172-31-90-103 docker]$ cd /etc/yum.repos.d/
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo kernel-livepatch.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo nano kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo kernel-livepatch.repo kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

The screenshot shows a terminal window with the title bar "ec2-user@ip-172-31-90-103:/etc/yum.repos.d". The window contains the configuration for the Kubernetes repository:

```
GNU nano 5.8                               kubernetes.repo                         Modified
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

At the bottom of the terminal window, there is a menu bar with the following options: Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

- Update packages list using **sudo yum update**

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum update
Kubernetes                                         125 kB/s | 17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- Install kubelet kubeadm kubectl

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:42 ago on sat Sep 14 04:08:20 2024.
Dependencies resolved.
=====
Package          Arch    Version           Repository      Size
=====
Installing:
  kubelet          x86_64  1.30.5-150500.1.1   kubernetes     17 M
  kubeadm         x86_64  1.30.5-150500.1.1   kubernetes     10 M
  kubectl          x86_64  1.30.5-150500.1.1   kubernetes     10 M
Installing dependencies:
  conntrack-tools x86_64  1.4.6-2.amzn2023.0.2  amazonlinux   208 k
  cri-tools        x86_64  1.30.1-150500.1.1   kubernetes     8.6 M
  kubernetes-cni  x86_64  1.4.0-150500.1.1   kubernetes     6.7 M
  libnetfilter_cthelper x86_64  1.0.0-21.amzn2023.0.2  amazonlinux   24 k
  libnetfilter_cttimeout x86_64  1.0.0-19.amzn2023.0.2  amazonlinux   24 k
  libnetfilter_queue  x86_64  1.0.5-2.amzn2023.0.2  amazonlinux   30 k
Transaction Summary
=====
Install 9 Packages
```

Total download size: 53 M  
 Installed size: 292 M  
 Downloading Packages:

|   |          |        |       |
|---|----------|--------|-------|
| (1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023 | 448 kB/s | 24 kB  | 00:00 |
| (2/9): libnetfilter_cthelper-1.0.0-21.amzn2023. | 409 kB/s | 24 kB  | 00:00 |
| (3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2. | 1.5 MB/s | 30 kB  | 00:00 |
| (4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86 | 1.8 MB/s | 208 kB | 00:00 |
| (5/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm   | 28 MB/s  | 8.6 MB | 00:00 |
| (6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm     | 23 MB/s  | 10 MB  | 00:00 |
| (7/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm     | 18 MB/s  | 10 MB  | 00:00 |
| (8/9): kubelet-1.30.5-150500.1.1.x86_64.rpm     | 37 MB/s  | 17 MB  | 00:00 |
| (9/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.r | 20 MB/s  | 6.7 MB | 00:00 |

Total 56 MB/s | 53 MB 00:00

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

Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- 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
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ . sudo swapoff -a
-bash: sudo: No such file or directory
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo swapoff -a
[ec2-user@ip-172-31-90-103 yum.repos.d]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

## 5. Perform this ONLY on the Master machine Initialize the Kubecluster

```
sudo kubeadm init --podnetwork-cidr=10.244.0.0/16
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo kubeadm init --pod-network-cidr=10
.244.0.0/16 --ignore-preflight-errors=all
I0914 04:12:17.448521 27990 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.30
[init] 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 (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0914 04:12:17.711154 27990 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificatedir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
```

```
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 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

➤ Save the token

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

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

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ 
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

## 6. Perform this ONLY on the worker machines

- Paste the below command on all 2 worker machines

```
sudo yum install iproute-tc -y
sudo systemctl enable kubelet
sudo systemctl restart kubelet
```

- Now use the token from earlier to join into worker instances

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

- **kubectl get nodes** to check whether master and worker nodes are connected successfully

| NAME                          | STATUS | ROLES         | AGE   | VERSION |
|-------------------------------|--------|---------------|-------|---------|
| ip-172-31-90-103.ec2.internal | Ready  | control-plane | 3m21s | v1.30.5 |

### Conclusion:

An EC2 instance was created on AWS Linux, and Docker, Kubernetes, Kubelet, Kubeadm, and Kubectl were installed. Kubernetes was initialized on the master node, which provided a token for connecting the master and worker nodes. On the slave node, iproute was installed, and Kubelet was enabled and restarted. However, there was an issue with joining the slave node to the cluster, resulting in only the master node being listed when running `kubectl get nodes`

## Experiment No. 4

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

**Steps:**

**1. Create an EC2 Linux Instances on AWS.**

The screenshot shows the 'Launch an instance' step in the AWS EC2 wizard. It includes sections for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Key pair (login)'. The 'Name and tags' section has a 'Name' field containing 'abhinav'. The 'Application and OS Images' section shows various AMI options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux, with a search bar and a 'Browse more AMIs' link. The 'Key pair (login)' section shows a 'Key pair name - required' dropdown set to 'serverkey-01' and a 'Create new key pair' button.

EC2 > Instances > Launch an instance

### Launch an instance Info

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

#### Name and tags Info

Name

abhinav

Add additional tags

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

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li

aws macos ubuntu windows redhat suse

Browse more AMIs

Including AMIs from AWS Marketplace

#### ▼ Key pair (login) Info

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

Key pair name - *required*

serverkey-01

Create new key pair

**▼ Network settings** [Info](#)

[Edit](#)

Network | [Info](#)  
vpc-0404d393731afabc3

Subnet | [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)  
Enable  
  
Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#)       [Select existing security group](#)

We'll create a new security group called '**launch-wizard-4**' with the following rules:

|   |   |
|---|---|
| <input checked="" type="checkbox"/> Allow SSH traffic from                | Anywhere<br>0.0.0.0/0   |
| Helps you connect to your instance  |   |
| <input checked="" type="checkbox"/> Allow HTTPS traffic from the internet | To set up an endpoint, for example when creating a web server |
|   |   |
| <input checked="" type="checkbox"/> Allow HTTP traffic from the internet  | To set up an endpoint, for example when creating a web server |
|   |   |

**⚠** 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. X

| <a href="#">Instances (1)</a> <a href="#">Info</a> |         | Last updated<br>less than a minute ago | <a href="#">C</a> | <a href="#">Connect</a> | <a href="#">Instance state ▾</a> | <a href="#">Actions ▾</a> | <a href="#">Launch instances ▾</a> |
|--|---------|--|-------------------|-------------------------|----------------------------------|---------------------------|------------------------------------|
|  |         |  | < 1 >             | ①                       |                                  |                           |                                    |
| <input type="checkbox"/>                           | Name ↴  | Instance ID                            | Instance state    | Running                 | Instance type                    | Status check              | Alarm status                       |
| <input type="checkbox"/>                           | abhinav | i-0a57da166d1f22307                    | Running           | 🕒                       | t2.micro                         | 🕒 2/2 checks passed       | <a href="#">View alarms</a> +      |

2. Then click on Id of that instance then click on connect

The screenshot shows the 'Connect to instance' page for an EC2 instance. At the top, there's a breadcrumb navigation: EC2 > Instances > i-0a57da166d1f22307 > Connect to instance. Below the breadcrumb, the title 'Connect to instance' has an 'Info' link. A note below the title says 'Connect to your instance i-0a57da166d1f22307 (abhinav) using any of these options'. There are four tabs at the top: EC2 Instance Connect (selected), Session Manager, SSH client, and EC2 serial console. A warning box is present, stating: 'Port 22 (SSH) is open to all IPv4 addresses' and explaining that port 22 is currently open to all IPv4 addresses, indicated by 0.0.0.0/0 in the inbound rule in the security group. It suggests restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. A 'Learn more' link is provided. The 'Instance ID' field contains 'i-0a57da166d1f22307 (abhinav)'. The 'Connection Type' section shows two options: 'Connect using EC2 Instance Connect' (selected) and 'Connect using EC2 Instance Connect Endpoint'. The 'Public IPv4 address' field contains '54.165.196.241'. The 'Username' field contains 'ec2-user'. A note in a box says: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.

3. SSH into the created machine instance

- Give permissions to the current user to the downloaded pem file using -  
`chmod 400 <security_filename.pem>`

```
abhin@Abhinavz-Acer MINGW64 ~ (master)
$ cd Downloads/

abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ chmod 400 "serverkey-01.pem"

abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ |
```

- Ssh using –  
**ssh -i (keyname).pem (username)@(public ipv4 dns address)**  
where keyname is name of the key you created. (server-01.pem). Other details can be found on the Instance dashboard.

```
abhin@Abhinavz-Acer MINGW64 ~/Downloads (master)
$ ssh -i "serverkey-01.pem" ec2-user@ec2-54-165-196-241.compute-1.amazonaws.com
,
#_
~\_ #####
~~ \##### Amazon Linux 2023
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~,-->
~~ / \
~~ / \
~~ /m/
[ec2-user@ip-172-31-90-103 ~]$
```

#### 4. Installation Of Docke007

- **sudo yum install docker -y**

```
[ec2-user@ip-172-31-90-103 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:13:41 ago on Sat Sep 14 03:42:27 2024.
Dependencies resolved.
=====
 Package           Arch      Version          Repository    size
=====
Installing:
 docker            x86_64   25.0.6-1.amzn2023.0.2  amazonlinux  44 M
Installing dependencies:
 containerd        x86_64   1.7.20-1.amzn2023.0.1  amazonlinux  35 M
 iptables-libc     x86_64   1.8.8-3.amzn2023.0.2  amazonlinux  401 k
 iptables-nft      x86_64   1.8.8-3.amzn2023.0.2  amazonlinux  183 k
 libcgroup         x86_64   3.0-1.amzn2023.0.1   amazonlinux  75 k
 libnetfilter_conntrack x86_64   1.0.8-2.amzn2023.0.2  amazonlinux  58 k
 libnftnlink       x86_64   1.0.1-19.amzn2023.0.2  amazonlinux  30 k
 libnftnl          x86_64   1.2.2-2.amzn2023.0.2  amazonlinux  84 k
 pigz              x86_64   2.5-1.amzn2023.0.3   amazonlinux  83 k
 runc              x86_64   1.1.13-1.amzn2023.0.1  amazonlinux  3.2 M

Transaction summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_2.2 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_6 2.5 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.1.3 MB/s | 58 kB  00:00
(5/10): libnftnlink-1.0.1-19.amzn2023.0.2.x86_938 kB/s | 30 kB  00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 1.7 MB/s | 83 kB  00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm 21 kB/s | 3.2 MB  00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64 31 kB/s | 35 MB  00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm 28 MB/s | 44 MB  00:01

Total                                         52 MB/s | 84 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing           :
  Installing         : runc-1.1.13-1.amzn2023.0.1.x86_64
  Installing         : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
  Installing         : pigz-2.5-1.amzn2023.0.3.x86_64
  Installing         : libnftnlink-1.2.2-2.amzn2023.0.2.x86_64
  Installing         : libnftnl-1.0.1-19.amzn2023.0.2.x86_64
  Installing         : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  Installing         : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  Installing         : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
```

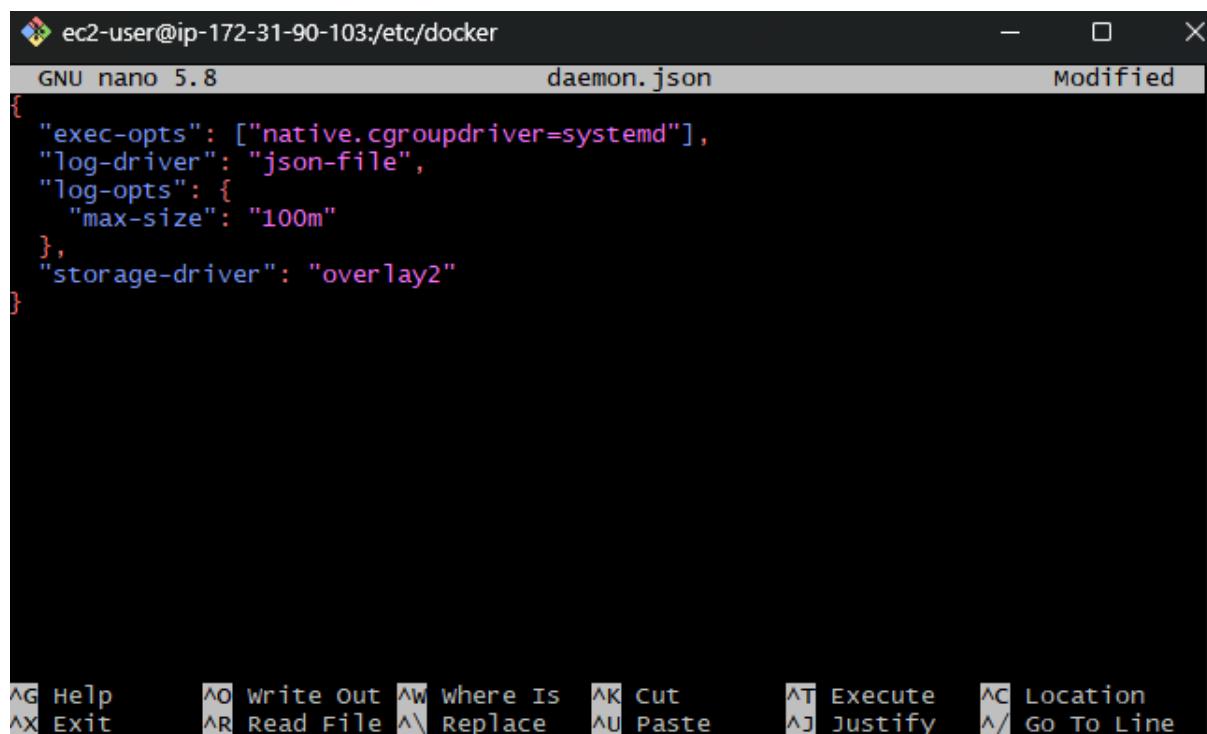
```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64           docker-25.0.6-1.amzn2023.0.2.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-90-103 ~]$
```

- Configure cgroup in a daemon.json  
(this can be done by creating the file and using **nano** text editor)

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

```
[ec2-user@ip-172-31-90-103 docker]$ sudo nano daemon.json
[ec2-user@ip-172-31-90-103 docker]$ |
```



- Enable and start docker and also load the daemon.json

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

```
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl enable docker
[sudo] password for ec2-user:
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-90-103 docker]$
sudo systemctl restart docker
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Check if docker is installed

```
[ec2-user@ip-172-31-90-103 docker]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-90-103 docker]$ |
```

## 5. Install Kubernetes

- SELinux needs to be disabled before configuring kubelet

```
sudo setenforce 0
```

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

```
[ec2-user@ip-172-31-90-103 docker]$ sudo setenforce 0
[sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Add Kubernetes using the repo

(this is done by creating **kubernetes.repo** file in **/etc/yum.repos.d** and configuring it using **nano** editor)

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
[ec2-user@ip-172-31-90-103 docker]$ cd /etc/yum.repos.d/
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo kernel-livepatch.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo nano kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo kernel-livepatch.repo kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

The screenshot shows a terminal window with the title bar "ec2-user@ip-172-31-90-103:/etc/yum.repos.d". The window contains the configuration for the Kubernetes repository:

```
GNU nano 5.8                               kubernetes.repo                         Modified
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

At the bottom of the terminal window, there is a menu bar with the following options: Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

- Update packages list using **sudo yum update**

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum update
Kubernetes                                         125 kB/s | 17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- Install kubelet kubeadm kubectl

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:42 ago on sat Sep 14 04:08:20 2024.
Dependencies resolved.
=====
Package          Arch    Version           Repository      Size
=====
Installing:
  kubelet          x86_64  1.30.5-150500.1.1   kubernetes     17 M
  kubeadm         x86_64  1.30.5-150500.1.1   kubernetes     10 M
  kubectl          x86_64  1.30.5-150500.1.1   kubernetes     10 M
Installing dependencies:
  conntrack-tools x86_64  1.4.6-2.amzn2023.0.2  amazonlinux   208 k
  cri-tools        x86_64  1.30.1-150500.1.1   kubernetes     8.6 M
  kubernetes-cni  x86_64  1.4.0-150500.1.1   kubernetes     6.7 M
  libnetfilter_cthelper x86_64  1.0.0-21.amzn2023.0.2  amazonlinux   24 k
  libnetfilter_cttimeout x86_64  1.0.0-19.amzn2023.0.2  amazonlinux   24 k
  libnetfilter_queue  x86_64  1.0.5-2.amzn2023.0.2  amazonlinux   30 k
Transaction Summary
=====
Install 9 Packages
```

Total download size: 53 M  
 Installed size: 292 M  
 Downloading Packages:

|   |          |        |       |
|---|----------|--------|-------|
| (1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023 | 448 kB/s | 24 kB  | 00:00 |
| (2/9): libnetfilter_cthelper-1.0.0-21.amzn2023. | 409 kB/s | 24 kB  | 00:00 |
| (3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2. | 1.5 MB/s | 30 kB  | 00:00 |
| (4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86 | 1.8 MB/s | 208 kB | 00:00 |
| (5/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm   | 28 MB/s  | 8.6 MB | 00:00 |
| (6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm     | 23 MB/s  | 10 MB  | 00:00 |
| (7/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm     | 18 MB/s  | 10 MB  | 00:00 |
| (8/9): kubelet-1.30.5-150500.1.1.x86_64.rpm     | 37 MB/s  | 17 MB  | 00:00 |
| (9/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.r | 20 MB/s  | 6.7 MB | 00:00 |

Total 56 MB/s | 53 MB 00:00

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

Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- 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
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ . sudo swapoff -a
-bash: sudo: No such file or directory
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo swapoff -a
[ec2-user@ip-172-31-90-103 yum.repos.d]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

## 6. Initialize the Kubecluster

```
sudo kubeadm init --podnetwork-cidr=10.244.0.0/16
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo kubeadm init --pod-network-cidr=10
.244.0.0/16 --ignore-preflight-errors=all
I0914 04:12:17.448521 27990 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.30
[init] 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 (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0914 04:12:17.711154 27990 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificatedir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
```

```
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 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

➤ Save the token

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

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

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ 
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

## 7. Deploy nginx server

- Apply deployment using this following command:

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

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
pod/nginx created
```

- use **kubectl get nodes** to check whether the pod gets created or not

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx    0/1     Pending   0          12s
```

- To convert state from pending to running use following command:

**kubectl describe pod nginx** (This command will help to describe the pods it gives reason for failure as it shows the untolerated taints which need to be untainted.)

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl describe pod nginx
Name:           nginx
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          <none>
Annotations:    <none>
Status:         Pending
IP:
IPs:
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k4lj6 (ro)
      Type:           Projected (a volume that contains injected data from m
ultiple sources)
      TokenExpirationSeconds: 3607
      ConfigMapName:   kube-root-ca.crt
      ConfigMapOptional: <nil>
      DownwardAPI:    true
    QoS Class:      BestEffort
    Node-Selectors: <none>
    Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 3
00s
                                         node.kubernetes.io/unreachable:NoExecute op=Exists for
                                         300s
Events:
  Type     Reason     Age     From           Message
  ----     ----     --     --           -----
  Warning  FailedScheduling  7s     default-scheduler  0/1 nodes are available: 1 no
de(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption:
0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

- check pod status

```
[ec2-user@ip-172-31-90-103 ~]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   1 (6s ago)  90s
```

- mention the port you want to host

```
[ec2-user@ip-172-31-90-103 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

## 8. Verify your deployment

- Open up a new terminal and ssh to your EC2 instance. Then, use this curl command to check if the Nginx server is running. `curl --head http://127.0.0.1:8080` 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.

```
[ec2-user@172-31-90-103 ~]$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Linux)
Date: Sat, 14 Sep 2024 12:31:53 GMT
Content-Type: text/html
Content-Length: 612
Connection: keep-alive
```

## Conclusion:

An AWS EC2 Linux instance was set up, and Docker and Kubernetes were installed. Kubernetes was initialized successfully, and the required commands were executed. Flannel was installed as a networking plugin. Although there was an initial error with the Nginx deployment, it was eventually deployed successfully using the `simple-pod.yml` file and accessed via localhost on port 8080.

Name: Abhinav Swaminathan

D15C

Roll No: 01

## Experiment No. 5

### Installation of Terraform

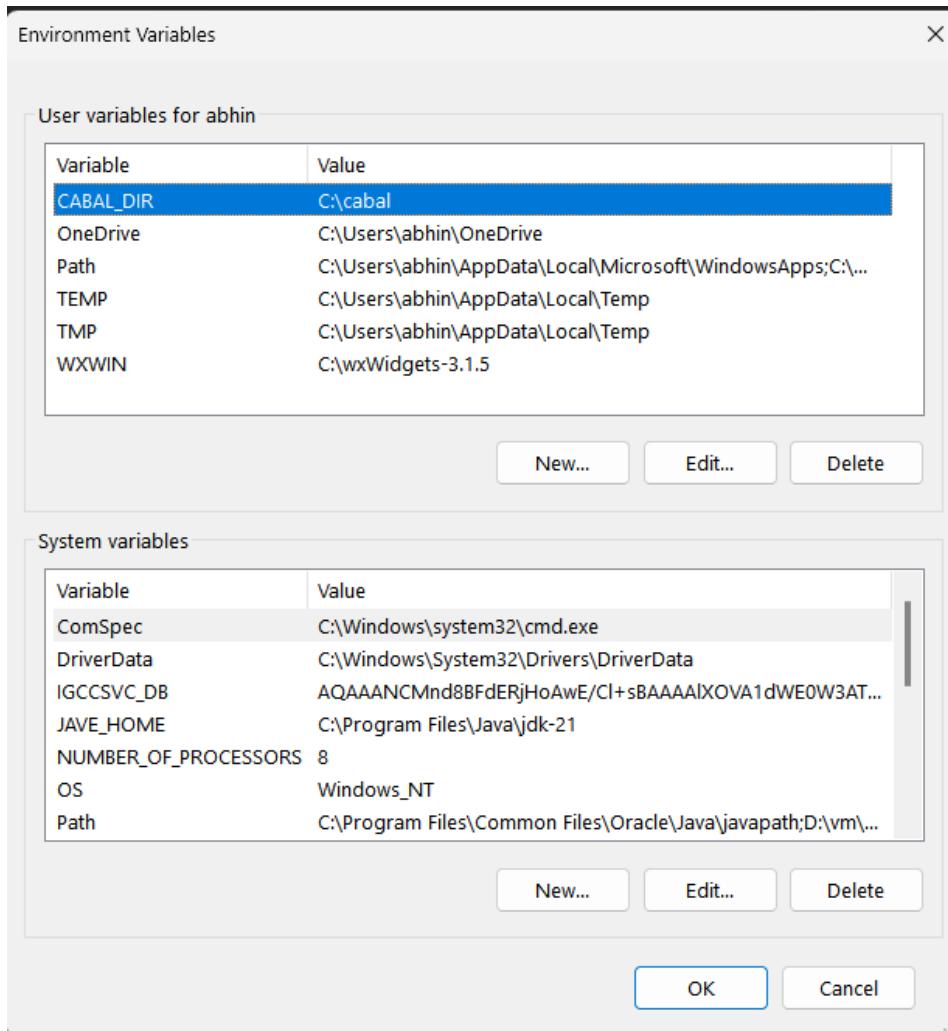
**Step 1:** To install Terraform, visit the official Terraform website mentioned below, go to the Downloads section, select Windows, and download the 64-bit version for your system. website: <https://www.terraform.io/downloads.html>

The screenshot shows the Terraform website's download section for Windows. It features two download options: '386 Version: 1.9.4' and 'AMD64 Version: 1.9.4'. Both have 'Download' buttons next to them. The left sidebar lists operating systems: macOS, Windows (selected), Linux, and FreeBSD. The right sidebar contains links for 'About Terraform', 'Defined cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.', 'Featured docs' (Introduction to Terraform, Configuration Language, Terraform CLI), and a search bar.

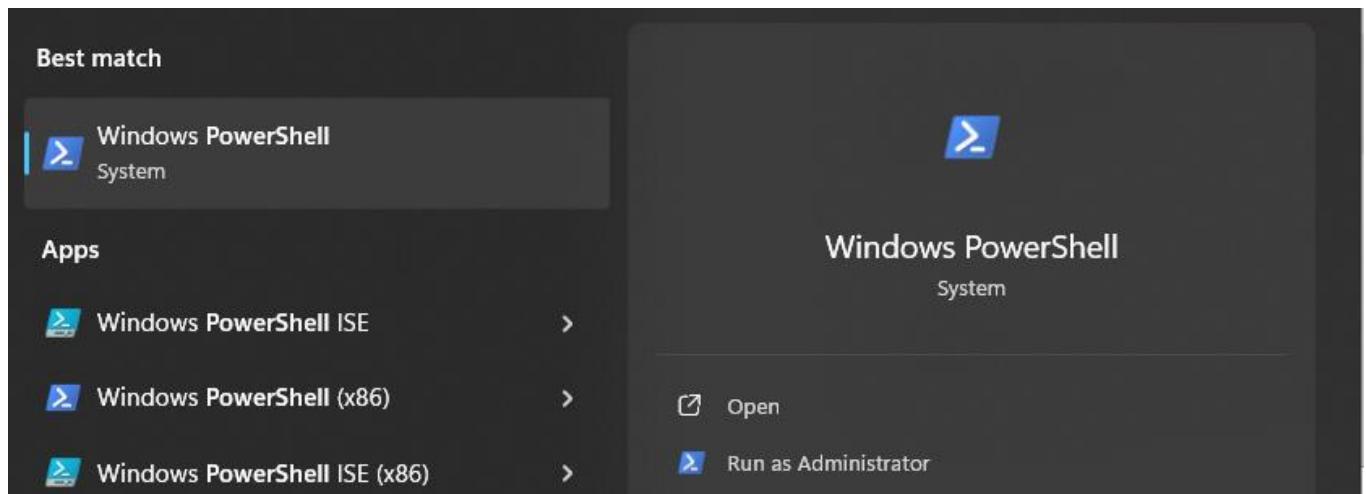
**Step 2:** Extract the downloaded `Terraform.exe` file to the `C:\Terraform` directory on your system.

The screenshot shows the 'Extract Compressed (Zipped) Folders' dialog box. It asks to 'Select a Destination and Extract Files'. The 'Files will be extracted to this folder:' field contains 'C:\terraform'. A 'Browse...' button is available to change the destination. A checked checkbox says 'Show extracted files when complete'. At the bottom are 'Extract' and 'Cancel' buttons.

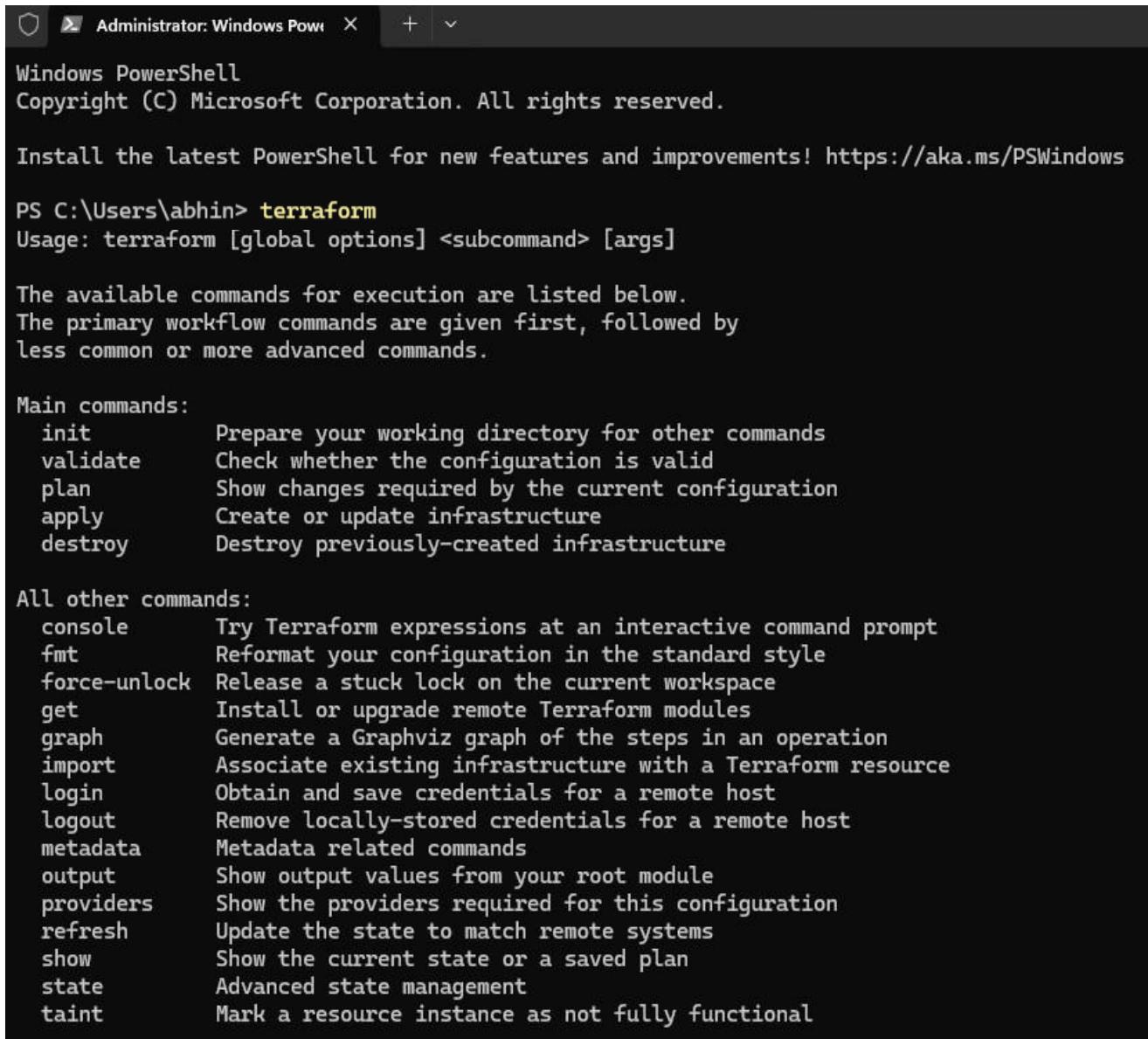
### Step 3: Set the System path for Terraform in Environment Variables



### Step 4: Run the windows powershell as administrator.



**Step 5:** Run “terraform” to verify its functionality. If you encounter any errors, double-check or update the Terraform path in your environment variables.



A screenshot of a Windows PowerShell window titled "Administrator: Windows Pow". The window shows the Terraform help output. It includes the standard PowerShell header, a note to install the latest version, and the Terraform usage command. It then lists main commands (init, validate, plan, apply, destroy) and their descriptions, followed by all other commands and their descriptions.

```
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\abhin> 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
```

## Experiment No. 6

- Creating a docker image using terraform

```
C:\Users\abhinav>docker --version
Docker version 26.1.3, build b72abbb

C:\Users\abhinav>docker

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

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

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

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



- Create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using a text editor and write the following contents into it to create a Ubuntu Linux container.

| This PC > Temporary Storage (D:) > Terraform_Scripts >        |           |                   |             |      |
|---|-----------|-------------------|-------------|------|
|   | Name      | Date modified     | Type        | Size |
| ss  | Docker    | 8/22/2024 4:10 PM | File folder |      |
| This PC > Temporary Storage (D:) > Terraform_Scripts > Docker |           |                   |             |      |
|   | Name      | Date modified     | Type        | Size |
| ls  | docker.tf | 8/22/2024 4:12 PM | TF File     | 1 KB |

## docker.tf - Notepad

```
File Edit Format View Help
terraform {
    required_providers {
        docker = {
            source = "kreuzwerker/docker"
            version = "2.21.0"
        }
    }
}
provider "docker" {
host = "npipe:///./pipe/docker_engine"
}

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

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

#### 4. Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

D:\Terraform_Scripts\Docker>
```

## 5. Execute Terraform plan to see the available resources

```
D:\Terraform_Scripts\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)

    + labels (known after apply)
}

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

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

6. 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”

```
D:\Terraform_Scripts\Dockers>terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach           = false
    + bridge           = (known after apply)
    + command          = (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)

    + labels (known after apply)
}

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

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

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

Enter a value: yes

docker image.ubuntu: Creating...
docker image.ubuntu: Still creating... [19s elapsed] docker image.ubuntu: Still creating... (20s elapsed) docker image.ubuntu: Still creating... [30s elapsed]
docker image.ubuntu: Creation complete after 30s [id=sha256:263966596d42ad38ae9914716692777ba9ff8779a62ad93a74fe82e3e1f ubuntu:latest] docker_container.foo: Creating...
```

7. Check Docker images, Before and After Executing Apply step

| REPOSITORY                                 | TAG                            | IMAGE ID     | CREATED     | SIZE   |
|--|--------------------------------|--------------|-------------|--------|
| mcr.microsoft.com/dotnet/framework/aspnet  | 4.8-windowsservercore-ltsc2022 | 0b1ef1176a57 | 6 weeks ago | 5.43GB |
| mcr.microsoft.com/dotnet/framework/sdk     | 4.8-windowsservercore-ltsc2022 | c3f8c2735565 | 6 weeks ago | 9.04GB |
| mcr.microsoft.com/dotnet/framework/runtime | 4.8-windowsservercore-ltsc2022 | e69ea8a5ec1b | 6 weeks ago | 5.1GB  |
| mcr.microsoft.com/windows/servercore       | ltsc2022                       | e60f47e635b7 | 7 weeks ago | 4.84GB |
| mcr.microsoft.com/windows/nanoserver       | ltsc2022                       | f0ca29645006 | 7 weeks ago | 292MB  |

| REPOSITORY                                 | TAG                            | IMAGE ID     | CREATED       | SIZE   |
|--|--------------------------------|--------------|---------------|--------|
| mcr.microsoft.com/dotnet/framework/aspnet  | 4.8-windowsservercore-ltsc2022 | 0b1ef1176a57 | 6 weeks ago   | 5.43GB |
| mcr.microsoft.com/dotnet/framework/sdk     | 4.8-windowsservercore-ltsc2022 | c3f8c2735565 | 6 weeks ago   | 9.04GB |
| mcr.microsoft.com/dotnet/framework/runtime | 4.8-windowsservercore-ltsc2022 | e69ea8a5ec1b | 6 weeks ago   | 5.1GB  |
| mcr.microsoft.com/windows/servercore       | ltsc2022                       | e60f47e635b7 | 7 weeks ago   | 4.84GB |
| mcr.microsoft.com/windows/nanoserver       | ltsc2022                       | f0ca29645006 | 7 weeks ago   | 292MB  |
| ubuntu                                     | Latest                         | 2dc39ba859dc | 2 minutes ago | 77.8MB |

8. Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
D:\Terraform_Scripts\Dockers>terraform destroy
docker_image.ubuntu: Refreshing state... [id:rsha256:2dc29b50dcfd2d30101475692777ba087762d92de0221fubuntu:latest]
Terrafore 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_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  - id          = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elfubuntu:latest" -> null
  - image_id    = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - Latest      = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - name        = "ubuntu:latest" -> null
  - repo digest = "ubuntu@sha256:204a3d7bb4d7723452be3923b06cd7043704030041c83c#7856c1" -> null
}

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

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

  Enter a value: yes

docker image.ubuntu: Destroying... [id:sha256:2de99b59cd42ade83814765692777ba5ff8779a62ad93ad62e3e1fubuntu:latest]
docker image.ubuntu: Destruction complete after is

Destroy complete! Resources: 1 destroyed.
```

9. Check Docker images, After Executing Destroy step

| REPOSITORY                                 | TAG                            | IMAGE ID     | CREATED     | SIZE   |
|--|--------------------------------|--------------|-------------|--------|
| mcr.microsoft.com/dotnet/framework/aspnet  | 4.8-windowsservercore-ltsc2022 | 0b1ef1176a57 | 6 weeks ago | 5.43GB |
| mcr.microsoft.com/dotnet/framework/sdk     | 4.8-windowsservercore-ltsc2022 | c3f8c2735565 | 6 weeks ago | 9.04GB |
| mcr.microsoft.com/dotnet/framework/runtime | 4.8-windowsservercore-ltsc2022 | e69ea8a5ec1b | 6 weeks ago | 5.1GB  |
| mcr.microsoft.com/windows/servercore       | ltsc2022                       | e60f47e635b7 | 7 weeks ago | 4.84GB |
| mcr.microsoft.com/windows/nanoserver       | ltsc2022                       | f0ca29645006 | 7 weeks ago | 292MB  |

```
D:\Terraform_Scripts\Dockers>_
```

## Experiment 7

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

### **Theory:**

#### **What is SAST?**

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

#### **What problems does SAST solve?**

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

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

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

#### **Why is SAST important?**

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

## **What are the key steps to run SAST effectively?**

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

1. **Finalize the tool.** Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.
2. **Create the scanning infrastructure, and deploy the tool.** This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.
3. **Customize the tool.** Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.
4. **Prioritize and onboard applications.** Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.
5. **Analyze scan results.** This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation.
6. **Provide governance and training.** Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

## **Integrating Jenkins with SonarQube:**

### **Prerequisites:**

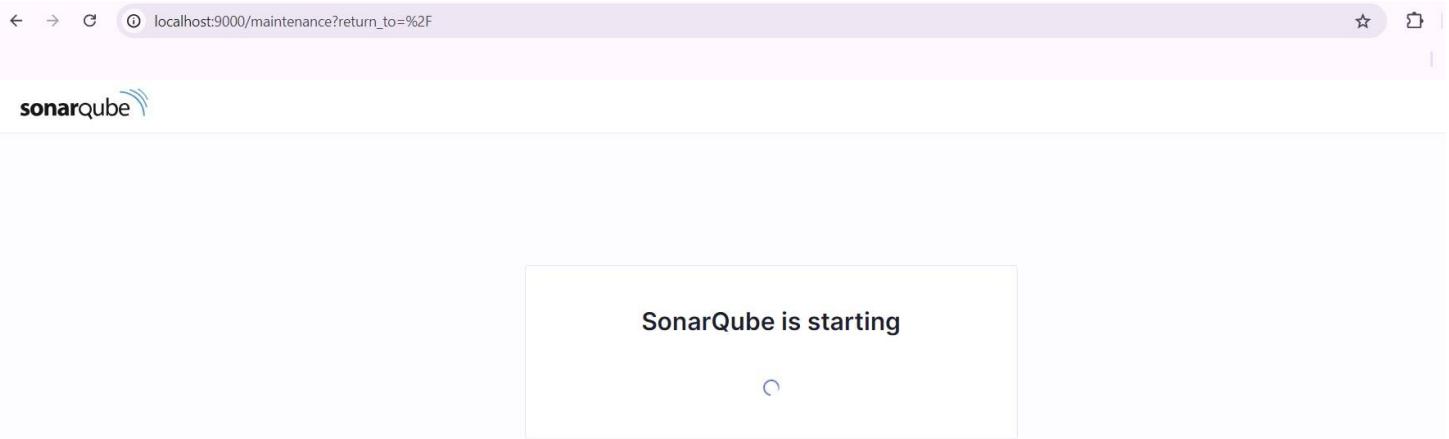
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

## Steps to integrate Jenkins with SonarQube

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

```
C:\Users\ADMIN>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9fec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
de76efbeef2054aeb442b86ba54c2916039b8757b388482d9780ffc69f5d8bbe
```

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



4. Login to SonarQube using username **admin** and password **admin**.
5. Create a manual project in SonarQube with the name **sonarqube**

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

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins Marketplace search results for 'sonar'. A search bar at the top contains the text 'sonar'. To the right of the search bar are two buttons: 'Install' and a dropdown menu. Below the search bar, there is a table with the following columns: 'Install', 'Name ↓', and 'Released'. A single result is listed: 'SonarQube Scanner 2.17.2'. The 'Install' button is highlighted with a blue border. Below the plugin name are two tabs: 'External Site/Tool Integrations' and 'Build Reports'. A description of the plugin follows: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' To the right of the description is the release date: '7 mo 9 days ago'.

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

The screenshot shows the 'SonarQube installations' configuration page. At the top, there is a breadcrumb navigation: Dashboard > Manage Jenkins > System >. The main section is titled 'SonarQube installations' and has a subtitle 'List of SonarQube installations'. It contains three fields: 'Name' (with value 'sonarqube'), 'Server URL' (with value 'http://localhost:9000'), and 'Server authentication token' (with a dropdown menu showing '- none -'). There is also a '+ Add ▾' button and an 'Advanced ▾' button.

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

The screenshot shows the 'Global Tool Configuration' page with the 'SonarQube Scanner' section selected. The 'Name' field is set to 'sonarqube'. The 'Install automatically' checkbox is checked. Below this, there is a 'Install from Maven Central' section with a 'Version' dropdown menu showing 'SonarQube Scanner 6.1.0.4477'. At the bottom of the configuration section is an 'Add Installer ▾' button.

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.

10. Choose this GitHub repository in Source Code Management.

[https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git)

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

[https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git)

Credentials [?](#)

- none -

[+ Add](#) ▾

Advanced ▾

It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

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

The screenshot shows the Jenkins configuration interface for a job. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected and highlighted in blue), and 'Post-build Actions'. The main content area is titled 'Build Steps' and contains a section for 'Execute SonarQube Scanner'. It includes fields for 'SonarQube Installation' (set to 'sonarqube'), 'JDK' (set to '(Inherit From Job)'), 'Path to project properties' (empty), 'Analysis properties' (containing the configuration: sonar.projectKey=sonarqube, sonar.login=admin, sonar.password=admin123, sonar.sources=C:\\\\ProgramData\\\\jenkins\\\\workspace\\\\SonarQube, sonar.host.url=http://127.0.0.1:9000), 'Additional arguments' (empty), and 'JVM Options' (-Dsonar.ws.timeout=300). At the bottom are 'Save' and 'Apply' buttons.

12. Go to [http://localhost:9000/<user\\_name>/permissions](http://localhost:9000/<user_name>/permissions) and allow Execute Permissions to the Admin user.

The screenshot shows the 'Permissions' page for the 'Administrator' user ('admin'). The top navigation bar includes links for 'Administer System', 'Administer', 'Execute Analysis', and 'Create'. Below the navigation, there are checkboxes for 'Quality Gates' (unchecked) and 'Quality Profiles' (checked). A note indicates that 'Execute Analysis' permissions are granted by default. The main content area shows the user's name and role ('Administrator admin') with a 'Edit' link.

13. Run The Build.

The screenshot shows the 'Status' page for a Jenkins job. The sidebar on the left lists 'Changes', 'Workspace', 'Build Now' (which is selected and highlighted in blue), 'Configure', 'Delete Project', 'SonarQube', and 'Rename'. The main content area displays the status of the build.

## Check the console output.

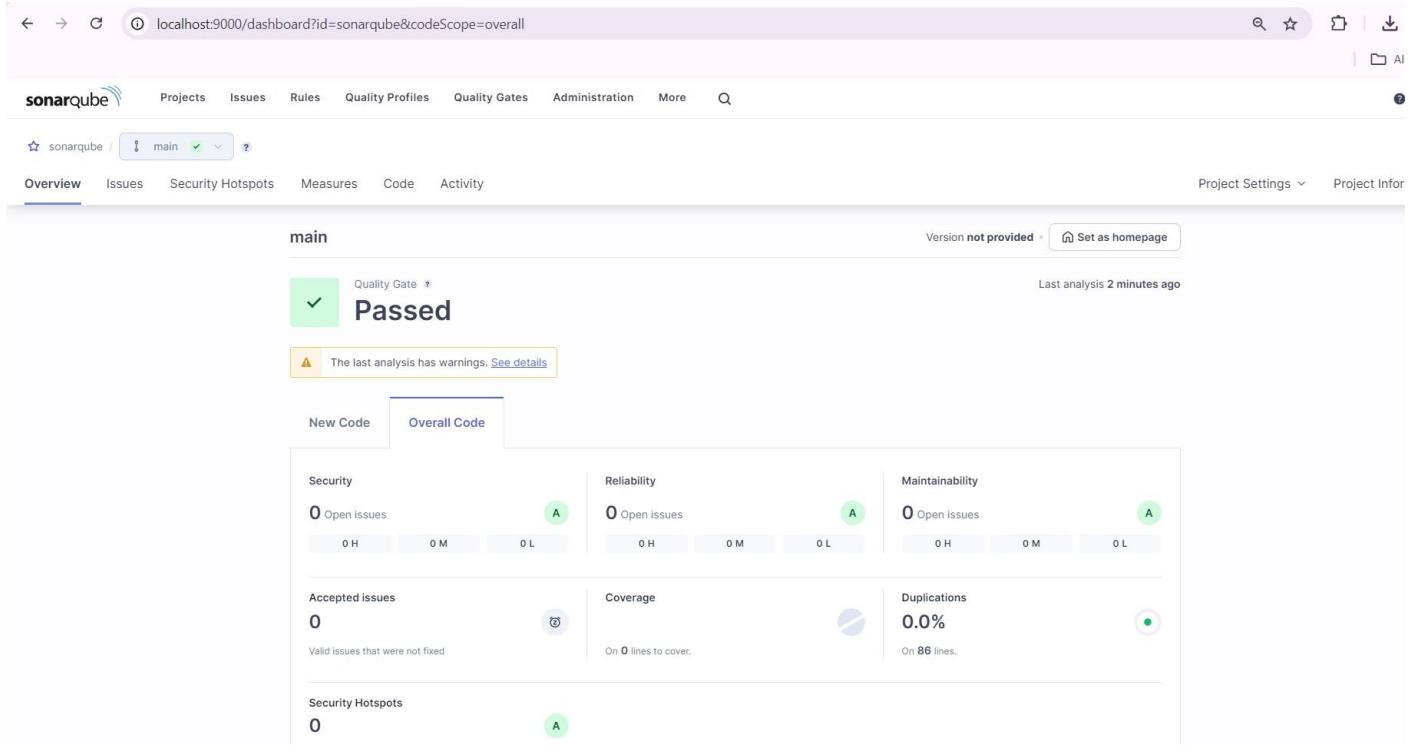
### Console Output

[Download](#)[Copy](#)[View as plain text](#)

```
Started by user Abhinav Swaminathan
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.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 2.45.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[SonarQube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -
Dsonar.login=admin -Dsonar.host.url=http://127.0.0.1:9000 -Dsonar.sources=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube -Dsonar.password=admin123 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
16:16:39.198 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://127.0.0.1:9000'
16:16:39.206 INFO Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
16:16:39.206 INFO Project root configuration file: NONE
16:16:39.230 INFO SonarScanner CLI 6.1.0.4477
16:16:39.230 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
16:16:39.230 INFO Windows 11 10.0. amd64
16:16:39.230 INFO SONAR_SCANNER_OPTS=-Dsonar.ws.timeout=300
16:16:39.254 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache

16:16:58.734 INFO Using git CLI to retrieve untracked files
16:16:58.791 INFO Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
16:16:58.856 INFO 14 source files to be analyzed
16:16:59.154 INFO 14/14 source files have been analyzed
16:16:59.154 INFO Sensor TextAndSecretsSensor [text] (done) | time=1306ms
16:16:59.163 INFO -----
16:16:59.373 INFO Sensor C# [csharp]
16:16:59.373 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see
https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
16:16:59.373 INFO Sensor C# [csharp] (done) | time=0ms
16:16:59.373 INFO Sensor Analysis Warnings import [csharp]
16:16:59.379 INFO Sensor Analysis Warnings import [csharp] (done) | time=0ms
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp]
16:16:59.379 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp] (done) | time=6ms
16:16:59.379 INFO Sensor Zero Coverage Sensor
16:16:59.389 INFO Sensor Zero Coverage Sensor (done) | time=10ms
16:16:59.389 INFO SCM Publisher SCM provider for this project is: git
16:16:59.389 INFO SCM Publisher 4 source files to be analyzed
16:16:59.838 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=449ms
16:16:59.846 INFO CPD Executor Calculating CPD for 0 files
16:16:59.846 INFO CPD Executor CPD calculation finished (done) | time=0ms
16:16:59.854 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
16:17:00.121 INFO Analysis report generated in 120ms, dir size=201.1 kB
16:17:00.195 INFO Analysis report compressed in 57ms, zip size=22.4 kB
16:17:00.393 INFO Analysis report uploaded in 195ms
16:17:00.394 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube
16:17:00.395 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
16:17:00.395 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=acd819f5-9e70-42ab-bff7-3cc893e2cae4
16:17:00.405 INFO Analysis total time: 18.743 s
16:17:00.408 INFO SonarScanner Engine completed successfully
16:17:00.494 INFO EXECUTION SUCCESS
16:17:00.494 INFO Total time: 21.288s
Finished: SUCCESS
```

#### 14. Once the build is complete, check the project in SonarQube.



The screenshot shows the SonarQube dashboard for the 'main' project. At the top, there's a green 'Passed' status with a checkmark icon. Below it, a message says 'The last analysis has warnings. See details'. The dashboard is divided into several sections: Security (0 Open issues), Reliability (0 Open issues), Maintainability (0 Open issues), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (0.0% on 86 lines). The last analysis was 2 minutes ago.

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

#### Conclusion:

In this experiment, the integration of Jenkins with SonarQube for performing Static Application Security Testing (SAST) was explored. SonarQube was set up within a Docker container, and Jenkins was configured to utilize the SonarQube scanner.

By creating a manual project in SonarQube and configuring the necessary authentication and tools in Jenkins, a seamless connection was established between Jenkins and SonarQube for static code analysis.

The integration was tested using a sample GitHub repository, successfully executing a build and analyzing the project's code quality through SonarQube.

This process provided valuable insights into the SAST process, Jenkins automation, and SonarQube's capabilities in identifying potential code vulnerabilities.

## Experiment No. 8

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

### **Theory:**

#### **What is SAST?**

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

#### **What problems does SAST solve?**

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

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

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

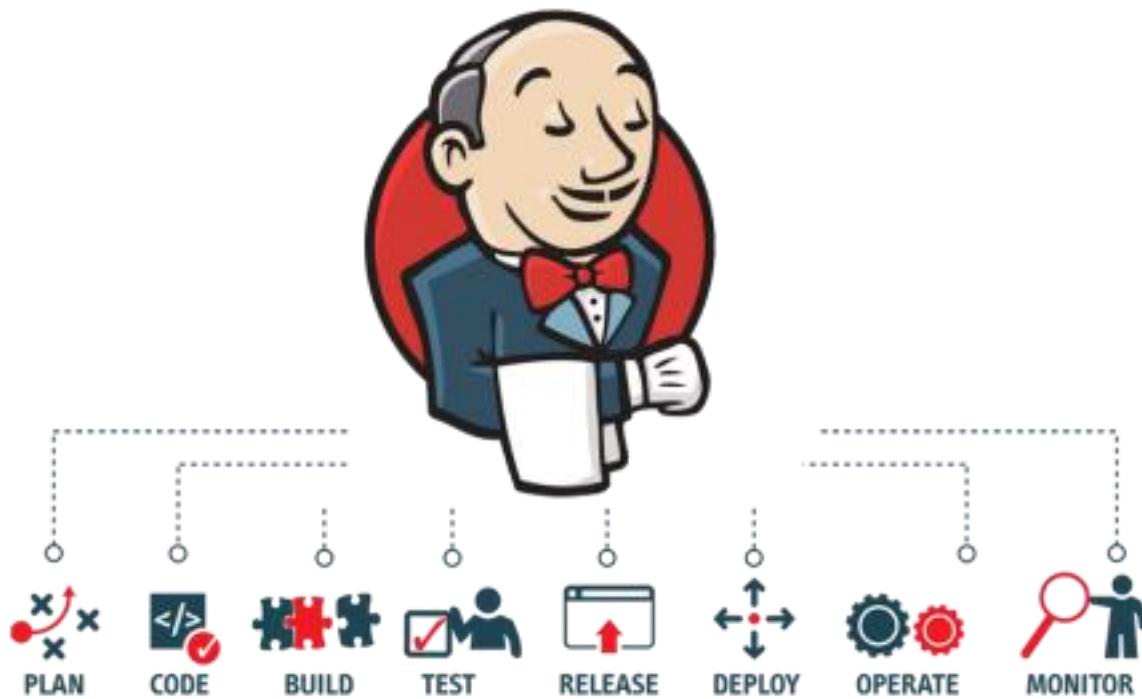
#### **Why is SAST important?**

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

## What is a CI/CD Pipeline?

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

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



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

## What is SonarQube?

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

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

## **Benefits of SonarQube**

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimizing the life of applications.
- **Increase productivity** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- **Quality code** - Code quality control is an inseparable part of the process of software development.
- **Detect Errors** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- **Increase consistency** - Determines where the code criteria are breached and enhances the quality
- **Business scaling** - No restriction on the number of projects to be evaluated
- **Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills

## **Integrating Jenkins with SonarQube:**

### **Prerequisites:**

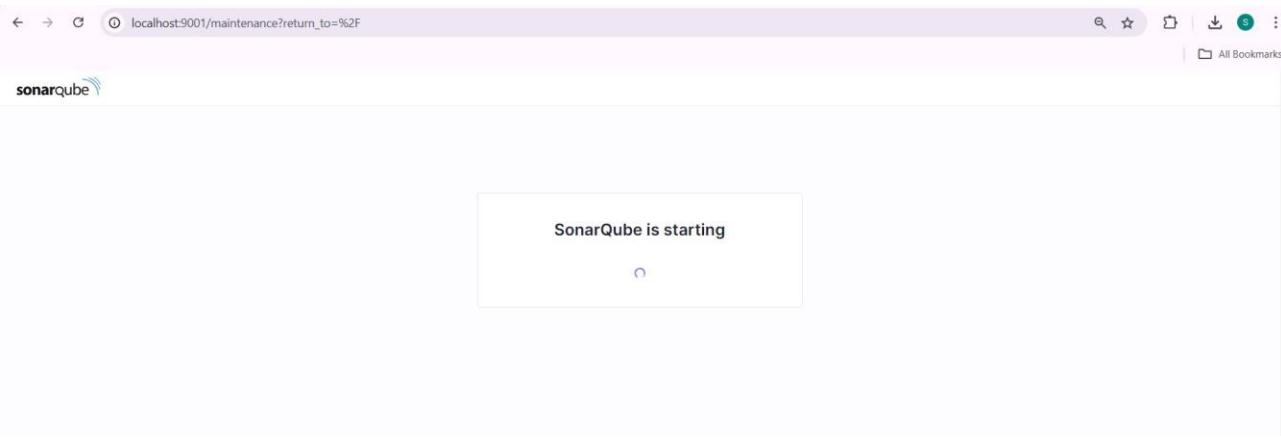
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

## **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.
2. Run SonarQube in a Docker container using this command –

```
C:\Users\ADMIN>docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest  
fda86b00e3989f3eb5aca8396b29b2a0adc95bcfe0fc5d85cf1237491e7678b9
```

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



4. Login to SonarQube using username *admin* and password *admin*.
5. Create a manual project in SonarQube with the name **sonarqube-test**

1 of 2

## Create a local project

Project display name \*

sonarqube-test



Project key \*

sonarqube-test



Main branch name \*

main

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

[Cancel](#)

[Next](#)

Setup the project and come back to Jenkins Dashboard.

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

### New Item

Enter an item name

SonarQube-8

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**

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/"  
        }  
    }  
}
```

Pipeline

Definition

Pipeline script

The screenshot shows the Jenkins Pipeline Script Editor. The 'Definition' tab is selected, and the 'Pipeline script' section contains the following Groovy code:

```
stage('Cloning the GitHub Repo') {  
    git 'https://github.com/shazforiot/GOL.git'  
}  
  
stage('SonarQube analysis') {  
    withSonarQubeEnv('sonarqube') {  
        bat """  
C:\\ProgramData\\Jenkins\\.jenkins\\tools\\hudson.plugins.sonar.SonarRunnerInstallation\\sonarqube\\bin\\sonar-scanner ^  
-D sonar.login=admin ^  
-D sonar.password=admin123 ^  
-D sonar.projectKey=sonarqube-test ^  
-D sonar.exclusions=vendor/**,resources/**,**/*.java ^  
-D sonar.host.url=http://127.0.0.1:9001/  
"""  
    }  
}
```

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

8. Run The Build.

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

**SonarQube-8**

- Status
- </> Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Stages
- Rename
- Pipeline Syntax

Build History trend

Filter... /

#4 Sep 26, 2024, 6:04PM

#3 Sep 26, 2024, 5:13PM

| Cloning the GitHub Repo    | SonarQube analysis |
|----------------------------|--------------------|
| 5s                         | 31min 25s          |
| 1s                         | 12min 7s           |
| 8s                         | 50min 43s aborted  |
| 10s                        |                    |
| #4 Sep 26 18:04 No Changes |                    |
| #3 Sep 26 17:13 No Changes |                    |
| #2 Sep 26 17:11 No Changes |                    |
| #1 Sep 26 17:11 No Changes |                    |

Dashboard > SonarQube-8 > #4

Status

</> Changes

Console Output

Skipping 4.248 KB. Full Log

Console Output

View as plain text

Edit Build Information

Delete build #4

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Replay

Pipeline Steps

Workspaces

Previous Build

```

18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 859. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1085. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 634. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.

references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 41. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 17. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 296. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 INFO CPD Executor CPD calculation finished (done) | time=15897ms
18:13:39.674 INFO SCM revision ID 'ba798ba7e1b576f04a461232b0412c5e6e1e5e4'
18:15:49.696 INFO Analysis report generated in 502ms, dir size=127.2 MB
18:16:08.759 INFO Analysis report compressed in 19048ms, zip size=29.6 MB
18:16:09.884 INFO Analysis report uploaded in 1125ms
18:16:09.887 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9001/dashboard?id=sonarqube-test
18:16:09.887 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:16:09.887 INFO More about the report processing at http://127.0.0.1:9001/api/ce/task?id=6f22c333-3777-4a21-b058-0ab4c049625c
18:16:22.970 INFO Analysis total time: 12:02.242 s
18:16:22.975 INFO SonarScanner Engine completed successfully
18:16:23.699 INFO EXECUTION SUCCESS
18:16:23.706 INFO Total time: 12:05.758s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## 10. After that, check the project in SonarQube.

The screenshot shows the SonarQube main dashboard for the 'main' project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation is a breadcrumb trail: sonarqube-test / main. The main content area has tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'main' tab is selected. A large green box indicates a 'Passed' status with a checkmark. Below it, a message says 'New analysis in progress'. The dashboard displays various metrics: Security (0 Open issues), Reliability (68k Open issues), Maintainability (164k Open issues), Accepted issues (0), Coverage (On 0 lines to cover), Duplications (50.6%), and Security Hotspots (3). The overall status is 'Passed'.

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

## 11. Code Problems -

The screenshot shows the SonarQube Issues tab for the 'main' project. The left sidebar includes filters for Software Quality (Security: 0, Reliability: 47k, Maintainability: 0), Severity (0), Type (Bug: 47k, Vulnerability: 0, Code Smell: 164k), and Scope. The main area lists three code problems under the 'gameoflife-core/build/reports/tests/all-tests.html' report:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element. Intentionality: accessibility wcag2-a  
Reliability: 0  
Severity: 0  
Type: Bug  
Last 2min effort - 4 years ago - ⚡ Bug - ⚡ Major
- Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency: user-experience  
Reliability: 0  
Severity: 0  
Type: Bug  
Last 5min effort - 4 years ago - ⚡ Bug - ⚡ Major
- Add "<th>" headers to this "<table>". Intentionality: accessibility wcag2-a  
Reliability: 0  
Severity: 0  
Type: Bug  
Last 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

A yellow warning box at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

## Code Smells

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar is collapsed, and the main area displays a list of code smells under the 'gameoflife-acceptance-tests/Dockerfile' component. The first item is a 'Code Smell' named 'Use a specific version tag for the image.' with an 'Intentionality' of 'Maintainability' and a priority of 'Major'. Below it are three other similar items. A sidebar on the left lists categories like Software Quality, Severity, Type (Bug, Vulnerability, Code Smell), and Scope. A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

## Intentional Issues

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar is collapsed, and the main area displays a list of intentional issues under the 'gameoflife-acceptance-tests/Dockerfile' component. The first item is an 'Intentionality' issue named 'Use a specific version tag for the image.' with an 'Intentionality' of 'Maintainability' and a priority of 'Major'. Below it are three other similar items. A sidebar on the left lists categories like Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility) and Software Quality (Security, Reliability, Maintainability). A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

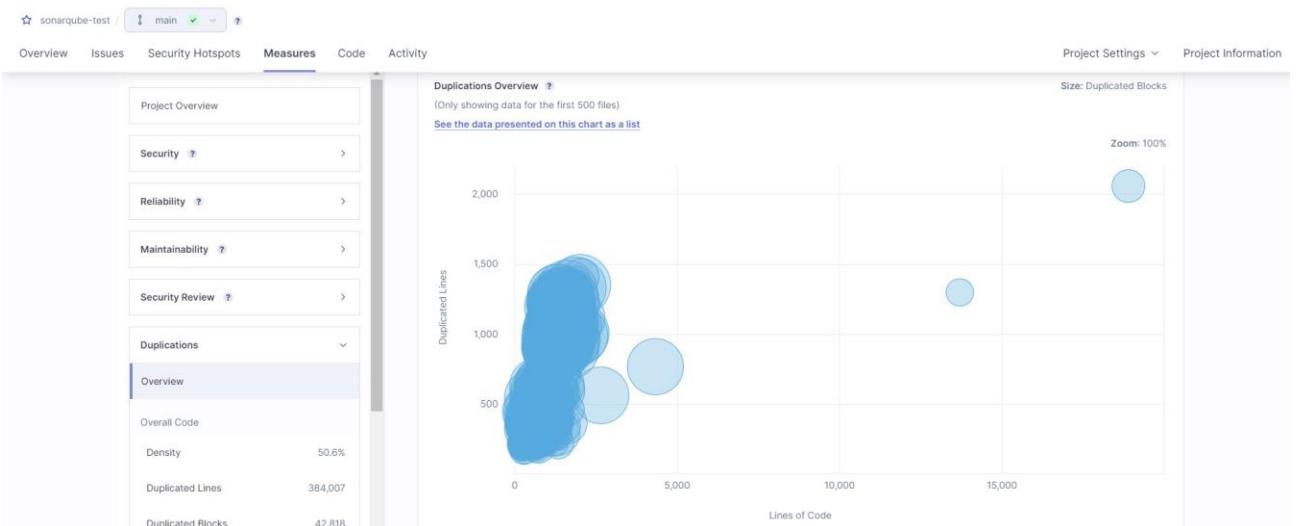
## Reliability Issue

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar is collapsed, and the main area displays a list of reliability issues under the 'gameoflife-core/build/reports/tests/all-tests.html' component. The first item is a 'Reliability' issue named 'Anchors must have content and the content must be accessible by a screen reader.' with a 'Consistency' of 'Maintainability' and a priority of 'Minor'. Below it are four other similar items. A sidebar on the left lists categories like Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility) and Software Quality (Security, Reliability, Maintainability). A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only.'

## Maintainability Issue

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Issues' tab is selected. On the left, there's a sidebar with filters for 'Clean Code Attribute' (Consistency: 164k, Intentionality: 15, Adaptability: 0, Responsibility: 0) and 'Software Quality' (Security: 0, Reliability: 21k, Maintainability: 164k). The main area displays a list of issues under the file 'gameoflife-core/build/reports/tests/all-tests.html'. The first issue is 'Remove this deprecated "width" attribute.' (Consistency, html5 obsolete, Major). Other issues listed include 'Remove this deprecated "align" attribute.', 'Remove this deprecated "align" attribute.', and 'Remove this deprecated "size" attribute.' All issues are marked as 'Open' and 'Not assigned'.

## Duplicates



In this way, we have created a CI/CD Pipeline with Jenkins and integrated it with SonarQube to find issues in the code like bugs, code smells, duplicates, cyclomatic complexities, etc.

## Conclusion:

In this experiment, I successfully created a CI/CD pipeline using Jenkins integrated with SonarQube for static code analysis on a sample Java application. I set up SonarQube in a Docker container and configured Jenkins to clone the GitHub repository and perform the analysis. The pipeline detected various issues, including bugs, code smells, and security vulnerabilities, which I reviewed in SonarQube. This experience enhanced my skills in configuring CI/CD tools and highlighted the importance of maintaining code quality through automation. Overall, I gained valuable insights into integrating tools for effective software development practices.

### **Adv DevOps Exp 09**

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

#### **Theory:**

##### **What is Nagios?**

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

Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

##### **Why We Need Nagios tool?**

Here are the important reasons to use Nagios monitoring tool:

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

##### **Features of Nagios**

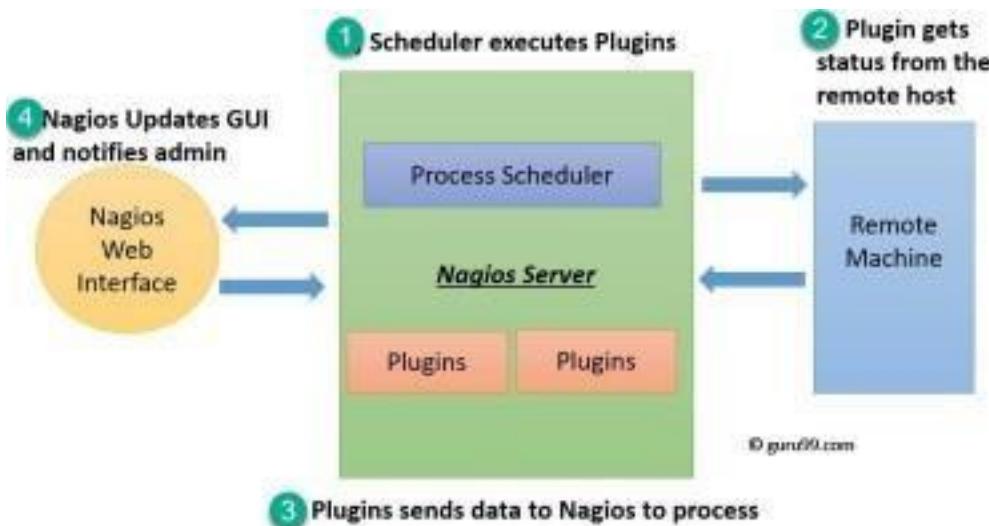
Following are the important features of Nagios monitoring tool:

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

- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

### Nagios Architecture

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



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

**Step 1: Create a security group with the required configurations |**  
 have created a new security group with a name 'newsecurity'

[EC2](#) > [Security Groups](#) > Create security group

### Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, provide a name and optional description, and select the VPC.

**Basic details**

Security group name Info  
 Name cannot be edited after creation.

Description Info

VPC Info

I have modified the INBOUND RULES as follows

| Type <small>Info</small> | Protocol <small>Info</small> | Port range <small>Info</small> | Source <small>Info</small> | Description - optional <small>Info</small> |                        |
|--------------------------|------------------------------|--------------------------------|----------------------------|--|------------------------|
| HTTP                     | TCP                          | 80                             | Anywh... ▾                 | <input type="text" value="::/0"/> X        | <a href="#">Delete</a> |
| HTTPS                    | TCP                          | 443                            | Anywh... ▾                 | <input type="text" value="0.0.0.0/0"/> X   | <a href="#">Delete</a> |
| SSH                      | TCP                          | 22                             | Anywh... ▾                 | <input type="text" value="0.0.0.0/0"/> X   | <a href="#">Delete</a> |
| All ICMP - IPv6          | IPv6 ICMP                    | All                            | Anywh... ▾                 | <input type="text" value="::/0"/> X        | <a href="#">Delete</a> |
| All ICMP - IPv4          | ICMP                         | All                            | Anywh... ▾                 | <input type="text" value="0.0.0.0/0"/> X   | <a href="#">Delete</a> |
| All traffic              | All                          | All                            | Anywh... ▾                 | <input type="text" value="0.0.0.0/0"/> X   | <a href="#">Delete</a> |
| Custom TCP               | TCP                          | 5666                           | Anywh... ▾                 | <input type="text" value="0.0.0.0/0"/> X   | <a href="#">Delete</a> |

**Step 2: Create ec2 instance**

Name it as nagios-host. Select instance type as amazon-linux and choose the already created key pair and security group

**Name and tags** [Info](#)

Name  
nagios-host [Add additional tags](#)

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

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

**Recents** **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Images (AMIs)**

**Key pair (login)** [Info](#)

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

Key pair name - *required*  
abhinav [Create new key pair](#)

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

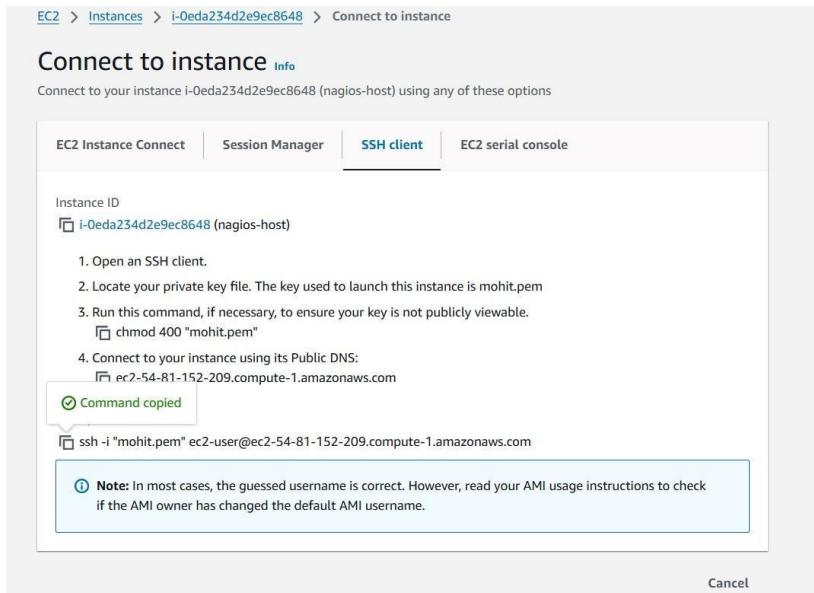
**Common security groups** [Info](#)

Select security groups  
newsecurity sg-05d7468fe3a2f7a8e X  
VPC: vpc-0aa3db8937df8678b

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Copy the given ssh command, as we will require it for logging into our nagios-host instance from our windows powershell



### Step 3: Open an administrative powershell and remotely login using the above mentioned ssh command

```

PS C:\Windows\system32> cd C:\Users\DeLL\Downloads
PS C:\Users\DeLL\Downloads> ssh -i "mohit.pem" ec2-user@ec2-54-81-152-209.compute-1.amazonaws.com
,#
~\### Amazon Linux 2023
~~\####\ https://aws.amazon.com/linux/amazon-linux-2023
~~\###| https://aws.amazon.com/linux/amazon-linux-2023
~~\###>
~~\###/
~~\###/
~~\###/
Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18
,#
~\### Amazon Linux 2023
~~\####\ https://aws.amazon.com/linux/amazon-linux-2023
~~\###| https://aws.amazon.com/linux/amazon-linux-2023
~~\###>
~~\###/
~~\###/
~~\###/
Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18
[ec2-user@ip-172-31-92-249 ~]$ sudo yum update
Last metadata expiration check: 0:13:13 ago on Mon Sep 30 09:23:03 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-92-249 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:13:23 ago on Mon Sep 30 09:23:03 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!

```

And then run these commands

**sudo yum update sudo yum install httpd php**

| Package                              | Architecture | Version                 | Repository  | Size  |
|--------------------------------------|--------------|-------------------------|-------------|-------|
| <b>Installing:</b>                   |              |                         |             |       |
| httpd                                | x86_64       | 2.4.62-1.amzn0923       | amazonlinux | 48 K  |
| php8.3                               | x86_64       | 0.3.10-1.amzn2023.0.1   | amazonlinux | 10 K  |
| <b>Installing dependencies:</b>      |              |                         |             |       |
| 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-htpd                   | noarch       | 18.0.0-12.amzn2023.0.3  | amazonlinux | 19 K  |
| httpd-foreground                     | x86_64       | 2.4.62-1.amzn0923       | amazonlinux | 14 K  |
| httpd-fs-filesystem                  | noarch       | 2.4.62-1.amzn0923       | amazonlinux | 14 K  |
| httpd-tools                          | x86_64       | 2.4.62-1.amzn0923       | amazonlinux | 81 K  |
| libbrotli                            | x86_64       | 1.0.9-4.amzn2023.0.2    | amazonlinux | 315 K |
| libbodium                            | x86_64       | 1.0.19-4.amzn0923       | amazonlinux | 176 K |
| libxml                               | x86_64       | 1.1.34-5.amzn0923.0.2   | amazonlinux | 241 K |
| mailcap                              | noarch       | 2.4.40-1.amzn0923.0.3   | amazonlinux | 31 K  |
| nginx-fs-filesystem                  | noarch       | 1.13.0-0.1.amzn0923.0.1 | amazonlinux | 9.8 K |
| psql                                 | x86_64       | 8.1.10-10.amzn0923.0.1  | amazonlinux | 3.7 M |
| php8.3-common                        | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 737 K |
| php8.3-process                       | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 45 K  |
| php8.3-xml                           | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 154 K |
| <b>Installing weak dependencies:</b> |              |                         |             |       |
| apr-util-devel                       | x86_64       | 1.6.3-1.amzn2023.0.1    | amazonlinux | 17 K  |
| mod_ssl                              | x86_64       | 2.0.27-1.amzn0923.0.3   | amazonlinux | 166 K |
| mod_lua                              | x86_64       | 2.4.62-1.amzn0923       | amazonlinux | 61 K  |
| php8.3-fpm                           | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 1.9 M |
| php8.3-mbstring                      | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 528 K |
| php8.3-openssl                       | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 379 K |
| php8.3-pdo                           | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 89 K  |
| php8.3-sodium                        | x86_64       | 8.3.10-10.amzn0923.0.1  | amazonlinux | 41 K  |

**sudo yum install gcc glibc glibc-common**

```
[ec2-user@ip-172-31-41-160 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:02 ago on Wed Oct  2 12:28:33 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      195 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
libltool-ltdl                 x86_64          2.4.7-1.amzn2023.0.3   amazonlinux      38 k
libcrypt-devel                 x86_64          4.4.33-7.amzn2023     amazonlinux      32 k
make                          x86_64          1:4.3-5.amzn2023.0.2   amazonlinux      534 k

Transaction Summary

Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]: y
Downloading Packages:
(1/13): annobin-docs-10.93-1.amzn2023.0.1.noarch.rpm          852 kB/s |  92 kB  00:00
(2/13): annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64.rpm    6.5 MB/s | 887 kB  00:00
(3/13): gc-8.0.4-5.amzn2023.0.2.x86_64.rpm                  2.3 MB/s | 105 kB  00:00
(4/13): glibc-devel-2.34-52.amzn2023.0.11.x86_64.rpm        1.1 MB/s | 27 kB  00:00
(5/13): cpp-11.4.1-2.amzn2023.0.2.x86_64.rpm                32 MB/s | 10 MB  00:00
(6/13): glibc-headers-x86-2.34-52.amzn2023.0.1.noarch.rpm    2.9 MB/s | 427 kB  00:00
(7/13): kernel-headers-6.1.109-118.189.amzn2023.x86_64.rpm   16 MB/s | 1.4 MB  00:00
(8/13): libmpc-1.2.1-2.amzn2023.0.2.x86_64.rpm            2.1 MB/s | 62 kB  00:00
(9/13): guile22-2.2.7-2.amzn2023.0.3.x86_64.rpm           27 MB/s | 6.4 MB  00:00
(10/13): libltool-ltdl-2.4.7-1.amzn2023.0.3.x86_64.rpm     322 kB/s | 38 kB  00:00
(11/13): libcrypt-devel-4.4.33-7.amzn2023.x86_64.rpm       1.4 MB/s | 32 kB  00:00
```

**sudo yum install gd gd-devel**

```
[ec2-user@ip-172-31-41-160 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:02:57 ago on Wed Oct  2 12:28:33 2024.
Dependencies resolved.

-----  

Package           Architecture Version      Repository  Size  

-----  

Installing:  

gd               x86_64       2.3.3-5.amzn2023.0.3  amazonlinux 139 k
gd-devel         x86_64       2.3.3-5.amzn2023.0.3  amazonlinux 38 k  

Installing dependencies:  

brotli          x86_64       1.0.9-4.amzn2023.0.2  amazonlinux 314 k
brotli-devel    x86_64       1.0.9-4.amzn2023.0.2  amazonlinux 31 k
bzip2-devel     x86_64       1.0.8-6.amzn2023.0.2  amazonlinux 214 k
cairo            x86_64       1.17.6-2.amzn2023.0.1  amazonlinux 684 k
cmake-filesystem x86_64       3.22.2-1.amzn2023.0.4  amazonlinux 16 k
fontconfig       x86_64       2.13.94-2.amzn2023.0.2  amazonlinux 273 k
fontconfig-devel x86_64       2.13.94-2.amzn2023.0.2  amazonlinux 128 k
fonts-filesystem noarch     1:12.0.5-12.amzn2023.0.2  amazonlinux 9.5 k
freetype          x86_64       2.13.2-5.amzn2023.0.1  amazonlinux 423 k
freetype-devel   x86_64       2.13.2-5.amzn2023.0.1  amazonlinux 912 k
glib2-devel      x86_64       2.74.7-689.amzn2023.0.2  amazonlinux 486 k
google-noto-fonts-common google-noto-sans-vf-fonts noarch     30201206-2.amzn2023.0.2  amazonlinux 15 k
graphite2        x86_64       1.3.14-7.amzn2023.0.2  amazonlinux 492 k
graphite2-devel  x86_64       1.3.14-7.amzn2023.0.2  amazonlinux 97 k
harfbuzz         x86_64       7.0.0-2.amzn2023.0.1  amazonlinux 21 k
harfbuzz-devel   x86_64       7.0.0-2.amzn2023.0.1  amazonlinux 868 k
harfbuzz-icu    x86_64       7.0.0-2.amzn2023.0.1  amazonlinux 404 k
jbigkit-langs   x86_64       2.1-21.amzn2023.0.2  amazonlinux 18 k
jbigpacks-core-font-en noarch     3.0-21.amzn2023.0.4  amazonlinux 54 k
libICE           x86_64       1.0.10-6.amzn2023.0.2  amazonlinux 10 k
libSM            x86_64       1.2.3-8.amzn2023.0.2  amazonlinux 71 k
libX11           x86_64       1.7.2-3.amzn2023.0.4  amazonlinux 42 k
libX11-common   noarch     1.7.2-3.amzn2023.0.4  amazonlinux 657 k
libX11-devel    x86_64       1.7.2-3.amzn2023.0.4  amazonlinux 152 k
libX11-xcb     x86_64       1.7.2-3.amzn2023.0.4  amazonlinux 939 k
libXau           x86_64       1.0.9-6.amzn2023.0.2  amazonlinux 12 k
libXau-devel    x86_64       1.0.9-6.amzn2023.0.2  amazonlinux 31 k
libXext          x86_64       1.3.4-6.amzn2023.0.2  amazonlinux 14 k
libXpm           x86_64       3.5.15-2.amzn2023.0.3  amazonlinux 41 k
libXpm-devel    x86_64       3.5.15-2.amzn2023.0.3  amazonlinux 65 k
libXrender       x86_64       0.9.10-14.amzn2023.0.2  amazonlinux 59 k
libXt             x86_64       1.2.0-4.amzn2023.0.2  amazonlinux 28 k
libbbikid-devel x86_64       2.37.4-1.amzn2023.0.4  amazonlinux 181 k

```

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

**sudo adduser -m nagios sudo  
passwd nagios**

```
[ec2-user@ip-172-31-41-160 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-41-160 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-41-160 ~]$
```

Create a new user group & create a new directory for Nagios downloads using the following commands **sudo groupadd nagcmd sudo usermod -a -G nagcmd nagios sudo usermod -a -G nagcmd apache mkdir ~/downloads cd ~/downloads**

Use **wget** to download the source zip files.

In this step, we are downloading, the latest version of nagios and the necessary plugins required to carry out the tasks of setting up a nagios server wget

<https://sourceforge.net/projects/nagios/files/latest/download>

Name: Abhinav Swaminathan

Div: D15C

Roll No: 01

```
[ec2-user@ip-172-31-41-160:~] wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-10-02 12:34:21-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NMNSZP2P-6la2Tltv00GCG7VVV7QGVH08n3tC24QehfMw7vHcoKbHg2iIRxbmfugII0LccNfxeAo[x3]zKG3W&3Dx3d&use_mirror=phoenixnap&r=[following]
--2024-10-02 12:34:21-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NMNSZP2P-6la2Tltv00GCG7VVV7QGVH08n3tC24QehfMw7vHcoKbHg2iIRxbmfugII0LccNfxeAo[x3]zKG3W&3Dx3d&use_mirror=phoenixnap&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://phoenixnap.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viafs=1
--2024-10-02 12:34:21-- https://phoenixnap.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viafs=1
Resolving phoenixnap.dl.sourceforge.net (phoenixnap.dl.sourceforge.net)... 184.164.141.26
Connecting to phoenixnap.dl.sourceforge.net (phoenixnap.dl.sourceforge.net)|184.164.141.26|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download          100%[=====] 1.97M 4.23MB/s   in 0.5s

2024-10-02 12:34:22 (4.23 MB/s) - 'download' saved [2065473/2065473]
```

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

```
[ec2-user@ip-172-31-41-160:~] wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 12:34:46-- 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 7.48MB/s   in 0.4s

2024-10-02 12:34:46 (7.48 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-92-249:~] cd ~/downloads
[ec2-user@ip-172-31-92-249:~/downloads] wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-09-30 09:54:56-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXTSVxrTZ6fgxJvhVA0zpB1bPgbyzLMcDDAAAlgEC1pOKr0cgJN23bKktar1cJ0tVfkx3Dx3d&use_mirror=netactuate&r=[following]
--2024-09-30 09:54:56-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXTSVxrTZ6fgxJvhVA0zpB1bPgbyzLMcDDAAAlgEC1pOKr0cgJN23bKktar1cJ0tVfkx3Dx3d&use_mirror=netactuate&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viafs=1
--2024-09-30 09:54:57-- https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viafs=1
Resolving netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)... 104.225.3.66
Connecting to netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)|104.225.3.66|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download          100%[=====] 1.97M ---KB/s   in 0.07s

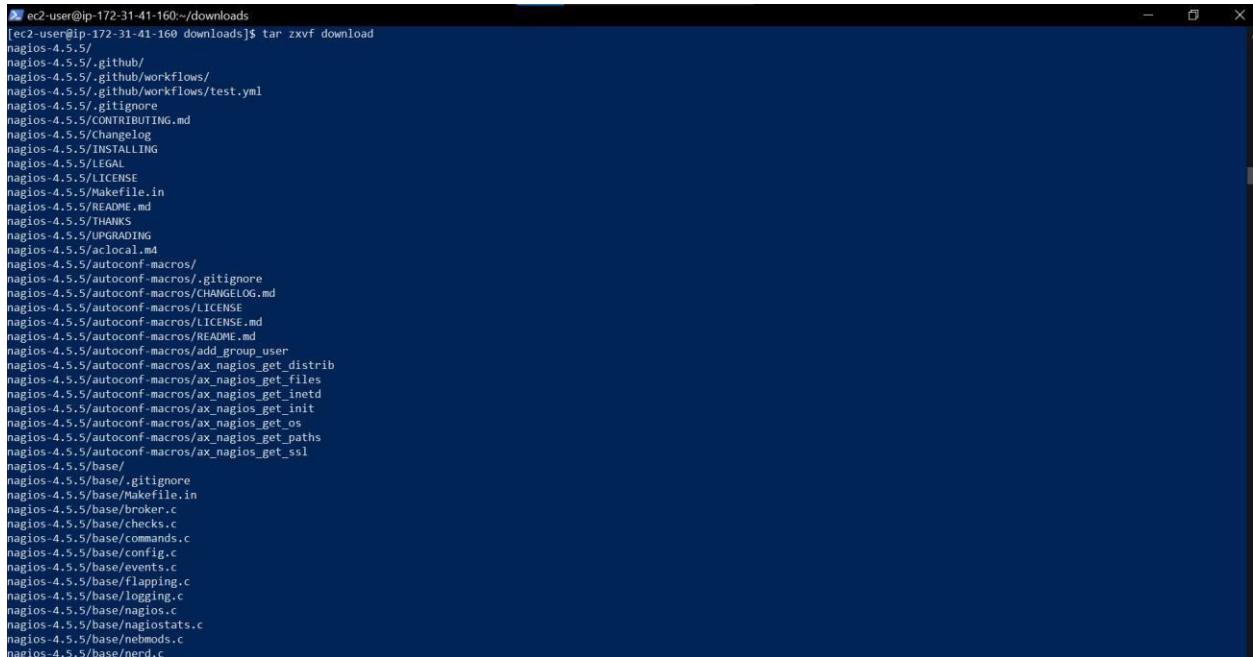
2024-09-30 09:54:57 (29.8 MB/s) - 'download' saved [2065473/2065473]

[ec2-user@ip-172-31-92-249:~/downloads] wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
--2024-09-30 09:56:53-- https://nagios-plugins.org/download/nagios-plugins-2.4.9.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: 2754403 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.9.tar.gz'

nagios-plugins-2.4.9.tar.gz  100%[=====] 2.63M 7.54MB/s   in 0.3s

2024-09-30 09:56:54 (7.54 MB/s) - 'nagios-plugins-2.4.9.tar.gz' saved [2754403/2754403]
```

Now, we run the next command in the following manner tar zxvf <nagios-4.5.5 version> (for me it has gotten saved as 'download') So i wrote **tar zxvf download**



```
[ec2-user@ip-172-31-41-160:~/downloads]$ tar zxvf download
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
nagios-4.5.5/autoconf-macros/ax_nagios_get_inetd
nagios-4.5.5/autoconf-macros/ax_nagios_get_init
nagios-4.5.5/autoconf-macros/ax_nagios_get_os
nagios-4.5.5/autoconf-macros/ax_nagios_get_paths
nagios-4.5.5/autoconf-macros/ax_nagios_get_ssl
nagios-4.5.5/base/
nagios-4.5.5/base/.gitignore
nagios-4.5.5/base/Makefile.in
nagios-4.5.5/base/broker.c
nagios-4.5.5/base/checks.c
nagios-4.5.5/base/commands.c
nagios-4.5.5/base/config.c
nagios-4.5.5/base/events.c
nagios-4.5.5/base/flapping.c
nagios-4.5.5/base/logging.c
nagios-4.5.5/base/nagios.c
nagios-4.5.5/base/nagiosstats.c
nagios-4.5.5/base/nemodns.c
nagios-4.5.5/base/nerd.c
```

After which we are supposed to **change our directory** over there

For eg. **cd nagios-4.5.5...** depending on the version that we have downloaded

Next, Run this command (make sure that you are working inside nagios-4.x.x directory)

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-41-160 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...
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for getopt.h... yes
checking for grp.h... yes
checking for libgen.h... yes
checking for limits.h... yes
checking for math.h... yes
checking for netdb.h... yes
checking for netinet/in.h... yes
checking for pwd.h... yes
checking for regex.h... yes
checking for signal.h... yes
checking for socket.h... no
checking for stdarg.h... yes
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ 
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for strerror... yes
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for strtoul... yes
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for unsetenv... yes
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for type of socket size... size_t
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for Kerberos include files... configure: WARNING: could not find include files
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for pkg-config... pkg-config
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ 
```

After running this command, we get an **error related to ssl header being absent**

For that purpose, we are to run the following command. **sudo yum install openssl-devel** (for ssl header)

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:12:11 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
openssl-devel    x86_64      1:3.0.8-1.amzn2023.0.14   amazonlinux  3.0 M
Transaction Summary
=====
Install 1 Package
Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64.rpm
                                                               26 MB/s | 3.0 MB   00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
Verifying  : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
Installed:
openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
Complete!
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ 
```

Now, Re-run **./configure --with-command-group=nagcmd**

After this, run **make all** command

```

ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 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.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o ..common/shared.o ..common/shared.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o workers.o workers.c
In function 'get_wproc_list':
  inlined from 'get_workers' at workers.c:277:12:
workers.c:253:17: warning: "%s" directive argument is null [-Wformat-overflow]
  253 |         log_debug_info(DHAVA_CHECKS, 1, "Found specialized worker(s) for '%s'. (%slash && *slash != '/') ? slash : cmd_name);"
   |
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o config.o config.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o events.o events.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o logging.o logging.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o macros-base.o ./common/macros.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o netutils.o netutils.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o notifications.o notifications.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o schedulers.o schedulers.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o utils.o utils.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o retention-base.o ./retention.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xretention-base.o ./xrddefaul.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o comments-base.o ./common/comments.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xcomments-base.o ./xdata/xccdefaul.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o objects-base.o ./common/objects.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xobjects-base.o ./xdata/xodtemplate.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o statusdata-base.o ./common/statusdata.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xstatusdata-base.o ./xdata/xsdefault.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o perfdata-base.o ./perfdata.c
gcc -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xperfdata-base.o ./xdata/xpdefault.c
GCC -Wall -I.. -I.. -I./lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o downtime-base.o ./common/downtime.c
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/lib'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5'
on doing this. Pay particular attention to the docs on
object configuration files, as they determine what/how
things get monitored!
make install-webconf
- This installs the Apache config file for the Nagios
web interface

make install-exfoliation
- This installs the Exfoliation theme for the Nagios
web interface

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

*** Support Notes ***

If you have questions about configuring or running Nagios,
please make sure that you:
  - Look at the sample config files
  - Read the documentation on the Nagios Library at:
    https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:
  - What version of Nagios you are using
  - What version of the plugins you are using
  - Relevant snippets from your config files
  - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:
  https://support.nagios.com
***** Enjoy. *****

```

Run the following set of commands to ensure that **sudo**  
**make install**

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
rm -f /usr/local/nagios/share/map.html
rm -f /usr/local/nagios/share/rss-*
rm -f /usr/local/nagios/share/graph-header.html
rm -f /usr/local/nagios/share/histogram.html
rm -f /usr/local/nagios/share/histogram-form.html
rm -f /usr/local/nagios/share/histogram-graph.html
rm -f /usr/local/nagios/share/histogram-links.html
rm -f /usr/local/nagios/share/infobox.html
rm -f /usr/local/nagios/share/map.php
```

**sudo make install-init**

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
```

**sudo make install-config**

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

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

**sudo make install-webconf**

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Next, we are supposed to create a nagiosadmin account for nagios login along with password.

Specify the password twice. **sudo htpasswd -c**

**/usr/local/nagios/etc/htpasswd.users nagiosadmin**

```
[ec2-user@ip-172-31-41-160 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-172-31-41-160 nagios-4.5.5]$
```

Restart Apache **sudo service**

**httpd restart**

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

**cd ~/downloads tar zxvf nagios-plugins-2.4.11.tar.gz**

```
[> ec2-user@ip-172-31-41-160:~/downloads
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-41-160 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/ltdmain.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
```

Compile and install plugins **cd**

**nagios-plugins-2.4.11**

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

Run the following command: **sudo**

**chkconfig --add nagios** On

running the above command

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios  
error reading information on service nagios: No such file or directory
```

If this is the output that one is getting, then it means that the init script is missing... We can check this by running **ls /etc/init.d/**

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$ ls /etc/init.d/  
README functions  
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$
```

With **ls** command, we must see a file named **nagios**, which i was not able to see

If the Init Script is Missing i.e If you don't see the **nagios** script in **/etc/init.d/**, you can create it manually. Here's how:

Run the following command: **sudo**

**nano /etc/init.d/nagios**

Within this file, paste the following script

```
#!/bin/bash  
  
# nagios      Startup script for Nagios  
#  
# chkconfig: 345 99 10  
# description: Nagios is a host/service/network monitoring program  
# processname: nagios  
# pidfile: /var/run/nagios/nagios.pid  
case "$1" in start) echo "Starting  
Nagios..."  
    /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg  
    ;;  
stop)  
    echo "Stopping Nagios..." kill `cat /var/run/nagios/nagios.pid`  
    ;;  
restart)  
    $0 stop  
    $0 start  
    ;;
```

```

status)
ps aux | grep nagios
;;
*)
echo "Usage: $0 {start|stop|restart|status}"
exit 1
;;
esac
exit 0

```

```

ec2-user@ip-172-31-92-249:~/downloads/nagios-plugins-2.4.11$ nano /etc/init.d/nagios
GNU nano 5.8
#!/bin/bash
# nagios      Startup script for Nagios
#
# chkconfig: 345 99 10
# description: Nagios is a host/service/network monitoring program
# processname: nagios
# pidfile: /var/run/nagios/nagios.pid
#
# case "$1" in
#   start)
#     echo "Starting Nagios..."
#     /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
#     ;;
#   stop)
#     echo "Stopping Nagios..."
#     kill -9 `cat /var/run/nagios/nagios.pid`
#     ;;
#   restart)
#     $0 stop
#     $0 start
#     ;;
#   status)
#     ps aux | grep nagios
#     ;;
#   *)
#     echo "Usage: $0 {start|stop|restart|status}"
#     exit 1
#   ;;
# esac
#
# exit 0

```

Make the Script Executable: After saving the file, run the following command to make it executable: **sudo chmod +x /etc/init.d/nagios**

Run **sudo chkconfig --add nagios** again And then run **sudo chkconfig nagios on**

```

[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo nano /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chmod +x /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$

```

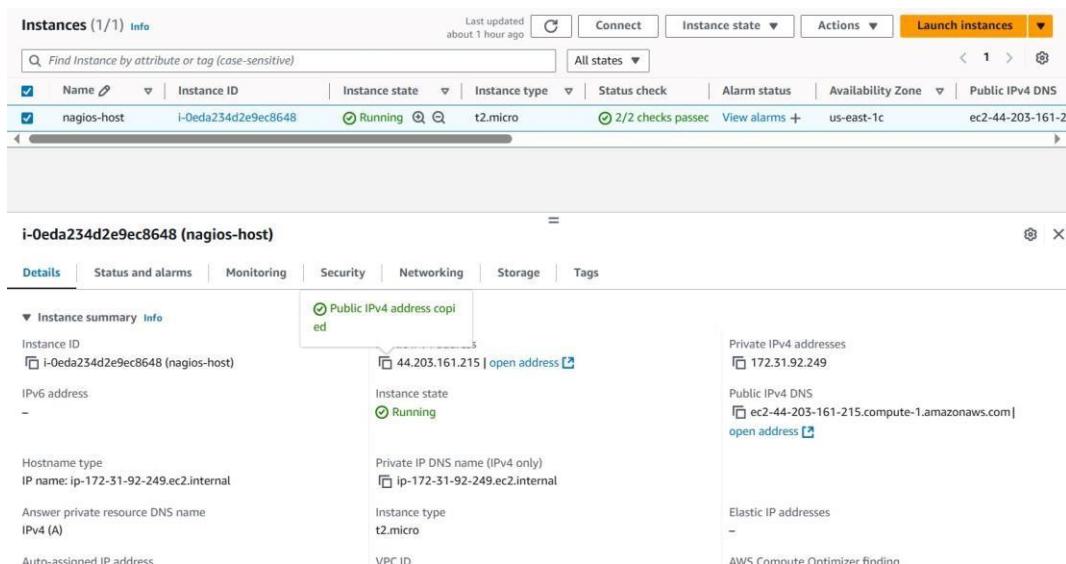
**sudo service nagios start**

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.11]$ sudo service nagios start
Starting Nagios...

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
Nagios 4.5.5 starting... (PID=72261)
Local time is Tue Oct 01 20:59:58 UTC 2024
wproc: Successfully registered manager as @wproc with query handler
wproc: Registry request: name=Core Worker 72265;pid=72265
wproc: Registry request: name=Core Worker 72264;pid=72264
wproc: Registry request: name=Core Worker 72263;pid=72263
wproc: Registry request: name=Core Worker 72262;pid=72262
Successfully launched command file worker with pid 72266
wproc: NOTIFY job 4 from worker Core Worker 72262 is a non-check helper but exited with return code 127
wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
```

Get your public IPv4 address from your instance. We will require it for connecting to our nginx server



Browse for this url: [http://<your\\_public\\_ip\\_address>/nagios](http://<your_public_ip_address>/nagios)

The browser may ask you for your nagios credentials which set in the earlier steps

The username is nagiosadmin and enter the password that you set earlier

The screenshot shows the Nagios Core 4.5.5 web interface. The top navigation bar indicates the URL as 34.229.45.75/nagios/. The main header features the Nagios logo with the tagline "Core™". A green checkmark icon with the text "Process running with PID 62668" is displayed. The left sidebar contains a navigation menu with sections like General, Current Status, Service Groups, Problems, Reports, and Configuration. The "Current Status" section is currently selected. The main content area includes a "Get Started" box with bullet points about monitoring infrastructure, changing the look, extending with addons, and getting support. It also features a "Quick Links" box with links to Nagios Library, Labs, Exchange, Support, and the official website. Below these are boxes for "Latest News" and "Don't Miss...". A vertical "Page Tour" button is located on the right side.

## Conclusion:

In this experiment, we successfully installed and configured Nagios Core on an Amazon Linux EC2 instance, showcasing its role in continuous monitoring within a DevOps environment. We learned about user management and service configuration, emphasizing Nagios's ability to monitor systems and networks effectively. This experience laid the groundwork for enhancing infrastructure reliability and integrating advanced monitoring strategies in future projects.

## Adv DevOps Exp 10

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

### Monitoring Using Nagios:

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

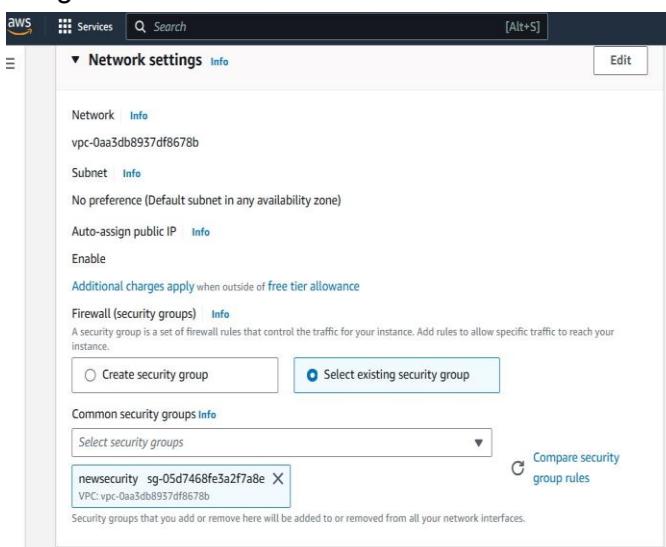
Run this command **sudo systemctl status**

```
ec2-user@ip-172-31-41-160:~/.downloads/nagios-plugins-2.4.11
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo systemctl status
● ip-172-31-41-160.ec2.internal
  State: running
    Units: 296 loaded (incl. loaded aliases)
      Jobs: 0 queued
     Failed: 0 units
      Since: Wed 2024-10-02 12:28:05 UTC; 33min ago
    systemd: 252.23-2.amzn2023
   Group: /
      └─init.scope
        └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
       └─system.slice
         └─acpid.service
           └─1938 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate-key --who=noah "---why=acpid instead" --mode=block /usr/sbin/acpid -f
             └─2059 /usr/sbin/acpid -f
           └─amazon-ssm-agent.service
             └─2141 /usr/bin/amazon-ssm-agent
           └─atd.service
             └─2152 /usr/sbin/atd -f
           └─audited.service
             └─1768 /sbin/audited
           └─chronynd.service
             └─2175 /usr/sbin/chronynd -F 2
           └─dbus-broker.service
             └─1946 /usr/bin/dbus-broker-launch --scope system --audit
               └─1954 dbus-broker --log 4 --controller 9 --machine-id ec2e4d759a3e2f6fe850b14e4cdacabe --max-bytes 536870912 --max-fds 4096 --max-matches 16384 --audit
             └─gssproxy.service
               └─1959 /usr/sbin/gssproxy -D
           └─httpd.service
             └─49553 /usr/sbin/httpd -DFOREGROUND
             └─49555 /usr/sbin/httpd -DFOREGROUND
             └─49556 /usr/sbin/httpd -DFOREGROUND
             └─49557 /usr/sbin/httpd -DFOREGROUND
             └─49558 /usr/sbin/httpd -DFOREGROUND
             └─62800 /usr/sbin/httpd -DFOREGROUND
           └─libstoragemgmt.service
             └─1940 /usr/bin/lsm -d
```

### Step 2: Before we begin,

To monitor a Linux machine, create an **Ubuntu 20.04 server** EC2 Instance in AWS.

Provide it with the **same security group** as the Nagios Host and name it 'nagios-client' alongside the host.



▼ Key pair (login) [Info](#)

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

Key pair name - *required*

[Create new key pair](#)

| Name          | Instance ID         | Instance state | Instance type | Status check      | Alarm status  | Availability Zone | Public IPv4 DNS   |
|---------------|---------------------|----------------|---------------|-------------------|---------------|-------------------|-------------------|
| nagios-host   | i-03facef442a77494d | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-east-1a        | ec2-34-229-45-75  |
| nagios-client | i-0b934b61f21351c1b | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-east-1a        | ec2-54-172-92-221 |

### Step 3: TO BE DONE IN THE Nagios-host TERMINAL

In the nagios-host terminal, run this command

**ps -ef | grep nagios**

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ ps -ef | grep nagios
ec2-user 63115 2315 0 13:03 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ ■
```

To become a root user, run '**sudo su**' and make two directories using the following commands. If one is running these commands in windows powershell, make sure that he/she copies it line by line as powershell might make an error while interpreting multiple lines **mkdir**

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

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

```
[ec2-user@ip-172-31-92-249 ~]$ sudo su
[root@ip-172-31-92-249 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-92-249 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-92-249 ec2-user]#
```

Copy the sample localhost.cfg file to linuxhost folder. Use the following mentioned command to achieve it **cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg**

**/usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg**

Open linuxserver.cfg using nano and make the following changes. This is a conf type file in which we will have to modify the configurations in way which will help us specify the hosts and clients to be monitored

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

**Changes to be made:**

1. Change the hostname to linux-server (EVERYWHERE ON THE FILE)
2. Change address to the public IP address of your LINUX CLIENT.
3. Change hostgroup\_name under hostgroup to linux-servers1

```
# HOST DEFINITION
#
#####
# Define a host for the local machine

define host {
    use          linux-server           ; Name of host template to use
                                         ; This host definition will inherit all variables that are defined
                                         ; in (or inherited by) the linux-server host template definition.

    host_name    linux-server
    alias        localhost
    address     54.172.92.226
}

#####

# Define an optional hostgroup for Linux machines

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

**IMP: Everywhere else on the file, change the hostname to linux-server instead of localhost.**

Open the Nagios Config file and add the following line

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

Add the following line in the file and save

**cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/**

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

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Verify the configuration files by running the following command

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

```
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-41-160 nagios-plugins-2.4.11]#
```

You are good to go if there are no errors.

Restart the nagios service **service**

**nagios restart**

And by running sudo systemctl status nagios, we can again check whether our server is running or not

```
root@ip-172-31-41-160:/tmp/nagios-plugins-2.4.11# sudo systemctl restart nagios
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status nagios
● nagios.service - Nagios Core 5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 13:20:17 UTC; 7s ago
       Docs: https://nagios.org/documentation
     Process: 78776 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 78777 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
      Main PID: 78778 (nagios)
        Tasks: 6 (limit: 1112)
       Memory: 4.0M
          CPU: 24ms
         CGroup: /system.slice/nagios.service
             ├─78778 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─78779 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─78780 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─78781 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─78782 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─78783 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: echo service query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: help for the query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: successfully registered manager @proc with query handler
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78782;pid=78782
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78781;pid=78781
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78780;pid=78780
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78779;pid=78779
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: Successfully launched command file worker with pid 78783
Oct 02 13:20:21 ip-172-31-41-160.ec2.internal nagios[78778]: HOST ALERT: linux-server:UP;SOFT;1;PING OK - Packet loss = 0%, RTA = 0.93 ms
Oct 02 13:20:24 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.0
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Docs: man:httdp.service(8)
   Main PID: 49553 (httpd)
     Status: "total requests: 26; Idle/Busy workers 100/0;Requests/sec: 0.0129; Bytes served/sec: 94.0/sec"
       Tasks: 230 (limit: 1112)
      Memory: 1.79M
        CPU: 1.416s
       CGroup: /system.slice/httpd.service
           ├─49553 /usr/sbin/httdp -DFOREGROUND
```

## Step 4: TO BE DONE IN THE Nagios-client TERMINAL

Now it is time to switch to the client machine.

SSH into the machine or simply use the EC2 Instance Connect feature.

```
PS C:\WINDOWS\system32> cd C:\Users\DEll\Downloads
PS C:\Users\DEll\Downloads> ssh -i "mohit.pem" ubuntu@ec2-54-172-92-226.compute-1.amazonaws.com
The authenticity of host 'ec2-54-172-92-226.compute-1.amazonaws.com (54.172.92.226)' can't be established.
ECDSA key fingerprint is SHA256:e/WkFQRuhSpqjq5hMaA0dku8msNhETN9S4gZEy53E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-172-92-226.compute-1.amazonaws.com,54.172.92.226' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

System information as of Wed Oct  2 13:26:11 UTC 2024

System load:  0.0              Processes:          104
Usage of /:   22.8% of 6.71GB   Users logged in:    0
Memory usage: 20%              IPv4 address for enx0: 172.31.36.100
Swap usage:   0%

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

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
```

Make a package index update and install gcc, nagios-nrpe-server and the plugins. Run the following commands to achieve the same.

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



Open nrpe.cfg file to make changes.

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

Under allowed\_hosts, add your nagios host IP address like so

```
ubuntu@ip-172-31-36-100: ~
GNU nano 7.2
#
# 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=127.0.0.1,34.229.45.75

#
# 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
```

Now restart the NRPE server by this command.

**sudo systemctl restart nagios-nrpe-server**

```
ubuntu@ip-172-31-36-100: ~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-36-100: ~$
```

Run the following command in the Nagios-host terminal

**sudo systemctl status nagios**

```
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-02 13:20:17 UTC; 15min ago
     Docs: https://www.nagios.org/documentation
 Main PID: 78778 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 4.3M
      CPU: 403ms
     CGroup: /system.slice/nagios.service
             └─78778 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL - 0% free (0 MB out of 0 MB)
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: NOTIFY job 3 from worker Core Worker 78782 is a non-check helper but exited with return code 127
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Oct 02 13:23:13 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Total Processes:OK;HARD:1;PROCS OK: 37 processes with STATE = RZDT
Oct 02 13:23:50 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Load:OK;HARD:1;OK - load average: 0.01, 0.07, 0.04
Oct 02 13:24:28 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Users:OK;HARD:1;USERS OK - 2 users currently logged in
Oct 02 13:24:46 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost;Current Users:OK;HARD:1;USERS OK - 2 users currently logged in
lines 1-26/26 (END)
```

**Step 5: Visiting your nagios server using your nagios-host ip address** Open up your browser and look for [http://<public\\_ip\\_address\\_of\\_nagios-host>/nagios](http://<public_ip_address_of_nagios-host>/nagios)

The screenshot shows the Nagios Core 4.5.5 dashboard. The top right corner displays the text "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". Below this, there are three main sections: "Get Started" (with a bulleted list of steps), "Latest News" (empty), and "Don't Miss..." (empty). To the right, a "Quick Links" sidebar lists several Nagios-related resources. On the left, a vertical navigation menu under "Current Status" includes links for "Tactical Overview", "Map", "Hosts", "Services", "Host Groups", "Problems", "Reports", and "Event Log".

Click on Hosts.

This screenshot focuses on the "Hosts" section of the Nagios Core 4.5.5 dashboard. It displays "Host Status Totals" with counts for Up (2), Down (0), Unreachable (0), Pending (0), and "Service Status Totals" with counts for Ok (12), Warning (1), Unknown (0), Critical (3), and Pending (0). Below these, a table titled "Host Status Details For All Host Groups" lists two hosts: "linux-server" and "localhost", both marked as UP. A red vertical bar on the right side of the screen has a "Page Tour" label.

Click on linux-server to view host information

The screenshot shows the Nagios web interface for the host 'localhost'. The 'Host Information' section displays details like last update, check interval, and configuration source. The 'Host State Information' section shows the host is 'UP' with a status message and various performance metrics. The 'Host Commands' section lists various actions that can be performed on the host, such as enabling or disabling checks. The 'Host Comments' section allows for adding comments to the host entry.

We can even navigate to the services section, which explicitly mentions the status, duration, checks, information about the numerous services present on our hosts

The screenshot shows the Nagios web interface displaying service status details for all hosts. It includes sections for current network status, host status totals, and service status totals. The main table lists various services across hosts, showing their status (e.g., OK, CRITICAL, WARNING), last check time, duration, and attempt count. A detailed view of the 'Service Status Details For All Hosts' table is provided, showing specific service details for each host.

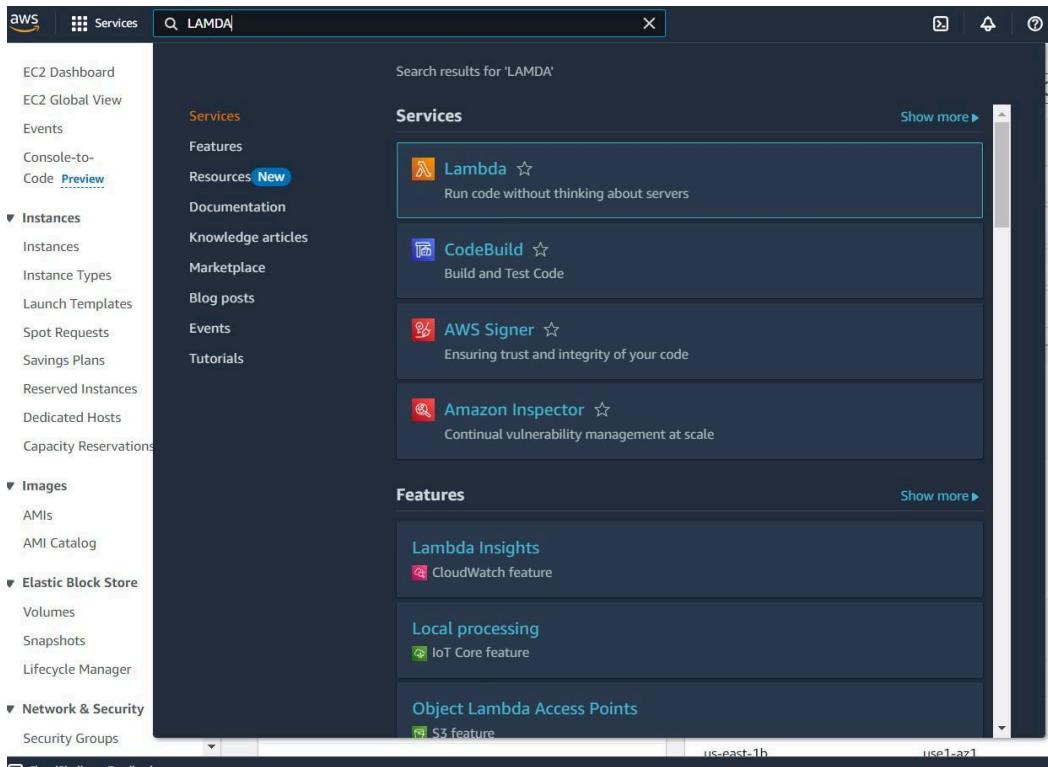
**Conclusion:** In conclusion, the experiment focused on monitoring ports, services, and a Linux server using Nagios. Through the step-by-step process, we successfully configured Nagios to monitor essential network services on the Linux server. By setting up both the Nagios host and client, we were able to track system performance, ensure service availability, and monitor key metrics like CPU and memory usage.

## Experiment 11

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

### Step 1: Accessing AWS

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



### Step 2: Creating a New Lambda Function

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

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

**How it works**

.NET | Java | **Node.js** | Python | Ruby | Custom runtime

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

Run | Next: Lambda responds to events

### Step 3: Configuring Basic Settings

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

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.

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

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

**Architecture** Info  
Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

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

**▶ Change default execution role**

**Permissions** [Info](#)

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

▼ Change default execution role

Execution role

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

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

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

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

▶ Additional Configurations

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

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

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

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

**Code source** [Info](#) Upload from ▾

File Edit Find View Go Tools Window Test ▾ Deploy ✖

Environment Go to Anything (Ctrl-P)

λ **lambda\_function** Environment Var +

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

## Step 4: Testing the Function

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

The screenshot shows the AWS Lambda console interface for testing a function. At the top, there's a "Test event" section with "Info" and "Save" buttons. Below it, a note says "To invoke your function without saving an event, configure the JSON event, then choose Test." Under "Test event action", "Create new event" is selected. In the "Event name" field, "MyEventName" is entered. The "Event sharing settings" section has "Private" selected. The "Template - optional" dropdown shows "hello-world". The "Event JSON" section contains the following JSON:

```
1  {
2      "key1": "value1",
3      "key2": "value2",
4      "key3": "value3"
5 }
```

At the bottom, the "Code source" tab is active, showing the "lambda\_function" file with the following Python code:

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

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

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

## Step 5: Running the Test

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

The screenshot shows the AWS Lambda function editor interface after running a test. The 'Test' button is now highlighted. The 'Execution result' tab is selected. The 'Test Event Name' dropdown shows 'demo1'. The 'Response' section displays the JSON output: { "statusCode": 200, "body": "\\"Hello this is Exp 11\\\""}'. The 'Function Logs' section shows the request and response details. A status bar at the bottom indicates 'Status: Succeeded | Max memory used: 32 MB | Time: 1.62 ms'. A green success message at the bottom of the screen says 'Successfully updated the function lambda\_demo.'

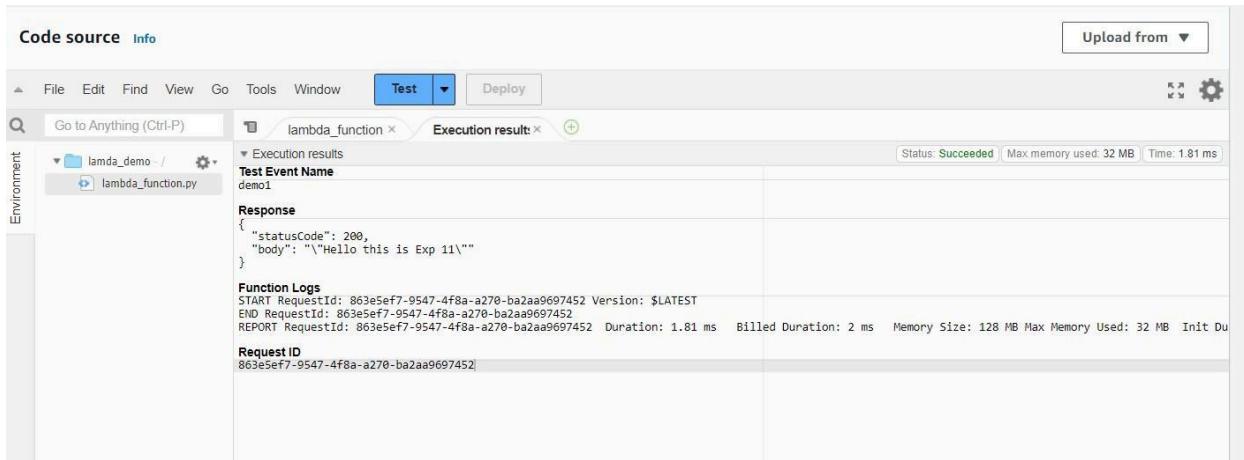
## Step 6: Editing and Deploying the Code

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



## Step 7: Final Testing

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



The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, a 'Test' button, and a 'Deploy' button. To the right of the 'Test' button is an 'Upload from' dropdown. On the left, there's a sidebar labeled 'Environment' with a dropdown menu showing 'lambda\_demo /' and 'lambda\_function.py'. The main area has tabs for 'Execution results' and 'Execution result'. Under 'Execution result', it shows 'Test Event Name: demo1'. Below that is a 'Response' section with the following JSON:

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

Under 'Function Logs', it shows the following log entries:

```
START RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Version: $LATEST
END RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452
REPORT RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Duration: 1.81 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 0 ms
RequestID: 863e5ef7-9547-4f8a-a270-ba2aa9697452
```

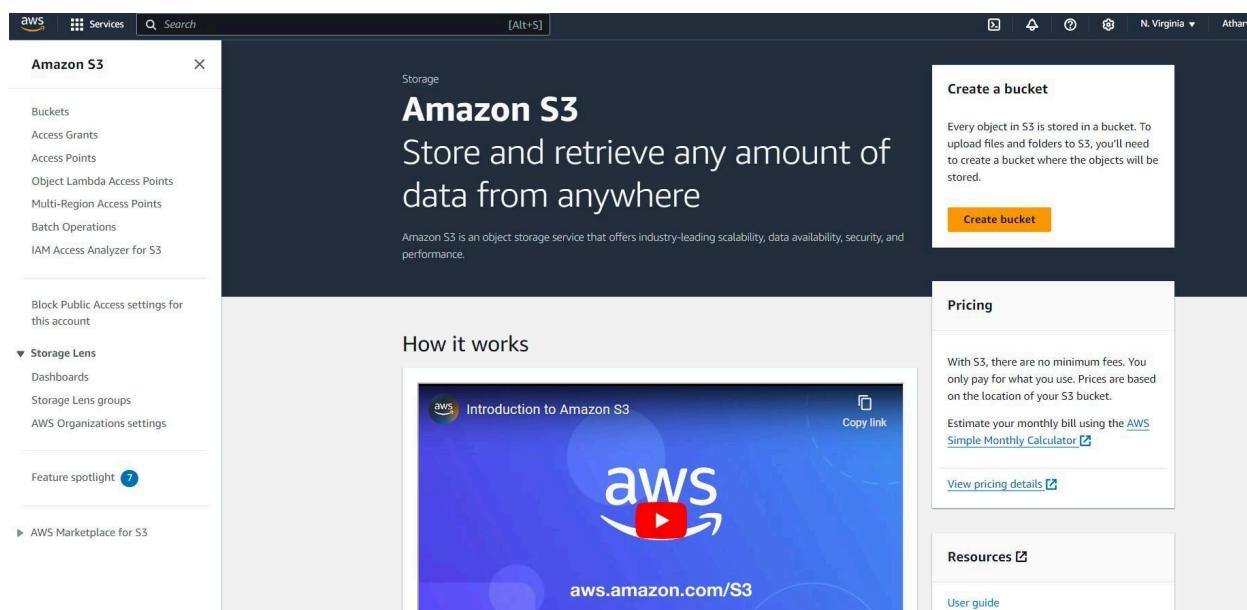
## Conclusion:

In this experiment, I created and tested your first AWS Lambda function using Python. I learned to navigate the AWS Management Console, configure basic settings, and modify function's code. This experience highlights the ease of deploying serverless applications with Lambda, allowing you to focus on coding rather than infrastructure management. I now have a foundational understanding to explore more complex serverless solutions and integrate AWS services for greater functionality.

## Experiment 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.

**Step 1: Create a s3 bucket. 1) Search for S3 bucket in the services search. Then click on create bucket.**



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

**General configuration**

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

**Bucket type** [Info](#)

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

**Directory**  
 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](#)

nishantbucket24

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

**Copy settings from existing bucket - optional**  
 Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

Format: s3://bucket/prefix

---

**Object Ownership** [Info](#)

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

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

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

**Object Ownership** [Info](#)

Bucket owner enforced

## Block Public Access settings for this bucket

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

### **Block all public access**

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

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

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

#### **Block public access to buckets and objects granted through any access control lists (ACLs)**

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

#### **Block public access to buckets and objects granted through new public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

#### **Block public and cross-account access to buckets and objects through any public bucket or access point policies**

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

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

The screenshot shows the AWS S3 Buckets page. At the top, a green banner indicates that a bucket named "s3lamdaexp11" has been successfully created. Below the banner, there's an "Account snapshot" section with an update frequency of "updated every 24 hours". The main area displays two buckets: "elasticbeanstalk-eu-north-1-010928207735" and "s3lamdaexp11". The "s3lamdaexp11" bucket is highlighted with a blue border. A "Create bucket" button is visible at the top right of the table header. The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date.

| Name   | AWS Region                      | IAM Access Analyzer                          | Creation date                         |
|--|---------------------------------|--|---------------------------------------|
| <a href="#">elasticbeanstalk-eu-north-1-010928207735</a> | Europe (Stockholm) eu-north-1   | <a href="#">View analyzer for eu-north-1</a> | August 14, 2024, 22:12:26 (UTC+05:30) |
| <a href="#">s3lamdaexp11</a>                             | US East (N. Virginia) us-east-1 | <a href="#">View analyzer for us-east-1</a>  | October 7, 2024, 09:40:50 (UTC+05:30) |

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

The screenshot shows the AWS S3 console. The URL is [Amazon S3 > Buckets > s3lamdaexp11](#). The bucket name is [s3lamdaexp11](#). The page displays the following interface:

- Top navigation: Objects, Properties, Permissions, Metrics, Management, Access Points.
- Toolbar: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload.
- Search bar: Find objects by prefix.
- Table headers: Name, Type, Last modified, Size, Storage class.
- Message: No objects. You don't have any objects in this bucket.
- Upload button: [Upload](#).

The screenshot shows the AWS S3 Upload interface. The URL is [Amazon S3 > Buckets > s3lamdaexp11 > Upload](#). The page displays the following interface:

- Section title: Upload [Info](#).
- Text: 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](#).
- Dropzone: Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
- Section title: Files and folders (1 Total, 990.9 KB).
- Table:
  - Header: All files and folders in this table will be uploaded.
  - Row: football.jpg
- Buttons: Remove, Add files, Add folder.
- Search bar: Find by name.
- Table headers: Name, Folder.
- Section title: Destination [Info](#).
- Text: Destination: <s3://s3lamdaexp11>.
- Section title: Destination details.
- Text: Bucket settings that impact new objects stored in the specified destination.
- Section title: Permissions.
- Text: Grant public access and access to other AWS accounts.
- Section title: Properties.
- Text: Specify storage class, encryption settings, tags, and more.

**6) The image has been uploaded to the bucket.**

⌚ Upload succeeded  
View details below.

**Upload: status**

The information below will no longer be available after you navigate away from this page.

**Summary**

|   |   |                               |
|---|---|-------------------------------|
| Destination<br><a href="#">s3://s3lambdaexp11</a> | Succeeded<br>⌚ 1 file, 990.9 KB (100.00%) | Failed<br>⌚ 0 files, 0 B (0%) |
|---|---|-------------------------------|

**Files and folders** Configuration

**Files and folders (1 Total, 990.9 KB)**

| Find by name |        |            |          |             |       |         |
|--------------|--------|------------|----------|-------------|-------|---------|
| Name         | Folder | Type       | Size     | Status      | Error | Actions |
| football.jpg | -      | image/jpeg | 990.9 KB | ⌚ Succeeded | -     |         |

## Step 2: Configure Lambda function

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

Lambda > Functions > Create function

**Create function** Info

Choose one of the following options to create your function.

- Author from scratch Start with a simple Hello World example.
- Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.
- Container image Select a container image to deploy for your function.

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

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

**Architecture** Info  
Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

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

Change default execution role

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

Lambda > Functions > lamdaexp12

lamdaexp12

Throttle Copy ARN Actions ▾

Function overview Info

Diagram Template

lamdaexp12

Layers (0)

+ Add trigger + Add destination

Description

Last modified 22 seconds ago

Function ARN arn:aws:lambda:us-east-1:010928207735:function:lamdaexp12

Function URL Info

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

Lambda > Add triggers

## Add trigger

Trigger configuration Info

S3 aws asynchronous storage

Lambda > Add triggers

## Add trigger

Trigger configuration Info

S3 aws asynchronous storage

**Bucket**  
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.  
s3/s3lamdaexp11

Bucket region: us-east-1

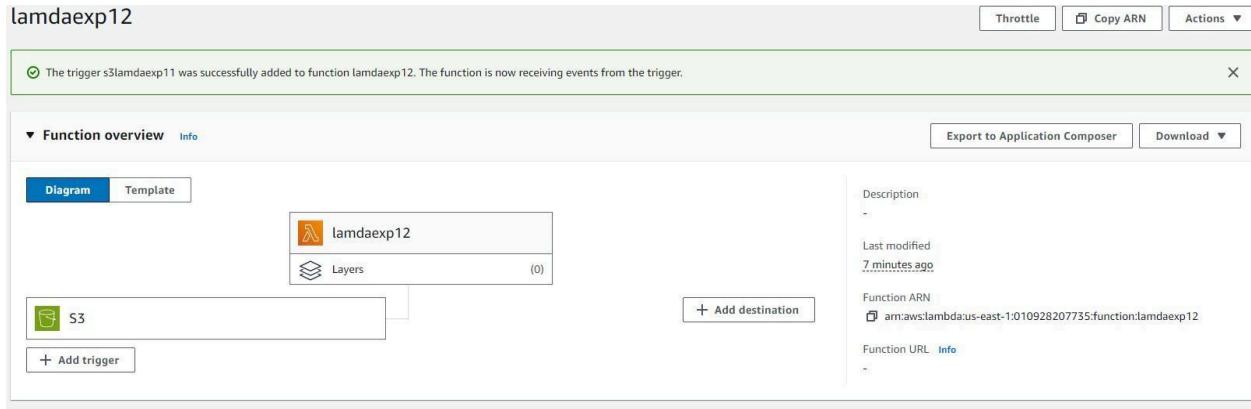
**Event types**  
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any **special characters** must be URL encoded.  
e.g. images/

**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any **special characters** must be URL encoded.  
e.g. .jpg

**Recursive invocation**  
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)



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

```

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

  // Extract bucket name and object key from the event
  const record = event.Records[0];
  const bucketName = record.s3.bucket.name;
  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys

  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
  console.log(`Event Source: ${record.eventSource}`);

  return {
    statusCode: 200,
    body: JSON.stringify('Log entry created successfully!')
  };
}

```

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

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is selected), 'Deploy', and 'Changes not deployed'. A 'Code source' tab is active, and an 'Info' tab is visible. On the left, there's an 'Environment' sidebar with a dropdown set to 'lambdaexp12' and a file tree showing 'index.mjs'. The main code editor area contains the following JavaScript code:

```

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

```

The bottom right corner of the editor shows '22:3 JavaScript Spaces: 2'.

The screenshot shows the AWS Lambda function editor interface, similar to the first one but with a different configuration. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is selected), 'Deploy', and 'Changes not deployed'. A 'Code source' tab is active, and an 'Info' tab is visible. On the left, there's an 'Environment' sidebar with a dropdown set to 'lambdaexp12' and a file tree showing 'index.mjs'. The main code editor area contains the same JavaScript code as the first screenshot. Additionally, a 'Configure test event' button is visible above the code editor. The bottom right corner of the editor shows '22:3 JavaScript Spaces: 2'.

Private

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

Shareable

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

Template - optional

hello-world

[Format JSON](#)

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

1:1 JSON Spaces: 2

[Cancel](#)

[Invoke](#)

[Save](#)

The test event myevent1 was successfully saved.

[Function URL](#) [Info](#)

### Step 3: Check the logs

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

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

**Services**

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

**Features**

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

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

CloudWatch > Log groups

**Log groups (1)**  
By default, we only load up to 10000 log groups.

Actions ▾ View in Logs Insights Start tailing

Filter log groups or try prefix search  Exact match < 1 > ⚙

| <input type="checkbox"/> Log group      | ▼ | Log class | ▼ | Anomaly detection         | ▼ | Data protection | ▼ | Sensitive data protection | ▼ | Retention                    | ▼ | Metric filters | ▼ | Contributor Insights |
|---|---|-----------|---|---------------------------|---|-----------------|---|---------------------------|---|------------------------------|---|----------------|---|----------------------|
| <a href="#">/aws/lambda/lambdaexp12</a> |   | Standard  |   | <a href="#">Configure</a> |   | -               | - | -                         | - | <a href="#">Never expire</a> |   | -              |   | -                    |

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

CloudWatch > Log groups > /aws/lambda/lamdaexp12

## /aws/lambda/lamdaexp12

Actions ▾ View in Logs Insights Start tailing Search log group

### Log group details

|                       |  |                            |   |                      |           |
|-----------------------|--|----------------------------|---|----------------------|-----------|
| Log class <b>Info</b> | Standard   | Stored bytes               | - | KMS key ID           | -         |
| ARN                   | arn:aws:logs:us-east-1:010928207735:log-group:/aws/lambda/lamdaexp12:* | Metric filters             | 0 | Anomaly detection    | Configure |
| Creation time         | 3 minutes ago  | Subscription filters       | 0 | Data protection      | -         |
| Retention             | Never expire   | Contributor Insights rules | - | Sensitive data count | -         |

Log streams Tags Anomaly detection Metric filters Subscription filters Contributor Insights Data protection

### Log streams (1)

Filter log streams or try prefix search  Exact match Show expired Info

Log stream  Last event time

2024/10/07/[\$LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00 2024-10-07 04:34:00 (UTC)

Create log stream Search all log streams

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

| CloudWatch > Log groups > /aws/lambda/lambda-p12 > 2024/10/07/[SLATEST]29f700136a7b40a2b79269533ea0969  |   |               |    |
|---|---|---------------|----|
| Log events  |   | Actions ▾     |    |
| You can use the filter bar below to search for and match terms, phrases, or values in your log events. <a href="#">Learn more about filter patterns</a> |   | Start tailing |    |
| <input type="text"/> Filter events - press enter to search  |   | Clear         | 1m |
| 30m 1h 12h Custom UTC timezone Display  |   |               |    |
| Timestamp   | Message   |               |    |
| No older events at this moment. <a href="#">Retry</a>   |   |               |    |
| 2024-10-07T04:45:00,475Z  | DLL_START Runtime Version: nodejs18-v8-19 Runtime Version: 4000 minvias:2minvias:exit-2> runtime:adde23be251ef04c5329e383824c4d4fc1ce1a4796d053399f980140001342f7   |               |    |
| 2024-10-07T04:45:00,412Z  | START RequestId: 9bbefab0-c4fe-40d9-a0e0-3e5140fb70fc Version: SLATEST  |               |    |
| 2024-10-07T04:45:00,432Z  | END RequestId: 9bbefab0-c4fe-40d9-a0e0-3e5140fb70fc   |               |    |
| 2024-10-07T04:45:00,432Z  | REPORT RequestId: 9bbefab0-c4fe-40d9-a0e0-3e5140fb70fc Duration: 10.74 ms Billed Duration: 10 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 13.57 ms |               |    |
| 2024-10-07T04:46:28,392Z  | START RequestId: 3cb6e10-4348-4ee5-9c87-e002259490fe Version: SLATEST   |               |    |
| 2024-10-07T04:46:28,394Z  | END RequestId: 3cb6e10-4348-4ee5-9c87-e002259490fe  |               |    |
| 2024-10-07T09:04:40,334Z  | [2024-10-07T09:04:40,1042 91711939-7136-4110-a650-12101515403] INFO An image has been added to the bucket example-bucket! test/key                                  |               |    |
| 2024-10-07T04:46:54,348Z  | START RequestId: e7d69c06-2818-4c39-9ead-5ed030574479 Version: SLATEST  |               |    |
| 2024-10-07T04:46:54,258Z  | END RequestId: e7d69c06-2818-4c39-9ead-5ed030574479   |               |    |
| 2024-10-07T04:46:54,258Z  | REPORT RequestId: e7d69c06-2818-4c39-9ead-5ed030574479 Duration: 1.46 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB                           |               |    |
| REPORT RequestId: a70ebc18-18b8-4c10-9add-5a1035798073 Duration: 1.48 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB               |   |               |    |
| No newer events at this moment. Auto-retry paused. <a href="#">Resume</a>   |   |               |    |

**Conclusion:**

In this experiment, In addition to demonstrating the integration of AWS Lambda with S3, this experiment showcases the scalability and flexibility of serverless architectures. By leveraging these services, we can build applications that respond in real-time to changes in data, such as the addition of files to S3 buckets, without the need for managing underlying server infrastructure. This not only enhances efficiency but also reduces operational costs, allowing developers to focus on building features rather than maintaining systems. Furthermore, the ability to log and monitor events through CloudWatch opens opportunities for further automation and analytics, paving the way for more complex workflows and data processing solutions.