

Name: Shreyash Kamat

Div: D15C

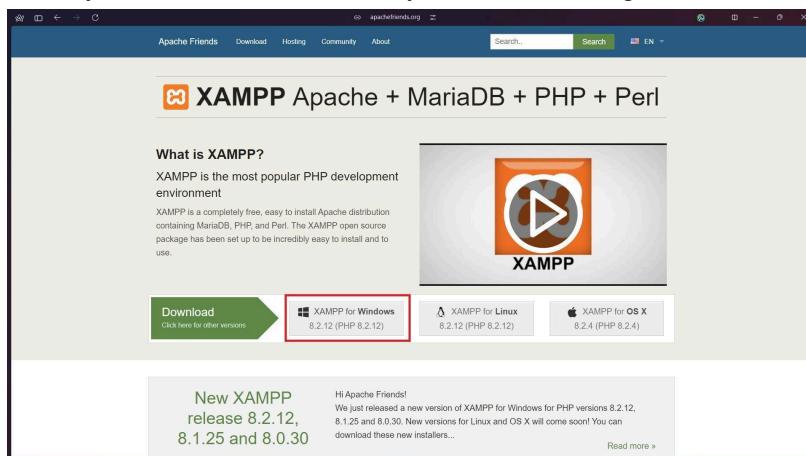
Roll no: 22

Static Hosting:

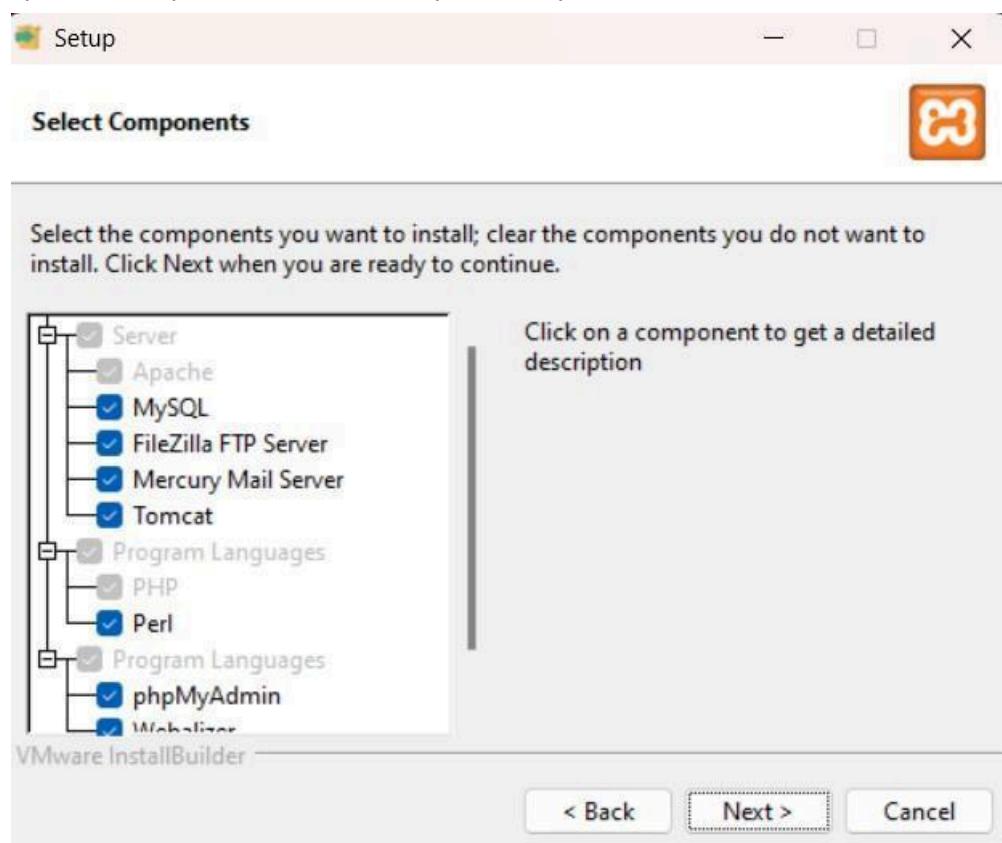
1) On local server (XAMPP)

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

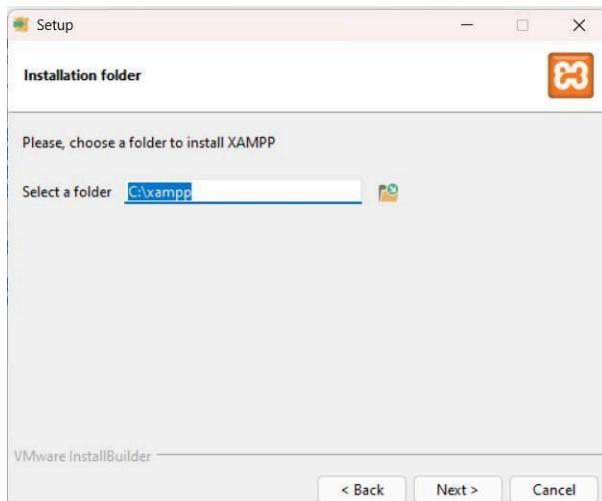
- 1) Select your OS. It will automatically start downloading.



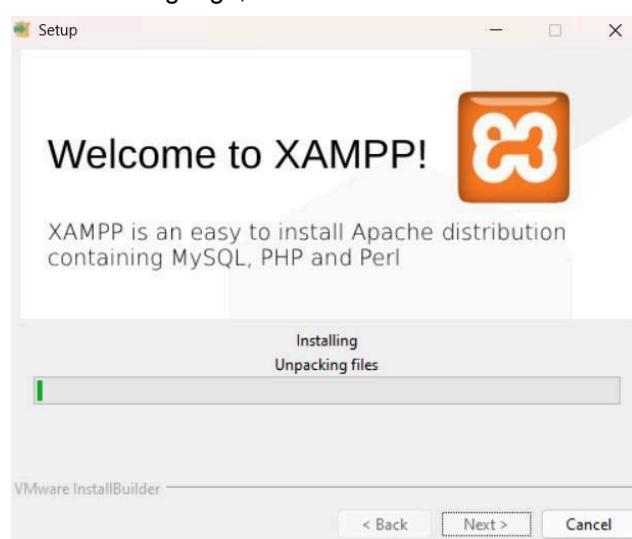
- 2) Open the setup file. Select all the required components and click next



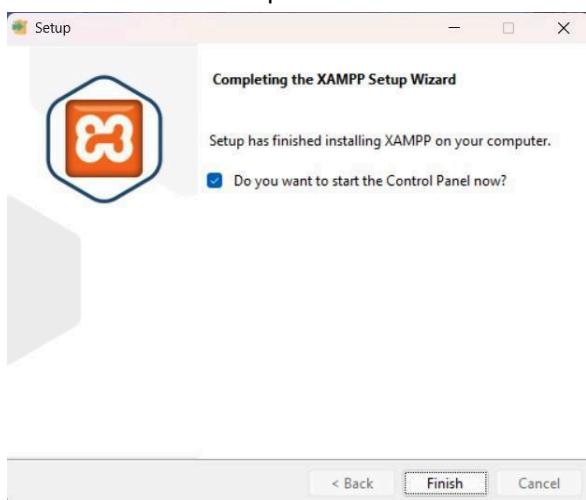
- 3) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



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



- 5) The installation is complete. Click Finish



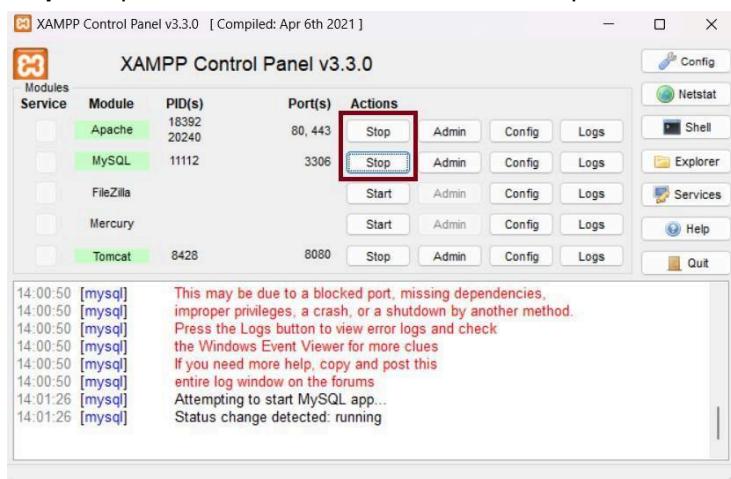
Step 2: Setup a file that is to be hosted on the server. Make sure the file has extension .php

test1	06-08-2024 22:48	PHP Source File	1 KB
-------	------------------	-----------------	------

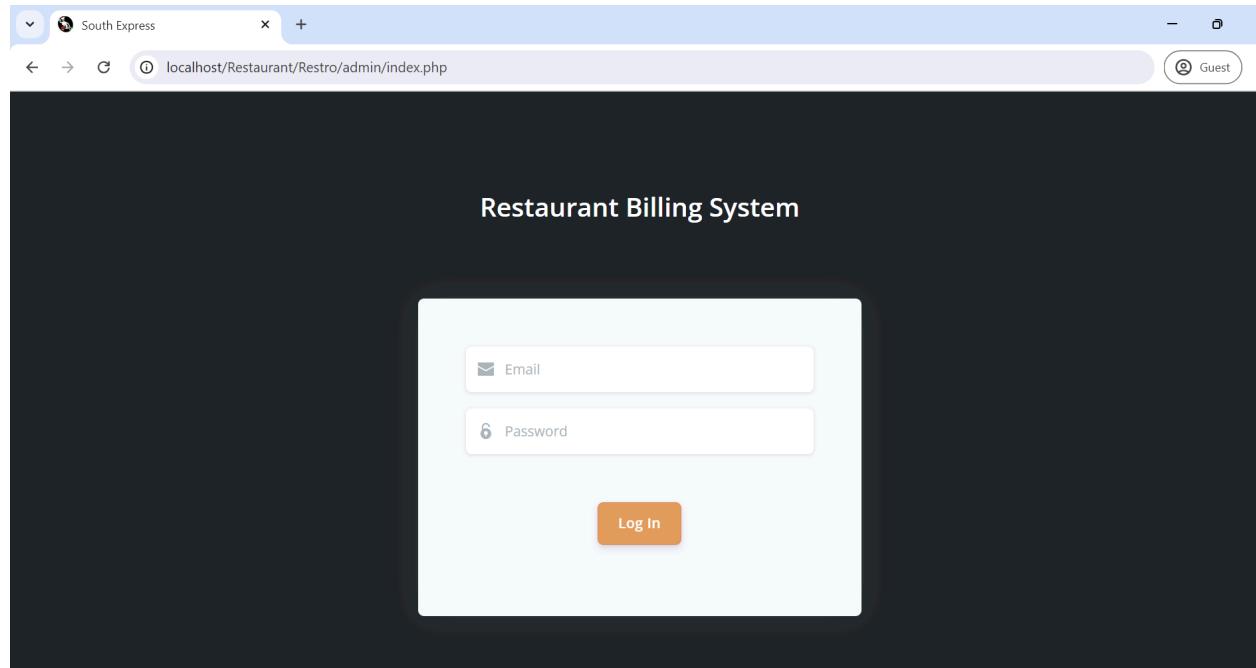
Step 3: Go to the directory where XAMPP was installed. Go to **htdocs** folder. Place your folder in this directory.

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
test1	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.



2) AWS S3

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

A screenshot of the AWS Management Console. The top navigation bar includes the AWS logo, a "Services" dropdown, a search bar, and user information. Below the search bar is a "Console Home" button and an "All services" link. The main content area is a grid of service icons and names. The "Storage" section is expanded, showing services like S3, EFS, FSx, S3 Glacier, Storage Gateway, and AWS Backup. Other sections visible include Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder, AWS App Runner, AWS SimSpace Weaver, Governance, AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, AWS Config, OpsWorks, Service Catalog, Systems Manager, Trusted Advisor, Control Tower, AWS Well-Architected Tool, AWS Chatbot, Launch Wizard, AWS Compute Optimizer, Resource Groups & Tag Editor, Amazon Grafana, Amazon Prometheus, AWS Resilience Hub, Incident Manager, AWS License Manager, Service Quotas, AWS Proton, Secrets Manager, GuardDuty, Amazon Inspector, Amazon Macie, IAM Identity Center, Certificate Manager, Key Management Service, CloudHSM, Directory Service, WAF & Shield, AWS Firewall Manager, AWS Artifact, Detective, AWS Signer, AWS Private Certificate Authority, Security Hub, AWS Audit Manager, Security Lake, Amazon Verified Permissions, AWS Payment Cryptography, IAM, and Cloud Financial.

Step 2: Click on Create Bucket

The screenshot shows the Amazon S3 landing page. At the top right, there is a call-to-action box titled "Create a bucket" with the sub-instruction: "Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored." Below this, there is a "Pricing" section stating: "With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket." At the bottom left, there is a "How it works" section with a thumbnail image labeled "Introduction to Amazon S3". The footer contains standard AWS links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

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

The screenshot shows the "General configuration" step of the bucket creation wizard. It includes fields for "Bucket name" (set to "statichosting22"), "AWS Region" (set to "US East (N. Virginia) us-east-1"), and "Bucket type" (set to "General purpose"). There are two options: "General purpose" (selected) and "Directory - New". The "Bucket name" field has a "Info" link. Below it, there is a note about uniqueness and naming rules, a "Copy settings from existing bucket - optional" section, and a "Choose bucket" button. The footer contains standard AWS links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

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

The screenshot shows the "Buckets" list in the Amazon S3 console. It displays one bucket named "statichosting22" under the "General purpose buckets" tab. The bucket details include its ARN, creation date (August 7, 2024), and a "View analyzer for us-east-1" link. There are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". The footer contains standard AWS links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS S3 console. In the top navigation bar, 'Amazon S3 > Buckets > statichosting22'. The main page displays the bucket 'statichosting22' with the status 'Info'. Below the status, there's a section titled 'Objects (0) Info' with a search bar and various actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A note says 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)'. There's also a 'Find objects by prefix' input field and a pagination control showing page 1 of 1. At the bottom, there's an 'Upload' button and a note 'No objects. You don't have any objects in this bucket.' The footer includes CloudShell, Feedback, and links to 2024, Privacy, Terms, and Cookie preferences.

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

The screenshot shows the 'Properties' tab of the AWS S3 bucket 'statichosting22'. It includes sections for Object Lock (Disabled), Requester pays (Disabled), and Static website hosting. The Static website hosting section has an 'Edit' button. A note says 'Use this bucket to host a website or redirect requests. [Learn more](#)'. Below it, 'Static website hosting' is set to 'Disabled'. The footer includes CloudShell, Feedback, and links to 2024, Privacy, Terms, and Cookie preferences.

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' configuration for the Static website hosting. Under 'Static website hosting', the 'Enable' radio button is selected. Under 'Hosting type', the 'Host a static website' radio button is selected, with a note: 'Use the bucket endpoint as the web address. [Learn more](#)'. The 'Redirect requests for an object' radio button is also present. A callout box provides information about making content publicly readable via S3 Block Public Access. Under 'Index document', the value 'index.html' is specified. The footer includes CloudShell, Feedback, and links to 2024, Privacy, Terms, and Cookie preferences.

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

The screenshot shows the AWS S3 console with the 'Objects' tab selected. The main area displays a message: 'No objects' and 'You don't have any objects in this bucket.' At the bottom right of the main area is a large orange 'Upload' button. Above the main area, there is a toolbar with various actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

Step 8: Click on Add files. Add all the files you want to upload. Then scroll down and click on Upload

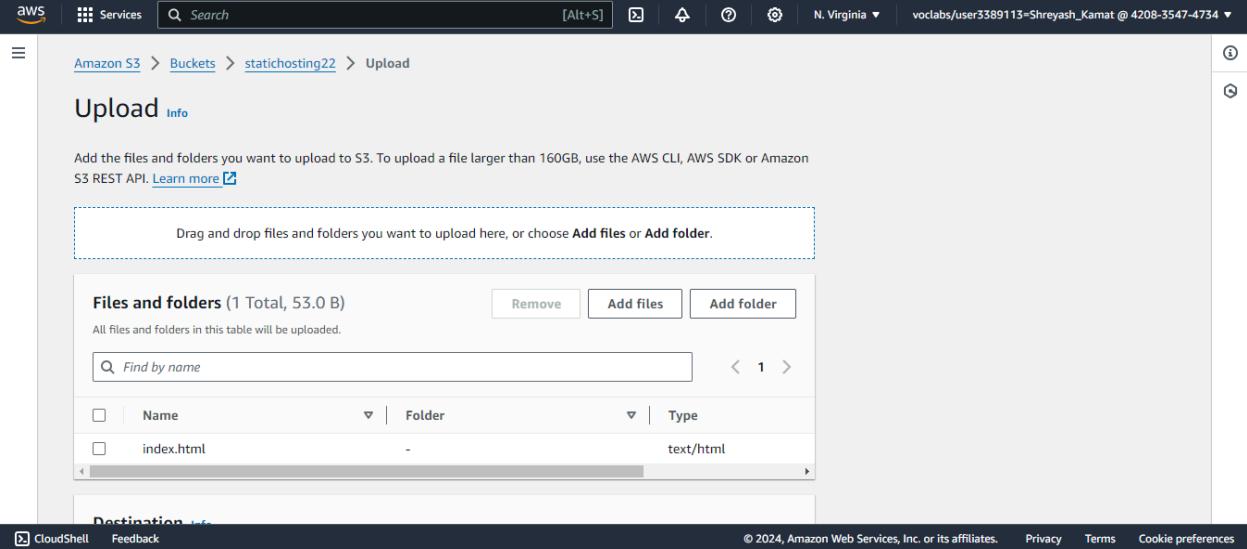
The screenshot shows the AWS S3 'Upload' screen. A dashed blue box highlights the 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' area. Below this, a table lists 'Files and folders (1 Total, 53.0 B)' with one item: 'index.html' (text/html). At the bottom of the screen, there is a 'Destination' section and a 'CloudShell' link.

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 'Properties' screen. Under the 'Static website hosting' section, the 'Bucket website endpoint' field is highlighted with a red border. It contains the URL 'http://statichosting27.s3-website-us-east-1.amazonaws.com'. Above this field, there is a note: 'When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket.' and a 'Learn more' link.

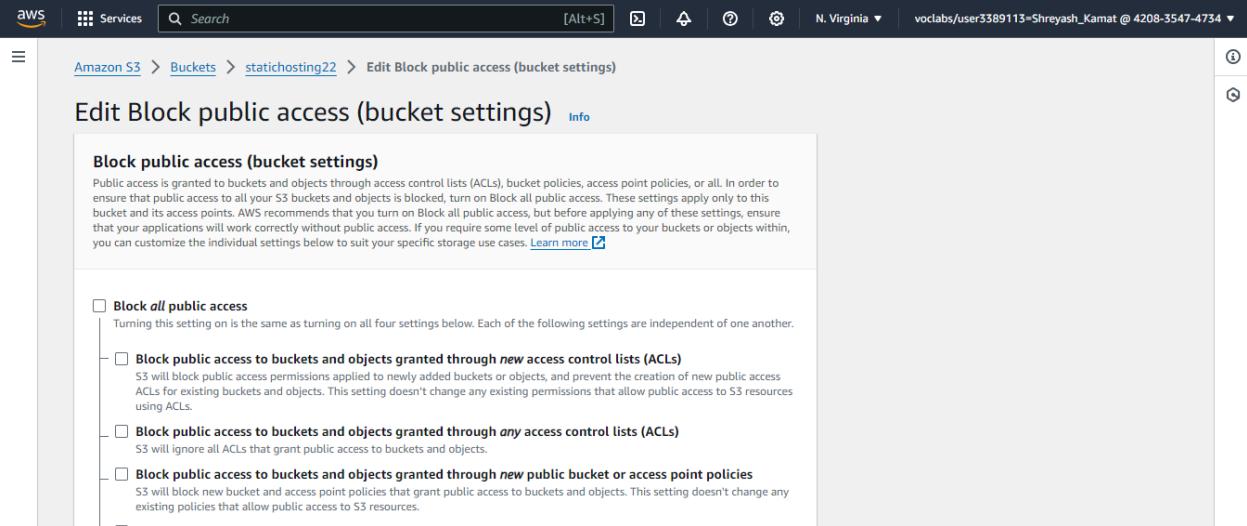
Step 10: Open the link. It will show a 403 forbidden error screen as the contents of the bucket are not

available for the public users. To change this, go to Permissions tab, go to Block public access and click on edit



The screenshot shows the AWS S3 'Upload' interface. At the top, there's a search bar and navigation links for 'Amazon S3 > Buckets > statichosting22 > Upload'. Below the navigation is a large central area with a dashed border for dragging and dropping files. A message says 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more.' Below this is a table titled 'Files and folders (1 Total, 53.0 B)' containing one item: 'index.html' (text/html). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name' is also present. At the bottom, there's a 'Destination' section and a footer with links like 'CloudShell', 'Feedback', and copyright information.

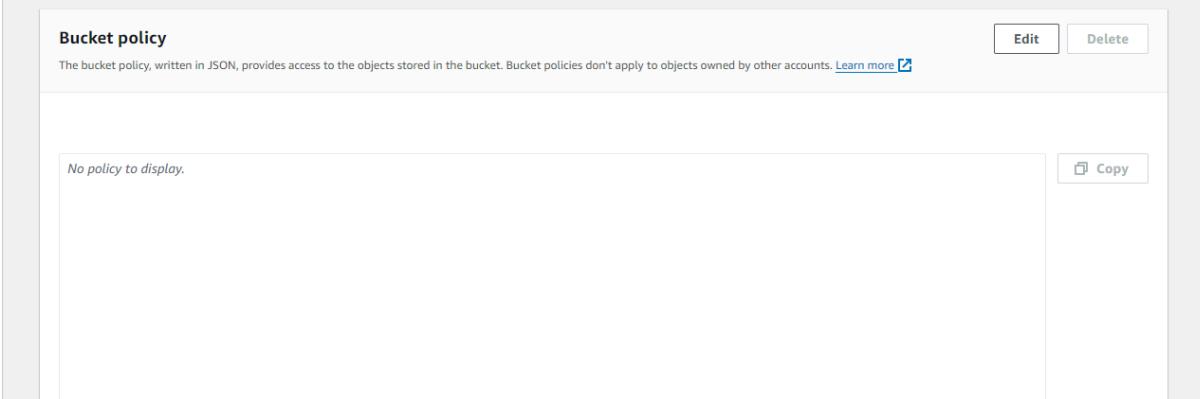
Step 11: Uncheck the Block all public access checkbox and click on save changes



The screenshot shows the 'Edit Block public access (bucket settings)' page. The title is 'Edit Block public access (bucket settings)'. Below it is a section titled 'Block public access (bucket settings)' with a descriptive paragraph about public access settings. Underneath is a list of checkboxes for different access control options:

- 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.

Step 12: Scroll down to bucket policy and click edit



The screenshot shows the 'Bucket policy' interface. The title is 'Bucket policy' with 'Edit' and 'Delete' buttons. A note says 'The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more.' Below this is a large text area containing the message 'No policy to display.' with a 'Copy' button to its right.

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.

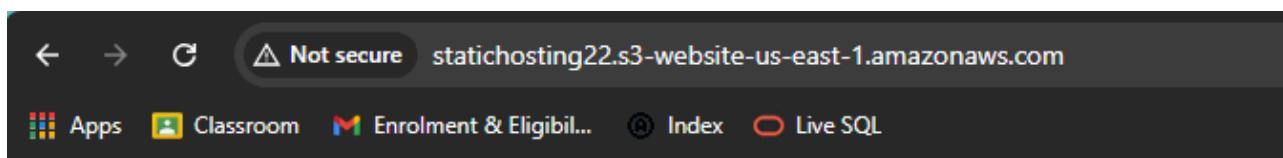
The screenshot shows the AWS Management Console interface for managing an S3 bucket. The top navigation bar includes 'Services', 'Search', and account information ('N. Virginia' and 'voclabs/user3389113=Shreyash_Kamat @ 4208-3547-4734'). The main content area displays the 'Bucket ARN' (arn:aws:s3:::statichosting22) and the 'Policy' section. The policy code is pasted into the 'Policy' textarea:

```

1▼ {
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "PublicReadGetObject",
6      "Effect": "Allow",
7      "Principal": {
8        "AWS": "*"
9      },
10     "Action": "s3:GetObject",
11     "Resource": "arn:aws:s3:::statichosting22/*"
12   }
13 }
```

To the right of the textarea, there is a sidebar with 'Edit statement' and 'Select a statement' buttons, and a note: 'Select an existing statement in the policy or add a new statement.' A 'CloudShell' link is at the bottom left, and a footer at the bottom right contains links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

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



Hello I am Shreyash Kamat

Exp 1b: Cloud9 Setup and Launch, Collaboration demonstration by creation of IAM groups and users.

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

The screenshot shows the 'Create environment' wizard in the AWS Cloud9 control panel. The 'Details' step is selected. The 'Name' field contains 'MohitCLOUD9'. The 'Description - optional' field contains 'This is my first cloud9 installation'. Under 'Environment type', the 'New EC2 instance' option is selected, with a note explaining that Cloud9 creates an EC2 instance in your account. The 'Existing compute' option is also available but not selected.

https://us-east-1.console.aws.amazon.com/cloud9control/home?region=us-east-1#/create/

Services Search [Alt+S] N. Virginia voclabs/user3385491+ Create environment Info

Details

Name

MohitCLOUD9

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

Description - optional

This is my first cloud9 installation

Limit 200 characters.

Environment type Info

Determines what the Cloud9 IDE will run on.

New EC2 instance

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

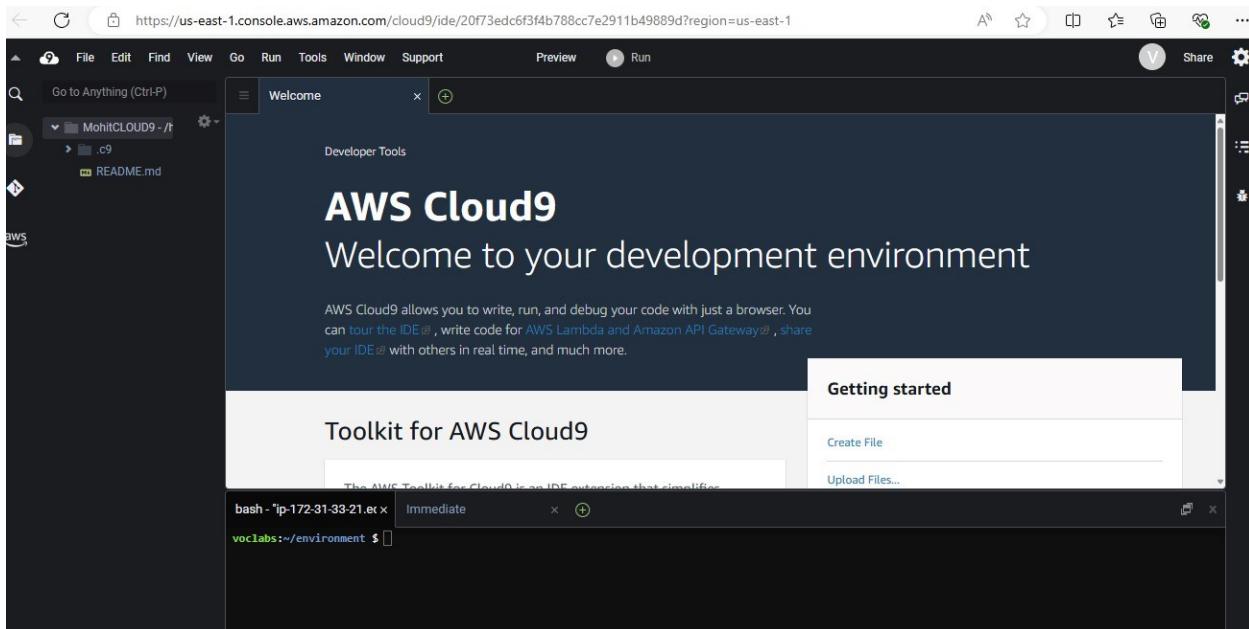
Existing compute

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

The screenshot shows the 'New EC2 instance' configuration page. At the top, there's a navigation bar with the URL <https://us-east-1.console.aws.amazon.com/cloud9control/home?region=us-east-1#/create/>. Below the navigation bar, the main title is 'New EC2 instance'. Under 'Instance type Info', three options are listed: 't2.micro (1 GiB RAM + 1 vCPU)' (selected), 't3.small (2 GiB RAM + 2 vCPU)', and 'm5.large (8 GiB RAM + 2 vCPU)'. Each option has a brief description. Below these, a link 'Additional instance types' leads to more options. Under 'Platform Info', it says 'Amazon Linux 2023' in a dropdown menu. Under 'Timeout', it says '30 minutes' in a dropdown menu.

2. We have successfully setup and launched our Cloud9 environment. Over here, we can build and develop programs as per our desire. We are also allowed to collaborate with multiple other users and access shared resources.

The screenshot shows the 'Specify user details' step of creating a new IAM user. The URL is <https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create>. The left sidebar shows 'Step 1: Specify user details', 'Step 2: Set permissions', and 'Step 3: Review and create'. The main area is titled 'User details' and contains a 'User name' field with 'sahil motiramani'. A note below says the user name can have up to 64 characters and lists valid characters. There's an optional checkbox for 'Provide user access to the AWS Management Console'. A callout box at the bottom right provides instructions for generating programmatic access keys. At the bottom right are 'Cancel' and 'Next' buttons.



3. Moving on, we are supposed to create a new user. Give a suitable name to the user and decide the password for the same.

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.

 Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

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

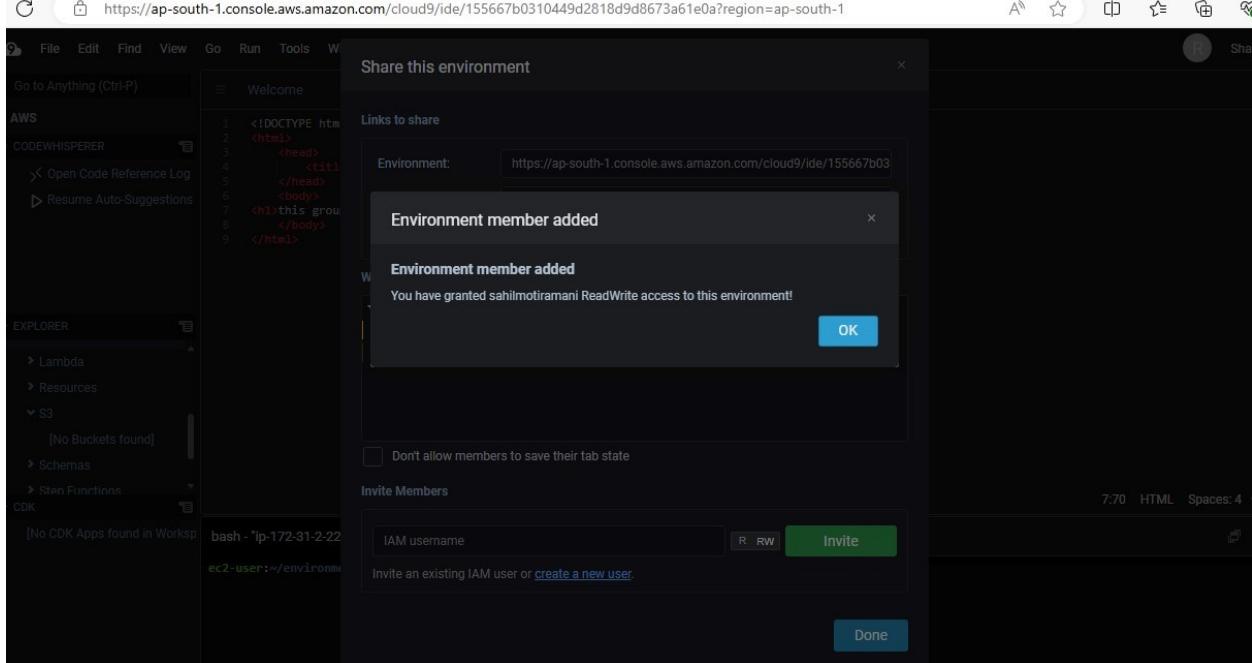
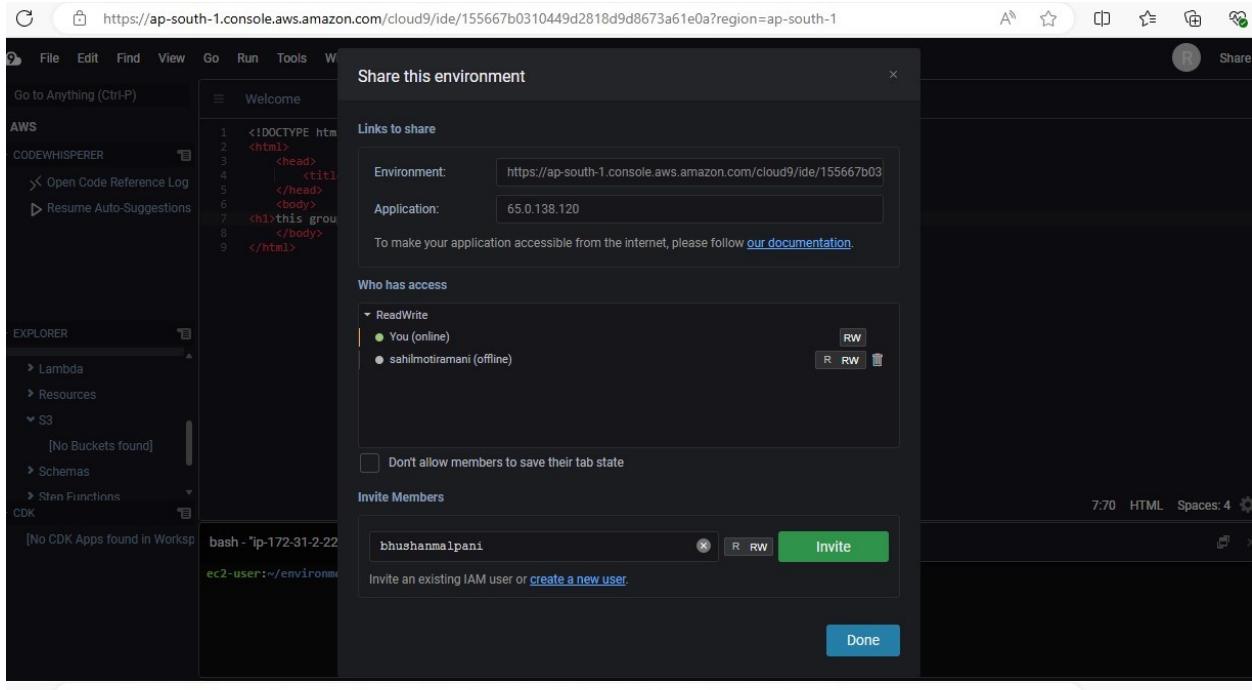
4. Similarly, create a new group and provide a suitable name for the same. Include the IAM users in this group together for our convenience i.e to provide similar kinds of permissions to the entire group rather than an individual user.

The screenshot shows the AWS IAM 'Create New User Group' wizard. Step 4 is completed, indicated by the green banner 'MSBCLLOUD9 user group created.' Below it, the 'Review and create' section shows three options: 'Add user to group' (selected), 'Copy permissions', and 'Attach policies directly'. The main table lists the newly created group 'MSBCLLOUD9' with details: Group name, Users (0), Attached policies (-), and Created (2024-07-29). A note to 'Set permissions boundary - optional' is present. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

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

The screenshot shows the AWS IAM 'Create New User' wizard. Step 4 is completed, indicated by the green banner 'sahilmotiramani user created.' The 'User details' section shows the user name 'sahilmotiramani', console password type 'Custom password', and 'Require password reset' set to 'Yes'. The 'Permissions summary' section lists two entities: 'IAMUserChangePassword' (AWS managed, used as Permissions policy) and 'MSBCLLOUD9' (Group, used as Permissions group). The 'Tags - optional' section notes that no tags are associated with the resource.

6. Go back to the cloud9 environment. Click on share this environment option so as to allow other collaborators to access your environment. Include your newly made IAM user in this environment and enable Read/Write permissions for it



7. Further, we are supposed to login from another browser using the credentials of the IAM user, so as to access the shared cloud9 environment with us.

These steps could not be completed because Cloud9 services have been disrupted and there is no access to the IAM user from the remote login

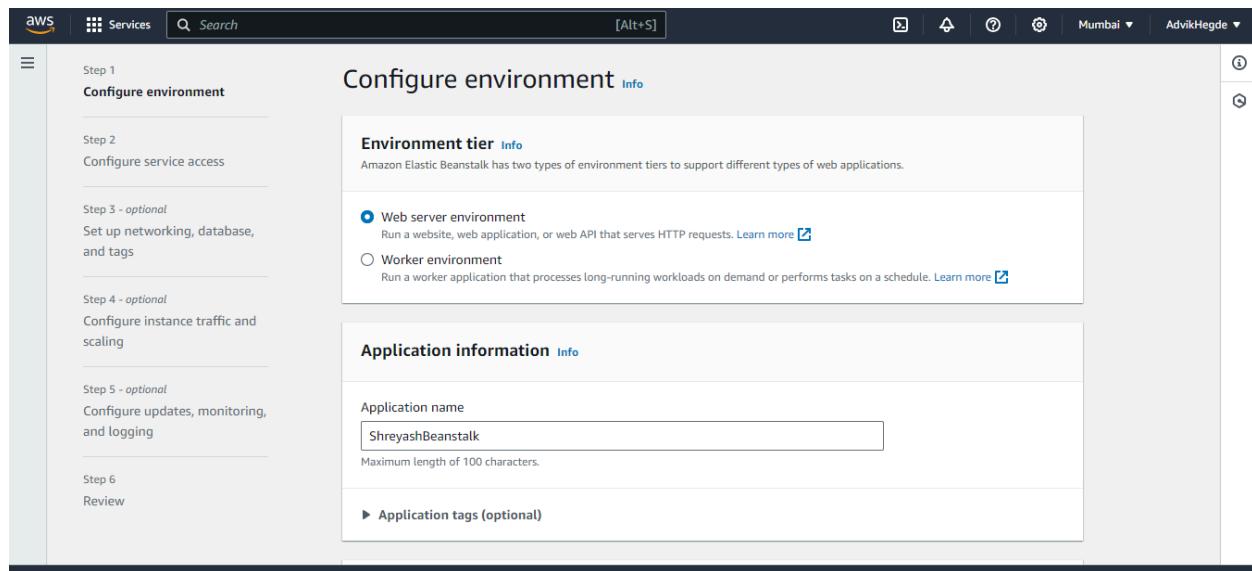
Name:Shreyash Kamat

Div/Roll No. : D15C/22

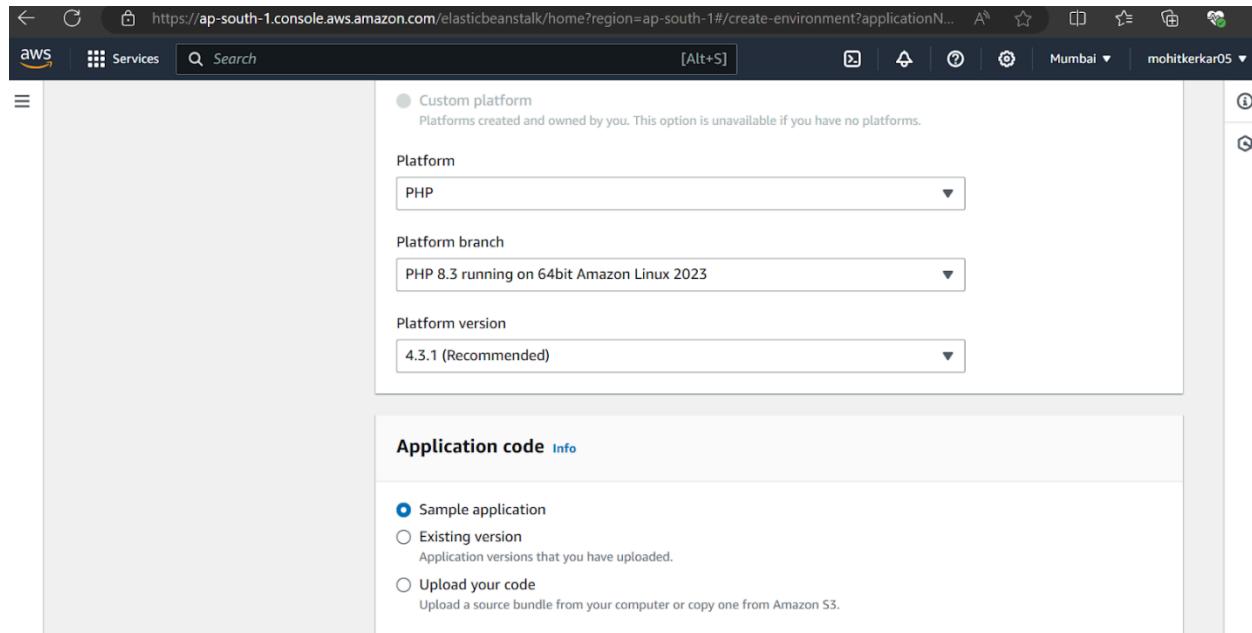
Exp 02:To Build Your Application using AWS Code Build and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS Code Deploy.

Step 1: Create our Elastic Beanstalk Environment

Login into your AWS account and navigate to services. Search for Elastic Beanstalk service and click on create application. Give your application a suitable name. For the platform, select PHP. The rest of the configuration settings are to be kept as default.



The screenshot shows the 'Configure environment' step of creating a new Elastic Beanstalk application. On the left, a sidebar lists steps from 1 to 6. Step 1 is selected, showing 'Configure environment'. The main area has two sections: 'Environment tier' and 'Application information'. In 'Environment tier', 'Web server environment' is selected. In 'Application information', the 'Application name' field contains 'ShreyashBeanstalk'. The browser address bar shows the URL for the AWS Elastic Beanstalk console.



The screenshot shows the 'Platform' configuration step of creating a new Elastic Beanstalk application. The sidebar shows step 1 selected. The main area includes fields for 'Platform' (set to 'PHP'), 'Platform branch' (set to 'PHP 8.3 running on 64bit Amazon Linux 2023'), and 'Platform version' (set to '4.3.1 (Recommended)'). Below this, the 'Application code' section is visible, showing options for 'Sample application' (selected), 'Existing version', and 'Upload your code'. The browser address bar shows the URL for the AWS Elastic Beanstalk console.

Now, while creating the environment, we are asked to provide an IAM role with the necessary EC2 permissions. We are supposed to make sure that we have made an existing IAM role with the following set of permissions:

1. AWSElasticBeanstalkWebTier
2. AWSElasticBeanstalkWorkerTier
3. AWSElasticBeanstalkMulticontainerDocker

We can skip the steps to follow after the initial few steps mentioned above and move straight to review the settings of our environment. After reviewing everything properly, our environment can successfully be created.

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar shows 'Applications' and 'Environments'. Under 'Environments', 'ShreyashBeanstalk-env' is selected. The main content area displays the 'Environment overview' for 'ShreyashBeanstalk-env'. It includes sections for 'Health' (with a warning icon), 'Domain' (ShreyashBeanstalk-env.eba-nqgmhcpy.ap-south-1.elasticbeanstalk.com), 'Environment ID' (e-pudixtyvb2), 'Application name' (ShreyashBeanstalk), and 'Platform' (PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2). The 'Events' tab is selected at the bottom. A green banner at the top says 'Environment successfully launched.'

Step 2: Fork the required repository onto our github account

The repository to be forked is - imoisharma/aws-codepipeline-s3-codeddeploy-linux-2.0

The screenshot shows the GitHub repository page for 'aws-codepipeline-s3-codeddeploy-linux-2.0' owned by 'imoisharma'. The repository is public and has 20 commits. A prominent 'Fork' button is visible in the top right corner. The commit history lists various files like README.md, .github, dist, scripts, CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE, and README.md, along with their respective dates and descriptions.

The screenshot shows the forked GitHub repository page for 'aws-codepipeline-s3-codeddeploy-linux-2.0' owned by 'Shreyash-664'. It is a fork of the original repository from 'imoisharma'. The repository is public and has 20 commits. A note at the top states 'This branch is up to date with imoisharma/aws-codepipeline-s3-codeddeploy-linux-2.0:master'. The commit history is identical to the original repository, showing the same files and their modifications.

This step is necessary for the execution of the steps to follow. It will be helpful in the creation of a pipeline.

Step 3: Creation of the Pipeline

Navigate to Codepipeline inside Developer Tools. Give a suitable name to the pipeline you want to create.

The screenshot shows the 'Choose pipeline settings' step in the AWS CodePipeline console. The pipeline name is set to 'ShreyashPipeline'. A note indicates that V1 pipelines cannot be created through the console, and recommends using the V2 pipeline type. The execution mode is set to 'Queued (Pipeline type V2 required)'. The sidebar shows steps 1 through 5: Choose pipeline settings (selected), Add source stage, Add build stage, Add deploy stage, and Review.

And click on next ...

The screenshot shows the 'Add source stage' step in the AWS CodePipeline console. The source provider is set to 'GitHub (Version 2)'. A note about GitHub version 2 actions is displayed. The connection search bar shows 'Connecting'. The sidebar shows steps 2 through 5: Add source stage (selected), Add build stage, Add deploy stage, and Review.

Step 4: GitHub connection

In this step, we are supposed to create a GitHub connection and add our existing repository over here i.e. the one we forked earlier.

The screenshot shows the AWS CodePipeline interface at Step 2 of 5. The left sidebar lists steps: Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5, and Review. The main panel is titled 'Source' and shows the 'Source provider' section. It indicates 'GitHub (Version 2)' is selected. A callout box provides information about the New GitHub version 2 (app-based) action, mentioning GitHub Apps and a link to learn more. Below this, the 'Connection' section shows a search bar with 'arn:aws:codeconnections:ap-south-1:221082173765:connection/479335fc-9a02-' and a 'Connecting' status. A green box at the bottom states 'Ready to connect' with a note that the GitHub connection is ready for use. The bottom navigation bar includes CloudShell, Feedback, and links to Privacy, Terms, and Cookie preferences.

We are supposed to enter our GitHub username so as to proceed towards making the connection.

aws Services More ▾

Developer Tools > ... > Create connection

Create a connection Info

Create GitHub App connection Info

Connection name

▶ Tags - *optional*

Connect to GitHub

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

Now to finalize our connection, we are to install an application which connects AWS to our GitHub account and repository.

Post the establishment of the connection, this is the message that is displayed. We can further select the branch of our repository that we want to connect.

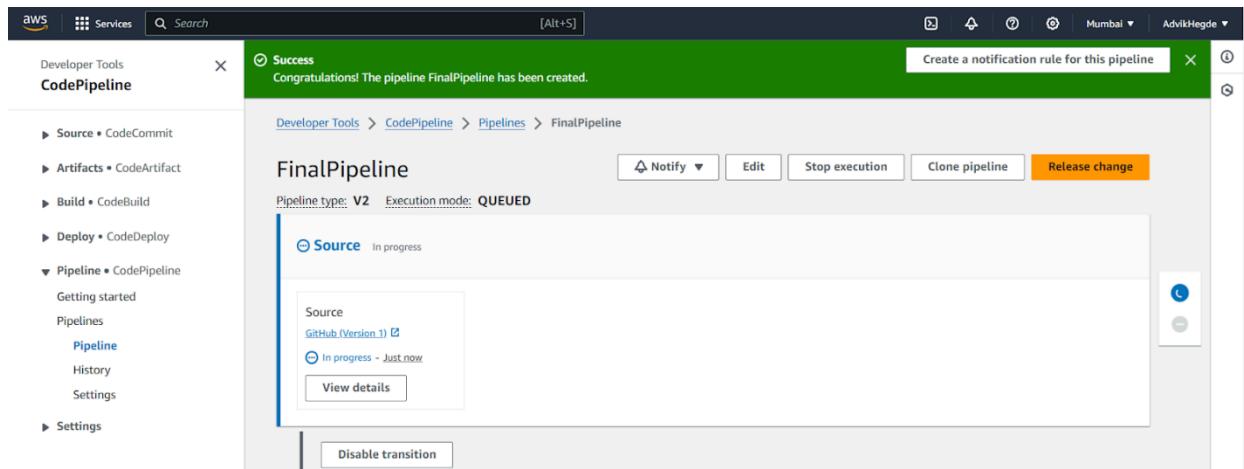
The screenshot shows the AWS CodePipeline setup interface. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar, and user information for 'AdvikHegde'. Below the navigation, the main content area has a 'Repository name' field containing 'Shreyash-664/aws-codepipeline-s3-codedeploy-linux-2.0'. A yellow warning box states: 'An unspecified error occurred. Check your network connectivity, and then check to see if there are any issues with the service at the [Service Health Dashboard](#). (Click here to retry)'. Below this, there's a 'Default branch' field set to 'master'. Under 'Output artifact format', the 'CodePipeline default' option is selected (indicated by a blue border), while 'Full clone' is also listed. At the bottom of the page, there are links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 5: Deployment stage:

We are expected to skip the build stage and move towards the deployment step. In the deployment step we are supposed to choose the Elastic Beanstalk application and the environment that we created earlier and proceed with our pipeline creation.

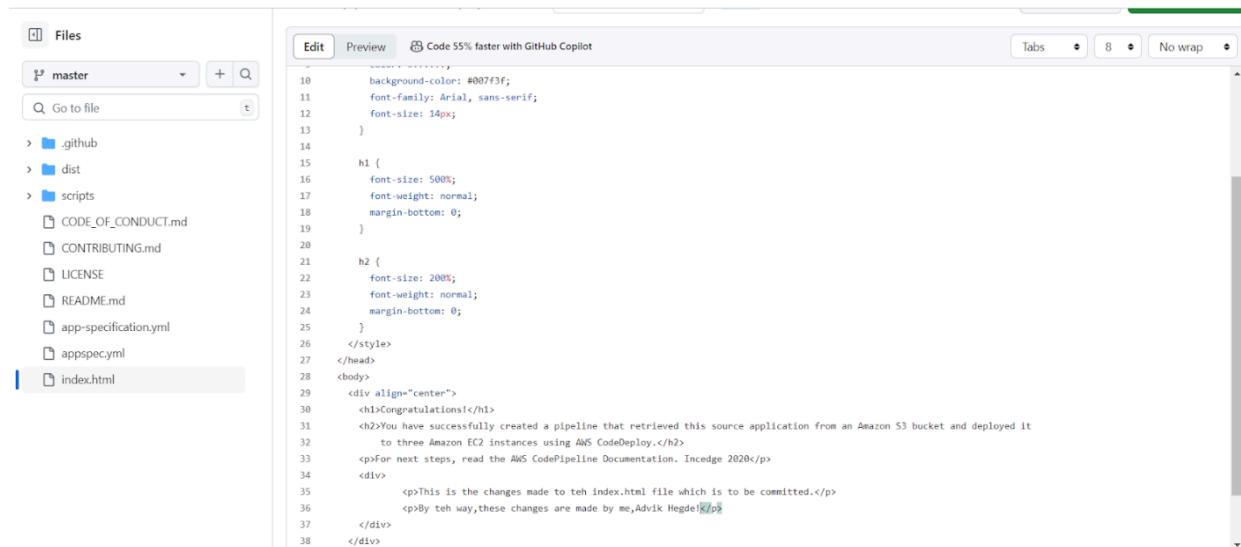
The screenshot shows the 'Add deploy stage' screen in the AWS CloudFormation console. The 'Deploy' tab is selected. The 'Deploy provider' dropdown is set to 'AWS Elastic Beanstalk'. The 'Region' dropdown is set to 'Asia Pacific (Mumbai)'. The 'Input artifacts' field is empty. The 'Application name' field contains 'ShreyashBeanstalk'. The 'Environment name' field contains 'ShreyashBeanstalk-env'. A checkbox for 'Configure automatic rollback on stage failure' is unchecked. The navigation bar at the top shows 'Services' and 'Search [Alt+S]'.

Step 6: Post deployment stage: When all the stages run successfully, this is what is displayed onto the screen. It shows us that our application and our environment have successfully been deployed using a dedicated pipeline created.



Step 7: Committing changes to your GitHub code

Now, we will go to our forked repository and make some changes to the index.html file. On making the desired changes, we are supposed to commit those changes on our forked repository. Write a good commit message so as to recognize it when it appears on the pipeline.



```
background-color: #007f3f;
font-family: Arial, sans-serif;
font-size: 14px;
}

h1 {
    font-size: 500px;
    font-weight: normal;
    margin-bottom: 0;
}

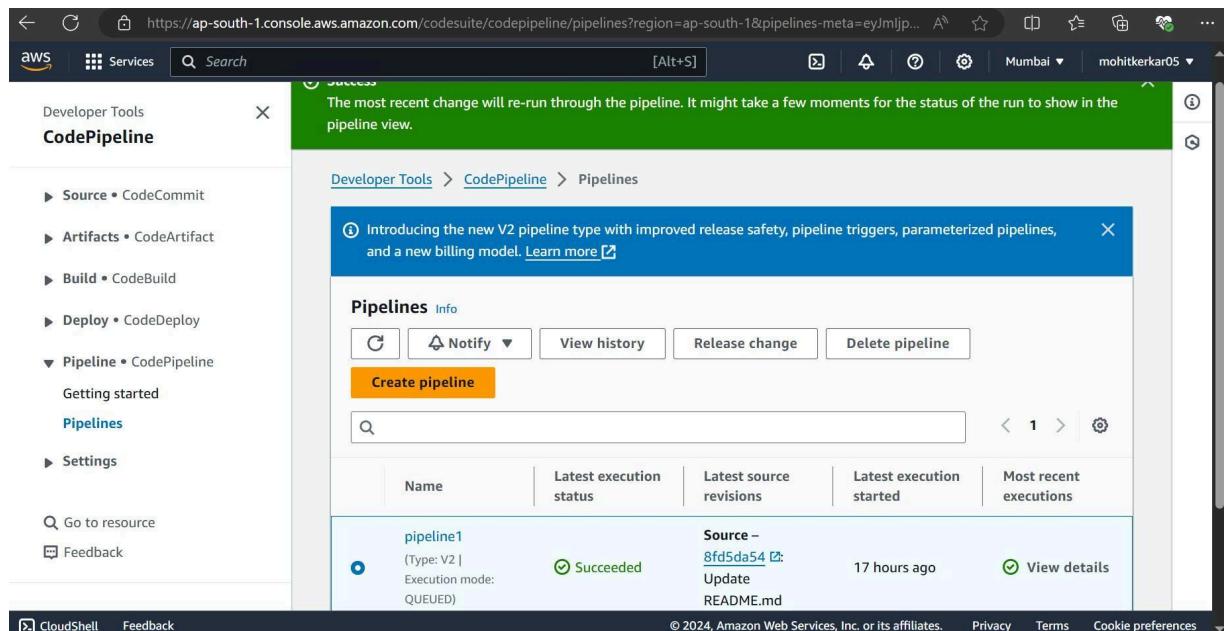
h2 {
    font-size: 200px;
    font-weight: normal;
    margin-bottom: 0;
}

</style>
</head>
<body>
<div align="center">
<h1>Congratulations!</h1>
<h2>You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.</h2>
<p>For next steps, read the AWS CodePipeline Documentation. Incide 2020</p>
<div>
<p>This is the changes made to teh index.html file which is to be committed.</p>
<p>By teh way,these changes are made by me,Advik Hegde</p>
</div>
</div>

```

Step 8: Apply the newly made changes in index.html onto our pipeline

Come back to the Codepipeline section and select the pipeline through which we successfully created and deployed our application. Click on the release change option to apply the latest changes/commits from our GitHub repository to our pipeline.



The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Developer Tools > CodePipeline > Pipelines

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. [Learn more](#)

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
pipeline1 (Type: V2 Execution mode: QUEUED)	Succeeded	Source - 8fd5da54 Update README.md	17 hours ago	View details

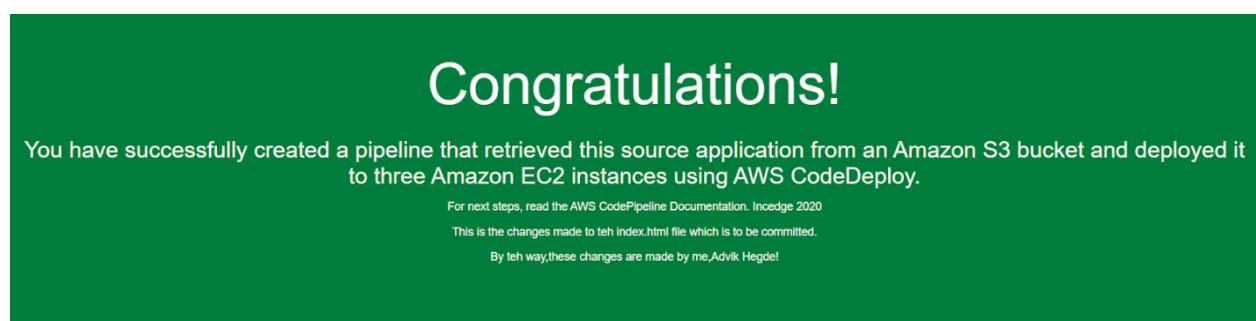
Once the changes have been applied, we see the commit message that we wrote for the latest commit on our repository being reflected on our pipeline. Over here, it would be seen somewhere near the bottom of the image that is attached. “Update index.html” was the latest commit message in the GitHub repository.

The screenshot shows the AWS CodePipeline console. At the top, there are two green success notifications: one stating 'Congratulations! The pipeline FinalPipeline has been created.' and another stating 'The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.' Below these, the pipeline name 'FinalPipeline' is displayed along with its type ('V2') and execution mode ('QUEUED'). A navigation bar at the top includes links for Developer Tools, CodePipeline, Pipelines, and FinalPipeline, along with buttons for Notify, Edit, Stop execution, Clone pipeline, and Release change. The main area shows the 'Source' stage, which has succeeded. It lists a GitHub commit: 'GitHub (Version 1)' with commit ID '3cf895ae' and a timestamp 'Succeeded - Just now'. A 'View details' button is present. To the right of the pipeline details, there are two small icons: a green checkmark and a blue circular arrow. The URL for the screenshot is <https://i.stack.imgur.com/0DfzJ.jpg>.

Step 9: Open the Domain of our Elastic Beanstalk environment

Now, we navigate back to our Elastic Beanstalk environment and open the environment domain of our deployed application.

The text in this image is clearly distinguishable from the earlier website's text meaning that the changes that we made to our code in index.html has successfully been applied to the website that we deployed.



Adv DevOps Lab Exp 03

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

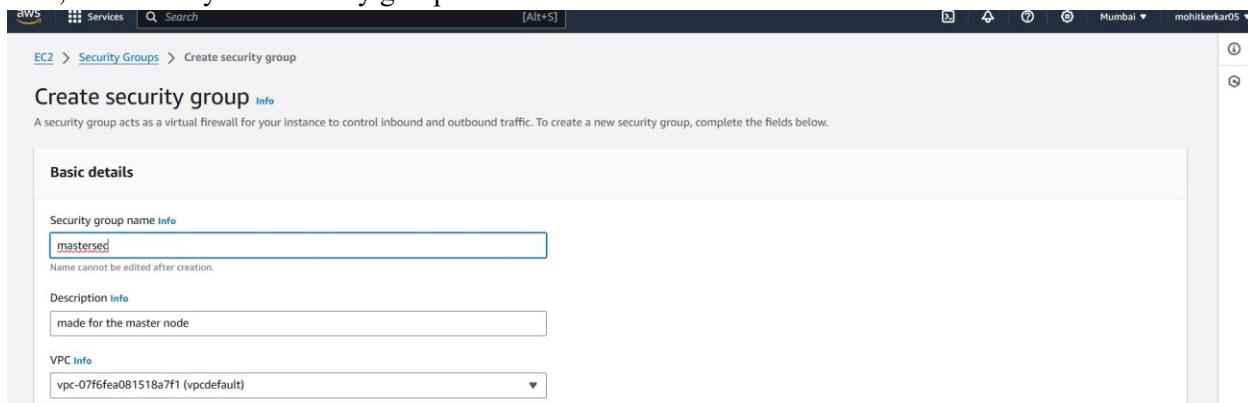
(I have performed this experiment on my personal AWS account)

Step 1: Create Key-pair, Security groups and required default VPCs

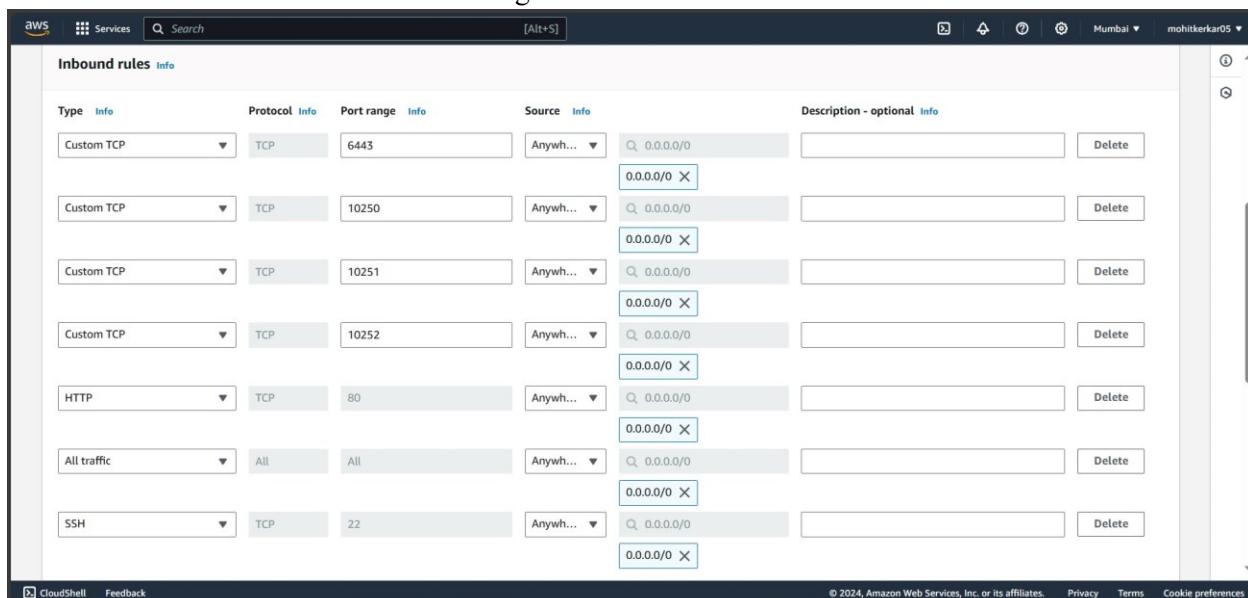
I started off by creating two separate security groups i.e one for the master instance (kubeadm to be initialized within it) and the other for the 2 worker instances. For creating any EC2 security groups, we require a VPC (Virtual private cloud) and a subnet along with it .. so that we can work with or allow inbound and outbound traffic and communication. Make sure you have a default VPC and a subnet already created which can be used for this experiment.

Also, make sure that you have a key pair installed of the type RSA with .pem extension on your local machine. Save it at a place which is accessible and where you can work from your terminal.

Here, i created my first security group



I modified the inbound rules in the following fashion for the master node



Then, i created a security group for the worker nodes

A screenshot of the AWS EC2 'Create security group' page. The 'Basic details' section contains the following information:

- Security group name:** workersec
- Description:** made for worker nodes
- VPC:** vpc-07f6fea081518a7f1 (vpcdefault)

Edited the inbound rules for the worker nodes in the following fashion

A screenshot of the AWS EC2 'Inbound rules' configuration page. The table lists the following rules:

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	Anywhere	0.0.0.0/0
HTTP	TCP	80	Anywhere	0.0.0.0/0
SSH	TCP	22	Anywhere	0.0.0.0/0
Custom TCP	TCP	10250	Anywhere	0.0.0.0/0
All TCP	TCP	0 - 65535	Anywhere	0.0.0.0/0
Custom TCP	TCP	30000 - 52767	Anywhere	0.0.0.0/0

Step 2: Create 3 EC2 instances i.e one master and other 2 workers

Now, in order to work with the instances, navigate to EC2 instances section and launch a few instances. Launch one instance and name it as masternode and the other two as worker.

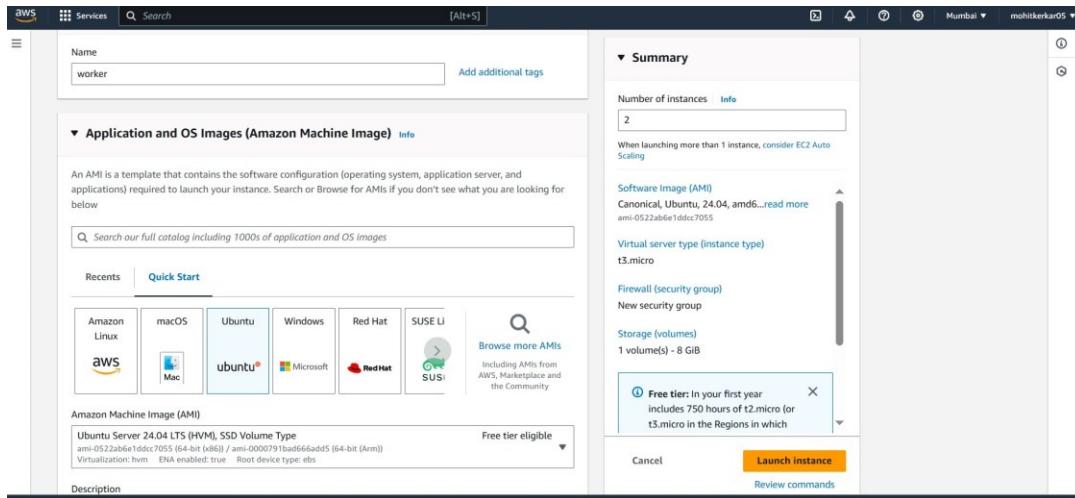
I selected **AMI as ubuntu** among the list of OS that we shown available in the free tier eligibility list

Following is the master instance node:

A screenshot of the AWS EC2 'Launch instance' page. The configuration includes:

- Name:** masternode
- Number of instances:** 1
- Software image (AMI):** Canonical, Ubuntu, 24.04, amd64... (ami-0522abef1ddcc7055)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB
- Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.

Following is the worker instance node:



IMPORTANT: Select t2.medium as the instance type for all 3 instances, as kubernetes requires at least 2 CPUs to work with and sufficient amount of other resources as well

Instance type

t2.medium

Family: t2 - 2 vCPU 4 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0496 USD per Hour

On-Demand Windows base pricing: 0.0676 USD per Hour

On-Demand RHEL base pricing: 0.0784 USD per Hour

On-Demand SUSE base pricing: 0.1496 USD per Hour

All generations

Compare instance types

Key pair (login)

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

exp3key

Create new key pair

Network settings

Network: vpc-07f6fea081518a7f1 | vpcdefault

Subnet:

Auto-assign public IP

Firewall (security groups)

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

Select security groups

mastersec sg-0b2b02a3572e6dc0c

VPC: vpc-07f6fea081518a7f1

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

After launching all the 3 instances, select one instance and press 'Connect'

Instances (3) <small>Info</small>								
		Last updated	Connect	Instance state	Actions	Launch instances	Mumbai	mohitkerkar05
		Find Instance by attribute or tag (case-sensitive)		All states				
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
masternode	i-0264ba2470966fc5	Running	t3.medium	Initializing	View alarms +	ap-south-1c	-	
worker	i-0f204d57806b582f	Running	t3.medium	Initializing	View alarms +	ap-south-1c	-	
worker	i-Qdf51ctf99646f4f	Running	t3.medium	Initializing	View alarms +	ap-south-1c	-	

Navigate to SSH client section and copy the command that's listed for connecting to the node remotely
 ssh -i "exp3key.pem" ubuntu@ec2-13-233-93-42.ap-south-1.compute.amazonaws.com...for the master node
 Similarly, do this for all nodes

Now, open 3 separate terminals and run change directories to the folder which contains the key we created earlier

Run their respective SSH commands (one in each terminal)

This helps us log onto those instances individually and remotely and work on them separately

```
ubuntu@ip-172-31-37-198: ~
PS C:\Users\Del1> cd C:\Users\Del1\Desktop\keypair
PS C:\Users\Del1\Desktop\keypair> ssh -i "exp3key.pem" ubuntu@ec2-3-111-188-84.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-111-188-84.ap-south-1.compute.amazonaws.com (3.111.188.84)' can't be established.
ECDSA key fingerprint is SHA256:Lm8ajmtdu/3LBU1qu0mUjWlCHZLqFwhD1jaUCAQAOqM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-111-188-84.ap-south-1.compute.amazonaws.com,3.111.188.84' (ECDSA) to the list
of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Thu Sep 26 09:43:59 UTC 2024

 System load:  0.34          Processes:      118
 Usage of /:   22.8% of 6.71GB  Users logged in:     0
 Memory usage: 6%            IPv4 address for enX0: 172.31.37.198
 Swap usage:   0%          

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.
```

```
PS C:\Windows\system32> cd C:\Users\DELL\Desktop\keypair
PS C:\Users\DELL\Desktop\keypair> ssh -i "exp3key.pem" ubuntu@ec2-13-234-38-84.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-234-38-84.ap-south-1.compute.amazonaws.com (13.234.38.84)' can't be established.
ECDSA key fingerprint is SHA256:P0+5r+bZ60Gpc0sL2A0+kkZYBNHMjfugne/ZBHPMHQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-234-38-84.ap-south-1.compute.amazonaws.com,13.234.38.84' (ECDSA) to the list
of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Thu Sep 26 09:46:24 UTC 2024

 System load: 0.08      Processes:          118
 Usage of /: 22.8% of 6.71GB  Users logged in:     0
 Memory usage: 5%           IPv4 address for enX0: 172.31.47.161
 Swap usage:  0%           
```

```
PS C:\Windows\system32> cd C:\Users\DELL\Desktop\keypair
PS C:\Users\DELL\Desktop\keypair> ssh -i "exp3key.pem" ubuntu@ec2-65-0-74-51.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-0-74-51.ap-south-1.compute.amazonaws.com (65.0.74.51)' can't be established.
ECDSA key fingerprint is SHA256:RTmw0J12RUM5s3vV2bYLL681AJE41Easupz4bVTCjFw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-0-74-51.ap-south-1.compute.amazonaws.com,65.0.74.51' (ECDSA) to the list of
known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Thu Sep 26 09:47:05 UTC 2024

 System load: 0.13      Processes:          119
 Usage of /: 22.8% of 6.71GB  Users logged in:     0
 Memory usage: 5%           IPv4 address for enX0: 172.31.33.106
 Swap usage:  0%           
```

Expanded Security Maintenance for Applications is not enabled.

Step 3: Docker installation:

Run the following command: These commands are used to install Docker on an Ubuntu system by adding Docker's official GPG key and configuring the Docker repository.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add curl
-fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null sudo add-apt-repository "deb
[arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-37-198: $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu > $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository...
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [15.3 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
```

Run the following commands to refresh your local package list to ensure the latest packages are available and install Docker Community Edition on your system without prompting for confirmation.

```
sudo apt-get update
sudo apt-get install -y docker-ce
```

```
ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-37-198: $ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7
  libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin
  libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 142 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]

ubuntu@ip-172-31-37-198: ~
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Run the following commands to create the Docker configuration directory and write a configuration file for Docker to use the systemd cgroup driver.

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

```
ubuntu@ip-172-31-37-198:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-37-198:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
> {
>   "exec-opts": ["native.cgroupdriver=systemd"]
> }
> EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
```

Run the following commands to ensure Docker starts automatically on system boot, reload systemd to apply new configurations, restart Docker to apply the new configuration.

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

```
ubuntu@ip-172-31-37-198:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-37-198:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-37-198:~$ sudo systemctl restart docker
```

Step 4: Kubernetes Installation:

Run the following commands to add the Kubernetes package repository to your Ubuntu system in order to install Kubernetes components like kubectl, kubelet, and kubeadm.

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

```
ubuntu@ip-172-31-37-198:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-37-198:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

Next, run these commands to refresh the local package index to include the newly added Kubernetes repository, install the main Kubernetes components (kubelet, kubeadm, kubectl), prevent the installed Kubernetes components from being automatically updated, ensuring cluster stability until manual updates are performed.

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

```

ubuntu@ip-172-31-37-198:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [533 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [129 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [376 kB]
Hit:8 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [155 kB]
Get:10 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
B]
Get:11 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 1324 kB in 1s (2118 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-37-198:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:

```



```

ubuntu@ip-172-31-37-198:~$ Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```



```

ubuntu@ip-172-31-37-198:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.

```

The command **sudo systemctl enable --now kubelet** enables the **kubelet** service, ensuring it starts automatically at boot and immediately starts running without needing a system reboot. The **sudo apt-get install -y containerd** command installs **containerd**, a container runtime that Kubernetes uses to manage and run containers.

sudo systemctl enable --now kubelet
sudo apt-get install -y containerd

```

Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-37-198: $ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4nets
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 142 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [85
99 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4
.1 [38.6 MB]
Fetched 47.2 MB in 1s (54.9 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.

```

The command `sudo mkdir -p /etc/containerd` creates the directory for containerd configuration files if it doesn't already exist. The next command, `sudo containerd config default | sudo tee /etc/containerd/config.toml`, generates a default configuration for containerd and saves it as `config.toml` in that directory. Together, these commands set up the necessary directory and create a default configuration file for containerd.

```

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

```

```

Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ sudo mkdir -p /etc/containerd
ubuntu@ip-172-31-37-198: $ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

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

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216

```

The command `sudo systemctl restart containerd` restarts the `containerd` service to apply any changes. `sudo systemctl enable containerd` sets it to start automatically at boot, while `sudo systemctl status containerd` checks its current status, showing whether it's active and any errors. Together, these commands manage the operation and health of the container runtime.

```

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

```

```
ubuntu@ip-172-31-37-198: $ sudo systemctl restart containerd
ubuntu@ip-172-31-37-198: $ sudo systemctl enable containerd
ubuntu@ip-172-31-37-198: $ sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-09-26 10:09:37 UTC; 14s ago
       Docs: https://containerd.io
   Main PID: 4771 (containerd)
      Tasks: 7
     Memory: 13.0M (peak: 13.4M)
        CPU: 123ms
      CGroup: /system.slice/containerd.service
              └─4771 /usr/bin/containerd

Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127045263Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127119958Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127202179Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127214694Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127239102Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127260747Z" level=info msg="Start"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127807399Z" level=info msg="servin"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.127910878Z" level=info msg="servin"
Sep 26 10:09:37 ip-172-31-37-198 containerd[4771]: time="2024-09-26T10:09:37.128176100Z" level=info msg="conta"
Sep 26 10:09:37 ip-172-31-37-198 systemd[1]: Started containerd.service - containerd container runtime.
```

Socat installation:

```
sudo apt-get install -y socat
```

```
ubuntu@ip-172-31-37-198: $ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 142 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (16.7 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.
```

Step 5: Kubernetes Cluster

Run this command only on the master instance sudo

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

It initializes a Kubernetes cluster with kubeadm and sets up the control plane (master node).

```
Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kubelet-start] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0926 10:13:04.272663    5083 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the
container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.
10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-37-198 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.37.198]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-37-198 localhost] and IPs [172.31.37.198 12
7.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-37-198 localhost] and IPs [172.31.37.198 127.
0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
```

```
Select ubuntu@ip-172-31-37-198: ~
luster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate a
nd key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.37.198:6443 --token gci45u.g7qowjugiqwlynrb \
  --discovery-token-ca-cert-hash sha256:a984e32c6c7c4815973519ebb45d64bf44a23f4e48dbf1195d8f0e695ea9409e
```

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

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

```
ubuntu@ip-172-31-37-198: ~ $ mkdir -p $HOME/.kube
ubuntu@ip-172-31-37-198: ~ $ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-37-198: ~ $ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-37-198: ~ $ kubectl get nodes
NAME           STATUS    ROLES          AGE   VERSION
ip-172-31-37-198  NotReady  control-plane  12m   v1.31.1
```

Connect master and worker nodes by running this command on the worker logged in terminals: I ran the following command to do this, `kubeadm join 172.31.37.198:6443 --token gci45u.g7qowjugiqw1ynrb \ --discovery-token-ca-cert-hash sha256:a984e32c6c7c4815973519ebb45d64bf44a23f4e48dbf1195d8f0e695ea9409e`

On node 1:

```
ubuntu@ip-172-31-47-161: ~
ubuntu@ip-172-31-47-161: $ sudo kubeadm join 172.31.37.198:6443 --token gci45u.g7qowjugiqw1ynrb \ --discovery-token-ca-cert-hash sha256:a984e32c6c7c4815973519ebb45d64bf44a23f4e48dbf1195d8f0e695ea9409e
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001901347s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

ubuntu@ip-172-31-47-161: $
```

On node 2:

```
ubuntu@ip-172-31-33-106: ~
ubuntu@ip-172-31-33-106: $ sudo kubeadm join 172.31.37.198:6443 --token gci45u.g7qowjugiqw1ynrb \ --discovery-token-ca-cert-hash sha256:a984e32c6c7c4815973519ebb45d64bf44a23f4e48dbf1195d8f0e695ea9409e
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.002345051s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

Run `kubectl get nodes` on the master instance to confirm the joining of the worker nodes. Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.

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

The command `kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml` deploys Calico, a networking and network security solution for Kubernetes, by applying the configuration specified in the provided YAML file from the Calico documentation. This sets up the necessary resources to enable networking capabilities within the Kubernetes cluster.

```
Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: ~ $ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-33-106   NotReady  <none>    17s    v1.31.1
ip-172-31-37-198   NotReady  control-plane  18m    v1.31.1
ip-172-31-47-161   NotReady  <none>    39s    v1.31.1
```

```
Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apixextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/ippreservations.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apixextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
```

sudo systemctl status kubelet

```
Select ubuntu@ip-172-31-37-198: ~
ubuntu@ip-172-31-37-198: $ sudo systemctl status kubelet
● kubelet.service - Kubernetes Node Agent
  Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
  Drop-In: /usr/lib/systemd/system/kubelet.service.d
            └─10-kubeadm.conf
  Active: active (running) since Thu 2024-09-26 10:13:42 UTC; 18min ago
    Docs: https://kubernetes.io/docs/
  Main PID: 5773 (kubelet)
    Tasks: 10 (limit: 4676)
   Memory: 32.6M (peak: 33.3M)
      CPU: 18.442s
     CGroup: /system.slice/kubelet.service
             └─5773 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=
```

Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.596915 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.596938 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.596958 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.596978 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.596997 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:24 ip-172-31-37-198 kubelet[5773]: I0926 10:32:24.597017 5773 reconciler_common.go:245] "operator" "Container runtime" "Container runt"
Sep 26 10:32:25 ip-172-31-37-198 kubelet[5773]: I0926 10:32:25.193047 5773 scope.go:117] "RemoveContainer" >
Sep 26 10:32:25 ip-172-31-37-198 kubelet[5773]: E0926 10:32:25.193206 5773 pod_workers.go:1301] "Error syncing pod"
Sep 26 10:32:27 ip-172-31-37-198 kubelet[5773]: E0926 10:32:27.544395 5773 kubelet.go:2902] "Container runt"
Sep 26 10:32:32 ip-172-31-37-198 kubelet[5773]: E0926 10:32:32.545352 5773 kubelet.go:2902] "Container runt"
lines 1-23/23 (END)
ubuntu@ip-172-31-37-198: \$

kubectl get nodes -o wide helps us get to know that the Status is ready.

```
ubuntu@ip-172-31-37-198: $ kubectl get nodes -o wide
NAME           STATUS   ROLES      AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE        KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-33-106 Ready    <none>    66s   v1.31.1   172.31.33.106  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-37-198 Ready    control-plane 19m   v1.31.1   172.31.37.198  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-47-161 Ready    <none>    88s   v1.31.1   172.31.47.161  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ubuntu@ip-172-31-37-198: $ kubectl label node ip-172-31-28-117 kubernetes.io/role=Node1
Error from server (NotFound): nodes "ip-172-31-28-117" not found
ubuntu@ip-172-31-37-198: $ kubectl label node ip-172-31-47-161 kubernetes.io/role=Node1
node/ip-172-31-47-161 labeled
ubuntu@ip-172-31-37-198: $ kubectl label node ip-172-31-33-106 kubernetes.io/role=Node2
node/ip-172-31-33-106 labeled
ubuntu@ip-172-31-37-198: $ kubectl get nodes -o wide
NAME           STATUS   ROLES      AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE        KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-33-106 Ready    Node2     3m26s  v1.31.1   172.31.33.106  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-37-198 Ready    control-plane 21m   v1.31.1   172.31.37.198  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
ip-172-31-47-161 Ready    Node1     3m48s  v1.31.1   172.31.47.161  <none>       Ubuntu 24.04 LTS   6.8.0-1012-aws   containerd://1.7.12
```

Or else run kubectl get nodes command

```
ubuntu@ip-172-31-37-198:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-33-106 Ready    Node2      3m42s   v1.31.1
ip-172-31-37-198 Ready    control-plane 21m     v1.31.1
ip-172-31-47-161 Ready    Node1      4m4s    v1.31.1
ubuntu@ip-172-31-37-198:~$
```

Conclusion: In this experiment, we successfully set up a Kubernetes cluster on AWS by creating the necessary infrastructure and configuring security groups. We installed Docker and Kubernetes components, initialized the master node, and connected the worker nodes. By deploying Calico for networking, we enabled effective communication within the cluster. This hands-on experience enhanced our understanding of Kubernetes architecture and equipped us with practical skills for managing containerized applications in a cloud environment.

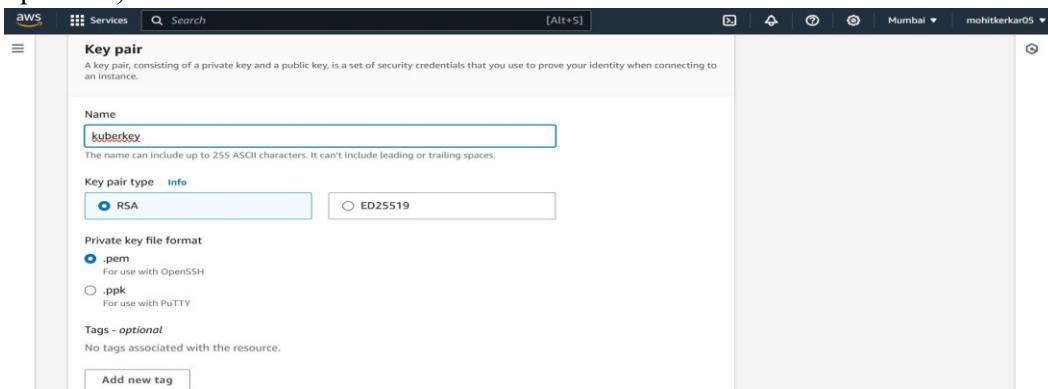
Adv DevOps Lab Exp 04

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

(I performed this Experiment on my personal AWS account)

Step 1: Creation of EC2 Instance, its Key-pair, remote login using SSH

Firstly, create a key pair with a desired name and type ‘RSA’ with .pem extension. (.pem is useful with OpenSSH)

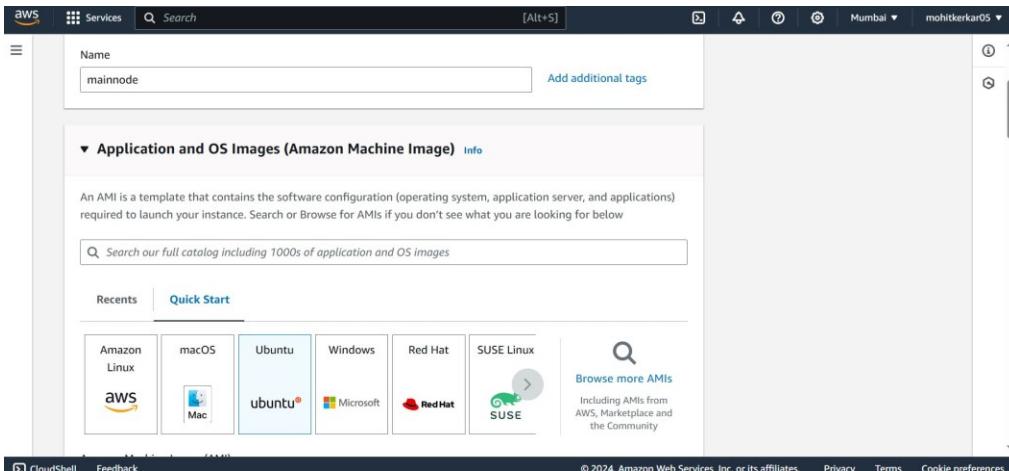


After creating the key, it automatically gets downloaded onto your machine. We are to create a new folder and place our within there. After this, we are to access our key and establish our connection with our instance by changing the directory to our key’s directory

Now, proceed to the creation of the EC2 instance. I selected Amazon machine image as **Ubuntu** and instance type as **t2.medium**, just for the purpose of convenience and also for successful execution of the experiment.

The execution of the experiment is determined by the fact that how many resources is the instance allowed to access or use. The Free Tier eligible instance type **t2.micro** **avails the instance with only one CPU**, whereas with t2.medium we get 2 CPU and much more execution space.

Be mindful that along with all this, **t2.medium does not have free tier eligibility**. So, use the instances and the resources that you allow them to access resourcefully and economically. Shut/terminate them as and when work related to them is over



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

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true
 On-Demand Linux base pricing: 0.0496 USD per Hour
 On-Demand Windows base pricing: 0.0676 USD per Hour
 On-Demand RHEL base pricing: 0.0784 USD per Hour
 On-Demand SUSE base pricing: 0.1496 USD per Hour

Additional costs apply for AMIs with pre-installed software

Instances (1/1) [Info](#)

Last updated less than a minute ago

[C](#) [Connect](#) [Actions](#) [Launch instances](#)

[Find Instance by attribute or tag \(case-sensitive\)](#) [All states](#)

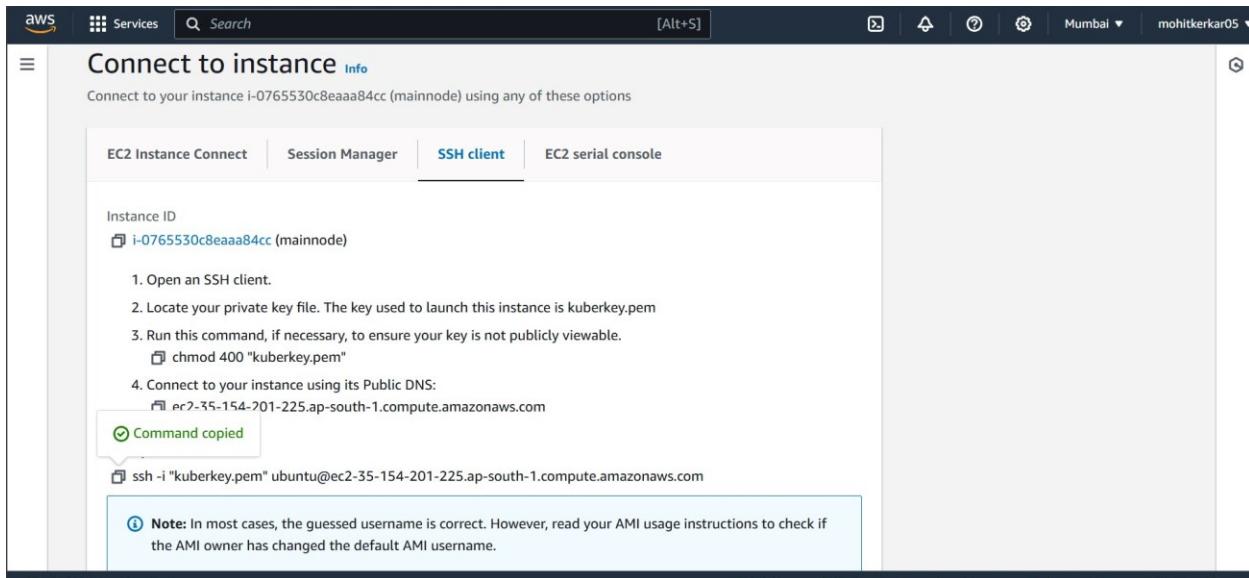
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
mainnode	i-0765530c8eaaa84cc	Running	t2.medium	Initializing	View alarms

Here are my instance details. From here on, after remotely logging in from my local terminal, you would see my IPv4 address in my terminal (Powershell)

i-0765530c8eaaa84cc (mainnode)

i-0765530c8eaaa84cc (mainnode)	35.154.201.225 open address	172.31.40.244
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-35-154-201-225.ap-south-1.compute.amazonaws.com
Hostname type	Private IP DNS name (IPv4 only)	open address
IP name: ip-172-31-40-244.ap-south-	open	

Select the instance and press connect. Navigate to the SSH client section and copy the SSH command provided to us. We will require this for remotely logging into our instance from our local terminal.



As one can see, I changed my directory to the folder in which i had stored my key (.pem) and ran the ssh command so as to remotely login onto my instance and perform the necessary tasks from my terminal itself

```
ubuntu@ip-172-31-40-244: ~
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Windows\system32> cd C:\Users\Del\Desktop\keypair
PS C:\Users\Del\Desktop\keypair> ssh -i "kuberkey.pem" ubuntu@ec2-35-154-201-225.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-35-154-201-225.ap-south-1.compute.amazonaws.com (35.154.201.225)' can't be established.
ECDSA key fingerprint is SHA256:GIs2bs4t0fWY44Hiy3aN3Li34BT8eDD+LxwJMADr7HU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-154-201-225.ap-south-1.compute.amazonaws.com,35.154.201.225' (ECDSA) to the
list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep 25 16:45:41 UTC 2024

System load:  0.0          Processes:      113
Usage of /:   22.8% of 6.71GB  Users logged in:    0
Memory usage: 5%
Swap usage:   0%
IPv4 address for enX0: 172.31.40.244

Expanded Security Maintenance for Applications is not enabled.
```

Step 2: Docker installation

Run the below mentioned commands in order to install docker on your instance

Adding key: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add **Adding key without apt-key:** curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null

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

```
ubuntu@ip-172-31-40-244: $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-40-244: $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-40-244: $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $ (lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [15.3 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

Run these commands to update the package list and install Docker:

`sudo apt update`

`sudo apt install docker-ce docker-ce-cli containerd.io`

```
ubuntu@ip-172-31-40-244: ~
nerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Setting up libldl17:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-40-244: $
```

Next, run this command.... `sudo mkdir -p`

`/etc/docker cat <<EOF | sudo tee`

`/etc/docker/daemon.json`

```
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

`EOF`

`sudo systemctl enable docker`

`sudo systemctl daemon-reload`

`sudo systemctl restart docker`

```
ubuntu@ip-172-31-40-244: $ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-40-244: $ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-40-244: $ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-40-244: $ sudo systemctl daemon-reload
ubuntu@ip-172-31-40-244: $ sudo systemctl restart docker
```

Step 3: Kubernetes Installation

Run these commands to install kubernetes: curl -fsSL

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg echo 'deb [signed-
by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
ubuntu@ip-172-31-40-244: $ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-40-244: $ echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /" | Out-File -FilePath "C:\path\to\your\directory\kubernetes.list" -Encoding utf8
Out-File: command not found
ubuntu@ip-172-31-40-244: $ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
> https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/
ubuntu@ip-172-31-40-244: $
```

After this step, run the following commands,

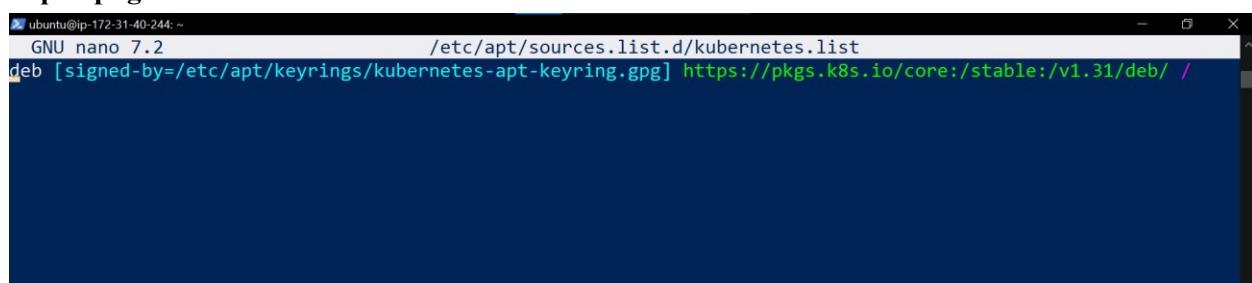
```
sudo apt-get update
sudo apt-get install -y
kubeadm kubelet kubectl
sudo apt-mark hold
kubeadm kubelet kubelet kubectl
```

Over here, while running sudo apt-get update if you encounter an error like this

```
ubuntu@ip-172-31-40-244: ~ $ sudo apt-get update
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
```

Then run the following command and ensure that the file contains the following texts:

```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```



Be sure to save this file and then proceed with these commands

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

```
ubuntu@ip-172-31-40-244: ~
ubuntu@ip-172-31-40-244: $ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 https://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/core:/stable:/v1.31/deb InRelease [1186 B]
]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 132 kB in 1s (199 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-40-244: $ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 142 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
```

```
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-40-244: $ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-40-244: $
```

```
sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

```
ubuntu@ip-172-31-40-244: ~
ubuntu@ip-172-31-40-244: $ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-40-244: $ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0925 17:22:15.942659    6444 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create new CRI runtime service: validate service connection: validate CR
I v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint
"unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...
To see the stack trace of this error execute with --v=5 or higher
```

Since running that command ran us into an error, we are expected to install containerd first

Run this command: sudo apt-get install -y containerd

Then run, sudo mkdir -p /etc/containerd

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

```
ubuntu@ip-172-31-40-244: $ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-40-244: $ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 142 not upgraded.
Need to get 47.2 MB of archives.
```

Post this step, run these commands one by one

sudo systemctl restart containerd sudo

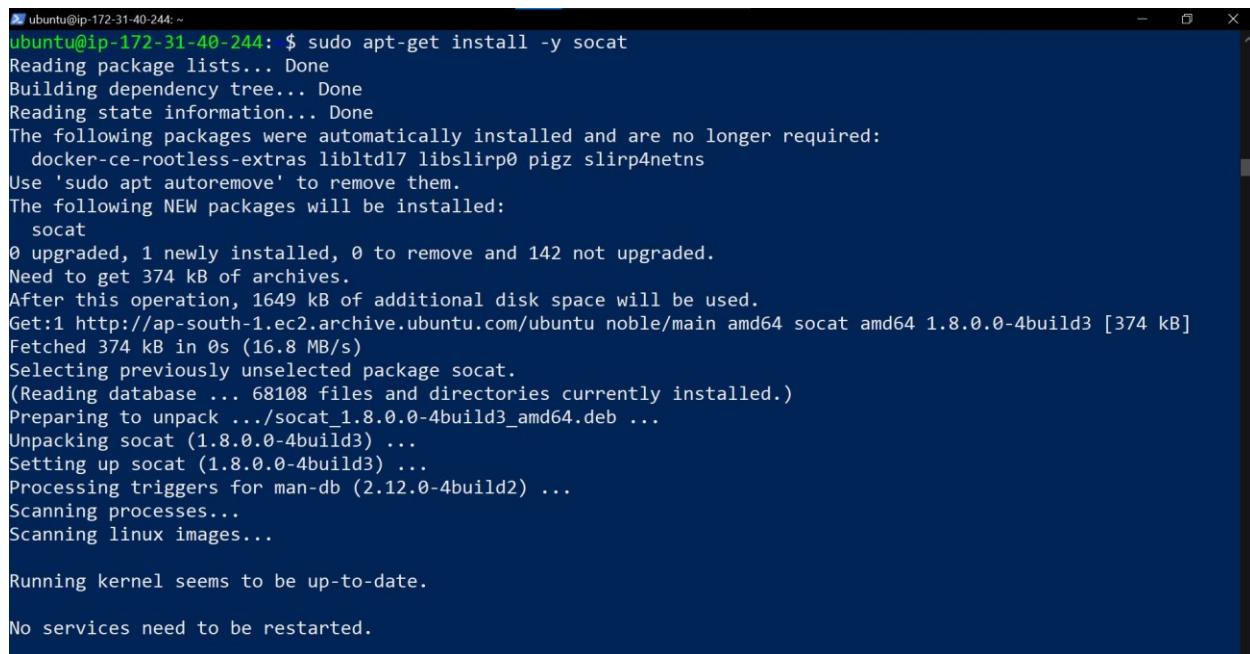
systemctl enable containerd sudo systemctl

status containerd

```
ubuntu@ip-172-31-40-244: $ sudo systemctl restart containerd
ubuntu@ip-172-31-40-244: $ sudo systemctl enable containerd
ubuntu@ip-172-31-40-244: $ sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-09-25 17:23:54 UTC; 17s ago
     Docs: https://containerd.io
     Main PID: 6555 (containerd)
        Tasks: 7
       Memory: 13.5M (peak: 14.1M)
         CPU: 121ms
        CGroup: /system.slice/containerd.service
                  └─6555 /usr/bin/containerd

Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231856399Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231887455Z" level=info msg="servin"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231896683Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231919602Z" level=info msg="servin"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231952908Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231972572Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231980403Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.231986601Z" level=info msg="Start"
Sep 25 17:23:54 ip-172-31-40-244 systemd[1]: Started containerd.service - containerd container runtime.
Sep 25 17:23:54 ip-172-31-40-244 containerd[6555]: time="2024-09-25T17:23:54.234185443Z" level=info msg="conta
```

After this, run sudo apt-get
install -y socat



```
ubuntu@ip-172-31-40-244: ~
ubuntu@ip-172-31-40-244: $ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libltdl17 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 142 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (16.8 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.
```

Step 4: Initialize kubecluster:

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

```
ubuntu@ip-172-31-40-244: ~
ubuntu@ip-172-31-40-244: $ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0925 17:31:39.850175    7057 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the
container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.
10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-40-244 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.40.244]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-40-244 localhost] and IPs [172.31.40.244 12
7.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-40-244 localhost] and IPs [172.31.40.244 127.
0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
```

```
ubuntu@ip-172-31-40-244: ~
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate a
nd key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.40.244:6443 --token egcpta.ggcdfxh328wg5vm3 \
  --discovery-token-ca-cert-hash sha256:e392dc659374560eb409e01b69491d214ea39cb415ee934aaecfcff6a7f21d5
ubuntu@ip-172-31-40-244: $
```

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

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

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

```
$HOME/.kube/config
```

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

```
kubectl apply -f
```

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

```
ubuntu@ip-172-31-40-244:~$ 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
ubuntu@ip-172-31-40-244:~$
```

Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment
 kubectl apply -f https://k8s.io/examples/application/deployment.yaml

```
ubuntu@ip-172-31-40-244:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-40-244:~$
```

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-brgb7	0/1	Pending	0	48s
nginx-deployment-d556bf558-s5rtj	0/1	Pending	0	48s

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

```
ubuntu@ip-172-31-40-244:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-40-244:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-40-244:~$
```

To untaint all nodes, run the following command

kubectl taint nodes --all [node-role.kubernetes.io/control-plane:NoSchedule-](https://kubernetes.io/docs/concepts/policy/taints-and-tolerations/)

Then run, kubectl get nodes

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-40-244	Ready	control-plane	16m	v1.31.1

Then run, kubectl get pods command again

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-brgb7	1/1	Running	0	7m15s
nginx-deployment-d556bf558-s5rtj	1/1	Running	0	7m15s

Since, Jenkins server/service is already occupying port 8080, i tried starting the nginx service at 8082 port number. Following is the command that i executed was, kubectl port-forward \$POD_NAME 8082:80

```
ubuntu@ip-172-31-40-244:~$ kubectl port-forward svc/nginx 8082:80
Forwarding from 127.0.0.1:8082 -> 80
Forwarding from [::1]:8082 -> 80
Handling connection for 8082
```

The last step consisted of opening a new terminal and logging in using the SSH method that we used earlier and running the following command so as to finally confirm that we have successfully deployed our kubernetes application over the nginx server, using nginx service (optional) curl --head http://127.0.0.1:8082

```
Last login: Wed Sep 25 19:13:52 2024 from 45.112.58.76
ubuntu@ip-172-31-40-244:~$ curl --head http://127.0.0.1:8082
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 25 Sep 2024 19:36:26 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

Conclusion:

In this experiment, we successfully deployed an Nginx web server on a Kubernetes cluster and exposed it using NodePort services. By creating the deployment and configuring the service, we demonstrated Kubernetes' ability to manage containerized applications efficiently.

The use of 'kubectl' commands allowed us to check pod status and access the Nginx server locally via port forwarding on the EC2 instance. This experience provided valuable insights into service management and networking concepts within Kubernetes.

Overall, this experiment reinforced the fundamental skills needed for deploying cloud-native applications and laid the groundwork for exploring more advanced Kubernetes features in future projects.

Name:Shreyash Kamat

Div:D15C

Roll no:22

Experiment 5

Aim:To Understand Terraform lifecycle,core concepts,technologies and install it on a Linux or Windows machine.

Installation and Configuration of Terraform in Windows

1: To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website: :<https://www.terraform.io/downloads.html>
Select the Operating System Windows followed by either 32 bit or 64 bit based on your OS type.

Windows

Binary download

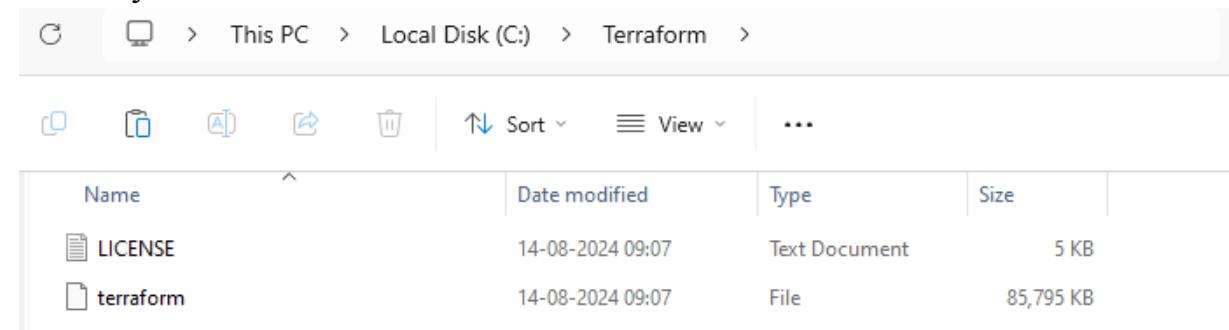
386
Version: 1.9.4

[Download](#)

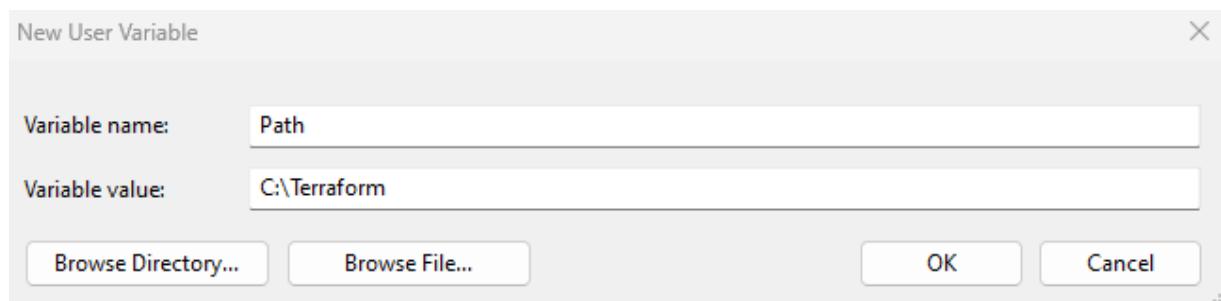
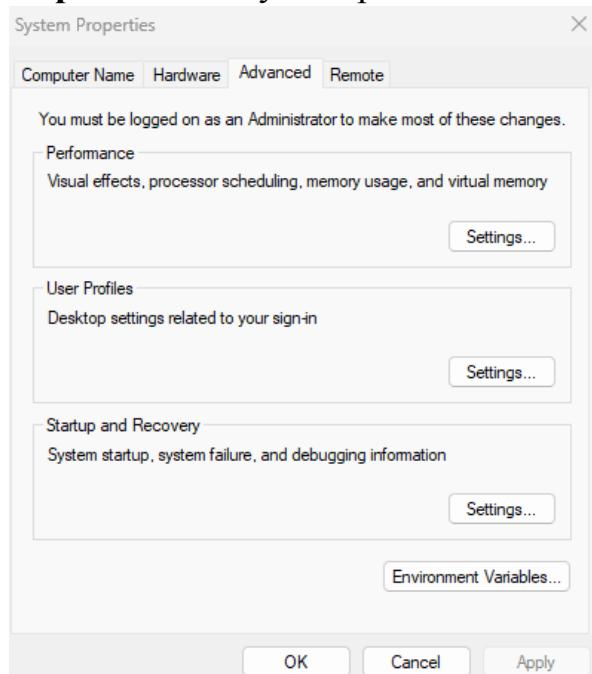
AMD64
Version: 1.9.4

[Download](#)

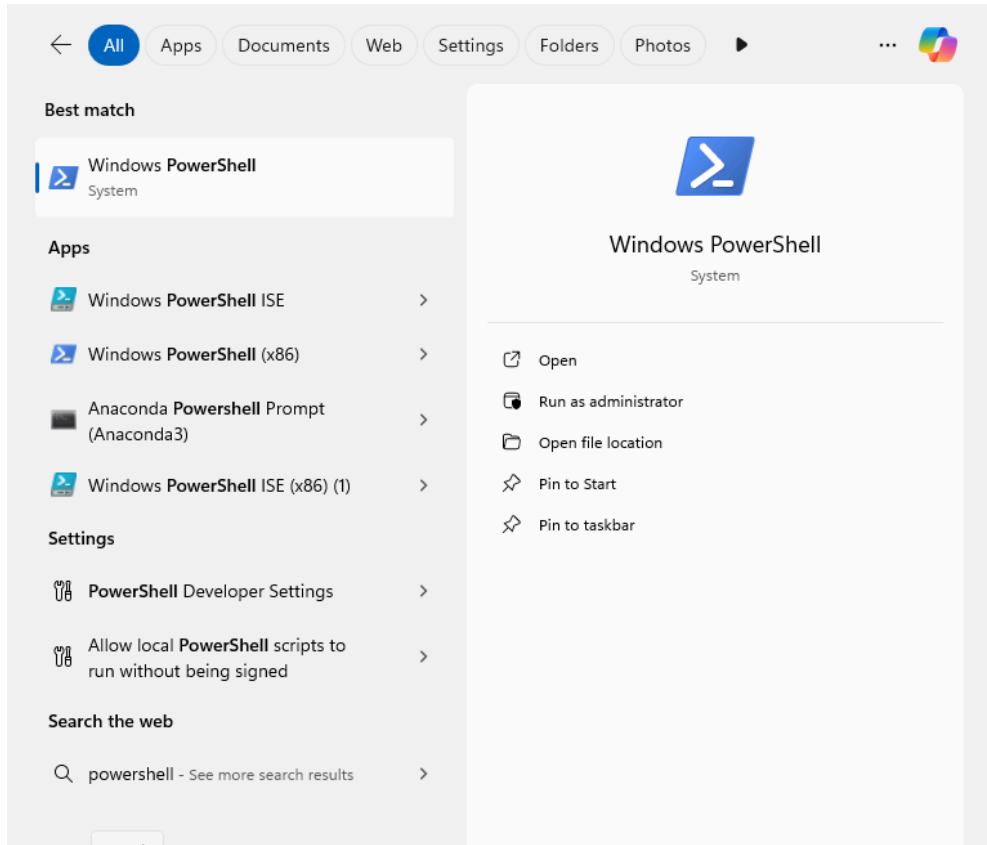
Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



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



Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality

```
PS C:\Users\student.VESIT505-17.000> 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
```

Name: Shreyash Kamat

Div/Roll no: D15C/22

Experiment No: 6

Implementation:

A. Creating docker image using terraform

Prerequisite:

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

Step 1: Check the docker functionality

```
PS C:\Users\INFT505-07> 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 (Docker Inc., v0.11.2-desktop.5)
  compose* Docker Compose (Docker Inc., v2.22.0-desktop.2)
  container  Manage containers
  context    Manage contexts
  dev*      Docker Dev Environments (Docker Inc., v0.1.0)
  extension* Manages Docker extensions (Docker Inc., v0.2.20)
  image     Manage images
  init*    Creates Docker-related starter files for your project (Docker Inc., v0.1.0-beta.8)
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks
  plugin    Manage plugins
  sbom*    View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scan*    Docker Scan (Docker Inc., v0.26.0)
```

```
PS C:\Users\INFT505-07> docker --version
Docker version 24.0.6, build ed223bc
PS C:\Users\INFT505-07>
```

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

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

Script:

terraform

```
{ required_providers
  docker = {
```

```

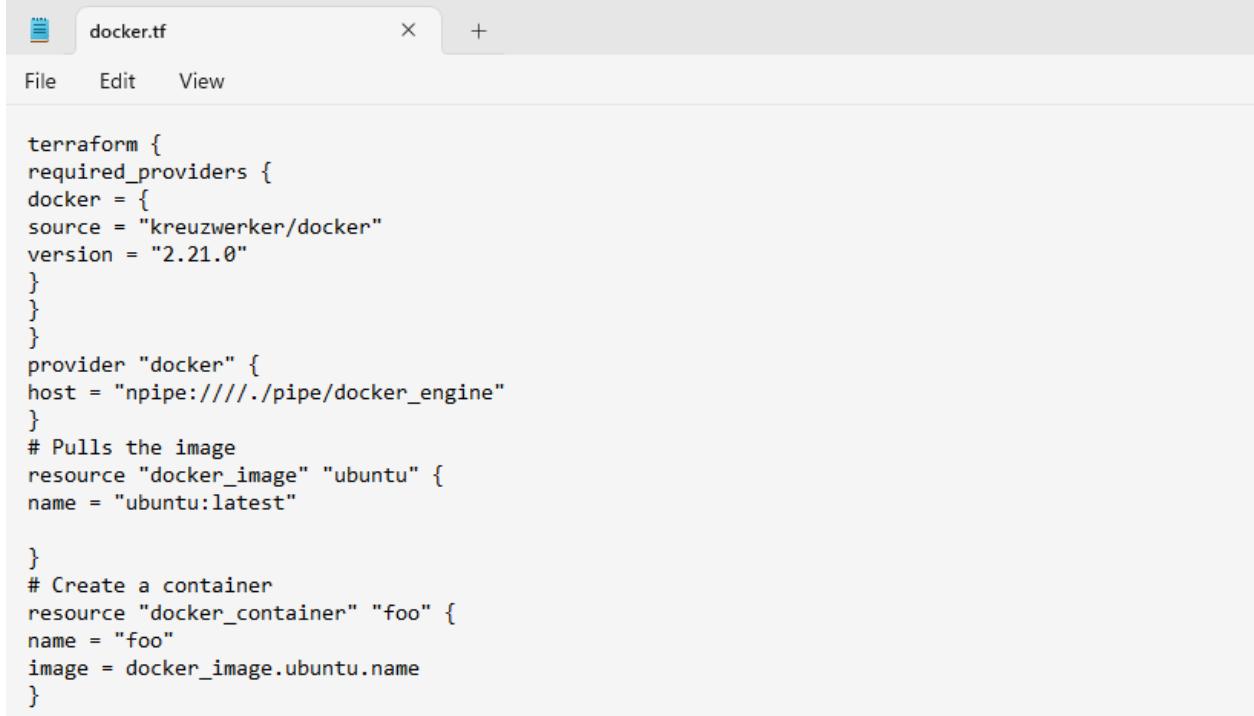
source = "kreuzwerker/docker"
version = "2.21.0"
}
}
}

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

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

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

```



The screenshot shows a code editor window with a tab labeled 'docker.tf'. The file contains Terraform configuration code:

```

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

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

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

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

```

Step 3: Execute Terraform Init command to initialize the resources

```
PS C:\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.
```

Step 4: Execute Terraform plan to see the available resources

```
PS C:\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           = "ubuntu:latest"
  + 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)
```

```
Administrator: Windows PowerShell + -v
+ 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.
```

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

```
+ name          = "foo"
+ network_data   = (known after apply)
+ read_only      = false
+ remove_volumes = true
+ restart        = "no"
+ rm             = false
+ runtime         = (known after apply)
+ security_opts   = (known after apply)
+ shm_size        = (known after apply)
+ start           = true
+ stdin_open      = false
+ stop_signal     = (known after apply)
+ stop_timeout    = (known after apply)
+ tty              = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

Enter a value: yes

docker_container.foo: Creating...
```

Docker images, Before Executing Apply step:

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
```

Docker images, After Executing Apply step:

```
PS C:\terraform_scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       edbfe74c41f8  2 weeks ago   78.1MB
```

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

```
PS C:\terraform_scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

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

Plan: 0 to add, 0 to change, 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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

Docker images After Executing Destroy step

```
PS C:\terraform_scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
```

Name:Shreyash Kamat

Div/Roll No:D15C/22

Exp 7:Understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

1. Open up Jenkins Dashboard on localhost, port 8090 or whichever port it is at for you.

The screenshot shows the Jenkins Dashboard with the following interface elements:

- Left sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views".
- Top right:** Buttons for "All" (highlighted), "+", and "Add description".
- Table:** A list of build projects with columns: Status (S), Warning (W), Name, Last Success, Last Failure, and Last Duration. The table contains the following data:

S	W	Name	Last Success	Last Failure	Last Duration
Green	Sunny	devops cicd pipes	1 mo 22 days #3	N/A	6.4 sec
Red	Cloudy	devops pipeline	N/A	1 mo 22 days #10	57 ms
Green	Sunny	Kspipeline	1 mo 6 days #1	N/A	5.7 sec
Green	Sunny	maven-tomcat1	1 mo 0 days #1	N/A	1 min 19 sec
Red	Cloudy	mavenproj	N/A	1 mo 22 days #1	1 min 36 sec

At the bottom, there are icons for "Icon: S M L" and a "More" button.

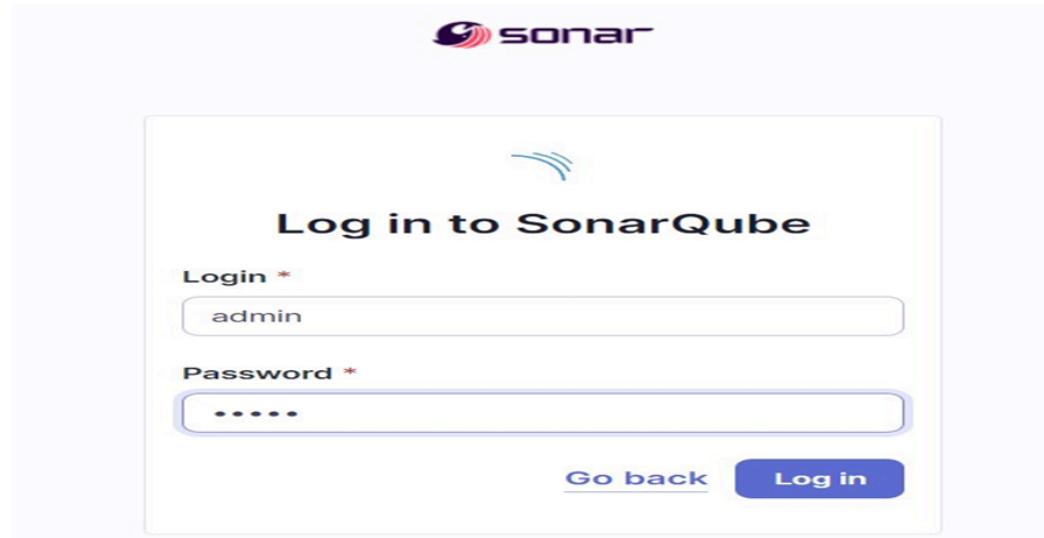
- Run SonarQube in a Docker container using this command -

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

-----Warning: run below command only once

```
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops> 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: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
77e678cded2ef5f989912d3d9e6991dd548eac03faaleed68dd906614be53acc
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops>
```

- 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.

The screenshot shows the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, a heading says "How do you want to create your project?". A note below it asks if the user wants to benefit from SonarQube's features like repository import and Pull Request decoration, and suggests creating a project from a favorite DevOps platform. It then asks if the user needs to set up a DevOps platform configuration. There are five main options for importing projects:

- Import from Azure DevOps (Setup button)
- Import from Bitbucket Cloud (Setup button)
- Import from Bitbucket Server (Setup button)
- Import from GitHub (Setup button)
- Import from GitLab (Setup button)

Below these options, a note asks if the user is testing or has an advanced use-case, suggesting to create a local project. A "Create a local project" button is shown in a box.

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

The screenshot shows the "Create a local project" form. It starts with a header "1 of 2" and "Create a local project". The first step is to enter the "Project display name *", which is "exp7". The second step is to enter the "Project key *", which is also "exp7". The third step is to enter the "Main branch name *", which is "main". Below these fields, a note says "The name of your project's default branch" with a "Learn More" link. At the bottom, there are "Cancel" and "Next" buttons.

Project display name *

exp7

Project key *

exp7

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.

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

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'sonar'. Below the search bar, there are four navigation links: 'Updates' (with a '39' badge), 'Available plugins' (which is highlighted in blue), 'Installed plugins', and 'Advanced settings'. To the right of the search bar is an 'Install' button and a refresh icon. The main area displays a list of plugins under the heading 'Name'. The first plugin listed is 'SonarQube Scanner 2.17.2', which is described as allowing an easy integration of SonarQube for continuous inspection of code quality. It was released 7 months and 8 days ago. The second plugin listed is 'Sonar Quality Gates 315.vff12b_e81a_3a_4', which fails the build whenever Quality Gates criteria are not met. It was released 29 days ago. The third plugin listed is 'Quality Gates 2.5', which also fails the build if Quality Gates status is different from 'Passed'. The entire list is contained within a light gray box with a thin border.

6. Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me
exp7

In **Server URL** Default is <http://localhost:9000>

The screenshot shows the Jenkins 'System > SonarQube servers' configuration page. At the top left, there is a section titled 'SonarQube servers' with a note that checked boxes allow job administrators to inject server configurations as environment variables. A checkbox labeled 'Environment variables' is checked. Below this is a section titled 'SonarQube installations' with a note about listing installations. The main form area contains fields for 'Name' (set to 'exp7'), 'Server URL' (set to 'Default is http://localhost:9000' and 'http://localhost:9000' in the input field), and 'Server authentication token' (a dropdown menu set to '- none -' with a '+ Add' button below it). At the bottom of the form is an 'Advanced' dropdown menu.

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

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for 'Gradle installations', 'SonarScanner for MSBuild installations', 'SonarQube Scanner installations', and 'Ant installations'. Each section has a 'Add [Tool]' button.

Check the “Install automatically” option. → Under name any name as identifier
→ Check the “Install automatically” option.

The screenshot shows the 'SonarQube Scanner' configuration dialog. It includes fields for 'Name' (set to 'sonarqube_exp7'), 'Install automatically' (checked), 'Version' (set to 'SonarQube Scanner 6.2.0.4584'), and an 'Add Installer' dropdown. A 'Add SonarQube Scanner' button is also present.

8. After the configuration, create a New Item in Jenkins, choose a freestyle project with name ks_exp7

New Item

Enter an item name
ks_exp7

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

OK

9. Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git

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

Dashboard > exp7 > Configuration

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

Credentials [?](#)
- none -
[+ Add](#)

Advanced [▼](#)

[Add Repository](#)

10. Under **Select project → Configuration → Build steps → 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 project named 'exp7'. The 'Configuration' screen is displayed, specifically the 'Build steps' section. A dropdown menu is open, showing various build step options: Execute SonarQube Scanner, Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, Set build status to "pending" on GitHub commit, SonarScanner for MSBuild - Begin Analysis, and SonarScanner for MSBuild - End Analysis. Below this, there is a link to 'Add build step'.

Under the 'Post-build Actions' section, there is a button to 'Add post-build action'.

At the bottom, there are 'Save' and 'Apply' buttons.

A detailed view of the 'Execute SonarQube Scanner' step is shown in a modal or expanded area:

- JDK**: Set to 'JDK 17'.
- Path to project properties**: An empty text input field.
- Analysis properties**: A text area containing the following properties:

```
sonar.projectKey=ks_exp7
sonar.projectName=ks_exp7
sonar.projectVersion=1.0
sonar.sources=C:/ProgramData/Jenkins/jenkins/workspace/ks_exp7
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.password=[REDACTED]
```
- Additional arguments**: An empty text input field.

 Status

 Changes

 Workspace

 Build Now

 Configure

 Delete Project

 SonarQube

 Rename

 SonarQube

Permalinks

- Last build (#7), 4 min 55 sec ago
- Last stable build (#7), 4 min 55 sec ago
- Last successful build (#7), 4 min 55 sec ago
- Last failed build (#6), 17 min ago
- Last unsuccessful build (#6), 17 min ago
- Last completed build (#7), 4 min 55 sec ago

 Build History

 Filter...

 trend

 #7

Sep 25, 2024, 3:09 PM

Console Output

 Download Copy View as plain

```

Started by user Shreyash Kamat
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[ks_exp7] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube1_exp7\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=ks_exp7 -Dsonar.projectName=ks_exp7 -Dsonar.host.url=http://localhost:9000 -
Dsonar.login=admin -Dsonar.projectVersion=1.0 -Dsonar.sources=C:/ProgramData/Jenkins/.jenkins/workspace/ks_exp7 -Dsonar.password=kshitij24 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7
15:09:08.473 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
```

11. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.

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

4 of 4 shown

12. Once the build is complete, check project on SonarQube

The screenshot shows the SonarQube interface for the 'main' project. At the top, there's a navigation bar with tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information dropdowns. The main content area has a title 'main' and a 'Version 1.0' indicator. A prominent green 'Passed' status is displayed next to a checkmark icon. Below it, a yellow warning box says 'The last analysis has warnings. See details'. Under the 'Overall Code' tab, there are three performance cards: Security (0 Open issues, A grade), Reliability (0 Open issues, A grade), and Maintainability (0 Open issues, A grade). The overall layout is clean and modern, typical of a software development tool.

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

Conclusion: In this project, we successfully integrated Jenkins with SonarQube to implement automated static application security testing (SAST). We initiated the process by deploying SonarQube via Docker, followed by configuring Jenkins with the required plugins and authentication methods. Next, we connected Jenkins to a GitHub repository and incorporated the SonarQube scanner as a build step. This setup allows for continuous code analysis, identifying vulnerabilities, code smells, and quality issues, thus ensuring automated reporting and ongoing enhancements in code quality.

Exp 8: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Theory:

What is SAST?

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

What problems does SAST solve?

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

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

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

Why is SAST important?

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines

of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.

What is a CI/CD Pipeline?

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

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



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

What is SonarQube?

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

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

Benefits of SonarQube

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

Integrating Jenkins with SonarQube:

Prerequisites:

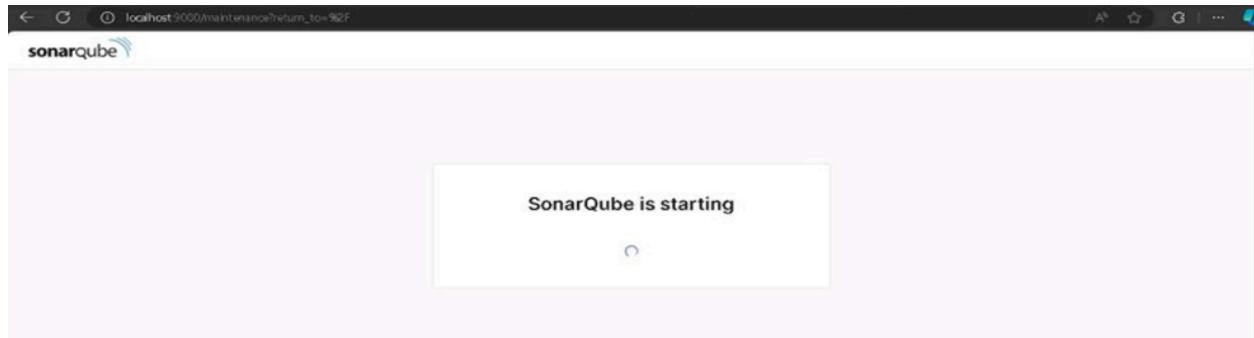
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

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 -

```
| PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8> docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
| 71fc67f0b15baa5be5bcd66966938e18682683d020beadcbc909dd027cf7a  
| PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8>
```

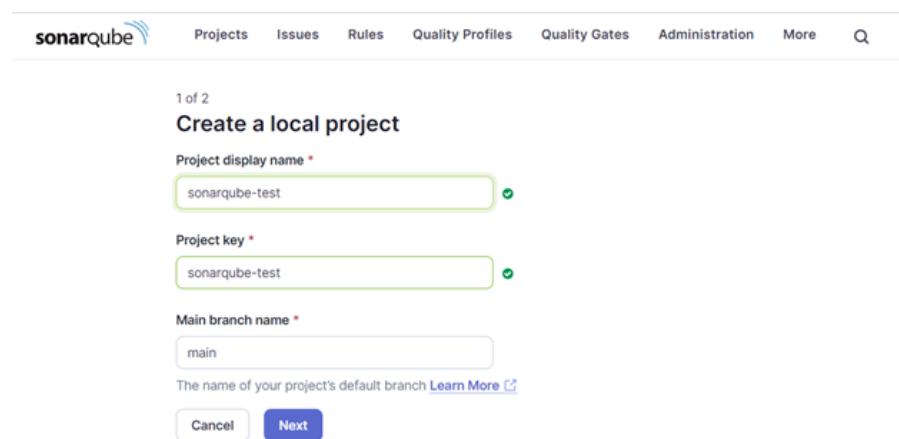
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



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



5. Create a manual project in SonarQube with the name **sonarqube-test**.
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-2

Select an item type



Freestyle project

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



Pipeline

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



Multi-configuration project

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



Folder

OK

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/" }
        }
    }
```

The screenshot shows the Jenkins Pipeline configuration page. The pipeline script is defined as follows:

```

1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') {
7       bat """
8         sh "C:/ProgramData/Jenkins/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqube/bin/sonar-scanner.bat"
9         -D sonar.login=<admin>
10        -D sonar.password=<password>
11        -D sonar.projectKey=<sonarqube-test>
12        -D sonar.exclusions=<vendor>**,resources/**,**/*.java
13        -D sonar.host.url=http://127.0.0.1:9000/
14        ...
15      """
16    }
17 }

```

Below the script, there is a checkbox labeled "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons.

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

The screenshot shows the Jenkins Pipeline status for the "SonarQube-2" pipeline. The pipeline has the following stages:

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

The "Stage View" section displays the message: "No data available. This Pipeline has not yet run."

SonarQube-2

Stage View

No data available. This Pipeline has not yet run.

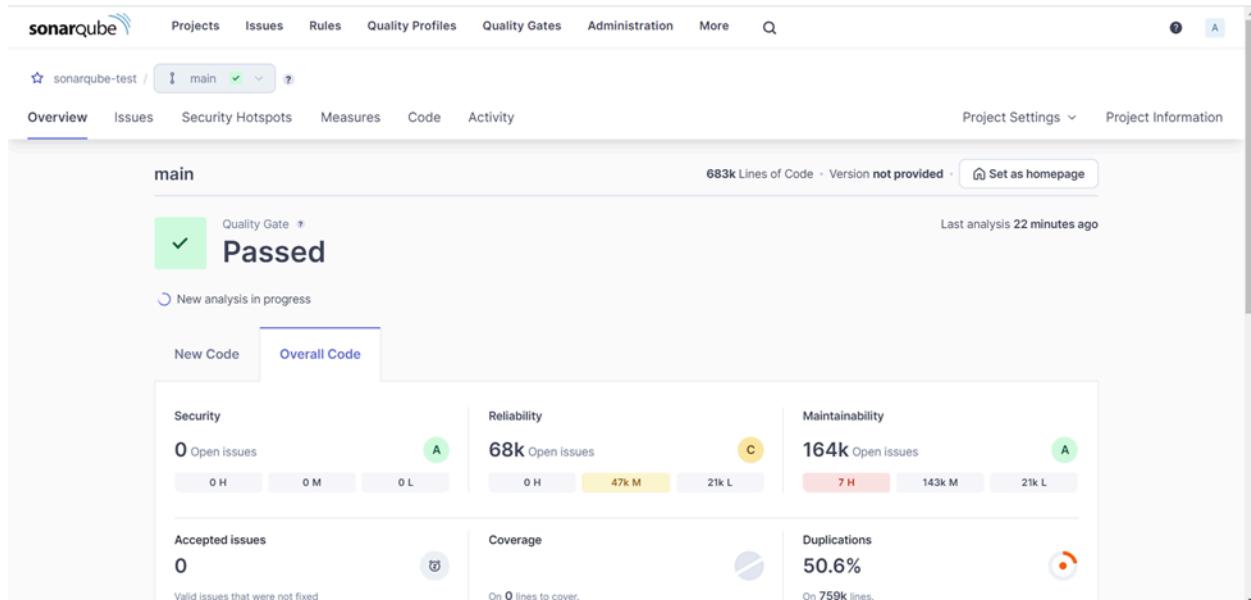
Permalinks

```

Dashboard > SonarQube-2 > #4
01:38:50.521 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.
01:38:50.321 INFO  CPD Executor CPD calculation finished (done) | time=238591ms
01:38:50.413 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
01:38:58.184 INFO  Analysis report generated in 525ms, dir size=127.2 MB
01:39:17.257 INFO  Analysis report compressed in 19081ms, zip size=29.6 MB
01:39:28.472 INFO  Analysis report uploaded in 11215ms
01:39:28.488 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
01:39:28.488 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
01:39:28.488 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=add9b398-1d74-4d8a-81ac-9b210f0e0b94
01:39:48.048 INFO  Analysis total time: 11:58.619 s
01:39:48.488 INFO  SonarScanner Engine completed successfully
01:39:49.184 INFO  EXECUTION SUCCESS
01:39:49.314 INFO  Total time: 12:15.607s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

10. After that, check the project in SonarQube.



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

11. Bugs

Responsibility

Add to selection Ctrl + click

Software Quality

Security 0

Reliability 33k

Maintainability 0

Severity ?

Type

Bug 33k

Vulnerability 0

Code Smell 164k

Add to selection Ctrl + click

Scope

Status

Select issues | Navigate to issue | 32,896 issues | 1369d effort

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

Insert a <!DOCTYPE> declaration to before this <html> tag.

Reliability 60

Consistency user-experience

Open Not assigned

L1 × 5min effort × 4 years ago • Bug • Major

gameoflife-core/build/reports/tests/allclasses-frame.html

Insert a <!DOCTYPE> declaration to before this <html> tag.

Reliability 60

Consistency user-experience

Open Not assigned

L1 × 5min effort × 4 years ago • Bug • Major

gameoflife-core/build/reports/tests/alltests-errors.html

Insert a <!DOCTYPE> declaration to before this <html> tag.

Reliability 60

Consistency user-experience

Open Not assigned

L1 × 5min effort × 4 years ago • Bug • Major

Code Smells

Responsibility

Add to selection Ctrl + click

Software Quality

Severity ?

Type

Bug 33k

Vulnerability 0

Code Smell 164k

Add to selection Ctrl + click

Scope

Status

Security Category

Select issues | Navigate to issue | 163,766 issues | 1705d effort

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

Remove this deprecated "width" attribute.

Maintainability 60

Consistency html5 obsolete

Open Not assigned

L9 × 5min effort × 4 years ago • Code Smell • Major

gameoflife-core/build/reports/tests/allclasses-frame.html

Remove this deprecated "align" attribute.

Maintainability 60

Consistency html5 obsolete

Open Not assigned

L11 × 5min effort × 4 years ago • Code Smell • Major

gameoflife-core/build/reports/tests/alltests-errors.html

Remove this deprecated "align" attribute.

Maintainability 60

Consistency html5 obsolete

Open Not assigned

L12 × 5min effort × 4 years ago • Code Smell • Major

gameoflife-core/build/reports/tests/alltests-xml.html

Remove this deprecated "size" attribute.

Maintainability 60

Consistency html5 obsolete

Open Not assigned

Intentional issues

The screenshot shows the SonarQube interface for the project `gameoflife-acceptance-tests/Dockerfile`. The sidebar on the left displays navigation links like 'Issues in new code', 'Clean Code Attribute', 'Software Quality', 'Severity', and 'Type'. Under 'Clean Code Attribute', 'Intentionality' is selected, showing 14k issues. The main panel lists several code smells under the 'Intentionality' category:

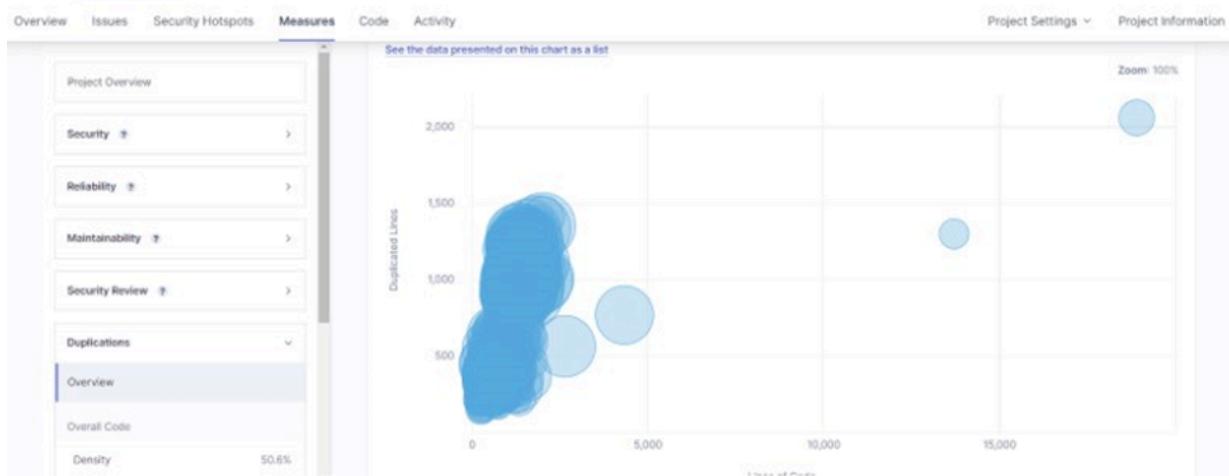
- Use a specific version tag for the image. (Maintainability) Intentionality No tags · L1 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Intentionality No tags · L12 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Intentionality No tags · L12 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Intentionality No tags · L12 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Major

Reliabilities issue

The screenshot shows the SonarQube interface for the project `gameoflife-core/build/reports/tests/all-tests.html`. The sidebar on the left displays navigation links like 'Issues in new code', 'Clean Code Attribute', 'Software Quality', 'Severity', and 'Type'. Under 'Clean Code Attribute', 'Reliability' is selected, showing 54k issues. The main panel lists several code smells under the 'Reliability' category:

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability) Consistency user-experience · L1 < 5min effort · 4 years ago · ⚡ Bug · ⚡ Major
- Anchors must have content and the content must be accessible by a screen reader. (Maintainability) Reliability Consistency accessibility · L29 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Minor
- Anchors must have content and the content must be accessible by a screen reader. (Maintainability) Reliability Consistency accessibility · L38 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Minor
- Anchors must have content and the content must be accessible by a screen reader. (Maintainability) Reliability Consistency accessibility · L38 < 5min effort · 4 years ago · ⚡ Code Smell · ⚡ Minor

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, we successfully integrated Jenkins with SonarQube to automate code quality checks within our CI/CD pipeline. We began by deploying SonarQube using Docker, configuring a project, and setting it up for comprehensive code analysis. Following this, we configured Jenkins by installing the SonarQube Scanner plugin, providing the necessary SonarQube server details, and establishing the scanner tool. We then created a Jenkins pipeline to streamline the process of cloning a GitHub repository and executing SonarQube analysis on the code. This integration facilitates continuous monitoring of code quality, enabling us to identify issues such as bugs, code smells, and security vulnerabilities throughout the development lifecycle.

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

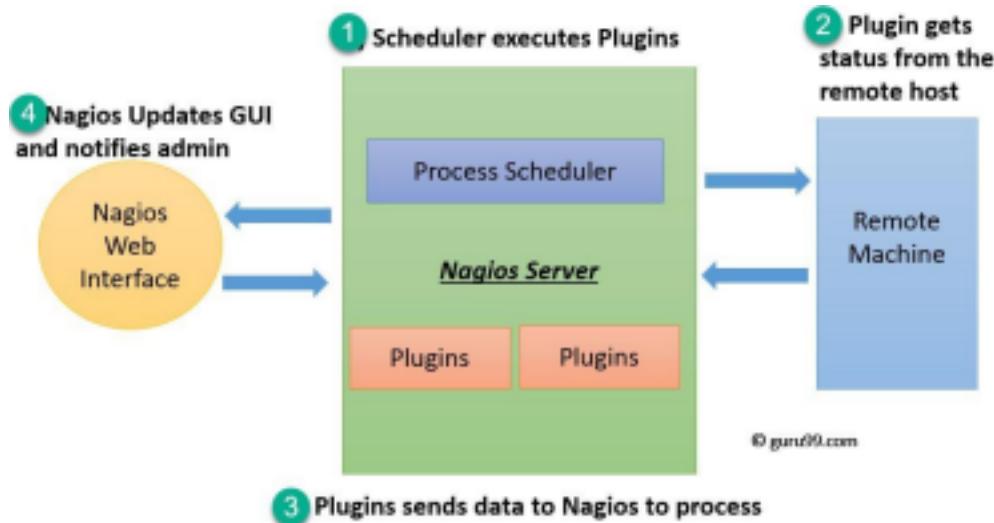
Following are the important features of Nagios monitoring tool:

- 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

I have created a new security group with a name 'newsecurity'

The screenshot shows the 'Create security group' wizard. At the top, there's a breadcrumb navigation: EC2 > Security Groups > Create security group. Below it, the title 'Create security group' has an 'Info' link. A descriptive text explains that a security group acts as a virtual firewall. The main section is titled 'Basic details' and contains a 'Security group name' field with 'Nagios' entered. A note below says 'Name cannot be edited after creation.'

I have modified the INBOUND RULES as follows

The screenshot shows the 'Inbound rules' list. At the top, there are tabs for 'Inbound rules' (which is selected), 'Outbound rules', and 'Tags'. Below the tabs, there's a search bar and a row of buttons: a refresh icon, 'Manage tags', and 'Edit inbound rules'. The main area displays a table of seven inbound rules. The columns are: Name, Security group rule..., IP version, Type, Protocol, and Port range. The rules are:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-041bc9f3bed4c124f	IPv6	All ICMP - IPv6	IPv6 ICMP	All
-	sgr-085dfc9eb063517e8	IPv4	HTTP	TCP	80
-	sgr-01afcfc27796f6bccf	IPv4	Custom TCP	TCP	5666
-	sgr-022555b6d76770...	IPv4	All ICMP - IPv4	ICMP	All
-	sgr-0cae92323695518...	IPv4	HTTPS	TCP	443
-	sgr-04169179aa4e58f18	IPv4	SSH	TCP	22
-	sgr-09646db0f6ed0ff0a	IPv4	All traffic	All	All

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
 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 (AMI)

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](#)

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)
vpc-00b2e9bc41e6d48bf

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

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

Nagios sg-0ea382a8493535f45 [X](#)
VPC: vpc-00b2e9bc41e6d48bf

[Compare security group rules](#)

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

Connect to instance [Info](#)

Connect to your instance i-07aa5e94e7f0c3ac2 (Nagios-host) using any of these options

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

Instance ID
 i-07aa5e94e7f0c3ac2 (Nagios-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Nagios.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Nagios.pem"
4. Connect to your instance using its Public DNS:
 ec2-18-212-111-59.compute-1.amazonaws.com

Example:
 ssh -i "Nagios.pem" ec2-user@ec2-18-212-111-59.compute-1.amazonaws.com

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

```
[ec2-user@ip-172-31-92-249~]
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

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
~~ v|'-'>
~~ /|/
~~ .-' /|/
~~ /| /|/
|m'/

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
~~ v|'-'>
~~ /|/
~~ .-' /|/
~~ /| /|/
|m'/

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

```
[ec2-user@ip-172-31-41-160~]$ sudo yum update
Last metadata expiration check: 0:01:37 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-41-160 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:45 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.

=====
Package           Architecture Version       Repository   Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023 amazonlinux 48 k
php8.3           x86_64      8.3.10-1.amzn2023.0.1    10 k

Installing dependencies:
argon2          x86_64      1.7.2-2.amzn2023.0.2 amazonlinux 120 k
curl             x86_64      1.6.3-1.amzn2023.0.1 amazonlinux 98 k
curl-util        x86_64      18.0.0-12.amzn2023.0.3 amazonlinux 19 k
generic-logos-httdt x86_64      2.4.62-1.amzn2023 amazonlinux 1.4 M
httpd-core       x86_64      2.4.62-1.amzn2023 amazonlinux 14 k
httpd-filesystem x86_64      2.4.62-1.amzn2023 amazonlinux 53 k
httpd-tools      x86_64      1.6.9-4.amzn2023.0.2 amazonlinux 315 k
libbsdodium      x86_64      1.0.19-4.amzn2023 amazonlinux 176 k
libedit          x86_64      1.1.34-5.amzn2023.0.2 amazonlinux 241 k
mod_wsgi         noarch     2.1.49-3.amzn2023.0.3 amazonlinux 33 k
nginx-filesystem x86_64      1.13.10-1.amzn2023.0.4 amazonlinux 9.8 k
php8.3-cgi      x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 3.7 M
php8.3-common   x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 737 k
php8.3-process  x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 45 k
php8.3-xml      x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 154 k

Installing weak dependencies:
argon2          x86_64      1.6.3-1.amzn2023.0.1 amazonlinux 17 k
curl             x86_64      2.6.27-1.amzn2023.0.3 amazonlinux 166 k
curl-util        x86_64      2.4.62-1.amzn2023 amazonlinux 61 k
httpd            x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 1.9 M
httpd-core       x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 528 k
httpd-filesystem x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 379 k
httpd-tools      x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 89 k
mod_wsgi         x86_64      8.3.10-1.amzn2023.0.1 amazonlinux 41 k

Transaction Summary
Install 25 Packages
```

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 | 105 k |
| glibc-devel      | x86_64      | 2.34-52.amzn2023.0.11 | amazonlinux | 27 k |
| glibc-headers-x86 | noarch      | 2.34-52.amzn2023.0.11 | amazonlinux | 427 k |
| guile22         | x86_64      | 2.2.7-2.amzn2023.0.3 | amazonlinux | 6.4 M |
| kernel-headers   | x86_64      | 6.1.109-118.189.amzn2023 | amazonlinux | 1.4 M |
| libmpc           | x86_64      | 1.2.1-2.amzn2023.0.2 | amazonlinux | 62 k |
| libtool-ltdl    | x86_64      | 2.4.7-1.amzn2023.0.3 | amazonlinux | 38 k |
| libxcrypt-devel  | x86_64      | 4.4.33-7.amzn2023 | amazonlinux | 32 k |
| make             | x86_64      | 1:4.3-5.amzn2023.0.2 | amazonlinux | 534 k |
|=====|
| Transaction Summary |             |              |            |       |
|=====|
| Install 13 Packages |             |              |            |       |
| Total download size: 52 M |             |              |            |       |
| Installed size: 168 M |             |              |            |       |
| Is this ok [y/N]: 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 kB/s | 10 MB | 00:00 |
| (6/13): glibc-headers-x86-2.34-52.amzn2023.0.11.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): libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64.rpm |             | 322 kB/s | 38 kB | 00:00 |
| (11/13): libxcrypt-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: 0:02:25 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      | 1:2.0.5-12.amzn2023.0.2 | amazonlinux | 9.5 k |
| freetype           | x86_64      | 2.13.2-5.amzn2023.0.1 | amazonlinux | 423 k |
| freetype-devel    | x86_64      | 2.13.2-5.amzn2023.0.1 | amazonlinux | 912 k |
| glib2-devel       | x86_64      | 2.71-7.689.amzn2023.0.2 | amazonlinux | 486 k |
| google-noto-fonts-common | noarch      | 20201206-2.amzn2023.0.2 | amazonlinux | 15 k |
| google-noto-sans-vf-fonts | noarch      | 20201206-2.amzn2023.0.2 | amazonlinux | 492 k |
| graphite2          | x86_64      | 1.3.14-7.amzn2023.0.2 | amazonlinux | 97 k |
| graphite2-devel   | x86_64      | 1.3.14-7.amzn2023.0.2 | amazonlinux | 21 k |
| harfbuzz           | x86_64      | 7.0.0-2.amzn2023.0.1 | amazonlinux | 868 k |
| harfbuzz-devel    | x86_64      | 7.0.0-2.amzn2023.0.1 | amazonlinux | 404 k |
| harfbuzz-icu      | x86_64      | 7.0.0-2.amzn2023.0.1 | amazonlinux | 18 k |
| jbigkit-libs       | x86_64      | 2.1-21.amzn2023.0.2 | amazonlinux | 54 k |
| langpacks-core-font-en | noarch      | 3.0-21.amzn2023.0.4 | amazonlinux | 10 k |
| libICE             | x86_64      | 1.0.10-6.amzn2023.0.2 | amazonlinux | 71 k |
| libSM              | x86_64      | 1.2.3-8.amzn2023.0.2 | amazonlinux | 42 k |
| libX11             | x86_64      | 1.7.2-3.amzn2023.0.4 | amazonlinux | 657 k |
| libX11-common      | noarch      | 1.7.2-3.amzn2023.0.4 | amazonlinux | 152 k |
| libX11-devel       | x86_64      | 1.7.2-3.amzn2023.0.4 | amazonlinux | 939 k |
| libX11-xcb         | x86_64      | 1.7.2-3.amzn2023.0.4 | amazonlinux | 12 k |
| libXau             | x86_64      | 1.0.9-6.amzn2023.0.2 | amazonlinux | 31 k |
| libXau-devel       | x86_64      | 1.0.9-6.amzn2023.0.2 | amazonlinux | 14 k |
| libXext             | x86_64      | 1.3.4-6.amzn2023.0.2 | amazonlinux | 41 k |
| libXpm              | x86_64      | 3.5.15-2.amzn2023.0.3 | amazonlinux | 65 k |
| libXpm-devel       | x86_64      | 3.5.15-2.amzn2023.0.3 | amazonlinux | 59 k |
| libXrender          | x86_64      | 0.9.10-14.amzn2023.0.2 | amazonlinux | 28 k |
| libXt               | x86_64      | 1.2.0-4.amzn2023.0.2 | amazonlinux | 181 k |
| libblkid-devel     | x86_64      | 2.37.4-1.amzn2023.0.4 | amazonlinux | 15 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>

```
ec2-user@ip-172-31-41-160:~/downloads$ 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_T3NMNSZPzP-61a2l1tv0o0CG7VVV7QGH08n3tC240ehfMw7vhcoKbHg2iIRxbmfugII0LccnFxta0ixg3jzKg3w3D0use_mirror-phoenixnaxpR= [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_T3NMNSZPzP-61a2l1tv0o0CG7VVV7QGH08n3tC240ehfMw7vhcoKbHg2iIRxbmfugII0LccnFxta0ixg3jzKg3w3D0use_mirror-phoenixnaxpR=
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?viasf=1 [following]
--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?viasf=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:~/downloads$ 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 (2.0M) [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-nVw9RdAvnMaShLf3gu4RXTSVxrTz6fGxJvhVA0zpB1bPgbyzLMcDDAALgtEC1pOKr0cgJNj23bKktaricJQYVfkgs30%20use_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-nVw9RdAvnMaShLf3gu4RXTSVxrTz6fGxJvhVA0zpB1bPgbyzLMcDDAALgtE
Cip0Kr0cgJNj23bKktaricJQYVfkgs30%20use_mirror=netactuate&e=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|: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 (2.0M) [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 ~]$ 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/nagios.stats.c
nagios-4.5.5/base/nebmods.c
nagios-4.5.5/base/nero.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]$ ./configure
checking for strstr... yes
checking for strtoul... yes
checking for unsetenv... yes
checking for type of socket size... size_t
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-41-160 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: 0:02:11 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-3.0.8-1.amzn2023.0.14.x86_64.rpm
                                                               26 MB/s | 3.0 MB  00:00
                                                               17 MB/s | 3.0 MB  00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
    Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
    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
 1/1
 1/1
 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$ make all
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ make all
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 -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o common/shared.o ./common/shared.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: %s directive argument is null [-Wformat-overflow=]
  253 |   log_debug_info(DEBUGL_CHECKS, 1, "found specialized worker(s) for '%s'", (slash & *slash != '/') ? slash : cmd_name);
   |   ~~~~~
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o logger.o logger.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros.o macros.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netutils.o netutils.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o notifications.o notifications.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o seahandlers.o seahandlers.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o utils.o utils.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o retention-base.o ./retention.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o xretention-base.o ./xdata/xrdddefault.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o comments-base.o ./common/comments.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o xcomments-base.o ./xdata/xcddefaul.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o objects-base.o ./common/objects.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o xobjects-base.o ./xdata/xodtemplate.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o statusdata-base.o ./common/statusdata.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o xstatusdata-base.o ./xdata/xsddefault.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o perfdata-base.o ./perfdata.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o xperfdata-base.o ./xdata/xpddefault.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -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/lib'
  - 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-classic
  - 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 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
[ec2-user@ip-172-31-41-160 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
[ec2-user@ip-172-31-41-160 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 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.cfg /usr/local/nagios/etc/cgi.cfg
/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:

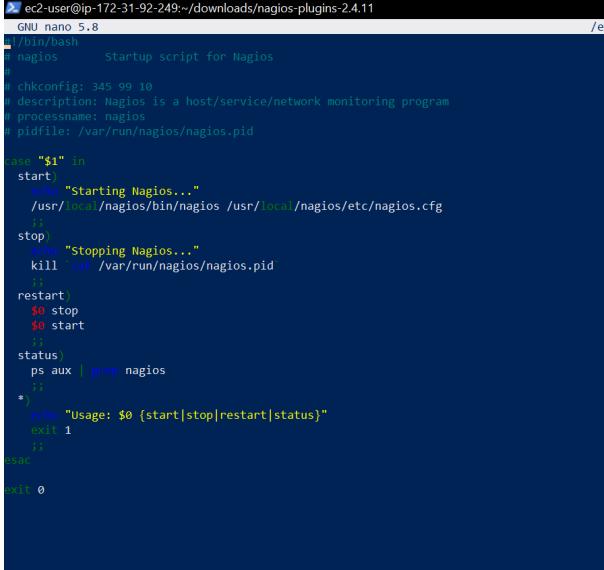
Run the following command:

```
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
```



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]$
```

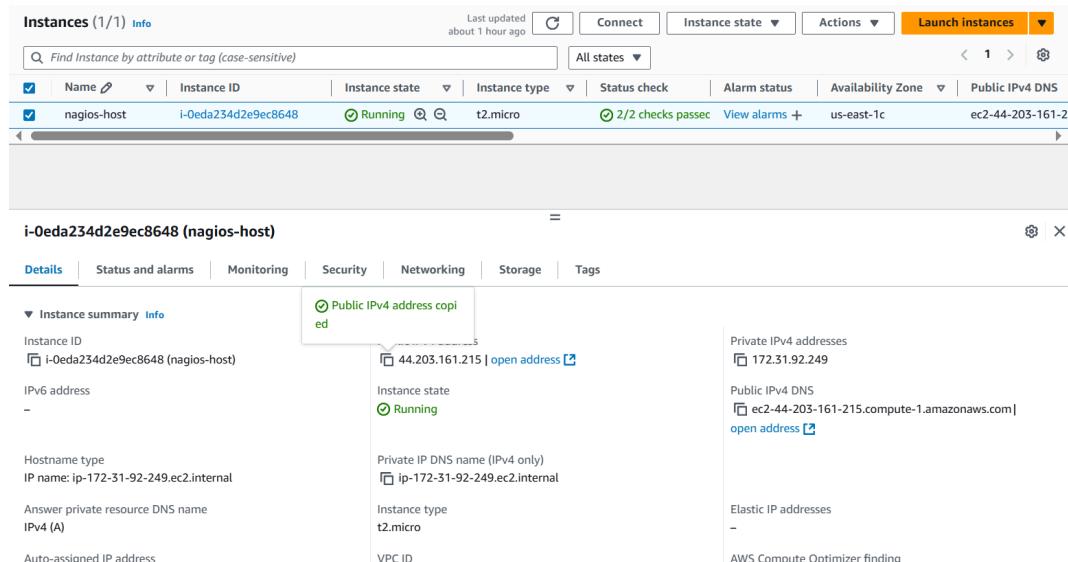
sudo service nagios start

```
[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

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 web interface. The top navigation bar indicates the URL as 34.229.45.75/nagios/. The main header features the Nagios Core logo with a gear icon. A green checkmark message says "Process running with PID 62668". The left sidebar contains a navigation menu with sections like General, Current Status, Service Groups, Problems, Reports, and a search bar. The Current Status section is expanded, showing links for Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid, and Service Groups. The Service Groups section is currently selected. The main content area includes a "Get Started" box with bullet points about monitoring, 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 "Latest News" and "Don't Miss..." sections, both of which are currently empty.

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).

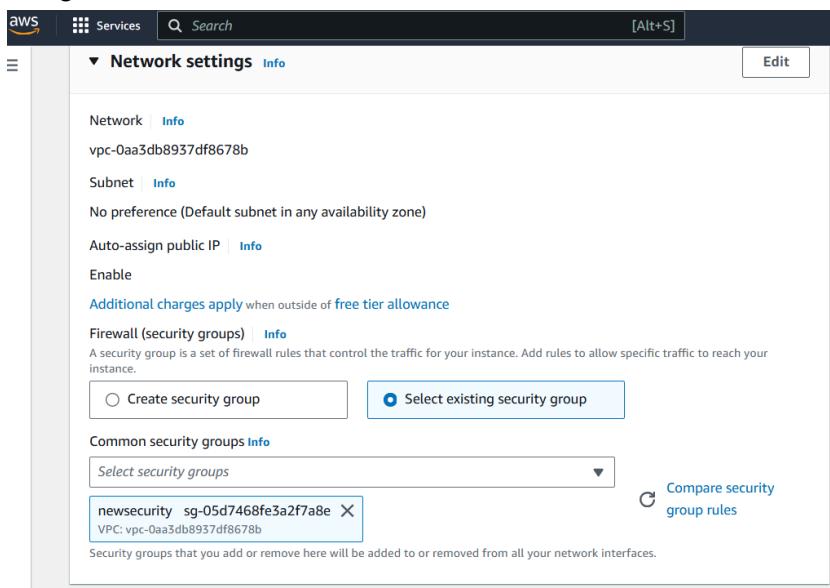
Run this command **sudo systemctl status**

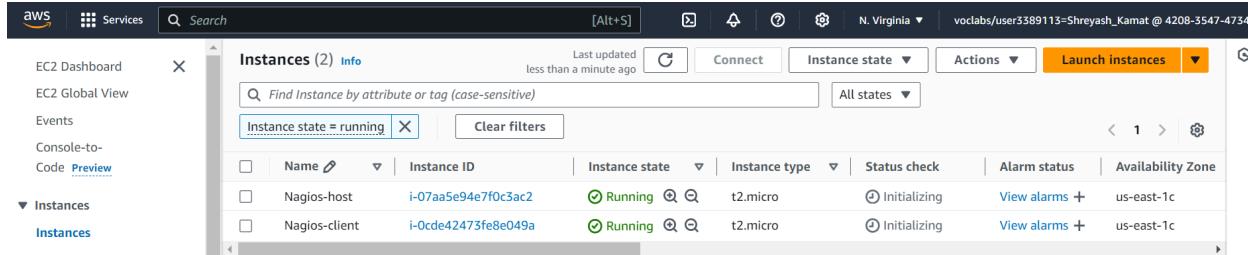
```
ec2-user@ip-172-31-41-160:~$ sudo systemctl status
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo systemctl status
● ip-172-31-41-160.ec2.internal
    State: running
      Units: 296 loaded (incl. loaded aliases)
        Jobs: 0 queued
       Failed: 0 units
     Since: Wed 2024-10-02 12:28:05 UTC; 33min ago
    Systemd: 252.23-2.amzn2023
   CGroup: /
           └─init.scope
             ├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
             ├─system.slice
             ├─acpid.service
             ├─amazon-ssm-agent.service
             ├─atd.service
             ├─auditd.service
             ├─chronynd.service
             ├─dbus-broker.service
             ├─gssproxy.service
             ├─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.





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/linuxhosts/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-server1         ; The name of the hostgroup
    alias        localhost             ; Long name of the group
    address      54.172.92.226        ; Comma separated list of hosts that belong to this group
}

#####
# Define an optional hostgroup for Linux machines

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

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 timperiods
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
[root@ip-172-31-41-160 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 4.5
   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: http://www.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

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 as @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]: wproc: 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 13:20:21 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)
     Drop-In: /usr/lib/systemd/system/httpd.service.d
       └─php-fpm.conf
     Active: active (running) since Wed 2024-10-02 12:47:56 UTC; 33min ago
       Docs: man:httpd.service(8)
     Main PID: 49553 (httpd)
       Status: "Total requests: 26; Idle/Busy workers 100/0;Requests/sec: 0.0129; Bytes served/sec: 94.8/sec"
     Tasks: 21 (limit: 1112)
     Memory: 21.7M
       CPU: 1.41ms
     CGroup: /system.slice/httpd.service
             └─49553 /usr/sbin/httpd -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/WkFQRuHSpqjqq5hdMaA0dku8msNheTN9SAgZey53E.
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
law.
```

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

```
ubuntu@ip-172-31-36-100:~$ sudo apt update -y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [388 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4576 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [275 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-security/universe amd64 Packages [8659 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.0 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2888 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [381 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [110 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [6676 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [388 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [157 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3688 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
```

```
ubuntu@ip-172-31-36-100:~$ apt update
ubuntu@ip-172-31-36-100:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core fonts-delaval-mono gcc-13 gcc-13-base
  gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libbam3 libasan libatomici libbinutils libc-dev-bin libc-devtools libc6-dev libgcc1-0 libcrypt-dev libctf-nobfd libctf0 libde265-0
  libdeflate libfontconfig libgcc-13-dev libgd3 libgomp1 liblprofng0 liblheif-plugin-aomenc liblheif-plugin-aomenc liblheif-plugin-libvpx liblhwasan0 libls23 libitm libjbg0
  libjpeg-turbo libjpeg0 liblrcb1 liblrcb3 liblrcdmath liblrcfamei liblsharpyuv0 liblttff libtsan2 libubsan1 libwepy1 libxpm4 linux-lc-dev manpages-dev rpcsvc-proto
Suggested packages:
  binutils-gdb gprofng gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdc gdc-doc gcc-13-multilib gcc-13-doc gdb-x86_64-linux-gnu glibc-doc
  liblheif1 liblheif-plugin-x265 liblheif-plugin-ffmpegdec liblheif-plugin-jpgedec liblheif-plugin-jpgedec liblheif-plugin-jx2dec liblheif-plugin-jxenc liblheif-plugin-ravie
liblheif-plugin-svc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core fonts-delaval-mono gcc gcc-13 gcc-13-base
  gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libbam3 libasan libatomici libbinutils libc-dev-bin libc-devtools libc6-dev libgcc1-0 libcrypt-dev libctf-nobfd libctf0 libde265-0
  libdeflate libfontconfig libgcc-13-dev libgd3 libgomp1 liblprofng0 liblheif-plugin-aomenc liblheif-plugin-aomenc liblheif-plugin-libvpx liblhwasan0 libls23 libitm libjbg0
  libjpeg-turbo libjpeg0 liblrcb1 liblrcb3 liblrcdmath liblrcfamei liblsharpyuv0 liblttff libtsan2 libubsan1 libwepy1 libxpm4 linux-lc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 67.8 MB of archives.
```

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
      CpuTime: 1.000us
     Corups: /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 13:22:54 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 kB)
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=2512; 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 = RSZDT
Oct 02 13:23:50 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Load;OK;HARD;1;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

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark indicates "Daemon running with PID 78778". The left sidebar contains a navigation menu with sections like General, Current Status, Reports, and Problems. The "Current Status" section is active, showing links for Tactical Overview, Map, Hosts, Services, Host Groups, and more. Below the sidebar are several informational boxes: "Get Started" with a list of steps, "Latest News" (empty), and "Don't Miss..." (empty). A "Quick Links" box on the right provides links to Nagios Library, Labs, Exchange, Support, and the official website.

Click on Hosts.

This screenshot shows the "Host Status Details For All Host Groups" section. It includes two summary tables: "Host Status Totals" and "Service Status Totals". The "Host Status Totals" table shows 2 hosts up, 0 down, 0 unreachable, and 0 pending. The "Service Status Totals" table shows 12 services ok, 1 warning, 0 unknown, 3 critical, and 0 pending. Below these are two tables: "Host Status Details For All Host Groups" and "Service Status Details For All Services". The "Host Status Details For All Host Groups" table lists two hosts: "linux-server" and "localhost", both marked as "UP". The "Service Status Details For All Services" table lists various services with their status, last check time, duration, and status information. A "Page tour" link is visible on the right side of the page.

Click on linux-server to view host information

The screenshot shows the Nagios web interface at 34.229.45.75/nagios/. The left sidebar has sections for General, Current Status (selected), and Reports. The main content area shows 'Host Information' for 'localhost' (linux-server). It includes details like last update, check attempts, and current status (UP). A 'Host State Information' section provides a detailed breakdown of the host's status. On the right, there's a 'Host Commands' panel with various options for managing the host. At the bottom, there are 'Host Comments' and system status indicators.

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 at 34.229.45.75/nagios/. The left sidebar has sections for General, Current Status (selected), and Reports. The main content area shows 'Service Status Details For All Hosts'. It lists services for 'localhost' and 'linux-server' with their respective statuses, last checks, and durations. The 'Status Information' column provides detailed logs for each service. A 'Results 1 - 16 of 16 Matching Services' message is at the bottom.

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.

Theory:

AWS Lambda

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

Lambda Workflow

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

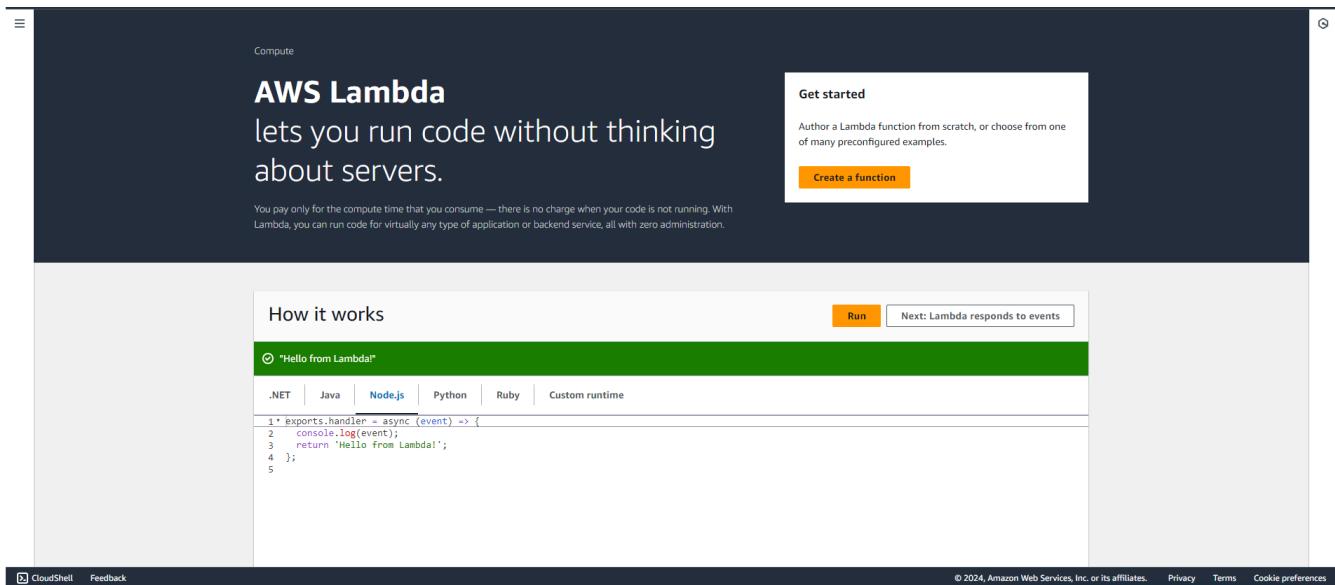
AWS Lambda Functions

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

Prerequisites: AWS Personal/Academy Account

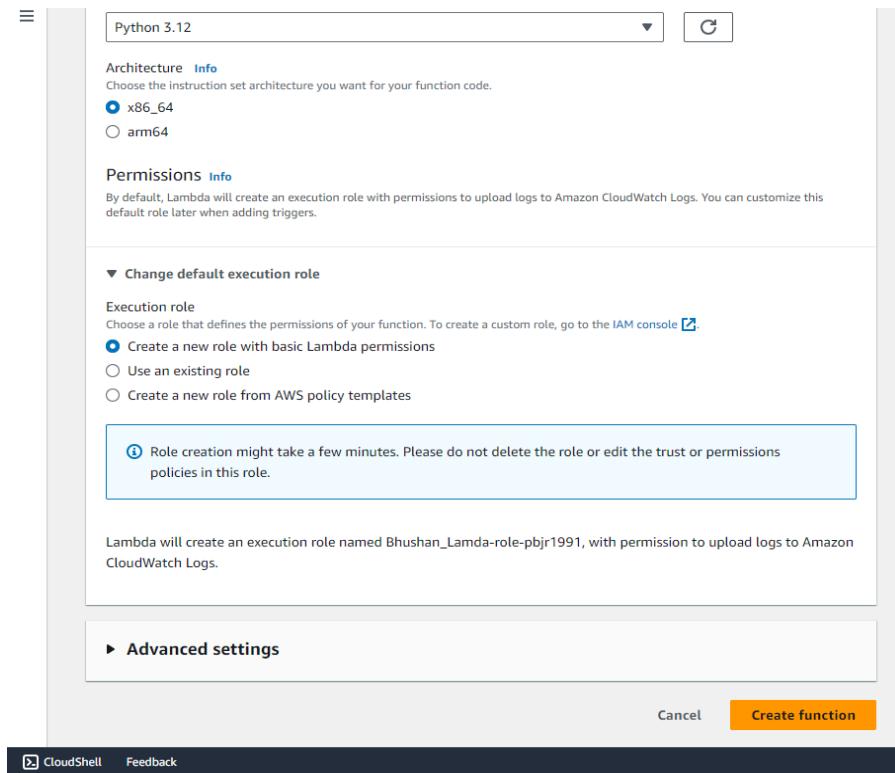
Steps To create the lambda function:

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



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

A screenshot of the 'Create function' wizard in the AWS Lambda console. The top navigation bar shows 'Lambda > Functions > Create function'. The main title is 'Create function' with an 'Info' link. Below it, a note says 'Choose one of the following options to create your function.' There are four options: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. The 'Basic information' section includes fields for 'Function name' (set to 'MyLambda'), 'Runtime' (set to 'Python 3.12'), and 'Architecture' (set to 'x86_64').



The screenshot shows the 'myLambda' function overview page. The top navigation bar includes 'Lambda > Functions > myLambda'. On the right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. The main area shows the function name 'myLambda' and a 'Layers' section with '(0)'. A 'Diagram' tab is selected. On the right, there's a sidebar with 'Description', 'Last modified' (4 minutes ago), 'Function ARN' (arn:aws:lambda:eu-north-1:860015268757:function:myLambda), and 'Function URL' (Info). Buttons for 'Export to Application Composer' and 'Download' are also present.

The screenshot shows the code editor for the 'myLambda' function. The top navigation bar includes 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code source' tab is selected. The code editor interface has tabs for 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test', and 'Deploy'. The code editor shows a file named 'lambda_function.py' with the following content:

```
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
```

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

The screenshot shows the AWS Lambda configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is highlighted in blue), Aliases, and Versions. On the left, a sidebar lists various configuration sections: General configuration (selected), Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area is titled "General configuration" and contains the following details:

Description	Memory	Ephemeral storage
-	128 MB	512 MB

Below this, there are fields for "Timeout" (set to "0 min 3 sec") and "SnapStart" (set to "None"). A large "Edit" button is located in the top right corner of the configuration panel.

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

The screenshot shows the "Edit basic settings" dialog box. The title is "Edit basic settings". The main section is titled "Basic settings" and contains the following fields:

- Description - optional:** Basic Settings
- Memory Info:** Your function is allocated CPU proportional to the memory configured. Value: 128 MB. Note: Set memory to between 128 MB and 10240 MB.
- Ephemeral storage Info:** You can configure up to 10 GB of ephemeral storage (/tmp) for your function. Value: 512 MB. Note: Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.
- SnapStart Info:** Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations. Value: None. Note: Supported runtimes: Java 11, Java 17, Java 21.
- Timeout:** Value: 0 min 1 sec.
- Execution role:** Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console. Options:
 - Use an existing role
 - Create a new role from AWS policy templates

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

Test event [Info](#)

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

Test event action

Create new event Edit saved event

Event name

MyEvent

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

Event sharing settings

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

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

Template - optional

hello-world

Event JSON

```
1 [ ]  
2   "key1": "value1",  
3   "key2": "value2",
```

Format JSON

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

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

Upload from ▾

File Edit Find View Go Tools Window Test Deploy

Configure test event Ctrl-Shift-C

• (unsaved) test event

Private saved events

MyEvent

Environment

myLambda - /

lambda_function.py

```
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
```

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

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

```

import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from D15C-12,15,22!')
    }

```

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

Execution results

Test Event Name
MyEvent

Response

```
{
  "statusCode": 200,
  "body": "Hello from D15C-12,15,22!"
}
```

Function Logs

START RequestId: 98e8ae50-5143-4945-9617-f127c5365ee6 Version: \$LATEST
END RequestId: 98e8ae50-5143-4945-9617-f127c5365ee6
REPORT RequestId: 98e8ae50-5143-4945-9617-f127c5365ee6 Duration: 1.92 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 94.54 ms

Request ID
98e8ae50-5143-4945-9617-f127c5365ee6

Conclusion:

In this experiment, we successfully implemented an AWS Lambda function, covering all the key steps involved. Starting with the function's setup in Python, we configured essential settings such as adjusting the timeout to 1 second. A test event was then created, followed by deploying the function and verifying its output. We also made code modifications to the Lambda function, redeployed it, and observed the real-time effects of these changes. This hands-on experience highlighted the ease and adaptability of AWS Lambda for building serverless applications, enabling developers to concentrate on writing code while AWS handles infrastructure and scalability.

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

Theory:

AWS Lambda and S3 Integration:

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

Workflow:

1. Create an S3 Bucket:

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

2. Create the Lambda Function:

- Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

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

4. Configure S3 Trigger:

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

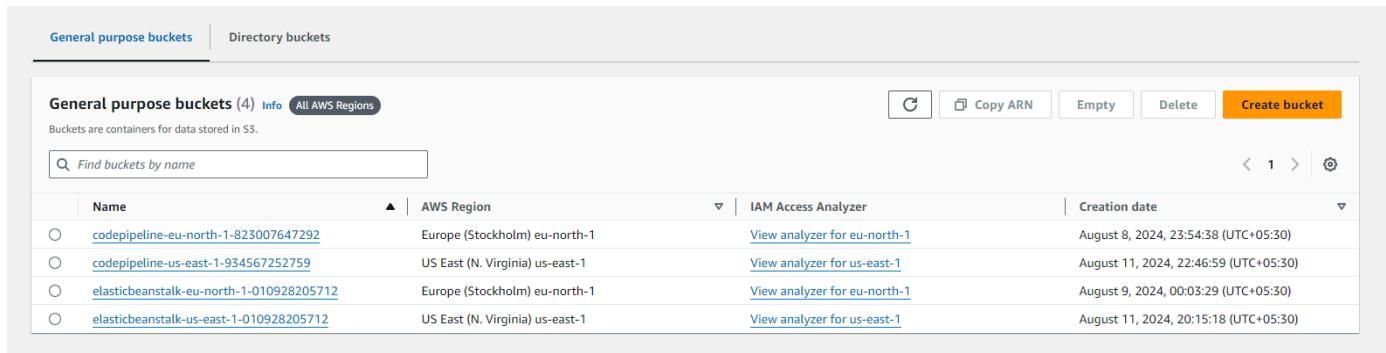
5. Test the Setup:

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

Prerequisites: AWS Personal Account

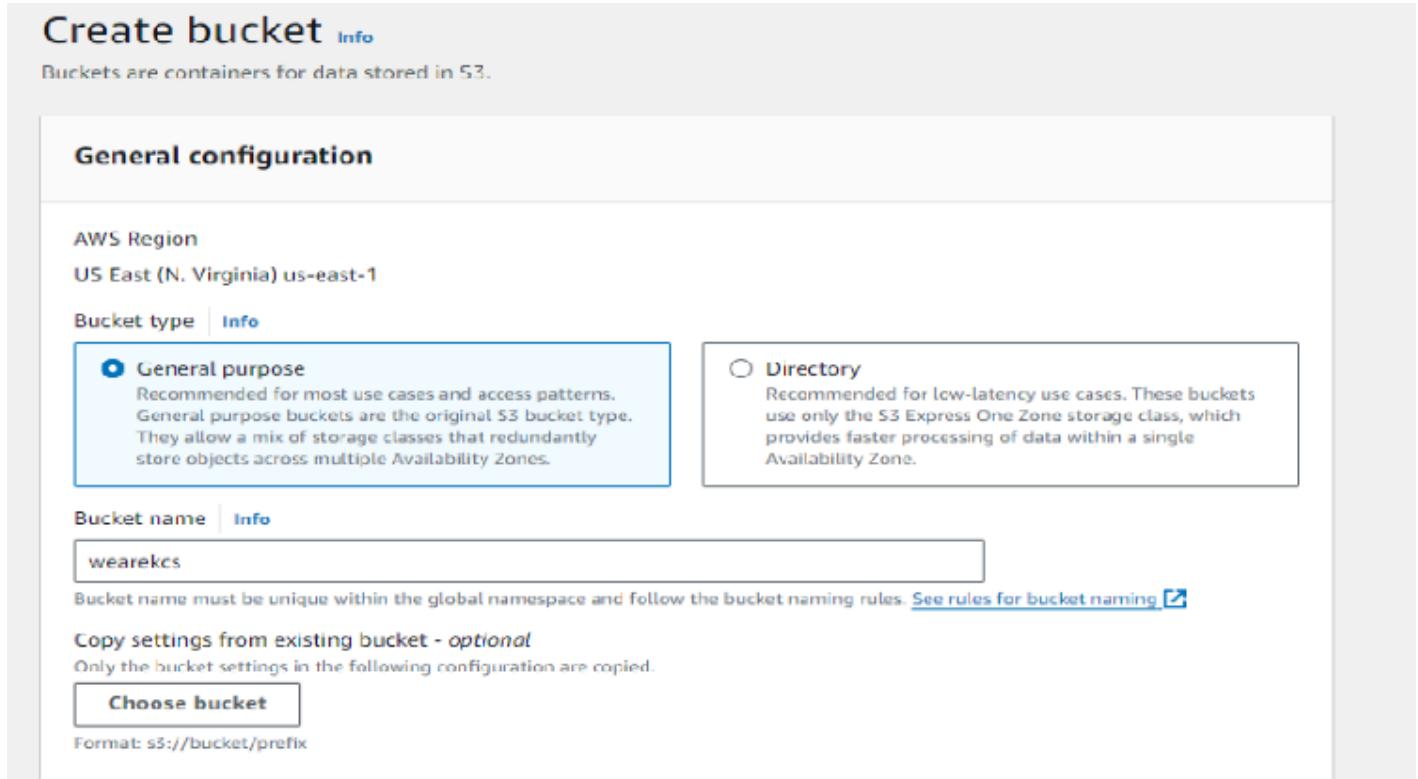
Steps To create the lambda function:

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



The screenshot shows the AWS S3 console with the 'General purpose buckets' tab selected. It displays a list of four existing buckets: 'codepipeline-eu-north-1-823007647292', 'codepipeline-us-east-1-934567252759', 'elasticbeanstalk-eu-north-1-010928205712', and 'elasticbeanstalk-us-east-1-010928205712'. Each entry includes the name, AWS Region, IAM Access Analyzer link, and creation date. A 'Create bucket' button is visible at the top right.

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



The screenshot shows the 'Create bucket' configuration page. Under 'General configuration', the 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. The 'Bucket type' is set to 'General purpose', which is highlighted with a blue border. The 'Bucket name' field contains 'wearekcs'. A note below the name states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'.

General configuration

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

Bucket type: General purpose

Bucket name: wearekcs

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
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.

Successfully created bucket "weeekis" [View details](#) [X](#)

In upload files and folders, or to configure additional bucket settings, choose [View details](#).

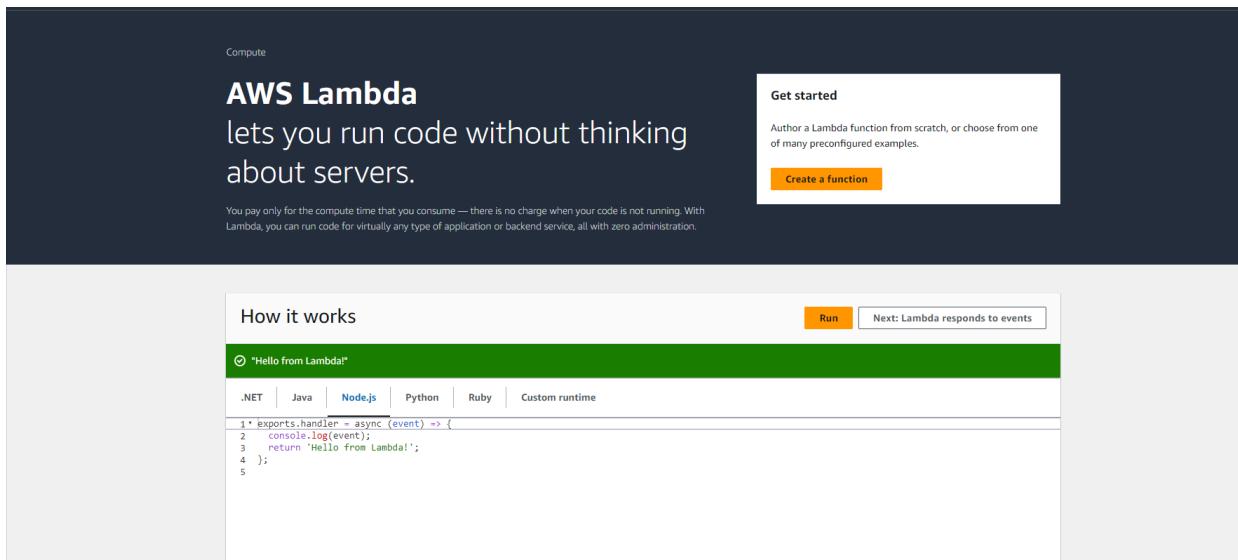
[Amazon S3](#) > Buckets

Account snapshot - updated every 24 hours [View details](#)
Storage lens provides visibility into storage usage and activity trends. [Learn more](#) [View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (1) Info All AWS Regions		Copy ARN	Empty	Delete	Create bucket
Bucket ARN: arn:aws:s3:::weeekis Bucket last modified: 10/1/2024 at 13:40:40 UTC+05:30		Edit	Empty	Delete	Create bucket
Find buckets by name		Name AWS Region IAM Access Analyzer Creation date			
weeekis		US East (N. Virginia) us-east-1	View analyzer for weeekis	October 1, 2024, 13:40:40 UTC+05:30	

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



Step 4: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12

, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

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.
- Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

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

Python 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.

x86_64
 arm64

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

▼ Change default execution role

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

Create a new role with basic Lambda permissions
 Use an existing role
 Create a new role from AWS policy templates

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

Lambda will create an execution role named Bhushan_Lambda-role-pbjr1991, with permission to upload logs to Amazon CloudWatch Logs.

► Advanced settings

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

[CloudShell](#) [Feedback](#)

[Lambda](#) > [Functions](#) > myLambda

myLambda

[Throttle](#) [Copy ARN](#) [Actions ▾](#)

▼ Function overview Info

[Diagram](#) [Template](#)

 **myLambda**
[Layers](#) (0)

[+ Add trigger](#) [+ Add destination](#)

[Export to Application Composer](#) [Download ▾](#)

Description
-

Last modified
4 minutes ago

Function ARN
[arn:aws:lambda:eu-north-1:860015268757:function:myLambda](#)

Function URL Info
-

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

Code source Info

[Upload from ▾](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test ▾](#) [Deploy](#)

Environment
myLambda /
[lambda_function](#) x Environment Vari +

```

lambda_function Environment Vari +
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

```

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

The screenshot shows the AWS Lambda Configuration page. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected and highlighted in blue), Aliases, and Versions. On the left, a sidebar menu lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area is titled "General configuration" and contains fields for Description (set to "-"), Memory (set to 128 MB), and Ephemeral storage (set to 512 MB). An "Edit" button is located in the top right corner of this section.

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

The screenshot shows the "Edit basic settings" page for a function named "Bhushan_Lambda". The top navigation bar shows the path: Lambda > Functions > Bhushan_Lambda > Edit basic settings. The main section is titled "Basic settings" and contains the following configuration options:

- Description - optional:** A text input field containing "Basic Settings".
- Memory:** Set to 128 MB. A note states: "Your function is allocated CPU proportional to the memory configured." A dropdown menu allows selecting memory sizes between 128 MB and 10240 MB.
- Ephemeral storage:** Set to 512 MB. A note states: "You can configure up to 10 GB of ephemeral storage (/tmp) for your function." A dropdown menu allows selecting storage sizes between 512 MB and 10240 MB.
- SnapStart:** Set to "None". A note states: "Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations." A dropdown menu allows selecting "None", "Java 8", "Java 11", "Java 17", or "Java 21".
- Timeout:** Set to 0 min 1 sec.
- Execution role:** A radio button group where "Use an existing role" is selected (indicated by a blue circle). The other option, "Create a new role from AWS policy templates", is unselected.

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

Test event [Info](#)

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

Test event action Create new event Edit saved event

Event name

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

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

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

Template - optional

Event JSON [Format JSON](#)

```

1 * []
2   "key1": "value1",
3     "key2": "value2",

```

Services [Search](#) [Alt+S]

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

Template - optional

Event JSON [Format JSON](#)

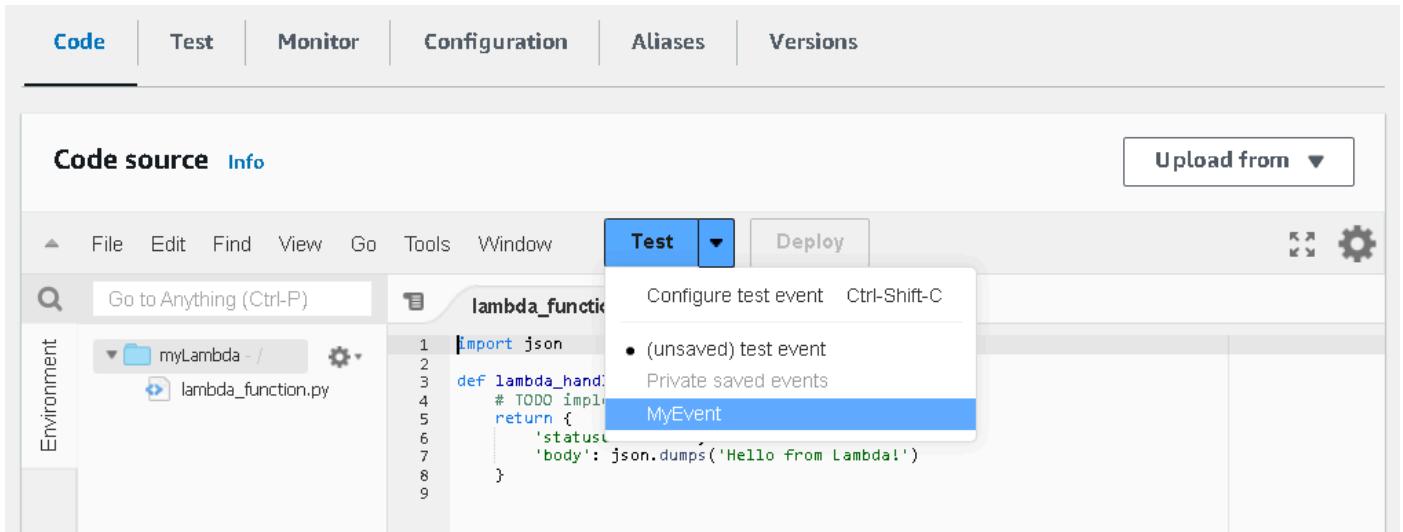
```

2 * "Records": [
3   {
4     "eventVersion": "2.0",
5     "eventSource": "aws:s3",
6     "awsRegion": "us-east-1",
7     "eventTime": "1970-01-01T00:00:00.000Z",
8     "eventName": "ObjectCreated:Put",
9     "userIdentity": {
10       "principalId": "EXAMPLE"
11     },
12     "requestParameters": {
13       "sourceIPAddress": "127.0.0.1"
14     },
15     "responseElements": {
16       "x-amz-request-id": "EXAMPLE123456789",
17       "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmnaaaaaaaaaaaaaaaa"
18     },
19     "s3": {
20       "s3SchemaVersion": "1.0",
21       "configurationId": "testConfigRule",
22       "bucket": {
23         "name": "example-bucket",
24         "ownerIdentity": {
25           "principalId": "EXAMPLE"
26         },
27         "arn": "arn:aws:s3:::example-bucket"
28       },
29       "object": {
30         "key": "test%2Fkey",
31         "size": 1024,

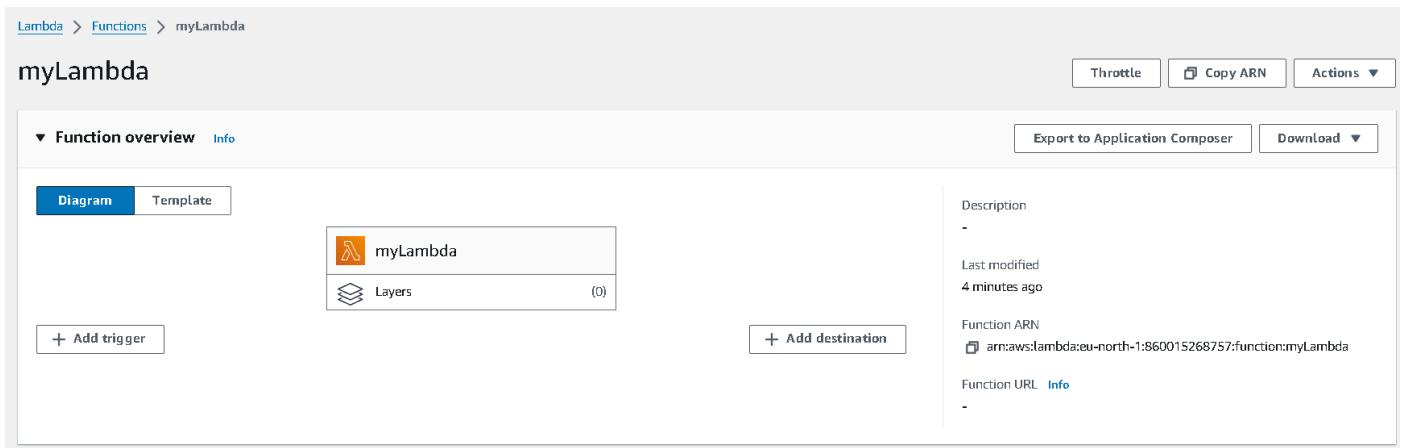
```

1:1 JSON Spaces: 2

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



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



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

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

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.

[X](#) [C](#)
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 [X](#)

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.

[C](#)

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.

[C](#)

Recursive invocation

Function overview [Info](#)

[Export to Application Composer](#) [Download](#) ▾

Diagram [Template](#)

myLambda
Layers (0)

S3

[+ Add destination](#)

[+ Add trigger](#)

Description
[Basic Settings](#)

Last modified
1 hour ago

Function ARN
[arn:aws:lambda:us-east-1:010928205712:function:Bhushan_Lambda](#)

Function URL [Info](#)

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

General configuration

- Triggers** [\(1\) Info](#)
- [Permissions](#)
- [Destinations](#)
- [Runas user](#)
- [Environment variables](#)
- [Tags](#)
- [VPC](#)
- [RDS databases](#)
- [Monitoring and operations tools](#)
- [Concurrency and recursion detection](#)
- [Asynchronous invocation](#)
- [Code signing](#)
- [File systems](#)
- [State machines](#)

Triggers (1) [Info](#)

		Edit	Delete	Add trigger		
<input type="checkbox"/>	Trigger	C	Fix errors	Edit	Delete	Add trigger
<input type="checkbox"/>	S3:wearekcs	arn:aws:s3:::wearekcs	Details			

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

The screenshot shows the AWS Lambda function editor. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The main area is titled "Code source" with a "Info" link. Below the title are standard menu options: File, Edit, Find, View, Go, Tools, Window. A toolbar with "Test" (highlighted in blue), Deploy, and a status message "Changes not deployed" is visible. On the left, there's a sidebar for "Environment" settings. The code editor contains the following Python code:

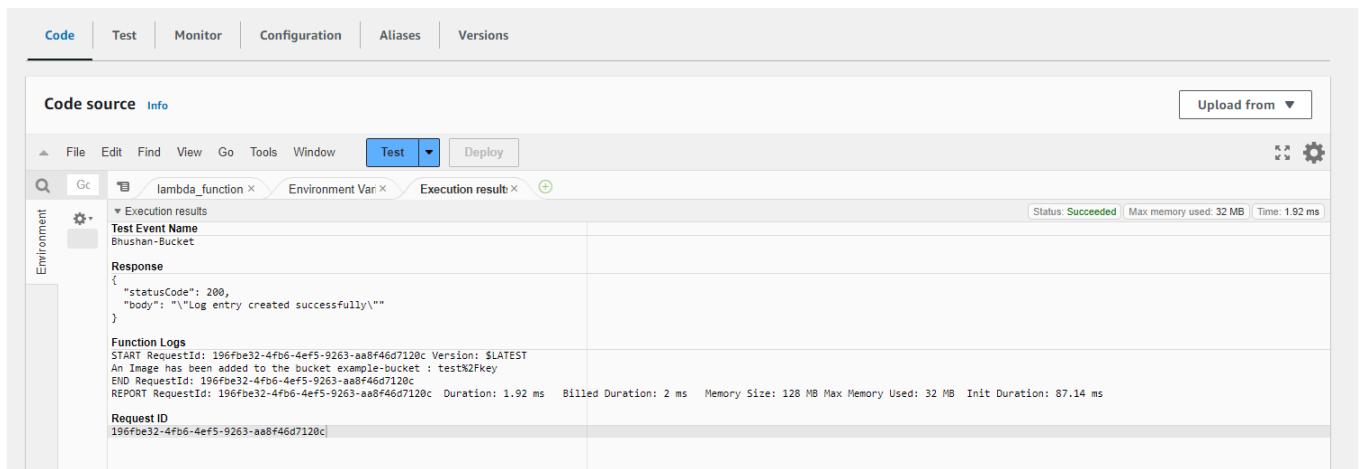
```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     bucket_name= event['Records'][0]['s3']['bucket']['name']
6     object_key= event['Records'][0]['s3']['object']['key']
7
8     print(f"An Image has been added to the bucket {bucket_name} : {object_key}")
9     return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

This screenshot shows the same AWS Lambda function editor after deployment. The "Changes not deployed" message has disappeared, indicating successful deployment. The code remains identical to the previous screenshot.

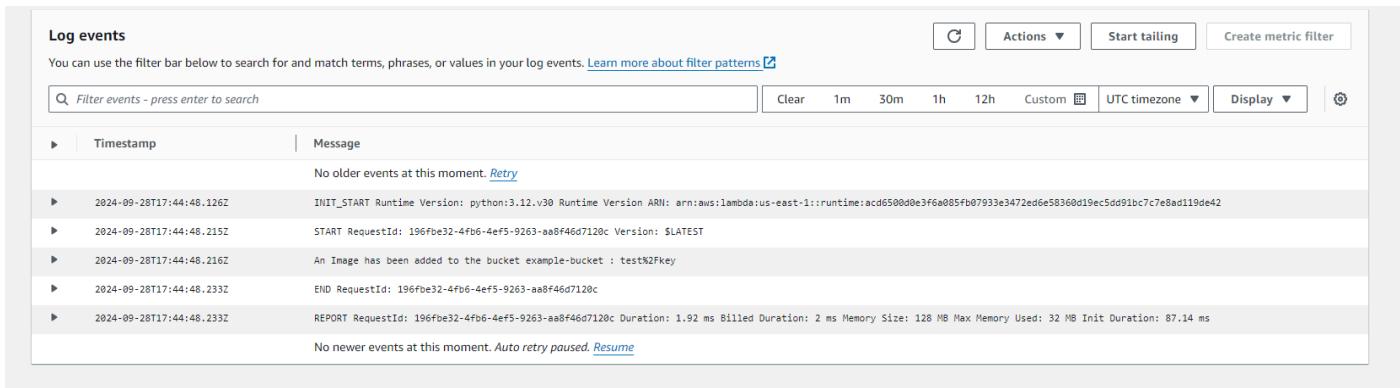
Step 9: Now upload any image to the bucket.

The screenshot shows the AWS S3 console. The path is "Amazon S3 > Buckets > wearables > Upload". The "Upload" tab is selected. The "Upload info" section contains instructions: "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." A link to "Learn more" is provided. Below this is a large dashed box for dragging files or choosing them. The "Files and folders" section shows one item: "F:\OUusXg\XXB2s.jpg" (1 Total, 957.0 KB). There are "Remove", "Add files", and "Add folder" buttons. The "Destination" section shows the destination as "s3://wearables". It includes "Destination details" (Bucket settings that impact new objects stored in the specified destination) and "Permissions" (Grant public access and access to other AWS accounts). The "Properties" section is partially visible. At the bottom right are "Cancel" and "Upload" buttons, with "Upload" being orange.

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



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



Conclusion:

In this experiment, we successfully created an AWS Lambda function that logs a message when an image is uploaded to an S3 bucket. It is important to note that we have to select S3-put template in the event otherwise code will give an error. The function was successfully triggered by S3 object uploads, validating the functionality of Lambda's event-driven architecture. This experiment demonstrated how Lambda can efficiently respond to S3 events and how to troubleshoot common issues with event structure.