

## EXPERIMENT 1A

**Aim:** To develop a website and host it on i) local machine or virtual machine  
ii) Amazon S3 Bucket

### Static Hosting:

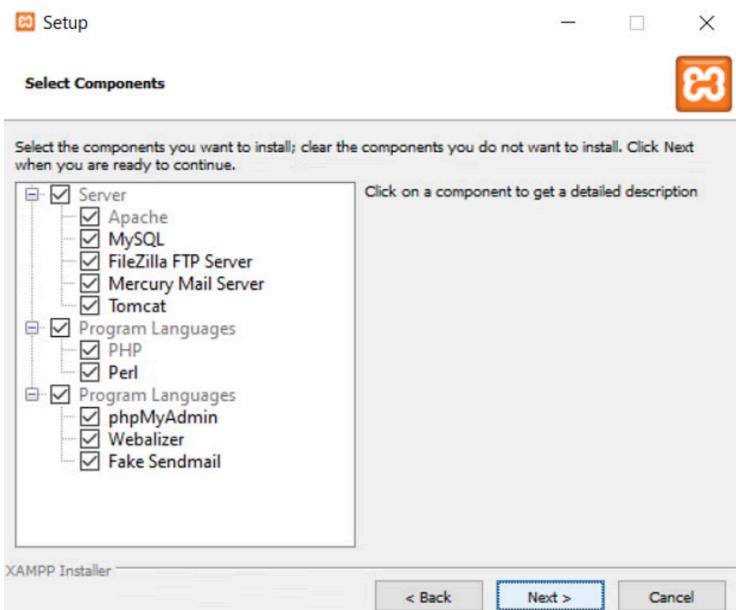
- 1) On local server (XAMPP)

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

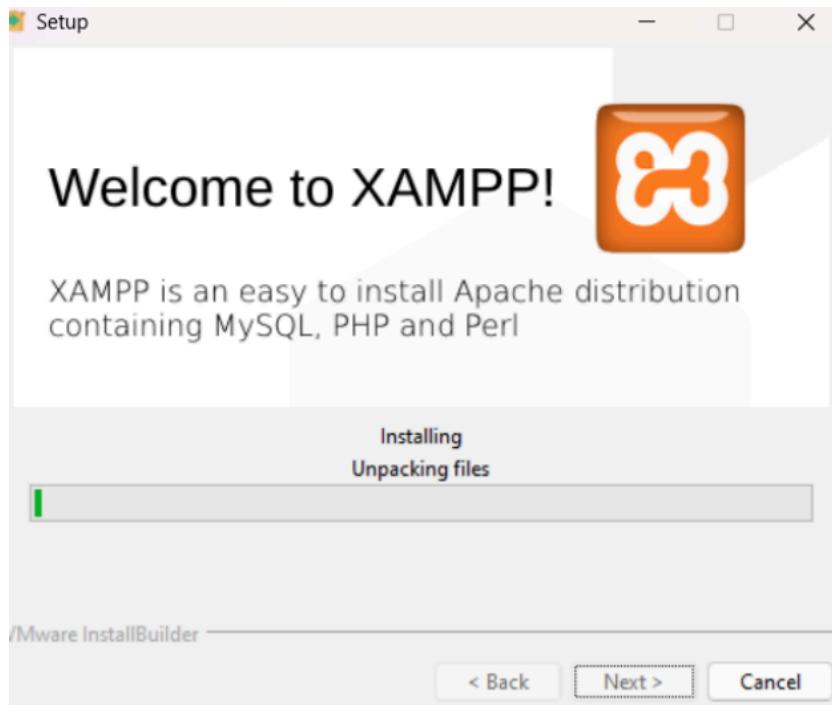
- 1) Select your OS. It will automatically start downloading.



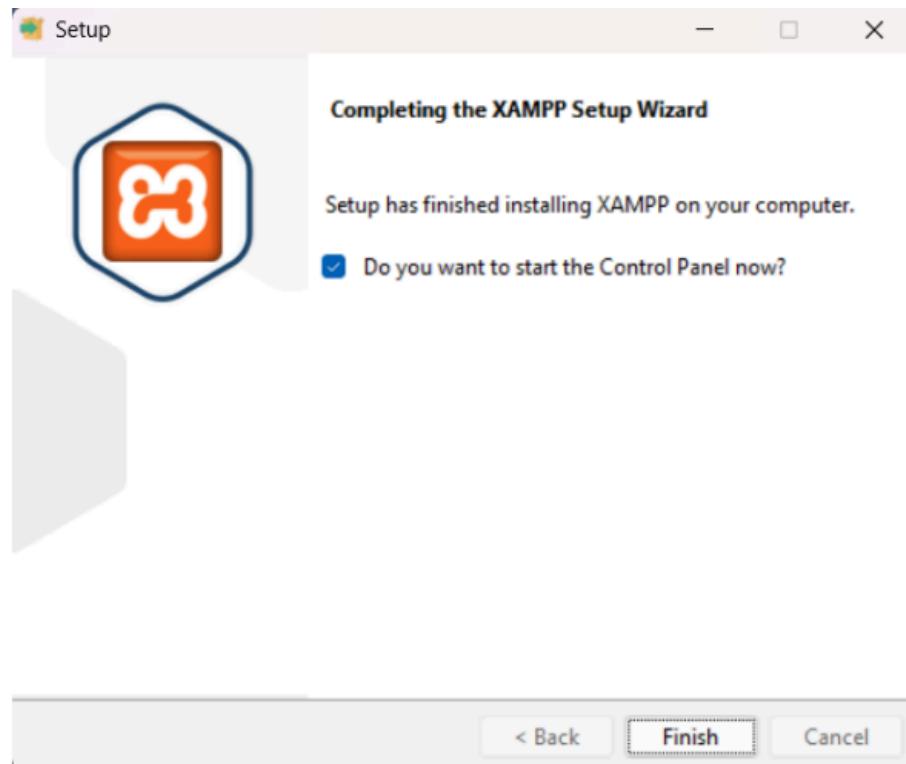
- 2) Open the setup file. Select all the required components and click next.



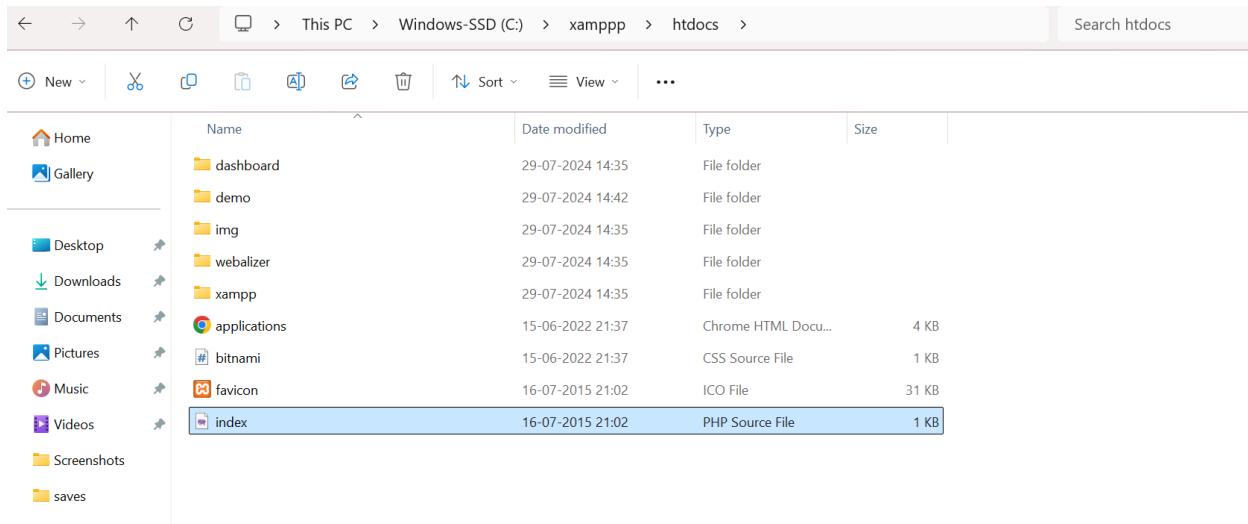
3) Select the language, click next. XAMPP starts to install.



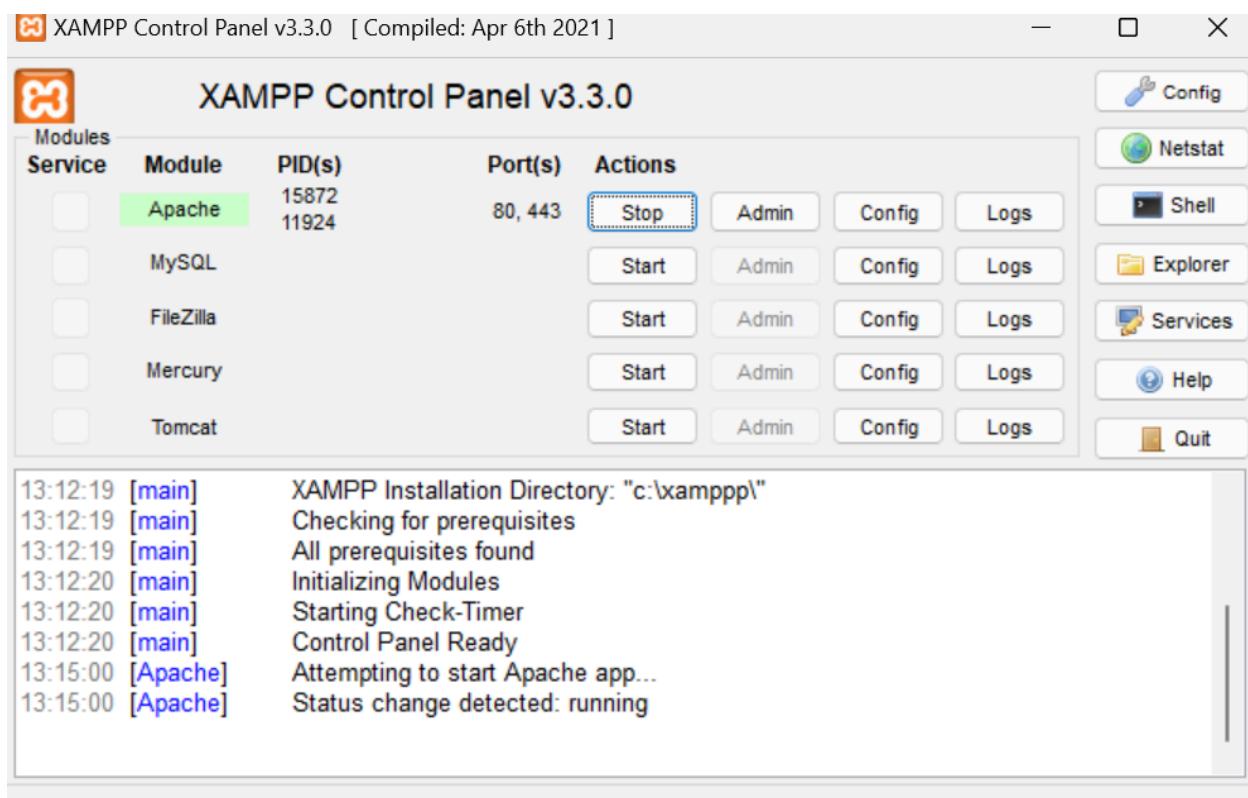
4) The installation is complete. Click Finish.



**Step 2:** Go to the directory where XAMPP was installed. Go to htdocs folder. Place your folder in this directory.



**Step 3:** Open XAMPP Control Panel, start the Apache service (Required) and mySQL service (if needed)

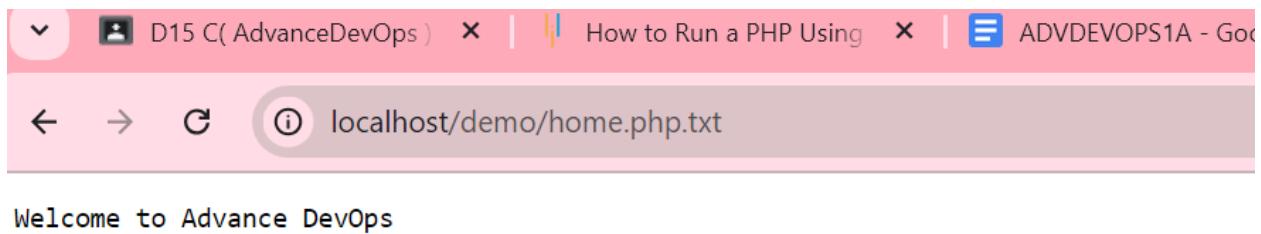


**Step 4:** Write a php file for your website.

```
File Edit View

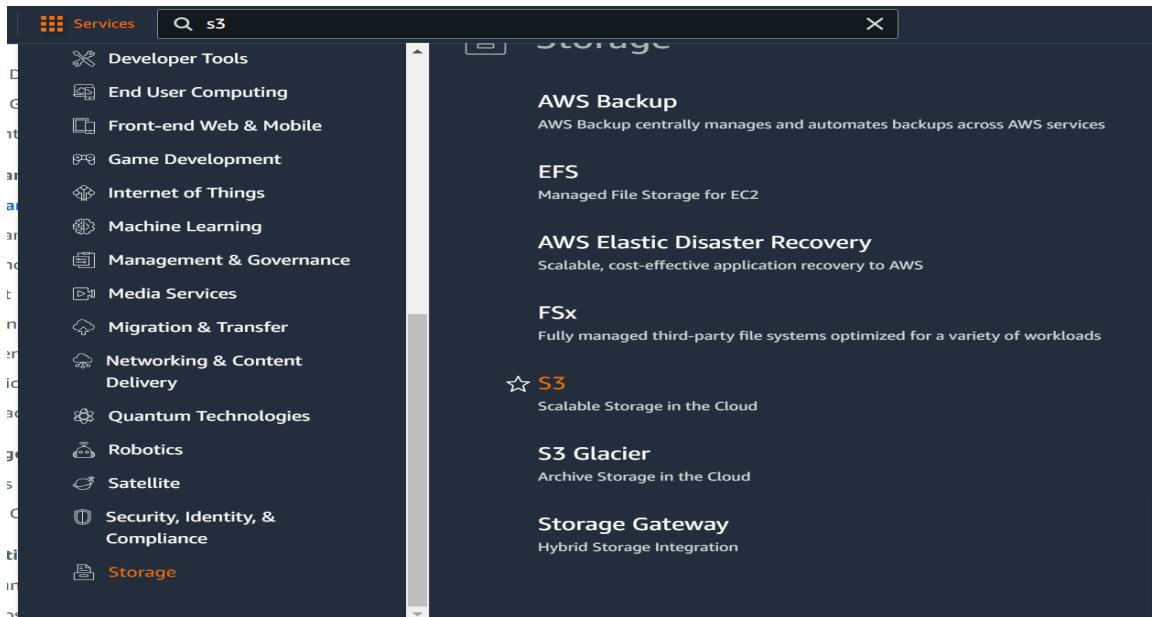
<?php
    echo "Welcome to Advance DevOps"
?>
```

**Step 5:** Open your web browser. Type localhost/YOUR\_FILENAME.php. This will open your website on your browser



## 2) AWS S3

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



**Step 2:** Click on Create Bucket. Give a name to your bucket, keeping other options default, scroll down and click on Create Bucket.

Amazon S3 > Buckets > Create bucket

### Create bucket Info

Buckets are containers for data stored in S3.

#### General configuration

AWS Region  
Asia Pacific (Sydney) ap-southeast-2

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

Copy settings from existing bucket - *optional*  
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

#### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**Default encryption** [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
 

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

**► Advanced settings**

**Info** After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

### Step 3: Go to the Objects tab and click on upload file.

[Services](#) [Search](#) [Alt+S] [Dashboard](#) [CloudWatch Metrics](#) [CloudWatch Logs](#) [AWS Lambda](#) [AWS Step Functions](#) [Amazon S3](#) [Buckets](#) [boomweb](#) Sydney bhumish

[boomweb](#) [Info](#)

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

**Objects (0) Info**

[Actions](#) [Create folder](#) [Upload](#)

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

< 1 > [Actions](#)

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

[Upload](#)

[Amazon S3](#) > [Buckets](#) > [boomweb](#) > [Upload](#)

**Upload** [Info](#)

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

Drag and drop files and folders you want to upload here, or choose [Add files](#) or [Add folder](#).

**Files and folders (0)**

All files and folders in this table will be uploaded.

Name	Folder	Type
No files or folders You have not chosen any files or folders to upload.		

**Step 4:** Click on Add files. Add all the files you want to upload. Then scroll down and click on Upload.

The screenshot shows the 'Upload' interface in the AWS S3 console. At the top, a table titled 'Files and folders (2 Total, 266.0 B)' lists two files: 'static.html.txt' in the 'staticweb/' folder and another 'static.html.txt' file. There are buttons for 'Remove', 'Add files', and 'Add folder'. Below the table is a search bar labeled 'Find by name' and navigation arrows. The main area has a header 'Destination' with a 'Info' link. Under 'Destination', it shows 's3://boomweb'. Below this, 'Destination details' and 'Bucket settings that impact new objects stored in the specified destination.' are listed. Further down are sections for 'Permissions' (Grant public access and access to other AWS accounts) and 'Properties' (Specify storage class, encryption settings, tags, and more). At the bottom right are 'Cancel' and 'Upload' buttons.

**Step 5:** This will take you to the Objects screen. Switch to Properties and scroll down to Static Website Hosting. There you would find the link (Bucket website endpoint) to your website.

The screenshot shows the 'Properties' tab of the 'boomweb' bucket's properties page. At the top, there's a breadcrumb trail: 'Amazon S3 > Buckets > boomweb'. Below that is the bucket name 'boomweb' with an 'Info' link. A navigation bar includes tabs for 'Objects', 'Properties' (which is selected), 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Bucket overview' section displays basic information: AWS Region (Asia Pacific (Sydney) ap-southeast-2), Amazon Resource Name (ARN) (arn:aws:s3:::boomweb), and Creation date (August 11, 2024, 22:53:06 (UTC+05:30)). The 'Bucket Versioning' section is present, with a note about versioning and a 'Edit' button. The 'Bucket Versioning' status is 'Disabled'. The 'Multi-factor authentication (MFA) delete' section indicates that MFA delete is required for changing Bucket Versioning settings and permanently deleting object versions. An additional note states: 'An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CloudTrail console.'

**Step 6:** Scroll down till you find Static website hosting, click on edit.

Static website hosting

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

Static website hosting

Disabled

Edit

**Step 7:** Enable static website hosting, in Index document, write the name of your document. Save your changes.

Edit static website hosting [Info](#)

Static website hosting

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

Static website hosting

Disable

Enable

Hosting type

Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

**For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#).**

Index document

Specify the home or default page of the website.

static.html

**Step 8:** Uncheck the Block all public access checkbox and click on save changes.

Edit Block public access (bucket settings) [Info](#)

Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets 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.

**Step 9:** Scroll down to bucket policy and click edit.

The screenshot shows the 'Edit bucket policy' page in the AWS Management Console. The policy document is as follows:

```
1▼ {
2  "Version": "2012-10-17",
3▼ "Statement": [
4▼   {
5    "Sid": "PublicReadGetObject",
6    "Effect": "Allow",
7▼     "Principal": {
8      "AWS": "*"
9    },
10   "Action": "s3:GetObject",
11   "Resource": "arn:aws:s3:::statichosting27/*"
12 }
13 ]
14 }
```

On the right side, there's a sidebar with options for editing statements, adding actions, and choosing services. The 'Included' section is expanded, showing 'S3' selected. Other available services listed are AMP, API Gateway, and API Gateway V2.

**Step 10:** You can access your website now.

The screenshot shows a web browser displaying the contents of the static website. The URL is `boomweb.s3.ap-southeast-2.amazonaws.com/static.html.txt`. The page content is:

My First Lab

My name is Bhumisha.  
Class: D15C  
Roll No: 38

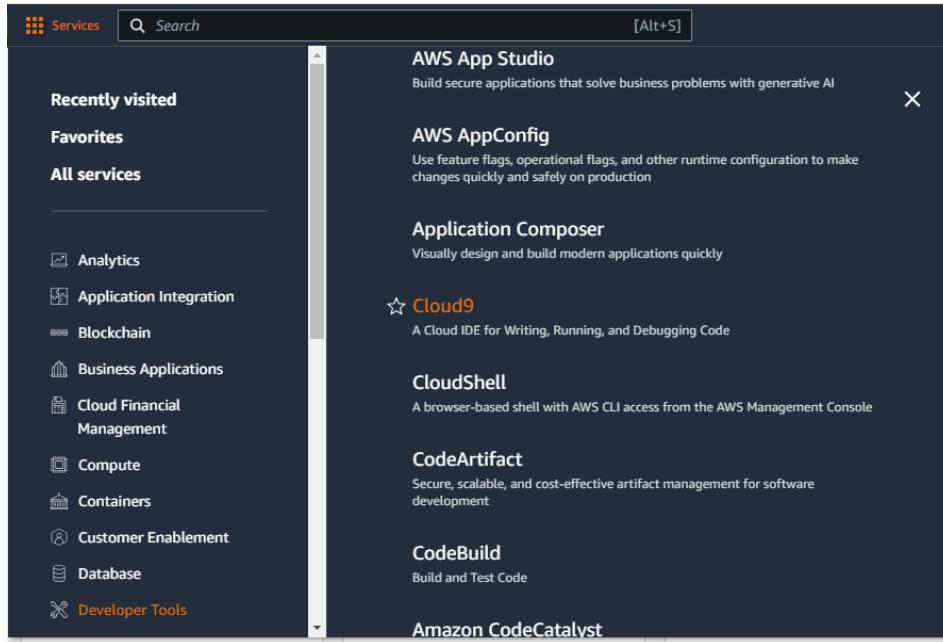
## EXPERIMENT 1(B)

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

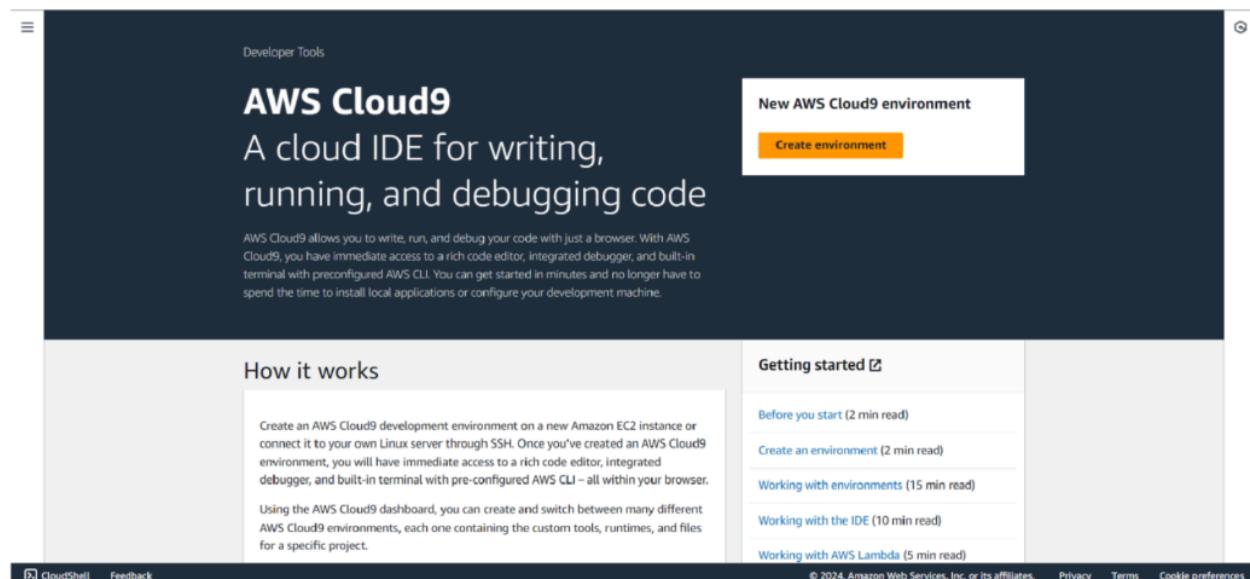
### Steps:

**Step 1:** Set up a Cloud9 environment.

1) Search Cloud9 in the services tab and open it



2) Click on Create Environment.



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

The screenshot shows the 'Create environment' page in the AWS Cloud9 interface. In the 'Details' section, the 'Name' field is filled with 'WebAppIDE'. The 'Description - optional' field is empty. Under 'Environment type', the 'New EC2 instance' option is selected, indicated by a blue border around its radio button. The 'Existing compute' option is unselected.

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

This screenshot shows the 'New EC2 instance' configuration section. Under 'Instance type', the 't2.micro (1 GB RAM + 1 vCPU)' option is selected, highlighted with a blue border. Other options shown are 't3.small (2 GB RAM + 2 vCPU)' and 'm5.large (8 GB RAM + 2 vCPU)'. A link 'Additional instance types' is also present. Under 'Platform Info', 'Amazon Linux 2023' is chosen. Under 'Timeout', '30 minutes' is set. The 'Network settings' section at the bottom is partially visible.

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

**Network settings**

**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**

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

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

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

## 6) The environment has been created.

Successfully created WebAppIDE. To get the most out of your environment, see [Best practices for using AWS Cloud9](#)

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

[AWS Cloud9](#) > Environments

Environments (1)					
	Name	Cloud9 IDE	Environment type	Connection	Permissions
<input type="radio"/>	WebAppIDE	<a href="#">Open</a>	EC2 instance	Secure	Owner

## 7) Search IAM on the services search bar and open it. Click on Create User. Give a username to your user and click Next.

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
Set permissions

Step 3  
Review and create

**Specify user details**

**User details**

User name

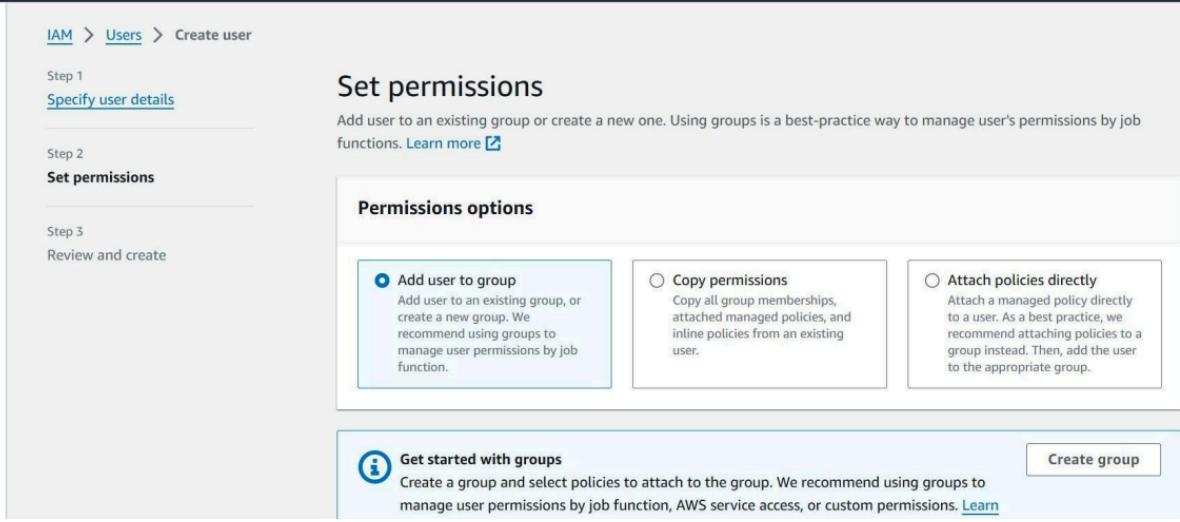
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

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

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

Cancel **Next**

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

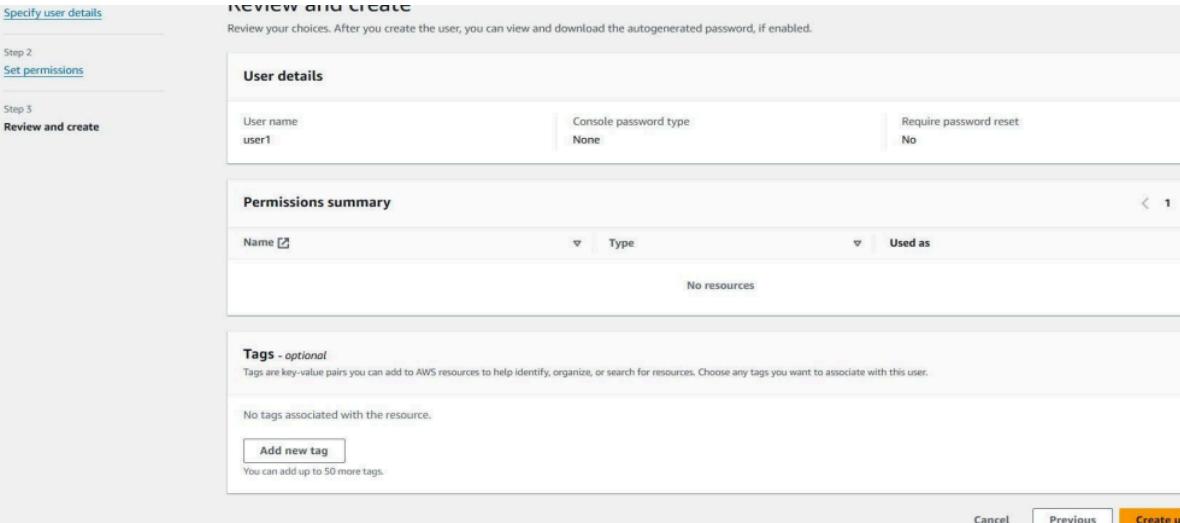


**Create user group**

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

**User group name**  
Enter a meaningful name to identify this group.  
  
Maximum 128 characters. Use alphanumeric and '+,-,@,\_' characters.

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



10) Go to permissions and click on Add permissions. Click on Attach Policies. Search for AWSCloud9EnvironmentMember, select it and click on Attach policies.

The screenshot shows the AWS IAM User Groups page. On the left, a sidebar lists navigation options: Dashboard, Access management, User groups (which is selected), Users, Roles, Policies, Identity providers, Account settings, and Access reports. The main content area displays the 'group1' user group details. The 'Summary' section shows the User group name as 'group1', Creation time as 'August 04, 2024, 16:59 (UTC+05:30)', and ARN as 'arn:aws:iam::010928206130:group/group1'. Below this, there are tabs for 'Users', 'Permissions' (which is selected), and 'Access Advisor'. The 'Permissions policies' section shows 0 managed policies attached to the group. A note states 'You can attach up to 10 managed policies.' Below this, a table lists other permission policies available for attachment, filtered by type ('All types') and search term ('AWSCloud9'). The table includes columns for Policy name, Type, Used as, and Description. One policy, 'AWSCloud9EnvironmentMember', is checked and highlighted in blue.

Policy name	Type	Used as	Description
AWSCloud9Administ...	AWS managed	None	Provides administrator access to AWS ...
<b>AWSCloud9Environ...</b>	AWS managed	None	Provides the ability to be invited into A...
AWSCloud9SSMInsta...	AWS managed	None	This policy will be used to attach a rol...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

## Experiment No. 2

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

### Step1:- Creation of role:-

1. Login to your AWS account and search for IAM

The screenshot shows the AWS search interface with the query 'iam' entered. The search results are categorized into 'Services' and 'Features'. Under 'Services', the 'IAM' card is highlighted, showing its purpose: 'Manage access to AWS resources'. Other services listed include IAM Identity Center, Resource Access Manager, and AWS App Mesh. Under 'Features', there are 24 results, but they are not fully visible. The top right corner shows the user's name 'bhumishap' and the region 'Sydney'.

2. Then go into the role section and click on create role.

The screenshot shows the AWS IAM service dashboard. On the left, there are two notifications: one about adding MFA to the root user and another about the root user having no active access keys. Below these are sections for 'IAM resources' (User groups: 0, Users: 0, Roles: 11, Policies: 7, Identity providers: 0) and 'What's new' (mentioning AWS IAM Access Analyzer). On the right, there are 'Quick Links' to 'My security credentials' and 'Tools' (Policy simulator). The top right corner shows the user's name 'bhumishap' and the region 'Global'.

Roles (11) <a href="#">Info</a>		<a href="#">C</a>	Delete	<a href="#">Create role</a>
<input type="text"/> Search		<a href="#">&lt;</a> <a href="#">1</a> <a href="#">&gt;</a> <a href="#">⚙️</a>		
<input type="checkbox"/>	Role name	Trusted entities		
<input type="checkbox"/>	<a href="#">AWSCodePipelineServiceRole-ap-southeast-2-booompipeline</a>	AWS Service: codepipeline		
<input type="checkbox"/>	<a href="#">AWSCodePipelineServiceRole-ap-southeast-2-FirstPipeline</a>	AWS Service: codepipeline		
<input type="checkbox"/>	<a href="#">AWSCodePipelineServiceRole-ap-southeast-2-FirtsPipeline</a>	AWS Service: codepipeline		
<input type="checkbox"/>	<a href="#">AWSCodePipelineServiceRole-ap-southeast-2-LabPipeline</a>	AWS Service: codepipeline		
<input type="checkbox"/>	<a href="#">AWSServiceRoleForAWSCloud9</a>	AWS Service: cloud9 (Service-Linked)		
<input type="checkbox"/>	<a href="#">AWSServiceRoleForCostOptimizationHub</a>	AWS Service: cost-optimization-hub.		
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked)		
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linked)		

3. Then select a trusted entity as AWS service.

### Select trusted entity [Info](#)

**Trusted entity type**

- AWS service  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

#### 4. Select use case as EC2.

Service or use case

Choose a use case for the specified service.

Use case

- EC2**  
Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**  
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**  
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**  
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**  
Allows EC2 Scheduled Instances to manage instances on your behalf.

**Cancel** **Next**

#### 5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.

**Permissions policies (2/953)** [Info](#) [C](#)

Choose one or more policies to attach to your new role.

Filter by Type

X  ▼ 2 matches < 1 > ⟳

<input checked="" type="checkbox"/>	Policy name <a href="#">X</a>	Type	Description
<input checked="" type="checkbox"/>	<a href="#"> AWSElasticBeansta...</a>	AWS managed	Provide the instances in your web server ...
<input checked="" type="checkbox"/>	<a href="#"> AWSElasticBeansta...</a>	AWS managed	Provide the instances in your worker envi...

▶ Set permissions boundary - *optional*

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

6. Give a name to Role. Here I have given my role name as aws -elasticbeanstalk -ec2 role.

Step 3 of 3

## Name, review, and create

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**aws-elasticbeanstalk-ec2-role**

Maximum 64 characters. Use alphanumeric and '+,-,@-\_ characters.

**Description**  
Add a short explanation for this role.  
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+=,. @-/\[\]!#\$%^&();;"`

**Step 1: Select trusted entities** **Edit**

7. Then the role gets created.

Services Search [Alt+S] Global ▾ bhumishap ▾

IAM > Roles > aws-elasticbeanstalk-ec2-role

### aws-elasticbeanstalk-ec2-role Info

Allows EC2 instances to call AWS services on your behalf. Delete

**Summary** Edit

Creation date August 16, 2024, 23:18 (UTC+05:30)	ARN <a href="#">arn:aws:iam::010928192223:role/aws-elasticbeanstalk-ec2-role</a>	Instance profile ARN <a href="#">arn:aws:iam::010928192223:instance-profile/aws-elasticbeanstalk-ec2-role</a>
Last activity -	Maximum session duration 1 hour	

**Permissions** Trust relationships Tags Access Advisor Revoke sessions

**Permissions policies (2) Info** Add permissions ▾

You can attach up to 10 managed policies.

Filter by Type All types

G Simulate Remove

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

## Step 2: Search for Elastic Beanstalk in the searchbar near services

The screenshot shows the AWS Management Console search results for 'elastic beanstalk'. The search bar at the top contains the query 'elastic beanstalk'. Below the search bar, there is a sidebar with 'Recent' and 'Services (13)' sections, including links for S3, Elasticsearch, CloudWatch, and AWS Lambda. The main search results are displayed under the 'Services' heading, with a link to 'See all 13 results'. The first result is 'Elastic Beanstalk' with a star icon, described as 'Run and Manage Web Apps'. Below it are 'Top features' for 'Applications' and 'Environments'. The second result is 'Elastic Transcoder' with a star icon, described as 'Easy-to-Use Scalable Media Transcoding'. The third result is 'Elastic Container Service' with a star icon, described as 'Highly secure, reliable, and scalable way to run containers'. The fourth result is 'Elastic Container Registry' with a star icon, described as 'Fully-managed Docker container registry : Share and deploy container software, publ...'. There is also a 'Features' section with a link to 'See all 29 results'.

## Step 3: Go to Elastic Beanstalk and click on Create Application

The screenshot shows the Amazon Elastic Beanstalk landing page. The top navigation bar includes 'aws', 'Services', a search bar, and account information for 'Sydney' and 'bhumishap'. The main heading is 'Amazon Elastic Beanstalk' with the subtext 'End-to-end web application management.' Below the heading is a brief description: 'Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.' To the right, there is a 'Get started' box with the subtext 'Easily deploy your web application in minutes.' and a prominent orange 'Create application' button. Another box below is titled 'Pricing' with the subtext 'There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.' At the bottom, there are links for 'CloudShell', 'Feedback', '© 2024 Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

**Step 4:** Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to a single instance. Click on NEXT

The screenshot shows the 'Configure environment' step of the Amazon Elastic Beanstalk setup wizard. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Configure environment', which is currently active. Step 2 is 'Configure service access', Step 3 is 'optional' networking, Step 4 is 'optional' instance traffic scaling, Step 5 is 'optional' updates and monitoring, and Step 6 is 'Review'. The main content area has two sections: 'Environment tier' and 'Application information'. Under 'Environment tier', 'Web server environment' is selected. Under 'Application information', the application name is set to 'boomweb'. In the bottom section, 'Platform type' is set to 'Managed platform' (selected), and the 'Platform' dropdown is set to 'PHP'. The 'Platform branch' dropdown is set to 'PHP 8.3 running on 64bit Amazon Linux 2023'. The 'Platform version' dropdown is set to '4.3.1 (Recommended)'.

Services Search [Alt+S] Sydney bhumishap

Configure environment [Info](#)

**Environment tier [Info](#)**  
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

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

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

**Application information [Info](#)**

Application name  
boomweb  
Maximum length of 100 characters.

▶ Application tags (optional)

**Platform [Info](#)**

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

**Custom platform**  
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform  
PHP

Platform branch  
PHP 8.3 running on 64bit Amazon Linux 2023

Platform version  
4.3.1 (Recommended)

**Step 5:** Use an existing service role and choose whatever service role is available on your account.

**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.

aws-elasticbeanstalk-ec2-role ▼

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

ec2 ▼

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role ▼

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) Next

**Step 6:** Review the settings that you have set up for your application and submit your application.

**Step 1: Configure environment** Edit

**Environment information**

Environment tier	Application name
Web server environment	boomweb
Environment name	Application code
Boomweb-env	Sample application
Platform	
arn:aws:elasticbeanstalk:ap-southeast-2::platform/PHP	
8.3 running on 64bit Amazon Linux 2023/4.3.2	

**Step 2: Configure service access**

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

Service role <code>arn:aws:iam::010928192223:role/aws-elasticbeanstalk-ec2-role</code>	EC2 key pair <code>ec2</code>	EC2 instance profile <code>aws-elasticbeanstalk-ec2-role</code>
---	----------------------------------	--

**Step 3: Set up networking, database, and tags**

**Networking, database, and tags Info**  
Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

No options configured

**Tags**

**Platform software**

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

X-Ray enabled  
Deactivated

**Environment properties**

Key	Value

Environment successfully launched.

Elastic Beanstalk > Environments

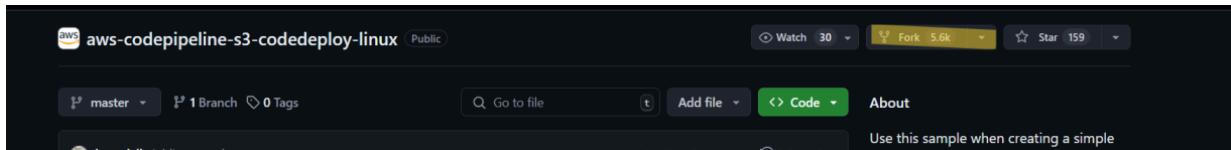
**Environments (1) Info**

**Create environment**

Environment name	Health	Applica...	Platform	Domain
<a href="#">Boomweb-env</a>	<span style="color: orange;">Warning</span>	<a href="#">boomweb</a>	PHP 8.3 r...	<a href="#">Boomweb-env.eba-5qbmwj7...</a>

**Step 7:** Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

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



**Step 8:** Click on Create Pipeline.

The screenshot shows the AWS Management Console interface. On the left, there's a sidebar with a 'Recently visited' section and a 'Services' list containing various AWS services such as Analytics, Application Integration, Blockchain, Business Applications, Cloud Financial Management, Compute, Containers, Customer Enablement, Database, Developer Tools, and End User Computing. The 'CodePipeline' service is highlighted. The main content area displays the 'CodePipeline' service details, including its description: 'Release Software using Continuous Delivery'. A prominent orange 'Create environment' button is located in the top right of this section. The URL in the browser bar is 'Developer Tools > CodePipeline > Pipelines'.

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

The screenshot shows the 'Choose pipeline settings' step of creating a new pipeline. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Pipeline settings'. It has a 'Pipeline name' field containing 'bhumisha' and a note that it cannot be edited after creation. Below it is a 'Pipeline type' section with a note about V1 pipelines being deprecated. Under 'Execution mode', the 'Queued (Pipeline type V2 required)' option is selected, with a note that executions are processed one by one.

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

The screenshot shows the 'Add source stage' step of creating a new pipeline. The sidebar shows Step 2 of 5. The main area is titled 'Source' and shows a 'Source provider' dropdown set to 'GitHub (Version 2)'. Below it is a note about GitHub version 2 actions. The 'Connection' section shows a search bar with 'arn:aws:codeconnections:ap-southeast-2:010928192223:connection/ab38d1' and a 'Connect to GitHub' button. At the bottom, a green box indicates 'Ready to connect'.

**Step 11:** Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

Repository name  
Choose a repository in your GitHub account.  
bhumishap/aws-codepipeline-s3-codedeploy-linux-2.0

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

Output artifact format  
Choose the output artifact format.

**CodePipeline default**  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

**Full clone**  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

**Step 12:** Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

**Trigger**

Trigger type  
Choose the trigger type that starts your pipeline.

**No filter**  
Starts your pipeline on any push and clones the HEAD.

**Specify filter**  
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

**Do not detect changes**  
Don't automatically trigger the pipeline.

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

Cancel Previous Next

**Step 13:** Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

ch [Alt+S] Sydney bhumishap

**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**

Asia Pacific (Sydney) ▾

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

SourceArtifact ▾  
No more than 100 characters

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

Q boomweb X

**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.

Q Boomweb-env X

Configure automatic rollback on stage failure

**Step 14:** Check all the information and click on create Pipeline.

**Review** [Info](#)

Step 5 of 5

**Step 1: Choose pipeline settings**

**Pipeline settings**

**Pipeline name**  
bhumisha

**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-ap-southeast-2-bhumisha

[Alt+S] Sydney | bhumishap | ⓘ ⌂ ⌃ ⌄ ⌅

### Step 2: Add source stage

Source action provider

Source action provider  
GitHub (Version 2)  
OutputArtifactFormat  
CODE\_ZIP  
DetectChanges  
false  
ConnectionArn  
arn:aws:codeconnections:ap-southeast-2:010928192223:connection/864c682d-86c9-45fd-9c50-12318b35e13e  
FullRepositoryId  
bhumishap/aws-codepipeline-s3-codeddeploy-linux-2.0  
Default branch  
master

### Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type  
No filter

### Step 3: Add build stage

Build action provider

Build stage  
No build

### Step 4: Add deploy stage

Deploy action provider

Deploy action provider  
AWS Elastic Beanstalk  
ApplicationName  
boomweb  
EnvironmentName  
Boomweb-env  
Configure automatic rollback on stage failure  
Disabled

Cancel Previous Create pipeline

**Step 15:** If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy.

The screenshot shows the AWS CodePipeline console with a successful pipeline execution. The pipeline is named "bhumisha" and is of type V2, currently in the QUEUED state. The execution ID is 14411bd1-9a5a-49e8-99de-54a52adae29b. The pipeline consists of two stages: Source and Deploy. The Source stage is succeeded, with a GitHub commit (commit ID 8fd5da54) from 2 minutes ago. The Deploy stage is also succeeded, with an AWS Elastic Beanstalk deployment from 1 minute ago. Both stages have green checkmarks indicating success. A "Release change" button is visible at the top right.

Developer Tools > CodePipeline > Pipelines > bhumisha

bhumisha

Notify ▾ Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: 14411bd1-9a5a-49e8-99de-54a52adae29b

Source GitHub (Version 2) Succeeded - 2 minutes ago 8fd5da54 View details

8fd5da54 Source: Update README.md

Disable transition

Services Search [Alt+S] Sydney bhumishap

Source • CodeCommit

Deploy • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline History Settings

to resource

Succeeded - 2 minutes ago 8fd5da54 View details

8fd5da54 Source: Update README.md

Disable transition

Deploy Succeeded Pipeline execution ID: 14411bd1-9a5a-49e8-99de-54a52adae29b Start rollback

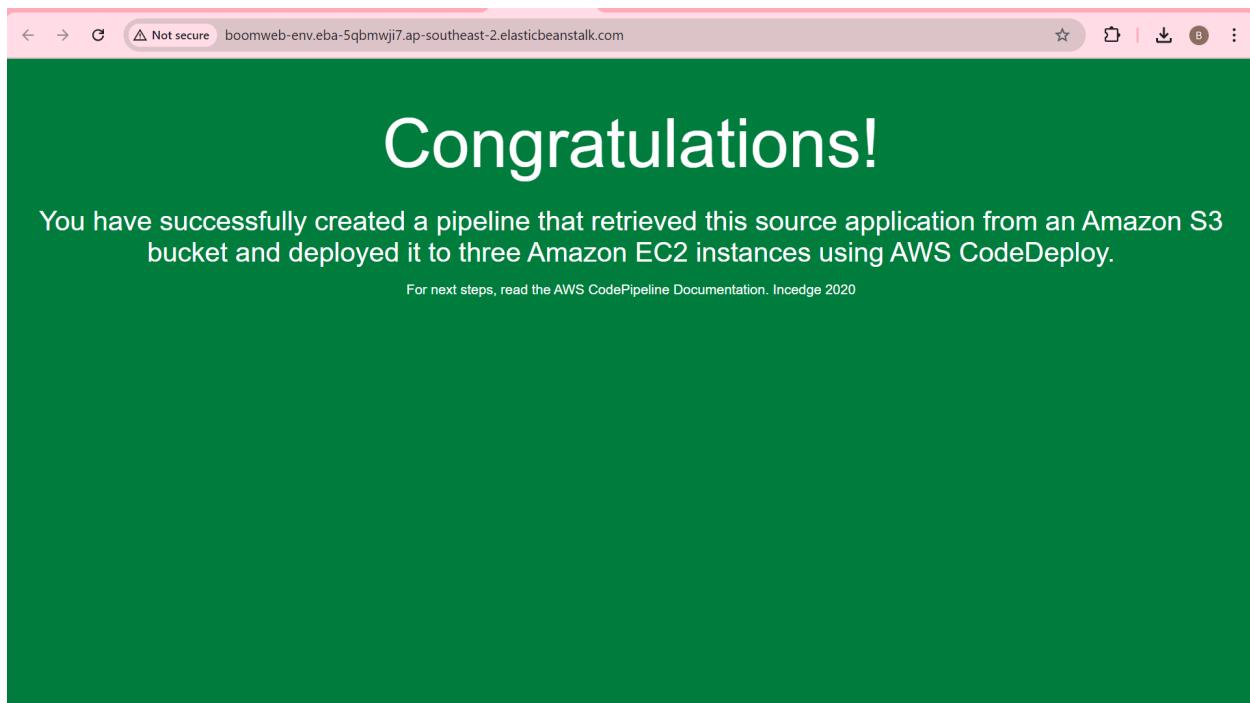
Deploy AWS Elastic Beanstalk Succeeded - 1 minute ago View details

8fd5da54 Source: Update README.md

**Step 16:** In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Beanstalk.

The screenshot shows the 'Environment overview' section of the AWS Elastic Beanstalk console. At the top, there's a breadcrumb navigation: 'Elastic Beanstalk > Environments > Boomweb-env'. Below it, the environment name 'Boomweb-env' is displayed with a blue 'Info' link. On the right, there are three buttons: a grey 'Actions' button with a dropdown arrow, an orange 'Upload and deploy' button, and a small circular icon with a question mark. The main area is titled 'Environment overview' and contains two columns of information. The left column includes 'Health' (with a warning icon and a 'View causes' link), 'Domain' (with the URL 'boomweb-env.eba-5qbmwji7.ap-southeast-2.elasticbeanstalk.com'), and a 'Platform' section. The right column includes 'Environment ID' (e-xt32eyinty), 'Application name' (boomweb), and a 'Change version' button. The entire interface has a light grey background with white and light blue text.

**Step 17:** This will successfully show the sample website hosted.



If you can see this, that means that you successfully created an automated software using CodePipeline.

## EXPERIMENT NO.3

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

Prerequisites :

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

Security group name	Security group ID	Description	VPC ID			
<a href="#">master</a>	<a href="#">sg-0ade163eb26ea436f</a>	<a href="#">Security group for master node</a>	<a href="#">vpc-0340ce013f393caf4</a>			
Owner	Inbound rules count	Outbound rules count				
<a href="#">010928192223</a>	8 Permission entries	1 Permission entry				
<hr/>						
<a href="#">Inbound rules</a> <a href="#">Outbound rules</a> <a href="#">Tags</a>						
<hr/>						
Inbound rules (8)						
<div style="display: flex; justify-content: space-between;"><div style="flex-grow: 1;"><input placeholder="Search" type="text"/></div><div><a href="#"><span>C</span></a> <a href="#">Manage tags</a> <a href="#">Edit inbound rules</a></div></div>						
<div style="display: flex; justify-content: space-between;"><div style="flex-grow: 1;"></div><div>&lt; 1 &gt; <span>⑤</span></div></div>						
<hr/>						
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0c2cb18dc157e06d6	IPv4	All TCP	TCP	0 - 65535
<input type="checkbox"/>	-	sgr-0c7aaa1be99c68fc0	IPv4	Custom TCP	TCP	10252
<input type="checkbox"/>	-	sgr-0f9940970a2c989e3	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-00f0537f09dd487c6	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-085af61d28e0ddd63	IPv4	Custom TCP	TCP	10251
<input type="checkbox"/>	-	sgr-041783d2b62755...	IPv4	All traffic	All	All
<input type="checkbox"/>	-	sgr-0491cabab6c1209cd9	IPv4	Custom TCP	TCP	6443
<input type="checkbox"/>	-	sgr-0a836004cb806ba...	IPv4	Custom TCP	TCP	10250

**Details**

Security group name node	Security group ID sg-0a0bac122aeec771f	Description Security Group for nodes	VPC ID <a href="#">vpc-0340ce013f393caf4</a>
Owner 010928192223	Inbound rules count 6 Permission entries	Outbound rules count 1 Permission entry	

**Inbound rules**   **Outbound rules**   **Tags**

**Inbound rules (6)**

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-02d17fd0ee1866a45	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0248b9fed2f393d80	IPv4	All traffic	All	All
<input type="checkbox"/>	-	sgr-09675ccdc361cd771	IPv4	Custom TCP	TCP	10250
<input type="checkbox"/>	-	sgr-0cd926e18bbb6...	IPv4	All TCP	TCP	0 - 65535
<input type="checkbox"/>	-	sgr-0e7072158495ca4...	IPv4	Custom TCP	TCP	30000 - 32767
<input type="checkbox"/>	-	sgr-0fb8f697c388fc27b	IPv4	HTTP	TCP	80

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

Master:

**Launch an instance** [Info](#)

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

**Name and tags** [Info](#)

Name  
master

**Application and OS Images (Amazon Machine Image)** [Info](#)

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

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

**Recents**   **Quick Start**

Amazon Linux   macOS   Ubuntu   Windows   Red Hat   SUSE Li

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd6... [read more](#)  
ami-0892a9c01908fafd1

Virtual server type (instance type)  
t2.medium

Firewall (security group)  
master

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

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

**Cancel**   **Launch instance**   [Review commands](#)

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-0892a9c01908fafd1 (64-bit (x86)) / ami-08a8dfbf1c5db5344 (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-0892a9c01908fafd1  
Username  
ubuntu  
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: 0.0584 USD per Hour  
On-Demand RHEL base pricing: 0.0872 USD per Hour  
On-Demand Windows base pricing: 0.0764 USD per Hour  
On-Demand SUSE base pricing: 0.1584 USD per Hour

All generations  
[Compare instance types](#)

**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  
ec2\_keypair [Create new key pair](#)

**Network settings** [Info](#)

Network [Info](#)  
vpc-0340ce013f393caf4

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

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

Additional charges apply when outside of free tier allowance

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

Create security group  Select existing security group

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

master sg-0ade163eb26ea436f X  
VPC: vpc-0340ce013f393caf4

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd6... [read more](#)  
ami-0892a9c01908fafd1

Virtual server type (instance type)  
t2.medium

Firewall (security group)  
master

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

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

[Launch instance](#) [Review commands](#)

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd6... [read more](#)  
ami-0892a9c01908fafd1

Virtual server type (instance type)  
t2.medium

Firewall (security group)  
master

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

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

[Launch instance](#) [Review commands](#)

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

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)  
vpc-0340ce013f393caf4

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

Auto-assign public IP | [Info](#)  
Enable  
Additional charges apply when outside of [free tier allowance](#)

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

Create security group  Select existing security group

Common security groups [Info](#)

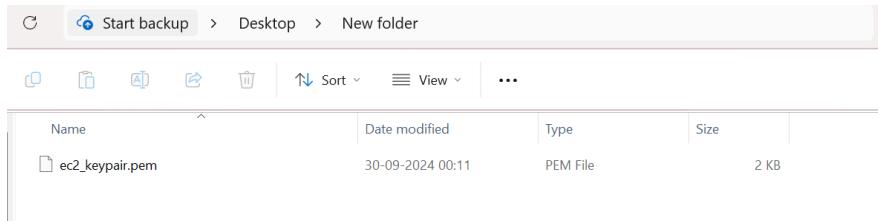
Select security groups ▾

node sg-0a0bac122aeec771f X  
VPC: vpc-0340ce013f393caf4

Compare security group rules

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

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



3. Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored. Then execute the ssh command.  
For example: ssh -i "ec2\_keypair.pem"  
<ssh://ubuntu@ec2-13-236-178-199.ap-southeast-2.compute.amazonaws.com>

Master:

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

Connect to your instance i-01ae5c4b1fe2ba178 (master) using any of these options

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

Instance ID  
i-01ae5c4b1fe2ba178 (master)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ec2\_keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
4. Connect to your instance using its Public DNS:

Example:

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.

```
PS C:\Users\bhumii> cd "C:\Users\bhumii\OneDrive\Desktop\New folder"
PS C:\Users\bhumii\OneDrive\Desktop\New folder> ssh -i "ec2_keypair.pem" ubuntu@ec2-13-236-178-199.ap-southeast-2.compute.amazonaws.com
The authenticity of host 'ec2-13-236-178-199.ap-southeast-2.compute.amazonaws.com (13.236.178.199)' can't be established.
ED25519 key fingerprint is SHA256:wumJngKU46dujGf/vlyRzXRAKpxR+AG0DmZRXBW4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-236-178-199.ap-southeast-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 29 18:53:56 UTC 2024

 System load:  0.08      Processes:          115
 Usage of /:   22.8% of 6.71GB  Users logged in:     0
 Memory usage: 6%           IPv4 address for enX0: 172.31.7.184
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

To run a command as administrator (user "root"), use "sudo <command>".
```

## Node 1:

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

Instance ID  
i-02f8edf3b90b9f4e2 (workernode-1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ec2\_keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
4. Connect to your instance using its Public DNS:

Example:

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

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

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

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

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

## Node 2:

Instance ID

i-0630b310934847c73 (workernode-2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ec2\_keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "ec2\_keypair.pem"
4. Connect to your instance using its Public DNS:  
 ec2-3-25-239-89.ap-southeast-2.compute.amazonaws.com

Example:

ssh -i "ec2\_keypair.pem" ubuntu@ec2-3-25-239-89.ap-southeast-2.compute.amazonaws.com

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

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

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

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

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

Thus the connection is successful.

4. Run on Master,Node 1, and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.  
curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo apt-key add -  
curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo tee

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

```
ubuntu@ip-172-31-7-184:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
-----END PGP PUBLIC KEY BLOCK-----
-bash: /etc/apt/trusted.gpg.d/docker.gpg: No such file or directory
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 https://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [15.3 kB]
Get:8 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:9 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:15 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:16 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:17 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8652 B]
Get:18 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [377 kB]
Get:19 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [156 kB]
Get:20 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:21 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.8 kB]
Get:22 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:23 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:24 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 kB]
Get:25 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:26 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:27 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:28 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:29 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:30 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:31 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:32 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:33 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
```

```
Get:34 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [1104 B]
Get:35 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:36 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:37 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:38 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [32.9 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 29.1 MB in 5s (5748 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

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

```

ubuntu@ip-172-31-7-184:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-southeast-2.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: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  pigz slirp4netns
Suggested packages:
  aufs-tools cgroups-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libltdl7
  libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 143 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
Get:6 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-buildx-plugin amd64 0.17.1-1~ubuntu.24.04-noble [30.3 MB]
Get:7 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-cli amd64 5.27.3.1-1~ubuntu.24.04-noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce amd64 5:27.3.1-1~ubuntu.24.04-noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-rootless-extras amd64 5:27.3.1-1~ubuntu.24.04-noble [9588 kB]
Get:10 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-compose-plugin amd64 2.29.7-1~ubuntu.24.04-noble [12.7 MB]
Fetched 123 MB in 2s (75.3 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_0.17.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.17.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.7-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...

```

```

Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.17.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

```

```

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

```

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

```
ubuntu@ip-172-31-7-184:~$ sudo mkdir -p /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
"exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
"exec-opts": ["native.cgroupdriver=systemd"]  
}
```

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

```
ubuntu@ip-172-31-7-184:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

5. Run the below command to install Kubernets.

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

```
ubuntu@ip-172-31-7-184:~$ sudo mkdir -p /etc/apt/keyrings  
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

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

```
deb [signed by /etc/apt/keyrings/kubernetes-keyring.gpg] https://pkgs.k8s.io/core/stable/v1.31/deb/ /  
ubuntu@ip-172-31-7-184:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease  
Get:5 https://prod-cdn.packages.k8s.io/repositories/istv:/kubernetes/:core:/stable:/v1.31/deb InRelease [1186 B]  
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:7 https://prod-cdn.packages.k8s.io/repositories/istv:/kubernetes/:core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 6051 B in 1s (10.9 kB/s)  
Reading package lists... Done  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  conntrack cri-tools kubernetes-cni  
The following NEW packages will be installed:  
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni  
0 upgraded, 6 newly installed, 0 to remove and 143 not upgraded.  
Need to get 87.4 MB of archives.  
After this operation, 311 MB of additional disk space will be used.  
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]  
Get:2 https://prod-cdn.packages.k8s.io/repositories/istv:/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/istv:/kubernetes/:core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]  
Get:4 https://prod-cdn.packages.k8s.io/repositories/istv:/kubernetes/:core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]  
Get:5 https://prod-cdn.packages.k8s.io/repositories/istv:/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/istv:/kubernetes/:core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
```

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

```

ubuntu@ip-172-31-7-184:~$ sudo systemctl enable --now kubelet
sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 143 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://ap-southeast-2.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 (61.8 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)

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.

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-172-31-7-184:~$ 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

```

```

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

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

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

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

```

```

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

```

```

ubuntu@ip-172-31-7-184:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-09-29 19:01:59 UTC; 268ms ago
     Docs: https://containerd.io
 Main PID: 4582 (containerd)
    Tasks: 7
      Memory: 13.3M (peak: 13.7M)
        CPU: 77ms
       CGroup: /system.slice/containerd.service
               └─4582 /usr/bin/containerd

Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453685323Z" level=info msg="Start subscribi>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453745481Z" level=info msg="Start recoverin>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453808448Z" level=info msg="Start event mon>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453816198Z" level=info msg=serving... addre>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453830731Z" level=info msg="Start snapshots>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453844510Z" level=info msg="Start cni netwo>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453854312Z" level=info msg="Start streaming>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453861743Z" level=info msg=serving... addre>
Sep 29 19:01:59 ip-172-31-7-184 containerd[4582]: time="2024-09-29T19:01:59.453945451Z" level=info msg="containerd succ>
Sep 29 19:01:59 ip-172-31-7-184 systemd[1]: Started containerd.service - containerd container runtime.

```

```

sudo apt-get install -y socat

```

```

ubuntu@ip-172-31-7-184:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 143 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (22.0 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

```

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

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

```

ubuntu@ip-172-31-7-184:~$ 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"
W0929 19:02:58.714702   4783 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-172-31-7-184 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.
[local] and IPs [10.96.0.1 172.31.7.184]
[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-172-31-7-184 localhost] and IPs [172.31.7.184 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-7-184 localhost] and IPs [172.31.7.184 127.0.0.1 ::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"

```

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 172.31.7.184:6443 --token asa7n9.0mkjob1gsfy3xuzy \
    --discovery-token-ca-cert-hash sha256:569daba7cee31b6f3c954325f206ce87c8d3fa2fa739e5f66641dde8d1bf13c
    --node-ip 172.31.7.184 --add-insecure-peer
```

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

```
mkdir -p $HOME/.kube
```

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

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

```
ubuntu@ip-172-31-7-184:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-7-184:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-7-184:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

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

```
ubuntu@ip-172-31-7-184:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE   VERSION
ip-172-31-7-184  NotReady control-plane  98s   v1.31.1
```

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

```
sudo kubeadm join 172.31.7.184:6443 --token asa7n9.0mkjob1gsfy3xuzy \
```

```
--discovery-token-ca-cert-hash
```

```
sha256:569daba7cee31b6f3c954325f206ce87c8d3fa2fa739e5f66641dde8d1bf13c
```

Node 1:

```
ubuntu@ip-172-31-3-88:~$ sudo kubeadm join 172.31.7.184:6443 --token asa7n9.0mkjob1gsfy3xuzy \
--discovery-token-ca-cert-hash sha256:569daba7cee31b6f3c954325f206ce87c8d3fa2fa739e5f66641dde8d1bf13c
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FVI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 500.662824ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

Node 2:

```
ubuntu@ip-172-31-7-177:~$ sudo kubeadm join 172.31.7.184:6443 --token asa7n9.0mkjob1gsfy3xuzy \
--discovery-token-ca-cert-hash sha256:569daba7cee31b6f3c954325f206ce87c8d3fa2fa739e5f66641dde8d1bf13c
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FVI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.00113109s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

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

9. Now Run the command kubectl get nodes to see the nodes after executing Join command on nodes.

```
ubuntu@ip-172-31-7-184:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE   VERSION
ip-172-31-3-88  NotReady <none>     44s   v1.31.1
ip-172-31-7-177  NotReady <none>     17s   v1.31.1
ip-172-31-7-184  NotReady control-plane 4m42s  v1.31.1
```

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

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

```
ubuntu@ip-172-31-7-184:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
```

```
sudo systemctl status kubelet
```

```
ubuntu@ip-172-31-7-184:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
   Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Sun 2024-09-29 19:04:03 UTC; 5min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 5478 (kubelet)
     Tasks: 10 (limit: 4676)
    Memory: 32.3M (peak: 32.8M)
      CPU: 6.520s
     CGroup: /system.slice/kubelet.service
             └─5478 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/ku
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454092 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454110 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454149 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454163 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454180 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454207 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:24 ip-172-31-7-184 kubelet[5478]: I0929 19:09:24.454222 5478 reconciler_common.go:245] "operationExecut>
Sep 29 19:09:29 ip-172-31-7-184 kubelet[5478]: E0929 19:09:29.041596 5478 kubelet.go:2902] "Container runtime networ>
Sep 29 19:09:34 ip-172-31-7-184 kubelet[5478]: E0929 19:09:34.042222 5478 kubelet.go:2902] "Container runtime networ>
Sep 29 19:09:35 ip-172-31-7-184 kubelet[5478]: I0929 19:09:35.871944 5478 scope.go:117] "RemoveContainer" containerI>
```

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

```
ubuntu@ip-172-31-7-184:~$ kubectl get nodes -o wide
NAME     STATUS   ROLES      AGE    VERSION
ip-172-31-3-88  Ready    <none>    2m2s   v1.31.1
ip-172-31-7-177  Ready    <none>    95s    v1.31.1
ip-172-31-7-184  Ready    control-plane  6m     v1.31.1

```

Now to Rename run this command kubectl label node ip-172-31-18-135

kubernetes.io/role=worker

**Rename to Node 1:** kubectl label node ip-172-31-3-88 kubernetes.io/role=Worker-Node1

**Rename to Node 2:** kubectl label node

ip-172-31-7-177 [kubernetes.io/role=Worker-Node2](#)

```
ubuntu@ip-172-31-7-184:~$ kubectl label node ip-172-31-3-88 kubernetes.io/role=Worker-Node1
node/ip-172-31-3-88 labeled
ubuntu@ip-172-31-7-184:~$ kubectl label node ip-172-31-7-177 kubernetes.io/role=Worker-Node2
node/ip-172-31-7-177 labeled
```

11. Now run kubectl get nodes

```
ubuntu@ip-172-31-7-184:~$ kubectl get nodes
NAME     STATUS   ROLES      AGE    VERSION
ip-172-31-3-88  Ready    Worker-Node1  4m22s   v1.31.1
ip-172-31-7-177  Ready    Worker-Node2  3m55s   v1.31.1
ip-172-31-7-184  Ready    control-plane  8m20s   v1.31.1
ubuntu@ip-172-31-7-184:~$ |
```

Hence we can see we have Successfully connected Node 1 and Node 2 to the Master.

Conclusion:

In this experiment, we successfully set up a Kubernetes cluster with one master and two worker nodes on AWS EC2 instances. After installing Docker, Kubernetes tools (kubelet, kubeadm, kubectl), and containerd on all nodes, the master node was initialized and the worker nodes were joined to the cluster. Initially, the nodes were in the NotReady state, which was resolved by installing the Calico network plugin. We also labeled the nodes with appropriate roles (control-plane and worker). The cluster became fully functional with all nodes in the Ready state, demonstrating the successful configuration and orchestration of Kubernetes.

## EXPERIMENT NO. 4

### Aim:

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

### Steps:

#### 1. Create a key pair.

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name  
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type | [Info](#)  
 RSA       ED25519

Private key file format  
 .pem  
For use with OpenSSH  
 .ppk  
For use with PuTTY

Tags - optional  
No tags associated with the resource.  
[Add new tag](#)  
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

Key pairs (2) <a href="#">Info</a>					
<input type="text"/> Find Key Pair by attribute or tag					
	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	rook	rsa	2024/09/14 22:42 GMT+5:30	40:38:ad:9e:d0:9d:51:f4:f1:20:b0:...	key-04
<input type="checkbox"/>	ec2	rsa	2024/08/05 13:07 GMT+5:30	15:77:76:82:87:d2:40:e4:db:c7:2a...	key-0b

The .pem file will be downloaded on your machine and will be required in the further steps.

#### 2. Now we will create an EC2 Ubuntu instance. Select the key pair which you just created while creating this instance.

Instances (1) <a href="#">Info</a>		Last updated less than a minute ago		<a href="#">Connect</a>	Instance state ▾	Actions ▾	<a href="#">Launch instances</a>	▼
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states ▾						
<input type="checkbox"/> Instance state = running		<a href="#">Clear filters</a>						
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available	
<input type="checkbox"/>	instance	i-051e99d82072f03cd	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	ap-so	

3. Now edit the inbound rules to allow ssh.

Edit inbound rules [Info](#)

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

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-009a122cf85a62854	SSH	TCP	22	Cus...	<input type="text"/> 0.0.0.0/0
-	All traffic	All	All	An...	<input type="text"/> 0.0.0.0/0

[Add rule](#)

4. Open git bash and go to the directory where pem file is located and use chmod to provide permissions.

```
bhumi@LAPTOP-RVJC2CFS MINGW64 ~/Downloads
$ chmod 400 rook.pem

bhumi@LAPTOP-RVJC2CFS MINGW64 ~/Downloads
$
```

5. Now use this command on the terminal: ssh -i <keyname>.pem ubuntu@ and replace
- Keyname with the name of your key pair, in our case test1.
  - As we are using amazon Linux instead of ubuntu we will have ec2-user
  - Replace public ip address with its value. Go to your instance and scroll down and you will find the public ip address there.

```
PS C:\Users\bhumi\Downloads> ssh -i "rook.pem" ubuntu@ec2-3-106-117-148.ap-southeast-2.compute.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 16:14:31 UTC 2024

System load: 0.56      Processes:          154
Usage of /: 55.3% of 6.71GB  Users logged in:   1
Memory usage: 69%          IPv4 address for enX0: 172.31.11.193
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Sun Sep 15 15:33:10 2024 from 106.216.240.125
```

## 6. Docker installation:

We will be installing docker by using “`sudo yum install docker -y`”

```
Last metadata expiration check: 0:05:38 ago on Sat Sep 14 17:38:25 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-libs     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
  libnftnl          x86_64    1.0.1-19.amzn2023.0.2  amazonlinux 30 K
  libliftnl         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          4.1 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm          6.8 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm             1.4 MB/s | 75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm  3.1 MB/s | 58 kB  00:00
(5/10): libnftnl-1.0.1-19.amzn2023.0.2.x86_64.rpm            1.2 MB/s | 30 kB  00:00
(6/10): libliftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            2.0 MB/s | 84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                  1.4 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          34 MB/s | 35 kB  00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm            32 MB/s | 44 MB  00:01
=====
Total                                         59 MB/s | 84 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                           1/1
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64  2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64          3/10
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64  4/10
  Installing : libliftnl-1.0.1-19.amzn2023.0.2.x86_64  5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
  Total                                         59 MB/s | 84 MB  00:01
  Preparing                                           1/1
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64  2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64          3/10
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64  4/10
  Installing : libliftnl-1.0.1-19.amzn2023.0.2.x86_64  5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
  Total                                         59 MB/s | 84 MB  00:01
  Preparing                                           1/1
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64  2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64          3/10
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64  4/10
  Installing : libliftnl-1.0.1-19.amzn2023.0.2.x86_64  5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
  Total                                         59 MB/s | 84 MB  00:01
  Preparing                                           1/10
  Installing : docker-25.0.6-1.amzn2023.0.2.x86_64          1/10
  Total                                         59 MB/s | 84 MB  00:01
Created symlink /etc/systemd/system/systemd-sysctl.service → /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   : libnftnl-1.0.1-19.amzn2023.0.2.x86_64          7/10
Verifying   : libliftnl-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          libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnftnl-1.0.1-19.amzn2023.0.2.x86_64
  libliftnl-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!
```

## 7. Then to configure cgroup in a daemon json file we will run

```
cd /etc/docker
```

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

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

```
EOF
```

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

```
[ec2-user@ip-172-31-3-10 ~]$ cd /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

## 8. 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
```

```

[ec2-user@ip-172-31-3-16 docker]$ sudo yum install -y kubelet kubeadm kubectl --disablereleases=kubernetes
Kubernetes
Last metadata expiration check: 0:00:02 ago on Sat Sep 14 17:47:29 2024.
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
  kubectl          x86_64     1.31.1-150500.1.1   kubernetes  15 M
  libnetfilter_cthelper x86_64   1.4.6-2.amzn2023.0.2   amazonlinux 208 k
  libnetfilter_cttimeout x86_64  1.0.0-19.amzn2023.0.2   amazonlinux 24 k
  libnetfilter_ctqueue x86_64   1.0.5-2.amzn2023.0.2   amazonlinux 24 k
  conntrack-tools   x86_64     1.4.6-2.amzn2023.0.2   amazonlinux 30 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_ctqueue 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 499 kB/s | 24 kB  00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 376 kB/s | 24 kB  00:00
(3/9): libnetfilter_ctqueue-1.0.5-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 30 kB  00:00
(4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm 1.7 MB/s | 208 kB  00:00
(5/9): kubelet-1.31.1-150500.1.1.x86_64.rpm 15 kB/s | 6.9 MB  00:00
(6/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm 21 kB/s | 1.9 MB  00:00
(7/9): kubectl-1.31.1-150500.1.1.x86_64.rpm 17 kB/s | 11 MB  00:00
(8/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm 21 kB/s | 7.1 MB  00:00
(9/9): kubelet-1.31.1-150500.1.1.x86_64.rpm 29 kB/s | 15 MB  00:00
=====
Total                                         45 MB/s | 51 MB  00:01
Kubernetes
Importing GPG key 0xA296436:
  Userid: <isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>>
  Fingerprint: D61A 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
  File: /etc/pki/rpm-gpg/k8s.io/stable:/v1.31/rpm/repodata/repomd.xml.key
Key imported successfully!
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : kubernetes-cni-1.5.1-150500.1.1.x86_64 1/9
  Installing : cri-tools-1.31.1-150500.1.1.x86_64 2/9
  Installing : libnetfilter_ctqueue-1.0.5-2.amzn2023.0.2.x86_64 3/9
  Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 4/9
  Installing : 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_ctqueue-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
  kubelet-1.31.1-150500.1.1.x86_64                    kubeadm-1.31.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64    kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64   libnetfilter_ctqueue-1.0.5-2.amzn2023.0.2.x86_64
=====
Complete!
[ec2-user@ip-172-31-3-16 docker]$

```

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

```
sudo swapoff -a
```

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

```
sudo sysctl -p
```

```

ubuntu@ip-172-31-11-193:~$ sudo swapoff -a
ubuntu@ip-172-31-11-193:~$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
ubuntu@ip-172-31-11-193:~$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
ubuntu@ip-172-31-11-193:~$ |

```

## 10. Initializing kubecluster:

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

```

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 172.31.3.16:6443 --token ekhyop.xkge2agz07jxxqgs \
    --discovery-token-ca-cert-hash sha256:8206263b4e2632eb03dafa4819c7c8505d47b21e8ba8c4901d5802c791c806f7

```

11. The mkdir command that is generated after initialization has to be copy pasted in the terminal.

```

ubuntu@ip-172-31-11-193:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

```

12. Then, add a common networking plugin called flannel:

```

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

```

```

ubuntu@ip-172-31-11-193:~$ 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

```

13. 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-172-31-11-193:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created

```

14. Use kubectl get pods to check if the pod is working correctly.

```

ubuntu@ip-172-31-11-193:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-49tlj   0/1     Pending   0          13s
nginx-deployment-d556bf558-qjqbg   0/1     Pending   0          13s

```

15. To change status from pending to running use the following command: kubectl describe pod nginx.

```

nginx-deployment-d556bf558-w2pd8  0/1  Pending   0          18s
[ec2-user@ip-172-31-3-16 docker]$ kubectl describe pod nginx
Name:           nginx-deployment-d556bf558-mvnj7
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:         app=nginx
Annotations:    pod-template-hash=d556bf558
Status:         Pending
IP:             <none>
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8cms7 (ro)
Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-8cms7:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <n/a>
    DownwardAPI:    true
  QoS Class:  BestEffort
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age   From           Message
  ----     ----            --   --            --
  Warning  FailedScheduling  57s   default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.

Name:           nginx-deployment-d556bf558-w2pd8
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:         app=nginx
Annotations:    pod-template-hash=d556bf558
Status:         Pending
IP:             <none>
Controlled By: ReplicaSet/nginx-deployment-d556bf558

Priority:       0
Service Account: default
Node:           <none>
Labels:         app=nginx
Annotations:    pod-template-hash=d556bf558
Status:         Pending
IP:             <none>
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6fl8b (ro)
Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-6fl8b:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <n/a>
    DownwardAPI:    true
  QoS Class:  BestEffort
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age   From           Message
  ----     ----            --   --            --
  Warning  FailedScheduling  57s   default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.

```

Use the below command to remove taints.

kubectl taint nodes –all node-role.kubernetes.io/control-plane-

```

[ec2-user@ip-172-31-3-16 ap-southeast-2.compute.internal ~]$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
[ec2-user@ip-172-31-3-16 ap-southeast-2.compute.internal ~]$

```

16. Check the pod status.

```

ubuntu@ip-172-31-11-193:~$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-68cf7659df-zx7vp   1/1     Running   0          19s

```

17. port forward the deployment to your localhost so that you can view it

```
ubuntu@ip-172-31-11-193:~$ kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

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

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

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

```
ubuntu@ip-172-31-11-193:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 16:14:46 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

Conclusion: In this experiment, we launched an EC2 instance and configured SSH access by updating the inbound rules. Next, we installed Docker and Kubernetes, and adjusted network settings to enable bridging. After completing the setup, we installed the Flannel networking plugin to ensure proper communication within the cluster. Once the cluster was up and running, we successfully deployed an NGINX server and verified its deployment.

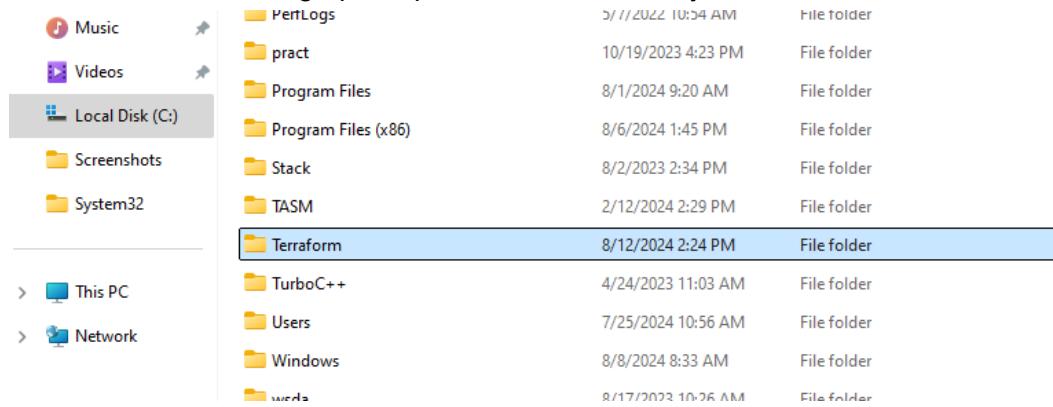
## EXPERIMENT NO. 5

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

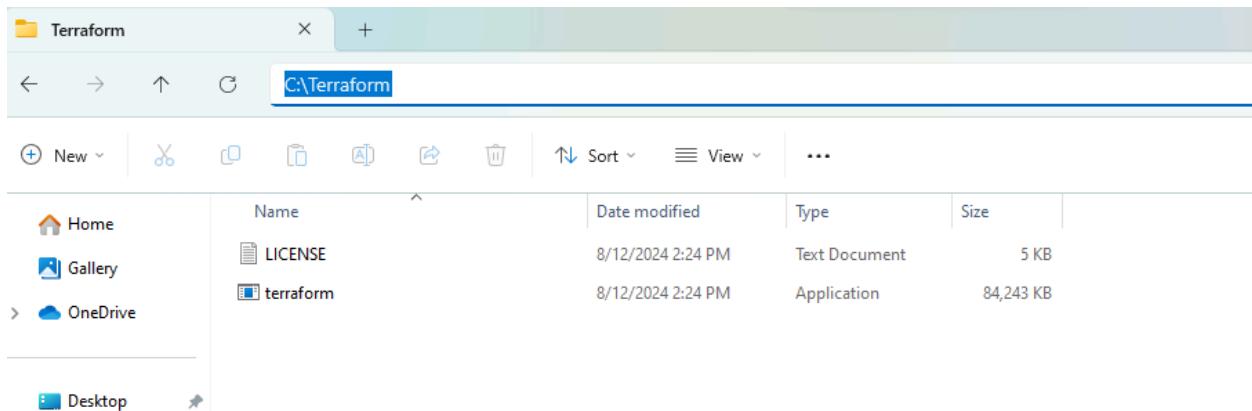
**Step 1:** Navigate to hashicorp.com page and download the AMD64 file for Terraform. Install it for Windows.

The screenshot shows the HashiCorp Terraform website. On the left, a sidebar lists 'Operating Systems' with 'Windows' selected. The main content area is titled 'Windows' and shows 'Binary download' options for '386' and 'AMD64' architectures, both at version 1.9.4. Below this, a section for 'Linux' shows package manager links for 'Ubuntu/Debian', 'CentOS/RHEL', 'Fedora', 'Amazon Linux', and 'Homebrew'. A terminal window at the bottom shows the command 'curl -O https://releases.hashicorp.com/terraform/1.9.4/terraform\_1.9.4\_windows\_amd64.zip' being run.

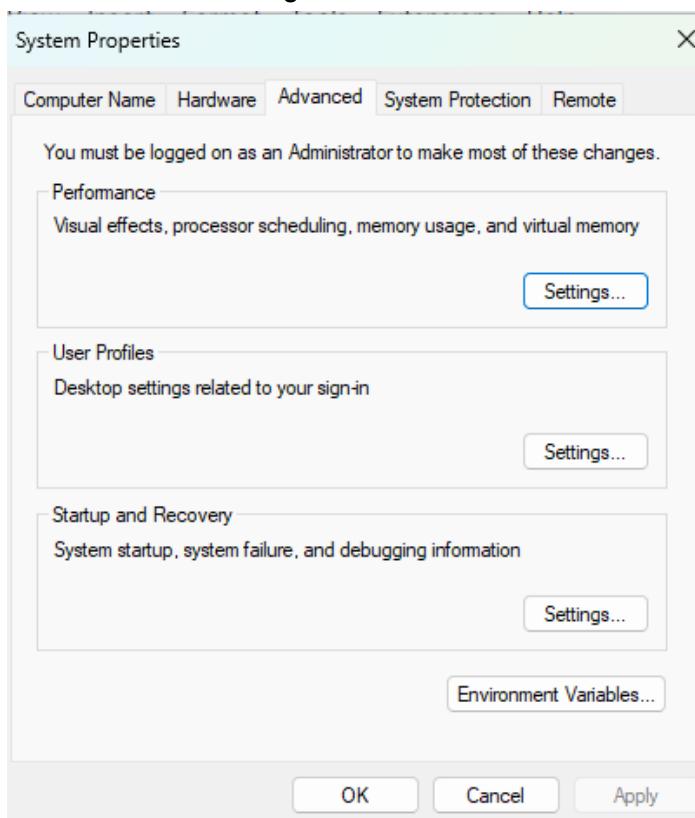
**Step 2:** Create a new folder inside your C drive named Terraform for your convenience. We would need it for setting up our path variable inside system variables.

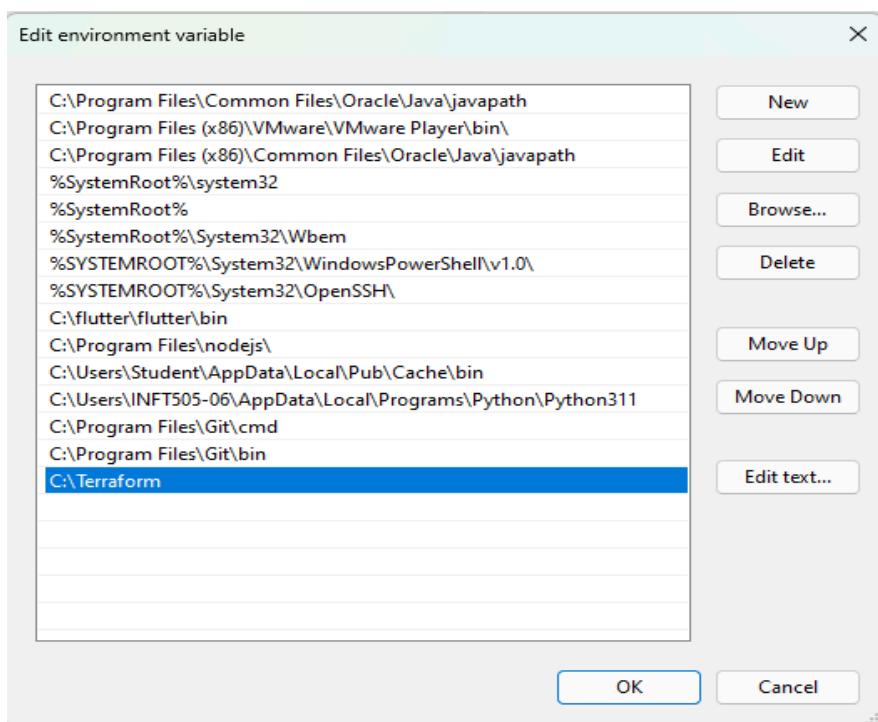
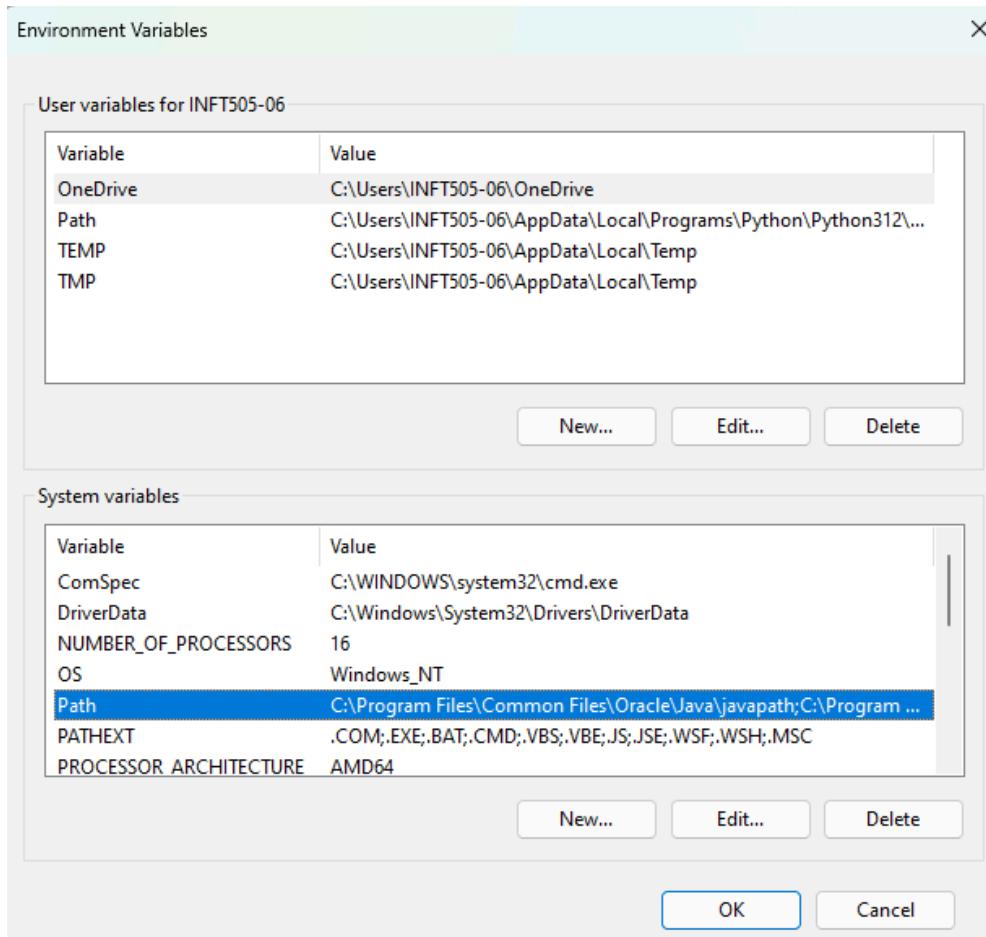


**Step 3:** Ensure that you have the following files appearing in your newly made Terraform folder in your local C drive.

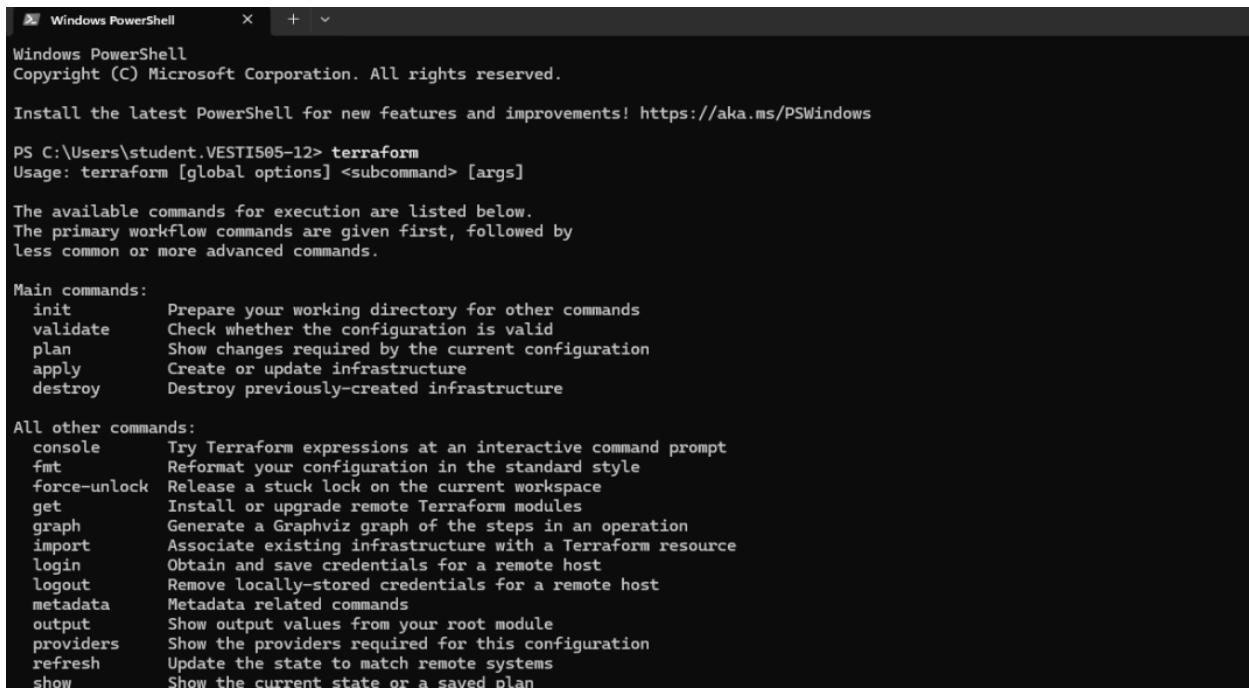


**Step 4:** Add the path of your Terraform folder in the PATH variable in the system environment variables inside settings.





**Step 5:** After opening Windows PowerShell, type the command ‘terraform’. If it is properly extracted and path is added to system variables, then following is the result we would be able to see. It will show us numerous other commands and options that we can try out so as to work with Terraform.



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the output of the "terraform" command. It includes the copyright notice, usage information, available commands, main commands, and all other commands. The text is white on a black background.

```
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\student.VESTI505-12> 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
```

## EXPERIMENT NO. 6

**Aim:** To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker) fdp.

Step 1: For this experiment, you need to install docker on your computer. Go to <https://www.docker.com/> and download the file according to the OS you have. Open the file and start the installation. Once installed, open your terminal and run 'docker' command. If this is your output, then docker is installed successfully.

```
C:\Users\bhumi>docker

Usage: docker [OPTIONS] COMMAND

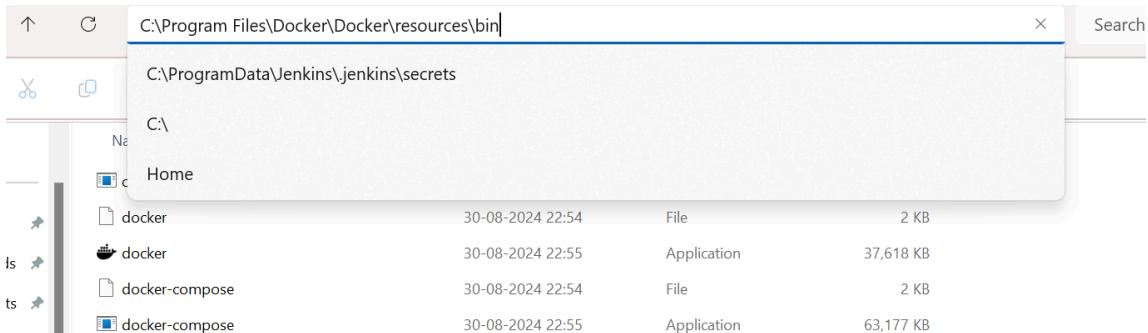
A self-sufficient runtime for containers

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

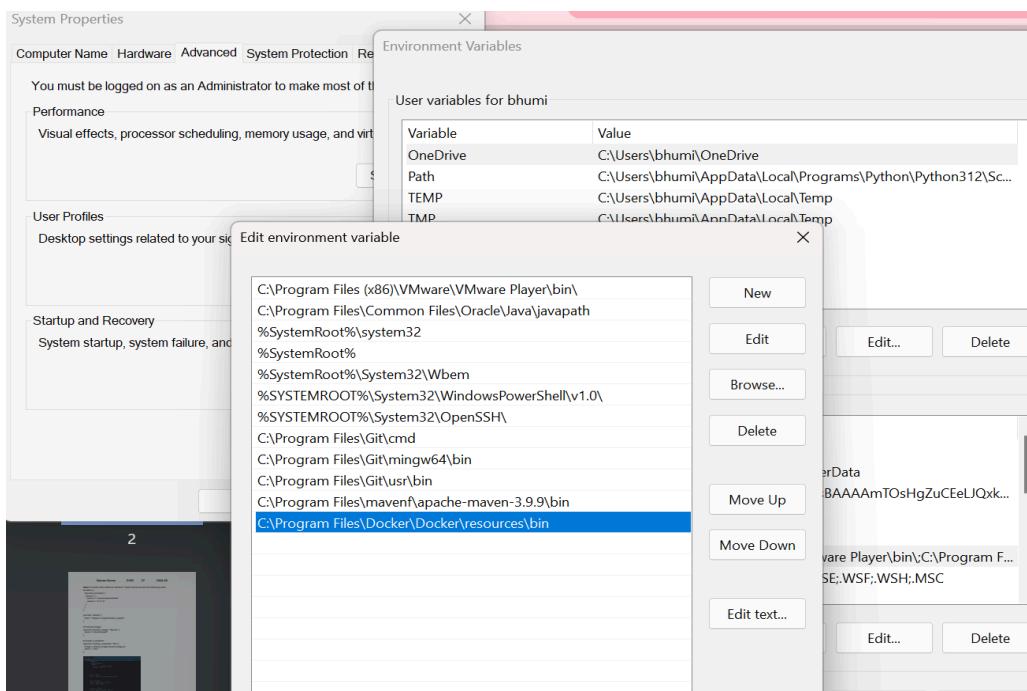
Management Commands:
  builder  Manage builds
  buildx*  Docker Buildx
  compose* Docker Compose
  container Manage containers
  context   Manage contexts
  debug*   Get a shell into any image or container
  desktop* Docker Desktop commands (Alpha)
  dev*    Docker Dev Environments
```

If you get an error like 'docker is not an internal or external command', you need to add the bin path of docker to your environment variables.

Go to File Explorer, and follow this path: C drive → Program Files → Docker → Docker → Resources → bin. Copy this path by clicking on the bar having the path and using shortcut CTRL + C



Open ‘Edit the System Environment Variables’ on your system. Click on Environment Variables. Now, check for a ‘Path’ variable under System variables, if it exists, click on it, then click on edit. Else, click on New and add the variable ‘Path’. If the variable existed, click on Edit, then on New. This will give you a text box. Paste the path you copied here and click on ok until you close all the tabs.



Now run the docker command again and the output would appear.  
Alternatively, you could also run ‘docker –version’ to check whether docker is started on the terminal.

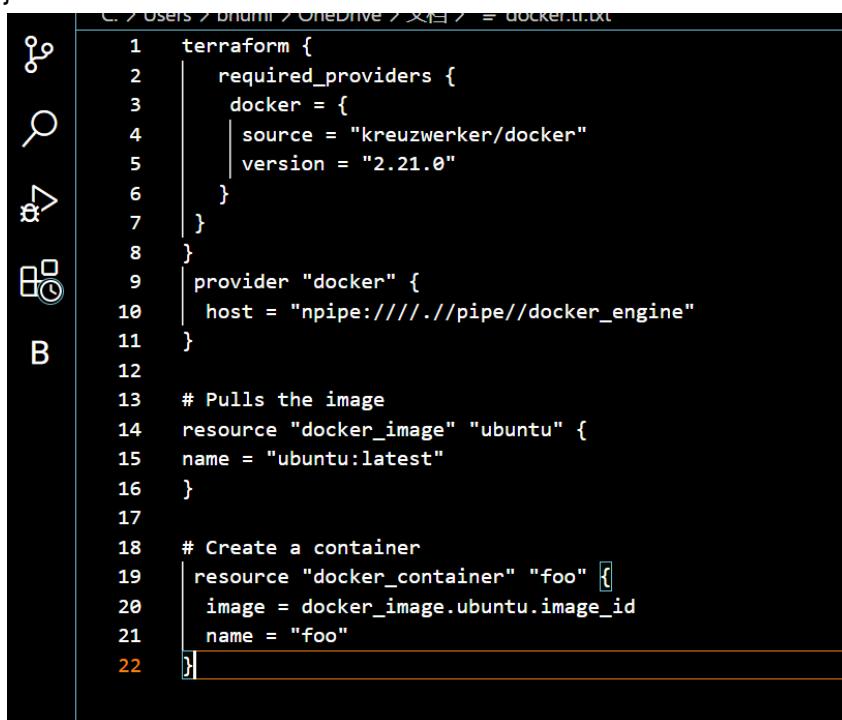
```
C:\Users\bhumis>docker --version
Docker version 27.1.1, build 6312585
```

Step 2: Create a file called 'docker.tf'. Open the file and put the following code.

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

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

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



```
C:\> users> brian> OneDrive> 项目> = docker.tf.txt
1  terraform {
2    required_providers {
3      docker = {
4        source = "kreuzwerker/docker"
5        version = "2.21.0"
6      }
7    }
8  }
9  provider "docker" {
10   host = "npipe://./pipe/docker_engine"
11 }
12
13 # Pulls the image
14 resource "docker_image" "ubuntu" {
15   name = "ubuntu:latest"
16 }
17
18 # Create a container
19 resource "docker_container" "foo" [
20   image = docker_image.ubuntu.image_id
21   name = "foo"
22 ]
```

Step 3: Open the folder where the docker.tf is present on your terminal. Execute the command 'terrafom init'. This will initialize terraform in the directory.

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

Terraform has been successfully initialized!

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

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

Step 4: Run the command 'terraform plan'. This creates an execution plan and lets you
overview changes that are going to happen in your infrastructure.

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

Terraform will perform the following actions:

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

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

Using the command : "terraform 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.

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 10s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...

```

## Docker images, Before Executing Apply step:

```
C:\Users\2022k\OneDrive\Desktop\Docker\Terraform Scripts>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
```

## Docker images, After Executing Apply step:

```
C:\Users\2022k\OneDrive\Desktop\Docker\Terraform Scripts>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       edbfe74c41f8  3 weeks ago  78.1MB
```

## Step 6: Execute Terraform destroy to delete the configuration, which will automatically Delete the ubuntu container.

```
C:\Users\2022k\OneDrive\Desktop\Terraform Scripts>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

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

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

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

Enter a value: yes

docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
```

## Docker images After Executing Destroy step.

```
C:\Users\2022k\OneDrive\Desktop\Docker\Terraform Scripts>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
```

## EXPERIMENT NO. 7

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

Prerequisites:

1) Docker

Run docker -v command.

Use this command to check if docker is installed and running on your system.

```
C:\Users\bhumii>docker -v
Docker version 27.2.0, build 3ab4256
```

2) Install SonarQube image Command:

docker pull sonarqube

This command helps you to install an image of SonarQube that can be used on the local system without actually installing the SonarQube installer.

```
C:\Users\bhumii>docker pull sonarqube
Using default tag: latest
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
docker.io/library/sonarqube:latest
```

3) Keep jenkins installed on your system.

Experiment Steps:

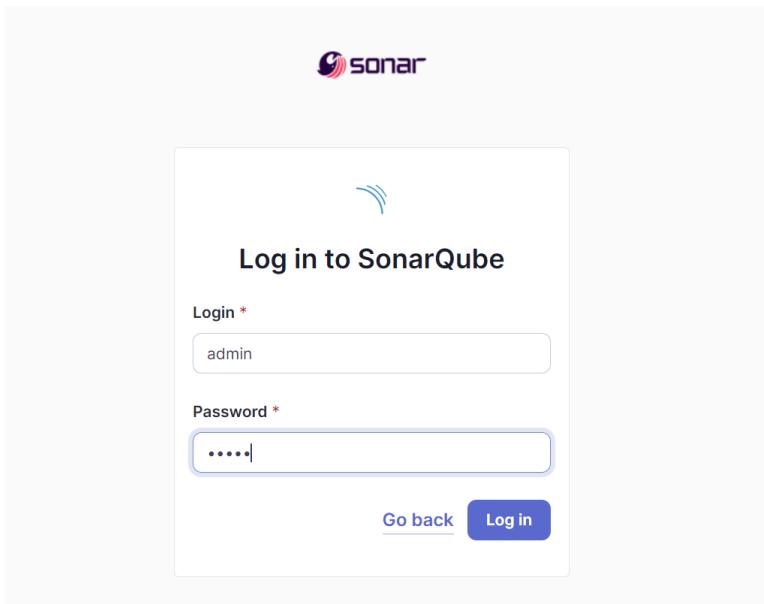
1. Run SonarQube in a Docker container using this command -

docker run -d --name sonarqube -e

SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\bhumii>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
7401befce9e7a7248c0e5648a2913e99c3843cce08e91d403e5af3a1479a151e
```

2. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
3. Login to SonarQube using username admin and password admin.



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

Click on Create a Local Project.

Here, I have given the name 'sonarqube'. This will be your project key as well. We'll use it later.

Keep the branch main only and click on next.

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](#)

Set up the project as required and click on create.

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. Open Jenkins on whichever port it is installed. (<http://localhost:<port number>>).

The screenshot shows the Jenkins dashboard. On the left, there are two expandable sections: "Build Queue" (which is currently empty) and "Build Executor Status". Under "Build Executor Status", there is one "Built-In Node" named "slave1" which is listed as "(offline)". On the right, there is a table titled "All" showing the last three builds. The columns are: S (Status), W (Warning), Name (bhumi, boompipeline, web), Last Success (27 days ago, #2, #1, #7), Last Failure (N/A, N/A, 27 days ago, #6), and Last Duration (2.7 sec, 4.8 sec, 0.36 sec). There is also a "Manage Jenkins" link at the top.

REST API Jenkins 2.462.1

6. Go to Manage jenkins → Plugins → Available Plugins  
Search for Sonarqube Scanner for Jenkins and install it.

The screenshot shows the "Available plugins" section of the Jenkins plugin manager. A search bar at the top right contains the text "sonarqube". Below the search bar, there is a table with columns: "Install", "Name", and "Released". The first row in the table is for the "SonarQube Scanner" plugin, version 2.17.2, released 7 months and 9 days ago. It has a checked checkbox under "Install". Other rows in the table include "Sonar Gerrit" (released 3 months and 23 days ago) and "SonarQube Generic Coverage" (released 5 years and 2 months ago). On the left side, there is a sidebar with links: "Updates" (14), "Available plugins" (selected), "Installed plugins", and "Advanced settings".

7. Under Jenkins 'Configure System', look for SonarQube Servers and enter the details.  
I have named the server 'sonarqube' and added the server url for jenkins.  
Enter the Server Authentication token if needed.

Dashboard > Manage Jenkins > System >

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name: sonarqube

Server URL: http://localhost:9000

Server authentication token: - none -

Advanced

Save Apply

The screenshot shows the 'SonarQube servers' configuration page in Jenkins. It includes fields for 'Name' (set to 'sonarqube'), 'Server URL' (set to 'http://localhost:9000'), and 'Server authentication token' (set to '- none -'). There is also an 'Advanced' section and 'Save' and 'Apply' buttons.

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

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name: sonarqube

Install automatically ?

☰ Install from Maven Central

Version: Sonarqube Scanner 6.1.0.4477

Add Installer ▾

The screenshot shows the 'SonarQube Scanner installations' configuration page in Jenkins. It includes a 'Name' field (set to 'sonarqube'), an 'Install automatically' checkbox (checked), and a 'Version' field (set to 'Sonarqube Scanner 6.1.0.4477'). There is also an 'Add Installer' button at the bottom right.

9. After configuration, create a New Item → choose a freestyle project and name your project. Here, I have given the ‘sonarqube’, then click on OK.

10. Choose the github repository [https://github.com/shazforiot/MSBuild\\_firstproject](https://github.com/shazforiot/MSBuild_firstproject). It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

Fork this repository.

11. Enable git and add the repository you forked.

The screenshot shows the Jenkins 'Configuration' screen for a job named 'bhumisha'. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. In the main area, under 'Repositories', there is a 'Repository URL' field containing 'https://github.com/bhumishap/MSBuild\_firstproject'. Below it is a 'Credentials' dropdown set to '- none -' and a '+ Add' button. An 'Advanced' dropdown is also present. A 'Branches to build' section follows, with a 'Branch Specifier (blank for 'any')' field. At the bottom of the page are 'Add Repository' and 'Save' buttons.

12. Under Build Steps, select Execute Sonarqube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

The screenshot shows the 'Build Steps' configuration for the 'bhumisha' job. It includes a 'Execute SonarQube Scanner' step. Under this step, the 'JDK' dropdown is set to '(Inherit From Job)'. The 'Path to project properties' field is empty. The 'Analysis properties' field contains the following configuration:

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

The 'Additional arguments' field is empty. At the bottom of the step configuration are 'Save' and 'Apply' buttons.

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

#### Global Permissions

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

All Users Groups  Search for users or groups...

	Administrator System	Administrator	Execute Analysis	Create
sonar-administrators System administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/> <input checked="" type="checkbox"/> Projects
sonar-users Every authenticated user automatically belongs to this group	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Projects
Anyone DEPRECATED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/> <input type="checkbox"/> Projects
Administrator admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/> <input type="checkbox"/> Projects

14. Go back to jenkins. Go to the job you had just built and click on Build Now.

Dashboard > sonarqube >

Status ✓ sonarqube

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

Build History trend ▾

SonarQube

Permalinks

- Last build (#1), 2 hr 53 min ago
- Last stable build (#1), 2 hr 53 min ago
- Last successful build (#1), 2 hr 53 min ago
- Last completed build (#1), 2 hr 53 min ago

15. Check the console Output.

```
Dashboard > sonarque > #1 > Console Output

[ Status ] [ Download ] [ Copy ] View as plain text

Console Output
Started by user unknown or anonymous
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\Jenkins\workspace\sonarque
The recommended git tool is NONE
No credential specified
> git@github.com: -> resolve-git-dir C:\ProgramData\Jenkins\Jenkins\workspace\sonarque.git # timeout=10
Fetching changes from the remote git repository
> git.exe config remote.origin.url https://github.com/bmuhapl/MsBuild_FirstProject # timeout=10
Fetching upstream changes from https://github.com/bmuhapl/MsBuild_FirstProject
> git pull --tags --prune --timeout=10
> git ->version # git version 2.46.1.windows.1
> git.exe fetch --tags --force --progress -- https://github.com/bmuhapl/MsBuild_FirstProject +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse --refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2b042e4cde72d27388baeaddfe70adef (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f f2b042e4cde72d27388baeaddfe70adef # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
Unpacking binary distribution from https://sonarsource-scanner-cli.s3-eu-west-1.amazonaws.com/sonarscanner-cli/sonar-scanner-CLI-1.10.4077/sonar-scanner-CLI-1.10.4077.zip to C:\ProgramData\Jenkins\Jenkins\workspace\sonarque on Jenkins
[stage] $ C:\ProgramData\Jenkins\Jenkins\workspace\sonarque\sonar-scanner\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarque -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=.
Docker password is passed -> sonar-projectKey=(src) [C:\ProgramData\Jenkins\Jenkins\workspace\sonarque]
23:04:51,587 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
23:04:51,595 INFO Scanner configuration file: C:\ProgramData\Jenkins\Jenkins\workspace\sonarque\plugins.sonar.sonarRunnerInstallation\sonarqubeScan\bin\conf\sonar-scanner.properties
23:04:51,603 INFO Project root configuration file: NONE
23:04:51,611 INFO Java: Java 17.0.2 Oracle Corporation (64-bit)
23:04:51,621 INFO Java 21 Oracle Corporation (64-bit)
23:04:51,631 INFO Windows 11 10.0.22621
23:04:51,640 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
23:04:51,648 INFO Using system properties: java.awt.image.ditherType, arc4random
23:05:10,124 INFO Communicating with Sonarque Server 10.6.0.9216
23:05:10,896 INFO Starting SonarScanner Engine...
23:05:10,896 INFO Java 17.0.2 Eclipse Adoptium (64-bit)
23:05:10,897 INFO Load global settings (done) | time=38ms
23:05:10,905 INFO Load global settings (done) | time=38ms
23:05:10,922 INFO Load project settings (done) | time=38ms
23:05:10,930 INFO Load project settings (done) | time=38ms
23:05:10,938 INFO Loading required plugins
23:05:09,262 INFO Load plugin index
23:05:09,534 INFO Load plugin index (done) | time=292ms
23:05:09,542 INFO Load plugin index
23:05:10,782 INFO Load/download plugins (done) | time=1248ms
23:05:10,821 INFO Process project properties
23:05:11,871 INFO Process project properties (done) | time=21ms
23:05:11,886 INFO Project key: sonarque
23:05:11,894 INFO Load component key: sonarque
23:05:11,892 INFO working dir: C:\ProgramData\Jenkins\Jenkins\workspace\sonarque\scannerwork
23:05:11,942 INFO Load project settings for component key: 'sonarque'
23:05:12,358 INFO Load project settings for component key: 'sonarque' (done) | time=41ms
23:05:12,416 INFO Load quality profiles
23:05:12,424 INFO Load quality profiles (done) | time=48ms
23:05:12,511 INFO Auto-configuring with CI 'Jenkins'
.....
#1 Console Output
[ Status ] [ Download ] [ Copy ] View as plain text

23:05:48,288 INFO Sensor IacCloudformationSensor [iac]
23:05:48,304 INFO @ source files have been analyzed
23:05:48,467 INFO Sensor IacCloudformationSensor [iac] (done) | time=196ms
23:05:48,467 INFO Sensor IacAzureResourceManagerSensor [iac]
23:05:48,484 INFO @ source files to be analyzed
23:05:48,712 INFO Sensor IacAzureResourceManagerSensor [iac] (done) | time=24ms
23:05:48,712 INFO Sensor IacJavaConfigSensor [iac]
23:05:48,728 INFO @ source files to be analyzed
23:05:48,736 INFO Sensor IacJavaConfigSensor [iac] (done) | time=24ms
23:05:48,736 INFO Sensor IacDockerSensor [iac] (done) | time=24ms
23:05:48,744 INFO @ source files to be analyzed
23:05:48,819 INFO @/source files have been analyzed
23:05:48,819 INFO Sensor IacDockerSensor [iac] (done) | time=83ms
23:05:48,819 INFO Sensor TextAndSecretsSensor [text]
23:05:48,919 INFO Available processors: 12
23:05:48,919 INFO Using thread for analysis.
23:05:49,014 INFO The property 'sonar.tests' is not set. To improve the analysis accuracy, we categorize a file as a test file if any of the following is true:
* The filename starts with "test"
* The filename contains "Test," or "tests."
* Any directory in the file path is named: "doc", "docs", "test" or "tests"
* Any directory in the file path has a name ending in "test" or "tests"

23:05:42,131 INFO Using git CLI to retrieve untracked files
23:05:42,131 INFO 14 source files to be analyzed
23:05:43,887 INFO 14/14 source file has been analyzed
23:05:43,143 INFO Sensor TextAndSecretsSensor [text] (done) | time=2260ms
23:05:43,112 INFO ..... Run sensors on project
23:05:43,120 INFO Sensor C# [charp]
23:05:43,421 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
23:05:43,429 INFO Sensor C# [charp] (done) | time=8ms
23:05:43,429 INFO Sensor AnalysisWarning [charp]
23:05:43,429 INFO Sensor AnalysisWarning [charp] (done) | time=8ms
23:05:43,429 INFO Sensor C# Cache Caching Sensor [charp]
23:05:43,429 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBasedir' property.
23:05:43,429 INFO Sensor C# Cache Caching Sensor [charp] (done) | time=8ms
23:05:43,429 INFO Sensor Zero Coverage Sensors
23:05:43,446 INFO Sensor Zero Coverage Sensors (done) | time=17ms
23:05:43,454 INFO SCM Publisher SON provider for this project is git
23:05:43,454 INFO SCM Publisher 4 source files to be analyzed
23:05:44,113 INFO Sensor SCM Publisher [git] (done) | time=658ms
23:05:44,119 INFO COP Executor COP calculation finished (done) | time=8ms
23:05:44,127 INFO SCM revision ID "f2b042e4cde72d27388baeaddfe70adef"
23:05:44,425 INFO Analysis report generated in 261ms, dir size=281.0 KB
23:05:44,718 INFO Analysis report compressed in 66ms, zip size=22.3 kB
23:05:45,068 INFO Analysis report uploaded in 94ms
23:05:45,068 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarque
23:05:45,068 INFO More about the report processing at http://localhost:9000/api/ca/TaskId=8e13-4095-820e-85a7fe54412
23:05:45,088 INFO Analysis total time: 14.809s
23:05:45,088 INFO SonarScanner Engine completed successfully
23:05:45,105 EXECUTION SUCCESS
23:05:45,165 INFO Total time: 49.576s
.....
@stuckin-a-currecc
```

16. Once the build is complete, go back to SonarQube and check the project linked.

The SonarQube dashboard shows a green 'Passed' status with 0 issues across all metrics: Security, Reliability, and Maintainability. The Jenkins dashboard shows a successful build history for the 'sonarqube' project, with the last build completed 2 hours ago.

## Conclusion:

In this experiment, we explored how to perform Static Application Security Testing (SAST) in Jenkins using SonarQube. Instead of installing SonarQube directly on the local system, we utilized a Docker image for a more streamlined setup. After configuring Jenkins with the necessary SonarQube settings, we integrated it with a GitHub repository containing code for analysis. Upon building the project, SonarQube successfully scanned the code, confirming that no errors were present. This process demonstrated how Jenkins and SonarQube can be effectively used together for automated code quality and security analysis.

## EXPERIMENT NO. 8

Aim: Create a Jenkins CI/CD 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.

Prerequisites:

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

```
C:\Users\bhumit>docker -v
Docker version 27.2.0, build 3ab4256

C:\Users\bhumit>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

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

- Download sonar scanner

The screenshot shows the SonarScanner CLI page on the SonarQube documentation site. The page has a sidebar with links like 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code' (expanded), 'Scanners' (expanded), and 'SonarScanner CLI'. The main content area features a large 'SonarScanner CLI' heading, a release note for version 6.2 (published 2024-09-17) about PKCS12 truststore support, download links for various platforms, and release notes. A sidebar on the right lists 'On this page' topics such as 'Configuring your project', 'Running SonarScanner CLI from the zip file', 'Running SonarScanner CLI from the Docker image', etc.

Extract the downloaded zip file in a folder.

### Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

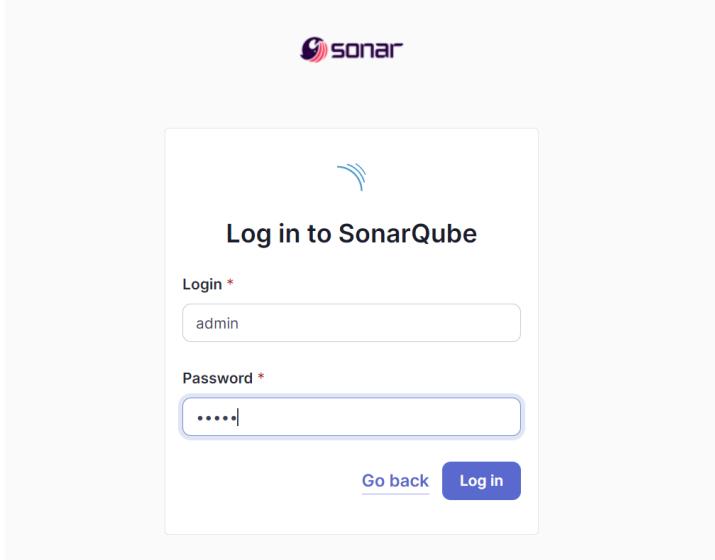
1. Run SonarQube image

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

This command will run the SonarQube image that was just installed using docker.

```
C:\Users\bhumit>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
7401befce9e7a7248c0e5648a2913e99c3843cce08e91d403e5af3a1479a151e
```

2. Once the SonarQube image is started, you can go to <http://localhost:9000> to find the SonarQube that has started.
3. Login to SonarQube using username admin and password admin.



4. Create a local project in SonarQube and enter a name.

Here I have given the name 'bhumiquibe' which is also the project key.

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

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.

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

Dashboard >

+ New Item      Add description

Build History      All      +

Manage Jenkins

S	W	Name ↓	Last Success	Last Failure	Last Duration
🕒	☀️	bhumi	27 days #2	N/A	2.7 sec
🕒	☀️	boompipeline	27 days #1	N/A	4.8 sec
🕒	☁️	web	27 days #7	27 days #6	0.36 sec

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

Icon: S M L      ...

1 Idle  
2 Idle  
slave1 (offline)

REST API      Jenkins 2.462.1

6. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.(we already installed it for exp 7 so you can skip)

Dashboard > Manage Jenkins > Plugins

Plugins

Updates      17

Available plugins

Installed plugins

Advanced settings

sonar

Name ↓	Enabled
SonarQube Scanner for Jenkins 2.17.2	<input checked="" type="checkbox"/>

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.  
[Report an issue with this plugin](#)

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

In SonarQube installations: Under Name add <project name of sonarqube> which is 'bhumiqube' for me. In Server URL Default is http://localhost:9000.

The screenshot shows the Jenkins system configuration page for SonarQube servers. The path is Dashboard > Manage Jenkins > System > SonarQube servers. A note says: "If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build." A checkbox labeled "Environment variables" is checked. Below it, under "SonarQube installations", there is a list titled "List of SonarQube installations". A single entry is shown with the "Name" field containing "bhumiqube" and the "Server URL" field containing "http://localhost:9000". There is also a "Server authentication token" section with a dropdown menu showing "- none -" and a "+ Add" button. An "Advanced" dropdown is also present.

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

Dashboard > Manage Jenkins > Tools > SonarQube Scanner

The screenshot shows the Jenkins Global Tool Configuration page for SonarQube Scanner installations. The path is Dashboard > Manage Jenkins > Tools > SonarQube Scanner. A note says: "SonarQube Scanner installations" and "Edited". A button "Add SonarQube Scanner" is visible. A configuration entry for "SonarQube Scanner" is shown with the "Name" field set to "bhumiqube". The "Install automatically" checkbox is checked. Under the "Install from Maven Central" section, the "Version" dropdown is set to "SonarQube Scanner 6.1.0.4477". An "Add Installer" dropdown is also present. At the bottom, another "Add SonarQube Scanner" button is visible.

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

1 of 2

## Create a local project

Project display name \*

bhumique



Project key \*

bhumique



Main branch name \*

main

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

Cancel

Next

10. Under Pipeline script, enter the following:

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
  
    stage('SonarQube analysis') {  
        withSonarQubeEnv('bhumique') {  
            bat """"  
                "C:\Program  
Files\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat" ^  
                    -D sonar.login=squ_19c75dfc2e1126a15d28436e4cf82fd4b8cac39a ^  
                    -D sonar.projectKey=bhumique ^  
                    -D sonar.exclusions=vendor/**,resources/**,*/*.java ^  
                    -D sonar.host.url=http://localhost:9000/  
                """  
        }  
    }  
}
```

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

## Pipeline

### Definition

Pipeline script

#### Script ?

```
1 * node {
2 *     stage('Cloning the GitHub Repo') {
3 *         git 'https://github.com/shazforiot/GOL.git'
4 *     }
5 *
6 *     stage('SonarQube analysis') {
7 *         withSonarQubeEnv('bhumiqueube'){
8 *             bat """
9 *             "C:\Program Files\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat" ^
10 *             -D sonar.login=squ_19c75dfc2e1126a15d28436e4cf82fd4b8cac39a ^
11 *             -D sonar.projectKey=bhumiqueube ^
12 *             -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
13 *             -D sonar.host.url=http://localhost:9000
14 *             """
15 *         }
16 *     }
17 * }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

## 11. Go to the job you had just built and click on Build Now.

Dashboard > bhumiqueube >

 Status

 Changes

 Build Now

 Configure

 Delete Pipeline

 SonarQube

 Stages

 Rename

 Pipeline Syntax

## 12. Once it is built, check the console output.

Dashboard > bhumique >

**bhumique**

- </> Changes
- ▷ Build Now
- ⚙ Configure
- 🗑 Delete Pipeline
- 🔍 SonarQube
- ☰ Stages
- 🔗 Rename

**Console Output**

[Download](#) [Copy](#) [View as plain text](#)

Skipping 4.249 KB.. [Full Log](#)

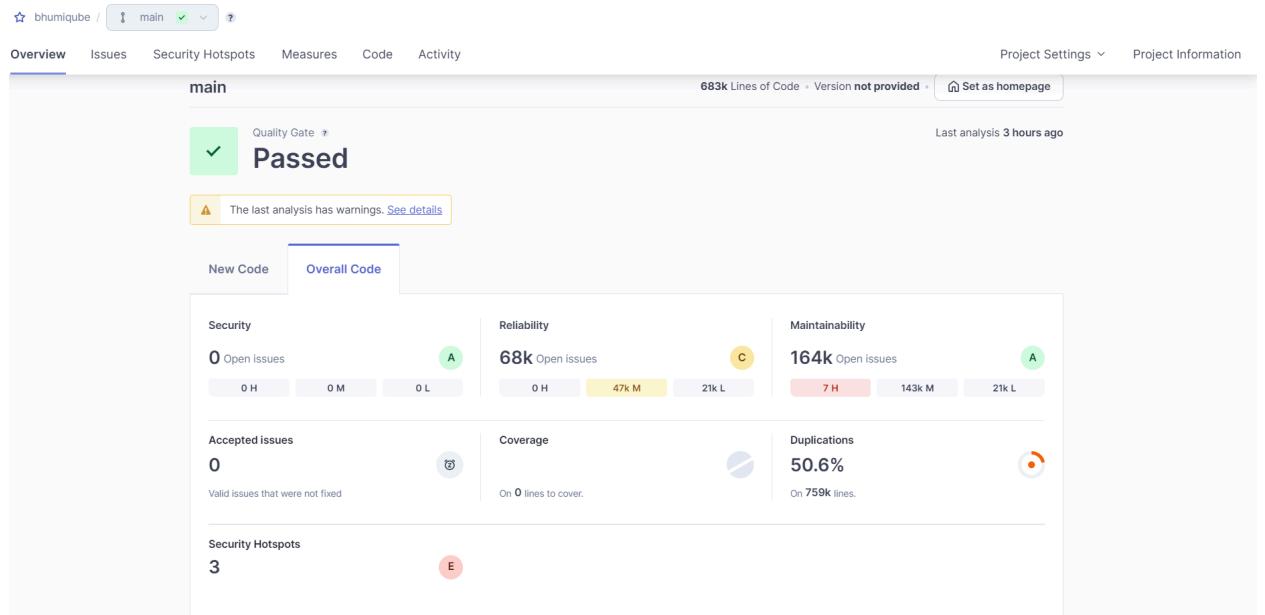
```

21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/writers/ReportSummary.html for block at line 41. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/writers/ReportSummary.html for block at line 17. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/writers/ReportSummary.html for block at line 303. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/writers/ReportSummary.html for block at line 312. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 315. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 649. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 312. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 651. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 17. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 312. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 315. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 653. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 655. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 315. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 325. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 32. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 312. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 697. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 315. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 64. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 707. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 64. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 40. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 74. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 41. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 17. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 136. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 136. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 655. Keep only the first 100 references.
21:00:36.356 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jorphan/math/StatCalculator.html for block at line 74. Keep only the first 100 references.
21:00:36.526 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/control/gui/package-tree.html for block at line 16. Keep only the first 100 references.
21:00:36.526 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/control/gui/package-tree.html for block at line 219. Keep only the first 100 references.

21:00:42.517 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/ldap/control/gui/LdapExtTestSamplerGui.html for block at line 155. Keep only the first 100 references.
21:00:42.643 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/ldap/control/gui/LdapExtTestSamplerGui.html for block at line 515. Keep only the first 100 references.
21:00:42.643 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/ldap/control/gui/LdapExtTestSamplerGui.html for block at line 768. Keep only the first 100 references.
21:00:42.643 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/ldap/control/gui/LdapExtTestSamplerGui.html for block at line 714. Keep only the first 100 references.
21:00:42.643 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/ldap/control/gui/LdapExtTestSamplerGui.html for block at line 668. Keep only the first 100 references.
21:00:42.643 INFO CPD Executor CPD calculation finished (done) | time=187786ms
21:00:42.660 INFO SCM revision ID 'ba799ba7eb576f04a4612322b0412c5e6e1e5e4'
21:02:50.519 INFO Analysis report generated in 5916ms, dir size=127.2 MB
21:03:11.285 INFO Analysis report compressed in 20750ms, zip size=29.6 MB
21:03:11.715 INFO Analysis report uploaded in 430ms
21:03:11.717 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=bhumique
21:03:11.717 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:03:11.717 INFO More about the report processing at http://localhost:9000/api/ce/task?id=2bf5a9f3-919a-4725-94d5-474d4773e610
21:03:26.163 INFO Analysis total time: 16:37.760 s
21:03:26.163 INFO SonarScanner Engine completed successfully
21:03:26.916 INFO EXECUTION SUCCESS
21:03:26.916 INFO Total time: 16:42.318s
[[Pipeline] ]
[[Pipeline] // withSonarQubeEnv
[[Pipeline] ]
[[Pipeline] // stage
[[Pipeline] ]
[[Pipeline] // node
[[Pipeline] End of Pipeline
Finished: SUCCESS

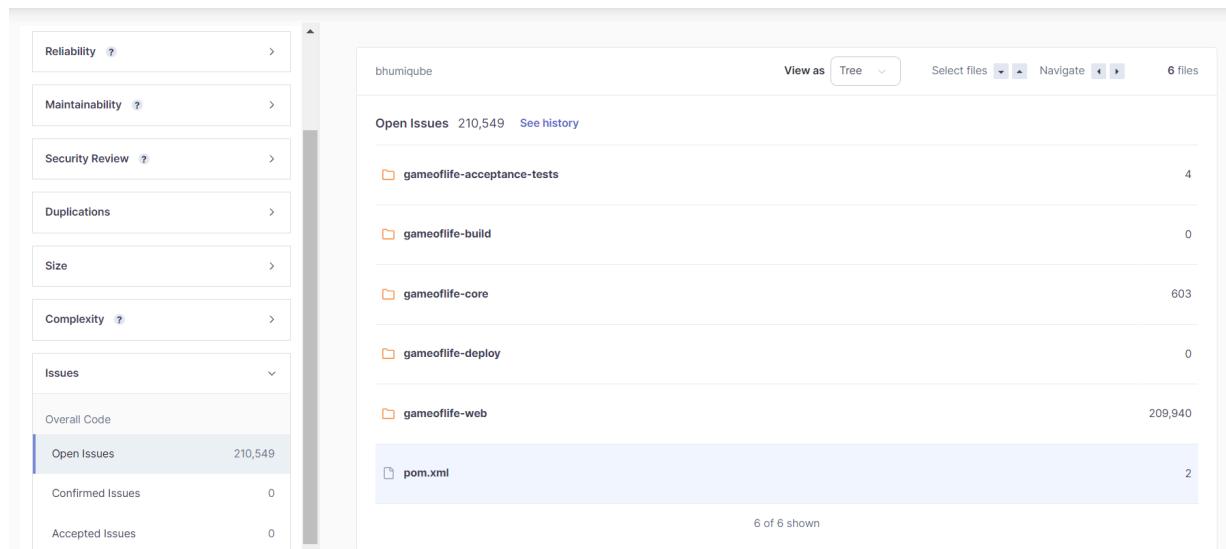
```

13. Once the build is complete, go back to SonarQube and check the project linked.



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

- Code Problems



## ● Consistency

The screenshot shows a software interface for managing code quality. The top navigation bar includes tabs for Overview, Issues (selected), Security Hotspots, Measures, Code, and Activity. On the right, Project Settings and Project Information are visible, showing 196,662 issues and 3075d effort.

The left sidebar features a 'Filters' section with a 'Clear All Filters' button. Under 'Issues in new code', there is a 'Clean Code Attribute' section expanded, showing:

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

An 'Add to selection Ctrl + click' link is present. Below this is a 'Software Quality' section with:

- Security: 0
- Reliability: 54k
- Maintainability: 164k

The main panel displays several code review items under the file 'gameoflife-core/build/reports/tests/all-tests.html':

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Consistency) Reliability: 1. Open Not assigned L1 - 5min effort - 4 years ago - Bug - Major
- Remove this deprecated "width" attribute. (Consistency) Maintainability: 1. Open Not assigned L9 - 5min effort - 4 years ago - Code Smell - Major
- Remove this deprecated "align" attribute. (Consistency) Maintainability: 1. Open Not assigned L11 - 5min effort - 4 years ago - Code Smell - Major
- Remove this deprecated "align" attribute. (Consistency) Maintainability: 1. Open Not assigned L11 - 5min effort - 4 years ago - Code Smell - Major

## ● Intentionality

The screenshot shows a software interface for managing code quality, similar to the previous one but with a different focus.

The top navigation bar includes tabs for Overview, Issues (selected), Security Hotspots, Measures, Code, and Activity. On the right, Project Settings and Project Information are visible, showing 13,887 issues and 59d effort.

The left sidebar features a 'Filters' section with a 'Clear All Filters' button. Under 'Issues in new code', there is a 'Clean Code Attribute' section expanded, showing:

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

An 'Add to selection Ctrl + click' link is present. Below this is a 'Software Quality' section with:

- Security: 0
- Reliability: 14k
- Maintainability: 15

The main panel displays several code review items under the file 'gameoflife-acceptance-tests/Dockerfile':

- Use a specific version tag for the image. (Intentionality) Maintainability: 1. Open Not assigned L1 - 5min effort - 4 years ago - Code Smell - Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) Maintainability: 1. Open Not assigned L12 - 5min effort - 4 years ago - Code Smell - Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) Maintainability: 1. Open Not assigned L12 - 5min effort - 4 years ago - Code Smell - Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) Maintainability: 1. Open Not assigned L12 - 5min effort - 4 years ago - Code Smell - Major

## ● Bugs

bhumiqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Software Quality

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

Severity

Type

- Bug: 14k (selected)
- Vulnerability: 0
- Code Smell: 268

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 "xmllang" attributes to this "<html>" element

Intentionality Reliability

Open Not assigned

L1 - 2min effort 4 years ago Bug Major

accessibility wcag2-a

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

Add "<th> headers to this "<table>".

Intentionality Reliability

Open Not assigned

L9 - 2min effort 4 years ago Bug Major

accessibility wcag2-a

Add "lang" and/or "xmllang" attributes to this "<html>" element

Intentionality Reliability

Open Not assigned

L1 - 2min effort 4 years ago Bug Major

accessibility wcag2-a

## ● Code Smells

bhumiqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Software Quality

- Security: 0
- Reliability: 253
- Maintainability: 15

Severity

Type

- Bug: 14k
- Vulnerability: 0
- Code Smell: 268 (selected)

Add to selection Ctrl + click

Scope

Status

Security Category

Select Issues Navigate to issue 268 issues 2d 5h effort

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image.

Intentionality Maintainability

Open Not assigned

L1 - 5min effort 4 years ago Code Smell Major

No tags

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Intentionality Maintainability

Open Not assigned

L1 - 5min effort 4 years ago Code Smell Major

No tags

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Intentionality Maintainability

Open Not assigned

L12 - 5min effort 4 years ago Code Smell Major

No tags

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Intentionality Maintainability

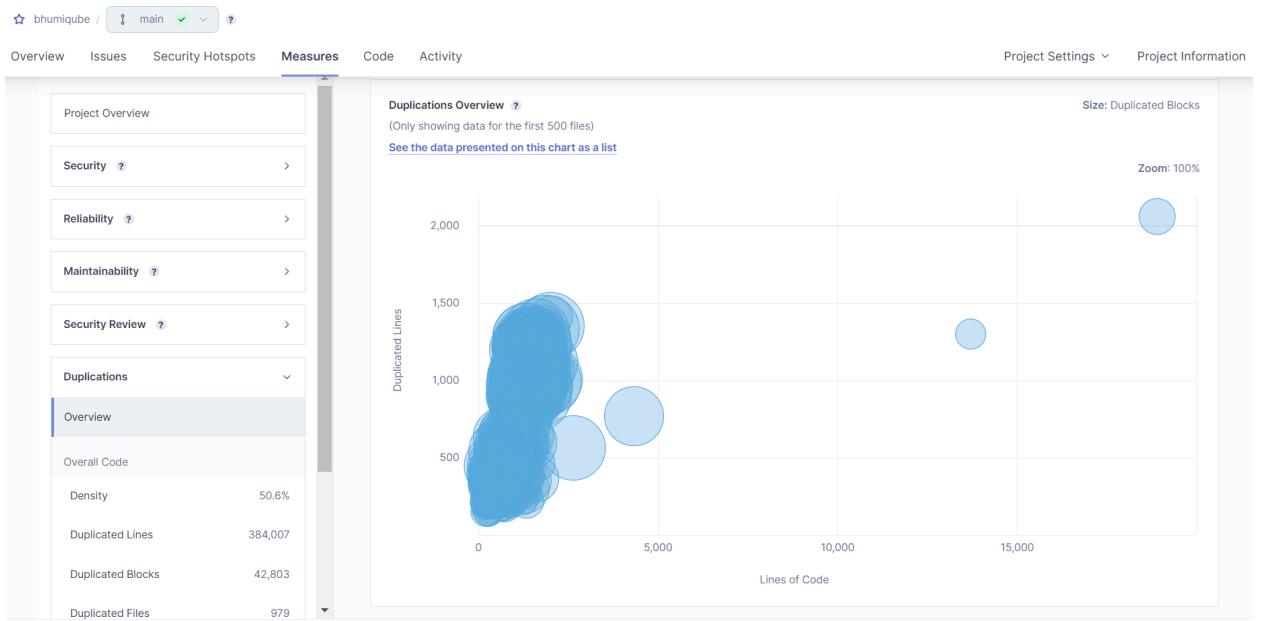
Open Not assigned

L12 - 5min effort 4 years ago Code Smell Major

No tags

Embedded database should be used for evaluation purposes only

## ● Duplications



## Conclusion:

In this experiment, we successfully integrated Jenkins with SonarQube to automate continuous monitoring of code quality within our CI/CD pipeline. The process involved deploying SonarQube using Docker, setting up a project for code analysis, and configuring Jenkins with the SonarQube Scanner plugin. After configuring the tools and providing the SonarQube server details, we developed a Jenkins pipeline that automatically clones code from GitHub and runs static analysis. This integration helps us identify bugs, code smells, and security vulnerabilities throughout the development process, ensuring better code quality and smoother development workflows.

## EXPERIMENT NO. 9

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

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

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services like CloudWatch, Systems Manager, S3, Lambda, EC2, CodePipeline, and Elastic Beanstalk. The main area has three main sections: 'Applications (0)', 'Cost and usage', and 'AWS Health'. The 'Cost and usage' section shows current month costs at \$0.00 and forecasted month end costs at \$0.00. The 'AWS Health' section has links for 'Getting started with AWS' and 'Training and certification'.

The screenshot shows the 'Security Groups' page with 14 entries. The columns are Name, Security group ID, Security group name, VPC ID, and Description. The entries include 'sg-0efffe1c9bf9a15f' (name launch-wizard-10), 'sg-0edf0ac973291ebbc' (name launch-wizard-9), 'sg-0508cc803b3cd17c9' (name default), 'sg-0ade163eb26ea436f' (name master), 'sg-0c385f9c589cd7d7d' (name launch-wizard-2), and 'sg-0a0bac122aeec771f' (name node).

Name	Security group ID	Security group name	VPC ID	Description
sg-0efffe1c9bf9a15f	launch-wizard-10	vpc-0340ce013f393caf4	launch-wizard-10 creat	
sg-0edf0ac973291ebbc	launch-wizard-9	vpc-0340ce013f393caf4	launch-wizard-9 create	
sg-0508cc803b3cd17c9	default	vpc-0340ce013f393caf4	default VPC security gr	
sg-0ade163eb26ea436f	master	vpc-0340ce013f393caf4	Security group for mas	
sg-0c385f9c589cd7d7d	launch-wizard-2	vpc-0340ce013f393caf4	launch-wizard-2 create	
sg-0a0bac122aeec771f	node	vpc-0340ce013f393caf4	Security Group for nod	

**Details**

Security group name <a href="#">nagios</a>	Security group ID <a href="#">sg-07366110359b11766</a>	Description <a href="#">nagios</a>	VPC ID <a href="#">vpc-0340ce013f393caf4</a>
Owner <a href="#">010928192223</a>	Inbound rules count 6 Permission entries	Outbound rules count 1 Permission entry	

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

**Inbound rules (6)**

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-02834ca100ea9e499	IPv4	All traffic	All	All
<input type="checkbox"/>	-	sgr-04f609e4c68588b2a	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-050341a0b8714c7...	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-0ae9c09a0dc55a21f	IPv4	Custom TCP	TCP	5666
<input type="checkbox"/>	-	sgr-057df07fa6d9b08c6	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-05eb320f1bc5e345e	IPv4	All ICMP - IPv4	ICMP	All

- Now Create a new EC2 instance. Name: nagios-host, AMI: Amazon Linux, Instance Type: t2.micro.

**Launch an instance** [Info](#)

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

**Name and tags** [Info](#)

Name  
 Add additional tags

**Application and OS Images (Amazon Machine Image)** [Info](#)

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

Recents   [Quick Start](#)

[Amazon Linux](#) [macOS](#) [Ubuntu](#) [Windows](#) [Red Hat](#) [SUSE Linux](#) [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Number of instances** [Info](#)

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.5.2...[read more](#)  
ami-0e8fd5cc56e4d158c

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
nagios

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

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

[Cancel](#) [Launch instance](#) [Review commands](#)

**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 the key pair, if you already have a key

with type RSA and extension .pem no need to create a new key but you must have that key downloaded.

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

Instance type

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true	
On-Demand SUSE base pricing: 0.0146 USD per Hour	
On-Demand Linux base pricing: 0.0146 USD per Hour	
On-Demand Windows base pricing: 0.0192 USD per Hour	
On-Demand RHEL base pricing: 0.029 USD per Hour	

All generations

[Compare instance types](#)

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](#)

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

Firewall (security groups) [Info](#)

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

Create security group  Select existing security group

Common security groups [Info](#)

Select security groups

nagios sg-07366110359b11766 X  
VPC: vpc-0340ce013f393caf4

Compare security group rules

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

▼ Configure storage [Info](#)

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

Add new volume

Click refresh to view backup information G

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...[read more](#)

ami-0e8fd5cc56e4d158c

Virtual server type (instance type)

t2.micro

Firewall (security group)

nagios

Storage (volumes)

1 volume(s) - 8 GiB

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

Cancel [Launch instance](#) Review commands

Instances (1) [Info](#)

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running X Clear filters

Launch instances

Instance state Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4 DNS

nagios-host i-0d111ce8296aa8a80 Running Initializing View alarms + ap-southeast-2a ec2-52-65-36-109.ap-s

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

**Connect to instance** Info

Connect to your instance i-0d111ce8296aa8a80 (nagios-host) using any of these options

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

Instance ID  
 [i-0d111ce8296aa8a80 \(nagios-host\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is ec2\_keypair.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "ec2\_keypair.pem"
- Connect to your instance using its Public DNS:  
 [ec2-52-65-36-109.ap-southeast-2.compute.amazonaws.com](#)

Example:  
 ssh -i "ec2\_keypair.pem" ec2-user@ec2-52-65-36-109.ap-southeast-2.compute.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.

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

```
PS C:\Users\bhumi> cd "C:\Users\bhumi\OneDrive\Desktop\New folder"
PS C:\Users\bhumi\OneDrive\Desktop\New folder> |
```

Successfully connected to the instance.

```
C:\Users\bhumi\OneDrive\Desktop\New folder>ssh -i "ec2_keypair.pem" ec2-user@ec2-52-65-36-109.ap-southeast-2.compute.amazonaws.com
The authenticity of host 'ec2-52-65-36-109.ap-southeast-2.compute.amazonaws.com (52.65.36.109)' can't be established.
ED25519 key fingerprint is SHA256:02bKhs1WIuCzaK4PY2Kj8a5S8oam+usgK/JBm3A4M6I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-65-36-109.ap-southeast-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
'___.###_          Amazon Linux 2023
~~_\###\_
~~ \###|
~~  '#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~  V~' '-->
~~  /_
~~  /_/
~~  /_/
[ec2-user@ip-172-31-40-157 ~]$ |
```

- Now Run the following command to make a new user and set the password.  
`sudo adduser -m nagios`

```
sudo passwd nagios
```

```
[ec2-user@ip-172-31-40-157 ~]$ 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:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-157 ~]$ |
```

- Now Run the following command to make a new user group.

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

```
[ec2-user@ip-172-31-40-157 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-157 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-40-157 ~]$ sudo usermod -a -G nagcmd apache
```

- Now make a new directory and go to that directory.

```
mkdir ~/downloads
cd ~/downloads
```

```
[ec2-user@ip-172-31-40-157 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-40-157 downloads]$ |
```

- Now to download the Nagios 4.5.5 and Nagios-plugins 2.4.11 run the following commands respectively.

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

```
[ec2-user@ip-172-31-40-157 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-04 22:45:39-- https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 52.54.96.194, 18.208.125.13, 3.215.172.219, ...
Connecting to go.nagios.org (go.nagios.org)|52.54.96.194|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93af2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 22:45:40-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93af2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
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|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93af2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 22:45:40-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=le9662c93af2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx          100%[=====]  1.97M  1.10MB/s    in 1.8s

2024-10-04 22:45:43 (1.10 MB/s) - '6kqcx' saved [2065473/2065473]
```

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

```
[ec2-user@ip-172-31-40-157 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 22:46:21-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz      100%[=====] 2.62M 1.16MB/s   in 2.3s
2024-10-04 22:46:24 (1.16 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

- Now to extract the files from the downloaded Nagios 4.5.5 run the following command.

```
tar zxvf 6kqcx
```

(Replace 6kqcx with your saved file name of Nagios 4.5.5 refer above screenshot)

```
[ec2-user@ip-172-31-40-157 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
```

```
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-40-157 downloads]$ |
```

- Now change the directory to nagios-4.5.5

```
cd nagios-4.5.5
```

```
[ec2-user@ip-172-31-40-157 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ |
```

10. Now run the following command to configure.

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

```
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking for shprint... yes
checking for asprintf... yes
checking for vasprintf... yes
checking for sigaction... yes
checking for C99 vsnprintf... yes
checking for library containing getservbyname... none required
checking for library containing connect... none required
checking for initgroups... yes
checking for setenv... yes
checking for strdup... yes
checking for strstr... yes
checking for strtoul... yes
checking for unsetenv... yes
checking for type of socket size... size_t
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ |
```

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

```
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ |
```

So run the following command to install ssl.

```
sudo yum install openssl-devel
```

```
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:16:22 ago on Fri Oct 4 22:35:57 2024.
Dependencies resolved.
=====
== Package           Architecture      Version       Repository      Size ==
=====
Installing:
  openssl-devel      x86_64          1:3.0.8-1.amzn2023.0.14          amazonlinux    3.0 M
Transaction Summary
=====
Install 1 Package
Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm          31 MB/s | 3.0 MB   00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          : 1/1
  Installing         : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Verifying          : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
Complete!
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ |
```

Now rerun the command

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

```
[ec2-user@ip-172-31-41-231 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
[ec2-user@ip-172-31-41-231 nagios-4.5.5]$
```

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

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

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

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

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

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

\*\*\*\*\*

Enjoy.

11. Now run the following commands to setup the Nagios.

```
sudo make install
```

```
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin;
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
```

```

*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
make install-basic
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/archives
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/spool/checkresults
chmod g+s /usr/local/nagios/var/spool/checkresults

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

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

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

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

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

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ 

```

### sudo make install-init

```

[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ 

```

### sudo make install-config

```

[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

```

### sudo make install-webconf

```

[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

```

12. To set the password

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

```
[ec2-user@ip-172-31-40-157 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-172-31-40-157 nagios-4.5.5]$ |
```

13. Now restart the httpd service and to do so, run the following command.

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ |
```

14. Now to extract the files from the downloaded Nagios plugin 2.4.11 run the following command first change the directory.

```
cd ~/downloads
```

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

```
[ec2-user@ip-172-31-40-157 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-40-157 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/_Noreturn.h
nagios-plugins-2.4.11/plugins-root/t/check_dhcp.t
nagios-plugins-2.4.11/plugins-root/t/check_icmp.t
nagios-plugins-2.4.11/po/
nagios-plugins-2.4.11/po/Makefile.in.in
nagios-plugins-2.4.11/po/remove-potcdate.sin
nagios-plugins-2.4.11/po/Makevars
nagios-plugins-2.4.11/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-40-157 downloads]$ |
```

```
nagios-plugins-2.4.11/plugins-root/t/check_dhcp.t
nagios-plugins-2.4.11/plugins-root/t/check_icmp.t
nagios-plugins-2.4.11/po/
nagios-plugins-2.4.11/po/Makefile.in.in
nagios-plugins-2.4.11/po/remove-potcdate.sin
nagios-plugins-2.4.11/po/Makevars
nagios-plugins-2.4.11/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-40-157 downloads]$ |
```

15. Now change the directory to nagios-plugins-2.4.11 and run the config command to configure.

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

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

```
[ec2-user@ip-172-31-40-157 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-40-157 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
```

```
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-40-157 nagios-plugins-2.4.11]$ |
```

16. Run the following commands to check nagios and start it. sudo chkconfig --add nagios

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

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-40-157 nagios-plugins-2.4.11]$ |
```

```

cd sudo service nagios start
[ec2-user@ip-172-31-40-157 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-40-157 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-40-157 ~]$ |

```

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

```

[ec2-user@ip-172-31-40-157 ~]$ 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.

```

sudo systemctl status nagios

```

[ec2-user@ip-172-31-40-157 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core Monitoring
   Loaded: loaded (/etc/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-09-04 23:20:25 UTC; 5min ago
     Main PID: 57764 (nagios)
       Tasks: 4 (limit: 1112)
      Memory: 4.1M
        CPU: 50ms
       CGroup: /system.slice/nagios.service
               └─57764 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
                  ├─57765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─57766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─57767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─57768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─57769 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg

Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: wproc: Registry request: name=Core Worker 57765;pid=57765
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: wproc: Registry request: name=Core Worker 57768;pid=57768
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: wproc: Registry request: name=Core Worker 57768;pid=57768
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: wproc: Registry request: name=Core Worker 57767;pid=57767
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: wproc: Registry request: name=Core Worker 57767;pid=57767
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: Successfully launched command file worker with pid 57769
Oct 04 23:20:25 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: Successfully launched command file worker with pid 57769
Oct 04 23:20:55 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(/usr/local/nagios
Oct 04 23:21:55 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(/usr/local/nagios
Oct 04 23:22:55 ip-172-31-40-157.ap-southeast-2.compute.internal nagios[57764]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(/usr/local/nagio
[ec2-user@ip-172-31-40-157 ~]$ |

```

17. We can see we have successfully launched Nagios now .

Open <http://nagios/> here it is <http://<ip address>/nagios> and we can see the running web page of nagios.

The screenshot shows the Nagios Core web interface. At the top, there's a navigation bar with icons for back, forward, search, and other browser functions. The URL bar shows "Not secure 13.211.113.220/nagios/". The main header features the "Nagios® Core™" logo with a gear icon. A green checkmark indicates "Daemon running with PID 74291". Below the header, the version information is displayed: "Nagios® Core™ Version 4.5.5" from "September 17, 2024" with a link to "Check for updates". On the left side, there's a sidebar with several sections: "General" (Home, Documentation), "Current Status" (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems), "Reports" (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and "System" (Comments, Downtime, Process Info, Performance Info). The main content area has several boxes: "Get Started" with a bulleted list of steps, "Latest News" (empty), "Don't Miss..." (empty), and "Quick Links" with links to Nagios Library, Labs, Exchange, Support, and the company website. At the bottom, there's a copyright notice: "Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors." A vertical "Page Tour" button is located on the right edge of the interface.

Conclusion: In this experiment, we have set up the Nagios core with plugins on Amazon Linux. Which will help us to understand Continuous monitoring and Installation. It is important to note that whatever set of rules we have added in step 1 are very important for this experiment.

## EXPERIMENT NO. 10

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

### 1. To Confirm Nagios is running on the server side

Perform the following command on your Amazon Linux Machine (Nagios-host).

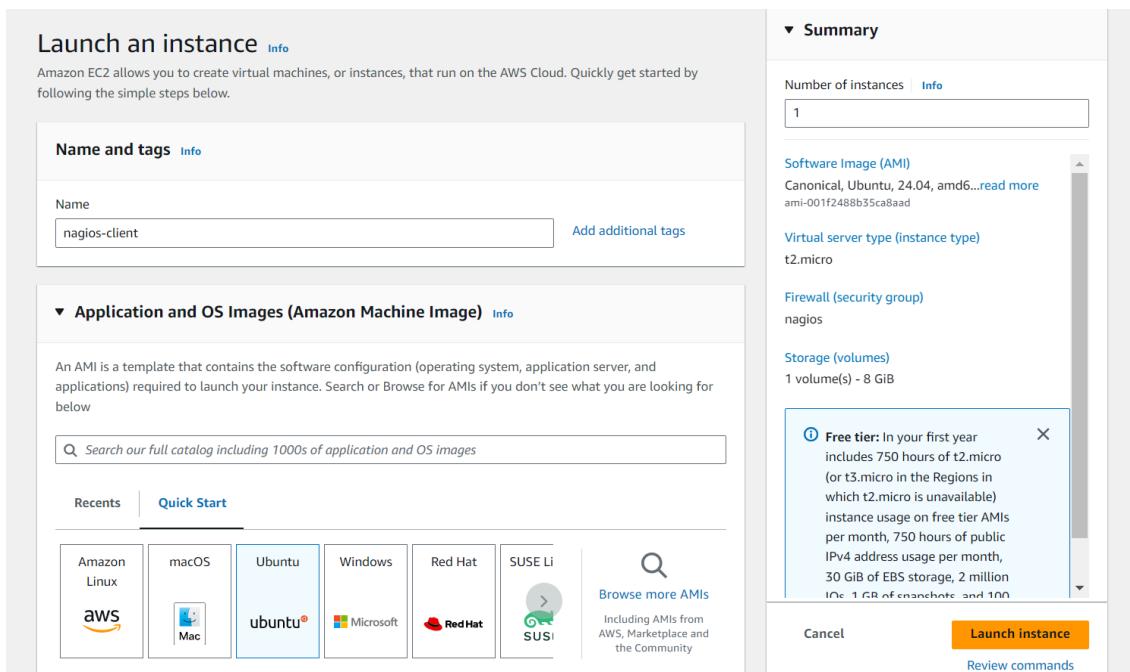
```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-13-224 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/etc/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-05 19:22:57 UTC; 2min 1s ago
     Main PID: 74867 (nagios)
        Tasks: 6 (limit: 1112)
       Memory: 5.6M
          CPU: 233ms
      CGroup: /system.slice/nagios.service
              ├─74867 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
              ├─74868 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─74869 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─74870 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─74871 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─74872 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg

Oct 05 19:22:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: wproc: Registry request: name=Core Worker 74869;pid=>
Oct 05 19:22:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: wproc: Registry request: name=Core Worker 74868;pid=>
Oct 05 19:22:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: Successfully launched command file worker with pid 7>
Oct 05 19:22:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: Successfully launched command file worker with pid 7>
Oct 05 19:22:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdo>
Oct 05 19:23:34 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;>
Oct 05 19:23:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdo>
Oct 05 19:24:12 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: SERVICE ALERT: localhost;Current Users;CRITICAL;HARD;>
Oct 05 19:24:49 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;1;(No ou>
Oct 05 19:24:57 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[74867]: HOST ALERT: localhost;DOWN;SOFT;3;(No output on stdo>
[lines 1-25/25 (END)]
```

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

### 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 the key pair if you already have a key with type RSA and extension .pem no need to create a 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).

- 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\bhumit> cd "C:\Users\bhumit\OneDrive\Desktop\New folder"
PS C:\Users\bhumit\OneDrive\Desktop\New folder> ssh -i "disco.pem" ubuntu@ec2-3-25-84-91.ap-southeast-2.compute.amazonaws.com
The authenticity of host 'ec2-3-25-84-91.ap-southeast-2.compute.amazonaws.com (3.25.84.91)' can't be established.
ED25519 key fingerprint is SHA256:zIXx3ATmWCr9U4e6inijfig4VG+Bji+Xm8lz0cDdHCC.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-25-84-91.ap-southeast-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

System information as of Sat Oct 5 18:51:05 UTC 2024

System load: 0.0 Processes: 105
Usage of /: 23.0% of 6.71GB Users logged in: 0
Memory usage: 20% IPv4 address for enX0: 172.31.5.17
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

Last login: Sat Oct 5 18:26:23 2024 from 13.239.158.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Successfully connected to the instance.

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

- Now on the server Nagios-host run the following command.

```
ps -ef | grep nagios
```

```
[ec2-user@ip-172-31-13-224 ~]$ ps -ef | grep nagios
nagios 74867 1 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
nagios 74868 74867 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 74869 74867 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 74870 74867 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 74871 74867 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 74872 74867 0 19:22 ? 00:00:00 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
ec2-user 75110 75013 0 19:26 pts/1 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-13-224 ~]$ |
```

- 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 75110 75013 0 19:26 pts/1 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-13-224 ~]$ sudo su
[root@ip-172-31-13-224 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-13-224 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-13-224 ec2-user]# |
```

- 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-172-31-13-224 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-13-224 ec2-user]# |
```

7. Open linuxserver.cfg using nano and make the following changes everywhere in the file.  
*Change hostname to linuxserver.*  
*Change address to the public IP of your Linux client.*  
*Set hostgroup\_name to linux-servers1.*

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

```
[root@ip-172-31-13-224 ec2-user]# sudo sed -i 's/^ *host_name.*/ host_name      linuxserver/' /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
sudo sed -i 's/^ *address.*/   address      3.25.70.155/' /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
sudo sed -i 's/^ *hostgroup_name.*/ hostgroup_name  linux-servers1/' /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-13-224 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-13-224 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg
```

```
#####
#  
# 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       3.25.70.155  
}
```

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

*Line to add : cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/*  
Run the command : nano /usr/local/nagios/etc/nagios.cfg

```
# 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/|
```

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

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

```
[root@ip-172-31-13-224 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 16 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 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
```

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

```
service nagios restart
```

```
[root@ip-172-31-13-224 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-13-224 ec2-user]# |
```

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-172-31-5-17:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:14 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:15 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:16 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
Get:17 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [159 kB]
```

```

Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2:4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-5-17:~$ |

```

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

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

```

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,3.25.70.155

```

13. Now restart the NRPE server by this command.

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

```

ubuntu@ip-172-31-5-17:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-5-17:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-5-17:~$ |

```

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 activate it.

```
sudo systemctl status nagios
```

```
[root@ip-172-31-13-224 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/etc/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-05 19:42:04 UTC; 22min ago
     Main PID: 76185 (nagios)
       Tasks: 6 (limit: 1112)
      Memory: 4.1M
        CPU: 252ms
      CGroup: /system.slice/nagios.service
              └─76185 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg

Oct 05 19:45:11 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: SERVICE ALERT: linuxserver;Root Partition;CRITICAL;HARD;1;(No op
Oct 05 19:45:49 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: SERVICE ALERT: linuxserver;SSH;CRITICAL;HARD;1;(No output on std
Oct 05 19:46:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;SOFT;5;(No output on stdout) std
Oct 05 19:46:26 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: SERVICE ALERT: linuxserver;Swap Usage;CRITICAL;HARD;1;(No ou
Oct 05 19:47:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: SERVICE ALERT: linuxserver;Total Processes;CRITICAL;HARD;1;(No >
Oct 05 19:47:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;SOFT;6;(No output on stdout) std
Oct 05 19:48:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;SOFT;7;(No output on stdout) std
Oct 05 19:49:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;SOFT;8;(No output on stdout) std
Oct 05 19:50:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;SOFT;9;(No output on stdout) std
Oct 05 19:51:04 ip-172-31-13-224.ap-southeast-2.compute.internal nagios[76185]: HOST ALERT: linuxserver;DOWN;HARD;10;(No output on stdout) std
lines 1-25/25 (END)
```

**sudo systemctl status httpd**

**sudo systemctl start httpd**

**sudo systemctl enable httpd**

```
[root@ip-172-31-13-224 ec2-user]# sudo systemctl status httpd
sudo systemctl start httpd
sudo systemctl enable httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Sat 2024-10-05 19:12:34 UTC; 51min ago
       Docs: man:httpd.service(8)
     Main PID: 49647 (httpd)
       Status: "Total requests: 24; Idle/Busy workers 100/0;Requests/sec: 0.00772; Bytes served/sec: 52 B/sec"
       Tasks: 230 (limit: 1112)
      Memory: 21.5M
        CPU: 1.806s
      CGroup: /system.slice/httpd.service
              ├─49647 /usr/sbin/httpd -DFOREGROUND
              ├─49654 /usr/sbin/httpd -DFOREGROUND
              ├─49655 /usr/sbin/httpd -DFOREGROUND
              ├─49656 /usr/sbin/httpd -DFOREGROUND
              ├─49657 /usr/sbin/httpd -DFOREGROUND
              └─77500 /usr/sbin/httpd -DFOREGROUND

Oct 05 19:12:34 ip-172-31-13-224.ap-southeast-2.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 05 19:12:34 ip-172-31-13-224.ap-southeast-2.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 05 19:12:34 ip-172-31-13-224.ap-southeast-2.compute.internal httpd[49647]: Server configured, listening on: port 80
[root@ip-172-31-13-224 ec2-user]#
```

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

Now Click on Hosts from left side panel

The screenshot shows the Nagios web interface at the URL 3.25.70.155/nagios/. The left sidebar is titled 'Current Status' and includes links for 'Tactical Overview', 'Map', 'Hosts', 'Services', 'Host Groups', 'Service Groups', and 'Problems'. The main content area displays 'Current Network Status' with last updated time as Sat Oct 5 21:17:24 UTC 2024. It shows 'Host Status Totals' with 2 Up, 0 Down, 0 Unreachable, and 0 Pending hosts. The 'Service Status Totals' shows 7 Ok, 1 Warning, 0 Unknown, 8 Critical, and 0 Pending services. Below this is a table titled 'Host Status Details For All Host Groups' with two entries: 'linuxserver' and 'localhost', both marked as 'UP'. The results table shows 1 of 2 matching hosts.

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

The screenshot shows the detailed host information for 'localhost' (linuxserver). The left sidebar includes 'Reports' and 'System' sections. The main content area shows 'Host Information' with details like last updated time, version, and status as 'UP'. It also shows 'Host State Information' with various metrics and status indicators. On the right, there is a 'Host Commands' section with a list of actions such as 'Locate host on map', 'Disable active checks of this host', and 'Stop accepting passive checks for this host'. At the bottom, there is a 'Host Comments' section with a link to 'Add a new comment'.

## Current Network Status

Not secure 3.25.70.155/nagios/

Host		Service		Status	Last Check	Duration	Attempt	Status Information	
linuxserver	Current Load	OK		OK	10-05-2024 21:17:41	0d 0h 2m 1s	1/4	OK - load average: 0.03, 0.13, 0.07	
	Current Users	OK		OK	10-05-2024 21:18:19	0d 0h 1m 23s	1/4	USERS OK - 6 users currently logged in	
	HTTP	<span style="color: red;">☒</span>	WARNING	WARNING	10-05-2024 21:18:56	0d 0h 0m 46s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time	
	PING	OK		OK	10-05-2024 21:19:34	0d 0h 5m 8s	1/4	PING OK - Packet loss = 0%, RTA = 0.66 ms	
	Root Partition	OK		OK	10-05-2024 21:15:11	0d 0h 4m 31s	1/4	DISK OK - free space / 5568 MB (68.61% inode=98%).	
	SSH	<span style="color: red;">☒</span>	OK	OK	10-05-2024 21:15:49	0d 0h 3m 53s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)	
	Swap Usage	<span style="color: red;">☒</span>	Critical	Critical	10-05-2024 21:16:26	0d 1h 33m 16s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.	
	Total Processes	OK		OK	10-05-2024 21:17:04	0d 0h 2m 38s	1/4	PROCS OK: 42 processes with STATE = R/SZDT	
localhost	Current Load	OK		OK	10-05-2024 21:18:34	0d 0h 1m 8s	1/4	OK - load average: 0.24, 0.17, 0.09	
	Current Users	OK		OK	10-05-2024 21:19:12	0d 0h 0m 30s	1/4	USERS OK - 6 users currently logged in	
	HTTP	<span style="color: red;">☒</span>	WARNING	WARNING	10-05-2024 21:14:49	0d 0h 4m 53s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time	
	PING	OK		OK	10-05-2024 21:15:27	0d 0h 4m 15s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms	
	Root Partition	OK		OK	10-05-2024 21:16:04	0d 0h 3m 38s	1/4	DISK OK - free space / 5568 MB (68.61% inode=98%).	
	SSH	<span style="color: red;">☒</span>	OK	OK	10-05-2024 21:16:42	0d 0h 3m 0s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)	
	Swap Usage	<span style="color: red;">☒</span>	Critical	Critical	10-05-2024 21:17:19	0d 1h 52m 23s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.	
	Total Processes	OK		OK	10-05-2024 21:17:57	0d 0h 1m 45s	1/4	PROCS OK: 42 processes with STATE = R/SZDT	

Results 1 - 16 of 16 Matching Services

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

## EXPERIMENT NO. 11

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

### Steps To create the lambda function:

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

The screenshot shows the AWS Lambda console homepage. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and account information ('Sydney' and 'bhumishap'). Below the header, a large banner features the text 'AWS Lambda' and 'lets you run code without thinking about servers.' It includes a 'Get started' section with a 'Create a function' button. A note below states: 'You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.' In the main content area, there's a 'How it works' section with tabs for '.NET', 'Java', 'Node.js' (which is selected), 'Python', 'Ruby', and 'Custom runtime'. A code snippet for Node.js is shown:

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

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

2. Now Give a name to your Lambda function.

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

The screenshot shows the 'Create function' wizard. It starts with a 'Basic information' step. Under 'Function name', the user has entered 'bhumi-lambda'. Under 'Runtime', 'Python 3.12' is selected. There are three options for creating the function: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Container image' option is described as 'Select a container image to deploy for your function.'

**Architecture** [Info](#)  
 Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

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

▼ Change default execution role

**Execution role**  
 Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
 Create a new role with basic Lambda permissions  
 Use an existing role  
 Create a new role from AWS policy templates

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

Lambda will create an execution role named bhumilambda-role-hzeu5hul, with permission to upload logs to Amazon CloudWatch Logs.

► Advanced settings

Cancel **Create function**

Thus, we have successfully created a lambda function named bhumilambda.

⌚ Successfully created the function **bhumilambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test". [X](#)

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

### bhumilambda

**Function overview** [Info](#)

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

 bhumilambda

 Layers (0)

+ Add trigger + Add destination

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

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

Description -

Last modified 7 seconds ago

Function ARN  arn:aws:lambda:ap-southeast-2:010928192223:function:bhumilambda

Function URL [Info](#) -

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

**Tutorials** [X](#)

Learn how to implement common use cases in AWS Lambda.

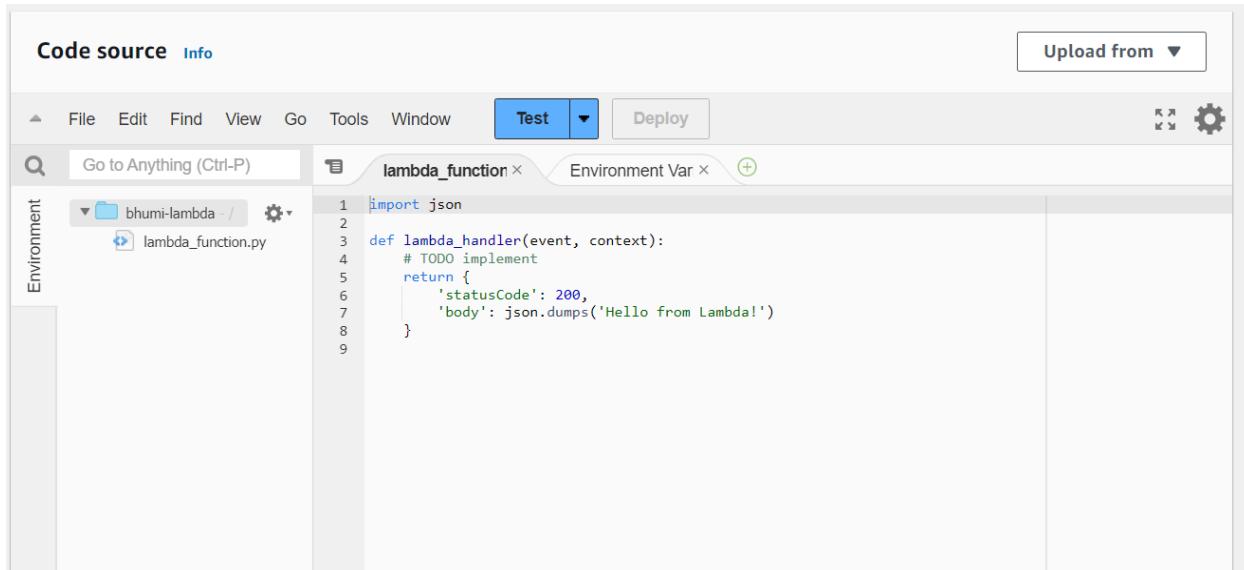
**Create a simple web app** [^](#)

In this tutorial you will learn how to:

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

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

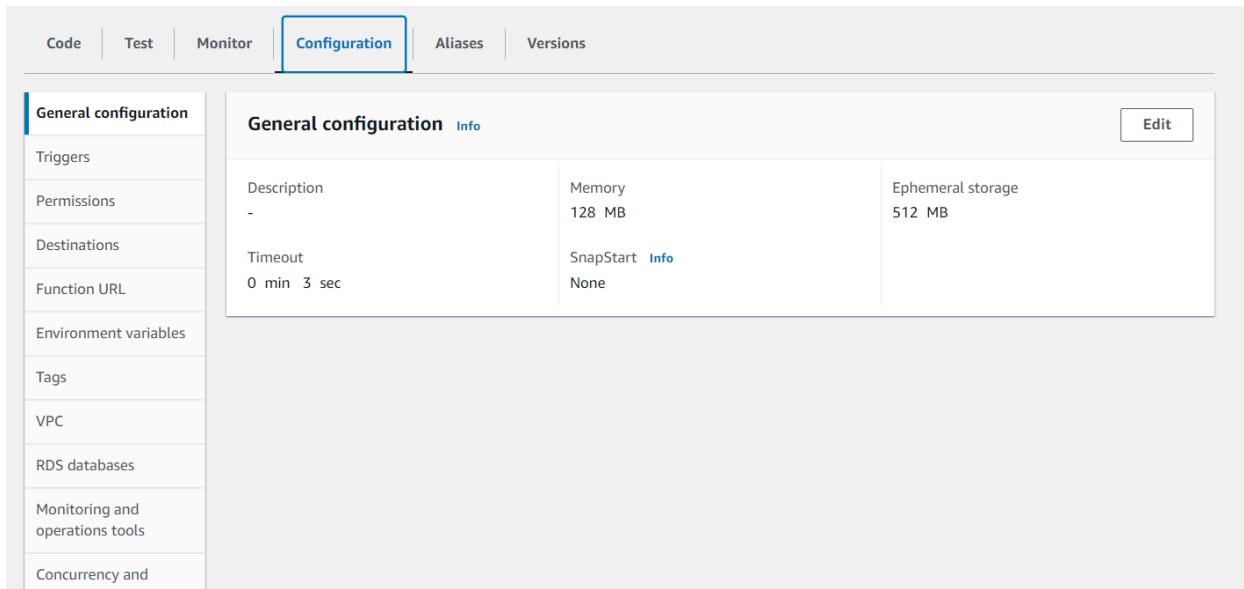
This is how the function is displayed.



The screenshot shows the AWS Lambda code editor interface. At the top, there are tabs for 'Code source' and 'Info', and a button 'Upload from'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, and buttons for 'Test' (which is highlighted in blue) and 'Deploy'. To the right of the menu are zoom and settings icons. The main area has a search bar 'Go to Anything (Ctrl-P)' and tabs for 'lambda\_function' and 'Environment Var'. On the left, there's a sidebar labeled 'Environment' with a tree view showing a folder 'bhumi-lambda' containing 'lambda\_function.py'. The code editor itself displays 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
```

Go to the Configuration tab to see the general configuration of our function.



The screenshot shows the AWS Lambda function configuration page. At the top, there are tabs for Code, Test, Monitor, Configuration (which is highlighted in blue), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, and Concurrency and. The main content area is titled 'General configuration' with an 'Info' link and an 'Edit' button. It shows the following configuration details:

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

We want to edit the timeout time and the rest can be kept the default.

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

To edit, we go to the Edit button seen in the above screenshot.

## Edit basic settings

**Basic settings** [Info](#)

Description - *optional*

**Memory** [Info](#)  
Your function is allocated CPU proportional to the memory configured.  
 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](#)  
 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](#).  
▼  
Supported runtimes: Java 11, Java 17, Java 21.

**Timeout**  
 min  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

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

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

**Test event** [Info](#) [Save](#) [Test](#)

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

Test event action  
 Create new event  Edit saved event

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

Event sharing settings  
 Private  
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)  
 Shareable  
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - *optional*

**Event JSON** [Format JSON](#)

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

**Test event** [Info](#)

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

Test event action

Create new event  Edit saved event

Event name

bhumi-event

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

Event sharing settings

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

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

Template - optional

hello-world

**Event JSON**

1 ↴  
2 "key1": "value1",  
3 "key2": "value2",  
4 "key3": "value3"  
5 ↴

[Format JSON](#)

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

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

**Code source** [Info](#)

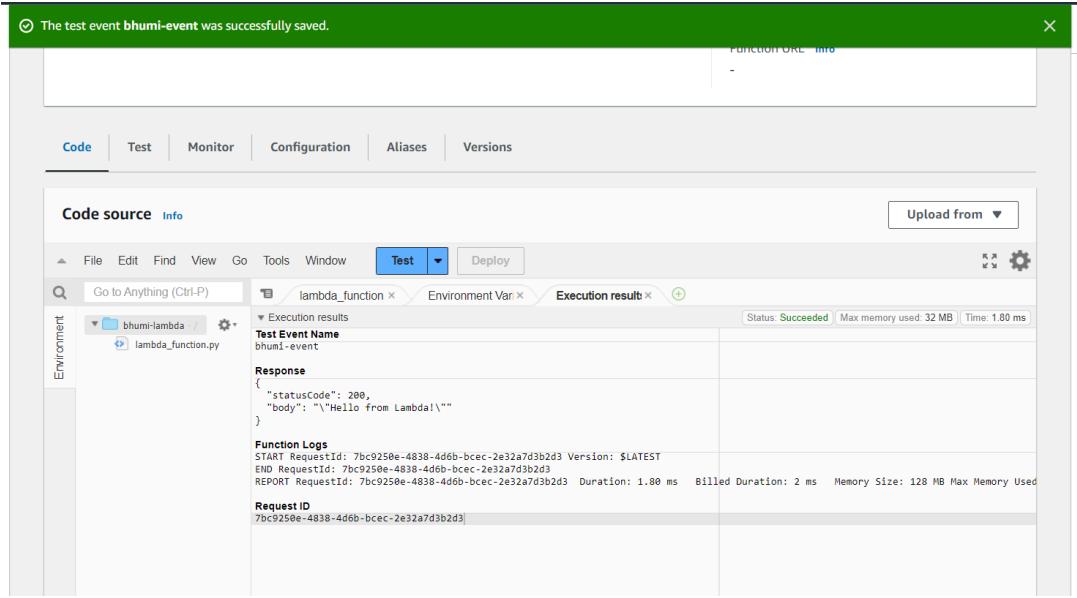
[Upload from](#)

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

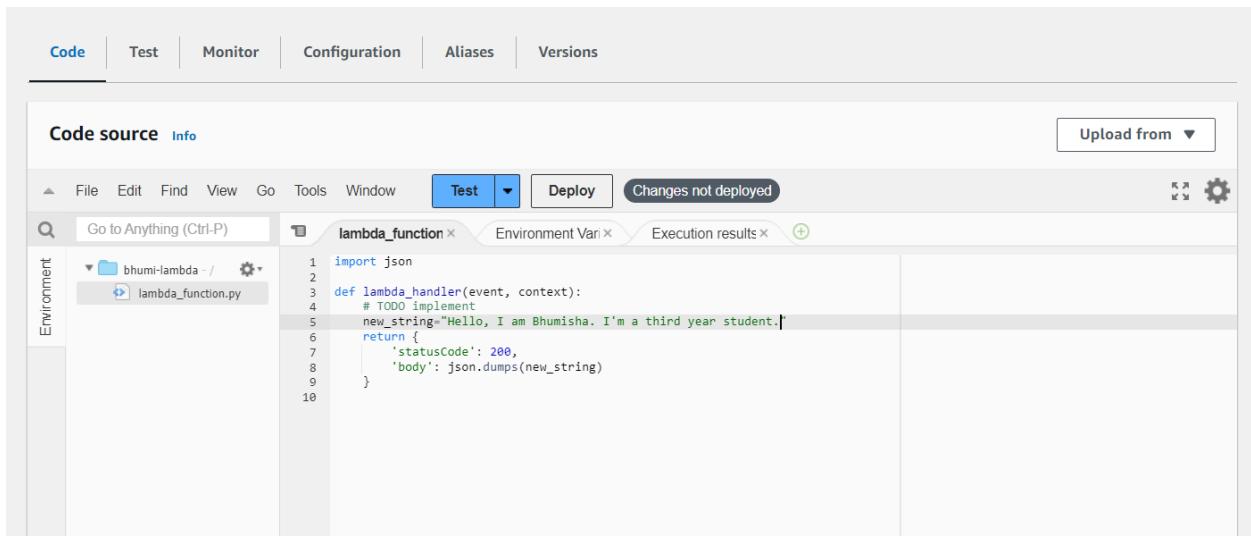
Configure test event Ctrl-Shift-C

Environment

```
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
```



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



Now **ctrl+s** to save and click on **deploy** to deploy the changes.

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a gear icon. On the left, there's a sidebar for Environment variables and a search bar labeled 'Go to Anything (Ctrl-P)'. The main area displays the code for 'lambda\_function.py':

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string="Hello, I am Bhumisha. I'm a third year student."
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10
```

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

The screenshot shows the AWS Lambda console interface after a test execution. The Test tab is selected. The Execution result panel shows the following details:

- Test Event Name: bhumi-event
- Status: Succeeded
- Max memory used: 32 MB
- Time: 1.53 ms
- Response (JSON):

```
{
    "statusCode": 200,
    "body": "\"Hello, I am Bhumisha. I'm a third year student.\""
}
```
- Function Logs:

```
START RequestId: db6d7b36-9858-4fcf-84aa-4f4eb0a8735d Version: $LATEST
END RequestId: db6d7b36-9858-4fcf-84aa-4f4eb0a8735d
REPORT RequestId: db6d7b36-9858-4fcf-84aa-4f4eb0a8735d Duration: 1.53 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```
- Request ID: db6d7b36-9858-4fcf-84aa-4f4eb0a8735d

Conclusion: In this experiment, we successfully created an AWS Lambda function and walked through its essential steps. After setting up the function with Python, we configured the basic settings, including adjusting the timeout to 1 second. We then created a test event, deployed the function, and validated the output. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time.

This practical experience demonstrated the simplicity and flexibility of AWS Lambda in creating serverless applications, allowing you to focus on code while AWS manages the infrastructure and scaling.

## EXPERIMENT NO. 12

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

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

The screenshot shows the Amazon S3 console. At the top, there is a header with the text "Amazon S3". Below it is a banner with the message "Account snapshot - updated every 24 hours" and a link to "All AWS Regions". On the right side of the banner is a button labeled "View Storage Lens dashboard". Underneath the banner, there are two tabs: "General purpose buckets" (which is selected) and "Directory buckets". A search bar with the placeholder "Find buckets by name" is located below the tabs. To the right of the search bar are navigation icons: back, forward, and refresh. Below the search bar is a table with the following columns: Name, AWS Region, IAM Access Analyzer, and Creation date. The table contains one row with the bucket name "elasticbeanstalk-ap-southeast-2-010928192223", AWS Region "Asia Pacific (Sydney) ap-southeast-2", IAM Access Analyzer link "View analyzer for ap-southeast-2", and Creation date "August 16, 2024, 15:31:00 (UTC+05:30)". Above the table are several buttons: "Create bucket" (orange), "Copy ARN", "Empty", and "Delete".

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

The screenshot shows the "Create bucket" wizard in the Amazon S3 console. The first step, "General configuration", is active. It includes fields for "Bucket name" (with "pawn" typed in), "AWS Region" (set to "Asia Pacific (Sydney) ap-southeast-2"), and "Object Ownership". The "Object Ownership" section has two options: "ACLs disabled (recommended)" (selected) and "ACLs enabled". The "ACLs disabled" option is described as "All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies." The "ACLs enabled" option is described as "Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs." Below the "Object Ownership" section, there is a note: "Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects." At the bottom of the page, there are "Next Step" and "Cancel" buttons.

**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.

**⚠ 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.

**General purpose buckets (2) [Info](#) [All AWS Regions](#)**

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">elasticbeanstalk-ap-southeast-2-010928192223</a>	Asia Pacific (Sydney) ap-southeast-2	<a href="#">View analyzer for ap-southeast-2</a>	August 16, 2024, 15:31:00 (UTC+05:30)
<a href="#">pawn</a>	Asia Pacific (Sydney) ap-southeast-2	<a href="#">View analyzer for ap-southeast-2</a>	October 6, 2024, 03:13:53 (UTC+05:30)

Thus, we have created a bucket named *pawn*.

### 3. Open lambda console and click on the create function button.

The screenshot shows the AWS Lambda console with the following interface elements:

- Services** tab selected in the top navigation bar.
- Compute** category selected in the sidebar.
- AWS Lambda** heading with the tagline "lets you run code without thinking about servers."
- Get started** callout box: "Author a Lambda function from scratch, or choose from one of many preconfigured examples." with a **Create a function** button.
- How it works** section: "Run" button and "Next: Lambda responds to events" link.
- Run code** section: "Run" button and "Next: Lambda responds to events" link.
- Run code** section: ".NET" runtime selected, with tabs for Java, Node.js, Python, Ruby, and Custom runtime.
- Code editor**: A snippet of Node.js code:

```

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

```
- CloudShell** and **Feedback** links at the bottom.

The screenshot shows the AWS Lambda console with the following interface elements:

- How it works** section: "Run" button and "Next: Lambda responds to events" link.
- Run code** section: ".NET" runtime selected, with tabs for Java, Node.js, Python, Ruby, and Custom runtime.
- Code editor**: A snippet of Node.js code:

```

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

```
- CloudShell** and **Feedback** links at the bottom.

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

Lambda > Functions > Create function

### Create function Info

Choose one of the following options to create your function.

- Author from scratch  
Start with a simple Hello World example.
- Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image  
Select a container image to deploy for your function.

#### Basic information

Function name Info  
Enter a name that describes the purpose of your function.  
**pawn-lambda**  
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.  
**Python 3.12**

Architecture Info  
Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

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

► Change default execution role

► Advanced settings

Cancel **Create function**

Thus, we have successfully created a lambda function named *pawn-lambda*.

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

[Lambda](#) > [Functions](#) > **pawn-lambda**

### pawn-lambda

**Function overview** Info

**Diagram** **Template**

**Layers** (0)

**Add trigger** **Add destination**

**Description**  
-

**Last modified**  
17 seconds ago

**Function ARN**  
arn:aws:lambda:ap-southeast-2:010928192223:function:pawn-lambda

**Function URL** Info  
-

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

**Code source** Info

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

Upload from **Import**

Go to Anything (Ctrl-P) **lambda\_function** Environment Vari **Import.json**

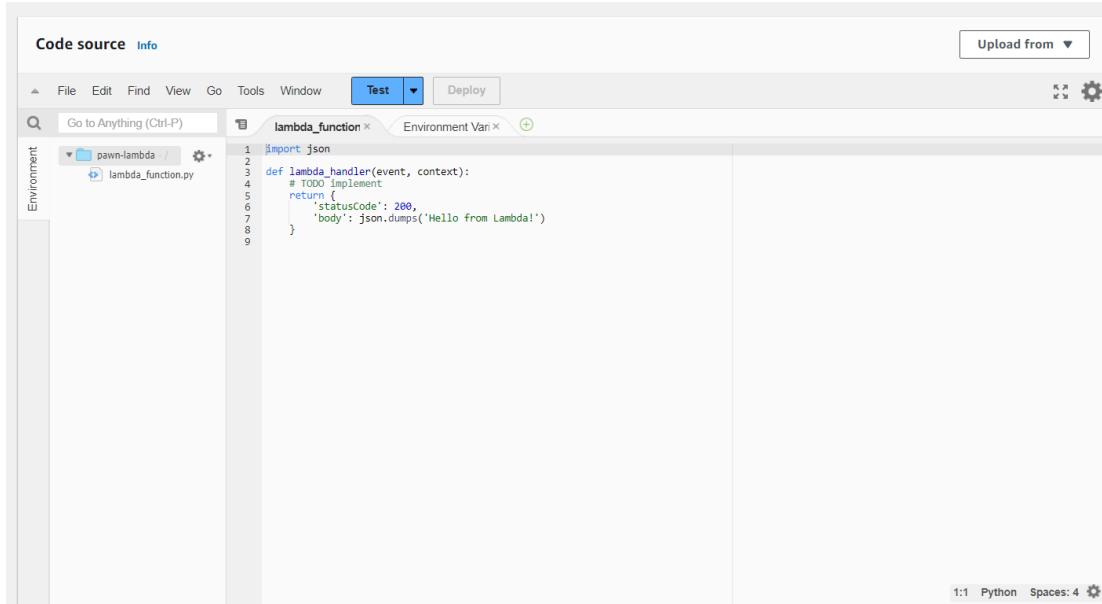
**Create a simple web app**

In this tutorial you will learn how to:

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

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

This is how the function is displayed.

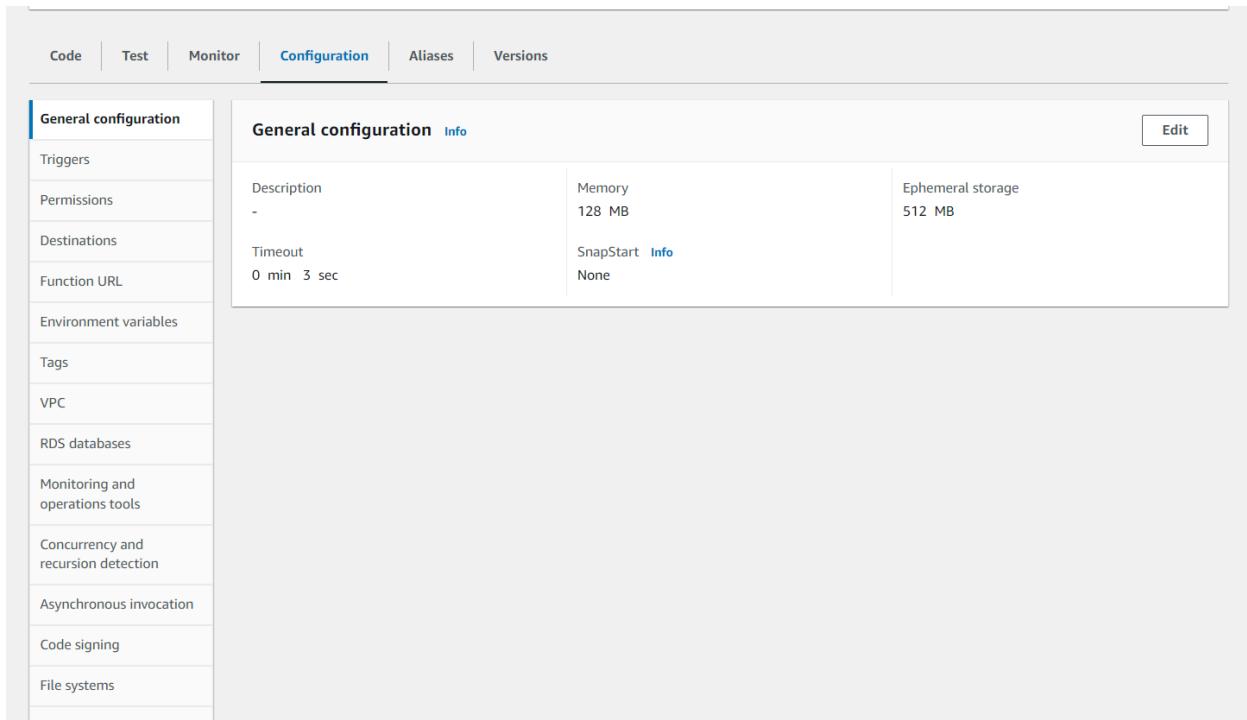


The screenshot shows the AWS Lambda code editor interface. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is currently selected), and 'Deploy'. On the right, there's an 'Upload from' button and a gear icon. The left sidebar has an 'Environment' section and a search bar 'Go to Anything (Ctrl-P)'. The main area displays the 'lambda\_function' file under the 'lambda\_function.py' tab. The code is as follows:

```
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
```

At the bottom right, it shows a 1:1 Python Spaces: 4 ratio.

Go to the Configuration tab to see the general configuration of our function.



The screenshot shows the AWS Lambda function configuration page. The top navigation bar has tabs for 'Code', 'Test', 'Monitor', 'Configuration' (which is selected and highlighted in blue), 'Aliases', and 'Versions'. On the left, a sidebar lists various configuration sections: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, and File systems. The main content area is titled 'General configuration' with an 'Info' link and an 'Edit' button. It shows the following settings:

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

We want to edit the timeout time and the rest can be kept the default.

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

To edit, we go to the Edit button seen in the above screenshot.

## Edit basic settings

### Basic settings Info

Description - *optional*

#### 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](#).

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

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

### Test event Info

Save

Test

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

#### Test event action

Create new event

Edit saved event

#### Event name

pawn-bucket

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

#### Event sharing settings

Private

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

Shareable

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

#### Template - *optional*

s3-put

Event JSON

Format JSON

Event JSON

```

1 * []
2 *   "Records": [
3 *     {
4       "eventVersion": "2.0",
5       "eventSource": "aws:s3",
6       "awsRegion": "us-east-1",
7       "eventTime": "1970-01-01T00:00:00.000Z",
8       "eventName": "ObjectCreated:Put",
9       "userIdentity": {
10         "principalId": "EXAMPLE"
11       },
12       "requestParameters": {
13         "sourceIPAddress": "127.0.0.1"
14       },
15       "responseElements": {
16         "x-amz-request-id": "EXAMPLE123456789",
17         "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGHIJ"
18       },
19       "s3": {
20         "s3SchemaVersion": "1.0",
21         "configurationId": "testConfigRule",
22         "bucket": {
23           "name": "example-bucket",
24           "ownerIdentity": {
25             "principalId": "EXAMPLE"
26           },
27           "arn": "arn:aws:s3:::example-bucket"
28         },
29         "object": {
30           "key": "test%2Fkey",
31         }
32       }
33     }
34   ]
35 
```

Format JSON

1:1 JSON Spaces: 2

6. Now in the Code section select the created event from the dropdown.

**Code** Test Monitor Configuration Aliases Versions

**Code source** Info

Upload from ▾

Test ▾ Deploy

Configure test event Ctrl-Shift-C

Go to Anything (Ctrl-P)

Environment pawn-lambda λ lambda\_function.py

```

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

7. Now In the Lambda function click on add trigger.

Lambda > Functions > pawn-lambda

pawn-lambda

Throttle Copy ARN Actions ▾

Function overview Info

Export to Application Composer Download ▾

Diagram Template

pawn-lambda

Layers (0)

+ Add trigger + Add destination

Description

Last modified 3 minutes ago

Function ARN arn:aws:lambda:ap-southeast-2:010928192223:function:pawn-lambda

Function URL Info

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

**Add trigger**

**Trigger configuration** [Info](#)

**Bucket**  
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.  
 [Copy](#)

**Bucket region:** ap-southeast-2

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

[X](#)

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

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

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocation.

**Lambda > Functions > pawn-lambda**

**pawn-lambda**

The trigger pawn was successfully added to function pawn-lambda. The function is now receiving events from the trigger.

**Function overview** [Info](#)

**Diagram** [Template](#)

**Triggers** [Info](#)

**S3**

**Layers** (0)

**Add destination**

**Add trigger**

**Description**

Last modified 5 minutes ago

Function ARN [arn:aws:lambda:ap-southeast-2:2010928192223:function:pawn-lambda](#)

Function URL [Info](#)

**Configuration**

**Triggers (1) [Info](#)**

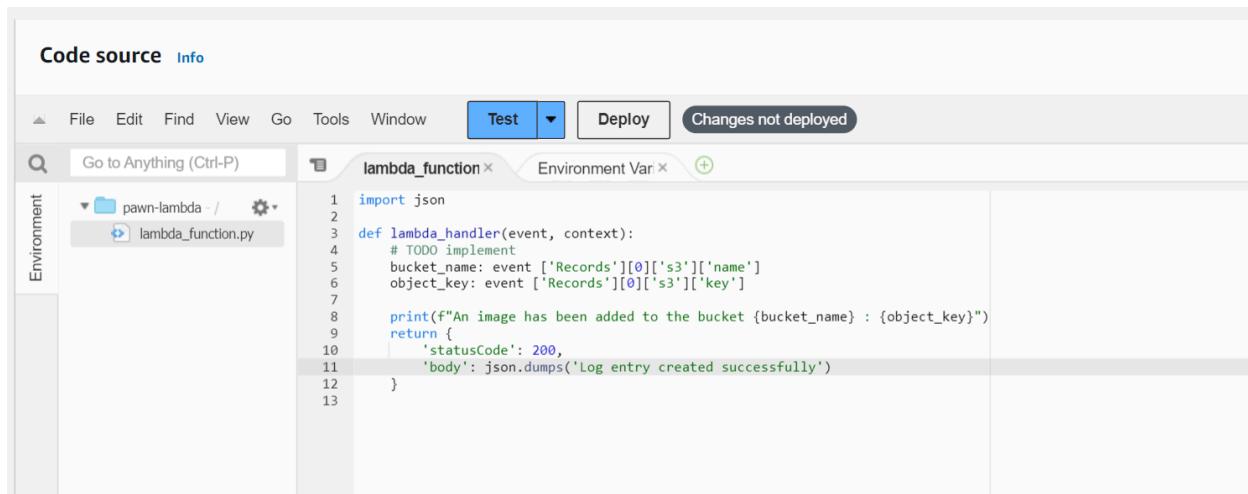
**Find triggers**

**Trigger**

**S3: pawn** [arn:aws:s3:::pawn](#)

**Details**

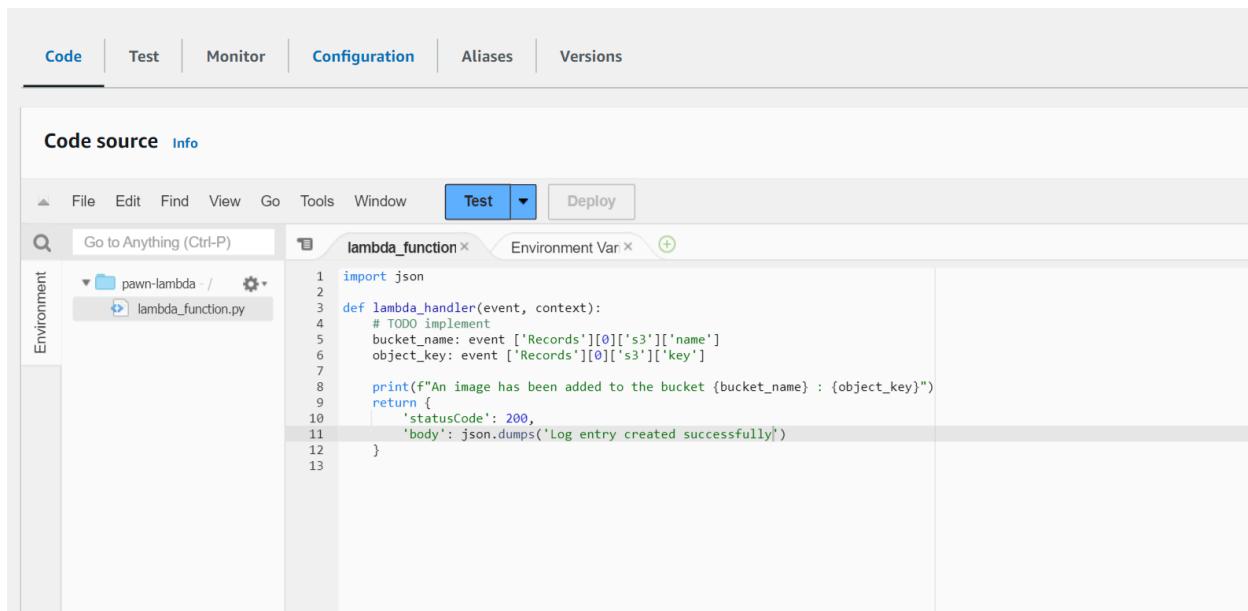
8. Now Write code that logs a message like “An Image has been added” when triggered. Save the file and click on deploy.



The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message 'Changes not deployed'. Below the navigation is a search bar 'Go to Anything (Ctrl-P)'. A sidebar on the left labeled 'Environment' shows a folder 'pawn-lambda - /' containing 'lambda\_function.py'. The main workspace displays the Python code for a lambda function:

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

Below the workspace, a navigation bar offers Code, Test, Monitor, Configuration, Aliases, and Versions. The Configuration tab is currently selected.



The screenshot shows the AWS Lambda function editor interface with the Configuration tab selected. The top navigation bar and sidebar are identical to the previous screenshot. The main workspace displays the same Python code as before. The bottom navigation bar now shows Configuration as the active tab.

9. Now upload any image to the bucket.

Amazon S3 > Buckets > pawn > Upload

## Upload [Info](#)

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

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

Files and folders (0)	
All files and folders in this table will be uploaded.	
<input type="text"/> Find by name	
<input type="checkbox"/>	Name
< 1 >	
No files or folders	
You have not chosen any files or folders to upload.	

**Destination** [Info](#)

Destination  
[s3://pawn](#)

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

[Permissions](#)

## 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, 77.3 KB)	
All files and folders in this table will be uploaded.	
<input type="text"/> Find by name	
<input type="checkbox"/>	Name
<input type="checkbox"/>	Screenshot 2024-09-10 220423.png
< 1 >	

**Destination** [Info](#)

Destination  
[s3://pawn](#)

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

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

The screenshot shows the AWS S3 console with a green header bar indicating "Upload succeeded". Below it, a modal window titled "Upload: status" displays the following information:

- Summary** table:
 

Destination s3://pawn	Succeeded 1 file, 77.3 KB (100.00%)	Failed 0 files, 0 B (0%)
--------------------------	--	-----------------------------
- Files and folders** tab selected, showing a table with one item:
 

Name	Folder	Type	Size	Status	Error
Screenshot 2...	-	image/png	77.3 KB	Succeeded	-

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

The screenshot shows the AWS Lambda function configuration page for a function named "pawn-lambda". The "Code source" tab is selected, showing the code editor with a Python file named "lambda\_function.py". The execution results section shows the following details:

- Test Event Name:** pawn-bucket
- Response:**

```
{
  "statusCode": 200,
  "body": "\"Log entry created successfully\""
}
```
- Function Logs:**

```
START RequestId: 69a05696-0896-495f-906f-cfa14acf7bb4 Version: $LATEST
An image has been added to the bucket example-bucket : test%2Fkey
END RequestId: 69a05696-0896-495f-906f-cfa14acf7bb4
REPORT RequestId: 69a05696-0896-495f-906f-cfa14acf7bb4 Duration: 1.67 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 0 ms
```
- Request ID:** 69a05696-0896-495f-906f-cfa14acf7bb4

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

The screenshot shows the AWS CloudWatch Log Groups interface. On the left, there's a navigation sidebar with options like Dashboards, Alarms, Logs (selected), Log Anomalies, Live Tail, Logs Insights, Metrics, X-Ray traces, Events, Application Signals, Network monitoring, and Insights. The main area displays 'Log groups (2)'. It includes a search bar, filter buttons (Exact match), and actions like 'Actions', 'View in Logs Insights', 'Start tailing', and 'Create log group'. Two log groups are listed: '/aws/lambda/bhum-lambda' and '/aws/lambda/pawn-lambda', both set to Standard Log class and Never expire retention.

The screenshot shows the AWS CloudWatch Log Events interface for the '/aws/lambda/pawn-lambda' log group. The path in the top bar is 'CloudWatch > Log groups > /aws/lambda/pawn-lambda > 2024/10/05/[\${LATEST}]9204e1e2d6d8477baab29bddea33a0d0'. The interface includes a filter bar, time range buttons (Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone), and a display dropdown. The log events table has columns for 'Timestamp' and 'Message'. It lists several log entries from October 5, 2024, such as INIT START, START RequestId, and REPORT RequestId. A message at the bottom indicates 'No newer events at this moment. Auto retry paused.'

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

