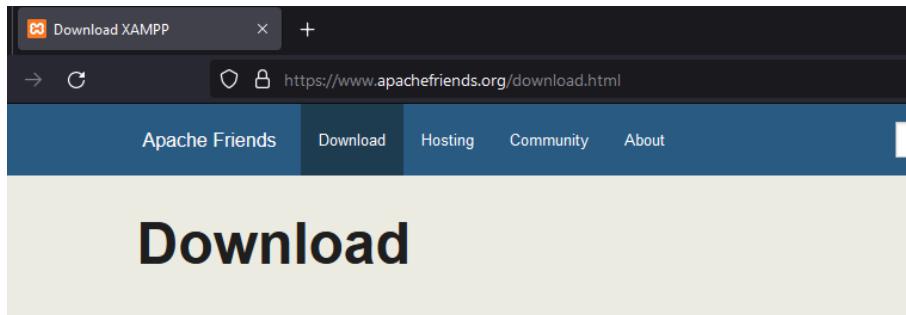


Experiment No. 1(A)

Aim: To develop a website and host it on local machine or virtual machine and Hosting a static website using Amazon S3 Bucket

1. To develop a website and host it on your local machine on a VM using XAMPP

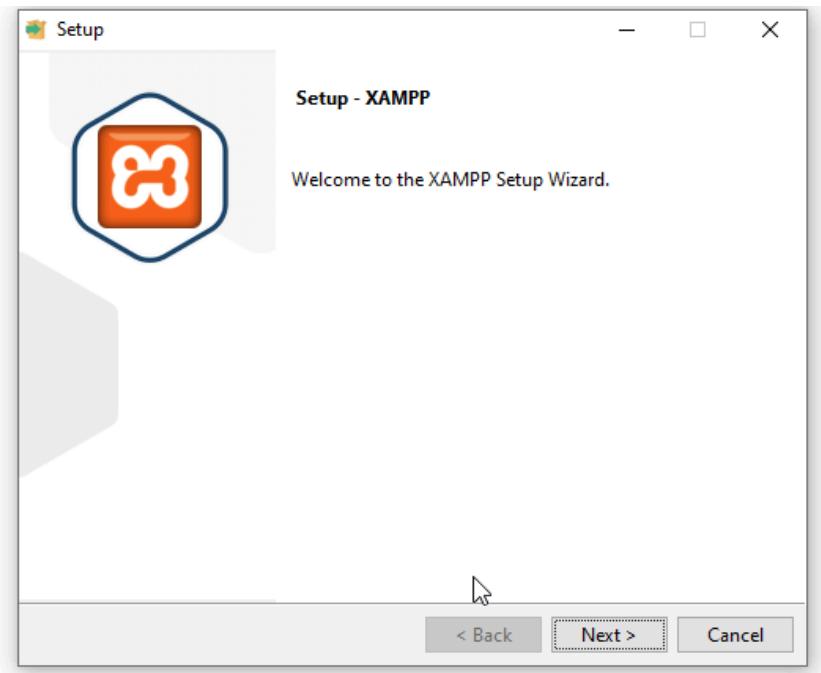
- 1) Goto official website of XAMPP and download the software as per your OS



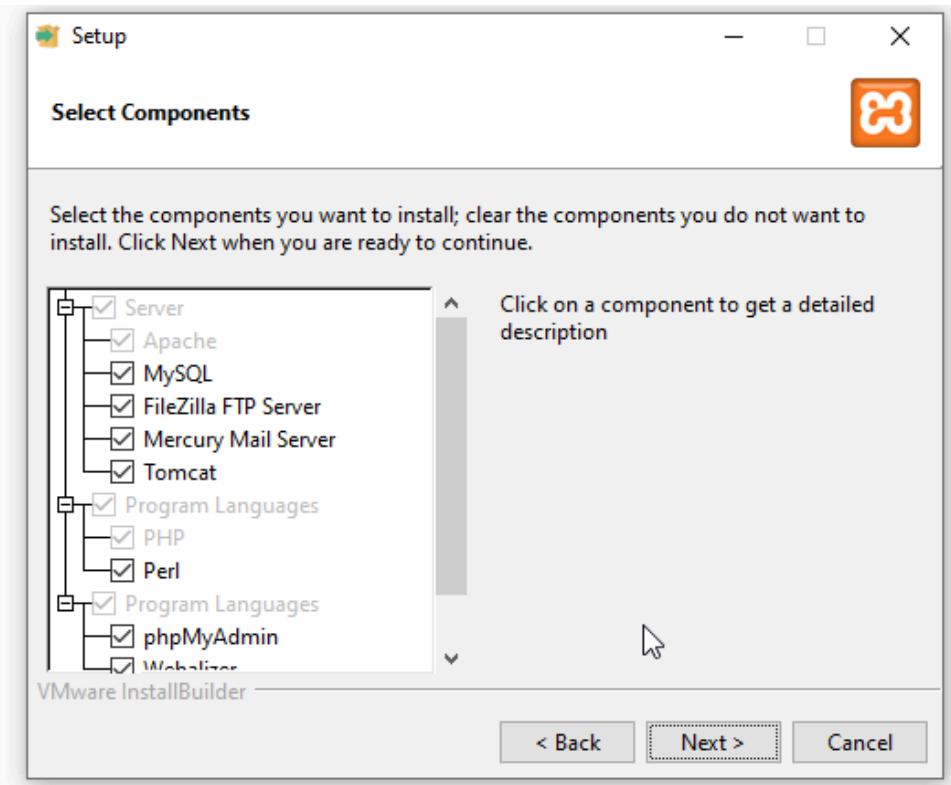
XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using [InstallBuilder](#).

Version	Checksum	Size
8.0.30 / PHP 8.0.30	What's Included? md5 sha1	Download (64 bit) 144 Mb
8.1.25 / PHP 8.1.25	What's Included? md5 sha1	Download (64 bit) 148 Mb
8.2.12 / PHP 8.2.12	What's Included? md5 sha1	Download (64 bit) 149 Mb

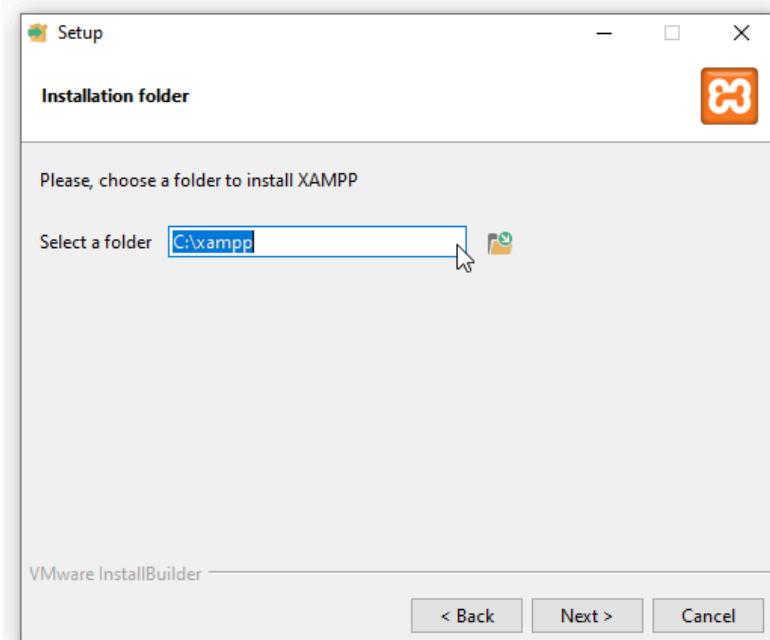
- 2) After Downloading, open it for installation, Click on next



3) Select components to install



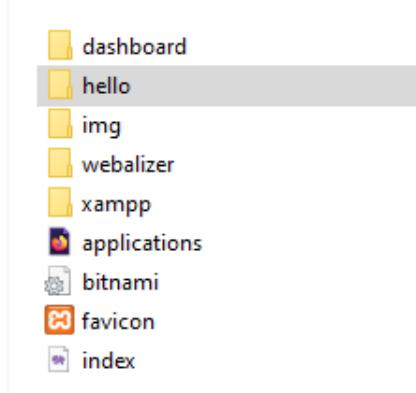
4) Select installation location, and confirm it



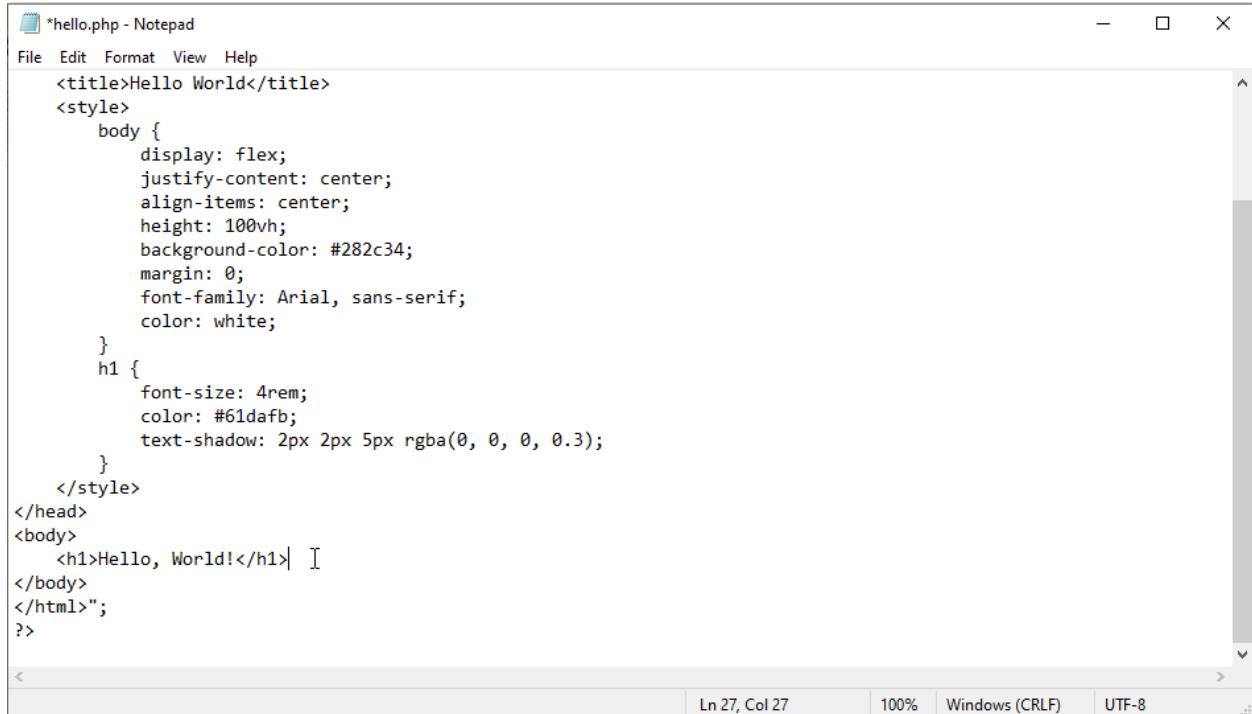
5) After installation, goto folder where XAMPP is installed, go to htdocs

anonymous	8/11/2024 2:29 AM
apache	8/11/2024 2:29 AM
cgi-bin	8/11/2024 2:33 AM
contrib	8/11/2024 2:29 AM
FileZillaFTP	8/11/2024 2:33 AM
htdocs	8/11/2024 2:29 AM
img	8/11/2024 2:29 AM
install	8/11/2024 2:33 AM
licenses	8/11/2024 2:29 AM
locale	8/11/2024 2:29 AM

6) Create new folder named anything we want our URI to be.



7) Create a file index.php and paste the following code.

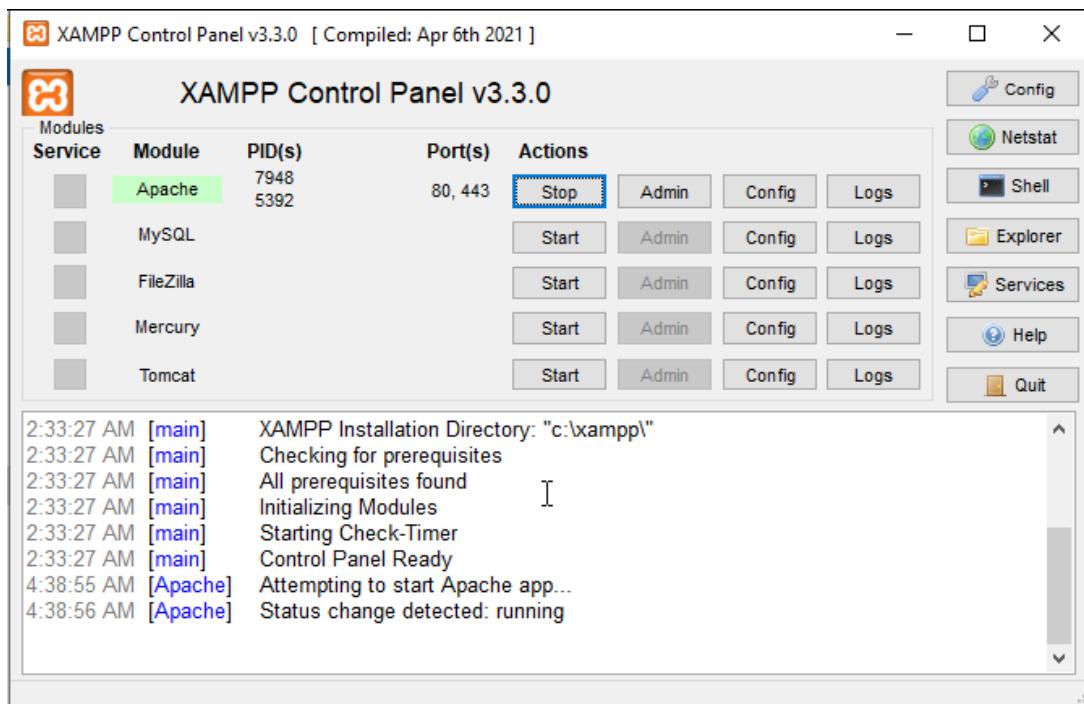


```
<?php
    $name = "Alok";
    $greet = "Hello, " . $name . "!";
    echo $greet;
?>
```

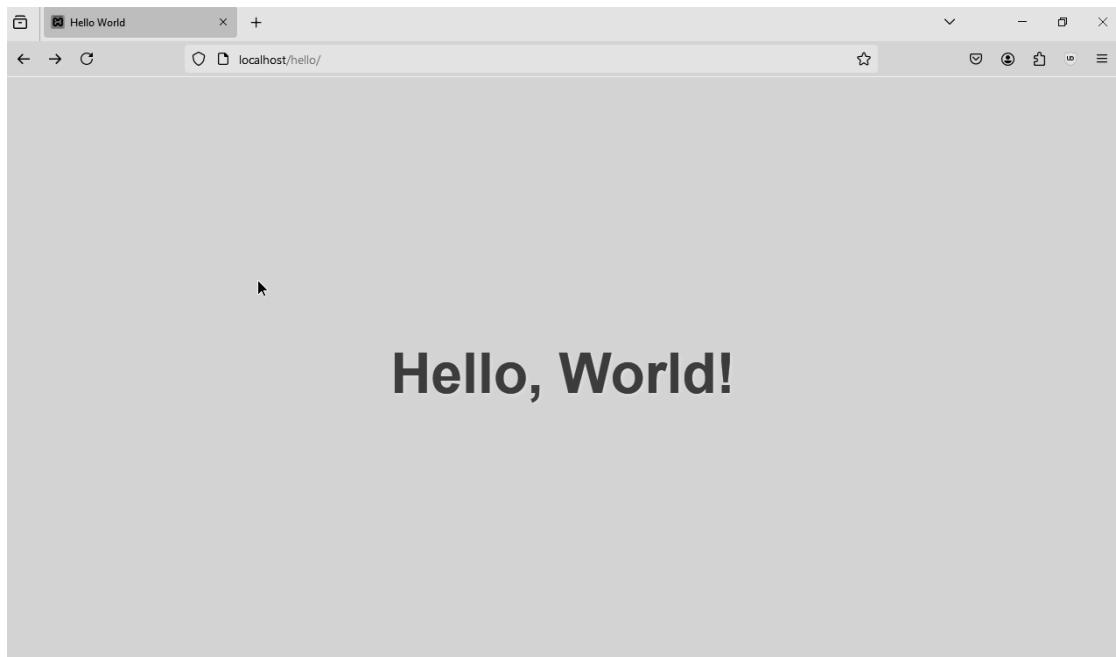
```
*hello.php - Notepad
File Edit Format View Help
<title>Hello World</title>
<style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        background-color: #282c34;
        margin: 0;
        font-family: Arial, sans-serif;
        color: white;
    }
    h1 {
        font-size: 4rem;
        color: #61dafb;
        text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
    }
</style>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>";
?>
```

Ln 27, Col 27 100% Windows (CRLF) UTF-8

8) Now go to XAMPP control panel and start Apache server

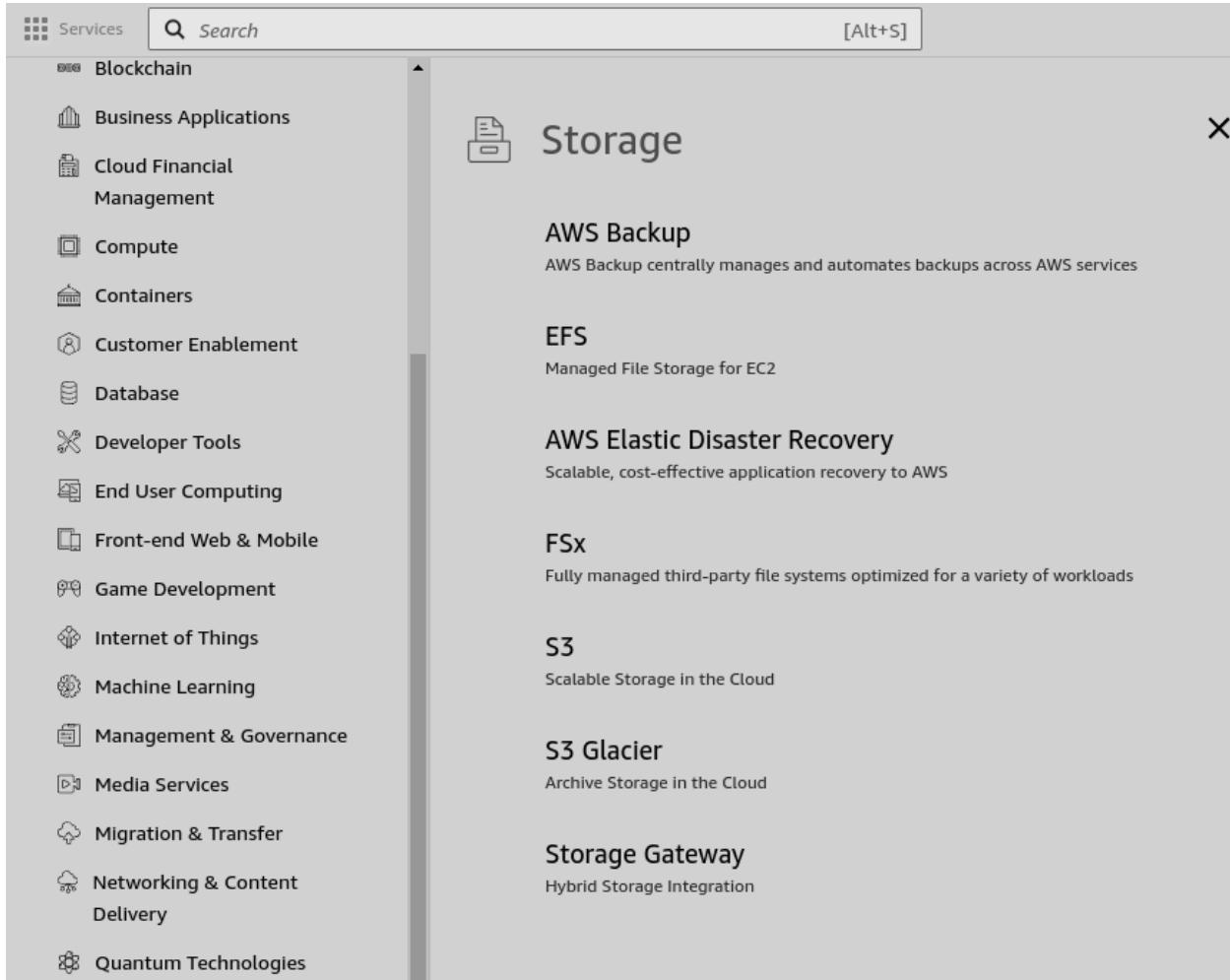


- 9) Now open browser and enter http://localhost/folder_name. This will open webpage locally on our machine



2. Hosting a static website on Amazon S3

- 1) Go to services then under Storage, select S3



The screenshot shows the AWS Management Console with the 'Storage' section selected. The left sidebar lists various AWS services, and the main content area displays the Storage services: AWS Backup, EFS, AWS Elastic Disaster Recovery, FSx, S3, S3 Glacier, and Storage Gateway. The S3 service is highlighted with a yellow box.

- 2) Create a new bucket



The screenshot shows the AWS S3 buckets list. It displays a single bucket entry: 'General purpose buckets (1)'. Below the list are buttons for 'Copy ARN', 'Empty', 'Delete', and a prominent yellow 'Create bucket' button.

- 3) Enter Name of the bucket

General configuration

AWS Region

Asia Pacific (Mumbai) ap-south-1

Bucket name [Info](#)

host-web-alok

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

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

- 4) Disable “Block all public access” to make the contents of bucket publicly accessible

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.

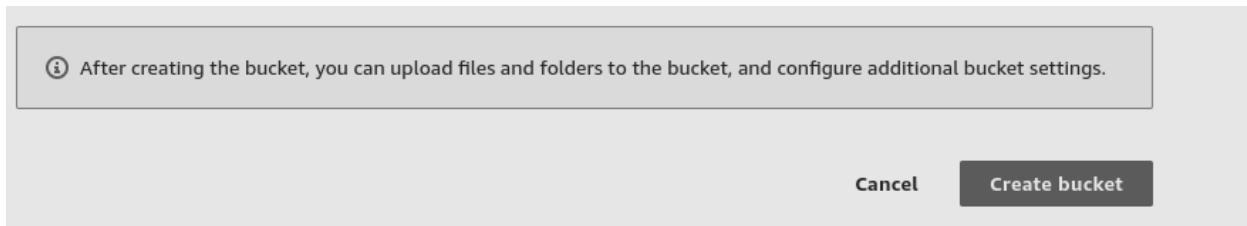
Block public access to buckets and objects granted through new public bucket or access point policies

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.

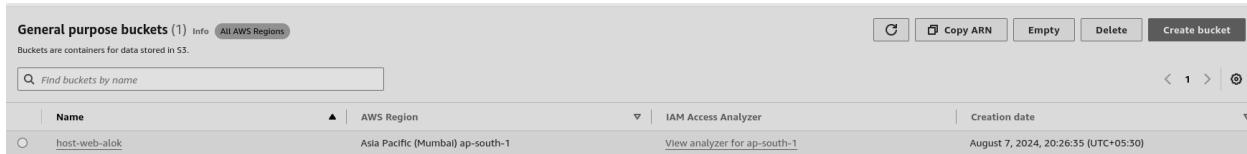
Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

- 5) Leaving the remaining configuration to it's default, click on Create Bucket



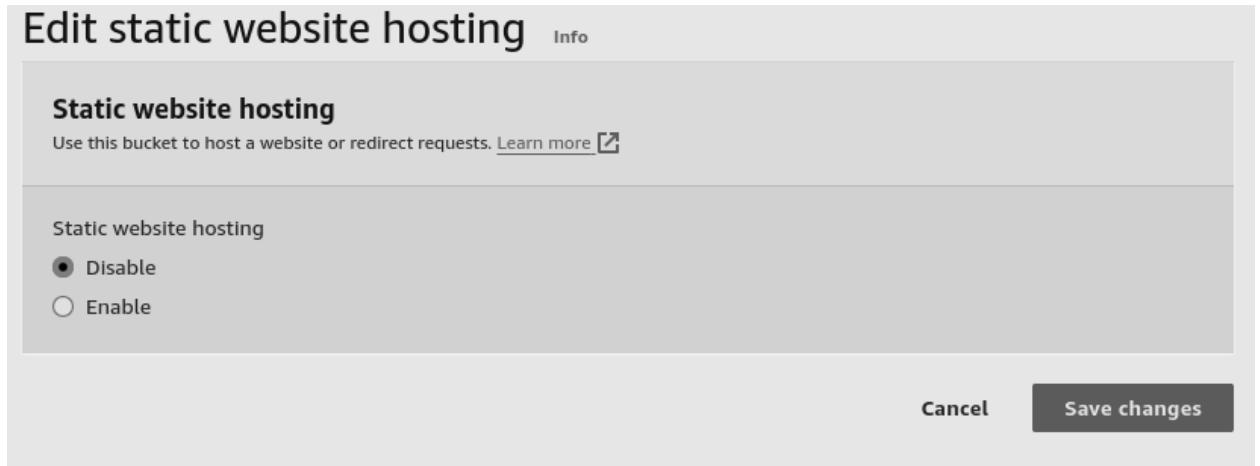
6) A new bucket will be created



7) Now to host a static web page, open the bucket we created, goto Properties, edit "Static Web hosting settings"



8) Enable Static web Hosting



9) After enabling, we need to enter the names of files two files, one for main file that would be displayed when page loads and the other when some error occurs. Name those files and save it

Hosting type

Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

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

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

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

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

Redirection rules - optional

- 10) After naming, go to objects and upload the files you mentioned in the static web hosting settings page

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**.

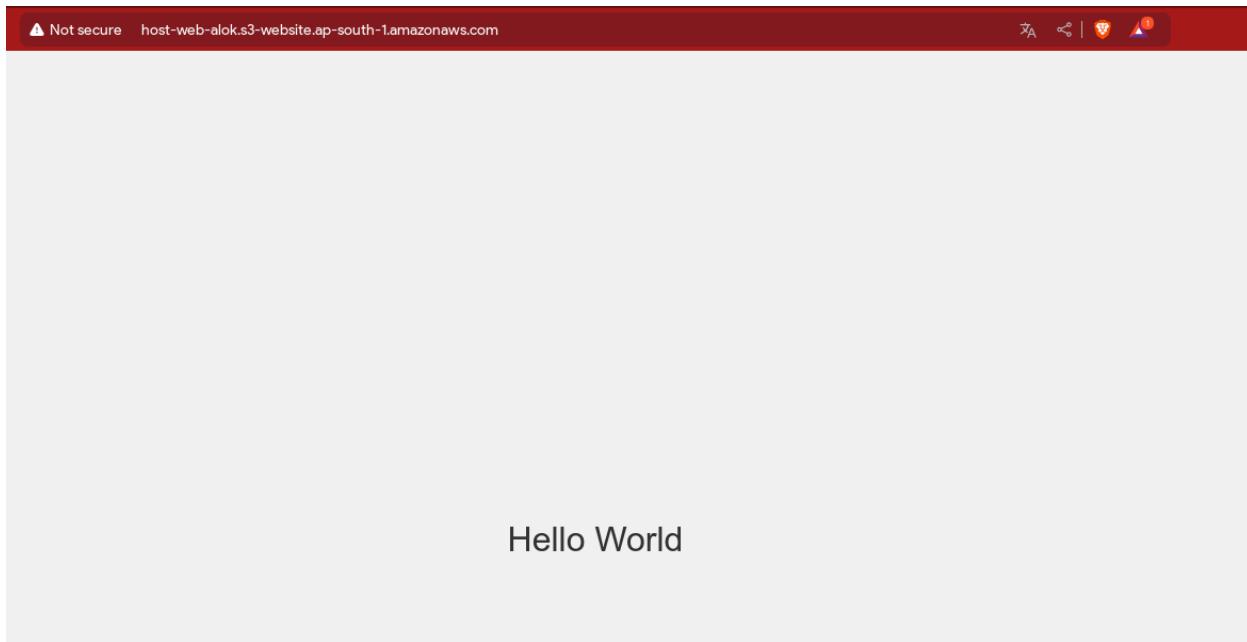
Files and folders (2 Total, 1.3 KB)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input style="width: 100%; height: 30px; border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;" type="text"/> Find by name		<< 1 >		
<input type="checkbox"/>	Name	▼	Folder	
<input type="checkbox"/>	index.html	-		
<input type="checkbox"/>	error.html	-		

- 11) Now to make sure our page is accessible to public, we need to change bucket policy. To change it, go to permissions and edit Bucket policy. Change resource arn as per your bucket name

Policy

```
1
2  {
3      "Version": "2012-10-17",
4      "Statement": [
5          {
6              "Sid": "PublicReadGetObject",
7              "Effect": "Allow",
8              "Principal": {
9                  "AWS": "*"
10             },
11             "Action": "s3:GetObject",
12             "Resource": "arn:aws:s3:::host-web-alok/*"
13         }
14     ]
15 }
```

- 12) Now we can go to home page of our bucket and our static page will be loaded



Experiment No: 1(B)

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

Step 1: Set up a Cloud9 environment.

- 1) Go to Cloud9 services under developers tool in All services

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a 'Services' button, a search bar, and account information: 'N. Virginia' and 'vocabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR @ 3805-5794-4473'. Below the navigation is a sidebar with a list of services: Direct Connect, AWS App Mesh, Global Accelerator, AWS Cloud Map, Route 53 Application Recovery Controller, AWS Private 5G, Developer Tools (CodeStar, CodeCommit, CodeBuild, CodeDeploy, CodePipeline, Cloud9, CloudShell, X-Ray, AWS FIS, CodeArtifact, Amazon CodeCatalyst, AWS AppConfig, Amazon Q Developer (Including Amazon CodeWhisperer), Application Composer, AWS App Studio), Customer Enablement (AWS IQ, Managed Services), Analytics (Athena, Amazon Redshift, CloudSearch, Amazon OpenSearch Service), and various other services like Amazon Kendra, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Transcribe, Amazon Translate, AWS DeepComposer, AWS DeepRacer, AWS Panorama, Amazon Monitron, AWS HealthLake, Amazon Lookout for Vision, Amazon Lookout for Equipment, Amazon Lookout for Metrics, Amazon Lex, Amazon Comprehend Medical, AWS HealthOmics, Amazon Bedrock, AWS HealthImaging, Amazon Q, Amazon Q Business, and more. The 'Cloud9' service is clearly visible under the 'Developer Tools' section. The bottom of the page includes a URL 'https://us-east-1.console.aws.amazon.com/cloud9/home?region=us-east-1', a copyright notice '© 2024, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

- 2) Click on create environment

The screenshot shows the AWS Cloud9 landing page. The top navigation bar is identical to the previous screenshot. The main content area features the 'AWS Cloud9' logo and the tagline 'A cloud IDE for writing, running, and debugging code'. Below this is a paragraph: 'AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.' To the right of this text is a large orange button labeled 'Create environment'. Below the main text are two sections: 'How it works' (with a sub-section about creating an environment on an Amazon EC2 instance) and 'Getting started' (with links to 'Before you start', 'Create an environment', 'Working with environments', and 'Working with the IDE'). The bottom of the page includes a 'CloudShell' button, a 'Feedback' link, and the standard copyright and link footer.

3) Give the name to your Environment ,keeping the other settings as default like environment type should be New EC2 instance

Details

Name: WebAppIDE

Description - optional:

Environment type: [Info](#)
 Determines what the Cloud9 IDE will run on.

New EC2 instance
 Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute
 You have an existing instance or server that you'd like to use.

New EC2 instance

Instance type: [Info](#)
 The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GB RAM + 1 vCPU)
 Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GB RAM + 2 vCPU)
 Recommended for small web projects.

m5.large (8 GB RAM + 2 vCPU)
 Recommended for production and most general-purpose development.

Platform: [Info](#)
 This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

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

30 minutes

Network settings: [Info](#)

4) Select the correct platform type as shown below and keep the others details as default like instance type as t2.micro which gives the user 1GB RAM + 1 Virtual CPU

New EC2 instance

Instance type: [Info](#)
 The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GB RAM + 1 vCPU)
 Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GB RAM + 2 vCPU)
 Recommended for small web projects.

m5.large (8 GB RAM + 2 vCPU)
 Recommended for production and most general-purpose development.

Additional instance types
 Explore additional instances to fit your need.

Platform: [Info](#)
 This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

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

30 minutes

Network settings: [Info](#)

5) Click on SSH under connection type in network settings if we go for AWS Manager(SSM) then it won't allow to create an environment then click on Create

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

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- **AWSCloud9SSMAccessRole and AWSCloud9SSMInstanceProfile** - 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](#)

6) Successfully created the environment so now click on open

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

[AWS Cloud9](#) Environments

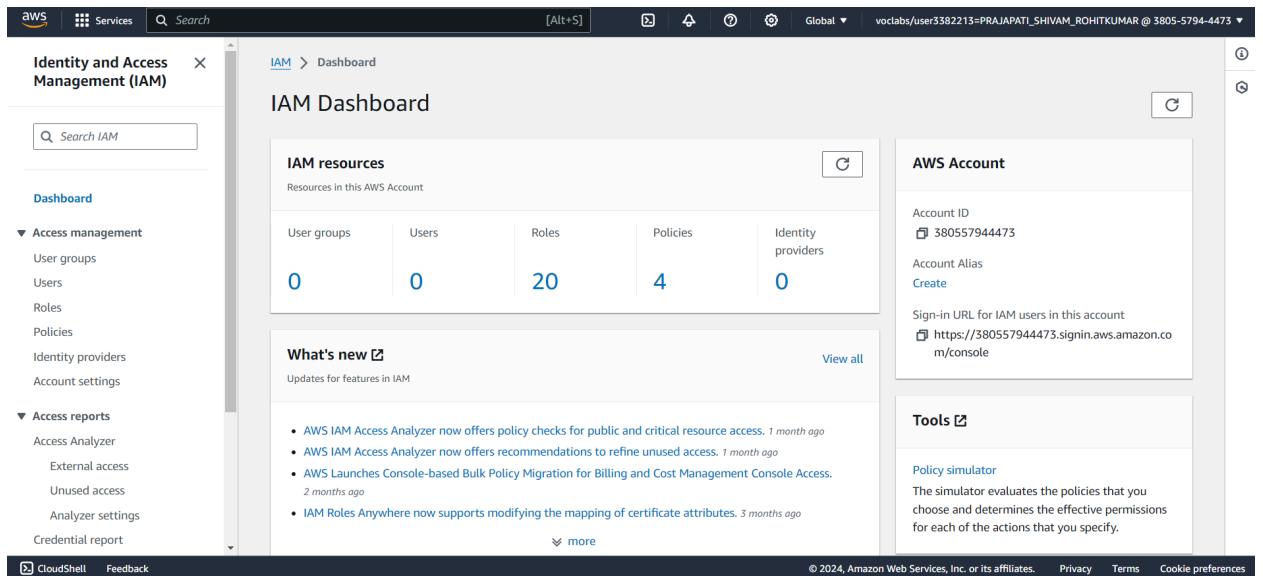
Environments (1)

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
WebAppIDE	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::38055794473:assumed-role/volcabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR

[Delete](#) [View details](#) [Open in Cloud9](#) [Create environment](#)

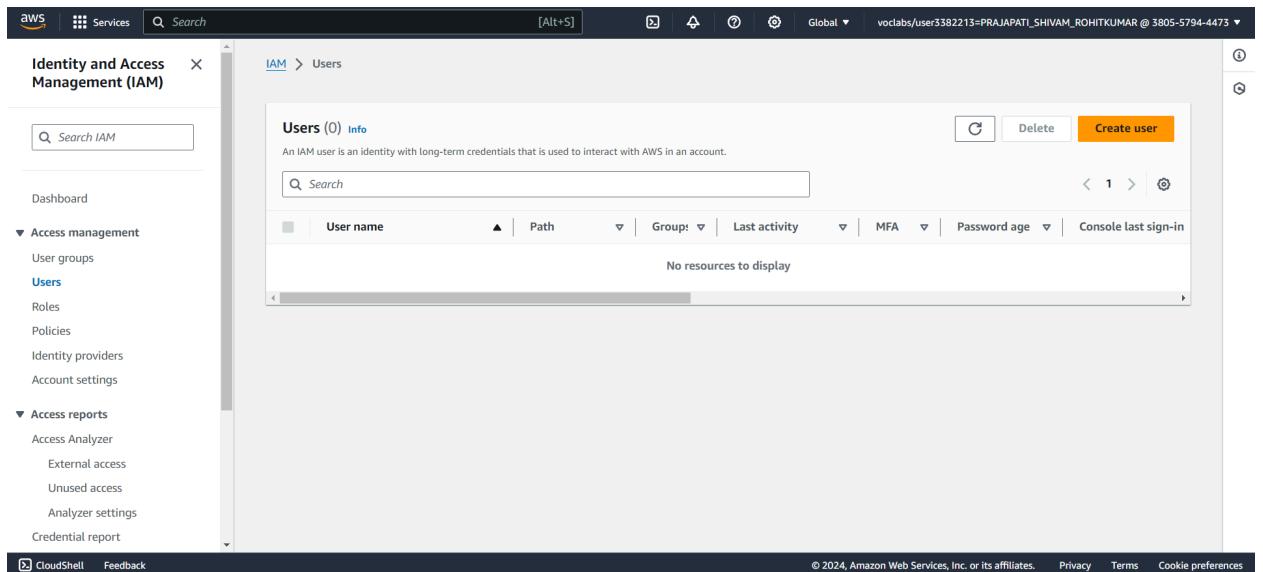
Step 2: Creating IAM user.

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



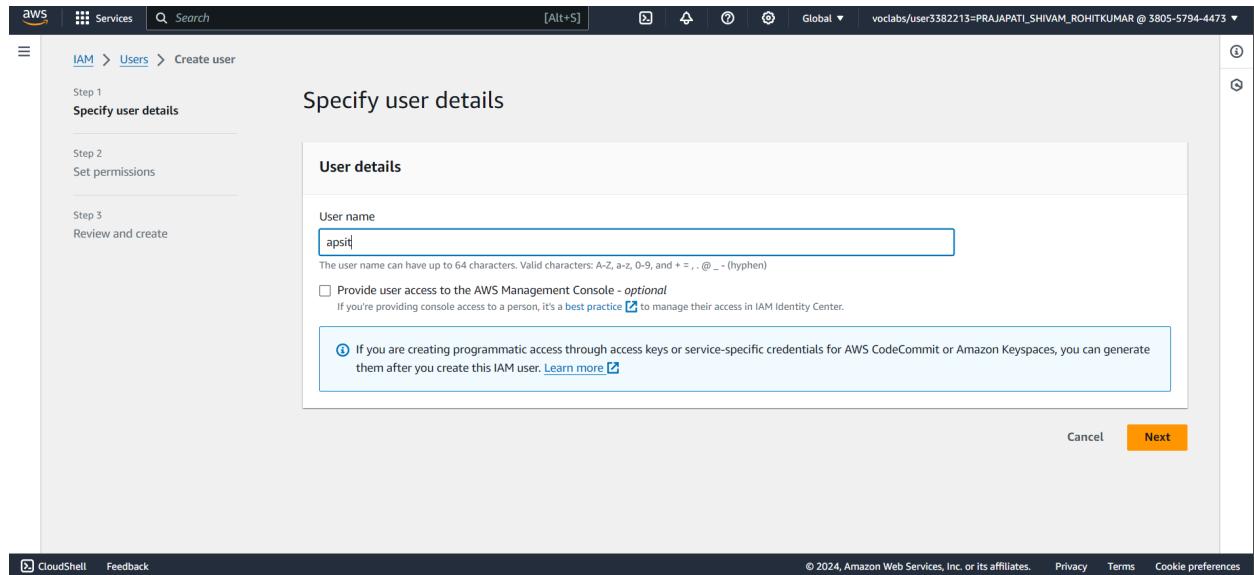
The screenshot shows the AWS IAM Dashboard. The left sidebar has 'Identity and Access Management (IAM)' selected. The main area displays 'IAM resources' with counts: 0 User groups, 0 Users, 20 Roles, 4 Policies, and 0 Identity providers. Below this is a 'What's new' section with several recent updates. On the right, there are sections for 'AWS Account' (Account ID: 38055794473, Account Alias: Create, Sign-in URL: https://38055794473.siginn.aws.amazon.co m/console) and 'Tools' (Policy simulator). The bottom right corner shows copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

2) Click on the create user

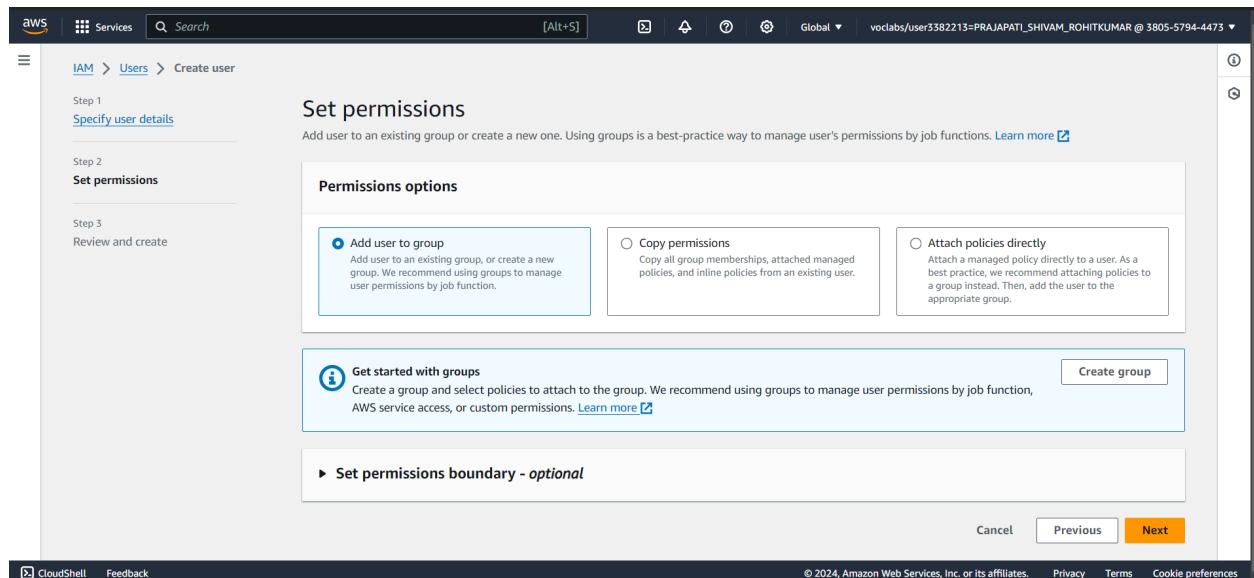


The screenshot shows the AWS IAM Users page. The left sidebar has 'Identity and Access Management (IAM)' selected. The main area shows a table titled 'Users (0) Info' with a note: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' The table has columns: User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. A message at the bottom of the table says 'No resources to display'. The bottom right corner shows copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

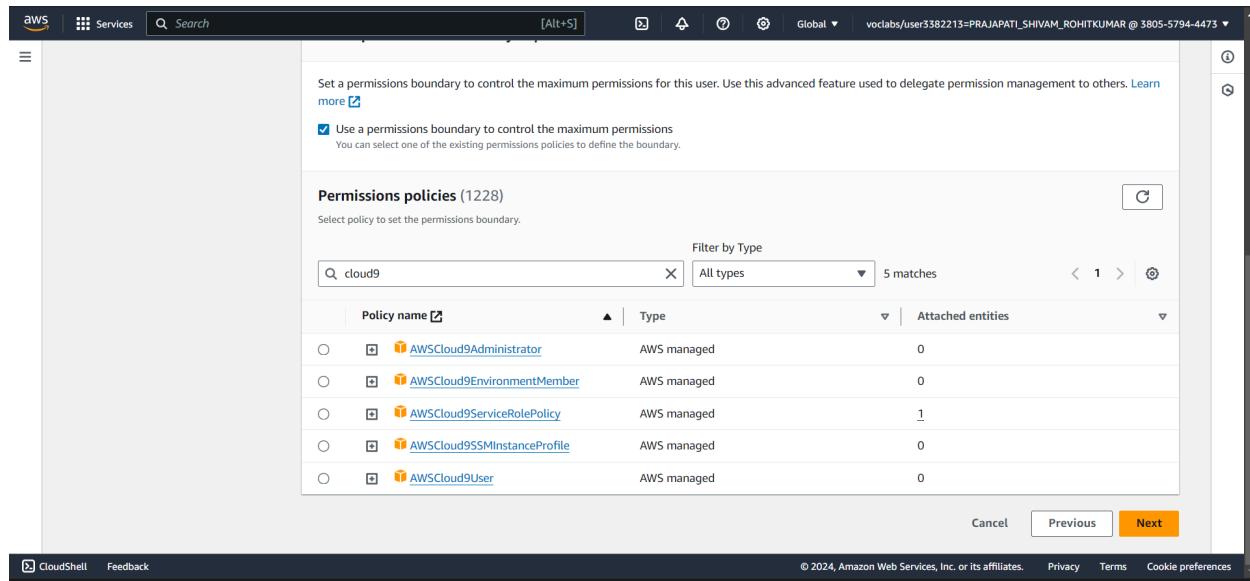
3) Write the name of the user you want to add and click on next



4) Select add User to Group. If there are no user groups on your accounts, you will have to create one. Click on Create Group. Click on the drop down menu of the set permissions boundary



5) Click on the checkbox and search for cloud9 under permissions policies ,click on next



Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

Use a permissions boundary to control the maximum permissions
You can select one of the existing permissions policies to define the boundary.

Permissions policies (1228)

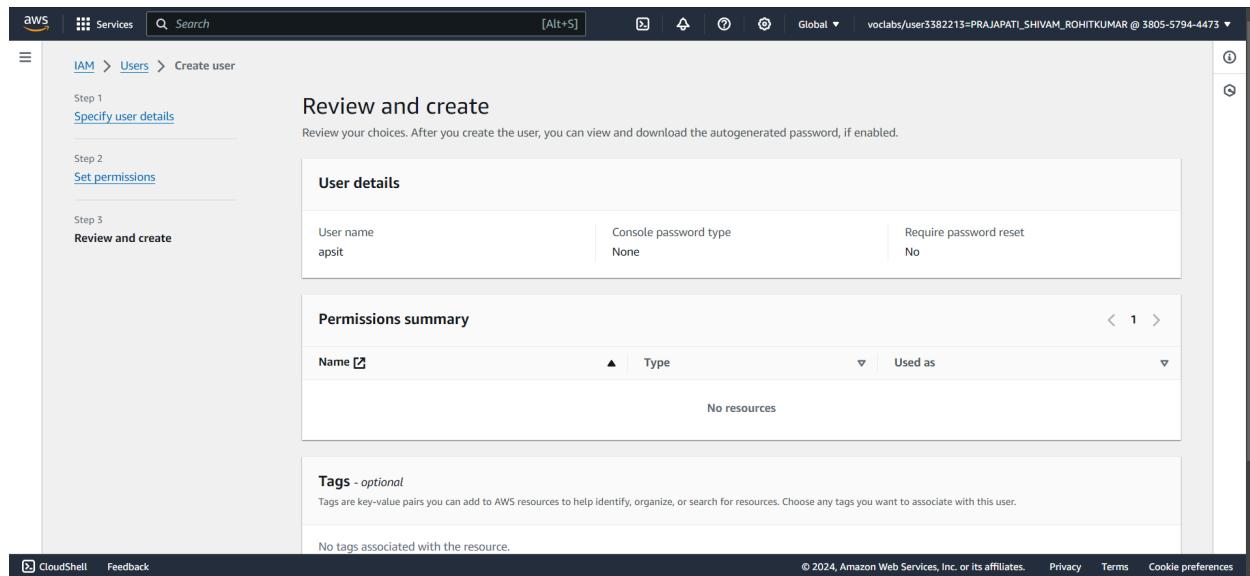
Select policy to set the permissions boundary.

Filter by Type

Policy name	Type	Attached entities
AWSCloud9Administrator	AWS managed	0
AWSCloud9EnvironmentMember	AWS managed	0
AWSCloud9ServiceRolePolicy	AWS managed	1
AWSCloud9SSMInstanceProfile	AWS managed	0
AWSCloud9User	AWS managed	0

Cancel Previous **Next**

6) Scroll down and click on create user



Step 1 [Specify user details](#)

Step 2 [Set permissions](#)

Step 3 [Review and create](#)

Review and create

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

User details

User name	apsit	Console password type	None	Require password reset	No
-----------	-------	-----------------------	------	------------------------	----

Permissions summary

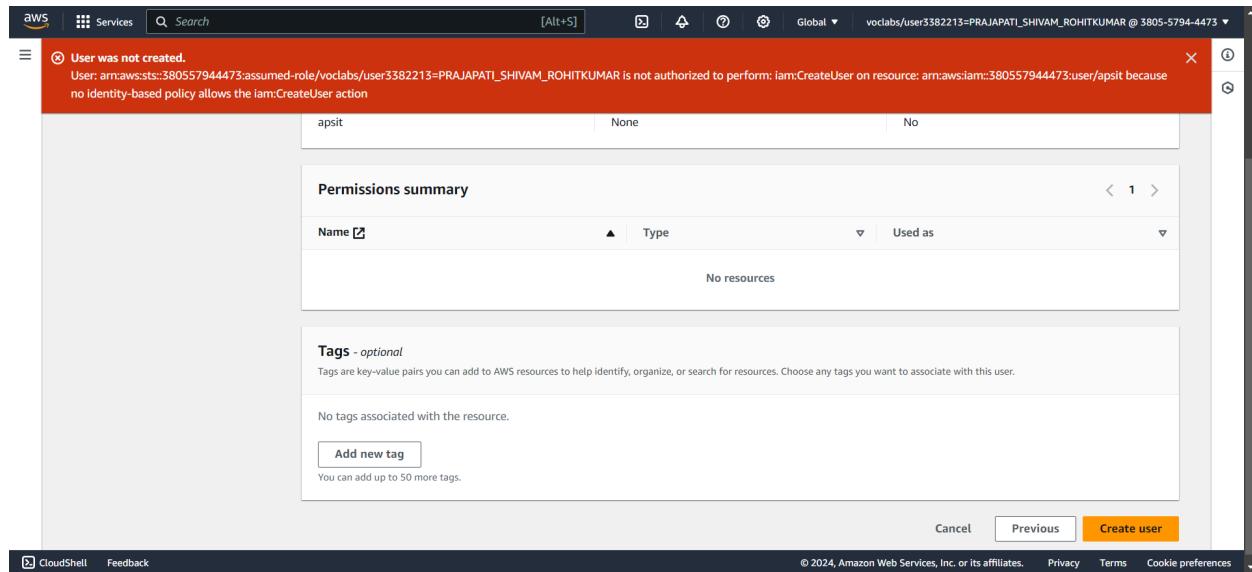
Name	Type	Used as
No resources		

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

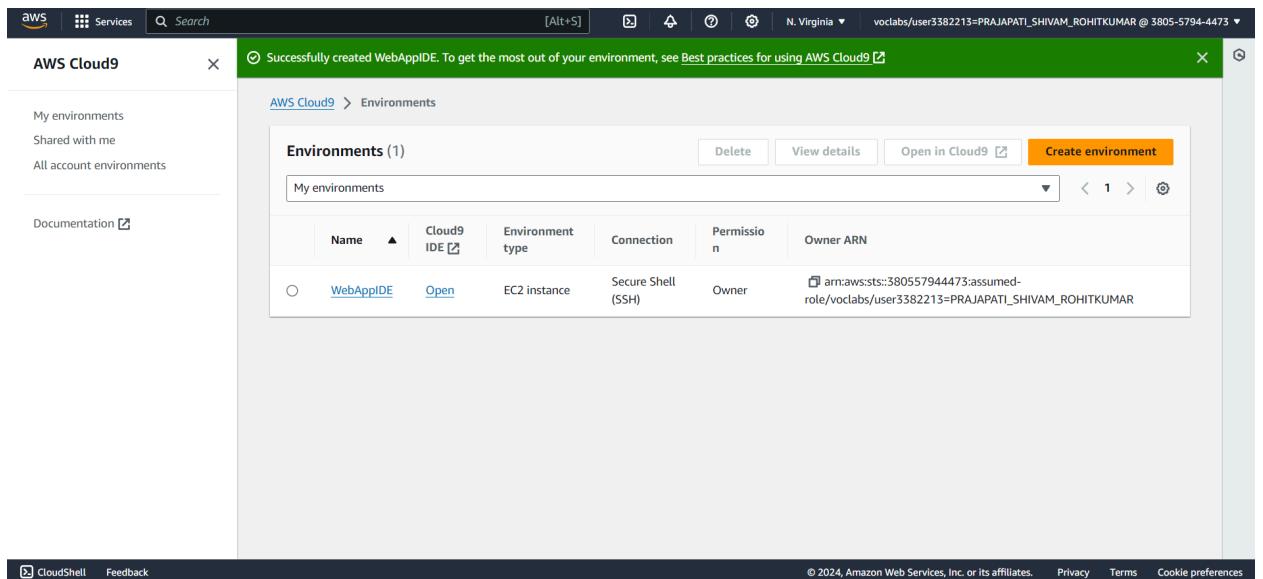
Next Step



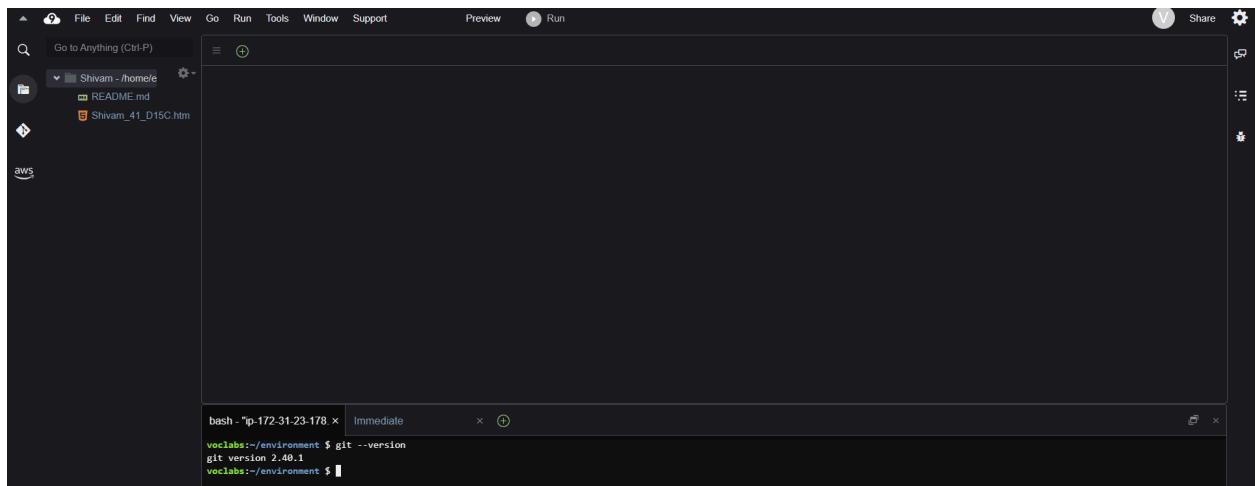
When we go to add user to a group, the AWS Academy account throws an error as we do not have the permissions to create a group. So we have to use our personal AWS account for this part.

Step 3: Working on Cloud9 IDE

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



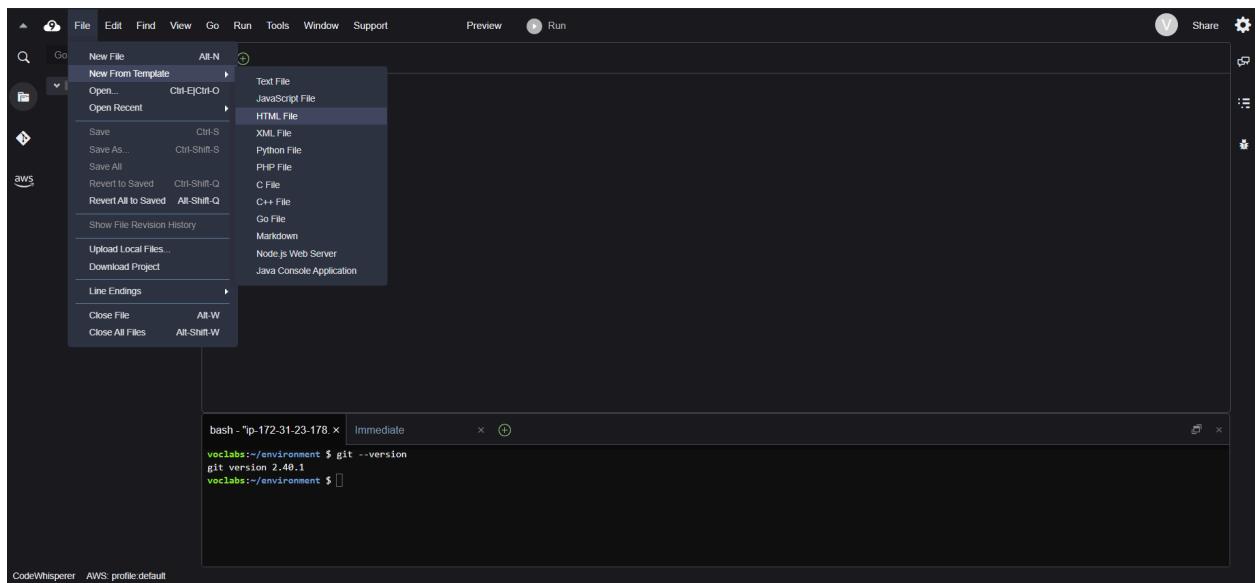
2) This is the Cloud9 IDE interface. The major part of the screen is the coding IDE. There is a command console just below it. For example, the command git --version is run.



The screenshot shows the Cloud9 IDE interface. On the left is a file explorer with a dark theme, showing a folder named 'Shivam - homefile' containing 'README.md' and 'Shivam_41_D15C.htm'. Below the file explorer is a sidebar with icons for 'aws' and other services. The main workspace is a large dark area. At the bottom is a terminal window with the following text:

```
bash - "ip-172-31-23-178.x" Immediate x ⓘ
voclabs:~/environment $ git --version
git version 2.40.1
voclabs:~/environment $
```

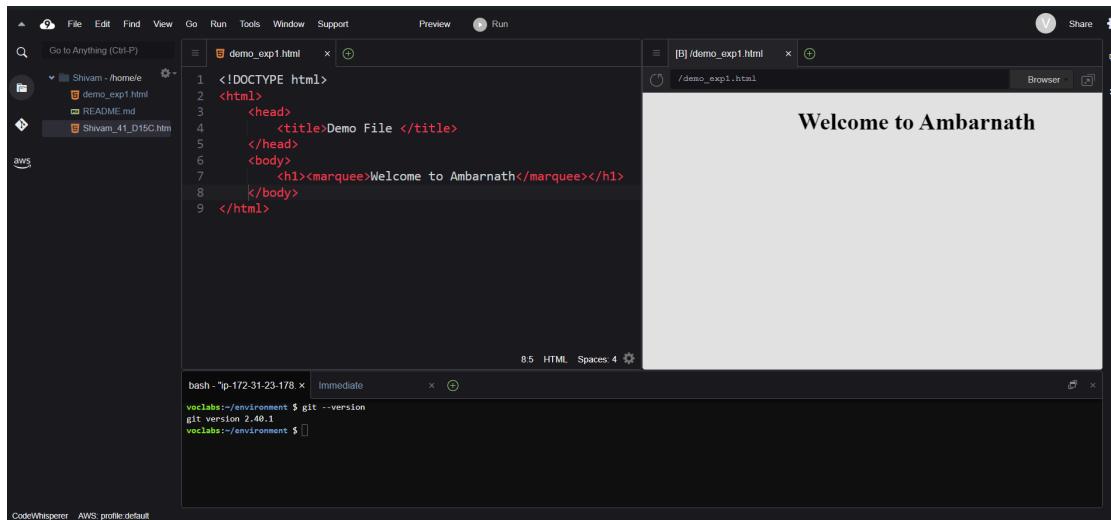
3) To add a file, click on file. For this experiment, we are to add an HTML file. So go to File → New From Template → HTML file. This gives a basic HTML template on the coding IDE



The screenshot shows the Cloud9 IDE interface with the 'File' menu open. The 'File' menu has the following options: New File (Alt+N), New From Template (highlighted), Open... (Ctrl+E/Ctrl+O), Open Recent, Save (Ctrl+S), Save As... (Ctrl+Shift+S), Save All, Revert to Saved (Ctrl+Shift+Q), Revert All to Saved (Alt+Shift+Q), Show File Revision History, Upload Local Files..., Download Project, Line Endings, Close File (Alt+W), and Close All Files (Alt+Shift+W). A sub-menu for 'New From Template' is open, showing options: Text File, JavaScript File, HTML File (highlighted), XML File, Python File, PHP File, C File, C++ File, Go File, Markdown, Node.js Web Server, and Java Console Application. Below the menu is a terminal window with the same text as the previous screenshot:

```
bash - "ip-172-31-23-178.x" Immediate x ⓘ
voclabs:~/environment $ git --version
git version 2.40.1
voclabs:~/environment $
```

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



The screenshot shows the CodeWhisperer IDE interface. On the left, there is a file tree with a file named 'demo_expt.html'. The main workspace shows the following HTML code:

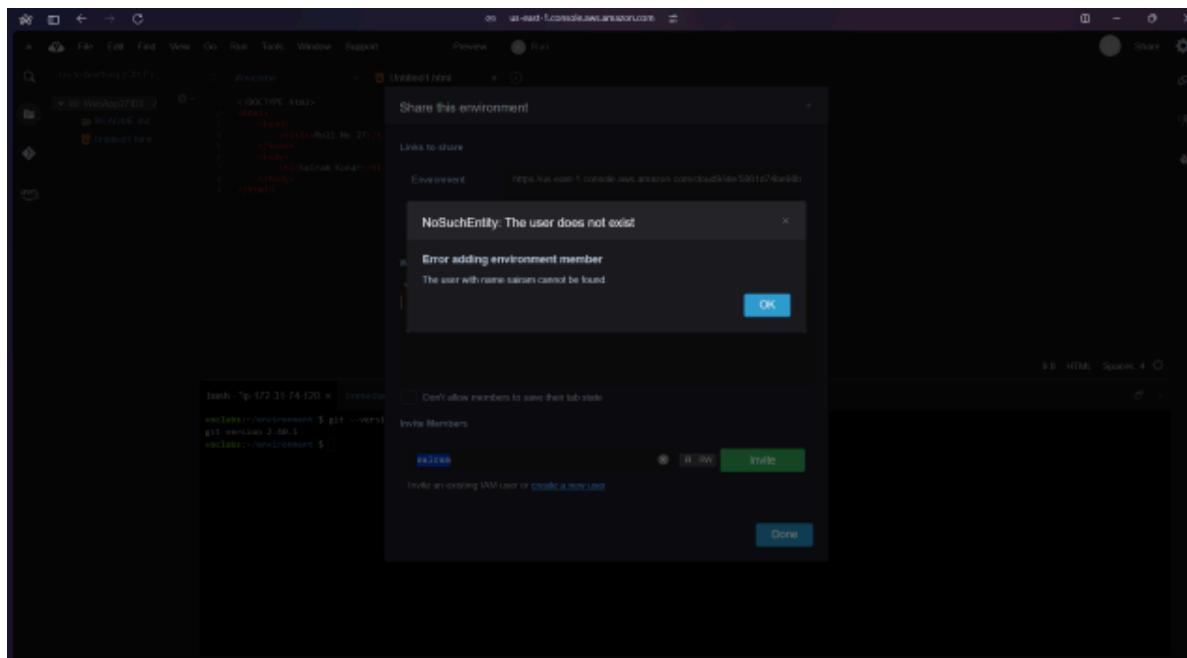
```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo File </title>
  </head>
  <body>
    <h1><marquee>Welcome to Ambarnath</marquee></h1>
  </body>
</html>
```

Below the code editor, there is a terminal window showing the following command:

```
git version 2.40.1
```

On the right, a browser window displays the rendered HTML with the heading 'Welcome to Ambarnath' in a marquee.

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



The screenshot shows the Cloud9 IDE interface. A 'Share this environment' dialog is open. Below it, an error message box displays:

NoSuchEntity: The user does not exist

Error adding environment member
The user with name saram cannot be found.

OK

At the bottom, there is an 'Invite Members' dialog with a list of users and an 'Invite' button.

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

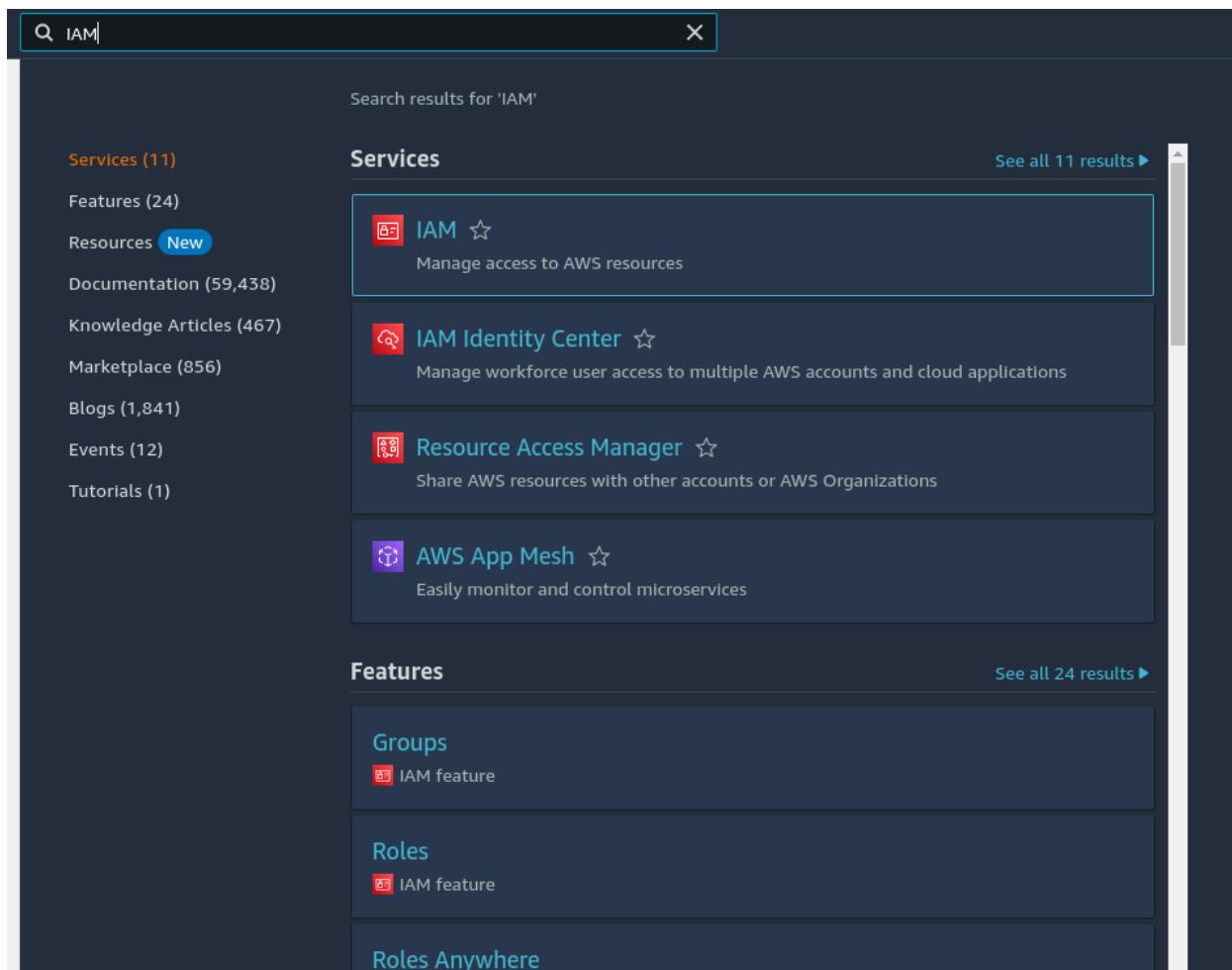
Experiment No. 2

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

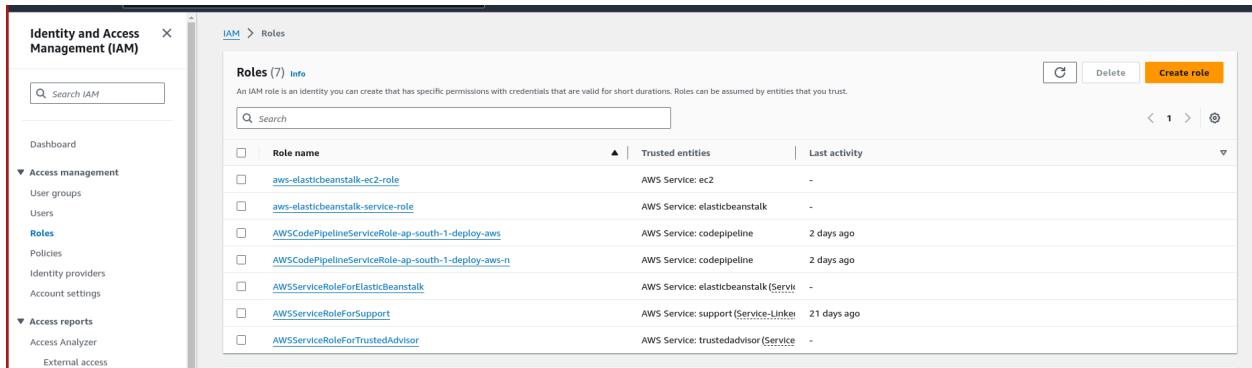
Steps:

Initially we will create a new Role in IAM

1. Search for IAM in search box



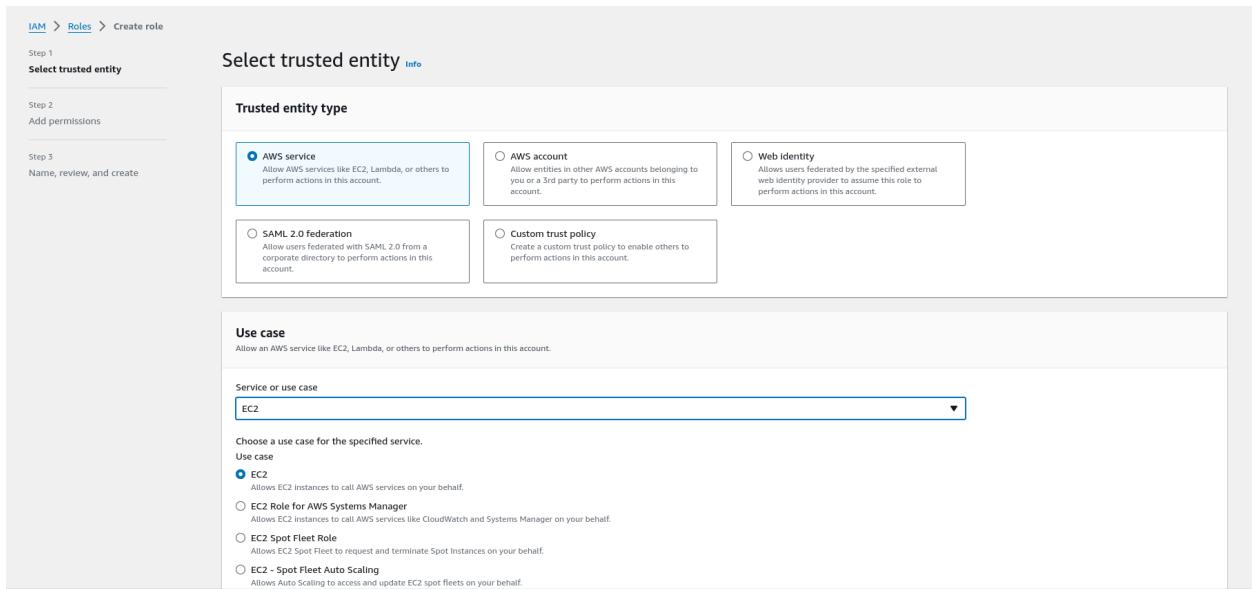
2. Go to Roles, and select AWSElasticBeanstalkWebtier and AWSElasticBeanstalkWorkertier, and create a role



The screenshot shows the AWS IAM 'Roles' list. There are 7 roles listed:

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	-
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSCodePipelineServiceRole-ap-south-1-deploy-aws	AWS Service: codepipeline	2 days ago
AWSCodePipelineServiceRole-ap-south-1-deploy-aws-n	AWS Service: codepipeline	2 days ago
AWSServiceRoleForElasticBeanstalk	AWS Service: elasticbeanstalk	-
AWSServiceRoleForSupport	AWS Service: support	21 days ago
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor	-

3. Select entity type as AWS Service



The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. The 'Trusted entity type' section shows the following options:

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

The 'Use case' section shows the following options:

- Service or use case: EC2
- Choose a use case for the specified service.
- Use case:
 - EC2: Allows EC2 instances to call AWS services on your behalf.
 - EC2 Role for AWS Systems Manager: Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
 - EC2 Spot Fleet Role: Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
 - EC2 - Spot Fleet Auto Scaling: Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

Step 3: Add tags

Add tags - *optional* info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

4. Give name to newly created role such as **aws-elasticbenstalk-ec2-role**

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

aws-elasticbenstalk-ec2-role

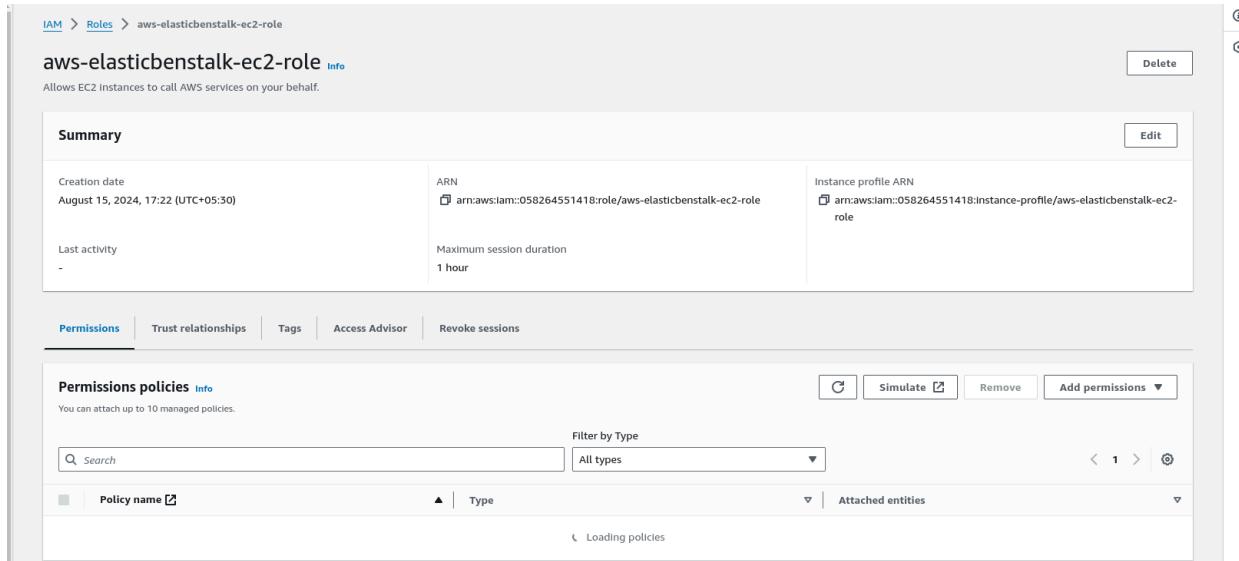
Description

Add a short explanation for this role.

Allows EC2 Instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,.@-_`

5. Summary of newly created role



IAM > Roles > aws-elasticbeanstalk-ec2-role

aws-elasticbeanstalk-ec2-role [Info](#)

Allows EC2 Instances to call AWS services on your behalf.

Summary	
Creation date	ARN
August 15, 2024, 17:22 (UTC+05:30)	arn:aws:iam::058264551418:role/aws-elasticbeanstalk-ec2-role
Last activity	Instance profile ARN
-	arn:aws:iam::058264551418:instance-profile/aws-elasticbeanstalk-ec2-role
Maximum session duration	1 hour

[Edit](#) [Delete](#) [@](#)

Permissions [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

Permissions policies [Info](#)

You can attach up to 10 managed policies.

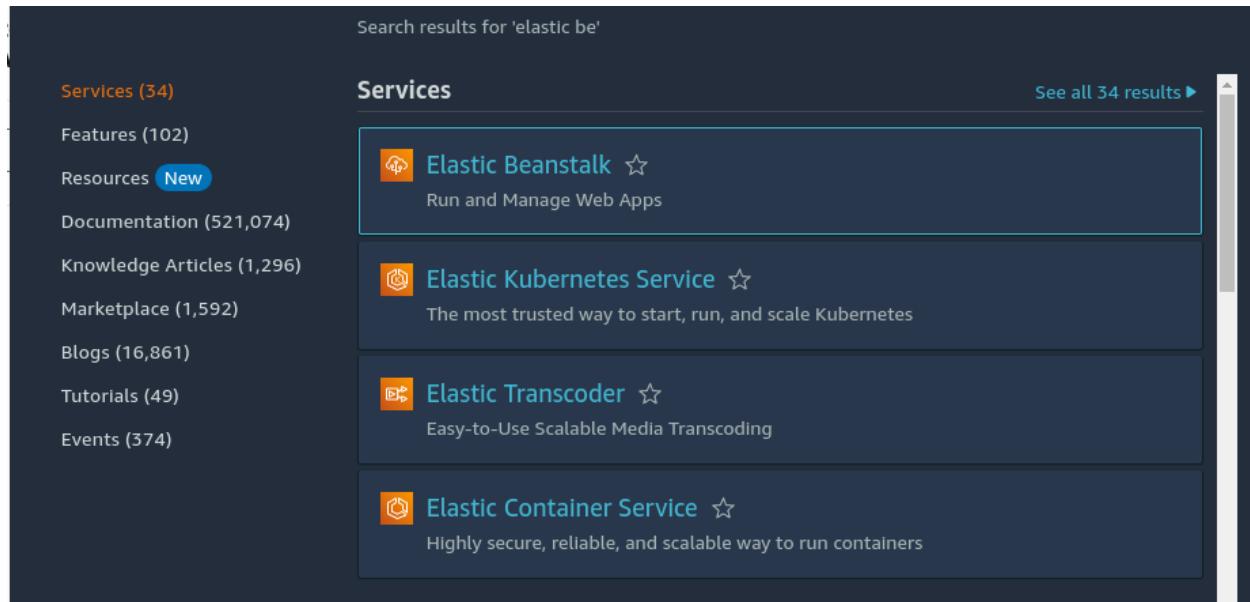
Filter by Type	
<input type="text" value="Search"/>	All types
<input type="checkbox"/> Policy name Info	▲ Type
▼ Attached entities	

Loading policies

[Add permissions](#)

Now we have to create Elastic Beanstalk Environment

6. Search **Elastic Beanstalk** and proceed with it



Search results for 'elastic be'

Services (34)

- Features (102)
- Resources [New](#)
- Documentation (521,074)
- Knowledge Articles (1,296)
- Marketplace (1,592)
- Blogs (16,861)
- Tutorials (49)
- Events (374)

Services

[See all 34 results ▶](#)

 Elastic Beanstalk ☆
Run and Manage Web Apps
 Elastic Kubernetes Service ☆
The most trusted way to start, run, and scale Kubernetes
 Elastic Transcoder ☆
Easy-to-Use Scalable Media Transcoding
 Elastic Container Service ☆
Highly secure, reliable, and scalable way to run containers

7. Create a new Environment and name it.

Step 1
Configure environment

Step 2
Configure service access

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

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure environment Info

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

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

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

Application information Info

Application name
alok-app

Maximum length of 100 characters.

► Application tags (optional)

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

Environment name
Alok-app-env

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

Domain
Leave blank for autogenerated value .ap-south-1.elasticbeanstalk.com [Check availability](#)

8. Select Platform as PHP, Application code as **Sample Application**, presets **Single Instance**

Platform Info

Platform type

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

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

Platform

PHP 

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023 

Platform version

4.3.2 (Recommended) 

Application code Info

Sample application

Existing version
Application versions that you have uploaded.

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

Presets Info

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

Configuration presets

Single instance (free tier eligible)

Single instance (using spot instance)

High availability

High availability (using spot and on-demand instances)

Custom configuration

9. Under Service access settings, select **Use an existing service role**. Name service role as **aws-elasticbeanstalk-service-role** and EC2

instance profile as **aws-elasticbeanstalk-ec2-role** which we had created earlier

Step 1
[Configure environment](#)

Step 2
Configure service access

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

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

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

Step 6
[Review](#)

Configure 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.

aws-elasticbeanstalk-service-role

EC2 key pair

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

ubuntu-linux

EC2 instance profile

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

aws-elasticbeanstalk-ec2-role

[View permission details](#)

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

10. Keep below settings as it as

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

Step 4 - optional
Configure instance traffic and scaling

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

Step 6
[Review](#)

Configure instance traffic and scaling - *optional* [Info](#)

▼ Instances [Info](#)
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

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

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

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

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

Monitoring interval

Instance metadata service (IMDS)
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv1, deactivate IMDSv1. [Learn more](#) 

IMDSv1
With the current setting, the environment enables only IMDSv2.
 Deactivated

EC2 security groups
Select security groups to control traffic.

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

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

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure 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

CloudWatch Custom Metrics - Environment

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
Keep logs after terminating environment

▼ Managed platform updates [Info](#)
Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates
 Activated

11. Our environment is created successfully

Elastic Beanstalk X

Environment successfully launched.

[Elastic Beanstalk](#) > [Environments](#) > **Alok-app-env**

Alok-app-env [Info](#) [Actions](#) [Upload and deploy](#)

Environment overview		Platform
Health	OK	Platform
Domain	Alok-app-env.eba-wcjuvikb.ap-south-1.elasticbeanstalk.com	PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2
		Running version
		Platform state Supported

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

12. Now we would deploy a codepipeline. For below repo in your github

Fork this repository in your github.

imoisharma / aws-codepipeline-s3-codedeploy-linux-2.0

Code Issues Pull requests Actions Projects Security Insights

aws-codepipeline-s3-codedeploy-linux-2.0 Public

Watch 3

master 1 Branch 0 Tags Go to file Add file Code

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

.github Adding template 7 years ago

dist Added dist folder 9 years ago

scripts s3 setup and s3 set cache control scripts 3 years ago

CODE_OF_CONDUCT.md Adding CONTRIBUTING/CoC 7 years ago

CONTRIBUTING.md Adding CONTRIBUTING/CoC 7 years ago

LICENSE Added AWS CodePipeline Sample 9 years ago

README.md Update README.md 3 years ago

app-specification.yml Create app-spec config file 3 years ago

appspec.yml Update appspec.yml 3 years ago

index.html Update index.html 3 years ago

imoisharma / aws-codepipeline-s3-codedeploy-linux-2.0

Type to search

Code Issues Pull requests Actions Projects Security Insights

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

aaaalok / aws-codepipeline-s3-cod

aws-codepipeline-s3-codedeploy-linux-2.0 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Wa

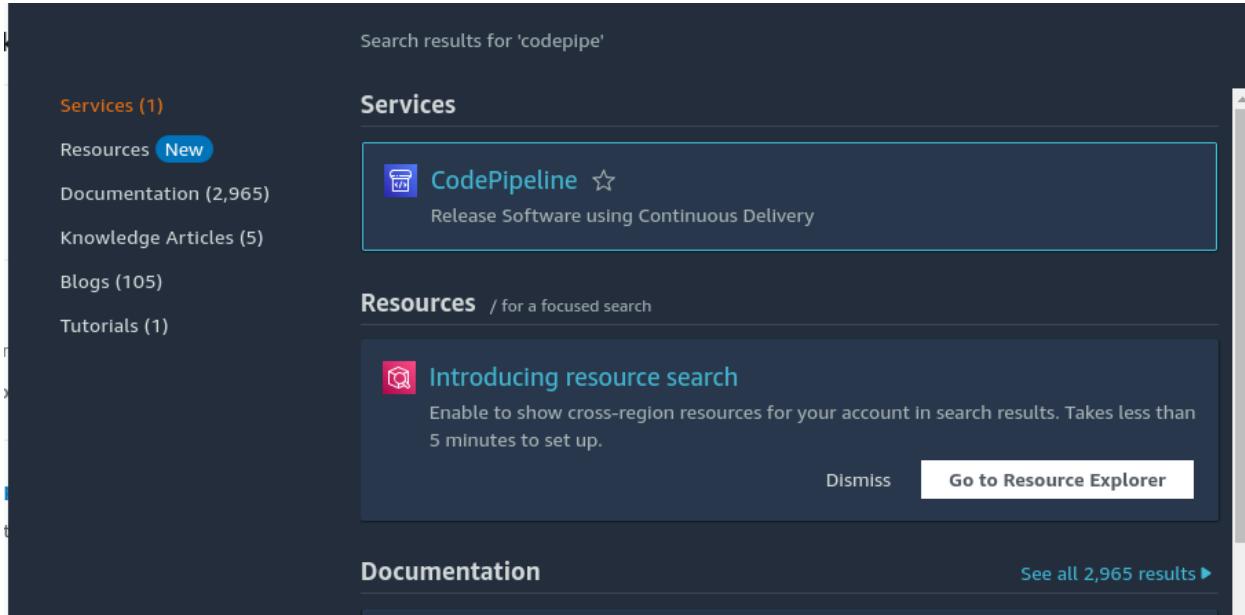
Copy the master branch only

Contribute back to imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0 by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

13. After forking, Now we would create a CodePipeline. Goto CodePipeline under services section



Search results for 'codepipe'

Services (1)

Resources [New](#)

Documentation (2,965)

Knowledge Articles (5)

Blogs (105)

Tutorials (1)

Services

CodePipeline ☆

Release Software using Continuous Delivery

Resources / for a focused search

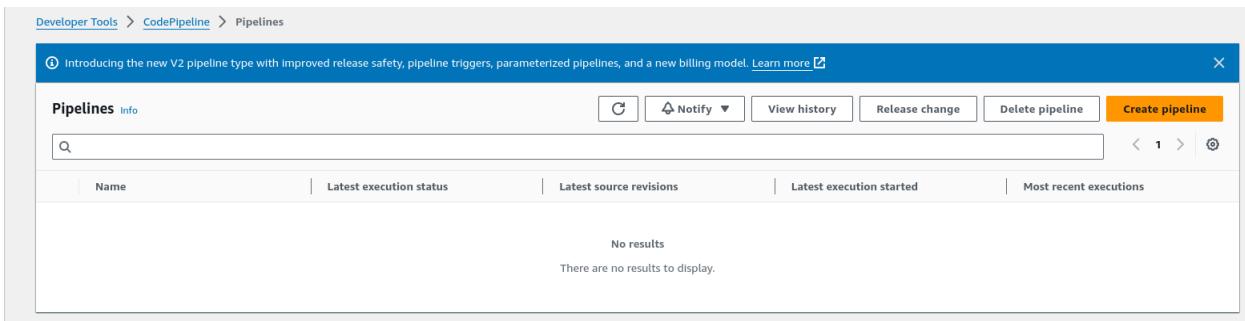
Introducing resource search

Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.

Dismiss [Go to Resource Explorer](#)

Documentation [See all 2,965 results ▶](#)

14. Create a new PipeLine



Developer Tools > [CodePipeline](#) > Pipelines

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more [\[?\]](#)

Pipelines [Info](#)

[Create pipeline](#)

Q

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

15. Name the pipeline leaving rest settings to its default

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings Info
Step 1 of 5

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Pipeline settings

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

Pipeline type

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

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

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

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

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

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name

Type your service role name

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

16. Under Source stage, select **Source Provider** as **GitHub (Version 2)**

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

GitHub (Version 2) ▾

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

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

or **Connect to GitHub**

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

X

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

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

X

Output artifact format
Choose the output artifact format.

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

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

17. Connect your GitHub account to AWS for it to build and deploy and track changes on repo

AWS Connector for GitHub by **Amazon Web Services** would like permission to:



Verify your GitHub identity (aaaalok)



Know which resources you can access



Act on your behalf

[Learn more](#)

[Learn more about AWS Connector for GitHub](#)

[Cancel](#)

[Authorize AWS Connector for GitHub](#)

Authorizing will redirect to

<https://redirect.codestar.aws>



Not owned or operated by GitHub



Created 4 years ago



More than 1K GitHub users

18. As of Now Skip **Build stage** under **Deploy Stage** enter AWS Elastic Beanstalk as Deploy Provider

Step 2
[Add source stage](#)

Step 3
[Add build stage](#)

Step 4
[Add deploy stage](#)

Step 5
Review

Deploy

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

AWS Elastic Beanstalk ▾

Region
Asia Pacific (Mumbai) ▾

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.



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.



Alok-app-env

Cancel  

19. Review the summary of Pipeline created and click on **Create Pipeline**

Choose pipeline settings Step 5 of 5

Step 2
[Add source stage](#)

Step 3
[Add build stage](#)

Step 4
[Add deploy stage](#)

Step 5
[Review](#)

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name: aloky

Pipeline type: V2

Execution mode: QUEUED

Artifact location: codepipeline-ap-south-1-13487781303

Service role name: AWSCodePipelineServiceRole-ap-south-1-aloky

Variables

Name	Default value	Description
No variables		

No variables defined at the pipeline level in this pipeline.

Step 2: Add source stage

Source action provider

Source action provider

Source action provider
GitHub (Version 2)

OutputArtifactFormat
CODE_ZIP

DetectChanges
true

ConnectionArn
arn:aws:codeconnections:ap-south-1:058264551418:connection/142a38a1-c5e6-4700-ab3f-73ffa5543ea4

FullRepositoryId
aaaalok/aws-codedepipeline-s3-codedeploy-linux-2.0

Default branch
master

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

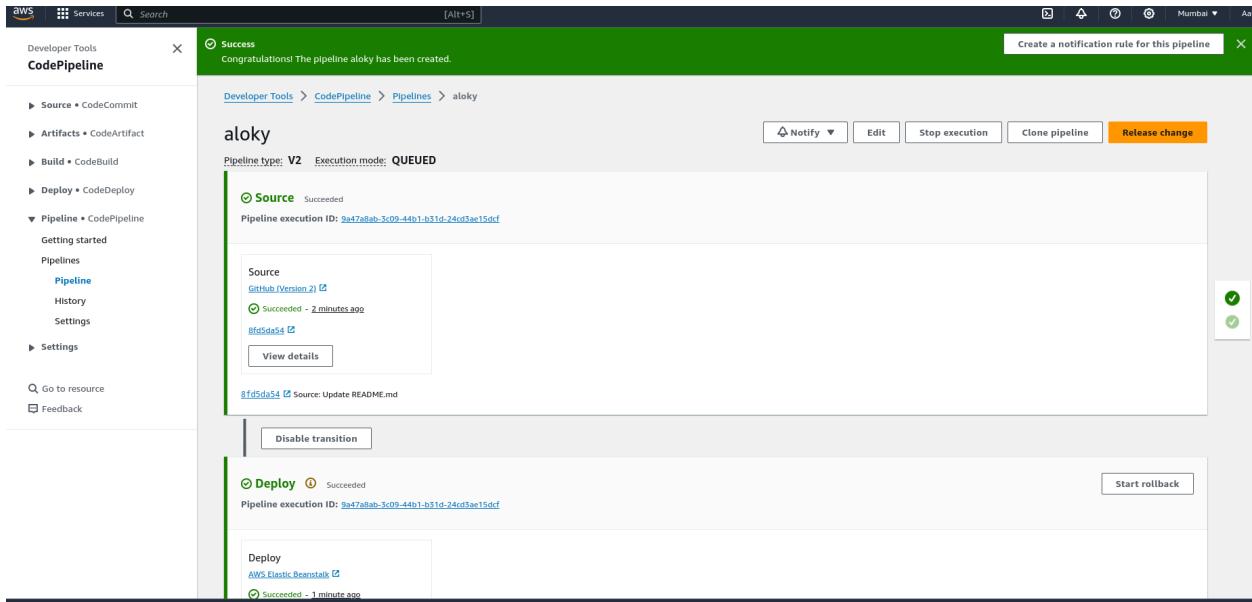
Trigger type
No filter

Step 3: Add build stage

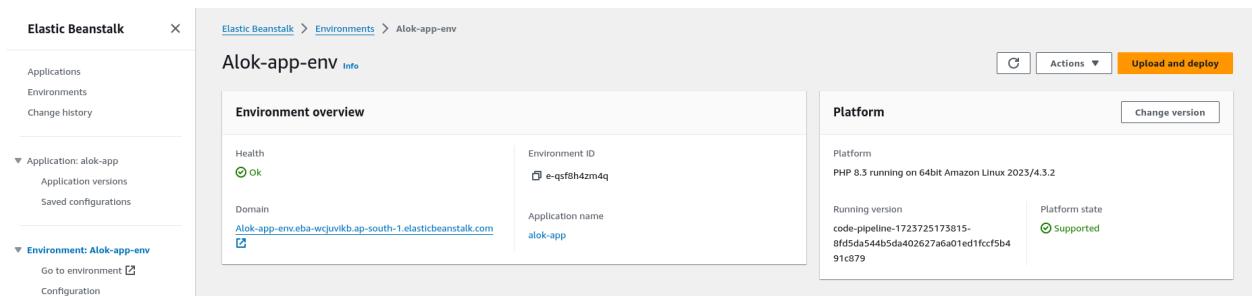
Build action provider

Build stage
No build

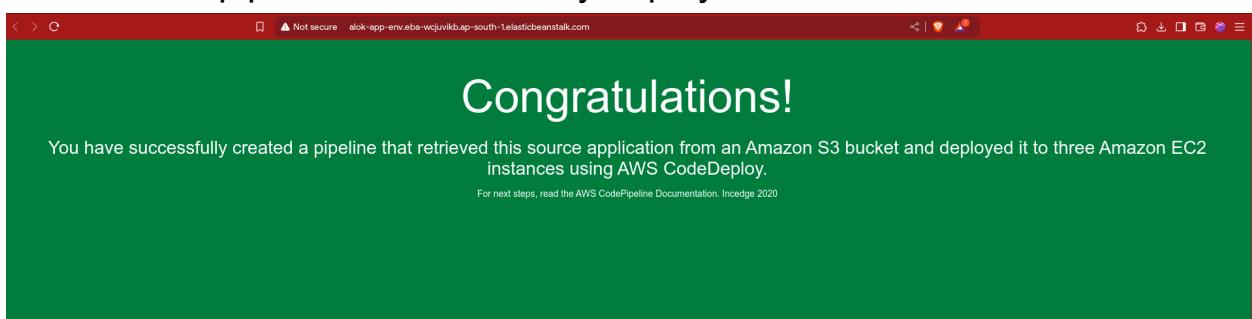
20. Our pipeline will be created and deployed in few minutes



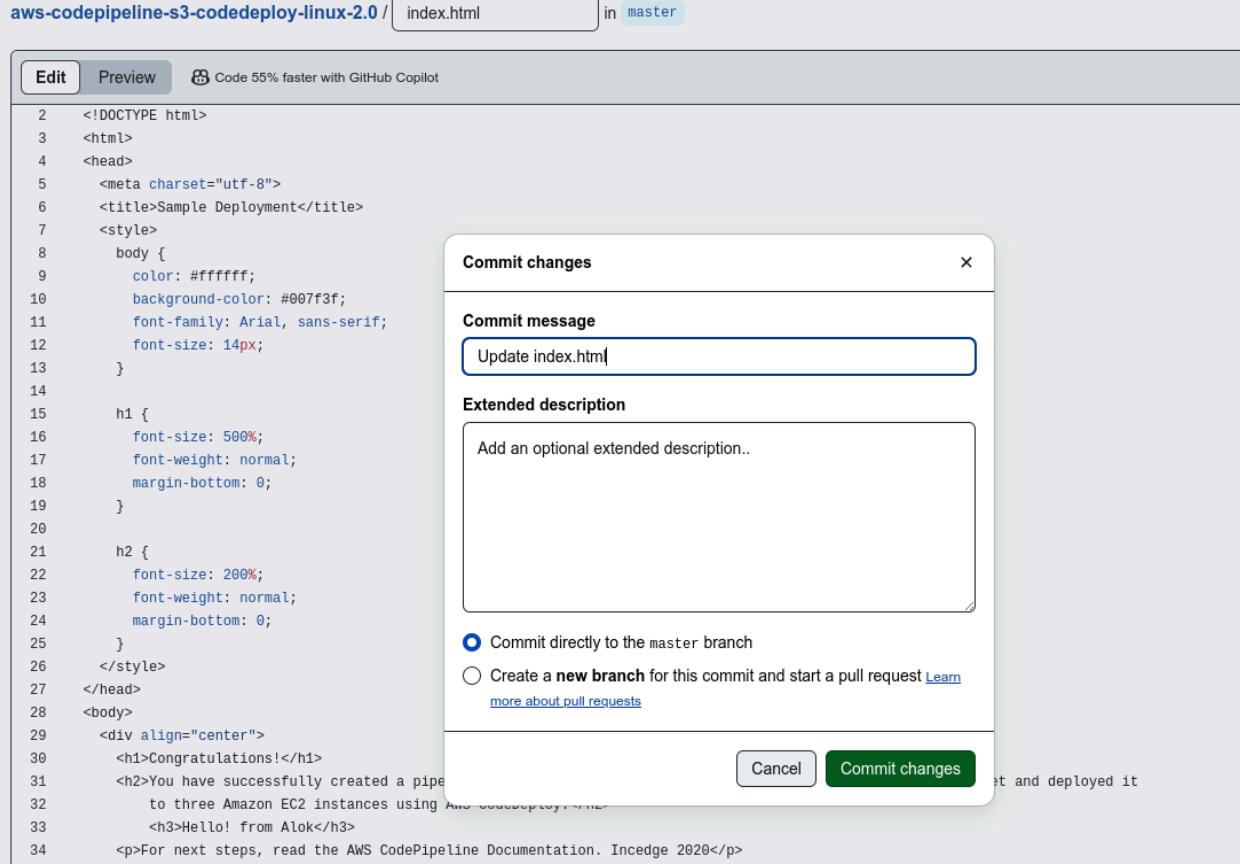
21. Go to Elastic Beanstalk environment that we had created previously and click on the URL given



22. Our pipeline is successfully deployed



23. Now to test our pipeline, we would do some changes in files in the repository



aws-codepipeline-s3-codedeploy-linux-2.0 / index.html in master

Commit changes

Commit message
Update index.html

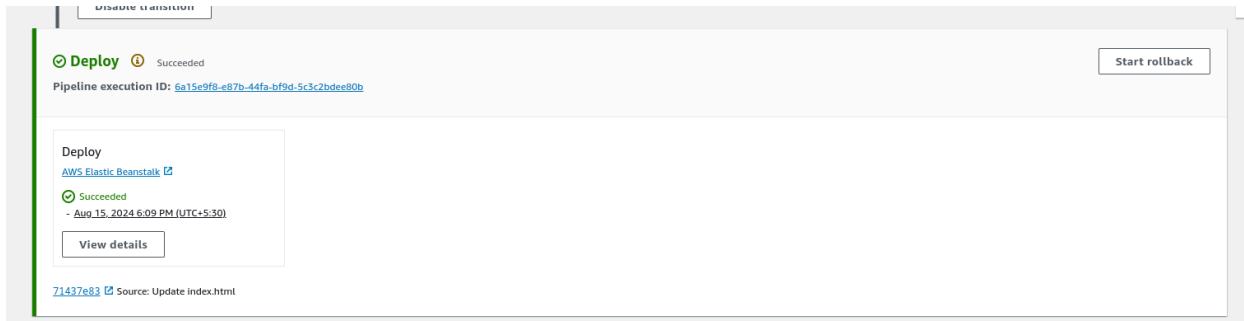
Extended description
Add an optional extended description..

Commit directly to the master branch
 Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

Cancel Commit changes

Use [control + shift + m] to toggle the [tab] key moving focus. Alternatively, use [esc] then [tab] to move to the next interactive element on the page.

24. Deploy process is automatically started on detecting change in repository



25. A new text **Hello! from Alok** is shown in the website.



EXPERIMENT NO. 3

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

To efficient run kubernetes cluster select instance type of at least t2.medium as kubernetes recommends at least 2 vCPU to run smoothly

	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	kube-master	i-00aa79ac09d7462c0	✓ Running	t2.medium
<input type="checkbox"/>	kube-worker1	i-0bab86cd3fbfcba0a	✓ Running	t2.medium
<input type="checkbox"/>	kube-worker2	i-00dcfd302ffd80dda	✓ Running	t2.medium

2. SSH into all 3 machines each in separate terminal

```
quantum@machine ~ ~/Downloads ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-3-88-111-183.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-111-183.compute-1.amazonaws.com (3.88.111.183)' can't be established.
ED25519 key fingerprint is SHA256:pQu+xs9foYbY3de1twjZcVVVA0zmGwGv6PHmVrUF/Q1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-111-183.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

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

Install Docker

```
sudo yum install docker -y
```

```
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
=====
 Package           Architecture
=====
Installing:
 docker           x86_64
Installing dependencies:
 containerd        x86_64
 iptables-libs    x86_64
 iptables-nft     x86_64
 libcgroup         x86_64
 libnetfilter_conntrack x86_64
 libnftnetlink    x86_64
 libnftnl          x86_64
 pigz             x86_64
 runc             x86_64

Transaction Summary
```

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

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

After configuring restart docker service service :

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

```
[ec2-user@ip-172-31-81-63 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-81-63 docker]$
```

4. Install Kubernetes on all 3 machines

SELinux needs to be disabled before configuring kubelet to avoid interference with kubernetes api server

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

```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

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/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-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
=====
Package                           Architecture      Version
=====
Installing:
kubeadm                         x86_64          1.30.4-150500.1.1
kubectl                          x86_64          1.30.4-150500.1.1
kubelet                          x86_64          1.30.4-150500.1.1
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
socat                            x86_64          1.7.4.2-1.amzn2023.0.2
Transaction Summary
=====
Install 10 Packages
```

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: 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 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
    --discovery-token-ca-cert-hash sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8c5
[ec2-user@ip-172-31-81-63 docker]$ mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-63 docker]$
```

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 instance)

```
kubeadm join 172.31.81.63:6443 --token
zh5jbb.a6ty3eujzc51d15d \
    --discovery-token-ca-cert-hash
sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6c
d2d2f6f8c5
```

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

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

```
[ec2-user@ip-172-31-81-63 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml  
namespace/kube-flannel created  
clusterrole.rbac.authorization.k8s.io/flannel created  
clusterrolebinding.rbac.authorization.k8s.io/flannel created  
serviceaccount/flannel created  
configmap/kube-flannel-cfg created  
daemonset.apps/kube-flannel-ds created
```

Check the created pod using this command

- kubectl get pods

6. Perform this ONLY on the worker machines

Paste the below command on all 2 worker machines

- sudo yum install iproute-tc socat -y (necessary packages required by kubernetes)
 - sudo systemctl enable kubelet
 - sudo systemctl restart kubelet
-
- kubeadm join 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
--discovery-token-ca-cert-hash

```
sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8  
c5
```

With the help of command the worker nodes are connected master node and is ready to do task assigned by master node

Now we can see in the master/control node of kubernetes that worker nodes are connected by typing **watch kubectl get nodes** in the **master node instance**

```
Every 2.0s: kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-81-63.ec2.internal	Ready	control-plane	29m	v1.30.4
ip-172-31-87-137.ec2.internal	Ready	<none>	5m58s	v1.30.4
ip-172-31-92-18.ec2.internal	Ready	<none>	5m53s	v1.30.4

Conclusion: We began with installation and configuration of necessary packages required by kubernetes. Some of them were available in the repository of the distribution of linux but some of them were not available so had to add their repository for installation. Even after setting up, the nodes were tainted which was the reason kubernetes api server was crashing, we then fixed it by making them untainted. We even disabled SELINUX to prevent any interference. In this experiment we successfully connected worker nodes with master nodes of kubernetes.

EXPERIMENT No.4

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

STEPS:

1. 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 with the text "AMZ - nginx" and a "Add additional tags" button. Below the search bar, a section titled "Application and OS Images (Amazon Machine Image)" is expanded, showing the following content:

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

Recent and Quick Start tabs: **Recent** (selected), **Quick Start**

AMI selection cards:

- Amazon Linux: AWS logo
- macOS: Mac logo
- Ubuntu: Ubuntu logo
- Windows: Microsoft logo
- Red Hat: Red Hat logo
- SUSE Li: SUSE logo

Search icon and "Browse more AMIs" link: **Browse more AMIs**
Including AMIs from AWS, Marketplace and the Community

Footer: **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+9Hn5KWLOYZMmzh1wg/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.
      \#/
      \~'-->
      /   A newer version of Amazon Linux is available!
      /_  Amazon Linux 2023, GA and supported until 2028-03-15.
      /m/  https://aws.amazon.com/Linux/amazon-linux-2023/
[ec2-user@ip-172-31-31-63 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-31-63 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-31-63 ~]$
```

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**

```
docker: version 20.10.8, build 396a81a
[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_VERSION}/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/${CRICCTL_VERSION}/crictl-${CRICCTL_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/${RELEASE_VERSION}/cmd/kr
el/templates/latest/kubelet/kubelet.service" | sed "s:/usr/bin:${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/${RELEASE_VERSION}/cmd/kr
el/templates/latest/kubeadm/10-kubeadm.conf" | sed "s:/usr/bin:${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 to **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 xsbsq1.6ro11sawnvtbsvu \
  --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
```

```
  cluster role, added. 0/0 total
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 :kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

```
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: 0/1 nodes are tainted by node-role.kubernetes.io/control-plane
```

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

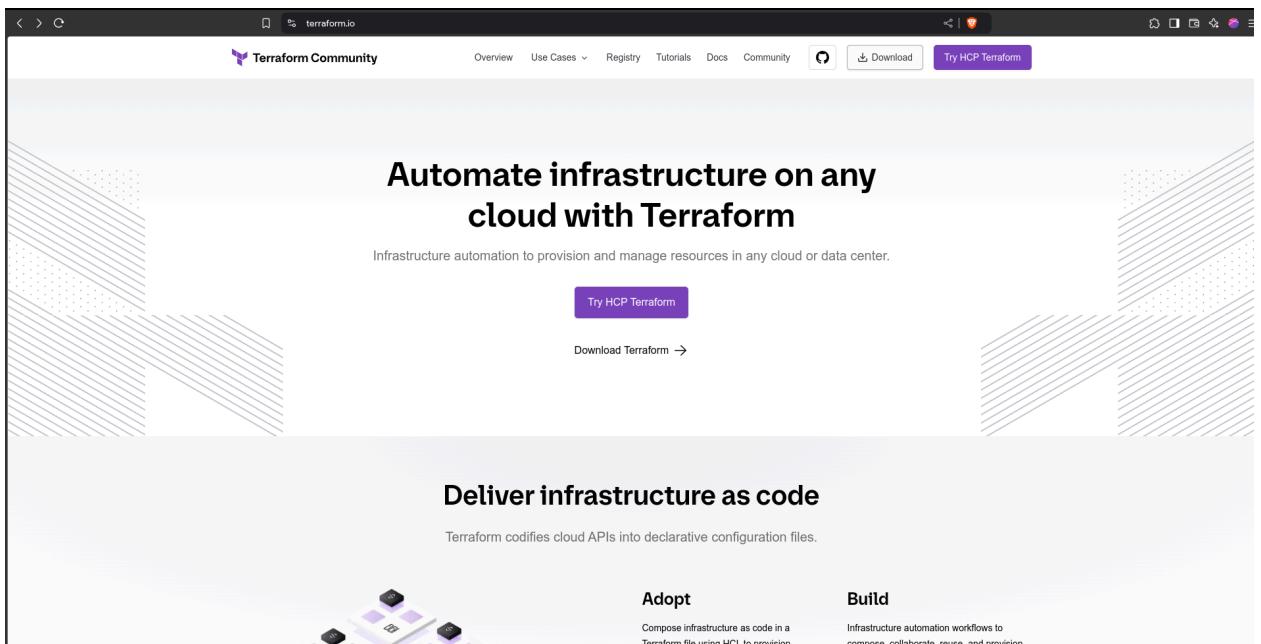
Conclusion: We began with installation and setup of docker and kubernetes. The kubernetes api server had some issues but were resolved after restarting kubelet service. The pods created were not running because nodes were tainted so we had to make them untainted. After solving all errors the nginx server pods were deployed successfully and can be accessed with the forwarded port. The nginx server can be accessed on different terminals or by making port forward process as background process by appending & after the command.

EXPERIMENT NO. 5

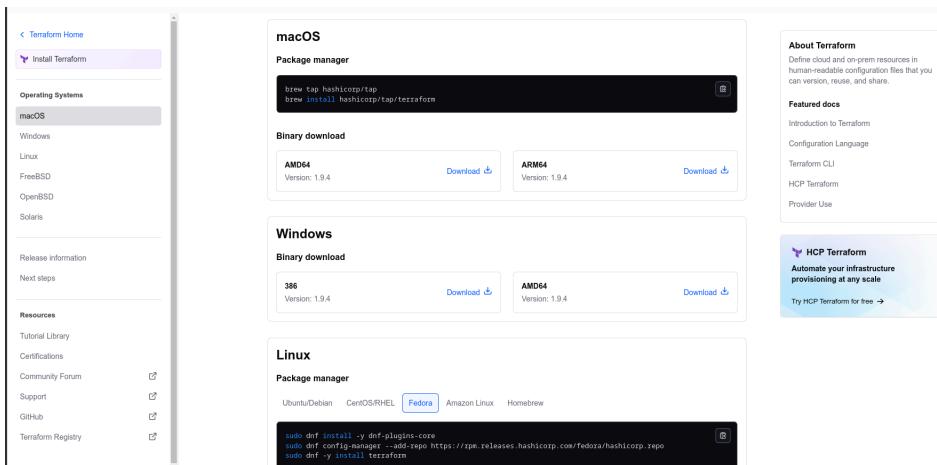
Aim: To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.

Steps:

1. Go to the official website of **Terraform** (<https://www.terraform.io/>). Click on Downloads located at top right corner of the page



2. Follow procedure according to your operating system (I'm installing on Linux)



3. Update your repository cache

```
quantum@machine ➔ sudo dnf5 update --refresh  
Updating and loading repositories:
```

4. Install Plugins Core

```
quantum@machine ~ sudo dnf install -y dnf-plugins-core
```

```
Fedora 40 - x86_64 - Updates
Package dnf-plugins-core-4.8.0-1.fc40.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

5. Add official package repository to your system

```
quantum@machine ~ ➜ sudo dnf config-manager --add-repo https://rpm.releases.hashicorp.com/fedora/hashicorp.repo
Adding repo from: https://rpm.releases.hashicorp.com/fedora/hashicorp.repo
```

6. Install it by typing **sudo dnf -y install terraform** in the terminal

✗ quantum@machine ~ sudo dnf5 install terraform

```
Repositories loaded.
Package Arch Version Repository Size
Installing:
terraform x86_64 1.9.4-1 hashicorp 84.9 MiB

Transaction Summary:
Installing: 1 packages

Total size of inbound packages is 27 MiB. Need to download 27 MiB.
After this operation 85 MiB will be used (install 85 MiB, remove 0 B).
Is this ok [y/N]: y
[1/1] terraform-0.1.9.4-1.x86_64
[1/1] Total 100% | 1.8 MiB/s | 26.7 MiB | 00m14s
[1/2] https://rpm.releases.hashicorp.com/gpg
[1/2] Total 100% | 1.8 MiB/s | 26.7 MiB | 00m14s
[1/2] https://rpm.releases.hashicorp.com/gpg
[1/2] Total 100% | 1.7 MiB/s | 3.9 KiB | 00m02s

[2/2] Total 100% | 1.8 MiB/s | 26.7 MiB | 00m14s
Importing PGP key 0xA621E701:
Userid : "HashiCorp Security (HashiCorp Package Signing) <security+packaging@hashicorp.com>"
Fingerprint: 798AEf65aE5C15428C8E2EAE14fC8CA621E701
From : https://rpm.releases.hashicorp.com/gpg
Is this ok [y/N]: y
The key was successfully imported.
[1/3] Verify package files
[2/3] Prepare transaction
[3/3] Installing terraform-0.1.9.4-1.x86_64
100% | 50.0 B/s | 1.0 B | 00m00s
100% | 1.0 B/s | 1.0 B | 00m01s
100% | 108.3 MiB/s | 84.9 MiB | 00m01s
```

7. Verify installation by checking version of Terraform installed

quantum@machine ~ terraform -v

Terraform v1.9.4
on linux_amd64

Prerequisite:

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

Steps:

1. Check docker installation of looking at it's help page

```
quantum@machine ~ docker -h
Flag shorthand -h has been deprecated, use --help

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

Management Commands:
  builder   Manage builds
  buildx*   Docker Buildx
  checkpoint Manage checkpoints
  compose*  Docker Compose
  container  Manage containers
  context    Manage contexts
  dev*      Docker Dev Environments
  extension* Manages Docker extensions
  feedback* Provide feedback, right in your terminal!
  image     Manage images
  init*     Creates Docker-related starter files for your project
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks
  plugin    Manage plugins
  sbom*     View the packaged-based Software Bill Of Materials (SBOM) for an image
  scan*     Docker Scan
  scout*    Docker Scout
  system    Manage Docker
  trust     Manage trust on Docker images
  volume   Manage volumes
```

```
quantum@machine ~ docker -v
Docker version 27.1.2, build d01f264
```

2. Create a new folder named “**Terraform Scripts**”, inside it create a new folder **docker** and create a new file named **docker.tf** with following contents inside it.

```
GNU nano 7.2                                            docker.tf
terraform {
    required_providers {
        docker = {
            source  = "kreuzwerker/docker"
            version = "2.21.0"
        }
    }
}

provider "docker" {
    host = "unix:///var/run/docker.sock"
}

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

# Create a container
resource "docker_container" "foo" {
    image = docker_image.ubuntu.image_id
    name  = "foo"
    command = [
        "/bin/bash",
        "-c",
        "while true; do sleep 3600; done"
    ]
}
```

3. To see list of providers for current configuration file use command **terraform providers**

```
quantum@machine ~/Downloads/advdevops/TerraformScripts/docker terraform providers
Providers required by configuration:
└── provider[registry.terraform.io/kreuzwerker/docker] 2.21.0
```

4. Execute command **terraform init** in the current directory

```
quantum@machine ~ ~/Downloads/advdevops/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.
```

5. Use command **terraform validate** to check for validation and syntax errors of config file

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/docker terraform validate
Success! The configuration is valid.
```

5. Execute command **terraform plan** to see the changes that will be made

```
quantum@machine:~/Downloads/advdevops/TerraformScripts/docker$ sudo terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

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

6. Check list of running docker containers using command **docker ps**

sudo docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

As we can see there are no active containers running

7. Execute **terraform apply** to apply configuration, which will automatically create and run the Ubuntu Linux container based on our configuration.

```
x quantum@machine:~/Downloads/advdevops/TerraformScripts/docker$ sudo terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:17c0145030df106e60e5d99149d69810db23b869ff0d3c9d23627a5a7bbb6b3ubuntu:latest]

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach           = false
  + bridge           = (known after apply)
  + command          = [
    + "/bin/bash",
    + "-c",
    + "while true; do sleep 3600; done",
  ]
  + 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             = "sha256:17c0145030df106e60e5d99149d69810db23b869ff0d3c9d23627a5a7bbb6b3"
  + init              = (known after apply)
  + ip_address        = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode          = (known after apply)
```

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

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

```
Enter a value: yes
```

```
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=d912cf0a2579f9b8c958d2d33426ed11e8251a553f85a0c08022d19ddf9eeecd]

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

8. After executing **terraform apply**

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/docker ➤ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d912cf0a2579 17c0145030df "/bin/bash -c 'while..." 2 minutes ago Up About a minute

```

After execution, a new docker image and container will be created

9. Execute **terraform destroy** to delete the configuration, which will automatically delete the Container.

```
# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
  - id          = "sha256:35a88802559dd2077e584394471ddaa1a2c5bfd16893b829ea57619301eb3908ubuntu:latest"
" -> null
  - image_id    = "sha256:35a88802559dd2077e584394471ddaa1a2c5bfd16893b829ea57619301eb3908" -> null
  - latest      = "sha256:35a88802559dd2077e584394471ddaa1a2c5bfd16893b829ea57619301eb3908" -> null
  - name        = "ubuntu:latest" -> null
  - repo_digest = "ubuntu@sha256:2e863c44b718727c860746568e1d54af13b2fa71b160f5cd9058fc436217b30" -> 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=8329fd298f0ccf122a391d74108a2809c40b5fbe383ffc31ee4fb6aeb6f0e3c7]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:35a88802559dd2077e584394471ddaa1a2c5bfd16893b829ea57619301eb3908ubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

10. Check whether docker image and container is removed or not

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/docker ➤ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
alpine          latest   05455a08881e  6 months ago  7.38MB
busybox         latest   3f57d9401f8d  7 months ago  4.26MB
sxcurity/gau   latest   5a5fc3cf7aa4  9 months ago  23.3MB
hello-world     latest   d2c94e258dcb  15 months ago 13.3kB
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/docker ➤ sudo docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
```

As we can see both containers and images are removed

EXPERIMENT NO. 6 (B)

AIM: Creating S3 Bucket using terraform

Prerequisite:

- 1) Any text editor to write and save scripts
- 2) Must have an AWS Access Key ID and Secret Access Key

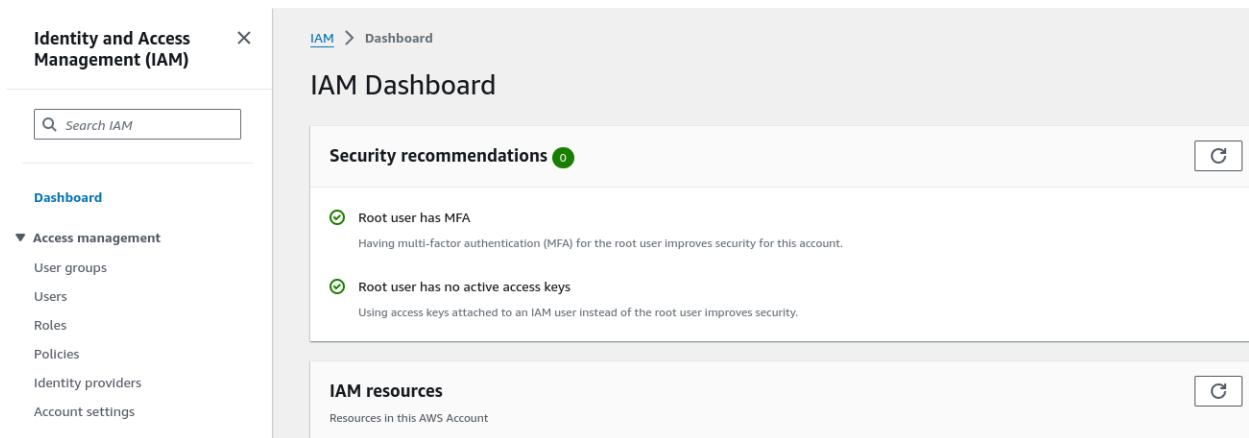
Step 1: Write a Terraform Script in Atom for creating S3 Bucket on Amazon AWS

```
resource "aws_s3_bucket" "aloky" {  
  bucket = "my-bj-terraform-test-bucket-aloky"  
  
  tags = {  
    Name      = "My bucket"  
    Environment = "Dev"  
  }  
}
```

Ensure your bucket name is globally unique

Step 2: Get secret key and access key from AWS account

1. Go to IAM dashboard



The screenshot shows the AWS IAM Dashboard. On the left, a sidebar menu includes 'Identity and Access Management (IAM)', 'Dashboard', and 'Access management' (which is expanded, showing 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', and 'Account settings'). The main content area is titled 'IAM Dashboard' and contains two sections: 'Security recommendations' and 'IAM resources'. The 'Security recommendations' section lists two items: 'Root user has MFA' (status: green checkmark) and 'Root user has no active access keys' (status: green checkmark). The 'IAM resources' section is titled 'Resources in this AWS Account' and includes a 'Create' button.

2. Go to Users and select an existing user if it exists or create a new one. I'm selecting existing one as i had already created before

Users (1) Info											
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.											
<input style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;" type="button" value="C"/> <input style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;" type="button" value="Delete"/> <input style="background-color: #ff7f0e; color: white; border: 1px solid #ff7f0e; padding: 2px 5px;" type="button" value="Create user"/>											
User name	▲	Path	▼	Group:	▼	Last activity	▼	MFA	▼	Password age	▼
aloky		/		0		-		-		Console last sign-in	▼
						-		-		Access key ID	▼
						-		-		Active key age	▼

3. On the user dashboard, click on **create access key**

aloky Info	<input type="button" value="Delete"/>
Summary	
ARN arn:aws:iam::058264551418:user/aloky	Console access Disabled
Created August 12, 2024, 20:39 (UTC+05:30)	Last console sign-in -

4. Select **Use case as Local Code** then Create access key

Use case

Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

Other
Your use case is not listed here.

5. After all above steps, access key is generated

is key

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
<input type="checkbox"/> AKIAQ3EGWP75IK42VSN3	<input type="checkbox"/> VekODARrlaVDCPbsTqLbHdOIQ+kc5HGfSeC9Cu7 Hide

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

6. Ensure newly created user has necessary permissions to edit and create S3 buckets to do so, click on **Add permissions**

Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.

[C](#) [Remove](#) [Add permissions ▾](#)

7. Select permission option as **Attach policies directly** and select **AmazonS3FullAccess** as permission policy from the list

\

Add permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1228)

Filter by Type [X](#) [▼](#) 12 matches [C](#) [1](#) [>](#) [@](#)

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonDMSRedshiftS3Role	AWS managed	0
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	0
<input type="checkbox"/> AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	0

Step 3: Create a new provider.tf file and write the following contents into it.

```
provider "aws" {  
  access_key = "AKIAQ3EGWP75IK42VSN3"  
  secret_key = "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"  
  region     = "ap-south-1"  
}
```

Save both the files in same directory TerraformScripts/S3

Step 4: Open terminal and go to TerraformScripts/S3 directory where our .tf files are stored

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/S3 ls  
provider.tf s3.tf
```

Step 5: Execute **terraform init** to initialize the resources

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/S3 terraform init  
Initializing the backend...  
Initializing provider plugins...  
- Finding latest version of hashicorp/aws...  
- Installing hashicorp/aws v5.63.1...  
- Installed hashicorp/aws v5.63.1 (signed by HashiCorp)  
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 6: Execute Terraform plan to see the available resources

```
quantum@machine ~ ~/Downloads/advdevops/TerraformScripts/S3 ➤ sudo terraform plan
[sudo] password for quantum:

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

Terraform will perform the following actions:

# aws_s3_bucket.kajal will be created
+ resource "aws_s3_bucket" "kajal" {
  + acceleration_status      = (known after apply)
  + acl                      = "public-read"
  + arn                      = (known after apply)
  + bucket                   = "my-bj-terraform-test-bucket"
  + bucket_domain_name       = (known after apply)
  + bucket_prefix             = (known after apply)
  + bucketRegionalDomainName = (known after apply)
  + force_destroy             = false
  + hostedZoneId             = (known after apply)
  + id                       = (known after apply)
  + objectLockEnabled         = (known after apply)
  + policy                   = (known after apply)
  + region                   = (known after apply)
  + requestPayer              = (known after apply)
  + tags                      = {
      + "Environment" = "Dev"
      + "Name"        = "My bucket"
    }
  + tags_all                 = {
      + "Environment" = "Dev"
      + "Name"        = "My bucket"
    }
  + website_domain           = (known after apply)
  + website_endpoint          = (known after apply)

  + cors_rule (known after apply)
  + grant (known after apply)
  + lifecycle_rule (known after apply)
}
```

```
+ logging (known after apply)
+ objectLockConfiguration (known after apply)
+ replicationConfiguration (known after apply)
+ serverSideEncryptionConfiguration (known after apply)
+ versioning (known after apply)
+ website (known after apply)
}

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

Warning: Argument is deprecated
with aws_s3_bucket.kajal,
on s3.tf line 3, in resource "aws_s3_bucket" "kajal":
  3:   acl    = "public-read"

Use the aws_s3_bucket_acl resource instead
(and one more similar warning elsewhere)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
```

Step 7: Execute **terraform apply** to apply the configuration, which will automatically create an S3 bucket based on our configuration.

```
quantum@machine ~/Downloads/advdevops/TerraformScripts/S3 ➤ sudo terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.aloky will be created
+ resource "aws_s3_bucket" "aloky" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "my-bj-terraform-test-bucket-aloky"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy             = false
    + hosted_zone_id           = (known after apply)
    + id                       = (known after apply)
    + object_lock_enabled       = (known after apply)
    + policy                   = (known after apply)
    + region                   = (known after apply)
    + request_payer             = (known after apply)
    + tags                     = {
        + "Environment" = "Dev"
        + "Name"        = "My bucket"
    }
    + tags_all                 = {
        + "Environment" = "Dev"
        + "Name"        = "My bucket"
    }
    + website_domain           = (known after apply)
    + website_endpoint          = (known after apply)

    + cors_rule (known after apply)
    + grant (known after apply)
}
```

```

+ grant (known after apply)

+ lifecycle_rule (known after apply)

+ logging (known after apply)

+ object_lock_configuration (known after apply)

+ replication_configuration (known after apply)

+ server_side_encryption_configuration (known after apply)

+ versioning (known after apply)

+ website (known after apply)
}


```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```

aws_s3_bucket.aloky: Creating...
aws_s3_bucket.aloky: Creation complete after 4s [id=my-bj-terraform-test-bucket-aloky]

```

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

Step 8: Go to **S3 Buckets** dashboard to see newly created S3 bucket using terraform

General purpose buckets		Directory buckets
General purpose buckets (4)	Info	All AWS Regions
Buckets are containers for data stored in S3.		
<input type="text" value="Find buckets by name"/>		Create bucket
Name	AWS Region	IAM Access Analyzer
codepipeline-ap-south-1-13487781303	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1
elasticbeanstalk-ap-south-1-058264551418	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1
host-web-alok	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1
my-bj-terraform-test-bucket-aloky	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1

Amazon S3 > Buckets > my-bj-terraform-test-bucket-aloky

my-bj-terraform-test-bucket-aloky Info

Objects Properties Permissions Metrics Management Access Points

Objects (0) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For more information, see [Amazon S3 Object](#).

Find objects by prefix

<input type="checkbox"/>	Name	<small>▲</small>	<small>Type</small>	<small>▼</small>	<small>Last modified</small>
No objects					

EXPERIMENT NO. 7

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

Prerequisites:

- Jenkins installed (Java JDK required)
 - Docker Installed (for SonarQube)
- docker engine service should be running (sudo systemctl start docker)

1. Install SonarQube Docker image

```
sudo docker pull sonarqube
```

```
quantum@machine ~ ➔ sudo docker pull sonarqube
[sudo] password for quantum:
Using default tag: latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Downloading [=====]
bd819c9b5ead: Download complete
4f4fb700ef54: Downloading [=====] 32B/32B
```

Wait for images to be pulled

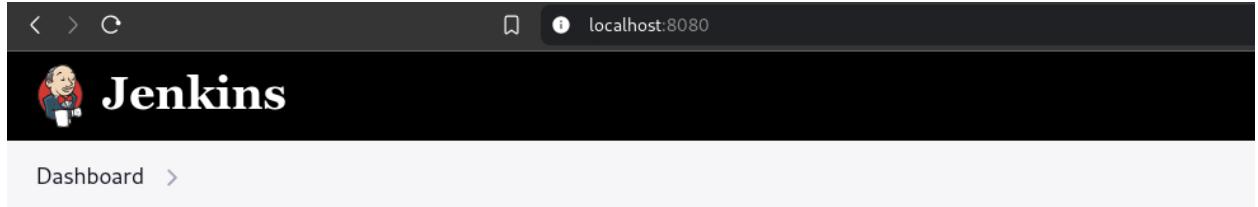
```
quantum@machine ~ ➔ sudo docker pull sonarqube
[sudo] password for quantum:
Using default tag: latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest
```

```
quantum@machine ~ ➔ sudo do
[sudo] password for quantum:
Using default tag: latest
latest: Pulling from library/so
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Downloading [=====
bd819c9b5ead: Download complete
4f4fb700ef54: Downloading [=====
```

Wait for images to be pulled

SonarQube installed successfully

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



Our jenkins is running on port 8080

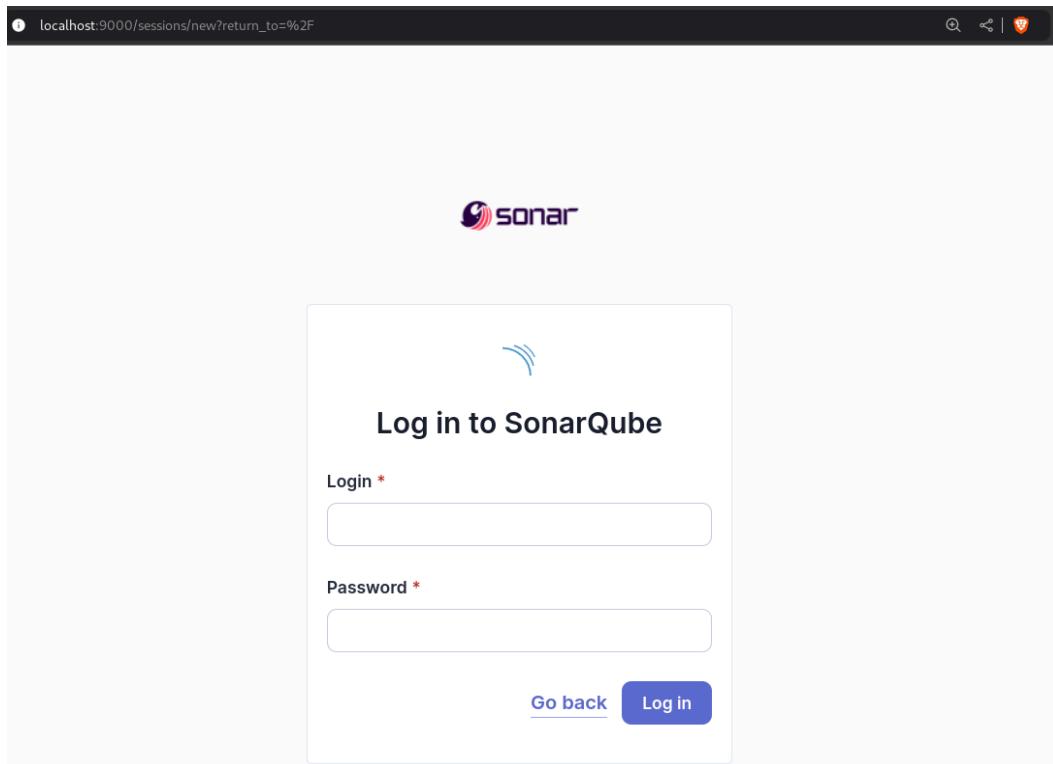
3. Run SonarQube in a Docker container using this command

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

Warning: run below command only once

```
quantum@machine ~ $ sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
[sudo] password for quantum:  
f9c595308e368210e19e099256a47ec1fe44affdc778eb58ccb53174163ce057
```

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



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

localhost:9000/account/reset_password

Update your password

⚠ This account should not use the default password.

Enter a new password

All fields marked with ***** are required

Old Password *

New Password *

Confirm Password *

Update

After logging, we have to change default password

6. Create a manual project in SonarQube with the name sonarqube
Click on **create a local project** on dashboard

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

⬇ Import from Azure DevOps	Setup	⬇ Import from Bitbucket Cloud	Setup
⬇ Import from GitHub	Setup	⬇ Import from GitLab	Setup

Are you just testing or have an advanced use-case? Create a local project.

⬇ Create a local project
--

1 of 2

Create a local project

Project display name *

 ✓

Project key *

 ✓

Main branch name *

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

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

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

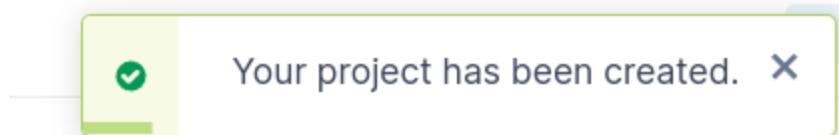
Use the global setting

[Previous version](#)

Any code that has changed since the previous version is considered new code.

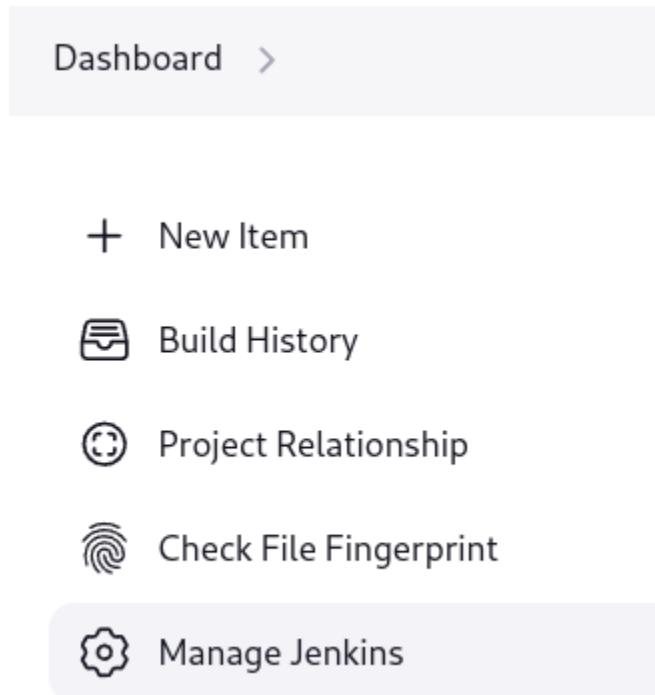
Recommended for projects following regular versions or releases.

We can either use new custom settings for the project or use global settings
Here I'm using global setting

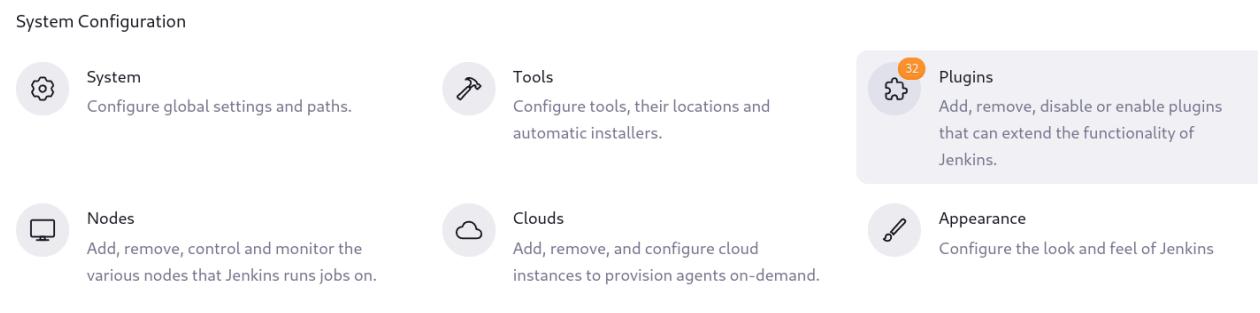


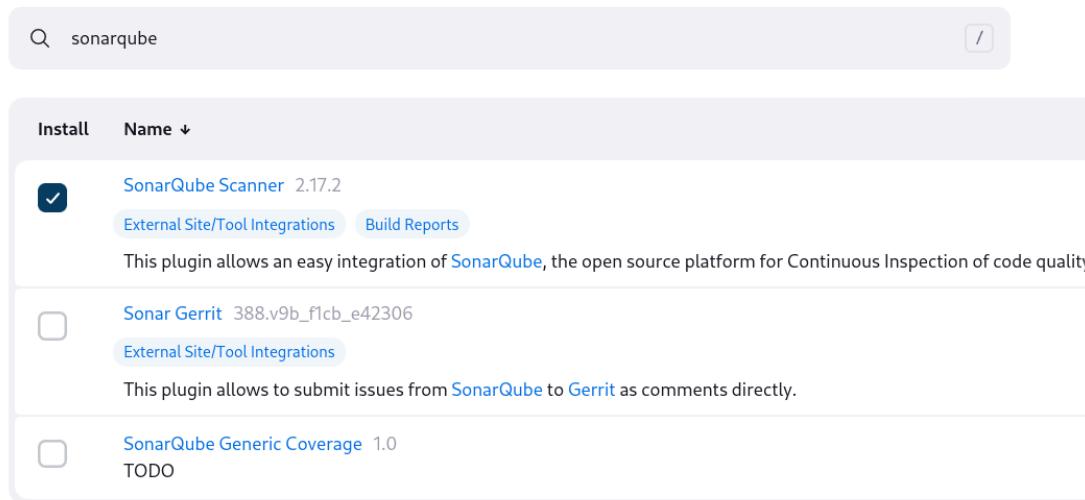
On successful creation of project, we get a popup for the same

7. After setting project in sonarqube, go to **Jenkins Dashboard**



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





The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text "sonarqube". Below the search bar, there are two tabs: "Install" and "Name ↓", with "Install" currently selected. A list of plugins is displayed, with the first plugin, "SonarQube Scanner 2.17.2", having a checked checkbox next to it. This plugin is categorized under "External Site/Tool Integrations" and "Build Reports". A description below the plugin states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." The other two plugins listed are "Sonar Gerrit" and "SonarQube Generic Coverage", both with unchecked checkboxes and descriptions indicating they are for "External Site/Tool Integrations".

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

 Installing

Loading plugin extensions

 Pending

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Our installation is in progress wait for it to download and install packages

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

 Success

Loading plugin extensions

 Success

Plugin installed successfully

8. Under Jenkins dashboard 'Configure System', look for SonarQube Servers and enter the details.

System Configuration



System

Configure global settings and paths.



Tools

Configure tools, their locations and automatic installers.



Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

SonarQube servers

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

Environment variables

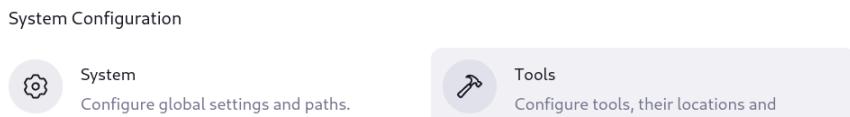
SonarQube installations

List of SonarQube installations

Name	<input type="text" value="sonarqube"/>	X
Server URL	Default is http://localhost:9000	
	<input type="text" value="http://localhost:9000"/>	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled.	
	<input type="text" value="- none -"/> + Add ▾	

[Advanced ▾](#)

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



SonarScanner for MSBuild installations

[Add SonarScanner for MSBuild](#)

SonarQube Scanner installations

[Add SonarQube Scanner](#)

Click on Add SonarQube Scanner

SonarQube Scanner installations



Add SonarQube Scanner

SonarQube Scanner

Name
sonarqube installer

Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Select Latest version and save configuration

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

Enter an item name

SonarQube

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

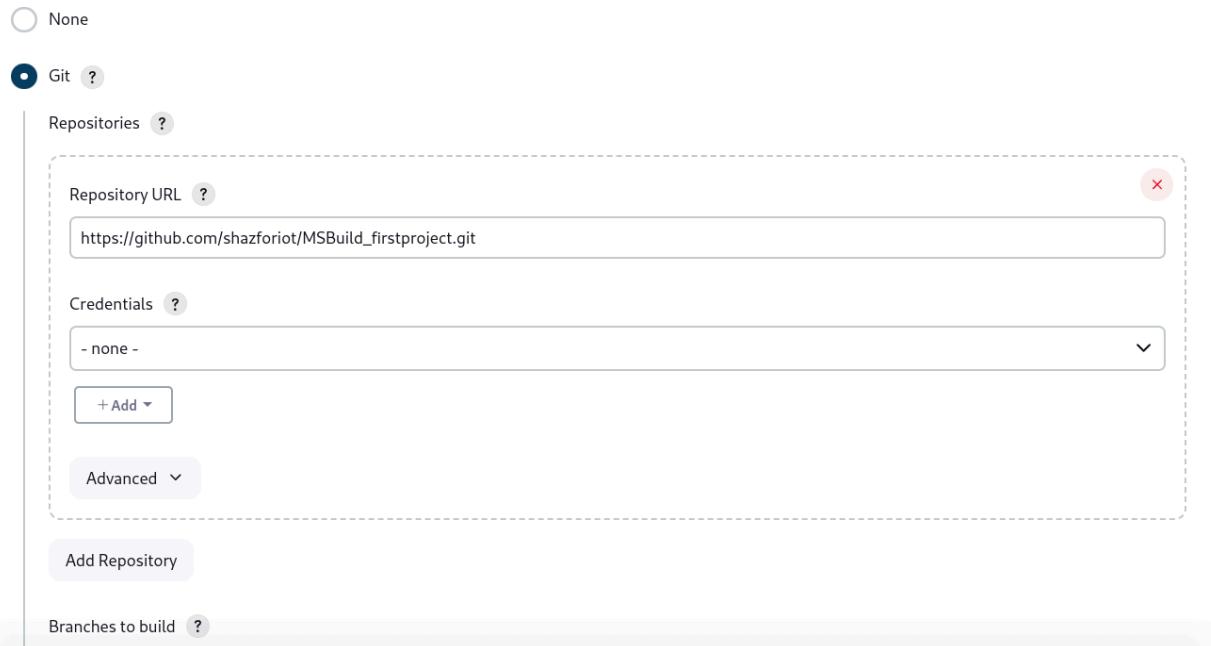
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

11. Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git It is a sample hello-world project with no vulnerabilities and issues, just to test the integration

Source Code Management

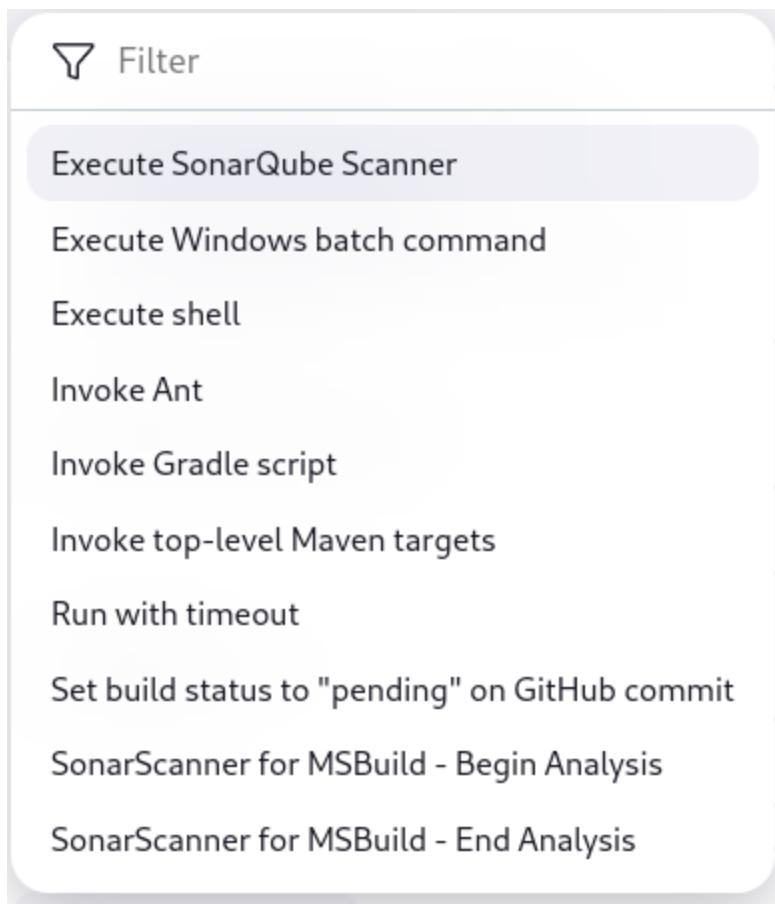


12. Under Build-> Execute SonarQube Scanner

Build Steps

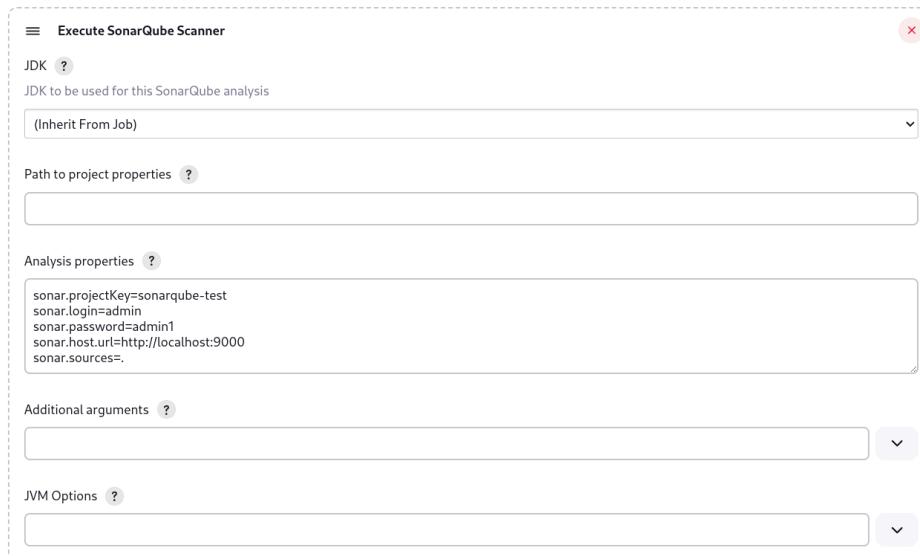
Add build step ▾

Click on add build steps

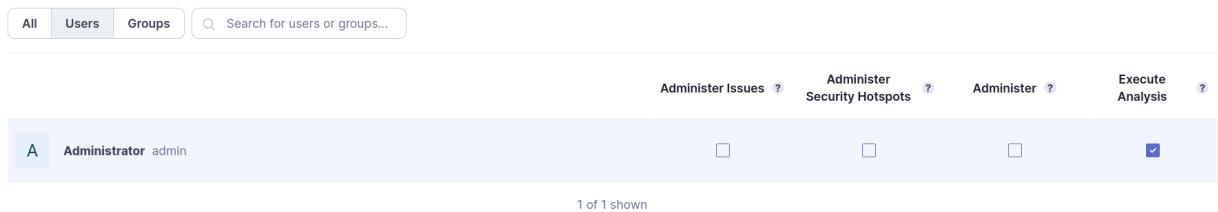


Then click on **Execute SonarQube Scanner**

13. Mention the SonarQube Project Key, Login, Password, Source path and Host URL in Analysis properties

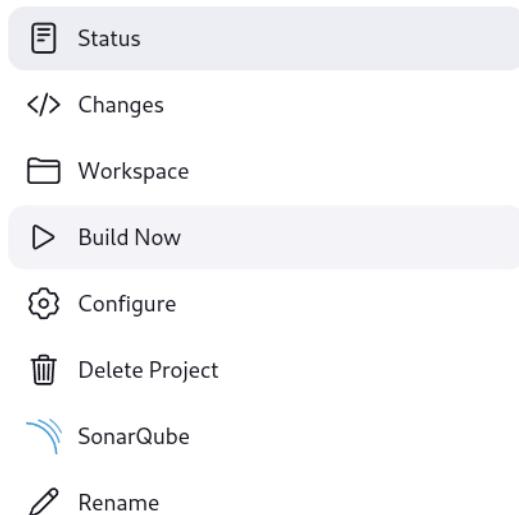


14. Go to http://localhost:9000/project_roles?id=<project_key> and allow Execute Permissions to the Admin user.



The screenshot shows the 'Project Roles' page in SonarQube. At the top, there are tabs for 'All', 'Users', and 'Groups', with a search bar. Below the tabs, there are four permission categories: 'Administer Issues', 'Administer Security Hotspots', 'Administer', and 'Execute Analysis'. Under the 'Administrator' user, the 'Execute Analysis' checkbox is checked. At the bottom, it says '1 of 1 shown'.

15. Run The Build.



16. Check the console output



The screenshot shows the Jenkins console output page. The title is 'Console Output' with a checkmark icon. There are download, copy, and plain text options at the top right. The log content is as follows:

```

Started by user unknown or anonymous
Running as SYSTEM
Building on the built-in node in workspace /home/quantum/.jenkins/workspace/SonarQube
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/quantum/.jenkins/workspace/SonarQube/.git # timeout=10
Fetching changes from the remote Git repository
> git 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 --version # timeout=10
> git --version # 'git version 2.46.0'
> git fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)

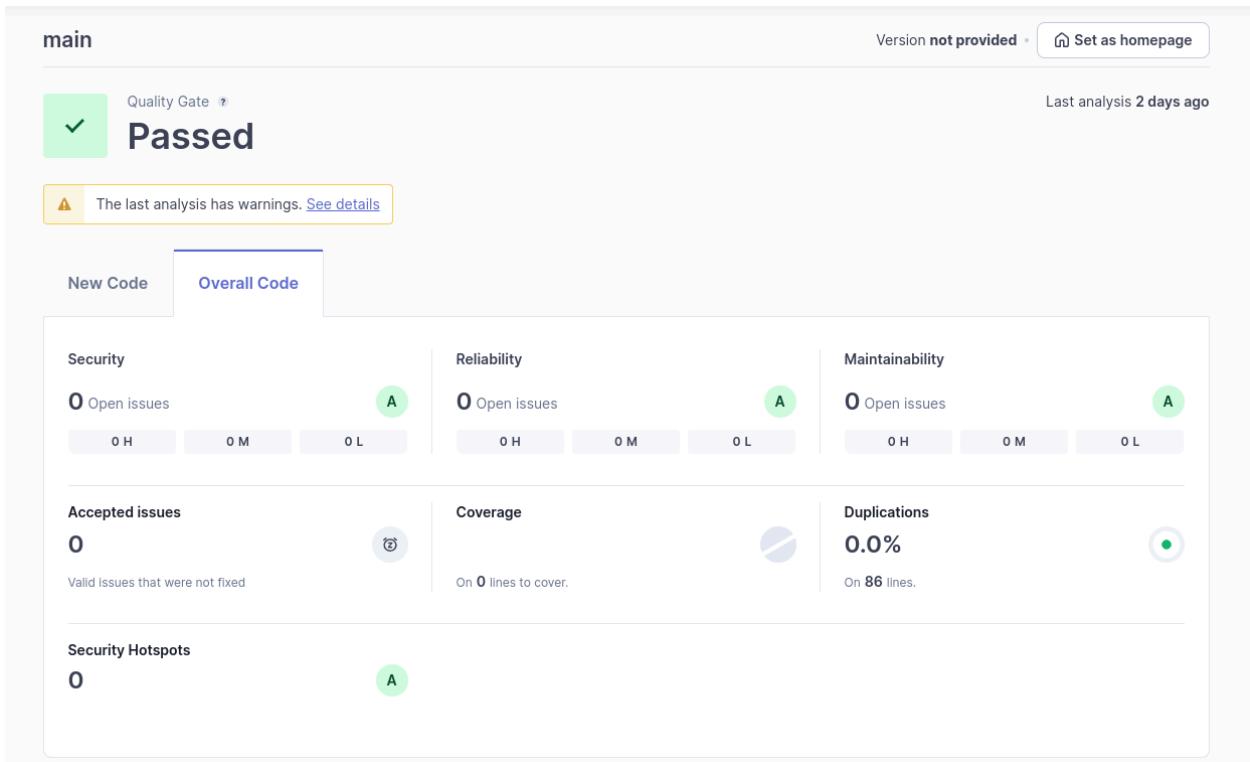
```

```

10:30:32.234 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:30:32.234 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:30:32.235 INFO More about the report processing at http://localhost:9000/api/ce/task?id=96795973-9667-4456-a15c-3311dbc9d067
10:30:32.244 INFO Analysis total time: 9.533 s
10:30:32.245 INFO SonarScanner Engine completed successfully
10:30:32.300 INFO EXECUTION SUCCESS
10:30:32.301 INFO Total time: 14.090s
Finished: SUCCESS

```

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



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

Conclusion: We began the experiment with installation of SonarQube Docker Image followed by setting up a new project in SonarQube. Then we installed the SonarQube scanner plugin and then created a new freestyle project in Jenkins with a Git repository for code analysis. Then we configured the Jenkins with appropriate settings to work with sonarqube. Gave permissions to Jenkins to perform code analysis. It is essential to provide correct properties in **Analysis Properties** for Jenkins to run correctly. The Jenkins project ran successfully with all tests passed in SonarQube.

EXPERIMENT NO. 8

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

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.



+

New Item

Build History

Project Relationship

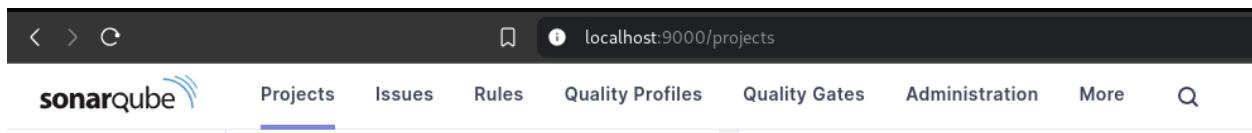
Check File Fingerprint

Manage Jenkins

2. Run SonarQube in a Docker container using this command

```
quantum@machine ~ ➔ sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

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



4. Login to SonarQube using your username and password

5. Create a manual project in SonarQube with the name sonarqube-test2

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

SonarQube-exp8

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

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



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

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/"
```

```
 }  
 }  
 }
```

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

8. Install sonar-scanner

Now we need to install **sonar-scanner** binary to perform code analysis To do so, go to

<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0.4584-linux-x64.zip> for linux OS
(<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0.4584-windows-x64.zip> for windows OS)

You will be prompted to download the zip file, download it and extract it and copy the path (absolute path) to <ROOT_DIRECTORY>/bin/sonar-scanner in my case it is /opt/sonar-scanner/bin/sonar-scanner (For linux)

For windows path would be

C:\\\\Users\\\\<USER_NAME>\\\\Downloads\\\\sonar-scanner-cli-6.2.0.4584-windows-x64\\\\bin\\\\sonar-scanner

Now we need to update the required details in the pipeline script as :

```
pipeline {  
    agent any  
    stages {  
        stage('Cloning the GitHub Repo') {  
            steps {  
                git 'https://github.com/shazforiot/GOL.git'  
            }  
        }  
        stage('SonarQube analysis') {  
            steps {  
                withSonarQubeEnv('sonarqube') {configuration  
                    sh """  
                        /opt/sonar-scanner/bin/sonar-scanner \ //replace with your actual path  
to sonar-scanner  
                        -D sonar.projectKey=sonarqube-test2 \  
                        -D sonar.sources=. \  
                        -D sonar.exclusions=vendor/**,resources/**,**/*.java \  
                }  
            }  
        }  
    }  
}
```

```

-D sonar.host.url=http://127.0.0.1:9000 \
-D sonar.login=admin \
-D sonar.password=admin1
    """
}
}
}
}

```

Pipeline

Definition

Pipeline script

```

Script ?
1 pipeline {
2     agent any
3
4     stages {
5         stage('Cloning the GitHub Repo') {
6             steps {
7                 git 'https://github.com/shazforiot/GOL.git'
8             }
9         }
10        stage('SonarQube analysis') {
11            steps {
12                withSonarQubeEnv('sonarqube') { // Ensure 'sonarqube' matches your Jenkins configuration
13                    sh """
14                     /opt/sonar-scanner/bin/sonar-scanner \
15                     -D sonar.projectKey=sonarqube-test2 \
16                     -D sonar.sources=. \
17                     -D sonar.exclusions=vendor/**,resources/**,*/**/*.java \
18                     -D sonar.host.url=http://127.0.0.1:9000 \
19                 """
20                }
21            }
22        }
23    }
24}

```

 Use Groovy Sandbox ?[Pipeline Syntax](#)[Save](#)[Apply](#)

9. Run the build

Dashboard > SonarQube-exp8 >

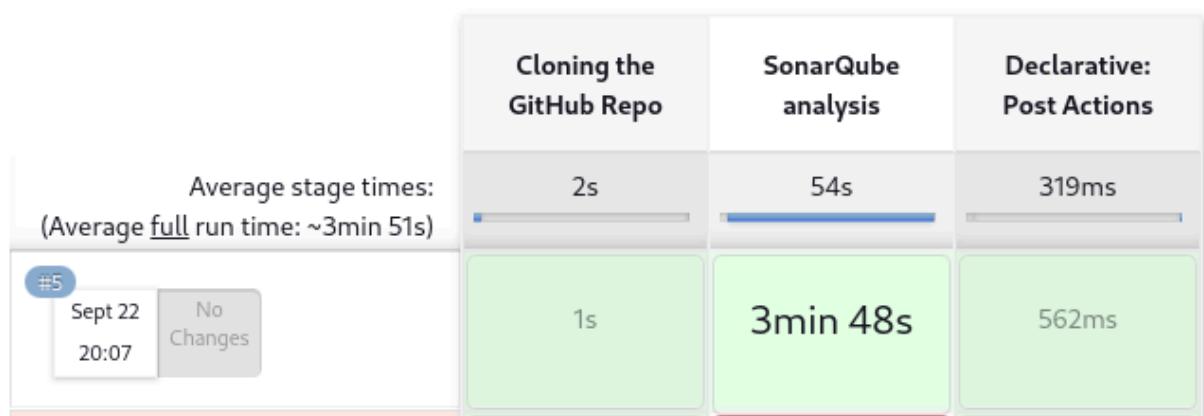
[Status](#)[Changes](#)[Build Now](#)[Configure](#)[Delete Pipeline](#)[Full Stage View](#)[Stages](#)[Rename](#)[Pipeline Syntax](#)

Check the status of Build

✅ SonarQube-exp8

SonarQube analysis for EXP 8

Stage View



As we can see the SonarQube analysis is completed

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

✅ Console Output

Skipping 4,226 KB.. [Full Log](#)

```

20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 296. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 17. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 298. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 300. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 225. Keep only the first 100 references.

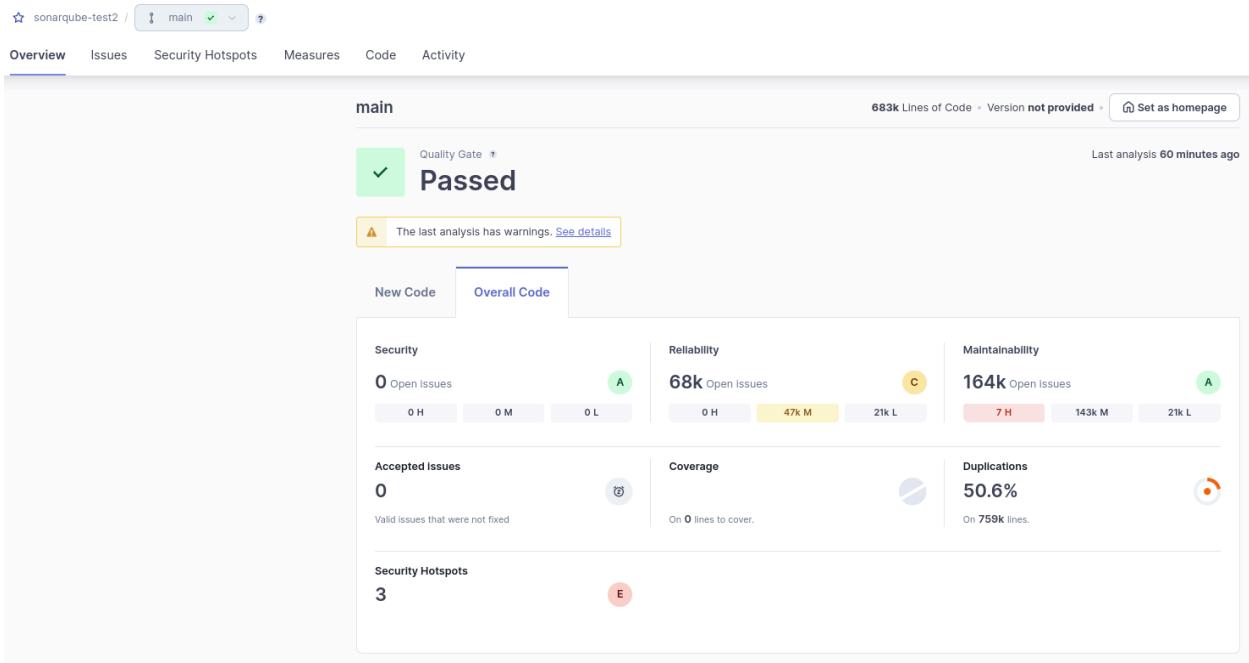
```

```

20:11:31.504 INFO CPD Executor CPD calculation finished (done) | time=104775ms
20:11:31.748 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:11:33.306 INFO Analysis report generated in 1507ms, dir size=127.2 MB
20:11:40.819 INFO Analysis report compressed in 7511ms, zip size=29.6 MB
20:11:42.147 INFO Analysis report uploaded in 1322ms
20:11:42.154 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test2
20:11:42.154 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:11:42.154 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=670aec3c-6c20-460b-ad52-ec6ce041fb1f
20:11:46.802 INFO Analysis total time: 3:42.216 s
20:11:46.835 INFO SonarScanner Engine completed successfully
20:11:48.173 INFO EXECUTION SUCCESS
20:11:48.296 INFO Total time: 3:47.113s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

11. After that, check the project in SonarQube.



Under different tabs, check all different issues with the code

12. Code Problems-

sonarqube-test2 / main

Overview Issues Security Hotspots Measures Code Activity Project Set

Filters

Issues in new code

Clean Code Attribute

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

Add to selection **Ctrl + click**

Software Quality

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

Severity

- High: 0
- Medium: 176k
- Low: 21k

Bulk Change

Select issues | Navigate to issue | 196,662 Issues | 3075d effort

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

- Insert a <!DOCTYPE> declaration to before this <html> tag.** Consistency
Reliability
- Remove this deprecated "width" attribute.** Consistency
Maintainability
- Remove this deprecated "align" attribute.** Consistency
Maintainability
- Remove this deprecated "align" attribute.** Consistency
Maintainability
- Remove this deprecated "size" attribute.** Consistency
html5 obsolete

L1 = 5min effort + 4 years ago - Bug - Major

L9 = 5min effort + 4 years ago - Code Smell - Major

L11 = 5min effort + 4 years ago - Code Smell - Major

L12 = 5min effort + 4 years ago - Code Smell - Major

L13 = 5min effort + 4 years ago - Code Smell - Major

Code Smells

sonarqube-test2 / main

Overview Issues Security Hotspots Measures Code Activity Project Set

Responsibility

Software Quality

- Security: 0
- Reliability: 21k
- Maintainability: 164k

Severity

- High: 7
- Medium: 143k
- Low: 21k

Type

- Bug: 47k
- Vulnerability: 0
- Code Smell: 164k

Add to selection **Ctrl + click**

Scope

- Main code: 164k
- Test code: 0

Status

Bulk Change

Select issues | Navigate to issue | 164,034 Issues | 1708d effort

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image.** Intentionality
Maintainability
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** Intentionality
Maintainability
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** Intentionality
Maintainability
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** Intentionality
Maintainability
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** Intentionality
Maintainability

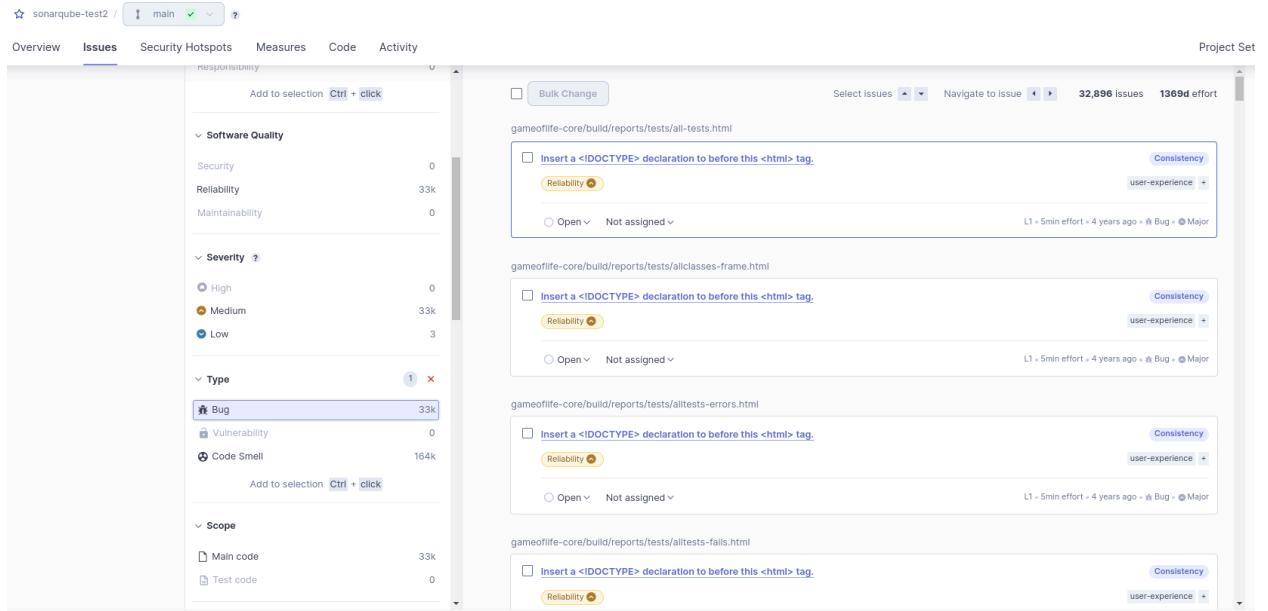
L1 = 5min effort + 4 years ago - Code Smell - Major

L12 = 5min effort + 4 years ago - Code Smell - Major

L13 = 5min effort + 4 years ago - Code Smell - Major

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

Bugs



The screenshot shows the SonarQube Issues page for a project named 'sonarqube-test2'. The 'Issues' tab is selected. The left sidebar contains filters for Responsibility, Software Quality, Severity, Type, and Scope. The main area lists issues grouped by file. Each group shows a summary of the issue type (e.g., Reliability, Consistency, user-experience) and a list of specific issues with their details: title, severity, assignee, and creation date. The 'gameoflife-core/build/reports/tests/all-tests.html' group has three issues: 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago), 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago), and 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago). The 'gameoflife-core/build/reports/tests/allclasses-frame.html' group has one issue: 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago). The 'gameoflife-core/build/reports/tests/alltests-errors.html' group has one issue: 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago). The 'gameoflife-core/build/reports/tests/alltests-fails.html' group has one issue: 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Reliability, Medium, Not assigned, 4 years ago). The total count of issues is 32,896 with 1369d effort.

Conclusion: We began the experiment with creating a new project in SonarQube and setting up a new Pipeline in Jenkins with proper configuration of pipeline script. Then we installed Sonar Scanner CLI so that jenkins can do code analysis of Git Repository. We can also configure the pipeline to use the installed Sonar Scanner plugin instead of locally installed Binary of Sonar Scanner. The pipeline ran successfully with all tests passed in SonarQube

EXPERIMENT NO. 9

Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core,

Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

STEPS TO INSTALL AND SETUP NAGIOS ON EC-2 INSTANCE

1. Create an Amazon Linux EC-2 instance and select either existing key pair or create new

EC2 > ... > Launch an instance

Launch an instance Info

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

Name and tags Info

Name

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

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

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

Recents Quick Start



Amazon Linux

aws



macOS

Mac



Ubuntu

ubuntu®



Windows

Microsoft



Red Hat

Red Hat



SUSE LI

SUS

Q Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Instances (1) <small>Info</small>										
Last updated less than a minute ago C Connect Instance state Actions Launch instances										
Find Instance by attribute or tag (case-sensitive) Running										
<input type="checkbox"/>	Name o	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	
<input type="checkbox"/>	nagio-host	I-01ad97aa822f75271	Running	Q t2.micro	Q Initializing	View alarms +	us-east-1c	ec2-52-207-222-231.co...	52.207.222.231	

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

Inbound rules (6)											
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	...	Description		
<input type="checkbox"/>	-	sgr-0be5f29d87c499251	IPv4	All traffic	All	All	0.0.0.0/0	-			
<input type="checkbox"/>	-	sgr-07cf6af90f9a8e6f6	IPv4	HTTP	TCP	80	0.0.0.0/0	-			
<input type="checkbox"/>	-	sgr-0646b85877794b...	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-			
<input type="checkbox"/>	-	sgr-0677298fb04c24d	IPv4	HTTPS	TCP	443	0.0.0.0/0	-			
<input type="checkbox"/>	-	sgr-012f4a15142687da3	IPv4	SSH	TCP	22	0.0.0.0/0	-			
<input type="checkbox"/>	-	sgr-0b3a173e17123d...	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0	-			

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

Connect to instance Info

Connect to your instance i-01ad97aa822f75271 (nagio-host) using any of these options

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

Instance ID
 i-01ad97aa822f75271 (nagio-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ec-2-ubuntu.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "ec-2-ubuntu.pem"
4. Connect to your instance using its Public DNS:
 ec2-52-207-222-231.compute-1.amazonaws.com

Example:
 ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-52-207-222-231.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.

```
> ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-52-207-222-231.compute-1.amazonaws.com
The authenticity of host 'ec2-52-207-222-231.compute-1.amazonaws.com (52.207.222.231)' can't be established.
ED25519 key fingerprint is SHA256:/s4gfn/CH0BLXNdwi7aFEQPNXd+F3gfT+c8lsjldby8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-207-222-231.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
X11 forwarding request failed on channel 0
,
#_
~\_\####_          Amazon Linux 2023
~~ \_\#####\
~~  \###|
~~  \#/  --> https://aws.amazon.com/linux/amazon-linux-2023
~~  V~'-->
~~_/
~~_. / /
~~_ / /
~~_ / /
[ec2-user@ip-172-31-33-179 ~]$
```

Now, we need to install necessary packages that are required to run nagios properly

sudo yum install httpd php

```
Installed:
  apr-1.6.2-2.0.2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64          apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64          generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.6-2.1.amzn2023.0.x86_64           httpd-core-2.4.6-2.1.amzn2023.0.x86_64           httpd-filesystem-2.4.6-2.1.amzn2023.noarch          httpd-tools-2.4.6-2.1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.x86_64          libbrotli-1.0.19-4.amzn2023.0.x86_64           libxml2-2.9.10-1.amzn2023.0.x86_64              mailcap-2.1.9-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.x86_64         mod_lua-2.4.6-2.1.amzn2023.0.x86_64           nginx-fslisten-1.1.24.0-1.amzn2023.0.4.noarch      phpb8.3-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-cl-8.3.10-1.amzn2023.0.1.x86_64       php8.3-common-8.3.10-1.amzn2023.0.1.x86_64      phpb8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64          phpb8.3-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64  php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64       phpb8.3-process-8.3.10-1.amzn2023.0.1.x86_64      phpb8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64
  phpb8.3-xmp-8.3.10-1.amzn2023.0.1.x86_64    phpb8.3-xml-8.3.10-1.amzn2023.0.1.x86_64       phpb8.3-xml-8.3.10-1.amzn2023.0.1.x86_64          phpb8.3-zip-8.3.10-1.amzn2023.0.1.x86_64
```

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

```
Installed:
  annobin-docs-10.93-1_amzn2023.0.1.noarch           annobin-plugin-gcc-10.93-1_amzn2023.0.1.x86_64    cpp-11.4.1-2_amzn2023.0.2.x86_64      glibc-8.0-4.5_amzn2023.0.2.x86_64
  gcc-11.4.1-2_amzn2023.0.2.x86_64                  glibc-devel-2.34-52_amzn2023.0.11.x86_64      glibc-headers-x86-2.34-52_amzn2023.0.11.noarch  glibc22-2.27-2_amzn2023.0.3.x86_64
  kernel-headers-6.1-109-118_amzn2023.x86_64        libmpc-1.2.1-2_amzn2023.0.2.x86_64            libtalloc-ltdl-2.4.7-1_amzn2023.0.3.x86_64    libcrypt-4.4.35-7_amzn2023.x86_64
  make-1.4.3-5_amzn2023.0.2.x86_64
```

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

```
Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-devel-1.13.2-15.4.amzn2023.0.2.x86_64
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
glibgbz-devel-2.74.7-689.amzn2023.0.2.x86_64
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
graphite2-zdev-devel-1.3.14-7.amzn2023.0.2.x86_64
langpacks-conc-font-otf-3.0.0.amzn2023.0.4.noarch
libX11-1.6.3-3.amzn2023.0.4.x86_64
libX11-xcb-1.7.1-2.3.amzn2023.0.4.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libffi-devel-3.4-4.1.amzn2023.0.1.x86_64
libjbig2-turbo-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
libsepol-devel-3.4-3.amzn2023.0.3.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64
fonts-filesystem-1.2.0-12.amzn2023.0.2.noarch
gd-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
graphite2-zdev-devel-1.3.14-7.amzn2023.0.2.x86_64
libICE-1.0.10.amzn2023.0.2.x86_64
libX11-common-1.7.3-3.amzn2023.0.4.noarch
libXau-0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libpcbi-67.1-7.amzn2023.0.3.x86_64
libjbig2-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
lippng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
lippng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
lintiff-4.4.0-4.amzn2023.0.18.x86_64
liwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
lxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
libX11-1.6.3-3.amzn2023.0.4.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
```

4. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation. This is required as to give separate permissions for nagios

sudo adduser -m nagios

sudo passwd nagios

```
ip-172-31-33-179.ec2.internal ~ ec2-user ~ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

6. Create a new user group

sudo groupadd nagcmd

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

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
ip-172-31-33-179.ec2.internal ~ ec2-user ~ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads

9. Use wget to download the installation source zip files.

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

```
ip-172-31-33-179.ec2.internal ~ ~/downloads ~ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-04 13:40:59-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00:f03c:92ff:feff: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%[=====] 2065473/2065473
2024-10-04 13:41:00 (4.15 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
```

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

```
# ip-172-31-33-179.ec2.internal ~ > downloads > wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 13:41:25-- 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%[=====] 2753049 2024-10-04 13:41:26 (7.55 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

10. Use tar to unzip the downloaded archive and change to that directory.

```
tar zxvf nagios-4.5.5.tar.gz
```

```
# ip-172-31-33-179.ec2.internal ~ > 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
```

11. Navigate to the extracted folder and Run the configuration script with the same group name you previously created.

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

```
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

We got error, because ssl headers library is not installed

It can be installed using **sudo yum install openssl-devel**

```
# ip-172-31-33-179.ec2.internal ~ > downloads > nagios-4.5.5 > sudo yum install openssl-devel
Last metadata expiration check: 0:28:57 ago on Fri Oct 4 13:16:25 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
```

Now rerun

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

```
Creating sample config files in sample-config/ ...
```

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

General Options:

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

Web Interface Options:

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

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

12. Compile the source code.

make all

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

command directory.

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > nagios-4.5.5 sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
```

```
# ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > nagios-4.5.5 sudo make install-init  
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system  
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
```

```
# ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > nagios-4.5.5 > sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timerperiods.cfg /usr/local/nagios/etc/objects/timerperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg
```

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > nagios-4.5.5 sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

14. Edit the config file and change the email address so that we can receive timely alerts about the status of our system.

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

```
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.alok.yadav@ves.ac.in; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
#
```

And change email with your email

15. Configure the web interface. This is used to set up web server configuration of nagios dashboard.

sudo make install-webconf

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > 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 ***
```

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

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

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > 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
```

17. Restart Apache

sudo service httpd restart

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > nagios-4.5.5 > sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

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

cd ~/downloads**tar zxvf nagios-plugins-2.4.11.tar.gz**

```
ip-172-31-33-179.ec2.internal ec2-user ~ > downloads > 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
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
```

19. Compile and install plugins

cd nagios-plugins-2.4.11**./configure --with-nagios-user=nagios --with-nagios-group=nagios**

```
  checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating gl/Makefile
config.status: creating nagios-plugins.spec
config.status: creating tools/build_perl_modules
config.status: creating Makefile
config.status: creating tap/Makefile
config.status: creating lib/Makefile
config.status: creating plugins/Makefile
config.status: creating lib/tests/Makefile
config.status: creating plugins-root/Makefile
config.status: creating plugins-scripts/Makefile
config.status: creating plugins-scripts/utils.pm
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
```

make**sudo make install**

```
# ip-172-31-33-179.ec2.internal ~ > downloads > nagios-plugins-2.4.11 > sudo make install
Making install in gl
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make  install-recursive
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
if test yes = no; then \
  case 'linux-gnu' in \
    darwin[56]*) \
      need_charset_alias=true ; \
    darwin* | cygwin* | mingw* | pw32* | cegcc*) \
      need_charset_alias=false ; \
    *) \
      need_charset_alias=true ; \
  esac ; \
else \
  need_charset_alias=false ; \
fi ; \
if $need_charset_alias; then \
  /bin/sh ../build-aux/mkinstalldirs /usr/local/nagios/lib ; \
fi ; \
```

20. Start Nagios and Add Nagios to the list of system services

sudo chkconfig --add nagios

sudo chkconfig nagios on

```
# ip-172-31-33-179.ec2.internal ~ > downloads > nagios-plugins-2.4.11 > sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
# ip-172-31-33-179.ec2.internal ~ > downloads > nagios-plugins-2.4.11 > sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

Verify the sample configuration files

```
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
```

As we can see no errors were detected

sudo service nagios start

```
# ip-172-31-33-179.ec2.internal ~ > downloads > nagios-plugins-2.4.11 > sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

21. Check the status of Nagios

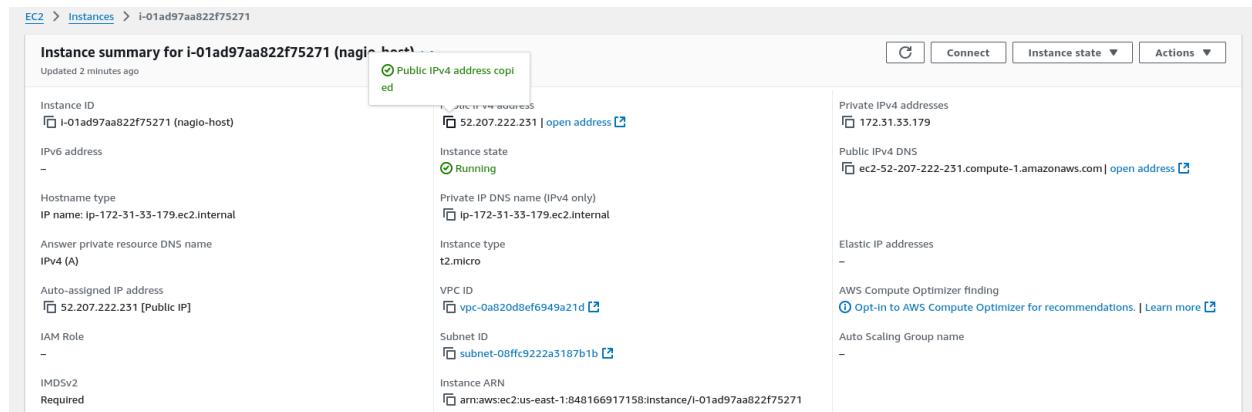
sudo systemctl status nagios

```
ip-172-31-33-179.ec2.internal ~ > downloads > 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 Fri 2024-10-04 14:05:21 UTC; 33s ago
    Docs: https://www.nagios.org/documentation
  Process: 67432 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 67433 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67434 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.6M
    CPU: 67ms
   CGroup: /system.slice/nagios.service
           └─67434 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67438 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67439 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67440 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67441 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67446 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: qh: core query handler registered
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: qh: echo service query handler registered
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: qh: help for the query handler registered
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: wproc: Successfully registered manager as @wproc with query handler
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: wproc: Registry request: name=Core Worker 67441;pid=67441
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: wproc: Registry request: name=Core Worker 67440;pid=67440
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: wproc: Registry request: name=Core Worker 67438;pid=67438
Oct 04 14:05:21 ip-172-31-33-179.ec2.internal nagios[67434]: wproc: Registry request: name=Core Worker 67439;pid=67439
Oct 04 14:05:22 ip-172-31-33-179.ec2.internal nagios[67434]: Successfully launched command file worker with pid 67446
```

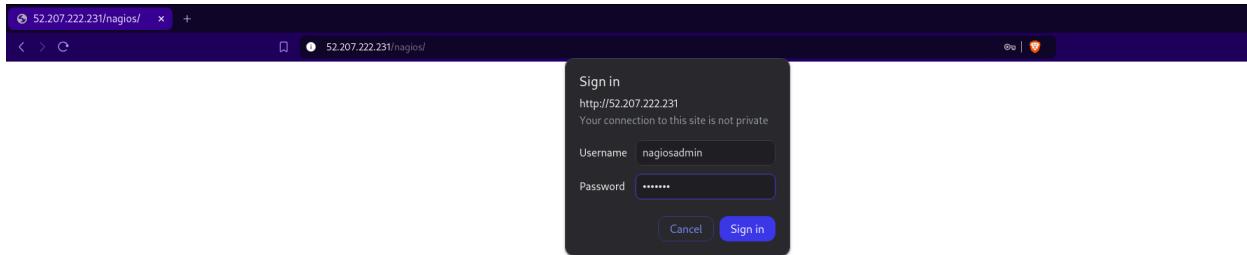
The nagios service is running and working normally

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

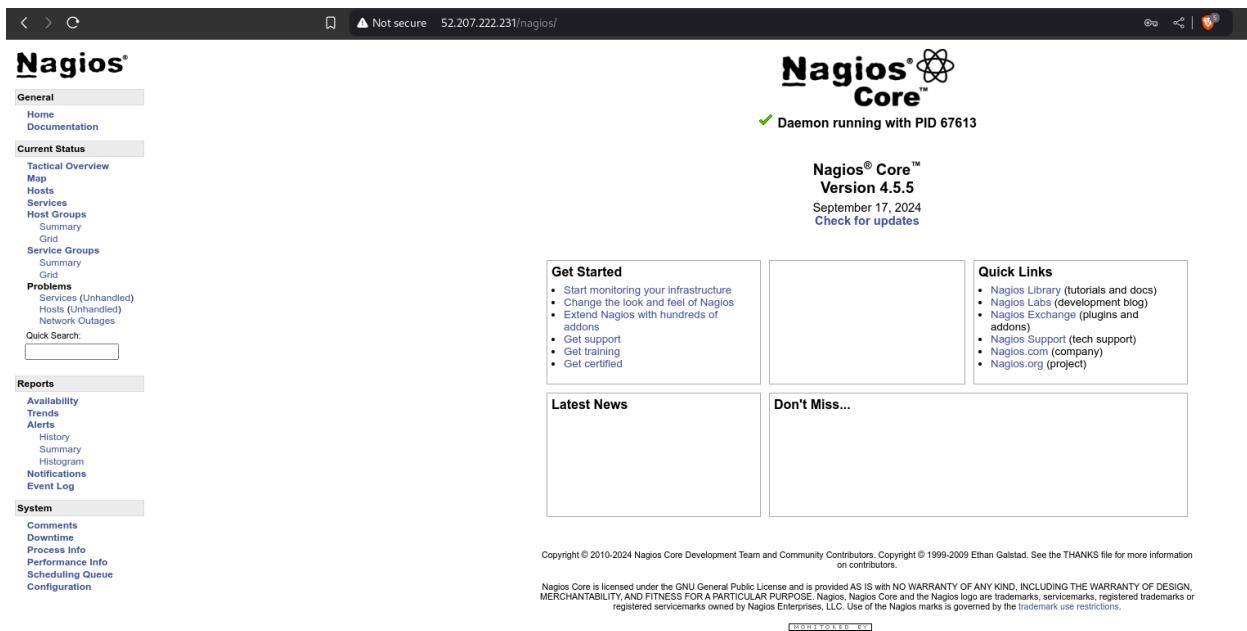


23. Open up your browser and look for http://<your_public_ip_address>/nagios

Enter username as nagiosadmin and password which we had set previously



24. After entering the correct credentials, you will see the Home page of Nagios.



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

Current Event Log

Last Updated: Fri Oct 4 14:14:42 UTC 2024
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

Log File Navigation

Fri Oct 4 00:00:00 UTC 2024 to Present..

File: /usr/local/nagios/var/nagios.log

October 04, 2024 14:00

[10-04-2024 14:12:43] wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
 [10-04-2024 14:12:43] wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
 [10-04-2024 14:12:43] wproc: early_limeout=0; exited_ok=1; wait_status=32512; error_code=0;
 [10-04-2024 14:12:43] wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
 [10-04-2024 14:12:43] wproc: NOTIFY job 3 from worker Core Worker 67616 is a non-check helper but exited with return code 127
 [10-04-2024 14:12:43] SERVICE ALERT: localhost:Swap Usage:CRITICAL:4:SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
 [10-04-2024 14:11:43] SERVICE ALERT: localhost:Swap Usage:CRITICAL:SOFT:3:SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
 [10-04-2024 14:10:43] SERVICE ALERT: localhost:Swap Usage:CRITICAL:SOFT:2:SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
 [10-04-2024 14:10:43] SERVICE ALERT: localhost:Swap Usage:CRITICAL:SOFT:1:SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
 [10-04-2024 14:09:43] SERVICE ALERT: localhost:Swap Usage:CRITICAL:SOFT:1:SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
 [10-04-2024 14:08:43] SERVICE ALERT: localhost:HTTP:WARNING:HARD:4:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.0001 second response time
 [10-04-2024 14:08:43] SERVICE ALERT: localhost:HTTP:WARNING:SOFT:3:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
 [10-04-2024 14:08:43] SERVICE ALERT: localhost:HTTP:WARNING:SOFT:2:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
 [10-04-2024 14:07:22] Successfully launched command file worker with pid 67622
 [10-04-2024 14:07:22] wproc: Registry request: name=Core Worker 67615;pid=67615
 [10-04-2024 14:07:22] wproc: Registry request: name=Core Worker 67616;pid=67616
 [10-04-2024 14:07:22] wproc: Registry request: name=Core Worker 67617;pid=67618
 [10-04-2024 14:07:22] wproc: Registry request: name=Core Worker 67617;pid=67617
 [10-04-2024 14:07:22] wproc: Successfully registered manager as @wproc with query handler
 [10-04-2024 14:07:22] qh: help for the query handler registered
 [10-04-2024 14:07:22] qh: core query handler registered
 [10-04-2024 14:07:22] qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
 [10-04-2024 14:07:22] LOG VERSION: 2.0
 [10-04-2024 14:07:22] Local time is Fri Oct 04 14:07:22 UTC 2024
 [10-04-2024 14:07:22] Nagios 4.5.5 starting... (PID=67613)
 [10-04-2024 14:07:22] Successfully shutdown... (PID=67434)
 [10-04-2024 14:07:22] Caught SIGTERM, shutting down...
 [10-04-2024 14:07:22] Caught SIGTERM, shutting down...
 [10-04-2024 14:07:22] Caught SIGTERM, shutting down...
 [10-04-2024 14:07:22] SERVICE ALERT: localhost:HTTP:WARNING:SOFT:1:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.002 second response time
 [10-04-2024 14:05:22] Successfully launched command file worker with pid 67446
 [10-04-2024 14:05:21] wproc: Registry request: name=Core Worker 67439;pid=67439
 [10-04-2024 14:05:21] wproc: Registry request: name=Core Worker 67438;pid=67438
 [10-04-2024 14:05:21] wproc: Registry request: name=Core Worker 67440;pid=67440
 [10-04-2024 14:05:21] wproc: Registry request: name=Core Worker 67441;pid=67441
 [10-04-2024 14:05:21] wproc: Successfully registered manager as @wproc with query handler

Now we can see system logs using nagios

Current Network Status

Last Updated: Fri Oct 4 14:20:53 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	0	Down	0	Unreachable	0	Pending	0
All Problems	0	All Types	1				
0	1						

Service Status Totals

Ok	6	Warning	1	Unknown	0	Critical	1	Pending	0
All Problems	2	All Types	8						
2	8								

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-04-2024 14:15:58	0d 0h 14m 55s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-04-2024 14:16:36	0d 0h 14m 17s	1/4	USERS OK - 1 users currently logged in
	HTTP	WARNING	10-04-2024 14:20:13	0d 0h 10m 40s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	10-04-2024 14:17:51	0d 0h 13m 2s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-04-2024 14:18:28	0d 0h 12m 25s	1/4	DISK OK - free space: / 6031 MB (74.31% inode=98%):
	SSH	OK	10-04-2024 14:19:06	0d 0h 11m 47s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-04-2024 14:17:43	0d 0h 8m 10s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-04-2024 14:20:21	0d 0h 10m 32s	1/4	PROCS OK: 37 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

Above is the status of all the services running on Host Machine

Conclusion:

We began the experiment by installing all the necessary packages required for Nagios. Next, we created a new user and group for Nagios, followed by installing the Nagios software through local compilation. Proper compilation is crucial to avoid errors during operation. After installation, we started both the `httpd` and Nagios services, enabling access to the Nagios dashboard, where critical system information is displayed.

EXPERIMENT NO. 10

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

Prerequisites:

Nagios Server running on Amazon Linux Machine.

STEPS:

1. To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host).

sudo systemctl status nagios

```
# ip-172-31-33-179.ec2.internal > ec2-user ~ > 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 Fri 2024-10-04 14:07:22 UTC; 1h 33min ago
     Docs: https://www.nagios.org/documentation
   Process: 67611 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 67612 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67613 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 5.0M
      CPU: 1.453s
     CGroup: /system.slice/nagios.service
             └─67613 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─67615 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67616 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67617 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67618 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67622 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─67615 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67616 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67617 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67618 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67622 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: stderr line 01: /bin/mail: No such file or directory
Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Oct 04 15:07:22 ip-172-31-33-179.ec2.internal nagios[67613]: Auto-save of retention data completed successfully.
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRIT
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: NOTIFY job 30 from worker Core Worker 67615 is a non-check helper but exited with return code 1
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: stderr line 01: /bin/mail: No such file or directory
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
```

As nagios service on main machine is running, we can proceed further

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

Launch an instance Info

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

Name and tags Info

Name

nagios-client Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

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

Recents **Quick Start**

Amazon Linux **macOS** **Ubuntu** **Windows** **Red Hat** **SUSE LI** Browse more AMIs 

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible 

ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Select the Existing Security Group and select the Security Group that we have created in

Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0a820d8ef6949a21d

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

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

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

Create security group Select existing security group

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

launch-wizard-26 sg-0a57acd1b72f678dd X
VPC: vpc-0a820d8ef6949a21d

Compare security group rules

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

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

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

ps -ef | grep nagios

```
# ip-172-31-33-179.ec2.internal ~ ps -ef | grep nagios
nagios  67613      1  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  67615  67613  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67616  67613  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67617  67613  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67618  67613  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  67622  67613  0 14:07 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 73660  73090  0 15:49 pts/1  00:00:00 grep --color=auto nagios
```

5. Now Become root user and create root directories.

sudo su

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

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

```
# ip-172-31-33-179.ec2.internal ➔ ec2-user ➔ ~ ➔ sudo su
[root@ip-172-31-33-179 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-33-179 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-33-179 ec2-user]#
```

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

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

7. Open linuxserver.cfg using nano and make the following changes in all Positions everywhere in file

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

Change hostname to **linuxserver**.

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

Set hostgroup_name to **linux-servers1**.

```
# Define a host for the local machine
define host {
    use          linux-server          ; Name of host template to use
    ; This host definition will inherit all variables that are defined
    ; in (or inherited by) the linux-server host template definition.
    host_name    linuxserver
    alias        localhost
    address      34.204.79.231
}

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

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

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

:

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

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

```
# You can specify individual object config files as shown below:  
cfg_file=/usr/local/nagios/etc/objects/commands.cfg  
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg  
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg  
cfg_file=/usr/local/nagios/etc/objects/templates.cfg  
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

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

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

```
Running pre-flight check on configuration data...  
  
Checking objects...  
    Checked 16 services.  
    Checked 2 hosts.  
    Checked 2 host groups.  
    Checked 0 service groups.  
    Checked 1 contacts.  
    Checked 1 contact groups.  
    Checked 24 commands.  
    Checked 5 time periods.  
    Checked 0 host escalations.  
    Checked 0 service escalations.  
Checking for circular paths...  
    Checked 2 hosts  
    Checked 0 service dependencies  
    Checked 0 host dependencies  
    Checked 5 timeperiods  
Checking global event handlers...  
Checking obsessive compulsive processor commands...  
Checking misc settings...  
  
Total Warnings: 0  
Total Errors: 0  
  
Things look okay - No serious problems were detected during the pre-flight check
```

We got no errors and warnings for current configuration

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

```
service nagios restart
```

```
[root@ip-172-31-33-179 ec2-user]# service nagios restart  
Redirecting to /bin/systemctl restart nagios.service
```

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

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

```
ubuntu@ip-172-31-45-81:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
```

```

Creating config file /etc/nagios-plugins/config/netware.cfg with new version
Creating config file /etc/nagios-plugins/config/nt.cfg with new version
Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version
Creating config file /etc/nagios-plugins/config/radius.cfg with new version
Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

```

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

```

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,52.207.222.231

```

13. Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server

```

ubuntu@ip-172-31-45-81:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-45-81:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-45-81:~$ 

```

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

sudo systemctl status nagios

```
[root@ip-172-31-33-179 ec2-user]# 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 Fri 2024-10-04 16:01:14 UTC; 6min ago
    Docs: https://www.nagios.org/documentation
   Process: 74405 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 74406 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 74407 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 4.2M
      CPU: 161ms
     CGroup: /system.slice/nagios.service
             ├─74407 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─74408 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74409 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74410 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74411 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─74412 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

sudo systemctl status httpd

```
[root@ip-172-31-33-179 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: active (running) since Fri 2024-10-04 13:56:26 UTC; 2h 11min ago
    Docs: man:httpd.service(8)
  Main PID: 52059 (httpd)
    Status: "Total requests: 220; Idle/Busy workers 100/0;Requests/sec: 0.0279; Bytes served/sec: 122 B/sec"
    Tasks: 230 (limit: 1112)
   Memory: 23.7M
      CPU: 5.641s
     CGroup: /system.slice/httpd.service
             ├─52059 /usr/sbin/httpd -DFOREGROUND
             ├─52061 /usr/sbin/httpd -DFOREGROUND
             ├─52065 /usr/sbin/httpd -DFOREGROUND
             ├─52066 /usr/sbin/httpd -DFOREGROUND
             ├─52067 /usr/sbin/httpd -DFOREGROUND
             └─67725 /usr/sbin/httpd -DFOREGROUND

Oct 04 13:56:25 ip-172-31-33-179.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 13:56:26 ip-172-31-33-179.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 13:56:26 ip-172-31-33-179.ec2.internal httpd[52059]: Server configured, listening on: port 80
[root@ip-172-31-33-179 ec2-user]#
```

Both nagios and httpd service is running fine on host system

15. Now to check Nagios dashboard go to <http://<nagios host ip>/nagios>
Eg. <http://34.207.68.187/nagios>

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

The dashboard features a header with the Nagios Core logo and a message indicating the daemon is running with PID 74407. Below the header, the version is listed as Nagios® Core™ Version 4.5.5, with the date September 17, 2024, and a link to check for updates. The left sidebar contains links for General, Current Status, Reports, and System. The Current Status section is expanded, showing links for Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, and Problems. The Reports section shows Availability, Trends, Alerts, History, Summary, Histogram, Notifications, and Event Log. The System section shows Comments and Downtime. The main content area includes a 'Get Started' box with a list of steps, a 'Latest News' box, a 'Don't Miss...' box, and a 'Quick Links' box with links to Nagios Library, Labs, Exchange, Support, and the official websites.

Now Click on Hosts from left side panel

The dashboard shows the 'Host Status Details For All Host Groups' table. The table has columns for Host, Status, Last Check, Duration, and Status Information. It lists two hosts: 'linuxserver' (UP, 10-04-2024 16:10:36, 0d 0h 10m 8s, PING OK - Packet loss = 0%, RTA = 0.79 ms) and 'localhost' (UP, 10-04-2024 16:09:43, 0d 2h 6m 0s, PING OK - Packet loss = 0%, RTA = 0.03 ms). The table includes a dropdown for 'Limit Results' set to 100, and filters for Host, Status, Last Check, Duration, and Status Information.

Our nagios client is showing up on nagios host dashboard

Host Information

Last Updated: Fri Oct 4 16:12:39 UTC 2024
 Updated every 90 seconds
 Nagios® Core™ 4.5.5 - www.nagios.org
 Logged in as nagiosadmin

View Status Detail For This Host
 View Alert History For This Host
 View Trends For This Host
 View Alert Histogram For This Host
 View Availability Report For This Host
 View Notifications For This Host

Host
localhost
 (linuxserver)

Member of
No hostgroups

34.204.79.231

Host State Information

Host Status:	UP (for 0d 0h 11m 25s)
Status Information:	PING OK - Packet loss = 0%, RTA = 0.79 ms
Performance Data:	rta=0.79000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-04-2024 16:10:36
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.158 seconds
Next Scheduled Active Check:	10-04-2024 16:15:36
Last State Change:	10-04-2024 16:01:14
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-04-2024 16:12:33 (0d 0h 0m 6s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Here we can see current status of nagios client machine which is up and running fine

Conclusion:

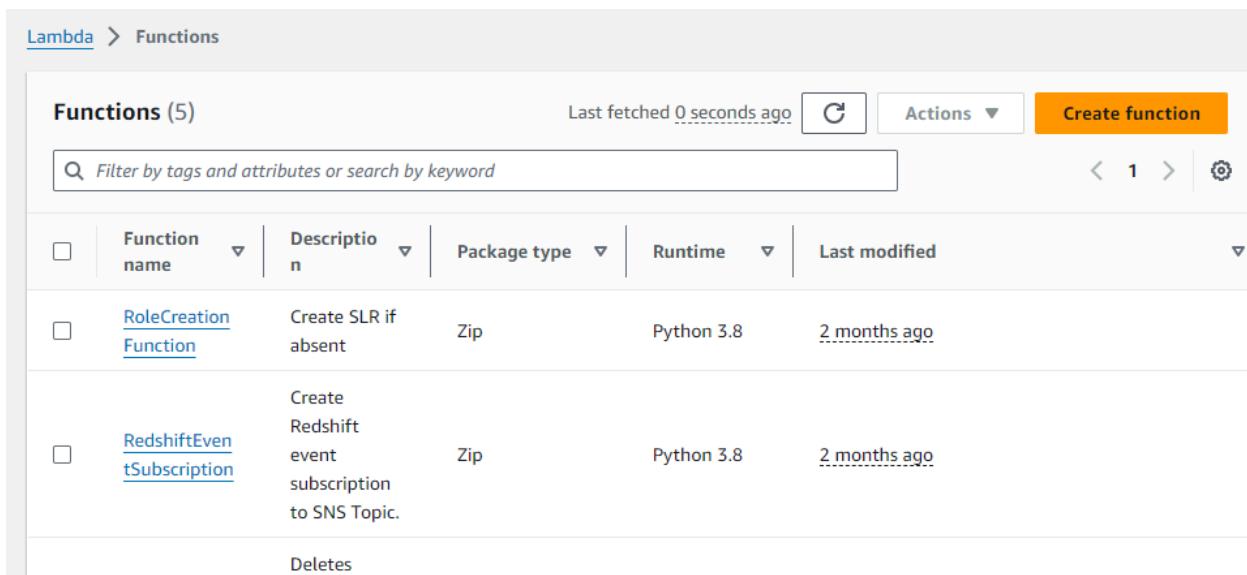
In this experiment, we created a new EC2 Linux instance and set up the Nagios client on it. We then connected the client to the Nagios host machine, allowing us to monitor alerts and status for both the host and clients on a single dashboard. It is important to configure the Nagios client with the correct host IP, as failure to do so will prevent the connection. After completing the setup, we were able to successfully view the system health details of both the client and host on the unified Nagios dashboard.

EXPERIMENT NO.11

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

Steps to create Lambda function in AWS :

1. Open up the Lambda Console and click on the Create button.
 Be mindful of where you create your functions since Lambda is region-dependent.



The screenshot shows the AWS Lambda Functions list page. At the top, there are navigation links for 'Lambda' and 'Functions'. Below that, a search bar with the placeholder 'Filter by tags and attributes or search by keyword' and a 'Create function' button. The main table has columns for 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. The table contains two rows of data:

Function name	Description	Package type	Runtime	Last modified
RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 months ago
RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 months ago

2. You can either create a function from scratch or select a blueprint, which is a pre-defined template by AWS that includes configuration settings for common use cases. Next, choose a runtime environment for your function; the dropdown will display all supported options, with Python, Node.js, .NET, and Java being the most popular. Finally, if you don't have an existing role, opt to create a new role with basic Lambda permissions.

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

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

Select proper Execution role

▼ 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

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 LabRole role](#) on the IAM console.

► Advanced settings

Cancel

Create function

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

[Lambda](#) > [Functions](#) > [lambda-alok](#)

lambda-alok

Throttle Copy ARN Actions ▾

Function overview Info Export to Application Composer Download ▾

Diagram Template

lambda-alok
Layers (0)

+ Add trigger + Add destination

Description -
Last modified 4 seconds ago
Function ARN arn:aws:lambda:us-east-1:848166917158:function:lambda-alok

Successfully created Lambda function

3. To view or change the basic settings, go to the 'Configuration' tab and click 'Edit' under 'General settings.' (THIS STEP IS OPTIONAL)

[Lambda](#) > [Functions](#) > [lambda-alok](#) > [Edit basic settings](#)

Edit basic settings

Basic settings [Info](#)

Description - *optional*
Demonstration of AWS Lambda

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

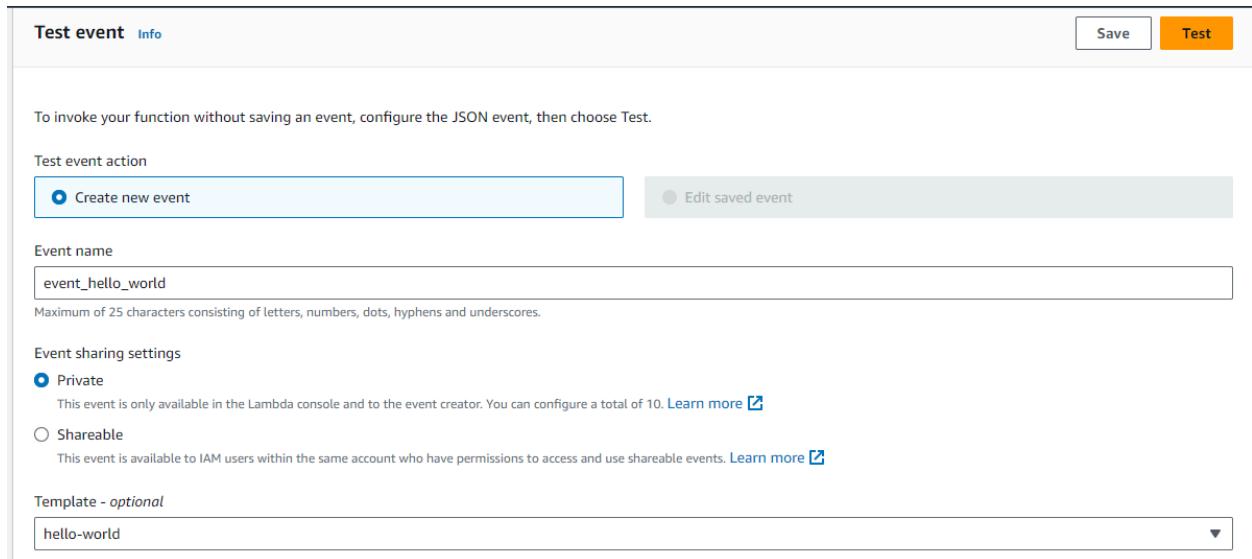
Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

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](#).

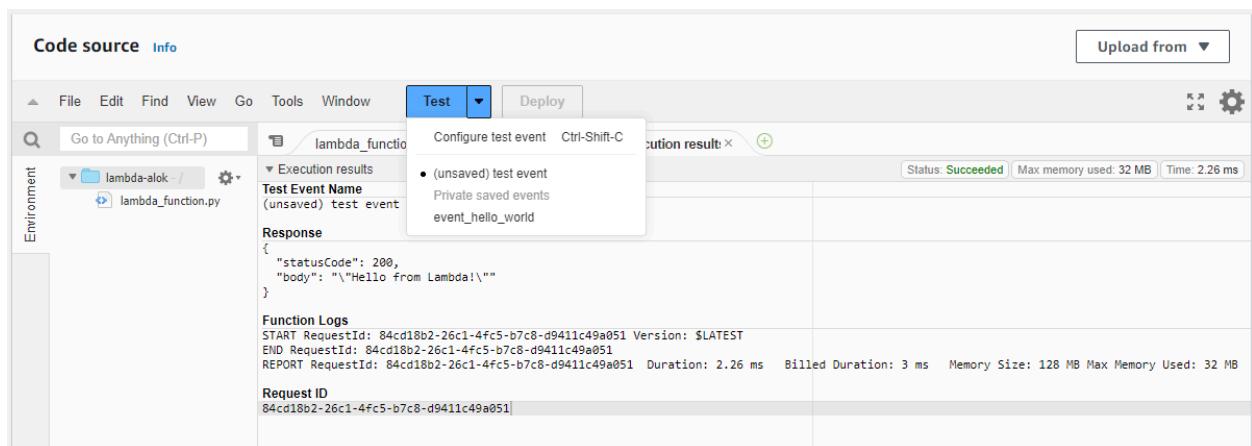
Added description.

4. Go to the 'Test' tab and click 'Create a new event.' Give the event a name, set 'Event Sharing' to private, and choose the 'hello-world' template.

We can create a new event to test and validate your Lambda function. By setting Event Sharing to private, we ensure its security, and selecting the "hello-world" template offers a straightforward framework for testing with minimal input complexity.



5. In the Code section, select the event you created from the dropdown menu under 'Test,' then click 'Test.' You should see the output below."



```

▼ Execution results
Test Event Name
event_hello_world

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

Function Logs
START RequestId: d13aac7-479f-4d8d-b24a-dd56a78d9cce Version: $LATEST
END RequestId: d13aac7-479f-4d8d-b24a-dd56a78d9cce
REPORT RequestId: d13aac7-479f-4d8d-b24a-dd56a78d9cce Duration: 2.28 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB

Request ID
d13aac7-479f-4d8d-b24a-dd56a78d9cce

```

Got above output from Lambda function

6. You can modify your Lambda function code. I've updated it to display the current Date and Time of AWS Server. After making your changes, save them using Control + S and then click on Deploy.

Function to get current Date and Time of Server:

```

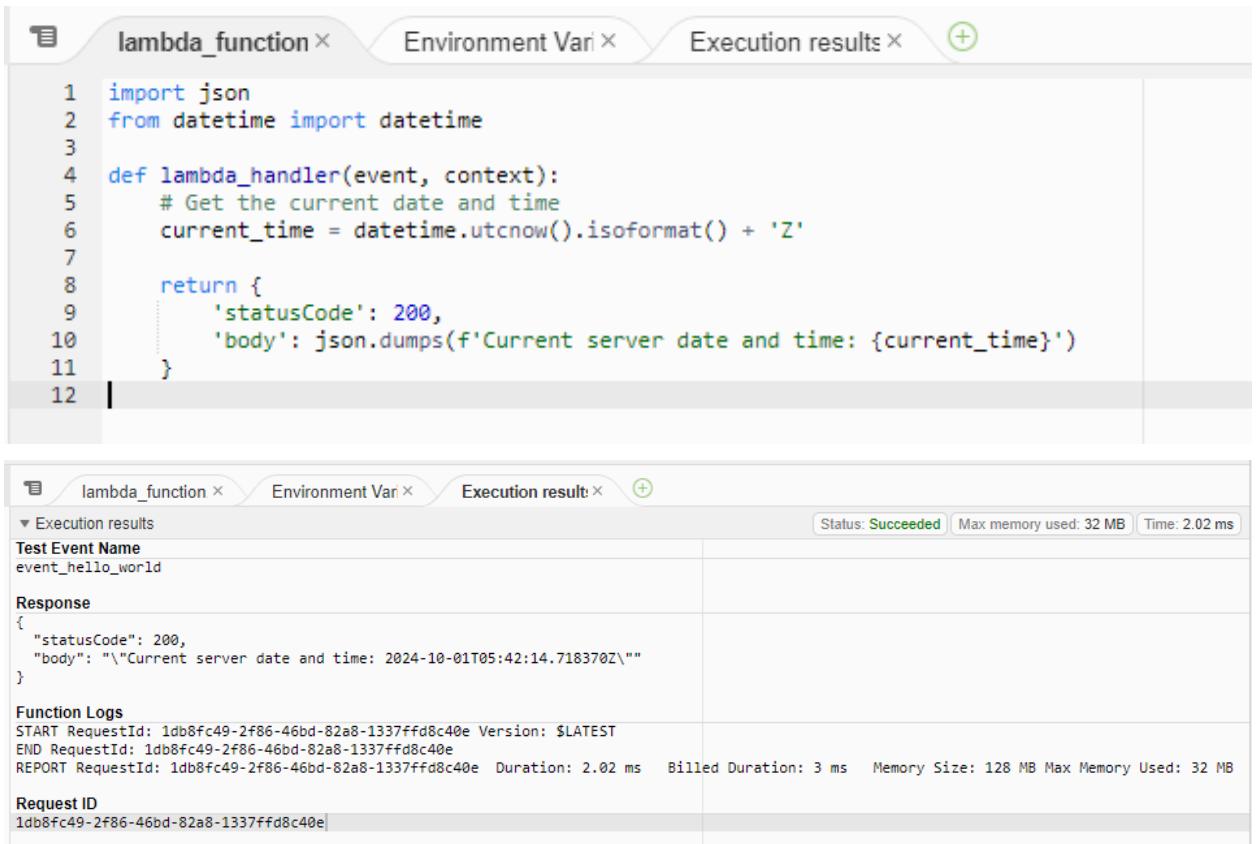
import json
from datetime import datetime

def lambda_handler(event, context):
    # Get the current date and time
    current_time = datetime.utcnow().isoformat() + 'Z'

    return {
        'statusCode': 200,
        'body': json.dumps(f'Current server date and time: {current_time}')
    }

```

Note: After making change in lambda function, ensure you deploy it before running test



The image shows two screenshots of the AWS Lambda function configuration and execution interface. The top screenshot displays the function code in the 'lambda_function' editor tab. The code is a simple Python function named 'lambda_handler' that returns the current UTC date and time in ISO format. The bottom screenshot shows the 'Execution results' tab for the same function. It includes a test event named 'event_hello_world', a successful response object with a status code of 200 and a body containing the current date and time, and detailed function logs showing the execution process.

```
1 import json
2 from datetime import datetime
3
4 def lambda_handler(event, context):
5     # Get the current date and time
6     current_time = datetime.utcnow().isoformat() + 'Z'
7
8     return {
9         'statusCode': 200,
10        'body': json.dumps(f'Current server date and time: {current_time}')
11    }
12
```

Execution results		Status: Succeeded	Max memory used: 32 MB	Time: 2.02 ms
Test Event Name	event_hello_world			
Response	{ "statusCode": 200, "body": "\"Current server date and time: 2024-10-01T05:42:14.718370Z\"" }			
Function Logs	START RequestId: 1db8fc49-2f86-46bd-82a8-1337ffd8c40e Version: \$LATEST END RequestId: 1db8fc49-2f86-46bd-82a8-1337ffd8c40e REPORT RequestId: 1db8fc49-2f86-46bd-82a8-1337ffd8c40e Duration: 2.02 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB			
Request ID	1db8fc49-2f86-46bd-82a8-1337ffd8c40e			

After changing the lambda function, we are getting Date and Time from the Server.

Conclusion:

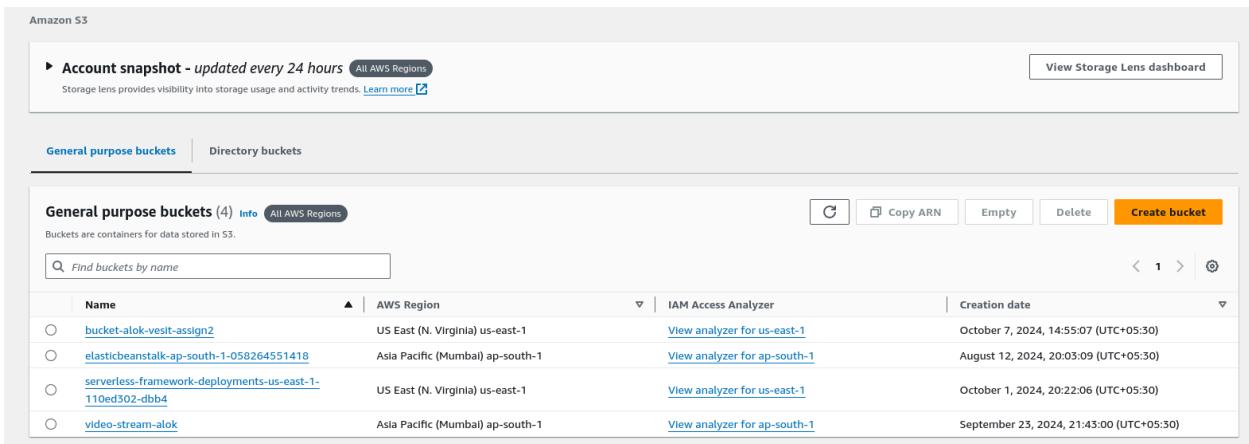
In conclusion, I conducted an experiment using the **Hello World** template and achieved the expected results. Initially, I encountered issues with my custom Lambda function, which was designed to retrieve the server's date and time, due to it not being deployed. After deploying the function, it executed correctly and returned the desired output. This experience underscored the critical importance of deploying changes to ensure they are properly reflected and take effect in production.

EXPERIMENT NO.12

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

Steps To create the lambda function:

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



Name	AWS Region	IAM Access Analyzer	Creation date
bucket-alok-visit-assign2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 7, 2024, 14:55:07 (UTC+05:30)
elasticbeanstalk-ap-south-1-058264551418	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 12, 2024, 20:03:09 (UTC+05:30)
serverless-framework-deployments-us-east-1-110ed302-dbb4	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 1, 2024, 20:22:06 (UTC+05:30)
video-stream-alok	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	September 23, 2024, 21:43:00 (UTC+05:30)

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

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

General purpose

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

Directory

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

Bucket name Info

alok-exp-12-lambda

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

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.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

Tags - optional (0)

Successfully created bucket "alok-exp-12-lambda"
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 5 [Info](#) All AWS Regions

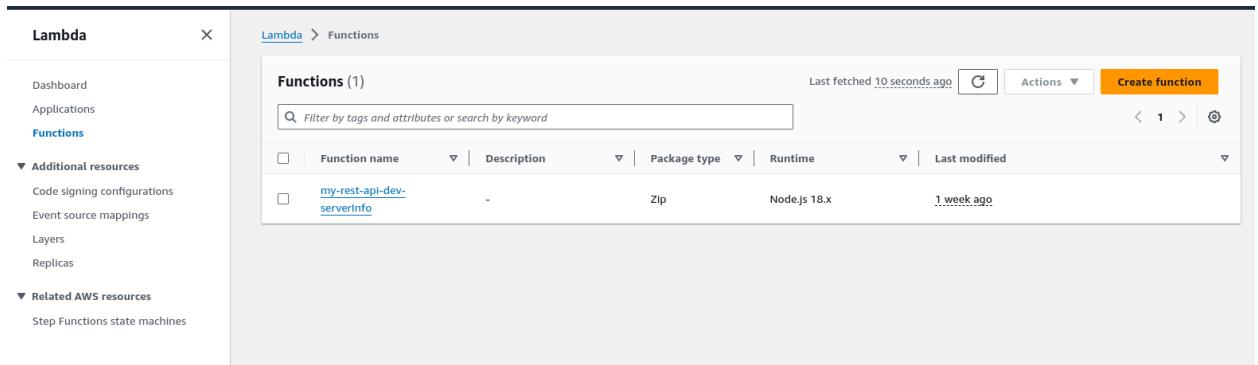
Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
alok-exp-12-lambda	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 10, 2024, 18:21:39 (UTC+05:30)
bucket-alok-visit-assign2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 7, 2024, 14:55:07 (UTC+05:30)
elasticbeanstalk-ap-south-1-059264551418	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 12, 2024, 20:03:09 (UTC+05:30)
serverless-framework-deployments-us-east-1-110ed302-dbb4	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 1, 2024, 20:22:06 (UTC+05:30)
video-stream-alok	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	September 23, 2024, 21:45:00 (UTC+05:30)

Bucket created successfully

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



Function name	Description	Package type	Runtime	Last modified
my-rest-api-dev-serverInfo	-	Zip	Node.js 18.x	1 week ago

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.

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

Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

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](#)

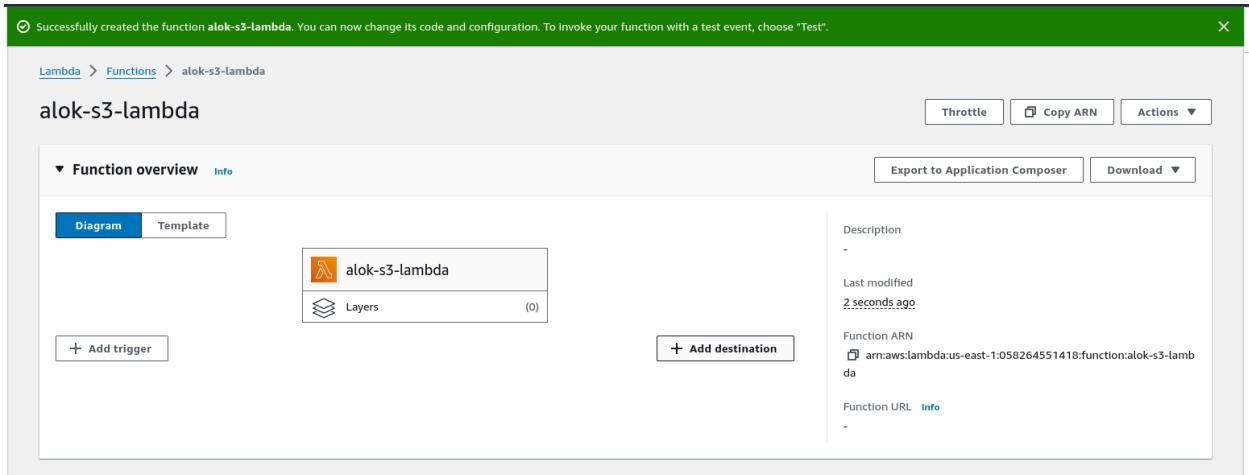
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role**Execution role**

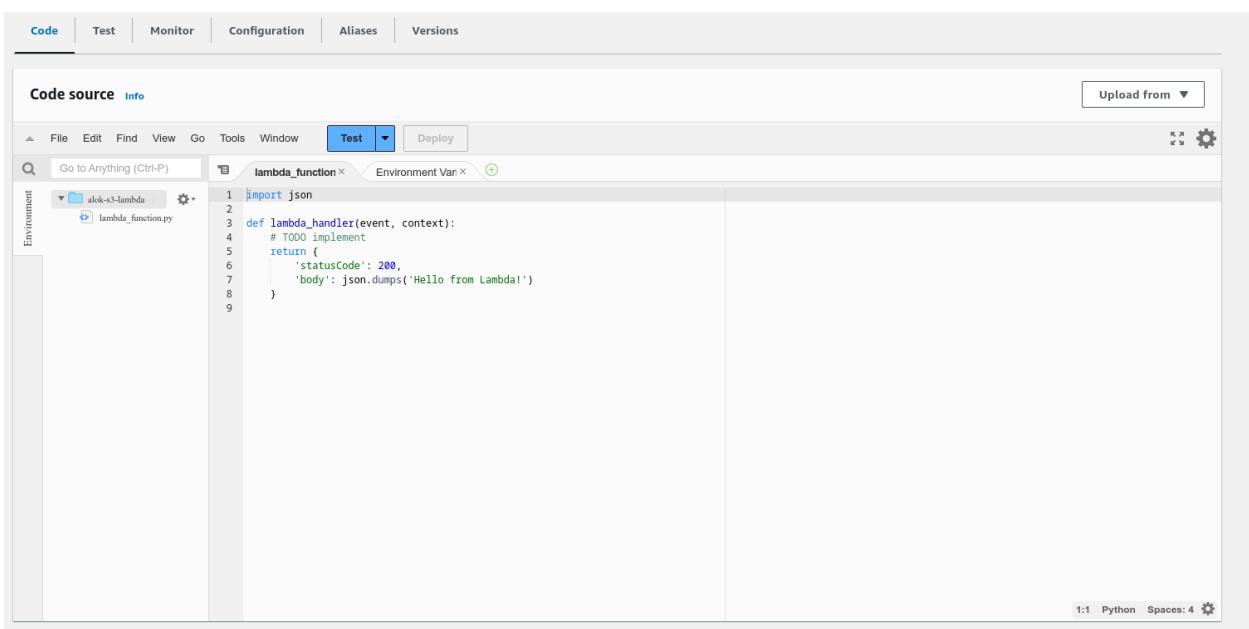
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#)

 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 function created successfully

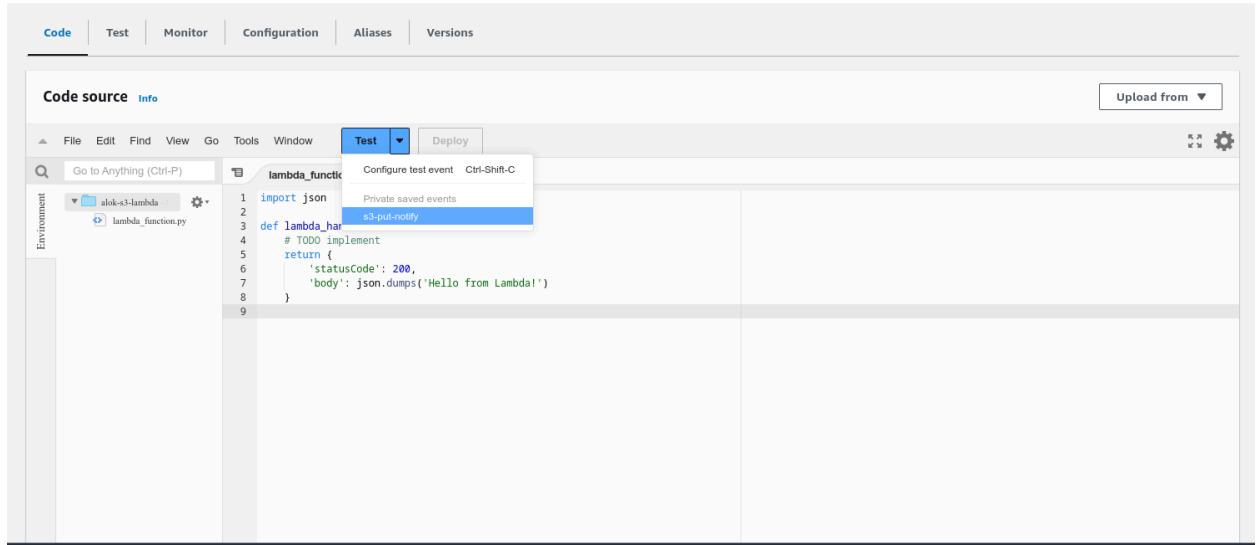


So to See or Edit the basic settings go to configuration then click on edit general setting.

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

- 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.

- Now In Code section select the created event from the dropdown .



The screenshot shows the AWS Lambda code editor interface. The 'Code' tab is selected. The 'Test' tab is highlighted in blue. A dropdown menu is open under the 'Test' tab, showing options: 'Configure test event' and 's3-put-notification'. The 's3-put-notification' option is highlighted with a blue selection bar. The code editor window shows a file named 'lambda_function.py' with the following content:

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

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.

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

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

The image shows two screenshots of the AWS Lambda console. The top screenshot is the 'Function overview' page for the function 'alok-s3-lambda'. It displays a diagram showing the function 'alok-s3-lambda' triggered by an 'S3' event. The bottom screenshot is the 'Configuration' tab of the same function, showing the 'Triggers' section with one trigger named 'S3: alok-exp-12-lambda'.

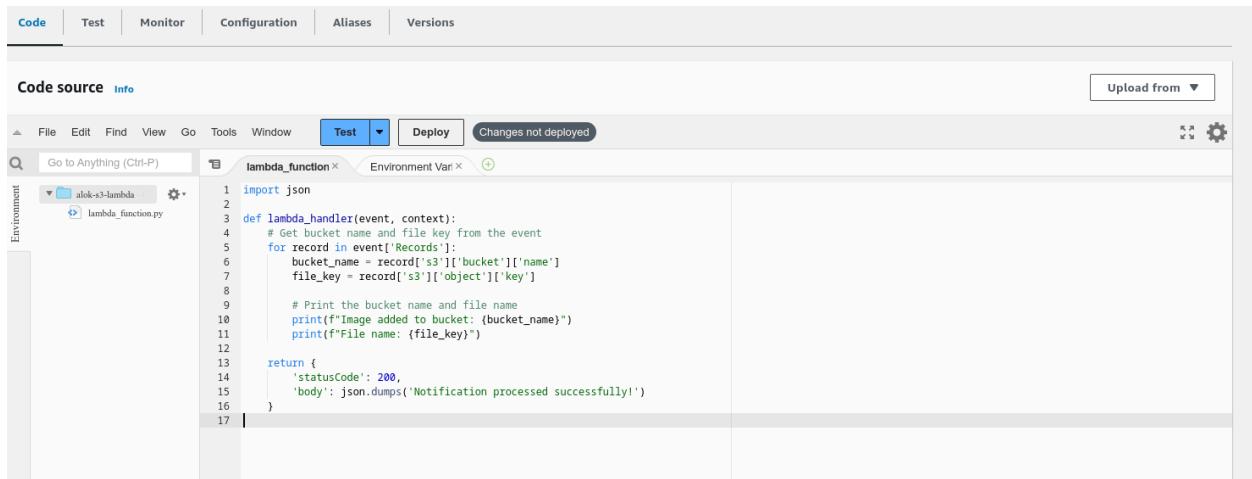
- Now Write code that logs a message like "Image added to bucket" when triggered. Save the file and click on deploy.

```
import json
```

```
def lambda_handler(event, context):
    # Get bucket name and file key from the event
    for record in event['Records']:
        bucket_name = record['s3']['bucket']['name']
        file_key = record['s3']['object']['key']

        # Print the bucket name and file name
        print(f"Image added to bucket: {bucket_name}")
        print(f"File name: {file_key}")

    return {
        'statusCode': 200,
        'body': json.dumps('Notification processed successfully!')
    }
```

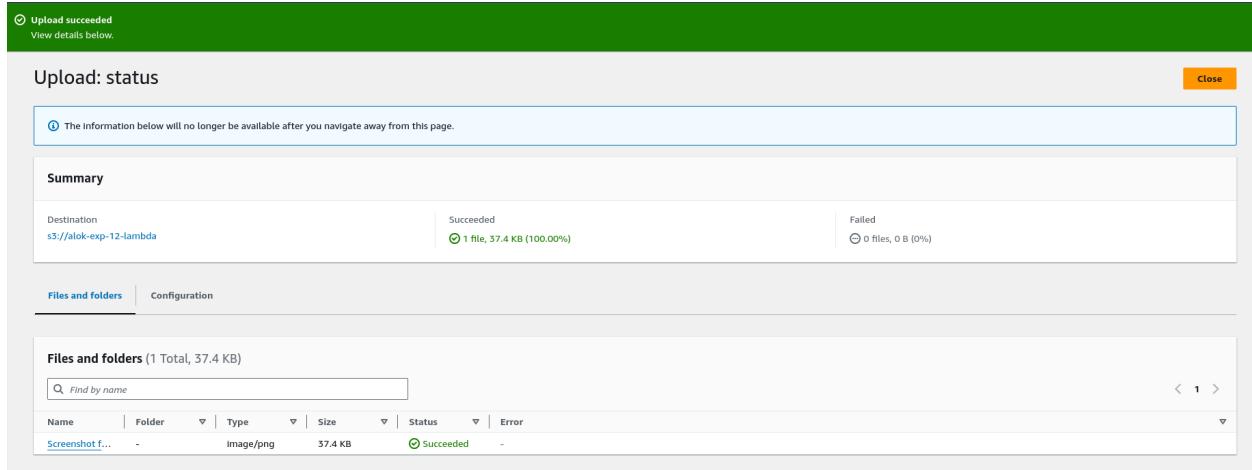


```

1 import json
2
3 def lambda_handler(event, context):
4     # Get bucket name and file key from the event
5     for record in event['Records']:
6         bucket_name = record['s3']['bucket']['name']
7         file_key = record['s3']['object']['key']
8
9         # Print the bucket name and file name
10        print(f"Image added to bucket: {bucket_name}")
11        print(f"File name: {file_key}")
12
13    return {
14        'statusCode': 200,
15        'body': json.dumps('Notification processed successfully!')
16    }
17

```

9. Now upload any image to the bucket.



Upload succeeded

View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://alok-exp-12-lambda	1 file, 37.4 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 37.4 KB)

Name	Folder	Type	Size	Status	Error
Screenshot f...	-	Image/png	37.4 KB	Succeeded	-

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

The screenshot shows the AWS Lambda Test interface. The 'Execution results' section displays the following output:

```

Test Event Name
s3-put-notify

Response
{
  "statusCode": 200,
  "body": "\"Notification processed successfully!\""
}

Function Logs
START RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266 Version: $LATEST
Image added to bucket: example-bucket
File name: test%2Fkey
END RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266
REPORT RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266 Duration: 1.68 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
Request ID
2a6c5f2c-dd25-4d19-9163-7d6e2bf42266
  
```

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

The screenshot shows the AWS CloudWatch Logs interface. The 'Log events' section displays the following log entries:

Timestamp	Message
2024-10-10T13:46:32.073Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:18809ca2e2714ff5637bd2bbe@06ceb81ec3bc480a0f277dab104c14cd814b081
2024-10-10T13:46:32.108Z	START RequestId: 39c8107c-b8a5-4ad4-b28d-a910ab099bbc Version: \$LATEST
2024-10-10T13:46:32.181Z	Image added to bucket: alok-exp-12-lambda
2024-10-10T13:46:32.181Z	File name: Screenshot+from+2024-10-10-19-11-54.png
2024-10-10T13:46:32.193Z	END RequestId: 39c8107c-b8a5-4ad4-b28d-a910ab099bbc
2024-10-10T13:46:32.193Z	REPORT RequestId: 39c8107c-b8a5-4ad4-b28d-a910ab099bbc Duration: 11.52 ms Billed Duration: 12 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 104.19 ms
2024-10-10T13:47:51.693Z	START RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266 Version: \$LATEST
2024-10-10T13:47:51.693Z	Image added to bucket: example-bucket
2024-10-10T13:47:51.711Z	File name: test%2Fkey
2024-10-10T13:47:51.711Z	END RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266
2024-10-10T13:47:51.712Z	REPORT RequestId: 2a6c5f2c-dd25-4d19-9163-7d6e2bf42266 Duration: 1.68 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB

As we can see our lambda function activity is recorded by Cloudwatch

Conclusion:

In this experiment, we successfully implemented an AWS Lambda function that logs a message when an image is uploaded to an S3 bucket. A key aspect was selecting the **S3 Object Created (Put)** event template, as failing to do so would result in errors due to an incompatible event structure. The Lambda function was successfully triggered by S3 object uploads, demonstrating the functionality and efficiency of AWS Lambda's event-driven architecture. This experiment highlighted Lambda's ability to seamlessly respond to S3 events and emphasized

the importance of correctly configuring event triggers to avoid common issues with event data.