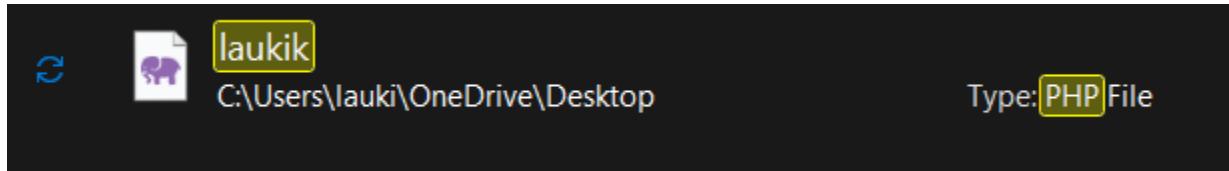


Exp 1a : Static Hosting

a) Hosting of a PHP file on Local virtual machine using Xampp

1. Create a .php file in some local repository



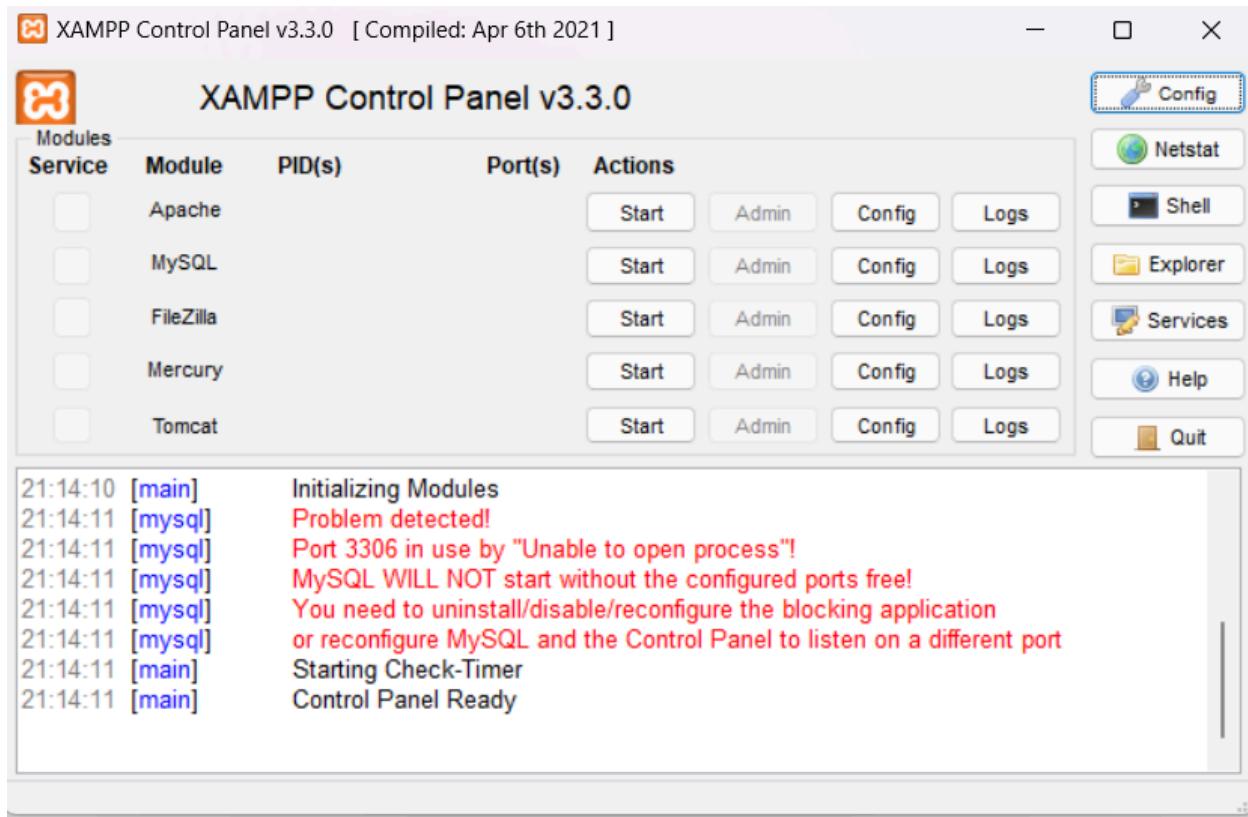
2. Make changes in that .php file as per your desire and save it

A screenshot of a code editor window titled "laukik.php". The window has a dark background. At the top, there is a menu bar with "File", "Edit", and "View" options. Below the menu, the code is displayed in a monospaced font:

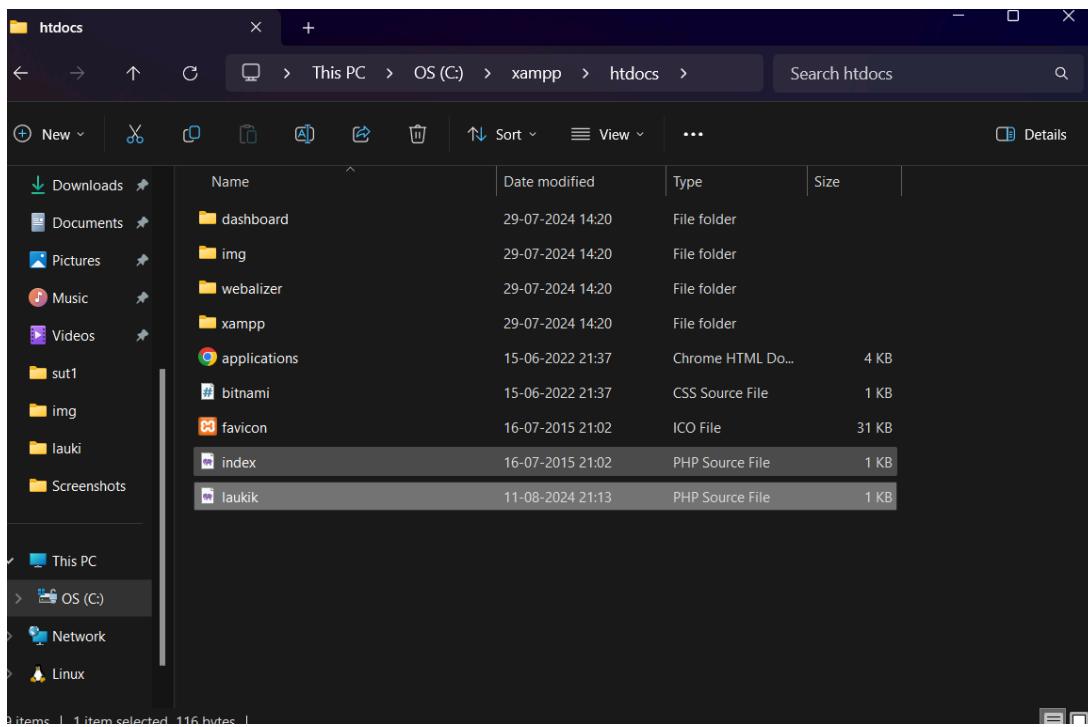
```
<html>
<head>
    <title>First PHP Program</title>
</head>
<body>
<?php
echo "I am laukik";
?>
</body>
</html?|
```

The code consists of basic HTML structure and a single PHP echo statement.

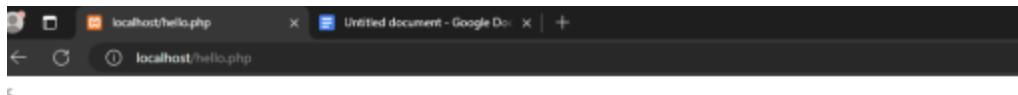
3. Make sure you have installed Xampp on your local machine. After the installation, start Xampp control panel and start modules named Apache and MySql



4. Ensure that you relocate your .php file in the htdocs folder inside the Xampp folder



5. Access the contents of the php file by typing localhost/your_file.php on your browser. We have successfully hosted our php file on our local machine using Xampp.



My first PHP page

Hello World!

b) Static hosting using AWS S3 bucket

1. Navigate to S3 inside services and create a bucket inside S3. Select the additional settings related to the bucket.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The 'Name and tags' section has 'My webserver' entered in the 'Name' field. The 'Application and OS Images (Amazon Machine Image)' section shows a search bar and a list of AMIs including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux Enterprise Server. The 'Summary' panel on the right shows 'Number of instances' set to 1, 'Software Image (AMI)' as Canonical, Ubuntu, 24.04 LTS, 'Virtual server type (instance type)' as t2.micro, and 'Storage (volumes)' as 1 volume(s) - 8 GiB. A tooltip for the 'Free tier' is visible, stating: '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'. The 'Launch instance' button is highlighted in orange at the bottom right.

The screenshot shows the 'Quick Start' page in the AWS console. It features a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux Enterprise Server. The 'Ubuntu' icon is selected. The 'Summary' panel on the right shows 'Number of instances' set to 1, 'Software Image (AMI)' as Canonical, Ubuntu, 24.04 LTS, 'Virtual server type (instance type)' as t2.micro, and 'Storage (volumes)' as 1 volume(s) - 8 GiB. A tooltip for the 'Free tier' is visible, stating: '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'. The 'Launch instance' button is highlighted in orange at the bottom right.

The screenshot shows the AWS S3 Buckets page. At the top, there's an account snapshot and a link to the Storage Lens dashboard. Below that, two tabs are visible: 'General purpose buckets' (selected) and 'Directory buckets'. A search bar labeled 'Find buckets by name' is present. A table lists the bucket details:

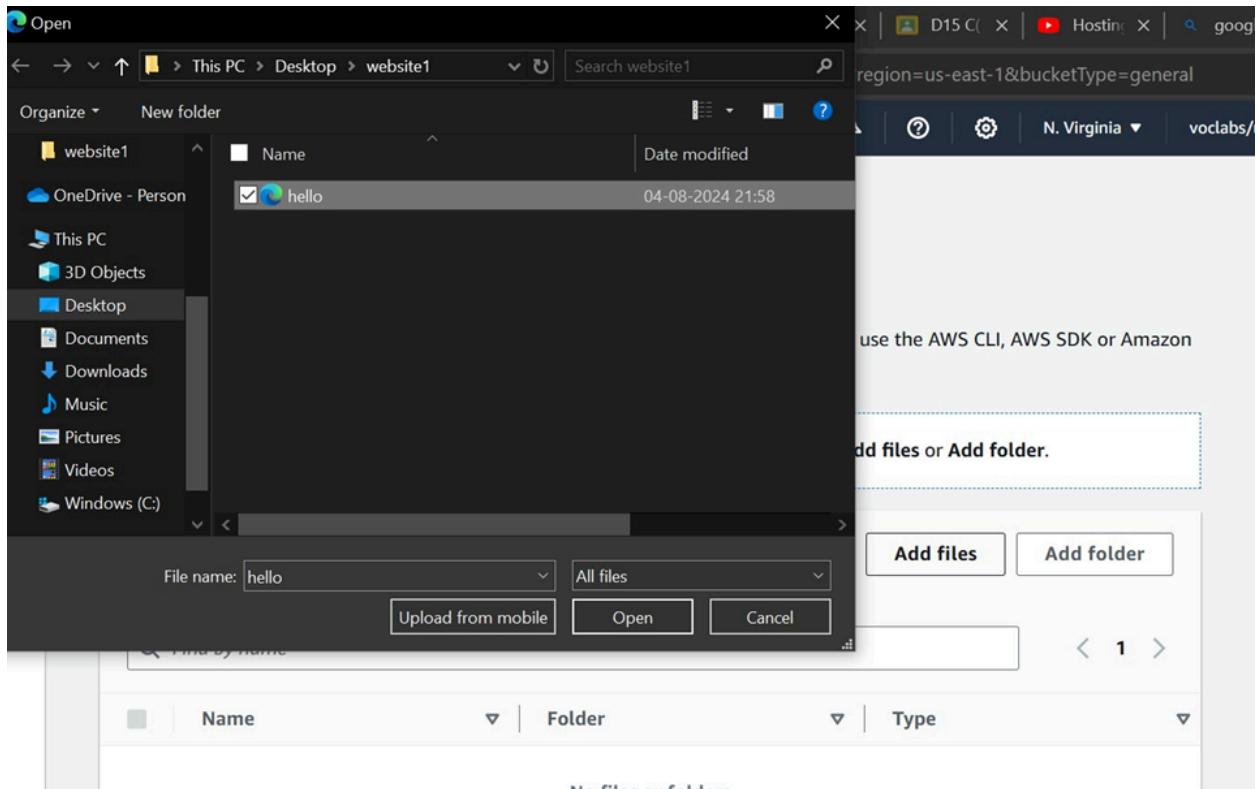
Name	AWS Region	IAM Access Analyzer	Creation date
laukikisgreat	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 29, 2024, 14:39:44 (UTC+05:30)

2. Our bucket was successfully created. Now, we would want to add/upload our local files onto our bucket

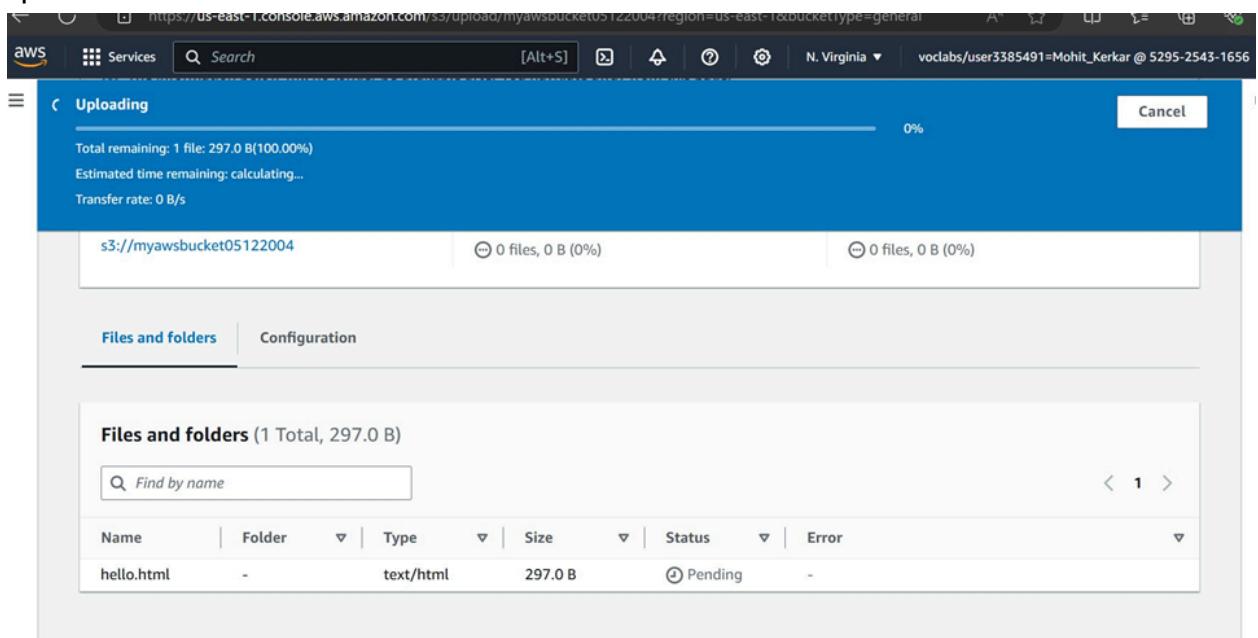
The screenshot shows the AWS S3 Buckets page again. The account snapshot and storage lens links are at the top. The 'General purpose buckets' tab is selected. A search bar is present. A table lists the buckets:

Name	AWS Region	IAM Access Analyzer	Creation date
laukikisgreat	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 29, 2024, 14:39:44 (UTC+05:30)
laukkkk	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 29, 2024, 14:52:02 (UTC+05:30)

3. Create a new html file, modify it and save it with a desired file name. Now, add/upload this file onto your S3 bucket.



4. For enabling static website hosting, we must enable this Static website hosting option



The screenshot shows the AWS S3 console with the 'Host a static website' option selected. A callout box highlights a note: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)'.

Index document
Specify the home or default page of the website.
index.html

Error document - optional
This is returned when an error occurs.
error.html

Redirection rules - optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

5. Open the website using the S3 bucket. You'd be able to see the contents of your html file. Thus, we have successfully and

The browser window shows the URL <https://myawsbucket05122004.s3.us-east-1.amazonaws.com/hello.html?response-content-disposition=inline&X-Amz-Secu...>. The page content is "this is a dummy website" followed by "i hope you get what i am saying".

Name: Laukik Padgaonkar

D15C

Roll No:37

Experiment No. 1B

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



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

A screenshot of the 'Create environment' form in the AWS Cloud9 interface. The form has a 'Details' section. Under 'Name', the value 'Laukik Padgaonkar' is entered. Under 'Description - optional', there is a text input field with the placeholder 'Limit 200 characters.' Below this, under 'Environment type', there are two options: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' option includes a note: 'Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.' The 'Existing compute' option includes a note: 'You have an existing instance or server that you'd like to use.'

Use the Secure Shell option in Network settings

Network settings [Info](#)

Connection
How your environment is accessed.

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

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

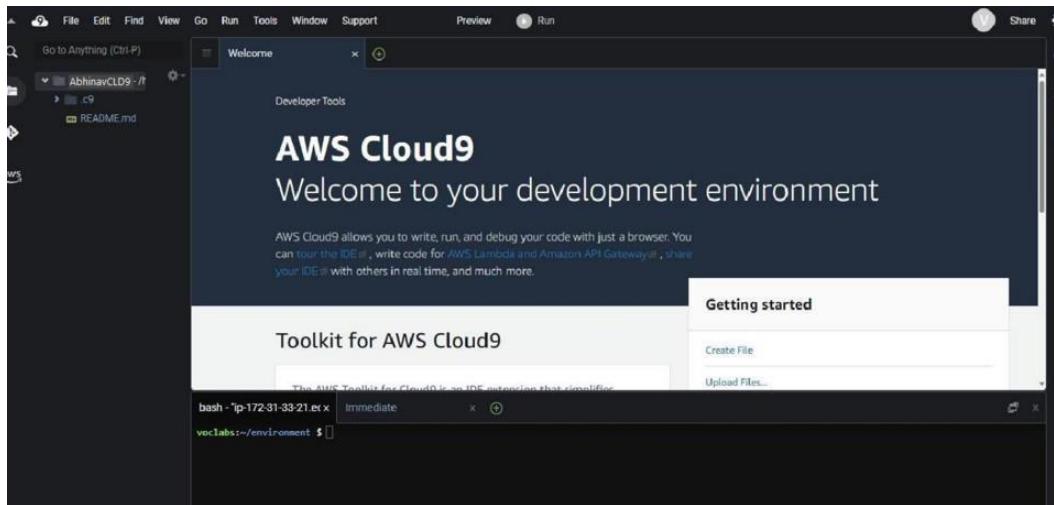
▶ [VPC settings](#) [Info](#)

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

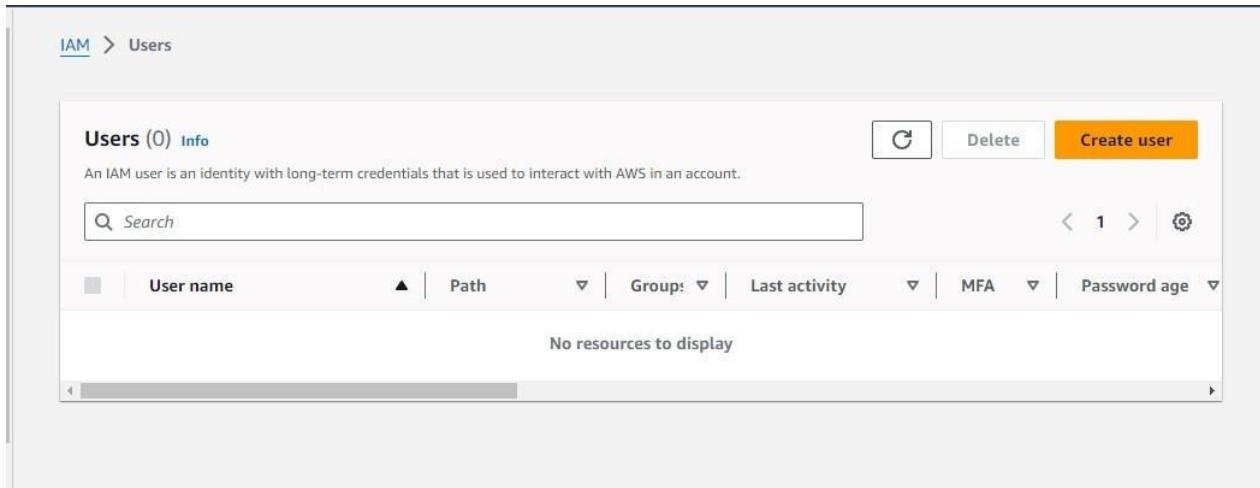
 The following IAM resources will be created in your account

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

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

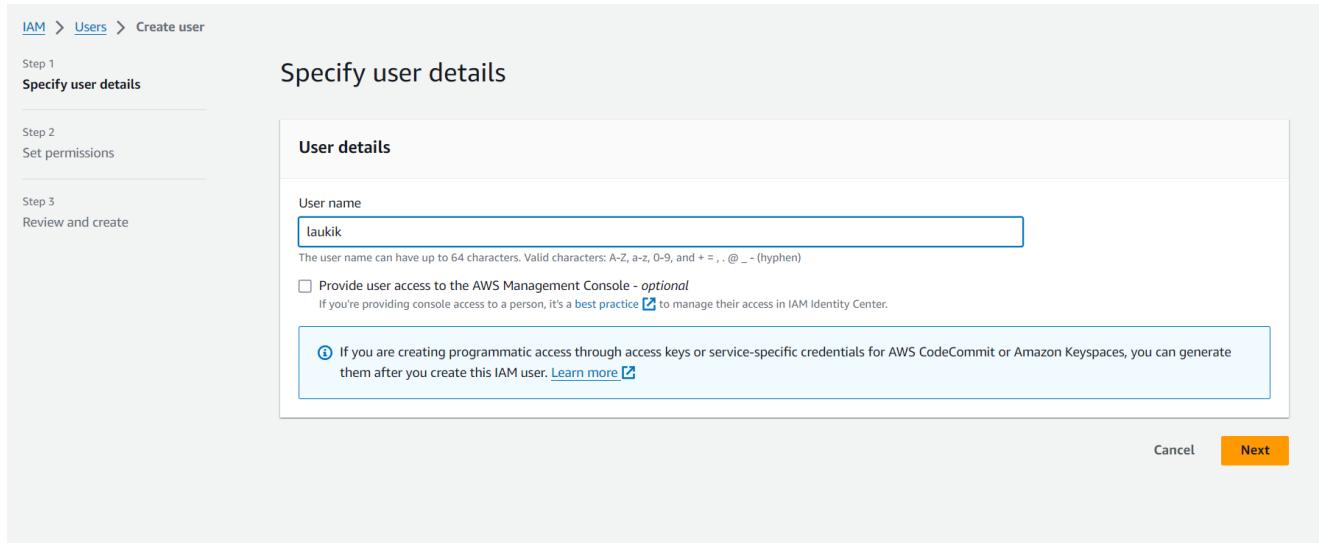


Open the aws account and search for IAM service. Then go to users tab and click on create user to create a new user.



The screenshot shows the AWS IAM service interface. In the top left, it says "IAM > Users". Below that, there's a heading "Users (0) Info" with a note: "An IAM user is an identity with long-term credentials that is used to interact with AWS in an account." There's a search bar labeled "Search" and a row of buttons: "Create user" (orange), "Delete" (grey), and "Edit" (grey). A table header is shown with columns: "User name", "Path", "Group", "Last activity", "MFA", and "Password age". Below the table, a message says "No resources to display".

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



The screenshot shows the "Create user" wizard at step 1: "Specify user details". On the left, there are three tabs: "Step 1 Specify user details" (selected), "Step 2 Set permissions", and "Step 3 Review and create". The main area has a title "User details" and a "User name" field containing "laukik". Below the field, a note says: "The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)". There's also an optional checkbox: "Provide user access to the AWS Management Console - optional". A note under it says: "If you're providing console access to a person, it's a best practice [link] to manage their access in IAM Identity Center." At the bottom right, there are "Cancel" and "Next" buttons.

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

Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended

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

I want to create an IAM user

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

Console password

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

- Must be at least 8 characters long



IAM > Users > Create user

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Step 4

Retrieve password

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

[Create group](#)

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

Create user group

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

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

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

Permissions policies (947)

Filter by Type
 All ty... ▾

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

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

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

Step 1
[Specify user details](#)

Step 2
[Set permissions](#)

Step 3
Review and create

Step 4
[Retrieve password](#)

Review and create

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

User details		
User name	Console password type	Require password reset
Abhinav	Custom password	No

Permissions summary

Name	Type	Used as
AdvanceDevOps_1	Group	Permissions group
AdvanceDevOps_2	Group	Permissions group
AdvDevOpsLab_3	Group	Permissions group

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

The screenshot shows the AWS Identity and Access Management (IAM) console. On the left, the navigation pane is visible with the following structure:

- Identity and Access Management (IAM)
- Dashboard
- Access management
 - User groups (selected)
 - Users
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access Analyzer
 - External access
 - Unused access

The main content area displays the details for a user group named "AdvanceDevOps_1". The "Summary" section includes the user group name, creation time (August 07, 2024, 09:33 (UTC+05:30)), and ARN (arn:aws:iam::022499016110:group/AdvanceDevOps_1). Below the summary, there are tabs for "Users (3)", "Permissions" (which is selected), and "Access Advisor". The "Permissions policies (0)" section indicates that up to 10 managed policies can be attached. A search bar and a filter dropdown for "All types" are also present.

Search for the “AWSCloud9EnvironmentMember” policy and attach it.

The screenshot shows the AWS IAM User Groups interface. A user group named "AdvanceDevOps_3_21_9" is selected. The "Add permissions" step is being completed. In the "Other permission policies" section, the "AWSCloud9EnvironmentMember" policy is selected and highlighted. The "Attach policies" button is visible at the bottom right of the search results.

Current permissions policies (0)

Other permission policies (945)

Filter by Type

Policy name	Type	Used as	Description
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
AdministratorAccess-Amp...	AWS managed	None	Grants account administrative permis...
AdministratorAccess-AWS...	AWS managed	None	Grants account administrative permis...
AlexaForBusinessDeviceS...	AWS managed	None	Provide device setup access to AlexaFo...

Current permissions policies (0)

Other permission policies (1/945)

Filter by Type

Policy name	Type	Used as	Description
AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS ...
AWSCloud9Environment...	AWS managed	None	Provides the ability to be invited into ...
AWSCloud9SSMInstanceP...	AWS managed	None	This policy will be used to attach a rol...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

Cancel Attach policies

The screenshot shows the "Policies attached to this user group" summary. It lists the user group name as "AdvanceDevOps_3_21_9", creation time as "August 07, 2024, 09:33 (UTC+05:30)", and ARN as "arn:aws:iam::022499016110:group/AdvanceDevOps_3_21_9". The "Permissions" tab is selected, showing one attached policy: "AWSCloud9EnvironmentMember".

Policies attached to this user group.

Summary

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

Users (3) Permissions Access Advisor

Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type

Policy name	Type	Attached entities
AWSCloud9EnvironmentMe...	AWS managed	3

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

Step 1: Create our ElasticBeanstalk Environment

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

The screenshot shows the 'Create new application' wizard in the AWS Elastic Beanstalk console. The 'Application information' section is visible, with the 'Application name' field set to 'beanstalk4'. The 'Tags' section is partially visible below it. The 'Platform' section is also visible, showing 'PHP' selected as the platform, with options for 'Platform branch' (PHP 8.3 running on 64bit Amazon Linux 2023) and 'Platform version' (4.3.1 (Recommended)). The 'Application code' section at the bottom shows 'Sample application' selected.

CloudShell Feedback https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-application [Alt+S] Mumbai mohitkerkar05

Elastic Beanstalk > Create application

Create new application Info

Application information

Application name
beanstalk4

Description

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive.

CloudShell Feedback https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-environment?applicationN... [Alt+S] Mumbai mohitkerkar05

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

Application code Info

Sample application

Existing version

Upload your code

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

CloudShell Feedback https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-environment?applicationN... [Alt+S] Mumbai mohitkerkar05

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

1. AWSElasticBeanstalkWebTier
2. AWSElasticBeanstalkWorkerTier
3. AWSElasticBeanstalkMulticontainerDocker

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

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for the environment 'Laukikapp-env'. At the top, a green banner indicates 'Environment successfully launched.' Below the banner, the navigation path is 'Elastic Beanstalk > Environments > Laukikapp-env'. The main content area is divided into two columns: 'Environment overview' and 'Platform'. The 'Environment overview' column contains information such as Health (Warning), Environment ID (e-kmfuu3urih), Domain (Laukikapp-env.eba-mmfxapm.eu-north-1.elasticbeanstalk.com), Application name (laukik_app), and a 'Change version' button. The 'Platform' column shows PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2, the Running version (empty), and the Platform state (Supported). Below this, the 'Source' section is visible, showing a dropdown for 'Source provider' set to 'GitHub (Version 2)'. A callout box provides information about GitHub version 2 actions. The 'Connection' section at the bottom allows selecting or creating a connection to a GitHub repository.

Step 2: Fork the required repository onto our github account

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

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

Codepipeline-s3-codedeploy-linux-2.0 Public
forked from [imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0](#)

master 1 Branch 0 Tags Go to file + <> Code About

This branch is up to date with [imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0:master](#).

Contribute Sync fork

imoisharma Update README.md 8fd5da5 · 3 years ago 20 Commits

- .github Adding template 7 years ago
- dist Added dist folder 9 years ago
- scripts s3 setup and s3 set cache control scripts 3 years ago
- CODE_OF_CONDUCT.md Adding CONTRIBUTING/CoC 7 years ago
- CONTRIBUTING.md Adding CONTRIBUTING/Coc 7 months ago

Readme Apache-2.0 license Code of conduct Activity 0 stars 0 watching 0 forks

Step 3: Creation of the Pipeline

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

And click on next ...

Environment successfully launched.

Elastic Beanstalk > Environments > Laukikapp-env

Laukikapp-env Info Actions Upload and deploy

Environment overview		Platform	Change version
Health	Environment ID	Platform	
⚠ Warning	e-kmfuu3urih	PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	
Domain	Application name	Running version	
Laukikapp-env.eba-mmfxapm.eu-north-1.elasticbeanstalk.com	laukik_app	-	
		Platform state	
		Supported	

Step 4: Github connection

In this step, we are supposed to create a github connection and add our existing repository over

here i.e the one we forked earlier

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

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.

Search bar: Q or Connecting

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

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

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

Q arn:aws:codeconnections:eu-north-1:010928207735:connection/59473412- X or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

Q 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.

Q

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 5: Deployment stage:

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

The screenshot shows the 'Add deploy stage' step in the AWS CodePipeline console. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage - currently selected), and Step 5 (Review). The main area has a heading 'Deploy' and a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk'. Below it is a 'Region' dropdown set to 'Asia Pacific (Mumbai)'. An 'Input artifacts' section is present but empty. A note at the top states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.'

The screenshot continues from the previous step. It shows the configuration for the 'AWS Elastic Beanstalk' provider. Under 'Application name', the search bar contains 'beanstalk4'. Under 'Environment name', the search bar contains 'Beanstalk4-env-1'. A checkbox for 'Configure automatic rollback on stage failure' is unchecked. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted.

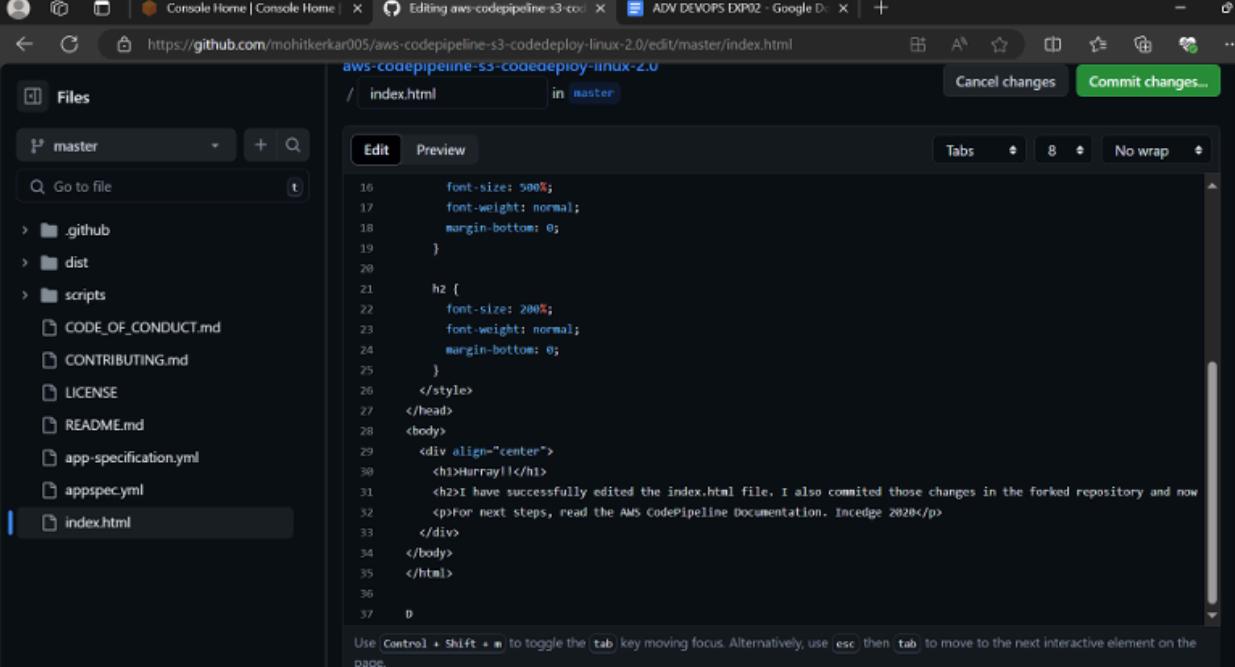
Step 6: Post deployment stage: When all the stages run successfully, this is what is displayed onto the screen. It shows us that our application and our environment have successfully been deployed using a dedicated pipeline created

The screenshot shows the AWS CodePipeline console. At the top, there is a green banner with the text "Success" and "Congratulations! The pipeline laukik_pipe has been created." A button to "Create a notification rule for this pipeline" is also present. Below the banner, the pipeline name "laukik_pipe" is displayed along with navigation links: Developer Tools > CodePipeline > Pipelines > laukik_pipe. To the right of the pipeline name are several buttons: Notify (with a dropdown arrow), Edit, Stop execution, Clone pipeline, and Release change (which is highlighted in orange). Below these buttons, it says "Pipeline type: V2" and "Execution mode: QUEUED". The main content area shows a single stage named "Source" which is currently "In progress". A timestamp indicates the progress was made "1 minute ago". A "View details" button is located at the bottom of this stage card. On the far right, there are two small circular icons.

The screenshot shows a success page for a Lambda function. The title is "Hi Laukik Padgaonkar". Below the title, a message states: "You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy." At the bottom of the page, there is a link: "For next steps, read the AWS CodePipeline Documentation."

Step 7: Committing changes to your github code

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

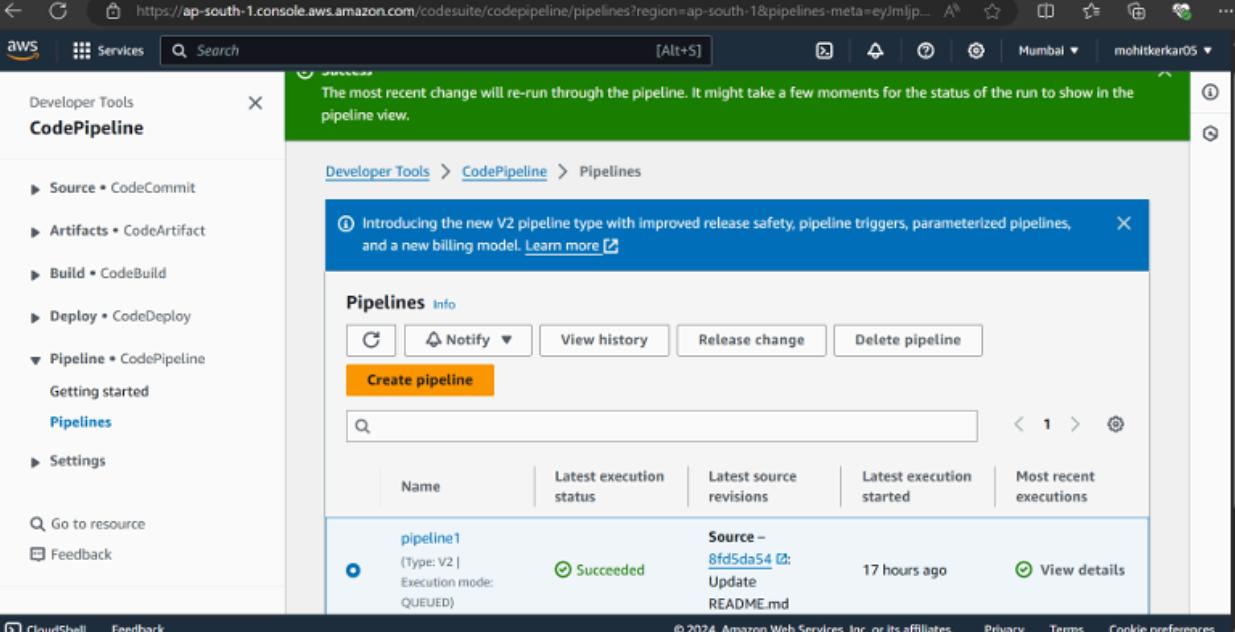


```

16     font-size: 500px;
17     font-weight: normal;
18     margin-bottom: 0;
19   }
20
21   h2 {
22     font-size: 200px;
23     font-weight: normal;
24     margin-bottom: 0;
25   }
26   </style>
27   </head>
28   <body>
29   <div align="center">
30     <h1>Hurray!!</h1>
31     <h2>I have successfully edited the index.html file. I also committed those changes in the forked repository and now for next steps, read the AWS CodePipeline Documentation. Incidege 2820</h2>
32     <p>For next steps, read the AWS CodePipeline Documentation. Incidege 2820</p>
33   </div>
34   </body>
35   </html>
36
37 
```

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

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

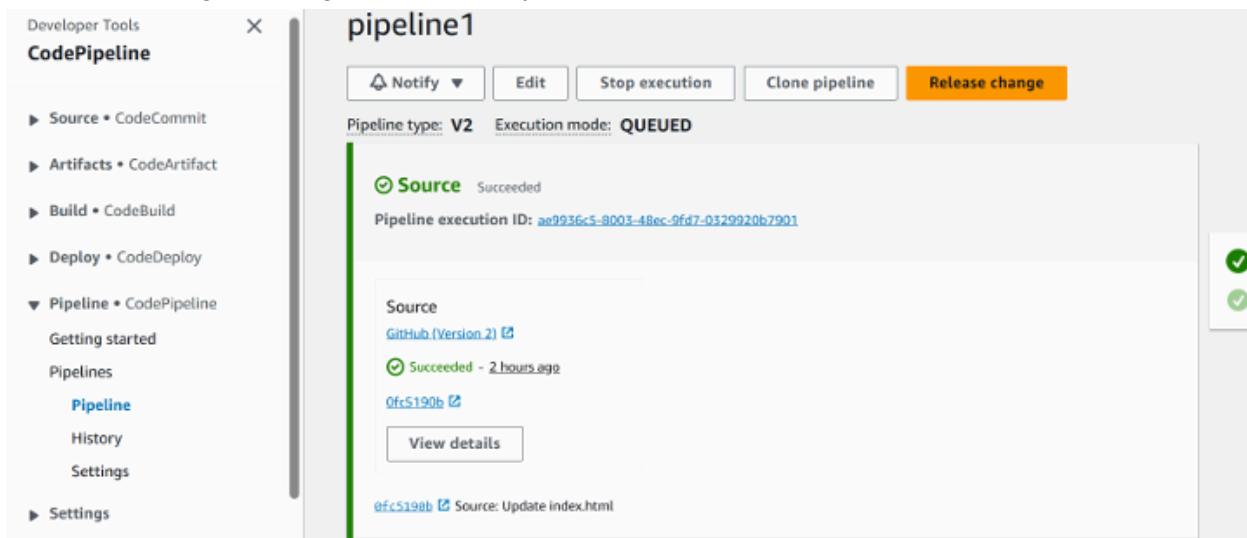


The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Pipelines

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
pipeline1	Succeeded	8fd5da54	17 hours ago	View details

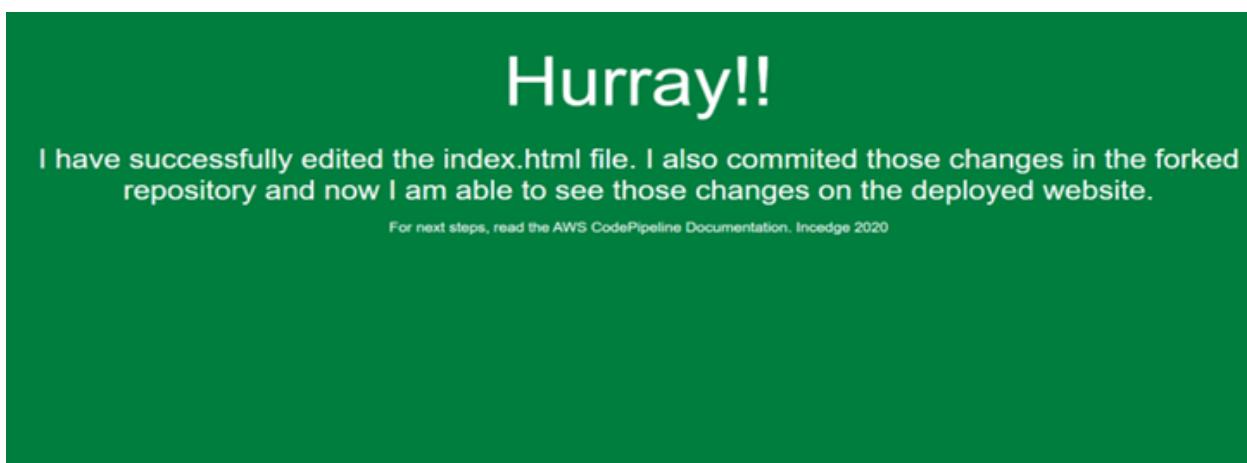
Once the changes have been applied, we see the commit message that we wrote for the latest commit on our repository being reflected on our pipeline. Over here, it would be seen somewhere near the bottom of the image that is attached. "Update index.html" was the latest commit message in the github repository



Step 9: Open the Domain of our Elastic Beanstalk environment

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

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



Advanced DevOps Lab

Experiment:3

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

Reference: <https://www.youtube.com/watch?v=Cz7hSJNq2GU>

Theory:

Container-based microservices architectures have profoundly changed the way development and operations teams test and deploy modern software. Containers help companies modernize by making it easier to scale and deploy applications, but containers have also introduced new challenges and more complexity by creating an entirely new infrastructure ecosystem.

Large and small software companies alike are now deploying thousands of container instances daily, and that's a complexity of scale they have to manage. So how do they do it?

Enter the age of Kubernetes.

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

Kubernetes makes it easy to deploy and operate applications in a microservice architecture. It does so by creating an abstraction layer on top of a group of hosts so that development teams can deploy their applications and let Kubernetes manage the following activities:

- Controlling resource consumption by application or team
 - Evenly spreading application load across a hosting infrastructure
 - Automatically load balancing requests across the different instances of an application
 - Monitoring resource consumption and resource limits to automatically stop applications from consuming too many resources and restarting the applications again
 - Moving an application instance from one host to another if there is a shortage of resources in a host, or if the host dies
 - Automatically leveraging additional resources made available when a new host is added to the cluster
-
- Easily performing canary deployments and rollbacks

Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

(Name 1 as Master, the other 2 as worker-1 and worker-2)

	Name	Instance ID	Instance state	
	kube-master	i-00aa79ac09d7462c0	Running	
	kube-worker1	i-0bab86cd3fbfcba0a	Running	
	kube-worker2	i-00dcfd302ffd80dda	Running	

2. Edit the Security Group Inbound Rules to allow SSH

The screenshot shows the AWS Management Console interface for managing security group inbound rules. A single rule is listed for port 22 (TCP) from anywhere (0.0.0.0/0). The rule is labeled 'SSH'. The 'Add rule' button is visible at the bottom left.

3. SSH into all 3 machines

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

```
quantum@machine ~ ~/Downloads ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-3-88-111-183.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-111-183.compute-1.amazonaws.com (3.88.111.183)' can't be established.
ED25519 key fingerprint is SHA256:pQu+xsoYbY3de1twjZcVVAOzmGwGv6PHmVruF/Q1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-111-183.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
,
~\_
###_ Amazon Linux 2023
~~ \#####\
~~ \##|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' '-->
~~ / /
~~-.-/
```

4. From now on, until mentioned, perform these steps on all 3 machines.

Install Docker

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

```
sudo apt-get update
sudo apt-get install -y docker-ce
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
```

Then, configure cgroup in a daemon.json file.

```
cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Install Kubernetes on all 3 machines

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

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

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

5. Perform this **ONLY on the Master machine**

Initialize the Kubecluster

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

```
-ignore-preflight-errors=all
[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'
W0914 09:57:27.006694    9935 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with kubelet. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Using existing ca certificate authority
[certs] Using existing apiserver certificate and key on disk
[certs] Using existing apiserver-kubelet-client certificate and key on disk
[certs] Using existing front-proxy-ca certificate authority
[certs] Using existing front-proxy-client certificate and key on disk
[certs] Using existing etcd/ca certificate authority
[certs] Using existing etcd/server certificate and key on disk
[certs] Using existing etcd/peer certificate and key on disk
[certs] Using existing etcd/healthcheck-client certificate and key on disk
[certs] Using existing apiserver-etcd-client certificate and key on disk
[certs] Using the existing "sa" key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
```

Copy the join command and keep it in a notepad, we'll need it later.

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

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

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

```
[ec2-user@ip-172-31-81-63 docker]$ 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-eof created
```

Check the created pod using this command

Now, keep a watch on all nodes using the following command

```
watch kubectl get nodes
```

6. Perform this **ONLY** on the worker machines

```
sudo kubeadm join <ip> --token <token> \
--discovery-token-ca-cert-hash <hash>
```

Now, notice the changes on the master terminal

```
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME                  STATUS   ROLES      AGE     VERSION
ip-172-31-85-89.ec2.internal  NotReady control-plane  119s   v1.26.0
ip-172-31-89-46.ec2.internal  NotReady <none>        19s    v1.26.0
ip-172-31-94-70.ec2.internal  NotReady <none>        12s    v1.26.0
[root@ip-172-31-85-89 ec2-user]#
```

That's it, we now have a Kubernetes cluster running across 3 AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed across these machines.

Conclusion: In this experiment, the goal was to set up and configure a Kubernetes cluster using kubeadm on a Linux environment with yum as the package manager.

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

Theory:

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

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

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 personal account and launch a new Ec2 Instance.

Select Ubuntu as AMI and **t2.medium** as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.

Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a 'Create New Function' button. Below that, a 'Function Overview' section displays the ARN, Region (us-east-1), Handler (index.handler), and Runtime (Node.js 18.x). The 'Code' tab is selected, showing the code editor with the provided Node.js code. The 'Logs' tab is also visible. On the right, there are tabs for 'Overview', 'Configuration', 'Monitoring', and 'Actions'.

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

The screenshot shows the AWS EC2 Instances page. It lists one instance named 'Experiment 4' with the following details: Name: Experiment 4, Instance ID: i-09f5752831db50f7d, State: Running, Type: t2.medium, Status: Initializing, and Public IP: ec2-54-165-99-170.co... The 'Actions' dropdown menu is open, showing options like Stop, Start, Terminate, and More options.

EC2 > Instances > i-09f3752831db50f7d > Connect to instance

Connect to instance Info

Connect to your instance i-09f3752831db50f7d (Experiment 4) using any of these options

EC2 Instance Connect | **Session Manager** | **SSH client** (selected) | **EC2 serial console**

Instance ID
 i-09f3752831db50f7d (Experiment 4)

1. Open an SSH client
2. Locate your private key file. The key used to launch this instance is Master_Ec2_Key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Master_Ec2_Key.pem"
4. Connect to your instance using its Private IP:
 172.31.20.171

Example:
 ssh -i "Master_Ec2_Key.pem" ubuntu@172.31.20.171

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

Cancel

Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)

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

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

PS C:\Users\bhush\OneDrive\Desktop>New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 07:58:53 UTC 2024

System load: 0.15      Processes:          152
Usage of /: 55.3% of 6.71GB   Users logged in:     1
Memory usage: 20%           IPv4 address for enX0: 172.31.20.171
Swap usage: 0%

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

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Sun Sep 15 07:27:23 2024 from 152.58.2.47
```

Step 4: Run the below commands to install and setup Docker.

```
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-172-31-20-171:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-20-171:~$ 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...
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Found existing deb-src entry in /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
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
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 https://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 0s (128 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has a
n 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 s
ection in apt-key(8) for details.
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-17-25:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy tru
ection 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-pl
  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 docker-ce-rootless-extras docker-
  libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 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
```

```

Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.1-1~ubuntu.24.04~noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.1-1~ubuntu.24.04~noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.1-1~ubuntu.24.04~noble [9571 kB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 MB]
Fetched 122 MB in 2s (71.3 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.


```

```

Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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


```

sudo mkdir -p /etc/docker

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

```

ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-20-171:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}


```

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

```
ubuntu@ip-172-31-17-25:~$ 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
```

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-172-31-20-171:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-20-171:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

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

```
ubuntu@ip-172-31-17-25:~$ sudo apt-get update-25:~$ 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 I
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb P
Fetched 6051 B in 1s (10.4 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trust
ection 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 133 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 2110 MB of additional disk space will be used
```

```
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

sudo systemctl enable --now kubelet

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

```
ubuntu@ip-172-31-17-25:~$ 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 Sun 2024-09-15 17:38:22 UTC; 272ms ago
     Docs: https://containerd.io
 Main PID: 9670 (containerd)
    Tasks: 7
   Memory: 13.4M (peak: 13.7M)
      CPU: 66ms
     CGroup: /system.slice/containerd.service
             └─9670 /usr/bin/containerd

Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.267775291Z" level=info msg="serving..." address=/run/contai
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268641365Z" level=info msg="serving..." address=/run/contai
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268711553Z" level=info msg="Start subscribing containerd e
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268738456Z" level=info msg="Start recovering state"
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268779583Z" level=info msg="Start event monitor"
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268794095Z" level=info msg="Start snapshots syncer"
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268801586Z" level=info msg="Start cni network conf syncer
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268807177Z" level=info msg="Start streaming server"
Sep 15 17:38:22 ip-172-31-17-25 containerd[9670]: time="2024-09-15T17:38:22.268848440Z" level=info msg="containerd successfully booted"
```

Now We have got an error.**So we have to perform some additional commands as follow.****sudo apt-get install -y containerd**

```
To see the stack trace of this error execute with --v=5 or higher    ubuntu@ip-172-31-20-171:~$ 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 130 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (74.5 MB/s)
(Reading database ... 68068 files and directories currently installed.)
Removing docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68048 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4.1) ...
Setting up runc (1.1.12-0ubuntu3.1) ...
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.
```

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

```
ubuntu@ip-172-31-20-171:~$ 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

[metrics]
  address = ""
  grpc_histogram = false

[plugins]

  [plugins."io.containerd.gc.v1.scheduler"]
    deletion_threshold = 0
```

...

```
[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
containerd sudo systemctl status
containerd
```

```
ubuntu@ip-172-31-20-171:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
ubuntu@ip-172-31-20-171:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; en>
     Active: active (running) since Sun 2024-09-15 07:49:23 UTC; 5s>
       Docs: https://containerd.io
      Main PID: 8398 (containerd)
        Tasks: 7
      Memory: 13.5M (peak: 14.0M)
        CPU: 70ms
      CGroup: /system.slice/containerd.service
              └─8398 /usr/bin/containerd

Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15">
Sep 15 07:49:23 ip-172-31-20-171 systemd[1]: Started containerd.ser>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15">
```

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

```
ubuntu@ip-172-31-17-25:~$ 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 libl
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 133 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
Fetched 374 kB in 0s (16.1 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

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

```
ubuntu@ip-172-31-17-25:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0915 17:42:02.713394    9994 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is in 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-172-31-17-25 kubernetes kubernetes.default kubernetes.default.svc kubernetes.local] and IPs [10.96.0.1 172.31.17.25]
[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-172-31-17-25 localhost] and IPs [172.31.17.25 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-17-25 localhost] and IPs [172.31.17.25 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
```

Copy the mkdir and chown commands from the top and execute them.**mkdir -p \$HOME/.kube****sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config****sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config**

```
ubuntu@ip-172-31-20-171:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
[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 post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover 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
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

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

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

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

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

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

```
kubeadm join 172.31.20.171:6443 --token 7acddu.inheshzwxti0372v \
--discovery-token-ca-cert-hash sha256:aed5faf97bac361d1bb7f33a89fb05d2bb28c7fc065024eac2302a734c330a36
```

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-172-31-17-25:~$ 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-172-31-17-25:~$ |
```

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-172-31-17-25:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-17-25:~$ |
```

`kubectl get pods`

```
ubuntu@ip-172-31-17-25:~$ kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fbphm   0/1     Pending   0          37s
nginx-deployment-d556bf558-ljnqb   0/1     Pending   0          37s
ubuntu@ip-172-31-17-25:~$ |
```

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

```
ubuntu@ip-172-31-20-171:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-20-171:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
```

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-172-31-20-171:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-20-171 untainted
ubuntu@ip-172-31-20-171:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE      VERSION
ip-172-31-20-171   Ready   control-plane   5m23s   v1.31.1
```

kubectl get pods

```
ubuntu@ip-172-31-17-25:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-fbphm   1/1     Running   0          14m
nginx-deployment-d556bf558-ljnqb   1/1     Running   0          14m
ubuntu@ip-172-31-17-25:~$ |
```

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

```
ubuntu@ip-172-31-20-171:~$ 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
Handling connection for 8080
```

Step 8: Verify your deployment

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

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

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

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

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

PS C:\Users\bhush\OneDrive\Desktop>New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 07:58:53 UTC 2024

 System load:  0.15      Processes:           152
 Usage of /:   55.3% of 6.71GB  Users logged in:    1
 Memory usage: 20%          IPv4 address for enX0: 172.31.20.171
 Swap usage:   0%

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

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Sun Sep 15 07:27:23 2024 from 152.58.2.47
```

```
ubuntu@ip-172-31-17-25:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 18:03:53 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

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

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

Conclusion:

In this experiment, we successfully installed Kubernetes on an EC2 instance and deployed an Nginx server using Kubectl commands. During the process, we encountered two main errors: the Kubernetes pod was initially in a pending state, which was resolved by removing the control-plane taint using `kubectl taint nodes --all`, and we also faced an issue with the missing `containerd` runtime, which was fixed by installing and starting containerd. We used a **t2.medium EC2 instance with 2 CPUs** to meet the necessary resource requirements for the Kubernetes setup and deployment.

Practical Manual on Terraform

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high level components such as DNS entries, SaaS features, etc.

Terraform can manage infrastructure on multiple cloud platforms. Terraform's state allows you to track resource changes throughout your deployments. You can commit your configurations to version control to safely collaborate on infrastructure. Terraform plugins called providers let Terraform interact with cloud platform and other services via their application programming interfaces (APIs).

A) Installation and Configuration of Terraform in Windows

Step 1: Download terraform

To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

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

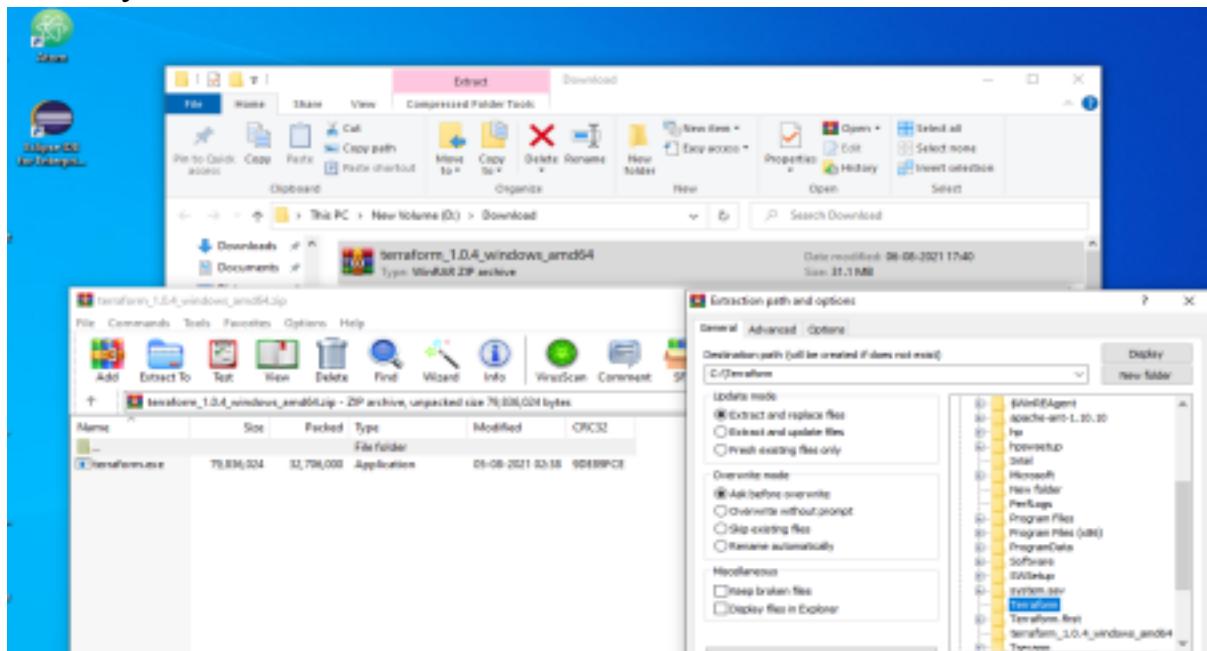
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.

The screenshot shows the Terraform download page for macOS. At the top, there is a purple square icon with a white 'T' logo, followed by the text "Install Terraform" and a dropdown menu showing "1.9.4 (latest)". Below this, the word "macOS" is prominently displayed. Under the heading "Package manager", there is a terminal-style box containing the commands:

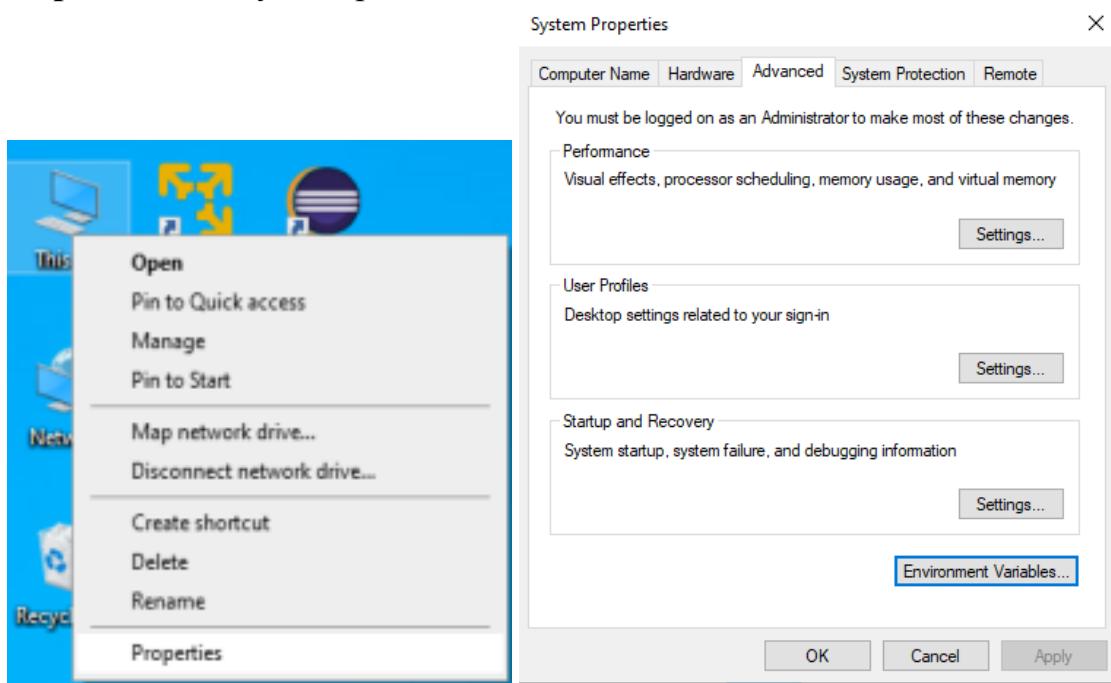
```
brew tap hashicorp/tap
brew install hashicorp/tap/terraform
```

Next to the terminal box is a copy icon. Below this section is another titled "Binary download". Under "Binary download", there are two options: "AMD64" and "ARM64", each with a "Download" button and a downward arrow icon.

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



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



Environment Variables

X

User variables for INFT

Variable	Value
OneDrive	C:\Users\INFT\OneDrive
Path	C:\Users\INFT\AppData\Local\Microsoft\WindowsApps;C:\Users\I...
TEMP	C:\Users\INFT\AppData\Local\Temp
TMP	C:\Users\INFT\AppData\Local\Temp

New...

Edit...

Delete

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

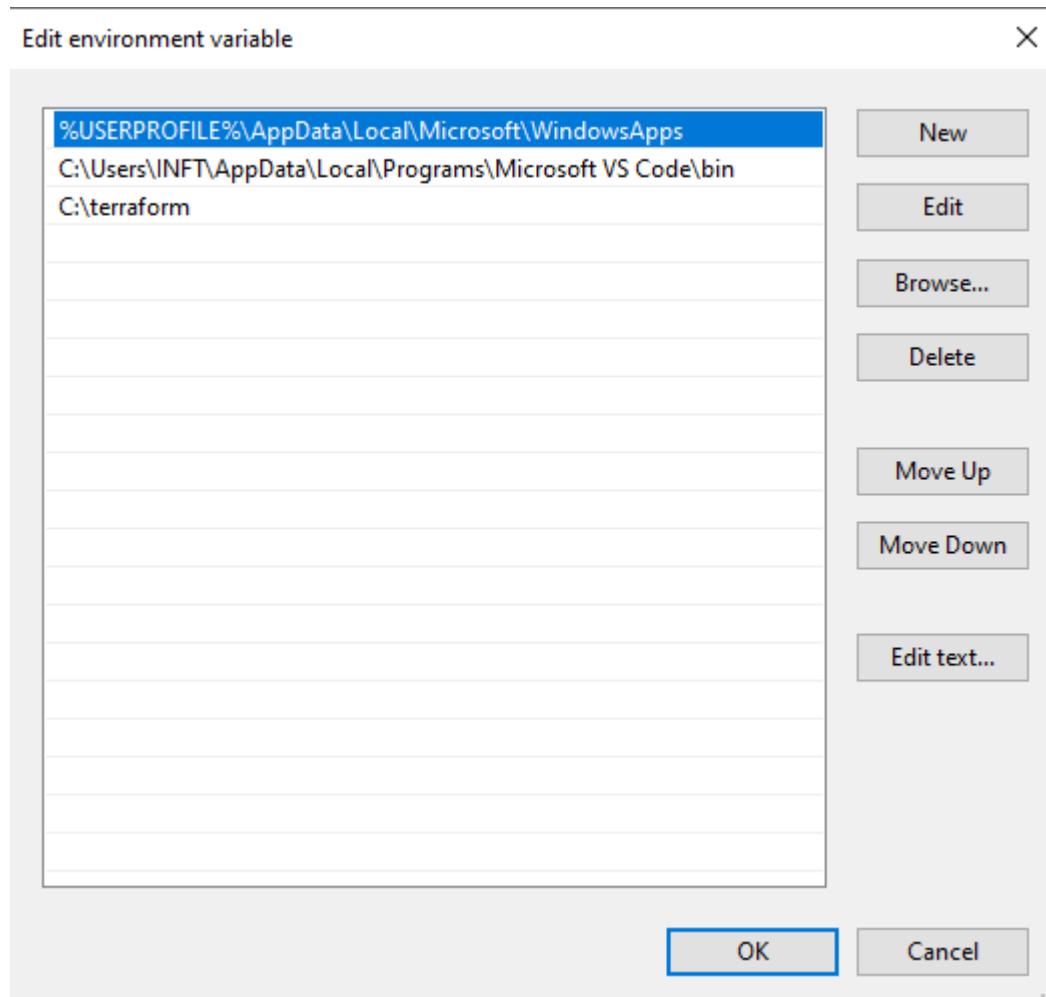
New...

Edit...

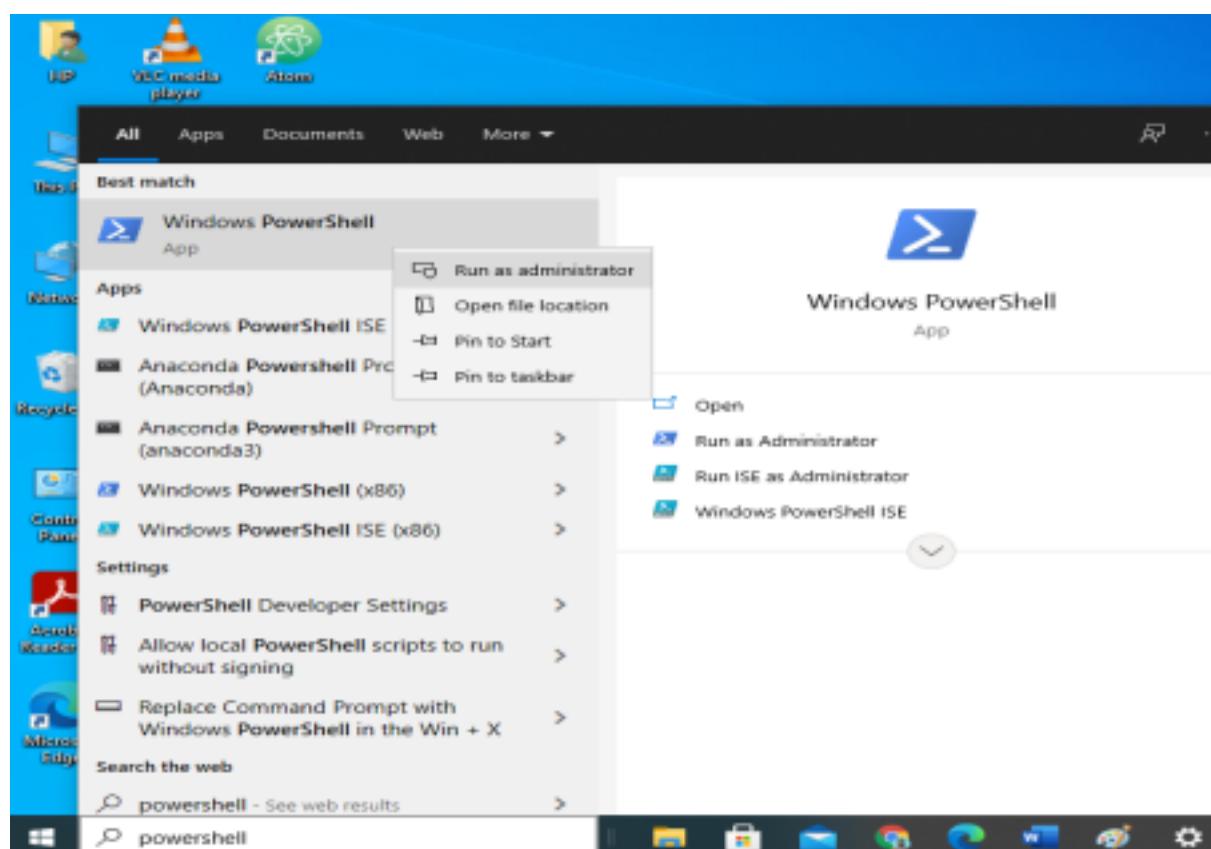
Delete

OK

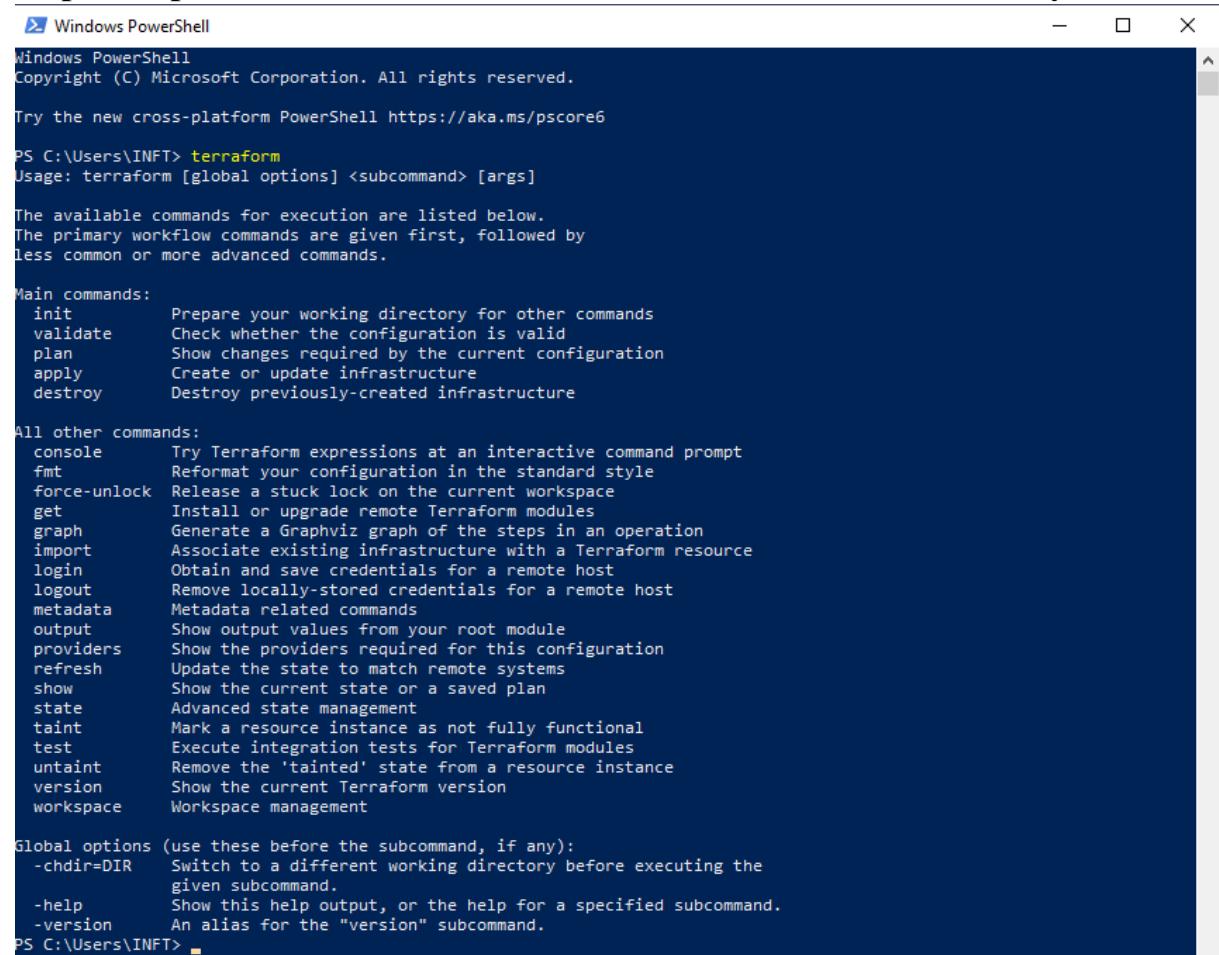
Cancel



Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality



The screenshot shows a Windows PowerShell window with the title 'Windows PowerShell'. The command 'terraform' was run, displaying the usage information and a detailed list of commands and their descriptions. The main commands listed are init, validate, plan, apply, and destroy. Other commands include console, fmt, force-unlock, get, graph, import, login, logout, metadata, output, providers, refresh, show, state, taint, test, untaint, version, and workspace. Global options like -chdir=DIR, -help, and -version are also listed.

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

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

PS C:\Users\INFT> 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.

PS C:\Users\INFT>
```

Note: If any error comes, then please recheck or set the path of Terraform in Environment variable again.

Experiment No.: 6

Implementation:

A. Creating docker image using terraform

Prerequisite:

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

Step 1: Check the docker functionality

```
Microsoft Windows [Version 10.0.22621.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\athar>docker --version
Docker version 27.1.1, build 6312585
```

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

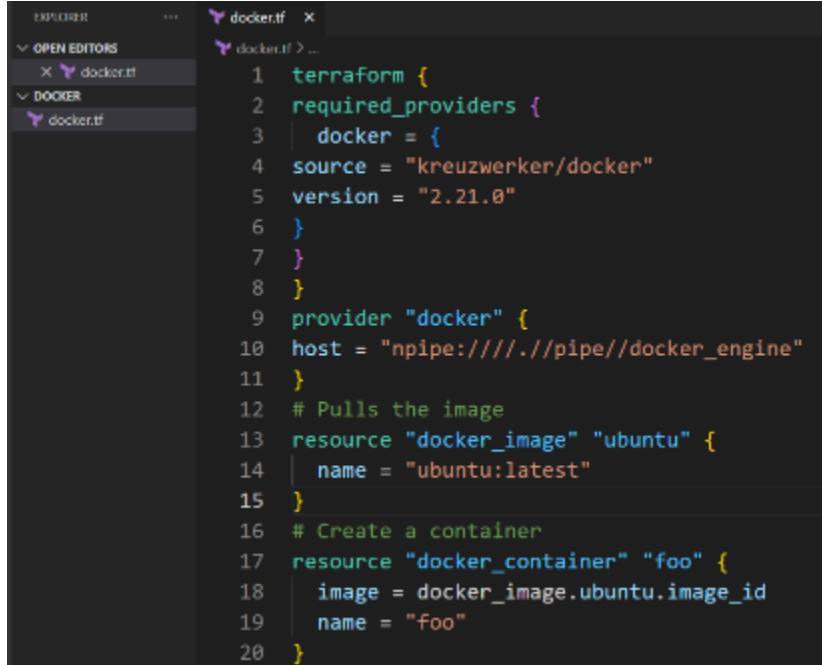
```
{ required_providers
  docker = {
    source = "kreuzwerker/docker"
    version = "2.21.0"
  }
}
```

```
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"
}
```



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

Step 3: Execute Terraform Init command to initialize the resources

```

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 BD080C4571C
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it
at 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,
commands will detect it and remind you to do so if necessary.
PS D:\all code\Terraform Scripts\Docker> 
```

Step 4: Execute Terraform plan to see the available resources

```

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]
} 
```

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”

```

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]
} 
```

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:

```
PS E:\Terraform Script\Docker> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
docker18tutorial   latest   e5bc7d28c9a8  38 minutes ago  28.9MB
nanavjawrani/docker18tutorial   latest   e5bc7d28c9a8  38 minutes ago  28.9MB
ubuntu              latest   2dc39ba859dc  2 weeks ago   77.8MB
alpine/git          latest   692618a8d74d  2 weeks ago   43.4MB
PS E:\Terraform Script\Docker> |
```

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

```
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Terraform> terraform destroy
docker_image.ubuntu: Refreshing state... [id:sha256:edbfef4c1f8a3981ce892e157cf29ea84ed33ebff8cd065b19b6a7761c299beubuntu:latest]
docker_container.foo: Refreshing state... [id:e001752417e17594f751208bd618354cb1b93037e0ad002301c1a671a29599]

Terraform used the selected provider to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create
  - destroy
  ! update

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
  - cpushares       = 8 => null
  - dns              = [] => null
  - dns_aaaa         = [] => null
  - dns_ns           = [] => null
  - dns_search       = [] => null
  - entrypoint       = [] => null
  - env              = [] => null
  - gateway         = "172.17.0.1" => null
  - group_ids        = [] => null
  - hostname         = "e001752417e17594f751208bd618354cb1b93037e0ad002301c1a671a29599" => null
  - id               = "e001752417e17594f751208bd618354cb1b93037e0ad002301c1a671a29599" => null
  - image            = "sha256:ef79c61f8a3591ce6d137c438a4d38a6f1c9d6719b6a7761c299be" => null
  - init             = false => null
  - ip_address       = "172.17.0.2" => null
  - ip_address_length = 16 => null
  - ip_mode          = "private" => null
  - links            = [] => null
  - log_driver       = "json-file" => null
  - log_opts          = [] => null
  - log_stream        = false => null
  - max_retry_count  = 0 => null
  - memory           = 8 => null
  - memory_swap      = 8 => null
  - restart_policy   = "true" => null
}
```

Docker images After Executing Destroy step

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

Logout

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

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

The screenshot shows the Jenkins dashboard with a sidebar on the left containing links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. A 'Build Queue' section indicates 'No builds in the queue.' Below the sidebar is a 'Build Executor Status' section showing 0/2 executors. The main area displays a table of projects with columns for Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration. The projects listed are My_First_Maven, MyPipeline1, Pipeline_01, and WebDriver.

S	W	Name	Last Success	Last Failure	Last Duration
Green	Cloud	My_First_Maven	23 days #2	23 days #1	20 sec
Green	Sun	MyPipeline1	28 days #1	N/A	9.2 sec
Green	Sun	Pipeline_01	1 mo 15 days #3	N/A	9.9 sec
Green	Cloud	WebDriver	1 day 16 hr #5	1 day 16 hr #4	13 sec

2. Run SonarQube in a Docker container using this command :-
a] docker -v
b] docker pull sonarqube
c] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Users\lauki> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
bd819c9b5ead: Download complete
80338217a4ab: Download complete
1a5fd5c7e184: Download complete
4f4fb700ef54: Download complete
7b87d6fa783d: Downloading [======>
7478e0ac0f23: Download complete
90a925ab929a: Download complete
7d9a34308537: Download complete
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is “**aditya**”.

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

4. Create a local project in SonarQube with the name **sonarqube**

1 of 2

Create a local project

Project display name *

 sonarqube 

Project key *

 sonarqube 

Main branch name *

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

2 of 2

X

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to 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

5. Setup the project and come back to Jenkins Dashboard. Go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner** in **Available Plugins** and install it.

The screenshot shows the Jenkins plugin marketplace. A search bar at the top contains the text "SonarQube Scanner". Below the search bar, there is a "Install" button with a blue background and white text, and a refresh icon. A dropdown menu is open, showing the "SonarQube Scanner" plugin. The plugin card includes the name "SonarQube Scanner 2.17.2", a "Released" status, and a timestamp "7 mo 9 days ago". It also features tabs for "External Site/Tool Integrations" and "Build Reports". A brief description states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality."

6. Under '**Manage Jenkins → System**', look for **SonarQube Servers** and enter these details.

Name : sonarqube

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

The screenshot shows the "SonarQube installations" configuration page. It has a "Name" field containing "sonarqube" and a "Server URL" field containing "http://localhost:9000". Below these fields is a "Server authentication token" section with a dropdown menu set to "- none -" and a "+ Add" button. At the bottom left, there is an "Advanced" button with a dropdown menu.

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

Manage Jenkins → Tools → SonarQube Scanner Installation



Add SonarQube Scanner

SonarQube Scanner

Name
sonarqube

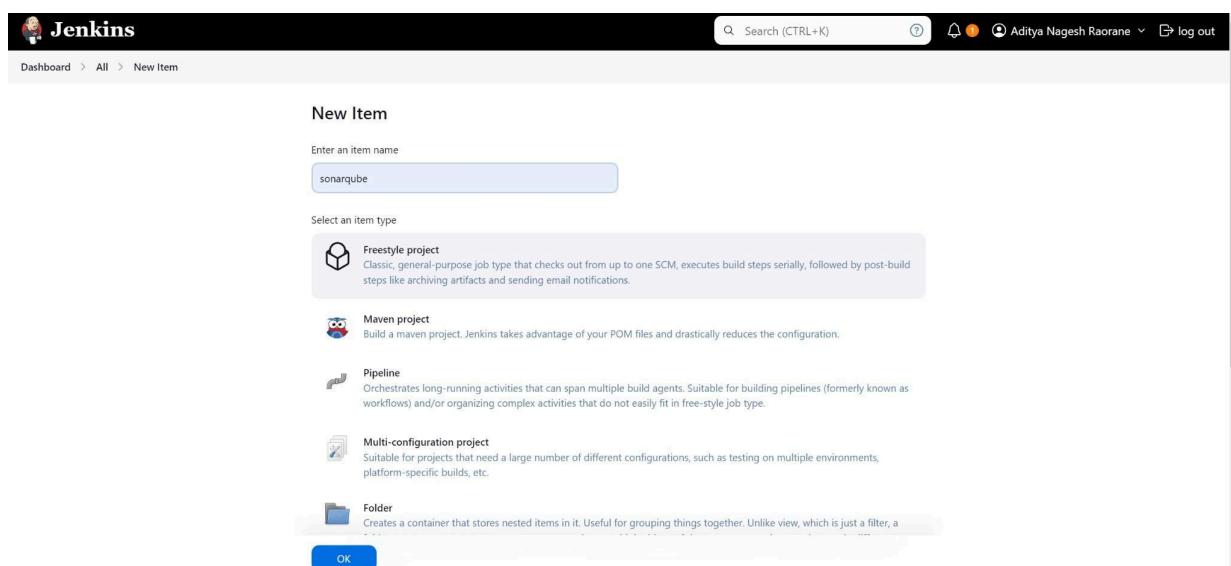
Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.2.0.4584

Add Installer ▾

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



Jenkins

Dashboard > All > New Item

New Item

Enter an item name
sonarqube

Select an item type

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

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

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

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

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

OK

9. Choose this GitHub repository in **Source Code Management**.

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

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

The screenshot shows the SonarQube configuration interface for a project named 'sonarqube'. The left sidebar has sections: General, Source Code Management (selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main area is titled 'Source Code Management' and shows 'Git' selected as the provider. Under 'Repositories', there is one entry with the URL 'https://github.com/shazforiot/MSBuild_firstproject.git'. Below it, 'Credentials' are set to 'none'. There are 'Add Repository' and 'Advanced' buttons. At the bottom are 'Save' and 'Apply' buttons.

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

sonar.projectKey=sonarqube

sonar.login=admin

sonar.password=aditya

sonar.sources=.

sonar.host.url=http://localhost:9000

Analysis properties ?

```
sonar.projectKey=sonarqube
sonar.login=admin
sonar.password=laukik
sonar.sources=-
sonar.host.url=http://localhost:9000
```

Additional arguments ?

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

The screenshot shows the SonarQube administration interface under the 'Global Permissions' section. It lists four groups: 'sonar-administrators', 'sonar-users', 'Anyone DEPRECATED', and 'Administrator admin'. The 'Administrator admin' group has 'Execute Analysis' checked for 'Quality Gates' and 'Quality Profiles'. Other groups have 'Execute Analysis' checked for 'Projects'.

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

12. Run The Build and check the console output.

The screenshot shows the Jenkins interface for the 'sonarqube' project. At the top, there's a navigation bar with a user icon, search bar, and log out link. Below the header, the project name 'sonarqube' is displayed with a green checkmark. To the left, there's a sidebar with various project management options like Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main content area shows a 'Builds' section with a table containing one row: a green checkmark icon, the build number '#1', and the timestamp '8:17PM'. On the right side of the main content area, there are three buttons: 'Download', 'Copy', and 'View as plain text'.

Console Output

Started by user laukik padgaonkar
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject
> git.exe --version # timeout=10
> git --version # 'git version 2.43.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
[sonarqube] \$ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=laukik -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
18:54:47.035 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
18:54:47.044 INFO Scanner configuration file:

Dashboard > sonarqube > #1 > Console Output

```

20:18:52.472 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
20:18:52.473 INFO Sensor C# [csharp] (done) | time=2ms
20:18:52.474 INFO Sensor Analysis Warnings import [csharp]
20:18:52.478 INFO Sensor Analysis Warnings import [csharp] (done) | time=4ms
20:18:52.479 INFO Sensor C# File Caching Sensor [csharp]
20:18:52.482 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
20:18:52.482 INFO Sensor C# File Caching Sensor [csharp] (done) | time=4ms
20:18:52.483 INFO Sensor Zero Coverage Sensor
20:18:52.510 INFO Sensor Zero Coverage Sensor (done) | time=28ms
20:18:52.515 INFO SCM Publisher SCM provider for this project is: git
20:18:52.518 INFO SCM Publisher 4 source files to be analyzed
20:18:53.806 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=1286ms
20:18:53.810 INFO CPD Executor Calculating CPD for 0 files
20:18:53.811 INFO CPD Executor CPD calculation finished (done) | time=0ms
20:18:53.822 INFO SCM revision ID: 'f2bc842c84d6e72427c388bcaceed6d6fee7b549adef'
20:18:54.975 INFO Analysis report generated in 240ms, dir size=201.0 kB
20:18:55.237 INFO Analysis report compressed in 114ms, zip size=22.4 kB
20:18:55.614 INFO Analysis report uploaded in 374ms
20:18:55.618 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
20:18:55.621 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:18:55.622 INFO More about the report processing at http://localhost:9000/api/ce/task?id=a2e28c04-ce64-4689-8023-5b03ea519fc9
20:18:55.653 INFO Analysis total time: 39.158 s
20:18:55.658 INFO SonarScanner Engine completed successfully
20:18:55.741 INFO EXECUTION SUCCESS
20:18:55.743 INFO Total time: 58.785s
Finished: SUCCESS

```

REST API Jenkins 2.473

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

The screenshot shows the SonarQube search interface. At the top, there is a search bar with the placeholder "Search for projects...". Below the search bar are several filters: "Perspective", "Overall Status", "Sort by", and "Name". On the right side, it shows "1 project(s)" with a house icon. The main content area displays a project card for "sonarqube PUBLIC". The card indicates the last analysis was 2 minutes ago and notes that the main branch is empty. A green checkmark icon with the word "Passed" is present. The overall status is shown as "Passed".

The screenshot shows the SonarQube project overview for the "main" branch. The top navigation bar includes links for "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", "More", and a search bar. Below the navigation, there are tabs for "Overview", "Issues", "Security Hotspots", "Measures", "Code", and "Activity". Project settings and information are also available. The main content area displays various quality gate metrics. A prominent green checkmark indicates the "Passed" status of the quality gate. It also mentions that the last analysis had warnings. Below this, there are sections for "New Code" and "Overall Code" under the "Reliability" category. Other metrics shown include Security (0 Open issues), Maintainability (0 Open issues), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (0.0%).

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

Conclusion:

In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.

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.

1. Open up Jenkins Dashboard on localhost:8080.

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

- Header:** Jenkins logo, search bar (Search (CTRL+K)), notifications, user info (Aditya Nagesh Raorane), and log out button.
- Left Sidebar:**
 - New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
 - My Views
- Build Queue:** Shows "No builds in the queue."
- Build Executor Status:** Shows "0/2" available executors.
- Job List:** A table listing five build jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	My_First_Maven	23 days #2	23 days #1	20 sec
✓	Sun	MyPipeline1	28 days #1	N/A	9.2 sec
✓	Sun	Pipeline_01	1 mo 15 days #3	N/A	9.9 sec
✓	Sun	sonarqube	13 min #1	N/A	1 min 2 sec
✓	Cloud	WebTestDriver	1 day 18 hr #5	1 day 18 hr #4	13 sec
- Bottom:** Icon selection dropdown (Icon: S M L) and a "ooo" link.

2. Run SonarQube in a Docker container using this command:
 - a] docker -v
 - b] docker pull sonarqube
 - c] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\adity>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\adity>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
4a6e73f4472de892b1ddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is

“**aditya**”.

4. Create a local project in SonarQube with the name **sonarqube-test**.

1 of 2

Create a local project

Project display name *

sonarqube-test

Project key *

sonarqube-test

Main branch name *

main

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

[Cancel](#)

[Next](#)

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to 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 become part of the overall code.

Recommended for projects following continuous delivery.

Reference branch

Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

[Back](#)

[Create project](#)

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins 'New Item' creation interface. In the 'Enter an item name' field, the text 'sonarqube-test' is entered. Below it, the 'Select an item type' section lists several options: 'Freestyle project' (selected), 'Maven project', 'Pipeline' (highlighted in blue), 'Multi-configuration project', and 'Folder'. At the bottom of the form is a blue 'OK' button.

7. Under **Pipeline Script**, enter the following -

```
node {  
    stage('Cloning the GitHub Repo')  
    {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
    stage('SonarQube analysis') {  
        withSonarQubeEnv('sonarqube') {  
            bat  
            "C:\\\\Users\\\\adity\\\\Downloads\\\\sonar-scanner-cli-6.1.0.4477-windows-x64\\\\sonar-  
            s scanner-6.1.0.4477-windows-x64\\\\bin\\\\sonar-scanner.bat" \\\n                -D sonar.login=<YOUR ID> \\\n                -D sonar.password=<YOUR PASSWORD> \\\n                -D sonar.projectKey=<YOUR PROJECT KEY> \\\n                -D sonar.exclusions=vendor/**,resources/**, */*.java \\\n                -D sonar.host.url=http://localhost:9000/"  
        }  
    }  
}
```

The screenshot shows the Jenkins Pipeline configuration page for a project named 'sonarqube-test'. The 'Pipeline' tab is selected under 'Definition'. A Groovy script is pasted into the 'Script' editor:

```
1 * node {
2     stage('Cloning the GitHub Repo') {
3         git 'https://github.com/charforiot/gol.git'
4     }
5     stage('SonarQube analysis') {
6         withSonarQubeEnv('sonarqube') {
7             sh "C:\Users\aditya\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner.bat \
8             -D sonar.login=admin \
9             -D sonar.password=aditya \
10            -D sonar.projectKey=sonarqube-test \
11            -D sonar.exclusions=vendor/resources/**, **/*.java \
12            -D sonar.host.url=http://localhost:9000/"
13        }
14    }
15 }
```

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

REST API Jenkins 2.473

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

8. Run The Build.

The screenshot shows the Jenkins dashboard for the 'sonarqube-test' project. The status is 'Idle'. Below the status, there are two buttons: '</> Changes' and '▷ Build Now'.

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

The screenshot shows the Jenkins Pipeline interface for the 'sonarqube-test' pipeline. On the left, there's a sidebar with various options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. The main area is titled 'Stage View' and displays two stages: 'Cloning the GitHub Repo' (1s) and 'SonarQube analysis' (14min 37s). Below this is a 'Permalinks' section with links to the last four builds. A 'Builds' section shows a single build (#1) from Sep 17 at 21:26 with 'No Changes'. The bottom section is titled 'Console Output' and shows a log of WARN messages about duplicate references in JMeter documentation files. Buttons for Download, Copy, and View as plain text are available in the top right of the console output area.

Console Output

Skiping 4,252 KB. [Full Log](#)

```
20:16:42.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.  
20:16:42.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.  
20:16:42.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.  
20:16:42.864 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.  
20:16:42.865 WARN Too many duplication references on file gameoflife-
```

Dashboard > sonarqube-test > #1

```

line 41. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 17. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 296. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 75. Keep only the first 100 references.
21:37:59.930 INFO CPD Executor CPD calculation finished (done) | time=153336ms
21:37:59.955 INFO SCH revision ID 'ba7990a7e1b576f04a4612322d0412c5e6e1e5e4'
21:40:14.276 INFO Analysis report generated in 5151ms, dir size=127.2 MB
21:40:35.678 INFO Analysis report compressed in 21388ms, zip size=29.6 MB
21:40:36.178 INFO Analysis report uploaded in 492ms

21:40:36.173 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
21:40:36.173 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:40:36.173 INFO More about the report processing at http://localhost:9000/api/ce/task?id=99fc1e5-df4e-4f9f-8688-3169741e0856
21:40:53.466 INFO Analysis total time: 14:32.336 s
21:40:53.468 INFO SonarScanner Engine completed successfully
21:40:54.148 INFO EXECUTION SUCCESS
21:40:54.150 INFO Total time: 14:35.645s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

REST API Jenkins 2.473

10. After that, check the project in SonarQube.

The screenshot shows the SonarQube interface with the following details:

- Projects:** sonarqube PUBLIC, sonarqube-test PUBLIC
- Status:** Passed (green checkmark)
- Filters:**
 - Quality Gate:** Passed (2), Failed (0)
 - Reliability:** A (1), B (0), C (1), D (0), E (0)
 - Security:** A (2)
- Metrics for sonarqube-test PUBLIC:**
 - Last analysis: 16 minutes ago - 683k Lines of Code, HTML, XML, ...
 - Security: 0
 - Reliability: 68k
 - Maintainability: 164k
 - Hotspots Reviewed: 0.0%
 - Coverage: 50.6%
 - Duplications: 0

The screenshot shows the SonarQube main dashboard for the 'main' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. The dashboard header indicates 683k Lines of Code and a Version not provided. A prominent green 'Passed' badge with a checkmark is displayed. Below it, a message states 'The last analysis has warnings. See details'. The dashboard is divided into several sections: Security (0 Open issues), Reliability (68k Open issues, C grade), Maintainability (164k Open issues, A grade), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6% on 759k lines). Buttons for New Code and Overall Code are present.

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

11. Code Problems - Open

The screenshot shows the SonarQube Measures tab for the 'main' project. The left sidebar includes links for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'Measures' tab is selected. The right panel displays a sidebar with categories like Security Review, Duplications, Size, Complexity, and Issues. The 'Issues' category is expanded, showing sub-sections for Open Issues (210,549), Confirmed Issues (0), Accepted Issues (0), and False Positive Issues (0). The main content area shows a tree view of the project structure under 'sonarqube-test', with nodes for gameoflife-acceptance-tests (4 issues), gameoflife-build (0 issues), gameoflife-core (603 issues), gameoflife-deploy (0 issues), gameoflife-web (209,940 issues), and pom.xml (2 issues). View options like Tree, Select files, and Navigate are available at the top of the tree view.

Issues

Consistency

sonarqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Filters **Clear All Filters**

Issues in new code

Clean Code Attribute 1 **Consistency** 197k

- Intentionality 14k
- Adaptability 0
- Responsibility 0

Add to selection **Ctrl + click**

Software Quality

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

- Insert a <!DOCTYPE> declaration to before this <html> tag. **Consistency** **user-experience** L1 - 5min effort - 4 years ago - ⚡ Bug - ⚡ Major
- Remove this deprecated "width" attribute. **Maintainability** html5 obsolete L9 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Remove this deprecated "align" attribute. **Maintainability** html5 obsolete L11 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

Embedded database should be used for evaluation purposes only

Intentionality

sonarqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Filters **Clear All Filters**

Issues in new code

Clean Code Attribute 1 **Intentionality** 14k

- Consistency 197k
- Adaptability 0
- Responsibility 0

Add to selection **Ctrl + click**

Software Quality

gameoflife-acceptance-tests/Dockerfile

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

Embedded database should be used for evaluation purposes only

Code Smells

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Severity: High (0), Medium (0), Low (253)

Type: Bug (14k), Vulnerability (0), Code Smell (253)

Scope, Status, Security Category

Bulk Change, Select issues, Navigate to issue, 253 issues, 2d 5h effort

gameoflife-web/tools/jmeter/printable_docs/building.html

Add an "alt" attribute to this image. Reliability: Low. Intentionality: accessibility wcag2-a. L29 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Minor

gameoflife-web/tools/jmeter/printable_docs/changes.html

Add an "alt" attribute to this image. Reliability: Low. Intentionality: accessibility wcag2-a. L31 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Minor

gameoflife-web/tools/jmeter/printable_docs/changes_history.html

Add an "alt" attribute to this image. Intentionality: accessibility wcag2-a.

Embedded database should be used for evaluation purposes only

Bugs

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Severity: High (0), Medium (14k), Low (0)

Type: Bug (14k), Vulnerability (0), Code Smell (253)

Scope, Status, Security Category

Bulk Change, Select issues, Navigate to issue, 13,619 issues, 56d effort

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element. Reliability: Low. Intentionality: accessibility wcag2-a. L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Add "<th>" headers to this "<table>". Reliability: Low. Intentionality: accessibility wcag2-a. L9 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element. Reliability: Low. Intentionality: accessibility wcag2-a.

Embedded database should be used for evaluation purposes only

Reliability

Clean Code Attribute

- Consistency: 33k
- Intentionality:** 14k
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

Software Quality

- Security: 0
- Reliability:** 14k
- Maintainability: 0

Severity

- High: 0
- Medium: 14k

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element **Intentionality**
Reliability

Open Not assigned

L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Add "<th>" headers to this "<table>". **Intentionality**
Reliability

Open Not assigned

L9 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element **Intentionality**
Reliability

Open Not assigned

L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Embedded database should be used for evaluation purposes only

Duplicates

Project Overview

Security

Reliability

Maintainability

Security Review

Duplications

Overview

Overall Code

Density: 50.6%

sonarqube-test

Duplicated Lines (%): 50.6% [See history](#)

View as: List 1,147 files

	Duplicated Lines (%)	Duplicated Lines
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../ReportCellRenderer.html	92.4%	1,282
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../RightAlignRenderer.html	92.4%	1,198
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../JMeterCellRenderer.html	92.1%	1,281
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../FunctionHelper.html	89.5%	1,070
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../AjpSamplerGui.html	89.0%	1,219

Security Hotspot

The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Security Hotspots' tab is selected. A prominent alert states: 'The tomcat image runs with root as the default user. Make sure it is safe here.' Below this, a 'Status: To review' message indicates the hotspot needs assessment. The 'Review priority' is set to 'Medium' under the 'Permission' category. The code snippet shown is a Dockerfile:

```

FROM tomcat:8-jre8
The tomcat image runs with root as the default user. Make sure it is safe here.

RUN rm -rf /usr/local/tomcat/webapps/*
COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war
EXPOSE 8080
CMD ["catalina.sh", "run"]

```

Cyclomatic Complexity

The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Measures' tab is selected. On the left, a sidebar lists various quality metrics, with 'Cyclomatic Complexity' highlighted. The main panel displays a tree view of files and their complexity counts:

- sonarqube-test (1,112)
- gameoflife-acceptance-tests
- gameoflife-build
- gameoflife-core (18)
- gameoflife-deploy
- gameoflife-web (1,094)
- pom.xml

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

Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, and security vulnerabilities on our sample Java application.

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

Theory:

What is Nagios?

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

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

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

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

Features of Nagios

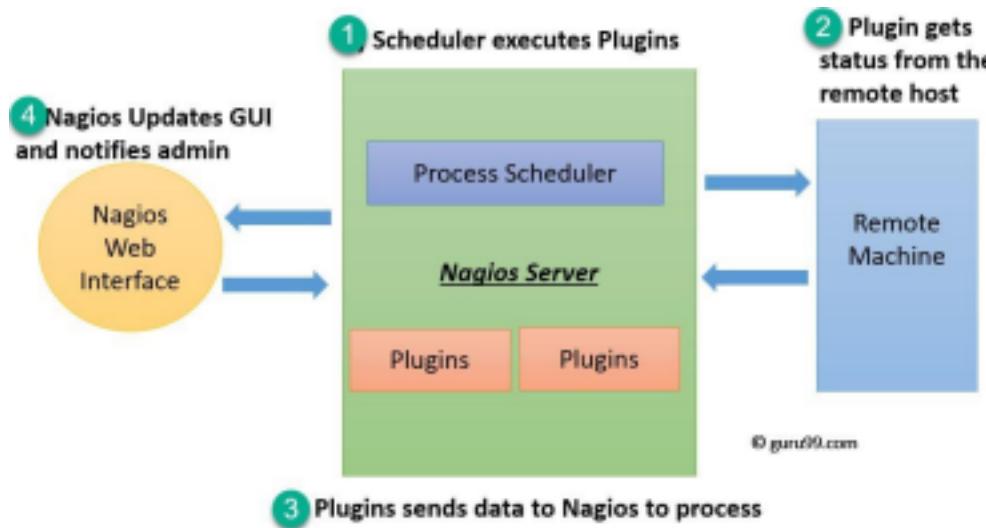
Following are the important features of Nagios monitoring tool:

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

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

Nagios Architecture

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



1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.

2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

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

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, and Instances. The main area is titled 'Instances (1) Info' and shows one instance: 'nagios-host' (Instance ID: i-028182fbe9c070820), which is 'Running' (status check: Initializing) and has an 't2.micro' instance type. There are buttons for 'Connect', 'Actions', and 'Launch instances'.

- Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.
You have to edit the inbound rules of the specified Security Group for this.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-04093d1a208295e38	SSH	TCP	22	C... 0.0.0.0/0	
-	HTTP	TCP	80	A... 0.0.0.0/0	
-	HTTPS	TCP	443	A... 0.0.0.0/0	
-	All ICMP - IPv4	ICMP	All	A... 0.0.0.0/0	

Add rule

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

Cancel Preview changes Save rules

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



4. Update the package indices and install the following packages using yum

```
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
yum install: error: unrecognized arguments: -Oy
[ec2-user@ip-172-31-38-150 ~]$ sudo yum install gd gd-devel -y
Last metadata expiration check: 0:05:12 ago on Sun Oct 6 11:15:04 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128
fnts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18
jbigkit-libs	x86_64	2.1-21.amzn2023.0.2	amazonlinux	54
langpacks-core-font-en	noarch	3.0-21.amzn2023.0.4	amazonlinux	10
libICE	x86_64	1.0.10-6.amzn2023.0.2	amazonlinux	71
libSM	x86_64	1.2.3-8.amzn2023.0.2	amazonlinux	42

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
```

```
Complete!
[ec2-user@ip-172-31-38-150 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-38-150 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-38-150 ~]$ █
```

6. Create a new user group

```
sudo groupadd nagcmd
```

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
```

8. Create a new directory for Nagios downloads

```

mkdir ~/downloads
cd ~/downloads
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-38-150 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-38-150 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-38-150 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-38-150 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-38-150 ~]$ cd ~/downloads
[ec2-user@ip-172-31-38-150 downloads]$ wget https://go.nagios.org/24-09-17/6kqcx

```

9. Use wget to download the source zip files.

```

wget https://go.nagios.org/1/975333/2024-09-17/6kqcx
wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz

```

```

[ec2-user@ip-172-31-38-150 downloads]$ wget https://go.nagios.org/24-09-17/6kqcx
--2024-10-06 11:23:50-- https://go.nagios.org/1/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org) ... 3.92.127.219, ...
Connecting to go.nagios.org (go.nagios.org) |3.92.1| HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagios.tar.gz?utm_source=Nagios.org&utm_content=Download+5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc3
.....
```

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

```

tar zxvf nagios-4.5.5.tar.gz
cd nagios-4.5.5
[ec2-user@ip-172-31-38-150 nagios-4.5.5]$ wget https://go.nagios.org/1/975333/2024-09-17/6kqcx
--2024-10-06 11:23:50-- https://go.nagios.org/1/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org) ... 3.92.120.28, 52.54.96.194, 3.21572.219, ...
Connecting to go.nagios.org (go.nagios.org) |3.92.120.28|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e24439d4a81d8 [following]
--2024-10-06 11:23:50-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f011e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com) ... 45.79.49.120, 2600:3c00:03c:92ff:feff:45ce
Connecting to assets.nagios.com (assets.nagios.com) |45.79.49.120|:80... con

```

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

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

12. Compile the source code.

```
make all
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". It displays the configuration summary for Nagios 4.5.5, followed by the output of the "make all" command. The configuration summary includes details like Nagios executable, user/group, command broker, and web interface URLs. The "make all" command output shows the compilation of various source files (nagios.c, broker.c, nebmods.c, common/shared.c, query-handler.c, workers.c) with gcc flags. A warning message is present in the compilation log.

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

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

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

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

[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
  |
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
```

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-172-31-38-150 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagio
s.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagi
os.conf; \
fi

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

[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

```
sudo nano /usr/local/nagios/etc/objects/contacts.cfg
=====
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo yum install openssl
evel
Last metadata expiration check: 0:13:31 ago on Sun Oct  6 11:15:04 2024.
Dependencies resolved.
=====
=====
Package           Architecture      Version
Repository        Size
=====
Installing:
openssl-devel      x86_64          1:3.0.8-1.amzn2023.0.14
amazonlinux        3.0 M
```

Transaction Summary

14. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagio
s.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagi
os.conf; \
fi

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

[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

15. 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-172-31-38-150 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
[sudo] password for nagiosadmin:
[sudo] password for nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

16. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

17. 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-172-31-38-150 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
```

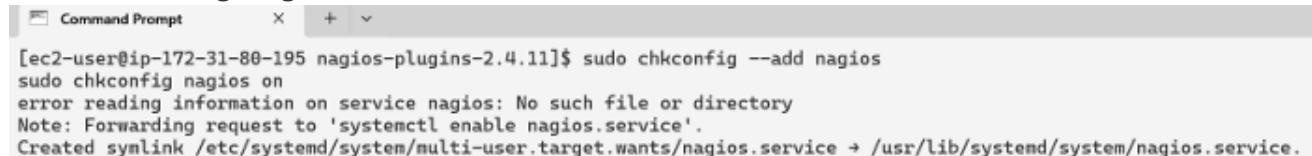
18. Compile and install plugins

```
cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
[ec2-user@ip-172-31-38-150 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
[ec2-user@ip-172-31-38-150 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
```

19. Start Nagios

Add Nagios to the list of system services

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



```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios  
sudo chkconfig nagios on  
error reading information on service nagios: No such file or directory  
Note: Forwarding request to 'systemctl enable nagios.service'.  
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

Verify the sample configuration files

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

```
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v  
/usr/local/nagios/etc/nagios.cfg  
/usr/local/nagios/etc/nagios.cfg  
  
Nagios Core 4.4.6  
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  
Copyright (c) 1999-2009 Ethan Galstad  
Last Modified: 2020-04-28  
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.
```

If there are no errors, you can go ahead and start Nagios.

```
sudo service nagios start
```

```
Things look okay - No serious problems were detected during the pre-flight check  
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo service nagios start  
Redirecting to /bin/systemctl start nagios.service  
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$
```

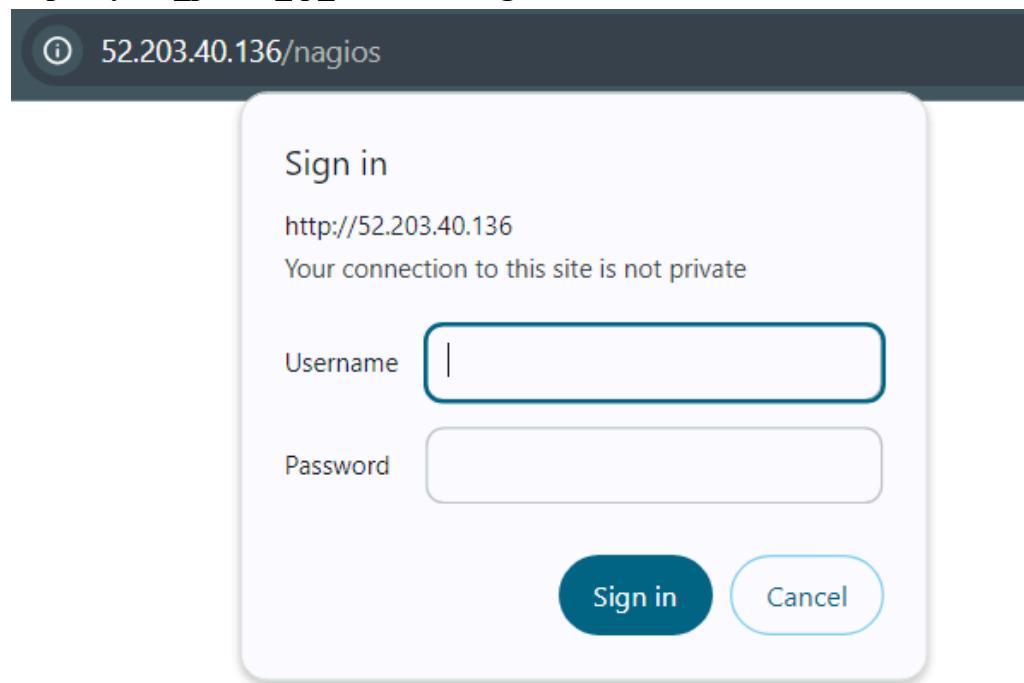
20. Check the status of Nagios

```
sudo systemctl status nagios
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
    Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
    Active: active (running) since Sun 2024-10-06 11:51:46 UTC; 1min 33s ago
      Docs: https://www.nagios.org/documentation
      Main PID: 89956 (nagios)
        Tasks: 6 (limit: 1112)
       Memory: 2.4M
          CPU: 36ms
        CGroup: /system.slice/nagios.service
                ├─89956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                ├─89957 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89958 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89959 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89960 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                └─89961 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 11:51:46 ip-172-31-38-150.ec2.internal nagios[89956]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfull
Oct 06 11:51:46 ip-172-31-38-150.ec2.internal nagios[89956]: qh: core query handler ready
```

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

22. 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.

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



The screenshot shows the Nagios Core 4.4.6 dashboard. At the top right, it displays "Nagios® Core™ Version 4.4.6" and the date "April 28, 2020". A prominent green checkmark indicates "Daemon running with PID 89956". Below this, a blue banner announces "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.5." On the left sidebar, there are links for General (Home, Documentation), Current Status (Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends (Legacy), Alerts, Notifications, Event Log), and a Quick Search bar. The main content area includes sections for Get Started (with a bulleted list of items like "Start monitoring your infrastructure"), Quick Links (with links to Nagios Library, Labs, Exchange, Support, .com, and .org), Latest News, and Don't Miss... (both currently empty).

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

Conclusion: We have successfully installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux machine. This enables us to effectively manage system performance and proactively address potential issues.

Advanced DevOps

Lab Experiment 10

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

Nagios. Steps:

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running **on the server side**, run this *sudo systemctl status nagios* on the “NAGIOS HOST”.

```
ec2-user@ip-172-31-81-4 ~ + × - × × Amazon Linux 2023 https://aws.amazon.com/linux/amazon-linux-2023
Last login: Mon Sep 23 15:36:29 2024 from 152.58.4.81
[ec2-user@ip-172-31-81-4 ~]$ 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 Mon 2024-09-23 16:32:36 UTC; 18min ago
     Docs: https://www.nagios.org/documentation
 Process: 1966 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1971 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1972 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 6.8M
    CPU: 320ms
   CGroup: /system.slice/nagios.service
           ├─1972 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─1974 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1975 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1976 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1977 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─1983 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

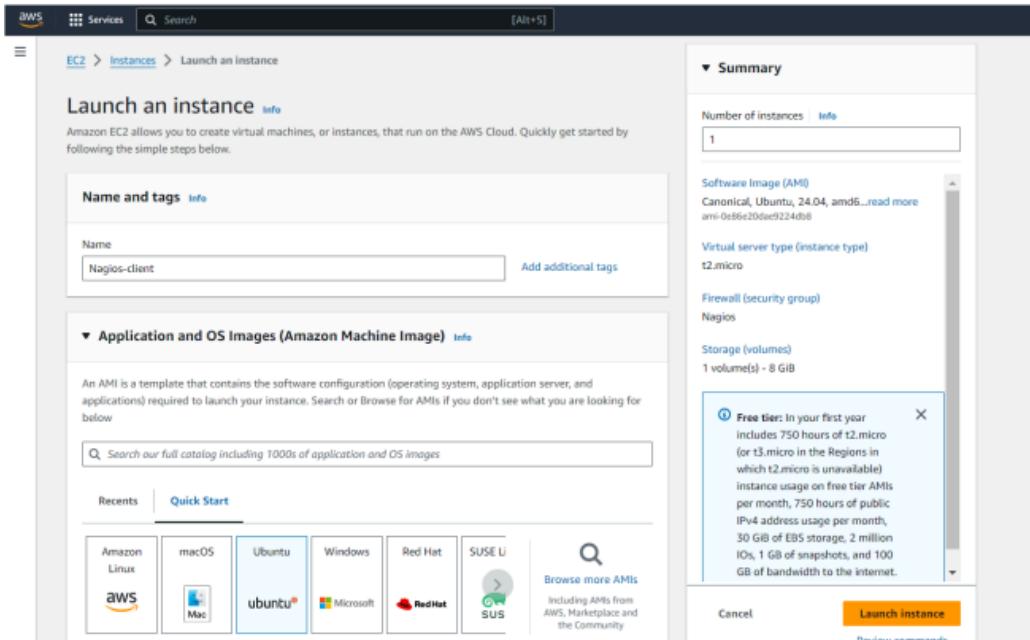
Sep 23 16:32:36 ip-172-31-81-4.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
```

You can proceed if you get this message.

- ## 2. Before we begin,

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

Provide it with the same security group as the Nagios Host and name it ‘linux-client’ alongside the host.



For now, leave this machine as is, and go back to your nagios HOST machine.

3. On the server, run this command

```
ps -ef | grep nagios
Last login: Sat Oct  5 16:58:17 2024 from 42.111.112.18
[ec2-user@ip-172-31-43-65 ~]$ ps -ef | grep nagios
nagios    97412      1  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios
nagios    97413    97412  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97414    97412  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97415    97412  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97416    97412  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97417    97412  0 17:34 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios
ec2-user   98423   98399  0 17:51 pts/2      00:00:00 grep --color=auto nagios
```

4. Become a root user and create 2 folders

```
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

```
[ec2-user@ip-172-31-43-65 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-43-65 ec2-user]# |
```

5. Copy the sample localhost.cfg file to linuxhost folder

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

```
[root@ip-172-31-81-4 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-81-4 ec2-user]# |
```

6. Open linuxserver.cfg using nano and make the following changes

nano

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

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
 Change address to the public IP address of your **LINUX CLIENT.**

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
#
# NOTE: This config file is intended to serve as an *extremely* simple
#       example of how you can create configuration entries to monitor
#       the local (Linux) machine.
#
#####

#####
# 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    localhost
    alias        localhost
    address      127.0.0.1
}

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

Change hostgroup_name under hostgroup to linux-servers1

Everywhere else on the file, change the hostname to linuxserver instead of localhost.

7. Open the Nagios Config file and add the following line

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

```
##Add this line
```

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

```
GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
#####
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#####
# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!
log_file=/usr/local/nagios/var/nagios.log

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

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

# Definitions for monitoring a Windows machine

^G Help      ^O Write Out   ^W Where Is     ^K Cut          ^T Execute      ^C Location    M-U Undo
^X Exit      ^R Read File   ^E Replace     ^P Paste         ^J Justify      ^Y Go To Line  M-E Redo
                                         M-A Set Mark  M-Z To Bracket
                                         M-C Copy      M-W Where Was
```

8. Verify the configuration files

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

```
Running pre-flight check on configuration data...

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

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

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-43-65 ec2-user]# |
```

You are good to go if there are no errors.

9. Restart the nagios service

```
service nagios restart
```

```
[root@ip-172-31-81-4 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-81-4 ec2-user]# sudo systemctl status nagi
```

Now it is time to switch to the client machine.

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

11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

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

```
ubuntu@ip-172-31-33-76:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [159 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 kB]
```

12. Open nrpe.cfg file to make changes.

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

Under allowed_hosts, add your nagios host IP address like so

```

ubuntu@ip-172-31-83-152:~$ nano /etc/nagios/nrpe.cfg
GNU nano 7.2                               /etc/nagios/nrpe.cfg *
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

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

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,3.86.12.126

# COMMAND ARGUMENT PROCESSING

```

Toolbar buttons: Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute Justify, Location Go To Line, Undo, Redo, Set Mark, Copy, To Bracket, Where Was.

13. Restart the NRPE server

```

sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-83-152:~$ sudo nano /etc/nagios/nrpe.cfg

ubuntu@ip-172-31-83-152:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-83-152:~$ 

```

14. Now, check your nagios dashboard and you'll see a new host being added.

Click on Hosts.

Nagios®

General

Current Status

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Logs

Problems

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

Reports

- Availability
- Trends
- Alerts
- History
- Summary
- Histogram
- Notifications
- Event Log

System

- Comments
- Downstream
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Nagios® Core™
Version 4.5.5
September 17, 2024
Check for updates

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Latest News

Don't Miss...

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS WITH NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios mark is governed by the trademark use restrictions.

Click on linuxserver to see the host details

Current Network Status

Last Updated: Sun Oct 6 17:55:03 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View Service Status Detail For All Host Groups](#)
[View Status Overview For All Host Groups](#)
[View Status Summary For All Host Groups](#)
[View Status Grid For All Host Groups](#)

Host Status Totals				Service Status Totals				
Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
2	0	0	0	6	1	0	1	0
All Problems All Types				All Problems All Types				
0 2				2 8				

Host Status Details For All Host Groups

Host		Status	Last Check	Duration	Status Information
linuxserver		UP	10-06-2024 17:50:12	0d 0h 24m 51s	PING OK - Packet loss = 0%, RTA = 0.77 ms
localhost		UP	10-06-2024 17:53:57	1d 0h 21m 53s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

You can click Services to see all services and ports being monitored.

Host Information

Last Updated: Sun Oct 6 17:43:35 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View Status Detail For This Host](#)
[View Alert History For This Host](#)
[View Alert Histogram For This Host](#)
[View Availability Report For This Host](#)
[View Notifications For This Host](#)

Host
localhost
(localhost)

Member of
linux-servers, lnx-servers1

Host State Information

Host Status: UP (for 1d 0h 10m 25s)	Status Information: PING OK - Packet loss = 0%, RTA = 0.03 ms
Performance Data: rta=0.030000ms;3000.000000;5000.000000;0.000000 pl=0%:80;100.0	Current Attempt: 1/10 (HARD state)
Last Check Time: 10-06-2024 17:38:57	Check Type: ACTIV/E
Check Latency / Duration: 0.000 / 4.140 seconds	Next Scheduled Active Check: 10-06-2024 17:43:57
Last State Change: 10-05-2024 17:33:10	Last Notification: N/A (notification 0)
Is This Host Flapping? NO (0.00% state change)	In Scheduled Downtime? NO
Last Update: 10-06-2024 17:43:34 (0d 0h 0m 1s ago)	

Active Checks: ENABLED
Passive Checks: ENABLED
Obsessing: ENABLED
Notifications: ENABLED
Event Handler: ENABLED
Flap Detection: ENABLED

Host Commands

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

Host Comments

Add a new comment Delete all comments

Entry Time Author Comment Comment ID Persistent Type Expires Actions

This host has no comments associated with it.

Current Network Status

Last Updated: Sun Oct 6 17:58:02 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View History For All hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

Host Status Totals				Service Status Totals				
Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
2	0	0	0	6	1	0	1	0
All Problems All Types				All Problems All Types				
0 2				2 8				

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-06-2024 17:56:27	1d 0h 24m 52s	1/4	OK - load average 0.00, 0.00, 0.00
localhost	Current Users	OK	10-06-2024 17:57:42	1d 0h 24m 14s	1/4	USERS OK - 6 users currently logged in
localhost	HTTP	WARNING	10-06-2024 17:53:57	0d 0h 19m 5s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
localhost	PING	OK	10-06-2024 17:55:12	1d 0h 22m 59s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
localhost	Root Partition	OK	10-06-2024 17:57:04	1d 0h 22m 22s	1/4	DISK OK - free space / 5567 MB (68.59% inode=98%):
localhost	SSH	CRITICAL	10-06-2024 17:53:19	1d 0h 21m 44s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	CRITICAL	10-06-2024 17:54:34	1d 0h 31m 7s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	10-06-2024 17:56:22	1d 0h 20m 29s	1/4	PROCS OK, 39 processes with STATE = RSDOT

Results 1 - 8 of 8 Matching Services

As you can see, we have our linuxserver up and running. It is showing critical status on HTTP due to permission errors and swap because there is no partition created.

In this case, we have monitored -

Servers: 1 linux server

Services: swap

Ports: 22, 80 (ssh, http)

Processes: User status, Current load, total processes, root partition, etc.

Recommended Cleanup

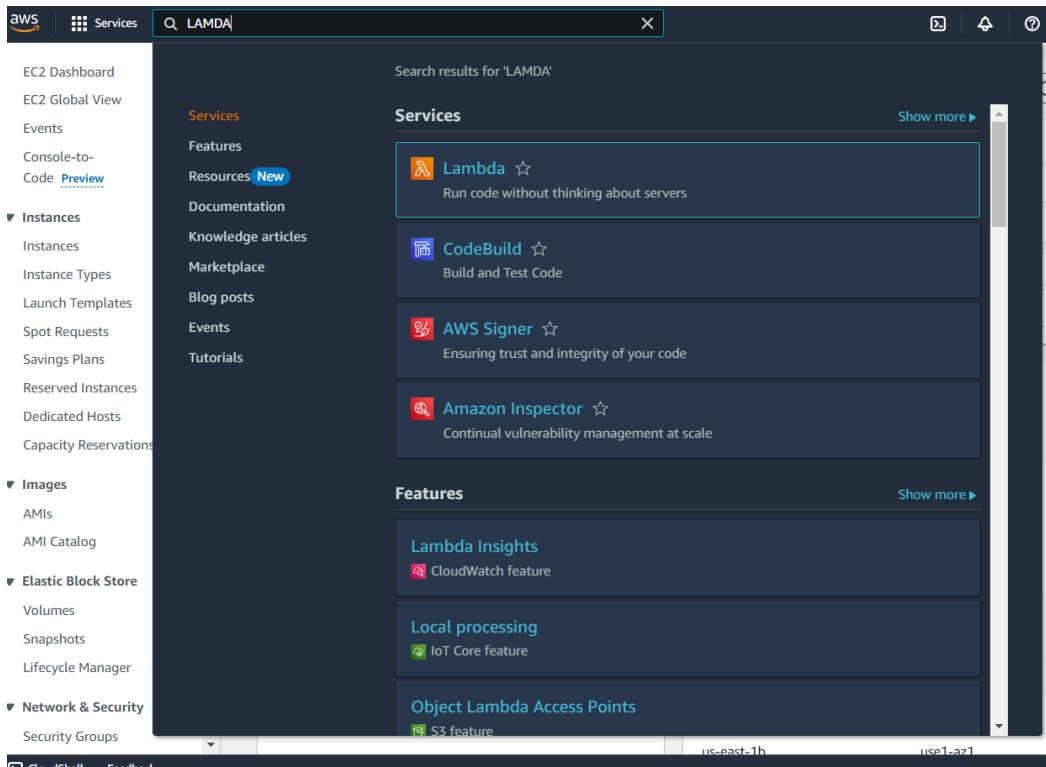
- Terminate both of your EC-2 instances to avoid charges.
- Delete the security group if you created a new one (it won't affect your bill, you may avoid it)

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

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

Step 1: Accessing AWS

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



Step 2: Creating a New Lambda Function

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

The screenshot shows the AWS Lambda landing page. At the top, there's a dark header with the word "Compute". Below it, the main title "AWS Lambda" is displayed in large, bold letters, followed by the subtitle "lets you run code without thinking about servers." A paragraph explains that users pay only for compute time consumed and can run code for various applications. To the right, a "Get started" box contains a link to "Create a function".

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

Create a function

How it works

Run Next: Lambda responds to events

.NET Java Node.js Python Ruby Custom runtime

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

Step 3: Configuring Basic Settings

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

The screenshot shows the "Basic information" configuration page for a Lambda function named "lamda_demo". The "Runtime" is set to "Python 3.12" and the "Architecture" is "x86_64". Under "Permissions", it says "By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers." A link "▶ Change default execution role" is visible at the bottom.

Basic information

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

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

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

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

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

▶ Change default execution role

Permissions [Info](#)

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

▼ Change default execution role

Execution role

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

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

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

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

▶ Additional Configurations

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

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

aws Services Search [Alt+S]

Successfully created the function lamda_demo. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamda_demo

lamda_demo

▼ Function overview [Info](#)

Throttle [Copy ARN](#) Actions ▾

Diagram Template

lambda_demo

+ Add trigger + Add destination

Description -

Last modified 26 seconds ago

Function ARN [arn:aws:lambda:eu-north-1:010928207735:function:lambda_demo](#)

Function URL [Info](#) -

Export to Application Composer Download ▾

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

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

[Learn more](#) [Start tutorial](#)

Code Test Monitor Configuration Aliases Versions

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

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

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

Step 4: Testing the Function

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

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

Test event action

Create new event Edit saved event

Event name

MyEventName

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

Event sharing settings

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

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

Template - optional

hello-world

Event JSON

```
1 [{"key1": "value1", "key2": "value2", "key3": "value3"}]
```

Format JSON

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_function.py

lambda_function

Configure test event Ctrl-Shift-C

Private saved events

● demo1

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

Code **Test** **Monitor** **Configuration** **Aliases** **Versions**

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_function.py

lambda_function

Execution result:

Status: Succeeded | Max memory used: 32 MB | Time: 1.35 ms

Execution results

Test Event Name: demo1

Response:

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

Function Logs

```
START RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Version: $LATEST
END RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa
REPORT RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Duration: 1.35 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa

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

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

Step 5: Running the Test

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

The screenshot shows the AWS Lambda function editor interface with the 'Execution result' tab selected. The top navigation bar is identical to the previous screenshot. The main area displays the execution results for the 'demo1' test event:

Status: Succeeded | Max memory used: 32 MB | Time: 1.62 ms

Test Event Name
demo1

Response

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

Function Logs

```
START RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Version: $LATEST
END RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec
REPORT RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Duration: 1.62 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Du
Request ID
298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec
```

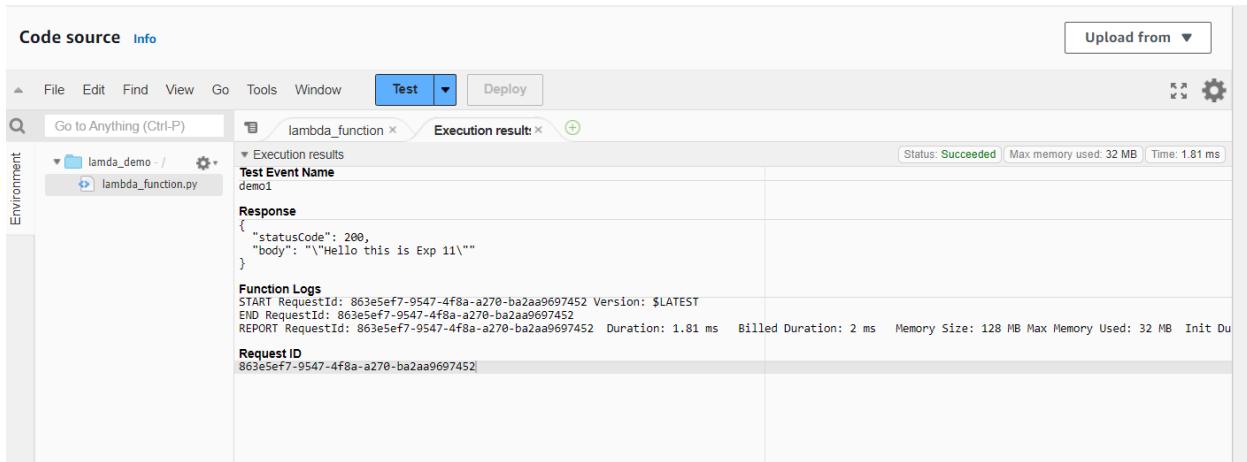
Step 6: Editing and Deploying the Code

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



Step 7: Final Testing

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



The screenshot shows the AWS Lambda console's "Test" tab for a function named "lambda_function". The "Execution result" section displays a successful test run with the following details:

- Status:** Succeeded
- Max memory used:** 32 MB
- Time:** 1.81 ms

The "Response" field shows the JSON output:

```
{"statusCode": 200, "body": "Hello this is Exp 11\\\""}  
The "Function Logs" section contains the following log entries:

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



The "Request ID" is listed as 863e5ef7-9547-4f8a-a270-ba2aa9697452.


```

Conclusion:

AWS Lambda simplifies the deployment of serverless applications by eliminating the need for server management, automatic scaling, and pay-as-you-go pricing. Creating Lambda functions with Python, Node.js, or Java is straightforward, making it accessible for developers to build efficient, scalable, and cost-effective solutions. Its ability to trigger functions based on events from various AWS services makes Lambda a powerful tool for automating cloud workflows.

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

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

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

3) Uncheck block all public access

Block Public Access settings for this bucket

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

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

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

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

⌚ Successfully created bucket "s3lamdaexp11"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

View details X

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](#) [Directory buckets](#)

General purpose buckets (2) Info All AWS Regions		Create bucket
Buckets are containers for data stored in S3.		
<input type="text" value="Find buckets by name"/>		< 1 > @
Name	AWS Region	IAM Access Analyzer
elasticbeanstalk-eu-north-1-01092807735	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1
s3lamdaexp11	US East (N. Virginia) us-east-1	View analyzer for us-east-1

5) 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 > s3lamdaexp11

s3lamdaexp11 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (0) [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](#)

Find objects by prefix

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

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

[Upload](#)

Amazon S3 > Buckets > s3lamdaexp11 > Upload

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, 990.9 KB)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input type="text"/> Find by name		< 1 >		
<input type="checkbox"/>	Name	▼	Folder	
<input type="checkbox"/>	football.jpg	-		

Destination [Info](#)

Destination
<s3://s3lamdaexp11>

▶ **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.

6) 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' with a link to 'View details below.' Below this, a message says 'The information below will no longer be available after you navigate away from this page.' The main area is titled 'Summary' and shows the destination 's3://s3lambdaexp11'. Under 'Succeeded', it lists '1 file, 990.9 KB (100.00%)'. Under 'Failed', it lists '0 files, 0 B (0%)'. There are tabs for 'Files and folders' and 'Configuration', with 'Files and folders' selected. A table below shows the uploaded file: Name (football.jpg), Folder (-), Type (image/jpeg), Size (990.9 KB), Status (Succeeded), and Error (-). A search bar at the top of the table says 'Find by name'.

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 'Create function' configuration page. The top navigation bar shows 'Lambda > Functions > Create function'. The main section is titled 'Create function' with an 'Info' link. It asks to choose one of three options: 'Author from scratch' (selected), 'Use a blueprint', or 'Container image'. Below this is a 'Basic information' section. In 'Function name', the value 'lamdaexp12' is entered. A note says the name must be unique to the Region and can't include spaces. In 'Runtime', 'Node.js 20.x' is selected. In 'Architecture', 'x86_64' is selected. In 'Permissions', there is a link to 'Change default execution role'. Other sections like 'Environment variables', 'Code', and 'Role' are partially visible at the bottom.

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

Lambda > Functions > lamdaexp12

lamdaexp12

Description

Last modified
22 seconds ago

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

Function URL [Info](#)

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

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

S3

aws asynchronous storage

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

S3

aws asynchronous storage

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

Bucket region: us-east-1

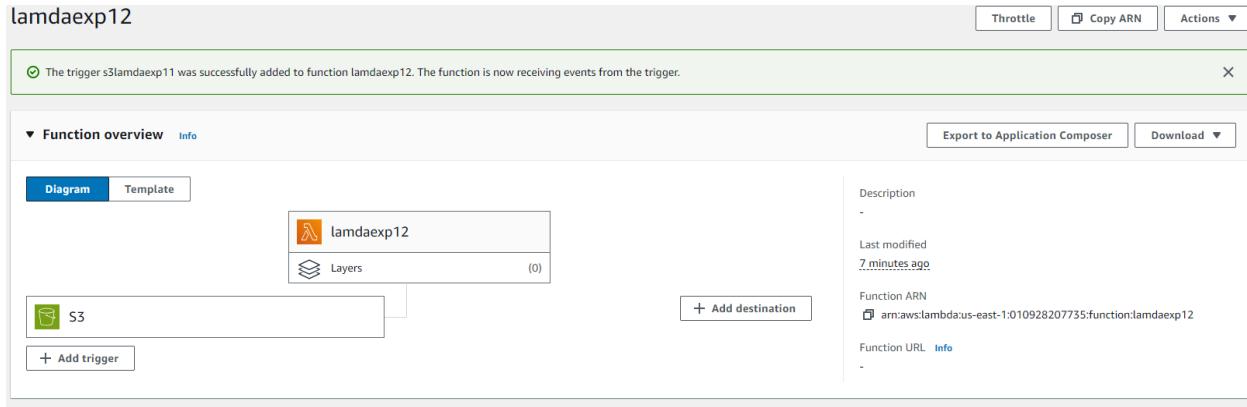
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events

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

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

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



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

```
export const handler = async (event) => {
if (!event.Records || event.Records.length === 0) {
console.error("No records found in the event.");
return {
statusCode: 400,
body: JSON.stringify('No records found in the event')
};
}
// Extract bucket name and object key from the event
const record = event.Records[0];
const bucketName = record.s3.bucket.name;
const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
console.log(`Event Source: ${record.eventSource}`);
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 JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.

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

```
1  exports.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 objectKey = decodeURIComponent(record.s3.object.key.replace(/\//g, ' ')); // Handle encoded keys
12   const bucketName = record.s3.bucket.name;
13   console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
14   console.log(`Event Source: ${record.eventSource}`);
15   console.log(`Event Source: ${record.eventSource}`);
16   console.log(`Event Source: ${record.eventSource}`);
17   console.log(`Event Source: ${record.eventSource}`);
18   return {
19     statusCode: 200,
20     body: JSON.stringify('Log entry created successfully!')
21   };
22 }
```

The status bar at the bottom right indicates '22:3 JavaScript Spaces: 2'.

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

Private

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

Shareable

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

Template - optional

hello-world

Format JSON

```
1 [ ]  
2 "Records": [  
3 {  
4     "eventVersion": "2.0",  
5     "eventSource": "aws:s3",  
6     "awsRegion": "us-east-1",  
7     "eventTime": "1970-01-01T00:00:00.000Z",  
8     "eventName": "ObjectCreated:Put",  
9     "userIdentity": {  
10         "principalId": "EXAMPLE"  
11     },  
12     "requestParameters": {  
13         "sourceIPAddress": "127.0.0.1"  
14     },  
15     "responseElements": {  
16         "x-amz-request-id": "EXAMPLE123456789",  
17         "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmabcdefghijklmnijkmnopqrstuvwxyzABCDEFGHIJKLMNOQRSTUVWXYZ"  
18     },  
19     "s3": {  
20         "s3SchemaVersion": "1.0",  
21         "configurationId": "testConfigRule",  
22         "bucket": {  
23             "name": "example-bucket",  
24             "ownerIdentity": {  
25                 "principalId": "EXAMPLE"  
26             },  
27             "arn": "arn:aws:s3:::example-bucket"  
28         },  
29         "object": {  
30             "key": "test%2Ekey"  
31         }  
32     }  
33 }
```

1:1 JSON Spaces: 2

Cancel **Invoke** **Save**

⌚ The test event **myevent1** was successfully saved.

Step 3: Check the logs

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

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

Services:

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

Features:

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

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

The screenshot shows the CloudWatch Log Groups page. It displays a single log group entry: '/aws/lambda/lambdaexp12'. The page includes filters for Log group, Log class, Anomaly detection, Data protection, Sensitive data protection, Retention, Metric filters, and Contributor Insights.

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

CloudWatch > Log groups > /aws/lambda/lamdaexp12

/aws/lambda/lamdaexp12

Log group details

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

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

Log streams (1)

Log stream	Last event time
2024/10/07/[\$.LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00	2024-10-07 04:34:00 (UTC)

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

CloudWatch > Log groups > /aws/lambda/lamdaexp12 > 2024/10/07/[\$.LATEST]29f700136a7b40a2b79269533eae8969

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
2024-10-07T04:45:01.475Z	INIT_START Runtime Version: nodejs:20.v39 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:ad9b2ae231dfc4c3325e183024ccb4d9de1aa14796d90295f898140041242f7
2024-10-07T04:45:01.612Z	START RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409ddd7c Version: \$LATEST
2024-10-07T04:45:01.632Z	END RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409ddd7c
2024-10-07T04:45:01.632Z	REPORT RequestId: 98ba5a56-c45d-4b89-a6dd-3e55409ddd7c Duration: 18.74 ms Billed Duration: 19 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 135.57 ms
2024-10-07T04:46:20.191Z	START RequestId: 3cc0ddc18-4346-4ea5-8c87-eabb930f86a Version: \$LATEST
2024-10-07T04:46:20.194Z	END RequestId: 3cc0ddc18-4346-4ea5-8c87-eabb930f86a
2024-09-30T09:24:40.324Z	2024-09-30T09:24:40.324Z 01723939-7200-421b-aa50-432105575475 INFO An image has been added to the bucket example-bucket: test/key
2024-10-07T04:46:54.248Z	START RequestId: a7de0c16-28c0-4c19-9aa0-5a616357447b Version: \$LATEST
2024-10-07T04:46:54.250Z	END RequestId: a7de0c16-28c0-4c19-9aa0-5a616357447b
2024-10-07T04:46:54.250Z	REPORT RequestId: a7de0c16-28c0-4c19-9aa0-5a616357447b Duration: 1.48 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB
	REPORT RequestId: a7de0c16-28c0-4c19-9aa0-5a616357447b Duration: 1.48 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 62 MB

No newer events at this moment. Auto retry paused. [Resume](#)

Conclusion: In this experiment, we successfully created a Lambda function that logs a message when an image is added to a specific S3 bucket. By configuring an S3 bucket trigger for the Lambda function, we demonstrated how AWS services can work together to automate tasks. The function logged important details about the event, including the bucket name and object key. After deploying the function and testing with a sample event, we verified the logs in CloudWatch, confirming that the Lambda function correctly detected and logged the addition of an image to the bucket. This experiment highlighted the seamless integration between AWS Lambda and S3 for event-driven processes.