

Experiment 1a

1)Static Hosting using XAMPP

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	19328 16548	80, 443	Stop	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat	8196	8080	Stop	Admin	Config	Logs

Logs

```
00:32:18 [mysql] Press the Logs button to view error logs and check
00:32:18 [mysql] the Windows Event Viewer for more clues
00:32:18 [mysql] If you need more help, copy and post this
00:32:18 [mysql] entire log window on the forums
00:32:40 [Apache] Attempting to stop Apache (PID: 2636)
00:32:40 [Apache] Attempting to stop Apache (PID: 13212)
00:32:41 [Apache] Status change detected: stopped
00:32:42 [Apache] Attempting to start Apache app...
00:32:42 [Apache] Status change detected: running
```

Select Category:

Music

Dance

Sports

About Category Contact Login Sign In

2)Static Hosting using Amazon S3

Static website hosting

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

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint

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

<http://bucket-aws34.s3-website-us-east-1.amazonaws.com>

[About](#) [Category](#) [Contact](#)

[Login](#) [Sign in](#)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
< >

Select Category:

here lastest news will come

[Category](#)

Music

Book

Book

Book

1)Elastic Beanstalk Environment

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with options like Applications, Environments, Change history, Application: my_web_34 (which is expanded to show Application versions and Saved configurations), and Recent environments (listing Myweb34-env). The main area displays the 'Application my_web_34 environments (1) Info' page. It shows one environment named 'Myweb34-env' with a yellow warning icon. The environment was created on August 11, 2024, and is running on PHP 8.3. The platform is listed as 'code-pipeline-1...'. There are buttons for Actions, Create new environment, and Support.

2)CodePipeline

The screenshot shows the AWS CodePipeline console. The pipeline is named 'my_pipeline' and is of type V2. The execution mode is 'QUEUED'. The pipeline has two stages: 'Source' and 'Deploy'. The 'Source' stage is shown as 'Succeeded' with a green checkmark. It has a transition to the 'Deploy' stage, which is currently 'In progress' with a blue circle. The pipeline execution ID is 'c3ed4dda-e394-4d65-bf80-5f5779048e02'. The pipeline has sections for Getting started, Pipelines, Pipeline, History, Settings, and Settings. There are also links for Go to resource and Feedback.

Website Before Change

Congratulations!

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.

For next steps, read the AWS CodePipeline Documentation.

Website After Change

Congratulations!

Hello, This is my_web_34

For next steps, read the AWS CodePipeline Documentation.

1)EC2 Instance in AWS

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there's a 'Details' section with the following information:

Name: Ayush Maurya	Owner ARN: arn:aws:sts::217253764927:assumed-role/voclabs/user3385518=MAURYA_AYUSH_SUBHASHCHANDRA	Status: Ready
Description: First Experence	Number of members: 1	Lifecycle status: Created
Environment type: EC2 instance		

Below the details, there are tabs for 'EC2 instance', 'Network settings', and 'Tags'. The 'EC2 instance' tab is selected. Under 'EC2 instance', the following details are shown:

ARN: arn:aws:cloud9:us-east-1:217253764927:environment:b096c95b7bb94be4bb2220da1765f6f6	Instance type: t2.micro (1 GiB RAM + 1 vCPU)
Platform: Amazon Linux 2023	Storage: EBS only

At the bottom right of the 'EC2 instance' section is a 'Manage EC2 instance' button.

2)IAM Services

The screenshot shows the AWS Cloud9 environments page. It displays a single environment named 'Ayush Maurya'.

Environment details:

Name: Ayush Maurya	Cloud9 IDE: Open	Environment type: EC2 instance	Connection: Secure Shell (SSH)	Permission: Owner	Owner ARN: arn:aws:sts::217253764927:assumed-role/voclabs/user3385518=MAURYA_AYUSH_SUBHASHCHANDRA
--------------------	------------------	--------------------------------	--------------------------------	-------------------	---

At the top of the page, there's a note: "For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#)".

User Group

The screenshot shows the AWS IAM User Groups creation wizard. Step 2, 'Set permissions', is active. It includes options for 'Add user to group', 'Copy permissions', and 'Attach policies directly'. Step 3, 'Review and create', is shown below.

Permissions options:

- Add user to group
Add user to an existing group, or create a new one. 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 (1):

Group name	Users	Attached policies	Created
ayushGroup	0	-	2024-08-11

Set permissions boundary - optional

At the bottom, there are 'Cancel', 'Previous', and 'Next' buttons.

User

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name: ayush

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](#).

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**

3)Cloud 9 and Cloud 9 IDE

AWS Cloud9

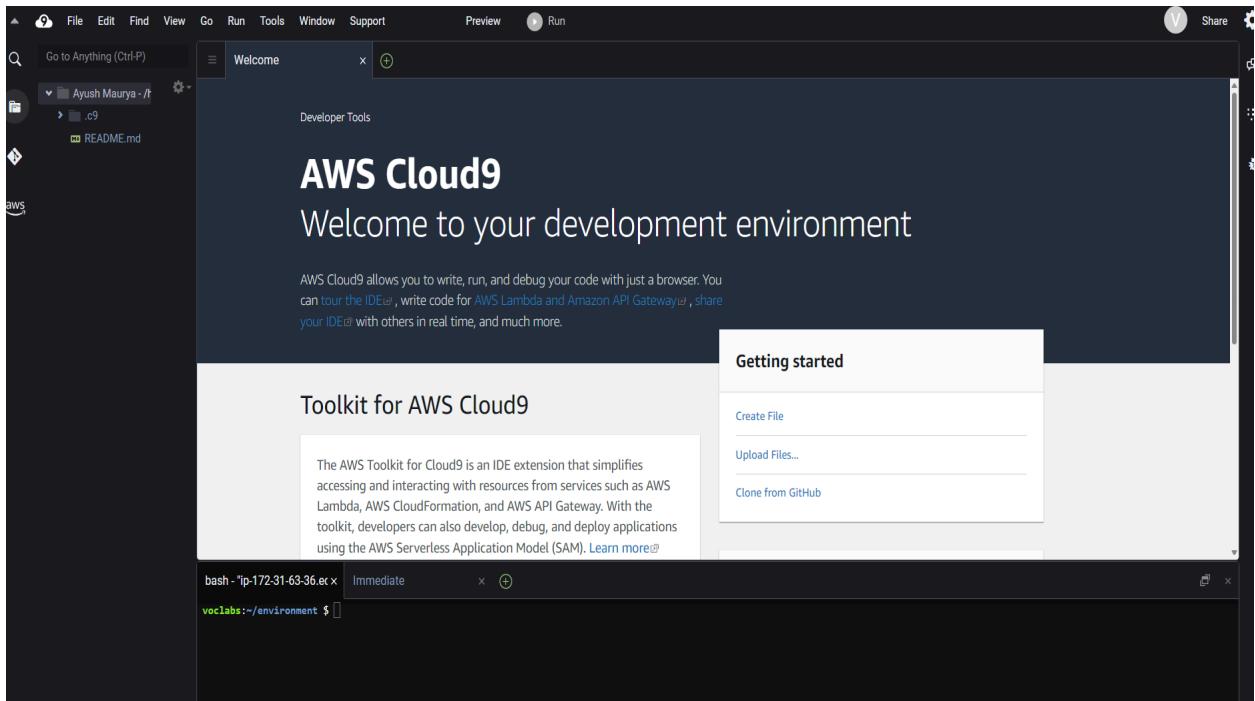
My environments
Shared with me
All account environments

Documentation [View](#)

AWS Cloud9 > Environments

Environments (1)

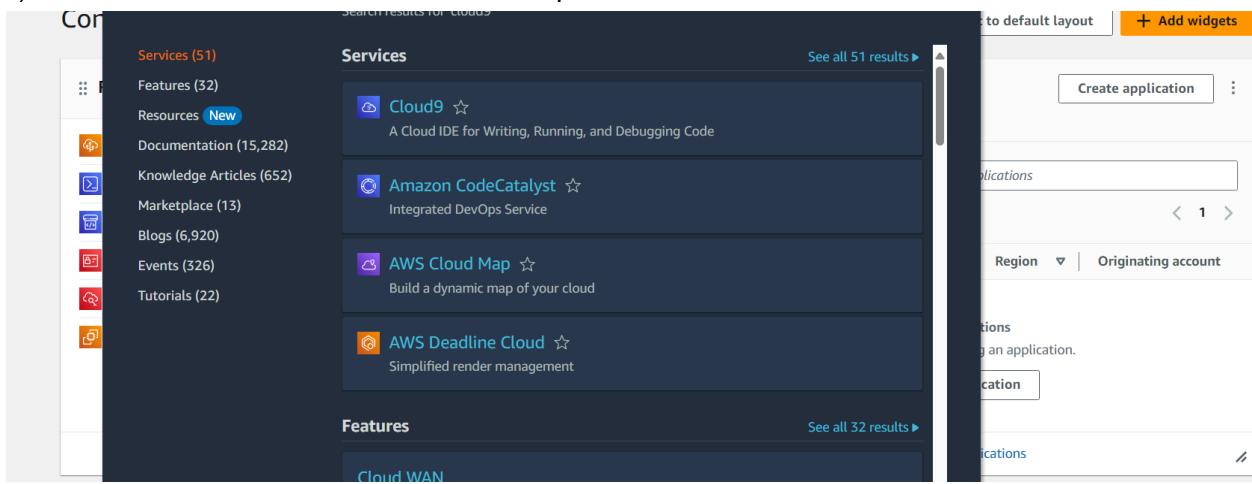
Name	Cloud9 IDE View	Environment type	Connection	Permission	Owner ARN
Ayush Maurya	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::217253764927:assumed-role/vclabs/user3385518=MAURYA_AYUSH_SUBHASHCHANDRA



Experiment 1b

Step 1: Set up Cloud9 environment.

1) Search Cloud9 in the services tab and open it



2) Click on Create Environment.

→ Give a name to your Cloud9 Environment. You can add a description if needed.

→ Select the option new EC2 instance if you do not have one ready for the environment. Give the specifications of that EC2 instance ahead

→ On the AWS Academy account, if we select AWS System Manager(SSM) in Network settings, it gives an error as the account does not have permissions to use the setting. So we select Secure Shell (SSH). After that click on Create.

A screenshot of the "Create environment" form. At the top, there is a breadcrumb trail: AWS Cloud9 > Environments > Create environment. The form has a header "Create environment" with a "Info" link. Below the header is a "Details" section. In the "Name" field, the value "Ayush Maurya" is entered. A note below the field states: "Limit of 60 characters, alphanumeric, and unique per user." In the "Description - optional" field, the value "First Experience" is entered. A note below the field states: "Limit 200 characters." Below these fields is a section titled "Environment type" with an "Info" link. It contains two options: "New EC2 instance" (selected, indicated by a blue border) and "Existing compute". The "New EC2 instance" description states: "Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation." The "Existing compute" description states: "You have an existing instance or server that you'd like to use."

Details

Name Ayush Maurya	Owner ARN <code>arn:aws:sts::217253764927:assumed-role/voclabs/user3385518=MAURYA_AYUSH_SUBHAS_HCHANDRA</code>	Status Ready
Description First Experience	Number of members 1	Lifecycle status Created
Environment type EC2 instance		

EC2 instance | Network settings | Tags

EC2 instance

ARN <code>arn:aws:cloud9:us-east-1:217253764927:environment:b096c95b7bb94be4bb2220da1765f6f6</code>	Instance type t2.micro (1 GiB RAM + 1 vCPU)
Platform Amazon Linux 2023	Storage EBS only

Manage EC2 instance

6) The environment is being created.

ⓘ For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#)

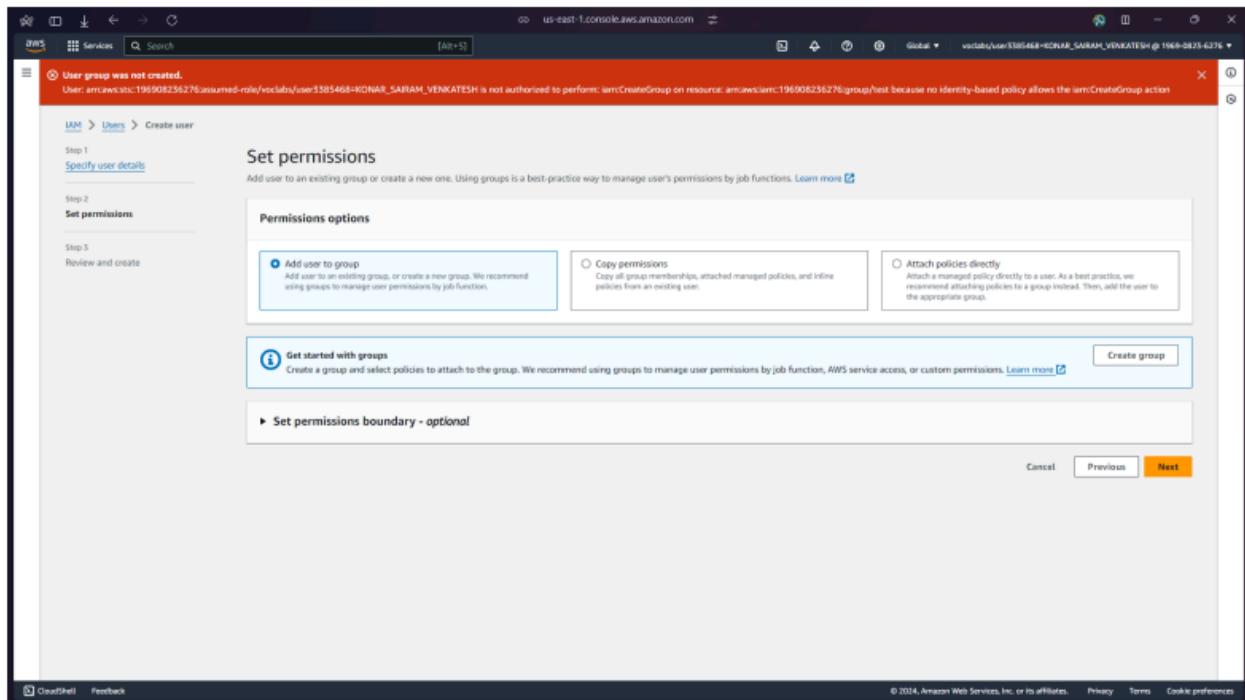
AWS Cloud9 > Environments

Environments (1)

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Ayush Maurya	Open	EC2 instance	Secure Shell (SSH)	Owner	<code>arn:aws:sts::217253764927:assumed-role/voclabs/user3385518=MAURYA_AYUSH_SUBHASHCHANDRA</code>

Step 2: Creating IAM user.

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



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

A screenshot of the AWS IAM 'Users' list page. The left sidebar shows 'Identity and Access Management (IAM)' with 'Users' selected. The main area shows a table with one row: 'User name' (empty). A note says 'No resources to display'. The top right has 'Create user' and other buttons.

2) Give a username to your user and click Next.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

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.

ⓘ 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**

3) Select add User to Group. If there are no user groups on your accounts, you will have to create one. Click on Create Group.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

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.

ⓘ 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**

4) Give a name to your user group. Then click on Create User Group.

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

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

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

Permissions policies (955)

Policy name	Type	Use...	Description
AdministratorAccess	AWS managed ...	None	Provides full access to AWS services
AdministratorAcce...	AWS managed	None	Grants account administrative perm
AdministratorAcce...	AWS managed	None	Grants account administrative perm
AlexaForBusinessD...	AWS managed	None	Provide device setup access to Alex
AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaForBusin
AlexaForBusinessG...	AWS managed	None	Provide gateway execution access t

Create user group

5) The group is created and shown under the groups area, select the group by clicking on the checkbox. Then click Next.

ayushGroup user group created.

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](#)

Step 2
Set permissions

Step 3
Review and create

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 (1)

Group name	Users	Attached policies	Created
ayushGroup	0	-	2024-08-11 ()

Set permissions boundary - optional

Next

6) Review all the Information, then click on Create user.

The screenshot shows the 'Review and create' step for creating a user. At the top, a green banner displays the message 'ayushGroup user group created.' Below this, the 'User details' section shows the user name 'ayush', console password type 'None', and a note that 'Require password reset' is set to 'No'. The 'Permissions summary' section indicates 'No resources'. The 'Tags - optional' section shows 'No tags associated with the resource' and a button to 'Add new tag'. A note states 'You can add up to 50 more tags.'

7) Open User Groups tab from the left side option. Click on the name of your group.

The screenshot shows the 'User groups' page. It lists one user group named 'ayushGroup'. The group has 0 users, 'Not defined' permissions, and was created 2 minutes ago. The page includes a search bar and buttons for 'Delete' and 'Create group'.

8) Go to permissions and click on Add permissions. Click on Attach Policies.

The screenshot shows the 'ayushGroup' permissions page. The 'Permissions' tab is selected. It shows 0 managed policies attached. There is a button to 'Add permissions'. The 'Summary' section provides basic information about the group, including its name, creation time (August 11, 2024, 18:14 (UTC+05:30)), and ARN (arn:aws:iam::011528263337:group/ayushGroup).

9) Search for AWSCloud9EnvironmentMember, select it and click on Attach policies

The screenshot shows the 'Attach permission policies to ayushGroup' dialog. At the top, there's a header 'Attach permission policies to ayushGroup'. Below it, a section titled 'Current permissions policies (0)' is shown. Underneath, a heading 'Other permission policies (1/953)' indicates one policy is available to attach. A search bar contains 'AWSCloud9' and a filter set to 'All types'. A table lists policies: 'AWSCloud9Administrator' (selected), 'AWSCloud9EnvironmentMember' (selected), 'AWSCloud9SSMInstanceProfile', and 'AWSCloud9User'. The 'AWSCloud9EnvironmentMember' row is highlighted with a blue border. At the bottom right are 'Cancel' and 'Attach policies' buttons.

10) The policies have been attached

The screenshot shows the 'ayushGroup' user group details page. The left sidebar includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups' selected), 'Identity providers', 'Account settings', 'Access reports', 'Credential report', and 'Organization activity'. The main panel shows the 'ayushGroup' summary: User group name 'ayushGroup', Creation time 'August 11, 2024, 18:14 (UTC+05:30)', and ARN 'arn:awsiam:011528263337:group/ayushGroup'. Below this, the 'Permissions' tab is selected, showing 'Permissions policies (1)'. A table lists the attached policy: 'AWSCloud9EnvironmentMember' (selected). At the bottom right are 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add permissions' buttons.

Step 3: Working on Cloud9 IDE

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

The screenshot shows the AWS Cloud9 interface. On the left, there's a sidebar with links for 'My environments', 'Shared with me', and 'All account environments'. Below that is a 'Documentation' link. The main area is titled 'Environments (1)' and shows a table with one row. The row contains the following information:

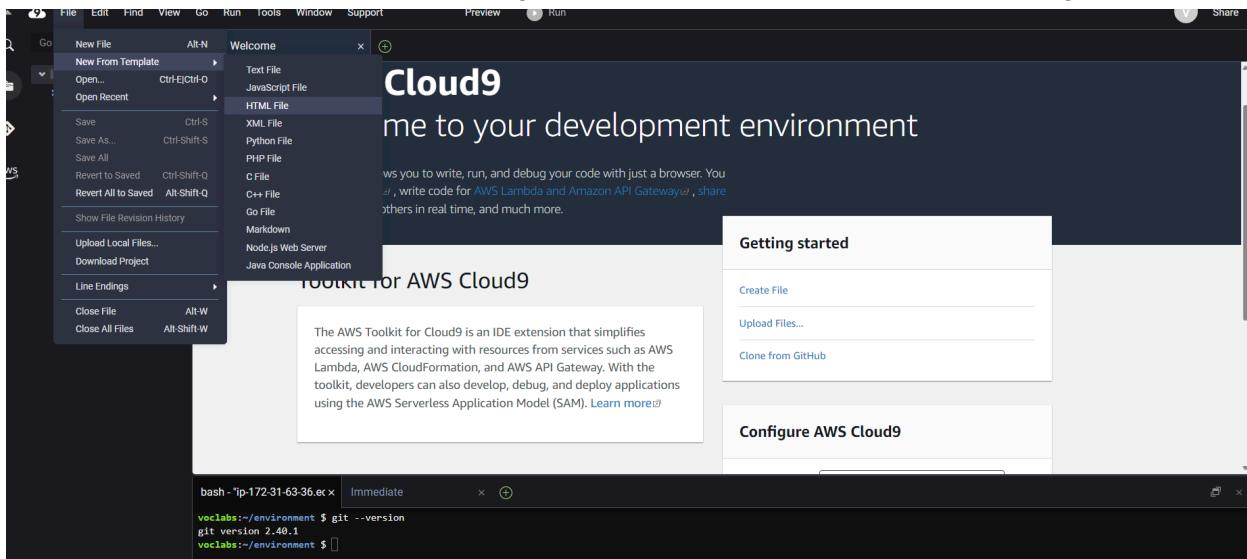
Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Ayush Maurya	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::217253764927:assumed-role/voclabs/user3385518=MAURYA_AYUSH_SUBHASHCHANDRA

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

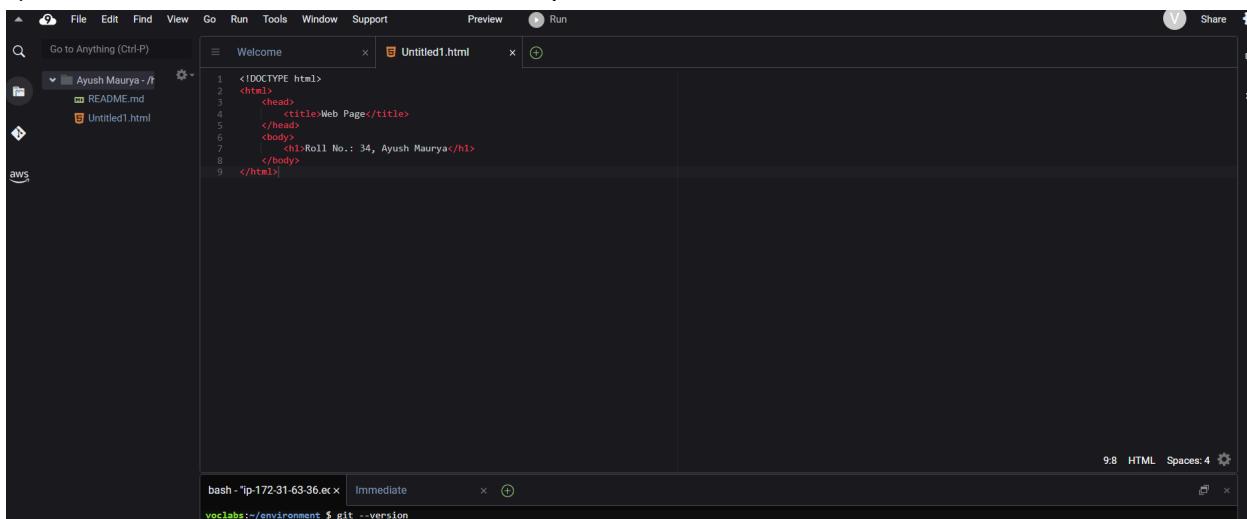
The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a navigation bar with links for File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run. The main area has a dark theme with a sidebar on the left containing a file tree showing 'Ayush Maurya - /' with subfolders '.9' and 'README.md'. The central area displays the AWS Cloud9 welcome message: 'AWS Cloud9' and 'Welcome to your development environment'. Below this is a 'Toolkit for AWS Cloud9' section with a description of the toolkit and a 'Learn more' link. At the bottom, there's a command-line interface (CLI) window showing a terminal session:

```
bash - *ip-172-31-63-36.ex ✘ Immediate ✘ +  
voclabs:~/environment $
```

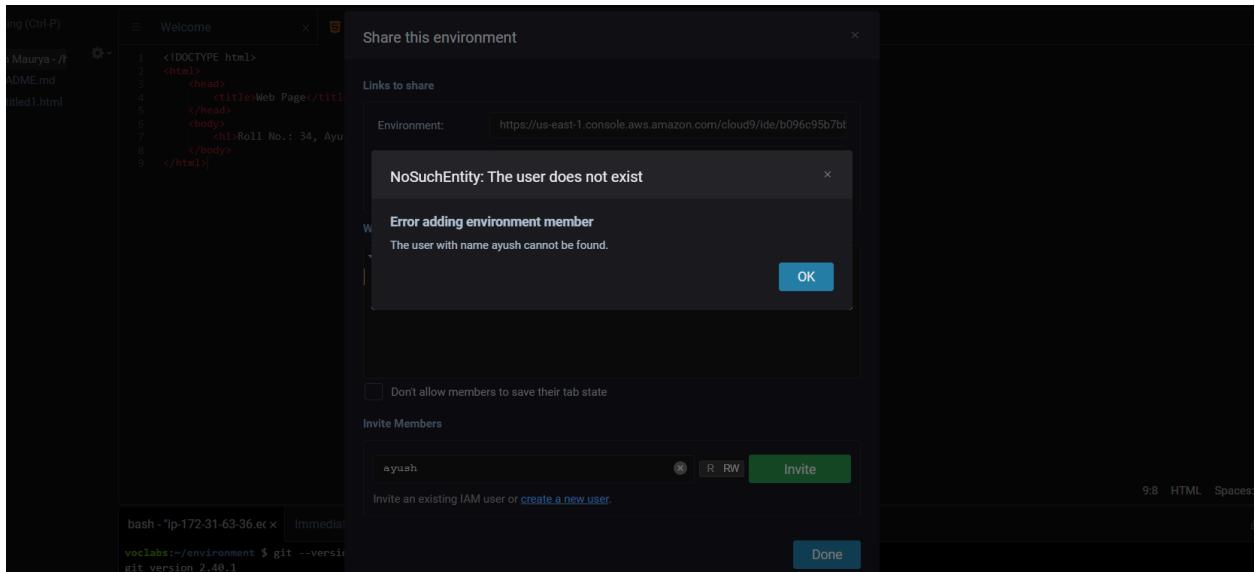
3) To add a file, click on file. For this experiment, we are to add an HTML file. So go to File → New From Template → HTML file. This gives a basic HTML template on the coding IDE.



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



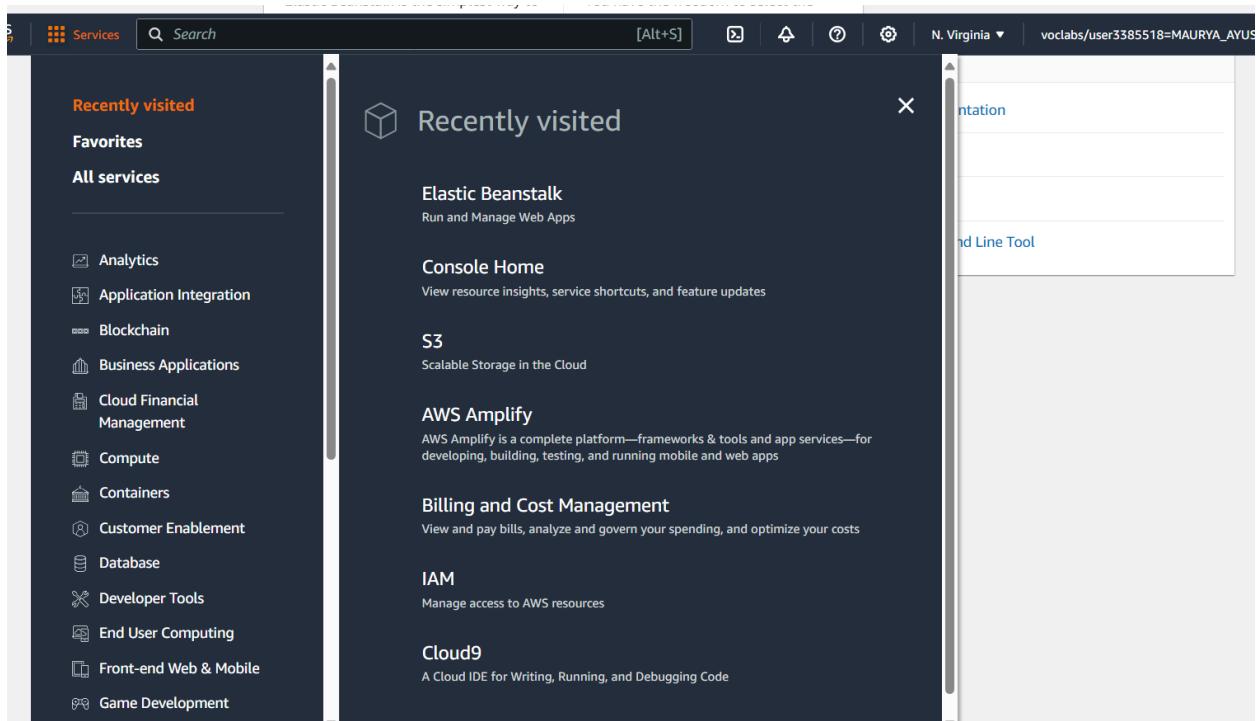
After saving, on the toolbar towards the far right, click on Share. Then put the username that you had put during creating IAM user.



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

Experiment 2

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



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

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. [Learn more](#)

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

[Getting started](#)

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

Step 1
Configure environment

Step 2
Configure service access

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

Step 4 - optional
Configure instance traffic and scaling

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment Info

Environment tier Info
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

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

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

Application information Info

Application name
my_web_34
Maximum length of 100 characters.

Application tags (optional)

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

Environment name
Myweb34-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain
Leave blank for autogenerated value .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
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform
PHP

Platform branch
PHP 8.3 running on 64bit Amazon Linux 2023

PHP 8.3 running on 64bit Amazon Linux 2023

[Alt+S] | | | | N. Virginia | vclabs/user3385518=

4.3.1 (Recommended)

Application code Info

Sample application

Existing version
Application versions that you have uploaded.

Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Presets Info

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

Configuration presets

Single instance (free tier eligible)

Single instance (using spot instance)

High availability

High availability (using spot and on-demand instances)

Custom configuration

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

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

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk setup wizard. On the left, a sidebar lists steps: 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 panel is titled 'Service access' and contains the following fields:

- Service role:** A radio button group where 'Use an existing service role' is selected. Below it, a dropdown menu shows 'AWSCloud9SSMAccessRole' with a clear button.
- EC2 key pair:** A dropdown menu labeled 'Choose a key pair' with a clear button.
- EC2 instance profile:** A dropdown menu labeled 'AWSCloud9SSMInstanceProfile' with a clear button. Below it is a 'View permission details' button.

Step 5 : Review the settings that you have set up for your application and submit your application.

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

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

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

Step 6
Review

Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	my_web_34
Environment name	Application code
Myweb34-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3	
running on 64bit Amazon Linux 2023/4.3.1	

Step 2: Configure service access [Edit](#)

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::011528263337:role/service-role/AWSCloud9SSMAccessRole	AWSCloud9SSMInstanceProfile

Step 3: Set up networking, database, and tags [Edit](#)

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System enhanced	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
—	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false
Updates		
Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	

[Alt+S]



Platform software

Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	—	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

Environment properties

Key	▲ Value
No environment properties	
There are no environment properties defined	

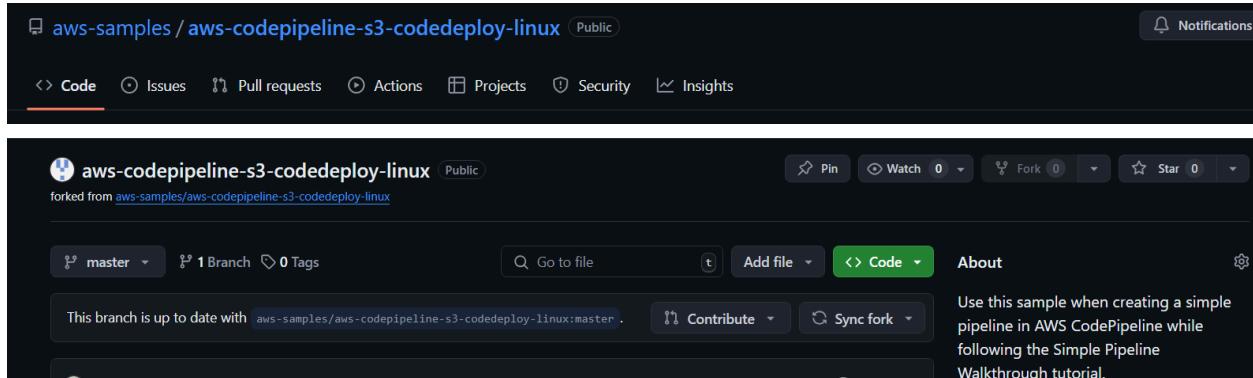
Cancel

Previous

Submit

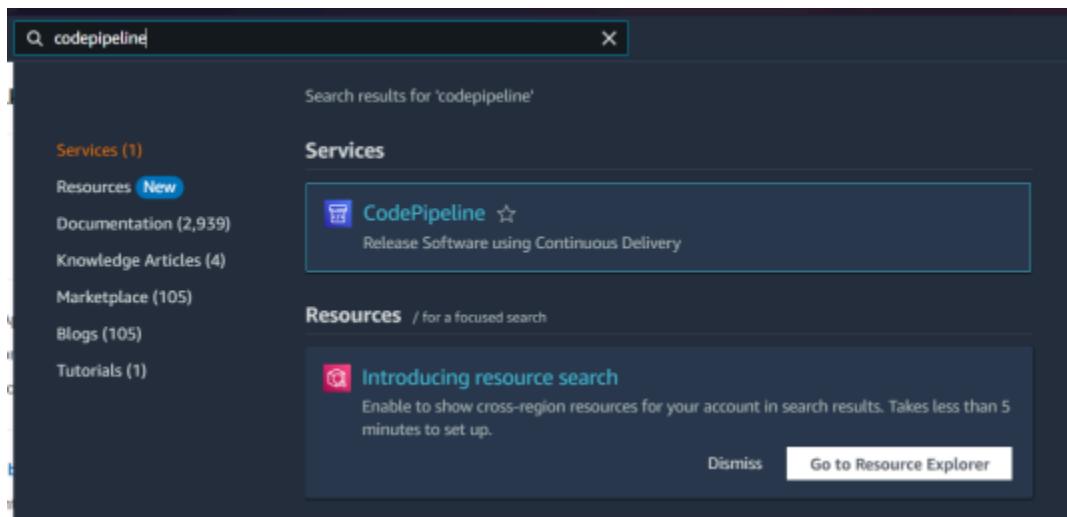
Step 6: Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

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



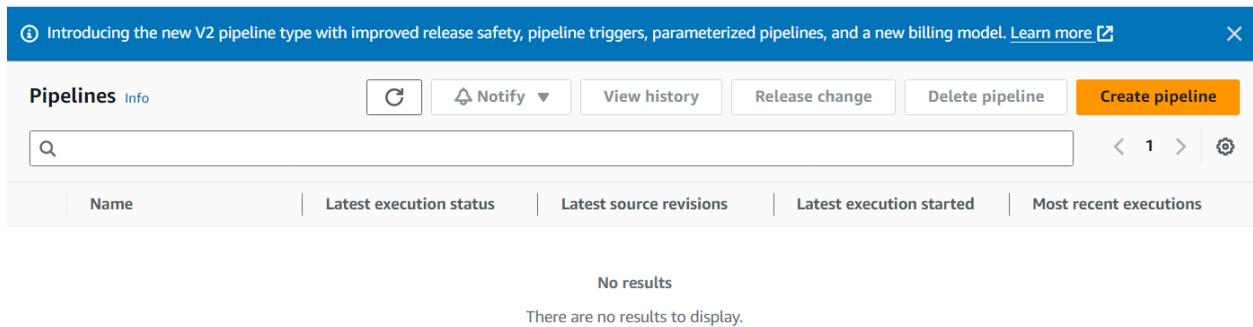
The image shows two screenshots of GitHub repository pages. The top screenshot is for the repository 'aws-samples / aws-codepipeline-s3-codedeploy-linux' (Public), which is a fork of 'aws-samples/aws-codepipeline-s3-codedeploy-linux'. The bottom screenshot is for the repository 'aws-codepipeline-s3-codedeploy-linux' (Public), also forked from the same source. Both pages display standard GitHub interface elements such as navigation tabs (Code, Issues, Pull requests, Actions, Projects, Security, Insights), repository statistics (master branch, 1 branch, 0 tags), search bars, and a note about the repository's purpose for AWS CodePipeline usage.

Step 7: Search CodePipeline in the services tab and click on it.



The image shows the AWS Services Catalog search results for 'codepipeline'. The 'Services' section is expanded, showing one result: 'CodePipeline' (Release Software using Continuous Delivery). A modal window titled 'Introducing resource search' is displayed, explaining how to enable cross-region resource search. The sidebar on the left lists other categories like Resources, Documentation, Knowledge Articles, Marketplace, Blogs, and Tutorials.

Step 8: Click on Create Pipeline.



The image shows the AWS CodePipeline Pipelines page. At the top, there is a banner about the new V2 pipeline type. Below it, there are buttons for Pipelines (Info), Refresh, Notify, View history, Release change, Delete pipeline, and Create pipeline. A search bar and pagination controls are also present. The main area displays a table with columns for Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. The table is currently empty, showing the message 'No results' and 'There are no results to display.'

Step 9: Give a name to your Pipeline. A new service role would be created with the name of the pipeline

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose pipeline settings Info

Step 1 of 5

Pipeline settings

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

Pipeline type

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

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

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

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

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

Service role

New service role Existing service role [Alt+S]

Role name

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

Variables

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

No variables defined at the pipeline level in this pipeline.

Add variable
You can add up to 50 variables.

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

Advanced settings

Cancel **Next**

Step 10 : Select a source provider (as Github (Version 2)). Click on Connect to Github to connect your github.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage Step 2 of 5

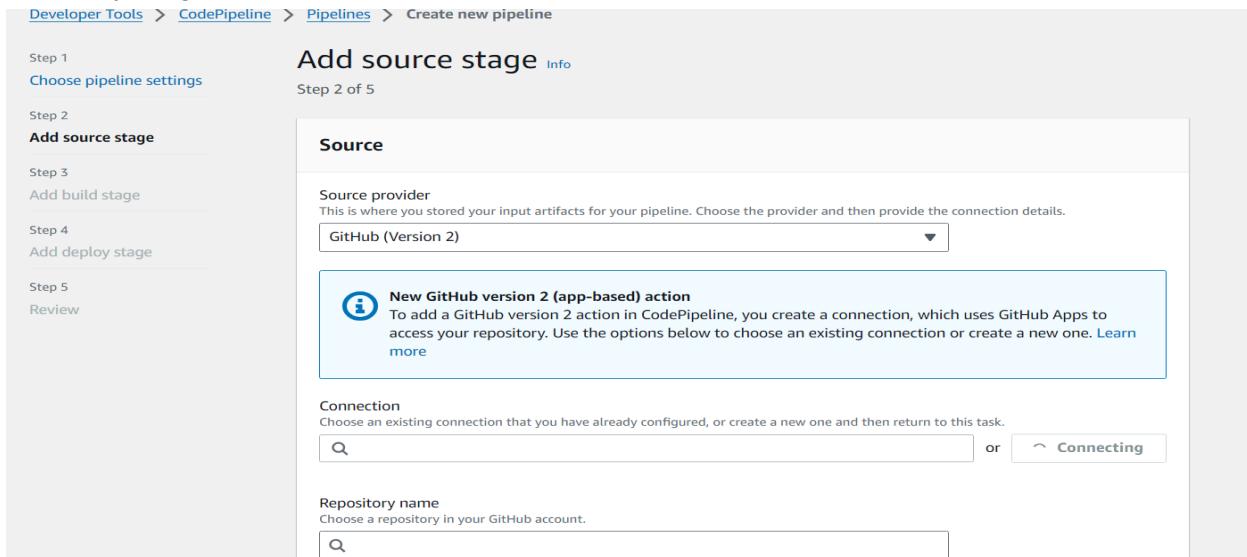
Source provider This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

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.

Repository name Choose a repository in your GitHub account.



Step 11: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app

Developer Tools > Connections > Create connection

Create a connection

Create GitHub App connection

Connection name MyGitHub

Tags - optional

Connect to GitHub



Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with no disconnections in the resource ARN. Resources with both service profiles will continue to display in the console. [Learn more](#)

Connect to GitHub

GitHub connection settings

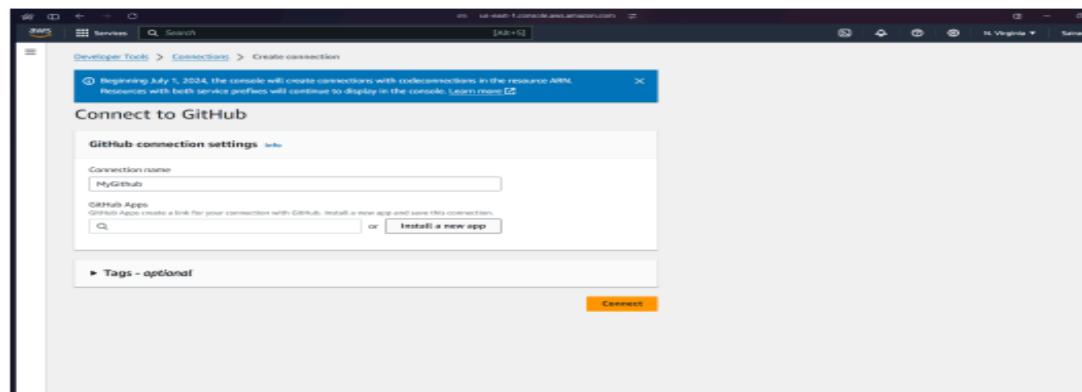
Connection name MyGitHub

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

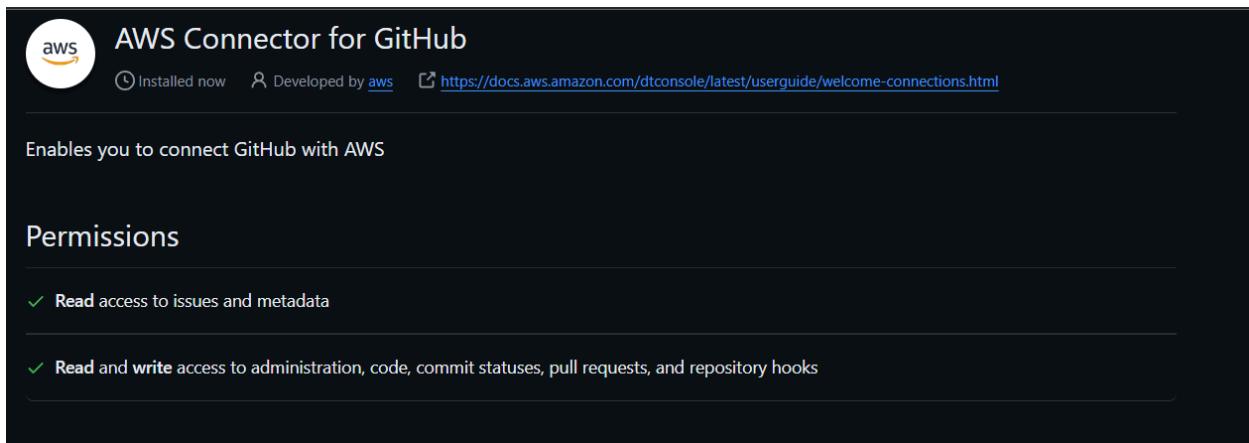
Install a new app or Connect

Tags - optional

Connect



Step 12 : This will direct you to install AWS Connector On Your GitHub. Install it to your account and give it its permissions.



Step 13: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use

A screenshot of the "Create connection" page for GitHub. The URL in the address bar is "Developer Tools > Connections > Create connection". A blue banner at the top contains a message: "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" with a link icon. Below the banner, the section title is "GitHub connection settings" with an "Info" link. The "Connection name" field contains "MyGithub". Under "GitHub Apps", there's a search bar with "53685085", an "X" button, and an "Install a new app" button. At the bottom, there's a "Tags - optional" section and a large orange "Connect" button.

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

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

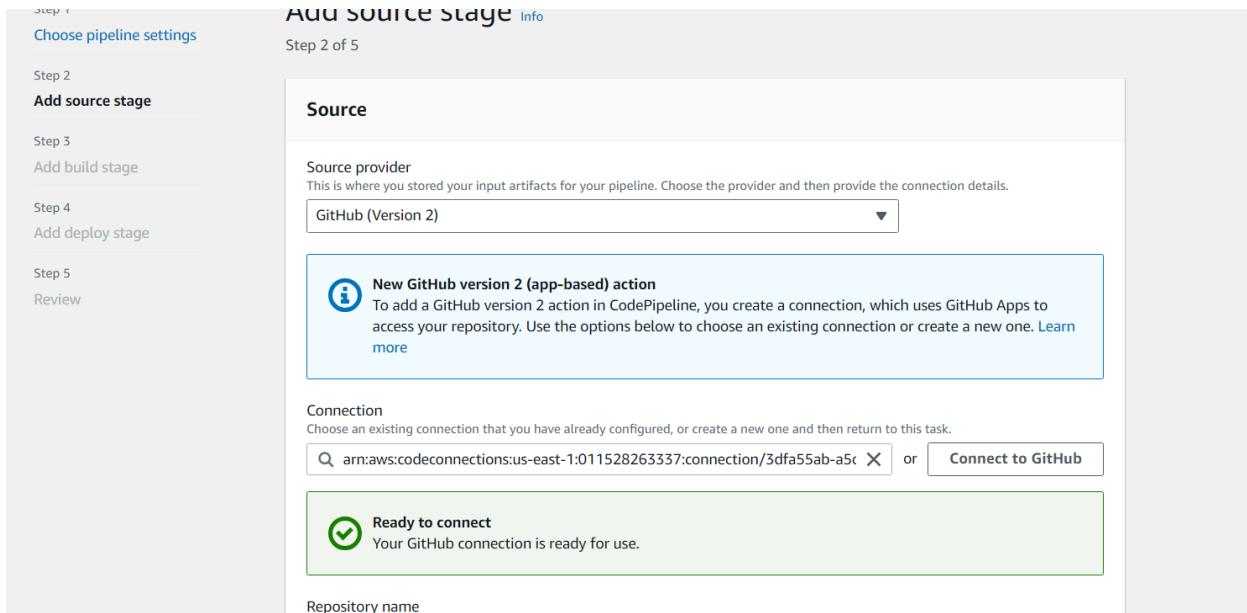
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:us-east-1:011528263337:connection/3dfa55ab-a5c X or [Connect to GitHub](#)

Ready to connect

Your GitHub connection is ready for use.

Repository name



Step 14: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

[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-1:011528263337:connection/3dfa55ab-a5c X or [Connect to GitHub](#)

Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

AyushMaurya3114/aws-codepipeline-s3-codedeploy-linux X

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.

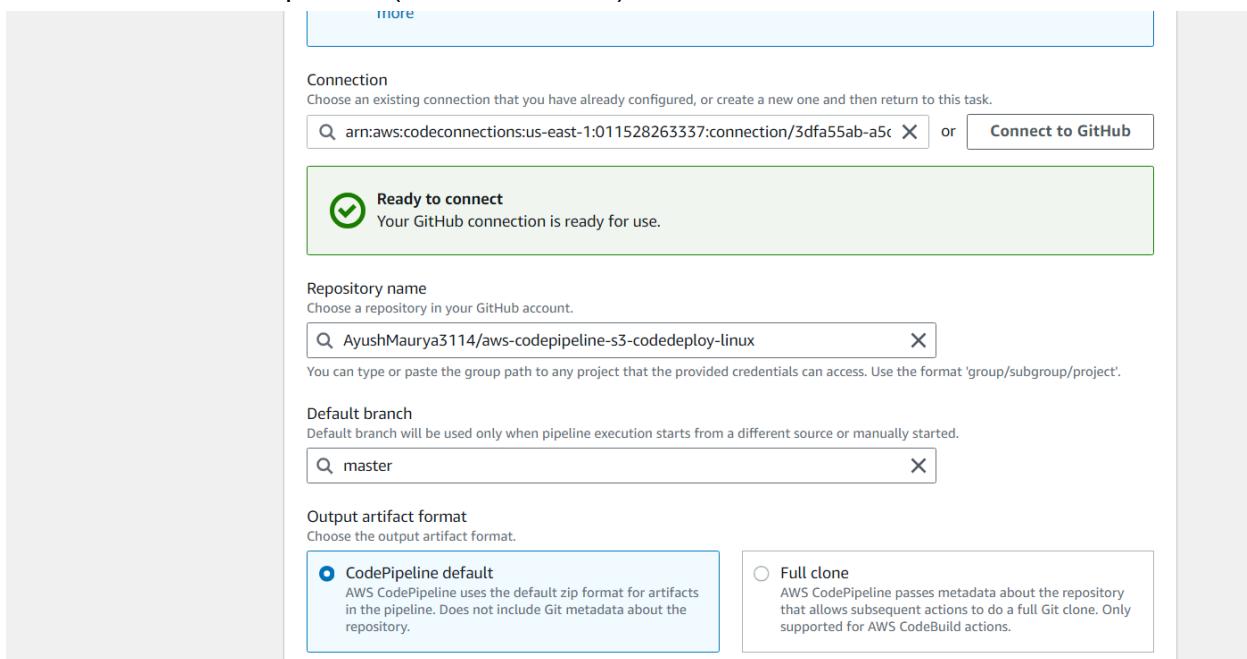
master X

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.



Step 15: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

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.

i You can add additional sources and triggers by editing the pipeline after it is created.

[Cancel](#) [Previous](#) **Next**

Step 16: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

Step 1 [Choose pipeline settings](#)
Step 2 [Add source stage](#)
Step 3 **Add build stage**
Step 4 [Add deploy stage](#)
Step 5 [Review](#)

Add build stage Info

Step 3 of 5

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

[Cancel](#) [Previous](#) [Skip build stage](#) **Next**

Add deploy stage

Step 5
Review

Deploy

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

AWS Elastic Beanstalk

Region
US East (N. Virginia)

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

SourceArtifact

No more than 100 characters

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

my_web_34

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.

Myweb34-env

Configure automatic rollback on stage failure

Cancel Previous Next

Step 17: Check all the information and click on create Pipeline.

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

my_web_34

EnvironmentName

Myweb34-env

Configure automatic rollback on stage failure

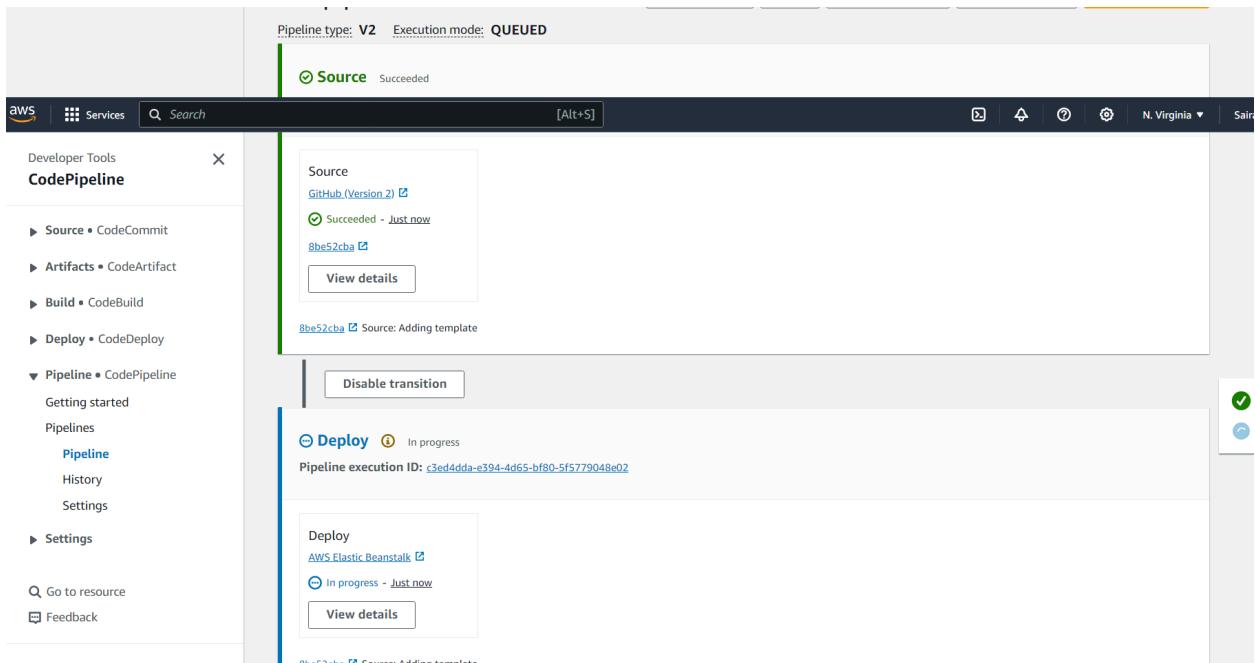
Disabled

Cancel

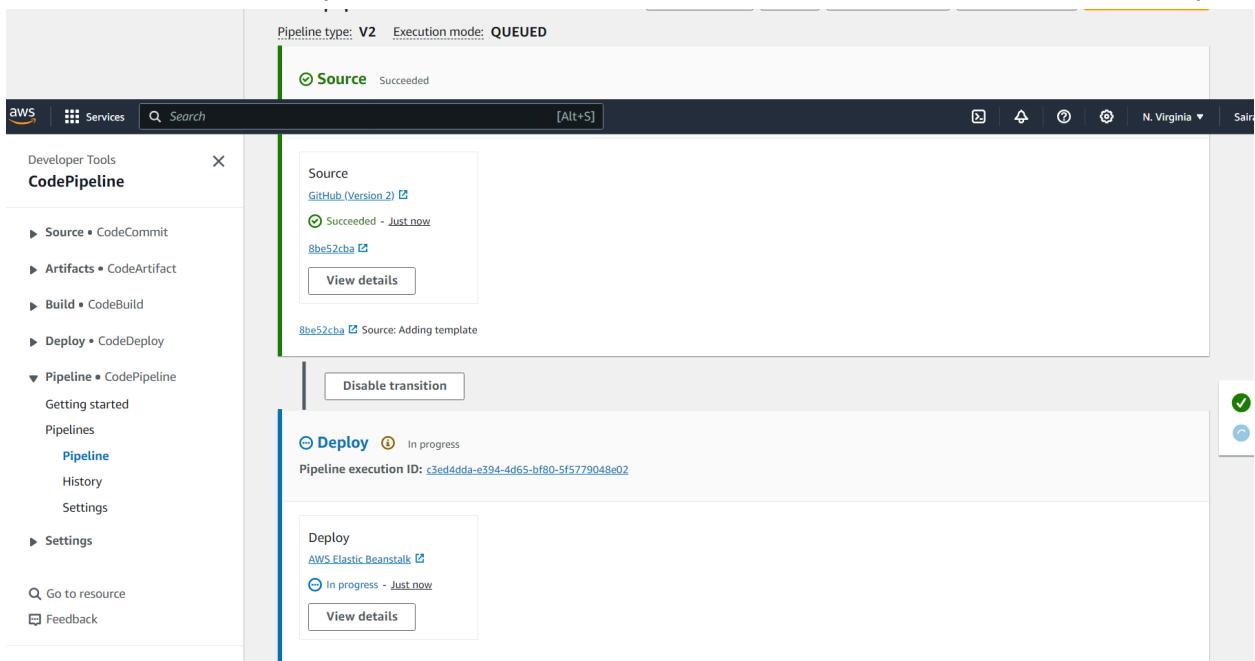
Previous

Create pipeline

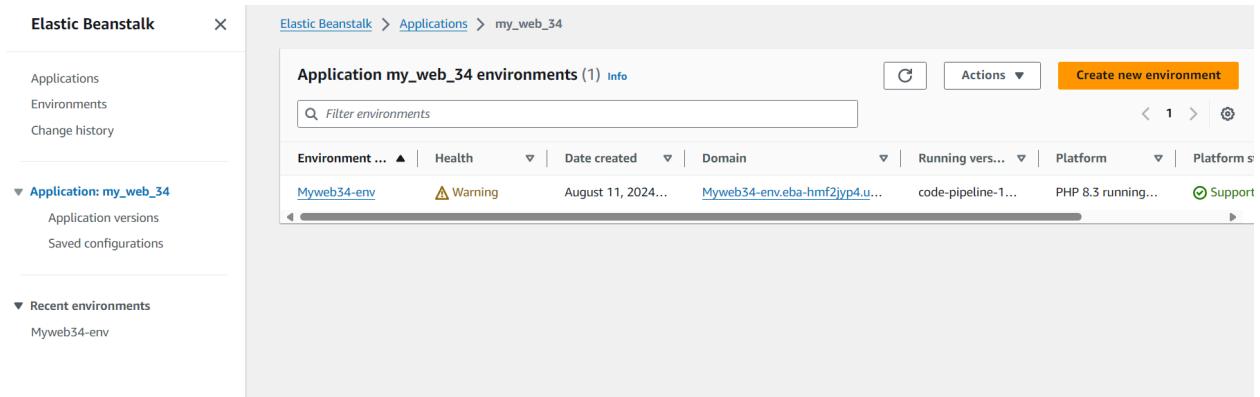
Step 18: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy.



Step 19: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



Step 20: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain.



The screenshot shows the AWS Elastic Beanstalk interface. On the left, there's a sidebar with options like Applications, Environments, Change history, Application: my_web_34 (selected), Application versions, Saved configurations, and Recent environments (Myweb34-env). The main area is titled 'Application my_web_34 environments (1) Info'. It lists one environment: 'Myweb34-env' with a warning icon, created on August 11, 2024, using the 'Myweb34-env.eba-hmf2jyp4.u...' domain, the 'code-pipeline-1...' platform, and PHP 8.3 running. There are buttons for Actions, Create new environment, and Support.

Step 21: This will successfully show the sample website hosted.



Step 22: Now, we make some changes to the index.html file in the github.

For eg: If you make some changes to the <h2>tag.

Once the changes are committed, when the website is refreshed, the changes can be seen.



Experiment 3

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

Theory:

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

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

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat. Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their applications while Kubernetes takes care of key tasks, including:

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

Necessary Requirements:

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

- **Minimum Requirements:**

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

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

Sign in to AWS Management Console:

Go to AWS Management Console.
Log in with your account credentials.

Navigate to EC2 Service:

In the AWS Console, search for EC2 and select it to open the EC2 dashboard.

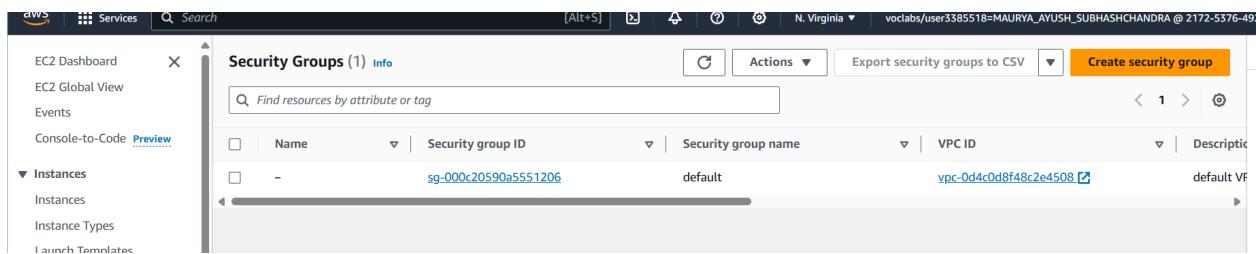
Go to "Security Groups":

On the left-hand navigation pane, under Network & Security, click on Security Groups.

Create Security Group:

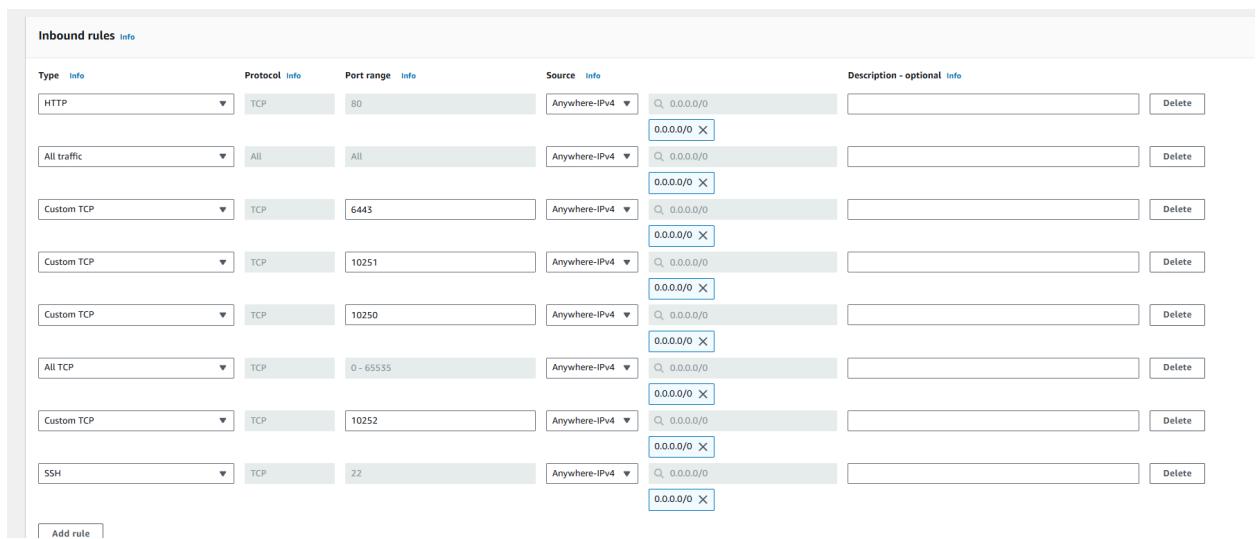
Click the Create Security Group button at the top.

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



The screenshot shows the AWS EC2 Security Groups page. At the top, there is a search bar and a 'Create security group' button. Below the header, a table lists one security group: 'sg-000c20590a5551206' with 'Name' as 'default', 'VPC ID' as 'vpc-0d4c0d8f48c2e4508', and 'Description' as 'default VPC'. The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, and Launch Templates.

MASTER:



The screenshot shows the 'Inbound rules' section of a security group. It lists ten rules:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere-IPv4	0.0.0.0/0
All traffic	All	All	Anywhere-IPv4	0.0.0.0/0
Custom TCP	TCP	6443	Anywhere-IPv4	0.0.0.0/0
Custom TCP	TCP	10251	Anywhere-IPv4	0.0.0.0/0
Custom TCP	TCP	10250	Anywhere-IPv4	0.0.0.0/0
All TCP	TCP	0 - 65535	Anywhere-IPv4	0.0.0.0/0
Custom TCP	TCP	10252	Anywhere-IPv4	0.0.0.0/0
SSH	TCP	22	Anywhere-IPv4	0.0.0.0/0

At the bottom left, there is a 'Add rule' button.

NODE:

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances. Select Ubuntu as AMI and t2.micro (because in academic account only t2.micro is present) as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder. We can use 3 Different keys or 1 common key also.

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

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0216 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

[Compare instance types](#)

[All generations](#)

[Additional costs apply for AMIs with pre-installed software](#)

Key pair (login) [Info](#)

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

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

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)

ami-0e86ec20dae9224db8

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

Network settings [Info](#)

Network [Info](#)
vpc-0d4c008f48c2e4508

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

Auto-assign public IP [Info](#)
Enable

Additional charges apply when outside of free tier allowance

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

Create security group Select existing security group

Common security groups [Info](#)

Select security groups [Compare security group rules](#)

MasterGroup sg-097fc30a345c1a537 X
VPC: vpc-0d4c008f48c2e4508

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

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)

ami-0e86ec20dae9224db8

Virtual server type (instance type)

t2.micro

Firewall (security group)

MasterGroup

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

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

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

Create security group Select existing security group

Common security groups [Info](#)

Select security groups [Compare security group rules](#)

NodeGroup sg-030c0a1b62a1e9894 X
VPC: vpc-0d4c008f48c2e4508

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

Firewall (security group)

NodeGroup

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

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Master	i-0c67658f4d6eeea8fc	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-208-20-141.compute-1.amazonaws.com	54.208.20.141
MASTER	i-0044a08ca3541e460	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-
nagios-host	i-04709b3512d97f50f	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-
node1	i-0414d4f92af63c03e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-35-170-201-119.compute-1.amazonaws.com	35.170.201.119
node2	i-0d57570c061c25ae1	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-184-73-147-235.compute-1.amazonaws.com	184.73.147.235
SLAVE	i-0dd0a5a729eb2e648	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-
SLAVE2	i-0426944f0d33bd940	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-

Step 3: Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.
ssh -i "<PATH TO FILE>exp3.pem" ubuntu@ec2-54-208-20-141.compute-1.amazonaws.com

MASTER:

CONNECT TO INSTANCE [intro](#)

Connect to your instance i-0c67658f4d6eea8fc (Master) using any of these options

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

Instance ID
[i-0c67658f4d6eea8fc \(Master\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp3.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "exp3.pem"`
4. Connect to your instance using its Public DNS:
`ec2-54-208-20-141.compute-1.amazonaws.com`

Example:
`ssh -i "exp3.pem" ubuntu@ec2-54-208-20-141.compute-1.amazonaws.com`

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

NODE 1&2:

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

Instance ID
[i-0414d4f92af63c03e \(node1\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp3.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "exp3.pem"`
4. Connect to your instance using its Public DNS:
`ec2-35-170-201-119.compute-1.amazonaws.com`

Example:
`ssh -i "exp3.pem" ubuntu@ec2-35-170-201-119.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.

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

Instance ID
[i-0d57570c061c25ae1 \(node2\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp3.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "exp3.pem"`
4. Connect to your instance using its Public DNS:
`ec2-184-73-147-235.compute-1.amazonaws.com`

Example:
`ssh -i "exp3.pem" ubuntu@ec2-184-73-147-235.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.

```

ubuntu@ip-172-31-82-192:~$.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-208-20-141.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 Thu Sep 26 15:21:26 UTC 2024

System load: 0.0      Processes:          104
Usage of /: 22.7% of 6.71GB  Users logged in:   0
Memory usage: 20%
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-82-192:~$
```



```

ubuntu@ip-172-31-86-84:~$.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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

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

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

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

ubuntu@ip-172-31-86-84:~$
```

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

```

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

```

ubuntu@ip-172-31-82-192:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

```
ubuntu@ip-172-31-82-192:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-82-192:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble
stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
[126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
[126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Pa
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiv
erse amd64 Components [212 B]
Get:51 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiv
erse amd64 c-n-f Metadata [116 B]
Get:52 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages
[15.3 kB]
Fetched 29.1 MB in 6s (4873 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s)
) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file h
as an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
ubuntu@ip-172-31-82-192:~$
```

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

```
ubuntu@ip-172-31-82-192:~$ 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 h
as an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  libltdl7 libslirp0 pigz
  slirp4netns
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-172-31-82-192:~$ |
```

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

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-192:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-82-192:~$ |
```

sudo systemctl enable docker

sudo systemctl daemon-reload

sudo systemctl restart docker

```
ubuntu@ip-172-31-82-192:~$ 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-172-31-82-192:~$ |
```

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-172-31-82-192:~$ # Add the Kubernetes GPG key and save it to the keyring
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

# Add the Kubernetes repository to your APT sources list
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

# Update package lists
sudo apt update
File '/etc/apt/keyrings/kubernetes-apt-keyring.gpg' exists. Overwrite? (y/N)
y
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/
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:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 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 (6276 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
142 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-82-192:~$ |
```

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
ubuntu@ip-172-31-82-192:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s)
) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file h
as an unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 142 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntr
ack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [11.2 MB]
Processing triggers for man-db (2.12.0-4ubuntu2) ...
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-172-31-82-192:~$ |
```

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

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

```
ubuntu@ip-172-31-82-192:~$ 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:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 142 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64
```

```
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.
```

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

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

```
ubuntu@ip-172-31-82-192:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml

disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""
```

```
[proxy_plugins]
[stream_processors]
[stream_processors."io.containerd.ocicrypt.decoder.v1.tar"]
    accepts = ["application/vnd.oci.image.layer.v1.tar+encrypted"]
    args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
    env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_ke
yprovider.conf"]
    path = "ctd-decoder"
    returns = "application/vnd.oci.image.layer.v1.tar"

[stream_processors."io.containerd.ocicrypt.decoder.v1.tar.gzip"]
    accepts = ["application/vnd.oci.image.layer.v1.tar+gzip+encrypted"]
    args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
    env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_ke
yprovider.conf"]
    path = "ctd-decoder"
    returns = "application/vnd.oci.image.layer.v1.tar+gzip"

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

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

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

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

```
ubuntu@ip-172-31-82-192:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 142 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat
  amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (15.7 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-82-192:~$ |
```

Step 6: Initialize the Kubercluster. Now perform this on Master Instance.

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

```

ubuntu@ip-172-31-82-192:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
[init] Using Docker version 20.10.12
[WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
[WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[preflight] Pulling images required for setting up a Kubernetes component
[preflight] This might take a few minutes, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using "kubeadm config images pull"
W0926 16:55:13.698118 9239 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] Generating certificates and keys for "kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-82-192 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.82.192]
[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/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-82-192 localhost] and IPs [172.31.82.192 127.0.0.1 :1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-82-192 localhost] and IPs [172.31.82.192 127.0.0.1 :1]
[certs] Generating "etcd-peer-etcd-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using namespace kubelet for kubeconfig files
[kubeconfig] Writing "ca" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest Folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kublet environment file with flags to file "/var/lib/kublet/kubeadm-flags.env"
[kubelet-start] Starting the kublet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for the kubelet to start listening on port 10250:10250<http://>
[kubelet-check] Waiting for the kubelet to be healthy on port 10250:10250<http://>
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 11.508580128s
[upload-certs] Uploading configuration for kubelet's ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kublet-config" in the "kube-public" Namespace
[upload-certs] Skipping phase. Please see "--upload-certs"
[mark-control-plane] Marking the node ip-172-31-82-192 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane:node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Using taints: kubernetes.io/not-ready:NoSchedule
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to update nodes
[bootstrap-token] Configured RBAC rules to allow the controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: CoreOS
[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 ip-172-31-82-192:6443 --token jnrrt-eyi0tdu03zqep \
--discovery-token-ca-cert-hash sha256:a6d3703c5688df5caadd8df018093d17c759db0b046a5cc908a99f2e52d2055

```

From this command we get token and ca-

Run this command on master and

```

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-172-31-82-192:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-82-192:~$

```

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

```

ubuntu@ip-172-31-82-192:~$ kubectl get nodes
NAME           STATUS      ROLES   AGE      VERSION
ip-172-31-82-192   NotReady   control-plane   7m21s   v1.31.1
ubuntu@ip-172-31-82-192:~$ 

```

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

```

sudo kubeadm join <your-master-node-ip>:6443 --token <your-token>
--discovery-token-ca-cert-hash sha256:<your-ca-cert-hash>

```

```
kubeadm join 172.31.82.192:6443 --token jrhztc.eyi07duk03zq4eqr \
--discovery-token-ca-cert-hash
sha256:a6037b3c6608d5fdadd8dfd100793d17c759dbeb046a5cc908a90f2e52d2055
(SLASH '\' MIGHT GIVE ERROR IF IT GIVES ERROR THEN TRY WITHOUT ERROR)
```

NODE1:

```
ubuntu@ip-172-31-86-84:~$ sudo kubeadm join 172.31.82.192:6443 --token jrhztc.eyi07duk03zq4eqr --discovery-token-ca-cert-hash sha256:a6037b3c6608d5fdadd8dfd100793d17c759dbeb046a5cc908a90f2e52d2055
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.004592587s
[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-172-31-86-84:~$
```

NODE2:

```
ubuntu@ip-172-31-86-8:~$ sudo kubeadm join 172.31.82.192:6443 --token jrhztc.eyi07duk03zq4eqr --discovery-token-ca-cert-hash sha256:a6037b3c6608d5fdadd8dfd100793d17c759dbeb046a5cc908a90f2e52d2055
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.003588565s
[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-172-31-86-8:~$
```

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

```
ubuntu@ip-172-31-82-192:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-82-192  NotReady  control-plane  55m    v1.31.1
ip-172-31-86-8  NotReady  <none>       7m44s   v1.31.1
ip-172-31-86-84 NotReady  <none>       7m5s    v1.31.1
ubuntu@ip-172-31-82-192:~$ |
```

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

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

```
ubuntu@ip-172-31-82-192:~$ 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/irreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-82-192:~$ |
```

sudo systemctl status kubelet

```
ubuntu@ip-172-31-82-192:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
   Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Thu 2024-09-26 16:55:46 UTC; 57min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 7822 (kubelet)
     Tasks: 10 (limit: 1130)
    Memory: 49.2M (peak: 72.1M)
      CPU: 44.924s
     CGroup: /system.slice/kubelet.service
             └─7822 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kube>
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: E0926 17:52:55.304654    7822 kuberuntime_>
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]:                      rpc error: code = Unknown desc = f>
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]:                      : unknown
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: > pod="kube-system/kube-scheduler-ip-172->
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: E0926 17:52:55.393676    7822 log.go:32] ">
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]:                      rpc error: code = Unknown desc = f>
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]:                      : unknown
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: > podSandboxID="af90c21fb06b79773120a5ffc">
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: E0926 17:52:55.393747    7822 kuberuntime_>
Sep 26 17:52:55 ip-172-31-82-192 kubelet[7822]: E0926 17:52:55.393869    7822 kubelet.go:1>
[lines 1-23/23 (END)]
```

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

```
ubuntu@ip-172-31-82-192:~$ kubectl get nodes -o wide
NAME           STATUS  ROLES   AGE    VERSION INTERNAL-IP    EXTERNAL-IP  OS-IMAGE         KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-82-192  Ready   control-plane   58m   v1.31.1  172.31.82.192  <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ip-172-31-86-8   Ready   <none>    16m   v1.31.1  172.31.86.8   <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ip-172-31-86-84  Ready   <none>    9m42s  v1.31.1  172.31.86.84  <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ubuntu@ip-172-31-82-192:~$ |
```

Now to Rename run this command

kubectl label node <node-ip> kubernetes.io/role=worker

Rename to Node 1:**kubectl label node ip-172-31-86-8 kubernetes.io/role=Node1**

Rename to Node 2:**kubectl label node ip-172-31-86-84 kubernetes.io/role=Node2**

```
ubuntu@ip-172-31-82-192:~$ kubectl label node ip-172-31-86-8 kubernetes.io/role=Node1
[[A^[[Anode/ip-172-31-86-8 labeled
ubuntu@ip-172-31-82-192:~$ kubectl label node ip-172-31-86-84 kubernetes.io/role=Node2
node/ip-172-31-86-84 labeled
ubuntu@ip-172-31-82-192:~$ |
```

Step 11: Run command **kubectl get nodes -o wide** . And Hence we can see we have

Successfully connected Node 1 and Node 2 to the Master.

```
ubuntu@ip-172-31-82-192:~$ kubectl get nodes -o wide
NAME           STATUS  ROLES   AGE    VERSION INTERNAL-IP    EXTERNAL-IP  OS-IMAGE         KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-82-192  Ready   control-plane   83m   v1.31.1  172.31.82.192  <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ip-172-31-86-8   Ready   Node1    35m   v1.31.1  172.31.86.8   <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ip-172-31-86-84  Ready   Node2    34m   v1.31.1  172.31.86.84  <none>        Ubuntu 24.04 LTS  6.8.0-1012-aws  containerd://1.7.12
ubuntu@ip-172-31-82-192:~$ kubectl get nodes -o wide|
```

Or run **kubectl get nodes**

```
^[[Aubuntu@ip-172-31-82-192 kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-172-31-82-192  Ready   control-plane   85m   v1.31.1
ip-172-31-86-8   Ready   Node1    37m   v1.31.1
ip-172-31-86-84  Ready   Node2    36m   v1.31.1
ubuntu@ip-172-31-82-192:~$ |
```

Conclusion:

In this Advanced DevOps Lab experiment, we began by setting up three EC2 Ubuntu instances on AWS, designating one as the Master node and the others as Worker nodes. We then installed Docker and Kubernetes on all instances, ensuring Docker was properly configured. The Kubernetes cluster was initialized on the Master node, and the Flannel networking plugin was applied to facilitate communication between nodes. Finally, we joined the Worker nodes to the cluster using the provided token and hash, resulting in a fully operational Kubernetes cluster ready for managing and scaling containerized applications.

Experiment 4

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

Theory:

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

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

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat. Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their applications while Kubernetes takes care of key tasks, including:

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

Necessary Requirements:

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

• Minimum Requirements:

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

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

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances. Select Ubuntu as AMI and t2.micro (because in academic account only t2.micro is present) as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.

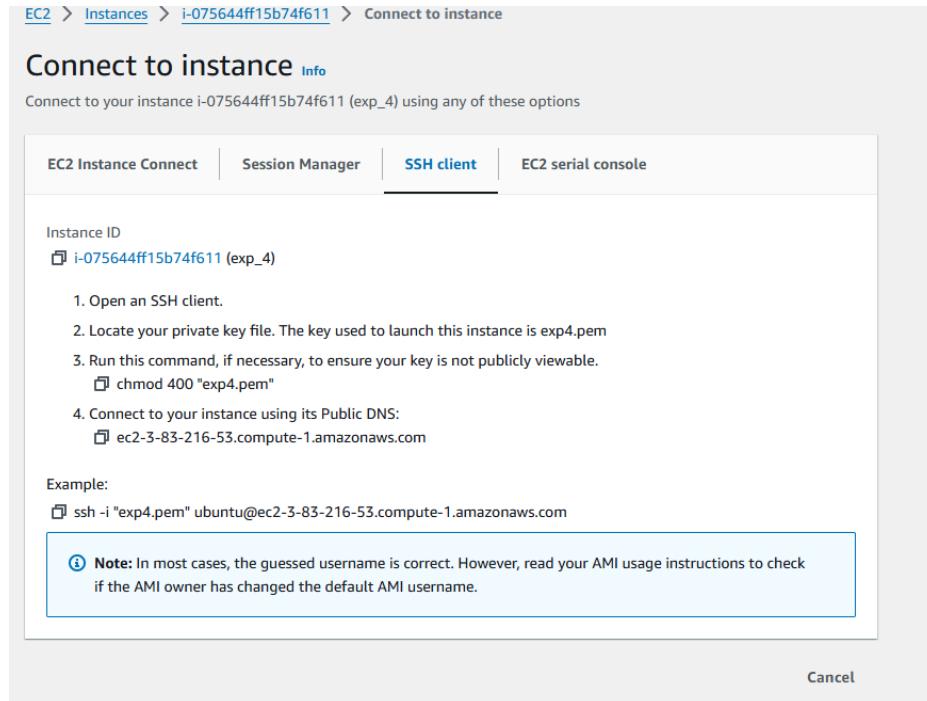
The screenshot shows the AWS CloudFormation console with the following details:

- Stack Name:** exp_4
- Template:** CloudFormation template (yaml)
- Outputs:**
 - MyWebAppURL: http://ec2-18-233-163-90.compute-1.amazonaws.com
- Next Step:** Create Change Set (highlighted in orange)

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

The screenshot shows the AWS EC2 Instances page with the following details:

- Instances:** exp_4, Master, node1, node2 (all running)
- Actions:** Connect (highlighted in orange)



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

```
ssh -i "<PATH TO FILE>exp3.pem" ubuntu@ec2-3-83-216-53.compute-1.amazonaws.com
```

```
C:\Users\Ayush Maurya>ssh -i "Downloads/exp4.pem" ubuntu@ec2-3-83-216-53.compute-1.amazonaws.com
The authenticity of host 'ec2-3-83-216-53.compute-1.amazonaws.com (3.83.216.53)' can't be established.
ED25519 key fingerprint is SHA256:jdAspK3Zoikd8Xh0vy+g6Ea5WqRKlgfnr5S66tYTRg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-83-216-53.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 Fri Sep 27 04:30:29 UTC 2024

 System load:  0.03           Processes:      105
 Usage of /:   22.8% of 6.71GB   Users logged in:  0
 Memory usage: 20%            IPv4 address for enp0: 172.31.93.95
```

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

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

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

```
ubuntu@ip-172-31-93-95:~$ |
```

Step 4: Run 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"
```

```
[root@ip-172-31-93-96 ~]# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl: https://download.docker.com/linux/ubuntu/gpg: failed to open stream: No such file or directory
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))
DE
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description: Docker CE stable components: stable
Architectures: codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository...
Pre-adding repository to continue via 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.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB]
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble universe InRelease [48.8 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [55.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Translation-en [115 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe multiverse Packages [389 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [80 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main i386 Packages [22 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [832 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c=1-f Metadata [10.3 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c=1-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [82.9 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c=1-f Metadata [428 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [82.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [208 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [208 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [394 kB]
Get:22 https://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [15.3 kB]
Get:23 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:24 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Translation-en [115 kB]
Get:25 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:26 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:27 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Translation-en [115 kB]
Get:28 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c=1-f Metadata [832 kB]
Get:29 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:30 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [10 kB]
Get:31 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Get:32 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe i386 Packages [10 kB]
Get:33 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [374 kB]
Get:34 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse i386 Components [374 kB]
Get:35 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [181 kB]
Get:36 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [181 kB]
Get:37 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c=1-f Metadata [428 kB]
Get:38 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [114 kB]
Get:39 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [532 kB]
Get:40 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 kB]
Get:41 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:42 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.6 kB]
Get:43 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse Translation-en [10.6 kB]
Get:44 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Packages [10.6 kB]
Get:45 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse Translation-en [10.6 kB]
Get:46 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.6 kB]
Get:47 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c=1-f Metadata [1194 kB]
Get:48 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted Translation-en [216 kB]
Get:49 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 kB]
Get:50 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse Translation-en [115 kB]
Get:51 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
Get:52 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c=1-f Metadata [116 kB]
Fetched 10.6 MB in 1s (2.6 MB/s)
Reading package lists... Done
```

```
sudo apt-get update
```

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

```
[root@ip-172-31-93-96 ~]# 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...
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 docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 143 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]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
...
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.
[root@ip-172-31-93-96 ~]#
```

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

```
ubuntu@ip-172-31-93-95:~$ sudo mkdir -p /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
"exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
"exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@ip-172-31-93-95:~$ |
```

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

```
ubuntu@ip-172-31-93-95:~$ 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-172-31-93-95:~$ A|
```

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-172-31-93-95:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /  
ubuntu@ip-172-31-93-95:~$ |
```

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

```
ubuntu@ip-172-31-93-95:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease  
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]  
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 6051 B in 1s (6278 B/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  
Reading state information... 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.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-93-95:~$ |
```

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

```
ubuntu@ip-172-31-93-95:~$ 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:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 143 not upgraded.
Need to get 47.1 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 (47.0 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04-noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)
Preparing to unpack ./runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
```

```
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-172-31-93-95:~$ |
```

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

```

ubuntu@ip-172-31-93-95:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

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

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

```

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

```

ubuntu@ip-172-31-93-95:~$ 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 Fri 2024-09-27 04:57:36 UTC; 400ms ago
       Docs: https://containerd.io
      Main PID: 4992 (containerd)
        Tasks: 6
       Memory: 14.9M (peak: 15.4M)
         CPU: 78ms
        CGroup: /system.slice/containerd.service
                  └─4992 /usr/bin/containerd

Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.325957262Z" level=info msg="serving..." address="/run/containerd/containerd.sock.ttrpc"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326155115Z" level=info msg="serving..." address="/run/containerd/containerd.sock"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326378383Z" level=info msg="Start subscribing containerd event"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326524660Z" level=info msg="Start recovering state"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326717096Z" level=info msg="Start event monitor"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326803827Z" level=info msg="Start snapshots syncer"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326823083Z" level=info msg="Start cni network conf syncer for default"
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.326834082Z" level=info msg="Start streaming server"
Sep 27 04:57:36 ip-172-31-93-95 systemd[1]: Started containerd.service - containerd container runtime.
Sep 27 04:57:36 ip-172-31-93-95 containerd[4992]: time="2024-09-27T04:57:36.330104560Z" level=info msg="containerd successfully booted in 0.049325s"
ubuntu@ip-172-31-93-95:~$ |

```

sudo apt-get install -y socat

```

ubuntu@ip-172-31-93-95:~$ 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 143 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (11.9 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-93-95:~$ |

```

Step 6: Initialize the Kubercluster.

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

```

ubuntu@ip-172-31-93-95:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[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'
W0927 05:00:10.999529    5254 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-93-95 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.93.95]
[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-172-31-93-95 localhost] and IPs [172.31.93.95 127.0.0.1 ::1]
[certs] Generating "etcd-peer" certificate and key

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 172.31.93.95:6443 --token wuhiw8.tqn7cnmejhk5kqey \
  --discovery-token-ca-cert-hash sha256:f9a9d75f6d99fdd71aaaf1b049f75e5ece76e902877e420624f4a305cf4125eb7
ubuntu@ip-172-31-93-95:~$ |

```

TOKEN

[kubeadm join 172.31.93.95:6443 --token wuhiw8.tqn7cnmejhk5kqey \| --discovery-token-ca-cert-hash sha256:f9a9d75f6d99fdd71aaaf1b049f75e5ece76e902877e420624f4a305cf4125eb7](#)

From this command we get token and ca-

```

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-172-31-93-95:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-93-95:~$ |
```

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-172-31-93-95:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-93-95:~$ |
```

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-172-31-93-95:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-93-95:~$ |
```

kubectl get pods

```
ubuntu@ip-172-31-93-95:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-wpb7n   0/1     Pending   0          30s
nginx-deployment-d556bf558-wvkzl   0/1     Pending   0          30s
ubuntu@ip-172-31-93-95:~$ |
```

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

```
ubuntu@ip-172-31-93-95:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-93-95:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-93-95:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-93-95:~$ |
```

Note : We have faced an error as pod status is pending so make it running run below commands

then again run above 2 commands.

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

untainted

kubectl get nodes

```
ubuntu@ip-172-31-93-95:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
kubectl get nodes
error: at least one taint update is required
NAME           STATUS   ROLES      AGE    VERSION
ip-172-31-93-95   Ready    control-plane   20m   v1.31.1
ubuntu@ip-172-31-93-95:~$
```

kubectl get pods

```
ubuntu@ip-172-31-93-95:~$ kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-wpb7n   0/1     Pending   0          19m
nginx-deployment-d556bf558-wvkzl   0/1     Pending   0          19m
ubuntu@ip-172-31-93-95:~$ |
```

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

```
ubuntu@ip-172-31-93-95:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

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

```
ubuntu@ip-172-31-93-95:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Fri, 27 Sep 2024 06:06:41 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
ubuntu@ip-172-31-93-95:~$
```

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: We successfully set up a Kubernetes cluster on AWS EC2, addressing issues related to component setup and residual configurations. We ensured proper cleanup of previous Kubernetes files and mounts, verified the kubelet service, and applied Flannel for networking. Finally, we resolved connectivity issues, and after a thorough review of logs and configuration, we deployed and exposed an NGINX server using Kubernetes services, preparing the cluster for efficient traffic management and scaling.

Experiment 5

Step 1: Download terraform

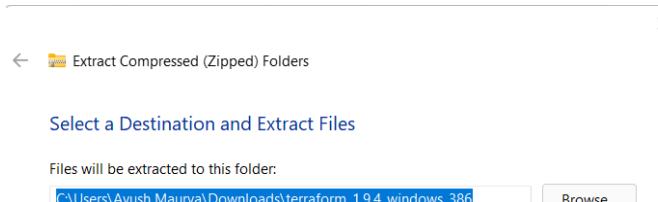
website:<https://www.terraform.io/downloads.html>

The screenshot shows the Terraform website's 'Install Terraform' section for macOS. At the top, there's a navigation bar with links for 'Developer', 'Terraform', 'Install', 'Tutorials', 'Documentation', 'Registry', and 'Try Cloud'. A search bar is also present. The main content area has a heading 'Install Terraform' with a purple logo. Below it, there's a terminal window showing the command: 'brew tap hashicorp/tap' and 'brew install hashicorp/tap/terraform'. Under the heading 'Binary download', there are two options: 'AMD64 Version: 1.9.4' with a 'Download' button, and 'ARM64 Version: 1.9.4' with a 'Download' button. On the right side, there's a sidebar with sections for 'About Terraform', 'Featured docs', and links to 'Introduction to Terraform', 'Configuration Language', 'Terraform CLI', and 'HCP Terraform'.

Step 2: Downlaod Windows Binary download 386

This screenshot shows the 'Install Terraform' section for Windows. The left sidebar lists operating systems: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, Solaris, Release information, and Next steps. The main content area has a terminal window with the command 'brew install hashicorp/tap/terraform'. Below it, under 'Binary download', there are two options: 'AMD64 Version: 1.9.4' with a 'Download' button, and 'ARM64 Version: 1.9.4' with a 'Download' button. In the bottom right corner of the main content area, there's a small note: 'Please click here to learn more about the 386 binary download'.

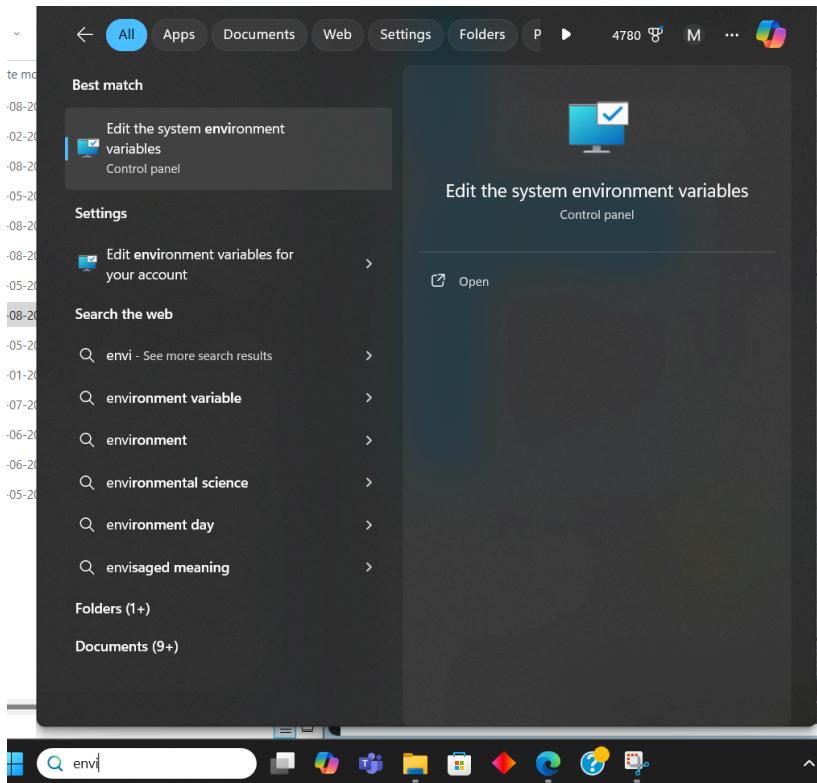
Step 3: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



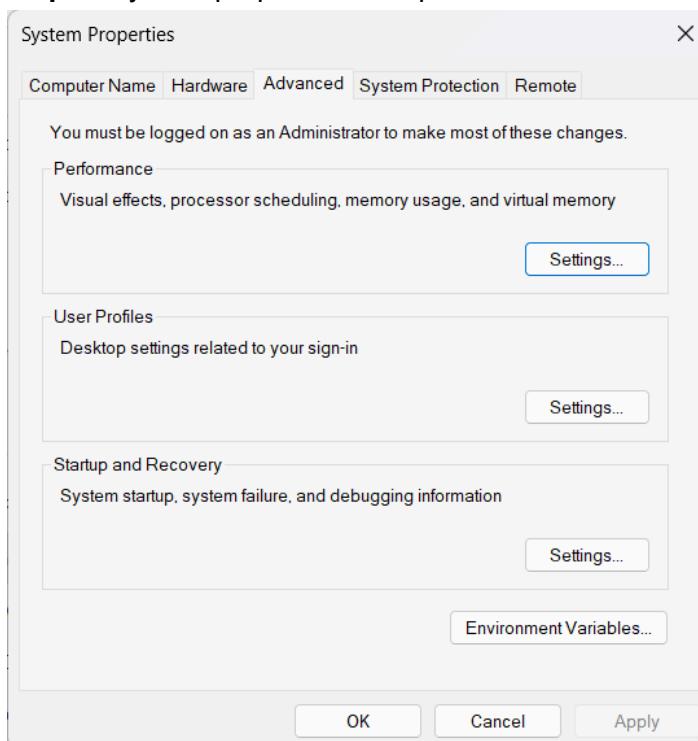
Step 4: Created folder in C: drive for terraform,

Name	Date modified	Type	Size
CFRBACKUP-HKTEXUEZ	12-08-2024 13:56	File folder	
LOGS	04-02-2024 16:03	File folder	
master_save	10-08-2024 10:53	File folder	
PerfLogs	07-05-2022 10:54	File folder	
Program Files	12-08-2024 13:48	File folder	
Program Files (x86)	09-08-2024 23:52	File folder	
temp	04-05-2024 10:52	File folder	
terraform	12-08-2024 14:27	File folder	
tmp	28-05-2024 17:59	File folder	
Users	03-01-2024 05:15	File folder	
Windows	10-07-2024 22:35	File folder	

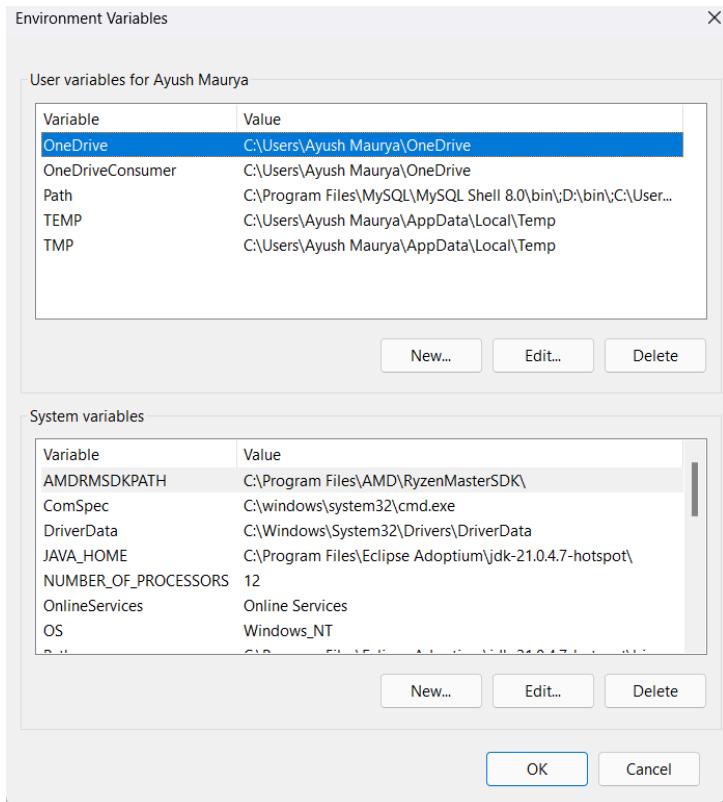
Step 5: Set the System path for Terraform in Environment Variable. Open Environment Variable.



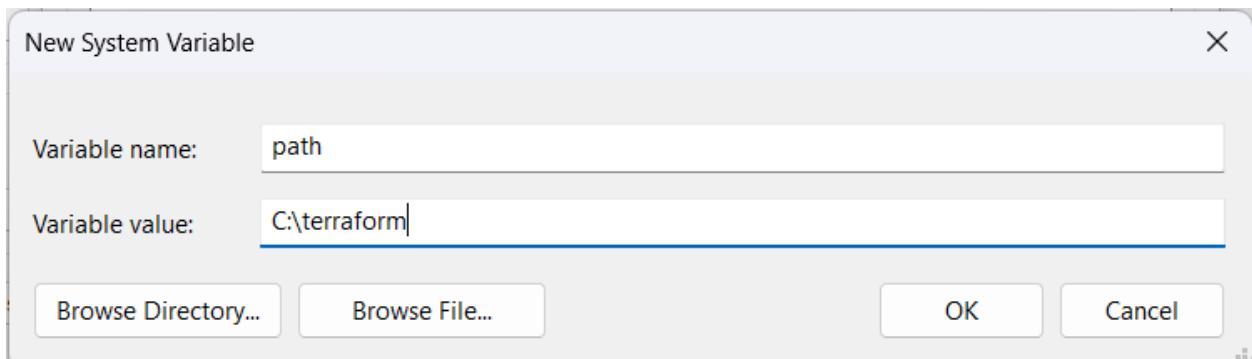
Step 6: System properties will open. Now click on Environment Variables On bottom.



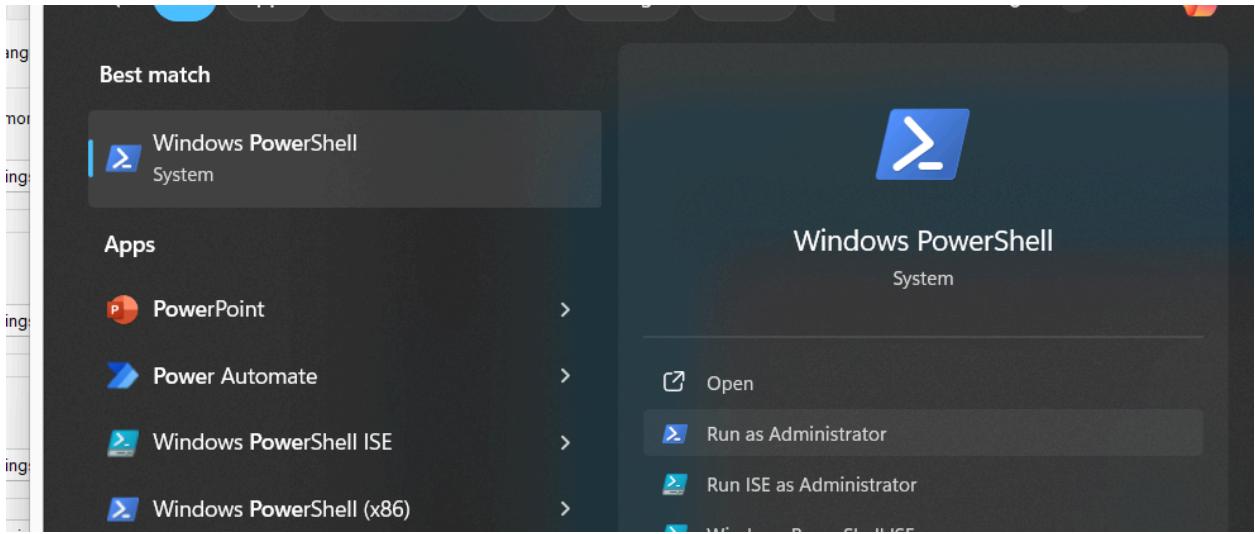
Step 7: Click on 'New' to create new system variables.



Step 8: Give variable name as 'path' or any and variable value as path of folder in C: drive where terraform is extracted.



Step 9: Open PowerShell with Admin Access



Step 10: Open Terraform in PowerShell and check its functionality

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

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

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

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

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

Experiment 6

A. Creating docker image using terraform

Prerequisite:

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

Step 1: Check the docker functionality

The screenshot shows a Windows Command Prompt window titled "Command Prompt". It displays the Docker help output and the result of the "docker --version" command.

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

C:\Users\Ayush Maurya>docker
Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

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

Management Commands:
  builder  Manage builds
  buildx* Docker Buildx
  checkpoint Manage checkpoints
  compose* Docker Compose
  container Manage containers
  context   Manage contexts
  debug*   Get a shell into any image or container
  desktop* Docker Desktop commands (Alpha)
  dev*    Docker Dev Environments
  extension* Manages Docker extensions
  feedback* Provide feedback, right in your terminal!
  image    Manage images
  init*   Creates Docker-related starter files for your project
  manifest Manage Docker image manifests and manifest lists
  network  Manage networks
  plugin   Manage plugins
  sbom*   View the packaged-based Software Bill Of Materials (SBOM) for an image

C:\Users\Ayush Maurya>docker --version
Docker version 27.0.3, build 7d4bcd8

C:\Users\Ayush Maurya>
```

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

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

Script:

terraform

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

```

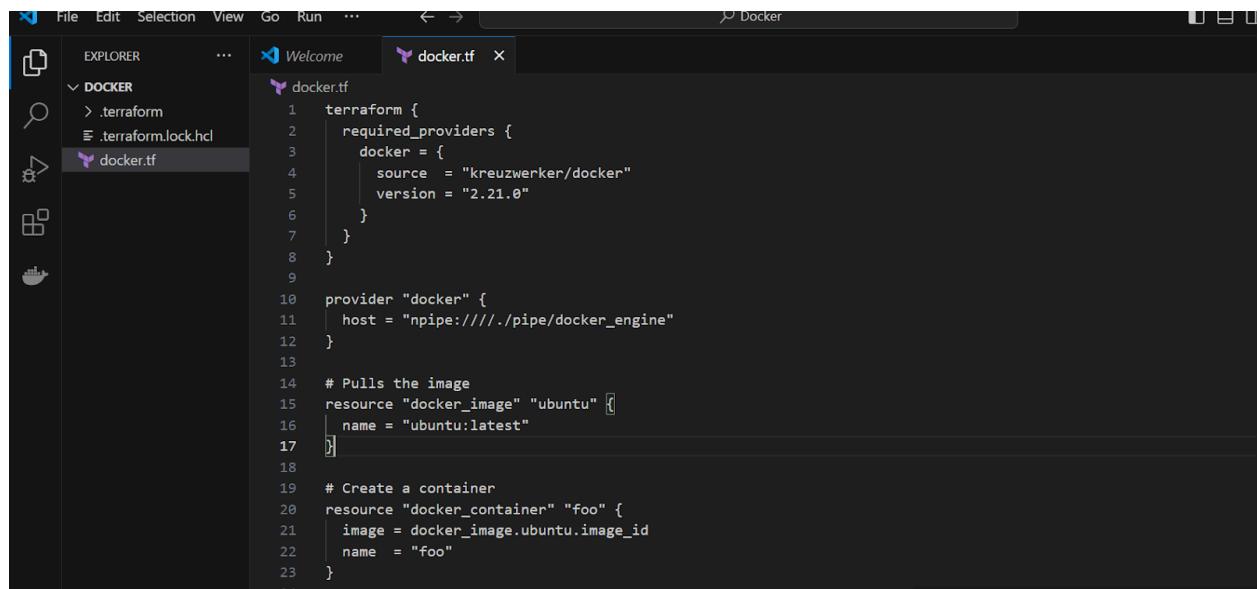
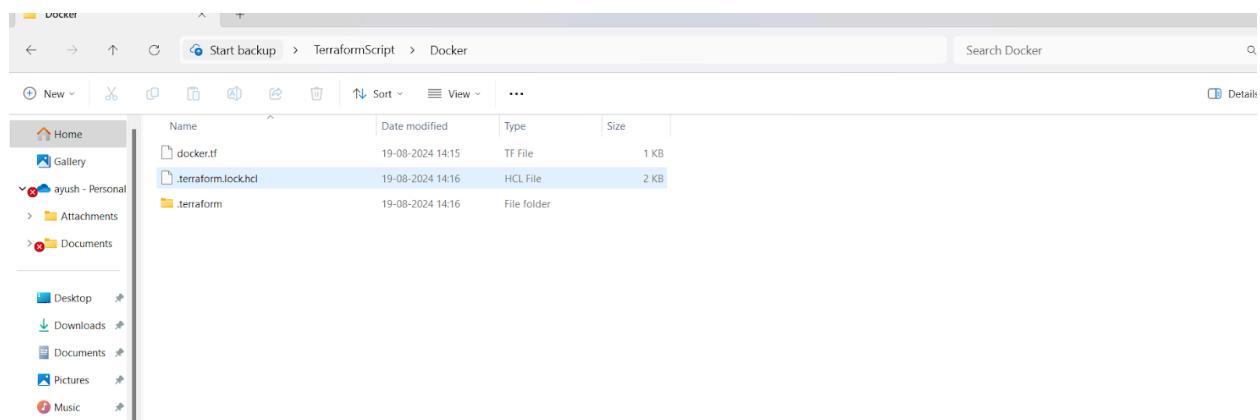
}

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

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

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

```



Step 3: Execute Terraform Init command to initialize the resources

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

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

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

```

+ start      = true
+ stdin_open = false
+ stop_signal = (known after apply)
+ stop_timeout = (known after apply)
+ tty         = false

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

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

```

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

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

```

PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Dockers> terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach      = false
  + bridge      = (known after apply)
  + command     = (known after apply)
  + container_logs = (known after apply)
  + entrypoint  = (known after apply)
  + env         = (known after apply)
  + exit_code   = (known after apply)
  + gateway     = (known after apply)
  + hostname    = (known after apply)
  + id          = (known after apply)
  + image       = (known after apply)
  + init        = (known after apply)
  + ip_address  = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode   = (known after apply)
  + log_driver  = (known after apply)
  + logs        = false
  + must_run    = true
  + name        = "foo"
  + network_data = (known after apply)
  + read_only   = false
  + remove_volumes = true
  + restart    = "no"
  + rm         = false
  + runtime     = (known after apply)

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

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

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

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

Enter a value: yes

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

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

```

Docker images, Before Executing Apply step:

```

C:\Users\Ayush Maurya>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
react-img      latest    d8b8903ee063   8 days ago   320MB

```

Docker images, After Executing Apply step:

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
react-img       latest       d8b8903ee063   8 days ago    320MB
ubuntu          latest       edbfe74c1f8a...   2 weeks ago   78.1MB
```

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

```
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c1f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=601782417e17994f75628bb648b3e4cbbb989037e3a3d302303c4a675a39599b]

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                 = [
    - "+sh",
    - "-c",
    - "-while true; do sleep 1; done",
  ] -> null
  - cpu_shares              = 0 -> null
  - dns                      = [] -> null
  - dns_opts                 = [] -> null
  - dns_search               = [] -> null
  - entrypoint               = [] -> null
  - env                      = [] -> null
  - gateway                 = "172.17.0.1" -> null
  - group_add                = [] -> null
  - hostname                 = "601782417e17" -> null
  - id                       = "601782417e17994f75628bb648b3e4cbbb989037e3a3d302303c4a675a39599b" -> null
  - image                     = "sha256:edbfe74c1f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - init                     = false -> null
  - ip_address               = "172.17.0.2" -> null
  - ip_prefix_length         = 16 -> null
  - ipc_mode                 = "private" -> null
  - links                    = [] -> null
  - log_driver               = "json-file" -> null
  - log_opts                 = {} -> null
  - logs                     = false -> null
  - max_retry_count          = 0 -> null
  - memory                   = 0 -> null
  - memory_swap              = 0 -> null
  - must_run                 = true -> null
}
```

```

- rm           = false -> null
- runtime     = "runc" -> null
- security_opts = [] -> null
- shm_size    = 64 -> null
- start        = true -> null
- stdio_open   = false -> null
- stop_timeout = 0 -> null
- storage_opts = {} -> null
- sysctls      = {} -> null
- tmpfs        = {} -> null
- tty          = false -> null
# (8 unchanged attributes hidden)
}

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

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

```

Docker images After Executing Destroy step

```

Destroy complete! Resources: 2 destroyed.
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
react-img latest d8b8903ee063 8 days ago 320MB
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> | Copilot

```

Experiment 7

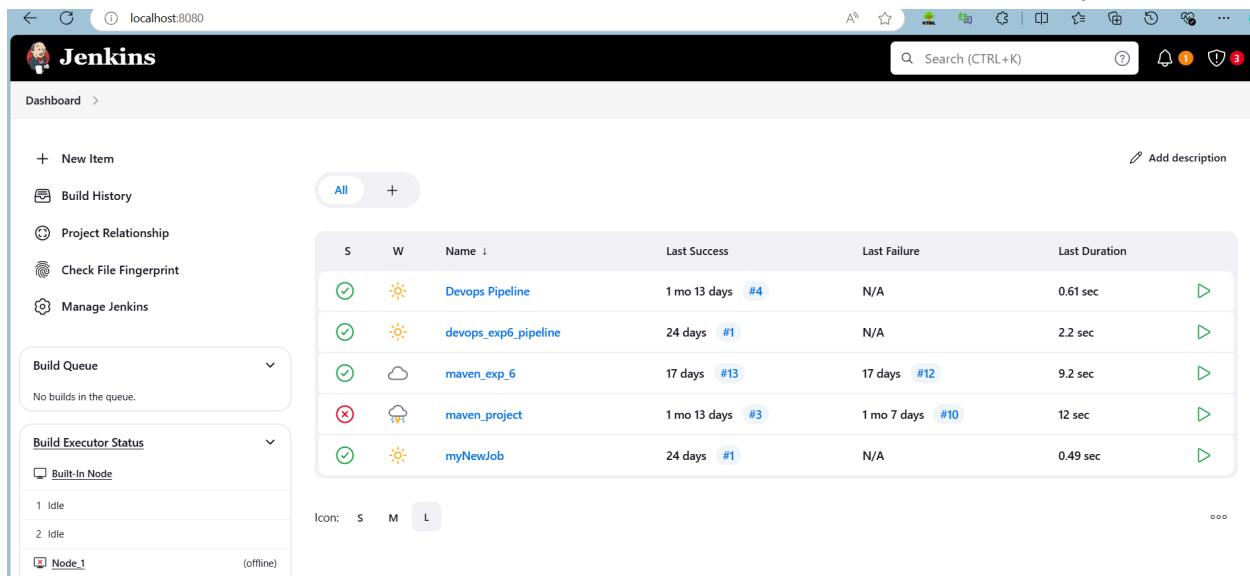
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

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



The screenshot shows the Jenkins dashboard at localhost:8080. On the left, there's a sidebar with links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. Below that are sections for 'Build Queue' (empty), 'Build Executor Status' (1 idle, 2 idle, 1 offline), and 'Cloud Bees' (Node_1). The main area displays a table of build jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Devops Pipeline	1 mo 13 days #4	N/A	0.61 sec
✓	☀️	devops_exp6_pipeline	24 days #1	N/A	2.2 sec
✓	☁️	maven_exp_6	17 days #13	17 days #12	9.2 sec
✗	☁️	maven_project	1 mo 13 days #3	1 mo 7 days #10	12 sec
✓	☀️	myNewJob	24 days #1	N/A	0.49 sec

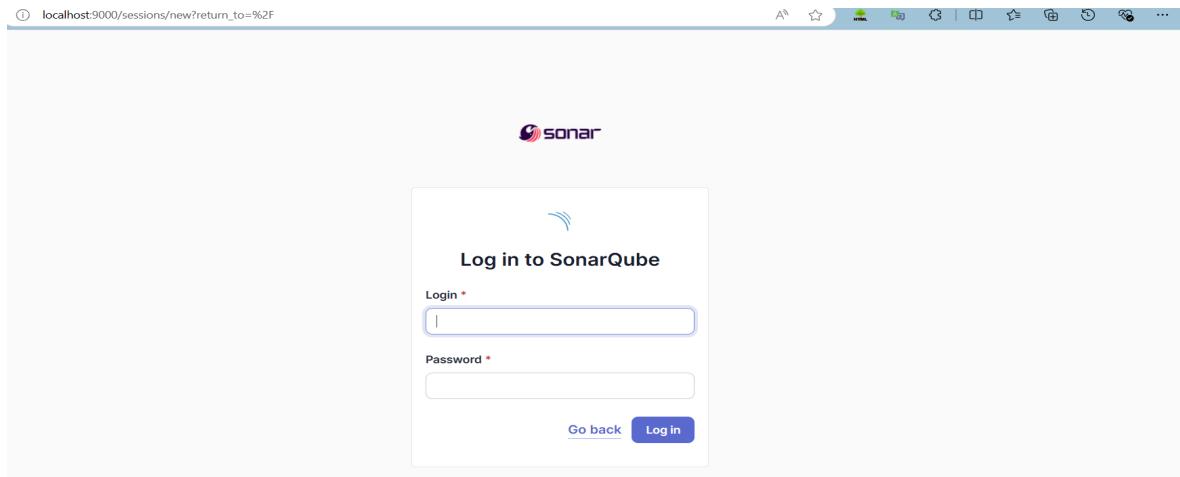
2. Run SonarQube in a Docker container using this command -

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

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

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Ayush Maurya> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59bed36c86
Status: Downloaded newer image for sonarqube:latest
4af48468290f95b22362652ee37b96c935b0bed754945c62cf3b0d51a2ac0c
PS C:\Users\Ayush Maurya>
```

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.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

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

Import from Azure DevOps Import from Bitbucket Cloud Import from Bitbucket Server
Import from GitHub Import from GitLab

Create a local project

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

1 of 2

Create a local project

Project display name *

 ✓

Project key *

 ✓

Main branch name *

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

Cancel Next

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. Follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

Use the global setting

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will be considered new code.
Recommended for projects following continuous delivery.

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. A search bar at the top contains the text 'sonarq'. Below the search bar, a table lists the 'SonarQube Scanner' plugin. The table has columns for 'Install', 'Name', and 'Released'. The 'SonarQube Scanner' entry shows 'Install' and 'Name' as 'SonarQube Scanner 2.17.2' and 'Released' as '6 mo 29 days ago'. A description below the table states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

Below the main table, there's a sidebar with links: 'Updates' (25), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. At the bottom of the sidebar, 'Download progress' is selected. The main area shows 'Download progress' for the SonarQube Scanner plugin, indicating 'Preparation' (Success) and 'SonarQube Scanner' (Success). It also shows 'Loading plugin extensions' (Success).

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

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me
adv_devops_7_sonarqube

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

The screenshot shows the 'SonarQube servers' configuration page. It includes sections for 'Environment variables' (unchecked), 'SonarQube installations' (list of installations), 'Name' (set to 'adv_devops_7_sonarqube'), 'Server URL' (set to 'https://localhost:9000'), and 'Server authentication token' (dropdown set to '- none -'). There is also an 'Advanced' button.

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

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. It lists several tool categories: 'Add Git', 'Gradle installations', 'SonarScanner for MSBuild installations', 'SonarQube Scanner installations', and 'Ant installations'. Under 'SonarQube Scanner installations', there is a button 'Add SonarQube Scanner'. A modal window is open for adding a new SonarQube Scanner configuration. It has fields for 'Name' (set to 'sonarqube_exp7') and 'Install automatically' (checkbox checked). Below these, there is a sub-section for 'Install from Maven Central' with a 'Version' field set to 'SonarQube Scanner 6.1.0.4477'. At the bottom of the modal is an 'Add Installer' button.

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

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name
sonarqube_exp7

Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.1.0.4477

Add Installer ▾

Add SonarQube Scanner

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

The screenshot shows the Jenkins 'New Item' dialog. In the 'Project Name' field, 'adv_devops_exp7' is entered. Below it, a note says '» Required field'. A list of project types is shown: 'Freestyle project' (selected), 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder'. Each item has a brief description. At the bottom left is an 'OK' button, and at the bottom right is a 'Branch Pipeline' button with a note: 'Creates a pipeline job for each branch of Pipeline projects according to detected branches in one SCM repository.'

9. Choose this GitHub repository in Source Code Management.

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

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

Source Code Management



10. Under **Select project → Configuration → Build steps → Execute SonarQube Scanner**, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

The screenshot shows the Jenkins configuration interface under 'Configure'. The 'Build Environment' section is selected. A dropdown menu for 'Build Steps' is open, showing options like 'Execute SonarQube Scanner', 'Execute Windows batch command', etc. The 'Execute SonarQube Scanner' option is highlighted. At the bottom of the dropdown, there is a 'Post-build Actions' section with an 'Add build step ^' button.

Execute SonarQube Scanner

JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?
[Empty input field]

Analysis properties ?
sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.sources=.

Additional arguments ?
[Empty input field]

JVM Options ?
[Empty input field]

Then save

Status

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

adv_devops_exp7

Add description

Disable Project

Permalinks

- Last build (#2), 1 day 20 hr ago
- Last stable build (#2), 1 day 20 hr ago
- Last successful build (#2), 1 day 20 hr ago
- Last completed build (#2), 1 day 20 hr ago

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

sonarQube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Administration

Configuration Security Projects System Marketplace

Global Permissions

Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

All Users Groups Search for users or groups...

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

4 of 4 shown

IF CONSOLE OUTPUT FAILED:

Step 1: Generate a New Authentication Token in SonarQube

1. Login to SonarQube:

- Open your browser and go to `http://localhost:9000`.
- Log in with your admin credentials (default username is `admin`, and the password is either `admin` or your custom password if it was changed).

2. Generate a New Token:

- Click on your `username` in the top-right corner of the SonarQube dashboard.
- Select **My Account** from the dropdown menu.
- Go to the **Security** tab.
- Under **Generate Tokens**, type a name for the token (e.g., "Jenkins-SonarQube").
- Click **Generate**.
- Copy the token and save it securely. You will need it in Jenkins.

Step 2: Update the Token in Jenkins

1. Go to Jenkins Dashboard:

- Open Jenkins and log in with your credentials.

2. Configure the Jenkins Job:

- Go to the job that is running the SonarQube scanner (`adv_devops_exp7`).
- Click **Configure**.

3. Update the SonarQube Token:

- In the SonarQube analysis configuration (either in the pipeline script or under "Build" section, depending on your job type), update the `sonar.login` parameter with the new token.

The screenshot shows the Jenkins configuration page for the 'Execute SonarQube Scanner' build step. It includes fields for JDK selection, path to project properties, analysis properties (containing SonarQube command-line parameters), additional arguments, and JVM options.

Execute SonarQube Scanner

JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
-Dsonar.login=sqp_568834b7b5e77a92843e4b3072e044643ce921c1
sonar.sources=.
```

Additional arguments ?

JVM Options ?

12. Run the Jenkins build.

The screenshot shows the Jenkins project page for 'adv_devops_exp7'. At the top, there are links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'SonarQube', and 'Rename'. On the right, there are buttons for 'Add description' and 'Disable Project'. Below these, the 'SonarQube' icon is present. A section titled 'Permalinks' lists recent builds: Last build (#10), 19 sec ago; Last stable build (#10), 19 sec ago; Last successful build (#10), 19 sec ago; Last failed build (#8), 22 min ago; Last unsuccessful build (#8), 22 min ago; and Last completed build (#10), 19 sec ago. A 'Build History' card for build #10 is shown, indicating it was run on Sep 18, 2024, at 2:36 PM.

Check the console Output

The screenshot shows the Jenkins build console output for build #10. The left sidebar includes links for 'Status', 'Changes', 'Console Output' (which is selected), 'View as plain text', 'Edit Build Information', 'Delete build #10', and 'Timings'. The main content area displays the command-line output of the build process, which includes cloning the repository from GitHub and fetching upstream changes.

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

The screenshot shows the SonarQube project page for the 'main' branch of 'adv_devops_7.sonarqube'. The top navigation bar includes links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search bar. The main content area shows a green 'Passed' status with a checkmark icon. It includes a message about keeping code clean and a button to 'Take the Tour'. Below this, there are tabs for 'New Code' and 'Overall Code'. At the bottom, there are sections for 'Security', 'Reliability', and 'Maintainability'. A note indicates that the last analysis was 5 minutes ago.

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

Conclusion:

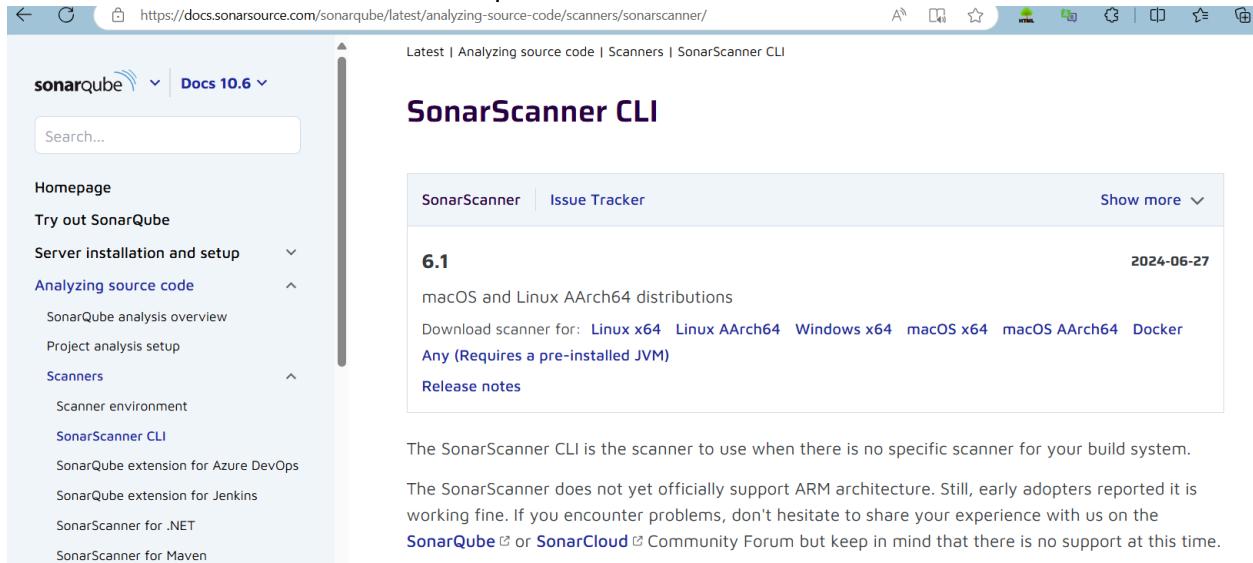
This integration has successfully built a powerful, automated system for enhancing code security and quality. By continuously scanning for vulnerabilities, code smells, and other potential issues, it ensures proactive maintenance of code standards. The seamless connection to GitHub facilitates easy tracking of changes and instant feedback. Automated reports provide valuable insights, allowing developers to address problems early in the development cycle. This streamlined process enhances both efficiency and security. As a result, the workflow becomes more reliable, with improved overall code integrity. Continuous improvement is ensured through consistent monitoring and analysis.

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.

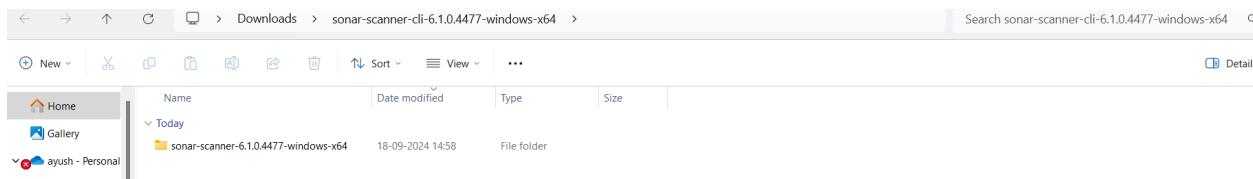
Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>
Visit this link and download the sonarqube scanner CLI.



The screenshot shows the SonarScanner CLI page on the SonarQube documentation site. The left sidebar contains navigation links for SonarQube, Docs 10.6, and various scanner extensions like SonarScanner CLI, SonarQube extension for Azure DevOps, etc. The main content area is titled "SonarScanner CLI" and includes sections for "SonarScanner" and "Issue Tracker". A specific release section for version 6.1 (published 2024-06-27) is highlighted, showing download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker, and Any (Requires a pre-installed JVM). Below this, there are "Release notes" and a note stating that the SonarScanner does not yet officially support ARM architecture.

Extract the downloaded zip file in a folder.



1. Install sonarqube image

Command: **docker pull sonarqube**

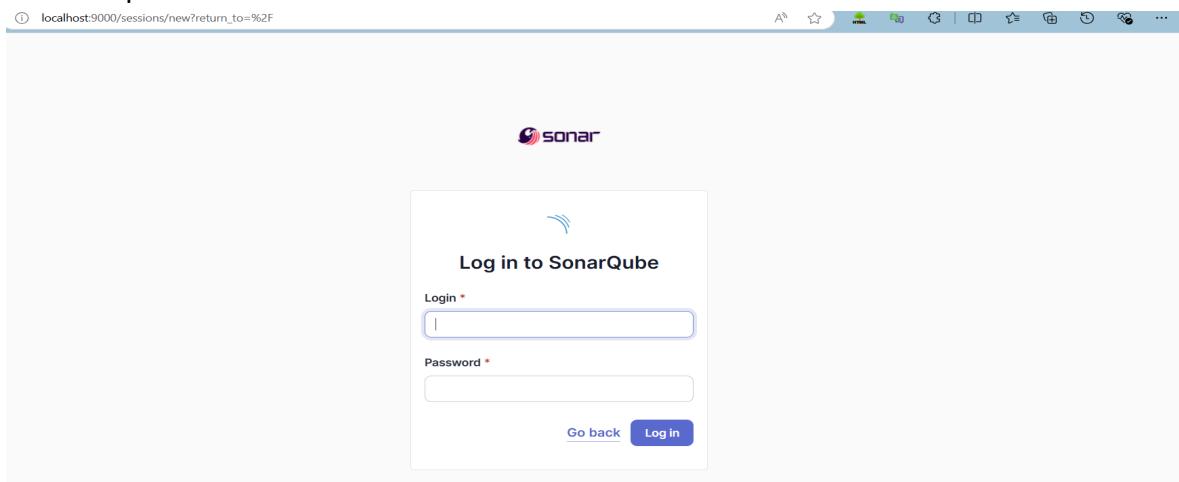
```
C:\Users\Ayush Maurya>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34388537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube

C:\Users\Ayush Maurya>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
dfe47cea99897c70999833dc0d7fa99279ef09d4f3038622910a74df2630afed5
```

2. Once the container is up and running, you can check the status of SonarQube at

localhost port 9000.



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

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

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

Import from Azure DevOps Import from Bitbucket Cloud Import from Bitbucket Server

Import from GitHub Import from GitLab

Create a local project

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

1 of 2

Create a local project

Project display name *

sonarqube

Project key *

sonarqube

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 at 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.

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

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like '+ New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. Below that are sections for 'Build Queue' (empty), 'Build Executor Status' (with one node 'Node_1' listed as idle), and 'Built-In Node' (also empty). The main area displays a table of projects with columns: S (Status), W (Workload), Name, Last Success, Last Failure, and Last Duration. Projects listed include 'Devops Pipeline', 'devops_exp6_pipeline', 'maven_exp_6', 'maven_project', and 'myNewJob'. Each row has a green checkmark icon and a yellow sun icon.

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

The screenshot shows the 'Manage Jenkins > Plugins' page. In the search bar, 'sonarq' is typed. A single result, 'SonarQube Scanner 2.17.2', is shown with an 'Install' button. The plugin description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' Below the search bar, there's a 'Download progress' section with a table showing the preparation step completed successfully. It also includes links to go back to the top page and restart Jenkins.

7. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me
adv_devops_7_sonarqube

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

Name

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

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

[Advanced ▾](#)

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

Dashboard > Manage Jenkins > Tools

Dashboard > Manage Jenkins > Tools

Add Git ▾

Gradle installations

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

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

≡ **SonarQube Scanner**

Name

Install automatically ?

≡ **Install from Maven Central**

Version

SonarQube Scanner 6.2.0.4584

Add Installer ▾

[Add SonarQube Scanner](#)

9. After configuration, create a New Item → choose a pipeline project.

The screenshot shows the Jenkins 'Enter an item name' dialog. The input field contains 'adv_devops_exp8'. Below the input field, there is a list of project types:

- 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. This option is highlighted with a blue border.
- 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.

At the bottom of the dialog are two buttons: 'OK' and 'Cancel Pipeline'.

10. Under Pipeline script, enter the following:

```
node {
stage('Cloning the GitHub Repo') {
    git 'https://github.com/shazforiot/GOL.git'
}

stage('SonarQube analysis') {
    withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
        sh """
            <PATH_TO SONARQUBE SCANNER 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=<SonarQube_URL>(default: http://localhost:9000/)
        """
    }
}
}
```

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

Pipeline

Definition

Pipeline script

```
Script ?  
1~ node {  
2~   stage('Cloning the GitHub Repo') {  
3~     git 'https://github.com/shafzoriot/GOL.git'  
4~   }  
5~  
6~   stage('SonarQube analysis') {  
7~     withSonarQubeEnv('sonarqube') { // Ensure this matches the SonarQube environment name in Jenkins  
8~       bat """  
9~         "C:\\\\Users\\\\Ayush Maurya\\\\Downloads\\\\sonar-scanner-cll-6.1.0.4477-windows-x64\\\\sonar-scanner-6.1.0.4477-windows-x64\\\\bin\\\\sonar-scanner.  
10~        -D sonar.login=admin ^  
11~        -D sonar.password=Ayush2114 ^  
12~        -D sonar.projectKey=sonarqube ^  
13~        -D sonar.exclusions=vendor/**,resources/**/*.java ^  
14~        -D sonar.host.url=http://localhost:9000/  
15~      """  
16~    }  
17~  }  
18~}  
19~
```

11. Build project

Dashboard > adv_devops_exp8 >

Status ✓ adv_devops_exp8

</> Changes

▷ Build Now

⌚ Configure

Delete Pipeline

Full Stage View

SonarQube

Stages

Rename

Pipeline Syntax

Build History trend ▾

Filter... /

#9 Sep 18 16:14 No Changes

#10 Sep 18 16:12 No Changes

#11 Sep 18 16:10 No Changes

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~6min 4s)	3s	40s
#9	2s	6min 2s
#10	2s	1s failed
#11	2s	120ms failed

12. Check console

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#9'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Replay

Pipeline Steps

Workspaces

← Previous Build

Console Output

Skipping 4,246 KB. Full Log

```
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 512. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 248. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 886. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 249. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 662. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 615. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 664. Keep only the first 100 references.  
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 913. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 810. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 668. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 548. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 543. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 152. Keep only the first 100 references.  
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 153. Keep only the first 100 references.
```

13. Now, check the project in SonarQube

The screenshot shows the SonarQube main dashboard for the 'main' project. At the top, there's a green 'Passed' status indicator. Below it, a message says 'The last analysis has warnings. See details'. The dashboard is divided into several sections: Security (0 Open Issues), Reliability (68k Open Issues, A grade), Maintainability (164k Open Issues, A grade), Accepted issues (0), Coverage (Coverage on 0 lines to cover), and Duplications (50.6% on 759k lines). The overall status is 'Passed'.

14. Code Problems

● Consistency

This screenshot shows the 'Issues' page for the 'gameoflife-core' project. It lists several consistency-related issues under the 'Consistency' category. One prominent issue is 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Severity: Bug, Major, 4 years ago). Other listed issues include removing deprecated 'width' and 'align' attributes (Maintainability, html5 obsolete, Major) and using specific version tags for images (Intentionality, No tags, Major).

● Intentionality

This screenshot shows the 'Issues' page for the 'gameoflife-acceptance-tests' project. It lists several intentionality-related issues under the 'Intentionality' category. One prominent issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (Severity: Code Smell, Major, 4 years ago). Other listed issues include using specific version tags for images (Intentionality, No tags, Major) and surrounding variables with double quotes (Intentionality, No tags, Major).

Bugs

Screenshot of a bug reporting interface showing three specific issues:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element.** (Intentionality: accessibility, wcag2-a) Status: Open, Not assigned. Created: L1 2min effort 4 years ago. Type: Bug, Priority: Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency: user-experience) Status: Open, Not assigned. Created: L1 5min effort 4 years ago. Type: Bug, Priority: Major.
- Add "<th>" headers to this "<table>".** (Intentionality: accessibility, wcag2-a) Status: Open, Not assigned. Created: L9 2min effort 4 years ago. Type: Bug, Priority: Major.

Code Smells

Screenshot of a code smell detection interface showing three specific issues:

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

Duplications



- Cyclomatic Complexities

The screenshot shows the SonarQube interface for a project named "gameoflife". The top navigation bar includes links for Overview, Issues, Security Hotspots, Measures (which is the active tab), Code, and Activity. On the right, there are Project Settings and Project Information options. The main content area is titled "Cyclomatic Complexity" with a value of 1,112 and a "See history" link. Below this, a list of files is shown with their respective complexity counts: gameoflife-acceptance-tests (1,112), gameoflife-build (18), gameoflife-core (18), gameoflife-deploy (18), gameoflife-web (1,094), and pom.xml (18). A note at the bottom indicates "6 of 6 shown".

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

Conclusion:

This experiment established a seamless integration of Jenkins with SonarQube to automate code quality assessments within the CI/CD pipeline. SonarQube was deployed using Docker, and after setting up a project, it was configured to analyze the codebase for potential quality issues. Jenkins was configured with the necessary SonarQube plugins, allowing automated code checks through a pipeline that cloned a GitHub repository and performed a SonarQube scan. This integration ensures continuous monitoring throughout the development cycle, effectively identifying and addressing bugs, code smells, and security vulnerabilities, thereby enhancing code quality and security.

Experiment 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.

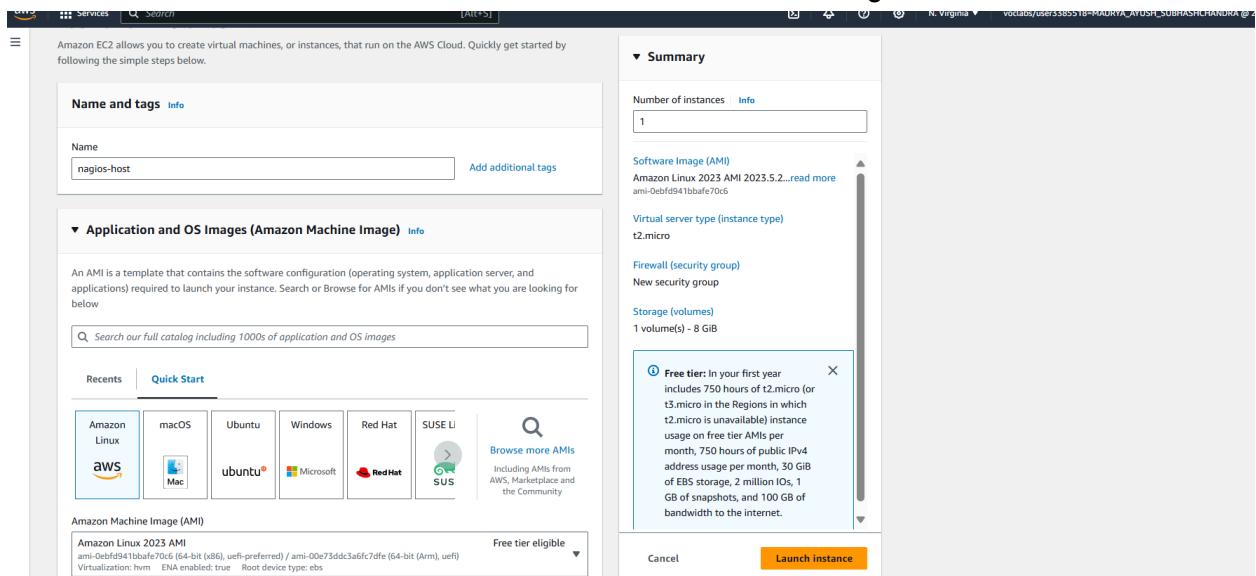
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host



Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour	
On-Demand SUSE base pricing: 0.0116 USD per Hour	
On-Demand RHEL base pricing: 0.026 USD per Hour	
On-Demand Linux base pricing: 0.0116 USD per Hour	

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

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

Key pair name - required

[Create new key pair](#)

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

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

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

- Allow SSH traffic from Anywhere
- Allow HTTPS traffic from the internet
- Allow HTTP traffic from the internet

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

EC2 Dashboard

Instances (1/5) [Info](#)

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
nagios-host	i-0011127bbfdb2f467	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-44-204-11-28.compute-1.amazonaws.com	44.204.11.
Master	i-0c67658f4d6ee8fc	Stopped	t2.micro	2/2 checks passed	View alarms +	us-east-1d	-	-
node1	i-0414d4f92af63c03e	Stopping	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-159-206-1.compute-1.amazonaws.com	54.159.206
node2	i-0d57570061c25ae1	Stopping	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-202-235-83.compute-1.amazonaws.com	44.202.235
exp_4	i-075644ff15b74f611	Stopped	t2.micro	2/2 checks passed	View alarms +	us-east-1d	-	-

i-0011127bbfdb2f467 (nagios-host)

Security details

IAM Role: -

Owner ID: 217253764927

Launch time: Sun Sep 29 2024 12:25:44 GMT+0530 (India Standard Time)

2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.

Security Groups (6) [Info](#)

[Find resources by attribute or tag](#)

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-07053550d576c53e	launch-wizard-2	vpc-0d4c0d8f48c2e4508	launch-wizard-2 created 2024-09-27T...	217253764927
-	sg-030c0a1b62a1e9894	NodeGroup	vpc-0d4c0d8f48c2e4508	Node	217253764927
-	sg-03f412e8ec9ec5946	launch-wizard-1	vpc-0d4c0d8f48c2e4508	launch-wizard-1 created 2024-09-27T...	217253764927
-	sg-000c20590a5551206	default	vpc-0d4c0d8f48c2e4508	default VPC security group	217253764927
-	sg-097fc30a345c1a537	MasterGroup	vpc-0d4c0d8f48c2e4508	Master	217253764927
-	sg-09d51590eb1851b46	launch-wizard-3	vpc-0d4c0d8f48c2e4508	launch-wizard-3 created 2024-09-29T...	217253764927

EC2 > Security Groups > sg-09d51590eb1851b46

sg-09d51590eb1851b46 - launch-wizard-3

[Actions ▾](#)

Details	
Security group name launch-wizard-3	Security group ID sg-09d51590eb1851b46
Owner 217253764927	Description launch-wizard-3 created 2024-09-29T06:49:51.498Z
	VPC ID vpc-0d4c0d8f48c2e4508
Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

[Inbound rules](#) [Outbound rules](#) [Tags](#)

Inbound rules (1)

[C](#) [Manage tags](#) [Edit inbound rules](#)

<input type="checkbox"/> Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/> -	sgr-0ec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-

Edit inbound rules info

Inbound rules control the incoming traffic that's allowed to reach the instance.

[Inbound rules info](#)

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional info
sgr-0ec19557ab9330565	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0 X
-	HTTP	TCP	80	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X
-	HTTPS	TCP	443	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X
-	All traffic	All	All	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	5666	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X
-	All ICMP - IPv4	ICMP	All	Anywhere... ▾	<input type="text"/> 0.0.0.0/0 X

[Add rule](#)

Security group name	Security group ID	Description	VPC ID
launch-wizard-3	sg-09d51590eb1851b46	launch-wizard-3 created 2024-09-29T06:49:51.498Z	vpc-0d4c0d8f48c2e4508
Owner 217253764927	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) [Outbound rules](#) [Tags](#)

Inbound rules (7)

[C](#) [Manage tags](#) [Edit inbound rules](#)

<input type="checkbox"/> Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/> -	sgr-034c500eff5e5fa00	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-038d0d3791dfcc60	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-0e8ad1dd008b14...	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-0ec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-00a0e56d560959f45	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-064c062d69916fa84	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-
<input type="checkbox"/> -	sgr-0613b7b6aa9d30def	IPv4	All traffic	All	All	0.0.0.0/0	-

You have to edit the inbound rules of the specified Security Group for this.

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.

The screenshot shows the 'Connect to instance' page in the AWS Management Console. The instance ID is i-0011127bbfdb2f467 (nagios-host). The 'SSH client' tab is selected. Below it, there are instructions for connecting via SSH:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp_09.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "exp_09.pem"
4. Connect to your instance using its Public DNS:
ec2-44-204-11-28.compute-1.amazonaws.com

Example:
ssh -i "exp_09.pem" ec2-user@ec2-44-204-11-28.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.

Or open command prompt and paste ssh command.

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ayush Maurya>ssh -i "Downloads/exp_09.pem" ec2-user@ec2-44-204-11-28.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-11-28.compute-1.amazonaws.com (44.204.11.28)' can't be established.
ED25519 key fingerprint is SHA256:v2OKH/ezl9iu7/RT6m8LWkgWzEJnnQIqrG9gKZwC14.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-11-28.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

' _#
~\_\####_      Amazon Linux 2023
~~ \_\#####\
~~ \###|
~~ #/ __-->
~~ V~' __->
~~ .-. /_
~~ / _/
~/m'

Last login: Sun Sep 29 07:11:40 2024 from 18.206.107.27
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo yum update

```
[ec2-user@ip-172-31-91-91 ~]$ 
sudo yum update
Last metadata expiration check: 0:19:03 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:29 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023
php8_3           x86_64      8.3.10-1.amzn2023.0.1
=====
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2
apr-util          x86_64      1.6.3-1.amzn2023.0.1
generic-logos-httd noarch     18.0.0-12.amzn2023
httpd-core        x86_64      2.4.62-1.amzn2023
httpd-filesystem noarch     2.4.62-1.amzn2023
httpd-tools       x86_64      2.4.62-1.amzn2023
libbrotli         x86_64      1.0.0-4.amzn2023.0.2
libsodium         x86_64      1.0.19-4.amzn2023
libxml2           x86_64      1.1.3H-5.amzn2023.0.2
mailcap           noarch     2.1.49-3.amzn2023.0.3
nginx-filesystem noarch     1.1.2H-0-1.amzn2023.0.4
php8_3-cli        x86_64      8.3.10-1.amzn2023.0.1
php8_3-common     x86_64      8.3.10-1.amzn2023.0.1
php8_3-process    x86_64      8.3.10-1.amzn2023.0.1
php8_3-xsl        x86_64      8.3.10-1.amzn2023.0.1
=====
Installing weak dependencies:
apr-util-openssl x86_64      1.6.3-1.amzn2023.0.1
mod_http2          x86_64      2.0.27-1.amzn2023.0.3
mod_lua            x86_64      2.4.62-1.amzn2023
php8_3-fpm         x86_64      8.3.10-1.amzn2023.0.1
php8_3-mbstring   x86_64      8.3.10-1.amzn2023.0.1
php8_3-opcache    x86_64      8.3.10-1.amzn2023.0.1
php8_3-pdo         x86_64      8.3.10-1.amzn2023.0.1
php8_3-sodium     x86_64      8.3.10-1.amzn2023.0.1
=====
Transaction Summary
=====
Total download size: 22 MB/s | 10 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : php8_3-common-8.3.10-1.amzn2023.0.1.x86_64 1/25
  Installing : apr-1.7.2-2.amzn2023.0.2.x86_64 2/25
  Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/25
  Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64 4/25
  Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 5/25
  Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch 6/25
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:41 ago on Sun Sep 29 06:56:15 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
gcc              x86_64      11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin-docs      noarch     10.93-1.amzn2023.0.1
annobin-plugin-gcc x86_64      10.93-1.amzn2023.0.1
cpp              x86_64      11.4.1-2.amzn2023.0.2
gc               x86_64      8.0.4-5.amzn2023.0.2
glibc-devel       x86_64      2.34-52.amzn2023.0.11
glibc-headers-x86 noarch     2.34-52.amzn2023.0.11
guile22          x86_64      2.2.7-2.amzn2023.0.3
kernel-headers   x86_64      6.1.18-118.189.amzn2023
libgcc            x86_64      1.2.1-2.amzn2023.0.2
libltool-ltdl    x86_64      2.4.7-1.amzn2023.0.3
libcrypt-devel   x86_64      4.4.33-7.amzn2023
make              x86_64      1:4.3-5.amzn2023.0.2
=====
Transaction Summary
=====
Install 13 Packages
Total download size: 52 M
=====
Installed:
annobin-docs-10.93-1.amzn2023.0.1.noarch
gcc-8.0.4-5.amzn2023.0.2.x86_64
glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
libmpc-1.2.1-2.amzn2023.0.2.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
=====
Complete!
```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:30 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
gd               x86_64      2.3.3-5.amzn2023.0.3
gd-devel          x86_64      2.3.3-5.amzn2023.0.3
=====
Installing dependencies:
brotli           x86_64      1.0.9-4.amzn2023.0.2
brotli-devel     x86_64      1.0.9-4.amzn2023.0.2
bz2ip-devel      x86_64      1.0.8-6.amzn2023.0.2
cairo             x86_64      1.17.6-2.amzn2023.0.1
cmake-filesystem x86_64      3.22.2-1.amzn2023.0.4
fontconfig        x86_64      2.13.94-2.amzn2023.0.2
=====
amazonlinux      139 k
amazonlinux      38 k
amazonlinux      314 k
amazonlinux      31 k
amazonlinux      214 k
amazonlinux      684 k
amazonlinux      16 k
amazonlinux      273 k
```

```

Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
glib2-devel-2.71-768.amzn2023.0.2.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-1.0.6-1.amzn2023.0.1.x86_64
libtracks-core-font-en-3.6-1.amzn2023.0.4.noarch
libX11-6.9.2-3.amzn2023.0.4.x86_64
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXrender-1.9.10-14.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libxcb-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
lible-1.2.1-1.amzn2023.0.1.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-devel-1.8.9-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64
libtiff-4.0.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!

```

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

sudo adduser -m nagios

sudo passwd nagios

(password : ayushmau)

```

[ec2-user@ip-172-31-91-91 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-91-91 ~]$

```

6. Create a new user group

sudo groupadd nagcmd

```

[ec2-user@ip-172-31-91-91 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-91-91 ~]$

```

7. Use these commands so that you don't have to use sudo for Apache and Nagios

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```

[ec2-user@ip-172-31-91-91 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-91-91 ~]$

```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads

```

[ec2-user@ip-172-31-91-91 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-91-91 ~]$

```

9. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ..
[ec2-user@ip-172-31-91-91 ~]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 09:11:59-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.g 100%[=====] 1.97M 5.07MB/s in 0.4s

2024-09-29 09:11:59 (5.07 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-91-91 downloads]$ |
```

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

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ cd ..
[ec2-user@ip-172-31-91-91 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-29 09:14:28-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4 100%[=====] 2.62M 6.92MB/s in 0.4s
```

10. Use tar to unzip and change to that directory.

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
```

11. Run the configuration script with the same group name you previously created.

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

Here we go an error

```
[ec2-user@ip-172-31-91-91 downloads]$ ./configure --with-command-group=nagcmd  
-bash: ./configure: No such file or directory  
[ec2-user@ip-172-31-91-91 downloads]$ |
```

Solution

Navigate to nagios folder in downloads

```
[ec2-user@ip-172-31-91-91 downloads]$ ls  
nagios-4.0.8  nagios-4.0.8.tar.gz  nagios-plugins-2.0.3.tar.gz  
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-4.0.8  
[ec2-user@ip-172-31-91-91 nagios-4.0.8]$ |
```

Error 2: Cannot find SSL headers.

Solution: Install openssl dev library

Steps:

sudo yum install openssl-devel

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo yum install openssl-devel  
Last metadata expiration check: 2:24:05 ago on Sun Sep 29 06:56:15 2024.  
Dependencies resolved.  
=====  
 Package           Arch      Version            Repository      Size  
=====  
 Installing:  
  openssl-devel    x86_64    1:3.0.8-1.amzn2023.0.14  amazonlinux   3.0 M  
  
Transaction Summary  
=====  
Install 1 Package  
  
Total download size: 3.0 M  
Installed size: 4.7 M  
Is this ok [y/N]: y  
Downloading Packages:
```

Now run

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

```
Event Broker: yes  
Install ${prefix}: /usr/local/nagios  
Install ${includedir}: /usr/local/nagios/include/nagios  
Lock file: /run/nagios.lock  
Check result directory: /usr/local/nagios/var/spool/checkresults  
Init directory: /lib/systemd/system  
Apache conf.d directory: /etc/httpd/conf.d  
Mail program: /bin/mail  
Host OS: linux-gnu  
IOBroker Method: epoll  
  
Web Interface Options:  
-----  
    HTML URL: http://localhost/nagios/  
    CGI URL: http://localhost/nagios/cgi-bin/  
Traceroute (used by WAP): /usr/bin/traceroute  
  
Review the options above for accuracy. If they look okay,  
type 'make all' to compile the main program and CGIs.  
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |
```

12. Compile the source code.

make all

```
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
```

13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ make all

sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflo
w=]
  253 |           log_debug_info(DEBUGL_CHECKS, 1, "Found specialized
worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
           | ^~~~~~
~~~~~
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE
```

14. Edit the config file and change the email address.

sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```

# CONTACTS
#
#####
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

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

#####
# CONTACT GROUPS
#
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name   admins
    alias              Nagios Administrators
    members            nagiosadmin
}

```

And change email with your email

15. Configure the web interface.

sudo make install-webconf

```

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$

```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

```

[ec2-user@ip-172-31-91-91 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-172-31-91-91 nagios-4.5.5]$ |

```

Password: Ayushmau

17. Restart Apache

sudo service httpd restart

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |
```

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

```
cd ~/downloads
```

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
```

19. Compile and install plugins

```
cd nagios-plugins-2.4.11
```

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd 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
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
```

```
make
```

```
sudo make install
```

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ make
sudo make install
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/
gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \
< ../build-aux/snippet/c++defs.h \
> c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.\ ifndef/, $p' \
< ../build-aux/snippet/warn-on-use.h \
> warn-on-use.h-t && \
mv warn-on-use.h-t warn-on-use.h
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/GL_ARG_NONNULL/, $p' \
< ../build-aux/snippet/arg-nonnull.h \
> arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
/usr/bin/mkdir -p arpa
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ 
0
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$
```

20. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
```

```
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo chkconfig --add nagio
s
sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with SysV service script with /usr/lib
/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service →
/usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ |
```

Verify the sample configuration files

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

Error

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool/checkresults' is not a valid directory)
  Error processing main config file!
```

Solution:

Create the missing directory: If the directory is missing, create it with the necessary permissions:

```
sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults
```

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$
```

Now run again

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

```
[ec2-user@ip-172-31-91-91 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
```

```
sudo service nagios start
```

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo service nagios start
Starting nagios (via systemctl): [ OK ]
```

21. Check the status of Nagios

```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
  Active: active (running) since Sun 2024-09-29 08:04:30 UTC; 37s ago
    Docs: man:systemd-sysv-generator(8)
 Process: 68037 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
     CPU: 47ms
    CGroup: /system.slice/nagios.service
            └─68059 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─68061 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68062 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68063 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68064 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─68065 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68063;pid=68063
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68062;pid=68062
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68064;pid=68064
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68061;pid=68061
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Warning: Could not open object cache file '/usr/local/nagios/var/objec...
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxp2N...
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Successfully launched command file worker with pid 68065
Sep 29 08:04:39 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxpImg...
Sep 29 08:04:49 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpAf...
Sep 29 08:04:59 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpCtQ...
lines 1-26/26 (END)
```

Error:

The log messages suggest that Nagios is unable to create temporary files, particularly in the directory `/usr/local/nagios/var/`. This is typically caused by permission issues, or the directory might not exist.

Solution:

Firstly check whether `/usr/local/nagios/var/` is there or not. If yes.....

```
ls -ld /usr/local/nagios/var/
```

Change ownership: Set the correct ownership for the Nagios user and group:

```
sudo chown -R nagios:nagcmd /usr/local/nagios/var
```

Set permissions: Ensure the directory has the right permissions:

```
sudo chmod -R 775 /usr/local/nagios/var
```

Restart Nagios: After adjusting the ownership and permissions, restart the Nagios service:

```
sudo systemctl restart nagios
```

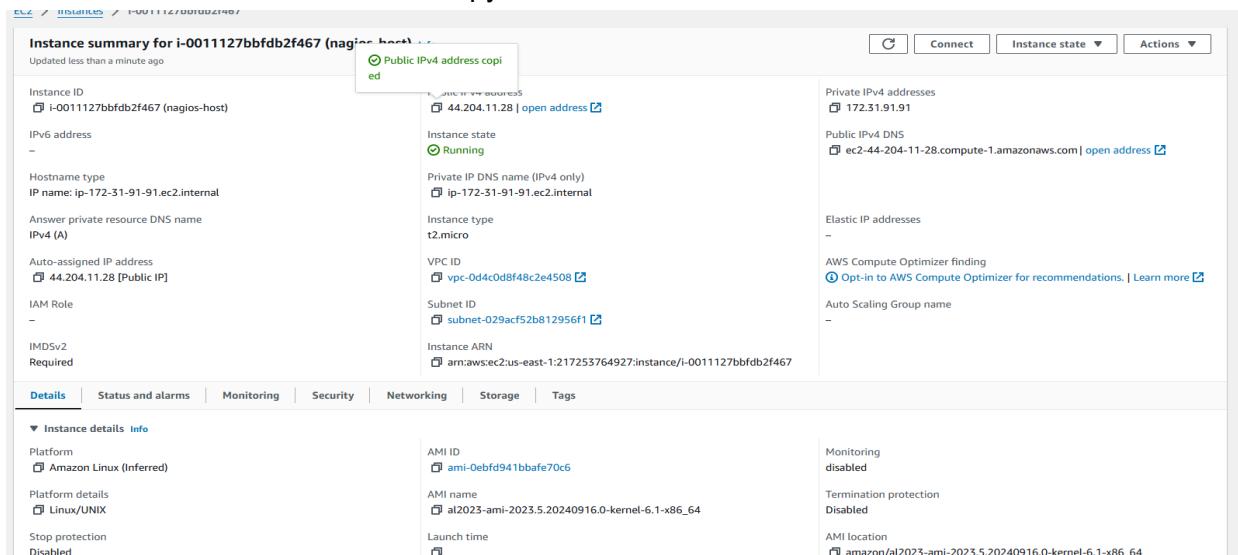
```
drwxr-xr-x. 4 root root 112 Sep 29 08:04 /usr/local/nagios/var/
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagcmd /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chmod -R 775 /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ |
```

Now run again

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-09-29 08:51:47 UTC; 42min ago
    Docs: https://www.nagios.org/documentation
   Tasks: 6 (limit: 1112)
  Memory: 2.9M
     CPU: 562ms
    CGroup: /system.slice/nagios.service
        └─71188 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
          ├─71190 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─71191 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─71192 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─71193 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          └─71194 /usr/local/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71191;pid=71191
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71190;pid=71190
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: Successfully launched command file worker with pid 71194
Sep 29 08:59:22 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes is greater than the configured threshold of 300
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE NOTIFICATION: nagiosadmin;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: NOTIFY job 10 from worker Core Worker 71192 is a non-check helper but exited with return code 1
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
lines 1-25/25 (END)
```

22. Go back to EC2 Console and copy the Public IP address of this instance



23. Open up your browser and look for http://<your_public_ip_address>/nagios

Enter username as nagiosadmin and password which you set in Step 16.

24. After entering the correct credentials, you will see this page.

The screenshot shows the Nagios Core web interface at the URL 44.204.11.28/nagios/. The page title is "Nagios® Core™ Version 4.5.5". A banner at the top right says "✓ Daemon running with PID 71188". The left sidebar has sections for General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Grid, Service Groups, Grid), Problems (Problems, Hosts (Unhandled), Network Outages, Quick Search), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue). The main content area includes a "Get Started" section with links to monitoring infrastructure, a "Quick Links" sidebar with Nagios resources, and two empty boxes for "Latest News" and "Don't Miss...". At the bottom, there is copyright information and a license notice.

This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion:

In this practical, we successfully installed and configured Nagios Core along with Nagios plugins and NRPE on an Amazon EC2 instance. We created a Nagios user, set up necessary permissions, and resolved common installation errors. Finally, we verified the setup by accessing the Nagios web interface, confirming that our monitoring system was fully operational.

Experiment 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).

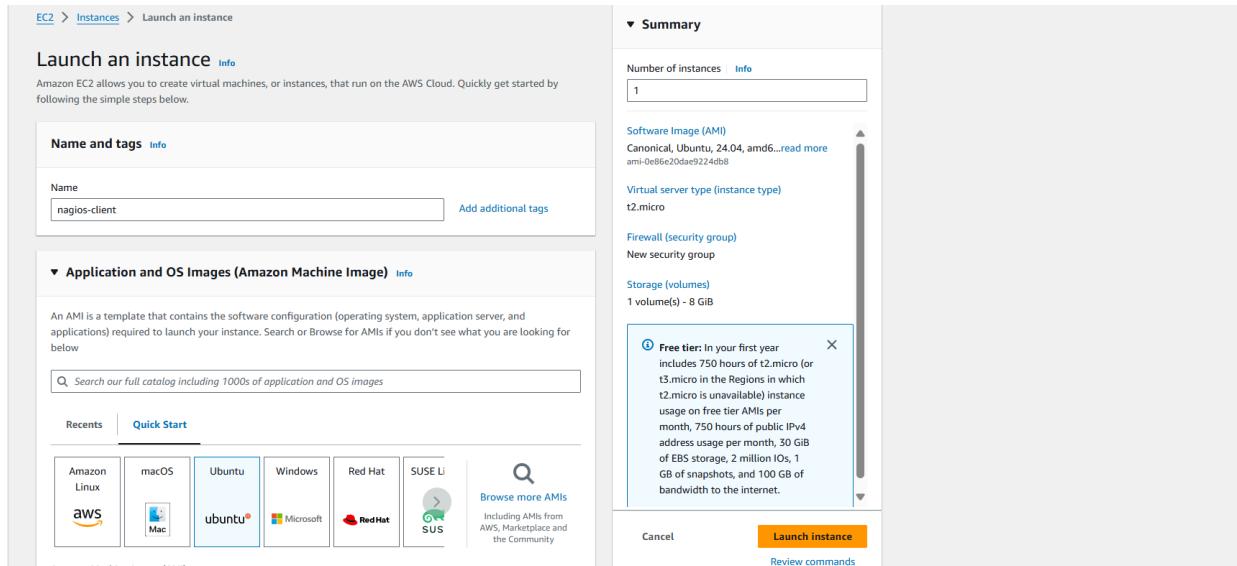
sudo systemctl status nagios

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 16:18:08 UTC; 21min ago
     Docs: https://www.nagios.org/documentation
 Process: 1942 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1944 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1916 (nagios)
   Tasks: 8 (limit: 1112)
    Memory: 7.7M
      CPU: 387ms
     CGroup: /system.slice/nagios.service
             ├─1946 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─1947 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1948 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1950 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─3088 /usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 3000.0,80% -c 5000.0,100% -p 5
             └─3089 /usr/bin/ping -n -U -w 30 -c 5 127.0.0.1

Sep 29 16:18:08 ip-172-31-91-91.ec2.internal systemd[1]: Starting nagios.service - Nagios Core 4.5.5...
Sep 29 16:18:08 ip-172-31-91-91.ec2.internal nagios[1946]: Started nagios.service - Nagios Core 4.5.5.
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE FLAPPING ALERT: localhost;HTTP;STARTED; Service appears to have started flapping (20.0%
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;4;connect to address 127.0.0.1 and port 80: Connecti
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRIT
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: NOTIFY job 2 from worker Core Worker 1948 is a non-check helper but exited with return co
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Lines 1-30/30 (END)
```

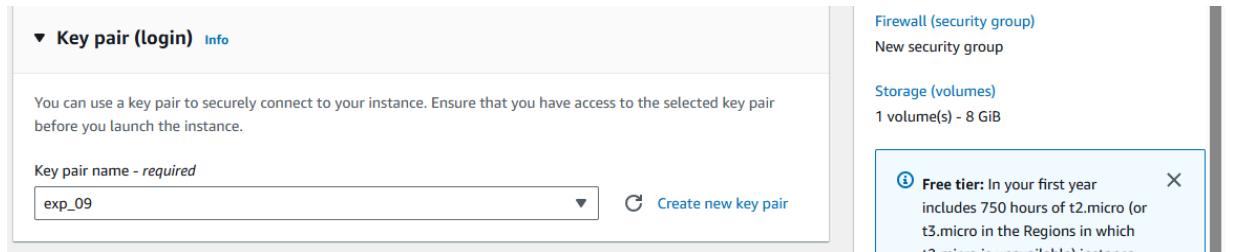
You can now proceed if you get the above message/output.

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

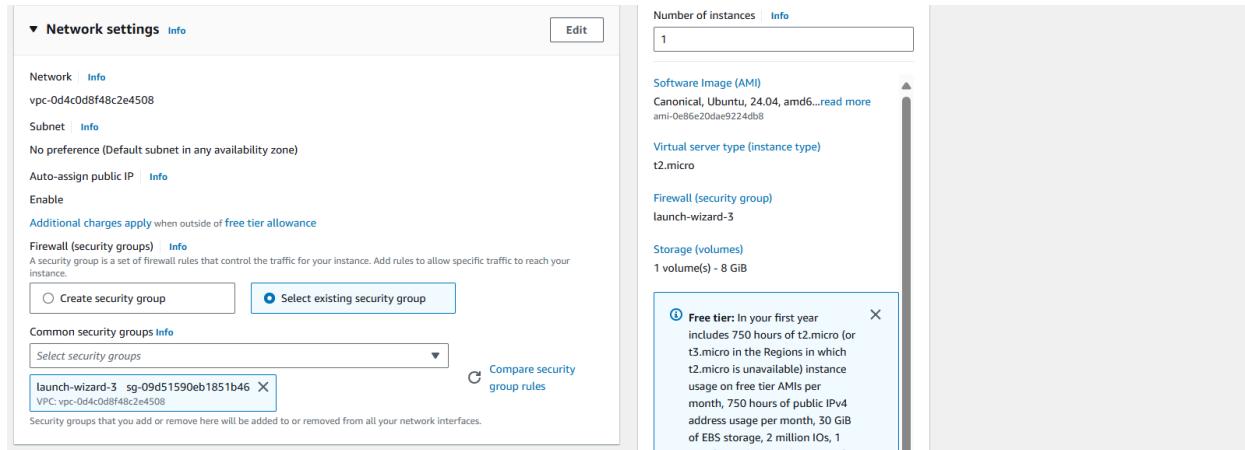


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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



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



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

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

```
PS C:\Users\Ayush Maurya> ssh -i "Downloads/exp_09.pem" ubuntu@ec2-44-206-245-149.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:DT+AA+mKcydh3kOJ2vEpm4ZsA6FL+LM4m1QSImdAHg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-245-149.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
```

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

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

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

ubuntu@ip-172-31-91-146:~$ |
```

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-172-31-91-91 ~]$ ps -ef | grep nagios
nagios 1946 1 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1947 1946 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1948 1946 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1949 1946 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1956 1946 0 16:18 ? 0:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/etc/nagios.cfg
root 3090 3055 0 16:40 pts/0 0:00:00 sudo systemctl status nagios
root 3092 3090 0 16:40 pts/1 0:00:00 sudo systemctl status nagios
root 3093 3092 0 16:40 pts/1 0:00:00 systemctl status nagios
ec2-user 3914 3890 0 16:59 pts/2 0:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-91 ~]$ |
```

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-172-31-91-91 ~]$ sudo su
[root@ip-172-31-91-91 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-91-91 ec2-user]# |
```

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

cp /usr/local/nagios/etc/objects/localhost.cfg

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

```
[root@ip-172-31-91-91 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-91-91 ec2-user]# |
```

Step 7: Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

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

Change hostname to **linuxserver**.

Change address to the **public IP of your Linux client**.

Set hostgroup_name to **linux-servers1**.

```

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

# Define a host for the local machine

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

    host_name      linuxserver
    alias          localhost
    address        172.31.92.146
}

#####
# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines

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

```

Step 8: Now update the Nagios config file .Add the following line in the file. Line to add :
> nano /usr/local/nagios/etc/nagios.cfg

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```

#
# 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

```

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-172-31-91-91 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

```

```

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 timemeperiods
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

```

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

```

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-91-91 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-172-31-92-146:~$ 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/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu/noble-security/main amd64 Packages [380 kB]
Get:8 http://security.ubuntu.com/ubuntu/noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu/noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu/noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Components [18632 B]
Get:14 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu/noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu/noble-security/multiverse Translation-en [2808 B]
Get:20 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Components [208 B]
Get:21 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse Translation-en [118 kB]

```

```

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: sshd[992,1102]
ubuntu @ session #7: sshd[1190,1248]
ubuntu@ip-172-31-92-146:~$ |

```

Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address.

sudo nano /etc/nagios/nrpe.cfg

```
# 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,::1,34.207.68.187

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

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

sudo systemctl restart nagios-nrpe-server

```
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
ubuntu@ip-172-31-92-146:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-146:~$ |
```

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

```
ec2-user@ip-172-31-91-91 ~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-09-29 17:20:07 UTC; 12min ago
       Docs: https://www.nagios.org/documentation
   Process: 4761 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 4762 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 4763 (nagios)
    Tasks: 6 (limit: 1112)
      Memory: 4.1M
        CPU: 234ms
       CGroup: /system.slice/nagios.service
           ├─4763 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─4764 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
           ├─4765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
           ├─4766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
           ├─4767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
           └─4768 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config fil
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/lo
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Successfully launched command file worker with pid 4768
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: SERVICE NOTIFICATION: nagiosadmin@localhost;Swap Usage:CRITICAL;notify-service-by-email;SWAP CRI
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: NOTIFY job 1 from worker Core Worker 4766 is a non-check helper but exited with return co
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: host=localhost; service=Swap Usage; contacts=nagiosadmin
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Lines 1-28/28 (END)
```

sudo systemctl status httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo systemctl start httpd

sudo systemctl enable httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-91-91 ~]$ |
```

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

Eg. <http://34.207.68.187/nagios>

Enter username as **nagiosadmin** and password which you set in Exp 9.

The screenshot shows the Nagios Core dashboard. On the left, there's a vertical navigation bar with sections for General, Current Status, Problems, Reports, and System. The 'Current Status' section is expanded, showing links like 'Tactical Overview', 'Map', 'Hosts', 'Services', 'Host Groups', 'Grid', and 'Service Groups'. The main content area features several boxes: 'Get Started' with a list of monitoring basics; 'Latest News' which is currently empty; 'Don't Miss...' which is also empty; and 'Quick Links' with a list of Nagios resources. The top right corner displays the Nagios logo and the text 'Nagios® Core™ Version 4.5.5 September 17, 2024 Check for updates'. A small green checkmark icon indicates a daemon is running with PID 4763.

Now Click on Hosts from left side panel

This screenshot shows the 'Host Status Details For All Host Groups' page. The left sidebar has a 'Current Status' section with a 'Hosts' link under 'Problems'. The main table lists two hosts: 'linuxserver' and 'localhost', both of which are marked as 'UP'. The table includes columns for 'Status', 'Last Check', 'Duration', and 'Status Information'. Below the table, it says 'Results 1 - 2 of 2 Matching Hosts'.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-29-2024 17:40:07	0d 0h 22m 43s	PING OK - Packet loss = 0%, RTA = 0.56 ms
localhost	UP	09-29-2024 17:40:00	0d 9h 37m 43s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Conclusion:

In this practical, we set up a Nagios host and client to monitor services and server performance on both Linux and Windows servers. We configured Nagios on an Amazon Linux machine to monitor critical services like HTTP, SSH, and system resources, ensuring their availability and health. By creating and configuring a new EC2 instance as the Nagios client, we enabled seamless communication between the client and host for efficient service monitoring. This setup helps ensure uptime and quick detection of issues across the infrastructure.

Experiment 11

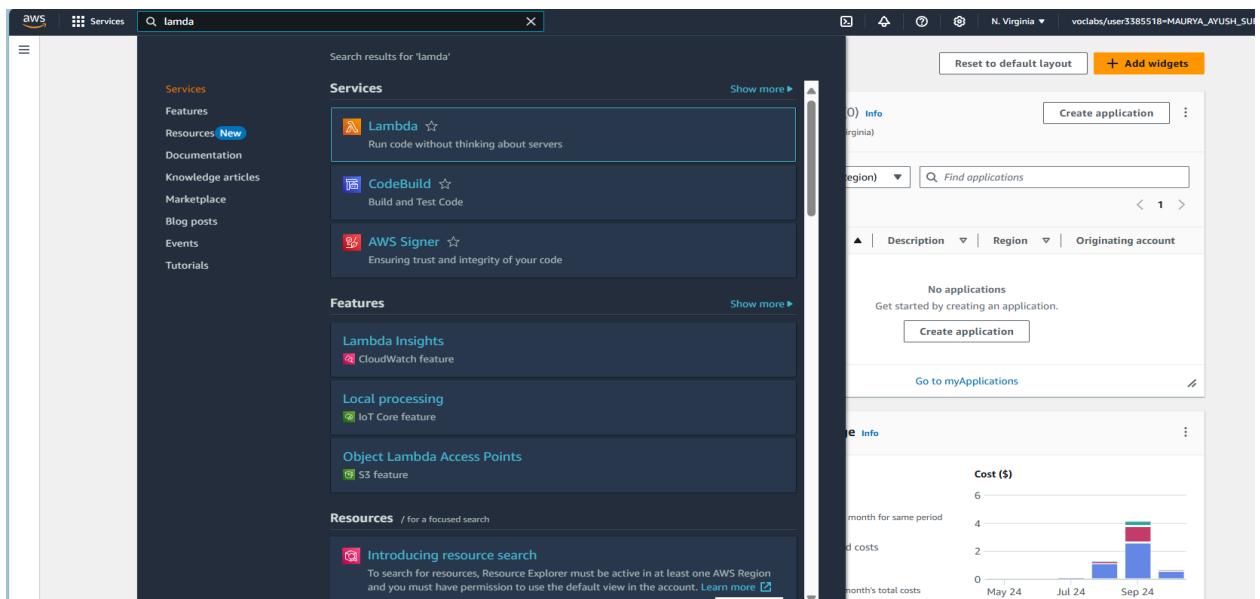
Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

Prerequisites:

- 1) AWS account (academy recommended)

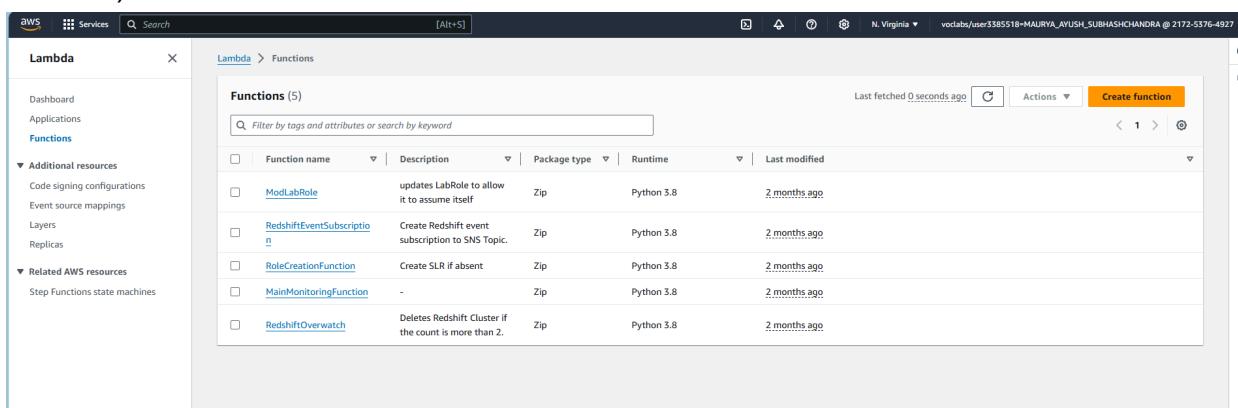
Step 1: Set up AWS Lambda Function

- 1) Search for Lambda in the services tab. Click on it once found.



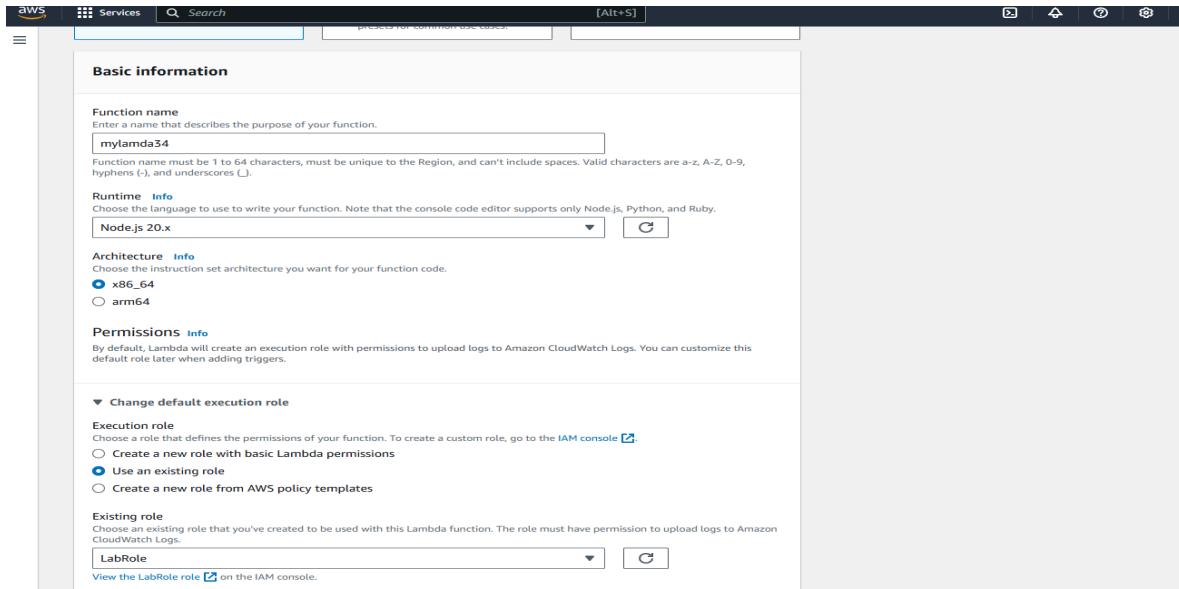
The screenshot shows the AWS Services console with a search bar at the top containing 'lambda'. The left sidebar has a 'Services' section with links like Resources (New), Documentation, and Tutorials. The main area displays search results for 'lambda', with 'Lambda' being the top result under 'Services'. Other results include 'CodeBuild' and 'AWS Signer'. Below the services list is a 'Features' section with 'Lambda Insights', 'Local processing', and 'Object Lambda Access Points'. A note at the bottom says 'Introducing resource search'. On the right side, there's a 'Create application' button and a chart showing costs over time.

- 2) Click on create functions.



The screenshot shows the AWS Lambda Functions page. The left sidebar has sections for Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main area shows a table titled 'Functions (5)' with columns for Function name, Description, Package type, Runtime, and Last modified. The functions listed are 'ModLabRole', 'RedshiftEventSubscription', 'RoleCreationFunction', 'MainMonitoringFunction', and 'RedshiftOverwatch'. At the top right of the table is a 'Create function' button.

3) Give a name to your Lambda function. Select the runtime as Node.js 20.x (You can also use python). Select the architecture as x86_64. Set the default execution role as LabRole if you are doing this on your academy account. (Use an existing role → LabRole)



4) Once the function is created, click on the name of the function.

Function name	Description	Package type	Runtime	Last modified
ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago
RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 months ago
RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 months ago
MainMonitoringFunction	-	Zip	Python 3.8	2 months ago
RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago
mylambda34	-	Zip	Node.js 20.x	in 2 minutes

5) This is the dashboard of our lambda function.

Function overview

- Function ARN: arn:aws:lambda:us-east-1:217253764927:function:mylambda34
- Last modified: in 34 seconds
- Description: -
- Code source: index.js

```

index.js
1 export const handler = async (event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from Lambda!'),
6     };
    
```

6) This function has the following default code, which is used to print “Hello from Lambda!”.

```

Code | Test | Monitor | Configuration | Aliases | Versions
Code source Info
File Edit Find View Go Tools Window Test Deploy
index.js Environment
mylambda34 / index.js
1 export const handler = async (event) => {
2     // ...
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello From Lambda!'),
6     };
7     return response;
8 }

```

Step 2: Set up configurations and test events

1) Just above the test code, you would find Configuration, click on it. Then click on Edit.

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

2) Here, change the Timeout to 1 sec. This is the time for which the function can be running before it is forcibly terminated.

Basic settings

Description - optional

Memory **128** MB

Ephemeral storage **512** MB

SnapStart **None**

Timeout **0 min 1 sec**

Execution role **Use an existing role**

3) We can see the executed changes.

The screenshot shows the AWS Lambda 'Configuration' tab. On the left, a sidebar lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', 'Function URL', 'Environment variables', 'Tags', and 'VPC'. The main panel displays 'General configuration' settings: Description is empty; Memory is set to 128 MB; Timeout is 0 min 1 sec; SnapStart is None; and Ephemeral storage is 512 MB. An 'Edit' button is in the top right corner.

4) Switch back to the code tab. Click on the dropdown arrow near test. Then select configure test event.

The screenshot shows the AWS Lambda 'Code source' tab. The 'Test' tab is selected. A dropdown menu is open, showing options: 'Configure test event' (which is highlighted in blue), 'Ctrl Shift C', and other options like 'Run', 'Deploy', and 'Edit'. The code editor shows the 'index.mjs' file with the following content:

```
1 // export const handler = async (event) => {
2 //   // TODO: Implement
3 //   const response = {
4 //     statusCode: 200,
5 //     body: JSON.stringify('Hello from Lambda!'),
6 //   };
7 //   return response;
8 // }
```

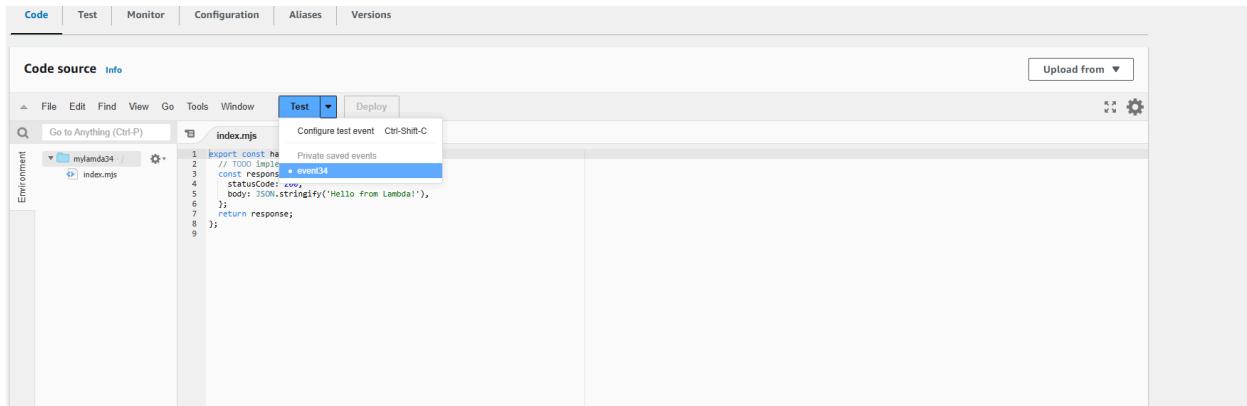
5) Here, create a new event, keep the other options default and save the event.

The screenshot shows the AWS Lambda 'Test' tab. A 'Configure test event' dialog is open. In the 'Test event action' section, 'Create new event' is selected. The 'Event name' field contains 'event34'. In the 'Event sharing settings' section, 'Private' is selected. The 'Template - optional' section shows 'hello-world' selected. The 'Event JSON' section contains the following JSON:

```
1 * []
2 "key1": "value1",
3 "key2": "value2",
4 "key3": "value3"
5 []
```

At the bottom of the dialog, there are 'Cancel', 'Invoke' (disabled), and 'Save' buttons. The 'Save' button is highlighted in orange.

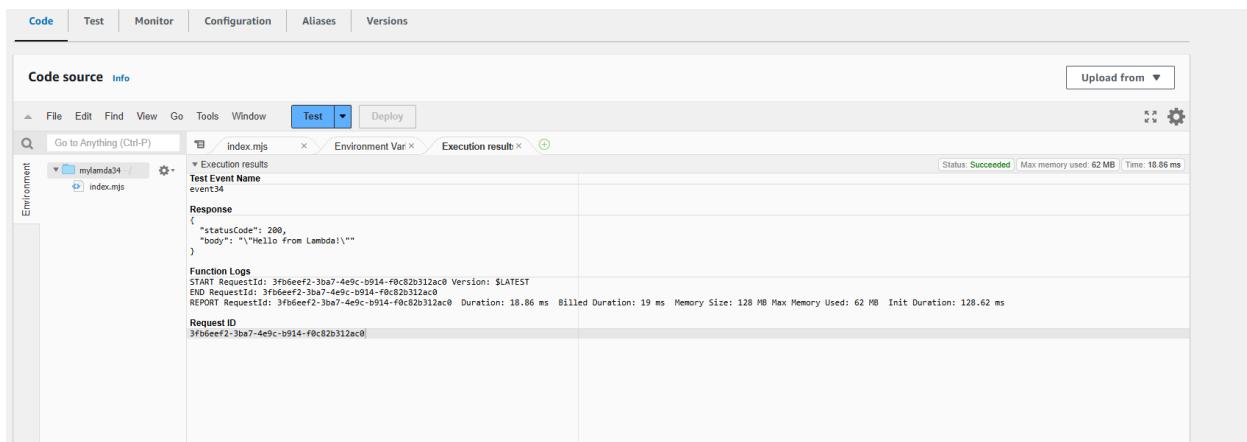
6) Now, again click on the dropdown. This time, select the event you have created. Then, click on TEST.



The screenshot shows the AWS Lambda console interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Test' tab is currently active. Below the navigation, there's a search bar labeled 'Go to Anything (Ctrl-P)' and a sidebar labeled 'Environment'. The main area displays a code editor for a file named 'index.mjs'. A context menu is open over the code, with the 'Test' option highlighted. Other options visible in the menu include 'Configure test event' (with a keyboard shortcut 'Ctrl-Shift-C') and 'Private saved events'. The code in the editor is as follows:

```
1 // export const handler = event => {
2 //   // TODO: Implement
3 //   const response = {
4 //     statusCode: 200,
5 //     body: JSON.stringify('Hello from Lambda!'),
6 //   };
7 //   return response;
8 //};
```

7) We can see the expected output for the sample code.



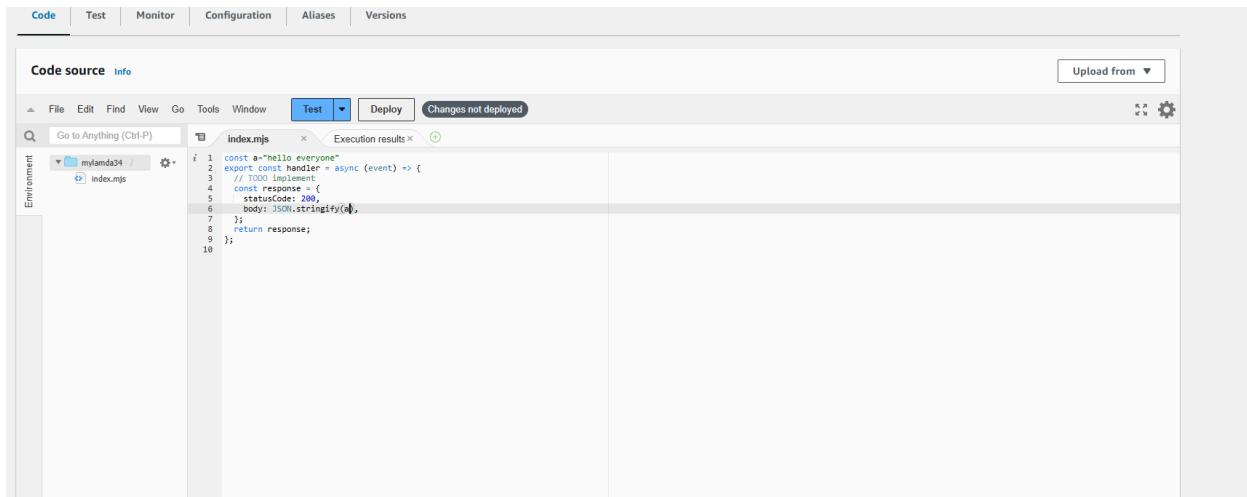
The screenshot shows the AWS Lambda console after a test run. The 'Test' tab is still active. The code editor for 'index.mjs' remains the same. To the right of the editor, there's a section titled 'Execution result' which displays the test results. It shows the 'Test Event Name' as 'event34'. Under the 'Response' section, the output is shown as:

```
{ "statusCode": 200, "body": "Hello from Lambda!" }
```

Below this, the 'Function Logs' section provides detailed log information:

```
START RequestId: 3fb6eeef2-3ba7-4e9c-b914-f0c82b312ac0 Version: $LATEST
END RequestId: 3fb6eeef2-3ba7-4e9c-b914-f0c82b312ac0
REPORT RequestId: 3fb6eeef2-3ba7-4e9c-b914-f0c82b312ac0 Duration: 18.86 ms Billed Duration: 19 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 128.62 ms
Request ID
3fb6eeef2-3ba7-4e9c-b914-f0c82b312ac0
```

8) For a test, declare a string and call it in line 6. After making the changes click on deploy.



The screenshot shows the AWS Lambda console after changes have been made to the code. The 'Test' tab is still active. The code editor now contains the following modified code:

```
i 1 const a="Hello everyone"
2 export const handler = async (event) => {
3   // TODO: Implement
4   const response = {
5     statusCode: 200,
6     body: JSON.stringify(a),
7   };
8   return response;
9 };
10};
```

To the right of the editor, the 'Execution results' section shows the message 'Changes not deployed'.

9) Run the test. We can see that the string we declared has come in the output.

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for 'Code' (selected), 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. Below the tabs, there's a toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (selected), and 'Deploy'. On the left, there's a sidebar titled 'Environment' with a dropdown menu. The main area shows a file tree with 'mylambda34' and 'index.mjs'. Under 'Execution results', there's a list item 'Test Event Name: event34'. The 'Execution result' tab is selected, displaying the following JSON response:

```
Response
{
  "statusCode": 200,
  "body": "Hello everyone!"
}
```

Below the response, the 'Function Logs' section shows the following log entries:

```
START RequestId: 78fe7dbe-a52e-4af6-9417-5a650abda3dd Version: $LATEST
END RequestId: 78fe7dbe-a52e-4af6-9417-5a650abda3dd
REPORT RequestId: 78fe7dbe-a52e-4af6-9417-5a650abda3dd Duration: 1.42 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB
```

The 'Request ID' is listed as 78fe7dbe-a52e-4af6-9417-5a650abda3dd.

Conclusion:

In this experiment, we explored AWS Lambda by creating and configuring Lambda functions using Node.js. We learned to set up a function, adjust configurations like timeout settings, and test it with custom events. This hands-on experience provided us with foundational skills in serverless computing, enabling us to develop scalable applications efficiently. Moving forward, we can investigate integrating AWS Lambda with other services to enhance our serverless applications.

Experiment 12

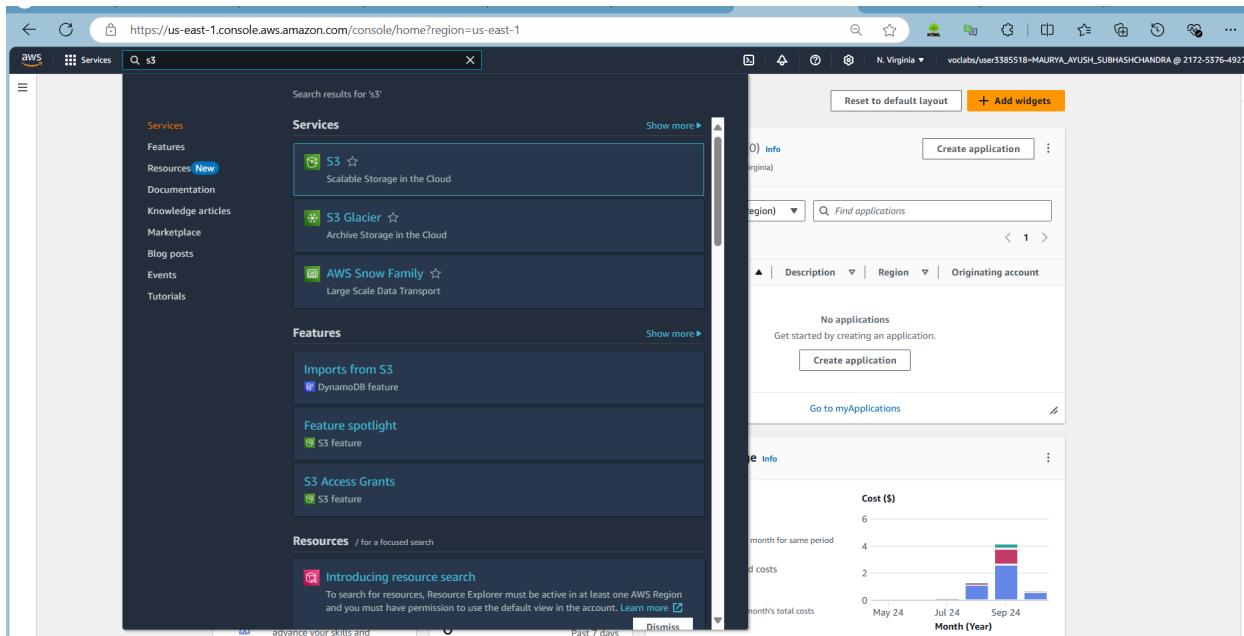
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

Prerequisites:

- 1) AWS account (academy preferable)
- 2) Lambda function (created in the previous experiment).

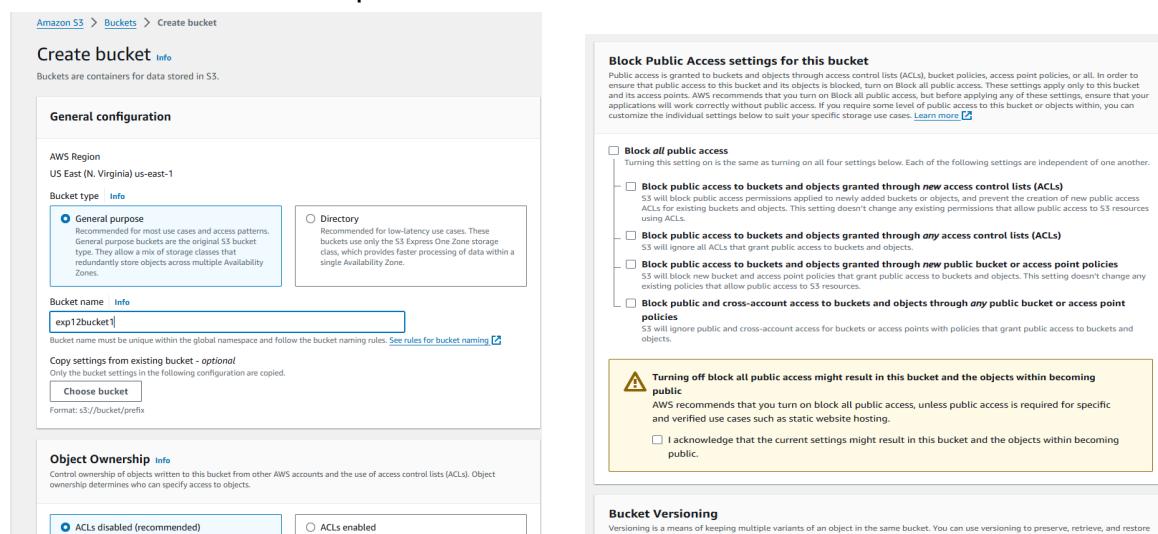
Step 1: Create a s3 bucket.

- 1) Search for S3 bucket in the services search. Then click on create bucket.



The screenshot shows the AWS Services search results for 's3'. The 'S3' service is highlighted and selected. The interface includes a sidebar with 'Services', 'Features', and 'Resources' sections, and a main content area with a bar chart showing costs over time.

- 2) Keep the bucket as a general purpose bucket. Give a name to your bucket.
- 3) Uncheck block all public access.



The screenshot shows the 'Create bucket' wizard. In the 'General configuration' step, the user has chosen 'General purpose' as the bucket type and named it 'exp12bucket1'. Under 'Block Public Access settings for this bucket', the 'Block all public access' checkbox is unchecked. A warning message states: 'Turning off block all public access might result in this bucket and the objects within becoming public.' The user has checked the acknowledgement checkbox.

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

Amazon S3 > Buckets

► Account snapshot - updated every 24 hours [All AWS Regions](#)

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

[View Storage Lens dashboard](#)

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

Buckets are containers for data stored in S3.

Q Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
bucket-aws34	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 5, 2024, 01:15:15 (UTC+05:30)
demo-bucket-i-013642310f50f2f8	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 24, 2024, 14:42:00 (UTC+05:30)

C Copy ARN Empty Delete Create bucket

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

Amazon S3 > Buckets > exp12bucket1 [Info](#)

Objects [Info](#)

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

Q Find objects by prefix

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

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

Upload

Amazon S3 > Buckets > exp12bucket1 > Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder.

Files and folders (1 Total, 4.1 MB) Remove Add files Add folder

All files and folders in this table will be uploaded.

Q Find by name

Name	Folder
<input type="checkbox"/> cube.png	-

Destination [Info](#)

Destination [s3://exp12bucket1](#)

► Destination details Bucket settings that impact new objects stored in the specified destination.

► Permissions Grant public access and access to other AWS accounts.

► Properties Specify storage class, encryption settings, tags, and more.

5) The image has been uploaded to the bucket.

The screenshot shows the AWS S3 'Upload: status' page. At the top, a green header bar indicates 'Upload succeeded'. Below it, the title 'Upload: status' is displayed. A message states 'The information below will no longer be available after you navigate away from this page.' The 'Summary' section shows the destination 's3://exp12bucket1' and two rows: 'Succeeded' (1 file, 4.1 MB (100.0%)) and 'Failed' (0 files, 0 B (0%)). Below this, tabs for 'Files and folders' and 'Configuration' are visible. The 'Files and folders' tab is selected, showing a table with one item: 'cube.png' (image/png, 4.1 MB, Succeeded). A search bar and navigation arrows are also present.

Step 2: Configure Lambda function

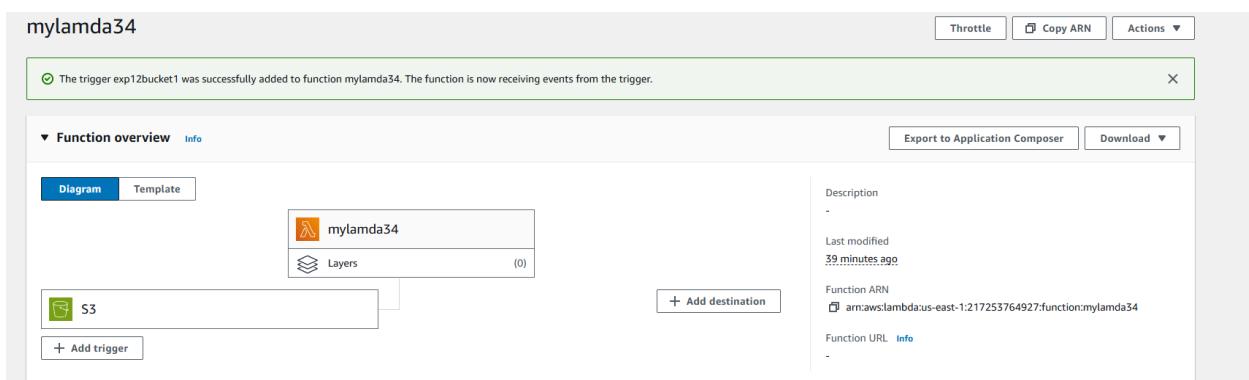
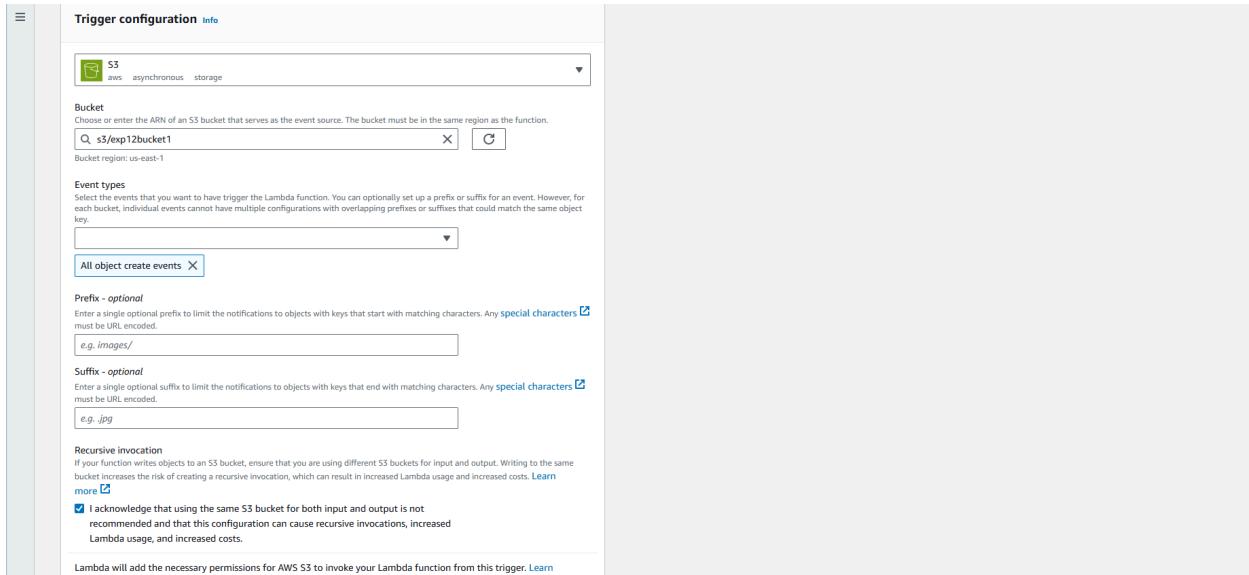
1) Go to the lambda function you had created before. (Services → Lambda → Click on name of function). Here, click on add trigger.

The screenshot shows the AWS Lambda function configuration page for 'mylamda34'. The 'Function overview' section is visible, showing the function name 'mylamda34', a description field, and a last modified time of '35 minutes ago'. The 'Code' tab is selected at the bottom. In the middle of the page, there is a 'Code source' section with a 'Test' dropdown and a 'Deploy' button. A prominent 'Add trigger' button is located in the center-left area. The right side of the page contains sections for 'Description', 'Last modified', 'Function ARN', and 'Function URL'.

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

The screenshot shows the 'Add trigger' configuration dialog. The 'Trigger configuration' section is open, with a dropdown menu titled 'Select a source' containing the option 's3'. Below this, a list of triggers is shown, with 'Batch/bulk data processing' and 'aws asynchronous storage' under the 's3' category. A 'Cancel' and 'Add' button are at the bottom right of the dialog.

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



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

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

  // Extract bucket name and object key from the event
  const record = event.Records[0];
```

```

const bucketName = record.s3.bucket.name;
const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys

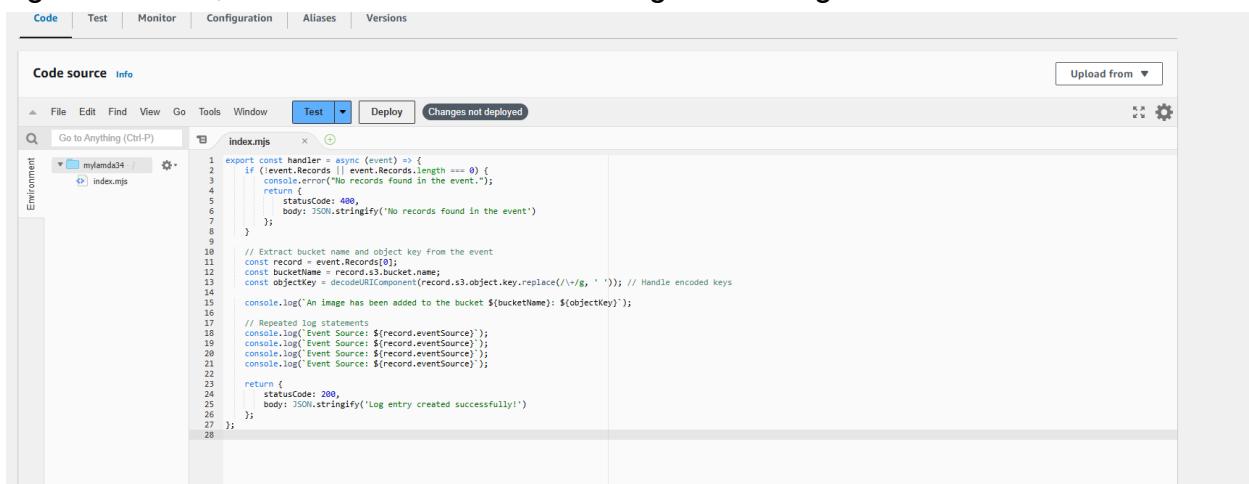
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);

// Repeated log statements
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);

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

```

This code checks for records in the event, extracts the bucket name and object key, logs the details, and returns a success message if an image is added to the bucket.



The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code source tab is selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a dropdown for Changes not deployed. On the far right of the toolbar are upload and settings icons. The main area displays the function's code in a code editor window titled 'index.mjs'. The code is identical to the one provided in the text block above, handling S3 events and logging details.

```

Code | Test | Monitor | Configuration | Aliases | Versions

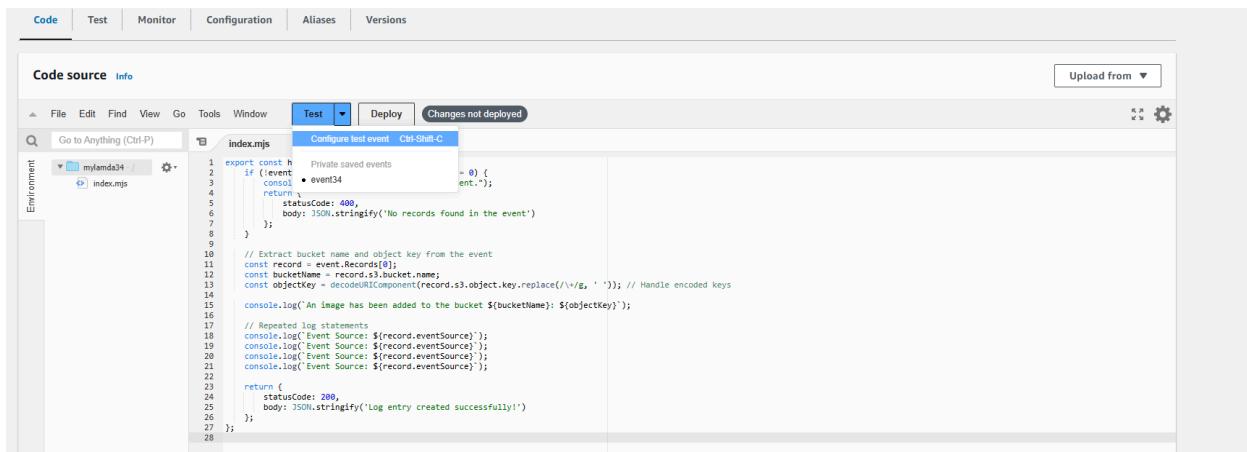
Code source Info

File Edit Find View Go Tools Window Test Deploy Changes not deployed

index.mjs
1 export const handler = async (event) => {
2   if (!event.Records || event.Records.length === 0) {
3     console.error('No records found in the event.');
4     return {
5       statusCode: 400,
6       body: JSON.stringify('No records found in the event')
7     };
8   }
9   // Extract bucket name and object key from the event
10  const record = event.Records[0];
11  const bucketName = record.s3.bucket.name;
12  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
13  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' '));
14  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' '));
15  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
16
17  // Repeated log statements
18  console.log(`Event Source: ${record.eventSource}`);
19  console.log(`Event Source: ${record.eventSource}`);
20  console.log(`Event Source: ${record.eventSource}`);
21  console.log(`Event Source: ${record.eventSource}`);
22
23  return {
24    statusCode: 200,
25    body: JSON.stringify('Log entry created successfully!')
26  };
27};
28

```

5) Now, click on the dropdown near the test, then click on the configure test event.



6) Here, select edit saved event. Select the event that you had created before. Under Event JSON, paste the following code.

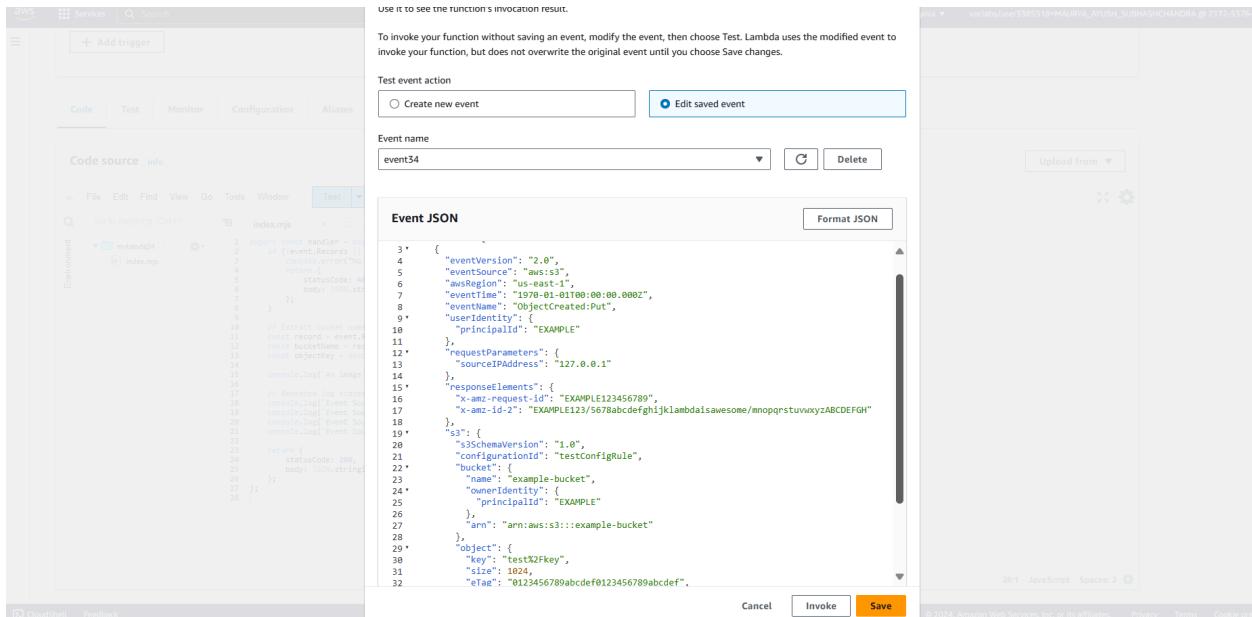
```
{  
  "Records": [  
    {  
      "eventVersion": "2.0",  
      "eventSource": "aws:s3",  
      "awsRegion": "us-east-1",  
      "eventTime": "1970-01-01T00:00:00.000Z",  
      "eventName": "ObjectCreated:Put",  
      "userIdentity": {  
        "principalId": "EXAMPLE"  
      },  
      "requestParameters": {  
        "sourceIPAddress": "127.0.0.1"  
      },  
      "responseElements": {  
        "x-amz-request-id": "EXAMPLE123456789",  
        "x-amz-id-2":  
          "EXAMPLE123/5678abcdefghijklmklambdaisawesome/mnopqrstuvwxyzABCDEFGH"  
      },  
      "s3": {  
        "s3SchemaVersion": "1.0",  
        "configurationId": "testConfigRule",  
        "bucket": {  
          "name": "example-bucket",  
          "ownerIdentity": {  
            "principalId": "EXAMPLE"  
          }  
        }  
      }  
    }  
  ]  
}
```

```

    },
    "arn": "arn:aws:s3:::example-bucket"
},
"object": {
    "key": "test%2Fkey",
    "size": 1024,
    "eTag": "0123456789abcdef0123456789abcdef",
    "sequencer": "0A1B2C3D4E5F678901"
}
}
]
}
}

```

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



Save the changes. Then deploy the code changes by clicking on deploy

- 7) After deploying, click on Test. The console output shows that 'an image has been added to the bucket'

The JSON response shows that the log entry was created successfully.

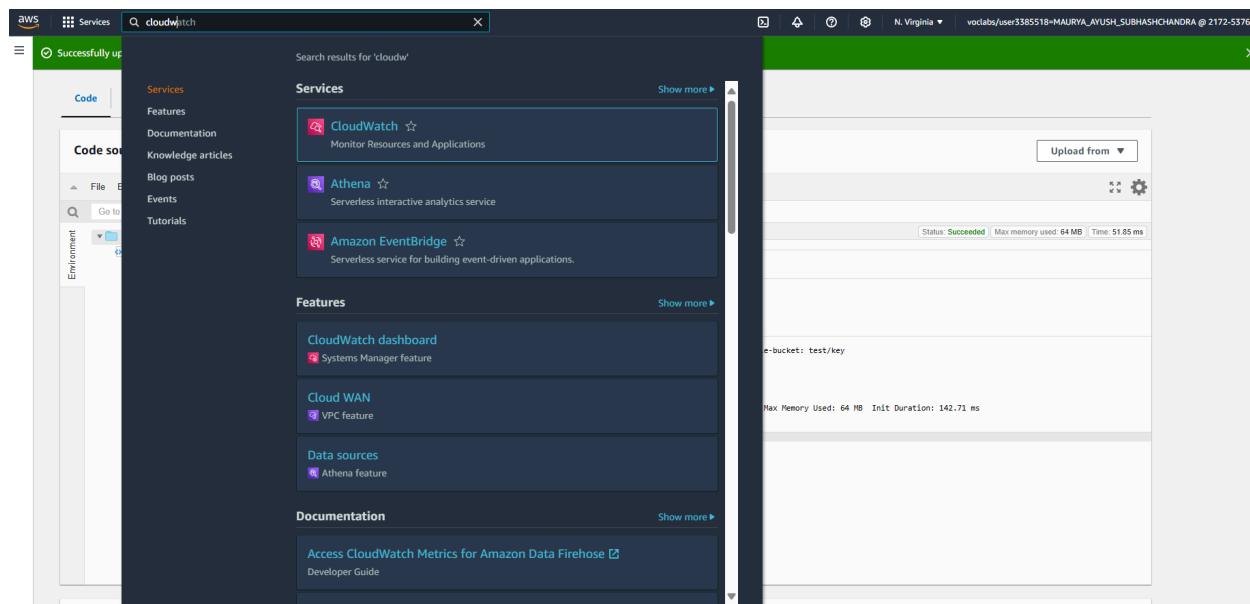
The screenshot shows the AWS Lambda console. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Test' tab is selected. Below the tabs, there's a file browser showing a folder named 'mylands34' containing an 'index.mjs' file. To the right of the file browser is a search bar and a dropdown menu for 'Execution results'. A sub-menu under 'Execution results' shows 'Test Event Name' and 'event:34'. The main content area displays the 'Response' and 'Function Logs' sections. The 'Response' section contains a JSON object with a 'statusCode' of 200 and a 'body' message stating 'log entry created successfully!'. The 'Function Logs' section shows a timestamped log entry indicating an image was added to a bucket:

```

START RequestId: b7410fe7-349b-4b16-972a-be40fac09fb3 Version: $LATEST
2024-10-06T17:18:35.376Z b7410fe7-349b-4b16-972a-be40fac09fb3 INFO An image has been added to the bucket example-bucket: test/key
2024-10-06T17:18:35.376Z b7410fe7-349b-4b16-972a-be40fac09fb3 INFO Event Source: aws:s3
2024-10-06T17:18:35.407Z b7410fe7-349b-4b16-972a-be40fac09fb3 INFO Event Source: aws:s3
2024-10-06T17:18:35.407Z b7410fe7-349b-4b16-972a-be40fac09fb3 INFO Event Source: aws:s3
2024-10-06T17:18:35.407Z b7410fe7-349b-4b16-972a-be40fac09fb3 INFO Event Source: aws:s3
END RequestId: b7410fe7-349b-4b16-972a-be40fac09fb3
REPORT RequestId: b7410fe7-349b-4b16-972a-be40fac09fb3 Duration: 51.85 ms Billed Duration: 52 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 142.71 ms
Request ID
b7410fe7-349b-4b16-972a-be40fac09fb3
  
```

Step 3: Check the logs

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



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

The screenshot shows the AWS CloudWatch Log Groups page. On the left, there's a navigation sidebar with options like Favorites and recent dashboards, Alarms, Logs (Log groups, Log anomalies, Live tail, Logs insights, Contributor insights), Metrics, X-Ray traces, Events, and Application Signals. The main area is titled "Log groups (4)" and shows a table with columns: Log group, Log class, Anomaly detection, Data protection, Sensitive data count, Retention, Metric filters, and Contributor Insights. The log group "/aws/lambda/mylambda34" is selected. At the top right, there are buttons for Actions, View in Logs Insights, Start tailing, and Create log group.

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

The screenshot shows the AWS CloudWatch Log streams page. It displays six log streams listed in a table with columns: Log stream, Last event time, and Duration. The log streams are: 2024/10/06/[\$LATEST]e0b1f30801d84fdb70f5cb8fe6d2530, 2024/10/06/[\$LATEST]4a738d5c654a4d8894df9e7e718a21bd, 2024/10/06/[\$LATEST]75cc563fb34c42d8a72e1b566b1014af, 2024/10/06/[\$LATEST]1eea5024119f46d3a99abae1f53b5d74, 2024/10/06/[\$LATEST]0b775a1bc8e645fdb712ef1db7c4704d, and 2024/10/06/[\$LATEST]b40efb0ce9264d918bdad02483add33c. Below the table are tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, and Data protection.

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

The screenshot shows the AWS CloudWatch Log events page. It lists log events for the log group "/aws/lambda/mylambda34". The events are timestamped and show Lambda runtime details: INIT, START, REPORT, and END requests. One event shows a REPORT request with a duration of 1.81 ms and a memory usage of 62 MB. The interface includes a search bar, filter buttons for Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone, and Display, and a button for Create metric filter.

Conclusion:

In this experiment, we successfully created and configured an AWS Lambda function to log when an image is added to a specific S3 bucket. We explored the integration between AWS Lambda and S3, demonstrating how event-driven processes can automate tasks. By setting up an S3 bucket trigger, we enabled the Lambda function to detect object uploads and log essential details like the bucket name and object key. After deploying and testing the function, we verified the logs in CloudWatch, confirming that the function worked as expected, accurately detecting and logging the addition of images to the bucket. This experiment showcases how AWS services can seamlessly collaborate for automation.