

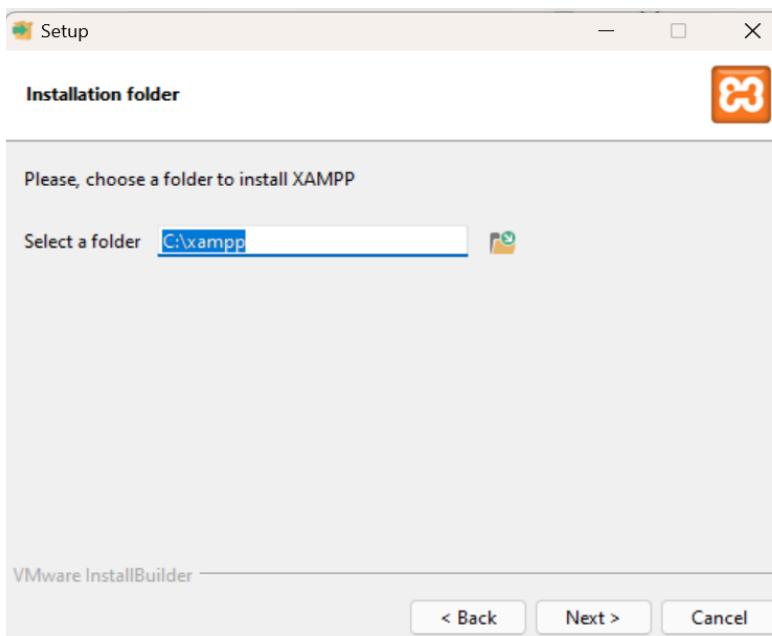
Experiment 1A

Hosting a static site using xampp

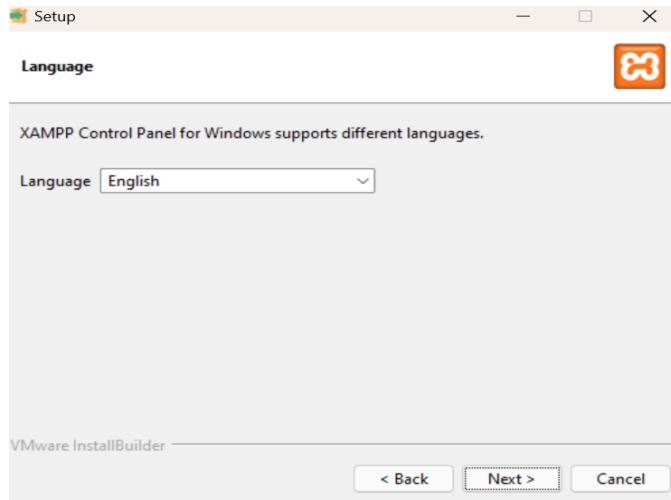
1. Download and open the xampp application and click on zip and extract it then click it.



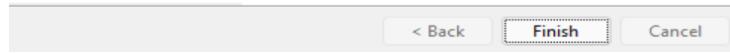
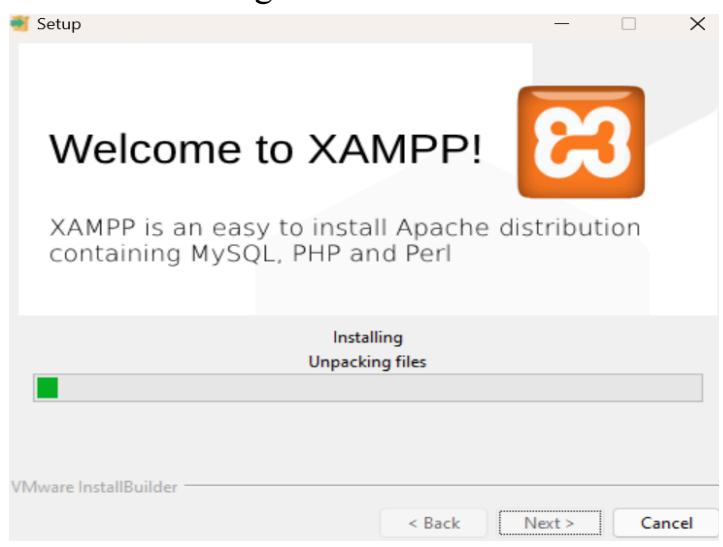
2. Then select the c directory.



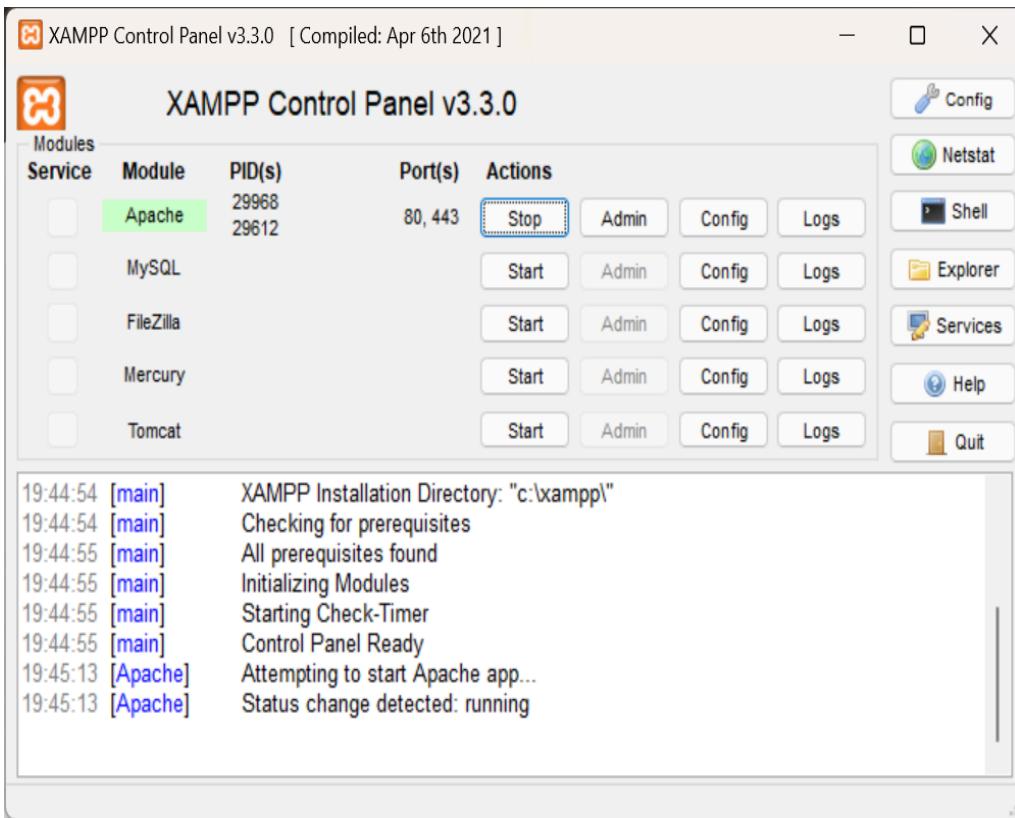
3. Then select language as english.



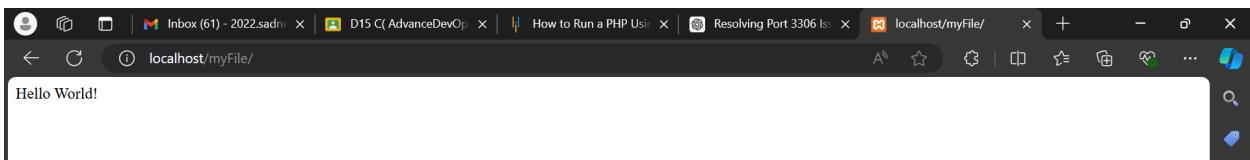
Then install and get started.



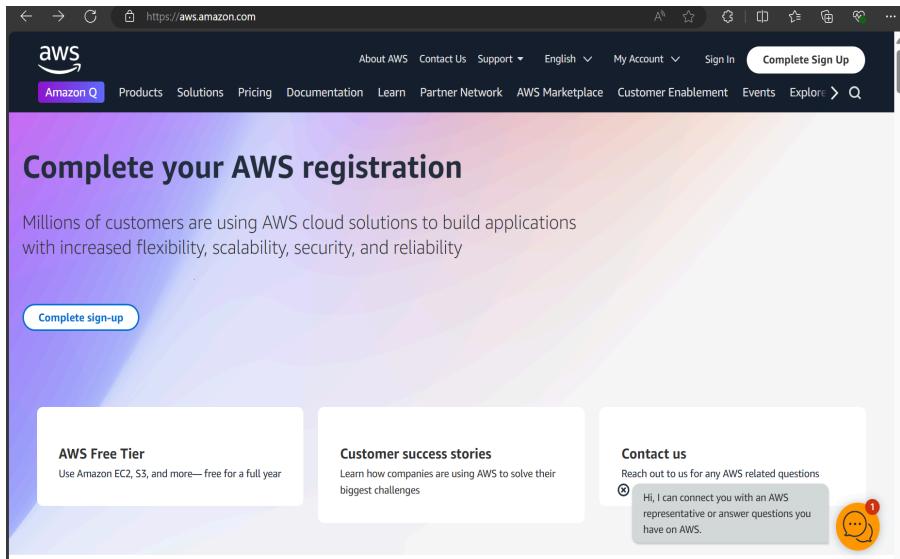
Then the final control panel opens.



Then I created a file myFile inside htdocs folder and uploaded a sample file.and then opened <https://localhost/myFile/in.php>



Hosting a static site using S3 bucket:

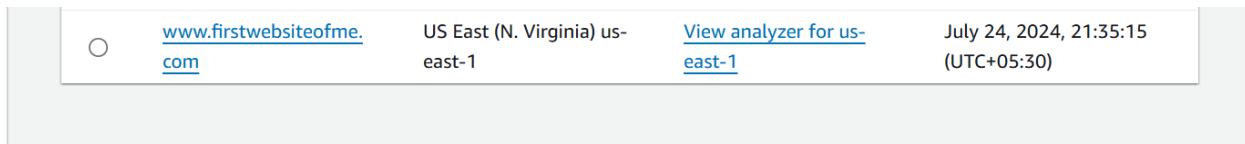


1. I clicked on create bucket.

This screenshot shows the 'Get Started' page for the Amazon S3 service in the EU-North-1 region. The main title is 'Amazon S3' with the subtitle 'Store and retrieve any amount of data from anywhere'. A brief description states: 'Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.' To the right, there's a 'Create a bucket' button and a 'Pricing' section. Below the main title, there's a 'How it works' section with a video thumbnail titled 'Introduction to Amazon S3' and a 'Copy link' button. The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

This screenshot shows the 'Get Started' page for the Amazon S3 service in the US-East-1 region. The layout is identical to the EU version, featuring the 'Amazon S3' title, 'Store and retrieve any amount of data from anywhere' subtitle, and the 'Create a bucket' button. It also includes the 'How it works' section with the 'Introduction to Amazon S3' video and the 'Pricing' section. The left sidebar contains a navigation menu with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, and Feature spotlight. The bottom of the page includes the same footer links as the EU version.

3. Then I created a bucket named www.firstwebsiteofme.com



I clicked on the bucket.

The screenshot shows the AWS S3 console for the 'www.firstwebsiteofme.com' bucket. The 'Objects' tab is selected, showing a single object named 'index.html' which is an HTML file (Type: html, Size: 62.0 B, Storage class: Standard). The 'Upload' button is highlighted in yellow. The left sidebar shows various AWS services like IAM Access Analyzer, Block Public Access settings, and Storage Lens.

4. Then I also S3 webhosting-bucket policy with this code .I entered in the permission section there is an option edit bucket policy. Where I uploaded the code and saved it.

The screenshot shows the 'Edit bucket policy' page. The policy is defined in JSON:

```
1 Version: "2012-10-17",
2 Statement: [
3   {
4     Sid: "PublicReadGetObject",
5     Effect: "Allow",
6     Principal: "*",
7     AWS: "*",
8   },
9   {
10    Action: "s3:GetObject",
11    Resource: "arn:aws:s3:::www.firstwebsiteofme/*"
12  }
13 ]
14 }
```

The right side of the screen has a 'Select a statement' dropdown and a '+ Add new statement' button.

4. Then i clicked on index.html file and opened its properties section.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various services like Buckets, Access Grants, and Storage Lens. The main area shows the object details for 'index.html' in the 'www.firstwebsiteofme.com' bucket. The 'Properties' tab is selected, displaying information such as Owner (2022.sadneya.samant), AWS Region (US East (N. Virginia) us-east-1), Last modified (July 24, 2024, 21:35:41 (UTC+05:30)), Size (62.0 B), Type (html), and Key (index.html). It also shows the S3 URI (s3://www.firstwebsiteofme.com/index.html), Amazon Resource Name (ARN) (arn:aws:s3:::www.firstwebsiteofme.com/index.html), Entity tag (Etag) (18f77726d0973067f7329656daaccd8), and Object URL (<https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html>). At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

4. Then I clicked on

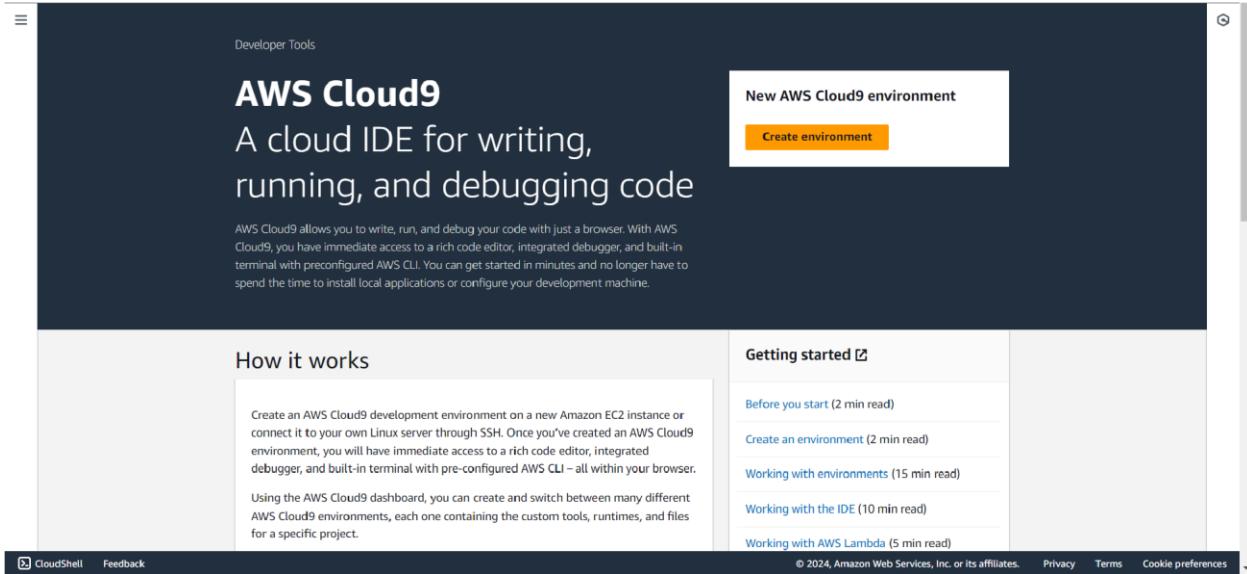
<https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html> which given me the final output.

The screenshot shows a web browser window with the URL <https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html>. The page content is "Hello World!" and below it, a small note says "This is just a sample page". The browser interface includes a back button, forward button, search bar, and a vertical toolbar on the right.

Name: Muskan Chandiramani
Div: D15C
Roll No.5

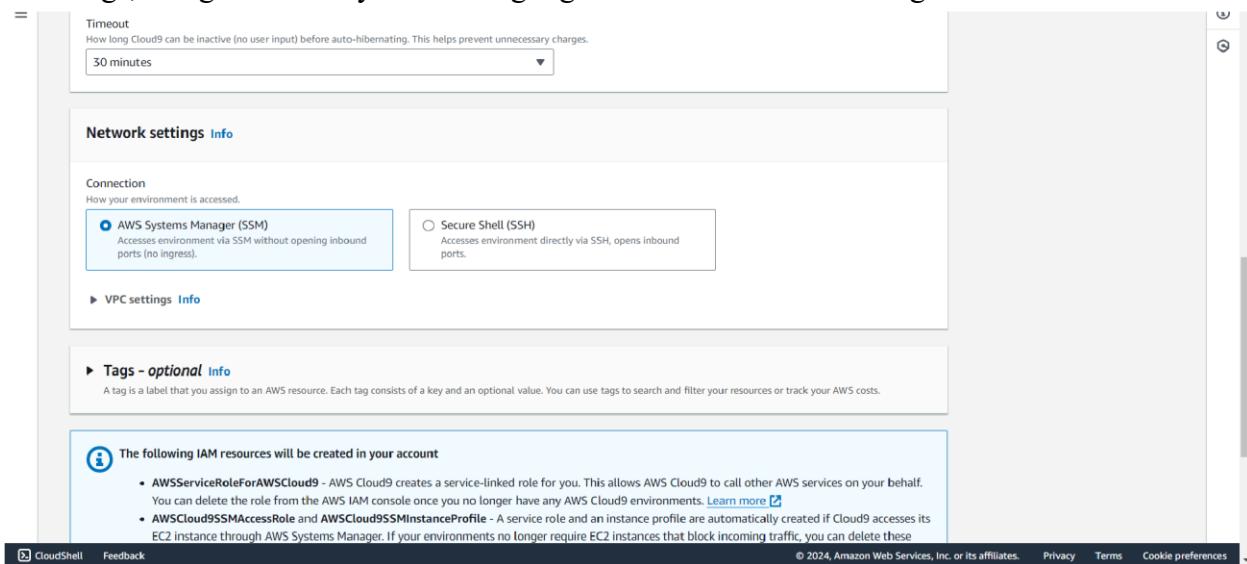
Experiment 1B: IAM and cloud9

1. Open the AWS account and search for Cloud9.



The screenshot shows the AWS Cloud9 landing page. At the top right, there is a prominent orange button labeled "Create environment". Below this, there is descriptive text about what Cloud9 is and how it works. On the left, there is a sidebar titled "How it works" with a detailed description of the Cloud9 development process. On the right, there is a sidebar titled "Getting started" with links to various documentation pages. At the bottom, there are links for "CloudShell", "Feedback", "Privacy", "Terms", and "Cookie preferences".

2. 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



The screenshot shows the "Create environment" configuration page. It includes sections for "Timeout" (set to 30 minutes), "Network settings" (with "AWS Systems Manager (SSM)" selected), "VPC settings" (indicated by a "▶" icon), "Tags - optional" (describing tags for AWS resources), and a section for "The following IAM resources will be created in your account" (listing "AWSServiceRoleForAWSCloud9", "AWSCloud9SSMAccessRole", and "AWSCloud9SSMInstanceProfile"). At the bottom, there are links for "CloudShell", "Feedback", "Privacy", "Terms", and "Cookie preferences".

The screenshot shows the 'VPC settings' section of the AWS Cloud9 configuration interface. It includes a 'Tags - optional' info section, a note about IAM resource creation, and an error message indicating issues with IAM roles and permissions.

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

- AWS*ServiceRoleForAWS*Cloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- AWS*Cloud9SSMAccessRole* and AWS*Cloud9SSMInstanceProfile* - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

Create [Cancel](#)

There was an error creating the IAM resources needed for SSM connection.

You don't have the permission required to perform this operation. Ask your administrator to give you permissions.

User: arn:aws:sts::354256622778:assumed-role/voclabs/user3404112=SHARMA__RAKSHIT_KUMAR is not authorized to perform: iam:CreateRole on resource: arn:aws:iam::354256622778:role/service-role/AWS*Cloud9SSMAccessRole* because no identity-based policy allows the iam:CreateRole action

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

3. Use the Secure Shell option in Network settings.

The screenshot shows the 'Network settings' section of the AWS Cloud9 configuration interface. It includes a 'Connection' section where 'Secure Shell (SSH)' is selected, and a note about IAM resource creation.

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

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

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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

The screenshot shows the AWS Cloud9 interface. On the left, there's a sidebar with links for 'My environments', 'Shared with me', and 'All account environments'. Below that is a 'Documentation' link. The main content area has a header indicating 'Creating MyEnvironment. This can take several minutes. While you wait, see Best practices for using AWS Cloud9'. It also contains a note about AWS Toolkits. The central part is titled 'Environments (1)' and shows a table with one row. The table columns are 'Name', 'Cloud9 IDE', 'Environment type', 'Connection', 'Permission', and 'Owner ARN'. The single row shows 'MyEnvironment' as the name, 'Open' as the Cloud9 IDE, 'EC2 instance' as the environment type, 'Secure Shell (SSH)' as the connection method, 'Owner' as the permission level, and the ARN 'arn:aws:sts::354256622778:assumed-role/voclabs/user3404112=SHARMA_RAKSHIT_KUMAR' as the owner ARN.

5. Cloud9 Environment.is opened when u click on the environment name

The screenshot shows the AWS Cloud9 IDE interface. The title bar says 'Welcome'. The left sidebar shows a file tree with 'MyEnvironment' expanded, containing 'c9' and 'README.md'. The main area has a large title 'AWS Cloud9' and sub-headline 'Welcome to your development environment'. Below this is a 'Toolkit for AWS Cloud9' section with a description of its features. To the right is a 'Getting started' sidebar with options like 'Create File', 'Upload Files...', and 'Clone from GitHub'. At the bottom is a terminal window showing a bash session with the command 'voclabs:~/environment \$'. The status bar at the bottom left says 'AWS profile default'.

IAM user creation steps

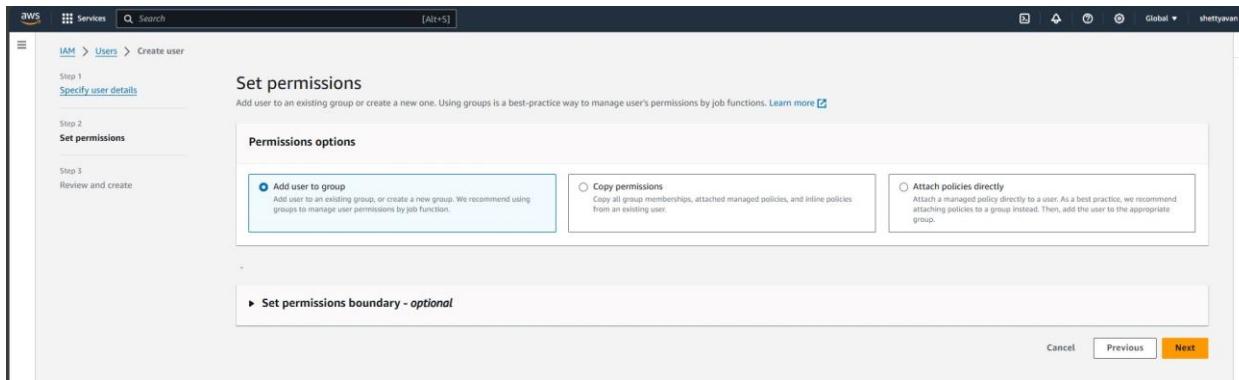
1. Open the aws account and search for IAM in service.

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with a search bar and links for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access Analyzer, External access, Unused access). The main area is titled 'IAM Dashboard' and contains 'Security recommendations' with a red notification dot (1). It lists two items: 'Add MFA for root user' (with a 'Add MFA' button) and 'Root user has no active access keys' (with a note about using access keys instead). Below this is a section titled 'IAM resources' with a 'Resources in this AWS Account' heading. A table shows the count of resources: User groups (1), Users (1), Roles (6), Policies (1), and Identity providers (0).

2. Select the users option from the left panel and click on create user button.Give the user name,

The screenshot shows the 'Specify user details' step of the 'Create user' wizard. The top navigation bar shows 'IAM > Users > Create user'. The left sidebar shows 'Step 1: Specify user details', 'Step 2: Set permissions', and 'Step 3: Review and create'. The main form is titled 'User details' and contains a 'User name' field with 'sample' entered. Below it is a note: '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 for 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a best practice [] to manage their access in IAM Identity Center'. At the bottom right are 'Cancel' and 'Next' buttons.

3. Click the add user option if you don't have an existing user group



4. Give a name to your user group and check the policies if required any

Name the group

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

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

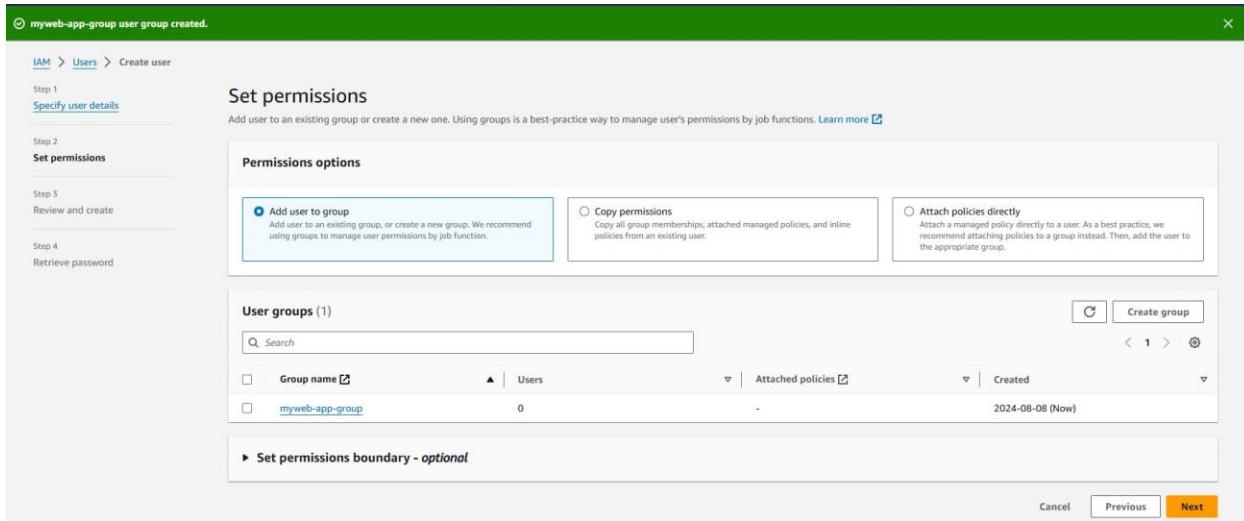
Add users to the group - Optional (1/1) info
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name	Groups	Last activity
sample	0	None

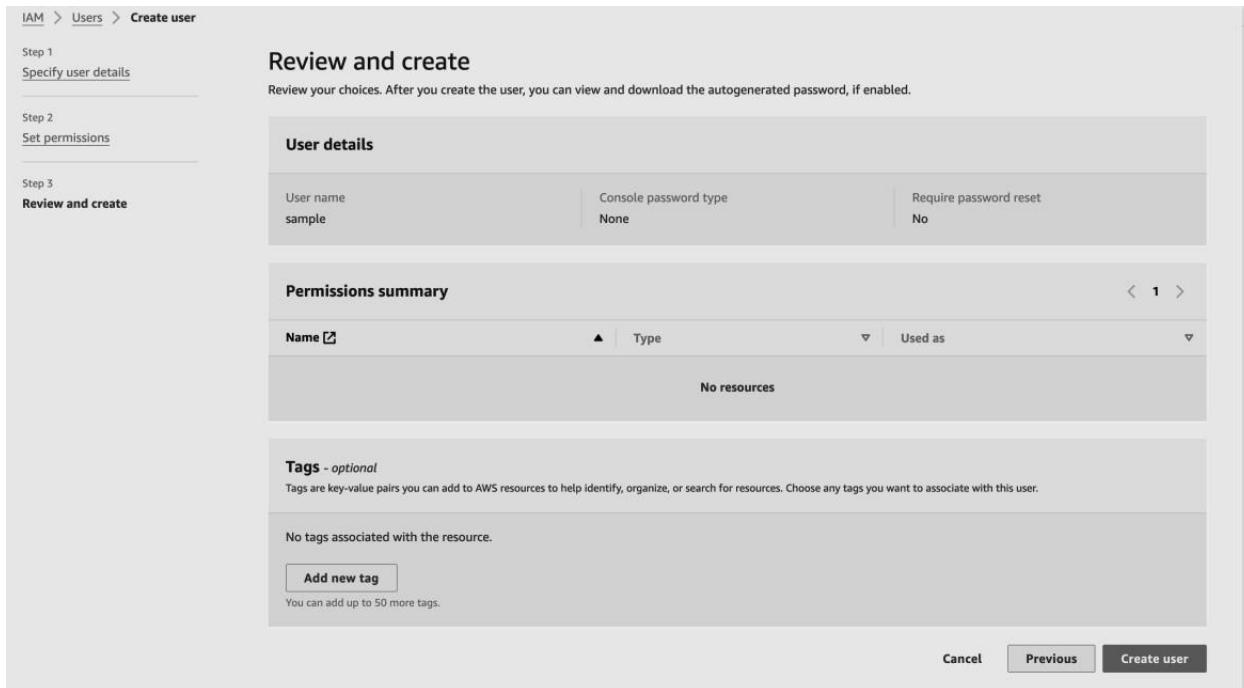
Attach permissions policies - Optional (945) Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Policy name	Type	Used as	Description
AdministratorAccess	AWS managed - job function	None	Provides full access to all AWS services.
AdministratorAccess-Amplify	AWS managed	None	Grants account-wide access to Amplify services.
AdministratorAccess-AWSElasticBeans...	AWS managed	None	Grants account-wide access to Elastic Beanstalk services.
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup for Alexa for Business.

5. Once the user group is created select the name and click next to create your user



6. Review the configuration details and check if you have missed any steps and then click on ‘Create user’ button



7. You will see the “user created successfully” message and incase you need then store your password by downloading the csv file

The screenshot shows the AWS IAM 'Users' page. A success message at the top states 'User created successfully'. Below it, a note says 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' A 'View user' button is available. The main table lists one user: 'sample' under 'User name'. The table includes columns for User name, Path, Groups, Last activity, MFA, Password age, Console last sign-in, Access key ID, and Active status.

8. 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 IAM 'User groups' page. It displays the 'sample_group' details, including its ARN: arn:aws:iam::434768569951:group/sample_group. The 'Users' tab is selected, showing one user named 'sample' added to the group. The 'Permissions' and 'Access Advisor' tabs are also visible.

9. Search for the “AWSCloud9EnvironmentMember” policy and attach it.

The screenshot shows the 'Attach permission policies to myweb-app-group' page. Under 'Other permission policies (1/945)', the 'Awscloud9E' policy is selected and highlighted. The 'Attach policies' button is visible at the bottom right.

Name: Muskan Chandiramani
Div: D10C
Roll No:5

Practical No 2 : Elastic Beanstalk

- 1) Go to services and choose elastic Beanstalk. following page will appear.

The screenshot shows the Amazon Elastic Beanstalk landing page. At the top, there's a dark header with the word 'Compute' and the Elastic Beanstalk logo. Below the header, the main title 'Amazon Elastic Beanstalk' is displayed in large bold letters, followed by the subtitle 'End-to-end web application management.' A brief description explains that Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. To the right, there's a 'Get started' section with a 'Create application' button. On the left, there's a 'Get started' section with a box containing text about uploading code. On the right, there's a 'Pricing' section stating that there's no additional charge for Elastic Beanstalk. At the bottom, there are links for CloudShell, Feedback, and various legal notices.

- 2) Configure the environment. Give the application name, check domain availability and choose PHP as platform. Then click next.

The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk setup wizard. It consists of three tabs: 'Environment tier', 'Application information', and 'Environment information'.
Environment tier: Set to 'Web server environment'. It explains that this tier runs a website, web application, or web API that serves HTTP requests.
Application information: Shows an 'Application name' field containing 'samplel'. It also includes an optional 'Application tags' section.
Environment information: Allows setting the name, subdomain, and description for the environment, noting that these cannot be changed later.

- 3) Configure the service access. 4) Choose one of the available VPC and instance subnet. Click next.

Set up networking, database, and tags - *optional* Info

Virtual Private Cloud (VPC)

VPC
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-0a482134962ed0c59 | (172.31.0.0/16)

Create custom VPC

Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
 Activated

Instance subnets			
<input type="text"/> Filter instance subnets			
Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1d	subnet-04a4cfde8...	172.31.0.0/20	

- 5) Configure instance traffic and scaling. Keep all the options as default.

Configure service access Info

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role
 Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.
 EMR_EC2_DefaultRole

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

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
 EMR_EC2_DefaultRole
 View permission details

Cancel **Skip to review** **Previous** **Next**

Configure instance traffic and scaling - optional Info

▼ Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

Size

The number of gigabytes of the root volume attached to each instance.

 GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

 MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

Availability Zones

Number of Availability Zones (AZs) to use.

Placement

Specify Availability Zones (AZs) to use.

Scaling cooldown

 seconds

Cancel

Skip to review

Previous

Next

- 6) Configure updates, monitoring, and logging. Keep everything as default and click next.

Configure updates, monitoring, and logging - optional Info

▼ Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

System

- Basic
 Enhanced

CloudWatch Custom Metrics - Instance

[Choose metrics](#) ▾

CloudWatch Custom Metrics - Environment

[Choose metrics](#) ▾

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

- Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. [Learn more](#)

Log streaming

(standard CloudWatch charges apply.)

- Activated

Retention

7

Lifecycle

[Keep logs after terminating envir...](#) ▾

Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

No environment properties have been configured.

[Add environment property](#)

[Cancel](#)

[Previous](#)

[Next](#)

7) In the review section, click submit.

false	false							
Platform software								
Lifecycle	Log streaming	Allow URL fopen						
false	Deactivated	On						
Display errors	Document root	Max execution time						
Off	-	60						
Memory limit	Zlib output compression	Proxy server						
256M	Off	nginx						
Logs retention	Rotate logs	Update level						
7	Deactivated	minor						
X-Ray enabled								
Deactivated								
Environment properties								
<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td colspan="2">No environment properties</td> </tr> <tr> <td colspan="2">There are no environment properties defined</td> </tr> </tbody> </table>			Key	Value	No environment properties		There are no environment properties defined	
Key	Value							
No environment properties								
There are no environment properties defined								
<input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input type="button" value="Submit"/>								

8) Environment has been created successfully.

Environment successfully launched.

Elastic Beanstalk > Environments > Sampel-env

Sampel-env Info

Environment overview

Health ⚠ Warning	Environment ID e-u7kfdezi3r
Domain kshitij.us-east-1.elasticbeanstalk.com	Application name sampie

Events | **Health** | **Logs** | **Monitoring** | **Alarms** | **Managed updates** | **Tags**

Events (10) Info

Time	Type	Details
August 9, 2024 21:25:13 (UTC+5:30)	⚠ WARN	Environment health has transitioned from Pending to Warning. Initialization completed 27 seconds ago and took 2 minutes. There are no instances. Unable to assume role "arn:aws:iam::996474913977:role/EMR_EC2_DefaultRole". Verify that the role exists and is configured correctly.

9) Deploy something on the recently created environment.

Upload and deploy

X

 To deploy a previous version, go to the [Application versions page](#)

Upload application

 Choose file

 File name: **Screenshot 2023-11-10 185456.png**

File must be less than 500MB max file size

Version label

Unique name for this version of your application code.

sampel-version-1

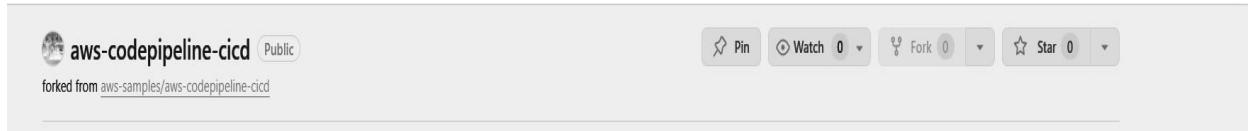
Current number of EC2 instances: 1

Cancel

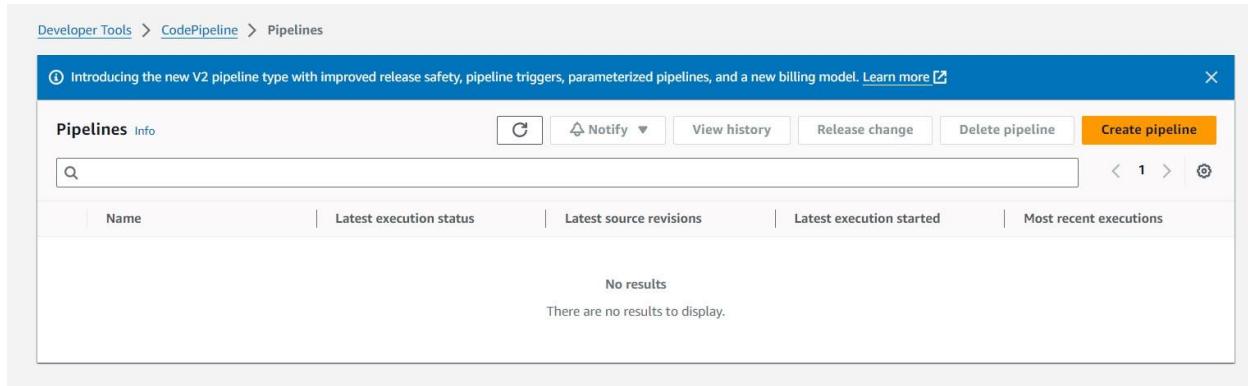
Deploy

Pipeline Creation:

- 1) Fork a github repository. This forked repository will act as source for your code pipeline.



- 2) Go to developer tools and select CodePipeline and create a new pipeline



- 3) Create a pipeline:

Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020

Choose pipeline settings Info

Step 1 of 5

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

firstpipeline

No more than 100 characters

Pipeline type

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

Execution mode

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

Superseded

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

Queued (Pipeline type V2 required)

Executions are processed one by one in the order that they are queued.

Parallel (Pipeline type V2 required)

4)

Add source stage Info

Step 2 of 5

Source

Source provider

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

GitHub (Version 1) ▾

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connect to GitHub



The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

GitHub webhooks (recommended)

Use webhooks in GitHub to automatically start my pipeline when a change occurs

AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

Add source stage Info

Step 2 of 5

Source

Source provider

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

GitHub (Version 1)



Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

You have successfully configured the action with the provider.



The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository

pixelbypixels/aws-codepipeline-cicd



Branch

main



Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

GitHub webhooks (recommended)

Use webhooks in GitHub to automatically start my pipeline when a change occurs

AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

5) Go to the deploy stage and ensure the following settings.

Add deploy stage Info

Step 4 of 5



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.

Deploy

Deploy provider

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

AWS Elastic Beanstalk



Region

US East (N. Virginia)



Input artifacts

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



No more than 100 characters

Application name

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

test_application



Environment name

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

test-application-env



Configure automatic rollback on stage failure

6) review the pipeline settings.

Review [Info](#)

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings	
Pipeline name	test_pipeline
Pipeline type	V2
Execution mode	QUEUED
Artifact location	A new Amazon S3 bucket will be created as the default artifact store for your pipeline
Service role name	AWSCodePipelineServiceRole-us-east-1-test_pipeline

7) Then go ahead and check the URL provided in the EBS environment.

Success
Congratulations! The pipeline firstpipeline has been created.

[Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > firstpipeline

firstpipeline

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: [cf4dc21b-a739-4463-a00a-e76d7579dcf3](#)

Source
[GitHub \(Version 2\)](#)
Succeeded • 2 minutes ago
[Bfd5d54](#)
[View details](#)

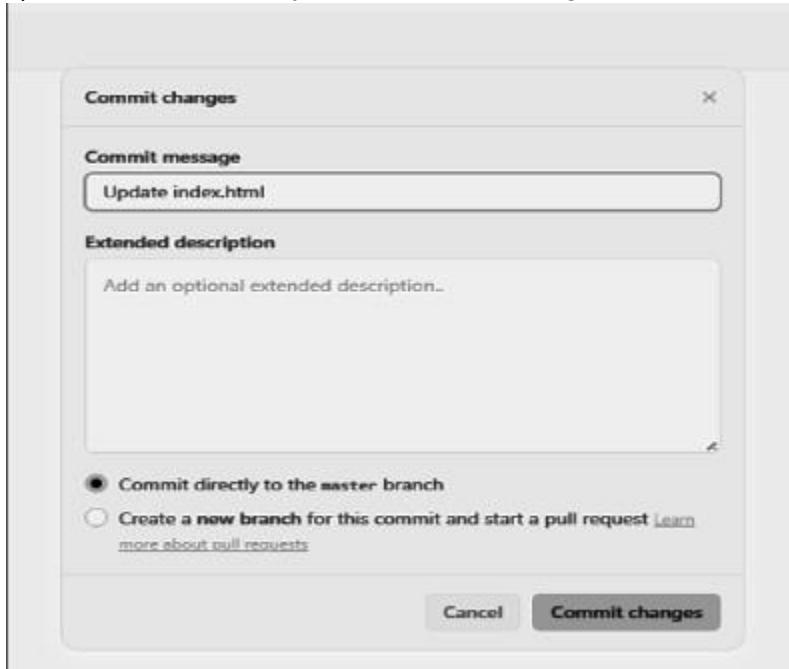
[fffd5d54](#) Source: Update README.md

[Disable transition](#)

Deploy Succeeded
Pipeline execution ID: [cf4dc21b-a739-4463-a00a-e76d7579dcf3](#)

Deploy

- 8) Go to the repository and make the changes in the index.html file and commit them



- 9) The changes that are committed can be noticed in the source panel in real time and to view the changes check the url (refresh it) and you can view the changes once the deployment section shows success.



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

Steps:

1. We will create 3 EC2 instances. One will be the master node and the other 2 will be slave/worker nodes.

Instances (3) Info		Last updated 43 minutes ago	C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
					All states ▾			
<input type="checkbox"/>	Name ✎ ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status		
<input type="checkbox"/>	worker2	i-0f554e25913aa17a0	🕒 Running ⊕ 🕒	t2.micro	🕒 2/2 checks passed	View alarms +		
<input type="checkbox"/>	master	i-09878736747637d9a	🕒 Running ⊕ 🕒	t2.micro	🕒 2/2 checks passed	View alarms +		
<input type="checkbox"/>	worker1	i-063256e0e8d824e95	🕒 Running ⊕ 🕒	t2.micro	🕒 2/2 checks passed	View alarms +		

2. After the instances have been created, we will connect them one by one.

Instances (1/3) Info Last updated less than a minute ago **Connect** **Instance state** Actions **Launch instances**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
worker2	i-0f554e25913aa17a0	Running	t2.micro	2/2 checks passed	View alarms +
master	i-09878736747637d9a	Running	t2.micro	2/2 checks passed	View alarms +
worker1	i-063256e0e8d824e95	Running	t2.micro	2/2 checks passed	View alarms +

security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.239.158.0/29. [Learn more.](#)

Instance ID: i-09878736747637d9a (master)

Connection Type:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address: 3.106.222.144

Username:
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel **Connect**

3. Docker installation:

This step has to be performed on all the 3 instances. The following command has to be run:

```
yum install docker -y
```

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-12-97 ~]$ sudo su
[root@ip-172-31-12-97 ec2-user]# yum install docker -y
Last metadata expiration check: 0:08:33 ago on Sat Sep 14 15:21:32 2024.
Dependencies resolved.

=====
Package           Architecture   Version      Repository  Size
=====
Installing:
  docker          x86_64        25.0.6-1.amzn2023.0.2    amazonlinux 44 M
Installing dependencies:
  containerd      x86_64        1.7.20-1.amzn2023.0.1    amazonlinux 35 M
  iptables-libc   x86_64        1.8.8-3.amzn2023.0.2    amazonlinux 401 k
  iptables-nft    x86_64        1.8.8-3.amzn2023.0.2    amazonlinux 183 k
  libcgroup       x86_64        3.0-1.amzn2023.0.1     amazonlinux 75 k
  libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2    amazonlinux 58 k

=====
aws Services Search [Alt+S]
=====
libn k x86_64 1.0.8-2.amzn2023.0.2    amazonlinux 58 k
libnfnetwork x86_64 1.0.1-19.amzn2023.0.2    amazonlinux 30 k
libnftnl x86_64 1.2.2-2.amzn2023.0.2    amazonlinux 84 k
pigz x86_64 2.5-1.amzn2023.0.3    amazonlinux 83 k
runc x86_64 1.1.13-1.amzn2023.0.1   amazonlinux 3.2 M

=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm          3.0 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm          6.6 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm          1.7 MB/s | 75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm  1.6 MB/s | 58 kB  00:00
(5/10): libnfnetwork-1.0.1-19.amzn2023.0.2.x86_64.rpm        823 kB/s | 30 kB  00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm          2.9 MB/s | 84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm          2.4 MB/s | 83 kB  00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm          15 MB/s | 3.2 MB 00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm        36 MB/s | 35 MB  00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm         30 MB/s | 44 MB  00:01

Total                                         56 MB/s | 84 MB  00:01

Run      : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64          8/10
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64          9/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64          10/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64          1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64          2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64          3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64          4/10
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64          5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnfnetwork-1.0.1-19.amzn2023.0.2.x86_64        7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64          8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64          9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64          10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64  docker-25.0.6-1.amzn2023.0.2.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  libcgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetwork-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
```

- After successfully docker has been installed it has to be started on all machines by using the command “`systemctl start docker`”

Complete!

```
[root@ip-172-31-12-97 ec2-user]# systemctl start docker
```

5 Kubernetes installation

Search kubeadm installation on your browser and scroll down and select red hat-based distributions.

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
Linux in permissive mode (effectively disabling it)
enforce 0
-i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
# This overwrites any existing configuration in /etc/yum.repos.d/
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

3. Install kubelet, kubeadm and kubectl:

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

4. (Optional) Enable the kubelet service before running kubeadm:

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

Copy the above given steps and paste in the terminal. This will create a Kubernetes repository, install kubelet, kubeadm and kubectl and also enable the services.

```
[root      ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-12-97 ec2-user]# yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.

=====
| Package           | Architecture | Version        | Repository | Size   |
|=====             |=====         |=====          |=====       |=====  |
| Installing:      |              |                |            |        |
| kubelet           | x86_64       | 1.31.1-150500.1.1 | kubernetes | 11 M   |
| kubeadm          | x86_64       | 1.31.1-150500.1.1 | kubernetes | 11 M   |
|                 |              |                |            |        |
| kube              | x86_64       | 1.31.1-150500.1.1 | kubernetes | 15 M   |
| Installing dependencies: |
| conntrack-tools  | x86_64       | 1.4.6-2.amzn2023.0.2 | amazonlinux | 208 k  |
| cri-tools         | x86_64       | 1.31.1-150500.1.1 | kubernetes | 6.9 M   |
| kubernetes-cni   | x86_64       | 1.5.1-150500.1.1 | kubernetes | 7.1 M   |
| libnetfilter_cthelper | x86_64 | 1.0.0-21.amzn2023.0.2 | amazonlinux | 24 k   |
| libnetfilter_cttimeout | x86_64 | 1.0.0-19.amzn2023.0.2 | amazonlinux | 24 k   |
| libnetfilter_queue | x86_64       | 1.0.5-2.amzn2023.0.2 | amazonlinux | 30 k   |
|                 |              |                |            |        |
| Transaction Summary |
| =====             |
| Install  9 Packages
|
Total download size: 51 M
Installed size: 269 M
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm      500 KB/s |  24 kB   00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm     475 KB/s |  24 kB   00:00
(3/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm      3.6 MB/s | 208 kB  00:00
(4/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm     1.4 MB/s | 30 kB   00:00
(5/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm      1.7 MB/s | 11 MB   00:00
(6/9): kubectl-1.31.1-150500.1.1.x86_64.rpm      15 MB/s | 11 MB   00:00
(7/9): cri-tools-1.31.1-150500.1.1.x86_64.rpm     8.0 MB/s | 6.9 MB  00:00
(8/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm    14 MB/s | 7.1 MB  00:00
(9/9): kubelet-1.31.1-150500.1.1.x86_64.rpm      25 MB/s | 15 MB   00:00
|
=====
| Ins          | libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 | 5/9 |
| Installing   | conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 | 6/9 |
| Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 | 6/9 |
| Installing   | kubelet-1.31.1-150500.1.1.x86_64 | 7/9 |
| Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 | 7/9 |
| Installing   | kubeadm-1.31.1-150500.1.1.x86_64 | 8/9 |
| Installing   | kubectl-1.31.1-150500.1.1.x86_64 | 9/9 |
| Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 | 9/9 |
| Verifying    | conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 | 1/9 |
| Verifying    | libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 | 2/9 |
| Verifying    | libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 | 3/9 |
| Verifying    | libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 | 4/9 |
| Verifying    | cri-tools-1.31.1-150500.1.1.x86_64 | 5/9 |
| Verifying    | kubeadm-1.31.1-150500.1.1.x86_64 | 6/9 |
| Verifying    | kubectl-1.31.1-150500.1.1.x86_64 | 7/9 |
| Verifying    | kubelet-1.31.1-150500.1.1.x86_64 | 8/9 |
| Verifying    | kubernetes-cni-1.5.1-150500.1.1.x86_64 | 9/9 |
|
Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64
kubeadm-1.31.1-150500.1.1.x86_64                    kubectl-1.31.1-150500.1.1.x86_64
kubelet-1.31.1-150500.1.1.x86_64                   kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
```

6. We can check if repository has been created by using yum repolist command.

```
[root@ip-172-31-14-85 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
kubernetes                               Kubernetes
```

7. Now we will be initializing the kubeadm. For that “kubeadm init” command has to be used. It may show errors but those can be ignored by using **--ignore-preflight-errors=all**

```
[root@ip-172-31-14-85 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
[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 15:50:31.271160 29520 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.14.85]
[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-172-31-14-85.ap-southeast-2.compute.internal localhost] and IPs [172.31.14.85 127.0.0.1 ::1]
```

```
85 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal localhost] and IPs [172.31.14.85 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
```

```
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 10.001658622s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 6lysht48enn4gmnho6ex8
[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!
```

8. On successful initialization we need to copy and paste the following commands on the master machine itself:

```
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
```

9. Next copy and paste the join link in the worker nodes so that the worker nodes can join the cluster.

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

```
kubeadm join 172.31.14.85:6443 --token 61ysht.48enn4gmnhof6ex8 \
--discovery-token-ca-cert-hash sha256:461819c971fe032e04a78e18fde8e28755825e8468d468a2c86d88c52dba4945
```

10. After performing join commands on the worker nodes, we will get following output:

```
This node has joined the cluster:
```

```
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.
```

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

11. Once again when you run kubectl get nodes you will now see all 3 nodes have joined the cluster.

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

Conclusion:

This experiment successfully demonstrated the creation of a Kubernetes cluster and the successful addition of all three nodes using various commands. Errors encountered during initialization can be addressed in two ways: 1) by ignoring the errors, or 2) by upgrading the instance type to t3.medium or t3.large if the issues are due to insufficient memory or CPU resources.

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.

Note:

AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account .).

If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance.

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

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.

Instances (1/1) Info										
Find Instance by attribute or tag (case-sensitive) All states										
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/>	Experiment 4	i-09f3752831db50f7d	Running View details Logs	t2.medium	Initializing View alarms +	View alarms +	us-east-1d	ec2-54-165-99-170.co...	54.165.99.170	-

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture

64-bit (x86) ▾

AMI ID

ami-0e86e20dae9224db8

Verified provider

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

Instance type

t2.medium

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

On-Demand Linux base pricing: 0.0464 USD per Hour

On-Demand RHEL base pricing: 0.0752 USD per Hour

On-Demand Windows base pricing: 0.0644 USD per Hour

On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

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

The screenshot shows the AWS EC2 Connect to instance page. At the top, there is a breadcrumb navigation: EC2 > Instances > i-09f3752831db50f7d > Connect to instance. Below the breadcrumb, the title "Connect to instance" has an "Info" link. A sub-instruction "Connect to your instance i-09f3752831db50f7d (Experiment 4) using any of these options" follows. There are four tabs: "EC2 Instance Connect", "Session Manager", "SSH client" (which is selected), and "EC2 serial console". Under "Instance ID", it shows "i-09f3752831db50f7d (Experiment 4)". Below this, a numbered list of steps is provided:

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`

Below the steps, there is an "Example:" section with a copy icon next to the command `ssh -i "Master_Ec2_Key.pem" ubuntu@172.31.20.171`. A note in a callout box states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." At the bottom right of the page is a "Cancel" button.

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

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

```
ubuntu@ip-172-31-20-171:~$ 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-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has 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.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
containerd.io docker-buildx-plugin docker-ce-cli
docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
slirp4netns
Suggested packages:
aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
containerd.io docker-buildx-plugin docker-ce docker-ce-cli
docker-ce-rootless-extras docker-compose-plugin libltdl7 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 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 https://download.docker.com/linux/ubuntu/noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
```

```

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_583a27.2.1-1~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-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
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}

```

EOF

```
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-172-31-20-171:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

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
kubeadm kubectl
sudo apt-mark hold
kubeadm kubectl**

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 6051 B in 0s (12.9 kB/s)
Reading package lists... Done
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 kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 130 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (77.1 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68011 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
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) ...
```

```

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-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0915 07:47:37.419191    7952 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create
new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: cod
e = Unimplemented desc = unknown service runtime.v1.RuntimeService
[WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock"
: rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal
with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher

```

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

```
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
```

...

sudo systemctl restart containerd
sudo systemctl enable 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
        CGroub: /system.slice/containerd.service
                  └─8398 /usr/bin/containerd

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

sudo apt-get install -y socat

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (12.1 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68112 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-172-31-20-171:~$ 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 07:49:42.979851    8570 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-20-171 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.20.171]
[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-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://172.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 502.777379ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 4.501245501s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[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
```

```
[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
```

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

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

kubectl apply -f

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

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

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-172-31-20-171:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-vz8rv	0/1	Pending	0	8s
nginx-deployment-d556bf558-wz5wc	0/1	Pending	0	8s

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

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-20-171	Ready	control-plane	5m23s	v1.31.1

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-vz8rv	1/1	Running	0	3m4s
nginx-deployment-d556bf558-wz5wc	1/1	Running	0	3m4s

```
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-172-31-20-171:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 07:59:03 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

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

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.

Experiment 5

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 with three main sections:

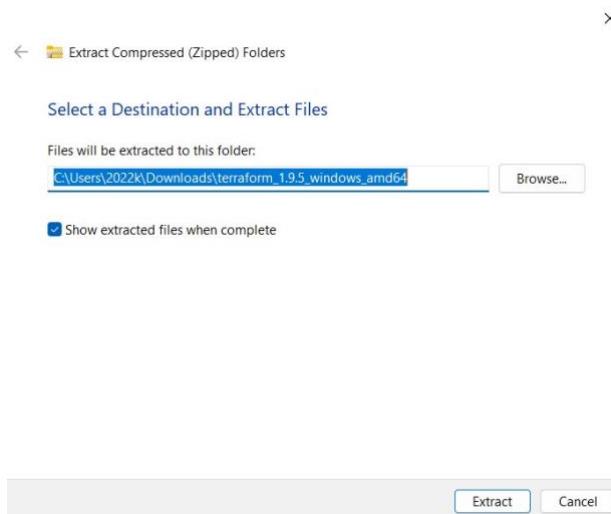
- macOS**:
 - Package manager:

```
brew tap hashicorp/tap
brew install hashicorp/tap/terraform
```
 - Binary download:
 - AMD64 Version: 1.9.5 [Download](#)
 - ARM64 Version: 1.9.5 [Download](#)
- Windows**:
 - Binary download:
 - 386 Version: 1.9.5 [Download](#)
 - AMD64 Version: 1.9.5 [Download](#)
- Linux**: (This section is partially visible at the bottom)

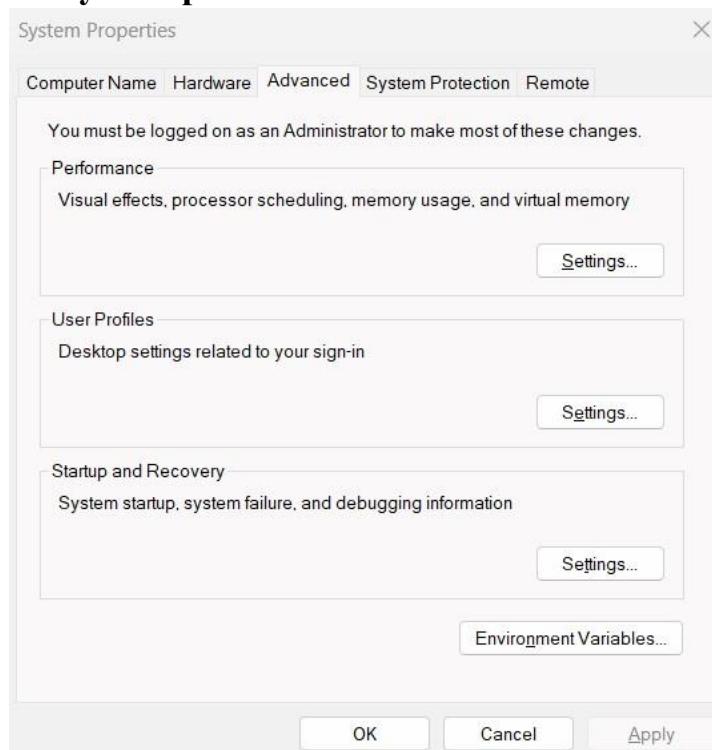
Step 2: Extract the downloaded setup file Terraform.exe in

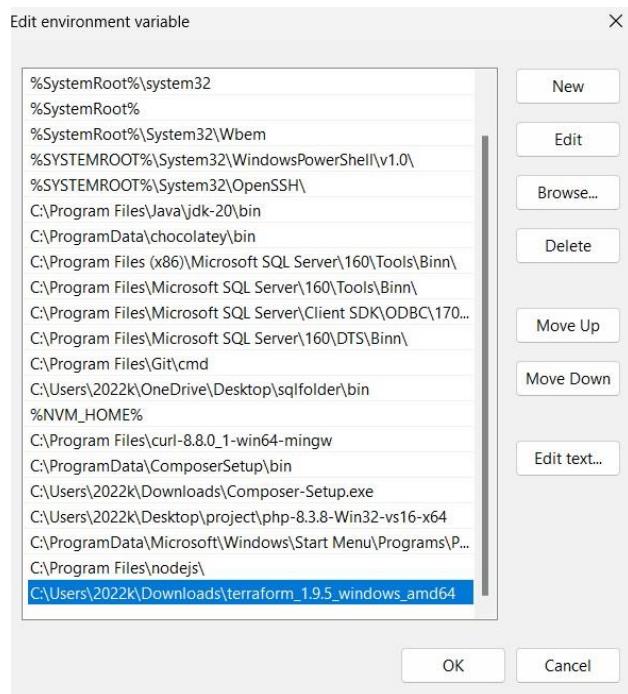


C:\Terraform directory

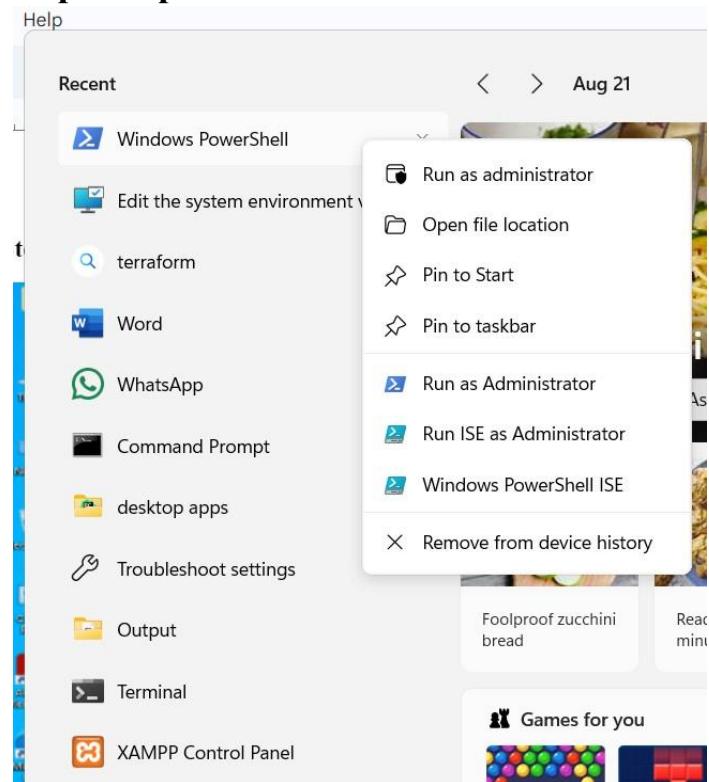


Step 3: Set the System path for Terraform in Environment Variables

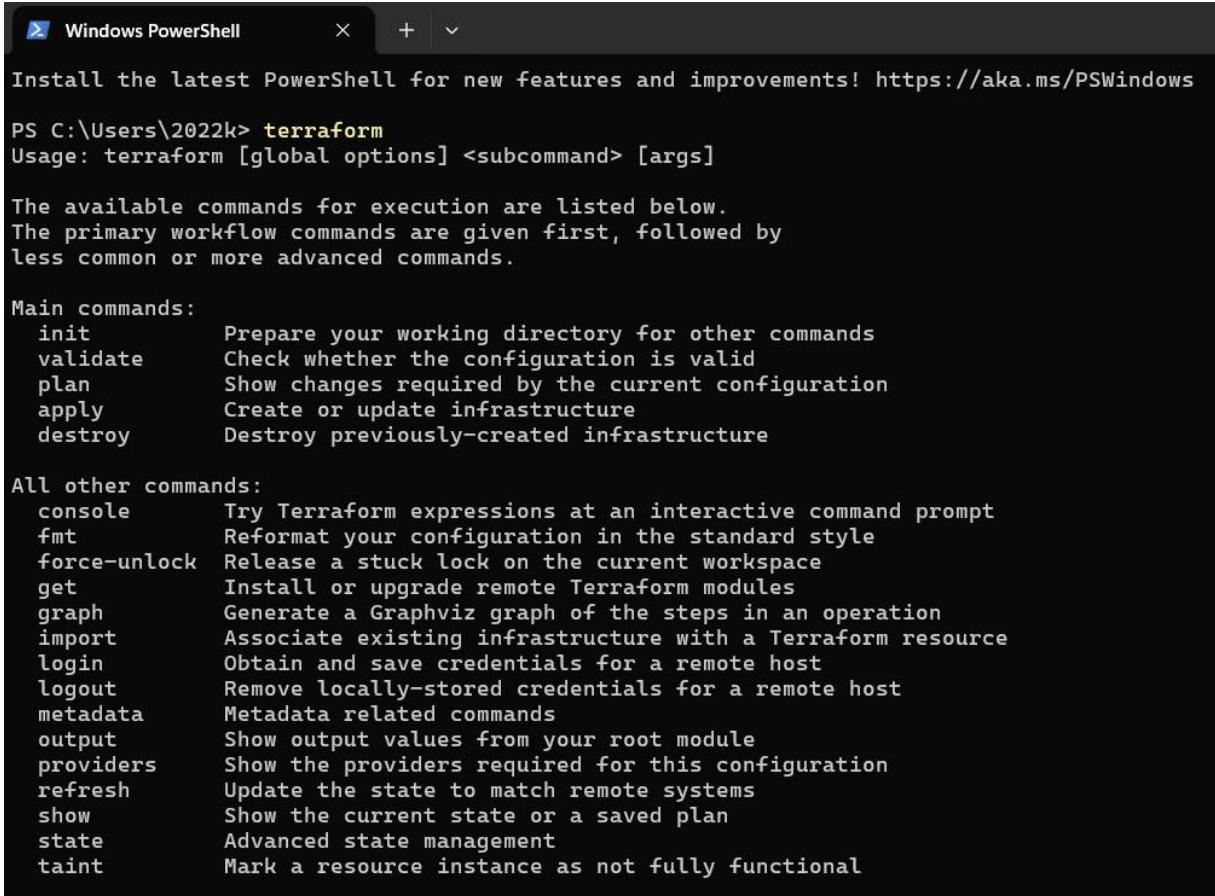




Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command "terraform" was run from the path "PS C:\Users\2022k>". The output displays the usage information and a detailed list of available commands:

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

PS C:\Users\2022k> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

Experiment No. 6

- Creating a docker image using terraform

```
PS C:\ProgramData\Microsoft\Windows\Start Menu> Docker --version
Docker version 27.1.1, build 6312585
PS C:\ProgramData\Microsoft\Windows\Start Menu> docker
```

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

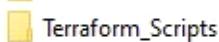
Common Commands:

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

Management Commands:

builder	Manage builds
buildx*	Docker Buildx
checkpoint	Manage checkpoints
compose*	Docker Compose
container	Manage containers
context	Manage contexts
debug*	Get a shell into any image or container
desktop*	Docker Desktop commands (Alpha)
dev*	Docker Dev Environments
extension*	Manages Docker extensions
feedback*	Provide feedback, right in your terminal!
image	Manage images

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



8/22/2024 4:10 PM

File folder

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



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

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

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

- Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

D:\Terraform_Scripts\Docker>
```

- Execute Terraform plan to see the available resources

```
D:\Terraform_Scripts\Docker>terraform plan

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

Terraform will perform the following actions:

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

  + healthcheck (known after apply)

  + labels (known after apply)
}

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

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

- 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”
- Check Docker images, Before and After Executing Apply step

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

D:\Terraform_Scripts\Docker>
```

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

D:\Terraform_Scripts\Docker>
```

- Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.
- Check Docker images, After Executing Destroy step

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

D:\Terraform_Scripts\Docker>
```

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 the following details:

- Left sidebar:** Includes 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.'
- Top right:** Includes a search bar ('Search (CTRL+K)'), a bell icon, a user profile ('Aditya Nagesh Raorane'), and a 'log out' link.
- Main area:** A table showing project details:

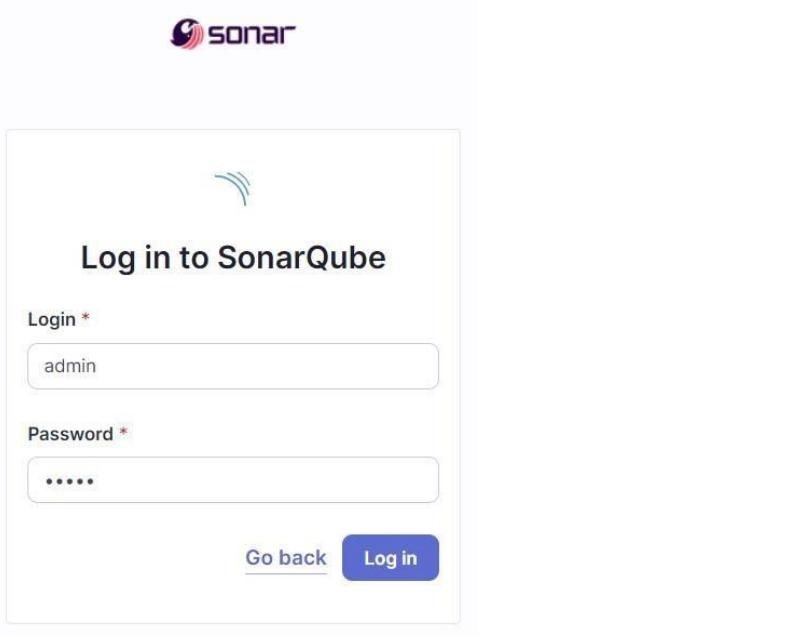
S	W	Name	Last Success	Last Failure	Last Duration
		My_First_Maven	23 days #2	23 days #1	20 sec
		MyPipeline1	28 days #1	N/A	9.2 sec
		Pipeline_01	1 mo 15 days #3	N/A	9.9 sec
		WebTestDriver	1 day 16 hr #5	1 day 16 hr #4	13 sec
- Bottom:** 'Build Executor Status' section showing 0/2 available executors, and icons for 'S' (Stable), 'M' (Medium), and 'L' (Long).

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\Muskan>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\aditya>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
7478be0ac0f23: 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
4a6e73f4472de892b1dddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

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 **mus12**



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 *

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](#)

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 Plugins page. In the top navigation bar, there is a search bar with the placeholder "Search (CTRL+K)" and a user profile icon for "Aditya Nagesh Raorane". Below the search bar, there are several tabs: "Dashboard", "Manage Jenkins", and "Plugins". Under the "Plugins" tab, there are four categories: "Updates" (with one update available), "Available plugins" (highlighted in blue), "Installed plugins" (with 1 item), and "Advanced settings". A search bar at the top right contains the text "sonarqube scanner". Below the search bar, a list of available plugins is shown, with "SonarQube Scanner 2.17.2" being the first item. This plugin is described as allowing an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. It has "External Site/Test Integrations" and "Build Reports" tabs. At the bottom of the list, it says "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." To the right of the plugin list, there are two buttons: "Install" and "Install after restart". The "Install" button is highlighted with a blue background and white text.

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

Name : sonarqube

Server URL : http://localhost:9000

The screenshot shows the Jenkins System configuration page. The top navigation bar includes "Dashboard", "Manage Jenkins", "System", and "Help". Below the navigation, there is a section titled "SonarQube servers". A note states: "If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build." There is a checked checkbox labeled "Environment variables". Below this, there is a "SonarQube installations" section with a link to "List of SonarQube installations". A new server configuration is being added, indicated by a dashed border around the form. The "Name" field contains "sonarqube". The "Server URL" field is set to "Default is http://localhost:9000" and contains "http://localhost:9000". The "Server authentication token" field has a dropdown menu showing "- none -" and a "+ Add +" button. An "Advanced" dropdown menu is partially visible. At the bottom of the form are "Save" and "Apply" buttons.

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

Manage Jenkins → Tools → SonarQube Scanner Installation

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. It displays the configuration for 'SonarQube Scanner installations'. A new installation named 'sonarqube' is being added. The 'Install from Maven Central' section is selected, showing the version 'SonarQube Scanner 6.2.0.4584'. The 'Install automatically' checkbox is checked. There are also sections for 'Ant installations' and 'Add SonarQube Scanner'.

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

The screenshot shows the Jenkins 'New Item' creation page. The 'Freestyle project' option is selected and highlighted. The 'Enter an item name' field contains 'sonarqube'. At the bottom, there is an 'OK' button.

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 'Configure' section of the SonarQube interface. Under 'Source Code Management', the 'Git' option is selected. A 'Repository URL' field contains 'https://github.com/shazforiot/MSBuild_firstproject.git'. Below it, a 'Credentials' dropdown is set to '- none -'. There are 'Save' and 'Apply' buttons at the bottom.

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

The screenshot shows the 'Configure' section of the SonarQube interface. Under 'Build Steps', the 'Execute SonarQube Scanner' step is selected. In the 'Analysis properties' field, the following values are entered: 'sonar.projectKey=sonarqube', 'sonar.login=admin', 'sonar.host.url=http://localhost:9000', and 'sonar.sources='.

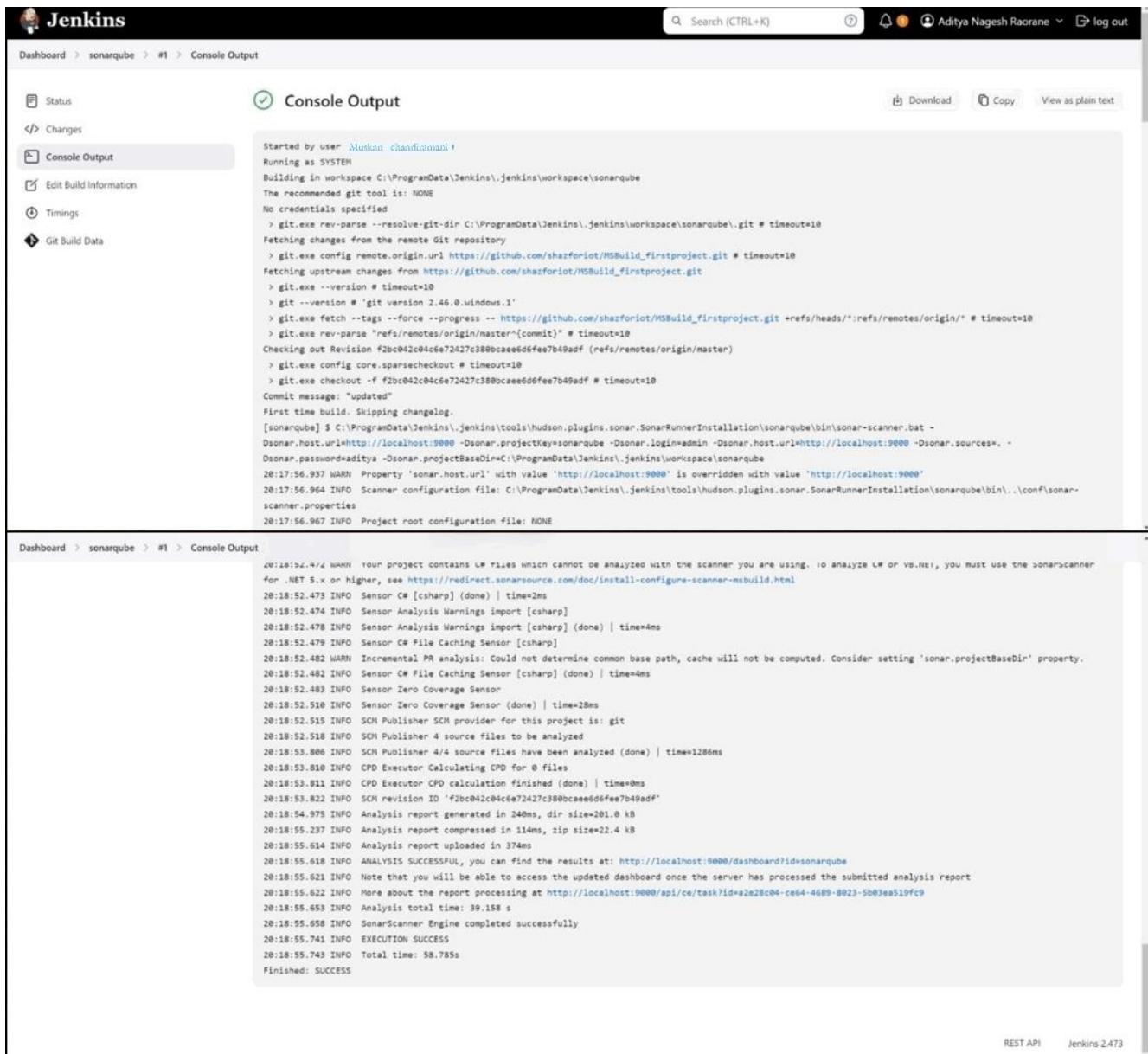
11. Go to <http://localhost:9000/admin/permissions> and allow Execute Permissions to the Admin user.

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

12. Run The Build and check the console output.

The Jenkins dashboard shows the following details:

- Project: sonarqube
- Status: Green (Success)
- Last build: #1, 2 min 3 sec ago
- Build Now button
- Configure, Delete Project, SonarQube, Rename links
- Builds table: #1 (8:17 PM)



The screenshot shows the Jenkins interface with the following details:

- Project Path:** Dashboard > sonarqube > #1 > Console Output
- Build Status:** Success (indicated by a green checkmark icon)
- Console Output Content:**

```

Started by user Muskan chandiramani
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\sonarqube.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git' version 2.46.0.windows.1'
> git.exe fetch -tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcace6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcace6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
[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=aditya -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
20:17:56.937 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
20:17:56.964 INFO Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-scanner.properties
20:17:56.967 INFO Project root configuration file: NONE

```
- Log Output (Bottom):**

```

20:18:52.474 WARN Your project contains L# files which cannot be analyzed with the scanner you are using. To analyze L# or VB.NET, you must use the sonarScanner
for .NET 5.x or higher, see https://reirect.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 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
20:18:53.866 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 'f2bc042c04c6e72427c380bcace6d6fee7b49adf'
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: 39.198 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

```

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

The screenshot shows the SonarQube interface with two main sections. The top section displays a summary of the 'sonarqube PUBLIC' project, indicating it has passed its last analysis 3 minutes ago. It shows 1 issue in the main branch. The bottom section provides a detailed overview of the 'main' branch, including quality gate status (Passed), security, reliability, maintainability, accepted issues, coverage, and duplications.

Category	Value	Grade
Quality Gate	Passed	Green
Security	0 Open issues	A
Reliability	0 Open issues	A
Maintainability	0 Open issues	A
Accepted issues	0	
Coverage	On 0 lines to cover.	
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)), 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"
- Main Content:**
 - Filter buttons: All, S, W, Name (dropdown).
 - Table header: S, W, Name, Last Success, Last Failure, Last Duration.
 - Table rows:

S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	My_First_Maven	23 days #2	23 days #1	20 sec
✓	Sun	MyPipeline_01	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
 - Icon legend: S, M, L.

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\Muskan>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 "**mus12**".



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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

2 of 2

Set up project for Clean as You Code

This 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**.

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
        canner-6.1.0.4477-windows-x64\\\\bin\\\\sonar-scanner.bat \\
        -D sonar.login=<YOUR ID> \\
        -D sonar.password=<YOUR PASSWORD> \\
        -D sonar.projectKey=<YOUR PROJECT KEY> \\
        -D sonar.exclusions=vendor/**,resources/**, */*.java \\
        -D sonar.host.url=http://localhost:9000/"}
    }
}

```

The screenshot shows the Jenkins Pipeline configuration page. At the top, there are tabs for 'Configure' (selected), 'General', 'Advanced Project Options', and 'Pipeline'. The 'Pipeline' tab is currently active, showing a 'Definition' dropdown set to 'Pipeline script'. Below this is a code editor containing Groovy script:

```

1 * node {
2   stage('Cloning the GitHub Repo') {
3     git "https://github.com/shafioriot/001.git"
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') {
7       bat "c:\Users\adity\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=sonar \
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 code editor are two buttons: 'Use Groovy Sandbox' (checked) and 'Pipeline Syntax'. At the bottom 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.



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

```

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 'ba799ba7e1b576f04ad61232b0d412c5e61e5e4'
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=99fcde5-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] }
[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 web interface. At the top, there are navigation links: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. A 'Create Project' button is also visible.

Filters:

- Quality Gate:**
 - Passed: 2
 - Failed: 0
- Reliability:**

Grade	Count
A	1
B	0
C	1
D	0
E	0
- Security:**

Grade	Count
A	2

Projects:

- sonarqube PUBLIC**: Passed, Last analysis: 1 hour ago. Status: The main branch of this project is empty.
- sonarqube-test PUBLIC**: Passed, Last analysis: 16 minutes ago - 683k Lines of Code - HTML, XML, ... Metrics: Security (0), Reliability (68k), Maintainability (164k), Hotspots Reviewed (0.0%), Coverage (—), Duplications (50.6%).

sonarqube-test / main

Overview: 683k Lines of Code · Version not provided · Set as homepage · Last analysis 38 minutes ago

Passed (Quality Gate)

New Code · **Overall Code**

Metrics:

- Security:** 0 Open issues (0 H, 0 M, 0 L)
- Reliability:** 68K Open issues (0 H, 47k M, 21k L)
- Maintainability:** 164K Open issues (7 H, 143k M, 21k L)
- Accepted issues:** 0 (Valid issues that were not fixed)
- Coverage:** 50.6% (On 0 lines to cover.)
- Duplications:** 50.6% (On 759k lines.)

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

11. Code Problems Open Issues

SonarQube interface showing the 'Measures' tab. The left sidebar displays metrics such as Security Review, Duplications, Size, Complexity, and Issues. The main panel shows the total number of open issues as 210,549. The right panel provides a detailed tree view of the issues across different files and packages, including gameoflife-acceptance-tests, gameoflife-build, gameoflife-core, gameoflife-deploy, gameoflife-web, and pom.xml.

Consistency

SonarQube interface showing the 'Issues' tab. The left sidebar filters for 'Consistency' issues. The main panel lists specific consistency-related code smells with checkboxes for fixing them. Examples include 'Insert a <!DOCTYPE> declaration to before this <html> tag.' and 'Remove this deprecated "width" attribute.'

Intentionality

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Filters My Issues All Clear All Filters

Issues in new code

Clean Code Attribute

- Consistency: 197k
- Intentionality**: 14k
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

Software Quality

Bulk Change Select issues Navigate to issue 13,887 issues 59d effort

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. **Maintainability** L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Open Not assigned
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Maintainability** L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Open Not assigned
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Maintainability** L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Open Not assigned

Intentionality

No tags

Embedded database should be used for evaluation purposes only

Code Smells

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

Add to selection Ctrl + click

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** accessibility wcag2-a
- Open Not assigned
- Add an "alt" attribute to this image. **Reliability** accessibility wcag2-a
- Open Not assigned

Intentionality

accessibility wcag2-a

L29 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Minor

L31 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Minor

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

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

Intentionality

Embedded database should be used for evaluation purposes only

The embedded database will not scale. It will not support connection to remote instances of SonarQube, and there is no support for retrieving issue data out of it into a different database provider.

Bugs

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)

Add to selection Ctrl + click

Scope

Status

Security Category

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

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

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

- Add "lang" and/or "xml:lang" attributes to this "<html>" element

Intentionality: Reliability: Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Embedded database should be used for evaluation purposes only

Reliability

Reliability

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

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)

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

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

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

- Add "lang" and/or "xml:lang" attributes to this "<html>" element

Intentionality: Reliability: Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Embedded database should be used for evaluation purposes only

Duplicates

	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/jo.../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 tomcat image runs with root as the default user. Make sure it is safe here. Medium

Running containers as a privileged user is security-sensitive docker:S6471

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review priority: Medium

Category: Permission

Assignee: Not assigned

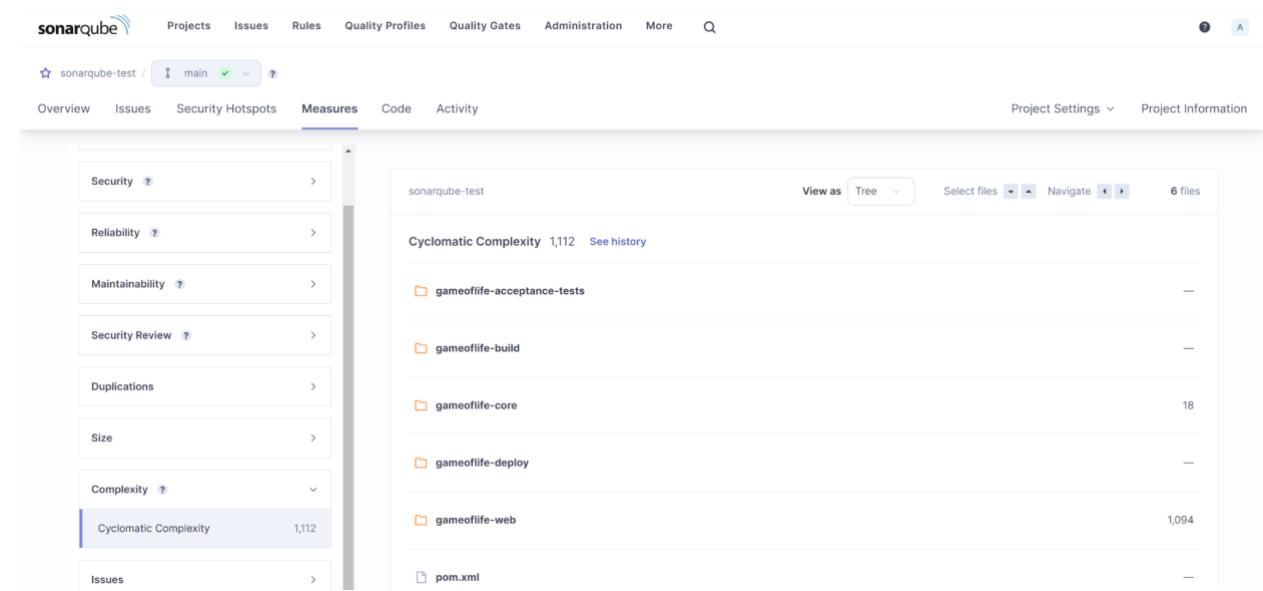
Where is the risk? gameoflife-web/Dockerfile

```

1 FROM tomcat:8-jre8
2
3
4
5
6
7
8
9
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



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.

Adv DevOps Practical 9

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

Theory:

What is Nagios?

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

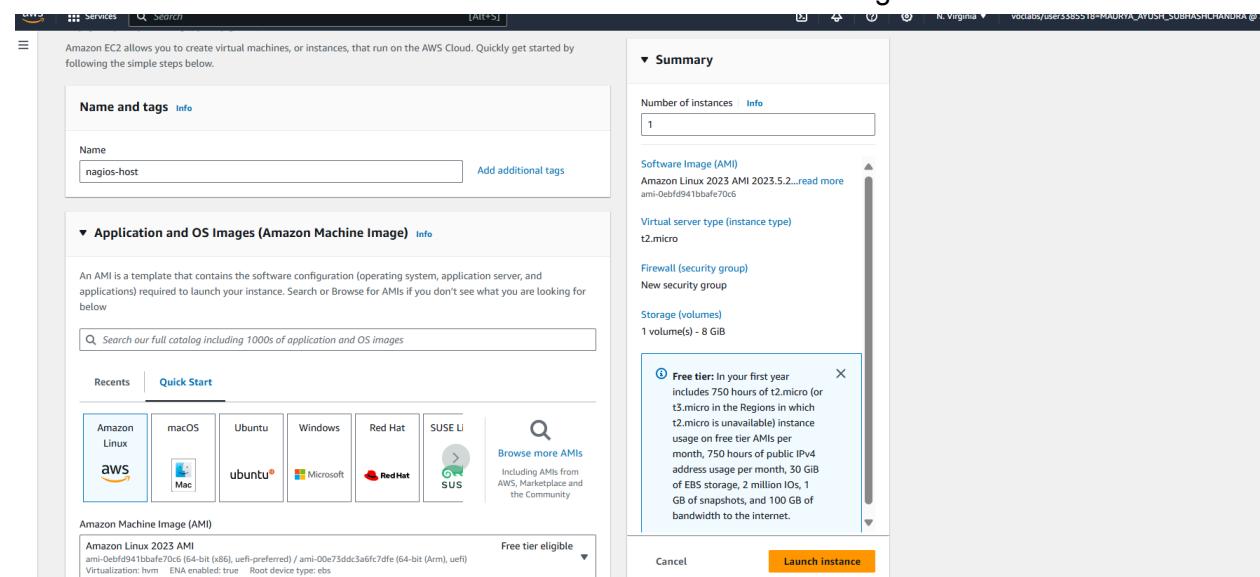
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

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



Instance type

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

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

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

Key pair name - required

[Create new key pair](#)

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or

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

Create security group Select existing security group

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

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

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

EC2 Dashboard

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

Last updated less than a minute ago

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

i-0011127bbfdb2f467 (nagios-host)

Security details

IAM Role: -

Owner ID: 217253764927

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

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

Security Groups (6) [Info](#)

[Find resources by attribute or tag](#)

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

EC2 > Security Groups > sg-09d51590eb1851b46

sg-09d51590eb1851b46 - launch-wizard-3

[Actions ▾](#)

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

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

Inbound rules (1)

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

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

Edit inbound rules info

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

[Inbound rules info](#)

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

[Add rule](#)

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

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

Inbound rules (7)

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

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

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

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

Connect to instance [Info](#)

Connect to your instance i-0011127bbfdb2f467 (nagios-host) using any of these options

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

Instance ID
i-0011127bbfdb2f467 (nagios-host)

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

Example:
ssh -i "exp_09.pem" ec2-user@ec2-44-204-11-28.compute-1.amazonaws.com

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

Or open command prompt and paste ssh command.

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

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

' _#
~ \_\#\#\#_      Amazon Linux 2023
~~ \_\#\#\#\#
~~ \#\#\#
~~ #/ __- https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' __->
~~ / /
~~ ._. / /
~/m'

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

sudo yum update

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

sudo yum install httpd php

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

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:41 ago on Sun Sep 29 06:56:15 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
gcc              x86_64      11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin-docs      noarch     10.93-1.amzn2023.0.1
annobin-plugin-gcc x86_64      10.93-1.amzn2023.0.1
cpp              x86_64      11.4.1-2.amzn2023.0.2
gc               x86_64      8.0.4-5.amzn2023.0.2
glibc-devel       x86_64      2.34-52.amzn2023.0.11
glibc-headers-x86 noarch     2.34-52.amzn2023.0.11
guile22          x86_64      2.2.7-2.amzn2023.0.3
kernel-headers    x86_64      6.1.10-12.amzn2023
libgcc            x86_64      1.2.1-2.amzn2023.0.2
libltool-ltdl     x86_64      2.4.7-1.amzn2023.0.3
libcrypt-devel    x86_64      4.4.33-7.amzn2023
make              x86_64      1:4.3-5.amzn2023.0.2
=====
Transaction Summary
=====
Install 13 Packages
Total download size: 52 M
=====
Installed:
annobin-docs-10.93-1.amzn2023.0.1.noarch
gcc-8.0.4-5.amzn2023.0.2.x86_64
glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
libmpc-1.2.1-2.amzn2023.0.2.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
=====
Complete!
```

sudo yum install gd gd-devel

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

```

Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
glib2-devel-2.71-768.amzn2023.0.2.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-1.0.6-1.amzn2023.0.1.x86_64
libtracks-core-font-en-3.6-1.amzn2023.0.4.noarch
libX11-2.3.4-2.amzn2023.0.4.x86_64
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXrender-1.9.10-14.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libxcb-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

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

Complete!

```

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

sudo adduser -m nagios

sudo passwd nagios

(password : ayushmau)

```

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

```

6. Create a new user group

sudo groupadd nagcmd

```

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

```

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

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```

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

```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads

```

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

```

9. Use wget to download the source zip files.

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

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

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

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

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

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

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

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

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

tar zxvf nagios-4.5.5.tar.gz

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

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

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

Here we go an error

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

Solution

Navigate to nagios folder in downloads

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

Error 2: Cannot find SSL headers.

Solution: Install openssl dev library

Steps:

sudo yum install openssl-devel

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

Now run

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

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

12. Compile the source code.

make all

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

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

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

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

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

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

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

```

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

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

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

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

define contactgroup {
    contactgroup_name   admins
    alias              Nagios Administrators
    members            nagiosadmin
}

```

And change email with your email

15. Configure the web interface.

sudo make install-webconf

```

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

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

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

```

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

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

```

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |

```

Password: Ayushmau

17. Restart Apache

sudo service httpd restart

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

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

```
cd ~/downloads
```

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

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

19. Compile and install plugins

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

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
```

```
make
```

```
sudo make install
```

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

20. Start Nagios

Add Nagios to the list of system services

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

```
sudo chkconfig nagios on
```

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

Verify the sample configuration files

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

Error

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

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

Solution:

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

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

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

Now run again

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

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0
```

```
sudo service nagios start
```

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

21. Check the status of Nagios

```
sudo systemctl status nagios
```

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

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

Error:

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

Solution:

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

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

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

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

Set permissions: Ensure the directory has the right permissions:

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

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

```
sudo systemctl restart nagios
```

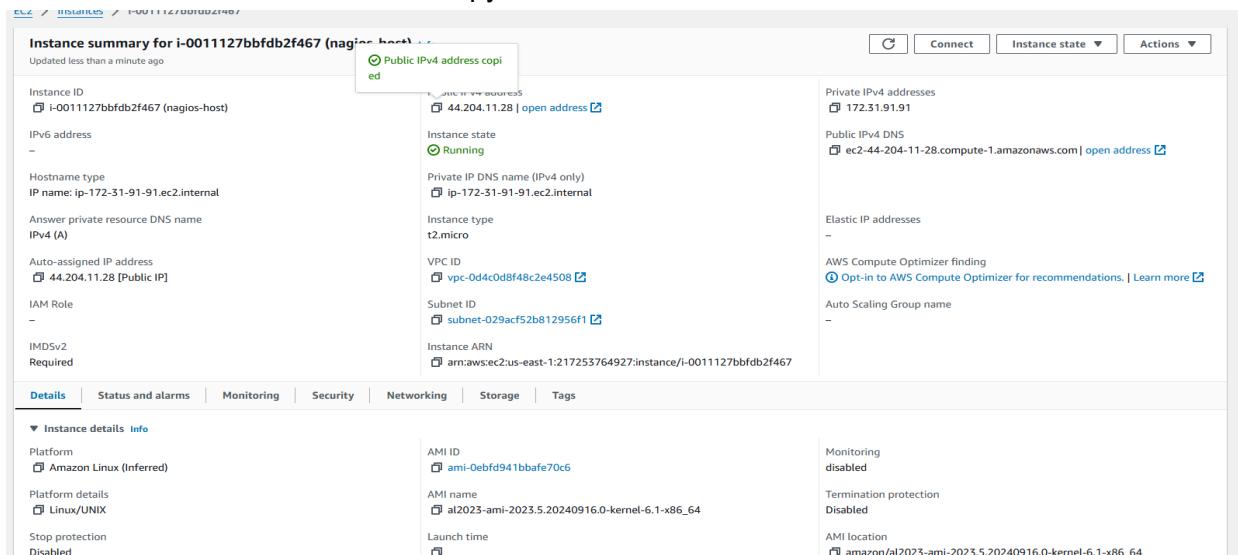
```
drwxr-xr-x. 4 root root 112 Sep 29 08:04 /usr/local/nagios/var/
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagcmd /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chmod -R 775 /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$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

Now run again

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

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

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



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

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

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

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

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

Conclusion:

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

Adv DevOps Practical 10

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

Theory:

Port and Service Monitoring

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

Windows/Linux Server Monitoring

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

Prerequisites:

AWS Academy or Personal account.

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

Monitoring Using Nagios:

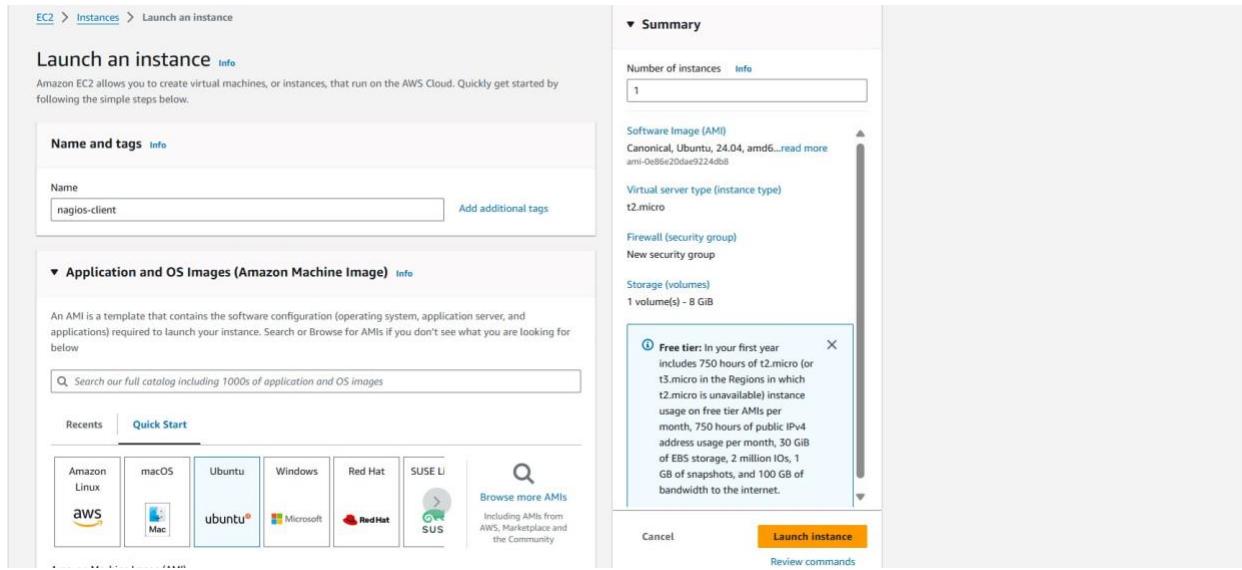
Step 1: To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host). **sudo systemctl status nagios**

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

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

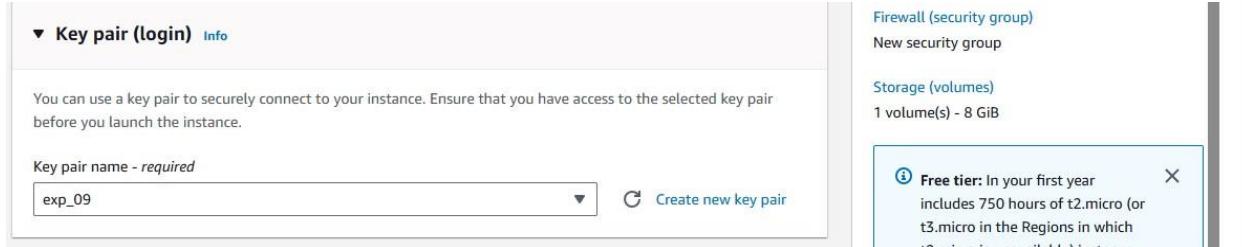
You can now proceed if you get the above message/output.

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

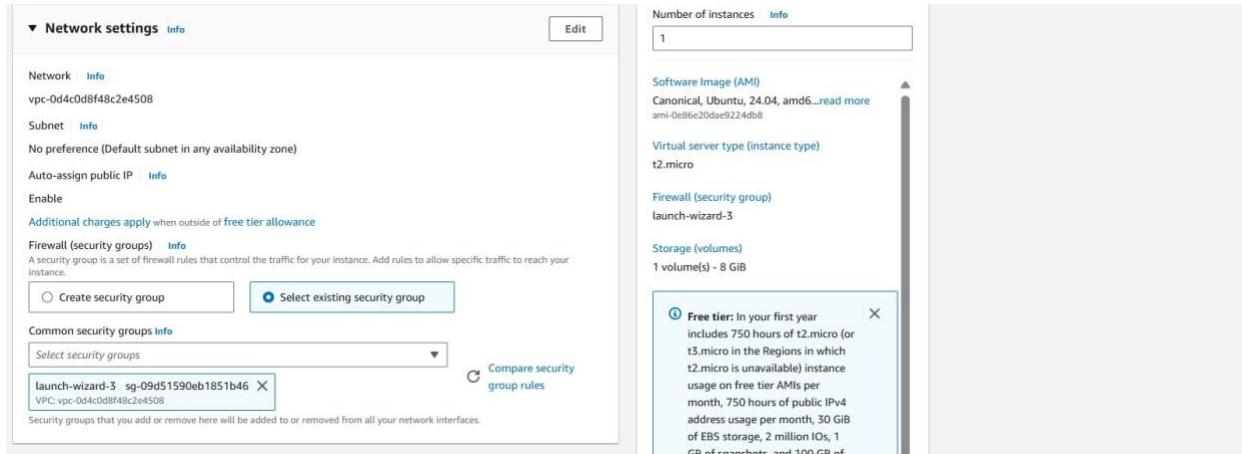


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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



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



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

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

```
PS C:\Users\ MUSKAANNN > ssh -i "Downloads/exp_09.pem" ubuntu@ec2-44-206-245-149.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:DT+AA+mKcydh3kOJ2vEpm4ZsA6FL+LM4mIQS1mddAHg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-245-149.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

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

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

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

ubuntu@ip-172-31-92-146:~$ |
```

Now perform all the commands on the Nagios-host till step 10

Step 4: Now on the server Nagios-host run the following command.

ps -ef | grep nagios

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

Step 5: Now Become root user and create root directories.

sudo su

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

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

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

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

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

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

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

Change hostname to **linuxserver**.

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

Set hostgroup name to **linux-servers1**.

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

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

    host_name      linuxserver
    alias          localhost
    address        172.31.92.146
}

#####
# HOST GROUP DEFINITION
#####
# Define an optional hostgroup for Linux machines

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

Step 8: Now update the Nagios config file .Add the following line in the file. Line to add :

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

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

```
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#####

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

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

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

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

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

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

Running pre-flight check on configuration data...

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

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

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

```
service nagios restart
```

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-91-91 ec2-user]# |
```

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

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

```
ubuntu@ip-172-31-92-146:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [389 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2888 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [381 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

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

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

```
User sessions running outdated binaries:
ubuntu @ session #2: sshd[992,1102]
ubuntu @ session #7: sshd[1198,1248]
ubuntu@ip-172-31-92-146:~$
```

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

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

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,34.207.68.187

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

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

```
sudo systemctl restart nagios-nrpe-server
```

```
monitoring-plugins is already the newest version (2.15.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
ubuntu@ip-172-31-92-146:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-146:~$ |
```

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

sudo systemctl status nagios

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-09-29 17:28:07 UTC; 12min ago
       Docs: https://www.nagios.org/documentation
    Process: 4761 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 4762 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 4763 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.1M
     CPU: 234ms
    CGroup: /system.slice/nagios.service
            └─4763 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─4764 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─4767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─4768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh

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

sudo systemctl status httpd

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

sudo systemctl start httpd

sudo systemctl enable httpd

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

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

<http://34.207.68.187/nagios>

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

Now Click on Hosts from left side panel

Conclusion:

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

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

STEP1:Go on your AWS console account and search for lambda and then go on create function Select the author from scratch, add function name and then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

The screenshot shows the AWS Lambda 'Create function' wizard. At the top, there are four options: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. Below this is a 'Basic information' section where the function name is set to 'lambdaexp11', runtime is 'Python 3.12', architecture is 'x86_64', and permissions are managed via a 'Change default execution role' link.

Create function Info

Choose one of the following options to create your function.

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

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

Basic information

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

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

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

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

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](#)

STEP 2: After the function is created successfully go on code write the default code and then configure them.

Successfully created the function lamdaexp11. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamdaexp11

lamdaexp11

Function overview [Info](#)

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

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

Diagram [Template](#)

lamdaexp11

Layers (0)

+ Add trigger [+ Add destination](#)

Description
-

Last modified
16 seconds ago

Function ARN
arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11

Code source [Info](#)

[Upload from ▾](#)

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

lambda_function Environment Vari
lambda_function.py

```

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

```

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

General configuration [Info](#) [Edit](#)

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

General configuration

- Triggers
- Permissions
- Destinations
- Function URL
- Environment variables
- Tags
- VPC

STEP 3: Then go on edit basic settings and add the description and then save it .

Lambda > Functions > lamdaexp11 > Edit basic settings

Edit basic settings

Basic settings [Info](#)

Description - *optional*

D15C

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.

128 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

STEP 4: Click on “use an existing role” option and then ahead add the role and save it.

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

None

Supported runtimes: Java 11, Java 17, Java 21.

Timeout

0 min 1 sec

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

Use an existing role
 Create a new role from AWS policy templates

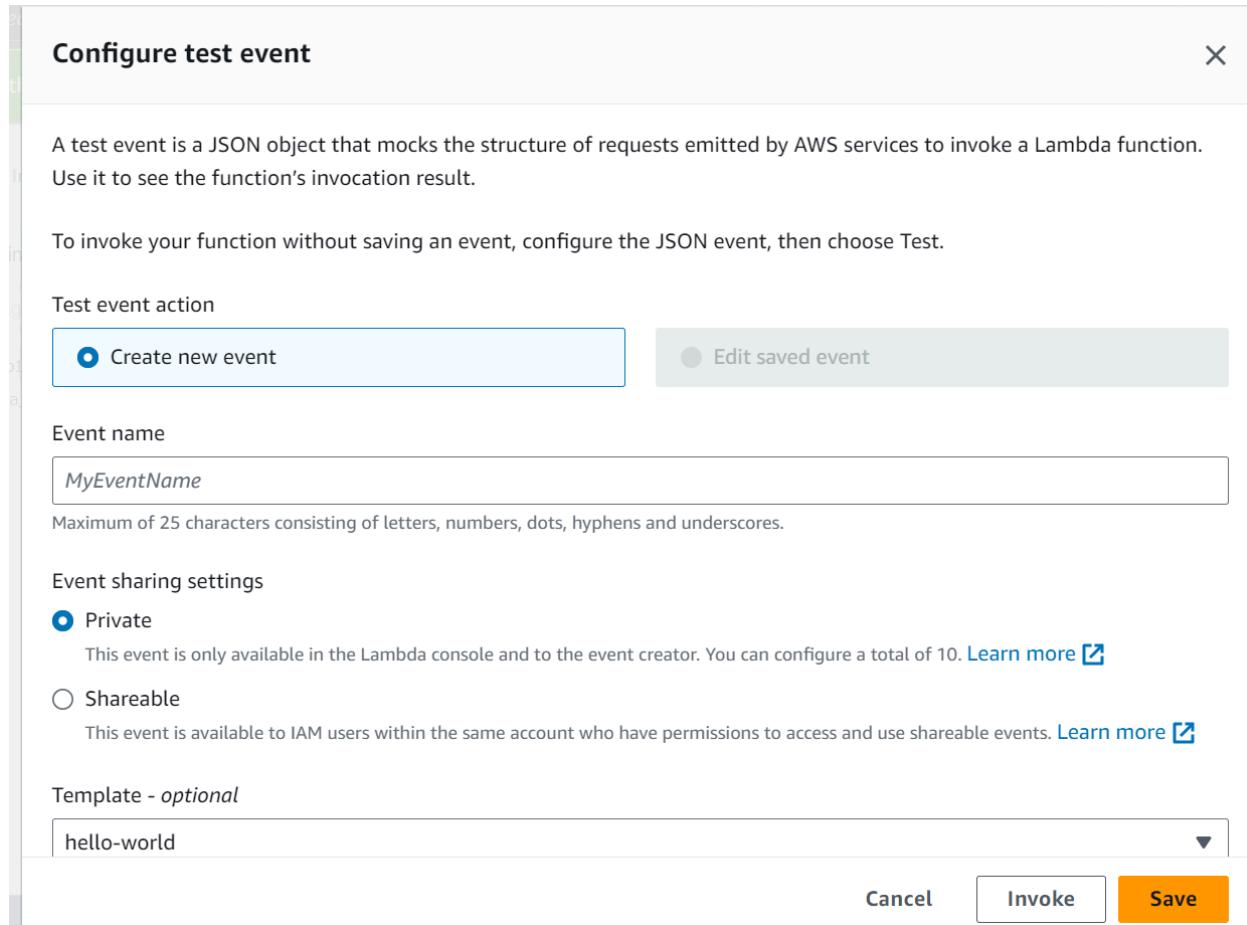
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/lamdaexp11-role-vj5j9g95

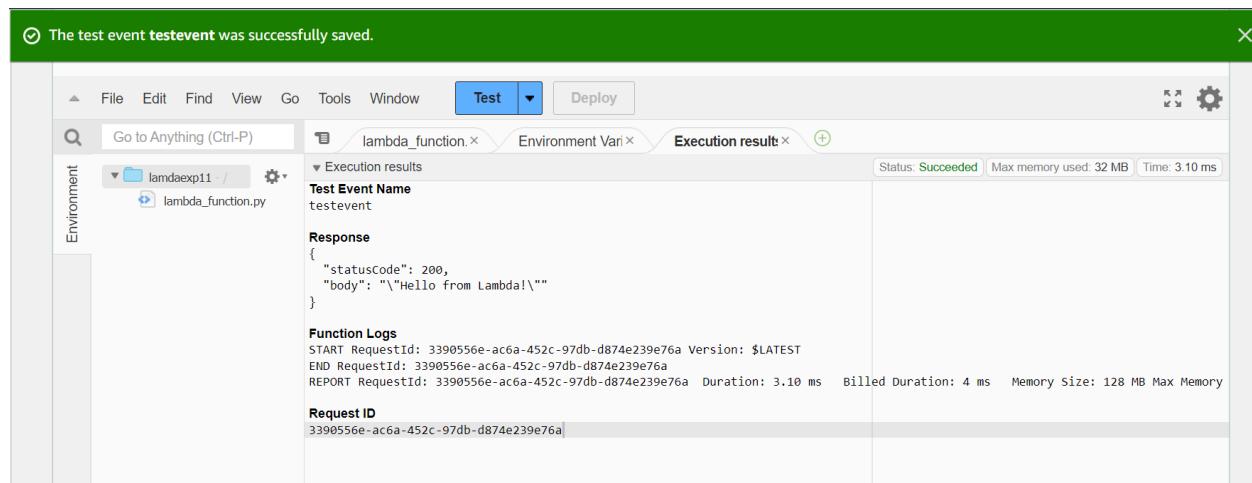
[View the lamdaexp11-role-vj5j9g95 role](#) on the IAM console.

Cancel **Save**

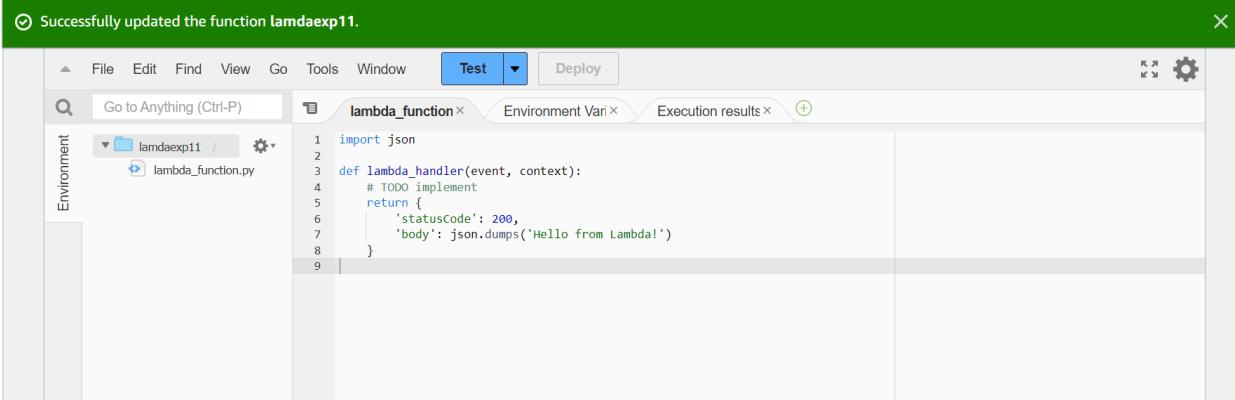
STEP 5: Go on configure test event click on “create new event” edit the event sharing accordingly and select hello world template for template option and then save it.



STEP 6: Click on the test and test the code.



STEP 7: The function is successfully added .



The screenshot shows the AWS Lambda function editor interface. At the top, a green banner displays the message "Successfully updated the function lambdaexp11.". Below the banner, the main window has a toolbar with "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test" (which is currently selected), and "Deploy". To the right of the toolbar are three small icons: a gear, a magnifying glass, and a plus sign. The left sidebar is titled "Environment" and contains a search bar labeled "Go to Anything (Ctrl-P)" and a folder icon labeled "lambdaexp11 - /". The main workspace is titled "lambda_function" and contains the following Python code:

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

Conclusion: In conclusion, the experiment successfully involved the creation, coding, and deployment of AWS Lambda function. By writing and refining the source code, we demonstrated the ability to implement specific functionality within the Lambda environment. The successful testing of the function confirmed its operational integrity and effectiveness in executing the desired tasks.

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

STEPS:

1. Create a S3 bucket and give it a bucket name

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [Info](#)

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

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
exp12d15c

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

2. Allow public access to the bucket as we are going to add this bucket as a trigger for our lambda function

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](#)

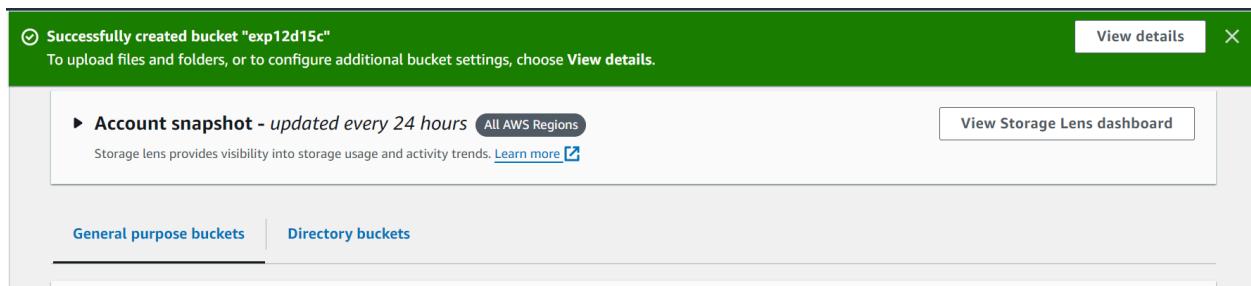
- Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions 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.

3. Give confirmation that you want to allow full public access and create the bucket

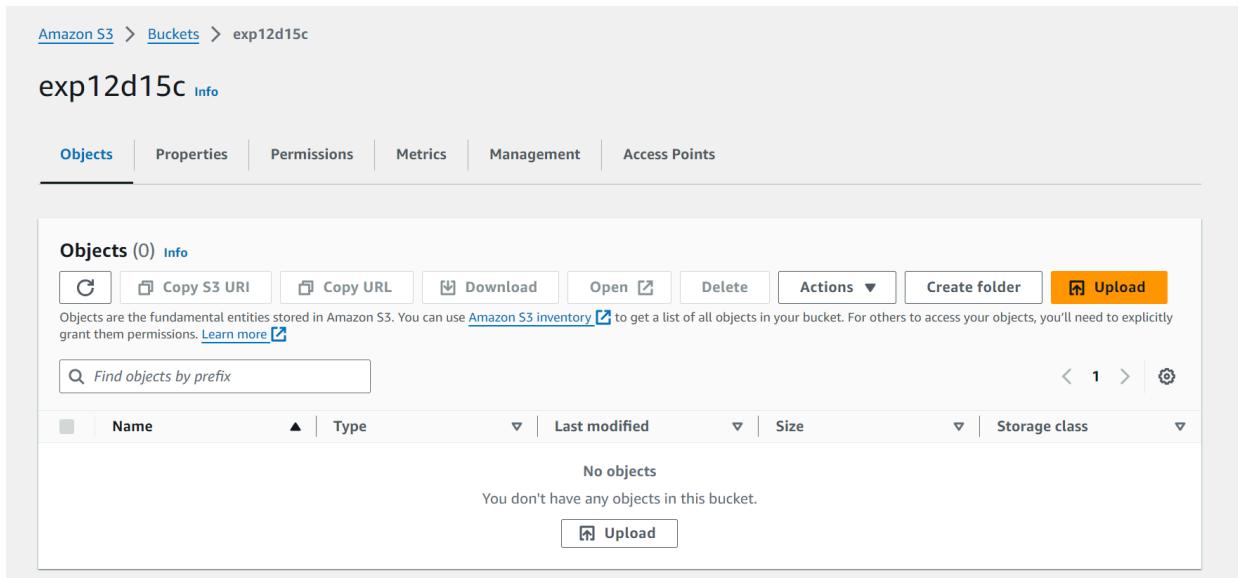
⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

4. You will see the confirmation that the bucket is created successfully



5. Now we need to upload something in the bucket so click on the upload button and add a file



6. I have added a .png extension file; You can upload a .txt file as well

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, 293.3 KB)

All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	AppBar(title Text('Guidelines'),),....	-	image/png

Remove **Add files** **Add folder**

7. Here you can see the confirmation that the upload was a success

Upload succeeded
View details below.

Summary

Destination	Succeeded	Failed
s3://exp12d15c	1 file, 293.3 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 293.3 KB)

Name	Folder	Type	Size	Status	Error
AppBar(title...	-	image/png	293.3 KB	Succeeded	-

8. Now go back to the aws dashboard and search for lamda function service, Open the function we created in experiment 10. We are going to add this bucket as a trigger to this function

9. On the function overview section of the dashboard you can see the “Add trigger” button.
Click on that

10. It will lead you to the trigger configuration tab; Where you have to select the service and the bucket you created. Add the required configuration information and then save.

11. Here you can see we have the confirmation message as well the the s3 bucket added to our triggers

The screenshot shows the AWS Lambda console. In the top navigation bar, it says "Lambda > Functions > lamdaexp11". The main title is "lamdaexp11". On the right, there are buttons for "Throttle", "Copy ARN", and "Actions". A success message box says "The trigger exp12d15c was successfully added to function lamdaexp11. The function is now receiving events from the trigger." Below this, the "Function overview" section is expanded, showing a "Diagram" tab selected. The diagram illustrates the function "lamdaexp11" with an "S3" trigger. There are buttons for "+ Add destination" and "+ Add trigger". To the right, detailed function metadata is listed: Description (D15C), Last modified (18 minutes ago), Function ARN (arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11), and Function URL (Info). There is also a "Layers" section which is currently empty (0).

12. Test the code by clicking on the Test tab ; Here as you can see our code ran successfully

The screenshot shows the "Test" tab for the "lamdaexp11" function. The "Code source" tab is selected. The "Test" tab has a dropdown menu. The "Execution results" section shows a successful run with the following details: Test Event Name (testevent), Response ({"statusCode": 200, "body": "\"Hello from Lambda!\""}, Function Logs (START RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Version: \$LATEST, END RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3, REPORT RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Duration: 1.97 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3), and Request ID (6e57026c-6bfe-41fe-898d-2cdbe72fb1a3).

Conclusion: In conclusion, the experiment successfully demonstrated the integration of an S3 bucket with an AWS Lambda function as a trigger. By creating the S3 bucket and configuring it to invoke the Lambda function upon object uploads, we established a seamless workflow for automated processing.

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

STEPS:

1. Create a S3 bucket and give it a bucket name

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [Info](#)

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

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
exp12d15c

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

2. Allow public access to the bucket as we are going to add this bucket as a trigger for our lambda function

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](#)

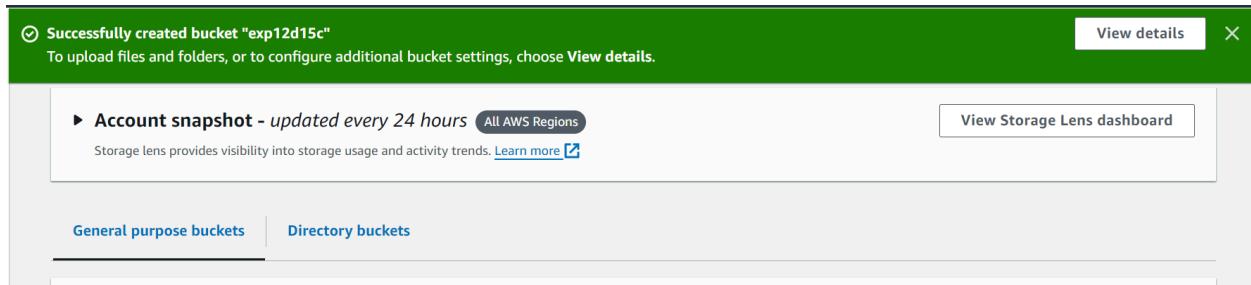
- Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions 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.

3. Give confirmation that you want to allow full public access and create the bucket

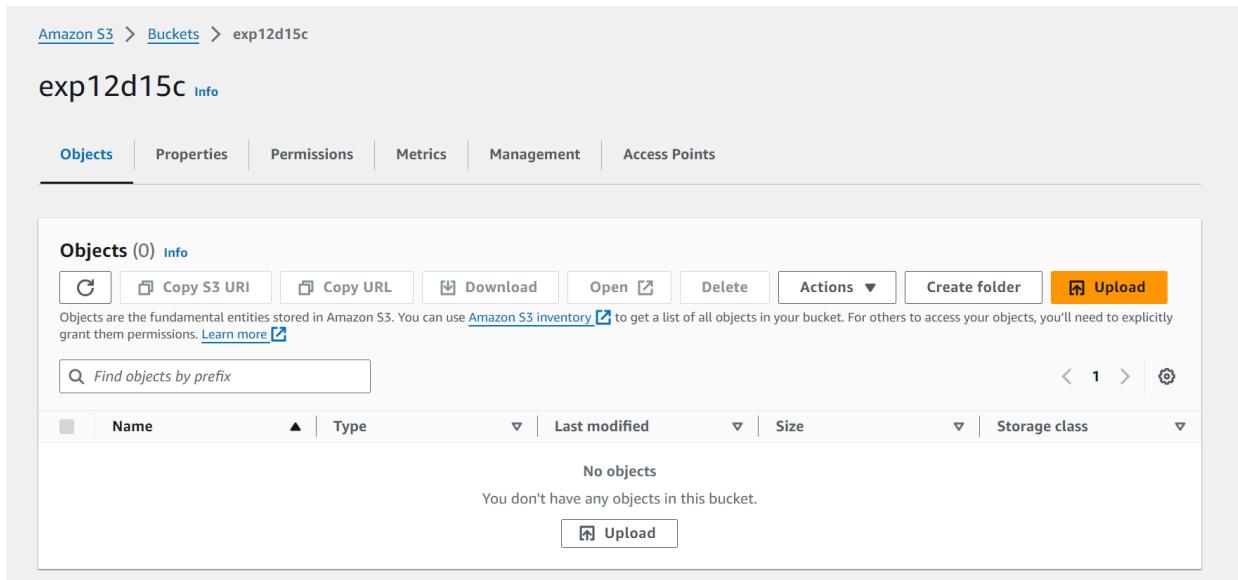
⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

4. You will see the confirmation that the bucket is created successfully



5. Now we need to upload something in the bucket so click on the upload button and add a file



6. I have added a .png extension file; You can upload a .txt file as well

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, 293.3 KB)

All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	AppBar(title Text('Guidelines'),),....	-	image/png

Remove **Add files** **Add folder**

7. Here you can see the confirmation that the upload was a success

Upload succeeded
View details below.

Summary

Destination	Succeeded	Failed
s3://exp12d15c	1 file, 293.3 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 293.3 KB)

Name	Folder	Type	Size	Status	Error
AppBar(title...	-	image/png	293.3 KB	 Succeeded	-

8. Now go back to the aws dashboard and search for lamda function service, Open the function we created in experiment 10. We are going to add this bucket as a trigger to this function

9. On the function overview section of the dashboard you can see the “Add trigger” button.
Click on that

10. It will lead you to the trigger configuration tab; Where you have to select the service and the bucket you created. Add the required configuration information and then save.

11. Here you can see we have the confirmation message as well the the s3 bucket added to our triggers

The screenshot shows the AWS Lambda console. In the top navigation bar, it says "Lambda > Functions > lamdaexp11". The main title is "lamdaexp11". On the right, there are buttons for "Throttle", "Copy ARN", and "Actions". A success message box says "The trigger exp12d15c was successfully added to function lamdaexp11. The function is now receiving events from the trigger." Below this, the "Function overview" section is expanded, showing a "Diagram" tab selected. The diagram illustrates the function "lamdaexp11" with an "S3" trigger. There are buttons for "+ Add destination" and "+ Add trigger". To the right, detailed function metadata is listed: Description (D15C), Last modified (18 minutes ago), Function ARN (arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11), and Function URL (Info). There is also a "Layers" section which is currently empty (0).

12. Test the code by clicking on the Test tab ; Here as you can see our code ran successfully

The screenshot shows the "Test" tab for the "lamda_function" test event. The "Execution results" section displays a successful run with the following details: Status: Succeeded, Max memory used: 32 MB, Time: 1.97 ms. The "Response" field shows the output: {"statusCode": 200, "body": "\"Hello from Lambda!\""}.

Conclusion: In conclusion, the experiment successfully demonstrated the integration of an S3 bucket with an AWS Lambda function as a trigger. By creating the S3 bucket and configuring it to invoke the Lambda function upon object uploads, we established a seamless workflow for automated processing.