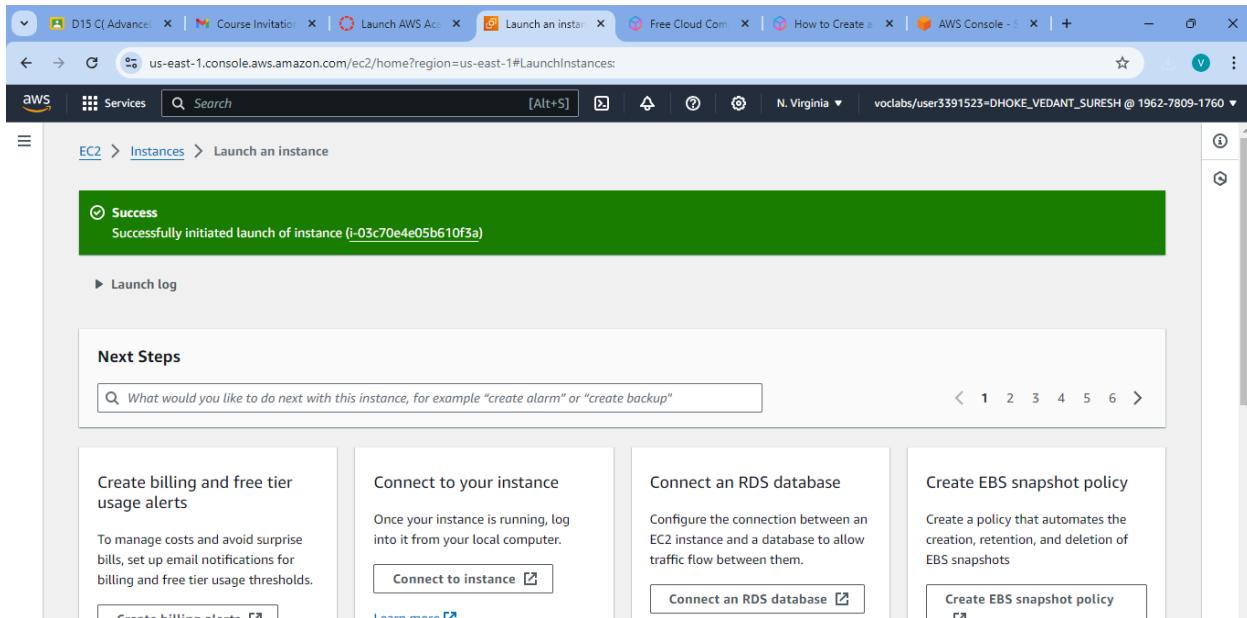


## AdvanceDevOps Exp 1A

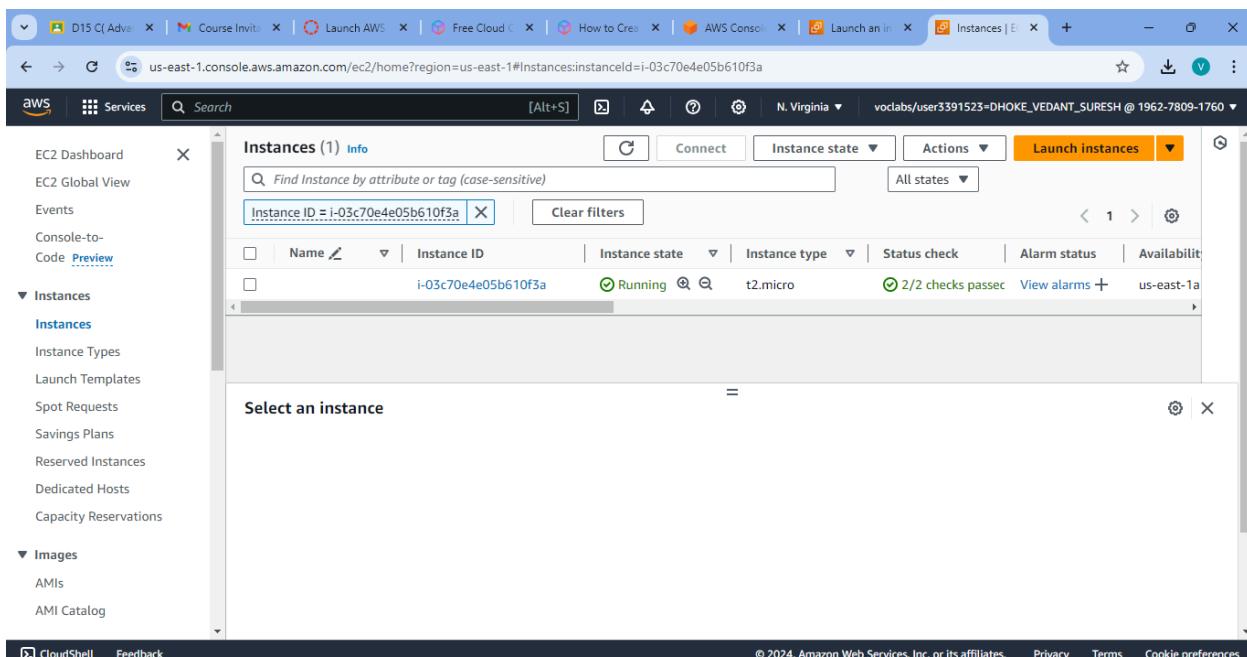
**AIM:** To understand the Creation of an EC2 Instance . To develop a website and host it on your local machine on a VM.Hosting a static website on Amazon S3.

Firstly we have to create an EC2 Instance.



The screenshot shows the AWS EC2 Instances launch interface. At the top, there's a green success message box stating "Successfully initiated launch of instance (i-03c70e4e05b610f3a)". Below this, under "Next Steps", there are several options: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". Each option has a corresponding button and a "Learn more" link. The "Connect to your instance" section includes a note about managing costs and avoiding surprise bills by setting up email notifications for billing and free tier usage thresholds.

### Launching our EC2 Instance



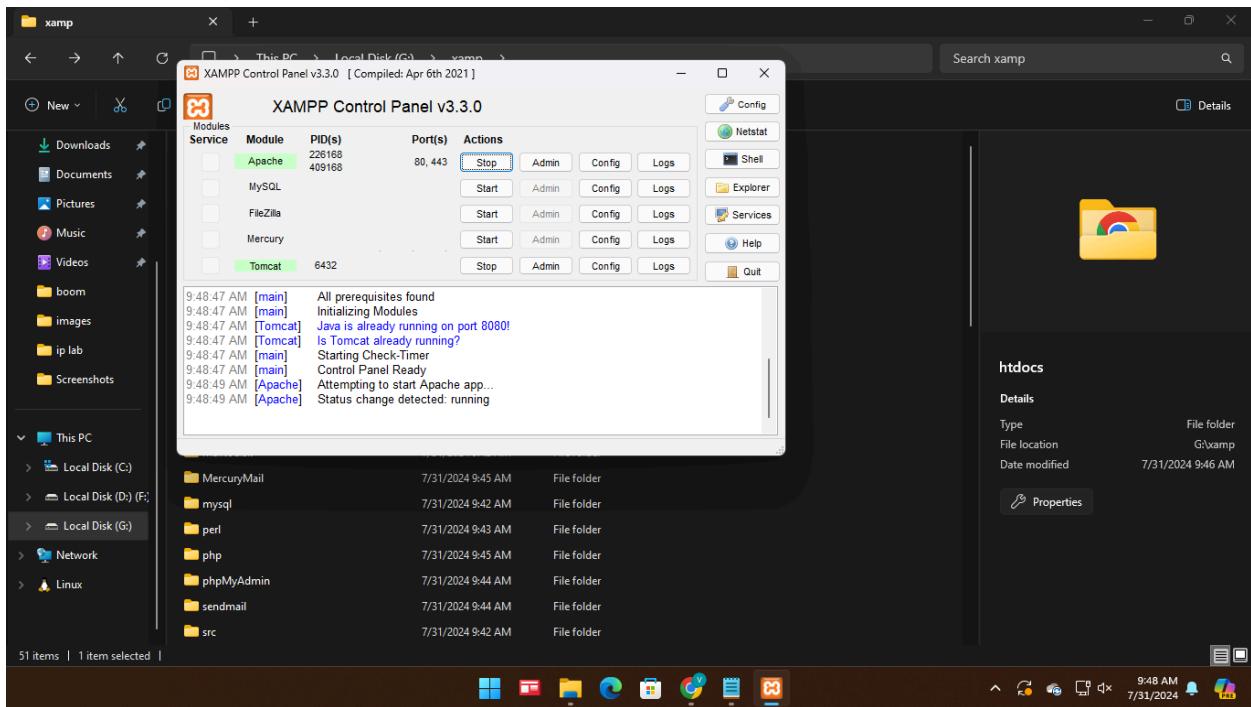
The screenshot shows the AWS EC2 Instances page with one instance listed. The instance details are as follows:

Instance ID	Name	Instance state	Instance type	Status check	Alarm status	Availability zone
i-03c70e4e05b610f3a		Running	t2.micro	2/2 checks passed	View alarms	us-east-1a

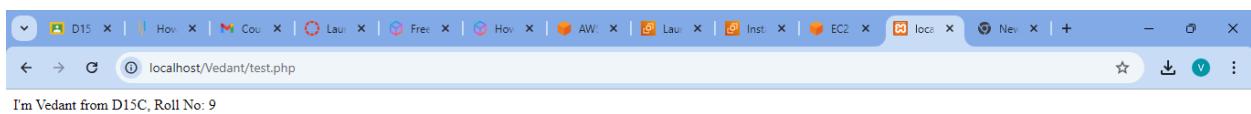
A modal window titled "Select an instance" is open at the bottom, showing the same instance information. The left sidebar shows navigation links for EC2 Dashboard, Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog.

nce

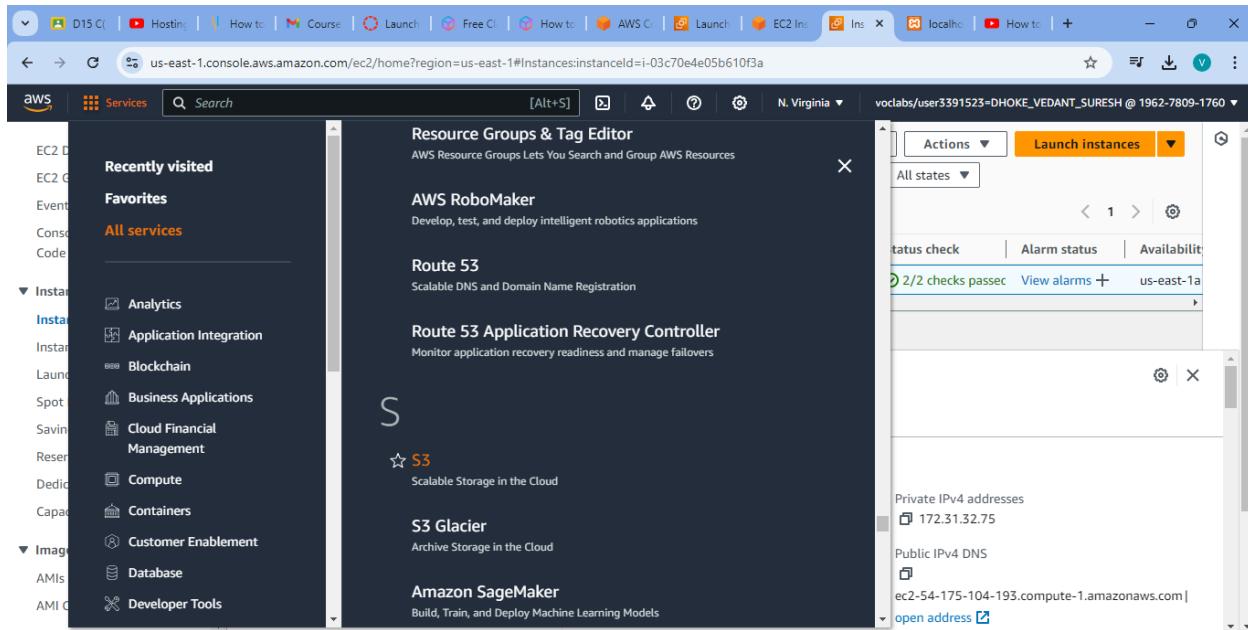
Download the XAMPP Server and Start it which will help us to host our website on local machine.



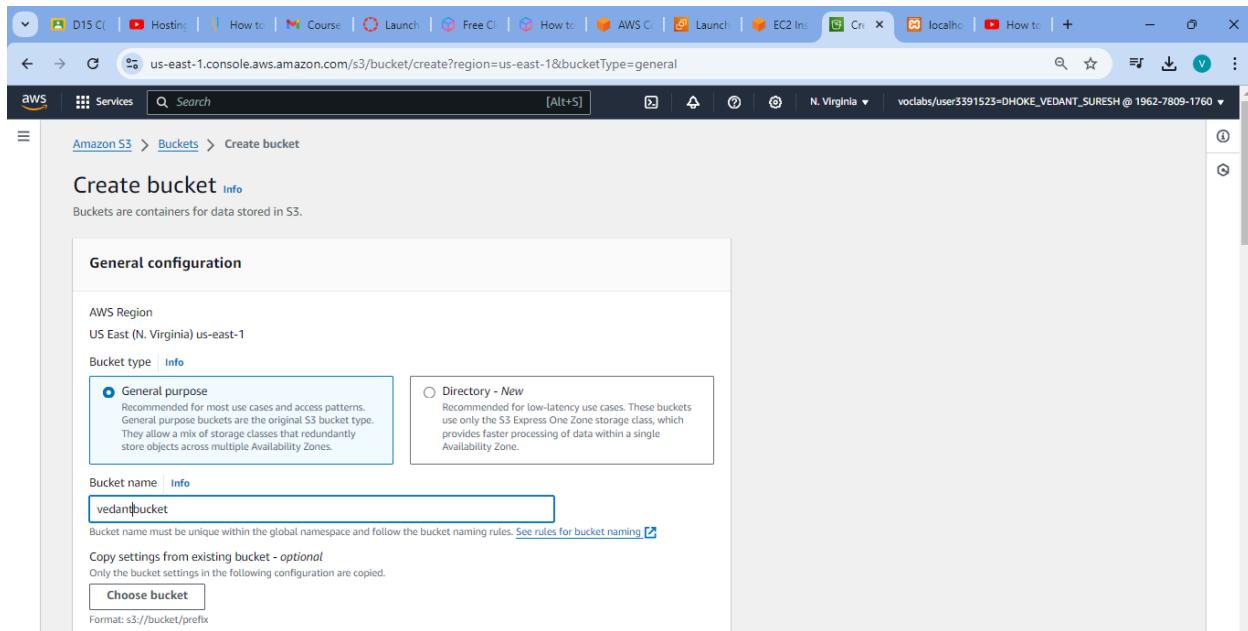
We can see that our website is running on the local machine.

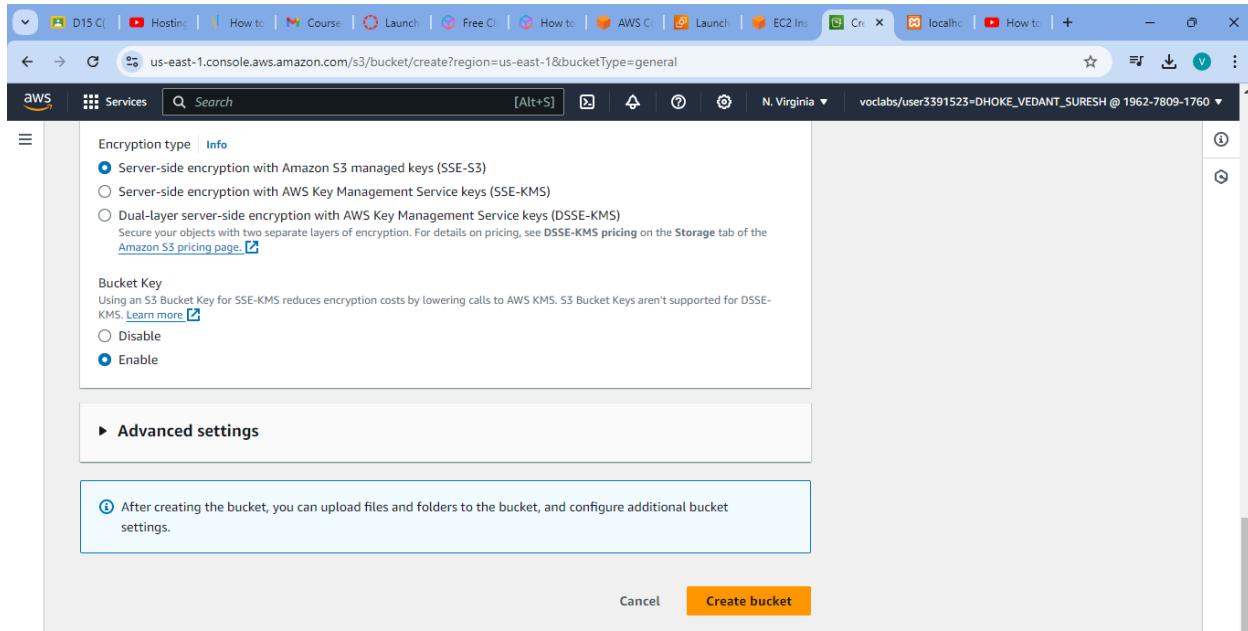


Now we want to create a S3 Bucket. Go to the services and select S3

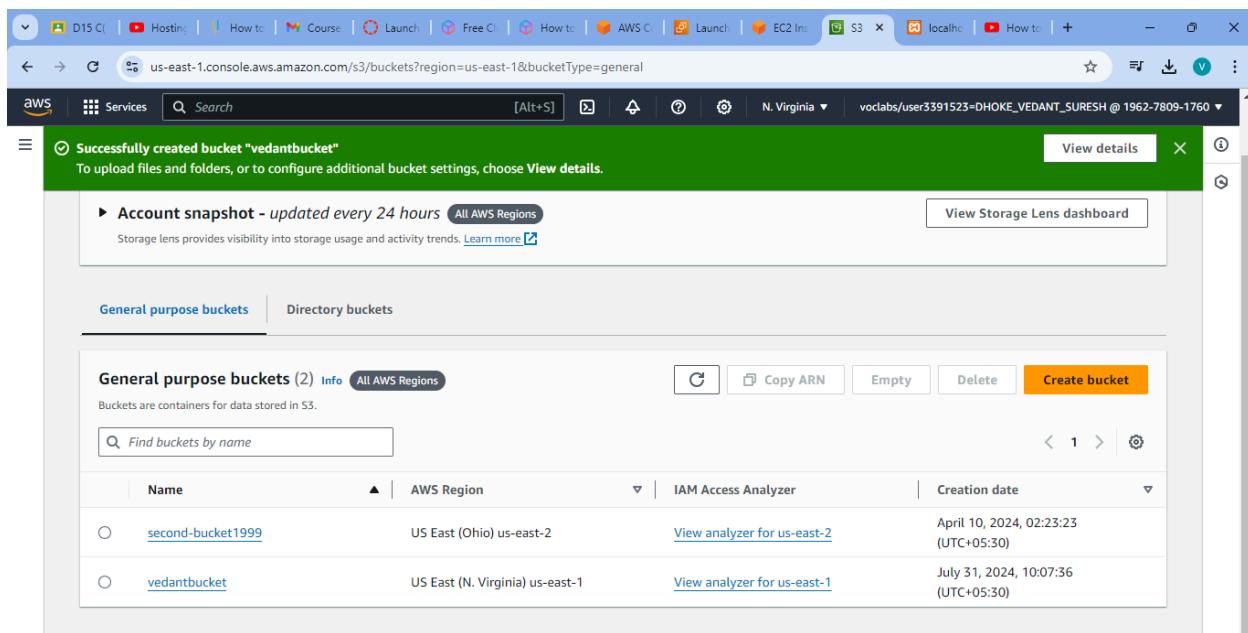


Give name to your bucket and configure the settings properly and click on create bucket.





Now we can see that our bucket named “vedantbucket” is created Successfully.



This is how our bucket's structure looks like , We have to upload our files into the bucket so click on Upload button

The screenshot shows the AWS S3 console interface. The URL in the browser is `us-east-1.console.aws.amazon.com/s3/buckets/vedantbucket?region=us-east-1&bucketType=general&tab=objects`. The page title is "vedantbucket". Below the title, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The "Objects" tab is selected. The main area displays a table with the heading "Objects (0) Info". Below the table, there is a search bar labeled "Find objects by prefix" and a pagination control showing "1". At the bottom of the table area is a large orange "Upload" button.

We now uploaded our test.php file in our bucket.

The screenshot shows the AWS S3 console interface, specifically the "Upload" step for the "vedantbucket". The URL in the browser is `us-east-1.console.aws.amazon.com/s3/upload/vedantbucket?region=us-east-1&bucketType=general`. The page title is "Upload". Above the upload area, there is a message: "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 dashed blue box with the text "Drag and drop files and folders you want to upload here, or choose Add files or Add folder.". Underneath this is a table titled "Files and folders (1 Total, 52.0 B)". The table shows one item: "test.php" located in the folder "Vedant/". There are "Remove", "Add files", and "Add folder" buttons above the table. Below the table is a "Destination" section with a "Info" link. The bottom of the screen shows the standard AWS navigation bar with links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with system status icons.

Finally we have uploaded test.php and index.html to our bucket.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs for 'Services' and 'Search'. Below it, the path 'Amazon S3 > Buckets > vedantbucket > Vedant/' is displayed. A 'Copy S3 URI' button is located in the top right corner. The main area is titled 'Objects (2) Info' and contains a table with two rows:

Name	Type	Last modified	Size	Storage class
index.html	html	August 8, 2024, 22:13:18 (UTC+05:30)	436.0 B	Standard
test.php	php	July 31, 2024, 10:14:51 (UTC+05:30)	52.0 B	Standard

By Clicking on the files we can see their properties and configurations.

The screenshot shows the AWS S3 object details page for 'test.php'. The URL in the browser is 'us-east-1.console.aws.amazon.com/s3/object/vedantbucket?region=us-east-1&bucketType=general&prefix=Vedant%2Ftest.php&tab=details'. The left sidebar has a 'Buckets' section with various options like Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area shows the file 'test.php' with the 'Properties' tab selected. The 'Object overview' section displays the following details:

Owner	s3://vedantbucket/Vedant/test.php
AWS Region	Amazon Resource Name (ARN)
US East (N. Virginia) us-east-1	arn:aws:s3:::vedantbucket/Vedant/test.php
Last modified	Entity tag (Etag)
July 31, 2024, 10:14:51 (UTC+05:30)	a77a1b00aff443f118b13215128e33a0
Size	Object URL
52.0 B	<a href="https://vedantbucket.s3.amazonaws.com/Vedant/test.php">https://vedantbucket.s3.amazonaws.com/Vedant/test.php</a>
Type	
php	
Key	

Now to avoid access denied issue we have to edit some of the properties.

### 1. Editing the hosting type as Host a static website.

The screenshot shows the AWS S3 console with the URL <https://us-east-1.console.aws.amazon.com/s3/bucket/vedantbucket/property/website/edit?region=us-east-1&bucketType=general>. The left sidebar shows 'Buckets' and other S3 management options. The main panel is titled 'Edit static website hosting' and contains a note about enabling public access via S3 Block Public Access settings.

### 2. Editing the Block public access.

The screenshot shows the AWS S3 console with the URL <https://us-east-1.console.aws.amazon.com/s3/bucket/vedantbucket/property/bpa/edit?region=us-east-1&bucketType=general>. The left sidebar shows 'Buckets' and other S3 management options. The main panel is titled 'Edit Block public access (bucket settings)' and lists several options under 'Block public access (bucket settings)'.

### 3. Editing the Public access status.

The screenshot shows a browser window with multiple tabs open, including 'Course Invitation - 2022.v...', 'Classroom Management...', 'Home', 'Launch AWS Academy Le...', 'Make objects public - S3 b...', 'vedantbucket.s3.amazonaws...', and 'voclabs/user3391523=DHOKE\_VEDANT\_SURESH @ 1962-7809-1760'. The main content area displays a green success message: 'Successfully edited public access' with a link to 'View details below.' Below this, a section titled 'Make public: status' shows a summary table:

Source	Successfully edited public access	Failed to edit public access
s3://vedantbucket	1 object, 52.0 B	0 objects

Below the summary, there are two tabs: 'Failed to edit public access' (selected) and 'Configuration'. Under 'Failed to edit public access', it says '(0)' and shows a table header for 'Name', 'Folder', 'Type', 'Last modified', 'Size', and 'Error'. A note at the bottom states 'No objects failed to edit'.

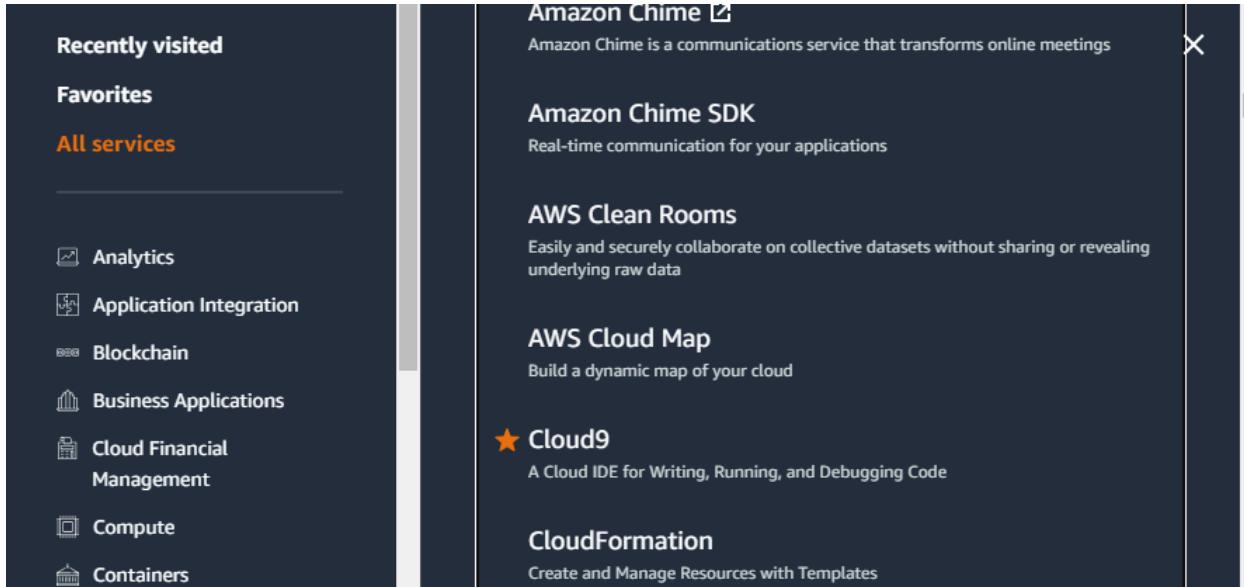
Finally our static website is hosted Amazon S3.

The screenshot shows a browser window with several tabs open, including 'Classroom Management...', 'Home', 'Course Invitation - 2022.v...', 'Launch AWS Academy Le...', 'Vedant/index.html - Object...', 'Document', and others. The active tab shows the URL 'vedantbucket.s3.amazonaws.com/Vedant/index.html'. The page content is the static website hosted on AWS, featuring the heading 'Welcome to Amazon Web Services (AWS)' and the subtext 'Explore the leading cloud platform with a comprehensive suite of services to build, deploy, and scale applications effortlessly.'

## AdvDevops Exp 1B

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

1. Login to your AWS account and navigate to the Cloud9.



## 2. We want to create an environment in Cloud9.

AWS Cloud9 > Environments > Create environment

## Create environment Info

**Details**

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

Description - *optional*  Limit 200 characters.

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

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

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

3. Configure the environment settings and click on **create** button.

Cloud9 after creation.

### New EC2 instance

**Instance type Info**  
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

- t2.micro (1 GiB RAM + 1 vCPU)  
Free-tier eligible. Ideal for educational users and exploration.
- t3.small (2 GiB RAM + 2 vCPU)  
Recommended for small web projects.
- m5.large (8 GiB RAM + 2 vCPU)  
Recommended for production and most general-purpose development.

Additional instance types  
Explore additional instances to fit your need.

**Platform Info**  
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

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

30 minutes

### Network settings Info

**Connection**  
How your environment is accessed.

- AWS Systems Manager (SSM)  
Accesses environment via SSM without opening inbound ports (no ingress).
- Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

► VPC settings Info

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

**The following IAM resources will be created in your account**

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

Cancel **Create**

#### 4. Our Environment is created Successfully.

The screenshot shows the AWS Cloud9 Environments page. At the top, there are two notifications: one about a successfully created environment and another about AWS Toolkits. Below the notifications is a breadcrumb navigation: AWS Cloud9 > Environments. The main area is titled "Environments (1)" and contains a table with one row. The table columns are: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The single environment listed is "Vedant Dhone" (with a link to "Open"), which is an "EC2 instance" connected via "Secure Shell (SSH)". The owner is "Owner" with the ARN "arn:aws:sts::196278091760:assumed-role/voclabs/user3391523=DHOKE\_VEDANT\_SURESH".

#### 5. It will take few minutes to create aws instance for your Cloud 9 Environment.

The screenshot shows the AWS Cloud9 IDE interface. The title bar says "Welcome". On the left, there's a sidebar with "Go to Anything (Ctrl-P)" and a file tree showing ".c9" and "README.md". The main area has a dark header with "AWS Cloud9" and "Welcome to your development environment". Below the header, there's a "Toolkit for AWS Cloud9" section with a sub-section about the AWS Toolkit for Cloud9. At the bottom, there's a terminal window showing a bash prompt: "bash - \*ip-172-31-61-229.x | Immediate | voclabs:~/environment \$".

#### 6. Now we want to Add users

The screenshot shows the AWS IAM Users page. The left sidebar includes "Identity and Access Management (IAM)", "Dashboard", "Access management" (with "Users" selected), "Policies", "Identity providers", "Account settings", "Access reports" (with "Analyzer settings" selected), and "Services" (with "Search" and "[Alt+S]"). The main area shows the "Users" section with a table. The table has columns: User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. A search bar at the top of the table is empty. The message "No resources to display" is centered below the table.

## 7. Add user provide manual password if you want and click on Next.

**User details**

User name  
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

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

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

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

I want to create an IAM user  
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, 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.

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

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

## 8. Set the Permissions ,click on create group

**Set permissions**

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

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

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

**Create group**

▶ **Set permissions boundary - optional**

**Cancel** **Previous** **Next**

9. Provide group name and click on create user group.

**Create user group** X

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

User group name  
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '+,-,@,\_' characters.

**Permissions policies (947)**  [Create policy](#)

Filter by Type Search All ty... 1 2 3 4 5 6 7 ... 48

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess</a>	AWS managed ...	None	Provides full access to AWS services
<input type="checkbox"/>	<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AlexaForBusinessD...</a>	AWS managed	None	Provide device setup access to Alexa
<input type="checkbox"/>	<a href="#">AlexaForBusinessF...</a>	AWS managed	None	Grants full access to AlexaForBusin

[Cancel](#) Create user group

## 10. Navigate back to user groups

### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

#### Permissions options

##### Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

##### Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

##### Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

#### User groups (3)

[Create group](#) Search[<](#) [1](#) [>](#)

<input type="checkbox"/>	Group name	▲	Users	▼	Attached policies	▼	Created
<input type="checkbox"/>	<a href="#">AdvanceDevOps_21_3_9</a>		0	-			2024-08-07 (1 minute ago)
<input type="checkbox"/>	<a href="#">AdvanceDevOps_3_21_9</a>		0	-			2024-08-07 (1 minute ago)
<input type="checkbox"/>	<a href="#">AdvDevOpsLab_9</a>		0	-			2024-08-07 (1 minute ago)

### Review and create

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

#### User details

User name

Vedant

Console password type

Custom password

Require password reset

No

#### Permissions summary

[<](#) [1](#) [>](#)

Name	▲	Type	▼	Used as	▼
<a href="#">AdvanceDevOps_21_3_9</a>		Group		Permissions group	
<a href="#">AdvanceDevOps_3_21_9</a>		Group		Permissions group	
<a href="#">AdvDevOpsLab_9</a>		Group		Permissions group	

#### Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

## 11. Click on create user

The screenshot shows the 'Tags - optional' section of the 'Create user' wizard. It includes a note about key-value pairs for AWS resources, a message stating 'No tags associated with the resource.', a 'Add new tag' button, and a note that up to 50 more tags can be added. At the bottom are 'Cancel', 'Previous', and 'Create user' buttons.

## 12. User is created Successfully

The screenshot shows the 'User created successfully' confirmation page. It features a green header bar with the message, a 'View user' link, and a close button. Below is a navigation breadcrumb ('IAM > Users > Create user'). A left sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Retrieve password' and contains 'Console sign-in details'. It shows the 'Console sign-in URL' (https://022499016110.signin.aws.amazon.com/console), 'User name' (Vedant), and 'Console password' (redacted). There is also a 'Email sign-in instructions' button. At the bottom are 'Cancel', 'Download .csv file', and 'Return to users list' buttons.

13.. Click on your group name which you have created and navigate to permission tab as shown:

The screenshot shows the AWS IAM User Groups page. The top navigation bar includes 'IAM' and 'User groups'. Below it, the group 'AdvDevOpsLab\_9' is selected. The main content area has a 'Summary' card with details: User group name (AdvDevOpsLab\_9), Creation time (August 07, 2024, 09:33 (UTC+05:30)), and ARN (arn:aws:iam::022499016110:group/AdvDevOpsLab\_9). Below the summary, there are tabs for 'Users (3)', 'Permissions' (which is selected and highlighted in blue), and 'Access Advisor'. The 'Permissions' tab displays a table for 'Permissions policies'. The table has columns for 'Policy name' (with a search bar and dropdown filter), 'Type', and 'Attached entities'. A message at the top of this section says 'You can attach up to 10 managed policies.' Below the table, a progress indicator says 'Loading policies'.

14. Search AWSCloud9 policy and click on Attach policies.

The screenshot shows the 'Other permission policies' search results for the 'AWSCloud9' policy. The search bar contains 'AWSCloud9'. The results table has columns: 'Policy name' (with a search bar and dropdown filter), 'Type', 'Used as', and 'Description'. One policy is checked: 'AWSCloud9EnvironmentMember' (AWS managed, None, Provides the ability to be invited into AW...). Other policies listed include 'AWSCloud9Administrator', 'AWSCloud9SSMInstanceProfile', and 'AWSCloud9User'. At the bottom right of the table are 'Cancel' and 'Attach policies' buttons.

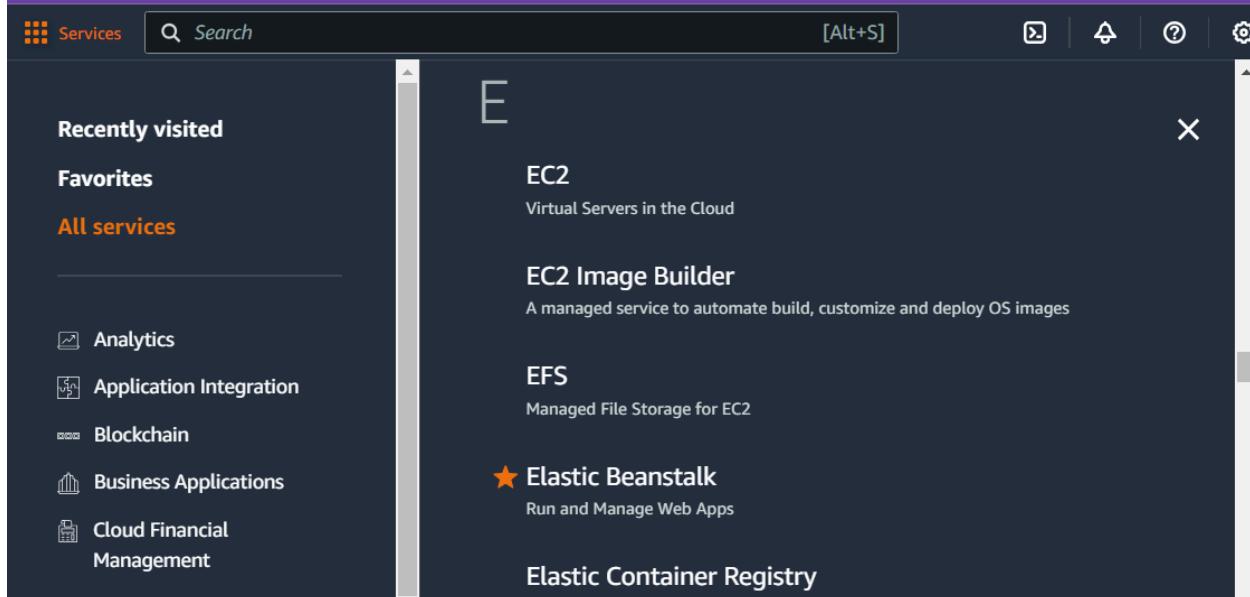
15. We can see that Policy is attached Successfully.

The screenshot shows the AWS IAM User Groups console. At the top, a green header bar indicates that there are policies attached to this user group. Below the header, the user group name is 'AdvDevOpsLab\_9'. On the right side of the header, there are 'Delete' and 'Edit' buttons. Under the 'Summary' section, there are three fields: 'User group name' (AdvDevOpsLab\_9), 'Creation time' (August 07, 2024, 09:33 (UTC+05:30)), and 'ARN' (arn:aws:iam::022499016110:group/AdvDevOpsLab\_9). Below the summary, there are three tabs: 'Users (3)', 'Permissions' (which is selected), and 'Access Advisor'. In the 'Permissions' section, it says 'Permissions policies (1) Info'. It shows one policy named 'AWSCloud9EnvironmentMember' attached. The policy details show it is an 'AWS managed' policy. There are buttons for 'Search', 'Simulate', 'Remove', and 'Add permissions'.

## AdvDevOps Exp 2

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

1. Login to your AWS account and search for Elastic Beanstalk in the search box.



2. Open the Elastic Beanstalk and name your web app as shown below.

The screenshot shows the 'Configure environment' step of the AWS Elastic Beanstalk setup wizard. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main area is titled 'Configure environment' with a 'Info' link. It contains two sections: 'Environment tier' and 'Application information'. Under 'Environment tier', the 'Web server environment' option is selected. Under 'Application information', the 'Application name' field is filled with 'Vedant'. A note states 'Maximum length of 100 characters.'

3. Automatically your environment name is set.

**Environment information** Info

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

Environment name

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

Domain

.us-east-1.elasticbeanstalk.com Check availability

Environment description

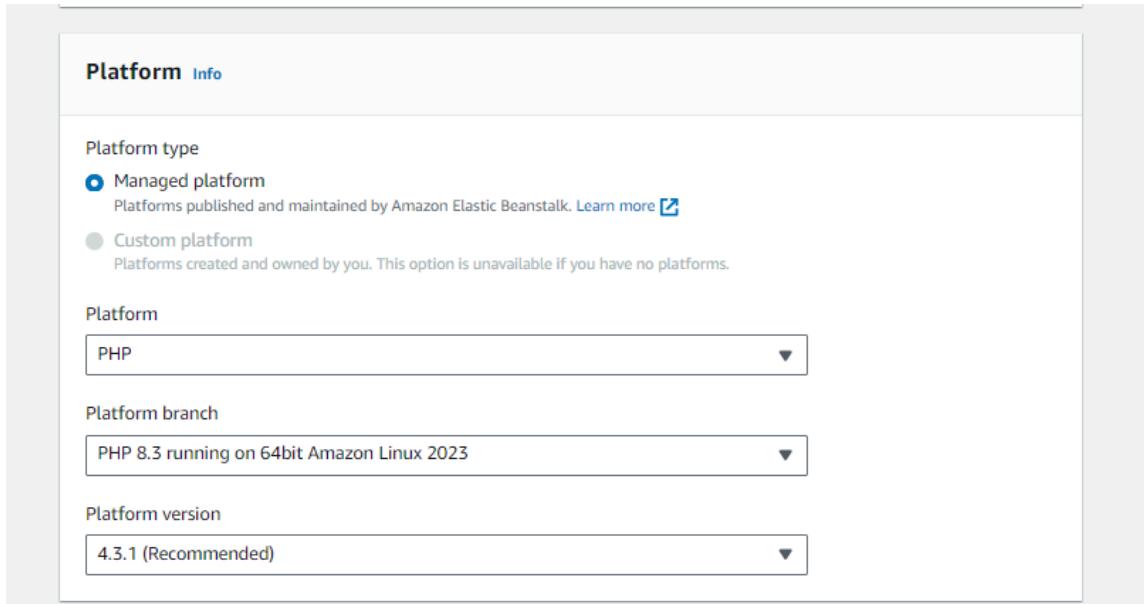
**Platform** Info

Platform type

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

Custom platform

4. Choose PHP from the drop-down menu and then click Create Application.



## 5. We can see that our Environment is created and launched successfully

Environment successfully launched.

VedantApp1-env

Environment overview

Health	Environment ID
Ok	e-rrgh85igup
Domain	Application name
VedantApp1-env.eba-zfgx7tjp.us-east-1.elasticbeanstalk.com	VedantApp1

Platform

Change version

Platform

PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

Running version

-

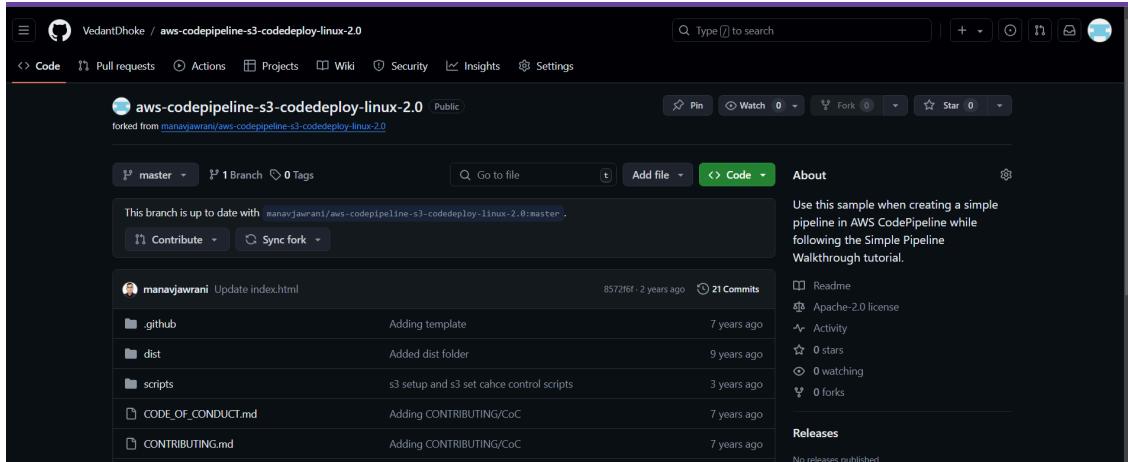
Platform state

Supported

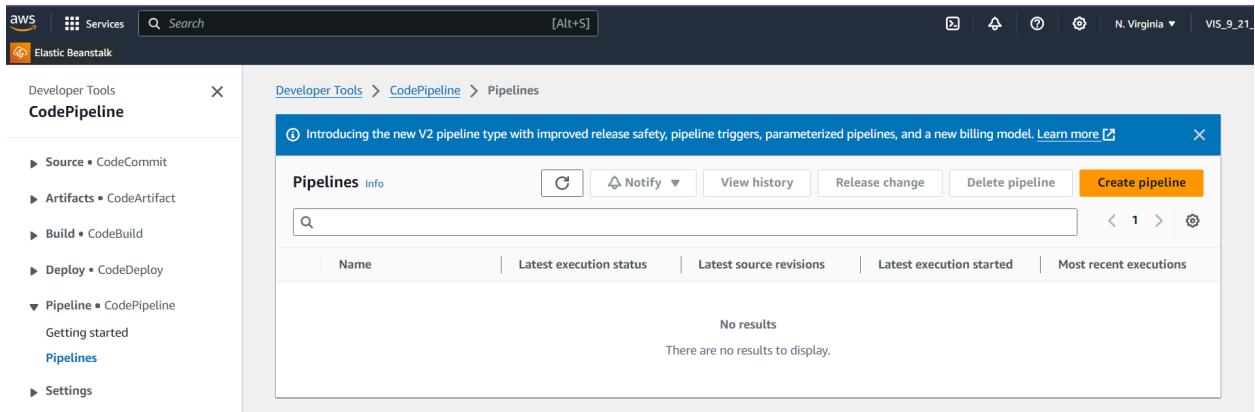
## 6. In this step, we'll obtain the sample code from a GitHub repository to host it later. The pipeline pulls the code from the source and executes specific actions.

To get started, navigate to the provided GitHub repository and fork it to your account. For this experiment, we will use a forked GitHub repository as our source.

We can see that in VedantDhoke repository the provided GitHub repository is forked.



7. Now we want to create a Pipeline. Navigate to Developer Tools -> CodePipeline and Click on create pipeline.



8. Give name to your pipeline and do the settings in step 1.

The screenshot shows the 'Pipeline settings' section of the AWS CodePipeline console. The pipeline name is set to 'VedantPipeline'. The pipeline type is selected as 'Queued (Pipeline type V2 required)'. The execution mode is 'Superseded'. The service role is configured with a new role named 'AWSCodePipelineServiceRole-us-east-1-VedantPipeline'. A checkbox 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' is checked.

Page	Pipeline settings
Page	<p><b>Pipeline name</b> Enter the pipeline name. You cannot edit the pipeline name after it is created. <input type="text" value="VedantPipeline"/></p> <p>No more than 100 characters</p>
Page	<p><b>Pipeline type</b></p> <p><input checked="" type="radio"/> You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.</p>
Page	<p><b>Execution mode</b> Choose the execution mode for your pipeline. This determines how the pipeline is run.</p> <p><input type="radio"/> Superseded A more recent execution can overtake an older one. This is the default.</p> <p><input checked="" type="radio"/> Queued (Pipeline type V2 required) Executions are processed one by one in the order that they are queued.</p> <p><input type="radio"/> Parallel (Pipeline type V2 required) Executions don't wait for other runs to complete before starting or finishing.</p>
Page	<p><b>Service role</b></p> <p><input checked="" type="radio"/> New service role Create a service role in your account</p> <p><input type="radio"/> Existing service role Choose an existing service role from your account</p>
Page	<p><b>Role name</b></p> <p><input type="text" value="AWSCodePipelineServiceRole-us-east-1-VedantPipeline"/></p> <p>Type your service role name</p> <p><input checked="" type="checkbox"/> Allow AWS CodePipeline to create a service role so it can be used with this new pipeline</p>

9. In the Source Stage, Choose GitHub(Version 2) and click on Connect to GitHub. You'll need your GitHub credentials to authorize and integrate AWS with your forked GitHub repository.

**Add source stage** info

Step 2 of 5

**Source**

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

GitHub (Version 2)

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

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

arn:aws:codeconnections:us-east-1:022499016110:connection/c8e5449b-9ft  or

**Ready to connect**  
Your GitHub connection is ready for use.

**Elastic Beanstalk**

**Developer Tools** > **Connections** > Create connection

**Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN.**   
Resources with both service prefixes will continue to display in the console. [Learn more](#)

**Connect to GitHub**

**GitHub connection settings** info

**Connection name**  
VedantConnect

**GitHub Apps**  
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.

53638499  or

**Tags - optional**

10. Next, select the forked repository and the appropriate branch from the search box. Click Continue, then skip the build stage and proceed directly to the Deployment stage.

**Repository name**  
Choose a repository in your GitHub account.

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

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

**Output artifact format**  
Choose the output artifact format.

- CodePipeline default**  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
- Full clone**  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

**Trigger**

**Trigger type**  
Choose the trigger type that starts your pipeline.

- No filter**  
Starts your pipeline on any push and clones the HEAD.
- Specify filter**  
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
- Do not detect changes**  
Don't automatically trigger the pipeline.

11. Select Deploy and specify the application name and environment name, then click "Next." Review the details and create the pipeline.

**Deploy**

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

**Region**

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

No more than 100 characters

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

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

Configure automatic rollback on stage failure

**Cancel** **Previous** **Next**

12. Review all the settings and click on create pipeline.

e > [Pipelines](#) > Create new pipeline

## Review Info

Step 5 of 5

### Step 1: Choose pipeline settings

#### Pipeline settings

Pipeline name

VedantPipeline

Pipeline type

V2

Execution mode

QUEUED

Artifact location

A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name

AWSCodePipelineServiceRole-us-east-1-VedantPipeline

### Step 4: Add deploy stage

#### Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

VedantApp1

EnvironmentName

VedantApp1-env

Configure automatic rollback on stage failure

Disabled

Cancel

Previous

Create pipeline

13. We can see our pipeline is successfully created. We need to ensure that both stages i.e Source and Deploy shows Succeeded as shown below.

The screenshot shows the AWS CodePipeline console with the pipeline named "VedantPipeline". The pipeline type is V2 and the execution mode is QUEUED. The "Source" stage is listed as Succeeded, with a Pipeline execution ID of 3370143f-2a4b-47fb-9d83-ba0681380088. The "Deploy" stage is also listed as Succeeded. A "Start rollback" button is visible next to the Deploy stage. The bottom of the screen includes standard AWS navigation links: © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

14. Later Click on the URL which is provided in the Domain. It will navigate us to our website.

The screenshot shows the AWS Elastic Beanstalk console with the environments page. It lists one environment: "VedantApp1-env", which is marked as "Ok" (healthy). The environment is running on "PHP 8.3" and is associated with the domain "VedantApp1-env.eba-zfgx7tjp...". The "Create environment" button is visible at the top right of the table. The left sidebar shows recent environments: "VedantApp1-env" and "Vedant-env".

15. This is our sample website.

The screenshot shows a web browser window with the URL "vedantapp1-env.eba-zfgx7tjp.us-east-1.elasticbeanstalk.com". The page displays a large "Congratulations!" message in white text on a green background. Below it, a summary states: "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." At the bottom, there is a link: "For next steps, read the AWS CodePipeline Documentation. Incedge 2020".

15. Committing the changes to update app .

1. Update the HTML file in your forked repository.
2. In GitHub, open index.html and modify the heading or paragraph tag. Commit the changes directly on GitHub.
3. Observe the real-time updates in the Source panel after committing.
4. Once the deployment is successful, view the changes on the website using the same URL.

The screenshot shows the AWS CodePipeline console with the pipeline named 'VedantPipeline'. The pipeline type is V2 and the execution mode is QUEUED. The pipeline has two stages: 'Source' and 'Deploy'. The 'Source' stage is succeeded, with a pipeline execution ID of c5797570-1430-4760-b04e-c1c5323e6e4e. It shows a GitHub (Version 2) action that succeeded just now, with a commit ID ce008b0d. A 'View details' button is available. The 'Deploy' stage is also succeeded, with the same pipeline execution ID. A 'Start rollback' button is present. On the right side, there are two green checkmarks indicating successful steps. A 'Disable transition' button is located below the Source stage.

16. Again click on the URL and we can see our Updated App.

The screenshot shows a web browser window with the URL vedantapp1-env.eba-zfgx7jp.us-east-1.elasticbeanstalk.com. The page displays a large green header with the text 'Congratulations! Vedant Dhone'. Below it, a message states: '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.' At the bottom, a small note says 'For next steps, read the AWS CodePipeline Documentation. Incedge 2020'.

**EXPERIMENT NO: 3**

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

**STEPS :****1.Create Security Groups with the following inbound rules**

The screenshot shows the AWS EC2 'Create security group' interface. It consists of two main sections: 'Basic details' and 'Inbound rules'.

**Basic details:**

- Security group name:** Master
- Description:** Allows SSH access to developers
- VPC:** vpc-06a8924e856c7a04d

**Inbound rules:**

**Top Tab (Visible):**

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere-1...	0.0.0.0/0
All traffic	All	All	Anywhere-1...	0.0.0.0/0
Custom TCP	TCP	6443	Anywhere-1...	0.0.0.0/0

**Bottom Tab (Visible):**

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere-1...	0.0.0.0/0
All traffic	All	All	Anywhere-1...	0.0.0.0/0
Custom TCP	TCP	6443	Anywhere-1...	0.0.0.0/0
Custom TCP	TCP	10251	Anywhere-1...	0.0.0.0/0
Custom TCP	TCP	10250	Anywhere-1...	0.0.0.0/0
All TCP	TCP	0 - 65535	Anywhere-1...	0.0.0.0/0
Custom TCP	TCP	10252	Anywhere-1...	0.0.0.0/0
SSH	TCP	22	Anywhere-1...	0.0.0.0/0

**Buttons:**

- Add rule (located at the bottom left of the bottom tab)
- Delete (multiple buttons located next to each rule entry)

EC2 > Security Groups > Create security group

### Create security group Info

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

**Basic details**

Security group name Info  
 Name cannot be edited after creation.

Description Info

VPC info

**Inbound rules Info**

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
HTTP	TCP	80	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
All traffic	All	All	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
Custom TCP	TCP	30000 - 32767	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete

**Inbound rules Info**

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
HTTP	TCP	80	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
All traffic	All	All	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
Custom TCP	TCP	30000 - 32767	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
Custom TCP	TCP	10250	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
SSH	TCP	22	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete
All TCP	TCP	0 - 65535	Anywhere-IP... 0.0.0.0/0	<input type="text"/> Delete

## 2.Create Instances:

We initiated the creation of three virtual machines or instances, naming them Master, node-1, and node-2. These instances will act as the nodes in our Kubernetes cluster.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. On the left, the 'Name and tags' section has 'master' entered in the 'Name' field. Below it, the 'Application and OS Images (Amazon Machine Image)' section shows a search bar and a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A 'Quick Start' tab is selected. On the right, the 'Summary' panel shows the configuration: 1 instance, Canonical Ubuntu 24.04 AMI, t2.medium instance type, and 1 volume (8 GiB). Buttons for 'Cancel', 'Launch instance', and 'Review commands' are at the bottom.

The screenshot continues the 'Launch an instance' wizard. The 'Network settings' section shows a key pair named 'key2309' and network configurations for 'vpc-06a8924e856c7a04d'. The 'Configure storage' section shows a root volume of 8 GiB using gp3 storage. Both sections have 'Edit' buttons. The right side of the screen displays the same 'Summary' panel as the previous step, including the 'Launch instance' button.

**Launch an instance**

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

**Name and tags**

Name: node

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

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

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

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

Browse more AMIs

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))

Free tier eligible

**Summary**

Number of instances: 2

When launching more than 1 instance, consider EC2 Auto Scaling

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

Virtual server type (instance type): t2.medium

Firewall (security group): node

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

Cancel      **Launch instance**      Review commands

**Additional costs apply for AMIs with pre-installed software**

**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: key2309

Create new key pair

**Network settings**

Network: vpc-06a8924e856c7a04d

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

Auto-assign public IP: Enabled

Additional charges apply when outside of free tier allowance

Firewall (security groups): Select existing security group

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

node\_sg-056d20b63b3793572 X  
VPC: vpc-06a8924e856c7a04d

Compare security group rules

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

**Summary**

Number of instances: 2

When launching more than 1 instance, consider EC2 Auto Scaling

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

Virtual server type (instance type): t2.medium

Firewall (security group): node

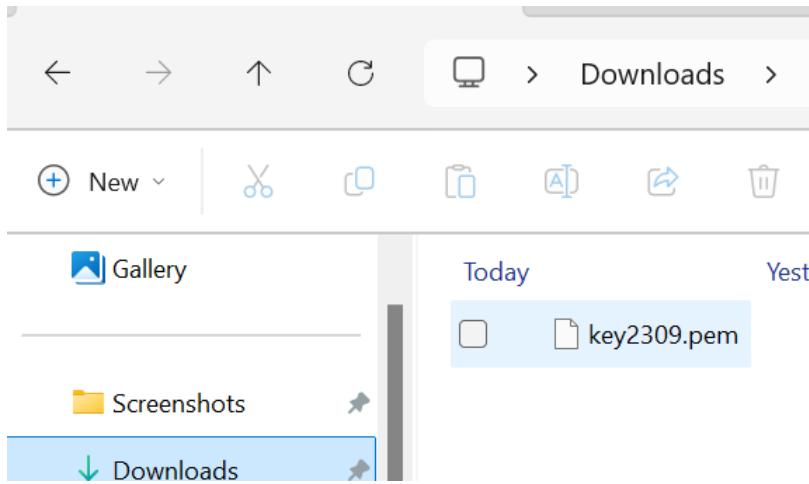
Storage (volumes): 1 volume(s) - 8 GiB

Cancel      **Launch instance**      Review commands

These the instances we have created successfully.

<input type="checkbox"/>	node-1	i-019d881a62651c90d	<span>Running</span>	<span>Q</span> <span>Q</span>	t2.medium	<span>⌚ Initializing</span>	<a href="#">View alarms</a> +	us-east-1b	ec2-3-86-95-118.comp...	3.86.95.118
<input type="checkbox"/>	node-2	i-05333ee8d27d26a64	<span>Running</span>	<span>Q</span> <span>Q</span>	t2.medium	<span>⌚ Initializing</span>	<a href="#">View alarms</a> +	us-east-1b	ec2-35-174-170-105.co...	35.174.170.105
<input type="checkbox"/>	master	i-011fbe3e0d5b69515	<span>Running</span>	<span>Q</span> <span>Q</span>	t2.medium	<span>⌚ Initializing</span>	<a href="#">View alarms</a> +	us-east-1b	ec2-54-89-251-73.com...	54.89.251.73

It is important to have the download file of your key.



Copy the command below in the example part and paste it in the cmd.

A screenshot of the AWS EC2 Connect interface. The top navigation bar shows 'EC2 &gt; Instances &gt; i-011fbe3e0d5b69515 &gt; Connect to instance'. Below this, the 'Connect to instance' section has an 'Info' link. It says 'Connect to your instance i-011fbe3e0d5b69515 (master) using any of these options'. There are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is selected), and 'EC2 serial console'. Under 'SSH client', the 'Instance ID' is listed as 'i-011fbe3e0d5b69515 (master)'. Below this are numbered steps: 1. Open an SSH client, 2. Locate your private key file. The key used to launch this instance is 'key2309.pem', 3. Run this command, if necessary, to ensure your key is not publicly viewable: 'chmod 400 "key2309.pem"', 4. Connect to your instance using its Public DNS: 'ec2-54-89-251-73.compute-1.amazonaws.com'. An 'Example:' section shows the command 'ssh -i "key2309.pem" ubuntu@ec2-54-89-251-73.compute-1.amazonaws.com'. A note in a box says: 'Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right is a 'Cancel' button.

## 2. Install Docker:

```
PS C:\Users\91900> ssh -i "key2309.pem" ubuntu@ec2-35-174-170-105.compute-1.amazonaws.com
The authenticity of host 'ec2-35-174-170-105.compute-1.amazonaws.com (35.174.170.105)' can't be established.
ED25519 key fingerprint is SHA256:777d6VJLUANXWdvC2Rqcqg6Jscu79S7iwHCjSgnMwM0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-174-170-105.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Mon Sep 23 15:19:47 UTC 2024

System load: 0.0          Processes:           112
Usage of /:   22.8% of 6.71GB   Users logged in:  0
Memory usage: 5%           IPv4 address for enX0: 172.31.86.235
Swap usage:   0%
```

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

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

```
ubuntu@ip-172-31-95-11:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mQINBFit2ioBEADhWpZ8/wvZ6hUTiX0wQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRFVMTSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYfOS8QcixNzHUYnb6emjLANyEVlZzeqo7XKL7
UrWV5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIVvWXqw0N
```

```
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 29.1 MB in 5s (6401 kB/s)
Reading package lists... Done
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.
```

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

```
ubuntu@ip-172-31-95-11:~$ sudo apt-get update  
sudo apt-get install -y docker-ce  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease  
Reading package lists... Done  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy format  
, see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done
```

```
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.
```

**No user sessions are running outdated binaries.**

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

**3.Start Docker:**

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

```
ubuntu@ip-172-31-95-11:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

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

```
ubuntu@ip-172-31-95-11:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

5: Run the below command to install Kubernets.

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

```
ubuntu@ip-172-31-95-11:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyring
s/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/ap
t/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

**Install Kubernetes components:**

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

```
ubuntu@ip-172-31-95-11:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]  
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 6051 B in 1s (11.5 kB/s)
```

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

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

```
ubuntu@ip-172-31-95-11:~$ sudo systemctl enable --now kubelet  
sudo apt-get install -y containerd  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  runc  
The following packages will be REMOVED:  
  containerd.io docker-ce  
The following NEW packages will be installed:
```

```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

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

```
ubuntu@ip-172-31-95-11:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2
```

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

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

sudo systemctl restart containerd

```
ubuntu@ip-172-31-95-11:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-09-23 15:35:29 UTC; 229ms ago
     Docs: https://containerd.io
 Main PID: 4614 (containerd)
    Tasks: 7
   Memory: 13.1M (peak: 14.4M)
      CPU: 64ms
     CGroup: /system.slice/containerd.service
             └─4614 /usr/bin/containerd

Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.093975568Z" level=info msg="serving..." address=/run/conta>
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094023950Z" level=info msg="serving..." address=/run/conta>
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094053439Z" level=info msg="Start subscribing containerrd>
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094080493Z" level=info msg="Start recovering state"
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094120880Z" level=info msg="Start event monitor"
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094128986Z" level=info msg="Start snapshots syncer"
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094137306Z" level=info msg="Start cni network conf syncer"
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094142936Z" level=info msg="Start streaming server"
Sep 23 15:35:29 ip-172-31-95-11 containerd[4614]: time="2024-09-23T15:35:29.094178857Z" level=info msg="containerd successfully bootstrapped"
Sep 23 15:35:29 ip-172-31-95-11 systemd[1]: Started containerd.service - containerd container runtime.
Lines 1-21/21 (END)
```

```
ubuntu@ip-172-31-95-11:~$ 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:
```

```
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
```

```
Running kernel seems to be up-to-date.
```

```
No services need to be restarted.
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

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

**6.Initialize kubeadm on Master Node:**

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

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

On the Master node, initialize the Kubernetes cluster using kubeadm. This process sets up the Kubernetes control plane and generates commands for joining worker nodes:

Copy the commands displayed in the output of the initialization process to configure permissions and obtain the join token. This includes a join command link needed for worker nodes to connect to the master.

```
ubuntu@ip-172-31-95-11:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kinit] 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'
W0923 15:48:50.266640    5352 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
```

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 172.31.95.11:6443 --token 16q9ib.pv09d9wixd10d50s \
--discovery-token-ca-cert-hash sha256:7ec275faa3af2297d3e809dea0a29175b2d2af6db63d2673456039bdbba306fc6
ubuntu@ip-172-31-95-11:~$
```

```
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-172-31-95-11:~$ 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-172-31-95-11:~$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-95-11	NotReady	control-plane	2m50s	v1.31.1

## 7.Join Worker Nodes:

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

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

```
ubuntu@ip-172-31-86-235:~$ sudo kubeadm join 172.31.95.11:6443 --token 16q9ib.pv09d9wixd10d50s \
--discovery-token-ca-cert-hash sha256:7ec275faa3af2297d3e809dea0a29175b2d2af6db63d2673456039bdbba306fc6
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.238074ms
[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-172-31-95-11:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-86-235  NotReady <none>    70s    v1.31.1
ip-172-31-91-211  NotReady <none>    61s    v1.31.1
ip-172-31-95-11   NotReady control-plane  5m25s   v1.31.1
ubuntu@ip-172-31-95-11:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created

ubuntu@ip-172-31-95-11:~$ kubectl get nodes -o wide
NAME           STATUS    ROLES      AGE     VERSION   INTERNAL-IP      EXTERNAL-IP   OS-IMAGE        KERNEL-VERSION   CONTA
INER-RUNTIME
ip-172-31-86-235  Ready    <none>    3m44s   v1.31.1   172.31.86.235  <none>       Ubuntu 24.04 LTS  6.8.0-1012-aws  conta
inerd://1.7.12
ip-172-31-91-211  Ready    <none>    3m35s   v1.31.1   172.31.91.211  <none>       Ubuntu 24.04 LTS  6.8.0-1012-aws  conta
inerd://1.7.12
ip-172-31-95-11   Ready    control-plane  7m59s   v1.31.1   172.31.95.11   <none>       Ubuntu 24.04 LTS  6.8.0-1012-aws  conta
inerd://1.7.12
```

```
ubuntu@ip-172-31-95-11:~$ kubectl label node ip-172-31-91-211 kubernetes.io/role=Node1
node/ip-172-31-91-211 labeled
```

```
ubuntu@ip-172-31-95-11:~$ kubectl label node ip-172-31-95-11 kubernetes.io/role=worker
node/ip-172-31-95-11 labeled
```

```
ubuntu@ip-172-31-95-11:~$ kubectl label node ip-172-31-86-235 kubernetes.io/role=Node2
error: 'kubernetes.io/role' already has a value (Node1), and --overwrite is false
```

```
ubuntu@ip-172-31-95-11:~$ kubectl label node ip-172-31-86-235 kubernetes.io/role=Node2 --overwrite
node/ip-172-31-86-235 labeled
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION
CONTAINER-RUNTIME								
ip-172-31-86-235	Ready	Node2 containerd://1.7.12	9m29s	v1.31.1	172.31.86.235	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws
ip-172-31-91-211	Ready	Node1 containerd://1.7.12	9m20s	v1.31.1	172.31.91.211	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws
ip-172-31-95-11	Ready	control-plane,worker containerd://1.7.12	13m	v1.31.1	172.31.95.11	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws

## CONCLUSION :

**Docker installation:** After installing docker on all instances, sometimes docker services may fail to restart.

**Network Configuration Issue:** Connectivity issue between master and worker modes might be caused by the firewall blocking that required ports.

**CrashLoopBackOff:** There are errors indicating that the containers for Kubernetes components are restarting repeatedly but failing to start properly.

## AdvanceDevops Experiment 4

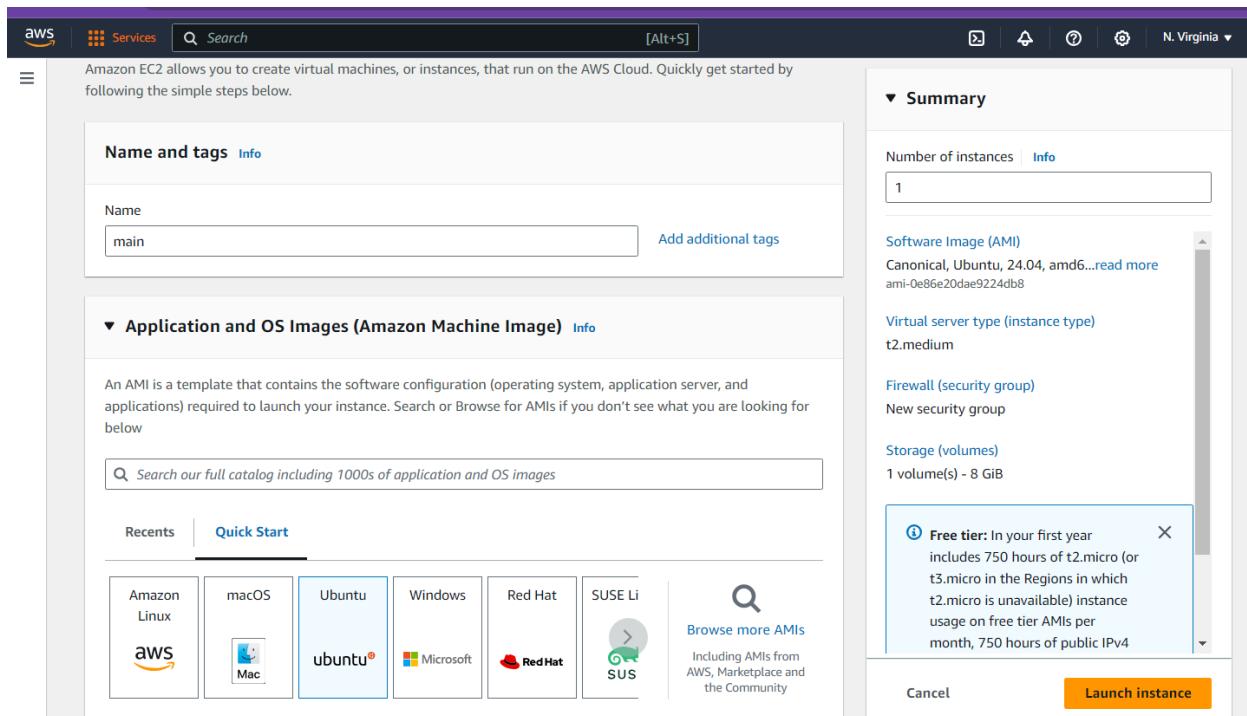
**Aim:** To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

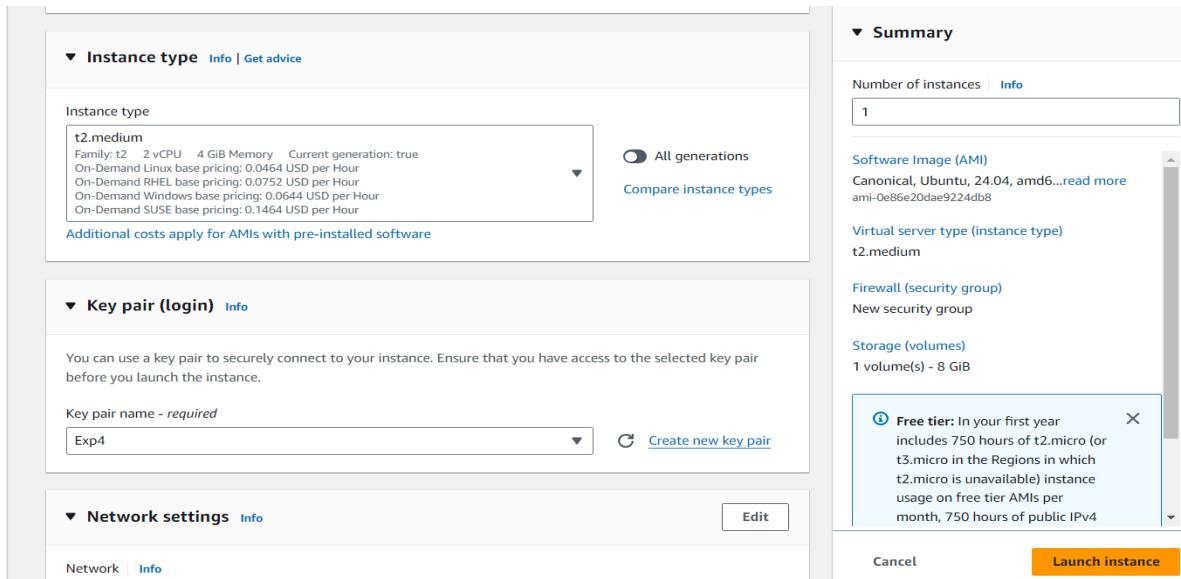
### Theory:

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

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

**Step 1:** Log in to your AWS Academy/personal account and launch a new Ec2 Instance. Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.





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

Instances (1) <a href="#">Info</a>										
Last updated less than a minute ago										
<a href="#">Connect</a> <a href="#">Actions</a> <a href="#">Launch instances</a>										
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public		
main	i-045e45a4940c82f2b	<span>Running</span>	t2.medium	<span>2/2 checks passed</span>	<a href="#">View alarms</a>	us-east-1c	ec2-54-89-80-249.com...	54.89		

EC2 > Instances > i-0c58e4cedc38ab19c > Connect to instance

### Connect to instance [Info](#)

Connect to your instance i-0c58e4cedc38ab19c (main1) using any of these options

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

Instance ID: [i-0c58e4cedc38ab19c \(main1\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Exp4.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 "Exp4.pem"`
- Connect to your instance using its Public DNS:  
`ssh -i "Exp4.pem" ubuntu@ec2-107-22-62-86.compute-1.amazonaws.com`

Example:  
`ssh -i "Exp4.pem" ubuntu@ec2-107-22-62-86.compute-1.amazonaws.com`

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

[Cancel](#)

**Step 3:** Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i ..... ) in the terminal.

Run the below commands to install and setup Docker.

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

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

```
ubuntu@ip-172-31-92-253:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg  
> /dev/null  
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable"  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'  
Description:  
Archive for codename: noble components: stable  
More info: https://download.docker.com/linux/ubuntu  
Adding repository.  
Press [ENTER] to continue or Ctrl-c to cancel.  
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.  
list  
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
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 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
```

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

```
ubuntu@ip-172-31-92-253:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroups-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 141 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
```

```
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.
ubuntu@ip-172-31-92-253:~$ |
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
ubuntu@ip-172-31-92-253:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-92-253:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-92-253:~$ |
```

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

```
ubuntu@ip-172-31-92-253:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-92-253:~$ |
```

**Step 5:** Run the below command to install Kubernets.

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

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

**/etc/apt/sources.list.d/kubernetes.list**

```
ubuntu@ip-172-31-92-253:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

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

```
ubuntu@ip-172-31-92-253:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 6051 B in 1s (11.3 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
```

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-92-253:~$ |
```

**sudo systemctl enable --now kubelet**

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

```
ubuntu@ip-172-31-92-253:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kinit] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0925 16:13:24.636784    4473 checks.go:1080] [preflight] WARNING: Couldn't create the interface used
for talking to the container runtime: failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: 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
ubuntu@ip-172-31-92-253:~$ |
```

**Now We have got an error.**

**So we have to perform some additional commands as follow.**

**sudo apt-get install -y containerd**

```
ubuntu@ip-172-31-92-253:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7
  libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 141 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (79.0 MB/s)
(Reading database ... 68064 files and directories currently installed.)
```

```
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-92-253:~$ |
```

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

```
ubuntu@ip-172-31-92-253:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-09-25 16:19:04 UTC; 255ms ago
     Docs: https://containerd.io
 Main PID: 5059 (containerd)
    Tasks: 8
   Memory: 13.4M (peak: 13.8M)
     CPU: 59ms
    CGroup: /system.slice/containerd.service
              └─5059 /usr/bin/containerd

Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.090155391Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091499805Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091559889Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091567736Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091598787Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091606039Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091607687Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091646696Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 containerd[5059]: time="2024-09-25T16:19:04.091701973Z" level=info
Sep 25 16:19:04 ip-172-31-92-253 systemd[1]: Started containerd.service - containerd container runtime
lines 1-21/21 (END)
```

**sudo apt-get install -y socat**

```
ubuntu@ip-172-31-92-253:~$ 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 141 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [37
4 kB]
Fetched 374 kB in 0s (17.2 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

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

```
ubuntu@ip-172-31-92-253:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kinit] 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 16:26:17.463609      5539 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 "regi
stry.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-172-31-92-253 kubernetes kubernetes.default
kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.92.253]
[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-172-31-92-253 localhost] and IPs [172.31
.92.253 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-92-253 localhost] and IPs [172.31.9
2.253 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
```

```
tificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.92.253:6443 --token 90n342.w0y0ehbtppbqeocz \
    --discovery-token-ca-cert-hash sha256:072239c6fcbbd2b8842bd4badd167478379f455ddc7525f46bb5902
87f594319
```

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

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

```
ubuntu@ip-172-31-92-253:~$ 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-172-31-92-253:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/D
ocumentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

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

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

```
ubuntu@ip-172-31-92-253:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

**kubectl get pods**

```
ubuntu@ip-172-31-92-253:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-llw22  0/1     Pending   0          25s
nginx-deployment-d556bf558-xbpvn  0/1     Pending   0          25s
```

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

**kubectl port-forward \$POD\_NAME 8080:80**

```
ubuntu@ip-172-31-92-253:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-92-253:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
```

**kubectl get nodes**

```
ubuntu@ip-172-31-92-253:~$ kubectl get nodes
NAME            STATUS   ROLES      AGE   VERSION
ip-172-31-92-253   Ready   control-plane   7m43s   v1.31.1
```

**get pods**

```
ubuntu@ip-172-31-92-253:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-llw22  1/1     Running   0          11m
nginx-deployment-d556bf558-xbpvn  1/1     Running   0          11m
```

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

**kubectl port-forward \$POD\_NAME 8080:80**

```
ubuntu@ip-172-31-92-253:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-92-253:~$ kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

**Step 8: Verify your deployment**

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

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

```
curl --head http://127.0.0.1:8080
```

```
PS C:\Users\LENOVO> ssh -i "Exp4.pem" ubuntu@ec2-54-164-58-232.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep 25 16:45:32 UTC 2024

System load: 0.07          Processes:           155
Usage of /: 55.5% of 6.71GB Users logged in:      1
Memory usage: 19%          IPv4 address for enX0: 172.31.92.253
Swap usage: 0%           

Expanded Security Maintenance for Applications is not enabled.

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

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

```
Last login: Wed Sep 25 16:12:02 2024 from 203.194.102.247
Last login: Wed Sep 25 16:12:02 2024 from 203.194.102.277
ubuntu@ip-172-31-92-253:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 25 Sep 2024 16:45:49 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

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

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

**Conclusion:**

- 1. EC2 Instance Launch Issues:** Incorrect AMI Selection: Selecting the wrong Amazon Machine Image (AMI) could cause issues, especially if it doesn't support the required software for Kubernetes.
- 2. Kubernetes Installation Issues:** Installation Fails Due to Network Issues: Sometimes, the installation of Kubernetes can fail due to network errors or misconfigured package repositories.
- 3. Nginx Deployment Issues:** Nginx Server Not Running: After deploying Nginx, the server might not start due to insufficient resources on the EC2 instance or missing configurations.

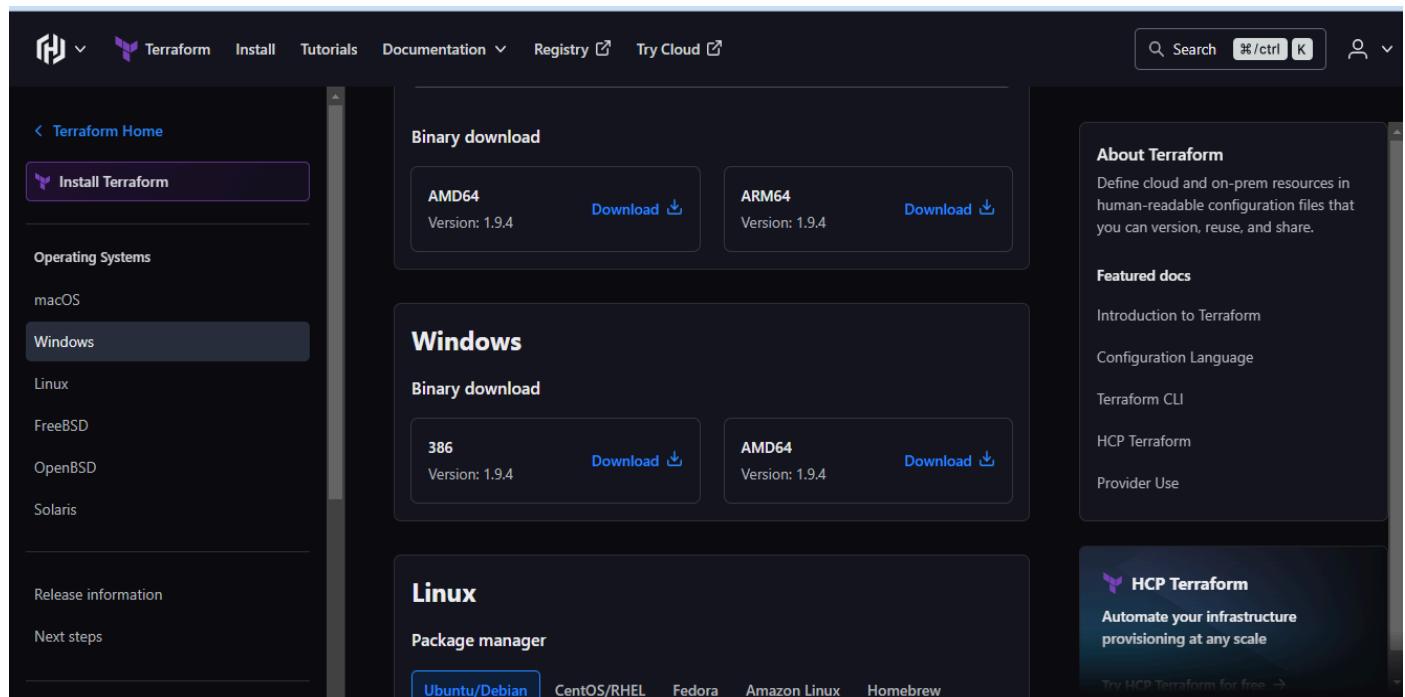
## Advance Devops Experiment 5

**Aim:** To understand terraform lifecycle, core concepts/terminologies and install it on Windows.

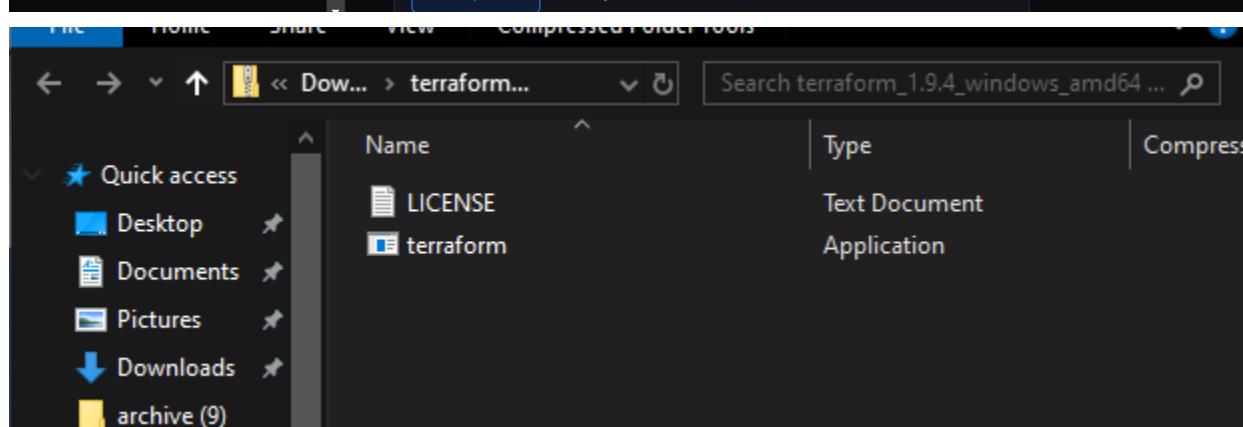
### 1. Download the Terraform:

click on the link <https://www.terraform.io/downloads.html>

Select the Operating System and download Terraform version according to configurations.



The screenshot shows the Terraform website's download section. On the left, a sidebar lists operating systems: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, Solaris, Release information, and Next steps. The main content area is titled "Binary download" and shows two options for Windows: "AMD64 Version: 1.9.4" and "ARM64 Version: 1.9.4", each with a "Download" button. Below this, there are sections for "Windows" (Binary download for 386 and AMD64) and "Linux" (Package manager for Ubuntu/Debian, CentOS/RHEL, Fedora, Amazon Linux, and Homebrew). To the right, there is an "About Terraform" summary, "Featured docs" links (Introduction to Terraform, Configuration Language, Terraform CLI, HCP Terraform, Provider Use), and an "HCP Terraform" advertisement.

The screenshot shows a file explorer window with the path "Dow... > terraform...". The contents of the folder are listed in a table:

Name	Type	Compress
LICENSE	Text Document	
terraform	Application	

### 3. Find the Terraform Executable Path:

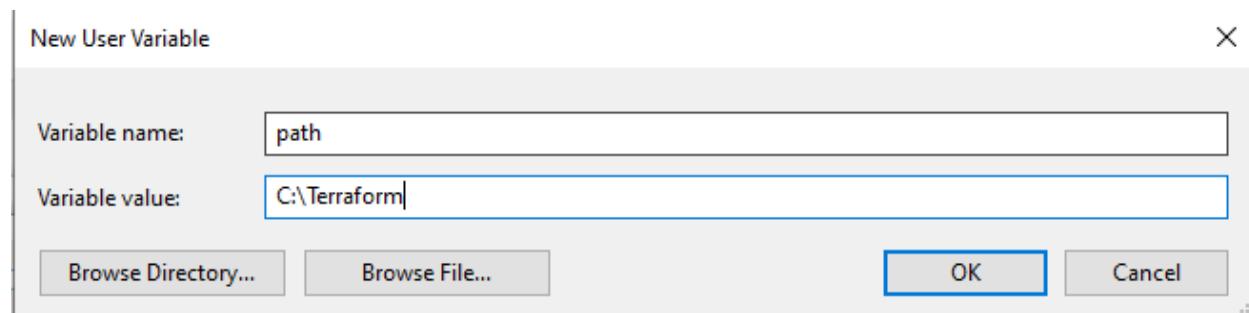
After installation, locate the directory where Terraform executable (terraform.exe) is present. For example, it could be in C:\terraform or any other path you chose during Installation.

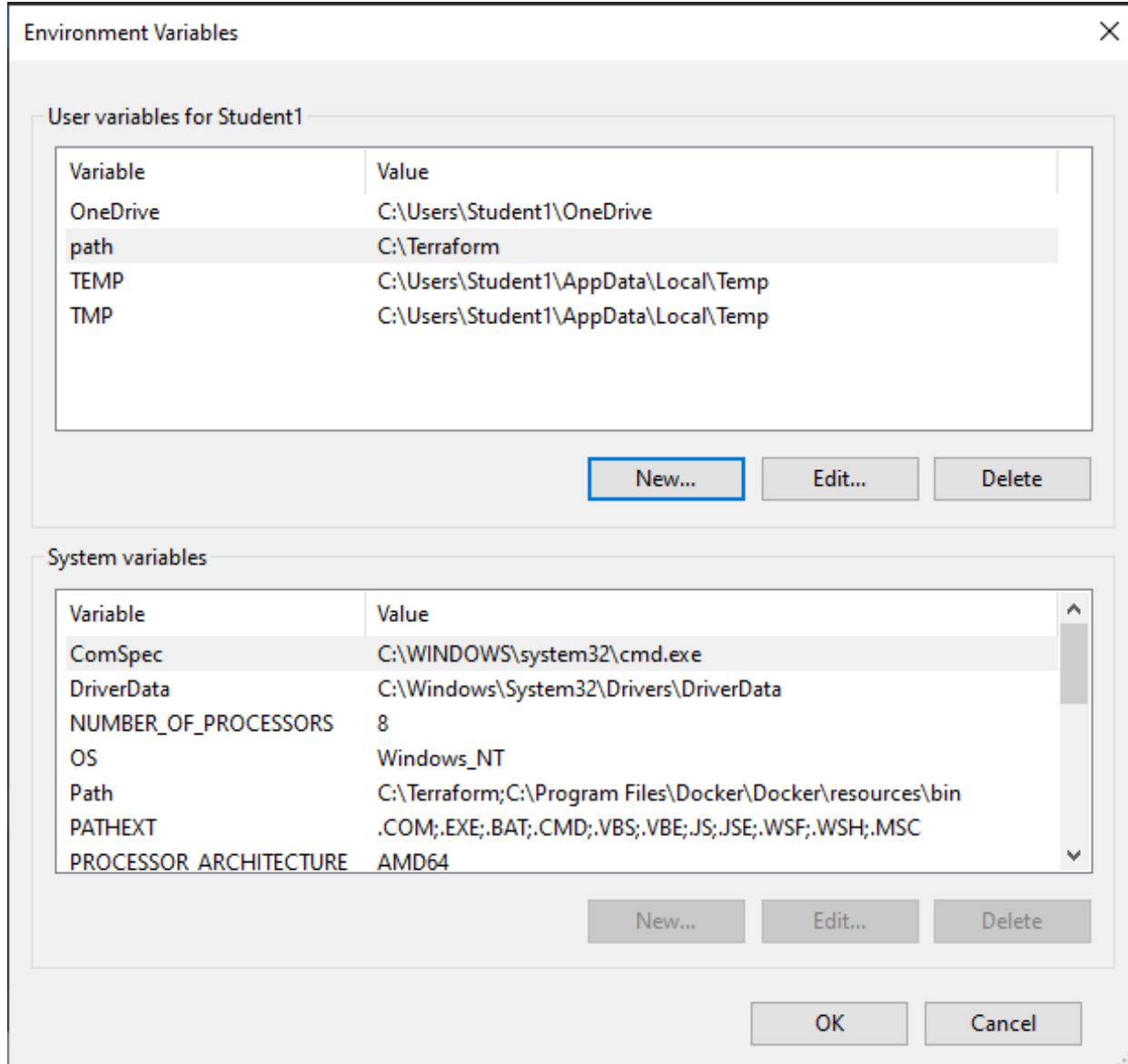
### 4. Edit System Environment Variables:

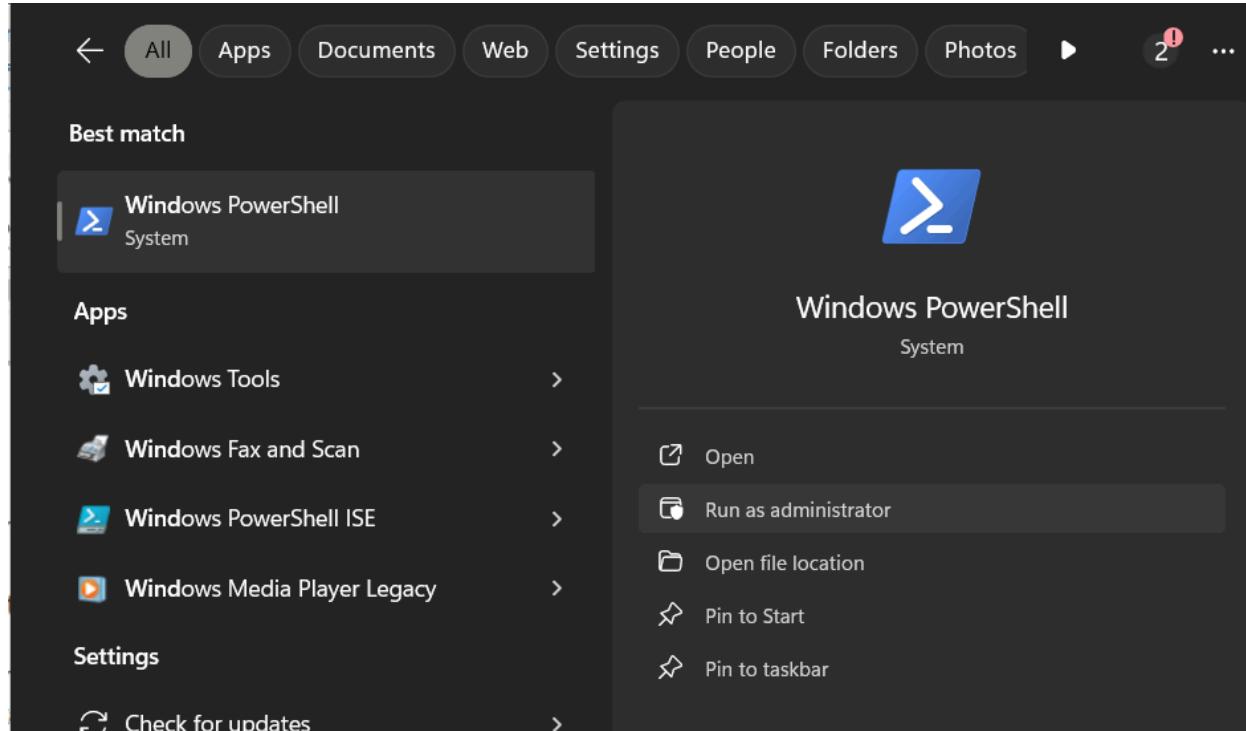
In the Environment Variables window (System Properties), under the section System Variables, find and select the variable named Path, then click Edit.

### 5. Add Terraform to Path:

In the Edit Environment Variable window, click New to add a new entry. Enter the path to the directory where Terraform executable (terraform.exe) is located. For example, if Terraform is installed in C:\terraform, add C:\terraform to the list.



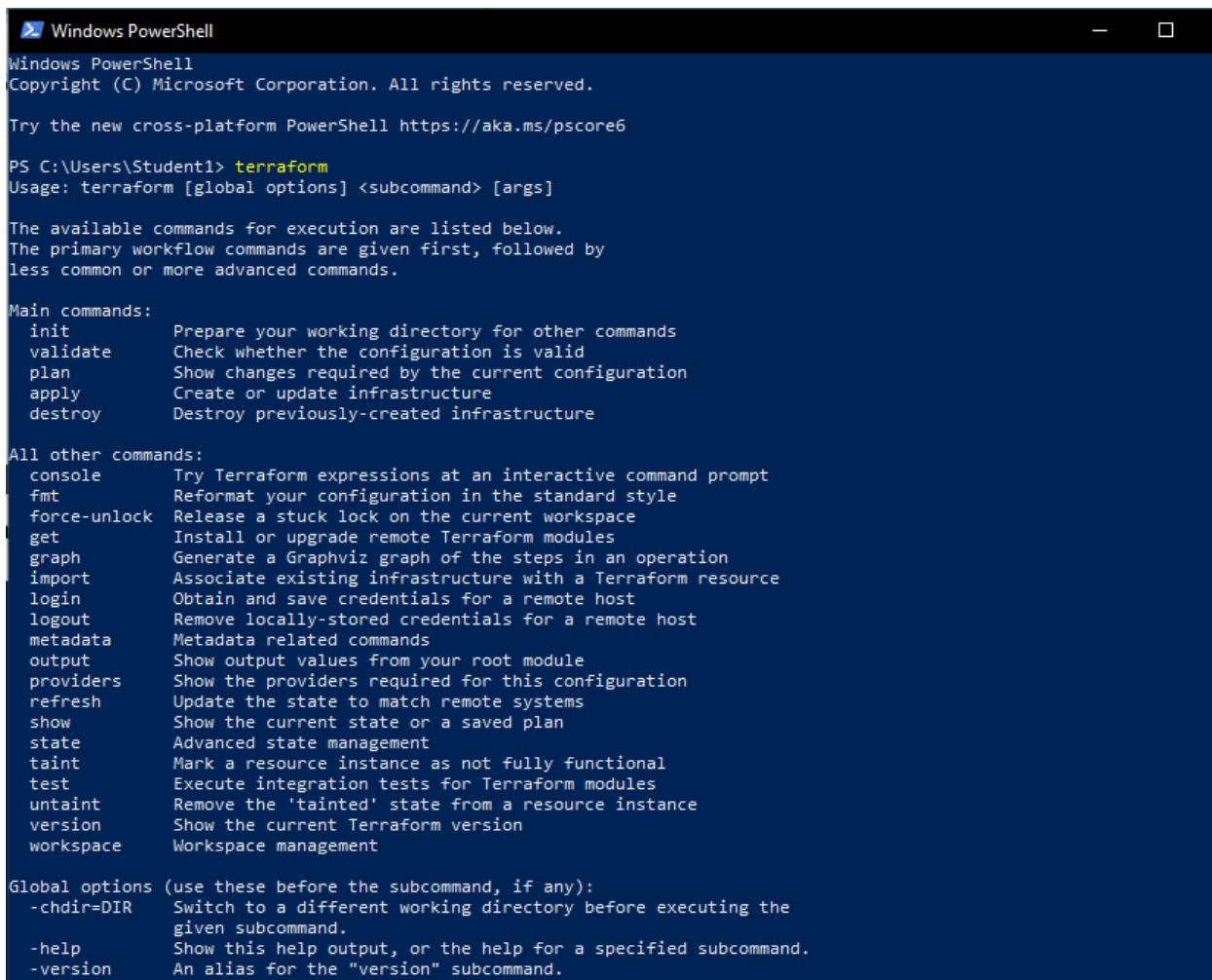


**6. Open PowerShell with Administrator Access.**

## 7.Verify Installation:

Open a new command prompt (to ensure it picks up the updated environment variables).

Type **terraform** and press Enter. You should see the Terraform version information printed on the screen if the path configuration was successful.



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

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

PS C:\Users\Student1> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

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

## AdvanceDevops Experiment: 6

**AIM: To Build, change, and destroy AWS /GCP/ Microsoft Azure/ Digital Ocean infrastructure using Terraform. (S3 bucket or Docker)**

### 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\INFT505-16> 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
```

```
PS C:\Users\INFT505-16> docker --version
Docker version 24.0.6, build ed223bc
PS C:\Users\INFT505-16> |
```

**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. Copy the Script into it.

**Script:**

```
terraform {  
    required_providers {  
        docker = {  
            source = "kreuzwerker/docker"  
            version = "2.21.0"  
        }  
    }  
}  
  
provider "docker" {  
    host = "npipe:///./pipe//docker_engine"  
}  
  
# Pulls the Ubuntu image  
resource "docker_image" "ubuntu" {  
    name = "ubuntu:latest"  
}  
  
# Create a container  
resource "docker_container" "foo" {  
    image = docker_image.ubuntu.image_id  
    name = "foo"  
}
```

```

 1 terraform {
 2   required_providers {
 3     docker = {
 4       source  = "kreuzwerker/docker"
 5       version = "2.21.0"
 6     }
 7   }
 8 }
 9
10 provider "docker" {
11   host = "npipe:///./pipe/docker_engine"
12 }
13
14 # Pull the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "3600"]
24 }
25
26

```

### Step 3: Execute Terraform Init command to initialize the resources

```
C:\Users\INFT505-16>cd desktop\TerraformScripts\Docker
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
    https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

Terraform has been successfully initialized!

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

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

**Step 4:** Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : “terraform apply”

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "3600",
    ]
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)

    + output          = (known after apply)
    + repo_digest     = (known after apply)
}

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Still creating... [40s elapsed]
docker_image.ubuntu: Still creating... [50s elapsed]
docker_image.ubuntu: Creation complete after 53s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=353bd0cae537e335931797ed86d2b603c682520b71c2af7d5b72f3c09eed2b11]
```

Docker images, After Executing Apply step:

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   edbfe74c41f8  3 weeks ago   78.1MB
sonarqube       latest   3183d6818c6e  10 months ago 716MB
```

**Step 5:** Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=353bd0cae537e335931797ed86d2b603c682520b71c2af7d5b72f3c09eed2b11]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach          = false -> null
    - command         = [
        - "sleep",
        - "3600",
    ] -> null
    - cpu_shares      = 0 -> null
    - dns              = [] -> null
    - dns_opts         = [] -> null
    - dns_search       = [] -> null
    - entrypoint       = [] -> null
    - env              = [] -> null
    - gateway          = "172.17.0.1" -> null
    - gateway_ip       = "" -> null
}

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=353bd0cae537e335931797ed86d2b603c682520b71c2af7d5b72f3c09eed2b11]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

**Step 6:** This command outputs the state or plan in a human-readable format, helping you review the details of your configuration..

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform show
# docker_container.foo:
resource "docker_container" "foo" {
    attach          = false
    bridge          = null
    command         = [
        "sleep",
        "3600",
    ]
    cpu_set          = null
    cpu_shares      = 0
    domainname      = null
    entrypoint       = []
    env              = []
    gateway          = "172.17.0.1"
    hostname         = "353bd0cae537"
    id               = "353bd0cae537e335931797ed86d2b603c682520b71c2af7d5b72f3c09eed2b11"
    image            = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest"
    init             = false
    ip_address       = "172.17.0.2"
    ip_prefix_length = 16
```

**Step 7:** This command generates a visual graph of your Terraform resources, which can help you understand the dependencies and relationships between them.

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform graph
digraph G {
    rankdir = "RL";
    node [shape = rect, fontname = "sans-serif"];
    "docker_container.foo" [label="docker_container.foo"];
    "docker_image.ubuntu" [label="docker_image.ubuntu"];
    "docker_container.foo" -> "docker_image.ubuntu";
}
```

**Step 8:** This command lists all the resources tracked by the Terraform state, allowing you to see which resources have been created and are being managed by Terraform..

```
C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>terraform state list
docker_container.foo
docker_image.ubuntu

C:\Users\INFT505-16\Desktop\TerraformScripts\Docker>
```

## AdvanceDevops Experiment 7

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

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

### What problems does SAST solve?

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

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

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

### Why is SAST important?

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

dramatic results in the overall quality of the code developed.

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

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

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

### **Integrating Jenkins with SonarQube:**

Windows installation

Step 1 Install JDK 1.8

Step 2 download and install jenkins

<https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows>

### Ubuntu installation

<https://www.digitalocean.com/community/tutorials/how-to-install-java-with-a-pt-on-ubuntu-20-04#installing-the-default-jre-jdk>

Step 1 Install JDK 1.8

sudo apt-get install openjdk-8-jre

sudo apt install default-jre

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04>

[Open SSH](#)

### Prerequisites:

- [Jenkins installed](#)
- [Docker Installed](#) (for SonarQube)  
(sudo apt-get install docker-ce=5:20.10.15~3-0~ubuntu-jammy  
docker-ce-cli=5:20.10.15~3-0~ubuntu-jammy containerd.io docker-compose-plugin)
- SonarQube Docker Image

### Steps to integrate Jenkins with SonarQube

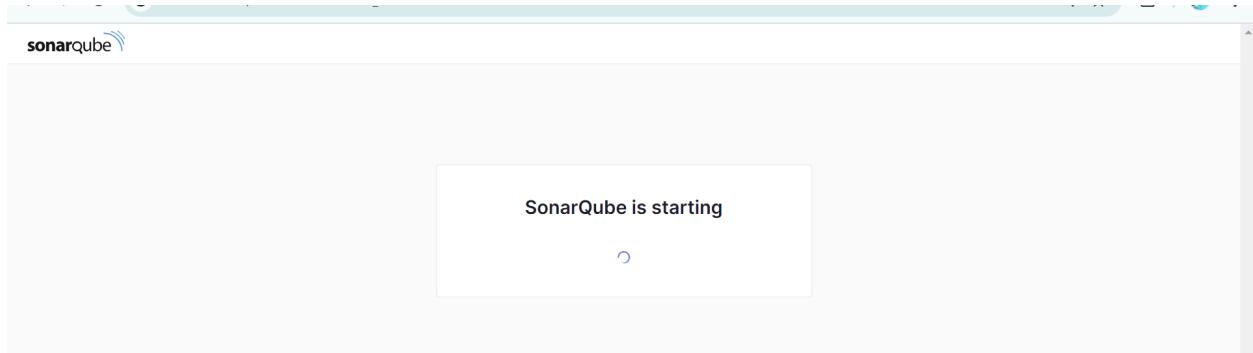
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -

```
PS C:\Users\91900> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:lates
t
8b2a833004ac39a9b009118bacac47e5808c9ec8df3f59f8657bd23fa23f48f2
```

**Warning: run below command only once**

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

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



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

A screenshot of the SonarQube login interface. At the top, there's a logo with the word "sonar" and a red circular icon. Below it is a "Log in to SonarQube" form. The form has two input fields: "Login \*" containing "admin" and "Password \*" containing "\*\*\*\*\*". At the bottom of the form are two buttons: "Go back" and a blue "Log in" button.

- Create a manual project in SonarQube with the name **sonarqube**

1 of 2

## Create a local project

**Project display name \***

sonarqube-test

**Project key \***

sonarqube-test

**Main branch name \***

main

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

2 of 2

## Set up project for Clean as You Code

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

Choose the baseline for new code for this project

 Use the global setting**Previous version**

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

Recommended for projects following regular versions or releases.

 Define a specific setting for this project Previous version

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

Recommended for projects following regular versions or releases.

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

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "sonar". Below the search bar, there are three tabs: "Updates" (with 34 notifications), "Available plugins" (selected), and "Installed plugins". In the "Available plugins" section, a search result for "SonarQube Scanner 2.17.2" is displayed. The result includes a checkbox, the name "SonarQube Scanner 2.17.2", the status "Released", and the date "7 mo 8 days ago". A brief description below states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality."

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

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name: sonarqube

Server URL: http://localhost:9000

Server authentication token: - none -

Advanced

**Save**   **Apply**

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

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name: sonarqube

Install automatically

☰ Install from Maven Central

Version: SonarQube Scanner 6.2.0.4584

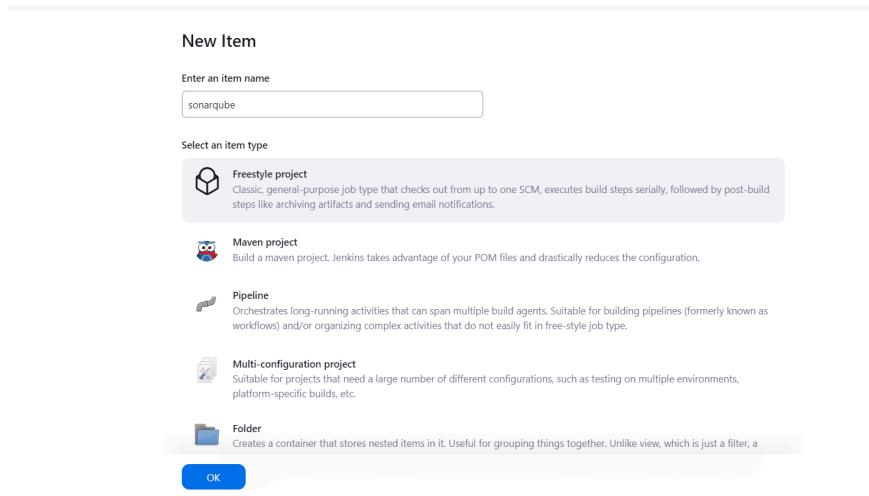
Add Installer

Add SonarQube Scanner

**Save**   **Apply**

---

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



9. Choose this GitHub repository in Source Code Management.

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

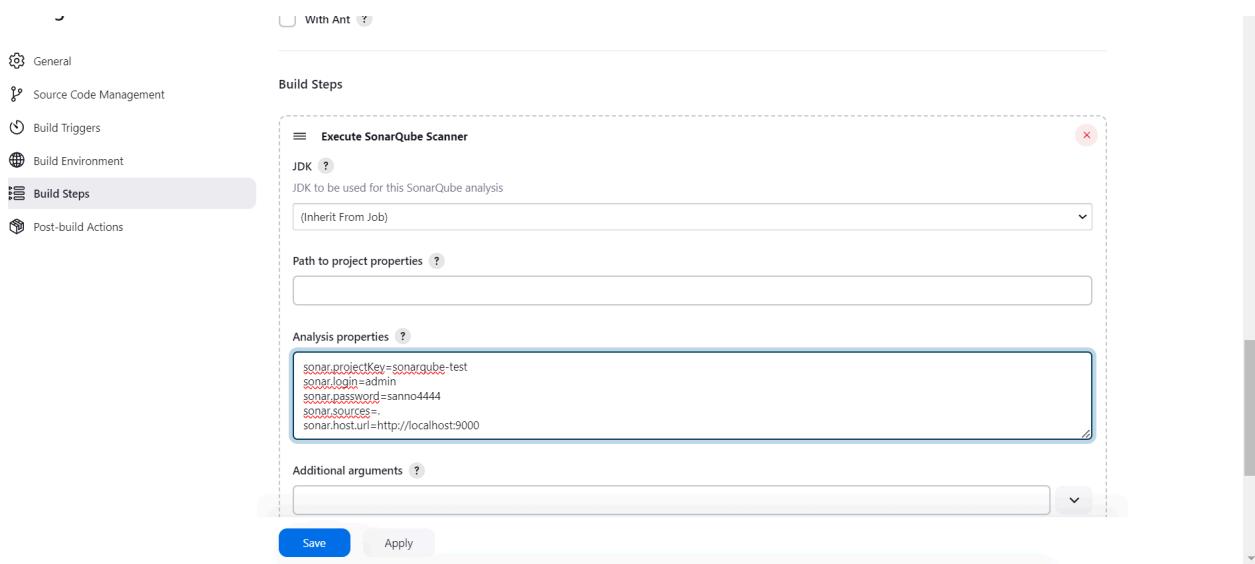
It is a sample hello-world project with no vulnerabilities and issues, just to

test

The screenshot shows the Jenkins 'Configure' screen for a new item. The left sidebar has tabs for General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Source Code Management' tab is selected. Under 'Source Code Management', the 'Git' section is active. The 'Repository URL' field contains the value 'https://github.com/shazforiot/MSBuild\_firstproject.git'. Below it, a red error message says 'Please enter Git repository.' The 'Credentials' dropdown is set to '- none -'. There is an 'Advanced' link and a 'Add Repository' button. At the bottom, there are 'Save' and 'Apply' buttons.

the integration.

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



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

The screenshot shows the SonarQube 'Administration' > 'Security' > 'Global Permissions' page. It lists global permissions for groups and users. The 'Administrator' user 'admin' is selected. The table shows the following permissions:

	Administrator System	Administer	Execute Analysis	Create
<b>sonar-administrators</b> 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
<b>sonar-users</b> 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
<b>Anyone DEPRECATED</b> 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
<b>Administrator admin</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects

At the bottom, it says '4 of 4 shown'.

12. Run The Build.

The screenshot shows the Jenkins dashboard for a 'sonarqube' project. On the left, a sidebar menu includes options like Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area displays a green checkmark icon next to the text 'sonarqube'. Below it is the SonarQube logo (three blue wavy lines) and the text 'Permalinks'. A bulleted list provides links to the last four builds: Last build (#1), Last stable build (#1), Last successful build (#1), and Last completed build (#1), all from 8 min 33 sec ago. At the bottom, a 'Build History' card shows a single build labeled '#1' from Sep 25, 2024, at 10:09 AM. It includes links for Atom feed for all and Atom feed for failures.

Check the console output.

The screenshot shows the Jenkins console output for build #1. The left sidebar includes Status, Changes, Console Output (which is selected), Edit Build Information, Delete build #1, Timings, and Git Build Data. The main area is titled 'Console Output' and shows the following log:

```

Started by user Vedant Dhone
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  Cloning repository https://github.com/shazforiot/MSBuild_firstproject.git
    > git.exe init C:\ProgramData\Jenkins\jenkins\workspace\sonarqube # timeout=10
  Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
    > git.exe --version # timeout=10
    > git --version # 'git' version 2.42.0.windows.2'
    > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
    > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
    > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
  Avoid second fetch
    > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
  Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
    > git.exe config core.sparsecheckout # timeout=10
    > git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
  Commit message: "updated"
  First time build. Skipping changelog.
  Unpacking https://repo1.maven.org/maven2/org/sonarsource/scanner/cli/sonar-scanner-cli/6.2.0.4584/sonar-scanner-cli-6.2.0.4584.zip to C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube on Jenkins
  [sonarqube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=.. -Dsonar.password=sanno444 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube

```

```
SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
10:10:57.397 INFO Sensor C# [csharp] (done) | time=2ms
10:10:57.397 INFO Sensor Analysis Warnings import [csharp]
10:10:57.399 INFO Sensor Analysis Warnings import [csharp] (done) | time=4ms
10:10:57.401 INFO Sensor C# File Caching Sensor [csharp]
10:10:57.405 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
10:10:57.405 INFO Sensor C# File Caching Sensor [csharp] (done) | time=5ms
10:10:57.405 INFO Sensor Zero Coverage Sensor
10:10:57.424 INFO Sensor Zero Coverage Sensor (done) | time=19ms
10:10:57.428 INFO SCM Publisher SCM provider for this project is: git
10:10:57.430 INFO SCM Publisher 4 source files to be analyzed
10:10:58.315 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=883ms
10:10:58.320 INFO CPD Executor Calculating CPD for 0 files
10:10:58.363 INFO CPD Executor CPD calculation finished (done) | time=0ms
10:10:58.372 INFO SCM revision ID 'f2bc042c04c6e72427c380bcace6d6fee7b49adf'
10:10:58.843 INFO Analysis report generated in 226ms, dir size=201.0 kB
10:10:58.903 INFO Analysis report compressed in 45ms, zip size=22.2 kB
10:10:59.397 INFO Analysis report uploaded in 491ms
10:10:59.401 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:10:59.402 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:10:59.403 INFO More about the report processing at http://localhost:9000/api/ce/task?id=2620d8fe-82f7-4a3c-999f-1ed8a7b15249
10:10:59.429 INFO Analysis total time: 30.223 s
10:10:59.431 INFO SonarScanner Engine completed successfully
10:10:59.519 INFO EXECUTION SUCCESS
10:10:59.521 INFO Total time: 47.815s
Finished: SUCCESS
```

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

The screenshot shows the SonarQube web interface. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation, the project 'sonarqube-test' is selected, and the branch 'main' is shown as 'Passed' with a green checkmark. A warning message indicates there are warnings in the last analysis. The main content area displays various quality gate metrics: Security (0 Open issues, A grade), Reliability (0 Open issues, A grade), Maintainability (0 Open issues, A grade), Accepted issues (0, with a note about valid issues not fixed), Coverage (0.0%, 86 lines), and Duplications (0.0%).

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

## Conclusion

**1. Docker Container Issues:** The SonarQube container might not start because your system doesn't have enough memory or processing power. SonarQube needs around 2GB of RAM to work properly, so if your system is low on resources, the container won't run.

**2. Login Problems in SonarQube:** You might have trouble logging in with the default username (admin) and password (admin). This could happen if there was a configuration issue with SonarQube or if the default password was changed during previous setups.

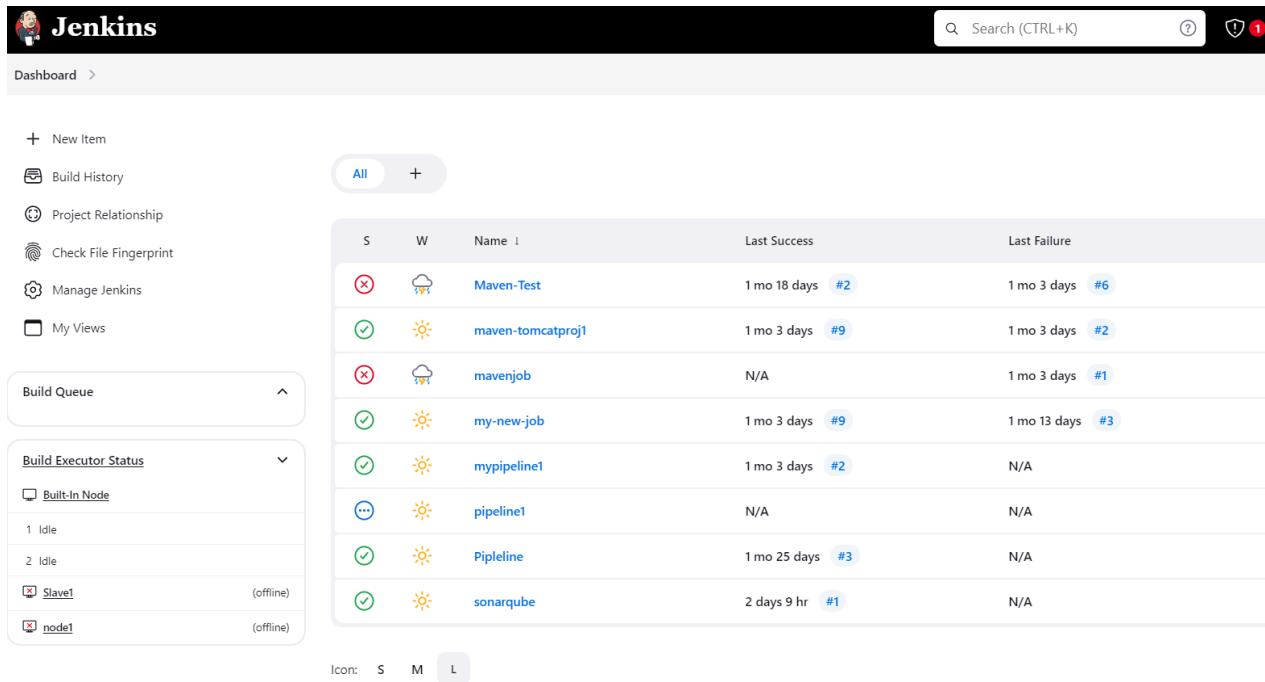
**3. Jenkins Plugin Installation Errors:** While installing the SonarQube Scanner plugin in Jenkins, you might encounter failures due to network issues or proxy settings, preventing the plugin from downloading correctly.

**4. Incorrect SonarQube Configuration in Jenkins:** While configuring SonarQube in Jenkins, entering the wrong project key, username, or password can cause the scan to fail. Ensuring accurate information is critical for a successful scan.

## Advance Devops Experiment 8

**Aim:** Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web Java / Python application. dive deep into this segment, let's first understand what is meant by the term 'pipeline'?

1. Open Jenkin dashboard.



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

- Header:** Jenkins logo, search bar, and a red notification badge with the number 1.
- Left Sidebar:**
  - New Item
  - Build History
  - Project Relationship
  - Check File Fingerprint
  - Manage Jenkins
  - My Views
- Build Queue:** A section showing the current build queue with no items.
- Build Executor Status:** A section showing the status of built-in nodes:
 

Node	Status	Idle
Slave1	(offline)	1
node1	(offline)	2
- Main Content:** A table listing Jenkins build jobs:
 

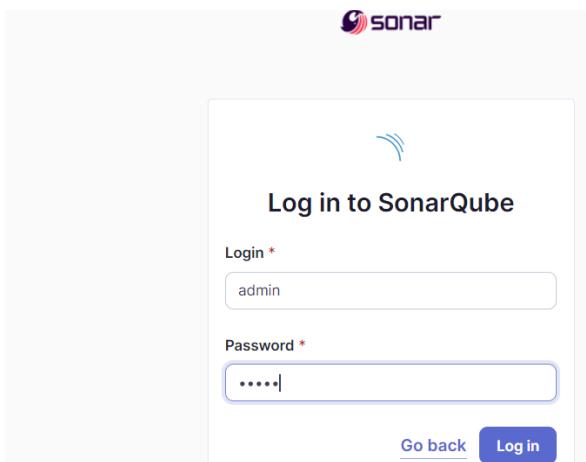
S	W	Name	Last Success	Last Failure
✗	☁️	Maven-Test	1 mo 18 days #2	1 mo 3 days #6
✓	☀️	maven-tomcatproj1	1 mo 3 days #9	1 mo 3 days #2
✗	☁️	mavenjob	N/A	1 mo 3 days #1
✓	☀️	my-new-job	1 mo 3 days #9	1 mo 13 days #3
✓	☀️	mypipeline1	1 mo 3 days #2	N/A
...	☀️	pipeline1	N/A	N/A
✓	☀️	Pipeline	1 mo 25 days #3	N/A
✓	☀️	sonarqube	2 days 9 hr #1	N/A
- Bottom:** Icon legend: S (Stable), M (Medium), L (Long).

2. Run SonarQube in a Docker container using this command -

```
C:\Users\91900>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
d23acccacd96c274f5f87912674ecf2d9adffff185a940c24740f44b29534485
```

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

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



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

1 of 2

### Create a local project

Project display name \*

 ✓

Project key \*

 ✓

Main branch name \*

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

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

Setup the project and come back to Jenkins Dashboard.

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

New Item

Enter an item name  
sonarqube14

Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

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 SonarQube Pipeline configuration interface. The top navigation bar includes 'Dashboard > sonarqube14 > Configuration'. Below this, the 'Configure' section is selected. Under 'Pipeline', the 'Definition' tab is chosen, and the 'Pipeline script' dropdown is set to 'Pipeline script'. A large text area contains a Groovy script for a Java project:

```
1 * node {
2     stage('Cloning the GitHub Repo') {
3         git 'https://github.com/shazforiot/GOL.git'
4     }
5     stage('SonarQube analysis') {
6         withSonarQubeContainer("sonarQube lab") {
7             bat """
8                 C:\Users\UJ1000\Downloads\sonar-scanner-cll-6.2.0.4584-windows-x64\bIn\sonar-scanner.bat ^
9                 -Dsonar.login=deshi123
10                -Dsonar.password=Shejal123
11                -Dsonar.projectKey=sonarqube14
12                -Dsonar.sources="**/*.*"
13                -Dsonar.host.url=http://localhost:9000/
14                -Dsonar.java.libraries="resources/**/*.*"
15            """
16        }
17    }
18 }
```

Below the script, there is a checkbox for 'Use Groovy Sandbox' which is checked. 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.

The screenshot shows the SonarQube pipeline status page. The top navigation bar includes 'Dashboard > sonarqube14 >'. Below this, the 'Status' tab is selected. On the left, a sidebar lists several options: 'Changes' (indicated by a '</>' icon), 'Build Now' (indicated by a play button icon), 'Configure' (indicated by a gear icon), 'Delete Pipeline' (indicated by a trash bin icon), 'Full Stage View' (indicated by a magnifying glass icon), 'SonarQube' (indicated by a signal icon), 'Stages' (indicated by a stack icon), 'Rename' (indicated by a pencil icon), and 'Pipeline Syntax' (indicated by a question mark icon).

**Stage View**

Build	Date	Changes	Time	Status
#11	Sep 27 21:03	No Changes	42s	Success (25min 37s)
#10	Sep 27 21:00	No Changes	3s	Aborted
#9	Sep 27 20:54	No Changes	2s	Failed (1s)
#8	Sep 27 20:52	No Changes	1s	Failed (83ms)
#7	Sep 27 20:51	No Changes	3s	Success (204ms)

**Build History**

- #11 | Sep 27, 2024, 9:03 PM
- #10 | Sep 27, 2024, 9:00 PM
- #9 | Sep 27, 2024, 8:54 PM

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

```

Dashboard > sonarqube14 > #11
21:22:54.863 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 75. Keep
only the first 100 references.
21:22:54.863 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 41. Keep
only the first 100 references.
21:22:54.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 17. Keep
only the first 100 references.
21:22:54.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 185. Keep
only the first 100 references.
21:22:54.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 185. Keep
only the first 100 references.
21:22:54.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 550. Keep
only the first 100 references.
21:22:54.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/BSFListener.html for block at line 75. Keep
only the first 100 references.
21:22:54.864 INFO CPD Executor CPD calculation finished (done) | time=443358ms
21:22:55.087 INFO SCM revision ID: '9a7909a7e1b57f0fa4d03222b0412c5e6e1e4e'
21:27:08.067 INFO Analysis report generated in 5834ms, dir size=127.2 MB
21:27:38.158 INFO Analysis report compressed in 3804ms, zip size=29.6 MB
21:27:52.892 INFO Analysis report uploaded in 14680ms
21:27:52.947 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube14
21:27:52.947 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:27:52.947 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=d45ad9df-fd29-4b0d-bfc8-78c329fd0549
21:28:36.631 INFO Analysis total time: 25:19.835 s
21:28:36.702 INFO SonarScanner Engine completed successfully
21:28:37.637 INFO EXECUTION SUCCESS
21:28:38.003 INFO Total time: 25:29.478s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

10. After that, check the project in SonarQube.

The screenshot shows the SonarQube interface for the 'main' project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation is a breadcrumb trail showing the project name 'sonarqube14 / main'. The main content area is titled 'main' and displays a 'Passed' status for the Quality Gate. It includes several cards with metrics: Security (0 Open Issues), Reliability (68k Open issues, A grade), Maintainability (164k Open issues, A grade), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6% on 759k lines). The overall code status is 'Overall Code'.

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

## 11. Code Problems -

### Consistency

The screenshot shows the SonarQube Issues tab for the 'main' project. The left sidebar has a 'Filters' section with 'My Issues' and 'All' buttons, and a 'Clear All Filters' button. It also includes dropdowns for 'Issues in new code', 'Clean Code Attribute' (selected 'Consistency' with 197k), 'Software Quality', 'Severity', 'Type', and 'Scope'. The main panel lists four issues under the 'Consistency' category:

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability)
- Remove this deprecated "width" attribute. (Maintainability)
- Remove this deprecated "align" attribute. (Maintainability)
- Remove this deprecated "align" attribute. (Maintainability)

Each issue includes a severity level (e.g., Reliability, Maintainability) and a 'Code Smell' tag.

## Intentionality

The screenshot shows the SonarQube Issues page for the project 'sonarqube14'. The left sidebar shows a summary of 197k issues: Consistency (197k), Intentionality (14k), Adaptability (0), and Responsibility (0). The main panel displays several Intentionality code smells found in the file 'gameoflife-acceptance-tests/Dockerfile'.

- Intentionality:** Use a specific version tag for the image.
- Maintainability:** Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
- Intentionality:** Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
- Maintainability:** Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
- Intentionality:** Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Each item includes a checkbox, a 'Bulk Change' button, and a detailed description with severity (L1-L3), effort (e.g., 5min), and date (e.g., 4 years ago). The right sidebar shows 'Project Settings' and 'Project Information'.

## Bugs and Code Smells

The screenshot shows the SonarQube Issues page for the project 'sonarqube14'. The left sidebar shows a summary of 47k issues: Medium (47k) and Low (3). The main panel displays several bugs and code smells found in the file 'gameoflife-core/build/reports/tests/all-tests.html'.

- Bug:** Add "lang" and/or "xml:lang" attributes to this "<html>" element.
- Vulnerability:** Insert a <!DOCTYPE> declaration to before this <html> tag.
- Code Smell:** Add "<th>" headers to this "<table>".

Each item includes a checkbox, a 'Bulk Change' button, and a detailed description with severity (L1-L3), effort (e.g., 2min-5min), and date (e.g., 4 years ago). The right sidebar shows 'Project Settings' and 'Project Information'.

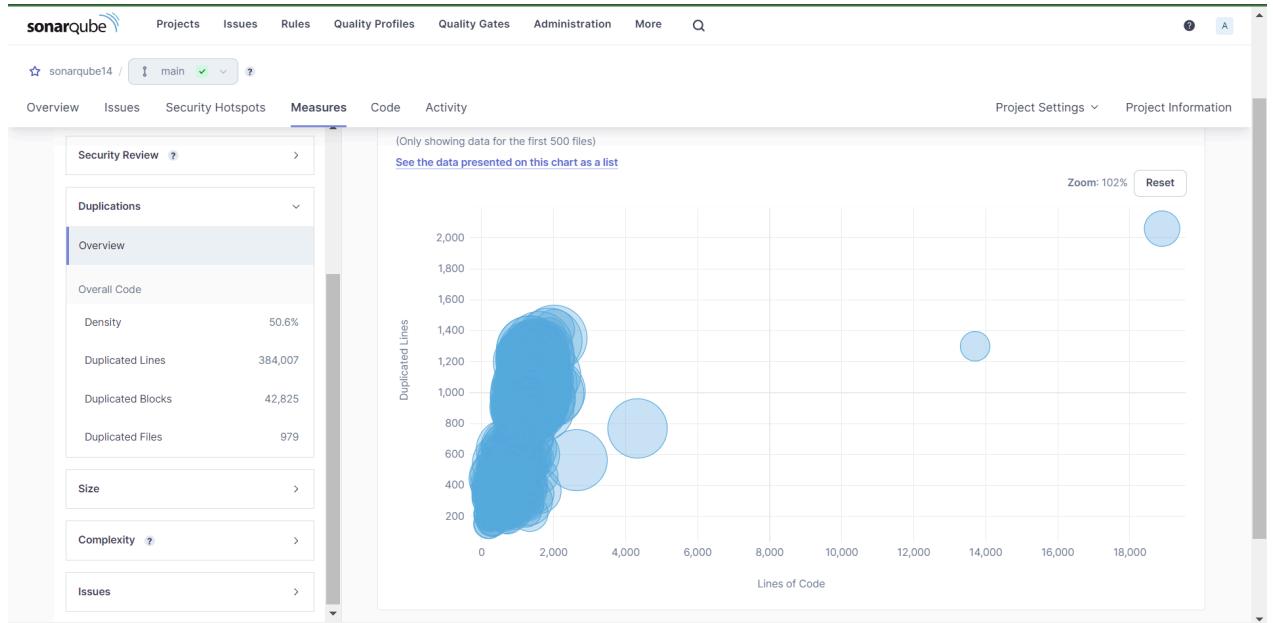
SonarQube Issues page for project sonarqube14/main. The left sidebar shows issues categorized by severity: Medium (143k), Low (21k), Type (Bug: 47k, Vulnerability: 0, Code Smell: 164k). The right panel displays two code smell findings for Dockerfile:

- gameoflife-acceptance-tests/Dockerfile**: Use a specific version tag for the image. (Intentionality: Maintainability) - Status: Open, Not assigned.
- gameoflife-acceptance-tests/Dockerfile**: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality: Maintainability) - Status: Open, Not assigned.

## Duplicates

SonarQube Measures page for project sonarqube14/main. The left sidebar shows measures: Security, Reliability, Maintainability, Security Review, and Duplications (Overview selected). The main panel displays a bubble chart titled "Duplications Overview" showing the relationship between Lines of Code (X-axis, 0 to 15,000) and Duplicated Lines (Y-axis, 0 to 2,000). The chart indicates that most duplicated lines are found in files with fewer than 5,000 lines of code. A tooltip states: "Size: Duplicated Blocks".

## Cyclomatic Complexities



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 cloned a GitHub repository and integrated it with SonarQube for comprehensive code analysis. SonarQube provided valuable insights into various types of program issues, such as:

Consistency: Detected non-adherence to coding standards and formatting rules.

Intentionality: Flagged potential logical or structural errors within the code.

Severity: Classified issues by their level of criticality (e.g., critical, major, minor).

Duplicates: Identified redundant code segments, suggesting potential optimization.

Cyclomatic Complexity: Measured the complexity of the code based on the number of control flow paths, highlighting sections that might be difficult to maintain or prone to errors.

**Issues Faced:**

**SonarQube Scanner Path Error:** Initially, Jenkins failed to detect the correct path for the SonarQube Scanner. This required manual intervention to adjust the pipeline script and specify the correct path for the scanner bash file, allowing the pipeline to execute successfully.

**SonarQube Login Issues:** Default credentials for logging into SonarQube (username: admin, password: admin) did not work due to a configuration reset or password change in a previous session. This necessitated troubleshooting and reconfiguration of access credentials.

**Slow Analysis Process:** During the scan of larger repositories, the SonarQube analysis process was notably slow.

## AdvanceDevops Experiment No: 9

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

### 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

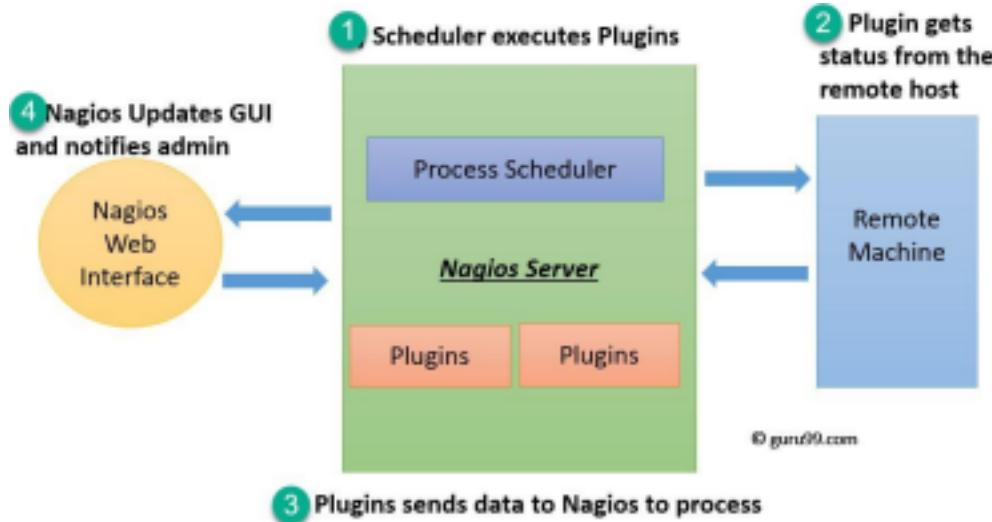
Following are the important features of Nagios monitoring tool:

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

Type	Protocol	Port range	Source	Description - optional
HTTPS	TCP	443	Anywhere-IPv4	
All traffic	All	All	Anywhere-IPv4	
HTTP	TCP	80	Anywhere-IPv4	
All ICMP - IPv6	IPv6 ICMP	All	Anywhere-IPv6	
SSH	TCP	22	Anywhere-IPv4	
All ICMP - IPv4	ICMP	All	Anywhere-IPv4	
Custom TCP	TCP	5666	Anywhere-IPv4	

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

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more  
ami-0ebfd941bbafe706

Virtual server type (instance type): t2.micro

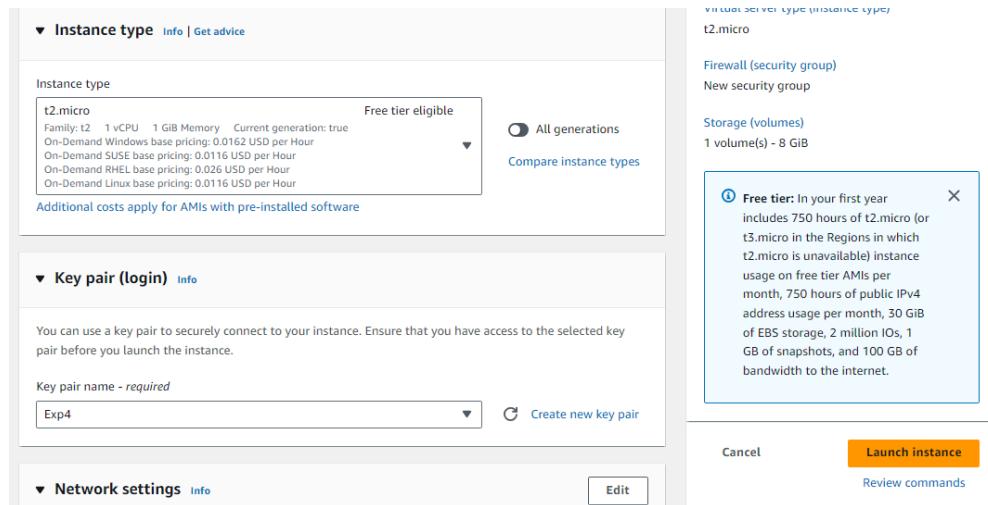
Firewall (security group): New security group

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

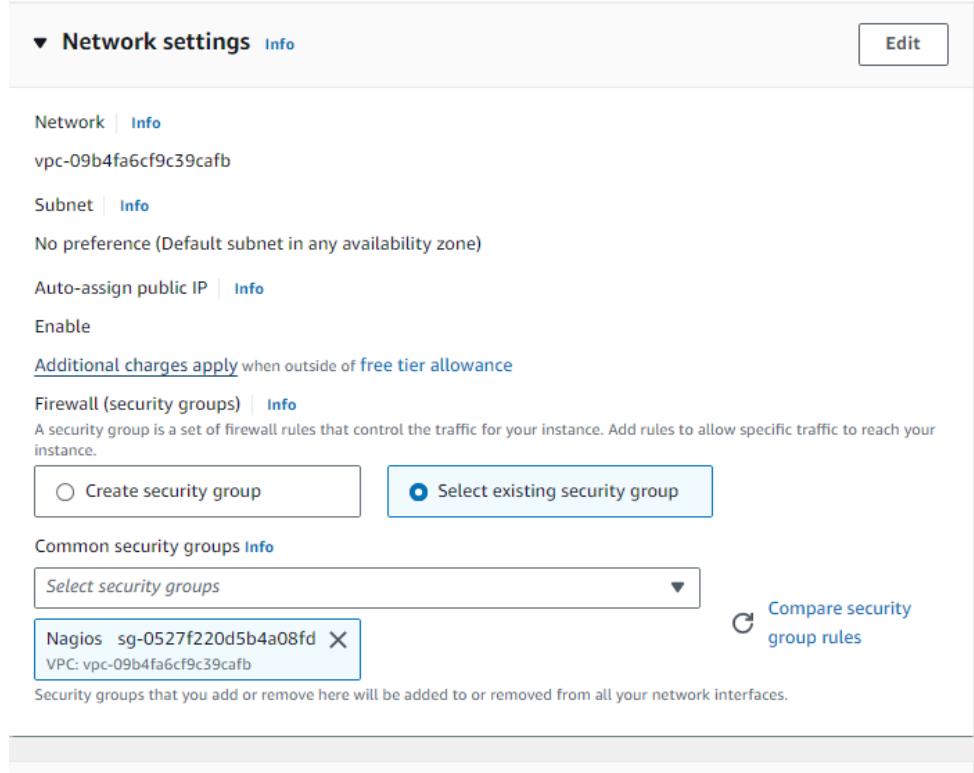
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOPS, 1 GB of snapshots, and 100 GB of bandwidth to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

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

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



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



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

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

Successfully connected to the instance.

```
C:\Users\LENOVO>ssh -i "Exp4.pem" ec2-user@ec2-54-210-4-52.compute-1.amazonaws.com
The authenticity of host 'ec2-54-210-4-52.compute-1.amazonaws.com (54.210.4.52)' can't be established.
ED25519 key fingerprint is SHA256:YqoGBku3mpyrgnwBwnKC060wMjlNnInQm5MPBeB1RM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-210-4-52.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
'_
~\_ ####_      Amazon Linux 2023
~~ \#####\
~~  \###|
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   \~' '--->
~~   / \
~~ .-.
~/-' /-
~/m/
[ec2-user@ip-172-31-38-62 ~]$ |
```

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

**sudo adduser -m nagios**

**sudo passwd nagios**

```
[ec2-user@ip-172-31-38-62 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-38-62 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-38-62 ~]$ |
```

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

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

```
[ec2-user@ip-172-31-38-62 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-38-62 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-38-62 ~]$ |
```

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

```
mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-38-62 ~]$ mkdir ~/downloads
cd ~/downloads
```

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

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

```
[ec2-user@ip-172-31-38-62 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-02 07:05:23-- https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 34.237.219.119, 3.92.120.28, 18.208.125.13, ...
Connecting to go.nagios.org (go.nagios.org)|34.237.219.119|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-02 07:05:23-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c0::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-02 07:05:23-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx          100%[=====] 1.97M 6.65MB/s   in 0.3s

2024-10-02 07:05:24 (6.65 MB/s) - '6kqcx' saved [2065473/2065473]

[ec2-user@ip-172-31-38-62 downloads]$ |
```

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

```
[ec2-user@ip-172-31-38-62 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 07:05:58-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.t 100%[=====] 2.62M 7.32MB/s in 0.4s
2024-10-02 07:05:59 (7.32 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-38-62 downloads]$ |
```

**Step 8:** Now to extract the files from the downloaded Nagios 4.5.5 run the following command.  
**tar zxvf 6kqcx**

```
[ec2-user@ip-172-31-38-62 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANX
```

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

```
[ec2-user@ip-172-31-38-62 downloads]$ cd nagios-4.5.5
```

**Step 10:** Now run the following command to configure.

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

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
```

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

```
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-172-31-38-62 nagios-4.5.5]$ |
```

So run following command to install ssl.

**sudo yum install openssl-devel**

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:10:57 ago on Wed Oct 2 06:57:40 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          21 MB/s | 3.0 MB  00:00
=====
Total                                         16 MB/s | 3.0 MB  00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :                                     1/1
  Installing         : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64          1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64          1/1
  Verifying          : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64          1/1

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

Complete!
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ |
```

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

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
```

#### web interface

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

#### \*\*\* Support Notes \*\*\*\*\*

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

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

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

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

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

\*\*\*\*\*

Enjoy.

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ |
```

**Step 11:** Now run the following commands to setup the Nagios.

**sudo make install**

```
[ec2-user@ip-172-31-38-62 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 nagcmd -d /usr/local/nagios/var/spool/checkresults
chmod g+s /usr/local/nagios/var/spool/checkresults
```

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

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

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

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

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

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ |
```

**sudo make install-init**

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/swtch.cfg /usr/local/nagios/etc/objects/swtch.cfg
*** Config files installed ***
```

**sudo make install-webconf**

```
[ec2-user@ip-172-31-38-62 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-172-31-38-62 nagios-4.5.5]$ |
```

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

```
[ec2-user@ip-172-31-38-62 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-172-31-38-62 nagios-4.5.5]$ |
```

Now to restart the httpd service run the following command.

**sudo service httpd restart**

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
-
```

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

**cd ~/downloads**

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

```
[ec2-user@ip-172-31-38-62 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-38-62 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
```

**Step 13:** Now change the directory to nagios-plugins-2.4.11 and run the config command to configure.  
**cd nagios-plugins-2.4.11**

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

```
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
-----
```

**Step 14:** Run the following commands to check nagios and start it.

**sudo chkconfig --add nagios**

```
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ |
```

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

```
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ |
```

**cd****sudo service nagios start**

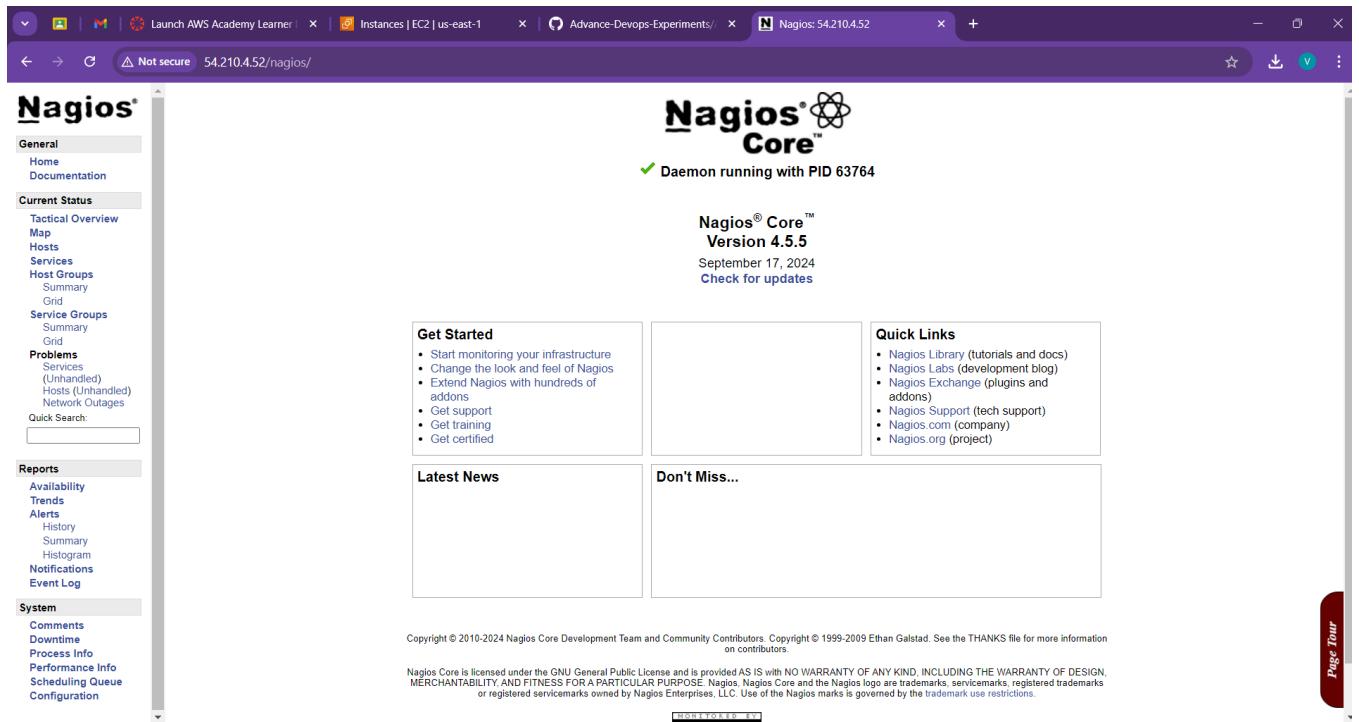
```
[ec2-user@ip-172-31-38-62 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-38-62 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-38-62 ~]$ |
```

**sudo systemctl status nagios**

```
[ec2-user@ip-172-31-38-62 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-10-02 07:41:50 UTC; 5s ago
     Docs: https://www.nagios.org/documentation
 Process: 63762 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=0)
 Process: 63763 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=0)
 Main PID: 63764 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.4M
    CPU: 74ms
   CGroup: /system.slice/nagios.service
           └─63764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─63765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─63766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─63767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─63768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─63769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: qh: core query handler registered
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: qh: echo service query handler registered
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: qh: help for the query handler registered
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: wproc: Successfully registered manager
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: wproc: Registry request: name=Core Work
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: wproc: Registry request: name=Core Work
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: wproc: Registry request: name=Core Work
Oct 02 07:41:50 ip-172-31-38-62.ec2.internal nagios[63764]: wproc: Registry request: name=Core Work
Oct 02 07:41:51 ip-172-31-38-62.ec2.internal nagios[63764]: Successfully launched command file work
Oct 02 07:41:51 ip-172-31-38-62.ec2.internal nagios[63764]: HOST ALERT: localhost;DOWN;SOFT;1;(No op
lines 1-28/28 (END)
```

**Step 15:** We can see we have successfully launched the Nagios now . Open <http://<instance public ip >/nagios/> here it is <http://54.210.4.52/nagios> we can see the running web page of nagios.



## Conclusion:

In this experiment, we successfully installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux machine within an AWS EC2 instance. The aim of continuously monitoring a remote system was achieved by integrating Nagios with the EC2 environment and allowing web access via the Nagios dashboard. We faced several challenges that required troubleshooting:

- **Security Group Configuration:** Setting up the correct inbound rules in the AWS security group was essential but prone to mistakes. Incorrectly configured ports could block HTTP and NRPE communication, preventing access to the Nagios dashboard or monitoring checks.
- **User and Group Permissions:** There were some issues when configuring user and group permissions, especially while adding users to the nagcmd group. If the commands weren't run correctly, Nagios failed to run properly due to incorrect access rights.
- **Dependencies and Package Installation:** While installing Nagios and its plugins, we encountered dependency issues, particularly with OpenSSL. Missing packages or libraries often halted the configuration process. Resolving these involved installing required dependencies and restarting the configuration steps.

## AdvanceDevOps Experiment No: 10

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

### Theory:

#### Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

#### Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

#### Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

#### Monitoring Using Nagios:

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

**sudo systemctl status nagios**

```
ec2-user@ip-172-31-46-196: ~ + - x
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
   Active: active (running) since Sun 2024-10-06 10:58:43 UTC; 4s ago
     Docs: https://www.nagios.org/documentation
 Process: 62217 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 62218 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 62219 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 5.4M
      CPU: 74ms
     CGroup: /system.slice/nagios.service
             └─62219 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
               ├─62220 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─62221 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─62222 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─62223 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─62224 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: qh: core query handler registered
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: qh: echo service query handler registered
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: qh: help for the query handler registered
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: wproc: Successfully registered manager as @wproc with query handler
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: wproc: Registry request: name=Core Worker 62223;pid=62223
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: wproc: Registry request: name=Core Worker 62221;pid=62221
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: wproc: Registry request: name=Core Worker 62222;pid=62222
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: wproc: Registry request: name=Core Worker 62220;pid=62220
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: Successfully launched command file worker with pid 62224
Oct 06 10:58:43 ip-172-31-46-196.ec2.internal nagios[62219]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/nagios/cgi-bin/nagios?&host=lo...
```

You can now proceed if you get the above message/output.

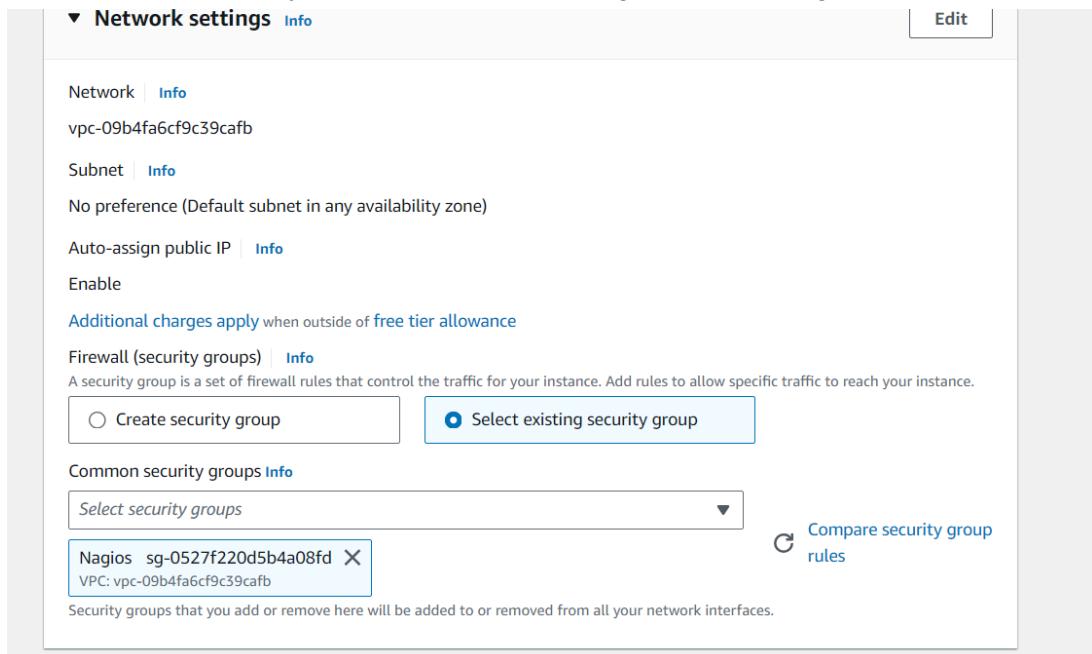
**Step 2:** Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the name 'Nagios-client' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Quick Start' tab is selected, showing options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar and a link to 'Browse more AMIs' are also present.

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

The screenshot shows the AWS EC2 'Create New Instance' wizard. In the 'Instance type' step, the 't2.micro' instance type is selected, which is free tier eligible. In the 'Key pair (login)' step, a key pair named 'devops' is selected. Other steps like 'Networking & Security' and 'Review: Launch instance' are partially visible at the bottom.

Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



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

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

```
PS C:\Users\Vedant> ssh -i "devops.pem" ubuntu@ec2-3-81-218-241.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-218-241.compute-1.amazonaws.com (3.81.218.241)' can't be established.
ED25519 key fingerprint is SHA256:7YtdUbwcFY6vK575h5D1fKqn10f220VC34blksm0Qcw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-218-241.compute-1.amazonaws.com' (ED25519) 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 Fri Sep 27 08:38:26 UTC 2024

System load: 1.36      Processes:          26
Usage of /home: unknown   Users logged in:    0
Memory usage: 4%        IPv4 address for eth0: 10.10.10.2
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

**Now perform all the commands on the Nagios-host till step 10****Step 4:** Now on the server Nagios-host run the following command.**ps -ef | grep nagios**

```
[ec2-user@ip-172-31-46-196 ~]$ ps -ef | grep nagios
nagios      2428      1  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagio
/etc/nagios.cfg
nagios      2430      2428  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local
nagios/rw/nagios.qh
nagios      2431      2428  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local
nagios/rw/nagios.qh
nagios      2432      2428  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local
nagios/rw/nagios.qh
nagios      2433      2428  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local
nagios/rw/nagios.qh
nagios      2437      2428  0 11:05 ?          00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagio
/etc/nagios.cfg
ec2-user     3367      3273  0 11:16 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-46-196 ~]$ |
```

**Step 5:** Now Become root user and create root directories.**sudo su****mkdir /usr/local/nagios/etc/objects/monitorhosts****mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts**

```
[ec2-user@ip-172-31-46-196 ~]$ sudo su
[root@ip-172-31-46-196 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-46-196 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-46-196 ec2-user]# |
```

**6:** Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)**cp /usr/local/nagios/etc/objects/localhost.cfg****/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg**

```
[root@ip-172-31-46-196 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-46-196 ec2-user]# |
```

**Step 7:** Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.Change **hostname** to **linuxserver**.Change **address** to the public IP of your Linux client.Set **hostgroup\_name** to **linux-servers1**.**nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg**

```

GNU nano 5.8      /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg      Modified

#####
#
# HOST DEFINITION
#
#####

# Define a host for the local machine

define host {

    use          linux-server ; Name of host template to use
                  ; This host definition will inherit all variables t>
                  ; in (or inherited by) the linux-server host templ>

    host_name    linux-server
    alias        localhost
    address     3.81.218.241
}

#####

#
# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines

define hostgroup {

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

```

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

**Step 8:** Now update the Nagios config file .Add the following line in the file.

**Line to add : cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/**

Run the command : **nano /usr/local/nagios/etc/nagios.cfg**

```

GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg                         Modified
#####
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####

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

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

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

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

# Definitions for monitoring a router/switch

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

```

**Step 9:** Now Verify the configuration files by running the following commands.

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

```
[root@ip-172-31-46-196 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 152)
Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 138)
Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 125)
Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 112)
Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 100)
Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 86)
Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 72)
Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios.cfg', starting on line 58)
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.

  Checked 0 host escalations.
  Checked 0 service escalations.

Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods

Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

[root@ip-172-31-46-196 ec2-user]#
```

**Step 10:** Now restart the services of nagios by running the following command.

**service nagios restart**

```
[root@ip-172-31-46-196 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-46-196 ec2-user]# |
```

**Step 11:** Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

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

```
ubuntu@ip-172-31-37-150:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
```

```
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up lib smbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-37-150:~\$ |

**Step 12:** Open nrpe.cfg file to make changes.Under allowed\_hosts, add your nagios host IP address.

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

```

GNU nano 1.2                               /etc/nagios/nrpe.cfg *

# NRPE USER
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

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

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,3.91.89.94| 

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

```

**Step 13:** Now restart the NRPE server by this command.

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

```

ubuntu@ip-172-31-37-150:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-37-150:~$ |

```

**Step 14:** Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

**sudo systemctl status nagios**

```
[root@ip-172-31-46-196 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
  Active: active (running) since Sun 2024-10-06 11:37:16 UTC; 9min ago
    Docs: https://www.nagios.org/documentation
 Process: 4481 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, st>
 Process: 4482 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, statu>
 Main PID: 4488 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.1M
    CPU: 108ms
   CGroup: /system.slice/nagios.service
           └─4488 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─4489 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─4490 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─4491 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─4492 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─4497 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 11:37:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;1;(No output on s>
Oct 06 11:38:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;2;(No output on s>
Oct 06 11:39:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;3;(No output on s>
Oct 06 11:40:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;4;(No output on s>
Oct 06 11:41:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;5;(No output on s>
Oct 06 11:42:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;6;(No output on s>
Oct 06 11:43:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;7;(No output on s>
Oct 06 11:44:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;8;(No output on s>
Oct 06 11:45:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;SOFT;9;(No output on s>
Oct 06 11:46:16 ip-172-31-46-196.ec2.internal nagios[4488]: HOST ALERT: linux-server;DOWN;HARD;10;(No output on s>
lines 1-28/28 (END)
```

**sudo systemctl status httpd****sudo systemctl start httpd****sudo systemctl enable httpd**

```
[root@ip-172-31-46-196 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
    Active: active (running) since Sun 2024-10-06 11:08:08 UTC; 42min ago
      Docs: man:httpd.service(8)
 Main PID: 2546 (httpd)
  Status: "Total requests: 48; Idle/Busy workers 100/0;Requests/sec: 0.0188; Bytes served/sec: 121 B/sec"
    Tasks: 230 (limit: 1112)
   Memory: 25.1M
     CPU: 1.834s
   CGroup: /system.slice/httpd.service
           ├─2546 /usr/sbin/httpd -DFOREGROUND
           ├─2548 /usr/sbin/httpd -DFOREGROUND
           ├─2554 /usr/sbin/httpd -DFOREGROUND
           ├─2555 /usr/sbin/httpd -DFOREGROUND
           ├─2556 /usr/sbin/httpd -DFOREGROUND
           └─2889 /usr/sbin/httpd -DFOREGROUND

Oct 06 11:08:07 ip-172-31-46-196.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 06 11:08:08 ip-172-31-46-196.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 06 11:08:08 ip-172-31-46-196.ec2.internal httpd[2546]: Server configured, listening on: port 80
[root@ip-172-31-46-196 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-46-196 ec2-user]# sudo systemctl enable httpd
[root@ip-172-31-46-196 ec2-user]# |
```

**Step 15:** Now to check Nagios dashboard go to <http://<Nagios-host ip>/nagios> .

**Now Click on Hosts from left side panel**

New Tab      Nagios: 3.91.89.94      +

Not secure 3.91.89.94/nagios/

# Nagios®

**General**

- Home
- Documentation

**Current Status**

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups

  - Summary
  - Grid

- Service Groups

  - Summary
  - Grid

- Problems

  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

- Quick Search:

Limit Results: 100 ▾

Host	Status	Last Check	Duration	Status Information
linux-server	UP	10-06-2024 12:01:16	0d 0h 4m 49s	PING OK - Packet loss = 0%, RTA = 1.00 ms
localhost	UP	10-06-2024 12:02:49	0d 0h 7m 1s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

**Current Network Status**

Last Updated: Sun Oct 6 12:06:05 UTC 2024  
 Updated every 90 seconds  
 Nagios® Core™ 4.5.5 - www.nagios.org  
 Logged in as nagiosadmin

**Host Status Totals**

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

**Service Status Totals**

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

**Host Status Details For All Host Groups**

We can see our linuxserver now click on it we can see the host information.

The screenshot shows the Nagios 4.5.5 interface for a host named "localhost" (linux-server). The host is currently "UP" (for 0d 0h 5m 50s). The status information shows PING OK - Packet loss = 0%, RTA = 0.72 ms. The performance data includes rta=0.721000ms;3000.000000;5000.000000;0.000000 and pl=0%;80;100;0. The host has been checked 1/10 (HARD state) since 10-06-2024 12:06:16. The check type is ACTIVE. The last state change was 10-06-2024 12:01:16. The host is not flapping. The last update was 10-06-2024 12:07:05 (0d 0h 0m 1s ago). Active checks are enabled, while passive checks, obsessing, notifications, event handler, and flap detection are also enabled. The host is a member of "No hostgroups". The IP address is 3.81.218.241. The host commands section lists various actions such as Locate host on map, Disable active checks of this host, Re-schedule the next check of this host, Submit passive check result for this host, Stop accepting passive checks for this host, Stop obsessing over this host, Disable notifications for this host, Send custom host notification, Schedule downtime for this host, Schedule downtime for all services on this host, Disable notifications for all services on this host, Enable notifications for all services on this host, Schedule a check of all services on this host, Disable checks of all services on this host, Enable checks of all services on this host, Disable event handler for this host, Disable flap detection for this host, and Clear flapping state for this host. A "Host Comments" section allows adding a new comment or deleting all comments.

## Current Network Status

The screenshot shows the Nagios web interface at [3.91.89.94/nagios/](http://3.91.89.94/nagios/). The dashboard includes:

- Current Network Status:** Last Updated: Sun Oct 6 12:09:17 UTC 2024. Updated every 90 seconds. Logged in as `nagiosadmin`.
- Host Status Totals:** Up: 2, Down: 0, Unreachable: 0, Pending: 0.
- Service Status Totals:** Ok: 6, Warning: 1, Unknown: 0, Critical: 1, Pending: 0.
- Service Status Details For All Hosts:** A table listing services for the host `localhost`. The table has columns: Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-06-2024 12:06:34	0d 0h 7m 43s	1/4	OK - load average: 0.00, 0.02, 0.00
	Current Users	OK	10-06-2024 12:07:12	0d 0h 7m 5s	1/4	USERS OK - 3 users currently logged in
	HTTP	WARNING	10-06-2024 12:07:49	0d 0h 6m 28s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	10-06-2024 12:08:27	0d 0h 5m 50s	1/4	PING OK - Packet loss = 0%, RTA = 0.02 ms
	Root Partition	OK	10-06-2024 12:09:04	0d 0h 10m 13s	1/4	DISK OK - free space: / 6122 MB (75.43% inode=98%).
	SSH	OK	10-06-2024 12:04:42	0d 0h 9m 35s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-06-2024 12:05:19	0d 0h 58m 58s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-06-2024 12:05:57	0d 0h 8m 20s	1/4	PROCS OK: 38 processes with STATE = RSZDT

## Conclusion:

In this experiment, we successfully implemented port, service, and Windows/Linux server monitoring using Nagios, but encountered a few challenges.

- **Configuration Issues:** Setting up monitoring hosts and editing files like `linuxserver.cfg` led to some errors in file paths and syntax, which required careful review.
- **NRPE Setup:** Configuring NRPE for remote monitoring was tricky due to firewall and permission issues, often causing connectivity problems between the Nagios host and clients.
- **Service Restarts:** Restarting Nagios and NRPE to apply changes didn't always work smoothly, with misconfigurations requiring troubleshooting.
- **Dashboard Access:** Accessing the Nagios dashboard was hindered by incorrect AWS security group rules, needing adjustments to allow proper HTTP and TCP traffic.

## **AdvanceDevOps Experiment No: 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**

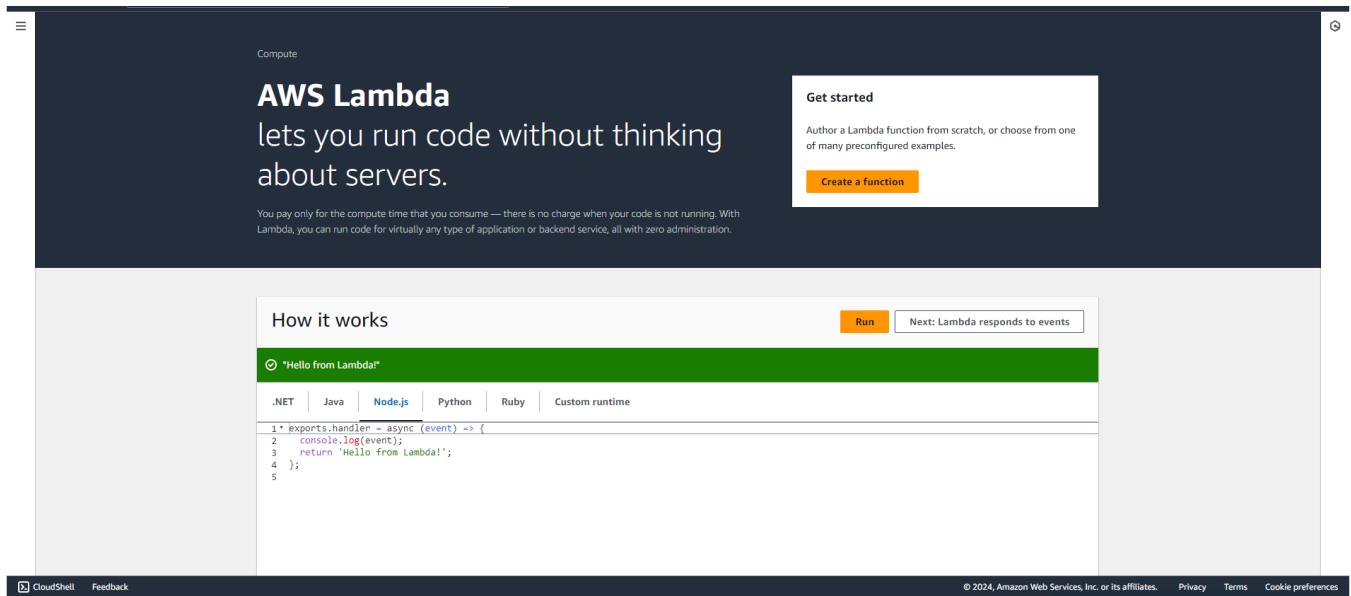
- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** 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.

**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.

The screenshot shows the AWS Lambda console interface. On the left, the navigation pane includes links for Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main content area displays a list of existing Lambda functions:

Function name	Description	Package type	Runtime	Last modified
<a href="#">ModLabRole</a>	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago
<a href="#">RedshiftOverwatch</a>	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago
<a href="#">RoleCreationFunction</a>	Create SLR if absent	Zip	Python 3.8	2 months ago

At the bottom, there's a search bar, a toolbar with various icons, and a status bar showing the date and time.

The second part of the screenshot shows the 'Create function' wizard. It starts with a choice between three options:

- 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.

Below this, the 'Basic information' section contains fields for:

- Function name: MY-lambda
- Runtime: Python 3.12
- Architecture: x86\_64

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

The screenshot shows the AWS Lambda 'Configuration' tab. On the left, a sidebar lists options: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main area displays 'General configuration' settings:

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

Below the table, there's an 'Advanced settings' section with a 'Create function' button at the bottom.

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.

This screenshot shows the 'Advanced settings' configuration page for a Lambda function. It includes fields for Memory, Ephemeral storage, SnapStart, Timeout, Execution role, and Existing role.

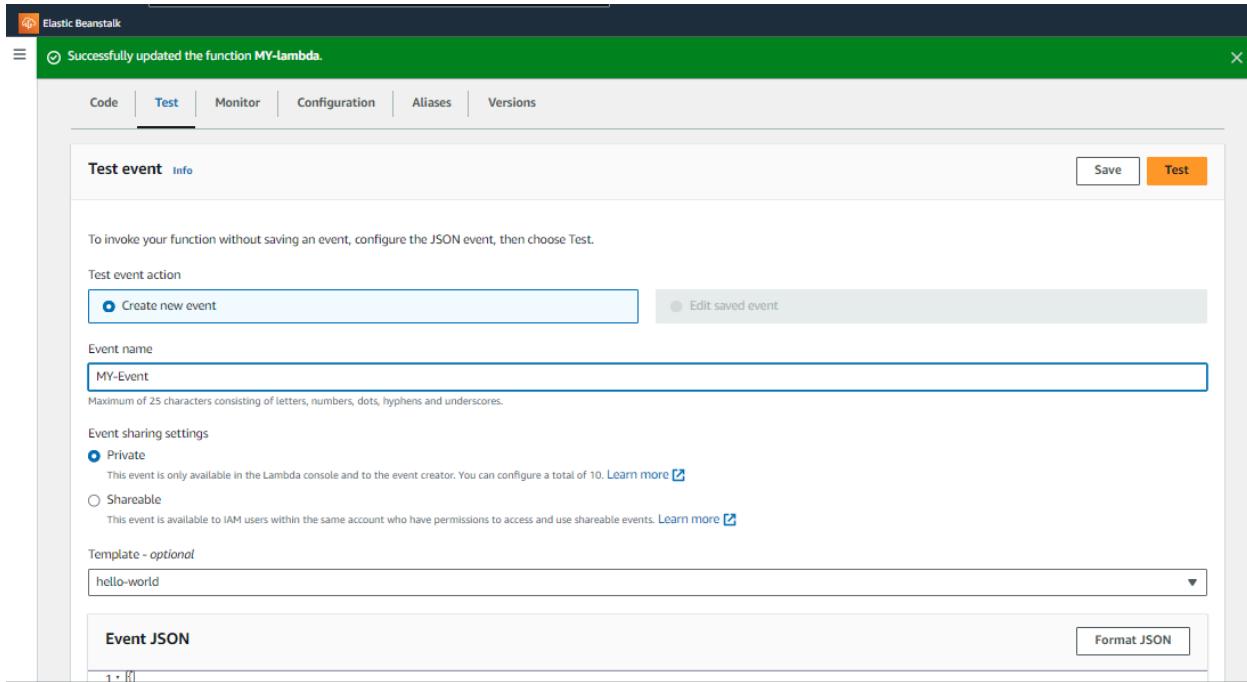
- Memory:** Set to 128 MB.
- Ephemeral storage:** Set to 512 MB.
- SnapStart:** Set to None.
- Timeout:** Set to 0 min 1 sec.
- Execution role:** Set to 'Use an existing role' with the role 'service-role/MY-lambda-role-en0dja4u' selected.
- Existing role:** A dropdown menu shows 'service-role/MY-lambda-role-en0dja4u'.

At the bottom are 'Cancel' and 'Save' buttons.

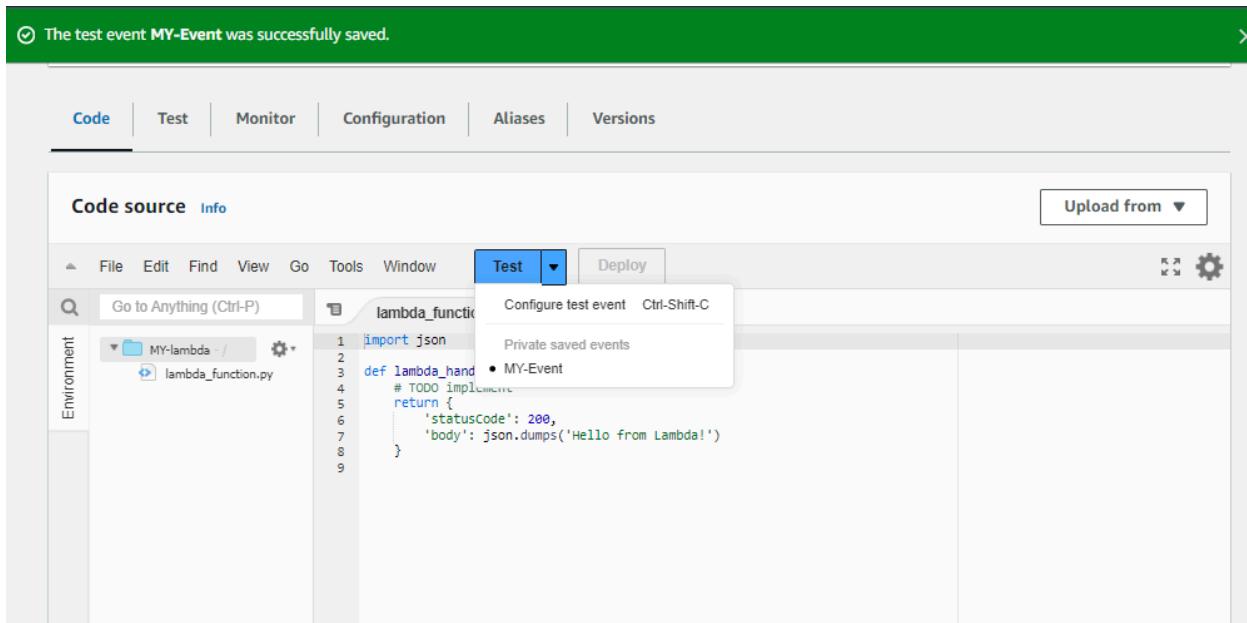
**Step 3:** Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.

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

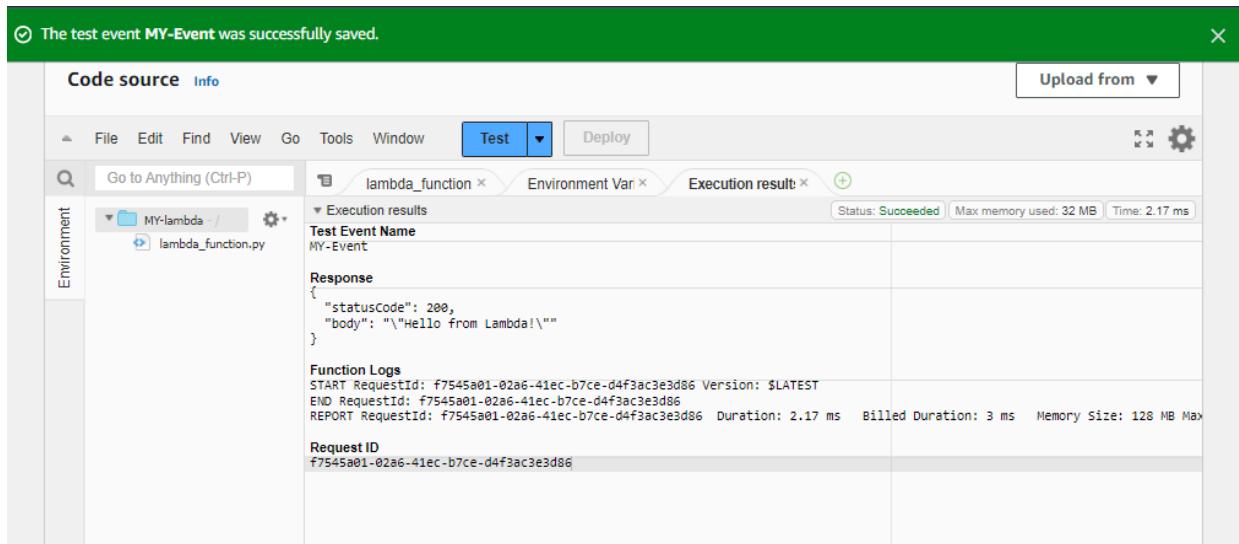
The screenshot shows the AWS Lambda function configuration interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, a search bar, and a Deploy button. A dropdown menu under the Test button shows 'Configure test event' and 'Ctrl-Shift-C'. A sub-menu lists 'Execution results', 'Test Event Name', and 'Bhushan\_Event', with 'Bhushan\_Event' highlighted in blue. The main area shows the code source 'lambda\_function.py' and its contents. On the left, there's an Environment sidebar with 'Bhushan\_Lambda' and 'lambda\_function.py'. Below the code editor, there's a 'Response' section showing the function's output. At the bottom, there's a detailed 'Function Logs' section with a table showing Request ID, Function Log, and Request ID again. A green status bar at the top of the interface says 'The test event Bhushan\_Event was successfully saved.'



**Step 5:** You can edit your lambda function code. I have changed the code to display the new String. Now ctrl+s to save and click on deploy to deploy the changes.



**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.



## Conclusion:

In this experiment, we successfully created and deployed our first AWS Lambda function using Python, gaining an understanding of its workflow and capabilities. The function was executed and tested, allowing us to observe the output and logs. While the overall process was smooth, there were several challenges that we encountered:

- **Role and Permissions Configuration:** Setting up the correct execution role with the necessary permissions was critical, as misconfigurations could prevent the function from running or accessing other AWS services. Debugging permission issues was time-consuming, especially when using new roles.
- **Timeout Issues:** Initially, the default timeout setting was insufficient for some operations. Adjusting the timeout to a value that suits the function's workload was necessary, especially for more complex operations beyond basic tasks.
- **Event Testing:** Configuring and testing events within Lambda required careful attention. Choosing the wrong template or incorrect event parameters often led to failures during the test phase, requiring adjustments to the event settings.

## **AdvanceDevOps Experiment No: 12**

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

### **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.

**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.

**Step 2:** I have used already created bucket v2bucket if you dont have then you can create a basic bucket for this experiment .

**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.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The path in the top navigation bar is 'Lambda > Functions > Create function'. The main section is titled 'Create function' with an 'Info' link. It asks to choose one of three options: 'Author from scratch' (selected), 'Use a blueprint', or 'Container image'. Below this is a 'Basic information' section with fields for 'Function name' (set to 'MY-lambda'), 'Runtime' (set to 'Python 3.12'), and 'Architecture' (set to 'x86\_64').

The screenshot shows the AWS Lambda console. At the top, a green success message says: "Successfully created the function MY-lambda. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the "Function overview" section displays the function name "MY-lambda". It includes tabs for "Diagram" (selected) and "Template". The diagram shows a single function icon labeled "MY-lambda" with "(0)" layers. Buttons for "+ Add trigger" and "+ Add destination" are present. On the right, there's a "Description" field, "Last modified" (3 seconds ago), "Function ARN" (arn:aws:lambda:us-east-1:022499016110:function:MY-lambda), and a "Function URL" link. A sidebar titled "Create a simple web app" provides a tutorial on how to build a simple web application using Lambda.

The screenshot shows the "Code source" tab of the Lambda function configuration. The interface includes tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions". The "Code" tab is selected. The "Code source" section shows the "lambda\_function" file with the following Python code:

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
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), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area displays the General configuration settings:

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

An "Edit" button is located in the top right corner of the configuration table.

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

The screenshot shows the AWS Lambda Edit basic settings page. The top navigation bar includes the AWS logo, Services, a search bar, and a [Alt+S] keyboard shortcut. The breadcrumb navigation shows Lambda > Functions > Bhushan\_Lambda > Edit basic settings. The main content area is titled "Edit basic settings" and contains the "Basic settings" section:

Basic settings	
Description - optional	<input type="text" value="Basic Settings"/>
Memory	<input type="text" value="128"/> MB
Your function is allocated CPU proportional to the memory configured. Set memory to between 128 MB and 10240 MB.	
Ephemeral storage	<input type="text" value="512"/> MB
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. <a href="#">View pricing</a>	
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.	
SnapStart	<input type="button" value="None"/>
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 <a href="#">SnapStart compatibility considerations</a> .	
Timeout	<input type="text" value="0"/> min <input type="text" value="1"/> sec
Supported runtimes: Java 11, Java 17, Java 21.	
Execution role	Choose a role that defines the permissions of your function. To create a custom role, go to the <a href="#">IAM console</a> .
<input checked="" type="radio"/> Use an existing role <input type="radio"/> Create a new role from AWS policy templates	

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

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

Test event action

Create new event     Edit saved event

Event name

event-exp\_12

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

```
s3-put
```

Template - optional

s3-put

Event 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": "EXAMPLE123/5678abcdefhijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
18 *     },
19 *     "s3": {
20 *       "s3SchemaVersion": "1.0",
21 *       "configurationId": "testConfigRule",
22 *       "bucket": "example-bucket",
23 *       "name": "example-key",
24 *       "ownerIdentity": {
25 *         "principalId": "EXAMPLE"
26 *       },
27 *       "arn": "arn:aws:s3:::example-bucket"
28 *     },
29 *     "object": {
30 *       "key": "testS3Key",
31 *       "size": 1024,
32 *     }
33 *
```

Format JSON

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

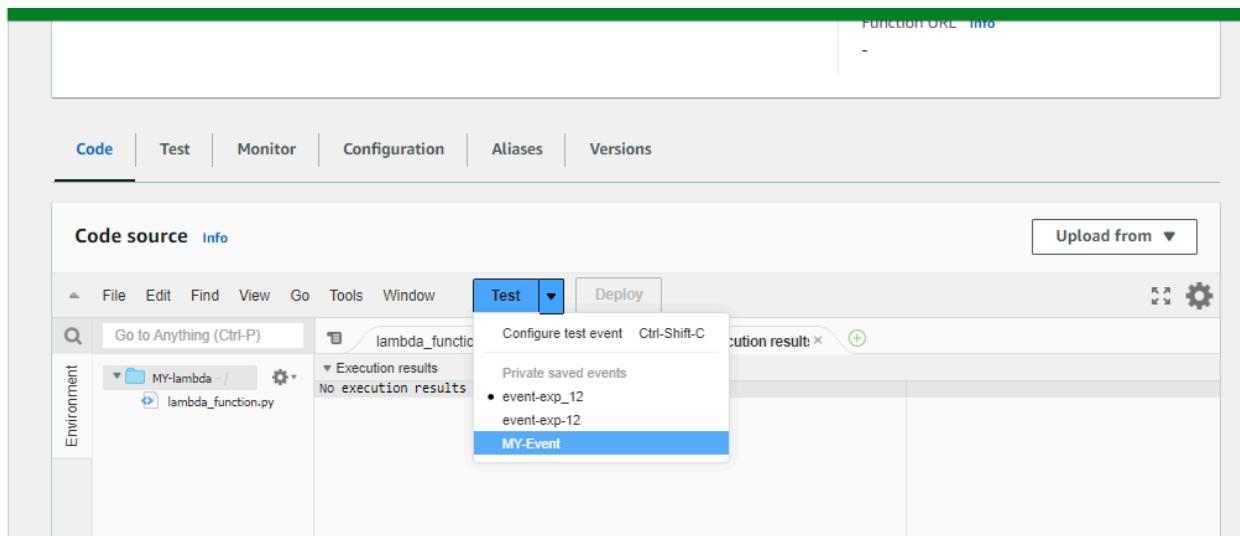
In this tutorial you will learn how to:

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

[Learn more](#)

[Start tutorial](#)

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



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

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

The screenshot shows the AWS Lambda Function Overview page for a function named "MY-lambda". The "Diagram" tab is selected, displaying a single function icon. Below the diagram are two buttons: "+ Add trigger" and "+ Add destination". To the right of the function icon, there is a "Description" section with a link to "Info". Further down, there are sections for "Last modified" (23 minutes ago), "Function ARN" (arn:aws:lambda:us-east-1:022499016110:function:MY-lambda), and "Function URL" (Info). At the bottom of the page, there are tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions".

The screenshot shows the "Add trigger" configuration page for an S3 event source. The "Trigger configuration" section is visible, showing the selected "S3" source. Below this, the "Bucket" section allows selecting an S3 bucket, with "s3/v2buckets" listed in the search bar. The "Event types" section lists "All object create events". At the bottom, there is a note about "Prefix - optional" and a warning about special characters.

The screenshot shows the configuration dialog for adding a trigger to a Lambda function. The 'Trigger type' is set to 'Amazon S3'. The 'Event name' is 'All object create events'. The 'Prefix - optional' field contains '23images'. The 'Suffix - optional' field contains 'e.g. jpg'. A note about recursive invocation is present, along with an acknowledgment checkbox. A note about Lambda permissions is also shown.

**Trigger configuration:**

- Trigger type: Amazon S3
- Event name: All object create events
- Prefix - optional: 23images
- Suffix - optional: e.g. jpg
- Recursive invocation note: If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)
- Acknowledgment checkbox: I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.
- Permissions note: Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

**Buttons:** Cancel, Add

**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 configuration page for the 'MY-lambda' function. It lists the function ARN and URL. On the left, there's a sidebar with tabs: General configuration, Triggers (selected), Permissions, Destinations, Function URL, and Environment. The main area shows the 'Triggers (1)' section with a table containing one row for an S3 trigger named 'v2buckets'.

Triggers (1) <a href="#">Info</a>	
<input type="checkbox"/>	Trigger S3: v2buckets arn:aws:s3:::v2buckets

**Function Details:**

- Last modified: 28 minutes ago
- Function ARN: arn:aws:lambda:us-east-1:022499016110:function:MY-lambda
- Function URL: [Info](#)

**Navigation:** Code, Test, Monitor, Configuration (selected), Aliases, Versions

**Step 9:** Now upload any image to the bucket.

The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > v2buckets > Upload'. Below that is a large 'Upload' button with an 'Info' link. A note below the button 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.' A dashed box area is labeled 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a table titled 'Files and folders (1 Total, 155.2 KB)'. It contains one item: 'Screenshot (1).png' (image/png). There are 'Remove', 'Add files', and 'Add folder' buttons at the top of the table. A search bar and pagination controls are also present. The 'Destination' section at the bottom shows the target bucket and path.

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

The screenshot shows the AWS Lambda function editor. The top navigation bar includes tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected. Below the tabs is a 'Code source' section with an 'Info' link. A toolbar above the code editor includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is currently selected), 'Deploy', and 'Changes not deployed'. The code editor displays a Python script named 'lambda\_function.py' under the 'MY-lambda' environment. The script content is as follows:

```
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} : {objecy_key}")
9     return {
10         'statusCode': 200,
11         'body': json.dumps('Log entry created successfully')
12     }
13
```

The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, and a 'Test' dropdown. The 'Test' dropdown is currently active. To the right of the menu is a 'Deploy' button and a gear icon. The main area has sections for 'Execution results', 'Test Event Name' (event-exp\_12), 'Response' (JSON object with statusCode: 200 and body: "Log entry created successfully"), 'Function Logs' (log output showing START, END, and REPORT events), and 'Request ID' (ef4bf0a2-8819-4f31-bfce-00f230170210). A status bar at the bottom indicates Success, 32 MB memory used, and 1.58 ms duration.

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

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar has a 'CloudWatch' header and sections for Favorites and recents, Dashboards, Alarms, Logs (selected), Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics, X-Ray traces, Events, and Application Signals. The main area shows a log group path: CloudWatch > Log groups > /aws/lambda/MY-lambda > 2024/10/05/[\$.LATEST]f42c0b3426b9499ea3d2b422849c0be. It includes a 'Log events' section with a search bar, timestamp filter (1m, 1h, UTC timezone), and display options. The log entries table lists several log entries with columns for Timestamp and Message. Some messages include error details like 'INIT\_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1:123456789012:function:MY-lambda' and '[ERROR] KeyError: 'Records' Traceback (most recent call last): File "/var/task/lambda\_function.py", line 10, in lambda\_handler event['Records'][0]

## Conclusion:

In this experiment, we successfully created a Lambda function designed to log the message “An Image has been added” upon the addition of an object to a specified S3 bucket. This process deepened our understanding of integrating AWS Lambda with S3 for event-driven applications. However, we encountered several challenges that provided valuable learning experiences:

- **S3 Bucket Configuration:** Setting up the S3 bucket correctly was crucial. We had to ensure that the bucket's public access settings were appropriately configured to allow the Lambda function to trigger without compromising security. Misconfigurations here could have led to access issues.
- **Trigger Setup:** Configuring the S3 bucket as a trigger for the Lambda function required careful attention. Selecting the correct bucket and event type was essential to ensure that the function executed as intended. Any oversight could result in the function not being triggered.
- **Testing and Monitoring:** After uploading images to the S3 bucket, verifying that the Lambda function logged the appropriate messages required us to navigate CloudWatch logs. Initially, we struggled to find the correct log group, which delayed our testing process.