

**Name: Kshitij Hundre**  
**Div: D15C**  
**Roll No:18**

## **Exp 1 :Static Hosting:**

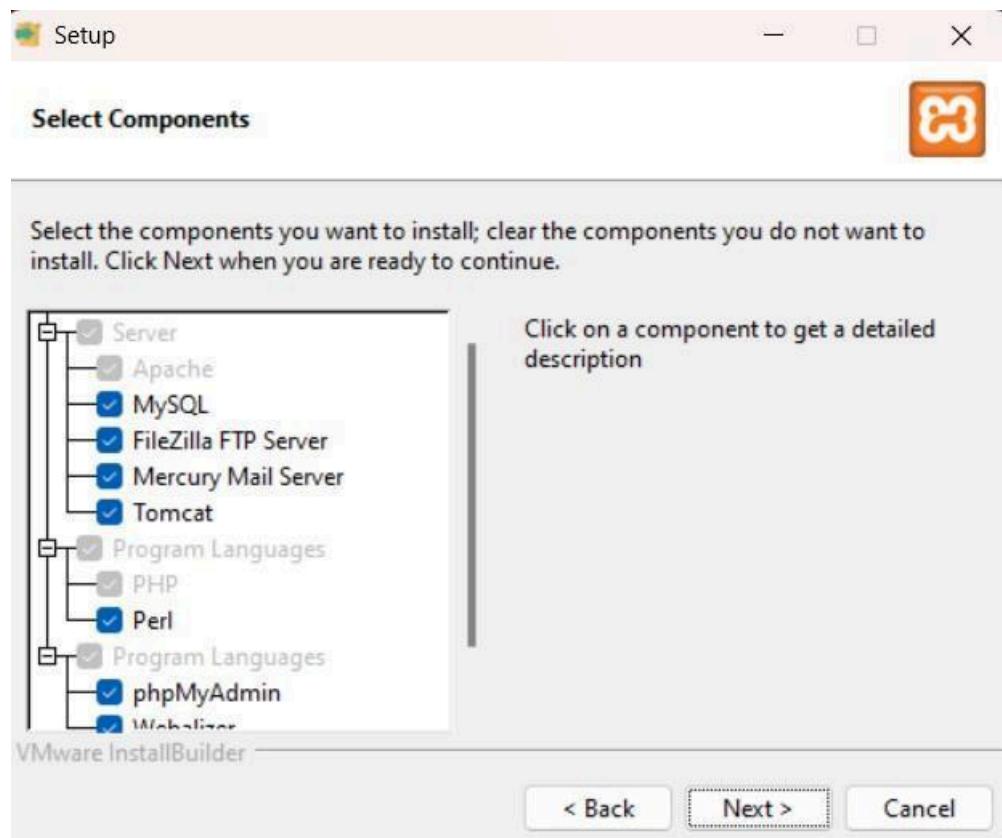
### **1) On local server (XAMPP)**

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

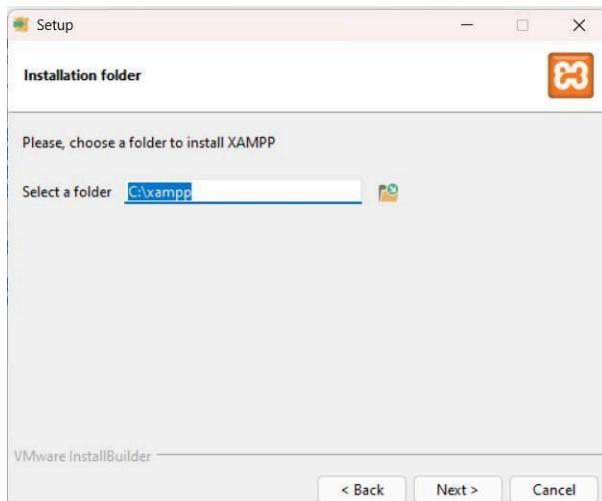
- 1) Select your OS. It will automatically start downloading.



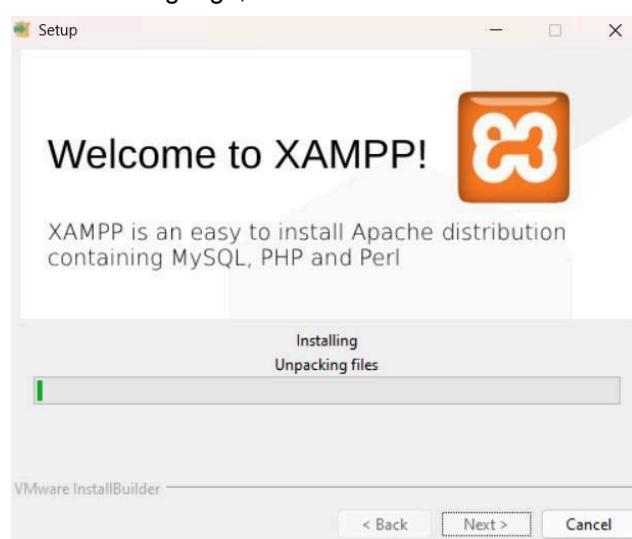
- 2) Open the setup file. Select all the required components and click next



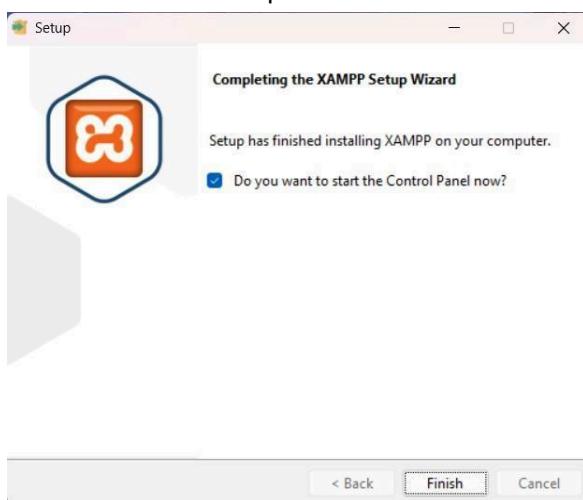
- 3) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



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



- 5) The installation is complete. Click Finish



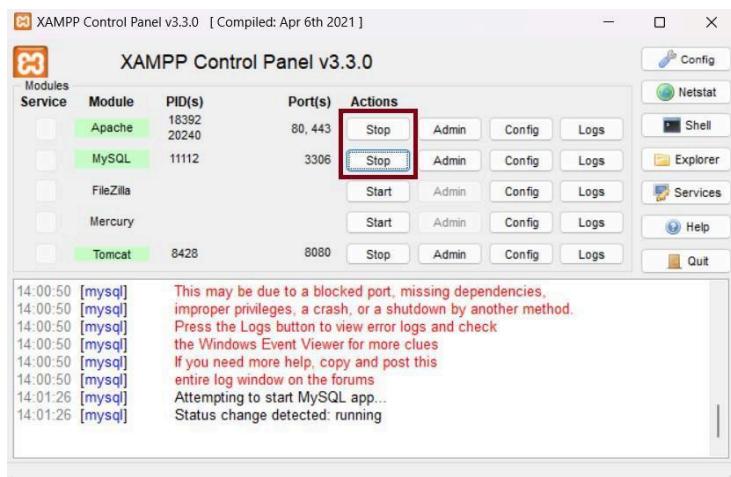
**Step 2:** Setup a file that is to be hosted on the server. Make sure the file has extension .php

test1	06-08-2024 22:48	PHP Source File	1 KB
-------	------------------	-----------------	------

**Step 3:** Go to the directory where XAMPP was installed. Go to **htdocs** folder. Place your folder in this directory.

Name	Date modified	Type	Size
dashboard	06-08-2024 20:42	File folder	
img	06-08-2024 20:42	File folder	
webalizer	06-08-2024 20:42	File folder	
xampp	06-08-2024 22:44	File folder	
applications	15-06-2022 21:37	Chrome HTML Do...	4 KB
bitnami	15-06-2022 21:37	CSS Source File	1 KB
favicon.ico	16-07-2015 21:02	ICO File	31 KB
index	16-07-2015 21:02	PHP Source File	1 KB
test1	06-08-2024 22:48	PHP Source File	1 KB
text	06-08-2024 22:23	PHP Source File	1 KB

**Step 4:** Open XAMPP Control Panel, start the Apache service (Required) and mySQL service (if needed)

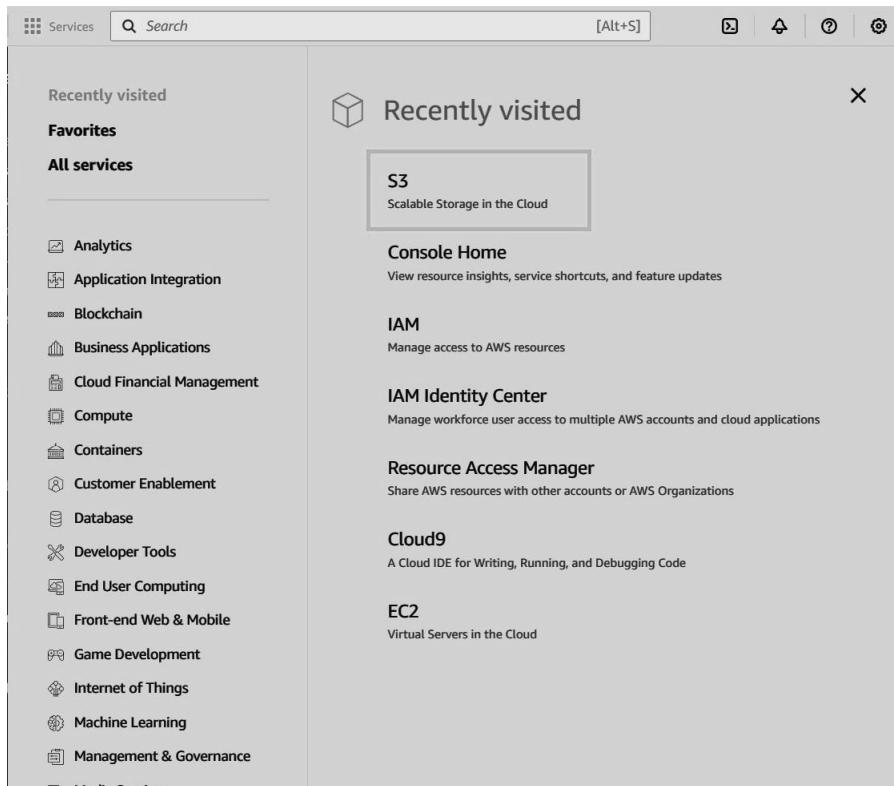


**Step 5:** Open your web browser. Type localhost/YOUR\_FILENAME.php. This will open your website on your browser.



## 2) AWS S3

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



## Step 2: Click on Create Bucket

The screenshot shows the AWS S3 landing page. On the right side, there is a prominent call-to-action box titled "Create a bucket". Below it, a text explains that every object in S3 is stored in a bucket and provides instructions to upload files and folders. A large orange "Create bucket" button is centered in this box. To the left of this box, there's a section titled "How it works" featuring a video thumbnail for "Introduction to Amazon S3". To the right, there are sections for "Pricing" (mentioning no minimum fees) and "Resources" (linking to the User guide). The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and copyright information.

## Step 3: Give a name to your bucket, keeping other options default, scroll down and click on Create Bucket

The screenshot shows the "Create bucket" configuration page. The "General configuration" tab is active. It includes fields for "Bucket name" (set to "statichosting27") and "AWS Region" (set to "US East (N. Virginia) us-east-1"). Under "Bucket type", the "General purpose" option is selected. There are two tabs: "General purpose" and "Directory - New". The "Object Ownership" tab is also visible at the bottom. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and copyright information.

#### Step 4: Click on the name of your bucket and goto Properties

Amazon S3 > Buckets

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

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

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (1) [Info](#) (All AWS Regions)

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
statichosting27	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 4, 2024, 15:30:03 (UTC+05:30)

CloudShell Feedback

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

Amazon S3 > Buckets > statichosting27 [Info](#)

statichosting27 [Info](#)

Objects (0) [Info](#)

Copy Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

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

Upload

#### Step 5: Scroll down till you find Static website hosting, click on edit

Transfer acceleration

Use an accelerated endpoint for faster data transfers. [Learn more](#)

Transfer acceleration

Disabled

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#)

Object Lock

Disabled

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)

Requester pays

Disabled

Static website hosting

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

Static website hosting

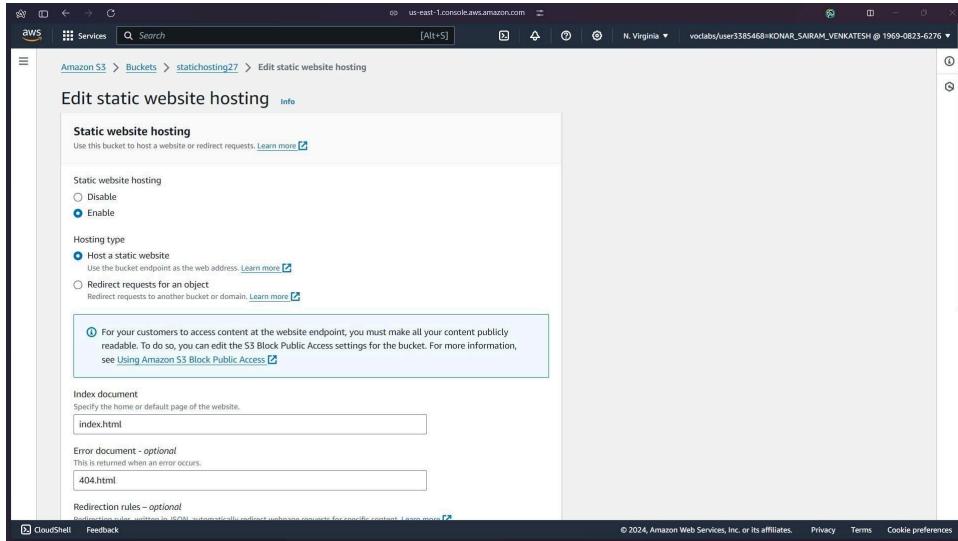
Disabled

Edit

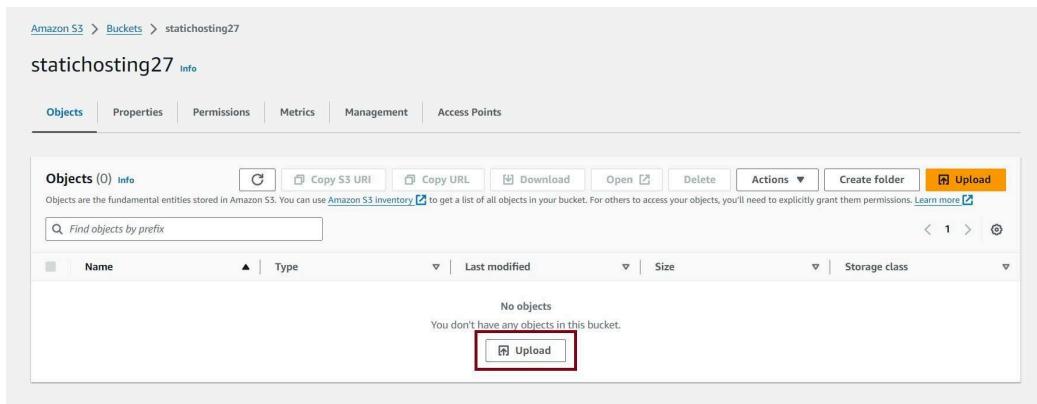
CloudShell Feedback

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

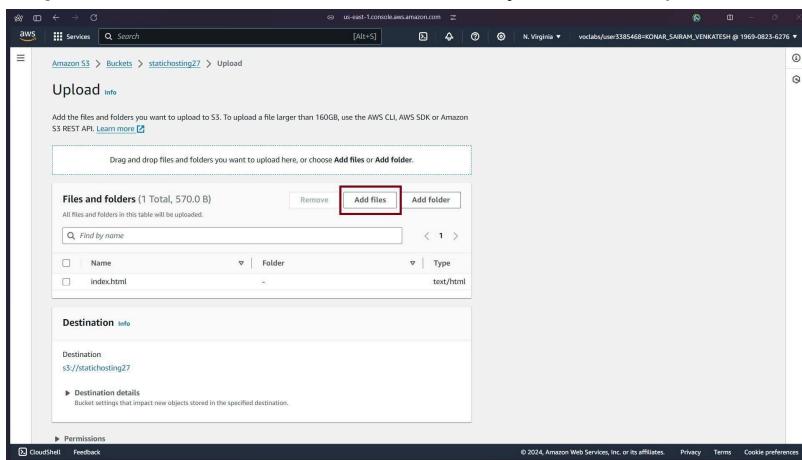
**Step 6:** Enable static website hosting, in Index document, write the name of your document and in error document, give name as 404.html. Save your changes.



**Step 7:** Go to Objects tab and click on upload file.



**Step 8:** Click on Add files. Add all the files you want to upload. Then scroll down and click on Upload



**Step 9:** This will take you to the Objects screen. Switch to Properties, scroll down to Static web hosting. There you would find the link (Bucket website endpoint) to your website.

The screenshot shows the 'Static website hosting' section of the AWS S3 Bucket Properties page. It includes fields for 'Hosting type' (set to 'Bucket hosting') and a 'Bucket website endpoint' (set to 'http://statichosting27.s3-website-us-east-1.amazonaws.com'). A red box highlights the endpoint URL.

**Step 10:** Open the link. It will show a 403 forbidden error screen as the contents of the bucket are not available for the public users. To change this, go to Permissions tab, go to Block public access and click on edit

The screenshot shows a 403 Forbidden error page. The browser address bar shows 'statichosting27.s3-website-us-east-1.amazonaws.com'. The main content area displays the error message '403 Forbidden' and a list of error details.

**403 Forbidden**

- Code: AccessDenied
- Message: Access Denied
- RequestId: 8TQ4EGP4TK06MVPB
- HostId: hF+ToadQUoCuDM8H+iFrSXdA28TGp+xikYbjb4CICS/t+3it4ihA/tvgA1Xr1xo+JL5AhkT6hJs=

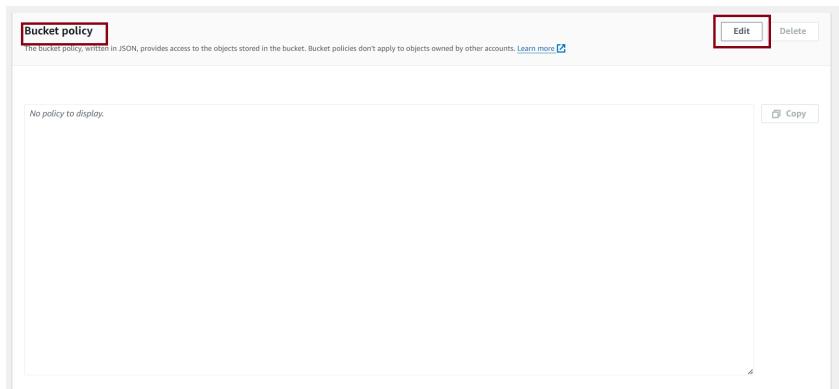
**An Error Occurred While Attempting to Retrieve a Custom Error Document**

- Code: AccessDenied
- Message: Access Denied

**Step 11:** Uncheck the Block all public access checkbox and click on save changes

The screenshot shows the 'Edit Block public access (bucket settings)' page. It features a 'Block public access (bucket settings)' section with a note about setting four options. A red box highlights the 'Block all public access' checkbox, which is currently checked. Below it are three other unchecked options: 'Block public access to buckets and objects granted through new access control lists (ACLs)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies'. At the bottom are 'Cancel' and 'Save changes' buttons.

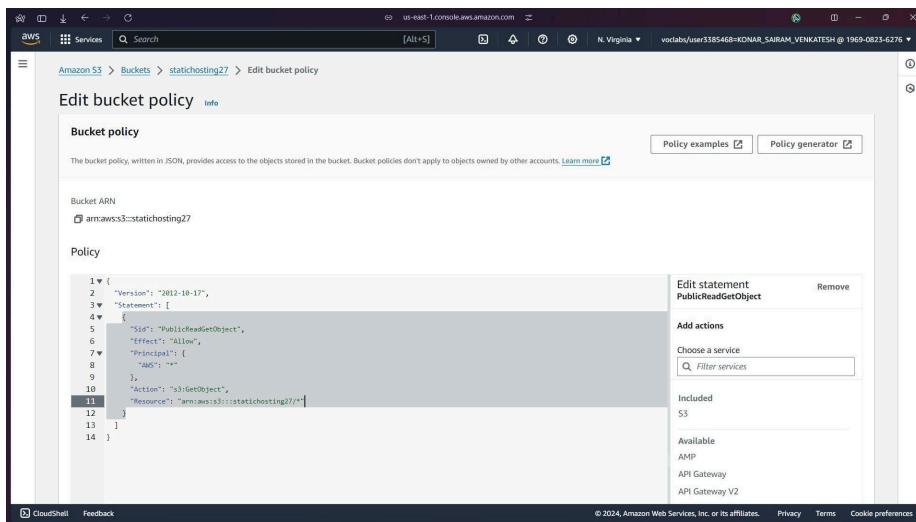
## Step 12: Scroll down to bucket policy and click edit



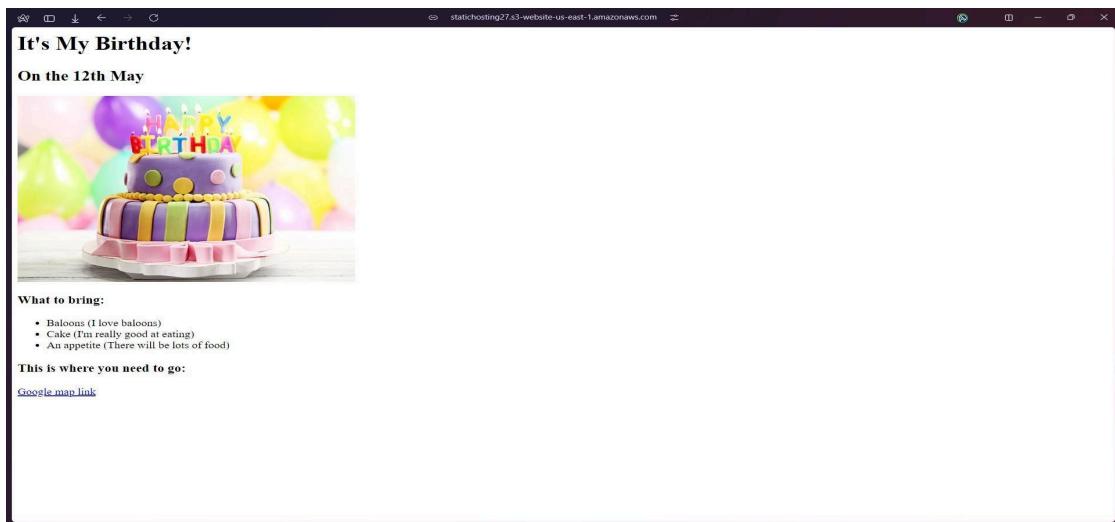
## Step 13:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "*"  
      },  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::YOUR-BUCKET-NAME-HERE/*"  
    }  
  ]  
}
```

Paste this code snippet in the policy textarea. Replace YOUR-BUCKET-NAME-HERE with the name you have given to your bucket. Save the changes.



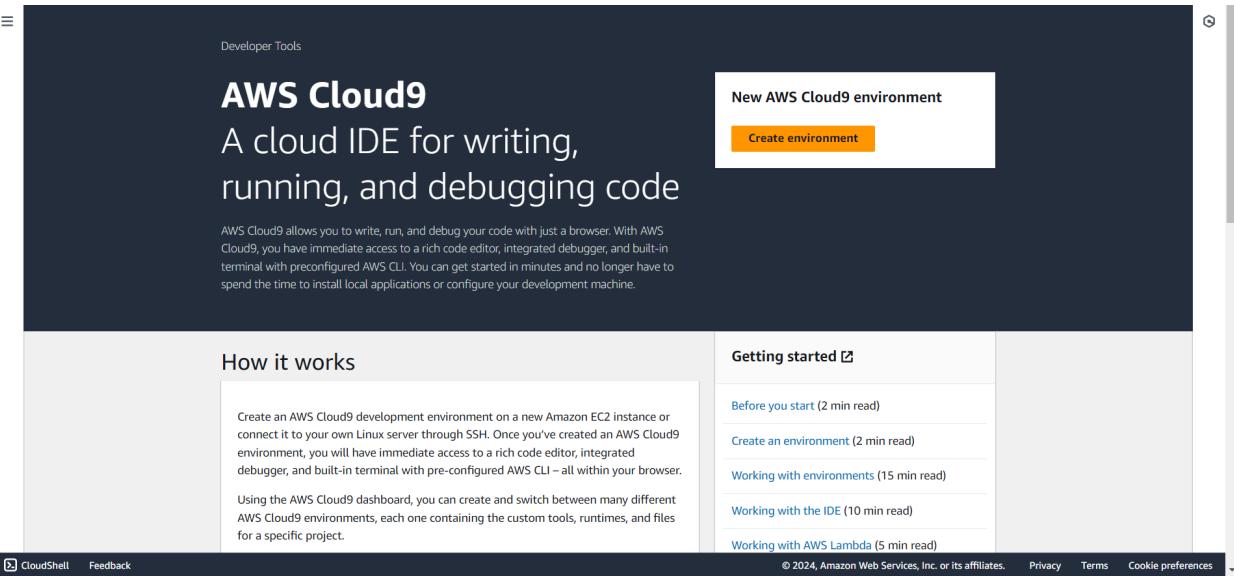
**Step 14:** Now reload the website. You can see your website



Name: Kshitij Hundre  
Div: D15C  
Roll No.18

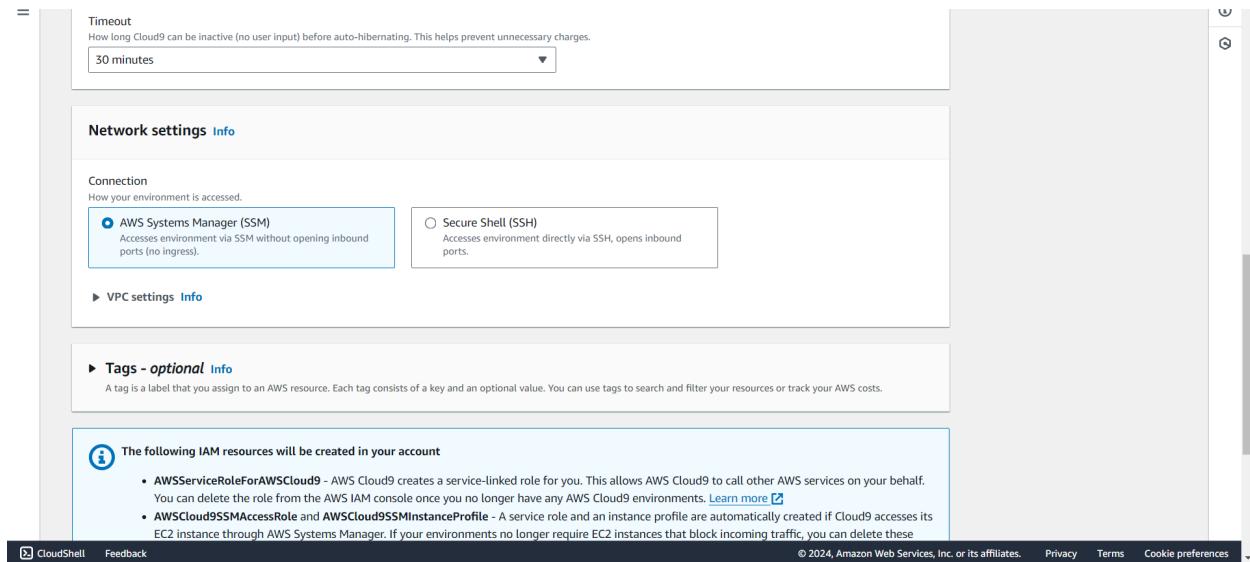
## Experiment 1B: IAM and cloud9

1. Open the AWS account and search for Cloud9.



The screenshot shows the AWS Cloud9 landing page. At the top right, there is a prominent orange "Create environment" button. Below it, the page title "AWS Cloud9" and subtitle "A cloud IDE for writing, running, and debugging code" are displayed. A descriptive paragraph explains that AWS Cloud9 allows users to write, run, and debug code with just a browser, providing immediate access to a rich code editor, integrated debugger, and built-in terminal. To the left, a "How it works" section provides a brief overview of the service's functionality. On the right, a "Getting started" sidebar lists several documentation links: "Before you start" (2 min read), "Create an environment" (2 min read), "Working with environments" (15 min read), "Working with the IDE" (10 min read), and "Working with AWS Lambda" (5 min read). The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

2. Enter the name and other required configuration for creating an environment. In network settings, using the AWS system manager gives an error while creating the environment



The screenshot shows the "Create environment" configuration page. It includes sections for "Timeout" (set to 30 minutes), "Network settings" (with options for AWS Systems Manager (SSM) or Secure Shell (SSH)), and "Tags - optional". A note at the bottom states: "The following IAM resources will be created in your account". This note lists two items: "AWS Service Role for AWS Cloud9" and "AWS Cloud9 SSM Access Role and AWS Cloud9 SSM Instance Profile". The bottom of the page features standard AWS navigation links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

▶ VPC settings [Info](#)

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

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

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

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

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

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

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

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

### 3. Use the Secure Shell option in Network settings.

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

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

**Connection**  
How your environment is accessed.

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

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

▶ VPC settings [Info](#)

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

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

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

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

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

The screenshot shows the AWS Cloud9 interface. On the left, there's a sidebar with links for 'My environments', 'Shared with me', and 'All account environments'. Below that is a 'Documentation' link. The main area is titled 'Environments (1)' and shows a single environment named 'MyEnvironment'. The table columns are 'Name', 'Cloud9 IDE', 'Environment type', 'Connection', 'Permission', and 'Owner ARN'. The 'Owner ARN' column contains the value: arn:aws:sts::354256622778:assumed-role/vclabs/user3404112=SHARMA\_RAKSHIT\_KUMAR. At the top right of the table is an orange 'Create environment' button. Above the table, there are two notifications: one about creating the environment and another about AWS Toolkits. The bottom of the screen shows standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

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

The screenshot shows the AWS Cloud9 development environment for the 'MyEnvironment' project. The left sidebar lists files: 'MyEnvironment', 'c9', and 'README.md'. The main area has a title 'AWS Cloud9' and a subtitle 'Welcome to your development environment'. It says 'AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.' A 'Getting started' sidebar on the right includes 'Create File', 'Upload Files...', and 'Clone from GitHub'. At the bottom, there's a terminal window showing a bash session with the command 'vclabs:~/environment \$'. The bottom left corner shows the text 'AWS profile.default'.

## IAM user creation steps

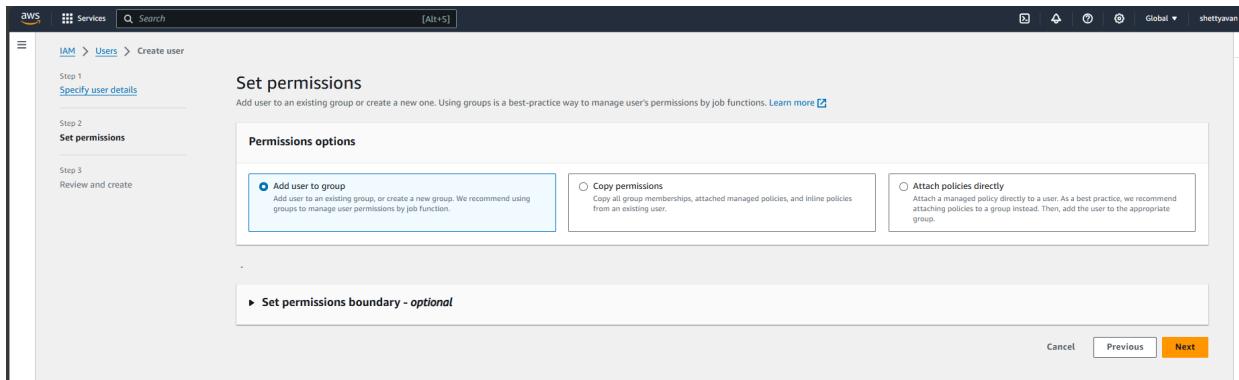
1. Open the aws account and search for IAM in service.

The screenshot shows the AWS IAM Dashboard. On the left, there's a navigation pane with options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'Account settings'. The main area displays 'Security recommendations' with two items: 'Add MFA for root user' (with a 'C' icon and 'Add MFA' button) and 'Root user has no active access keys' (with a 'C' icon). Below that is a section for 'IAM resources' with a table showing counts: User groups (1), Users (1), Roles (6), Policies (1), and Identity providers (0).

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

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. It's Step 1 of 3. The left sidebar shows 'Step 1: Specify user details', 'Step 2: Set permissions', and 'Step 3: Review and create'. The main form has a 'User details' section with a 'User name' field containing 'sample'. A note below says the name can have up to 64 characters and lists valid characters. There's an optional checkbox for 'Provide user access to the AWS Management Console - optional' with a note about best practices. A note at the bottom explains how to generate programmatic access keys. At the bottom right are 'Cancel' and 'Next' buttons.

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



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

**Name the group**

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

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

**User name**  sample

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

Policy name	Type	Used as	Description
<input type="checkbox"/> <a href="#">AdministratorAccess</a>	AWS managed - job function	None	Provides full ac...
<input type="checkbox"/> <a href="#">AdministratorAccess-Amplify</a>	AWS managed	None	Grants account...
<input type="checkbox"/> <a href="#">AdministratorAccess-AWSElasticBeans...</a>	AWS managed	None	Grants account...
<input type="checkbox"/> <a href="#">AlexaForBusinessDeviceSetup</a>	AWS managed	None	Provide device...

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

The screenshot shows the 'Set permissions' step of the 'Create user' wizard. In the 'Permissions options' section, the 'Add user to group' option is selected, with a note: 'Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job functions.' Below this are three other options: 'Copy permissions', 'Attach policies directly', and a 'User groups' table. The table lists one group: 'myweb-app-group'. The 'Next' button is highlighted in orange at the bottom right.

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

The screenshot shows the 'Review and create' step of the 'Create user' wizard. It displays the 'User details' section with a user name 'sample', 'Console password type' set to 'None', and 'Require password reset' set to 'No'. The 'Permissions summary' section shows 'No resources'. The 'Tags' section indicates 'No tags associated with the resource' and has an 'Add new tag' button. The 'Create user' button is highlighted in orange at the bottom right.

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

The screenshot shows the AWS IAM Users page. At the top, a success message says "User created successfully" with a link to "View user". Below it, the "Users (1) Info" section indicates "An IAM user is an identity with long-term credentials that is used to interact with AWS in an account." A table lists one user: "sample" with "User name", "Path /", "Groups 0", "Last activity -", "MFA -", "Password age -", "Console last sign-in -", "Access key ID -", and "Active".

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

The screenshot shows the AWS IAM User Groups page. It displays a "sample\_group" entry with "User group name sample\_group", "Creation time August 07, 2024, 09:50 (UTC+05:30)", and "ARN arn:aws:iam::434768569951:group/sample\_group". Below this, the "Users (1)" tab is selected, showing "Users in this group (1)". The table lists "sample" with "User name", "Groups 1", "Last activity None", and "Creation time 7 minutes ago".

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

The screenshot shows the AWS IAM Attach permission policies page for the "myweb-app-group". It lists "Other permission policies (1/945)" with "AWSCloud9EnvironmentMember" selected. The policy details show "Policy name AWSCloud9EnvironmentMember", "Type AWS managed", "Used as None", and "Description Provides the ability to be invited into AW...". The "Attach policies" button is highlighted.

Name: Kshitij Hundre  
Div: D10C  
Roll No:18

## Practical No 2 : Elastic Beanstalk

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

The screenshot shows the Amazon Elastic Beanstalk landing page. At the top, it says "Compute" and "Amazon Elastic Beanstalk: End-to-end web application management." Below this, a description states: "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's a "Get started" button with the subtext "Easily deploy your web application in minutes." and a "Create application" button. On the left, there's a "Get started" section with a subtext about uploading code and automatic handling of deployment, load balancing, and scaling. On the right, there's a "Pricing" section stating "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", "Getting started", and "Cookie preferences".

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

The screenshot shows the "Configure environment" step in the AWS Elastic Beanstalk setup wizard. It has three main sections: "Environment tier", "Application information", and "Environment information".  
**Environment tier:** Shows "Web server environment" selected. Subtext: "Run a website, web application, or web API that serves HTTP requests."  
**Application information:** Shows "Application name" set to "sample". Subtext: "Maximum length of 100 characters."  
**Environment information:** Subtext: "Choose the name, subdomain and description for your environment. These cannot be changed later."

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

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

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

Domain  
 .us-east-1.elasticbeanstalk.com [Check availability](#)  
( kshitij.us-east-1.elasticbeanstalk.com is available)

Environment description  
  
[Edit](#)

**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  
 [Edit](#)

Platform branch  
 [Edit](#)

[View details](#)

### 3) Configure the service access.

**Configure service access** [Info](#)

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

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

Existing service roles  
 Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.  
 [Edit](#)

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

EC2 instance profile  
 Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.  
 [Edit](#)

[View permission details](#)

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

- 4) Choose one of the available VPC and instance subnet. Click next.

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

### Virtual Private Cloud (VPC)

#### VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.

[Learn more](#)

vpc-0a482134962ed0c59 | (172.31.0.0/16)



[Create custom VPC](#)

### Instance settings

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

#### Public IP address

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

Activated

### Instance subnets

Filter instance subnets

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

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

## Configure instance traffic and scaling - optional Info

### Instances Info

Configure the Amazon EC2 instances that run your application.

#### Root volume (boot device)

##### Root volume type

(Container default)

##### Size

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

8 GB

##### IOPS

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

100 IOPS

##### Throughput

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

125 MiB/s

#### Amazon CloudWatch monitoring

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

##### Monitoring interval

5 minute

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

**Choose x86 instance types**

t3.micro X   t3.small X

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

ami-083f545ce1a73bf03

**Availability Zones**  
Number of Availability Zones (AZs) to use.

Any ▼

**Placement**  
Specify Availability Zones (AZs) to use.

**Choose Availability Zones (AZs)** ▼

**Scaling cooldown**

360 seconds

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

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

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

**▼ Monitoring [Info](#)**

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

**System**  
 Basic  
 Enhanced

**CloudWatch Custom Metrics - Instance**

**Choose metrics** ▼

**CloudWatch Custom Metrics - Environment**

**Choose metrics** ▼

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

**Log streaming**  
 Activated (standard CloudWatch charges apply.)

**Retention**

7 ▼

**Lifecycle**

**Instance log streaming to CloudWatch logs**

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

**Log streaming**  
(standard CloudWatch charges apply.)

**Activated**

**Retention**

7

**Lifecycle**

Keep logs after terminating envir...

**Environment properties**

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

No environment properties have been configured.

[Add environment property](#)

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

7) In the review section, click submit.

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

[Cancel](#) [Previous](#) [Submit](#)

## 8) Environment has been created successfully.

Sampel-env [Info](#)

[Events](#) [Health](#) [Logs](#) [Monitoring](#) [Alarms](#) [Managed updates](#) [Tags](#)

**Events (10)** [Info](#)

Filter events by text, property or value

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

## 9) Deploy something on the recently created environment.

**Upload and deploy**

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

Upload application

**Screenshot 2023-11-10 185456.png**

File must be less than 500MB max file size

Version label

Unique name for this version of your application code.

sampel-version-1

Current number of EC2 instances: 1

[Cancel](#) [Deploy](#)

⌚ Environment successfully launched.

⌚ Successfully uploaded file Screenshot 2023-11-10 185456.png to S3, created application version and started deployment with new application version

Elastic Beanstalk > Environments > Sampel-env

## Sampel-env Info

Events Health Logs Monitoring Alarms Managed updates Tags

Change version

**Environment overview**

Health	Environment ID
<span style="color: orange;">⚠ Warning</span>	e-u7kfdezi3r
Domain	Application name
kshitij.us-east-1.elasticbeanstalk.com 	sampel

**Platform**

Platform	Platform state
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	<span style="color: green;">✔ Supported</span>
Running version	-

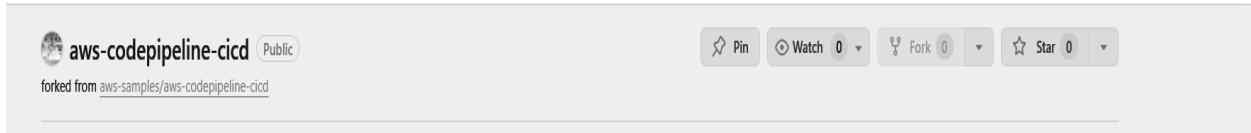
**Events (1)** Info

Filter events by text, property or value

Time	Type	Details
August 9, 2024 21:25:22 (UTC+5:30)	<span style="color: orange;">⚠ WARN</span>	Service role "arn:aws:iam::996474913977:role/EMR_EC2_DefaultRole" is missing permissions required to check for

# Pipeline Creation:

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



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

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
No results There are no results to display.				

- 3) Create a pipeline:

# Congratulations!

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

For next steps, read the [AWS CodePipeline Documentation](#). Incedge 2020

## Choose pipeline settings Info

Step 1 of 5

### Pipeline settings

#### Pipeline name

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

No more than 100 characters

#### Pipeline type

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

#### Execution mode

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

**Superseded**

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

**Queued (Pipeline type V2 required)**

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

**Parallel (Pipeline type V2 required)**

4)

## Add source stage Info

Step 2 of 5

### Source

#### Source provider

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



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



#### The GitHub (Version 1) action is not recommended

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

#### Change detection options

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

GitHub webhooks (recommended)

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

AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

S

## Add source stage Info

Step 2 of 5

### Source

#### Source provider

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



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

You have successfully configured the action with the provider.



#### The GitHub (Version 1) action is not recommended

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

#### Repository



#### Branch



#### Change detection options

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

GitHub webhooks (recommended)

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

AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

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

Add deploy stage [Info](#)

Step 4 of 5

**You cannot skip this stage**  
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

**Deploy**

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

**Region**

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#)  
  
No more than 100 characters

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

Configure automatic rollback on stage failure

6) review the pipeline settings.

Review [Info](#)

Step 5 of 5

**Step 1: Choose pipeline settings**

**Pipeline settings**

**Pipeline name**  
test\_pipeline

**Pipeline type**  
V2

**Execution mode**  
QUEUED

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

**Service role name**  
AWSCodePipelineServiceRole-us-east-1-test\_pipeline

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

Success  
Congratulations! The pipeline firstpipeline has been created.

Developer Tools > CodePipeline > Pipelines > firstpipeline

## firstpipeline

Pipeline type: V2 Execution mode: QUEUED

**Source** Succeeded  
Pipeline execution ID: [c4dc21b-af39-4463-a00a-e76d7579dcf3](#)

Source  
[GitHub \(Version 2\)](#) [Succeeded - 2 minutes ago](#)  
[8fd5da54](#)

[View details](#)

[Add data](#) Source: Update README.md

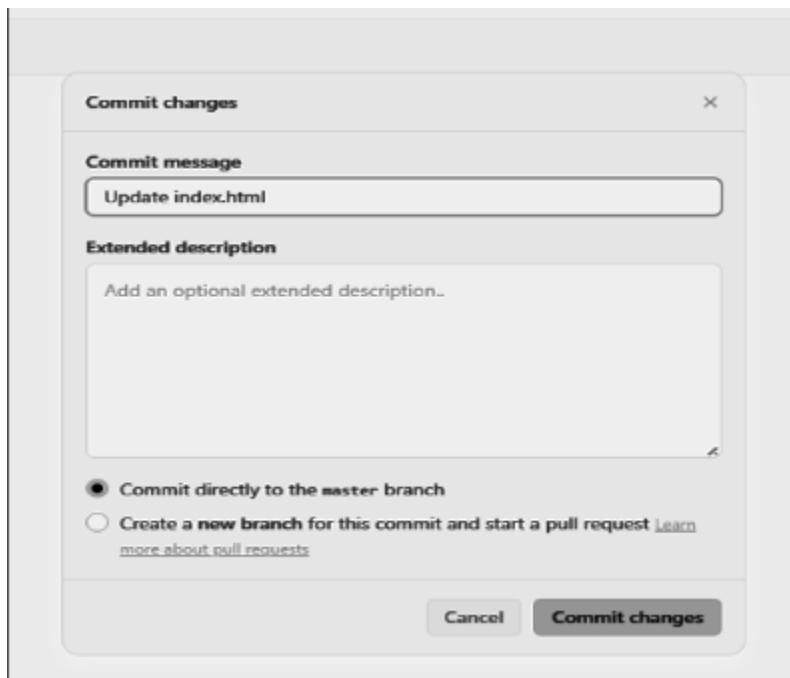
[Disable transition](#)

**Deploy** Succeeded  
Pipeline execution ID: [c4dc21b-af39-4463-a00a-e76d7579dcf3](#)

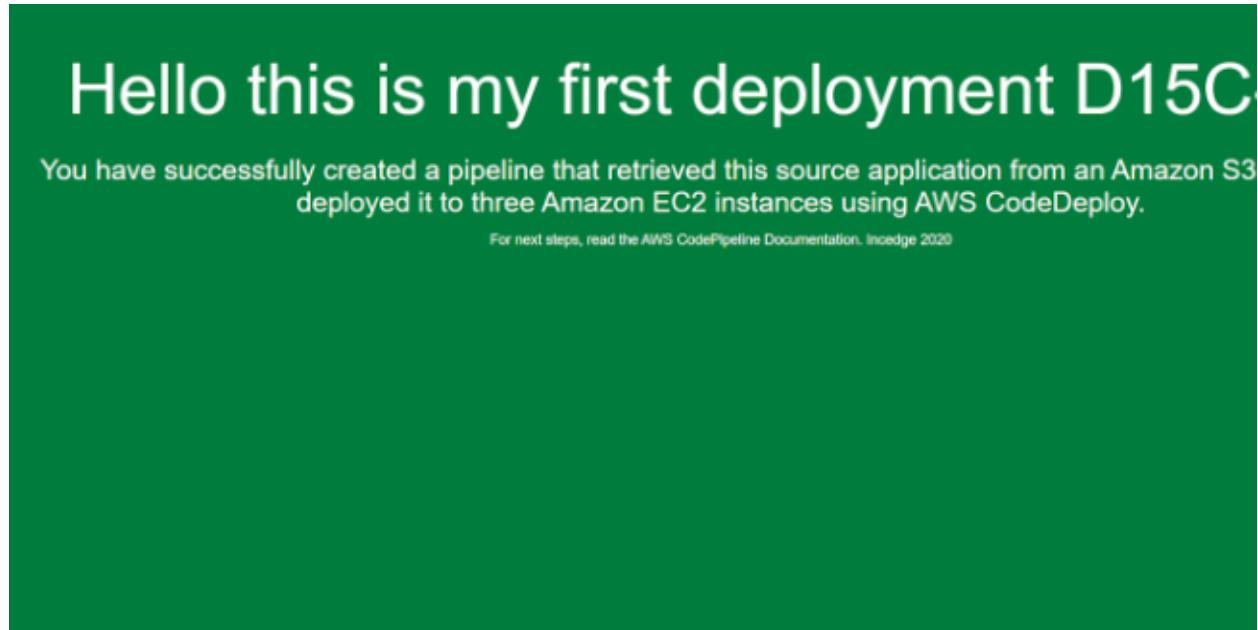
[Deploy](#)

Notify Edit

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



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



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

1. Create 3 EC-2 instances with all running on Amazon Linux as OS with inbound SSH allowed and the proper key

The screenshot shows the AWS EC2 'Launch an instance' wizard. The process is at step 1: 'Name and tags'. The 'Name' field contains 'master'. Below it, under 'Application and OS Images (Amazon Machine Image)', the 'Amazon Linux 2023 AMI' is selected. The 'Virtual server type (instance type)' is set to 't2.medium'. A summary panel on the right indicates 1 instance will be launched, using the 'Amazon Linux 2023 AMI'. It also shows the 'Free tier' information: 750 hours of t2.micro or t3.micro usage per month, 750 hours of public IPv4 address usage, 30 GB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth. Buttons for 'Launch instance' and 'Review commands' are visible.

To efficiently run a Kubernetes cluster, select an instance type of at least t2.medium as Kubernetes recommends at least 2 vCPU to run smoothly.

**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](#)

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

[Edit](#)

Network [Info](#)  
vpc-0deb6a82b5be91aae

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

We'll create a new security group called 'launch-wizard-10' with the following rules:

<input checked="" type="checkbox"/> <b>Allow SSH traffic from</b> <small>Helps you connect to your instance</small>	<input type="text" value="Anywhere"/> <small>0.0.0.0/0</small>
<input checked="" type="checkbox"/> Allow HTTPS traffic from the internet <small>To set up an endpoint, for example when creating a web server</small>	
<input checked="" type="checkbox"/> Allow HTTP traffic from the internet <small>To set up an endpoint, for example when creating a web server</small>	

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

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.5.2...[read more](#)  
ami-0182f373e66f89c85

Virtual server type (instance type)  
t2.medium

Firewall (security group)  
New security group

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

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

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

In this way create 3 instances namely master, worker-1 and worker-2

Instances (3) <a href="#">Info</a>												
<a href="#">Find Instance by attribute or tag (case-sensitive)</a> <span style="float: right;">All states <a href="#">▼</a></span>												
<span style="float: left;">Last updated <a href="#">C</a></span> <span style="float: right;">Connect <a href="#">Instance state ▾</a> Actions <a href="#">▼</a> Launch instances <a href="#">▼</a></span>												
Name <a href="#">▼</a>	Instance ID	Instance state <a href="#">▼</a>	Instance type <a href="#">▼</a>	Status check <a href="#">▼</a>	Alarm status	Availability Zone <a href="#">▼</a>	Public IPv4 DNS <a href="#">▼</a>	Public IPv4 ... <a href="#">▼</a>	Elastic IP	IPv6		
master	i-0604dbbc26a0a4f01	<span style="color: green;">Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.medium	<span style="color: grey;">Initializing</span>	<a href="#">View alarms +</a>	us-east-1a	ec2-54-85-79-186.com...	54.85.79.186	-	-		
worker-1	i-05548bd7fe0a7292f	<span style="color: green;">Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.medium	<span style="color: grey;">Initializing</span>	<a href="#">View alarms +</a>	us-east-1a	ec2-54-196-211-209.co...	54.196.211.209	-	-		
worker-2	i-007a15dade39c85b0	<span style="color: green;">Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.medium	<span style="color: grey;">Initializing</span>	<a href="#">View alarms +</a>	us-east-1a	ec2-18-209-62-85.com...	18.209.62.85	-	-		

2. SSH into all 3 machines each in separate terminal
  - a. You can do it through the aws console directly

```
C:\Users\Avan>ssh -i "C:\Users\Avan\Downloads\kub1.pem" ec2-user@3.85.237.93
The authenticity of host '3.85.237.93 (3.85.237.93)' can't be established.
ED25519 key fingerprint is SHA256:Nz2iC26abFyhATf/8i4F0IgWmDoxTXBbzY9/NkMwYyM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.85.237.93' (ED25519) to the list of known hosts.

      #_
      ~\_ #####_          Amazon Linux 2023
      ~~ \_#####\_
      ~~  \###|_
      ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
      ~~    V~' '-->
      ~~~   /
      ~~.~. /_/
      _/m/' /_/
Last login: Thu Sep 12 13:11:49 2024 from 18.206.107.29
```

Or

*b. Locate your key from the Downloads folder and open it in cmd and paste this command*

**ssh -i <-your-key->.pem ec2-user<ip-address of instance>**

```
C:\Users\Avan\Downloads>ssh -i C:\Users\Avan\Downloads\kub1.pem ec2-user@54.85.79.186
The authenticity of host '54.85.79.186 (54.85.79.186)' can't be established.
ED25519 key fingerprint is SHA256:ffz946cqlxbNvBsPqtNcxLlfXmhW8VJhyLD4n4jStto.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.85.79.186' (ED25519) to the list of known hosts.

      #_
      ~\_ #####_          Amazon Linux 2023
      ~~ \_#####\_
      ~~  \###|_
      ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
      ~~    V~' '-->
      ~~~   /
      ~~.~. /_/
      _/m/' /_/
Last login: Sat Sep 14 06:03:27 2024 from 18.206.107.27
[ec2-user@ip-172-31-20-75 ~]$
```

With this you can continue your commands through local terminal

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

### Install Docker

`sudo yum install docker -y`

```
[ec2-user@ip-172-31-212 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:33:43 ago on Thu Sep 12 13:11:13 2024.
Dependencies resolved.
=====
Transaction Summary
=====
Installing:
  docker                                     x86_64          25.0.6-1.amzn2023.0.2
  containerd                                 x86_64          1.7.20-1.amzn2023.0.1
  installing dependencies:
    containerd                                x86_64          1.7.20-1.amzn2023.0.1
    iptables-libs                             x86_64          1.8.8-3.amzn2023.0.2
    iptables-nft                            x86_64          1.8.8-3.amzn2023.0.2
    libcgroup                                x86_64          3.0-1.amzn2023.0.1
    libnetfilter_conntrack                   x86_64          1.0.8-2.amzn2023.0.2
    libnftnl                                 x86_64          1.0.1-19.amzn2023.0.2
    libnfnetlink                           x86_64          1.2.2-2.amzn2023.0.2
    pigz                                    x86_64          2.5-1.amzn2023.0.3
    runc                                    x86_64          1.1.13-1.amzn2023.0.1
=====
Transaction Summary
```

Then, configure cgroup in a daemon.json file by using following commands

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

```
[ec2-user@ip-172-31-20-75 ~]$ cd /etc/docker
[ec2-user@ip-172-31-20-75 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-20-75 docker]$ ls
daemon.json  kubectl
```

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

```
[ec2-user@ip-172-31-31-212 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-31-212 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-31-212 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-31-212 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-31-212 docker]$ █
```

#### 4. Install Kubernetes on all 3 machines

SELinux needs to be disabled before configuring kubelet

- sudo setenforce 0
- sudo sed -i 's/^SELINUX=enforcing\$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-26-2 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-26-2 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-2 docker]$ █
```

Add kubernetes repository (paste in terminal)

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

Type following commands to install set of kubernetes packages:

- sudo yum update
- sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

```
[ec2-user@ip-172-31-212 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:23 ago on Thu Sep 12 14:09:10 2024.
Dependencies resolved.
=====
Package           Architecture   Version
=====
Installing:
kubeadm          x86_64        1.30.5-150500.1.1
kubectl          x86_64        1.30.5-150500.1.1
kubelet           x86_64        1.30.5-150500.1.1
Installing dependencies:
conntrack-tools  x86_64        1.4.6-2.amzn2023.0.2
cri-tools         x86_64        1.30.1-150500.1.1
kubernetes-cni   x86_64        1.4.0-150500.1.1
libnetfilter_cthelper x86_64    1.0.0-21.amzn2023.0.2
libnetfilter_cttimeout x86_64    1.0.0-19.amzn2023.0.2
libnetfilter_queue x86_64        1.0.5-2.amzn2023.0.2
=====
Transaction Summary
Install 9 Packages

Total download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm
(3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm
(4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm
(5/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm
(6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm
(7/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm
(8/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.rpm
(9/9): kubelet-1.30.5-150500.1.1.x86_64.rpm
=====
Total
Kubernetes
Importing GPG key 0x9A296436:
Userid      : "jsv:kubernetes OBS Project <jsv:kubernetes@build.opensuse.org>"
Fingerprint: DB15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
From       : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
```

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

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

## 5. Perform this ONLY on the Master machine

Initialize kubernetes by typing below command

- sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all

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

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.26.2:6443 --token 6cj5z0.8ei243v0zn9k7erg \
    --discovery-token-ca-cert-hash sha256:abd917ec30e12c5616bf647a3d174bef3d271e92c30b8f2f7768cfb3181341d4
[ec2-user@ip-172-31-26-2 docker]$ █
```

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

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

**Copy this join link and save it in clipboard (copy from your output as it different for each master instance)**

Example :

```
kubeadm join 172.31.20.75:6443 --token 66kg9u.2bc0kze31hrwbzvr \
    --discovery-token-ca-cert-hash
sha256:5e478da328b199e17d9b5da68e78bc9a6daab2043b05860552f4c184a7b3cb66
```

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

**Command:**

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

```
[ec2-user@ip-172-31-26-2 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-26-2 docker]$ █
```

## 6. Perform this ONLY on the worker machines

Paste the below command on all 2 worker machines

- sudo yum install iproute-tc -y
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

```
ec2-user@ip-172-31-212-docker:~$ sudo yum install iproute-tc -y
Last metadata expiration check: 0:15:14 ago on Thu Sep 12 14:09:10 2024.
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
installing:
iproute-tc        x86_64      5.10.0-2.amzn2023.0.5
                                                               amazonlinux 455 k
transaction Summary
install 1 Package
=====
total download size: 455 k
installed size: 928 k
amazonlinux
iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm
=====
total
running transaction check
transaction check succeeded.
running transaction test
transaction test succeeded.
running transaction
preparing transaction
installing iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
Verifying iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
=====
installed:
iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
=====
complete!
ec2-user@ip-172-31-212-docker:~ [ ]
```

```
[ec2-user@ip-172-31-17-184 docker]$ sudo systemctl enable kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[ec2-user@ip-172-31-17-184 docker]$ sudo systemctl restart kubelet
[ec2-user@ip-172-31-17-184 docker]$ [ ]
```

Now paste the hash that you copied in these worker notes to connect to master cluster

- kubeadm join 172.31.20.75:6443 --token 66kg9u.2bc0kze31hrwbzvr \
   
 --discovery-token-ca-cert-hash
   
 sha256:5e478da328b199e17d9b5da68e78bc9a6daab2043b05860552f4c184a7b3cb66

Now we can see in the master/control node of Kubernetes that worker nodes are connected by this command

- **watch kubectl get nodes**
- (in the master node instance)

Errors faced during the execution :

1. In the end kubelet might not respond or the connectivity of nodes to master might not happen
2. You can see this error

```
[ec2-user@ip-172-31-20-75 docker]$ kubectl get nodes
E0914 06:14:55.956919 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s": connection refused
E0914 06:14:55.957758 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s": connection refused
E0914 06:14:55.959507 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s": connection refused
E0914 06:14:55.960160 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s": connection refused
E0914 06:14:55.961526 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s": connection refused
[ec2-user@ip-172-31-20-75 docker]$
```

3. Try to restart the kubelet from worker instance and try the commands again

### Conclusion :

In this experiment, we set out to deploy Kubernetes on Docker by connecting a master node to two worker nodes. We encountered several issues, starting with misconfigured SSH inbound rules, which were resolved by correctly enabling the necessary access rules. It became evident that using t2.medium or t3 instances was crucial to provide adequate resources for running Kubernetes efficiently. However, despite these adjustments, the worker nodes were unable to join the cluster. While the master node was successfully initialized, the issue seemed to lie in the worker nodes, possibly due to misconfigurations in the kubelet setup or networking challenges. This included the worker nodes being unable to communicate with the master node's API server, which might have been caused by incorrect firewall settings, missing API server certificates, or errors during the kubeadm join process on the worker nodes.

## EXPERIMENT-4 ADV-DEVOPS

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

### STEPS:

1. Select Amazon linux as OS image (You can use any but then modify commands accordingly)

The screenshot shows the AWS Lambda console interface. At the top, there is a search bar containing "AMZ - nginx" and a button labeled "Add additional tags". Below this, a section titled "Application and OS Images (Amazon Machine Image)" has a "Info" link. A descriptive text explains what an AMI is: "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." Below this text is a search bar with the placeholder "Search our full catalog including 1000s of application and OS images". Under the search bar, there are two tabs: "Recents" and "Quick Start", with "Quick Start" being active. Below the tabs, there are six cards representing different AMIs: "Amazon Linux" (selected), "macOS", "Ubuntu", "Windows", "Red Hat", and "SUSE LI". Each card has its respective logo. To the right of these cards is a "Browse more AMIs" section with a magnifying glass icon and the text "Including AMIs from AWS, Marketplace and the Community". At the bottom of the screenshot, there is a footer bar with the text "Amazon Machine Image (AMI)".

2. Make ssh connection in terminal

Note: If you have directly made connection through browser then skip this part

```

quantum@machine ~/Downloads> ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-54-162-208-25.compute-1.amazonaws.com
The authenticity of host 'ec2-54-162-208-25.compute-1.amazonaws.com (54.162.208.25)' can't be established.
ED25519 key fingerprint is SHA256:1BrdxB+9Hn5KWL0YZNmzh1wg/R4s+e7QDAMBJPvf/E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-162-208-25.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #
      _###_
      \####\      Amazon Linux 2
      \###\      AL2 End of Life is 2025-06-30.
      \#/
      V~'-->
      /      A newer version of Amazon Linux is available!
      /      Amazon Linux 2023, GA and supported until 2028-03-15.
      /      https://aws.amazon.com/linux/amazon-linux-2023/
      /m'/

[ec2-user@ip-172-31-63 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-63 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-63 ~]$
```

### 3. Install Docker

**sudo dnf update**

**sudo dnf install docker**

**sudo systemctl enable docker**

**sudo systemctl start docker**

To test whether docker is successfully running, use command **sudo docker run hello-world**

```

[ec2-user@ip-172-31-24-190 ~]$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Then, configure cgroup in a daemon.json file. This allows kubernetes to manage host more efficiently

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

## **4. Install Kubernetes**

**Note:** I'm directly installing binary package you may install from package repository of your distribution

**Install CNI plugins (required for most pod network):**

```
CNI_PLUGINS_VERSION="v1.3.0"  
ARCH="amd64"  
DEST="/opt/cni/bin"  
sudo mkdir -p "$DEST"  
curl -L  
"https://github.com/containernetworking/plugins/releases/download/${CNI_PLUGINS_V  
ERSION}/cni-plugins-linux-${ARCH}-${CNI_PLUGINS_VERSION}.tgz" | sudo tar -C  
"$DEST" -xz
```

**Define the directory to download command files:**

```
DOWNLOAD_DIR="/usr/local/bin"  
sudo mkdir -p "$DOWNLOAD_DIR"
```

**Optionally install crictl (required for interaction with the Container Runtime Interface (CRI), optional for kubeadm):**

```
CRICCTL_VERSION="v1.31.0"  
ARCH="amd64"
```

```
curl -L  
"https://github.com/kubernetes-sigs/cri-tools/releases/download/${CRICTL_VERSION}/crictl-${CRICTL_VERSION}-linux-${ARCH}.tar.gz" | sudo tar -C $DOWNLOAD_DIR -xz
```

**Install kubeadm, kubelet and add a kubelet systemd service:**

```
RELEASE=$(curl -sSL https://dl.k8s.io/release/stable.txt)"  
ARCH="amd64"  
cd $DOWNLOAD_DIR  
sudo curl -L --remote-name-all  
https://dl.k8s.io/release/${RELEASE}/bin/linux/${ARCH}/{kubeadm,kubelet}  
sudo chmod +x {kubeadm,kubelet}
```

```
RELEASE_VERSION="v0.16.2"  
curl -sSL  
"https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kre/templates/latest/kubelet/kubelet.service" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" |  
sudo tee /usr/lib/systemd/system/kubelet.service  
sudo mkdir -p /usr/lib/systemd/system/kubelet.service.d  
curl -sSL  
"https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kre/templates/latest/kubeadm/10-kubeadm.conf" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" |  
sudo tee /usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
```

**Now we need to install kubectl**

Set up repository:

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo  
[kubernetes]  
name=Kubernetes  
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/  
enabled=1  
gpgcheck=1  
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo  
md.xml.key  
EOF
```

```
sudo yum install -y kubectl
```

```
ec2-user@ip-172-31-24-190 ~ $ kubectl version
Client Version: v1.31.1
Kustomize Version: v5.4.2
```

We have installed successfully installed kubernetes

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
```

```
[root@ip-172-31-24-190 bin]# sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
[root@ip-172-31-24-190 bin]#
```

Disable SELINUX

Type **sudo nano /etc/selinux/config** and set the value of **SELINUX=disabled** instead of **SELINUX=permissive**

Save the file by pressing **ctrl+o** then press enter then press **ctrl+x**

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#     grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#     grubby --update-kernel ALL --remove-args selinux
#
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Then reboot the system using **sudo reboot**

After rebooting we need to make ssh connection with machine after it gets disconnected

Now if we type command **sestatus**, then it show disabled

```
ec2-user@ip-172-31-24-190 ~ $ sestatus
SELinux status:                 disabled
```

## 5. Initialize the Kubecluster

Install packages socat and iproute-tc and conntrack to avoid prelight errors

**sudo dnf install socat iproute-tc conntrack-tools -y**

**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 172.31.24.190:6443 --token xsbsq1.6ro11sawnvttbsv \ 
--discovery-token-ca-cert-hash sha256:10d2b67f4f4749b51854065a554c74e6a956e4782d9ab4bb79b8591648b3edef
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
```

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

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

**sudo systemctl restart kubelet**

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

**kubectl apply -f**

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

```
 ubectl RC, etc. environment
ec2-user@ip-172-31-24-190 ~ $ 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
```

### **Now type kubectl get nodes**

The connection to the server 172.31.24.190:6443 was refused - did you specify the right host or port?

ec2-user@ip-172-31-24-190 ~ \$ kubectl get nodes

The connection to the server 172.31.24.190:6443 was refused - did you specify the right host or port?

ec2-user@ip-172-31-24-190 ~ \$ kubectl get nodes

**^[[AError from server (Forbidden): nodes is forbidden: User "kubernetes-admin" cannot list resource "nodes" in**

**ec2-user@ip-172-31-24-190 ~ \$ kubectl get nodes**

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-24-190.ec2.internal	Ready	control-plane	34m	v1.31.0

ec2-user@ip-172-31-24-190 ~ \$ kubectl get nodes

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-24-190.ec2.internal	Ready	control-plane	34m	v1.31.0

ec2-user@ip-172-31-24-190 ~ \$

**Note: If any time of get error of connection refused just restart the kubelet service (sudo systemctl restart kubelet)**

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

```
ec2-user@ip-172-31-24-190 ~ $ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

Use 'kubectl get pods' to verify if the deployment was properly created and the pod is working correctly.

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-mwd8p   0/1     Pending   0          7s
nginx-deployment-d556bf558-zc25s   0/1     Pending   0          7s
```

As we can see our pods are in pending state

On checking logs to we came to know the pods are in tainted state (using command **kubectl describe pod nginx-deployment-d556bf558-mwd8p**)

```
Events:
Type  Reason           Age   From            Message
----  ----             --   --              --
Warning FailedScheduling 56s   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:
taint for scheduling
```

To make pods untainted

Type kubectl get nodes to see name of node

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
NAME                  STATUS   ROLES      AGE   VERSION
ip-172-31-24-190.ec2.internal   Ready    control-plane   43m   v1.31.0
ec2-user@ip-172-31-24-190 ~ $
```

Copy the name of the node (ip-172-31-24-190.ec2.internal)

Then type command **kubectl taint nodes <NODE\_NAME> - -all**

In my case **kubectl taint nodes ip-172-31-24-190.ec2.internal node-role.kubernetes.io/control-plane-**

```
ec2-user@ip-172-31-24-190 ~ $ kubectl taint nodes ip-172-31-24-190.ec2.internal node-role.kubernetes.io/control-plane-
node/ip-172-31-24-190.ec2.internal untainted
```

After executing above command, check again status of pods if still pending then restart kubelet wait for 1-2 minutes and check again

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-mwd8p   1/1     Running   2 (73s ago)   12m
nginx-deployment-d556bf558-zc25s   1/1     Running   2 (73s ago)   12m
```

As we can see our pods are running

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

**kubectl port-forward <POD\_NAME> 8080:80**

In my case : **kubectl port-forward nginx-deployment-d556bf558-mwd8p 8080:80**

Note: if you are getting connection refused error then restart kubelet

```
ec2-user@ip-172-31-24-190 ~ $ kubectl port-forward nginx-deployment-d556bf558-mwd8p 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

As port forwarding is active so we cannot type other commands.

Open new terminal window and make ssh connection to same machine OR we can open instance of same machine in new browser tab

And type command **curl --head http://127.0.0.1:8080**

```
ec2-user@ip-172-31-24-190 ~ $ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 14 Sep 2024 06:54:21 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

ec2-user@ip-172-31-24-190 ~ $ █
3 1:ec2-user@ip-172-31-24-190:~# - 2:ec2-user@ip-172-31-24-190:~* 3:~/Downloads
```

Response status 200 (OK) indicates that our nginx server is running successfully on kubernetes

**Conclusion:** We started by installing and setting up Docker and Kubernetes. Initially, there were issues with the Kubernetes API server, but restarting the kubelet service resolved them. The pods weren't running at first due to node tainting, which we corrected by untainting the nodes. After addressing all the errors, we successfully deployed the NGINX server pods, which are now accessible via the forwarded port. The NGINX server can be accessed either through different terminals or by running the port forwarding process in the background, achieved by appending `&` to the command.

Name: Kshitij Hundre

Div: D15C

Roll No:18

### Step 1) Install Terraform

The screenshot shows the Terraform download page. At the top right is a dropdown menu set to "1.9.4 (latest)". Below it, under "macOS", there's a "Package manager" section with the command "brew tap hashicorp/tap" and "brew install hashicorp/tap/terraform". Under "Binary download", there are two options: "AMD64" (Version: 1.9.4) and "ARM64" (Version: 1.9.4), each with a "Download" button. Below this, under "Windows", there's a "Binary download" section with "386" (Version: 1.9.4) and "AMD64" (Version: 1.9.4), each with a "Download" button.

### Step 2) Setup path in environment variables.

The screenshot shows the Windows "Environment Variables" dialog. It has two main sections: "User variables for 91773" and "System variables".

**User variables for 91773:**

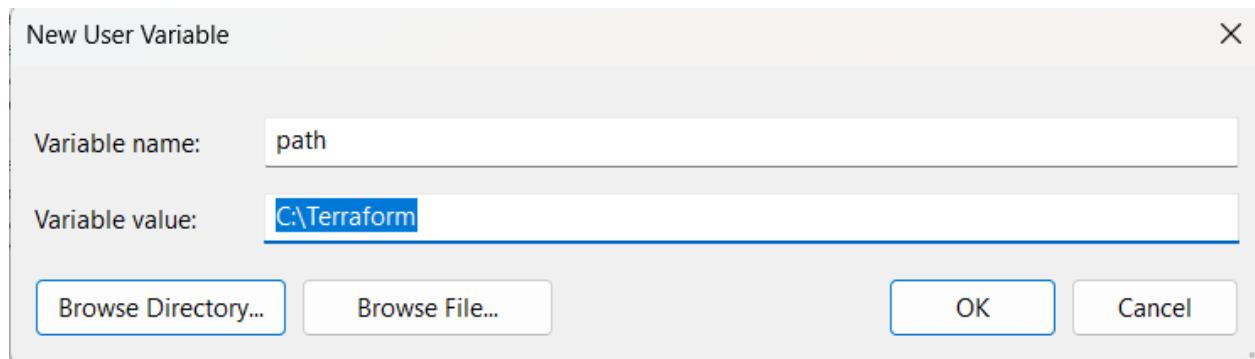
Variable	Value
IntelliJ IDEA Community E...	C:\Users\91773\OneDrive\Desktop\java neew\IntelliJ IDEA Co...
JAVA_HOME	C:\Users\91773\AppData\Local\Programs\Eclipse Adoptium\j...
OneDrive	C:\Users\91773\OneDrive
OneDriveConsumer	C:\Users\91773\OneDrive
Path	C:\Users\91773\AppData\Local\Programs\Eclipse Adoptium\j...
TEMP	C:\Users\91773\AppData\Local\Temp
TMP	C:\Users\91773\AppData\Local\Temp

**System variables:**

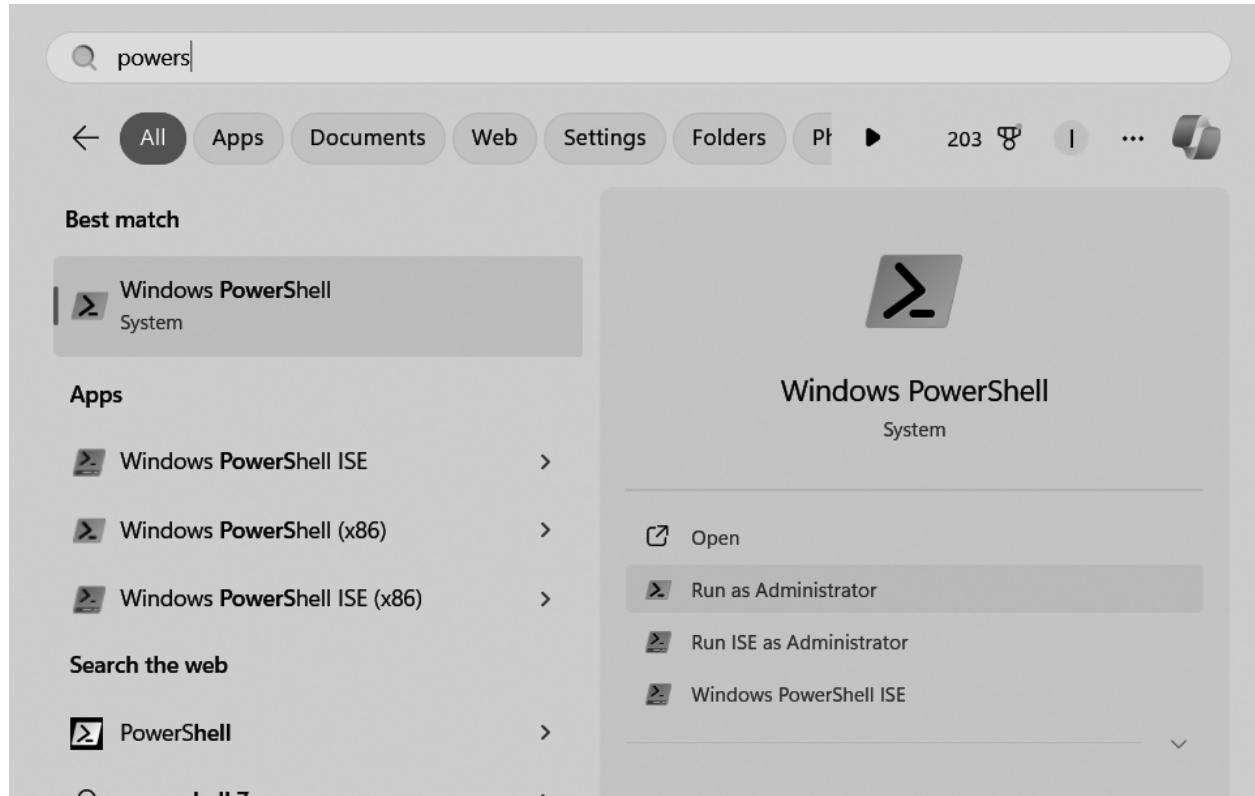
Variable	Value
ACSetupSvcPort	23210
ACSvcPort	17532
ANDROID_HOME	D:\Flutter Dev\ANDROID_SDK
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
EnableLog	INFO
JAVA_HOME	D:\Flutter Dev\JDK
NUMBER_OF_PROCESSORS	12

At the bottom are "New...", "Edit...", "Delete", "OK", and "Cancel" buttons.

Step 3) Select the C drive Terraform folder as variable value.



Step 4) Open Windows powershell as Administrator.



## Step 5) Run the Terraform command in powershell.

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

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

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

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

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

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

Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
              given subcommand.
  -help       Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.
PS C:\WINDOWS\system32> ■
```

Name: KSHITIJ HUNDRE  
Div: D15C  
Roll NO:18

# Experiment No.: 6

## Implementation:

### A. Creating docker image using terraform

Prerequisite:

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

**Step 1:** Check the docker functionality

```
PS C:\Users\91773> 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
```

```
PS C:\Users\91773> docker --version
Docker version 27.0.3, build 7d4bcd8
PS C:\Users\91773> |
```

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

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

Script:

```
terraform {  
    required_providers {  
        docker = {  
            source = "kreuzwerker/docker"  
            version = "2.21.0"  
        }  
    }  
}  
  
provider "docker" {  
    host = "npipe:///./pipe/docker_engine"  
}  
  
# Pull the image  
resource "docker_image" "ubuntu" {  
    name = "ubuntu:latest"  
}  
  
# Create a container  
resource "docker_container" "foo" {  
    image = docker_image.ubuntu.image_id  
    name = "foo"  
    command = ["sleep", "3600"]  
}
```

```
docker.tf  X
```

```
docker.tf
```

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

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

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
    https://www.terraform.io/docs/cli/plugins/signing.html
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

### Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
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)
```

```

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

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

```

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

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

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad76
tu:latest]

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "3600",
    ]
}

```

Docker images, Before Executing Apply step:

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

```

Docker images, After Executing Apply step:

```

appy complete: resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       edbfe74c41f8   2 weeks ago   78.1MB
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> |

```

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

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a]
ubuntu:latest]
docker_container.foo: Refreshing state... [id=f03a28e4658896c23c9992f7a98eb1011befc7d014e997ea9fc6372da70b7903]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
  - attach                  = false -> null
  - command                 = [
    - "sleep",
    - "3600",
  ] -> null
  - cpu_shares              = 0 -> null
}

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=f03a28e4658896c23c9992f7a98eb1011befc7d014e997ea9fc6372da70b7903]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a]
est]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.

```

Docker images After Executing Destroy step

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE

```

## Adv DevOps Practical 7

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

### Integrating Jenkins with SonarQube:

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

### Steps to integrate Jenkins with SonarQube

1. Open up Jenkins Dashboard on localhost, port 8090 or whichever port it is at for you.

The screenshot shows the Jenkins dashboard interface. On the left, there's a sidebar with links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. Below these are sections for Build Queue and Build Executor Status, both with expand/collapse arrows. The main area displays a table of projects with columns for Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration. The projects listed are: devops cicd pipes (Status: Green, Last Success: 1 mo 22 days, Last Failure: N/A, Duration: 6.4 sec), devops pipeline (Status: Red, Last Success: N/A, Last Failure: 1 mo 22 days, Duration: 57 ms), Kspipeline (Status: Green, Last Success: 1 mo 6 days, Last Failure: N/A, Duration: 5.7 sec), maven-tomcat1 (Status: Green, Last Success: 1 mo 0 days, Last Failure: N/A, Duration: 1 min 19 sec), and mavenproj (Status: Red, Last Success: N/A, Last Failure: 1 mo 22 days, Duration: 1 min 36 sec). At the bottom of the dashboard, there are icons for S, M, and L, and a 'Add description' button.

S	W	Name ↓	Last Success	Last Failure	Last Duration
Green	Sun	devops cicd pipes	1 mo 22 days #3	N/A	6.4 sec
Red	Cloud	devops pipeline	N/A	1 mo 22 days #10	57 ms
Green	Sun	Kspipeline	1 mo 6 days #1	N/A	5.7 sec
Green	Sun	maven-tomcat1	1 mo 0 days #1	N/A	1 min 19 sec
Red	Cloud	mavenproj	N/A	1 mo 22 days #1	1 min 36 sec

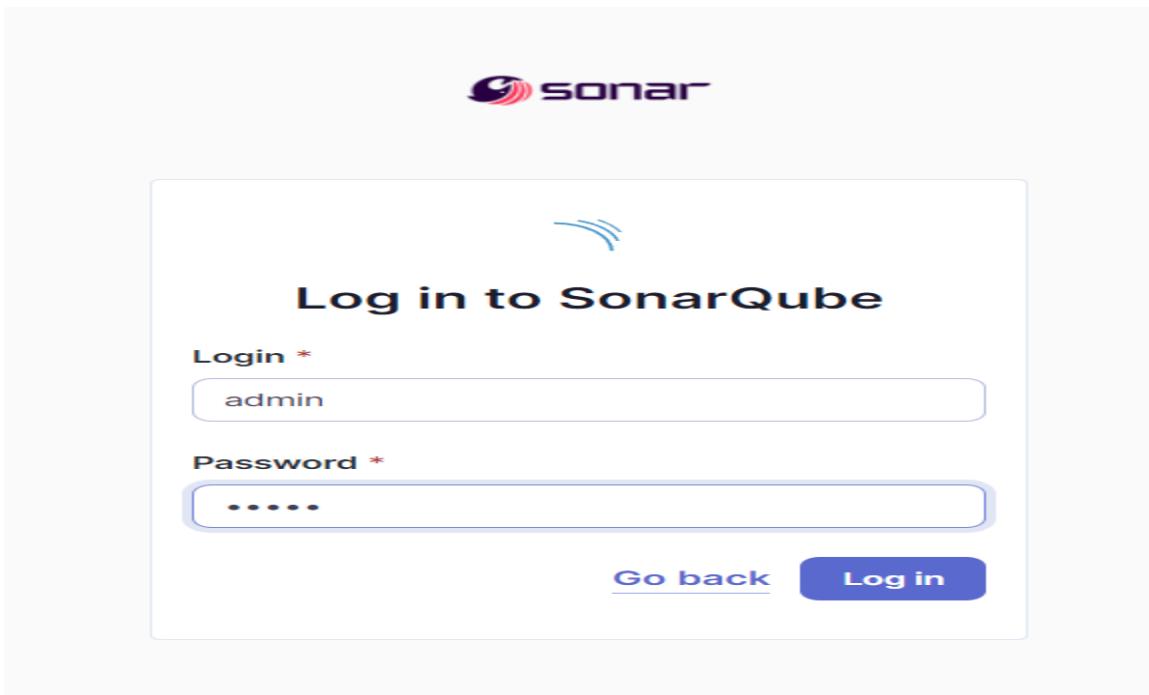
2. Run SonarQube in a Docker container using this command -

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

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

```
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
77e678cded2ef5f989912d3d9e6991dd548eac03faaleed68dd906614be53acc
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops>
```

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



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

The screenshot shows the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, a heading asks "How do you want to create your project?". It provides several options: "Import from Azure DevOps" (Setup), "Import from Bitbucket Cloud" (Setup), "Import from Bitbucket Server" (Setup), "Import from GitHub" (Setup), "Import from GitLab" (Setup), and a "Create a local project" button. A note at the bottom says "Are you just testing or have an advanced use-case? Create a local project."

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

This screenshot shows the "Create a local project" form. At the top left, it says "1 of 2". The main title is "Create a local project". The first field is "Project display name \*", with the value "exp7" entered. The second field is "Project key \*", also with the value "exp7". The third field is "Main branch name \*", with the value "main". Below these fields is a note: "The name of your project's default branch [Learn More](#)". At the bottom are two buttons: "Cancel" and a blue "Next" button.

Setup the project and come back to Jenkins Dashboard.

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

Plugins

Available plugins

SonarQube Scanner 2.17.2

External Site/Tool Integrations Build Reports

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

Sonar Quality Gates 315.vf1f12b\_e81a\_3a\_4

Library plugins (for use by other plugins) analysis Other Post-Build Actions

Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")

Quality Gates 2.5

Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")

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

Enter the Server Authentication token if needed.

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

**sahilexp7**

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

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	<input type="text" value="exp7"/>	<span style="color: red;">X</span>
Server URL	Default is <a href="http://localhost:9000">http://localhost:9000</a>	
	<input type="text" value="http://localhost:9000"/>	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled.	
	<input type="text" value="- none -"/> <span style="float: right;">▼</span>	
	<span style="border: 1px solid #ccc; padding: 2px;">+ Add ▾</span>	
<span style="border: 1px solid #ccc; padding: 2px;">Advanced ▾</span>		

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

**Dashboard > Manage Jenkins > Tools**

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for:

- Gradle installations:** Contains a 'Add Gradle' button.
- SonarScanner for MSBuild installations:** Contains a 'Add SonarScanner for MSBuild' button.
- SonarQube Scanner installations:** Contains a 'Add SonarQube Scanner' button.
- Ant installations:** Contains a 'Add Ant' button.

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

The screenshot shows the 'SonarQube Scanner' configuration dialog. It includes fields for:

- Name:** sonarqube\_exp7
- Install automatically:** A checked checkbox.
- Install from Maven Central:** A section containing:
  - Version:** SonarQube Scanner 6.2.0.4584
  - Add Installer:** A dropdown menu.
- Add SonarQube Scanner:** A button at the bottom.

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

**New Item**

Enter an item name  
ks\_exp7

Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple

**OK**

9. Choose this GitHub repository in Source Code Management.

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

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

Dashboard > exp7 > Configuration

**Source Code Management**

None

Git [?](#)

**Repositories** [?](#)

**Repository URL** [?](#)  [X](#)

**Credentials** [?](#)

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

The screenshot shows the Jenkins configuration interface for a job named 'exp7'. In the 'Build steps' section, the 'Execute SonarQube Scanner' option is selected, indicated by a blue border around its dropdown menu. The menu lists several other build step options: Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, Set build status to "pending" on GitHub commit, SonarScanner for MSBuild - Begin Analysis, and SonarScanner for MSBuild - End Analysis. Below the build steps, there is a 'Post-build Actions' section with a 'Save' button highlighted in blue. The main configuration area is expanded to show the 'Execute SonarQube Scanner' step's configuration details:

- JDK**: A dropdown menu set to 'JDK 17'.
- Path to project properties**: An empty input field.
- Analysis properties**: A code editor containing the following SonarQube analysis properties:

```
sonar.projectKey=ks_exp7
sonar.projectName=ks_exp7
sonar.projectVersion=1.0
sonar.sources=C:/ProgramData/Jenkins/jenkins/workspace/ks_exp7
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.password=kshitij24
```
- Additional arguments**: An empty input field.

Status: ks\_exp7

SonarQube

Permalinks

- Last build (#7), 4 min 55 sec ago
- Last stable build (#7), 4 min 55 sec ago
- Last successful build (#7), 4 min 55 sec ago
- Last failed build (#6), 17 min ago
- Last unsuccessful build (#6), 17 min ago
- Last completed build (#7), 4 min 55 sec ago

Build History: trend

#7 | Sep 25, 2024, 3:09 PM

## Console Output

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

```

Started by user Kshitij Hundre
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git' version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[ks_exp7] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube1_exp7\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=ks_exp7 -Dsonar.projectName=ks_exp7 -Dsonar.host.url=http://localhost:9000 -
Dsonar.login=admin -Dsonar.projectVersion=1.0 -Dsonar.sources=C:/ProgramData/Jenkins/.jenkins/workspace/ks_exp7 -Dsonar.password=kshitij24 -
Dsonar.projectBaseDir=C:/ProgramData/Jenkins/.jenkins/workspace/ks_exp7
15:09:08.473 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
```

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

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

4 of 4 shown

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

The screenshot shows the SonarQube interface for the 'main' project. At the top, there's a navigation bar with tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information dropdowns. Below the navigation, the project name 'main' is displayed along with its version 'Version 1.0'. A large green 'Passed' button indicates the quality gate status. To the right of the button, it says 'Last analysis 10 minutes ago'. Under the 'Overall Code' tab, there are four cards: 'Security' (0 open issues), 'Reliability' (0 open issues), and 'Maintainability' (0 open issues), all marked with an 'A'. A yellow warning box at the bottom left states: '⚠️ The last analysis has warnings. See details'.

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

### Conclusion:

In this project, we integrated Jenkins with SonarQube for automated static application security testing (SAST). We set up SonarQube using Docker, configured Jenkins with the necessary plugins and authentication, and linked it to a GitHub repository. The SonarQube scanner was added as a build step, enabling continuous code analysis for vulnerabilities, code smells, and quality issues, ensuring automated reporting and continuous code quality improvement.

## Expt No. 08 Advanced DevOps Lab

**Aim:** Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

### **Theory:**

#### **What is SAST?**

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

#### **What problems does SAST solve?**

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

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

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

#### **Why is SAST important?**

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster

than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.

### What is a CI/CD Pipeline?

CI/CD pipeline refers to the Continuous Integration/Continuous Delivery pipeline. Before we dive deep into this segment, let's first understand what is meant by the term 'pipeline'?

A pipeline is a concept that introduces a series of events or tasks that are connected in a sequence to make quick software releases. For example, there is a task, that task has got five different stages, and each stage has got some steps. All the steps in phase one have to be completed, to mark the latter stage to be complete.



Now, consider the CI/CD pipeline as the backbone of the DevOps approach. This Pipeline is responsible for building codes, running tests, and deploying new software versions. The Pipeline executes the job in a defined manner by first coding it and then structuring it inside several blocks that may include several steps or tasks.

## What is SonarQube?

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

## Benefits of SonarQube

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- **Increase productivity** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- **Quality code** - Code quality control is an inseparable part of the process of software development.
- **Detect Errors** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- **Increase consistency** - Determines where the code criteria are breached and enhances the quality
- **Business scaling** - No restriction on the number of projects to be evaluated
- **Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills

## Integrating Jenkins with SonarQube:

### Prerequisites:

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

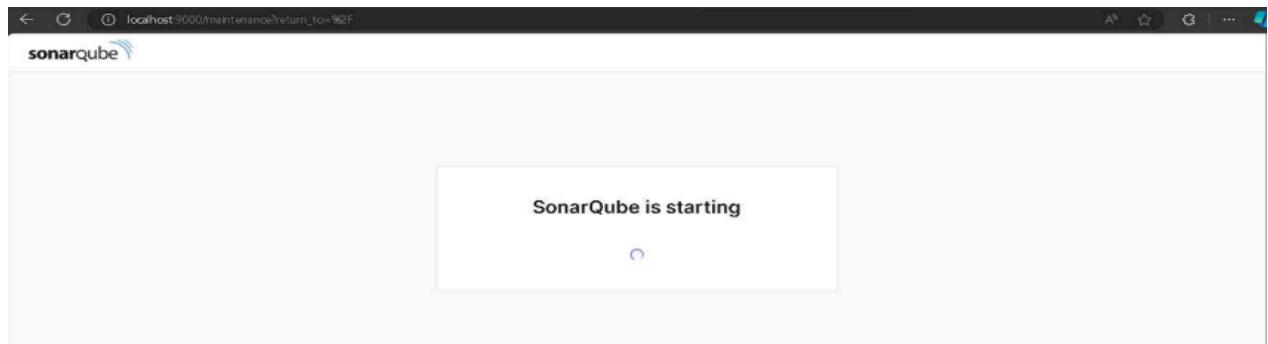
## Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

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

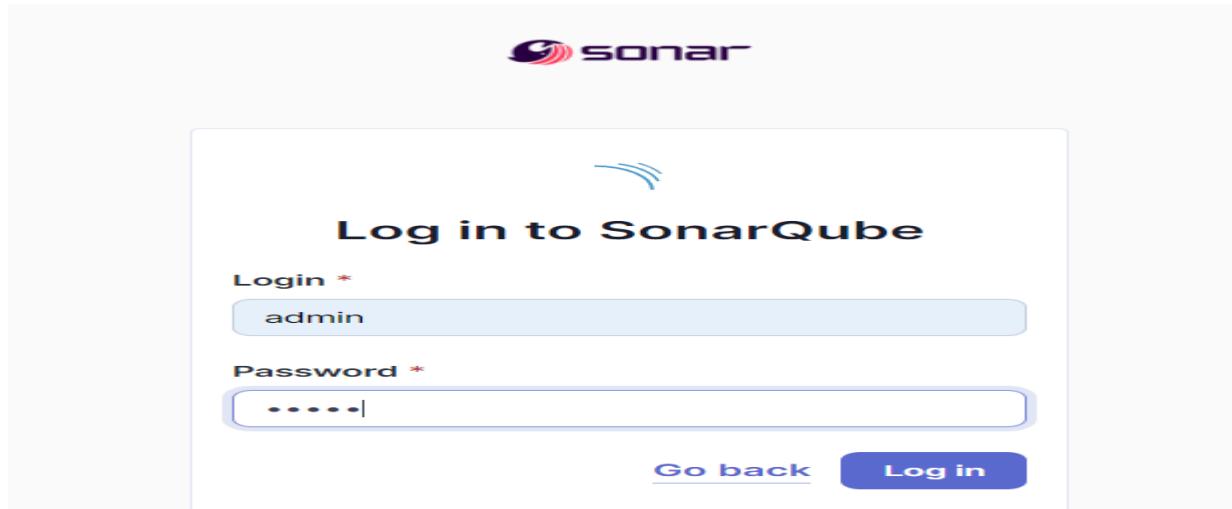
2. Run SonarQube in a Docker container using this command -

```
| PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8> docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS
| _DISABLE=true -p 9000:9000 sonarqube:latest
| 71fc67f0b15baa5be5bdccdd66966938e18682683d020beadc9c909dd027cfe7a
PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8>
```

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



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



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

1 of 2

## Create a local project

**Project display name \***

**Project key \***

**Main branch name \***

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

[Cancel](#)

[Next](#)

Setup the project and come back to Jenkins Dashboard.

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

Enter an item name

Select an item type



**Freestyle project**

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



**Maven project**

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



**Pipeline**

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



**Multi-configuration project**

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

[OK](#)

7. Under Pipeline Script, enter the following -

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
    stage('SonarQube analysis') {  
        withSonarQubeEnv('sonarqube') {  
            sh "<PATH_TO SONARQUBE FOLDER>/bin//sonar-scanner \  
-D sonar.login=<SonarQube_USERNAME> \  
-D sonar.password=<SonarQube_PASSWORD> \  
-D sonar.projectKey=<Project_KEY> \  
-D sonar.exclusions=vendor/**,resources/**,**/*.java \  
-D sonar.host.url=http://127.0.0.1:9000/"  
        }  
    }  
}
```

Configure

The screenshot shows the Jenkins Pipeline configuration interface. On the left, there are tabs for General, Advanced Project Options, and Pipeline. The Pipeline tab is selected, and the sub-tab Pipeline script is also selected. The main area contains a code editor with the following Groovy script:

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
    stage('SonarQube Analysis') {  
        withSonarQubeEnv('exp8') {  
            sh """  
                C:\Program Files\Sonar Scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat  
                -Dsonar.login=admin  
                -Dsonar.password=kshitij24  
                -Dsonar.projectKey=sonarqube-test  
                -Dsonar.exclusions=vendor/**,resources/**,**/*.java  
                -Dsonar.host.url=http://127.0.0.1:9000  
            """  
        }  
    }  
}
```

Below the code editor, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom of the screen are "Save" and "Apply" buttons.

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

8. Run The Build.

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

**KsSonarQube**

- </> Changes
- ▷ Build Now
- ⚙ Configure
- 🗑 Delete Pipeline
- 🔍 Full Stage View
- ⚡ SonarQube
- 📦 Stages
- ✍ Rename
- ❓ Pipeline Syntax

**Stage View**

Cloning the GitHub Repo	SonarQube Analysis
2s	1min 44s
2s	8min 33s
4s	835ms failed
2s	3s failed
2s	3s failed

Average stage times:  
(Average full run time: ~8min 36s)

#9 Sep 25 20:49 No Changes  
#8 Sep 25 20:44 No Changes  
#7 Sep 25 20:42 No Changes  
#6 Sep 25 20:31 No Changes

**Build History**

trend ▾

Filter... /

#9 Sep 25, 2024, 8:49 PM

**Console Output**

- </> Changes
- Console Output
- View as plain text
- Edit Build Information
- >Delete build '#9'
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Replay
- Pipeline Steps
- Workspaces
- ← Previous Build

Skipping 4,247 KB.. [Full Log](#)

```

20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
```

Kshitij Hundre > My Views > All > KsSonarQube > #9

```

for block at line 17. Keep only the first 100 references.
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
20:56:18.456 INFO CPD Executor CPD calculation finished (done) | time=107093ms
20:56:18.490 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:57:50.106 INFO Analysis report generated in 3149ms, dir size=127.2 MB
20:57:56.943 INFO Analysis report compressed in 6828ms, zip size=29.6 MB
20:57:58.685 INFO Analysis report uploaded in 1732ms
20:57:58.688 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
20:57:58.688 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:57:58.688 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=18847db4-4f06-4766-9ad4-ee006448353c
20:58:06.225 INFO Analysis total time: 8:22.672 s
20:58:06.231 INFO SonarScanner Engine completed successfully
20:58:06.824 INFO EXECUTION SUCCESS
20:58:06.857 INFO Total time: 8:31.713s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## 10. After that, check the project in SonarQube.

sonarqube-test / main

**main**

Quality Gate **Passed**

683k Lines of Code - Version not provided - Set as homepage

Last analysis 16 minutes ago

New Code      Overall Code

Security	Reliability	Maintainability
0 Open issues	68k Open issues	164k Open issues
0 H 0 M 0 L	0 H 47k M 21k L	7 H 143k M 21k L

Accepted issues	Coverage	Duplications
0	On 0 lines to cover.	50.6% On 759k lines.

Valid Issues that were not fixed

Security Hotspots  
3

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

## 11. Bugs

The screenshot shows a software interface for managing code quality issues. On the left, there is a sidebar with navigation links: Responsibility, Software Quality, Severity, Type, Scope, Status, and Security Category. Under Type, 'Bug' is selected, showing 33k issues. Other options include Vulnerability (0) and Code Smell (164k). The main area displays three tabs of issues:

- gameoflife-core/build/reports/tests/all-tests.html**: Contains an issue titled "Insert a <!DOCTYPE> declaration to before this <html> tag." with a severity of Reliability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and user-experience. Effort: L1 = 5min effort × 4 years ago × ⚡ Bug × ⚡ Major.
- gameoflife-core/build/reports/tests/allclasses-frame.html**: Contains an issue titled "Insert a <!DOCTYPE> declaration to before this <html> tag." with a severity of Reliability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and user-experience. Effort: L1 = 5min effort × 4 years ago × ⚡ Bug × ⚡ Major.
- gameoflife-core/build/reports/tests/alltests-errors.html**: Contains an issue titled "Insert a <!DOCTYPE> declaration to before this <html> tag." with a severity of Reliability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and user-experience. Effort: L1 = 5min effort × 4 years ago × ⚡ Bug × ⚡ Major.

## Code Smells

The screenshot shows a software interface for managing code quality issues. On the left, there is a sidebar with navigation links: Responsibility, Software Quality, Severity, Type, Scope, Status, and Security Category. Under Type, 'Code Smell' is selected, showing 164k issues. Other options include Bug (33k) and Vulnerability (0). The main area displays four tabs of issues:

- gameoflife-core/build/reports/tests/all-tests.html**: Contains an issue titled "Remove this deprecated 'width' attribute." with a severity of Maintainability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and html5 obsolete. Effort: L9 = 5min effort × 4 years ago × ⚡ Code Smell × ⚡ Major.
- gameoflife-core/build/reports/tests/alltests-errors.html**: Contains an issue titled "Remove this deprecated 'align' attribute." with a severity of Maintainability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and html5 obsolete. Effort: L11 = 5min effort × 4 years ago × ⚡ Code Smell × ⚡ Major.
- gameoflife-core/build/reports/tests/alltests-errors.html**: Contains an issue titled "Remove this deprecated 'size' attribute." with a severity of Maintainability (Medium), status Open, and assigned to Not assigned. It is categorized under Consistency and html5 obsolete. Effort: L12 = 5min effort × 4 years ago × ⚡ Code Smell × ⚡ Major.

## Intentional issues

The screenshot shows the SonarQube interface for the project `gameoflife-acceptance-tests/Dockerfile`. The left sidebar displays navigation links like 'Issues in new code', 'Clean Code Attribute', 'Software Quality', 'Severity', and 'Type'. Under 'Type', 'Bug' is selected. The main panel shows four intentional issues:

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

## Reliability issues

The screenshot shows the SonarQube interface for the project `gameoflife-core/build/reports/tests/all-tests.html`. The left sidebar displays navigation links like 'Issues in new code', 'Clean Code Attribute', 'Software Quality', 'Severity', and 'Type'. Under 'Type', 'Reliability' is selected. The main panel shows four reliability issues:

- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency, Reliability) - L1 • 5min effort • 4 years ago • Bug • Major
- Anchors must have content and the content must be accessible by a screen reader.** (Consistency, Reliability) - L29 • 5min effort • 4 years ago • Code Smell • Minor
- Anchors must have content and the content must be accessible by a screen reader.** (Consistency, Reliability) - L38 • 5min effort • 4 years ago • Code Smell • Minor
- Anchors must have content and the content must be accessible by a screen reader.** (Consistency, Reliability) - No tags

## Duplicates



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

### **Conclusion:**

In this experiment, we integrated Jenkins with SonarQube to enable automated code quality checks within our CI/CD pipeline. We started by deploying SonarQube using Docker, setting up a project, and configuring it to analyze code quality. Next, we configured Jenkins by installing the SonarQube Scanner plugin, adding SonarQube server details, and setting up the scanner tool. We then developed a Jenkins pipeline to automate the process of cloning a GitHub repository and running SonarQube analysis on the code. This integration helps ensure continuous monitoring of code quality, detecting issues such as bugs, code smells, and security vulnerabilities throughout the development process.

Name : Kshitij Hundre  
Div: D15C  
Roll No:18

## Adv-Devops Exp 9

Steps to perform the experiment:

Step1) Create an EC2 instance. keep the settings as default.

**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

nagios\_host\_exp9\_kcs

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

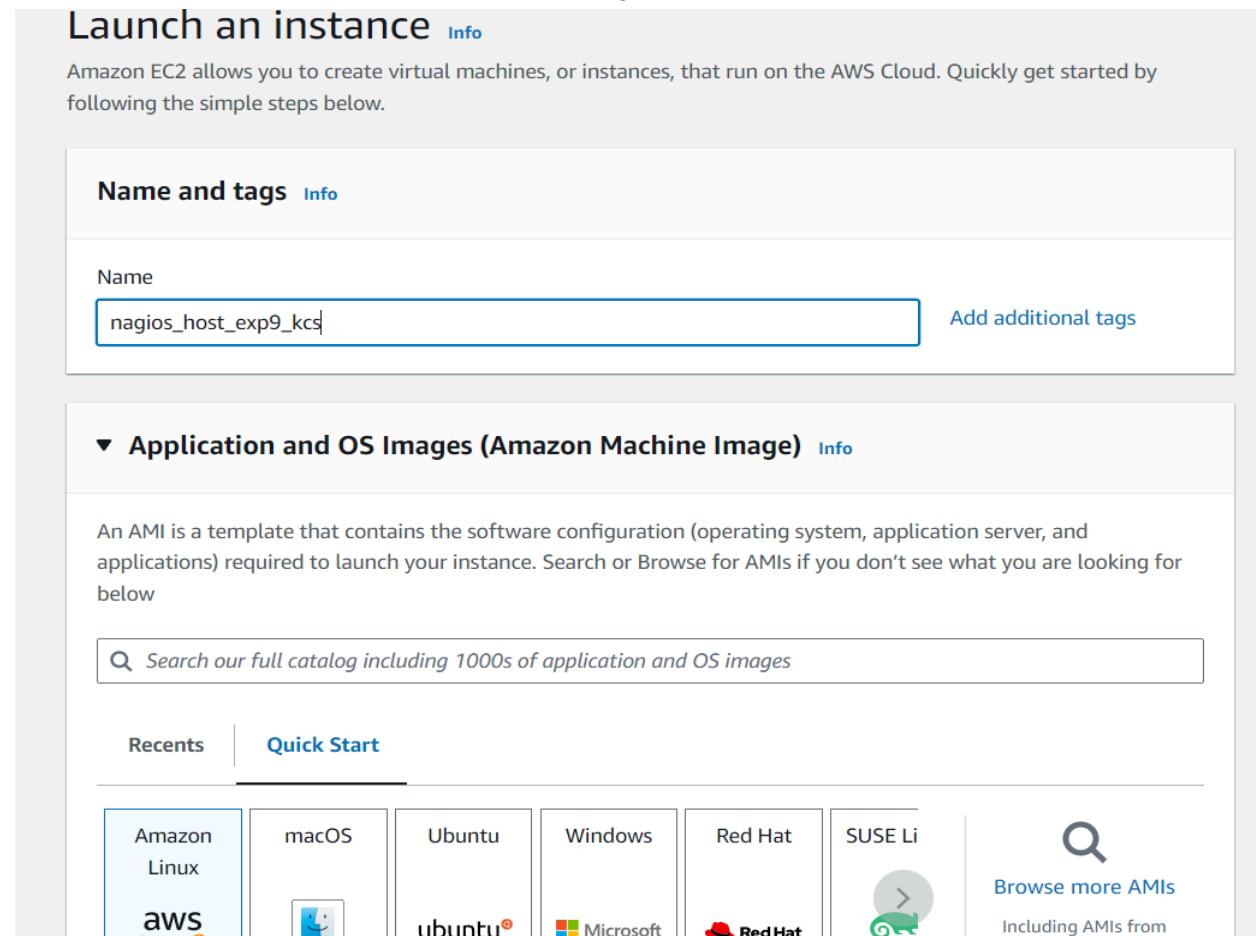
Search our full catalog including 1000s of application and OS images

Recents      Quick Start

Amazon Linux    macOS    Ubuntu    Windows    Red Hat    SUSE Li

aws                ubuntu®    Microsoft    Red Hat    

 Browse more AMIs  
Including AMIs from



Create a new key pair login and save the downloaded file in a folder of your local desktop.

Also create a new security group. In my case its name will 'launch-wizard-10'.  
Later we will edit rules of this security group.

**▼ 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](#)

**▼ Network settings [Info](#)**

Edit

Network [Info](#)  
vpc-0a482134962ed0c59

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

We'll create a new security group called '**launch-wizard-10**' with the following rules:



**Exp9 advdevops** X +

← → ↑ ↓ ⌂ Start backup > College Resources > Exp9 advdevops

New X C Sort View ...

Home	Name	Date modified	Type	Size
Gallery	nagios_exp9.pem	29-09-2024 11:43	PEM File	2 KB

Now to edit security groups, select your security group and click on edit inbound rules. Add these security rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	HTTP	TCP	80	Anywhere... ▾	Q. 0.0.0.0/0 ▾ :/0 X
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere... ▾	Q. 0.0.0.0/0 ▾ :/0 X
-	HTTPS	TCP	443	Anywhere... ▾	Q. 0.0.0.0/0 X
-	All traffic	All	All	Anywhere... ▾	Q. 0.0.0.0/0 X
-	Custom TCP	TCP	5666	Anywhere... ▾	Q. 0.0.0.0/0 X
-	All ICMP - IPv4	ICMP	All	Anywhere... ▾	Q. 0.0.0.0/0 X
-	SSH	TCP	22	Custom ▾	Q. 0.0.0.0 X

[Add rule](#)

Details			
Security group name <a href="#">launch-wizard-10</a>	Security group ID <a href="#">sg-070b8d0a96b7916ca</a>	Description <a href="#">launch-wizard-10 created 2024-09-29T06:09:17.335Z</a>	VPC ID <a href="#">vpc-0a482134962ed0c59</a>
Owner <a href="#">996474913977</a>	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	<a href="#">Actions ▾</a>

now navigate to instances, click on the instance which was created earlier and click on connect. now copy the ssh command and just replace the .pem file with its actual location in your computer.

EC2 > Instances > i-00cad8f00eebfe889 > Connect to instance

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

Connect to your instance i-00cad8f00eebfe889 (nagios\_host\_exp9\_kcs) using any of these options

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

Instance ID  
[i-00cad8f00eebfe889 \(nagios\\_host\\_exp9\\_kcs\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is `nagios_exp9.pem`
- Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 "nagios_exp9.pem"`
- Connect to your instance using its Public DNS:  
`ec2-54-164-8-234.compute-1.amazonaws.com`

Example:  
`ssh -i "nagios_exp9.pem" ec2-user@ec2-54-164-8-234.compute-1.amazonaws.com`

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

[Cancel](#)

ssh -i "nagios\_exp9.pem" [ec2-user@ec2-54-164-8-234.compute-1.amazonaws.com](mailto:ec2-user@ec2-54-164-8-234.compute-1.amazonaws.com)...paste this command in terminal..just replace your.pem file path.

```
PS C:\Users\91773\Desktop\College Resources\Exp9 advdevops> ssh -i "C:\Users\91773\Desktop\College Resources\Exp9 advdevops\nagios_exp9.pem" ec2-user@ec2-54-164-8-234.compute-1.amazonaws.com
The authenticity of host 'ec2-54-164-8-234.compute-1.amazonaws.com (54.164.8.234)' can't be established.
ED25519 key fingerprint is SHA256:scPnbaAMCV+FHEbUBCnwZdAj6MQStqfeD/Rh06wDSY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-164-8-234.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #_
      ~\_ #####_
      ~~ \#####\
      ~~ \###|
      ~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
      ~~ V~' `-->
      ~~~ /`_
      ~~ .-/`_/
      ~~ /`_/
      _/`_/
      _/m/`_
```

sudo yum update

```
-/ \ \ [ec2-user@ip-172-31-86-195 ~]$ sudo yum update
Last metadata expiration check: 0:32:14 ago on Sun Sep 29 06:16:51 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-86-195 ~]$
```

sudo yum install httpd php

Select y when asked i prompt.

```
[ec2-user@ip-172-31-86-195 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:33:12 ago on Sun Sep 29 06:16:51 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    48 k
php8.3           x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   10 k
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2   amazonlinux   129 k
apr-util         x86_64      1.6.3-1.amzn2023.0.1   amazonlinux   98 k
generic-logos-httd x86_64      18.0.0-12.amzn2023.0.3  amazonlinux   19 k
httpd-core       x86_64      2.4.62-1.amzn2023   amazonlinux   1.4 M
httpd-filesystem x86_64      2.4.62-1.amzn2023   amazonlinux   14 k
httpd-tools      x86_64      2.4.62-1.amzn2023   amazonlinux   81 k
libbrotli        x86_64      1.0.9-4.amzn2023.0.2   amazonlinux   315 k
libsodium         x86_64      1.0.19-4.amzn2023   amazonlinux   176 k
libxslt          x86_64      1.1.34-5.amzn2023.0.2  amazonlinux   241 k
mailcap          noarch     2.1.49-3.amzn2023.0.3  amazonlinux   33 k
nginx-filesystem x86_64      1:1.24.0-1.amzn2023.0.4  amazonlinux   9.8 k
php8.3-cli       x86_64      8.3.10-1.amzn2023.0.1   amazonlinux   3.7 M
php8.3-common    x86_64      8.3.10-1.amzn2023.0.1   amazonlinux   737 k
php8.3-process   x86_64      8.3.10-1.amzn2023.0.1   amazonlinux   45 k
php8.3-xml       x86_64      8.3.10-1.amzn2023.0.1   amazonlinux   154 k
Installing weak dependencies:
apr-util-openssl x86_64      1.6.3-1.amzn2023.0.1   amazonlinux   17 k
```

```
sudo yum install gcc glibc glibc-common
```

```
[ec2-user@ip-172-31-86-195 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:35:10 ago on Sun Sep 29 06:16:51 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
<hr/>				
Installing:				
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	32 M
<hr/>				
Installing dependencies:				
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux	92 k
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux	887 k
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	10 M
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux	195 k
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux	27 k
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 k
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libxcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 k

```
sudo yum install gd gd-devel
```

```
[ec2-user@ip-172-31-86-195 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:36:28 ago on Sun Sep 29 06:16:51 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
<hr/>				
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
<hr/>				
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cmake-fs	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-fs	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912 k
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21 k

```
sudo adduser -m nagios
```

```
sudo passwd nagios
```

```
[ec2-user@ip-172-31-86-195 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-86-195 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-86-195 ~]$
```

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

```
[ec2-user@ip-172-31-86-195 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-86-195 ~]$
```

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

```
[ec2-user@ip-172-31-86-195 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-86-195 downloads]$ |
```

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

```
[ec2-user@ip-172-31-86-195 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 07:22:16-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz      100%[=====]  1.97M  6.21MB/s    in 0.3s
2024-09-29 07:22:17 (6.21 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
[ec2-user@ip-172-31-86-195 downloads]$
```

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

```
[ec2-user@ip-172-31-86-195 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-29 07:23:16-- 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  6.58MB/s    in 0.4s
2024-09-29 07:23:16 (6.58 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-86-195 downloads]$ tar zxvf nagios-4.5.5.tar.gz
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
```

**Now we have to first navigate to the nagios-4.5.5 folder in downloads.**

- **commands to enter:**

ls (verify whether nagios-4.5.5 exists). Then go inside nagios 4.5.5 using cd.

```
[ec2-user@ip-172-31-86-195 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-86-195 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$
```

### **we now have to install openssl dev library**

The OpenSSL development library, or openssl-devel contains include files that help develop applications that use cryptographic algorithms and protocols

#### **commands to enter:**

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

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 1:13:15 ago on Sun Sep 29 06:16:51 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                                         22 MB/s | 3.0 MB  00:00

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!
```

**Then finally we can run the commands like usual.**

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

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
```

**make all**

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nebmod.o nebmod.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o common/shared.o ./common/shared.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o workers.o workers.c
```

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.

**sudo make install**

**sudo make install-init**

**sudo make install-config**

**sudo make install-commandmode**

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
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'
```

**Now the next command will take us to nano editor:**

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

```
GNU nano 5.8
/usr/local/nagios/etc/objects/contacts.cfg
#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####

#####
# CONTACTS
#
#####

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

define contact {

    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin        ; Full name of user
    email             nagios@localhost ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####

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

**Change your email**

```
define contact {

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

}

#####

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

**Press ctrl + O and enter**

**Press ctrl + X**

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ |
```

**sudo make install-webconf**

```
[ec2-user@ip-172-31-86-195 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 ***
```

## Adding password for nagios admin

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

```
[ec2-user@ip-172-31-86-195 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-172-31-86-195 nagios-4.5.5]$
```

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$
```

```
cd ~/downloads
```

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

```
[ec2-user@ip-172-31-86-195 nagios-4.5.5]$ cd ~/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
```

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

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

```
.....
[ec2-user@ip-172-31-86-195 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
```

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

```
make
```

```
sudo make install
```

```
[ec2-user@ip-172-31-86-195 nagios-plugins-2.4.11]$ make
sudo make install
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
  cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \  
 \
```

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

```
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-86-195 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

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

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

Checking objects...
    Checked 8 services.
    Checked 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-172-31-86-195 nagios-plugins-2.4.11]$
```

```
sudo service nagios start
```

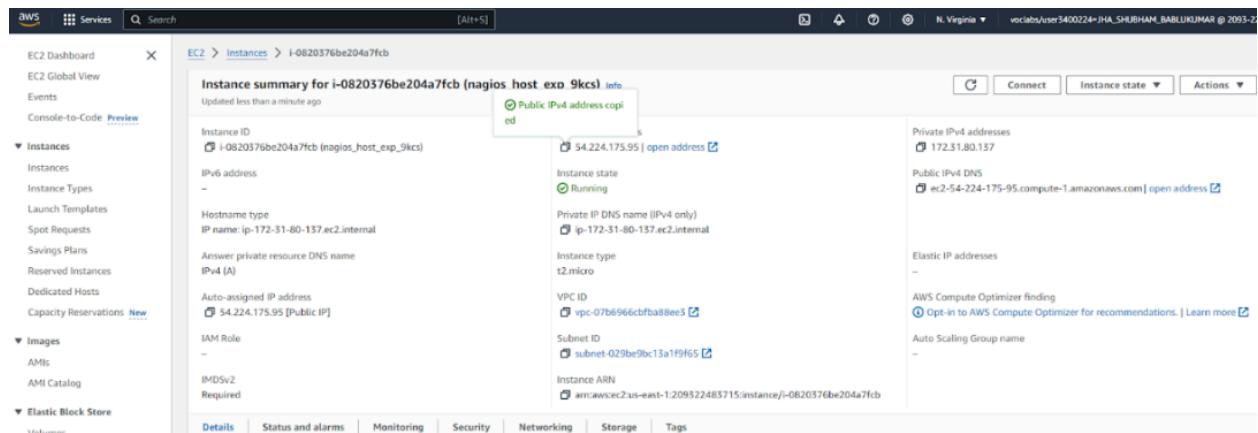
```
[ec2-user@ip-172-31-86-195 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-86-195 nagios-plugins-2.4.11]$
```

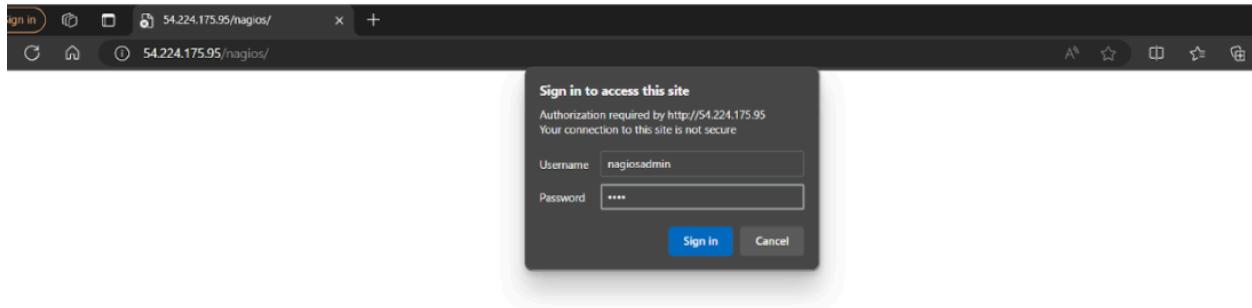
```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-86-195 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-09-29 08:00:47 UTC; 1min 5s ago
    Docs: https://www.nagios.org/documentation
 Process: 66625 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0)
 Process: 66626 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU
 Main PID: 66627 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.8M
     CPU: 90ms
    CGroup: /system.slice/nagios.service
            └─66627 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─66628 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─66629 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─66630 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─66631 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─66632 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successful
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: qh: core query handler registered
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: qh: echo service query handler registered
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: qh: help for the query handler registered
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: wproc: Successfully registered manager as @wproc with quer
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: wproc: Registry request: name=Core Worker 66630;pid=66630
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: wproc: Registry request: name=Core Worker 66631;pid=66631
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: wproc: Registry request: name=Core Worker 66628;pid=66628
Sep 29 08:00:47 ip-172-31-86-195.ec2.internal nagios[66627]: wproc: Registry request: name=Core Worker 66629;pid=66629
Sep 29 08:00:48 ip-172-31-86-195.ec2.internal nagios[66627]: Successfully launched command file worker with pid 66632
lines 1-28/28 (END)
```

Now, go to EC2 instance and click on instance id. Then, click on the copy icon just before the public ip address on public IP.





## Conclusion:

In this experiment, we successfully installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux machine for continuous monitoring. Nagios proves to be an essential tool in DevOps culture by detecting network and server issues in real-time, ensuring infrastructure health. Its scalability, security, and ability to send automated alerts enhance monitoring efficiency. By integrating NRPE, we extended monitoring to remote hosts, allowing proactive troubleshooting. Overall, Nagios is highly customizable with its plugin support and architecture, making it invaluable for maintaining service availability and operational stability.



Name: Kshitij Hundre  
Div: D15C  
Roll No:18

## Adv Devops Exp:10

### Aim :

The aim of this experiment is to set up and configure Nagios for comprehensive monitoring of ports, services, and both Windows and Linux servers. The objective is to ensure real-time monitoring, detect potential issues, and provide timely alerts for system administrators to take preventive or corrective actions, ensuring optimal system performance and uptime.

## Steps

### 1) Launch an instance

Launch an ec2 instance.

Select Ubuntu as the OS to give a meaningful name of the instance.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the instance is named 'exp10client'. Under 'Application and OS Images (Amazon Machine Image)', the 'Ubuntu' option is selected. The 'Summary' section on the right shows one instance being launched with the AMI 'Canonical, Ubuntu, 24.04, a ami-0e86e20dae9224db8'. It also lists the instance type 't2.micro', security group 'launch-wizard-5', and storage '1 volume(s) - 8 GiB'. A note about the free tier is displayed.

Select the same security group as given to the exp9 machine.

**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**

**Browse more AMIs**  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible
--	--------------------

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

**Architecture** 64-bit (x86) | **AMI ID** ami-0e86e20dae9224db8 | **Username** ubuntu | **Verified provider**

**Summary**

Number of instances: 1

Software Images: Canonical, Ubuntu ami-0e86e20dae...

Virtual server type: t2.micro

Firewall (security groups): launch-wizard-1

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

**Free tier**  
750 hours in the Region. The Region is unavailable for this tier. AMI public. If used for a month, it costs \$0.01 million. 100 GB of internet bandwidth per month.

**Cancel**

Make sure to select the same key-pair login used in the exp9 machine.

**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** nagios\_exp\_9 | [Create new key pair](#)

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

**Network** vpc-07b6966cbfba88ee3

**Subnet** 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

**Software Images**: Canonical, Ubuntu ami-0e86e20dae...

**Virtual server type**: t2.micro

**Firewall (security groups)**: launch-wizard-1

**Storage (volumes)**: 1 volume(s)

**Free tier**  
750 hours in the Region. The Region is unavailable for this tier. AMI public. If used for a month, it costs \$0.01 million. 100 GB of internet bandwidth per month.

**Cancel**

click on launch instance.

Now connect with this client machine using the ssh through your terminal(open a new terminal in your local machine and we will need both of the terminals open)

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

Last updated 2 minutes ago [C](#) [Connect](#) [Instance state](#) [Act](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Master	i-0ab175e9c60cc3a23	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-3-82-156-160.com...
node-1	i-08ad30b7114767ca2	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-3-85-110-80.comp...
node-2	i-03c70d364fb762af5	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-54-226-209-38.co...
nagios_host_e...	i-0820376be204a7fcb	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-54-224-175-95.co...
exp10client	i-0994ca5a178801a54	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1b	ec2-54-173-58-143.co...

EC2 > Instances > i-0994ca5a178801a54 > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-0994ca5a178801a54 (exp10client) using any of these options

EC2 Instance Connect Session Manager [SSH client](#) EC2 serial console

Instance ID [i-0994ca5a178801a54 \(exp10client\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is nagios\_exp\_9.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 "nagios_exp_9.pem"`
4. Connect to your instance using its Public DNS:  
`ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com`

**Command copied**

`ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com`

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

[Cancel](#)

Note to change the path of the .pem file.

```

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\Lenovo> ssh -i "C:\Users\Lenovo\Downloads\nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com

The authenticity of host 'ec2-54-173-58-143.compute-1.amazonaws.com (54.173.58.143)' can't be established.
ED25519 key fingerprint is SHA256:IA3XH7f011spK084wDcZFmqRgNn0iJZ7itI2pBMmHP4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-58-143.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sat Sep 28 10:43:28 UTC 2024

System load:  0.01      Processes:          107
Usage of /:   22.8% of 6.71GB  Users logged in:  0
Memory usage: 19%          IPv4 address for enx0: 172.31.82.77

```

## 2) Go to nagios host machine (Host machine)

Perform the following commands

`ps -ef | grep nagios`

```

[ec2-user@ip-172-31-80-137 ~]$ ps -ef | grep nagios
nagios  3152      1  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  3153  3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3154  3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3155  3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3156  3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3160  3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
ec2-user 11528  2972  0 10:44 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-137 ~]$ |

```

`sudo su`

`mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts`

```
[root@ip-172-31-80-137 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-137 ec2-user]# ls
```

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

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

```
[root@ip-172-31-80-137 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

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

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg|
```

Change hostname and alias to linuxserver

Change address to public ip address of client instance (Ubuntu instance) you can get the

ip address by clicking on the instance id on the instances section there you will get the public ipv4 address

Instance summary for i-0994ca5a178801a54 (exp10client) C Connect Instance state ▾ Actions ▾

Updated less than a minute ago

Instance ID  
i-0994ca5a178801a54 (exp10client)

IPv6 address  
-

Hostname type  
IP name: ip-172-31-82-77.ec2.internal

Answer private resource DNS name  
IPv4 (A)

Auto-assigned IP address

Public IPv4 address copied  
54.173.58.143 | open address

Instance state  
Running

Private IP DNS name (IPv4 only)  
ip-172-31-82-77.ec2.internal

Instance type  
t2.micro

VPC ID

Private IPv4 addresses  
172.31.82.77

Public IPv4 DNS  
ec2-54-173-58-143.compute-1.amazonaws.com | open address

Elastic IP addresses  
-

AWS Compute Optimizer finding

## 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
    ; in (or inherited by) the linux-server template
    host_name    linuxserver
    alias        linuxserver
    address      54.173.58.143
}
```

Change hostgroup\_name to linux-servers1

```
# Define an optional hostgroup for Linux machines

define hostgroup {
    hostgroup_name      linux-servers1          ; The name of the hostgroup
    alias               Linux Servers           ; Long name of the group
    members             localhost              ; Comma separated list of host>
}

|
```

Change the occurrences of hostname further in the document from localhost to linuxserver  
example like:

host

host\_name location  
--- ---  
changed to DTNC

6

```
define service {
    use local-service ; Name of service template
    host_name linuxserver
    service_description PING
    check_command check_ping!100.0,20%!500.0,60%
}
```

This is the last one

```
define service {
    use local-service ; Name of service template to >
    host_name linuxserver
    service_description HTTP
    check_command check_http
    notifications_enabled 0
```

now ctrl+O and enter to save and then ctrl+X for exiting.

Open nagios configuration file and add the line shown below  
nano /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

```
##Add this line below the opened nano interface where similar lines are commented.
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

```

GNU nano 5.8
/usr/local/nagios/etc/nagios.cfg

# 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

# Definitions for monitoring the local (Linux) host
:cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

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

# Definitions for monitoring a router/switch
:cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
:cfg_file=/usr/local/nagios/etc/objects/printer.cfg

# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

:cfg_dir=/usr/local/nagios/etc/servers
:cfg_dir=/usr/local/nagios/etc/printers
:cfg_dir=/usr/local/nagios/etc/switches
:cfg_dir=/usr/local/nagios/etc/routers
:cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts up. The cache need object definitions from

```

ctrl+o and enter for saving and ctrl+x to exit nano editor.

## Verify configuration files

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

```
[root@ip-172-31-80-137 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 0 service dependencies
Checked 0 host dependencies
Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# |
```

Restart nagios service.

```
service nagios restart
```

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-80-137 ec2-user]# |
```

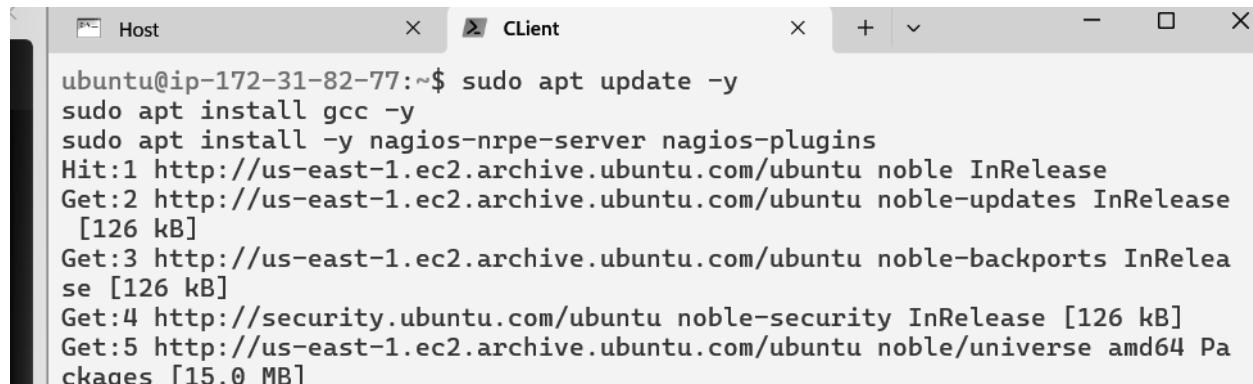
### 3) Go to client machine (ubuntu machine)

Perform the following commands

```
sudo apt update -y
```

```
sudo apt install gcc -y
```

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



A screenshot of a terminal window titled "Client". The window contains the following text:

```
ubuntu@ip-172-31-82-77:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
[126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Pa
ckages [15.0 MB]
```

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

Restarting services...

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

0 containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #1: sshd[990,1101]
ubuntu @ user manager service: systemd[996]

0 VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Open the nrpe.cfg file in nano editor  
sudo nano /etc/nagios/nrpe.cfg

Under allowed\_hosts, add the nagios host ip address (public)

```
# You can either supply a username or a UID.  
#  
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
nrpe_user=nagios  
  
# NRPE GROUP  
# This determines the effective group that the NRPE daemon should run as.  
# You can either supply a group name or a GID.  
#  
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
nrpe_group=nagios  
  
# ALLOWED HOST ADDRESSES  
# This is an optional comma-delimited list of IP address or hostnames  
# that are allowed to talk to the NRPE daemon. Network addresses with a bit  
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not curr  
# 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=127.0.0.1,54.224.175.95  
  
# COMMAND ARGUMENT PROCESSING  
# This option determines whether or not the NRPE daemon will allow clients  
again save and exit the nano editor.
```

## 4) Go to nagios dashboard and click on hosts

The screenshot shows the Nagios Core dashboard at the URL <https://54.224.175.95/nagios/>. The top navigation bar indicates "Not secure" and the address. The main header features the "Nagios® Core™" logo with a green checkmark and the text "Daemon running with PID 13935". On the left, a sidebar menu includes sections for General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Grid, Service Groups, Grid), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime Info, Process Info, Performance Info, Scheduling Queue, Configuration). The central content area has a "Get Started" section with links to monitoring infrastructure, changing look, extending Nagios, and getting certified. It also features "Latest News" and "Don't Miss..." sections. A "Quick Links" sidebar on the right provides links to Nagios Library, Labs, Exchange, Support, and official websites. At the bottom, there is copyright information and a "Nagios" logo.

Click on hosts

This screenshot shows the "Current Status" section of the Nagios Core dashboard. On the left, a vertical sidebar lists navigation links: "Tactical Overview" (which is currently selected and highlighted in blue), "Map", "Hosts", "Services", and "Host Groups". The main content area is currently empty, indicating no hosts are currently selected or visible.

## 5) Click on linux server

**Nagios®**

**Current Network Status**  
Last Updated: Sat Sep 28 11:33:24 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**General**  
Home Documentation

**Current Status**  
Tactical Overview Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**  
Availability Trends Alerts History Summary Histogram Notifications Event Log

**Host Status Totals**  
Up Down Unreachable Pending  
2 0 0 0  
All Problems All Types  
0 2

**Service Status Totals**  
Ok Warning Unknown Critical Pending  
12 1 0 3 0  
All Problems All Types  
4 16

**Host Status Details For All Host Groups**

Limit Results: 100 Host Status Last Check Duration Status Information

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 11:29:10	0d 0h 8m 36s	PING OK - Packet loss = 0%, RTA = 1.18 ms
localhost	UP	09-28-2024 11:32:18	0d 3h 53m 7s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

**Nagios®**

**Host Information**  
Last Updated: Sat Sep 28 11:33:39 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**General**  
Home Documentation

**Current Status**  
Tactical Overview Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**  
Availability Trends Alerts History Summary Histogram Notifications Event Log

**Host**  
**linuxserver**  
(linuxserver)

**Member of**  
No hostgroups

54.173.58.143

**Host State Information**

Host Status:	Status Information:
UP	(for 0d 0h 8m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.18 ms
Performance Data:	rta=1.184000ms,3000.000000,5000.000000,0.000000,p=0%;80;100,0
Current Attempt:	1/10 (HARD state)
Last Check Time:	09-28-2024 11:29:10
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.066 seconds
Next Scheduled Active Check:	09-28-2024 11:34:10
Last State Change:	09-28-2024 11:24:48
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	09-28-2024 11:33:37 ( 0d 0h 0m 2s ago)

**Host Commands**

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

**Host Comments**

Add a new comment

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it.							

## 6) Click on nagios services

[Documentation](#)

The screenshot shows the Nagios 'Current Status' interface. On the left, a sidebar lists navigation options: General, Home, Documentation, Current Status (selected), Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid, and Service Groups. The main content area displays 'Current Network Status' with a last update timestamp of 09-28-2024 11:33:58 UTC 2024. It shows host status totals (Up: 2, Down: 0, Unreachable: 0, Pending: 0) and service status totals (Ok: 12, Warning: 1, Unknown: 0, Critical: 3, Pending: 0). Below these are two tables: 'Host Status Details For All Hosts' and 'Service Status Details For All Hosts'. The 'Service Status Details For All Hosts' table lists services for 'linuxserver' and 'localhost', including CPU load, memory usage, disk space, and swap usage. For 'localhost', there are entries for 'HTTP' (CRITICAL), 'PING' (OK), 'Root Partition' (OK), 'SSH' (OK), 'Swap Usage' (CRITICAL), and 'Total Processes' (OK). The 'localhost' row for 'HTTP' has a note: 'connect to address 54.173.58.143 and port 80: Connection refused'.

## Conclusion:

In this experiment, we successfully set up Nagios to monitor a remote Linux server (Ubuntu instance) from a Nagios host machine, both operating within an EC2 environment. The process involved careful configuration of both the Nagios host (referred to as the "exp9 machine") and the remote client machine, allowing for effective real-time monitoring of server performance. By launching an Ubuntu EC2 instance as the client machine, we ensured that consistent security group policies and key-pairs were maintained, enabling seamless SSH access to both machines for configuration purposes.

The configuration on the Nagios host involved creating a new directory for the remote client's monitoring configuration, where we modified the **localhost.cfg** file to reflect the IP address of the remote client, adjusted the hostgroup, and ensured proper integration with the Nagios setup.

We also verified these changes through a Nagios configuration check before restarting the service, ensuring that the Nagios dashboard was ready to monitor the remote machine.

On the client side, we installed the **nagios-nrpe-server** and **nagios-plugins**, which allowed communication between the Nagios host and client machine. Correctly configuring the **nrpe.cfg** file, particularly the **allowed\_hosts** directive, was crucial in establishing connectivity. Once the configuration was complete and the Nagios service was restarted, we successfully confirmed that the remote Linux server was being monitored on the Nagios dashboard, achieving the experiment's objective.

This experiment not only highlighted the steps for setting up monitoring between a Nagios host and a remote Linux client but also demonstrated key troubleshooting techniques. By resolving common issues such as restarting the Apache server on the Nagios host, ensuring proper directory structure, and configuring the NRPE correctly on the client machine, we gained valuable insights into server monitoring and network management using Nagios. The hands-on experience provided a deeper understanding of how Nagios can be effectively used for real-world infrastructure monitoring.

Name: Kshitij Hundre  
Div: D15C  
Roll No:18

## Adv DevOps Exp-11

### Aim:

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

### Theory:

#### AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

#### Lambda Workflow:

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring reserved concurrency to manage traffic.
- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

#### AWS Lambda Functions:

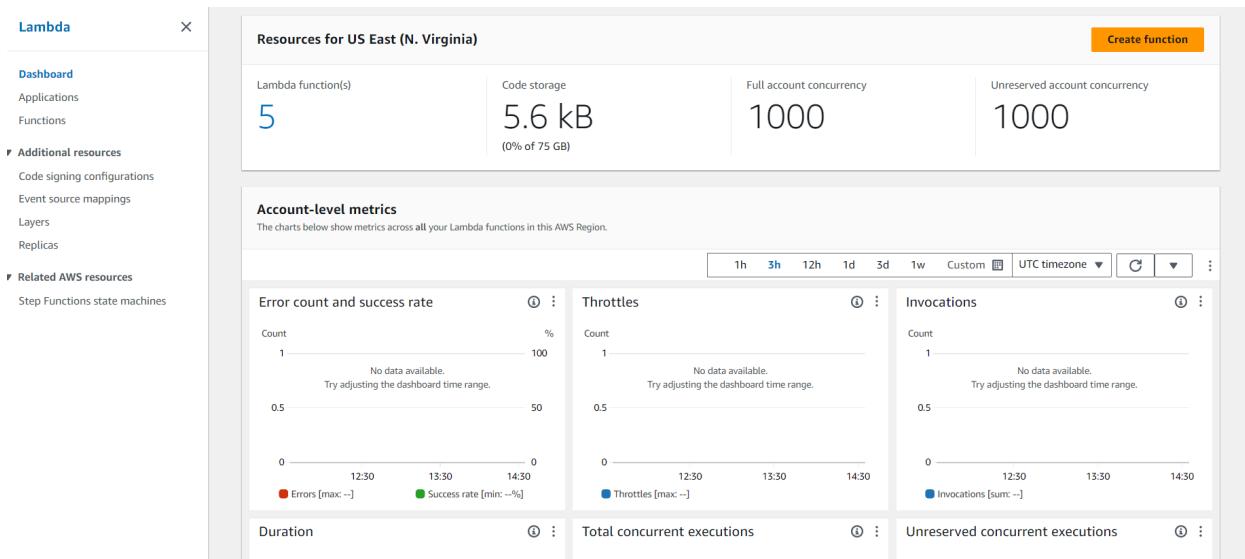
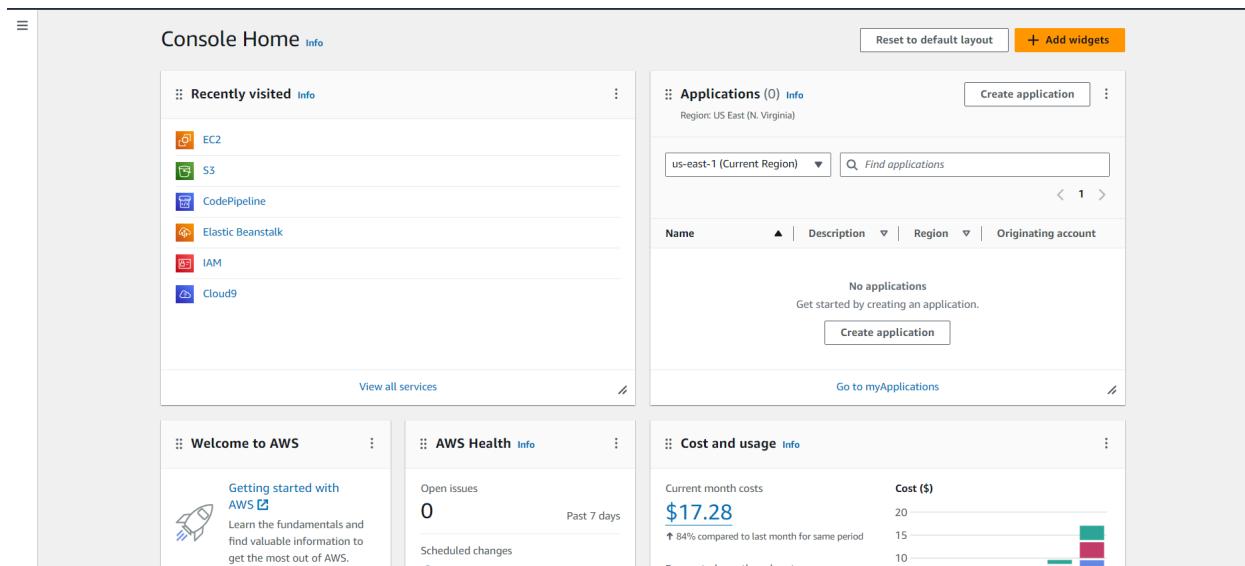
- **Python:** Great for quick development with its rich standard library and support for lightweight tasks.
- **Java:** Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

Prerequisites: AWS Personal/Academy Account

Prerequisites: AWS Personal/Academy Account

## Steps To create the lambda function:

**Step 1:** Login to your AWS Personal/Academy Account. Open Lambda and click on create function button.



**Step 2:** Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

AWS Services Search [Alt+S]

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

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

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

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

▼ Change default execution role

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

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

Lambda will create an execution role named KCS\_Lambda-role-kssqesm9, with permission to upload logs to Amazon CloudWatch Logs.

▶ Advanced settings

Cancel Create function

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

[Lambda](#) > [Functions](#) > KCS\_Lambda

## KCS\_Lambda

Throttle    [Copy ARN](#)    Actions ▾

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

[Diagram](#)    [Template](#)

 KCS\_Lambda  
 Layers (0)

+ Add trigger    + Add destination

Description  
-

Last modified  
3 seconds ago

Function ARN  
[arn:aws:lambda:us-east-1:235494807211:function:KCS\\_Lambda](#)

Function URL [Info](#)  
-

Export to Application Composer    Download ▾

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

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

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

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

Go to Anything (Ctrl-P)

Environment

KCS\_Lambda - lambda\_function Environment Var +

```

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

```

Upload from ▾

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

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

General configuration	
Triggers	<a href="#">Edit</a>
Permissions	
Destinations	
Function URL	
Environment variables	
Tags	
VPC	
RDS databases	
Monitoring and operations tools	

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

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout 0 min 3 sec	SnapStart <a href="#">Info</a> None	

Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 2 sec since that is sufficient for now.

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

**Description - optional**  
The supreme leader(KCS) wants to change the basic settings

**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

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

View the [KCS\\_Lambda-role-9nzyyxbk](#) role [on the IAM console](#).

☰ Successfully updated the function KCS\_Lambda. X

[Lambda](#) > [Functions](#) > KCS\_Lambda

**KCS\_Lambda**

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

**Function overview** [Info](#) [Export to Application Composer](#) [Download ▾](#)

**Diagram** [Template](#)

KCS\_Lambda  
 Layers (0)

[+ Add trigger](#) [+ Add destination](#)

**Description**  
The supreme leader(KCS) wants to change the basic settings

**Last modified**  
2 seconds ago

**Function ARN**  
 arn:aws:lambda:us-east-1:235494807211:function:KCS\_Lambda

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

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

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  
KCS\_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

The test event KCS\_Event was successfully saved. X

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  
KCS\_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

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

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

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

Go to Anything (Ctrl-P)

Environment KCS\_Lambda / lambda\_function.py

lambda\_function

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string = "Hey there. I am KCS!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
```

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment KCS\_Lambda / lambda\_function.py

lambda\_function x Environment Var x Execution result x

Test Event Name KCS\_Event

Status: Succeeded | Max memory used: 32 MB | Time: 2.07 ms

Response

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

Function Logs

```
START RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f Version: $LATEST
END RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f
REPORT RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Mem
```

Request ID 9b8874c5-da6e-4026-9098-134c4fee787f

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

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

Go to Anything (Ctrl-P)

Environment KCS\_Lambda / lambda\_function.py

lambda\_function x Environment Var x Execution results x

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string = "Hey there. I am KCS!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10
```

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

The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. On the right, there's a 'Upload from' button. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, and a 'Test' dropdown which is currently selected. To the right of the dropdown is a 'Deploy' button and a gear icon. The main area has sections for 'Execution results', 'Test Event Name' (set to 'KCS\_Event'), 'Response' (containing JSON output), 'Function Logs' (listing START, END, and REPORT events), and 'Request ID' (showing the value '8cc12d43-7137-4c05-9ecf-315440b7226d'). A sidebar on the left is titled 'Environment' and lists 'lambda\_function' and 'lambda\_function.py'.

You can see the desired output.

**Conclusion:** In this experiment, we successfully developed an AWS Lambda function, covering the key steps involved. Starting with the Python-based setup, we configured the function's fundamental settings, including setting the timeout to 1 second. We proceeded to create a test event, deployed the function, and verified its output. Additionally, we made updates to the Lambda function's code and redeployed it, observing the real-time changes. This hands-on experience highlighted AWS Lambda's efficiency and adaptability, enabling rapid serverless application development while AWS handles infrastructure and scaling effortlessly.

## Adv DevOps Exp-12

**Aim:**

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

**Theory:****AWS Lambda and S3 Integration:**

AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

**Workflow:****1. Create an S3 Bucket:**

- First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

**2. Create the Lambda Function:**

- Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

**3. Set Up Permissions:**

- Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

**4. Configure S3 Trigger:**

- Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

**5. Test the Setup:**

- Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.  
Prerequisites: AWS Personal Account

**Prerequisites:** AWS Personal Account

## Steps To create the lambda function:

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

The screenshot shows the Amazon S3 service page. On the left, there's a large 'Amazon S3' logo with the tagline 'Store and retrieve any amount of data from anywhere'. Below the logo, a brief description states: 'Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.' On the right, a prominent orange 'Create a bucket' button is visible. Below it, a 'Pricing' section provides information about costs and links to a calculator. In the center, a 'How it works' section includes a thumbnail image of a presentation slide titled 'Introduction to Amazon S3' featuring the AWS logo.

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

The screenshot shows the 'Create bucket' configuration page. It starts with a summary: 'Buckets are containers for data stored in S3.' Below this, the 'General configuration' section is shown. Under 'AWS Region', 'US East (N. Virginia) us-east-1' is selected. The 'Bucket type' section has 'General purpose' selected (indicated by a blue border). A detailed description of general purpose buckets follows. The 'Bucket name' field contains 'wearekcs'. A note below says the name must be unique and follow naming rules, with a link to 'See rules for bucket naming'. The 'Copy settings from existing bucket - optional' section is present but empty. At the bottom, a note says '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.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership  
Bucket owner enforced

**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.

**A** Turning off block all public access might result in this bucket and the objects within becoming

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

Amazon S3 > Buckets

**Account snapshot - updated every 24 hours** All AWS Regions  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

General purpose buckets | Directory buckets

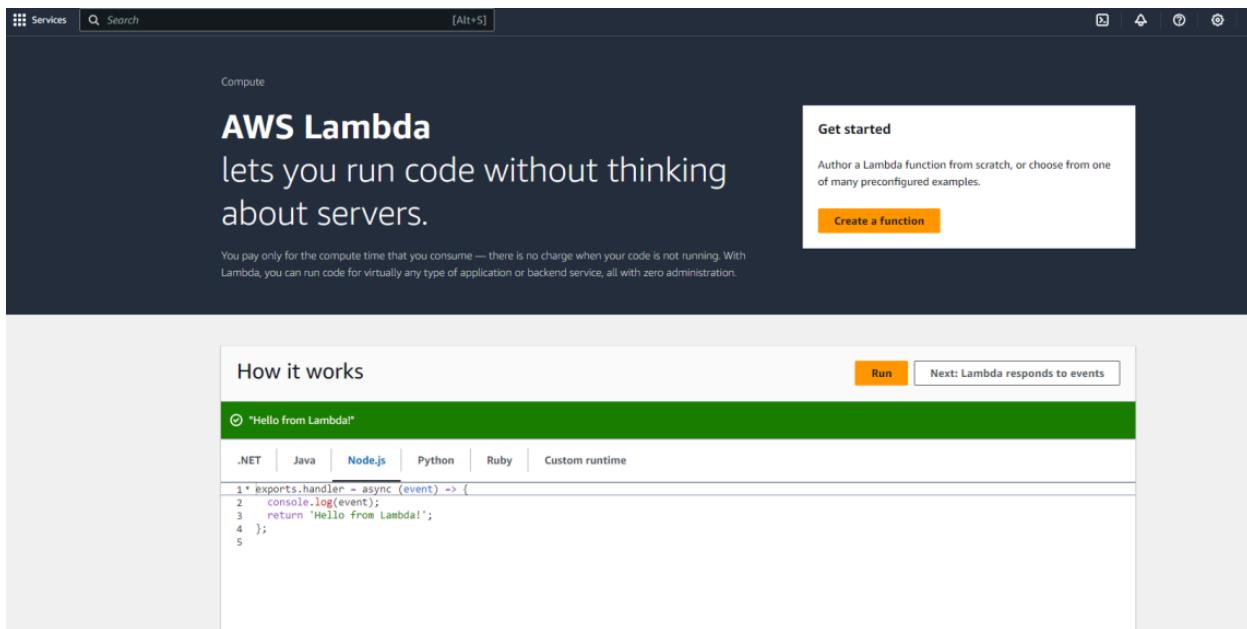
General purpose buckets (1) [Info](#) All AWS Regions  
Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
wearekes	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 1, 2024, 13:40:40 (UTC+05:30)

[Create bucket](#)

### Step 3: Open lambda console and click on create function button



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

The screenshot shows the 'Create function' wizard. The top navigation bar includes 'Lambda > Functions > Create function'. The main section is titled 'Create function' with an 'Info' link. It says: 'Choose one of the following options to create your function.' Three radio buttons are shown: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' step is currently active. It includes fields for 'Function name' (set to 'KCS\_Exp12'), 'Runtime' (set to 'Python 3.12'), 'Architecture' (set to 'x86\_64'), and 'Permissions' (with a note about default execution role). Other sections like 'Advanced settings' and 'Change default execution role' are also visible.

Name: Kshitij Hundre

Div:D15C

Roll No:18

The screenshot shows the AWS Lambda Function Overview page for a function named 'KCS\_Exp12'. At the top, a green banner indicates 'Successfully created the function KCS\_Exp12. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' Below the banner, the function name 'KCS\_Exp12' is displayed. The 'Function overview' tab is selected, showing a diagram where the function is represented by a box labeled 'KCS\_Exp12' with a lambda icon, and below it, a 'Layers' section with '(0)' listed. There are buttons for '+ Add trigger' and '+ Add destination'. On the right side, there are fields for 'Description' (empty), 'Last modified' (3 seconds ago), 'Function ARN' (arn:aws:lambda:us-east-1:235494807211:function:KCS\_Exp12), and 'Function URL' (Info). Below the overview, tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions' are visible. Under the 'Code' tab, the 'Code source' section is active, showing a code editor with Python code for a lambda function named 'lambda\_function'. The code defines a single handler 'lambda\_handler' that returns a JSON response with status code 200 and body 'Hello from Lambda!'. The code editor has tabs for 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (selected), and 'Deploy'.

The screenshot shows the AWS Lambda Code Source editor for the 'KCS\_Exp12' function. The interface includes tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected, showing a code editor with Python code for a lambda function named 'lambda\_function'. The code defines a single handler 'lambda\_handler' that returns a JSON response with status code 200 and body 'Hello from Lambda!'. The code editor has tabs for 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (selected), and 'Deploy'.

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

The screenshot shows the AWS Lambda Configuration page for the 'KCS\_Exp12' function. The 'Configuration' tab is selected. On the left, a sidebar lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', 'Function URL', and 'Environment variables'. The 'General configuration' section is expanded, showing fields for 'Description' (KCS exp12), 'Memory' (128 MB), 'Ephemeral storage' (512 MB), 'Timeout' (0 min 2 sec), 'SnapStart' (Info, None), and an 'Edit' button. The main area displays the configuration details.

Change any setting of your choice. Here I have set a timeout of 2 secs.Then save changes

Lambda > Functions > KCS\_Exp12 > Edit basic settings

## Edit basic settings

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

Description - optional  
KCS exp12

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 2 sec

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

Existing role  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
service-role/KCS\_Exp12-role-0q6h1t4r  
[View the KCS\\_Exp12-role-0q6h1t4r role](#) on the IAM console.

[Save](#) [Test](#)

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

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

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

Test event [Info](#)  
 Create new event  Edit saved event

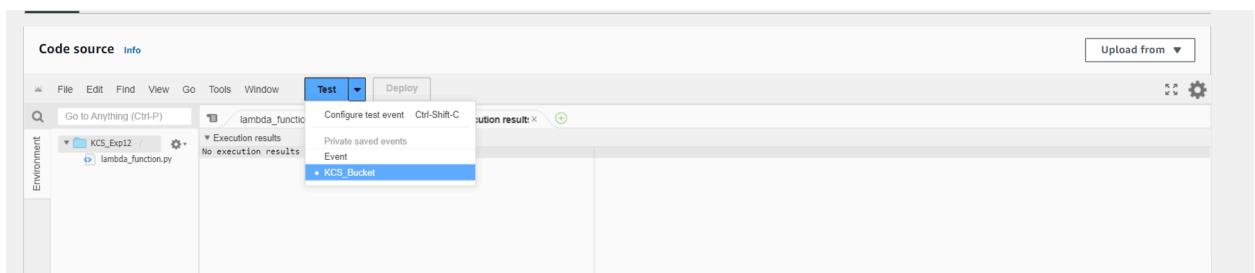
Event name  
KCS\_Bucket  
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

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](#)

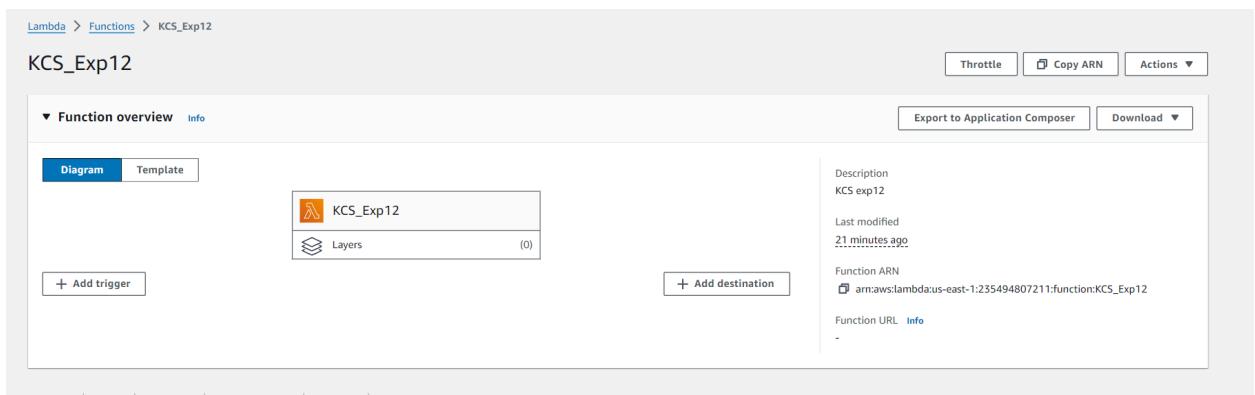
Template - optional  
s3-put

Event JSON [Format JSON](#)

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



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



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

Lambda > Add triggers

## Add trigger

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

**S3** aws asynchronous storage

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

[X](#) [C](#)

Bucket region: us-east-1

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

**All object create events** [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**

KCS\_Ex12

The trigger wearekcs was successfully added to function KCS\_Ex12. The function is now receiving events from the trigger. [X](#)

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

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

**KCS\_Ex12**

**Description** KCS\_Ex12

**Last modified** 26 minutes ago

**Function ARN** arn:aws:lambda:us-east-1:235494807211:function:KCS\_Ex12

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

**S3** [+ Add destination](#)

[+ Add trigger](#)

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

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

The screenshot shows the AWS Lambda Configuration page. The left sidebar contains a list of configuration options: General configuration, Triggers (which is selected), Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, File systems, and State machines. The main panel is titled 'Triggers (1) Info' and shows one trigger named 'S3: wearekcs' with the ARN 'arnawsS3:wearekcs'. There are buttons for 'C' (Create), 'Fix errors', 'Edit', 'Delete', and 'Add trigger'.

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

The screenshot shows the AWS Lambda Code source editor. The code in `lambda_function.py` is:

```

import json
def lambda_handler(event, context):
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = event['Records'][0]['s3']['object']['key']
    print(f'An image has been added to the bucket {bucket_name} : {object_key}')
    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }

```

The status bar at the bottom of the editor window displays a green message: "Successfully updated the function KCS\_Exp12."

Below the editor, there is another screenshot of the AWS Lambda Configuration page showing the execution results of a test event. The test event name is 'KCS\_Bucket' and it shows a successful execution with a status of 'Succeeded', a maximum memory used of 32 MB, and a duration of 2.00 ms. The log output includes the printed message and the function logs.

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

Amazon S3 > Buckets > wearekcs > Upload

## Upload Info

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

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

Files and folders (1 Total, 957.0 KB)		<a href="#">Remove</a>	<a href="#">Add files</a>	<a href="#">Add folder</a>
All files and folders in this table will be uploaded.				
<input type="text"/> <a href="#">Find by name</a>		< 1 >		
<input type="checkbox"/> Name	▼	Folder		
<input type="checkbox"/> F_i0UxsXgAAxB2s.jpg	-			

### Destination Info

Destination  
[s3://wearekcs](#)

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

▶ **Permissions**  
Grant public access and access to other AWS accounts.

▶ **Properties**  
Specify storage class, encryption settings, tags, and more.

[Cancel](#) [Upload](#)

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

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