

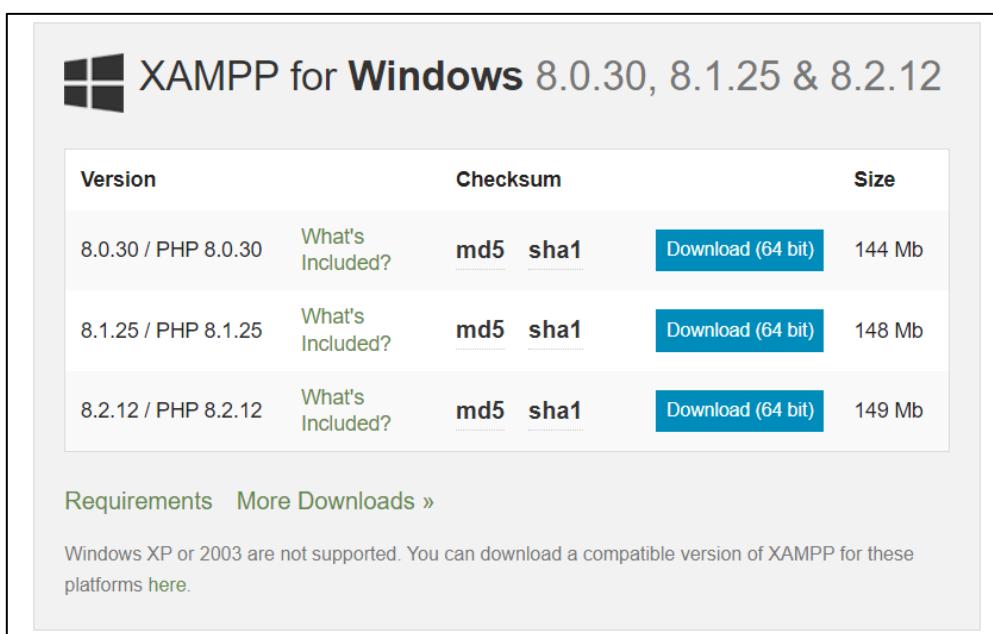
Advance DevOps

Experiment 1a

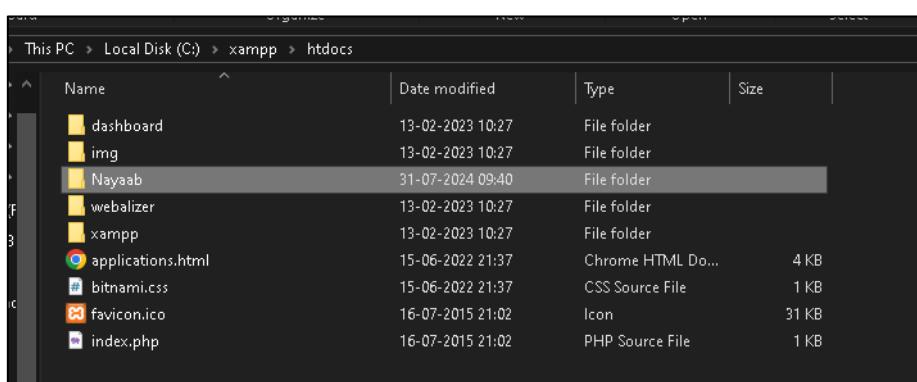
Part 1: To develop a website and host it on your local machine on a VM

Following are the steps to be followed:

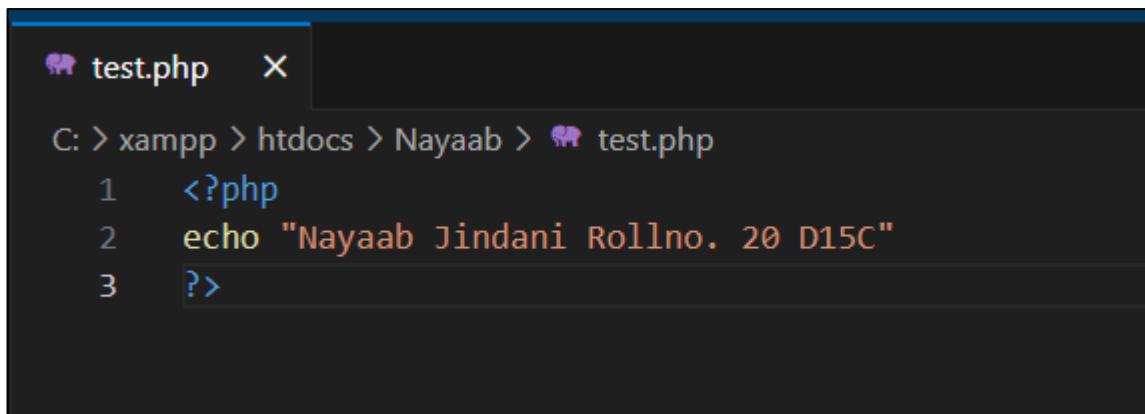
1. First step is to download XAMPP through the official website, <https://www.apachefriends.org/download.html> and then complete the installation process.



2. To run a php script go to C:\xampp\htdocs and create a folder (here my folder is named Nayaab).



3. Inside the folder create a file and save it as filename.php and write php script inside this file.



```

test.php  X

C: > xampp > htdocs > Nayaab > test.php

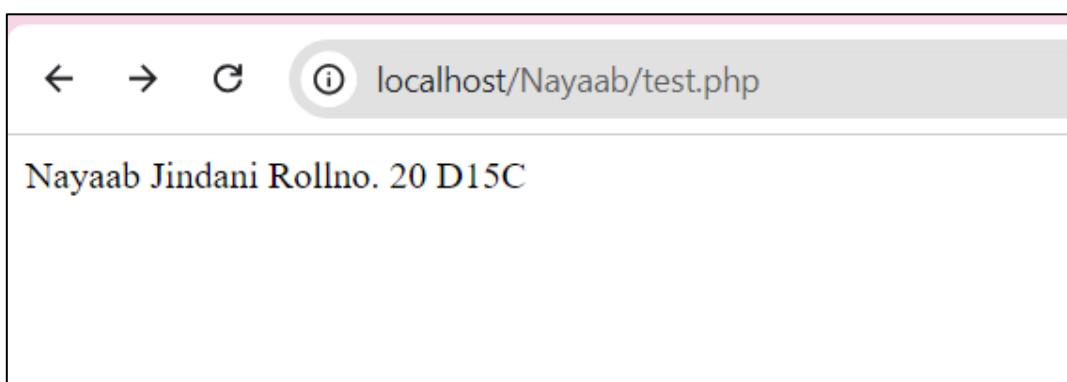
1 <?php
2 echo "Nayaab Jindani Rollno. 20 D15C"
3 ?>

```

4. Now go to XAMPP control panel and start Apache server.



5. Now type `http://localhost/folder_name/filename.php` in your browser and you will get to see the output.



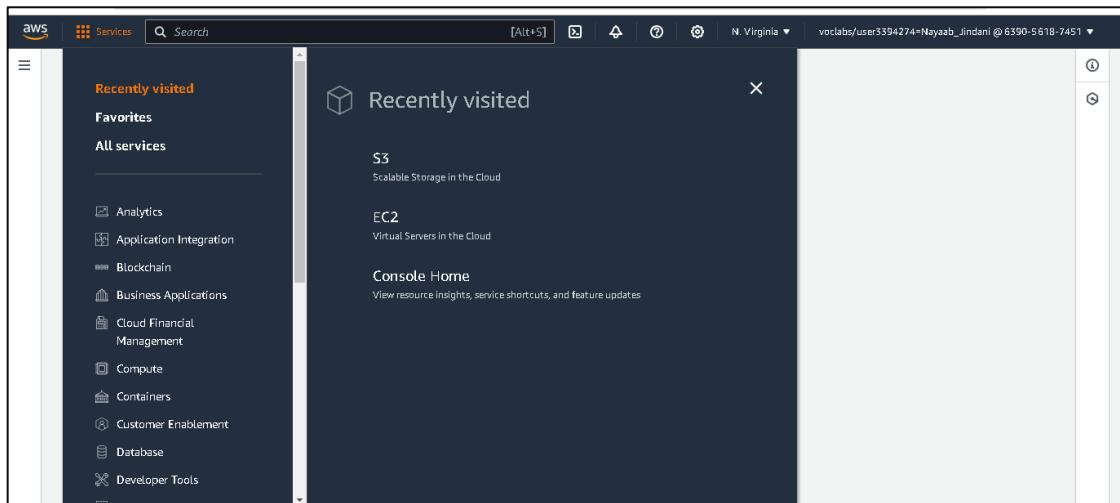
Advance DevOps

Experiment 1a

Part 2: Hosting a static website on Amazon S3

Steps:

1. Go to AWS console home -> services and select S3.



2. Select create a bucket

A screenshot of the Amazon S3 service page. The main header says 'Amazon S3' and 'Store and retrieve any amount of data from anywhere'. Below it, a sub-header says 'How it works' and shows a video thumbnail titled 'Introduction to Amazon S3'. To the right, there's a 'Create a bucket' button inside a box with explanatory text: 'Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.' Further down, there's a 'Pricing' section with text about no minimum fees and links to the AWS Simple Monthly Calculator.

3. Enter bucket name and select create bucket.

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 - New

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](#)

nayaab

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.

4. Now upload the folder or files of the website to be hosted.

Upload [Info](#)

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

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

Files and folders (1 Total, 103.0 B)

All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	test_doc.html	test_bucket/	text/html

Find by name

< 1 >

5. After uploading the folder or files when you select the object URL of your file it will show access denied:

```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>7R75XT2NFV2HWY48</RequestId>
  <HostId>PnyRp52k1NIJqU5T/ITdnTgvivnUnyOf3JzgesbK6QFmGKLL48p91uPmr78+eYFHa1LoftVm9zbs9qiEFwxirfHTxmhHjs4v1/aKuKJg2Q=</HostId>
</Error>

```

- To provide access we need to enable static website hosting and to do so go to buckets -> your bucket -> properties -> static website hosting and enable static website hosting. Also Specify the home or default page of the website. Then click on save changes.

Amazon S3 > Buckets > nayaab > Edit static website hosting

Edit static website hosting Info

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

Static website hosting
 Disable
 Enable

Hosting type
 Host a static website
 Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
 Redirect requests to another bucket or domain. [Learn more](#)

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

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

- Now select the permission tab which is besides properties and uncheck block all public access.

Edit Block public access (bucket settings) Info

Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

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

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

- Now we have to make all objects related to our website publicly accessible and that can be done in 2 ways: either we can select each object and provide permission or use the bucket policy and provide permission for all files at once.

To select a file and give permission first go to permissions and enable ACL.

Amazon S3 > Buckets > nayaab > Edit Object Ownership

Edit Object Ownership info

Object Ownership

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.

Now select the objects related to your website click on actions and then click on make public using ACL. Select make public.

Actions ▾ **Create folder**

- Share with a presigned URL
- Calculate total size
- Copy
- Move
- Initiate restore
- Query with S3 Select
- Edit actions**
- Rename object
- Edit storage class
- Edit server-side encryption
- Edit metadata
- Edit tags
- Make public using ACL**

6. Now when you open the URL you will be able to see the website.



Advance DevOps

Experiment 1b

Steps to setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and perform collaboration:

1. Login in your AWS account. Search Cloud9 in services.

The screenshot shows the AWS search interface with the query 'cloud9'. On the left, there's a sidebar with categories like Services (51), Features (32), Resources (New), Documentation (15,277), Knowledge Articles (652), Marketplace (13), Blogs (6,912), Events (327), and Tutorials (22). The main area displays three results: 'Cloud9' (a Cloud IDE for writing, running, and debugging code), 'Amazon CodeCatalyst' (an integrated DevOps service), and 'AWS Cloud Map' (a service for building dynamic maps of your cloud).

2. Select create environment.

The screenshot shows the AWS Cloud9 landing page. It features the heading 'AWS Cloud9' and the subtext 'A cloud IDE for writing, running, and debugging code'. Below this is a paragraph about AWS Cloud9's benefits and a link to its documentation. On the right side, there's a callout box with the text 'New AWS Cloud9 environment' and a prominent orange 'Create environment' button.

3. Provide the name, description (optional).

AWS Cloud9 > Environments > Create environment

Create environment [Info](#)

Details

Name
WebAppIDE

Description - optional

Limit 200 characters.

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.

4. Select Secure Shell (SSH) as the connection type and 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.

5. Your Cloud9 IDE has been created.

Environments (1)						Delete	View details	Open in Cloud9	Create environment
My environments									
Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN				
<input type="radio"/> WebAppIDE	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::639056187451:assumed-role/voclabs/user3394274=Nayaab_Jindani				

6. Now we will create user. For that go to IAM -> Users -> Create user.

The screenshot shows the AWS IAM service interface. On the left, there's a navigation sidebar with options like Dashboard, Access management, and Access reports. Under Access management, the 'Users' section is selected, showing a list of users with one entry: 'User name' (Nayaab). The main panel displays the 'Users (0) Info' section, which defines an IAM user as an identity with long-term credentials used to interact with AWS. A prominent orange 'Create user' button is located at the top right of the main content area.

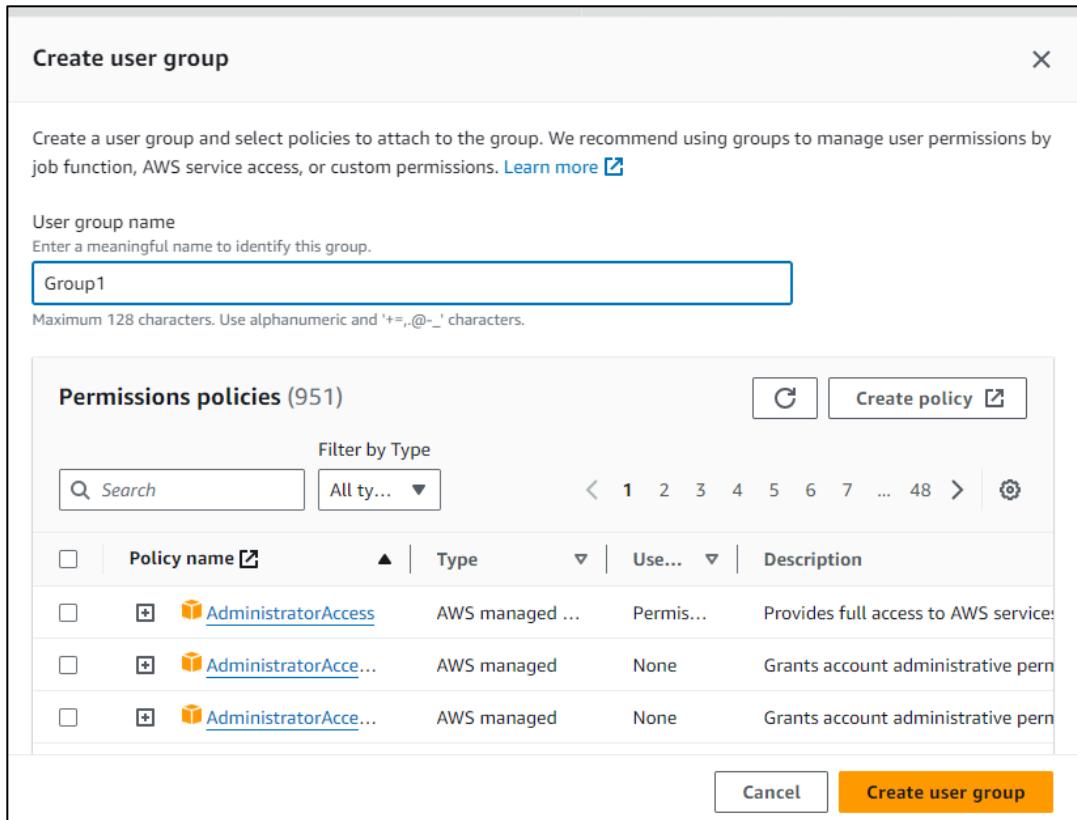
7. Enter user details like user name, password and click on next.

This screenshot shows the 'User details' creation form. The 'User name' field is filled with 'Nayaab'. The 'Provide user access to the AWS Management Console - optional' checkbox is checked. The 'Console password' section shows 'Custom password' selected, with a password entered as '*****'. Below the password field are two bullet points: 'Must be at least 8 characters long' and 'Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | '.

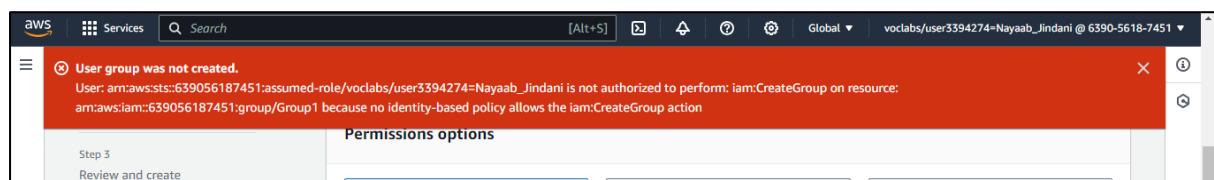
8. Select create group in the next section.

This screenshot shows the 'Get started with groups' section. It contains a callout box with the heading 'Get started with groups' and the text: 'Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions.' A blue 'Learn more' link is present. To the right of the callout is a 'Create group' button.

9. Enter group name, select permission policies and select create user group.



10. Due to authorization issues the group was not created and the below given error was displayed.



11. Now we will sign in to the AWS account to continue with the further procedure and sign out from the console we were working on earlier. After logging in search for IAM in services and click on it.

Search results for 'iam'

Services (11)

- Features (24)
- Resources **New**
- Documentation (59,387)
- Knowledge Articles (471)
- Marketplace (857)
- Blogs (1,838)

Services

IAM ☆ Manage access to AWS resources

Top features

Groups Users Roles Policies Access Analyzer

See all 11 results ▶

12. Now again we will follow the same procedure to create user and group as done above.

IAM > Users

Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in
No resources to display						

Create user

User type

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

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

Console password

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } { }

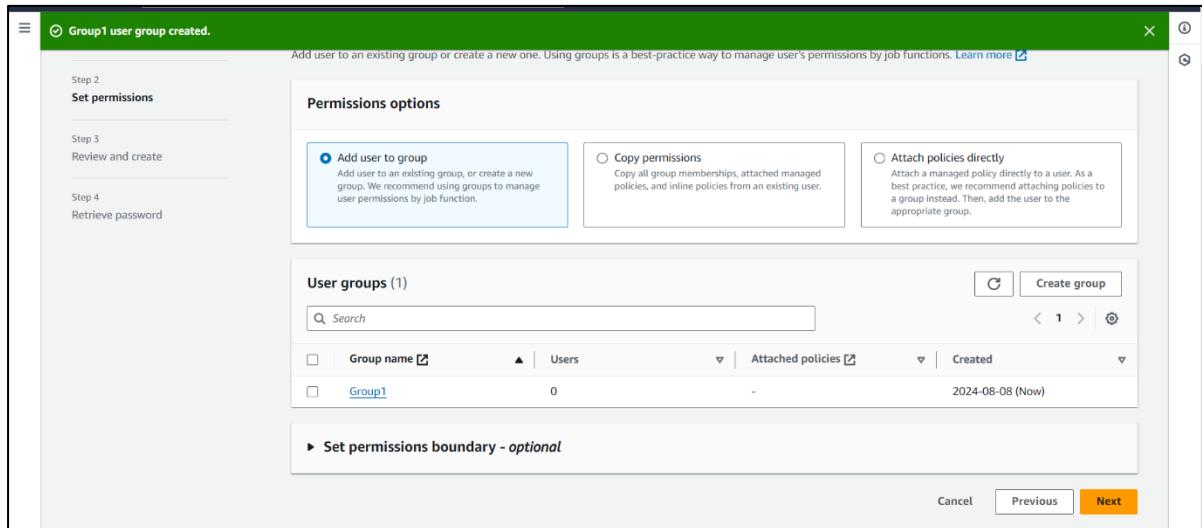
Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

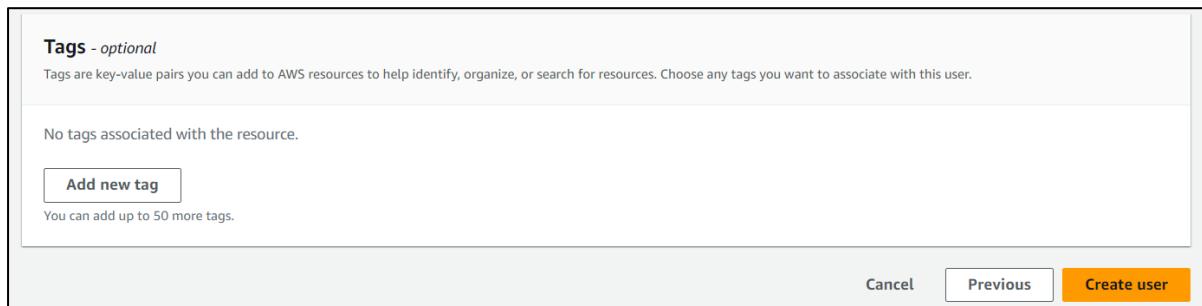
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

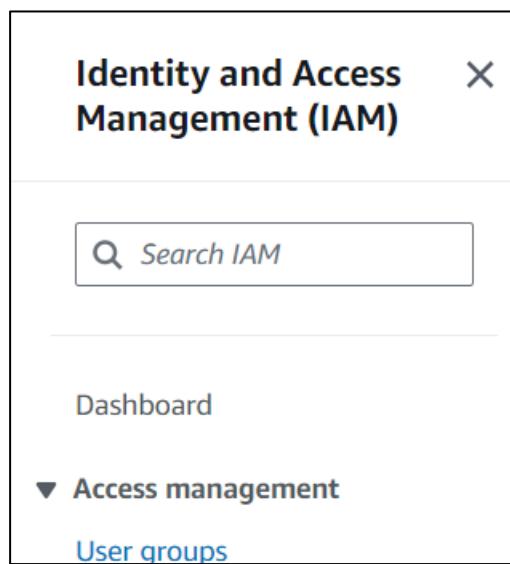
13. Select add user to group and click on next.



14. Add tags (optional) and select create user.



15. In the left pane select user groups.



16. Click on the group you created and go to permissions tab and select attach policies from add permissions.

The screenshot shows the 'Permissions' tab selected in the AWS IAM console. The 'Permissions policies' section displays a table with one row: 'No resources to display'. At the top right, there are several buttons: 'Add permissions' (highlighted in blue), 'Simulate', 'Remove', 'Attach policies' (also highlighted in blue), and 'Create inline policy'. Below these buttons are search and filter fields. The table has columns for 'Policy name', 'Type', and 'Attached entities'.

17. Attach the policy AWSCloud9EnvironmentMember by selecting it and click on attach policies.

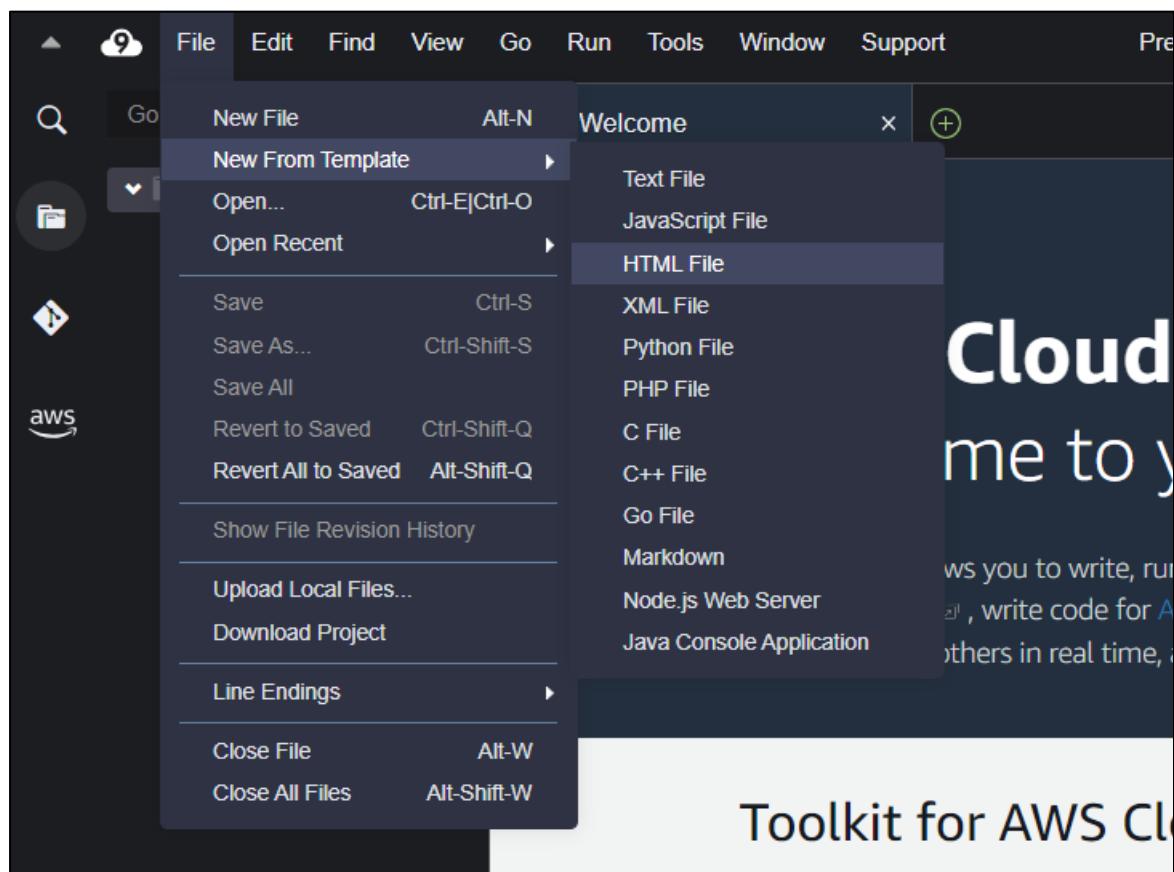
The screenshot shows a modal dialog titled 'Select Policy' with a list of AWS managed policies. The 'AWScloud9EnvironmentMember' policy is selected, indicated by a checked checkbox. At the bottom right of the dialog are 'Cancel' and 'Attach policies' buttons, with 'Attach policies' being highlighted in orange.

Policy Name	Type	Description
AmazonCloudWatchRUMFullAccess	AWS managed	Grants full access permissions for the ...
AmazonCloudWatchRUMReadOnly...	AWS managed	Grants read only permissions for the A...
AmazonDMSCloudWatchLogsRole	AWS managed	Provides access to upload DMS replicat...
AmazonGrafanaCloudWatchAccess	AWS managed	This policy grants access to Amazon Cl...
AmazonSageMakerPartnerService...	AWS managed	Service role policy used by the AWS Cl...
AmazonSageMakerServiceCatalog...	AWS managed	Service role policy used by the AWS Cl...
AWSAppSyncPushToCloudWatchL...	AWS managed	Allows AppSync to push logs to user's ...
AWScloud9Administrator	AWS managed	Provides administrator access to AWS ...
<input checked="" type="checkbox"/> AWScloud9EnvironmentMember	AWS managed	Provides the ability to be invited into ...
AWSCloud9SSMInstanceProfile	AWS managed	This policy will be used to attach a rol...
AWScloud9User	AWS managed	Provides permission to create AWS Clo...
AWScloudFormationFullAccess	AWS managed	Provides full access to AWS CloudForma...
AWScloudFormationReadOnlyAccess	AWS managed	Provides access to AWS CloudFormatio...
AWScloudHSMFullAccess	AWS managed	Provides full access to all CloudHSM re...
AWScloudHSMReadOnlyAccess	AWS managed	Provides read only access to all Cloud...

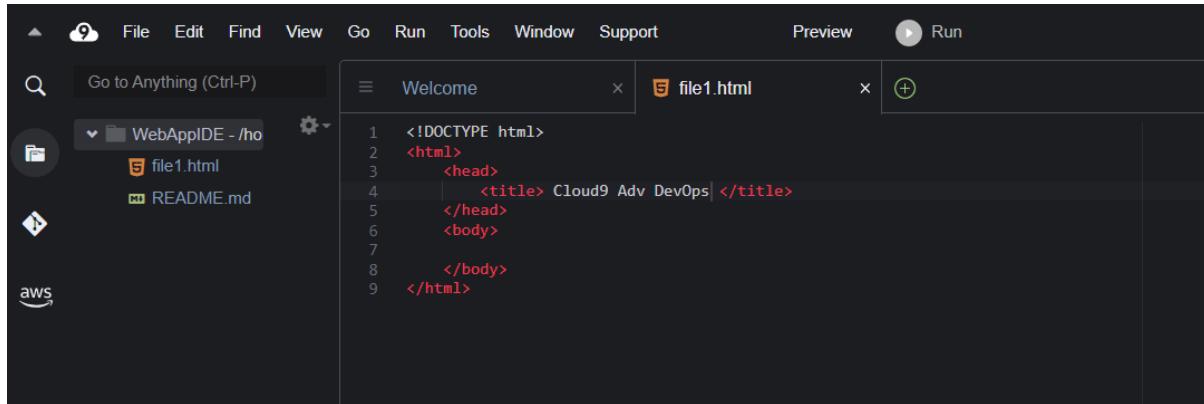
18. Policy has been successfully attached to the user group

The screenshot shows the AWS IAM User Groups console. The navigation path is IAM > User groups > Group1. The main view displays the 'Group1' details, including its name, creation time, and ARN. Below this, the 'Permissions' tab is selected, showing a list of attached policies. One policy, 'AWSCloud9EnvironmentMember', is listed under 'Attached entities'. There are buttons for 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add permissions'.

19. Now we will open the Cloud9 IDE we created earlier. Then go to file -> new from template -> HTML file (any file type can be selected).

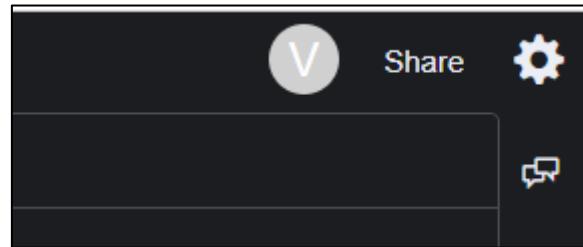


20. You can now edit the html file and save it.



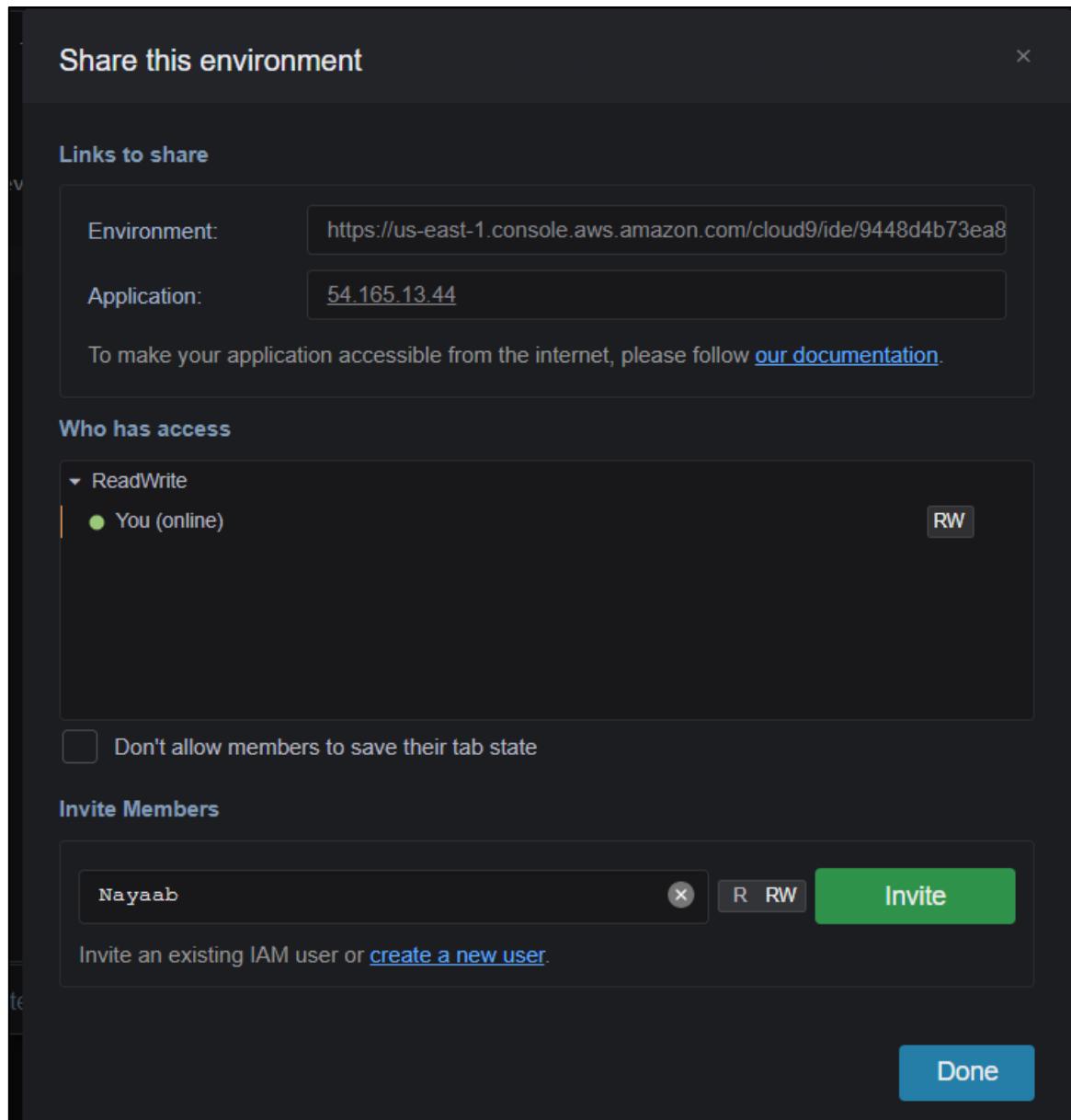
```
<!DOCTYPE html>
<html>
  <head>
    <title> Cloud9 Adv DevOps </title>
  </head>
  <body>
    </body>
  </html>
```

21. We can collaborate with other members by sharing this file. To do so click on the share option to the top right of the screen.



22. Here you can give read or read/write permission to your team members and click on invite.

Your teammate will get an invite in “shared with you” and after he/she selects open IDE they will be able to see the same interface as yours and now you and your team members can collaborate in real time.



Advance DevOps

Experiment 2

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:

1. Firstly, we will create an environment and for that search elastic beanstalk in the services.

The screenshot shows the AWS search interface with the query 'elastic beanstalk' entered in the search bar. The results are filtered under the 'Services' category, which contains 12 items. The top result is 'Elastic Beanstalk', described as 'Run and Manage Web Apps'. Below it are 'Elastic Transcoder' (Easy-to-Use Scalable Media Transcoding) and 'Elastic Container Service' (Highly secure, reliable, and scalable way to run containers). To the left of the main results, there's a sidebar with links to other AWS services like Features, Resources, and Documentation.

2. Select create application.

The screenshot shows a 'Get started' page with the heading 'Get started' and the subtext 'Easily deploy your web application in minutes.' A prominent orange button labeled 'Create application' is centered at the bottom of the white box.

3. Now we will configure the environment. In environment tier select web server environment and give application name.

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

Maximum length of 100 characters.

► Application tags (optional)

4. Select a platform, here I have selected php.

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

Platform branch

Platform version

5. Keep all other settings at their default values

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

6. Now we will configure service access. Select an existing service role, EC2 key pair and EC2 instance profile from the dropdowns given.

Configure service access [Info](#)

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

Service role
 Create and use new service role
 Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.
EMR_EC2_DefaultRole ▼

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

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

[View permission details](#)

7. In the setup network section select a vpc and instance subnets then click on next.

Set up networking, database, and tags - optional Info

Virtual Private Cloud (VPC)

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

vpc-000515808c77a2ee6 | (172.31.0.0/16)

[Create custom VPC](#)

Instance subnets

Filter instance subnets

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-1e	subnet-048c3b524...	172.31.48.0/20	
<input type="checkbox"/>	us-east-1d	subnet-06b262b3f...	172.31.80.0/20	
<input type="checkbox"/>	us-east-1a	subnet-098581ca2...	172.31.16.0/20	
<input checked="" type="checkbox"/>	us-east-1b	subnet-0d832834f...	172.31.32.0/20	
<input type="checkbox"/>	us-east-1f	subnet-0e49b4a70...	172.31.64.0/20	
<input checked="" type="checkbox"/>	us-east-1c	subnet-0fe29786e...	172.31.0.0/20	

8. In the next step select EC2 security groups and set instance type as t2.micro

EC2 security groups
Select security groups to control traffic.

EC2 security groups (3)

Filter security groups

	Group name	Group ID	Name
<input checked="" type="checkbox"/>	aws-cloud9-WebAppIDE-94...	sg-0ef761e90503c99e9	
<input type="checkbox"/>	default	sg-0b49cb198da1d9424	
<input type="checkbox"/>	launch-wizard-1	sg-084839deaf51584f0	

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

Choose x86 instance types ▾

t2.micro X

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

ami-083f545ce1a73bf03

9. Next is the review page where you can check the configurations that have been set in the previous steps. Click on submit.

Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information	
Environment tier	Application name
Web server environment	Application1
Environment name	Application code
Application1-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3	
running on 64bit Amazon Linux 2023/4.3.1	

Step 2: Configure service access [Edit](#)

Service role	EC2 key pair	EC2 instance profile

10. Your environment will be successfully created.

Environment successfully launched.

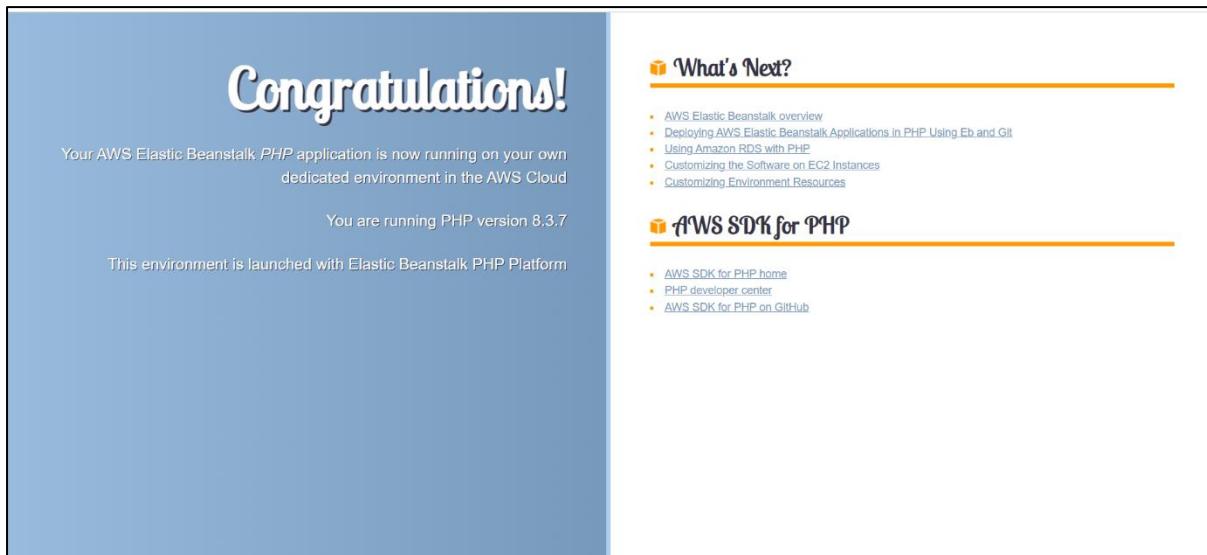
Elastic Beanstalk > Environments > Application1-env

Application1-env [Info](#)

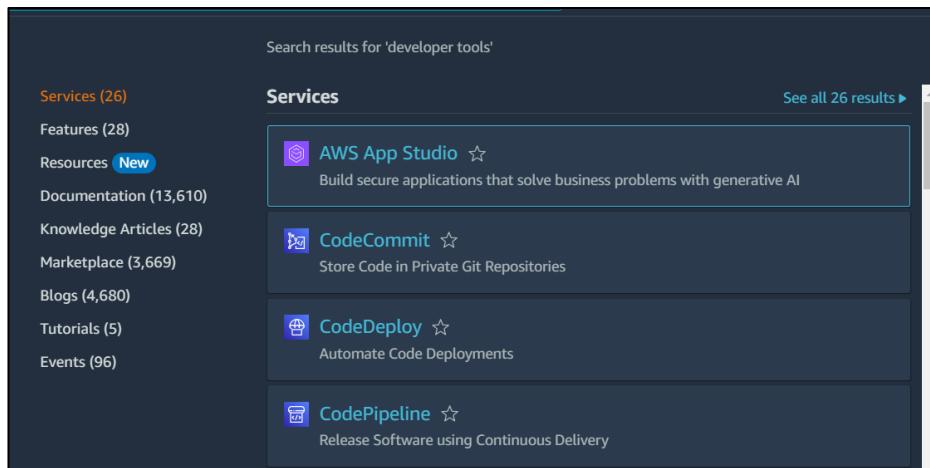
Environment overview

Health	Environment ID
Grey	e-dwvx2qmicx
Domain	Application name
Application1-env.eba-8p8z2p3c.us-east-1.elasticbeanstalk.com	Application1

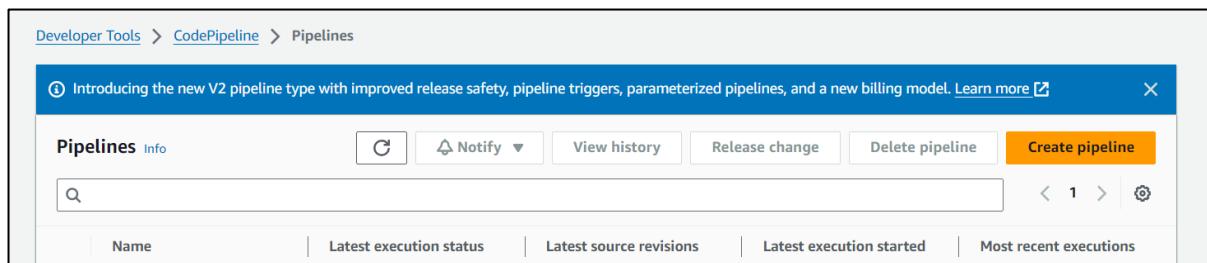
11. After clicking on domain link the page given below will open:



12. Now we will create a pipeline. Go to services and select CodePipeline.



13. Select create pipeline.



14. In step 1 give the pipeline name

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

15. In Add source stage select GitHub version2 as the source provider and then connect your AWS to GitHub account and if there is a connection which already exists then select that.

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

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

16. After connecting to GitHub select the repository name and default branch

The screenshot shows the 'Connection' configuration page for GitHub. At the top, there's a search bar with the ARN 'arn:aws:codeconnections:us-east-1:025066268342:connection/22f7e0a1-83f...' and a 'Connect to GitHub' button. Below this, a green box indicates 'Ready to connect' with the message 'Your GitHub connection is ready for use.' In the 'Repository name' section, a search bar contains 'Nayaab-Jindani05/aws-codepipeline-s3-codedeploy-linux'. A note below says, 'You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project''. In the 'Default branch' section, a search bar contains 'master'. A note below says, 'Default branch will be used only when pipeline execution starts from a different source or manually started.'

17. Skip the build stage and go to deploy stage

The screenshot shows the 'Add build stage' configuration screen. It's step 3 of 5. The 'Build - optional' section has a 'Build provider' field with a note: 'This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.' At the bottom, there are four buttons: 'Cancel', 'Previous', 'Skip build stage' (which is highlighted in orange), and 'Next'.

18. In the deploy stage select AWS Elastic Beanstalk as the deploy provider and also select the application name, environment name created earlier.

Deploy

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

Region

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

No more than 100 characters

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

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.

Configure automatic rollback on stage failure

19. In the review stage check all the settings that have been done and select create pipeline.

Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

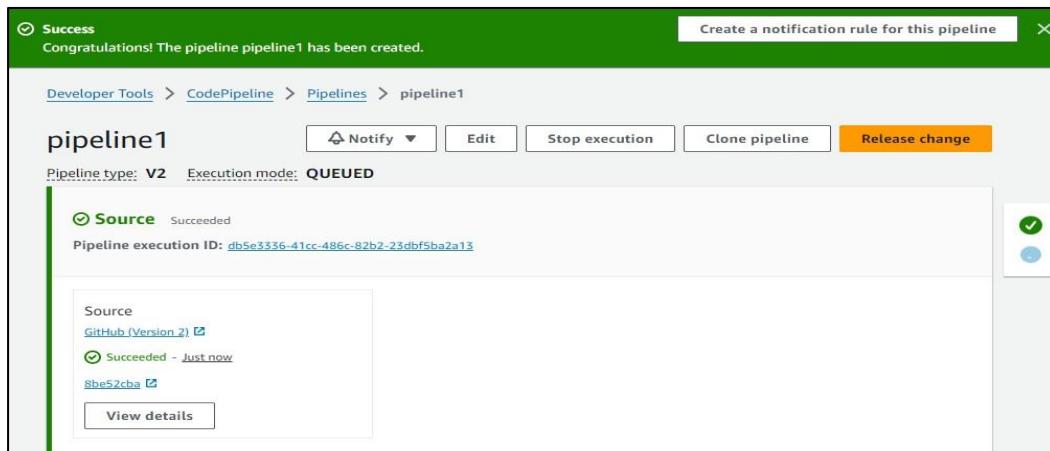
Deploy action provider
AWS Elastic Beanstalk

ApplicationName
Application1

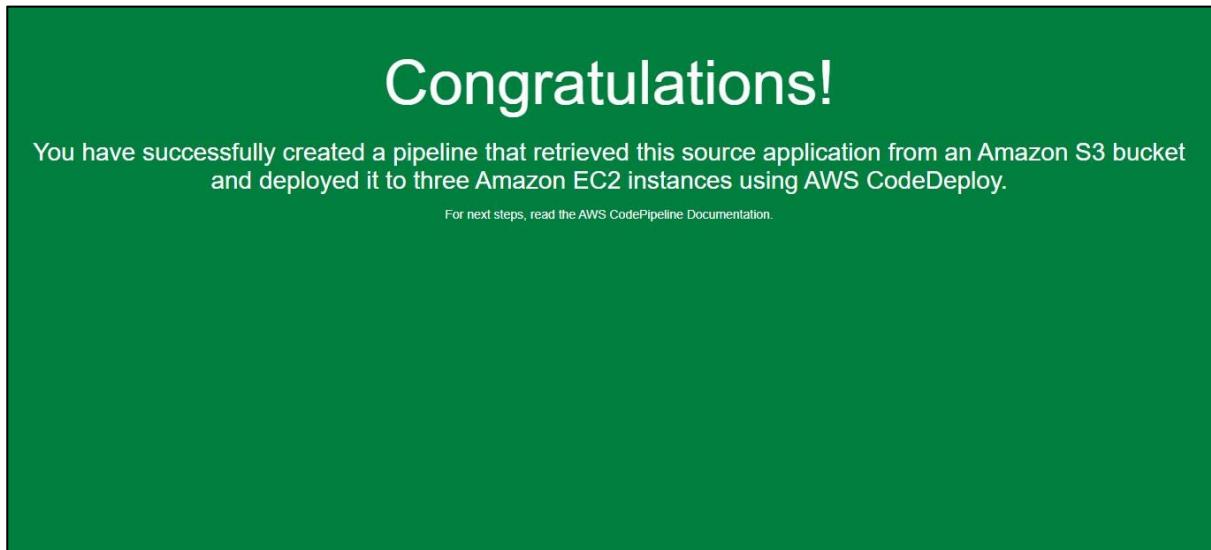
EnvironmentName
Application1-env

Configure automatic rollback on stage failure
Disabled

20. This screen means that pipeline creation is successful



21. Now we can select the URL and it will open a sample website that we have created.



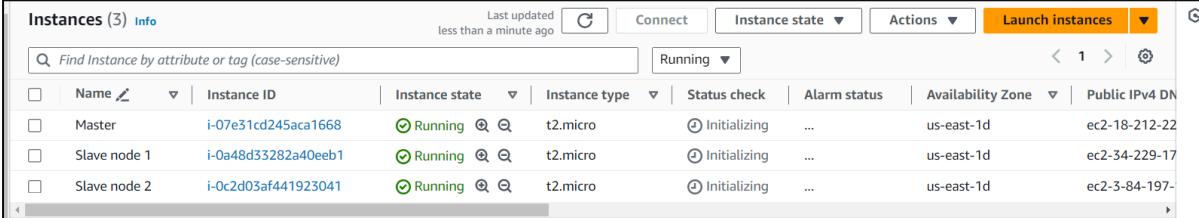
Advance DevOps

Experiment 3

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

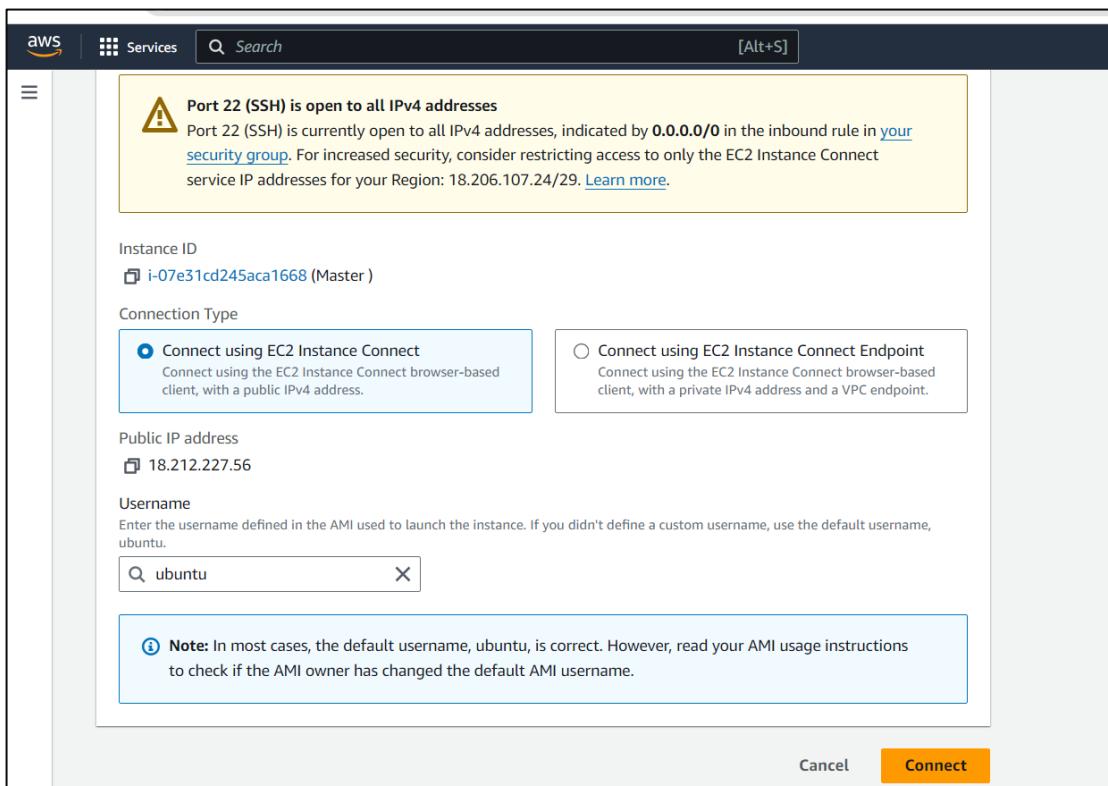
Steps:

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



Name	Value	Unit	Timestamp
Master	100.00	%	2023-09-15T12:00:00Z
Slave node 1	0.00	%	2023-09-15T12:00:00Z
Slave node 2	0.00	%	2023-09-15T12:00:00Z

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



3. Edit the inbound rule to allow all traffic

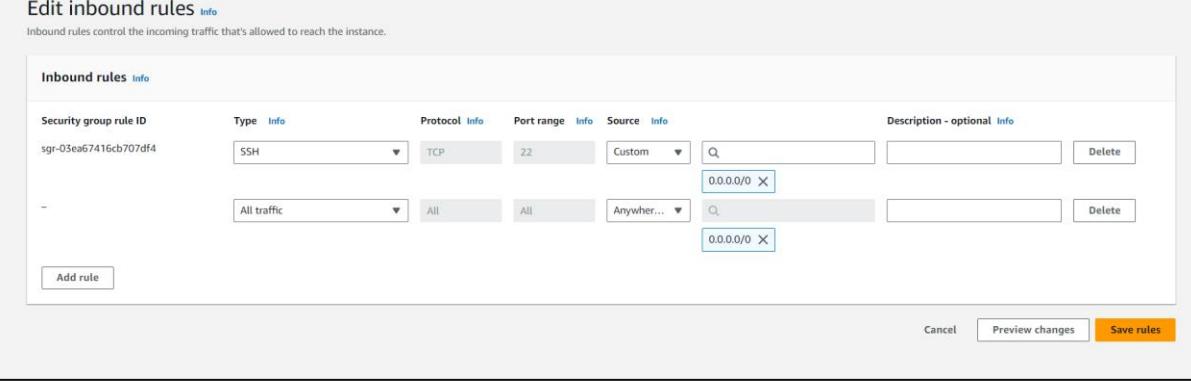
Edit inbound rules Info

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-05ea67416cb707df4	SSH	TCP	22	Custom	0.0.0.0 X
-	All traffic	All	All	Anywhere...	0.0.0.0 X

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)



4. Docker installation:

This step has to be performed on all the 3 instances.

The following command has to be run:

```
yum install docker -y
```

```
[ec2-user@ip-172-31-24-23 ~]$ sudo su
[root@ip-172-31-24-23 ec2-user]# yum install docker -y
Last metadata expiration check: 0:02:51 ago on Sun Sep 8 08:49:45 2024.
Dependencies resolved.
=====
Package           Architecture   Version      Repository  Size
=====
Installing:
docker            x86_64        25.0.6-1.amzn2023.0.2  amazonlinux 44 M
Installing dependencies:
containerd         x86_64        1.7.20-1.amzn2023.0.1  amazonlinux 35 M
iptables          x86_64        1.8.8-3.amzn2023.0.2  amazonlinux 401 K
iptables-nft      x86_64        1.8.8-3.amzn2023.0.2  amazonlinux 183 K
libcgroup          x86_64        3.0-1.amzn2023.0.1   amazonlinux 75 K
libnetfilter_conntrack x86_64        1.0.6-2.amzn2023.0.2  amazonlinux 58 K
libnftnl           x86_64        1.0.1-19.amzn2023.0.2  amazonlinux 30 K
libnftnl           x86_64        1.2.2-2.amzn2023.0.2  amazonlinux 84 K
pigz              x86_64        2.5-1.amzn2023.0.3   amazonlinux 83 K
```

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

```
[root@ip-172-31-24-23 ec2-user]# systemctl start docker
[root@ip-172-31-24-23 ec2-user]#
```

6. Kubernetes installation:

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

Debian-based distributions
Red Hat-based distributions

[Without a package manager](#)

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
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/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

3. Install kubelet, kubeadm and kubectl:

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

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

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

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

```
[root@ip-172-31-24-23 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.
=====
Transaction Summary
=====
Install  10 Packages

Total download size: 51 M
Installed size: 270 M
Downloading Packages:
(1/10): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm           459 kB/s | 24 kB     00:00
(2/10): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm                 1.4 MB/s | 30 kB     00:00
=====
=====
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                         7/10
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                  7/10
Installing : kubelet-1.31.0-150500.1.1.x86_64                                8/10
Running scriptlet: kubelet-1.31.0-150500.1.1.x86_64                          8/10
Installing : kubeadm-1.31.0-150500.1.1.x86_64                               9/10
Installing : kubectl-1.31.0-150500.1.1.x86_64                            10/10
Running scriptlet: kubectl-1.31.0-150500.1.1.x86_64                         10/10
Verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                      1/10
Verifying   : kubelet-1.31.0-150500.1.1.x86_64                        2/10
Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64            3/10
Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64          4/10
Verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                5/10
Verifying   : socat-1.7.4.2-1.amzn2023.0.2.x86_64                           6/10
Verifying   : cri-tools-1.31.0-150500.1.1.x86_64                         7/10
Verifying   : kubeadm-1.31.0-150500.1.1.x86_64                         8/10
Verifying   : kubectl-1.31.0-150500.1.1.x86_64                         9/10
Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64                     10/10
=====
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64      kubeadm-1.31.0-150500.1.1.x86_64
  kubelet-1.31.0-150500.1.1.x86_64                 kubelet-1.31.0-150500.1.1.x86_64      kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
  socat-1.7.4.2-1.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-24-23 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-24-23 ec2-user]#
```

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

```
[root@ip-172-31-24-23 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
kubernetes                             Kubernetes
```

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

```
[root@ip-172-31-24-23 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
  [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-24-23 ec2-user]# ^C
[root@ip-172-31-24-23 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
  [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0908 09:11:46.376173 28683 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-24-23.ec2.internal kubernetes kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.24.23]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key

[root@ip-172-31-82-191 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0913 15:43:09.666583 2255 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unavailable desc = connection error: desc = "transport: Error while dialing: dial unix /var/run/containerd/containerd.sock: connect: no such file or directory"
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-82-191 ec2-user]# ^C
[root@ip-172-31-82-191 ec2-user]# systemctl start docker
[root@ip-172-31-82-191 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0913 15:44:18.815161 2565 checks.go:946] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-82-191.ec2.internal kubernetes kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.82.191]
```

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

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

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

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

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

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

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

```
kubeadm join 172.31.82.191:6443 --token 8450pt.tdprcovwa61rqyo1 \
--discovery-token-ca-cert-hash sha256:b11f191f3df19a2e9112a5c19b4461bf feadd8b5be8625ad8451019aecc043c
```

11. We can check the nodes that have joined the cluster using kubectl get nodes. Right now, there is only one node which is the master node.

```
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME                  STATUS    ROLES      AGE      VERSION
ip-172-31-85-89.ec2.internal   NotReady  control-plane  72s    v1.26.0
```

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

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

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

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

```
[root@ip-172-31-34-212 ec2-user]# kubectl get nodes
NAME                  STATUS    ROLES      AGE      VERSION
ip-172-31-34-212.ec2.internal   Ready    control-plane  18m    v1.31.1
ip-172-31-37-229.ec2.internal   Ready    <none>     13m    v1.31.1
ip-172-31-45-98.ec2.internal   Ready    <none>     14m    v1.31.1
[root@ip-172-31-34-212 ec2-user]#
```

Conclusion:

This experiment enabled the creation of a Kubernetes cluster and the successful joining of all 3 nodes using various commands. Errors during initialization can be handled in two ways: 1. By ignoring the errors, or 2. By changing the instance type to t3.medium or t3.large if the issue is related to insufficient memory space or CPU resources. Also, it is to be ensured that the inbound rules and outbound rules allow all traffic or else it leads to connectivity issues between master node and worker nodes.

Advance DevOps

Experiment 4

Aim:

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

Steps:

1. Create a key pair

The screenshot shows the AWS Key Pairs list page. It displays one key pair entry:

Name	Type	Created	Fingerprint	ID
test	rsa	2024/08/28 09:58 GMT+5:30	36:80:ad:4d:22:3d:50:d3:8f:77:bb:47:ea:3b:ace2:7...	key-07334c8c589...

The screenshot shows the 'Create key pair' wizard. Step 1: Key pair details. The form includes fields for Name (set to 'test1'), Key pair type (set to RSA), Private key file format (.pem selected), and Tags - optional (empty). At the bottom are 'Cancel' and 'Create key pair' buttons.

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

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

The screenshot shows the AWS Instances list page. It displays one instance entry:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Instance1	i-0e052e86e6a7d7725	Running	t3.medium	2/2 checks pa	View alarms +	us-east-1d	ec2-98-81-152-195.co...	98.81.152

3. Now edit the inbound rules to allow ssh

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0f9dae3bbadfb973	All traffic SSH	All TCP	All 22	Custom Anywhere...	

Add rule Delete Cancel Preview changes Save rules

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

```
Dell@DESKTOP-OVNIAIM MINGW64 ~/Downloads (master)
$ chmod 400 test1.pem
```

5. Now use this command on the terminal: ssh -i <keyname>.pem
ubuntu@<public_ip_address> and replace

- Keyname with the name of your key pair, in our case test1.
- As we are using amazon Linux instead of ubuntu we will have ec2-user
- Replace public ip address with its value. Go to your instance and scroll down you will find the public ip address there.

```
Dell@DESKTOP-OVNIAIM MINGW64 ~/Downloads (master)
$ ssh -i "test1.pem" ec2-user@ec2-44-204-14-37.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-14-37.compute-1.amazonaws.com (44.204.14.37)' can't be established.
ED25519 key fingerprint is SHA256:CtxhAZnv4MFbUai03z96MQzMKK6JxuN/nwlIerDSazI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-14-37.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-81-24 ~]$
```

6. Docker installation:

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

```
[root@ip-172-31-81-24 ec2-user]# sudo yum install docker -y
Last metadata expiration check: 0:01:25 ago on Sat Sep 14 06:42:34 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |             |            |       |
| docker            | x86_64       | 25.0.6-1.amzn2023.0.2 | amazonlinux | 44 M  |
| Installing dependencies: |             |             |            |       |
| containerd        | x86_64       | 1.7.20-1.amzn2023.0.1 | amazonlinux | 35 M  |
| libatomic         | x86_64       | 1.1.1-1.amzn2023.0.2 | amazonlinux | 404 k |
| libcap             | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
| libcgroup          | x86_64       | 3.0-1.amzn2023.0.1   | amazonlinux | 75 k  |
| libnetfilter_conntrack | x86_64       | 1.0-8-2.amzn2023.0.2 | amazonlinux | 58 k  |
| libnfnetlink       | x86_64       | 1.0-19-2.amzn2023.0.2 | amazonlinux | 38 k  |
| libtunctl          | x86_64       | 1.2.2-2.amzn2023.0.2 | amazonlinux | 84 k  |
| pigrz              | x86_64       | 2.5-1.amzn2023.0.3   | amazonlinux | 83 k  |
| runc               | x86_64       | 1.1.13-1.amzn2023.0.1| amazonlinux | 3.2 M  |

Transaction Summary
=====
| Install 10 Packages |
Total download size: 84 M
Installed size: 317 M
Downloaded Packages:
(1/10): iptables-libns-1.8.8-3.amzn2023.0.2.x86_64.rpm           4.3 MB/s | 401 kB  00:00
(2/10): libatomic-1.1.1-1.amzn2023.0.2.x86_64.rpm                 3.3 MB/s | 183 kB  00:00
(3/10): libcgroup-1.0-1.amzn2023.0.1.x86_64.rpm                  1.6 MB/s | 75 kB  00:00
(4/10): libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64.rpm    1.6 MB/s | 58 kB  00:00
(5/10): libnfnetlink-1.0-1-19.amzn2023.0.2.x86_64.rpm            940 kB/s | 30 kB  00:00
(6/10): libtunctl-1.2.2-2.amzn2023.0.2.x86_64.rpm                1.0 MB/s | 84 kB  00:00
(7/10): pigrz-2.5-1.amzn2023.0.3.x86_64.rpm                   2.1 MB/s | 83 kB  00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm                 19 MB/s | 3.2 MB  00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm            27 MB/s | 35 MB  00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm              27 MB/s | 44 MB  00:01

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/10
  Writing  : 1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64

=====
| 49 MB/s | 84 MB  00:01 |
| 1/10   |
| 2/10   |
| 3/10   |
| 4/10   |
| 5/10   |
| 6/10   |
| 7/10   |
| 8/10   |
| 9/10   |
| 10/10  |
```

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

cd /etc/docker

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

{

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

}

EOF

sudo systemctl enable docker sudo

systemctl daemon-reload sudo

systemctl restart docker

```
[root@ip-172-31-81-24 ec2-user]# cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

8. Kubernetes installation:

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

Debian-based distributions

Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
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/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

3. Install kubelet, kubeadm and kubectl:

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

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

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

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

```
[root@ip-172-31-81-24 docker]# sudo setenforce 0
sudo sed -i '/^SELINUX=enforcing/ { s/SELINUX=enforcing/SELINUX=permissive/; p }' /etc/selinux/config
[root@ip-172-31-81-24 docker]# cat <>EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core/stable/v1.31/rpm/repo/epm
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core/stable/v1.31/rpm/repo/epm/repo/epm.key
exclude=kubelet kubelet kubelet cri-tools kubernetes-cni
EOF
[[kubernetes]]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core/stable/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core/stable/v1.31/rpm/repo/epm/repo/epm.key
exclude=kubelet kubelet kubelet cri-tools kubernetes-cni
[[root@ip-172-31-81-24 docker]# sudo yum install -y kubelet kubeadm kubectl --disallow-excludes=kubernetes
Kubernetes
Dependencies resolved.

Transaction Summary
=====================================================================
Install 9 Packages
Total download size: 51 M
Installed size: 269 M
(0/9): kubelet-1.31.1-150500.1.1.x86_64.rpm
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm
(2/9): libnetfilter_cthelper-0.0-21.amzn2023.0.2.x86_64.rpm
(3/9): libnetfilter_ctqueue-0.5-2.amzn2023.0.2.x86_64.rpm
(4/9): contract-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm
(5/9): libnl-3-1.31.1-150500.1.1.x86_64.rpm
(6/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm
(7/9): kubecfg-1.31.1-150500.1.1.x86_64.rpm
(8/9): kubenetes-cni-1.5.1-150500.1.1.x86_64.rpm
(9/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm

Total
Kubernetes
Importing GPG key 0xA296436:
-----
Using existing key 0xA296436
----- Project <https://github.com/opensuse/obs-build.opensuse.org>
-----
[travis@ip-172-31-81-24 ~]$ curl https://github.com/opensuse/obs-build.opensuse.org/
-----
```

```
(6/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm
(7/9): kubelet-1.31.1-150500.1.1.x86_64.rpm
(8/9): kubenetes-cni-1.5.1-150500.1.1.x86_64.rpm
(9/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm

Total
Kubernetes
Importing GPG key 0xA296436:
----- Using existing key 0xA296436
----- Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 3346 54DA 9A29 6436
----- From : https://pkgs.k8s.io/core/stable/v1.31/rpm/repo/epm/repo/epm.key
Key imported successfully
Running transaction test
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : .
  Installing  : kubernetes-cni-1.5.1-150500.1.1.x86_64
  Installing  : cri-tools-1.31.1-150500.1.1.x86_64
  Installing  : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  Installing  : libnetfilter_cthelper-0.0-21.amzn2023.0.2.x86_64
  Installing  : libnetfilter_ctqueue-0.5-2.amzn2023.0.2.x86_64
  Installing  : contract-tools-1.4.6-2.amzn2023.0.2.x86_64
  Installing  : libnl-3-1.31.1-150500.1.1.x86_64
  Installing  : kubeadm-1.31.1-150500.1.1.x86_64
  Installing  : kubelet-1.31.1-150500.1.1.x86_64
  Installing  : kubecfg-1.31.1-150500.1.1.x86_64
  Installing  : kubernetes-cni-1.5.1-150500.1.1.x86_64
  Installing  : libnetfilter_ctqueue-0.5-2.amzn2023.0.2.x86_64
  Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64
  Verifying   : cri-tools-1.31.1-150500.1.1.x86_64
  Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  Verifying   : libnetfilter_cthelper-0.0-21.amzn2023.0.2.x86_64
  Verifying   : libnetfilter_ctqueue-0.5-2.amzn2023.0.2.x86_64
  Verifying   : contract-tools-1.4.6-2.amzn2023.0.2.x86_64
  Verifying   : libnl-3-1.31.1-150500.1.1.x86_64
  Verifying   : kubeadm-1.31.1-150500.1.1.x86_64
  Verifying   : kubelet-1.31.1-150500.1.1.x86_64
  Verifying   : kubecfg-1.31.1-150500.1.1.x86_64
  Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64

Installed:
  contract-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64      kubeadm-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64      kubernetes-cni-1.5.1-150500.1.1.x86_64      libnetfilter_cthelper-0.0-21.amzn2023.0.2.x86_64
  libnetfilter_ctqueue-0.5-2.amzn2023.0.2.x86_64

Completed:
[root@ip-172-31-81-24 docker]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-81-24 docker]#
```

- 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
 /etc/sysctl.conf sudo
 sysctl -p

```
[root@ip-172-31-81-24 docker]# sudo swapoff -a
[root@ip-172-31-81-24 docker]# echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[root@ip-172-31-81-24 docker]# sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[root@ip-172-31-81-24 docker]#
```

10. Initializing kubecluster: sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
  [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with `--v=5` or higher
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
```

```
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 172.31.81.24:6443 --token 4a91z3.yz6rwmmkf9yncyd2 \
  --discovery-token-ca-cert-hash sha256:3404bd1bcd9cf90a003673f622d1672acb4c6ce7c15c4738c80a0a1560fe70d
[root@ip-172-31-81-24 docker]# |
```

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

```
[root@ip-172-31-81-24 docker]# mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-81-24 docker]# |
```

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

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

```
[root@ip-172-31-81-24 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
[root@ip-172-31-81-24 docker]# |
```

13. Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment kubectl apply -f https://k8s.io/examples/application/deployment.yaml

```
[root@ip-172-31-81-24 docker]# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
[root@ip-172-31-81-24 docker]# |
```

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

```
[root@ip-172-31-81-24 docker]# kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-8jdlf   0/1     Pending   0          18s
```

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

```
Name:           nginx-deployment-d556bf558-gw8v8
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          app=nginx
                 pod-template-hash=d556bf558
Annotations:    <none>
Status:         Pending
IP:
IPs:
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:

Conditions:
  Type      Status
  PodScheduled  False
Volumes:
  kube-api-access-f9k9s:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
QoS Class:      BestEffort
Node-Selectors: <none>
Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Warning  FailedScheduling  114s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane-}. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning  FailedScheduling  3m18s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane-}. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Use the below command to remove taints

```
[root@ip-172-31-81-24 docker]# kubectl taint nodes --all node-role.kubernetes.io/control-plane-
```

16. Check the pod status

```
[root@ip-172-31-81-24 docker]# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   1 (6s ago)  90s
```

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

```
[root@ip-172-31-81-24 docker]# kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

18. Verify your deployment

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

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

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

If you see your nginx server name and response code is 200 then the deployment was successful.

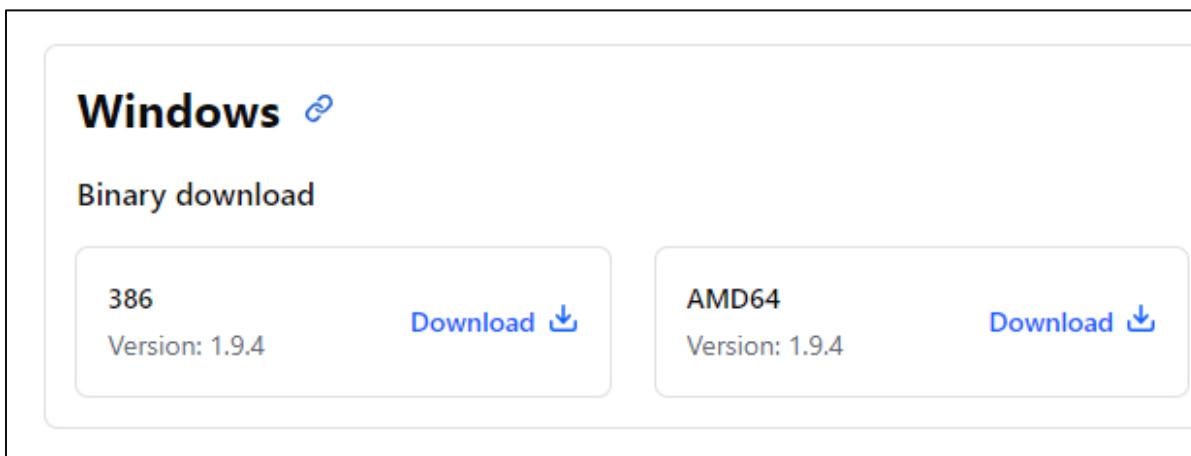
```
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 14 Sep 2024 06:54:21 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

Conclusion: In this experiment we created an ec2 instance, enabled ssh by editing the inbound rules. After that we installed docker and Kubernetes and configured internet options to allow bridging. Once this setup got completed, we added a common networking plugin called flannel. Once the cluster started running we deployed nginx server on this cluster and verified deployment.

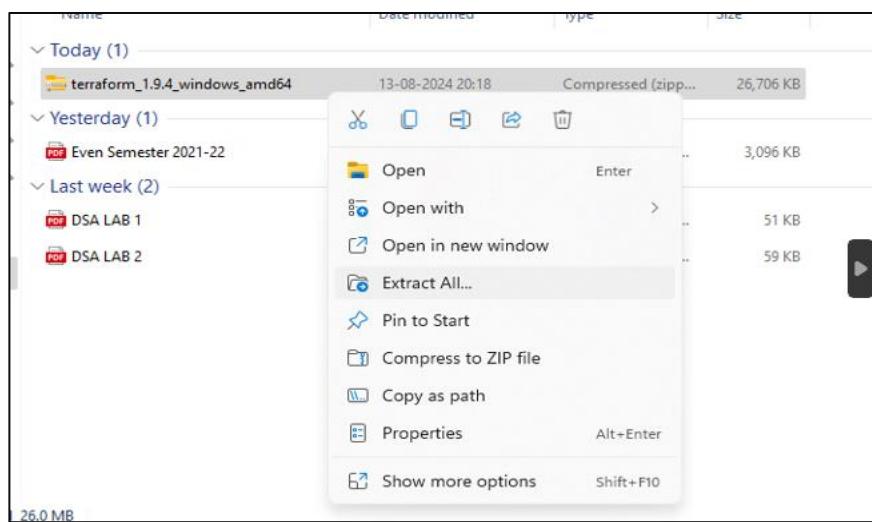
Advance DevOps

Experiment 5

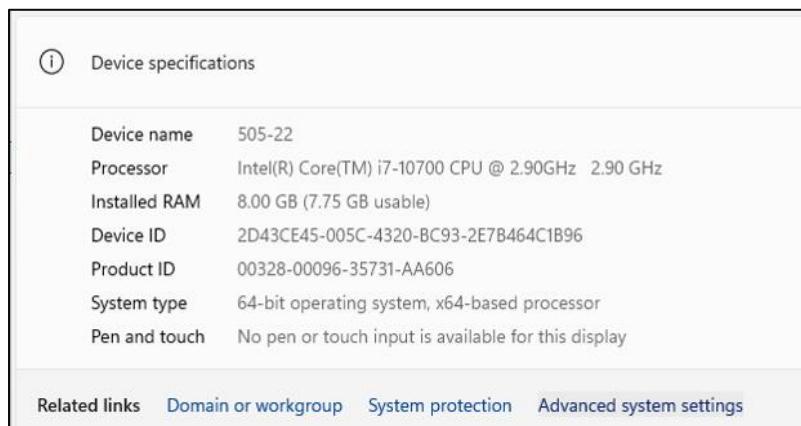
Step 1: To install terraform first download the Terraform CLI Utility for Windows from terraform official website www.terraform.io/downloads.html



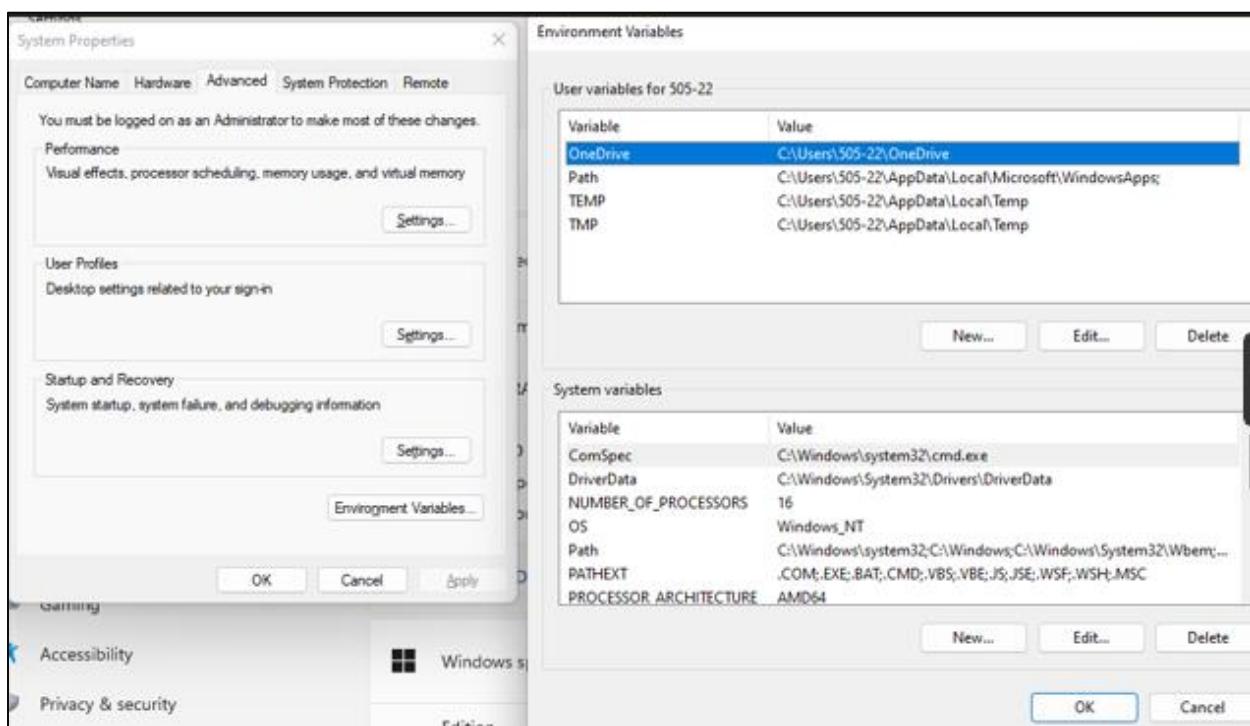
Step 2: After the zip file is downloaded, we will right click on it and click on extract all files.



Step 3: Go to advance system settings in settings



Step 4: In advance settings select environment variables and set the path of Terraform.



Step 5: In order to check whether terraform was successfully installed we have to run the command “terraform -v” which will give us the version of our terraform and hence terraform has been downloaded correctly.

```
C:\Users\DELL\Downloads\terraform_1.9.4_windows_amd64>terraform -v
Terraform v1.9.4
on windows_amd64
```

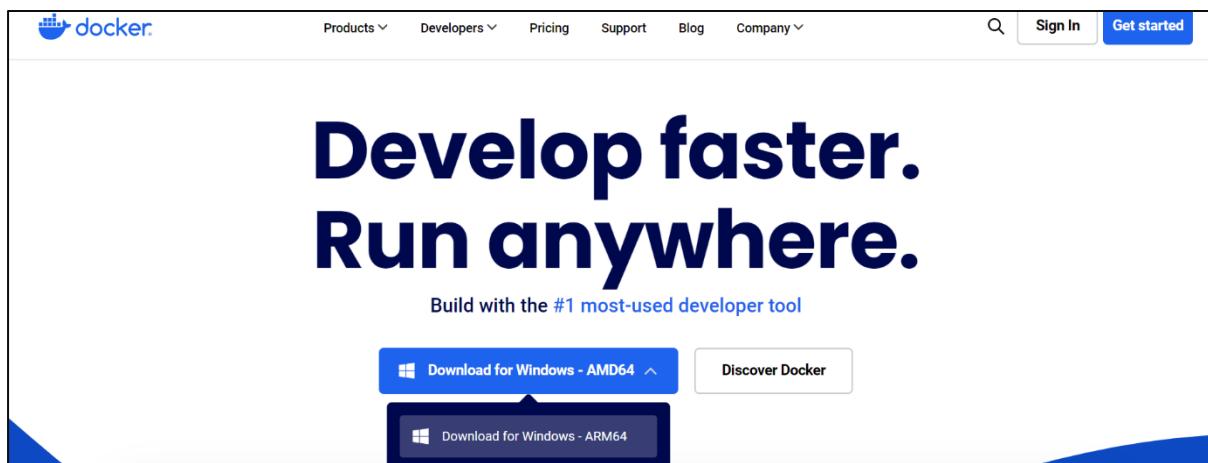
Advance DevOps

Experiment 6

Creating docker image using terraform

Steps:

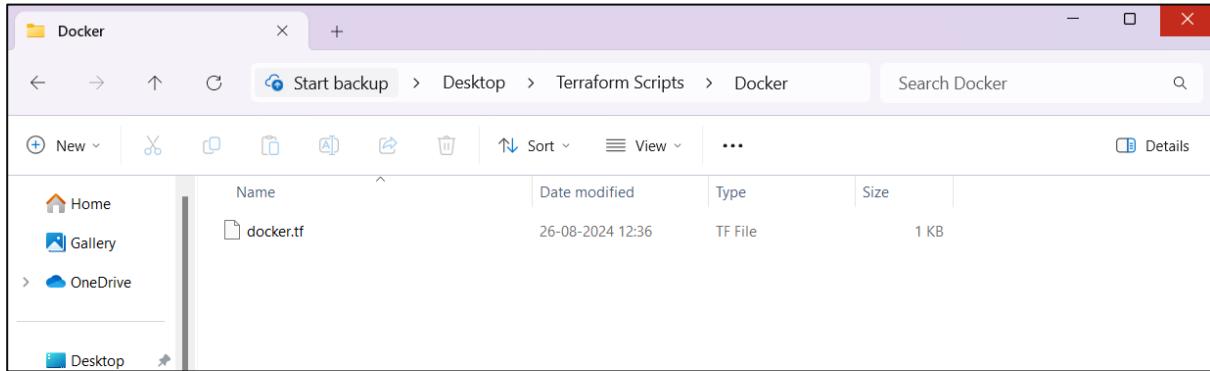
1. Firstly, we will have to download and install docker from its official website.



2. To check if docker has been correctly installed we can use the command “docker – version”. If it returns the version then installation has been successful.

A screenshot of a Windows PowerShell window titled 'Windows PowerShell (x86)'. The command 'docker --version' is run, and the output shows 'Docker version 27.0.3, build 7d4bcd8'. The window has a dark theme.

3. Now we will create a folder “Terraform Scripts” and inside this folder we will create another folder “Docker” where our docker script will be saved. Using an editor create a file named docker and save it with .tf extension inside the above-mentioned directory structure.



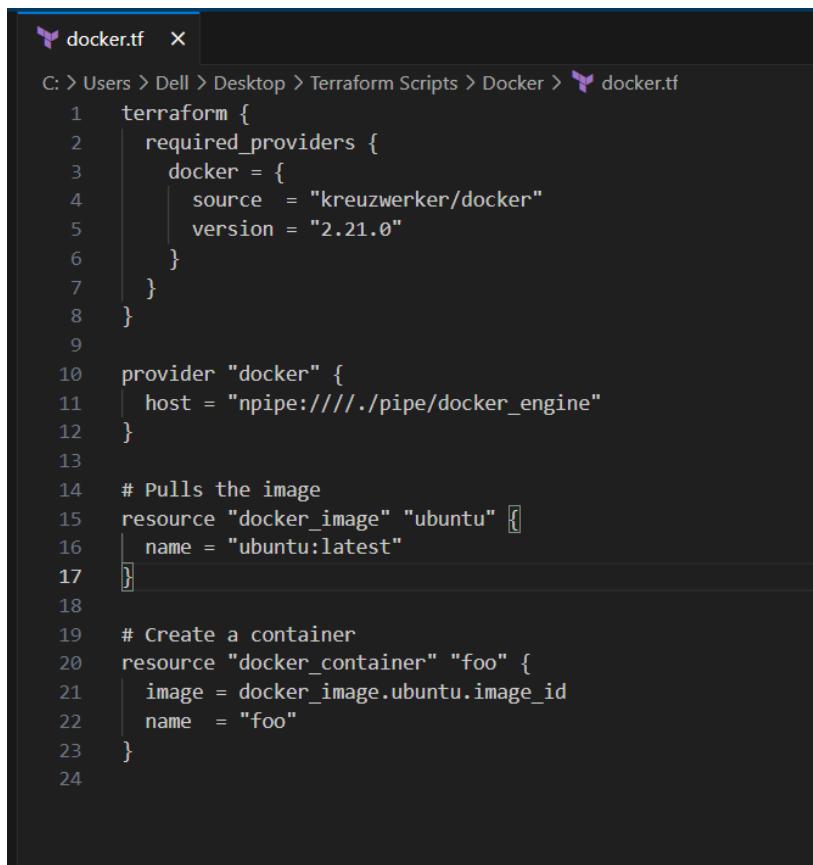
4. The docker.tf file will contain the following:

```
terraform {  
    required_providers {  
        docker = {  
            source  = "kreuzwerker/docker"  
            version = "2.21.0"  
        }  
    }  
}  
  
provider "docker" {  
    host = "npipe:///./pipe/docker_engine"  
}  
  
# Pulls the image  
resource "docker_image" "ubuntu" {  
    name = "ubuntu:latest"  
}
```

```
# Create a container

resource "docker_container" "foo" {
    image = docker_image.ubuntu.image_id
    name  = "foo"
}
```

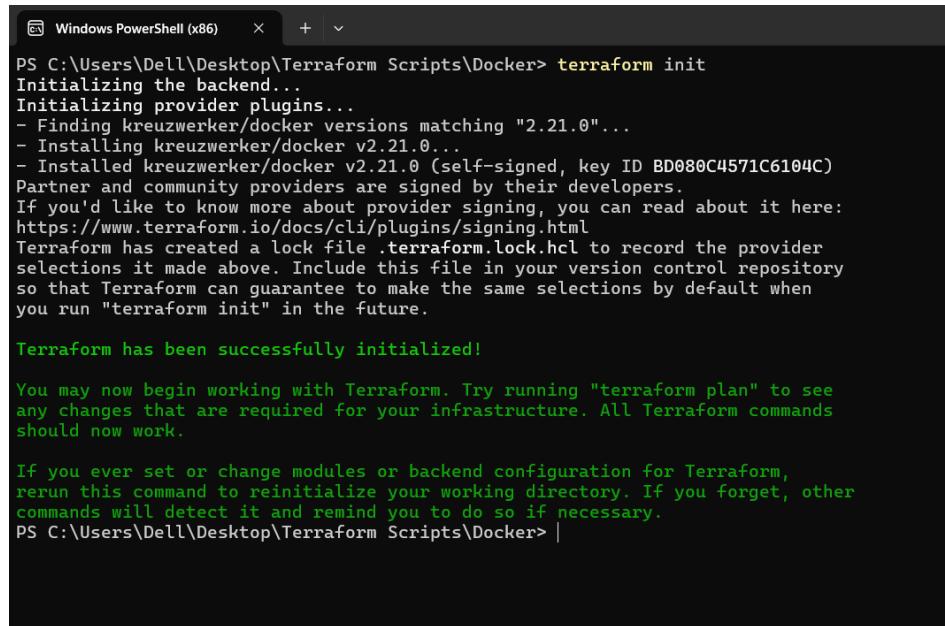
Ensure correct block formatting to avoid errors.



```
docker.tf x

C: > Users > Dell > Desktop > Terraform Scripts > Docker > docker.tf
1  terraform {
2      required_providers {
3          docker = {
4              source  = "kreuzwerker/docker"
5              version = "2.21.0"
6          }
7      }
8  }
9
10 provider "docker" {
11     host = "npipe:////./pipe/docker_engine"
12 }
13
14 # Pulls the image
15 resource "docker_image" "ubuntu" {
16     name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21     image = docker_image.ubuntu.image_id
22     name  = "foo"
23 }
24
```

5. Now execute “terraform init” command in power shell. This command will initialize your working directory and download the necessary provider plugins. (Make sure you are in the “Docker” directory)



```

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

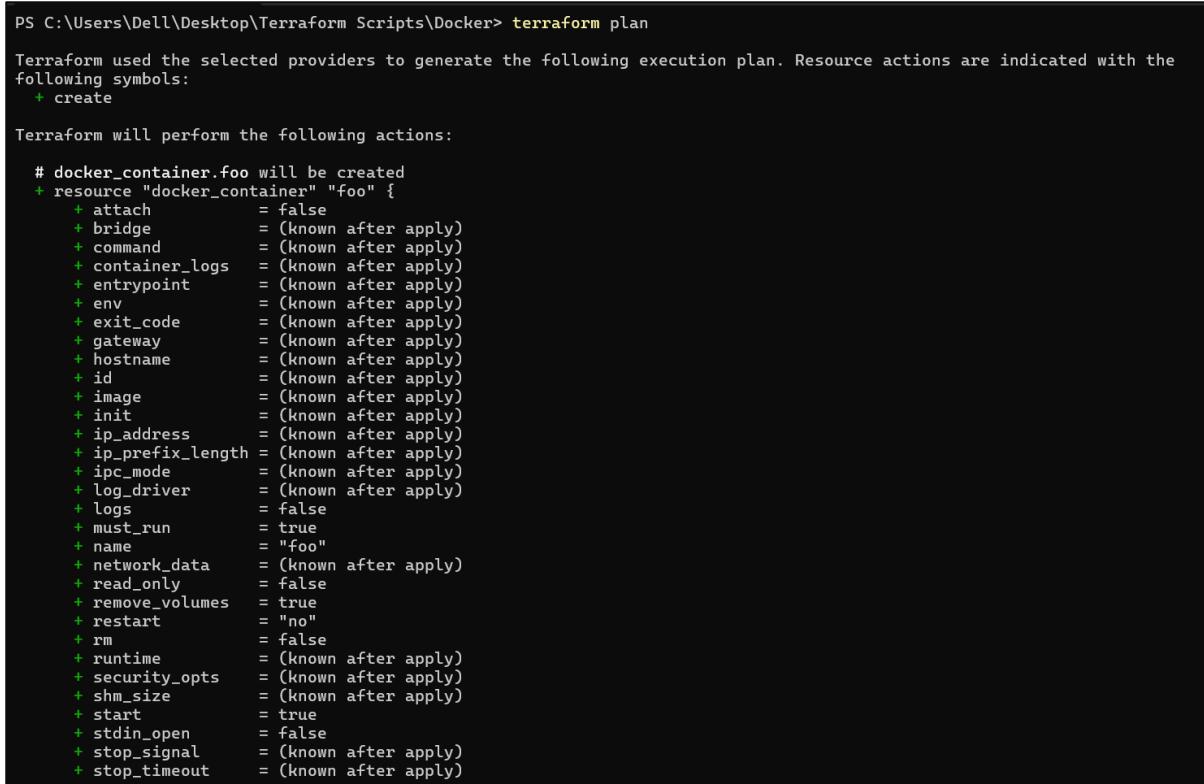
Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\DELL\Desktop\Terraform Scripts\Docker>

```

6. Next execute “terraform plan” command. The terraform plan command is used to create an execution plan for terraform. This plan shows you what changes terraform will make to your infrastructure based on your current configuration files and the state of your existing infrastructure.



```

PS C:\Users\DELL\Desktop\Terraform Scripts\Docker> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach                  = false
    + bridge                  = (known after apply)
    + command                 = (known after apply)
    + container_logs          = (known after apply)
    + 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)
}
```

7. After this execute the “terraform apply” command. The terraform apply command executes the actions proposed in terraform plan. It is used to deploy your infrastructure.

In this step I got an error saying “container exited immediately” which indicates that the Docker container created by terraform configuration exited immediately after being started. In order to keep the container running I updated the script file with the following code:

```
command = ["sleep", "infinity"] # Keeps the container running
```

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

8. After updating the script, the error will be solved and we can again run apply command.

```
PS C:\Users\DELL\Desktop\Terraform Scripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach           = false
  + bridge           = (known after apply)
  + command          = [
    + "sleep",
    + "infinity",
  ]
  + 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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  + 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
}
```

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

+ healthcheck (known after apply)

+ labels (known after apply)
}

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 0s [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]

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

Docker image has been created.

9. The image created can be checked by using docker images command. It will show the repository, tag, image id, time of creation and size as shown below.

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

10. Using terraform destroy we can destroy the container.

```
PS C:\Users\DELL\Desktop\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach                  = false -> null
    - command                 = [
        - "sleep",
        - "infinity",
    ] -> null
    - cpu_shares              = 0 -> null
    - dns                      = [] -> null
    - dns_opts                 = [] -> null
    - dns_search                = [] -> null
    - entrypoint               = [] -> null
    - env                      = [] -> null
    - gateway                  = "172.17.0.1" -> null
    - group_add                = [] -> null
    - hostname                 = "334f576e574f" -> null
    - id                       = "334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81" -> null
    - image                     = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - init                      = false -> null
    - ip_address                = "172.17.0.2" -> null
    - ip_prefix_length          = 16 -> null
    - ipc_mode                  = "private" -> null
    - links                     = [] -> null
    - log_driver                = "json-file" -> null
    - log_opts                  = {} -> null
    - logs                      = false -> null
    - max_retry_count           = 0 -> null
    - memory                    = 0 -> null
    - memory_swap                = 0 -> null
    - must_run                  = true -> null
    - name                      = "foo" -> null
    - network_data              = [
```

```
    - rm                      = false -> null
    - runtime                 = "runc" -> null
    - security_opts            = [] -> null
    - shm_size                 = 64 -> null
    - start                    = true -> null
    - stdio_open                = false -> null
    - stop_timeout              = 0 -> null
    - storage_opts              = {} -> null
    - sysctls                  = {} -> null
    - tmpfs                     = {} -> null
    - tty                      = false -> null
  ] # (8 unchanged attributes hidden)
}

# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
    - id                       = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
    - image_id                 = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - latest                   = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - name                      = "ubuntu:latest" -> null
    - repo_digest               = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
}

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

Now when we run docker images we will see that the container information is not visible that means it has been deleted successfully.

```
PS C:\Users\DELL\Desktop\Terraform Scripts\Docker> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
PS C:\Users\DELL\Desktop\Terraform Scripts\Docker> |
```

Advance DevOps

Experiment 7

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

Steps:

- Firstly, we will ensure whether docker is installed or not by running docker -v in the command prompt.

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

- Run docker login command and add your username and password for docker.

```
C:\Users\DELL>docker login
Authenticating with existing credentials...
Stored credentials invalid or expired
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username (dimple866): dimple866
Password:
Login Succeeded
```

- Run docker pull SonarQube command to install SonarQube image.

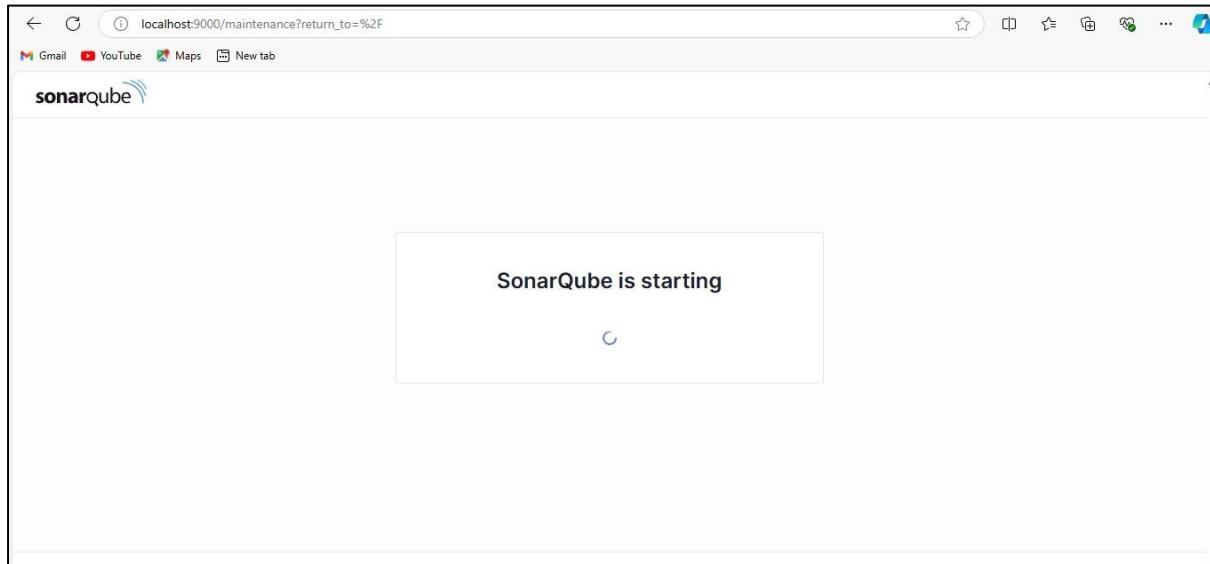
```
C:\Users\DELL>docker pull sonarqube
Using default tag: latest
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

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

- Run docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Command to run the sonarqube.

```
C:\Users\DELL>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
ac1f985dedebc00a642a4c69a502d611389e8f9fa46610febe75aa5021767cab
```

5. Once the container is running go to your web browser and check status of SonarQube at port 9000.



6. Once SonarQube is started it will redirect you to login page. The login and password both for SonarQube is 'admin'

A screenshot of the SonarQube login page. At the top, there is a logo with the word 'sonar' and a stylized blue bird icon. Below the logo, the text 'Log in to SonarQube' is displayed. There are two input fields: 'Login *' and 'Password *'. Underneath the password field is a link 'Forgot your password?'. At the bottom of the form are two buttons: 'Go back' and a blue 'Log in' button.

7. Change the password for your SonarQube account.

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

8. After changing the password, you will be directed to this screen. Click on Create a Local Project.

The screenshot shows the SonarQube interface for creating a local project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, a section titled "How do you want to create your project?" lists several import options:

- Import from Azure DevOps (Setup button)
- Import from Bitbucket Cloud (Setup button)
- Import from Bitbucket Server (Setup button)
- Import from GitHub (Setup button)
- Import from GitLab (Setup button)

Below these options, a message says, "Are you just testing or have an advanced use-case? Create a local project." There is a "Create a local project" button. In the bottom right corner, a dark overlay box contains the text "Get the most out of SonarQube!" followed by a brief description of SonarLint and two buttons: "Learn More" and "Dismiss".

9. Add name of the project and project key and select the main branch name and click on next.

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](#)

10. Set up the project as required and click on create.

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

Use the global setting

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

Reference branch
Choose a branch as the baseline for the new code.
Recommended for projects using feature branches.

[Back](#) [Create project](#)

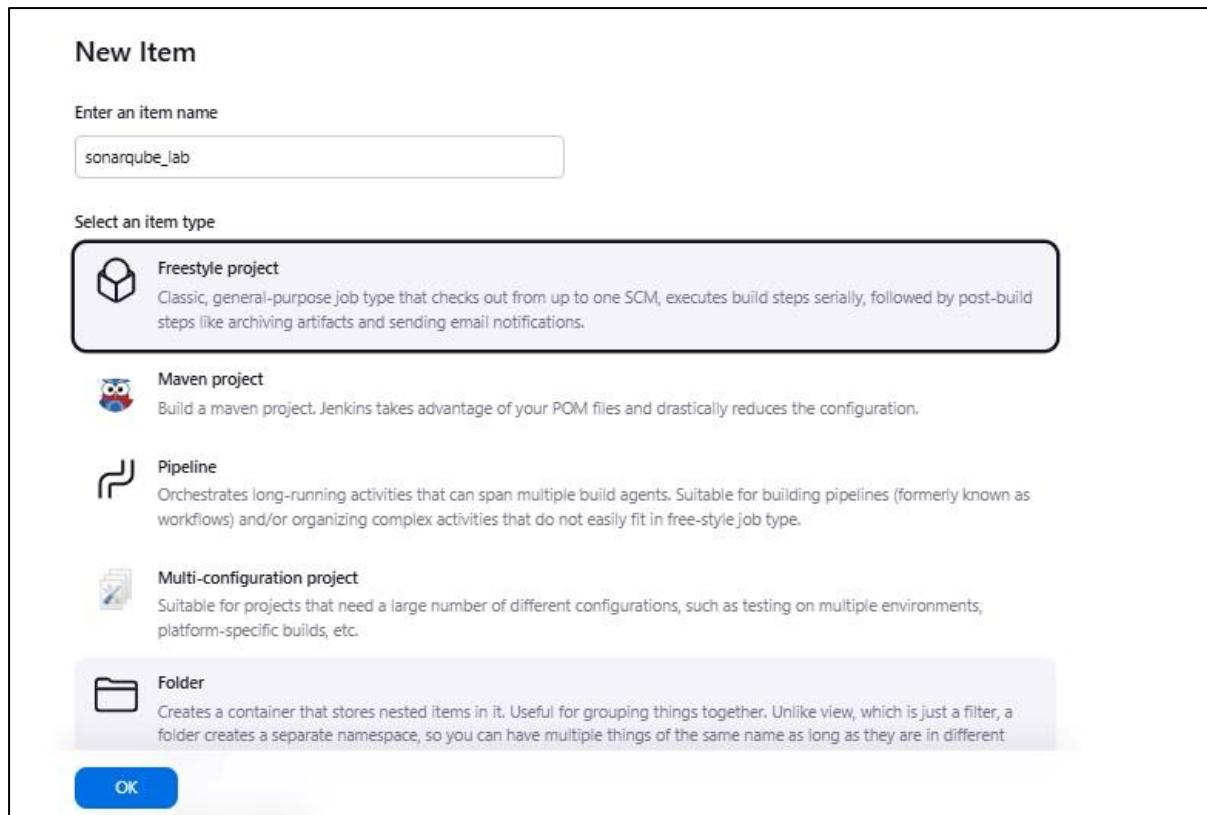
11. Go to Jenkins dashboard->Manage Jenkins->System and scroll down to SonarQube installations. Enter the name and URL in the fields and save the changes.

The screenshot shows the Jenkins configuration interface for SonarQube servers. It includes sections for 'SonarQube servers' (with a note about injecting server config as environment variables), 'SonarQube installations' (listing one entry), and a detailed configuration panel for a specific installation named 'sonarqube'. The configuration fields include 'Name' (sonarqube), 'Server URL' (http://localhost:9000), and 'Server authentication token' (set to '- none -'). There is also an 'Advanced' section and a 'Save' button at the bottom.

12. In SonarQube Scanner add the latest version then apply the changes and save it.

The screenshot shows the Jenkins configuration interface for SonarQube Scanner. It includes sections for 'SonarQube Scanner' (with an 'Install automatically' checkbox checked) and 'Ant installations'. A configuration panel for 'sonarqube_lab' is shown, with 'Name' set to 'sonarqube_lab', 'Install from Maven Central' selected, and 'Version' set to 'SonarQube Scanner 6.2.0.4584'. There is also an 'Add Installer' dropdown and a 'Save' button at the bottom.

13. Go to Jenkins and then create a new item, enter the item name and select “Freestyle project” and then click on ok.



14. Use this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject



15. In Analysis properties, mention the SonarQube Project Key, Login, Password, Source path and Host URL.

JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?
[Empty input field]

Analysis properties ?
sonar.projectKey=sonarqube
sonar.login=admin
sonar.password=123456
sonar.hostUrl=http://localhost:9000
sonar.sources=.

Additional arguments ?
[Empty input field]

JVM Options ?
[Empty input field]

16. Now, you need to grant the local user (here admin user) permissions to Execute the Analysis stage on SonarQube. For this go to http://localhost:<port_number>/admin/permissions and check the 'Execute Analysis' checkbox under Administrator.

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

4 of 4 shown

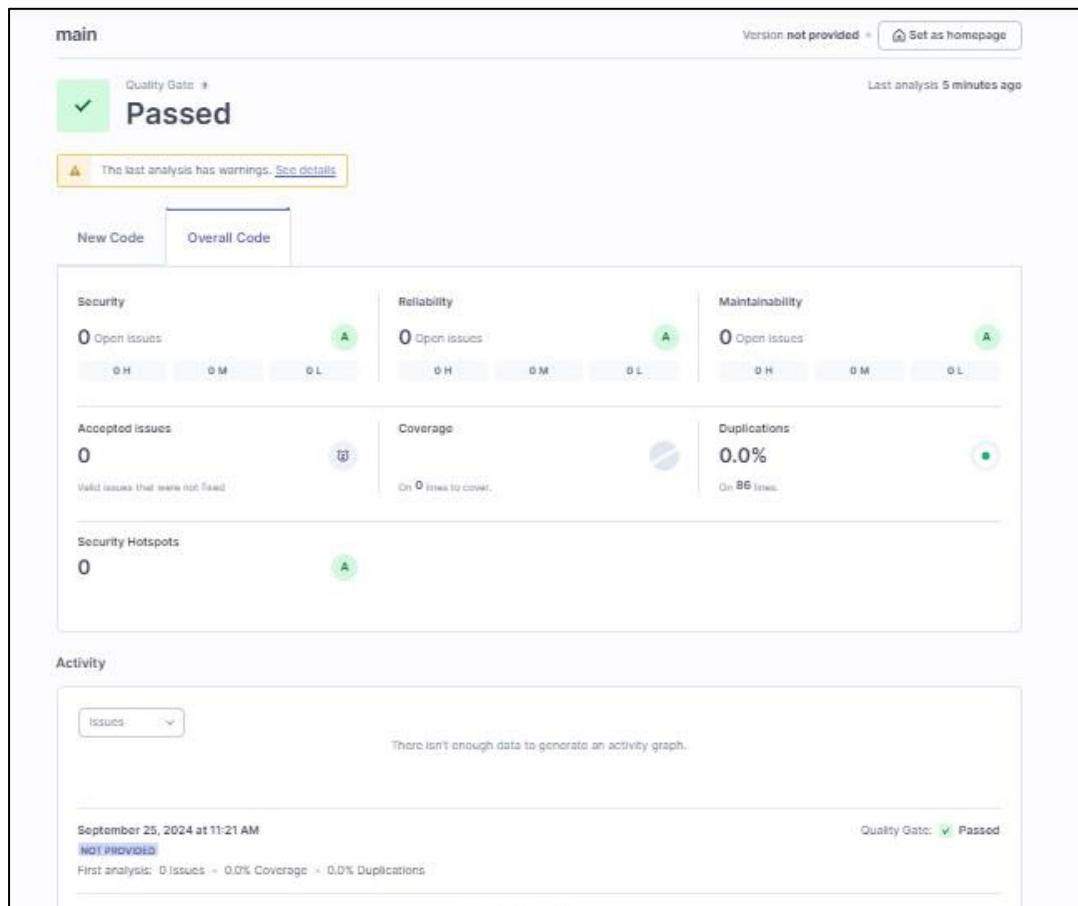
17. Go to the job you have just built and click on Build Now.

The screenshot shows the Jenkins dashboard for a job named 'sonarqube-lab'. The build number is #10, which was started by user 'Dimple Dalwani' on Sep 25, 2024, at 11:21:17 AM. The build status is green. The 'git' section shows the revision f2bc042c04c6e72427c380bcaeef6d6fee7b49adf and the repository https://github.com/shazforiot/MSBuild_firstproject. The 'Timings' section indicates a total duration of 1 min 2 sec from scheduled to completion. The 'Console Output' link is visible on the left sidebar.

18. Check the console Output

```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO Analysis report uploaded in 3391ms
23:15:59.132 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO Analysis total time: 10:30.120 s
23:16:05.636 INFO SonarScanner Engine completed successfully
23:16:06.248 INFO EXECUTION SUCCESS
23:16:06.273 INFO Total time: 10:47.728s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

19. Go back to SonarQube and check the project.



Conclusion: While performing this experiment there was an issue in creating sonarqube docker image and we resolved it by logging in to the docker desktop and performing it through the terminal. Other than this we created a freestyle project and entered the sonarqube credentials and then performed build.

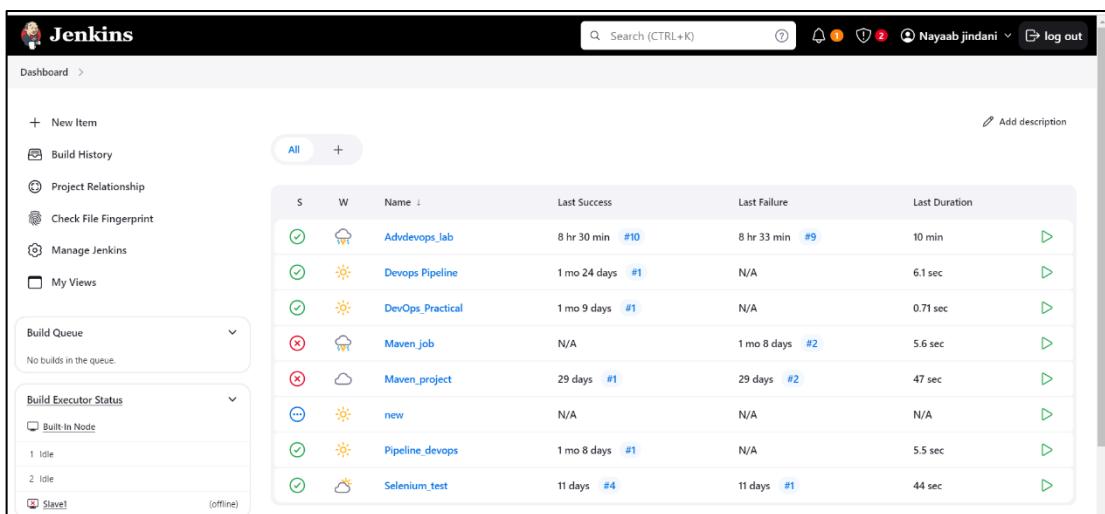
Advance DevOps

Experiment 8

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

Steps:

1. Open the Jenkins dashboard.



The screenshot shows the Jenkins dashboard with the following details:

- Left Sidebar:**
 - + New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
 - My Views
 - Build Queue: No builds in the queue.
 - Build Executor Status: 1 Idle, 2 Idle, 1 Slave (offline)
- Central Area:**
 - Filter: All
 - Table Headers: S, W, Name, Last Success, Last Failure, Last Duration
 - Table Data:

S	W	Name	Last Success	Last Failure	Last Duration
Green	Cloud	Advdevops_lab	8 hr 30 min #10	8 hr 33 min #9	10 min
Green	Sun	Devops Pipeline	1 mo 24 days #1	N/A	6.1 sec
Green	Sun	DevOps_Practical	1 mo 9 days #1	N/A	0.71 sec
Red	Cloud	Maven_job	N/A	1 mo 8 days #2	5.6 sec
Red	Cloud	Maven_project	29 days #1	29 days #2	47 sec
Blue	Sun	new	N/A	N/A	N/A
Green	Sun	Pipeline_devops	1 mo 8 days #1	N/A	5.5 sec
Green	Cloud	Selenium_test	11 days #4	11 days #1	44 sec

2. First, we will pull the latest version of sonar qube image from the docker hub using the command:

```
docker pull sonarqube:latest
```

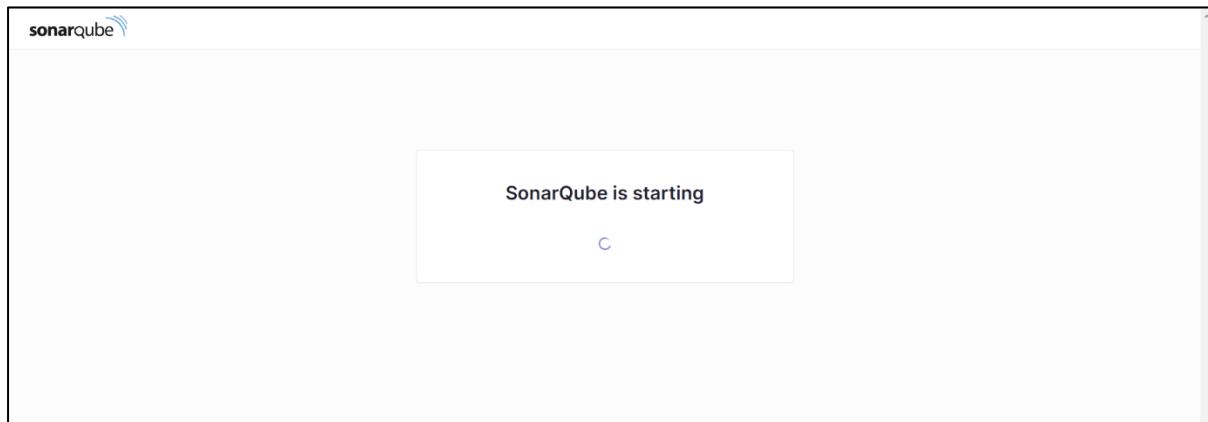
```
docker pull sonarqube:latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700eff54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

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

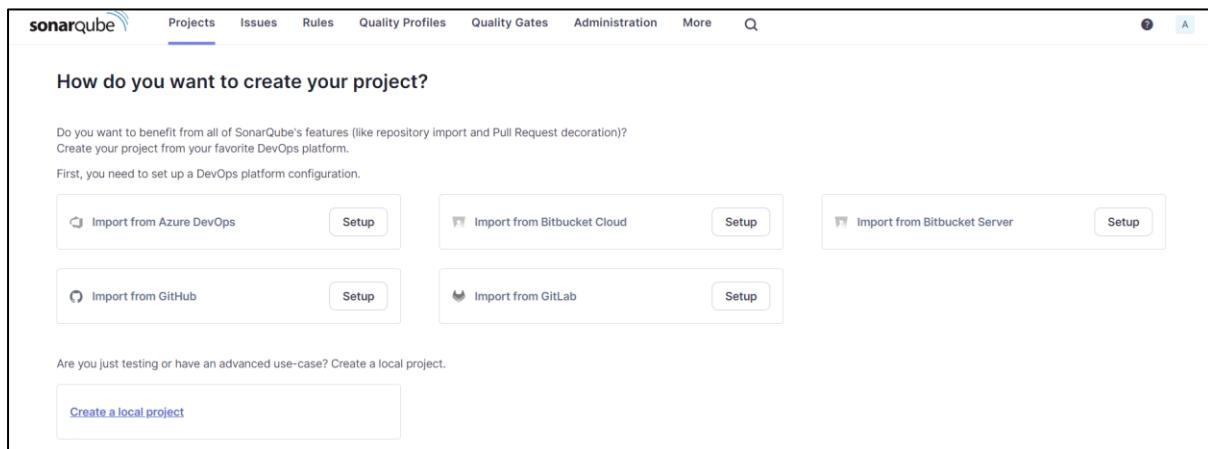
3. Next, we will run SonarQube in a docker container

```
PS C:\Users\DELL> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
67aeea599cf48e12d50da592eff01d8257f58e6c1bffff50446066e5f2a8844
```

4. Once the container is running, we will check the status of sonar qube on the port 9000. It will show “Sonar qube is starting”



5. Now login to SonarQube using username and password.



6. Click on create a local project option from the dashboard and give a name to the project, click on next and complete the setup.

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](#)

7. Go to Jenkins dashboard and create a new item by giving a name and select pipeline option.

Enter an item name

» Required field

 **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.

8. Scroll down to pipeline script and enter the following script:

```
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/"  
        }  
    }  
}
```

(Change the path and credentials)

The screenshot shows the Jenkins Pipeline configuration interface. At the top, there's a dropdown menu labeled "Pipeline script". Below it is a "Script" editor area containing the Groovy pipeline code provided earlier. The code is syntax-highlighted, with lines numbered from 1 to 17. A "Use Groovy Sandbox" checkbox is checked. At the bottom of the editor, there are "Save" and "Apply" buttons.

```
Definition  
Pipeline script  
Script ?  
1 node {  
2     stage('Cloning the GitHub Repo') {  
3         git 'https://github.com/shazforiot/GOL.git'  
4     }  
5     stage('SonarQube analysis') {  
6         withSonarQubeEnv('sonarqube') {  
7             sh """  
8                 C:/Users/Dell/Downloads/sonar-scanner-cli-6.2.0.4584-windows-x64/bin/sonar-scanner \  
9                     -D sonar.login=admin \  
10                    -D sonar.password=nayaab \  
11                    -D sonar.projectKey=sonarqube-test \  
12                    -D sonar.exclusions=vendor/**,resources/**,**/*.java \  
13                    -D sonar.host.url=http://127.0.0.1:9000/  
14             """  
15         }  
16     }  
17 }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

9. Now run the build.

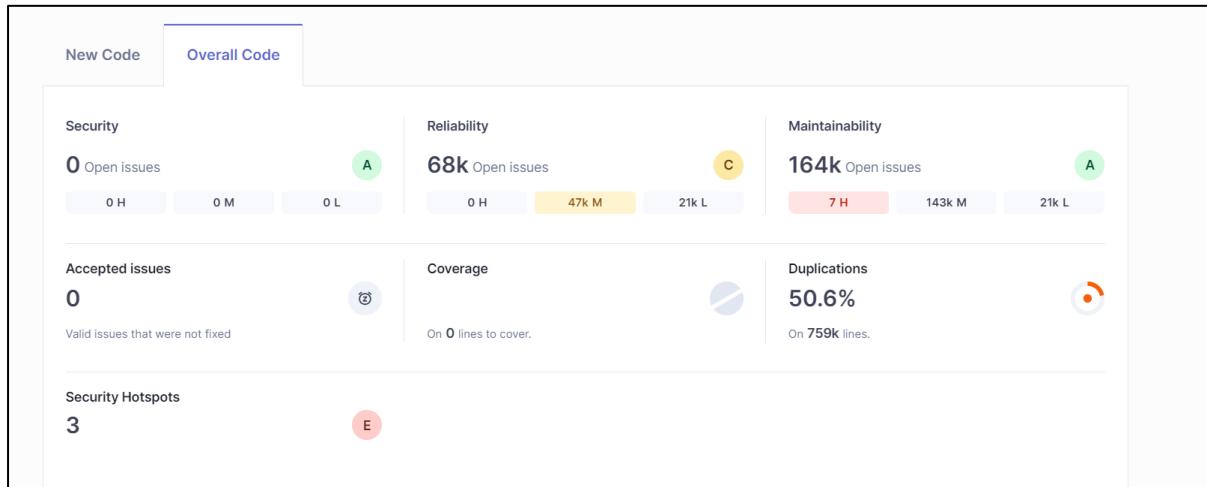
Console Output

```
Started by user Nayaab jindani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.2.windows.1'
> git.exe fetch -tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheck # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
```

```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO Analysis report uploaded in 3391ms
23:15:59.132 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO Analysis total time: 10:30.120 s
23:16:05.636 INFO SonarScanner Engine completed successfully
23:16:06.248 INFO EXECUTION SUCCESS
23:16:06.273 INFO Total time: 10:47.728s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The build is successful.

10. Go to sonar qube and check the different errors, code problems, bugs present in the code.



The screenshot shows the SonarQube analysis results for the file 'gameoflife-acceptance-tests/Dockerfile'. The results are listed in a table format:

Issue Type	Description	Severity	Effort	Last Updated	Tags
Intentionality	<input type="checkbox"/> Use a specific version tag for the image.	Maintainability	No tags	L1 - 5min effort	4 years ago
Intentionality	<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Maintainability	No tags	L12 - 5min effort	4 years ago
Intentionality	<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Maintainability	No tags	L12 - 5min effort	4 years ago
Intentionality	<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Maintainability	No tags	L12 - 5min effort	4 years ago

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

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

Reliability ? accessibility wcag2-a +

Open Not assigned L1 - 2min effort - 4 years ago • Bug • Major

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

Reliability ? accessibility wcag2-a +

Open Not assigned L3 - 2min effort - 4 years ago • Bug • Major

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

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

Reliability ? accessibility wcag2-a +

Open Not assigned L1 - 2min effort - 4 years ago • Bug • Major

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

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image. Intentionality

Maintainability ? No tags +

Open Not assigned L1 - 5min effort - 4 years ago • Code Smell • Major

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

Maintainability ? No tags +

Open Not assigned L12 - 5min effort - 4 years ago • Code Smell • Major

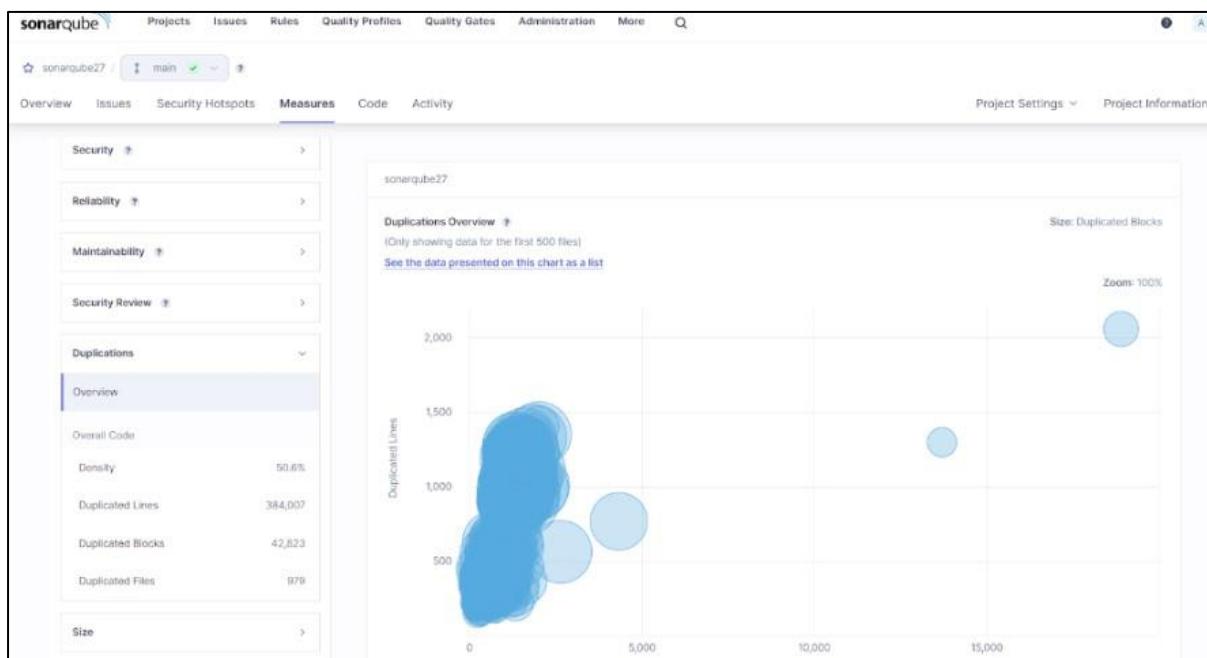
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality

Maintainability ? No tags +

Open Not assigned L12 - 5min effort - 4 years ago • Code Smell • Major

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

Maintainability ? No tags +



Conclusion:

In this experiment we created a Jenkins CICD Pipeline to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample code. It is to be checked whether the sonar scanner plugin is installed in Jenkins or not and also provide the correct path and credentials in the pipeline script or else it leads to the failure of the build.

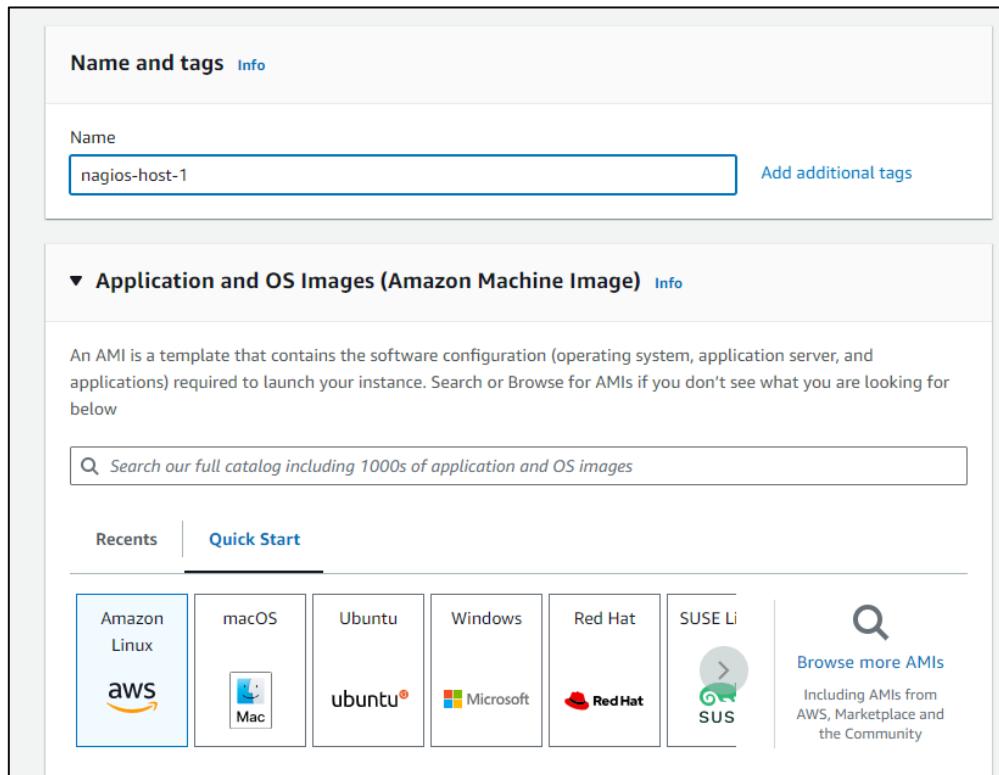
Advance DevOps

Experiment 9

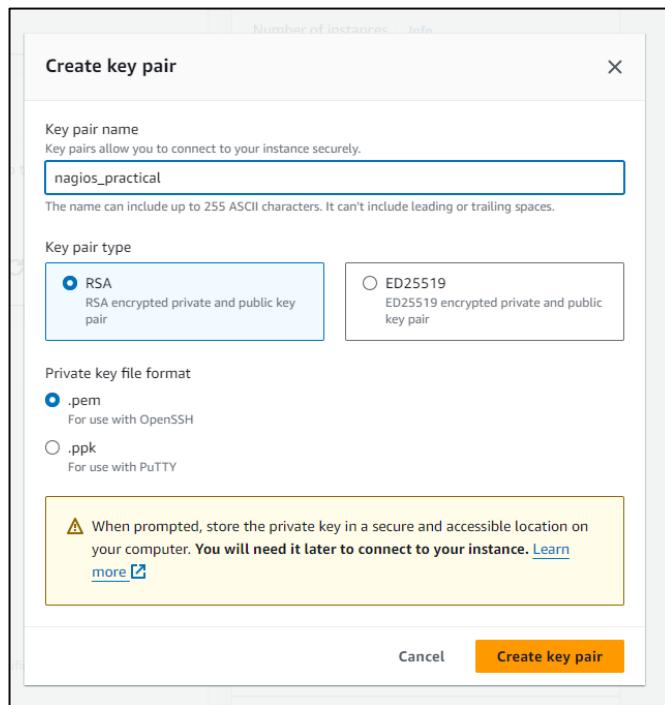
Aim: To understand continuous monitoring and installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Steps:

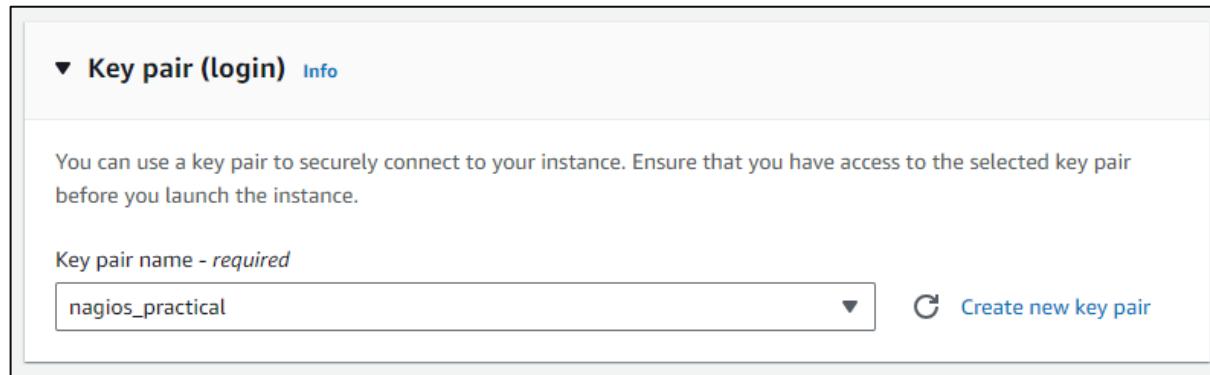
1. Create an ec2 instance and select amazon Linux as the OS



2. Now we will create a key pair.



3. Use the created key pair while creating the instance



4. Once the instance is successfully initiated go to security groups and select the security group id of the instance you just created.

(The security group name is visible during instance creation and also on the ec2 instances dashboard)

Security Groups (33) Info						
	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-06e44bf6931c962f3	launch-wizard-13	vpc-0d1089189551d9d25	launch-wizard-13 created 2024-09-08...	025066268342
<input type="checkbox"/>	-	sg-05448f01f173b8d27	launch-wizard-21	vpc-0d1089189551d9d25	launch-wizard-21 created 2024-09-13...	025066268342
<input type="checkbox"/>	-	sg-02f32431c127a01bc	launch-wizard-14	vpc-0d1089189551d9d25	launch-wizard-14 created 2024-09-08...	025066268342
<input type="checkbox"/>	-	sg-0588ff70648d484edd	launch-wizard-32	vpc-0d1089189551d9d25	launch-wizard-32 created 2024-10-01...	025066268342
<input type="checkbox"/>	-	sg-0b4f7ff506f304d6c	launch-wizard-15	vpc-0d1089189551d9d25	launch-wizard-15 created 2024-09-08...	025066268342
<input type="checkbox"/>	-	sg-077e1908baa1282b8	default	vpc-0d1089189551d9d25	default VPC security group	025066268342
<input type="checkbox"/>	-	sg-00ad94946f13866a7	launch-wizard-10	vpc-0d1089189551d9d25	launch-wizard-10 created 2024-08-29...	025066268342

5. To edit the inbound rules select the “Edit inbound rules” button

Inbound rules (1)						
	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0d219a020a411ab6c...	IPv4	SSH	TCP	22

6. Add the rules as given in the screenshot below

Edit inbound rules [Info](#)

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

Inbound rules Info						
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0d219a020a411ab6c	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0"/> X	Delete
-	HTTP	TCP	80	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete
-	HTTPS	TCP	443	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete
-	All traffic	All	All	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete
-	Custom TCP	TCP	5666	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete
-	All ICMP - IPv4	ICMP	All	Anywhere-IPv4	<input type="text" value="0.0.0.0"/> X	Delete

[Add rule](#)

7. Connect the instance

Instances (1/1) Info										
Find Instance by attribute or tag (case-sensitive)										
Instance state = running		Clear filters		Actions		Launch instances				
<input checked="" type="checkbox"/>	Name X	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4	DNS	Public IPv
<input checked="" type="checkbox"/>	nagios-host-1	i-0b21c79e1e222bc9d	Running Q Q	t2.micro	2/2 checks pa View alarms +	us-east-1c	ec2-34-230-73-94.com...	34.230.73		

8. Copy the ssh command given in the ssh client section.

The screenshot shows the AWS EC2 Instance Connect interface. The top navigation bar has four tabs: EC2 Instance Connect, Session Manager, SSH client (which is highlighted in blue), and EC2 serial console. Below the tabs, the Instance ID is listed as i-0b21c79e1e222bc9d (nagios-host-1). A list of steps for connecting via SSH is provided:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is nagios_practical.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "nagios_practical.pem"
4. Connect to your instance using its Public DNS:
ec2-34-230-73-94.compute-1.amazonaws.com

Below the steps, there is an example command:

```
ssh -i "nagios_practical.pem" ec2-user@ec2-34-230-73-94.compute-1.amazonaws.com
```

A callout box contains a note:

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

At the bottom right of the interface is a 'Cancel' button.

9. In your terminal paste the copied command, just replace the .pem file name with the actual location where the .pem file is downloaded in your system

```
C:\Users\DELL>ssh -i "C:\Users\DELL\Downloads\nagios_practical.pem" ec2-user@ec2-34-230-73-94.compute-1.amazonaws.com
The authenticity of host 'ec2-34-230-73-94.compute-1.amazonaws.com (34.230.73.94)' can't be established.
ED25519 key fingerprint is SHA256:05N8h6uRSMLgvSCs4gVzGHutukB+uAOHp4xAZI+rerA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-230-73-94.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     /_###_      Amazon Linux 2023
    /##\###\
   /##|###|
  /#/_/---> https://aws.amazon.com/linux/amazon-linux-2023
 /~\/_/ /
 /~/\_/
 /m/_
Last login: Tue Oct 1 15:23:21 2024 from 18.206.107.27
[ec2-user@ip-172-31-87-75 ~]$ |
```

10. Now install the following packages using yum:

sudo yum update

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum update
Last metadata expiration check: 0:30:09 ago on Tue Oct 1 15:04:44 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:31:22 ago on Tue Oct 1 15:04:44 2024.
Dependencies resolved.

=====
Package           Architecture   Version      Repository    Size
=====
Installing:
httpd            x86_64        2.4.62-1.amzn2023
php8.3           x86_64        8.3.10-1.amzn2023.0.1
Installing dependencies:
apr              x86_64        1.7.2-2.amzn2023.0.2
apr-util         x86_64        1.6.3-1.amzn2023.0.1
generic-logos-httpd noarch      18.0.0-12.amzn2023.0.3
httpd-core       x86_64        2.4.62-1.amzn2023
httpd-filesystem noarch      2.4.62-1.amzn2023
httpd-tools      x86_64        2.4.62-1.amzn2023
libbrotli        x86_64        1.0.9-4.amzn2023.0.2
libsodium         x86_64        1.0.19-4.amzn2023
libxmlslt        x86_64        1.1.34-5.amzn2023.0.2
mailcap          noarch      2.1.49-3.amzn2023.0.3
nginx-filesystem noarch      1:1.24.0-1.amzn2023.0.4
php8.3-cli       x86_64        8.3.10-1.amzn2023.0.1
php8.3-common   x86_64        8.3.10-1.amzn2023.0.1
php8.3-process  x86_64        8.3.10-1.amzn2023.0.1
php8.3-xml      x86_64        8.3.10-1.amzn2023.0.1
Installing weak dependencies:
apr-util-openssl x86_64        1.6.3-1.amzn2023.0.1
mod_http2        x86_64        2.0.27-1.amzn2023.0.3
mod_lua          x86_64        2.4.62-1.amzn2023
php8.3-fpm       x86_64        8.3.10-1.amzn2023.0.1
php8.3-mbstring  x86_64        8.3.10-1.amzn2023.0.1
php8.3-opcache   x86_64        8.3.10-1.amzn2023.0.1
php8.3-pdo       x86_64        8.3.10-1.amzn2023.0.1
php8.3-sodium   x86_64        8.3.10-1.amzn2023.0.1

Transaction Summary

=====
Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-filesystem-2.4.62-1.amzn2023.noarch
libsodium-1.0.19-4.amzn2023.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
php8.3-8.3.10-1.amzn2023.0.1.x86_64
php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64
php8.3-mbstring-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
php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

Unchanged:
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-core-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64

Complete!
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:35:45 ago on Tue Oct 1 15:04:44 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.

=====
Package           Architecture   Version      Repository    Size
=====
Installing:
gcc              x86_64        11.4.1-2.amzn2023.0.2
Installing dependencies:
anno2bin-docs    noarch      18.93-1.amzn2023.0.1
anno2bin-plugin-gcc x86_64        10.93-1.amzn2023.0.1
cpp              x86_64        11.4.1-2.amzn2023.0.2
gc               x86_64        8.0.4-5.amzn2023.0.2
glibc-devel      x86_64        2.34-52.amzn2023.0.11
glibc-headers-x86 noarch      2.34-52.amzn2023.0.11
guile22         x86_64        2.2.7-2.amzn2023.0.3
kernel-headers   x86_64        6.1.109-118.189.amzn2023
libmpc          x86_64        1.2.1-2.amzn2023.0.2
libtool-ltdl    x86_64        2.4.7-1.amzn2023.0.3
libcrypt-devel  x86_64        4.4.33-7.amzn2023
make             x86_64        1:4.3-5.amzn2023.0.2

Transaction Summary
=====
Install 13 Packages
```

```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch
  gcc-8.0.4-5.amzn2023.0.2.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64

  annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64

  cpp-11.4.1-2.amzn2023.0.2.x86_64
  glibc-devel-2.34-52.amzn2023.0.11.x86_64
  kernel-headers-6.1.109-118.189.amzn2023.x86_64
  libxcrypt-devel-4.4.33-7.amzn2023.x86_64

Complete!
```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:38:32 ago on Tue Oct 1 15:04:44 2024.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
  gd                  x86_64        2.3.3-5.amzn2023.0.3      amazonlinux  139 k
  gd-devel            x86_64        2.3.3-5.amzn2023.0.3      amazonlinux  38 k
Installing dependencies:
  brotli              x86_64        1.0.9-4.amzn2023.0.2      amazonlinux  314 k
  brotli-devel        x86_64        1.0.9-4.amzn2023.0.2      amazonlinux  31 k
  bzlib2-devel        x86_64        1.0.8-6.amzn2023.0.2      amazonlinux  214 k
  cairo               x86_64        1.17.6-2.amzn2023.0.1     amazonlinux  684 k
  cmake-fs             x86_64        3.22.2-1.amzn2023.0.4     amazonlinux  16 k
  fontconfig          x86_64        2.13.94-2.amzn2023.0.2     amazonlinux  273 k
  fontconfig-devel    x86_64        2.13.94-2.amzn2023.0.2     amazonlinux  128 k
  fonts-fs             noarch       1:2.0.5-12.amzn2023.0.2    amazonlinux  9.5 k
  freetype             x86_64        2.13.2-5.amzn2023.0.1     amazonlinux  423 k
  freetype-devel      x86_64        2.13.2-5.amzn2023.0.1     amazonlinux  912 k
  glib2-devel          x86_64        2.74.7-68.amzn2023.0.2     amazonlinux  486 k
  google-noto-fonts-common x86_64        20201206-2.amzn2023.0.2    amazonlinux  15 k
  google-noto-sans-vf-fonts x86_64        20201206-2.amzn2023.0.2    amazonlinux  492 k
  graphite2            x86_64        1.3.14-7.amzn2023.0.2     amazonlinux  97 k
  graphite2-devel     x86_64        1.3.14-7.amzn2023.0.2     amazonlinux  21 k
  harfbuzz              x86_64        7.0.0-2.amzn2023.0.1     amazonlinux  868 k
  harfbuzz-devel      x86_64        7.0.0-2.amzn2023.0.1     amazonlinux  404 k
  harfbuzz-icu         x86_64        7.0.0-2.amzn2023.0.1     amazonlinux  18 k
  jbigkit-libs         x86_64        2.1-21.amzn2023.0.2      amazonlinux  54 k
  langpacks-core-font-en x86_64        3.0-21.amzn2023.0.4      amazonlinux  10 k
```

```
Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64
  bzlib2-devel-1.0.8-6.amzn2023.0.2.x86_64
  cmake-fs-3.22.2-1.amzn2023.0.4.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
  freetype-2.13.2-5.amzn2023.0.1.x86_64
  gd-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-68.amzn2023.0.2.x86_64
  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
  libICE-1.0.10-6.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64
  libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64
  libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
  libicu-devel-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
  libpng-2.1.6.37-10.amzn2023.0.6.x86_64
  libselinux-devel-3.4-5.amzn2023.0.2.x86_64
  libtiff-4.0.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
  pixman-0.40.0-3.amzn2023.0.3.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64

  brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
  cairo-1.17.6-2.amzn2023.0.1.x86_64
  fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  fonts-fs-1:2.0.5-12.amzn2023.0.2.noarch
  freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
  gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
  graphite2-1.3.14-7.amzn2023.0.2.x86_64
  harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
  langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
  libSM-1.2.3-8.amzn2023.0.2.x86_64
  libX11-common-1.7.2-3.amzn2023.0.4.noarch
  libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
  libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
  libXpm-3.5.15-2.amzn2023.0.3.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
  libicu-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64
  libsep-devel-3.4-3.amzn2023.0.3.x86_64
  libtiff-devel-4.0.0-4.amzn2023.0.18.x86_64
  libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
  pcre2-devel-10.40-1.amzn2023.0.3.x86_64
  pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
  sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[ec2-user@ip-172-31-87-75 ~]$ |
```

11. Create a new Nagios User with its password using the below given commands.

sudo adduser -m nagios

sudo passwd nagios

```
[ec2-user@ip-172-31-87-75 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-87-75 ~]$ |
```

12. Create a new user group

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-87-75 ~]$ sudo groupadd nagcmd  
[ec2-user@ip-172-31-87-75 ~]$ |
```

13. Next execute these commands so that you don't have to use sudo for Apache and Nagios:

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

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

```
[ec2-user@ip-172-31-87-75 ~]$ sudo usermod -a -G nagcmd nagios  
sudo usermod -a -G nagcmd apache  
[ec2-user@ip-172-31-87-75 ~]$ |
```

14. Create a new directory for Nagios downloads

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

```
[ec2-user@ip-172-31-87-75 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-87-75 downloads]$ |
```

15. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-87-75 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz  
--2024-10-01 15:55:47-- 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:fef7:45ce  
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2065473 (2.0M) [application/x-gzip]  
Saving to: 'nagios-4.5.5.tar.gz'  
  
nagios-4.5.5.tar.gz          100%[=====] 1.97M  5.54MB/s   in 0.4s  
2024-10-01 15:55:47 (5.54 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]  
[ec2-user@ip-172-31-87-75 downloads]$ |
```

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

```
[ec2-user@ip-172-31-87-75 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-01 15:57:19-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz      100%[=====] 2.62M 4.14MB/s in 0.6s

2024-10-01 15:57:20 (4.14 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

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

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-87-75 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
```

17. We have to now change the directory to nagios-4.5.5, for this first verify whether nagios-4.5.5 exists by using ls command.

```
[ec2-user@ip-172-31-87-75 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
```

18. As nagios-4.5.5 is present we will now use cd command to change directory.

```
[ec2-user@ip-172-31-87-75 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ |
```

19. Now we will install openssl dev library by using the command:

sudo yum install openssl-devel

```

Transaction Summary
=====
Install 1 Package

Total download size: 3.0 M
Installed size: 0.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm           23 MB/s | 3.0 MB   00:00
Total                                         18 MB/s | 3.0 MB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :                               1/1
Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
Verifying  : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1

Installed:
openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64

```

20. Run the configuration script

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

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
```

```
*** 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.
```

21. To compile the source code run “make all”

22. To install binaries, init script and sample config files run

sudo make install

sudo make install-init

`sudo make install-config`

```
sudo make install-commandmode
```

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
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
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
```

23. In the config file edit the email address

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

```
#####
# CONTACTS
#
#####

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

define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin        ; Full name of user
    email             d2022.nayaab.jindani@ves.ac.in ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}
```

24. To configure the web interface run:

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-87-75 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 ***

[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ | Search
```

25. Create a nagios admin account and password

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

```
[ec2-user@ip-172-31-87-75 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
```

26. Restart apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ |
```

27. Go back to the downloads folder by using “cd ~/downloads” and unzip the plugins zip file using

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

```
[ec2-user@ip-172-31-87-75 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  
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h  
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h  
nagios-plugins-2.4.11/build-aux/test-driver  
nagios-plugins-2.4.11/config_test/  
nagios-plugins-2.4.11/config_test/Makefile  
nagios-plugins-2.4.11/config_test/run_tests  
nagios-plugins-2.4.11/config_test/child_test.c  
nagios-plugins-2.4.11/gl/  
nagios-plugins-2.4.11/gl/...
```

28. Compile and install the plugins

```
cd nagios-plugins-2.0.3
```

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

```
[ec2-user@ip-172-31-87-75 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
```

make

sudo make install

```
[ec2-user@ip-172-31-87-75 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 ; \
if test -f /usr/local/nagios/lib/charset.alias; then \
  sed -f ref-add.sed /usr/local/nagios/lib/charset.alias > /usr/local/nagios/lib/charset.tmp ; \
  /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
  rm -f /usr/local/nagios/lib/charset.tmp ; \
else \
  if $need_charset_alias; then \
    sed -f ref-add.sed charset.alias > /usr/local/nagios/lib/charset.tmp ; \
    /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
  fi ; \
fi ; \

```

29. Run below given commands to start nagios:

sudo chkconfig --add nagios

sudo chkconfig nagios on

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

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

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$
```

If the message says no errors detected then run “sudo service nagios start”

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ |
```

`sudo systemctl status nagios`

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

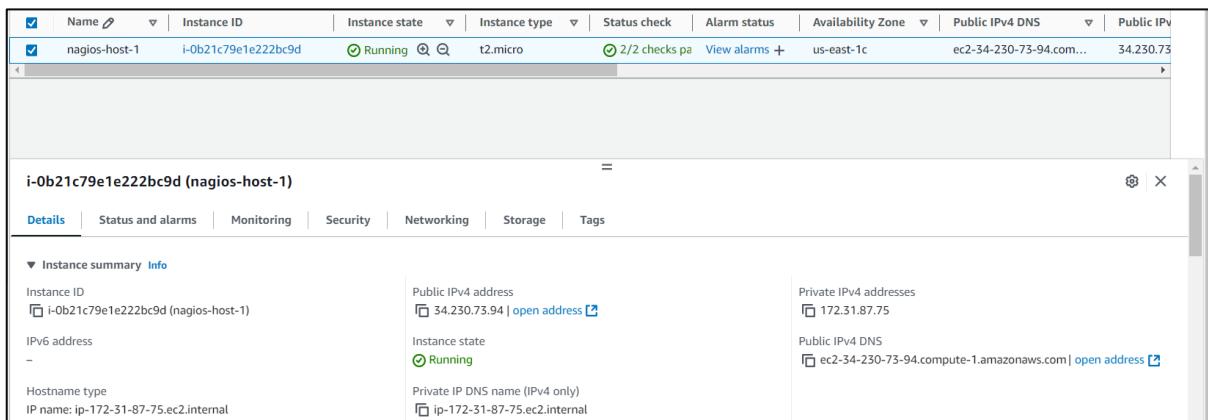
Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-87-75 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 Tue 2024-10-01 16:32:27 UTC; 48s ago
    Docs: https://www.nagios.org/documentation
 Process: 66684 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 66693 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 66694 (nagios)
   Tasks: 6 (limit: 1112)
     Memory: 5.7M
        CPU: 82ms
      CGroup: /system.slice/nagios.service
           ├─66694 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─66695 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─66696 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─66697 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─66698 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─66739 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

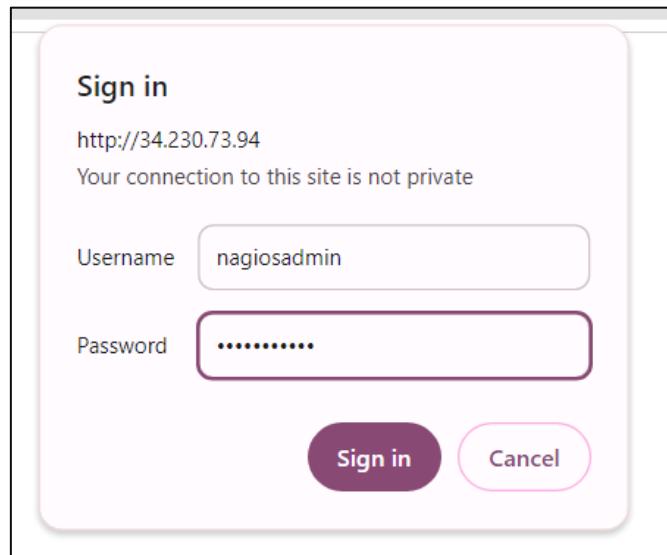
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: core query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: echo service query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: help for the query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Successfully registered manager as @wproc with query handler
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66696;pid=66696
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66695;pid=66695
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66697;pid=66697
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66698;pid=66698
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: Successfully launched command file worker with pid 66739
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ |
```

We can see that Nagios has been initialized correctly and its status is active.

30. Go back to your instances and copy the public IPv4 address



31. Lastly, go to your web browser and type “<http://<public-IPv4-address>/nagios>”
Replace public-IPv4-address with the public ip address of your instance which you copied.
You will get a prompt to enter the username and password that have been set for nagios admin in step 25.



You will see the below shown page after entering credentials.

The image shows the Nagios Core dashboard. The top navigation bar indicates 'Not secure' and the URL '34.230.73.94/nagios/'. The main header features the 'Nagios® Core™' logo and a green checkmark icon with the text '✓ Daemon running with PID 66694'. The left sidebar contains several navigation menus: 'General' (Home, Documentation), 'Current Status' (Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid), 'Problems' (Services (Unhandled), Hosts (Unhandled), Network Outages), 'Reports' (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and 'System' (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). The main content area includes a 'Get Started' section with a list of monitoring tips, a 'Latest News' section, a 'Don't Miss...' section, and a 'Quick Links' sidebar with links to Nagios Library (tutorials and docs), Nagios Labs (development blog), Nagios Exchange (plugins and addons), Nagios Support (tech support), Nagios.com (company), and Nagios.org (project). The footer contains copyright information for Nagios Core Development Team and Community Contributors, dated September 17, 2024, and a note about the GNU General Public License.

Conclusion: In this experiment the main issue that I faced was accessing the nagios web interface due to a “Forbidden: You do not have permission to access this resource” error. This issue was resolved by adjusting the inbound security rules and ensuring that all necessary files were installed in the correct directories. It is also important to restart apache after doing any changes for the changes to get reflected. Once Nagios was activated without errors, the output was successfully displayed on the web interface.

Advance DevOps

Experiment 10

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

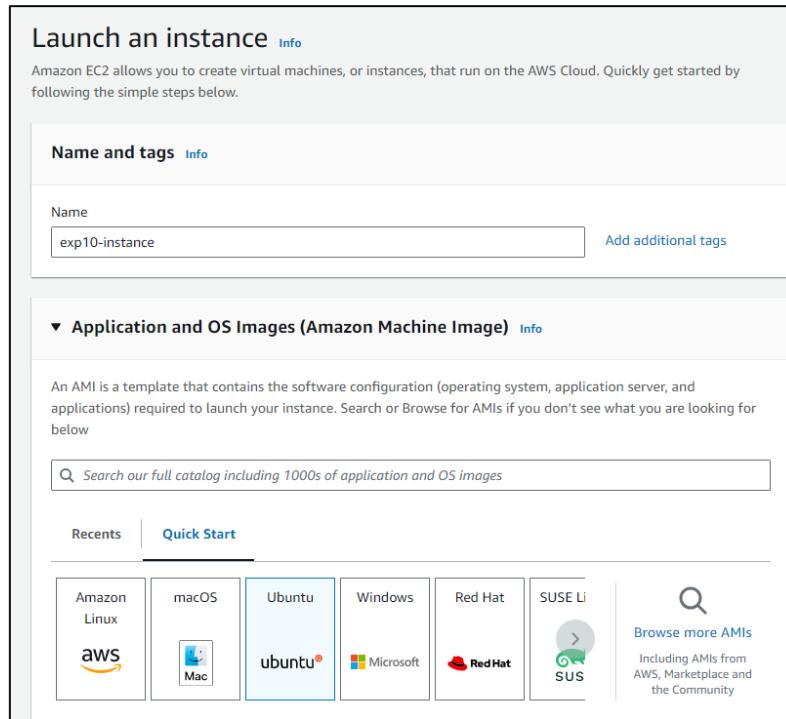
Steps:

1. Firstly, we will check whether nagios is running on the server side by using the command “sudo systemctl status nagios” on the host machine (host machine is the instance connected to the terminal in experiment 9, ensure that you have started the instance created for exp9, also check status of apache).

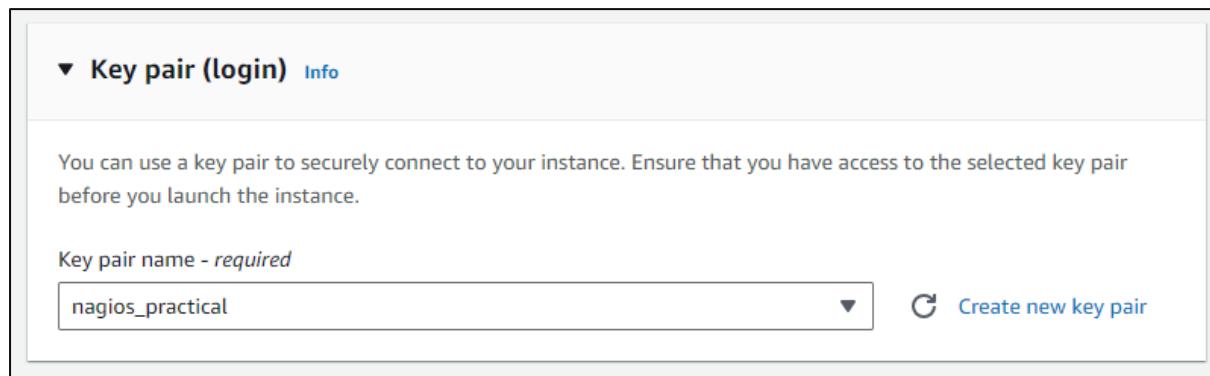
```
[ec2-user@ip-172-31-87-75 ~]$ sudo systemctl status
nagios
● ip-172-31-87-75.ec2.internal
  State: running
  Units: 295 loaded (incl. loaded aliases)
    Jobs: 1 queued
  Failed: 0 units
    Since: Wed 2024-10-02 06:17:29 UTC; 2min 42s ago
  systemd: 252.23-2.amzn2023
  CGroup: /
```

```
[ec2-user@ip-172-31-87-75 ~]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-87-75 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service;
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: active (running) since Wed 2024-10-02 06:26:51
    Docs: man:httpd.service(8)
  Main PID: 3242 (httpd)
    Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 1112)
     Memory: 13.1M
        CPU: 47ms
  CGroup: /system.slice/httpd.service
          ├─3242 /usr/sbin/httpd -DFOREGROUND
          ├─3243 /usr/sbin/httpd -DFOREGROUND
          ├─3244 /usr/sbin/httpd -DFOREGROUND
          ├─3245 /usr/sbin/httpd -DFOREGROUND
          ├─3246 /usr/sbin/httpd -DFOREGROUND
```

2. Now we will launch a new instance. Select ubuntu for the OS.



3. Select the key pair which was created and used in the exp 9.



4. Select existing security group and from the list of options select the security group created for exp 9. Previously it was launch wizard 32 and so here I have selected the same.

Network settings [Info](#)

[Edit](#)

Network [Info](#)
vpc-0d1089189551d9d25

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-32 sg-0588f70648d484edd [X](#)
VPC: vpc-0d1089189551d9d25

[Compare security group rules](#)

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

5. Open a new terminal to connect to the client machine. Copy the SSH command provided in the SSH client section during connection of instance. When pasting the command into your terminal, ensure you specify the full path to your .pem file instead of just the file name.

```
PS C:\Users\DELL> ssh -i "C:\Users\DELL\Downloads\nagios_practical.pem" ubuntu@ec2-18-207-191-20.compute-1.amazonaws.com
The authenticity of host 'ec2-18-207-191-20.compute-1.amazonaws.com (18.207.191.20)' can't be established.
ED25519 key fingerprint is SHA256:NPJP0UfSGZXQUXK8GQ9sw/fIzAFXOnabRJiCiAFdyiU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-207-191-20.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

System information as of Wed Oct  2 06:36:10 UTC 2024

System load:  0.15           Processes:          106
Usage of /:   22.9% of 6.71GB  Users logged in:     0
Memory usage: 21%            IPv4 address for enX0: 172.31.40.130
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

6. Now go back to your host machine and run the following command

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

```
[ec2-user@ip-172-31-87-75 ~]$ ps -ef | grep nagios
nagios    2002      1  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    2003    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    2004    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    2005    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    2006    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    2007    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   4389    2306  0 06:43 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-87-75 ~]$ | grep --color=auto nagios
```

7. Now perform these commands on the host terminal

```
sudo su
```

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

```
[root@ip-172-31-87-75 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-87-75 ec2-user]# |
```

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

```
[root@ip-172-31-87-75 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-87-75 ec2-user]# |
```

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

The above given command will open the nano text editor wherein you have to do the following changes:

- i. Change the hostgroup name to linux-servers1

```
#####
#
# 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             linuxserver            ; Comma separated list of hosts that belong to this group
}
```

- ii. Change host name and alias from localhost to linuxserver everywhere in the file

```
# Define a service to "ping" the local machine

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

- iii. Change the address to the public IPv4 address of the ubuntu instance (You will find the ip address when you select the instance on the ec2 instances dashboard)

```
# 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           linuxserver
    address         18.207.191.20
}
```

8. Open the Nagios Config file by using this command:
 nano /usr/local/nagios/etc/nagios.cfg
 nano text editor will get opened

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

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

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

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

# Definitions for monitoring a Windows machine
```

9. In the text editor add “cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/” this line

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

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

10. Now we will verify the configuration files

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

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 16 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-87-75 ec2-user]# |
```

If there are no errors we can proceed further

11. We will now restart the nagios service
service nagios restart

```
[root@ip-172-31-87-75 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-87-75 ec2-user]# |
```

12. Now on the client machine (The ubuntu machine we created for this experiment) run the following command:

`sudo apt update -y`

```
ubuntu@ip-172-31-40-130:~$ sudo apt update -y
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 [380 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 [535 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8676 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [380 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [156 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
```

`sudo apt install gcc -y`

```
ubuntu@ip-172-31-40-130:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core
fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libao-m3 libatomic1 libbinutils libc-dev-bin libc-devtools
libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0
libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libis123 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblrc4
liblsan0 libmpc3 libquadmath0 libsframe1 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxml4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gcc-13-doc
gdb-x86_64-linux-gnu libgc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-ravle libheif-plugin-svtenc
The following NEW packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core
fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libao-m3 libatomic1 libbinutils libc-dev-bin libc-devtools
libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0
libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libis123 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblrc4
liblsan0 libmpc3 libquadmath0 libsframe1 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxml4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
```

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

```
ubuntu@ip-172-31-40-130:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
libsnp-base libsnpmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins-basic monitoring-plugins-common
monitoring-plugins-standard mysql-common python3-gpg python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common
samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
Suggested packages:
cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib fping postfix
| sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients python3-dnspython cifs-utils
The following NEW packages will be installed:
libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
libsnp-base libsnpmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins monitoring-plugins-basic
monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg python3-ldb python3-markdown python3-samba
python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
```

13. Open nrpe.cfg file to make changes.

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

Under allowed_hosts, add your nagios host public IPv4 address:

```
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,54.163.184.143

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
```

14. Now restart the NRPE server

`sudo systemctl restart nagios-nrpe-server`

```
ubuntu@ip-172-31-40-130:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-40-130:~$ |
```

15. Go to the nagios dashboard and click on hosts

Click on linux server

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-02-2024 07:54:42	0d 0h 20m 46s	PING OK - Packet loss = 0%, RTA = 1.75 ms
localhost	UP	10-02-2024 07:54:24	0d 15h 23m 39s	PING OK - Packet loss = 0%, RTA = 0.03 ms

We can see the host state information:

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

If you want to see all the services and ports being monitored then select the services option and you will see the page as shown below:

Current Network Status
 Updated: Wed Oct 2 07:58:01 UTC 2024
 Updated every 30 seconds
 Nagios® Core™ 4.5.5 - www.nagios.org
 Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems	All Types		
0	2		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0
All Problems	All Types			
4	16			

View History For All hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Service Status Details For All Hosts

Limit Results: 100 ▾

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
linuxserver	Current Load	OK	10-02-2024 07:55:57	0d 0h 22m 4s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:56:35	0d 0h 21m 26s	1/4	USERS OK - 3 users currently logged in
	HTTP	CRITICAL	10-02-2024 07:55:12	0d 0h 17m 49s	4/4	connect to address 18.207.191.20 and port 80: Connection refused
	PING	OK	10-02-2024 07:57:50	0d 0h 20m 11s	1/4	PING OK - Packet loss = 0%, RTA = 2.11 ms
	Root Partition	OK	10-02-2024 07:53:27	0d 0h 19m 34s	1/4	DISK OK - free space: / 6114 MB (75.33% inode=98%)
	SSH	OK	10-02-2024 07:54:05	0d 0h 18m 56s	1/4	SSH OK - OpenSSH_9_6p1 Ubuntu-Subuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 07:57:42	0d 0h 15m 19s	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	10-02-2024 07:55:20	0d 0h 17m 41s	1/4	PROCS OK: 38 processes with STATE = R/SZDT
localhost	Current Load	OK	10-02-2024 07:53:09	0d 1h 24m 57s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:53:47	0d 1h 24m 19s	1/4	USERS OK - 3 users currently logged in
	HTTP	WARNING	10-02-2024 07:54:24	0d 1h 28m 37s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	10-02-2024 07:55:02	0d 1h 23m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-02-2024 07:55:39	0d 1h 22m 27s	1/4	DISK OK - free space: / 6114 MB (75.33% inode=98%)
	SSH	OK	10-02-2024 07:56:17	0d 1h 21m 49s	1/4	SSH OK - OpenSSH_8_7 protocol 2.0
Swap Usage	CRITICAL	10-02-2024 07:56:54	0d 1h 18m 12s	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	10-02-2024 07:57:32	0d 1h 20m 34s	1/4	PROCS OK: 38 processes with STATE = R/SZDT	

Results 1 - 16 of 16 Matching Services

Conclusion: For performing this experiment it is necessary to start the instance of the previous experiment as that will act as the host and the instance created in this experiment will be the client machine. There were errors when I tried to run the command to verify the Nagios configuration file and in order to resolve those errors I reinstalled the nagios plugins and restarted the nagios service after which the errors were fixed.

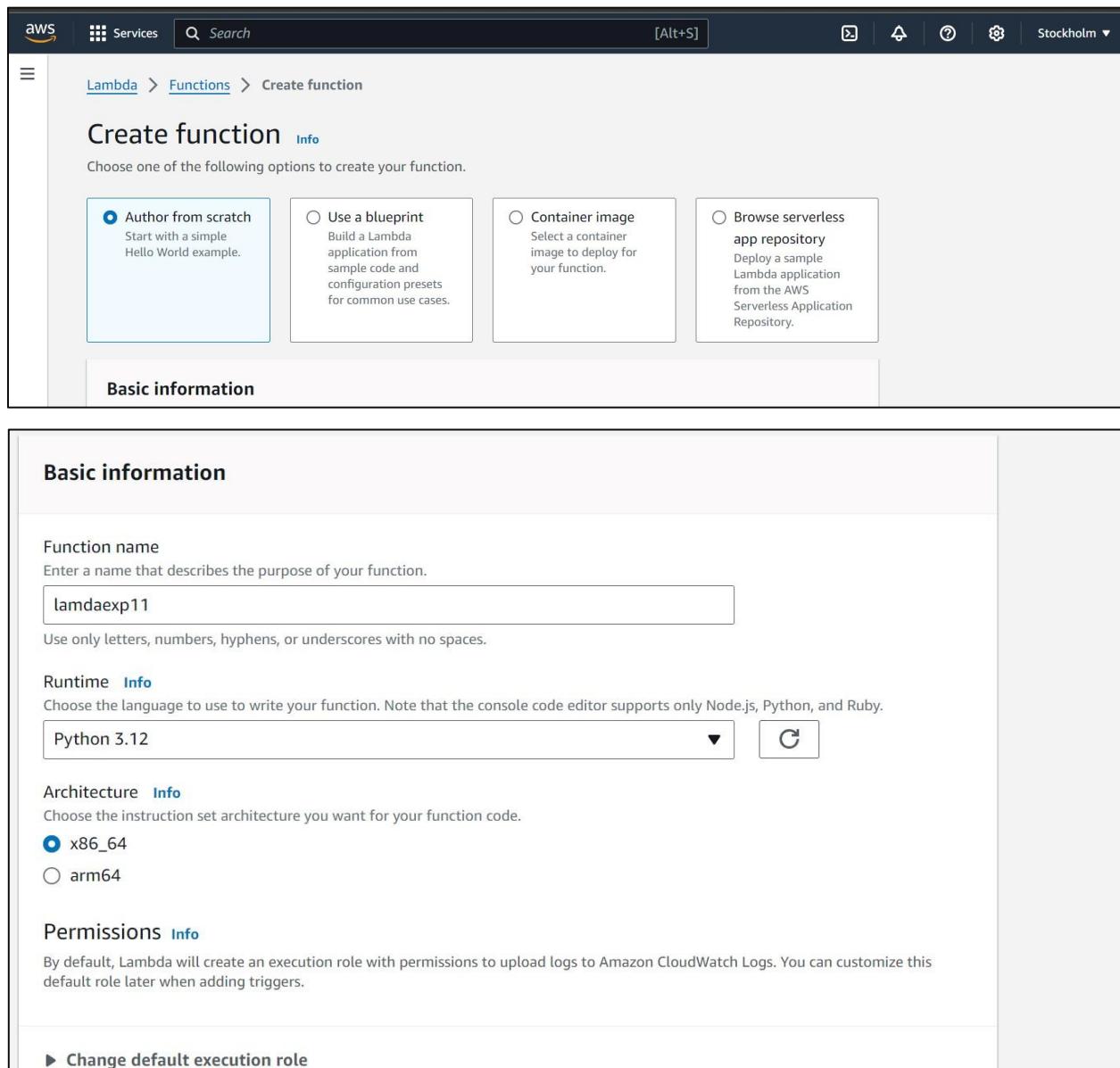
Advance DevOps

Experiment 11

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

Steps:

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



The screenshot shows the AWS Lambda 'Create function' wizard. The top navigation bar includes 'Services', a search bar, and a location dropdown set to 'Stockholm'. The main title is 'Create function' with an 'Info' link. Below it, a sub-header says 'Choose one of the following options to create your function.' There are four options:

- Author from scratch: Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.
- Browse serverless app repository: Deploy a sample Lambda application from the AWS Serverless Application Repository.

The 'Basic information' step is currently selected. It contains fields for 'Function name' (set to 'lamdaexp11'), 'Runtime' (set to 'Python 3.12'), 'Architecture' (set to 'x86_64'), and 'Permissions' (with a note about default execution role). At the bottom, there's a link to 'Change default execution role'.

2. After the function is created successfully go on code, write the default code and then configure it.

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

Lambda > Functions > lamdaexp11

lamdaexp11

Function overview Info

Description
-

Last modified
16 seconds ago

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

Diagram **Template**

Layers (0)

+ Add trigger + Add destination

Throttle Copy ARN Actions ▾ Export to Application Composer Download ▾

Code source Info

File Edit Find View Go Tools Window Test Deploy

Upload from ▼

Environment Vari +

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

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

Code Test Monitor Configuration Aliases Versions

General configuration Info Edit

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

Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC

Lambda > Functions > lamdaexp11 > Edit basic settings

Edit basic settings

Basic settings [Info](#)

Description - optional
D15C

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

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

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

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

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).
None

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

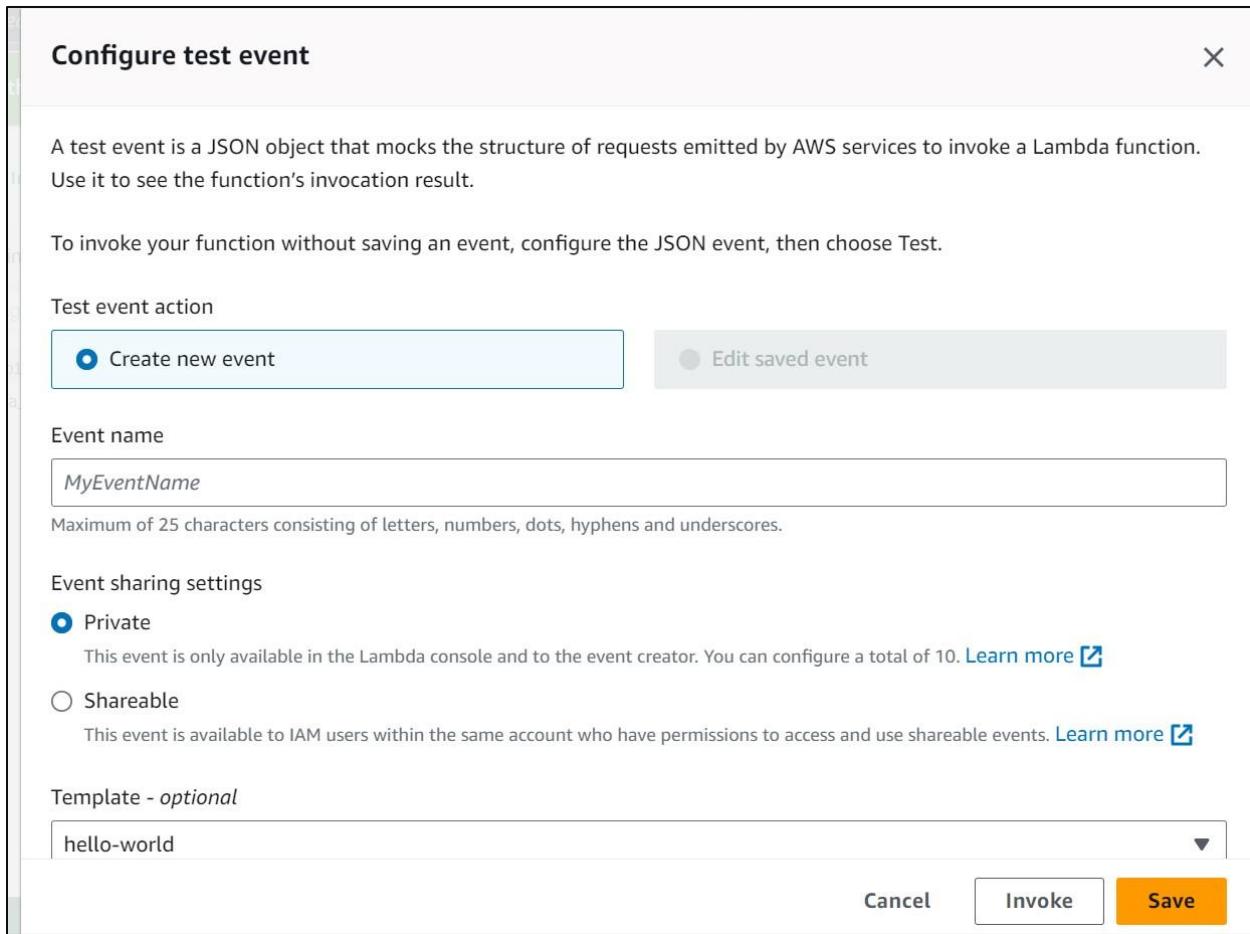
Timeout
0 min 1 sec

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

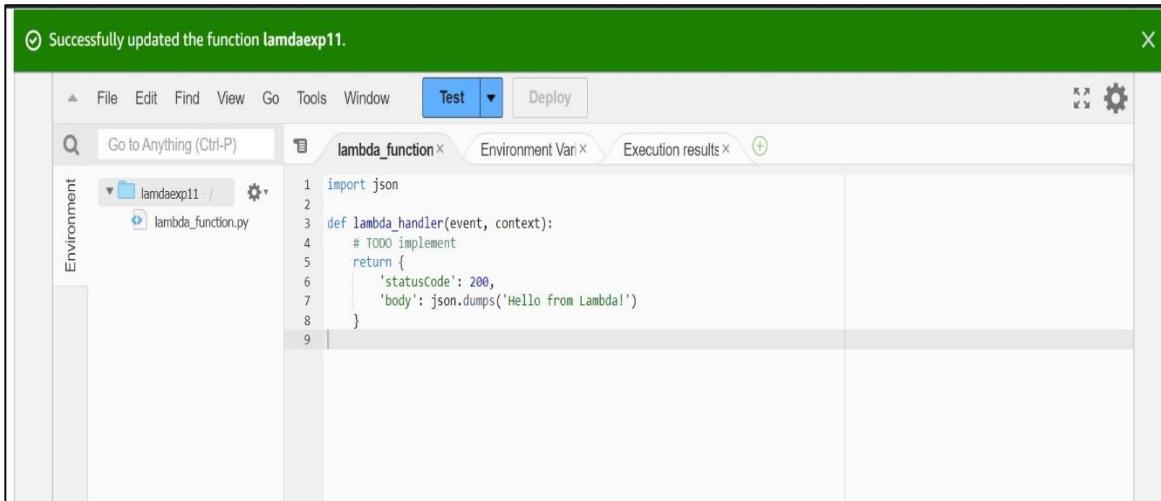
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/lamdaexp11-role-vj5j9g95 [View the lamdaexp11-role-vj5j9g95 role](#) on the IAM console.

Cancel [Save](#)

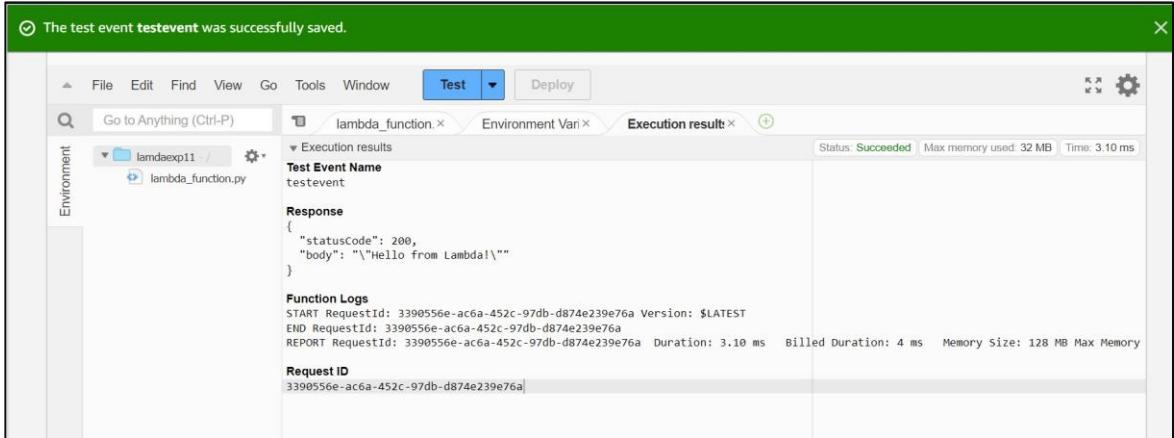
5. Go on configure test event click on “create new event”. Add an event name, set the event sharing settings and select hello-world template option. After this select save.



6. Once the test event is successfully created and saved click on test option to test the code.



7. The function has been added successfully.



The screenshot shows the AWS Lambda Test interface. At the top, a green banner displays the message: "The test event **testevent** was successfully saved." Below the banner, the interface has a toolbar with File, Edit, Find, View, Go, Tools, Window, a Test dropdown, Deploy, and settings icons. The main area is titled "lambda_function.x" and shows the "Execution results" tab selected. It displays the following information:

- Test Event Name:** testevent
- Status:** Succeeded | Max memory used: 32 MB | Time: 3.10 ms
- Response:**

```
{  
    "statusCode": 200,  
    "body": "\"Hello from Lambda!\""  
}
```
- Function Logs:**

```
START RequestId: 3390556e-ac6a-452c-97db-d874e239e76a Version: $LATEST  
END RequestId: 3390556e-ac6a-452c-97db-d874e239e76a  
REPORT RequestId: 3390556e-ac6a-452c-97db-d874e239e76a Duration: 3.10 ms Billed Duration: 4 ms Memory Size: 128 MB Max Memory
```
- Request ID:** 3390556e-ac6a-452c-97db-d874e239e76a

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

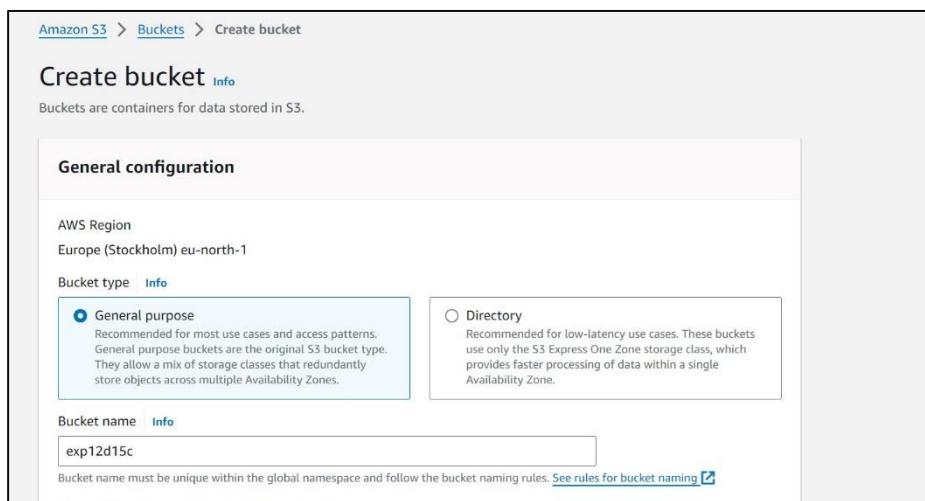
Advance DevOps

Experiment 12

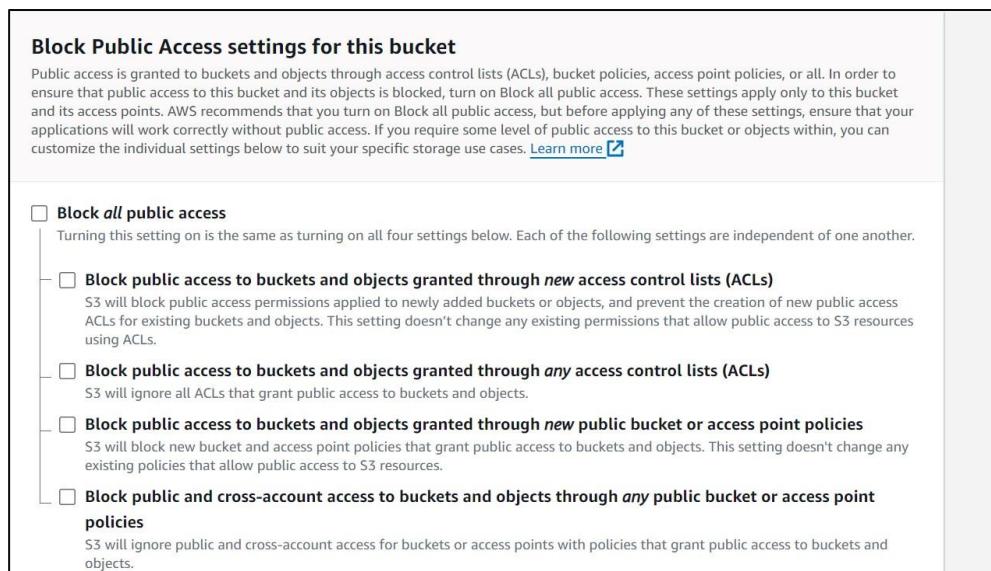
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Steps:

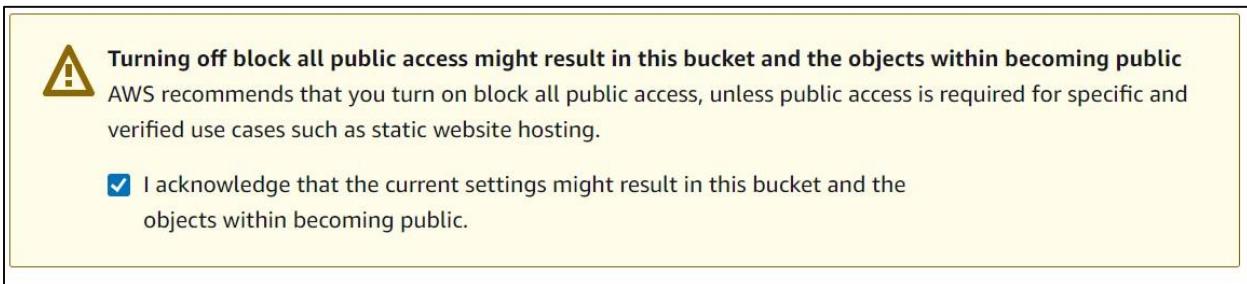
1. First, we have to create a S3 bucket for which give a name to the bucket.



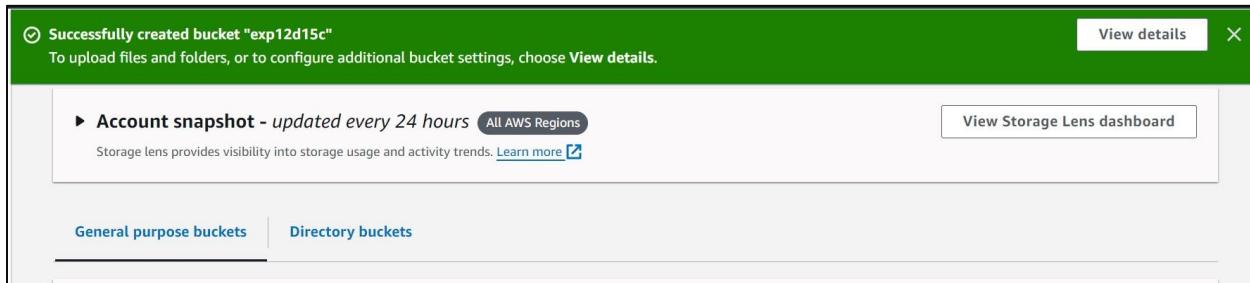
2. To allow public access to the bucket, uncheck the block all public access option.



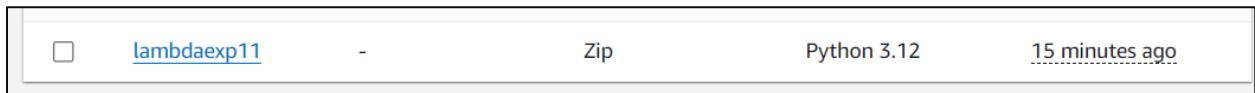
3. Give confirmation that you want to allow full public access and create the bucket



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



5. After creating the bucket go to the AWS dashboard and search for lambda function service, select the function we created in experiment 11 (lambdaexp11). We are going to add this bucket as a trigger to this function.



6. After selecting the function scroll down to the code section and add the following lines of code to display the required message when image is added to the bucket.

Code source [Info](#)

[Upload from](#)

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

[Environment](#) [Go to Anything \(Ctrl-P\)](#)

lambda_function [Environment Var](#) [+](#)

```

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

```

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

8. It will lead you to the trigger configuration tab. Here select the service as S3 and choose the bucket you just created.

9. Here you can see we have the confirmation message as well as the S3 bucket has been added to our triggers.

The screenshot shows the Lambda function overview for 'lamdaexp11'. At the top, there is a success message: 'The trigger exp12d15c was successfully added to function lamdaexp11. The function is now receiving events from the trigger.' Below this, the 'Function overview' section is displayed. It includes a 'Diagram' tab (selected), a 'Template' tab, and a visual representation of the function structure. The diagram shows 'lamdaexp11' at the top, followed by a 'Layers' section with '(0)' listed. Below this is an 'S3' box with a plus sign and 'Add destination' text. To the right of the diagram, there are details: 'Description D15C', 'Last modified 18 minutes ago', 'Function ARN arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11', and 'Function URL Info'. Buttons for 'Export to Application Composer' and 'Download' are also present.

10. Now we will upload a file in the S3 bucket to test. Here I have uploaded an image file.

The screenshot shows the Amazon S3 'Objects' page for the bucket 'exp12d15c'. The top navigation bar shows 'Amazon S3 > Buckets > exp12d15c'. The main area displays the 'Objects (0) Info' section. A prominent orange 'Upload' button is visible. Below it, a message states 'No objects' and 'You don't have any objects in this bucket.'. A search bar labeled 'Find objects by prefix' and a pagination control with '1' are also present.

The screenshot shows the AWS S3 console interface. At the top, there is a green banner indicating "Upload succeeded". Below it, a message says "The information below will no longer be available after you navigate away from this page." The main section is titled "Summary" and shows the destination as "s3://exp12d15c". It displays two rows: "Succeeded" with "1 file, 50.5 KB (100.00%)" and "Failed" with "0 files, 0 B (0%)". Below this, there are tabs for "Files and folders" and "Configuration", with "Files and folders" selected. Under "Files and folders", it says "(1 Total, 50.5 KB)". A table lists the file "4.2.png" with details: Name (4.2.png), Folder (-), Type (image/png), Size (50.5 KB), Status (Succeeded), and Error (-).

11. To check the log search for CloudWatch in services

The screenshot shows the AWS search results for the term "watch". The search bar at the top contains "watch". The results are categorized under "Services" and "Features". Under "Services", there are four items: CloudWatch (Monitor Resources and Applications), Athena (Serverless interactive analytics service), Amazon EventBridge (Serverless service for building event-driven applications), and Batch (Fully managed batch processing at any scale). Under "Features", there are two items: CloudWatch dashboard (Systems Manager feature) and Data sources (Athena feature). There are "Show more" buttons for both categories.

12. Select the function created in log groups

The screenshot shows the AWS CloudWatch Log Groups interface. At the top, there is a header with the title "Log groups (3)" and a "Create log group" button. Below the header is a search bar with the placeholder "Filter log groups or try prefix search". To the right of the search bar are filter options: "Exact match", a page number "1", and a settings gear icon. The main table lists three log groups:

Log group	Log class	Anomaly d...	Data p...	Sensiti...	Retenti...	Metric
/aws/lambda/lambdaexp11	Standard	Configure	-	-	Never expire	-
/aws/lambda/myLambdaFunction	Standard	Configure	-	-	Never expire	-
/aws/lambda/testfunc	Standard	Configure	-	-	Never expire	-

13. Select the log stream

The screenshot shows the AWS CloudWatch Log Streams interface. At the top, there is a header with the title "Log streams (1)" and a "Create log stream" button. Below the header is a search bar with the placeholder "Filter log streams or try prefix search". To the right of the search bar are filter options: "Exact match", "Show expired", and a settings gear icon. The main table lists one log stream:

Log stream	Last event time
2024/10/08/[\$LATEST]6ab0c7431681419a92e9154a255e1ad3	2024-10-08 16:31:29 (UTC)

14. Here we can see that the message has been displayed in the logs.

The screenshot shows the AWS CloudWatch Log Events interface. The table displays log entries for a Lambda function execution:

Timestamp	Message
No older events at this moment. Retry	
2024-10-08T18:22:36.145Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ca2e2714ff5637bd2bbe06c...
2024-10-08T18:22:36.264Z	START RequestId: 784e2efc-219c-4d70-8197-a0ca3df2f568 Version: \$LATEST
2024-10-08T18:22:36.265Z	An image has been added to the bucket exp12d15c: 4.2.png
2024-10-08T18:22:36.267Z	END RequestId: 784e2efc-219c-4d70-8197-a0ca3df2f568
2024-10-08T18:22:36.270Z	REPORT RequestId: 784e2efc-219c-4d70-8197-a0ca3df2f568 Duration: 2.17 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory U...
2024-10-08T18:22:46.119Z	START RequestId: 57941134-711d-4df0-b862-6d58b80f3a61 Version: \$LATEST
2024-10-08T18:22:46.121Z	An image has been added to the bucket exp12d15c: 4.2.png
2024-10-08T18:22:46.121Z	END RequestId: 57941134-711d-4df0-b862-6d58b80f3a61
2024-10-08T18:22:46.121Z	REPORT RequestId: 57941134-711d-4df0-b862-6d58b80f3a61 Duration: 1.65 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory U...
No newer events at this moment. Auto retry paused . Resume	

Conclusion: In this experiment, we successfully created an S3 bucket and configured it as a trigger for the Lambda function created in the previous experiment. Upon the occurrence of an event, such as uploading a file to the S3 bucket, the Lambda function was triggered, producing the output message: "An image has been added to the bucket." We have successfully demonstrated the effective integration of S3 and Lambda, allowing automatic responses to file uploads in the bucket.