

Exp No.1A

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

Hosting Using Xampp

1.Create a File in your local directory.

Name	Date modified	Type	Size
phphello	11-08-2024 21:12	PHP Source File	1 KB

2 .Write the PHP in the File.

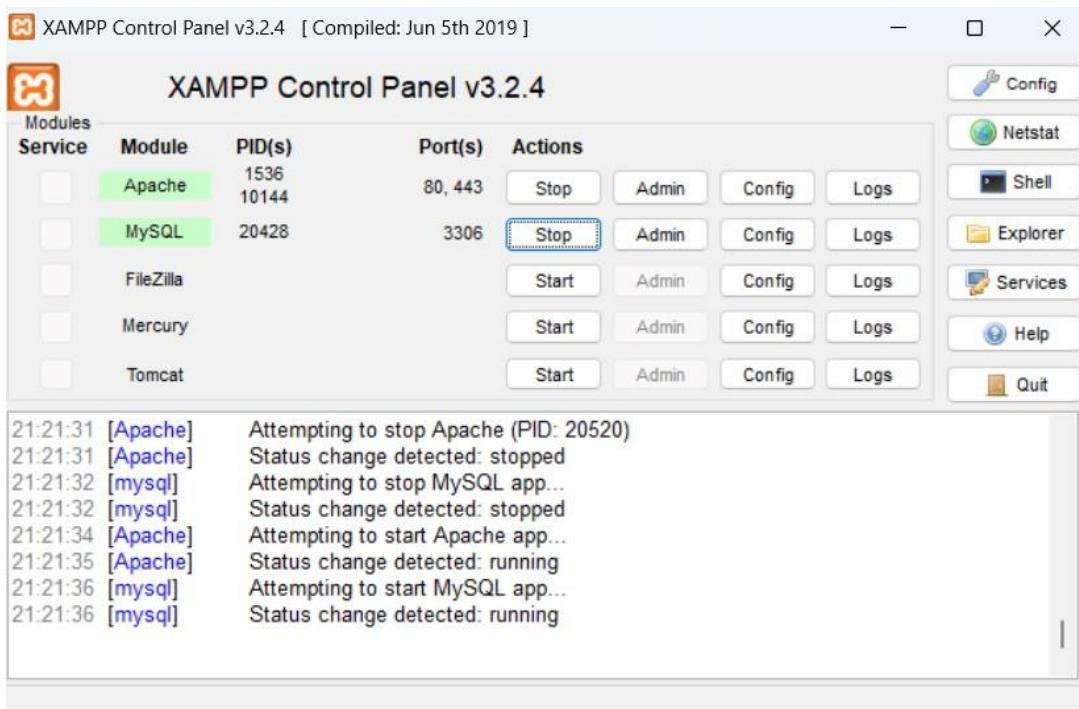
```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

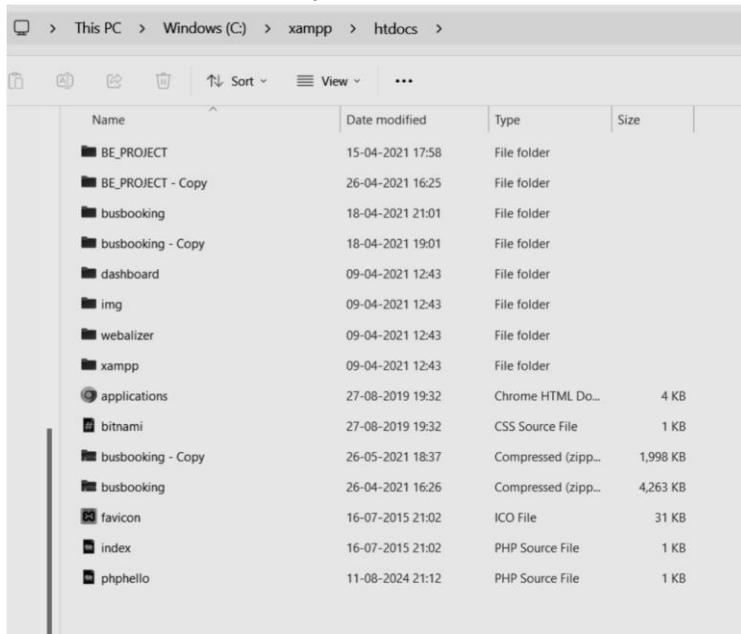
<?php
echo "Hello World!";
?>

</body>
</html>
```

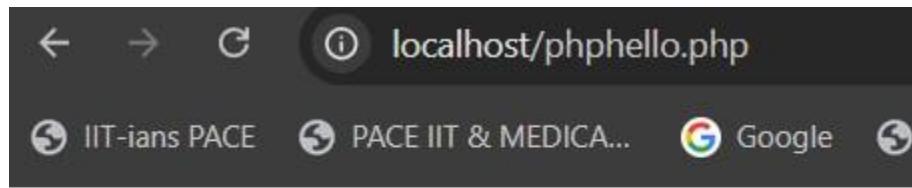
3. Turn on Xampp Control panel.
And Start the Apache Module and MySQL Module.



4. Paste the File Which you have made in This PC → Windows(c) → Xammp → htdocs folder



5.Type localhost.<filename>.php in the browser Here the File name is “phphello”.



My first PHP page

Hello Sahil

Exp 1A Part 2 :Static Hosting Using AWS

1.Start the AWS lab and Navigate to the services and choose S3 bucket and Click on create Bucket.

Name the bucket according to you.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The top navigation bar shows 'Amazon S3 > Buckets > Create bucket'. The main title is 'Create bucket' with an 'Info' link. A sub-instruction says 'Buckets are containers for data stored in S3.' Below this is a 'General configuration' section. Under 'AWS Region', 'US East (N. Virginia) us-east-1' is selected. Under 'Bucket type', 'General purpose' is selected (indicated by a blue dot). A detailed description of General purpose buckets follows. There is also a 'Directory - New' option with its own description. The 'Bucket name' field contains 'sahilm'. A note below it states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)' with a help icon. A 'Copy settings from existing bucket - optional' section is present, with a 'Choose bucket' button and a note about copied settings. A 'Format: s3://bucket/prefix' placeholder is shown at the bottom. The overall background is light gray with white and light blue UI elements.

2. Bucket will be created.

3. Configure the Edit Static website hosting page.

The screenshot shows the 'Edit Static website hosting' page for the 'sahilm' bucket in the AWS S3 console. At the top, a green banner says 'Successfully created bucket "sahilm"' and 'To upload files and folders, or to configure additional bucket settings, choose View details.' with a 'View details' button. Below this is the 'Buckets' navigation bar. The main content area shows an 'Account snapshot' with an update frequency of 'updated every 24 hours' and a 'All AWS Regions' link. It includes a 'Storage lens provides visibility into storage usage and activity trends' note and a 'View Storage Lens dashboard' button. Below this is a 'General purpose buckets' section. It lists 'General purpose buckets (1)' with an 'Info' link and an 'All AWS Regions' link. A note says 'Buckets are containers for data stored in S3.' A search bar 'Find buckets by name' is available. The table lists one bucket: 'sahilm' (Name), 'US East (N. Virginia) us-east-1' (AWS Region), 'View analyzer for us-east-1' (IAM Access Analyzer), and 'August 4, 2024, 21:44:05 (UTC+05:30)' (Creation date). Action buttons include 'Create bucket' (orange), 'Copy ARN', 'Empty', and 'Delete'. Navigation icons like back, forward, and refresh are at the bottom right. The overall background is white with green and orange UI elements.

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.

sahil.html

Error document - optional

This is returned when an error occurs.

error.html

4. Upload the File Which want to Host.

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, 432.0 B)

[Remove](#) [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	index.html	-	text/html

Destination [Info](#)

Destination
<s3://sahilm>

5.The file Will be uploaded.

The screenshot shows a green header bar with the message "Upload succeeded" and "View details below." Below this, a note 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://sahilm". Under "Succeeded", it lists "1 file, 432.0 B (100.00%)". Under "Failed", it lists "0 files, 0 B (0%)". At the bottom, there are tabs for "Files and folders" (selected) and "Configuration".

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	432.0 B	Succeeded	-

6. Edit the Bucket Policies and use this code replace the sahilm with the bucket name Here my bucket name is sahilm.

The screenshot shows the "Edit bucket policy" page. It includes sections for "Bucket ARN" (arn:aws:s3:::sahilm) and "Policy". The policy JSON is as follows:

```

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

```

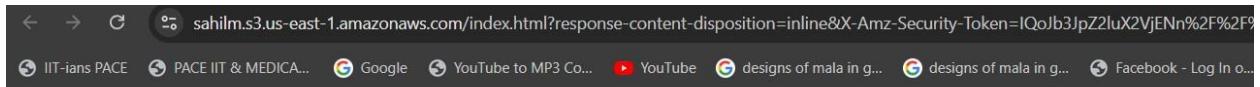
A portion of the JSON code (lines 5-9) is highlighted with a gray background.

Name:Sahil Motiramani

Roll no 35

Div D15C

7. After editing Policy the link will be generated of static hosting.

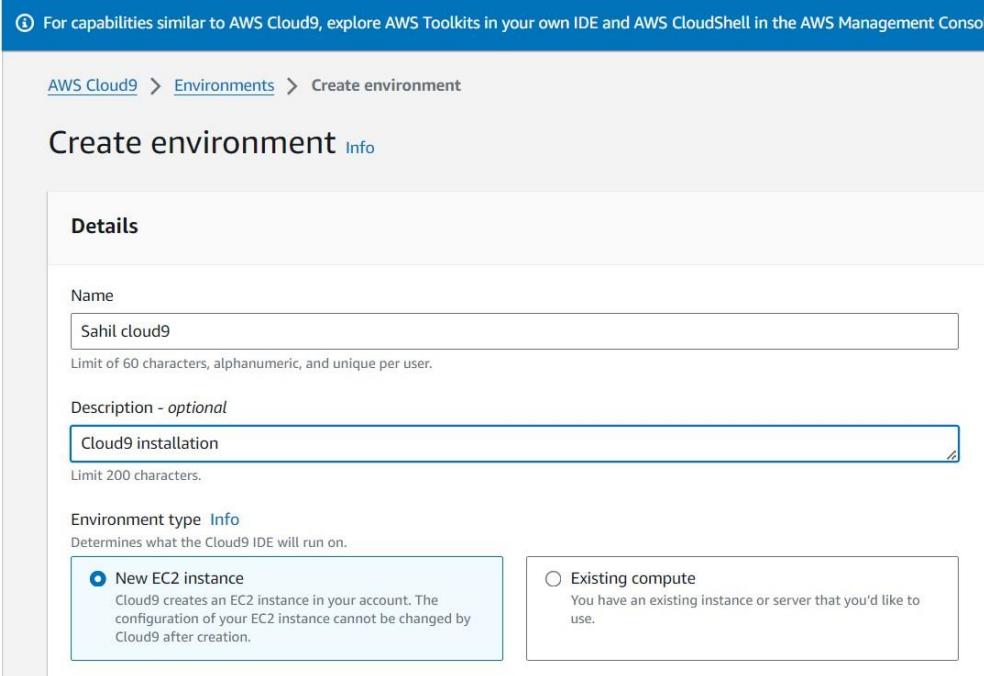


My Portfolio

Name:Sahil Motiramani

Exp 1b: Cloud9 Setup and Launch, Collaboration demonstration by creation of IAM groups and users.

1. Open your AWS account and search for Cloud9 service inside Developer tools. Create a new Cloud9 environment by filling in the required details. Make sure you use an EC2 instance to create your environment.



The screenshot shows the 'Create environment' wizard in the AWS Management Console. The current step is 'Details'. The 'Name' field is filled with 'Sahil cloud9'. The 'Description - optional' field contains 'Cloud9 installation'. Under 'Environment type', the 'New EC2 instance' option is selected. The 'Instance type' section shows options for t2.micro, t3.small, m5.large, and additional instance types. The 'Platform' section is set to 'Amazon Linux 2023'. The 'Timeout' section is set to '30 minutes'.

Details

Name
Sahil cloud9
Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional*
Cloud9 installation
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.

New EC2 instance

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

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

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

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

Additional instance types
Explore additional instances to fit your need.

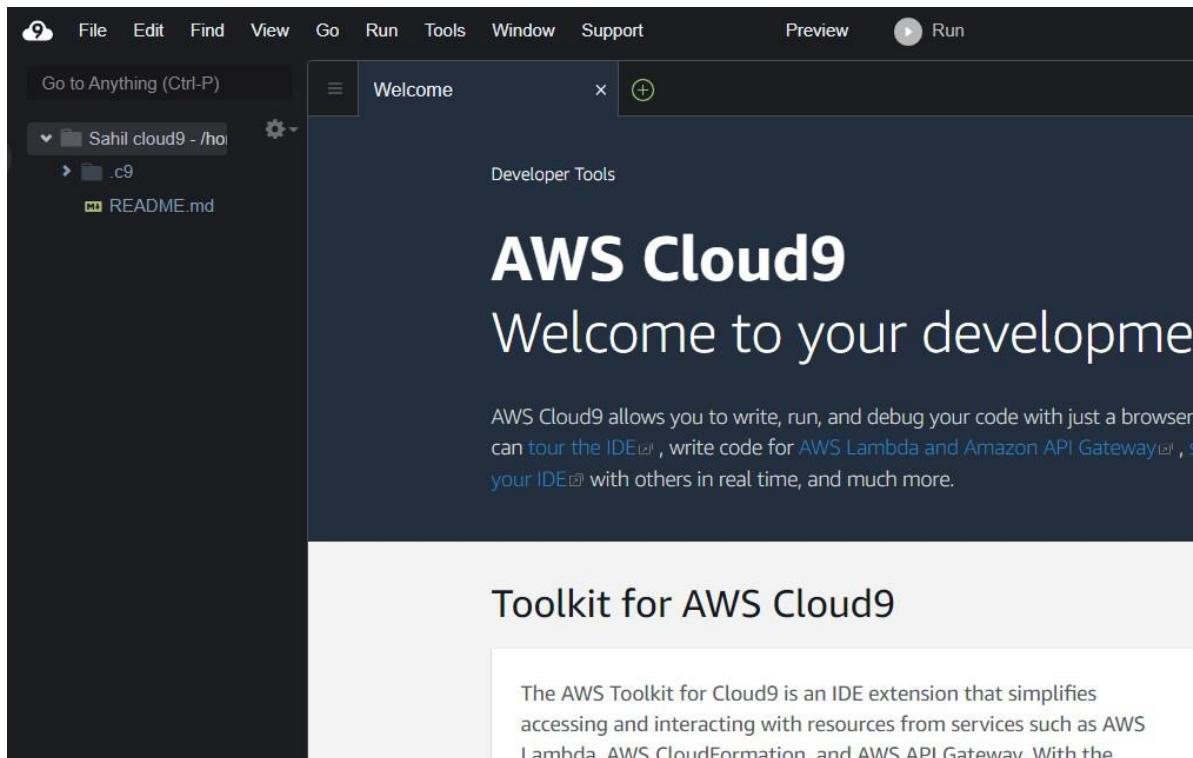
Platform [Info](#)
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

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

30 minutes

2. We have successfully setup and launched our Cloud9 environment. Over here, we can build and develop programs as per our desire. We are also allowed to collaborate with multiple other users and access shared resources.



3. Moving on, we are supposed to create a new user. Give a suitable name to the user and decide the password for the same.

Specify user details

User details

User name

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

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

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

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.

 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

- Similarly, create a new group and provide a suitable name for the same. Include the IAM users in this group together for our convenience i.e to provide similar kinds of permissions to the entire group rather than an individual user.

The screenshot shows the AWS IAM User Groups creation interface. At the top, a green banner indicates "MSBCLOUD9 user group created". Below this, there are three options: "Add user to group" (selected), "Copy permissions", and "Attach policies directly". The "User groups (1/1)" section shows a table with one item: "MSBCLOUD9" (Group name), 0 Users, and 2024-07-29 (Created). At the bottom, there is an optional step to "Set permissions boundary". Navigation buttons "Cancel", "Previous", and "Next" are visible at the bottom right.

5. The user has successfully been created i.e There is a custom made username and a password for the IAM user.

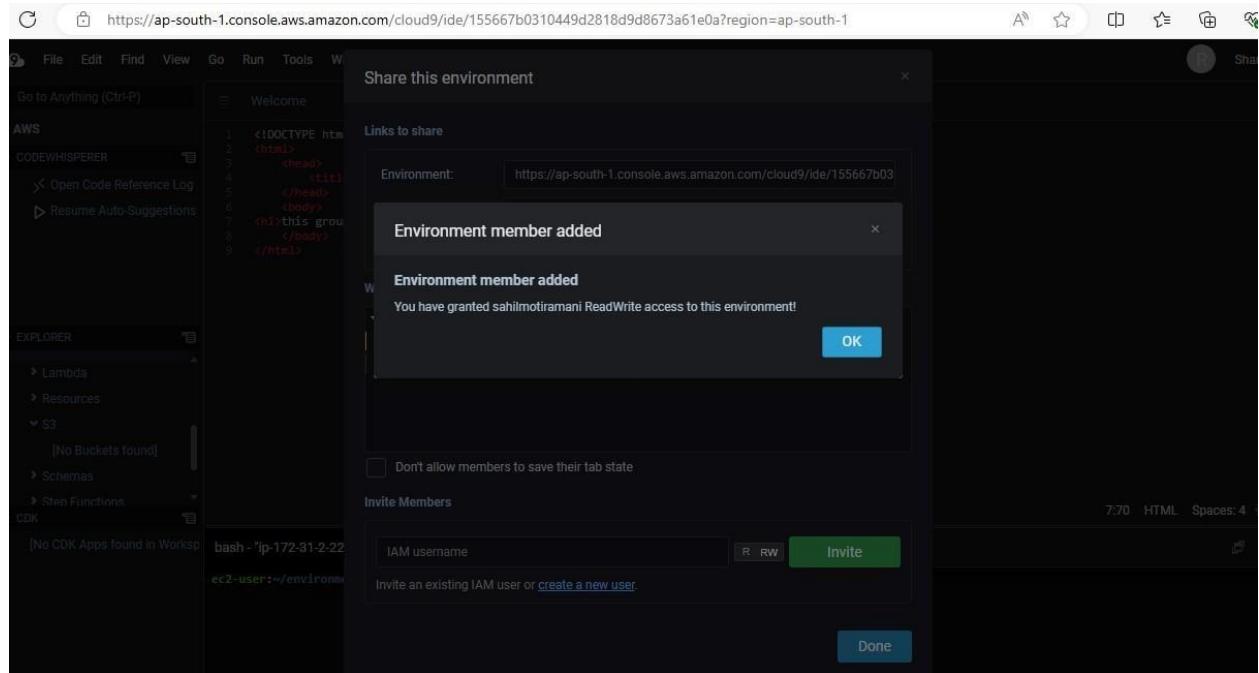
The screenshot shows the AWS IAM 'Create New User' page. At the top, a green banner says 'JD9 user group created.' Below it, the 'User details' section shows a user named 'sahilmotiramani' with a 'Custom password' type and 'Yes' for 'Require password reset'. In the 'Permissions summary' section, two policies are listed: 'IAMUserChangePassword' (AWS managed, used as Permissions policy) and 'MSBCLLOUD9' (Group, used as Permissions group). The 'Tags - optional' section indicates no tags are associated with the resource.

6. Go back to the cloud9 environment. Click on share this environment option so as to allow other collaborators to access you environment. Include your newly made IAM user in this environment and enable Read/Write permissions for it

The screenshot shows the 'Share this environment' dialog in the AWS Cloud9 interface. It displays sharing links for the environment ('https://ap-south-1.console.aws.amazon.com/cloud9/ide/155667b0310449d2818d9d8673a61e0a?region=ap-south-1') and application ('65.0.138.120'). Under 'Who has access', 'ReadWrite' permissions are granted to 'You (online)' and the newly created IAM user 'sahilmotiramani (offline)'. A checkbox for 'Don't allow members to save their tab state' is present. In the 'Invite Members' section, the user 'bhushanmalpani' is invited, with 'R RW' permissions selected. A 'Done' button is at the bottom right.

7. Further, we are supposed to login from another browser using the credentials of the IAM user, so as to access the shared cloud9 environment with us.

These steps could not be completed because Cloud9 services have been disrupted and there is no access to the IAM user from the remote login



Exp:2

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

- 1.Login to your AWS account and got to services , search for Elastic Beanstalk in search box open up Elastic Beanstalk and Create Your environment.

Environment name	Health	Application	Platform	Domain	Running
Advdevops33-env	Suspended	advdevop...	PHP 8.3 r...	Advdevops33-env.eba-icmyh9...	code-pip...
Beanstalk4-env-1	Suspended	beanstalk4	PHP 8.3 r...	Beanstalk4-env-1.eba-rs5bqky...	code-pip...

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

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

Application information [Info](#)

Application name:

Maximum length of 100 characters.

Application tags (optional)

Environment information [Info](#)

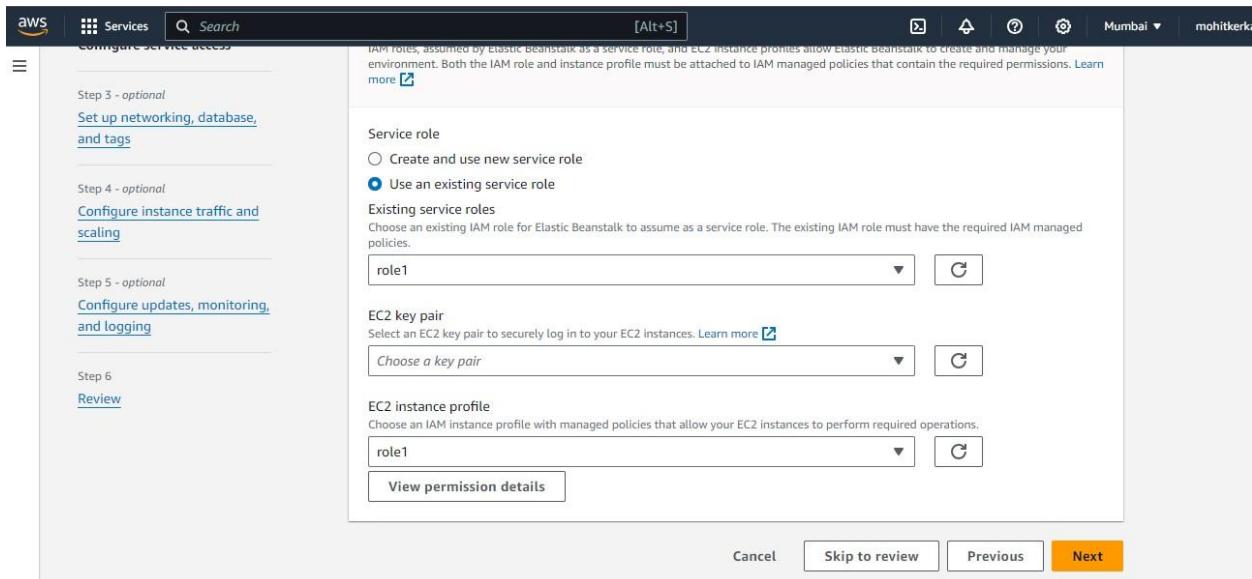
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name:

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

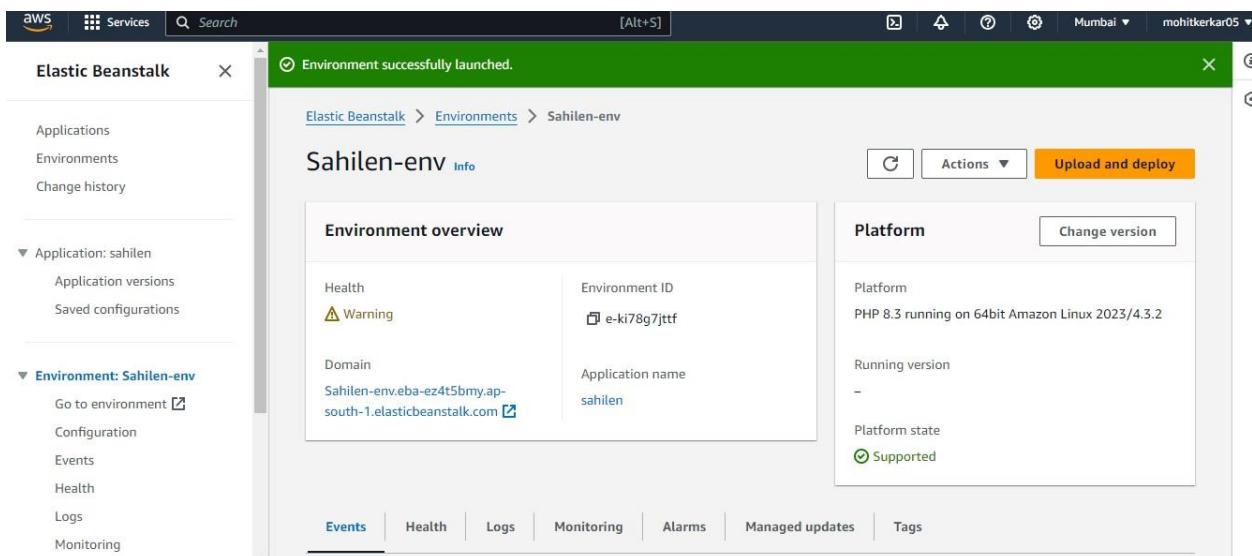
2. Give Name to tour application.

3. Configure the Service roles.



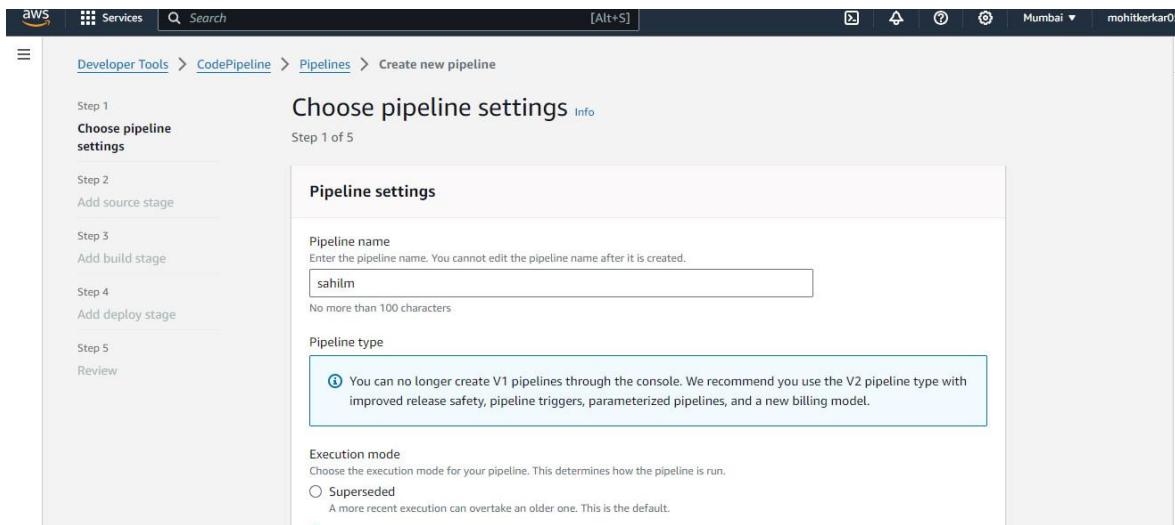
Skip the Further steps and directly go to the review part.

4.Click on Create environment,Then your environment will be created.

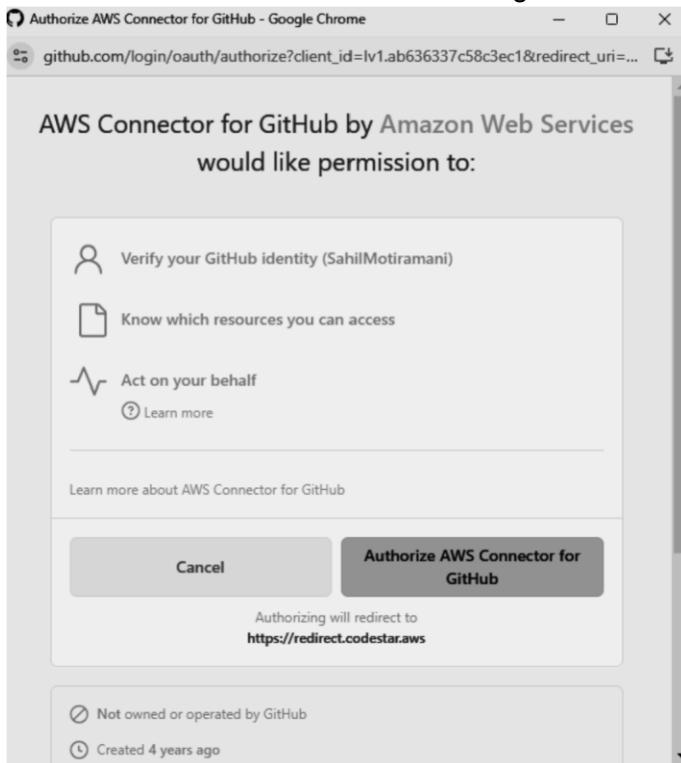


Step 2: Create a new Pipeline.

1. Now, Navigate to the services and Search for Code Pipeline → Pipelines → Create new pipeline.



2. Connect AWS with GitHub and give the access.



3.Connect to GitHub.

The screenshot shows the 'Create connection' page for GitHub within the AWS CodePipeline interface. The URL in the browser is ap-south-1.console.aws.amazon.com/codesuite/settings/connections/create/github?r.... The page title is 'Create connection | CodePipeline | ap-south-1 - Google Chrome'. The navigation bar includes 'aws' logo, 'Services', search, and 'More' dropdown. Below the navigation is a breadcrumb trail: 'Developer Tools > ... > Create connection'. The main heading is 'Connect to GitHub'. A section titled 'GitHub connection settings' contains a 'Connection name' input field with the value 'SahilMotiramani'. An optional 'App installation' section allows searching for an app by ID (53842597) or installing a new one. A 'Tags' section is also present. At the bottom right is a large orange 'Connect' button. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice: '© 2024, Amazon Web Services, Inc. or its affiliates.'

4.Fork the repository named aws-codepipeline-s3-codedeploy-linux-2.0.

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

Code Pull requests Actions Projects Wiki Security Insights Settings

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

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

master 1 Branch 0 Tags Go to file Add file <> Code About

This branch is up to date with imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0:master .

Contribute Sync fork

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

.github Adding template 7 years ago

dist Added dist folder 9 years ago

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

CODE_OF_CONDUCT.md Adding CONTRIBUTING/CoC 7 years ago

CONTRIBUTING.md Adding CONTRIBUTING/CoC 7 years ago

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

Readme Apache-2.0 license Activity 0 stars 0 watching 0 forks

Releases

5.select the repository which you have forked.

aws Services Search [Alt+S] Mumbai mohitkerkar05

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

am:aws:codeconnections:ap-south-1:010526275007:connection/779d2672- or Connect to GitHub

Ready to connect Your GitHub connection is ready for use.

Repository name Choose a repository in your GitHub account.

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

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

master

Output artifact format Choose the output artifact format.

CodePipeline default AWS CodePipeline uses the default zip format for artifacts

Full clone AWS CodePipeline passes metadata about the repository

6.select the application and environment name.

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

Region
Asia Pacific (Mumbai)

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

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.
sahilen

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.
Sahilen-env

Configure automatic rollback on stage failure

7.The pipeline will be created.

Success
Congratulations! The pipeline Sahilm has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > Sahilm

Sahilm

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: g11df979-ce02-4d28-a944-4c57a34c6722

Source GitHub (Version 2) Succeeded - 1 minute ago 8fd5da54

View details

Source: Update README.md

8. Make some changes in the forked repository using github

Here, I have made a change by adding my name “Sahil” after “Congratulations”.

```
14
15     h1 {
16         font-size: 500%;
17         font-weight: normal;
18         margin-bottom: 0;
19     }
20
21     h2 {
22         font-size: 200%;
23         font-weight: normal;
24         margin-bottom: 0;
25     }
26     </style>
27 </head>
28 <body>
29     <div align="center">
30         <h1>Congratulations Sahil!</h1>
31         <h2>You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it
32             to three Amazon EC2 instances using AWS CodeDeploy.</h2>
33         <p>For next steps, read the AWS CodePipeline Documentation. Incedge 2020</p>
34     </div>
35 </body>
36 </html>
37
38
```

9.Commit the changes and Reload the environment the changes will be seen.

Congratulations Sahil!

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

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

EXP No.:3

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

Step 1: Create 3 EC2 instances with all running on Amazon Linux as OS Kubernetes-master, kuberene-node1, kubernetes-node2.

Instances (3) Info								
Last updated less than a minute ago C Connect Instance state ▾ Actions ▾ Launch instances ▾								
Find Instance by attribute or tag (case-sensitive) All states ▾								
Instance state = running X Clear filters								
	Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
<input type="checkbox"/>	kuberene-node1	i-020814342b31e498d	Running Q Q	t2.micro	Initializing	View alarms +	us-east-1f	ec2-44-210-7
<input type="checkbox"/>	kubernetes-node1	i-07178de832f08d434	Running Q Q	t2.micro	Initializing	View alarms +	us-east-1f	ec2-44-200-3
<input type="checkbox"/>	kubernetes-node2	i-01f1908139e14493e	Running Q Q	t2.micro	Initializing	View alarms +	us-east-1f	ec2-3-238-71

Step 2: SSH into all 3 machines each in separate terminal for each instance,

```
ubuntu@ip-172-31-73-42: ~
HP@LAPTOP-B2H2GRPI MINGW64 ~
$ cd downloads
HP@LAPTOP-B2H2GRPI MINGW64 ~/downloads
$ chmod 400 "sahil.pem"

HP@LAPTOP-B2H2GRPI MINGW64 ~/downloads
$ ssh -i "sahil.pem" ubuntu@ec2-44-222-111-79.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sat Sep 14 13:13:00 UTC 2024

 System load:  0.0      Processes:          107
 Usage of /:   8.4% of 18.33GB   Users logged in:     1
 Memory usage: 20%           IPv4 address for enx0: 172.31.73.42
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

Last login: Sat Sep 14 13:10:15 2024 from 103.160.108.205
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-73-42:~$ |
```

Step 3:Now install docker in all 3 instances;

`sudo apt-get intall -y docker.io`

```
ubuntu@ip-172-31-73-42:~$ sudo apt-get install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 7 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willsync1 [4450 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (74.6 MB/s)
```

Then, Configure File daemon.json;

```
ubuntu@ip-172-31-73-42:~$ cd /etc/docker
ubuntu@ip-172-31-73-42:/etc/docker$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
ubuntu@ip-172-31-73-42:/etc/docker$ |
```

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

```
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl enable docker
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl daemon-reload
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl restart docker
ubuntu@ip-172-31-73-42:/etc/docker$ docker -v
Docker version 24.0.7, build 24.0.7-0ubuntu4.1
ubuntu@ip-172-31-73-42:/etc/docker$ |
```

Step 4:Install Kubernetes in all three instances:

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

Add kubernetes repository (paste in terminal);

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

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

```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
=====
Package           Architecture Version
=====
Installing:
kubeadm          x86_64      1.30.4-150500.1.1
kubectl          x86_64      1.30.4-150500.1.1
kubelet          x86_64      1.30.4-150500.1.1
Installing dependencies:
conntrack-tools   x86_64      1.4.6-2.amzn2023.0.2
cri-tools         x86_64      1.30.1-150500.1.1
kubernetes-cni    x86_64      1.4.0-150500.1.1
libnetfilter_cthelper x86_64  1.0.0-21.amzn2023.0.2
libnetfilter_cttimeout x86_64  1.0.0-19.amzn2023.0.2
libnetfilter_queue x86_64      1.0.5-2.amzn2023.0.2
socat             x86_64      1.7.4.2-1.amzn2023.0.2
Transaction Summary
=====
Install 10 Packages
```

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

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

Step 5: Perform this ONLY on the Master machine Initialize kubernetes by

typing below command

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

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

```
Your Kubernetes control-plane has initialized successfully!
```

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

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

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

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

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

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

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

```
Copy the mkdir and chown commands from the top and execute them
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Copy this join link and save it in clipboard (copy from your output as it different for
each instance)
kubeadm join 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
--discovery-token-ca-cert-hash
```

```
sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6c d2d2f6f8c5
```

```
Then, add a common networking plugin called flannel file as mentioned in the code.
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

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

Check the created pod using this command

- `kubectl get pods`

Step:6. For nodes only;

Use the below command on all 2 node machines

- `Sudo yum install iproute-tc -y`
- `sudo systemctl enable kubelet`
- `sudo systemctl restart kubelet`
- `kubeadm join 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \ --discovery-
token-ca-cert-hash`

Name: Sahil Motiramani

Div:D15C

Roll No:35

sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8

c5 Master control nodes;

Every 2.0s: kubectl get nodes				
NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-81-63.ec2.internal	Ready	control-plane	29m	v1.30.4
ip-172-31-87-137.ec2.internal	Ready	<none>	5m58s	v1.30.4
ip-172-31-92-18.ec2.internal	Ready	<none>	5m53s	v1.30.4

Exp:04

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

Theory `kubectl` is the command-line tool used to interact with Kubernetes clusters.

It serves as the primary interface for managing and orchestrating containers in a Kubernetes environment. By sending commands to the Kubernetes API server, `kubectl` allows you to control clusters, manage workloads, and inspect resource states.

To begin using Kubernetes, installing `kubectl` is essential. The installation process varies based on the operating system (Linux, Windows, or macOS). After installing, `kubectl` connects to the Kubernetes cluster using the `kubeconfig` file, which stores details like cluster name, server address, and access credentials. With this connection established, you can use `kubectl` to perform a variety of operations, such as creating, updating, scaling, and deleting applications.

When deploying your first application with Kubernetes, the process involves defining the application in a configuration file (usually YAML) that specifies its requirements, such as images, replicas, and networking settings. `kubectl` interprets this configuration and relays it to the Kubernetes cluster, which then manages the lifecycle of the application.

Step 1:Create an EC2 instance use ubuntu application and select t2 .medium category in instance type create a new key rsa type save it in local machine in an folder:

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a 'Create Function' button. Below that, a section titled 'HelloWorld' is shown with the message 'Function created successfully'. The 'Overview' tab is selected. Under the 'Handler' section, 'index.handler' is listed under 'File name'. The 'Runtime' is set to 'Node.js 14.x'. The 'Memory size' is 128 MB and the 'Timeout' is 3 seconds. The 'Logs' section shows a log entry with the timestamp '2021-09-14T10:45:10.000Z'. The 'Environment' section lists 'AWS_LAMBDA_FUNCTION_NAME' and 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE'. The 'Logs' section shows a log entry with the timestamp '2021-09-14T10:45:10.000Z'.

Step 2:Click create to create the instance:

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a 'Launch instances' button. Below that, a table lists one instance: 'Exp4' (Instance ID: i-09a06120376188a8d, State: Running, Type: t2.medium). The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The 'Actions' column contains a 'Stop' button for each instance.

Step 3:

Navigate to ssh client copy the key:

Connect to instance Info

Connect to your instance i-09a06120376188a8d (Exp4) using any of these options

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

Instance ID
 i-09a06120376188a8d (Exp4)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is sahilexp4key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "sahilexp4key.pem"
4. Connect to your instance using its Public DNS:
 ec2-52-201-236-39.compute-1.amazonaws.com

Command copied

ssh -i "sahilexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com

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

[Cancel](#)

Step 4:navigate to the folder open the terminal and paste the ssh command:

```
ssh -i "sahilexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com
```

```
PS C:\Users\HP\Desktop\Adv devops exp4> ssh -i "sahilexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com
The authenticity of host 'ec2-52-201-236-39.compute-1.amazonaws.com (52.201.236.39)' can't be established.
ED25519 key fingerprint is SHA256:nITHk14RA95WSu1ku3jdi5jaDWtYkXby4850GXZFDU8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-201-236-39.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 20:25:29 UTC 2024

System load:  0.02      Processes:          120
Usage of /:   22.8% of 6.71GB   Users logged in:     0
Memory usage: 5%
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

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

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

Last login: Sun Sep 15 20:25:31 2024 from 18.206.107.29
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 5:Install docker

Use the commands given below to install docker curl -fsSL

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

```
ubuntu@ip-172-31-21-243:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://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]
```

Use:

sudo apt-get update

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored
  in apt-key(8) for details.
```

Use:

sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
```

Now the docker is installed;

Now lets enable the docker:

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

```
ubuntu@ip-172-31-21-243:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-21-243:~$ cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"] } EOF
```

sudo systemctl enable docker

```
ubuntu@ip-172-31-21-243:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-21-243:~$ |
```

sudo systemctl daemon-reload

sudo systemctl restart docker

```
ubuntu@ip-172-31-21-243:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-21-243:~$ 
sudo systemctl restart docker
ubuntu@ip-172-31-21-243:~$ |
```

Step 6:Now lets install kubernetes; curl -fsSL

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-21-243:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-21-243:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ |
ubuntu@ip-172-31-21-243:~$ |
```

sudo apt-get update

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 6051 B in 0s (12.6 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc
ecktion in apt-key(8) for details.
ubuntu@ip-172-31-21-243:~$ |
```

sudo apt-get install -y kubelet kubeadm kubectl

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 133 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (71.7 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68007 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
```

sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-21-243:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-21-243:~$ |
```

```
sudo systemctl enable --now kubelet
//Skip:sudo kubeadm init --pod-network-cidr=10.244.0.0/16
ubuntu@ip-172-31-21-243:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-21-243:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0915 20:41:32.398638      4595 checks.go:1080] [preflight] WARNING: Couldn't create t
new CRI runtime service: validate service connection: validate CRI v1 runtime API +
e = Unimplemented desc = unknown service runtime.v1.RuntimeService
      [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet
[preflight] You can also perform this action beforehand using 'kubeadm config images
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI
: rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[+
with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-21-243:~$
```

sudo apt-get install -y containerd

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 133 not upgraded.
```

sudo mkdir -p /etc/containerd sudo containerd config default | sudo tee /etc/containerd/config.toml

```
ubuntu@ip-172-31-21-243:~$ sudo mkdir -p /etc/containerd
ubuntu@ip-172-31-21-243:~$ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0
```

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

```
ubuntu@ip-172-31-21-243:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-09-15 20:45:11 UTC; 228ms ago
     Docs: https://containerd.io
     Main PID: 5089 (containerd)
        Tasks: 7
       Memory: 13.9M (peak: 14.3M)
         CPU: 58ms
        CGroup: /system.slice/containerd.service
                  └─5089 /usr/bin/containerd

Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275924779Z" level=info msg=servi
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275966419Z" level=info msg=servi
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275967171Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276021207Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276080044Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276092900Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276101205Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276111208Z" level=info msg="Starti
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276200348Z" level=info msg="conta
Sep 15 20:45:11 ip-172-31-21-243 systemd[1]: Started containerd.service - containerd container runtime.
[1 lines 1-21/21] (END)
```

sudo apt-get install -y socat

```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 133 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.
Fetched 374 kB in 0s (10.9 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68107 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
[1]+ 0 Sudo: 172 31 21 243 +
```

Step 7: Initialize the kubernetes:

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

```
ubuntu@ip-172-31-21-243:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0915 20:47:01.807699      5328 checks.go:846] detected that the sandbox image "registry.k8s.io/pa
used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
```

```
ubuntu@ip-172-31-21-243:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-21-243:~$ |
```

kubectl apply -f

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

```
ubuntu@ip-172-31-21-243:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yaml
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
ubuntu@ip-172-31-21-243:~$ |
```

Step 8:Now we can deploy our nginx server on this cluster using following steps:

```
kubectl apply -f https://k8s.io/examples/application/deployment.yaml
ubuntu@ip-172-31-21-243:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-21-243:~$ |
```

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-54h6b	0/1	Pending	0	28s
nginx-deployment-d556bf558-jw5xg	0/1	Pending	0	28s

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
```

```
ubuntu@ip-172-31-21-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
ubuntu@ip-172-31-21-243:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-21-243:~$ |
```

kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
 kubectl get nodes

```
ubuntu@ip-172-31-21-243:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
ubuntu@ip-172-31-21-243:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-21-243 Ready    control-plane   6m14s   v1.31.1
ubuntu@ip-172-31-21-243:~$ |
```

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-54h6b	0/1	Pending	0	2m31s
nginx-deployment-d556bf558-jw5xg	0/1	Pending	0	2m31s

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
```

```
ubuntu@ip-172-31-21-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
ubuntu@ip-172-31-21-243:~$ |
```

Step 9 :check deployment:

Open new terminal in folder,

Paste the ssh key,

Type

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

```
PS C:\Users\HP\Desktop\Adv devops exp4> ssh -i "sahilexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 21:05:11 UTC 2024

System load: 0.05           Processes:          155
Usage of /: 55.3% of 6.71GB   Users logged in:     1
Memory usage: 19%            IPv4 address for enX0: 172.31.21.243
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Sun Sep 15 20:25:36 2024 from 103.160.108.205
ubuntu@ip-172-31-21-243:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 21:05:56 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
```

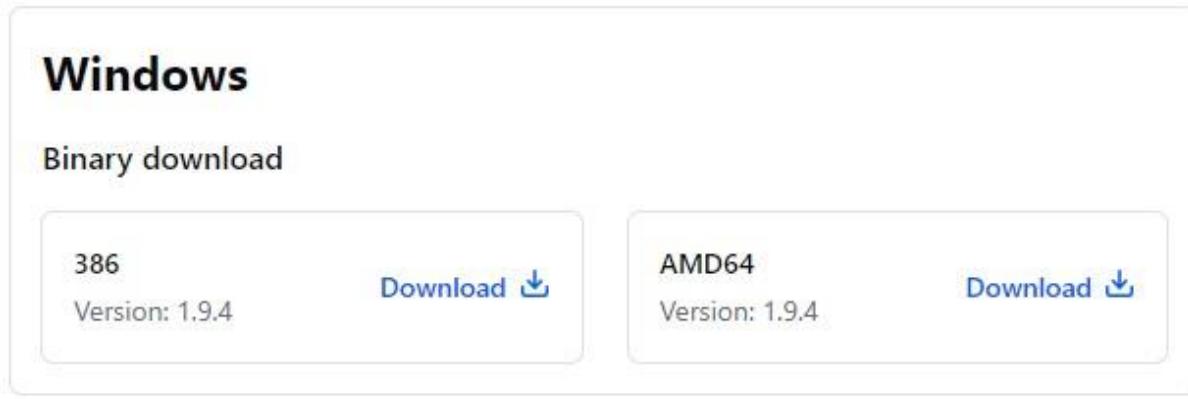
Now we have successfully deployed our nginx server on our ec2 instance.

Conclusion

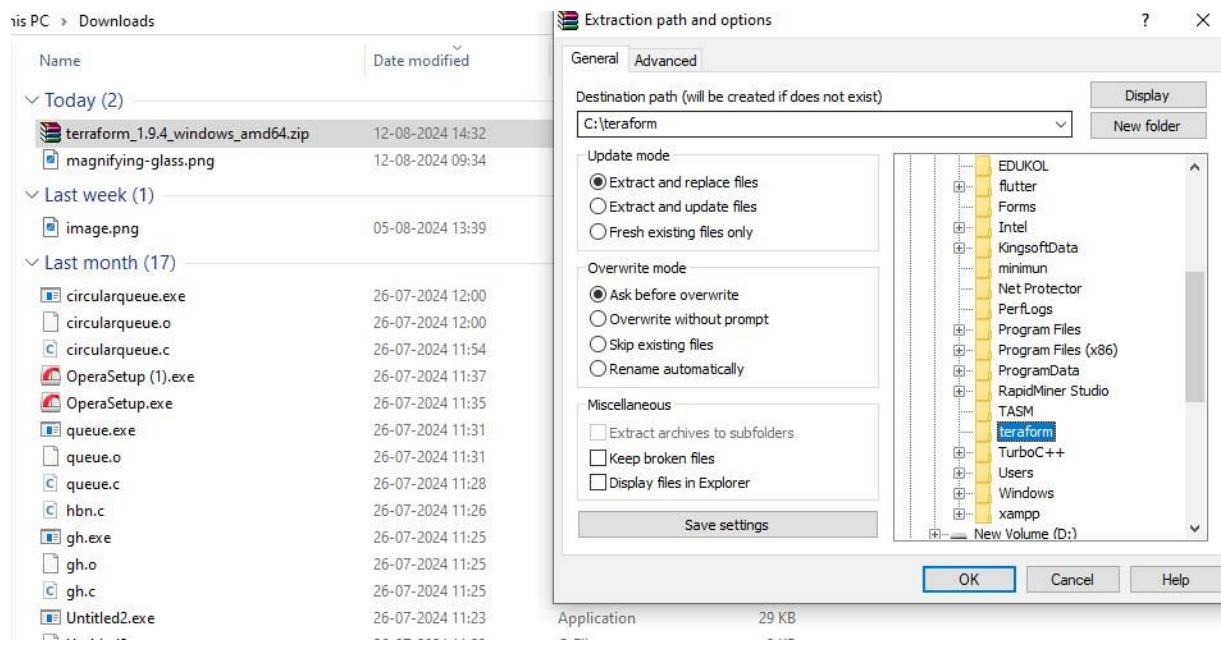
Installing `kubectl` and using it effectively is a crucial part of managing Kubernetes clusters. As the main interface to Kubernetes, `kubectl` empowers you to deploy, monitor, and troubleshoot applications, providing full control over cluster resources. Deploying an application for the first time serves as an introduction to Kubernetes' ability to orchestrate containers seamlessly. It showcases the power of declarative configurations and automated scaling, which are central to Kubernetes' efficiency in managing modern applications. Understanding how to install and operate `kubectl` lays the foundation for more advanced interactions with Kubernetes, enabling both developers and administrators to harness its full potential in creating resilient, scalable, and portable containerized applications. This knowledge is fundamental for any professional aiming to work within a Kubernetes-driven infrastructure.

Exp 05: To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.

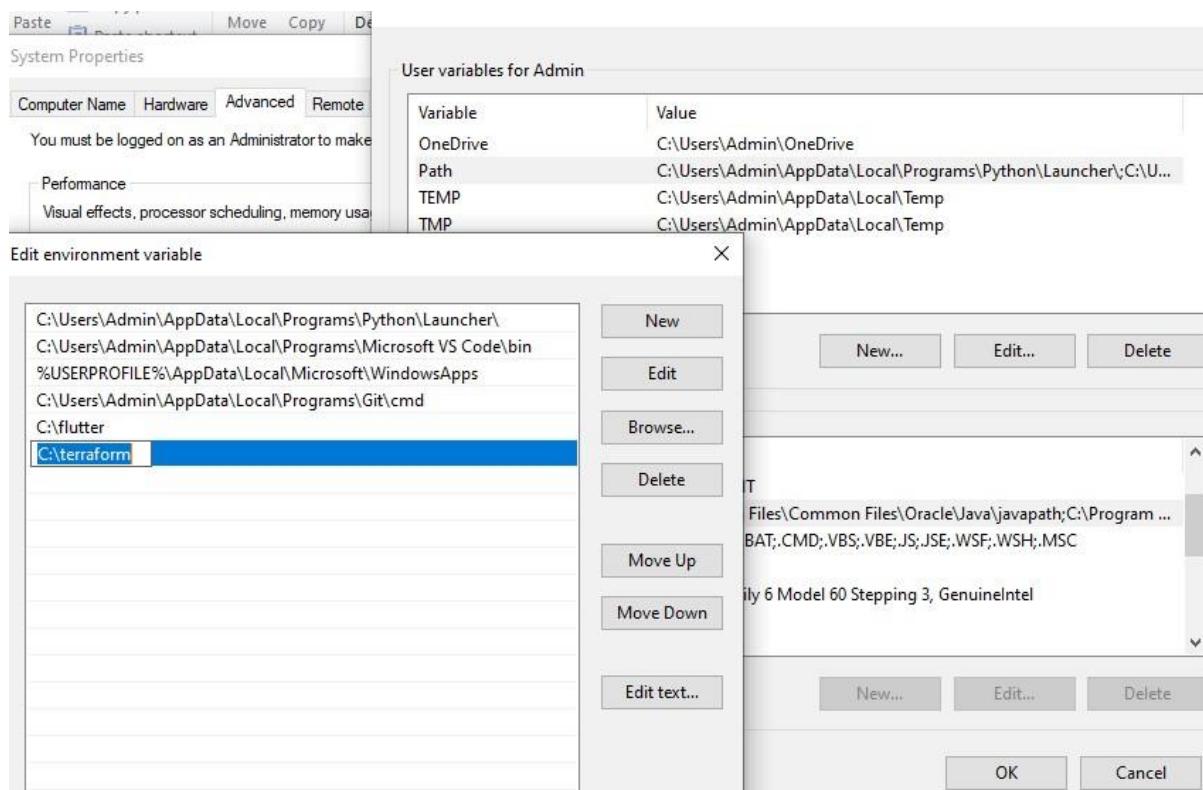
Step1: Go to hashicorp.com page and download the AMD 64 file for terraform. Install it in Windows.



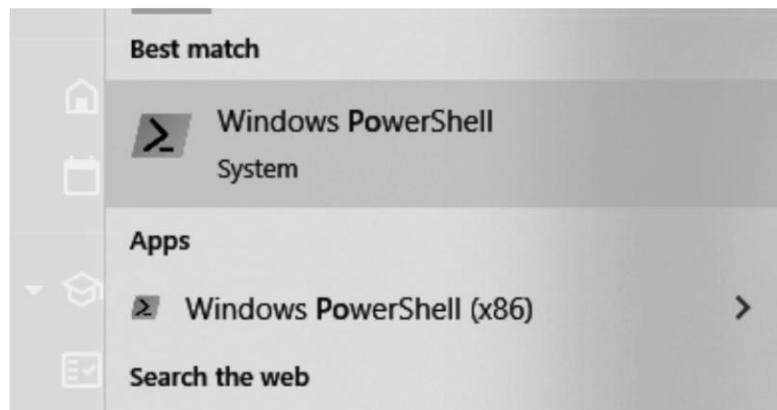
Step 2: After Downloading Extract the file and Save it inside the C drive by naming the folder terraform.



Step 3: Edit Path inside Environment Variables Add the new path of terraform
I.e C:\terraform



Step 4: Open Windows PowerShell



Step 5. Type terraform in PowerShell if There are no errors then that means the Terraform is installed successfully or else there will be a error in Environment Variables.

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

S C:\Users\Admin> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

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

A. Creating docker image using terraform

Step 1: Download Docker and Check the docker functionality

```
C:\Users\student>docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

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

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

```
C:\Users\student>docker --version
Docker version 27.0.3, build 7d4bcd8

C:\Users\student>
```

Step 2:Create a folder named 'TerraformScripts'

Then,go in that folder and again create a folder named 'Docker' Then,create a file named docker.tf And write the following code in it.

```
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"
    command = ["sleep", "3600"] |
  }
```

Step 3: Run the command terraform init.

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

Terraform has been successfully initialized!

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

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

Step 4:Run the terraform plan to see all the files initialized.

```
C:\Users\INFT\Desktop\TerraformScript\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)
```

```

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

+ healthcheck  (known after apply)

+ labels        (known after apply)
}

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

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

```

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

C:\Users\INFT\Desktop\TerraformScript\Docker>

This is image before apply:

```
C:\Users\INFT\Desktop\TerraformScript\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
```

```
C:\Users\INFT\Desktop\TerraformScript\Docker>
```

Step 5:execute the command `terraform apply`:

It will apply the configuration

```
C:\Users\INFT\Desktop\TerraformScript\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest    edbfe74c41f8  2 weeks ago  78.1MB
```

```
C:\Users\INFT\Desktop\TerraformScript\Docker>
```

```
C:\Users\INFT\Desktop\TerraformScript\Docke>terraform apply

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

Terraform will perform the following actions:

# 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

  + name          = "foo"
  + network_data = (known after apply)
  + read_only     = false
  + remove_volumes = true
  + restart       = "no"
  + rm            = 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 1s [id=6b4b627b5597135995916aaa25dad8226b7961c9b0db969caf119b02ffbc06ea]

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

Docker images,After executing terraform apply:

Step 6:Execute terraform destroy to delete the configurations;

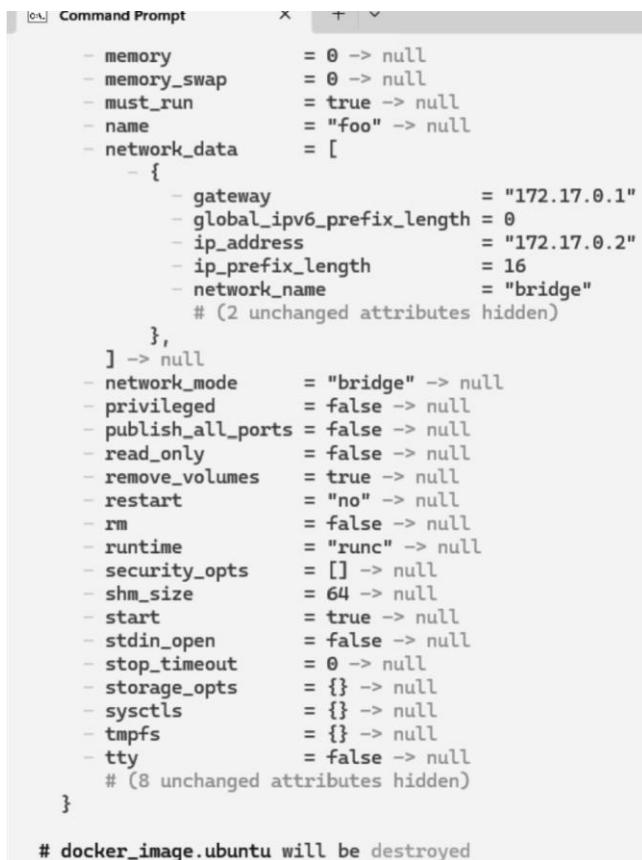
It will automatically delete the container:

```
C:\Users\INFT\Desktop\TerraformScript\Dockers>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=6b4b627b5597135995916aaa25dad8226b7961c9b0db969caf119b02ffbc06ea]

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",
    - "3600",
  ] -> 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          = "6b4b627b5597" -> null
  - id               = "6b4b627b5597135995916aaa25dad8226b7961c9b0db969caf119b02ffbc06ea" -> null
  - image             = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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
}
```



```
Command Prompt x + ▾
- memory           = 0 -> null
- memory_swap      = 0 -> null
- must_run         = true -> null
- name              = "foo" -> null
- network_data      = [
  - {
    - gateway          = "172.17.0.1"
    - global_ipv6_prefix_length = 0
    - ip_address        = "172.17.0.2"
    - ip_prefix_length   = 16
    - network_name       = "bridge"
    # (2 unchanged attributes hidden)
  },
] -> null
- network_mode      = "bridge" -> null
- privileged         = false -> null
- publish_all_ports = false -> null
- read_only          = false -> null
- remove_volumes    = true -> null
- restart            = "no" -> null
- rm                 = false -> null
- runtime             = "runc" -> null
- security_opts      = [] -> null
- shm_size           = 64 -> null
- start              = true -> null
- stdin_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
```

```

# 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=6b4b627b5597135995916aaa25dad8226b7961c9b0db969caf119b02ffbc06ea]
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.

```

Docker images After destroying,

```

C:\Users\INFT\Desktop\TerraformScript\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE

C:\Users\INFT\Desktop\TerraformScript\Docker>

```

Adv DevOps Practical 7

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

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

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

S	W	Name	Last Success	Last Failure	Last Duration
Green	Sun	sahil 7	24 days #2	N/A	96 ms
Green	Sun	Sahil exp6	24 days #3	N/A	1 sec
Blue	Sun	SahilExp6	N/A	N/A	N/A
Red	Cloud	sahiljob	N/A	24 days #1	1.5 sec

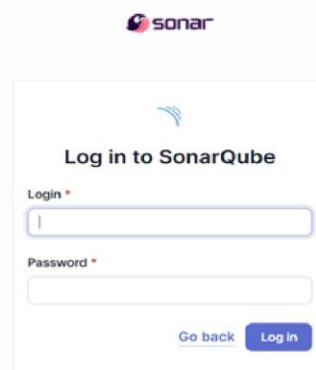
2. Run SonarQube in a Docker container using this command -

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

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

```
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
st
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
c1c57f6ace6123c4160745bf57640c75fcbe70225eabb02b7bf7a40450e8001
```

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



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

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

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

Import from Azure DevOps Import from Bitbucket Cloud Import from Bitbucket Server
Import from GitHub Import from GitLab

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

Create a local project

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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

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

Setup the project and come back to Jenkins Dashboard.

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

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	7 mo 6 days ago
<input type="checkbox"/>	External Site/Tool Integrations Build Reports	
<input type="checkbox"/>	Sonar Gerrit 388.v9b_ftcb_e42306	3 mo 20 days ago
<input type="checkbox"/>	SonarQube Generic Coverage 1.0 TODO	5 yr 1 mo ago

- Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **sahilexp7**

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

SonarQube installations
List of SonarQube installations

Name: sahilexp7

Server URL: http://localhost:9000

Server authentication token: - none -

Advanced ▾

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

Dashboard > Manage Jenkins > Tools

Dashboard > Manage Jenkins > Tools

Add Git ▾

Gradle installations

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

Add Ant

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

The screenshot shows the Jenkins Manage Jenkins > Tools page. A new SonarQube Scanner tool is being configured. The 'Name' field is set to 'sonarqube_exp7'. The 'Install automatically' checkbox is checked. Under 'Install from Maven Central', the 'Version' is set to 'SonarQube Scanner 6.2.0.4584'. There is also an 'Add Installer' button. Below this, there is another 'Add SonarQube Scanner' button. At the bottom, there are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins New Item creation page. In the 'Enter an item name' field, 'exp7' is typed. Under 'Select an item type', the 'Freestyle project' option is selected, described as a '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.' Other options shown include 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom is an 'OK' button.

9. Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git

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

The screenshot shows the 'Source Code Management' configuration page. Under the 'Git' tab, a repository URL is set to `https://github.com/shazforiot/MSBuild_firstproject.git`. The 'Credentials' dropdown is set to '- none -'. There is an 'Advanced' button and a 'Save' button at the bottom.

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

The screenshot shows the 'Build Environment' configuration page. A modal window is open, listing various build steps. The 'Execute SonarQube Scanner' option is highlighted. Other options include 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', 'Set build status to "pending" on GitHub commit', 'SonarScanner for MSBuild - Begin Analysis', and 'SonarScanner for MSBuild - End Analysis'. At the bottom of the modal is an 'Add build step' button.

Then save

The screenshot shows the SonarQube dashboard for the project 'exp7'. On the left, there's a sidebar with icons for Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area displays a green checkmark icon and the project name 'exp7'. Below this is the SonarQube logo and the word 'Permalinks'. To the right, a list of recent builds is shown:

- Last build (#20), 4 min 15 sec ago
- Last stable build (#20), 4 min 15 sec ago
- Last successful build (#20), 4 min 15 sec ago
- Last failed build (#18), 6 min 38 sec ago
- Last unsuccessful build (#19), 4 min 55 sec ago
- Last completed build (#20), 4 min 15 sec ago

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

The screenshot shows the SonarQube Administration - Security page. It lists groups and their permissions:

Group	Administer System	Administer	Execute Analysis	Create
sonar-administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
sonar-users	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
Anyone DEPRECATED	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
Administrator admin	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects

At the bottom, it says '4 of 4 shown'.

IF CONSOLE OUTPUT FAILED:

Step 1: Generate a New Authentication Token in SonarQube

1. Login to SonarQube:

- Open your browser and go to `http://localhost:9000`.
- Log in with your admin credentials (default username is `admin`, and the password is either `admin` or your custom password if it was changed).

2. Generate a New Token:

- Click on your `username` in the top-right corner of the SonarQube dashboard.
- Select **My Account** from the dropdown menu.
- Go to the **Security** tab.
- Under **Generate Tokens**, type a name for the token (e.g., "Jenkins-SonarQube").
- Click **Generate**.
- Copy the token and save it securely. You will need it in Jenkins.

Step 2: Update the Token in Jenkins

1. Go to Jenkins Dashboard:

- Open Jenkins and log in with your credentials.

2. Configure the Jenkins Job:

- Go to the job that is running the SonarQube scanner (`adv_devops_exp7`).
- Click **Configure**.

3. Update the SonarQube Token:

- In the SonarQube analysis configuration (either in the pipeline script or under "Build" section, depending on your job type), update the `sonar.login` parameter with the new token.

Dashboard > exp7 > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Execute SonarQube Scanner

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

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=sahilexp
sonar.host.url=http://localhost:9000
sonar.login=sqa_6a06a6333aecc198878c652a50fcf3f1e9cf82c1
sonar.sources=.
```

Additional arguments ?

JVM Options ?

Save **Apply**

12. Run the Jenkins build.

Dashboard > exp7 >

Status exp7

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

SonarQube

Permalinks

- Last build (#20), 8 min 21 sec ago
- Last stable build (#20), 8 min 21 sec ago
- Last successful build (#20), 8 min 21 sec ago
- Last failed build (#18), 10 min ago
- Last unsuccessful build (#19), 9 min 1 sec ago
- Last completed build (#20), 8 min 21 sec ago

Build History trend ▾

Filter... /

#20 | Sep 23, 2024, 7:49 PM

Check the console Output

Console output;

The screenshot shows the Jenkins interface with the path: Dashboard > exp7 > #20 > Console Output. The 'Console Output' tab is selected. The log output is as follows:

```

Started by user Sahil Motiramani
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\exp7\.git # t
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # t
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.43.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git

```

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

The screenshot shows the SonarQube main dashboard for the project 'sahilexp7'. The 'Overview' tab is selected. The 'main' branch is shown with a green checkmark icon and the word 'Passed'. A warning message says: 'The last analysis has warnings. See details'. Below the main section, there is an 'Activity' section.

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

Conclusion:

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

Exp:8

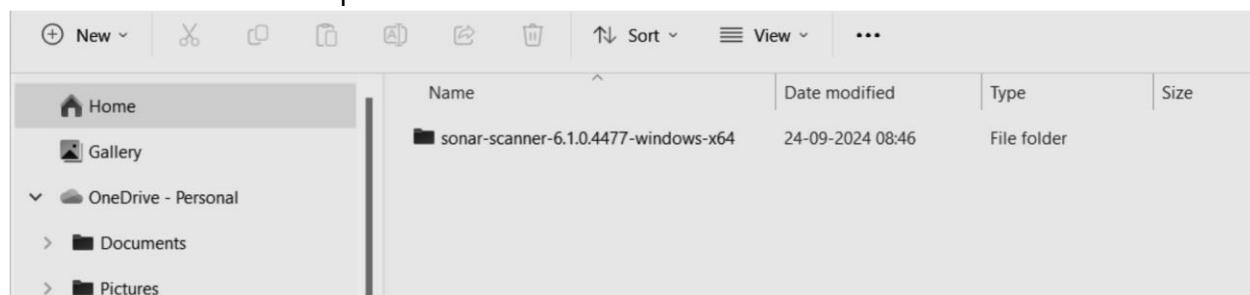
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application. Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>

The screenshot shows the 'SonarScanner CLI' page under the 'Analyzing source code' section. It features a sidebar with links like 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Scanners', and 'SonarScanner CLI'. The main content area displays version 6.1 details, including download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM), and Release notes. A sidebar on the right lists 'On this page' topics such as Configuring your project, Running SonarScanner CLI from the zip file, and Scanning C, C++, or Objective-C projects.

Visit this link and download the sonarqube scanner CLI.

Extract the downloaded zip file in a folder.



1. Install sonarqube image

Command: docker pull sonarqube

```
C:\Users\HP\Desktop\sem5\advdevops8>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

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

C:\Users\HP\Desktop\sem5\advdevops8>
```

```
C:\Users\HP\Desktop\sem5\advdevops8>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
a57154161e14bed00ec141b755fa197a52321bf5c0688b825ff4dfbeaf712099
```

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



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

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

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

Import from Azure DevOps Import from Bitbucket Cloud Import from Bitbucket Server

Import from GitHub Import from GitLab

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

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

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

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](#)

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Admin

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. T
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.

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

The Jenkins dashboard displays the following information:

- Build History:** Shows recent builds for various projects like exp7, sahil 7, Sahil exp6, and SahilExp6.
- Build Executor Status:** Shows the status of built-in nodes, with one node labeled "Sahil" marked as offline.

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

Dashboard > Manage Jenkins > Plugins

Plugins

Updates Available plugins (16) Installed plugins Advanced settings

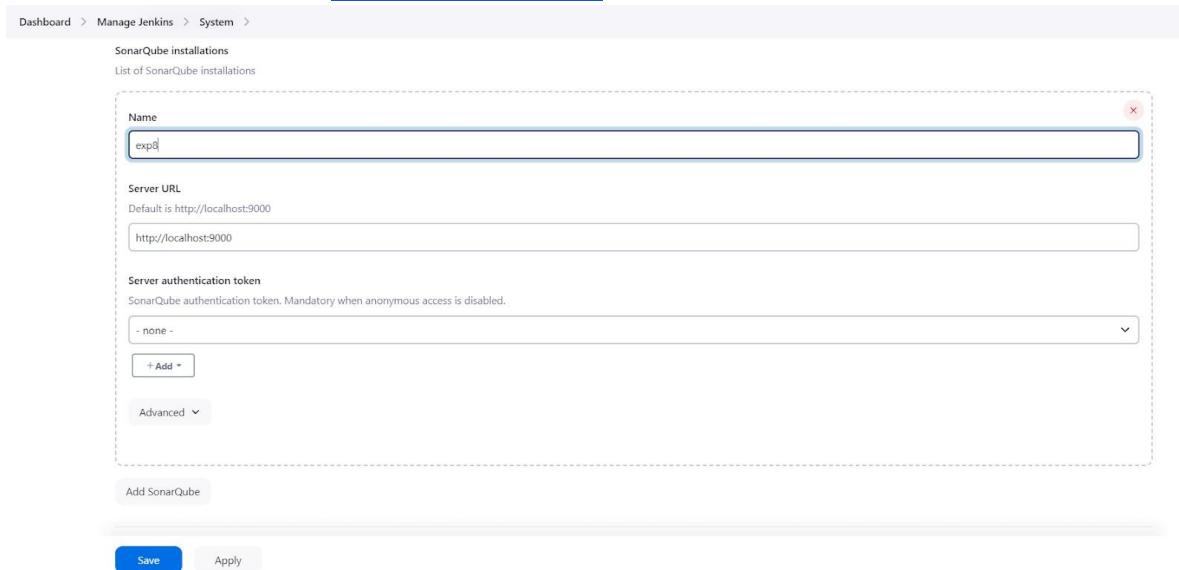
Search: sonarq

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
<input type="checkbox"/>	Sonar Gerrit 388.v9b_f1cb_e42306	External Site/Tool Integrations This plugin allows to submit issues from SonarQube to Gerrit as comments directly.
<input type="checkbox"/>	SonarQube Generic Coverage 1.0	TODO

7. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for SonarQube Servers and enter the details.
Enter the Server Authentication token if needed.

In SonarQube installations: Under Name add <project name of sonarqube> for me
adv_devops_7_sonarqube

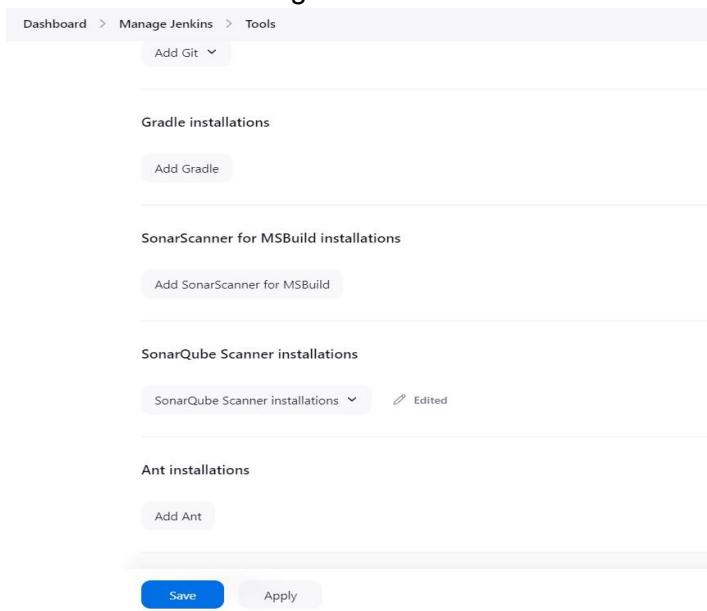
In Server URL Default is <http://localhost:9000>



The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'System' section. A new SonarQube installation is being configured. The 'Name' field contains 'exp0'. The 'Server URL' field is set to 'http://localhost:9000'. The 'Server authentication token' dropdown is currently set to 'none'. There is an 'Advanced' button for more options. At the bottom, there are 'Save' and 'Apply' buttons.

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

Dashboard > Manage Jenkins > Tools



The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. It lists several global tool configurations: 'Gradle installations' (with an 'Add Gradle' button), 'SonarScanner for MSBuild installations' (with an 'Add SonarScanner for MSBuild' button), 'SonarQube Scanner installations' (with a dropdown menu showing 'SonarQube Scanner installations' and 'Edited' status), and 'Ant installations' (with an 'Add Ant' button). At the bottom, there are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins 'Manage Jenkins' interface with the 'Tools' section selected. A 'SonarQube Scanner installations' card is open, showing a single configuration entry. The configuration details are as follows:

- Name:** sonarqube_exp8
- Install automatically:**
- Install from Maven Central:**
 - Version:** SonarQube Scanner 6.2.0.4584
 - Add Installer:** A dropdown menu.

At the bottom of the card are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins 'New Item' dialog. The 'Item name' field contains 'advdevops_exp8'. The 'Select an item type' section shows four options:

- Freestyle project**: Described as a classic general-purpose job type.
- Pipeline**: Described as orchestrating long-running activities across multiple build agents.
- Multi-configuration project**: Described as suitable for projects with many configurations.
- Folder**: Described as a container for nested items.

An 'OK' button is at the bottom of the dialog.

10. Under Pipeline script, enter the following:

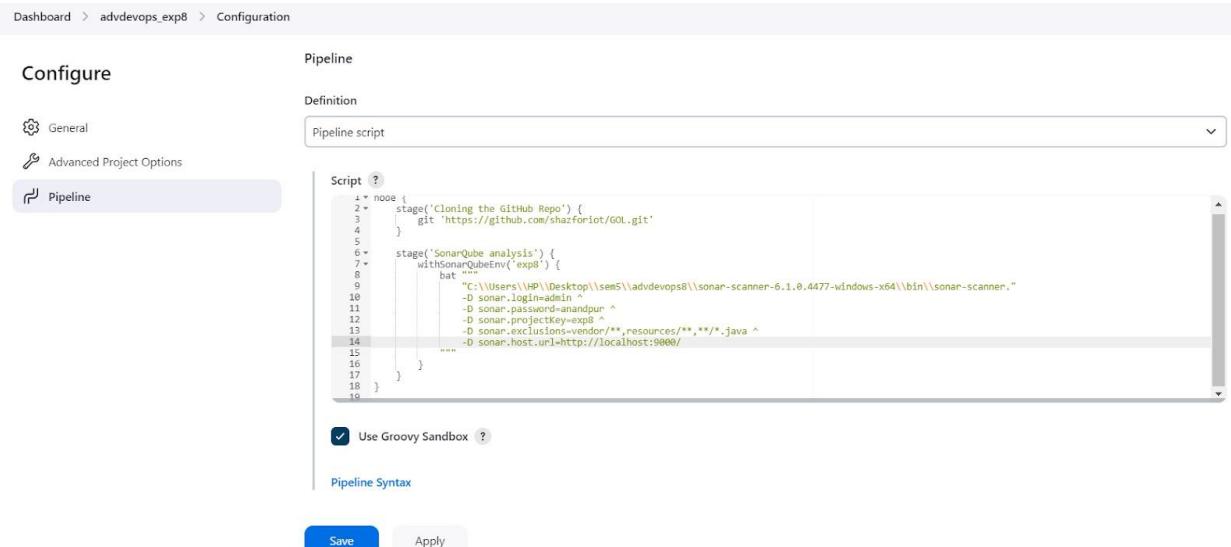
```

node {
stage('Cloning the GitHub Repo') { git
  'https://github.com/shazforiot/GOL.git'
}

stage('SonarQube analysis') {
  withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
    sh """
      <PATH_TO SONARQUBE SCANNER 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=<SonarQube_URL>(default: http://localhost:9000)
      """
    }
  }
}

```

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



The screenshot shows the Jenkins Pipeline configuration page for a pipeline named 'advdevops_exp8'. The 'Pipeline' tab is selected. The 'Definition' dropdown is set to 'Pipeline script'. The script content is as follows:

```

node {
  stage('Cloning the GitHub Repo') {
    git 'https://github.com/shazforiot/GOL.git'
  }

  stage('SonarQube analysis') {
    withSonarQubeEnv('exp8') {
      bat """
        <C:\Users\A\P\Desktop\sem5\advdevops8\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner. ^
        -D sonar.login=admin ^
        -D sonar.password=anandpur ^
        -D sonar.projectKey=exp8 ^
        -D sonar.exclusions=vendor/**,resources/**, **/*.java ^
        -D sonar.host.url=http://localhost:9000
      """
    }
  }
}

```

Below the script, there is a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom, there are 'Save' and 'Apply' buttons.

11. Build project

The screenshot shows the Jenkins Pipeline interface for 'advdevops_exp8'. On the left, a sidebar lists options like Status, Changes, Build Now, Configure, Delete Pipeline, SonarQube, Stages, Rename, and Pipeline Syntax. The main area displays the pipeline's history with the last 10 builds. A prominent green checkmark icon indicates the latest build (#10) was successful.

Build	Status	Time Ago
#10	Success	17 min ago
#9	Unsuccessful	19 min ago
#8	Completed	17 min ago
#7	Failed	27 min ago
#6	Successful	17 min ago
#5	Successful	17 min ago
#4	Successful	17 min ago
#3	Successful	17 min ago
#2	Successful	17 min ago
#1	Successful	17 min ago

12. Check console

The screenshot shows the Jenkins Pipeline interface for 'advdevops_exp8' focusing on the 'Console Output' tab for build #10. The output log is displayed, showing the Jenkins pipeline starting, cloning the GitHub repository, and performing a git pull operation.

```

Started by user Sahil Motiramani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\advdevops_exp8
[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_exp8\.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 2.43.0.windows.1
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/*
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a46123220d412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a46123220d412c5e6e1e5e4 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10

```

13. Now, check the project in SonarQube

The screenshot shows the SonarQube main dashboard for the 'main' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. The main content area displays the project's status as 'Passed' with a green checkmark icon. It provides an overview of code quality metrics: Security (0 Open issues), Reliability (68k Open issues), and Maintainability (164k Open issues). Below these are sections for Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6% on 759K lines). The dashboard also indicates 683k Lines of Code and a Version not provided.

14. Code Problems

- Consistency

The screenshot shows the SonarQube Issues tab for the 'main' project. The left sidebar features a 'Filters' section with 'My Issues' and 'All' buttons, and dropdown menus for 'Issues in new code', 'Clean Code Attribute' (selected 'Consistency'), and 'Software Quality'. The main pane displays a list of issues under the heading 'gameoflife-core/build/reports/tests/all-tests.html'. The first issue is 'Insert a <DOCTYPE> declaration to before this <html> tag.' (Severity: Reliability, Status: Open). The second is 'Remove this deprecated "width" attribute.' (Severity: Maintainability, Status: Open). The third is 'Remove this deprecated "align" attribute.' (Severity: Maintainability, Status: Open). The top right of the interface shows '196,662 issues' and '3075d effort'.

- Intentionality

Issues

The screenshot shows the SonarQube Issues page for the project 'exp8'. The main navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search bar. The 'Issues' tab is selected. On the left, there's a sidebar with 'My Issues' and 'All' buttons, and a 'Filters' section with a 'Clear All Filters' button. Below that are sections for 'Issues in new code' and 'Clean Code Attribute' (Consistency: 197k, Intentionality: 14k), 'Adaptability': 0, 'Responsibility': 0. A 'Software Quality' section is also present. The main content area displays a list of issues under 'gameoflife-acceptance-tests/Dockerfile'. Each issue entry includes a checkbox for 'Bulk Change', a title, a severity level (e.g., 'Intentionality'), and a detailed description. At the top right of the main content area, it says '13,887 issues' and '59d effort'.

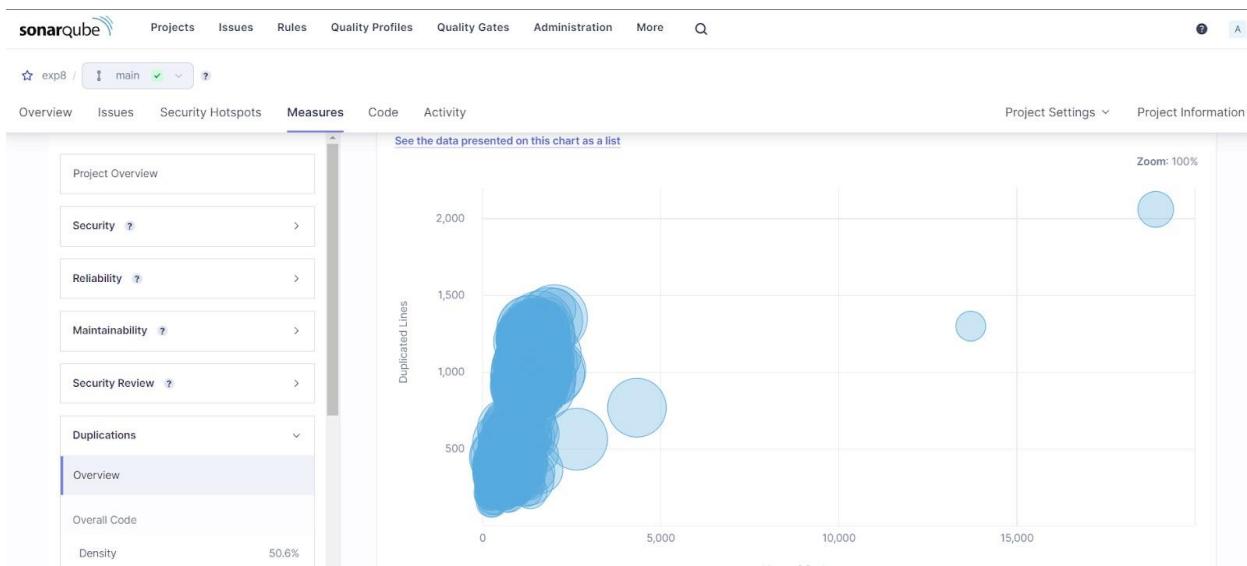
● Bugs

The screenshot shows the SonarQube Bugs page for the project 'exp8'. The main navigation bar is identical to the Issues page. The 'Bugs' tab is selected. The left sidebar shows 'Software Quality' (Security: 0, Reliability: 14k, Maintainability: 0), 'Severity' (0), 'Type' (Bug: 14k, Vulnerability: 0, Code Smell: 268), and 'Scope'. The main content area shows a list of bugs under 'gameoflife-core/build/reports/tests/all-tests.html'. Each bug entry has a checkbox for 'Bulk Change', a title, a severity level (e.g., 'Intentionality'), and a detailed description. At the top right, it says '13,619 issues' and '56d effort'.

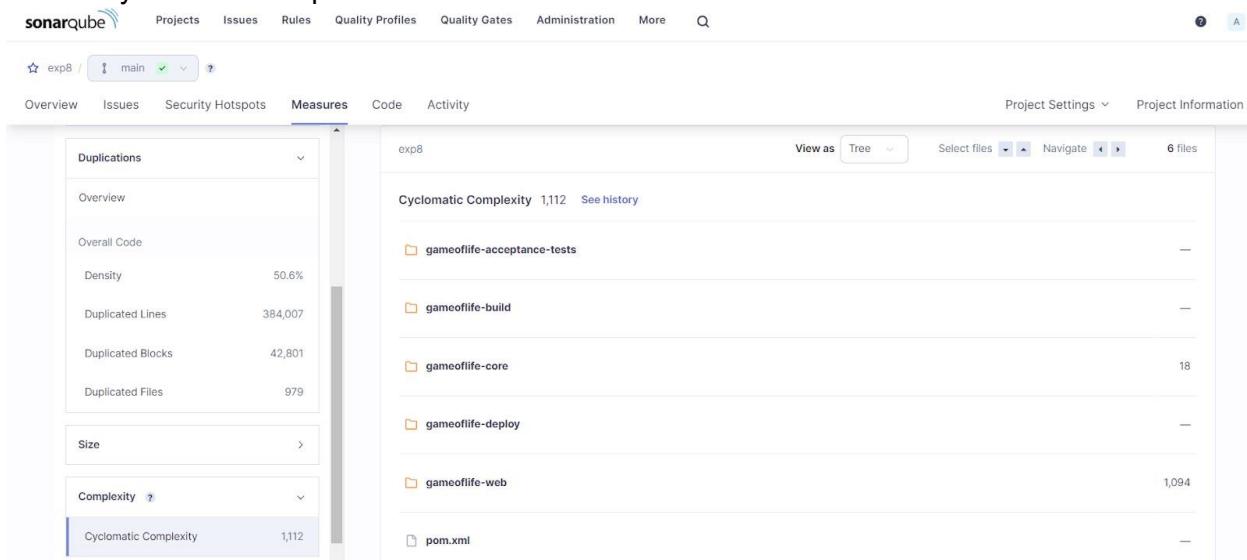
● Code Smells

The screenshot shows the SonarQube Code Smells page for the project 'exp8'. The main navigation bar is identical. The 'Code Smells' tab is selected. The left sidebar shows 'Software Quality' (Security: 0, Reliability: 253, Maintainability: 15), 'Severity' (0), 'Type' (Bug: 14k, Vulnerability: 0, Code Smell: 268), and 'Scope'. The main content area shows a list of code smells under 'gameoflife-acceptance-tests/Dockerfile'. Each smell entry has a checkbox for 'Bulk Change', a title, a severity level (e.g., 'Intentionality'), and a detailed description. At the top right, it says '268 issues' and '2d 5h effort'.

● Duplications



● Cyclomatic Complexities



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

Conclusion:

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

Exp:09

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

Theory :**What is Nagios?**

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

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

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes ● Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures ● You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

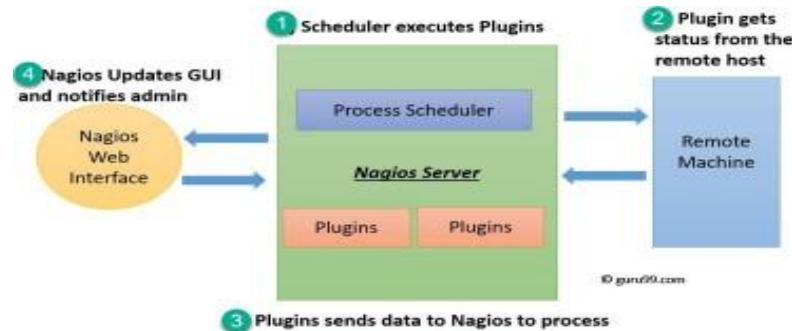
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes

- If the services are running fine, then there is no need to do check that host is alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, SSH, SNMP, FTP, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.

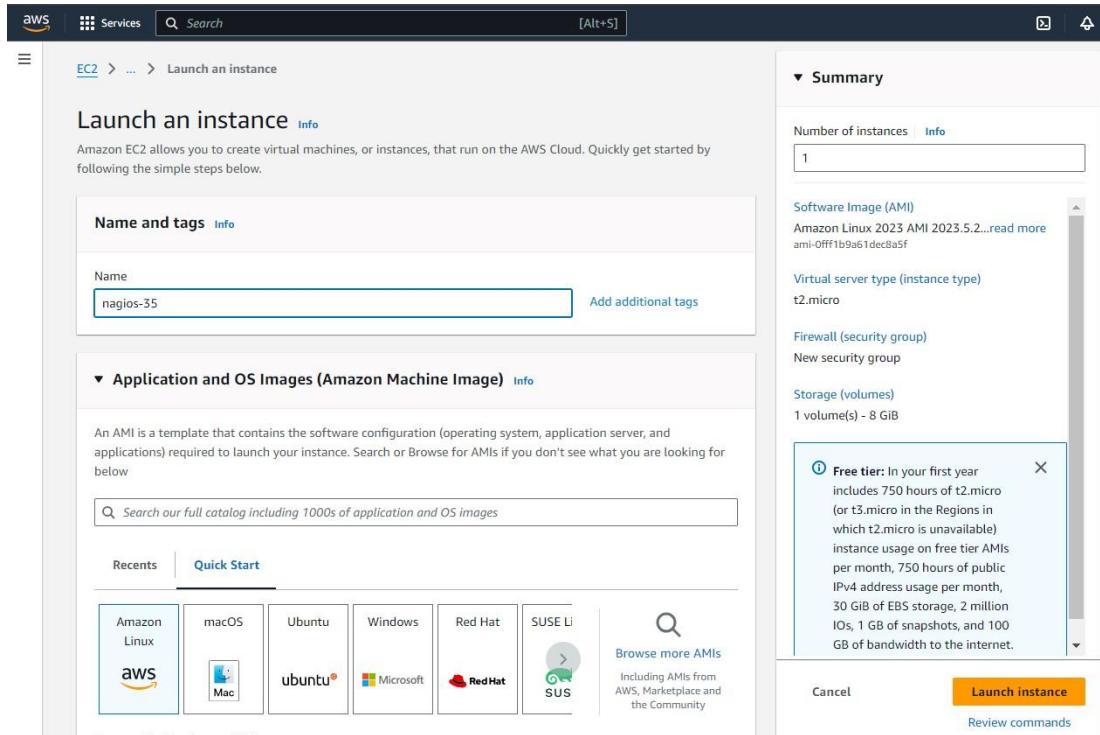


1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

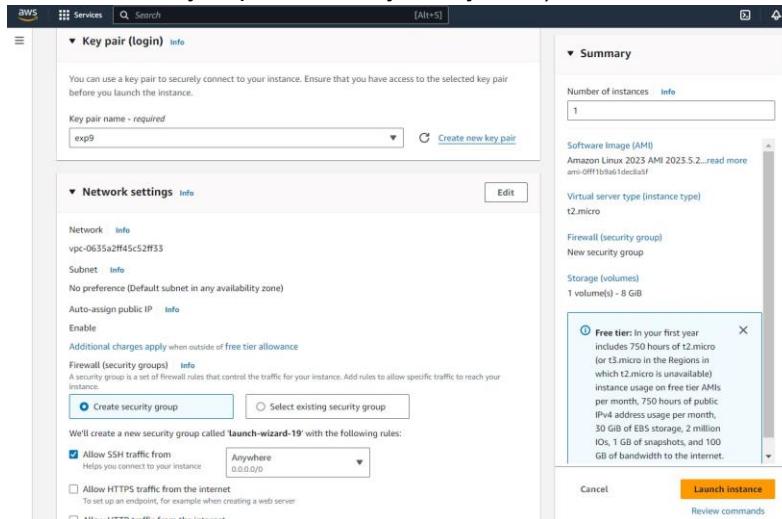
Prerequisites: AWS Free Tier

Step 1: Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.



Select The OS Image as Amazon Linux.

Step 2: If you do not have a private key created or a .pem file created, click on create a key pair. Else select the key pair that you had created before. (Make sure you know where the .pem file for that key is present on your system)



AWS will create a security group for this instance. Keep the name of that instance saved.

Name:Sahil Motiramani

Div:D15C

Rollno:35

Step 3: After creating the instance, click on Security Groups from the left side pane. Find the security group that was created for your instance. Click on the instance ID for that group.

The screenshot shows the AWS EC2 Security Groups page. The left sidebar lists various EC2-related options like Dashboard, Global View, Instances, and Instance Types. The main area displays a table titled 'Security Groups (22) Info' with columns for Name, Security group ID, Security group name, VPC ID, Description, and Owner. The table lists several security groups, including 'launch-wizard-16', 'default', and multiple entries for 'launch-wizard'. Each row includes a 'View details' link and a 'Delete' button.

Here, click on Edit Inbound Rules.

The screenshot shows the 'Details' tab for the security group 'sg-0155e5f9b619e6e06 - launch-wizard-16'. It displays basic information: Security group name (launch-wizard-16), Security group ID (sg-0155e5f9b619e6e06), Description (launch-wizard-16 created 2024-10-05T04:42:45.023Z), Owner (767378259677), Inbound rules count (1 Permission entry), and Outbound rules count (1 Permission entry). Below this, the 'Inbound rules' tab is selected, showing a table with one rule: 'sgr-0d9a65ebb737dd8ed' (Protocol: TCP, Port range: 22, Source: 0.0.0.0/0, Type: SSH).

Now, click on add rules, and add teh rules for the following protocols:

HTTP, All ICMP - IPv6, HTTPS, All traffic, Custom TCP (Port 5666), All ICMP - IPv4

The screenshot shows the 'Edit inbound rules' page for the same security group. It lists eight new rules that have been added:

Name	Type	Protocol	Port range	Source	Description
-	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTP	TCP	80	Anywhere-...	/0
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere-...	/0
-	HTTPS	TCP	443	Anywhere-...	0.0.0.0/0
-	All traffic	All	All	Anywhere-...	0.0.0.0/0
-	Custom TCP	TCP	5666	Anywhere-...	0.0.0.0/0
-	All ICMP - IPv4	ICMP	All	Anywhere-...	0.0.0.0/0

Click on save. This will add all the inbound rules to the security group.

Name:Sahil Motiramani

Div:D15C

Rollno:35

Security group name: launch-wizard-16
Owner: 767378259677
Security group ID: sg-0155e5f9b619e6e06
Description: launch-wizard-16 created 2024-10-05T04:42:45Z
VPC ID: vpc-0635a2ff45c52ff33

Inbound rules	Outbound rules	Tags																																																																
Inbound rules (7) Search: <input type="text"/> <table border="1"><thead><tr><th>Name</th><th>Security group rule...</th><th>IP version</th><th>Type</th><th>Protocol</th><th>Port range</th><th>Source</th><th>Description</th></tr></thead><tbody><tr><td>-</td><td>sgr-dau2d85959sec7616...</td><td>IPv4</td><td>All traffic</td><td>All</td><td>All</td><td>0.0.0.0/0</td><td>-</td></tr><tr><td>-</td><td>sgr-01c78e117ddc91f7d</td><td>IPv6</td><td>HTTP</td><td>TCP</td><td>80</td><td>::/0</td><td>-</td></tr><tr><td>-</td><td>sgr-dab571c02cfb9504</td><td>IPv4</td><td>All ICMP - IPv4</td><td>ICMP</td><td>All</td><td>0.0.0.0/0</td><td>-</td></tr><tr><td>-</td><td>sgr-0951f5f24dc41b5a5</td><td>IPv4</td><td>Custom TCP</td><td>TCP</td><td>5666</td><td>0.0.0.0/0</td><td>-</td></tr><tr><td>-</td><td>sgr-07663f5c31d0d975c</td><td>IPv4</td><td>HTTPS</td><td>TCP</td><td>443</td><td>0.0.0.0/0</td><td>-</td></tr><tr><td>-</td><td>sgr-065ffbf768228beb</td><td>IPv6</td><td>All ICMP - IPv6</td><td>IPv6 ICMP</td><td>All</td><td>::/0</td><td>-</td></tr><tr><td>-</td><td>sgr-0d9a65eb737dd...</td><td>IPv4</td><td>SSH</td><td>TCP</td><td>22</td><td>0.0.0.0/0</td><td>-</td></tr></tbody></table>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	-	sgr-dau2d85959sec7616...	IPv4	All traffic	All	All	0.0.0.0/0	-	-	sgr-01c78e117ddc91f7d	IPv6	HTTP	TCP	80	::/0	-	-	sgr-dab571c02cfb9504	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-	-	sgr-0951f5f24dc41b5a5	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-	-	sgr-07663f5c31d0d975c	IPv4	HTTPS	TCP	443	0.0.0.0/0	-	-	sgr-065ffbf768228beb	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-	-	sgr-0d9a65eb737dd...	IPv4	SSH	TCP	22	0.0.0.0/0	-		
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description																																																											
-	sgr-dau2d85959sec7616...	IPv4	All traffic	All	All	0.0.0.0/0	-																																																											
-	sgr-01c78e117ddc91f7d	IPv6	HTTP	TCP	80	::/0	-																																																											
-	sgr-dab571c02cfb9504	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-																																																											
-	sgr-0951f5f24dc41b5a5	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-																																																											
-	sgr-07663f5c31d0d975c	IPv4	HTTPS	TCP	443	0.0.0.0/0	-																																																											
-	sgr-065ffbf768228beb	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-																																																											
-	sgr-0d9a65eb737dd...	IPv4	SSH	TCP	22	0.0.0.0/0	-																																																											

Step 4: Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect.

Instances (1/1) Info
Last updated less than a minute ago
Find Instance by attribute or tag (case-sensitive)
All states
Name: nagios-35
Instance ID: i-002e443a2612e41be
Instance state: Running
Instance type: t2.micro
Status check: 2/2 checks passed
Alarm status: View alarms
Availability Zone: us-east-1b
Public IPv4 DNS: ec2-54-207-239-4.compute-1.amazonaws.com
Public IP: 54.207.2

Click on SSH client. Copy the example command.

EC2 > Instances > i-002e443a2612e41be > Connect to instance

Connect to instance info

Connect to your instance i-002e443a2612e41be (nagios-35) using any of these options

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

Instance ID: i-002e443a2612e41be (nagios-35)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp9.pem.
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "exp9.pem"
4. Connect to your instance using its Public DNS:
 ssh -i "exp9.pem" ec2-user@ec2-54-207-239-4.compute-1.amazonaws.com

Command copied

ssh -i "exp9.pem" ec2-user@ec2-54-207-239-4.compute-1.amazonaws.com

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

Step 5: Now, we have to connect our local OS terminal to the instance using SSH. For this, Open terminal wher the private key file is located (.pem) Paste the copied SSH command and run it.

```
Microsoft Windows [Version 10.0.22631.4249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\Desktop\sem5\advdevops 9>ssh -i "exp9.pem" ec2-user@ec2-34-207-239-4.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-239-4.compute-1.amazonaws.com (34.207.239.4)' can't be established.
ED25519 key fingerprint is SHA256:UuegxDkRL6R+3iPxqY2jG36UhsDv85Y3Wp21RVCzobo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-239-4.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

[ec2-user@ip-172-31-39-94 ~]$ |
```

Step 6: Now we start working on this terminal. First run the command
sudo yum update

This command will check for any updates for the YUM library.

```
[ec2-user@ip-172-31-39-94 ~]$ sudo yum update
Last metadata expiration check: 0:19:35 ago on Sun Oct  6 10:11:16 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-39-94 ~]$ |
```

Step 7: We are going to install an Apache server and a PHP on this instance. For that, run this command.

sudo yum install httpd php

```
[ec2-user@ip-172-31-39-94 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:20:07 ago on Sun Oct  6 10:11:16 2024.
Dependencies resolved.
=====
Package           Architecture Version   Repository  Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023            48 k
php8.3           x86_64      8.3.10-1.amzn2023.0.1        10 k
Installing dependencies:
apr               x86_64      1.7.2-2.amzn2023.0.2        129 k
apr-util          x86_64      1.6.3-1.amzn2023.0.1        98 k
generic-logos-httpd noarch     18.0.0-12.amzn2023.0.3        19 k
httpd-core        x86_64      2.4.62-1.amzn2023            1.4 M
httpd-filesystem noarch     2.4.62-1.amzn2023            14 k
httpd-tools        x86_64      2.4.62-1.amzn2023            81 k
libbrotli          x86_64      1.0.9-4.amzn2023.0.2        315 k
libsodium          x86_64      1.0.19-4.amzn2023            176 k
libxml2            x86_64      1.1.34-5.amzn2023.0.2        241 k
mailcap            noarch     2.1.49-3.amzn2023.0.3        33 k
nginx-filesystem  noarch     1:1.24.0-1.amzn2023.0.4        9.8 k
php8.3-cli         x86_64      8.3.10-1.amzn2023.0.1        3.7 M
php8.3-common      x86_64      8.3.10-1.amzn2023.0.1        737 k
php8.3-process     x86_64      8.3.10-1.amzn2023.0.1        45 k
php8.3-xml          x86_64      8.3.10-1.amzn2023.0.1        154 k
Installing weak dependencies:
apr-util-openssl  x86_64      1.6.3-1.amzn2023.0.1        17 k
mod_http2          x86_64      2.0.27-1.amzn2023.0.3        166 k
mod_lua             x86_64      2.4.62-1.amzn2023            61 k
php8.3-fpm          x86_64      8.3.10-1.amzn2023.0.1        1.9 M
php8.3-mbstring     x86_64      8.3.10-1.amzn2023.0.1        528 k
php8.3-opcache      x86_64      8.3.10-1.amzn2023.0.1        379 k
php8.3-pdo           x86_64      8.3.10-1.amzn2023.0.1        89 k
php8.3-sodium        x86_64      8.3.10-1.amzn2023.0.1        41 k
```

Name:Sahil Motiramani

Div:D15C

Rollno:35

```
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
  libodium-1.19.4-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-pdo-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

apr-util-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.62-1.amzn2023.x86_64
httpd-tools-2.4.62-1.amzn2023.x86_64
liblxml-1.1.34-5.amzn2023.0.2.x86_64
mod_lua-2.4.62-1.amzn2023.x86_64
php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
php8.3-process-8.3.10-1.amzn2023.0.1.x86_64

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
```

Step 8: Next we install C/C++ compiler (GCC) along with the necessary C libraries required for compiling and running C programs. Use the following command. `sudo yum install gcc glibc glibc-common`

```
[ec2-user@ip-172-31-39-94 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:21:56 ago on Sun Oct 6 10:11:16 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.
```

Package	Architecture	Version	Repository	Size
Installing:				
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	32 M
Installing dependencies:				
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux	92 k
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux	887 k
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	10 M
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux	185 k
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux	27 k
glIBC-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 k
guile2	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.189-118.189.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
lbttool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 k

```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch
  gc-8.0.4-5.amzn2023.0.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
  liblpc-1.2.1-2.amzn2023.0.2.x86_64
  make-1.4.3-5.amzn2023.0.2.x86_64

Available Updates:
  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

Complete!
[ec2-user@ip-172-31-39-94 ~]$ |
```

Step 9: We would also need GD library and its development tools. For that, run this command
`sudo yum install gd gd-devel`

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
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	604 k
cmake-filesystem	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-filesystem	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
Installed:				
brotli-1.0.9-4.amzn2023.0.2.x86_64	brotli-devel-1.0.9-4.amzn2023.0.2.x86_64	brotli-devel-1.0.9-4.amzn2023.0.2.x86_64	bzip2-devel-1.0.8-4.amzn2023.0.2.x86_64	
cairo-1.17.6-2.amzn2023.0.1.x86_64	cmake-filesystem-2.2-2.1.amzn2023.0.4.x86_64	fontconfig-2.13.94-2.amzn2023.0.2.x86_64	fontconfig-2.13.94-2.amzn2023.0.2.x86_64	
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64	fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch	freetype-2.13.2-5.amzn2023.0.1.x86_64	freetype-2.13.2-5.amzn2023.0.1.x86_64	
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64	gd-2.3.3-5.amzn2023.0.3.x86_64	google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch	google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch	
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64	google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch	graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64	graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64	
graphite2-1.3.14-7.amzn2023.0.2.x86_64	harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64	harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64	harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64	
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64	libcICE-1.0.10-6.amzn2023.0.2.x86_64	jbigkit-libs-2.1-21.amzn2023.0.2.x86_64	jbigkit-libs-2.1-21.amzn2023.0.2.x86_64	
langpacks-core-font-ent-3.0-21.amzn2023.0.4.noarch	libcXX-1.0.9-6.amzn2023.0.2.x86_64	libBZM-1.2-3-8.amzn2023.0.2.x86_64	libBZM-1.2-3-8.amzn2023.0.2.x86_64	
libXi-1.7.3-1.amzn2023.0.4.x86_64	libcXX-common-1.0.9-6.amzn2023.0.2.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libXau-1.0.9-6.amzn2023.0.2.x86_64	libXau-1.0.9-6.amzn2023.0.2.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libXext-1.3.4-6.amzn2023.0.2.x86_64	libXpm-3.5.15-2.amzn2023.0.3.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libXrender-1.9.10-14.amzn2023.0.2.x86_64	libXpm-3.5.15-2.amzn2023.0.3.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libXf86-1.9.10-14.amzn2023.0.2.x86_64	libXt-1.2.0-4.amzn2023.0.2.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libfb1-devel-3.4.4-1.amzn2023.0.1.x86_64	libXt-1.2.0-4.amzn2023.0.2.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libjpegturbo-2.1.4-2.amzn2023.0.5.x86_64	libICU-67.1-7.amzn2023.0.3.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libpng-2.1.6.37-10.amzn2023.0.6.x86_64	libjpegturbo-devel-2.1.4-2.amzn2023.0.5.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libsepul-3.4-3.amzn2023.0.3.x86_64	libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libwebrtc-1.2-4-1.amzn2023.0.6.x86_64	libtiff-4.0-4.amzn2023.0.18.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
libxcb-1.13.1-17.amzn2023.0.2.x86_64	libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
pcr2-utf16-10-40-1.amzn2023.0.3.x86_64	libxcb-devel-2.10-4.amzn2023.0.6.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
sysprof-capture-devel-3.40.1-1.amzn2023.0.2.x86_64	libxcb-fb-1.0-1.amzn2023.0.6.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	
zx-devel-5.2.5-9.amzn2023.0.2.x86_64	libxcb-glx-1.0-1.amzn2023.0.6.x86_64	libI18N-1.7.2-3.amzn2023.0.4.noarch	libI18N-1.7.2-3.amzn2023.0.4.noarch	

Step 10: Now, we create a user called as 'nagios' and make sure that it has a home directory, and set up a password for it. sudo adduser -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-39-94 ~]$ 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-172-31-39-94 ~]$ |
```

Step 11: Create a user group called as 'nagcmd' to execute nagios commands.
sudo groupadd nagcmd

```
[ec2-user@ip-172-31-39-94 ~]$ sudo groupadd nagcmd  
[ec2-user@ip-172-31-39-94 ~]$ |
```

Step 12: Add users apache and nagios to this user group.

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

```
[ec2-user@ip-172-31-39-94 ~]$ sudo usermod -a -G nagcmd nagios  
sudo usermod -a -G nagcmd apache  
[ec2-user@ip-172-31-39-94 ~]$ |
```

Step 13: We create a directory downloads, to store the files of nagios server that are downloaded. mkdir ~/downloads cd ~/downloads

```
[ec2-user@ip-172-31-39-94 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-39-94 downloads]$ |
```

Step 14:Now we need to install the latest versions of nogios-core and nagios-plugins. Go to the respective websites and check whether a better version is available.If newer versions are available, then right click on the download button → Copy link address.

Paste this link address in place of the current link in command.
If not run these commands.

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

Name:Sahil Motiramani

Div:D15C

Rollno:35

```
[ec2-user@ip-172-31-83-157 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-28 04:04:23-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[=====] 1.97M  5.36MB/s   in 0.4s

2024-09-28 04:04:24 (5.36 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-83-157 downloads]$
```

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

```
[ec2-user@ip-172-31-39-94 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-06 10:37:57-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[=====] 1.97M  6.17MB/s   in 0.3s

2024-10-06 10:37:58 (6.17 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-39-94 downloads]$ |
```

Step 15:Now, we need to extract nagios-core file into the same directory. For that, we will use tar command.

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-39-94 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/xdata/.gitignore
nagios-4.5.5/xdata/Makefile.in
nagios-4.5.5/xdata/xcdefault.c
nagios-4.5.5/xdata/xcdefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-39-94 downloads]$ |
```

Step 16: We need to ensure that Nagios uses a specific group (in this case, `nagcmd`) for executing external commands.

Name:Sahil Motiramani

Div:D15C

Rollno:35

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

An **error** could be encountered here: **./configure: no such path or directory**

Solution: Navigate to the nagios-4.5.5 folder in downloads. (version could vary)

Steps: ls

```
[ec2-user@ip-172-31-39-94 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz
[ec2-user@ip-172-31-39-94 downloads]$ |
```

- cd nagios-4.5.5 (use the version shown by your ls command)
- ./configure --with-command-group=nagcmd

Another **error** could be **Cannot find SSL headers**.

To solve this, we need to install OpenSSL Dev Library

Steps:

sudo yum install openssl-devel

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:30:53 ago on Sun Oct  6 10:11:16 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
openssl-devel    x86_64          1:3.0.0-1.amzn2023.0.14
                                                               amazonlinux  3.0 M
Transaction Summary
=====
Install 1 Package

Total download size: 3.0 M
Is this ok [Y/N]: y
Downloading Packages:
openssl-devel-3.0.0-1.amzn2023.0.14.x86_64.rpm          26 MB/s | 3.0 MB  00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Pretrans:
Installing : openssl-devel-1:3.0.0-1.amzn2023.0.14.x86_64          1/1
Running scriptlet: openssl-devel-1:3.0.0-1.amzn2023.0.14.x86_64
Verifying   : openssl-devel-1:3.0.0-1.amzn2023.0.14.x86_64          1/1

Installed:
openssl-devel-1:3.0.0-1.amzn2023.0.14.x86_64

Complete!
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$
```

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

Name:Sahil Motiramani

Div:D15C

Rollno:35

```
[ec2-user@ip-172-31-39-94 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
```

```
*** 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.
```

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ |
```

Step 17: We need to compile all components of this software according to the instruction in the Makefile. For that, use this command:

make all

Then, sudo make install sudo
make install-init sudo make
install-config sudo make install-
commandmode

```
[ec2-user@ip-172-31-39-94 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: cannot stat 'nagios': No such file or directory
make[1]: *** [Makefile:188: install] Error 1
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
```

*** Config files installed ***

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

```
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
```

*** External command directory configured ***

Step 18: We need to update the email linked with this server to our email for it to send notifications (if any needed). sudo nano

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

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####
#
# CONTACTS
#
#####
#
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.
#
define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defi
    alias             Nagios Admin         ; Full name of user
    email             2022.sahil.motiramani@ves.ac.in| ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}
#
#
# CONTACT GROUPS
#
#####
```

Here, change the email under 'define contact{}' to your email address.

To save this use the following shortcut sequence CTRL+O→Enter→CTRL+X.

Name:Sahil Motiramani

Div:D15C

Rollno:35

CTRL+O: Overwrite the existing file with edited file

CTRL+X: Exit nano editor.

Step 19: We need to install the necessary configuration files for the Nagios web interface.

Run the following command. sudo make install-webconf

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ |
```

Step 20: Now we need to setup a user to access this nagios web interface. So we run this command to create a user called 'nagiosadmin'.

Keep this username and password saved as it is needed to login to the web interface.

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

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ |
```

Step 21: Restart the apache server to apply all the recent configurations.

sudo service httpd restart

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ |
```

Step 22: Now we go back to the downloads folder and extract the files of nagios plugin. cd ~/downloads tar zxvf nagios-plugins-2.4.11.tar.gz (Version may vary)

Name:Sahil Motiramani

Div:D15C

Rollno:35

```
[ec2-user@ip-172-31-39-94 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/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-39-94 downloads]$ |
```

Step 23: Again, we need to install the configurations for these files.

cd nagios-plugins-2.4.11 (version may vary)

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

```
[ec2-user@ip-172-31-39-94 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-94 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
```

```
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-39-94 nagios-plugins-2.4.11]$ |
```

Step 24: We need to compile all components of this software according to the instruction in the Makefile. For that, use the commands:

make sudo

make install

```
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-39-94 nagios-plugins-2.4.11]$ |
```

Step 25: We need to register the Nagios service with the system, which would make it able to manage the server status. So run the following commands sudo chkconfig --add nagios
sudo chkconfig nagios on

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl enable nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl start nagios
```

Step 26: We need to verify the Nagios configuration for any syntax errors or issues before starting or restarting the Nagios service.

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

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -a 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl enable nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl start nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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 0 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 0 hosts.
    Checked 0 service dependencies.
    Checked 0 host dependencies.
    Checked 5 timperiods.
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-172-31-39-94 nagios-4.4.6]$ |
```

Step 27: sudo service nagios start

nagios start

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ |
```

sudo systemctl status nagios

Name:Sahil Motiramani

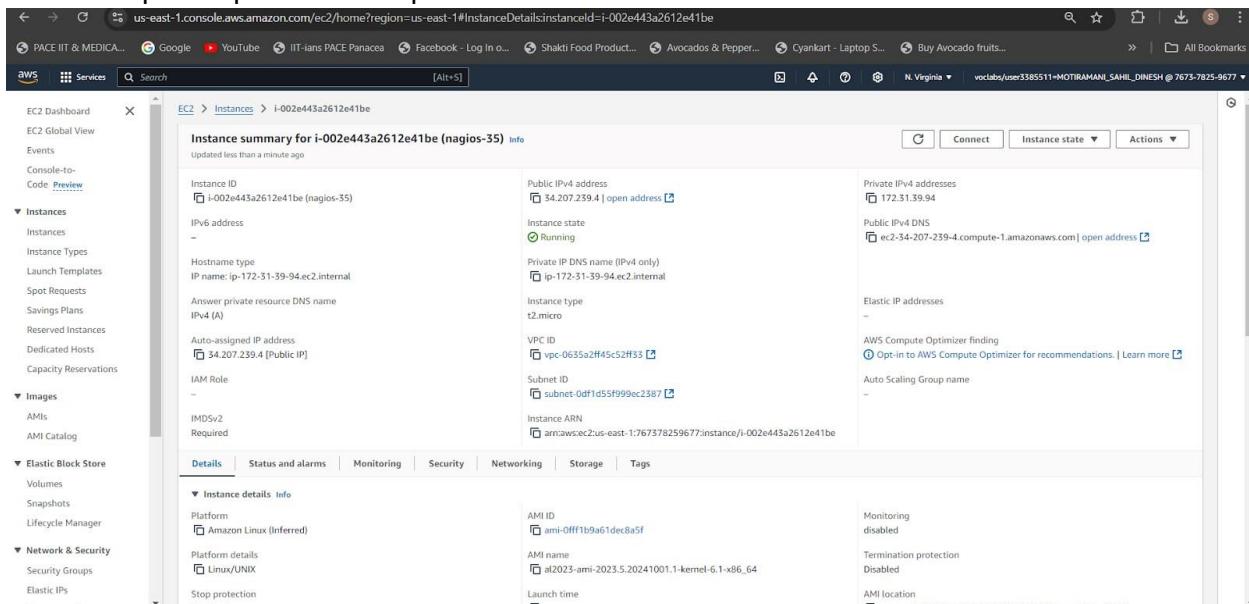
Div:D15C

Rollno:35

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-10-06 11:02:08 UTC; 2min 4s ago
       Docs: https://www.nagios.org/documentation
   Process: 66287 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 66288 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 66289 (nagios)
   Tasks: 6 (limit: 1112)
     Memory: 2.4M
        CPU: 42ms
      CGroup: /system.slice/nagios.service
              └─66289 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─68289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─68291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─68292 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─68293 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─68294 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: qh core query handler registered
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: qh echo service query handler registered
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: qh help for the query handler registered
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: wproc: Successfully registered manager as @wproc with query handler
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: wproc: Registry request: name=Core Worker 68291;pid=68291
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: wproc: Registry request: name=Core Worker 68292;pid=68292
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: wproc: Registry request: name=Core Worker 68293;pid=68293
Oct 06 11:02:08 ip-172-31-39-94.ec2.internal nagios[66289]: wproc: Registry request: name=Core Worker 68299;pid=68299
Oct 06 11:02:10 ip-172-31-39-94.ec2.internal nagios[66289]: Successfully launched command file worker with pid 68294
Oct 06 11:04:00 ip-172-31-39-94.ec2.internal nagios[66289]: SERVICE ALERT: localhost:HTTP;WARNING;SOFT;1;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.003 second response time
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ ]
```

Step 28: Now, go to EC2 instance and click on instance id. Then, click on the copy icon just before the public ip address on public IP.



Step 29: Open a new tab. In the address bar type <http://<publicipaddress>/nagios>. This would be in the output



Conclusion:

In this experiment, we have learned how to install and set up Nagios Core, Nagios Plugins, and NRPE on a Linux machine. We are working with an Amazon Linux OS instance that has been configured with the necessary security rules. It's crucial to ensure that the links for Nagios Core and Nagios Plugins used in the wget commands are current and up-to-date. After downloading, we need to extract and configure these files to prevent any issues when starting the server. Once the setup is complete, we can launch the Nagios server. By using the public IP address of the EC2 instance, we can access the Nagios dashboard by entering that IP into a web browser.

Exp:10

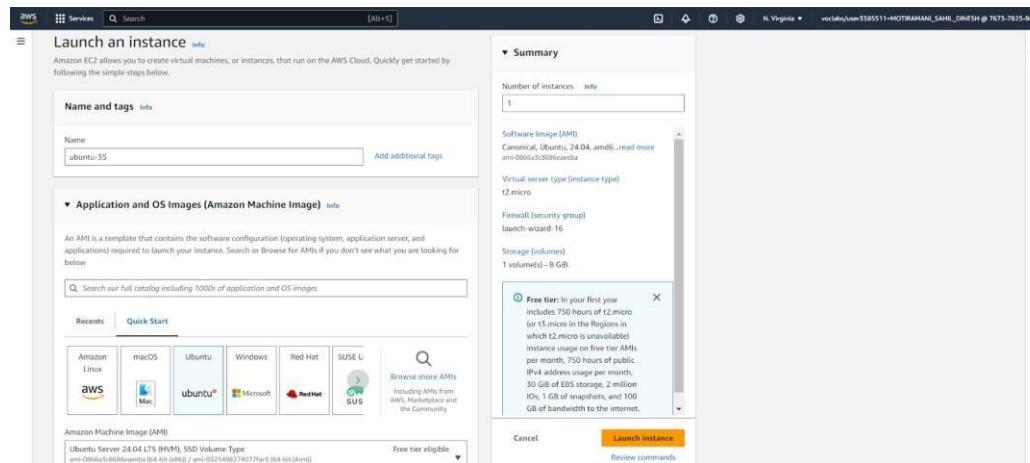
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Prerequisites:

- 1) An Amazon Linux instance with nagios already set up.

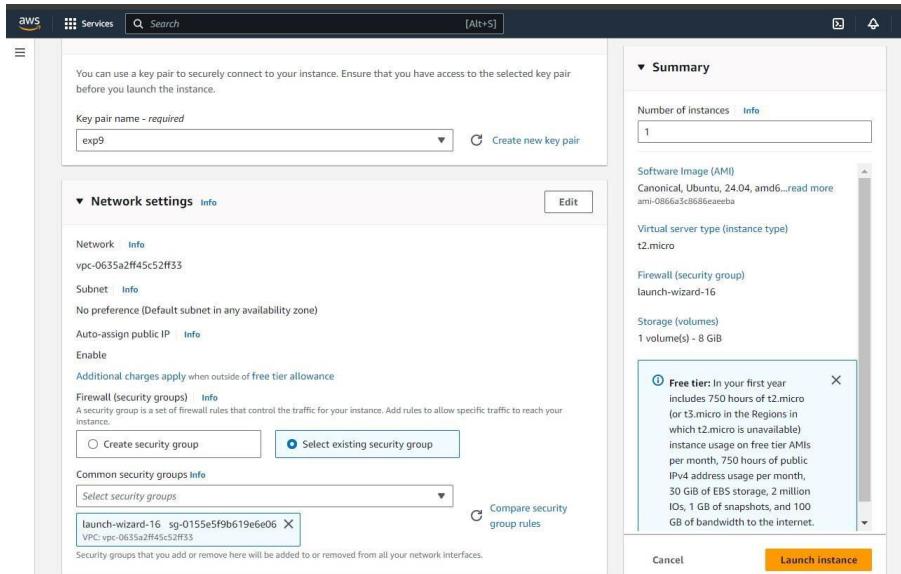
Step 1: Set up ubuntu instance

- 1) Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.



Select The OS Image as Ubuntu.

- 2) Make sure to select the same private key that you created for the Amazon Linux instance. Also select the same security group as you created for the Linux instance.



- 3) Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect. Click on SSH client. Copy the example command. Now, we have to connect our local OS terminal to the instance using SSH. For this, open terminal where the private key file is located (.pem). Paste the copied SSH command and run it.

Step 2: Execute the following on Nagios Host machine (Linux)

- 1) We need to verify whether the nagios service is running or not. For that, run this command.

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

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ ps -ef | grep nagios
nagios  68289      1  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  68290  68289  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  68291  68289  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  68292  68289  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  68293  68289  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  68294  68289  0 11:02 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
ec2-user  71786    2942  0 11:51 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ |
```

- 2) Now, make yourself as the root user, and create a folder with the path '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts' sudo su mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-39-94 nagios-4.4.6]# |
```

- 3) We need to create a config file in this folder. So, copy the contents of the existing localhost config to the new file 'linuxserver.cfg'. cp /usr/local/nagios/etc/objects/localhost.cfg

`/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg` 4) We

need to make some changes in this config file. Open it using nano editor.

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

Change hostname and alias to linuxserver

Change address to public ip address of client instance (Ubuntu instance)

```
# 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     localhost
    alias         localhost
    address       3.80.168.49
}
```

Change hostgroup_name to linux-servers1

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

Change the **occurrences of hostname** further in the document from **localhost** to **linuxserver**

- 5) Now, we need to edit the nagios configuration file to add this directory. `nano /usr/local/nagios/etc/nagios.cfg` Run this command and add the following line

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

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

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

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

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

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

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

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

- 6) Now we verify the configuration files.

Name:Sahil Motiramani

Div:D15C

RollNo:35

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

```
[root@ip-172-31-39-94 nagios-4.4.6]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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 timperiods

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-172-31-39-94 nagios-4.4.6]
```

- 7) Once the files are verified, we need to restart the server.

service nagios restart

```
[root@ip-172-31-39-94 nagios-4.4.6]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-39-94 nagios-4.4.6]# |
```

Step 3: Execute the following on Nagios Client machine (Ubuntu)

- 1) First, we check for any new updates, then we install gcc, nagios nrpe server and nagios plugins.

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

Name:Sahil Motiramani

Div:D15C

RollNo:35

```
ubuntu@ip-172-31-44-65:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-44-65:~$ |
```

- 2) We need to add the public IP address of our host Nagios machine (Linux) to the nrpe configuration file.

sudo nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add the nagios host ip address (public)

```
# NRPE USER
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

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

#
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,34.207.239.4
```

Step 4: Check the Nagios Dashboard 1)

Go to Nagios dashboard, click on hosts.

Here, we can see that the linuxserver is also added as a host.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-07-2024 04:54:21	0d 0h 14m 37s	PING OK - Packet loss = 0%, RTA = 1.24 ms
localhost	UP	10-07-2024 04:50:52	0d 17h 52m 26s	PING OK - Packet loss = 0%, RTA = 0.03 ms

2) Click on linuxserver. Here, we can check all the information about linuxserver host.

The screenshot shows the Nagios web interface at <http://18.215.236.11/nagios/>. The left sidebar has sections for General, Current Status, Reports, and System. The main content area displays 'Host Information' for the host 'localhost' (linuxserver). It shows the host is up with 0% packet loss and RTA of 1.24 ms. The 'Host State Information' section shows the host is UP (0d 0h 16m 0s). The 'Host Commands' section contains various monitoring and control options. Below these are 'Host Comments' and a table for adding comments.

3) Click on services. Here we can see all the services that are being monitored by linuxserver.

The screenshot shows the Nagios web interface at <http://18.215.236.11/nagios/>. The left sidebar has sections for General, Current Status, Reports, and System. The main content area displays 'Service Status Details For All Hosts'. It lists services for 'linuxserver' and 'localhost'. For 'linuxserver', services like Current Load, Current Users, and SSH are OK. However, the HTTP service is CRITICAL due to connection refused. For 'localhost', the SSH service is also CRITICAL. A status summary table at the top right shows 12 OK, 1 WARNING, 0 UNKNOWN, and 3 CRITICAL services across all types.

Name:Sahil Motiramani

Div:D15C

RollNo:35

In this case, we have monitored -

Servers: 1 linux server

Services: swap

Ports: 22, 80 (ssh, http)

Processes: User status, Current load, total processes, root partition, etc.

Conclusion:

In this experiment, we set up port and server monitoring using Nagios.

Linux Instance: Hosts the Nagios dashboard and server.

2. Ubuntu Instance : Acts as the second monitored host.

3. Configuration:

- Add the Ubuntu instance's IP to the Nagios server's configuration.
- On the Ubuntu instance, configure the NRPE server and allow the Nagios server's IP.

4. Restart NRPE: After configuration, restart the NRPE service on Ubuntu.

5. Monitor: The Ubuntu instance will appear as "linuxserver" on the Nagios dashboard for monitoring.

Exp:11

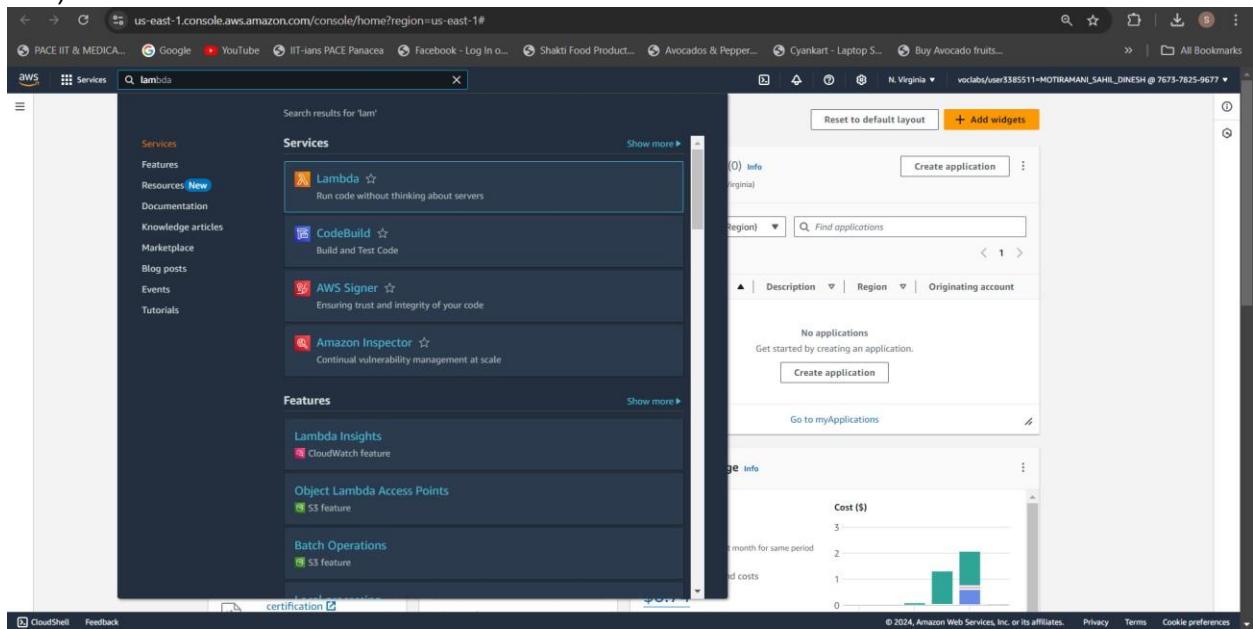
Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

Prerequisites:

- 1) AWS account (academy recommended)

Step 1: Set up AWS Lambda Function

- 1) Search for Lambda in the services tab. Click on it once found.



The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with navigation links like Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main area is titled 'Functions (5)' and lists five Lambda functions. Each function entry includes a checkbox, the function name, a brief description, the package type (Zip), the runtime (Python 3.8), and the last modified date (3 months ago).

Function Name	Description	Package type	Runtime	Last modified
RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	3 months ago
RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	3 months ago
RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	3 months ago
MainMonitoringFunction	-	Zip	Python 3.8	3 months ago
ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	3 months ago

- 2) Click on create functions.
- 3) Give a name to your Lambda function. Select the runtime as Node.js 20.x (You can also use python). Select the architecture as x86_64. Set the default execution role as LabRole if you are doing this on your academy account. (Use an existing role → LabRole)

The screenshot shows the 'Create Function' wizard. It consists of several tabs: 'Create function' (selected), 'Info', 'Basic information', 'Permissions info', 'Additional configurations', and 'Advanced configurations'. The 'Basic information' tab is active, showing fields for 'Function name' (sahil55), 'Runtime' (Node.js 20.x), and 'Architecture' (x86_64). The 'Permissions info' tab shows the 'Execution role' dropdown set to 'LabRole'. The 'Additional configurations' tab at the bottom indicates that additional configurations like code signing, function URL, tags, and Amazon VPC access will be set up later.

- 4) Once the function is created, click on the name of the function (myLambda27 in my case).

Functions (7)						
	Function name	Description	Package type	Runtime	Last modified	
<input type="checkbox"/>	myLambda27	-	Zip	Node.js 20.x	5 days ago	
<input type="checkbox"/>	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 months ago	
<input type="checkbox"/>	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago	
<input type="checkbox"/>	myLambda27_12	-	Zip	Python 3.12	5 days ago	
<input type="checkbox"/>	MainMonitoringFunction	-	Zip	Python 3.8	2 months ago	
<input type="checkbox"/>	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 months ago	
<input type="checkbox"/>	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago	

5) This is the dashboard of our lambda function.

Successfully created the function sahil35. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > sahil35

sahil35

Throttle Copy ARN Actions ▾

Function overview Info Export to Application Composer Download ▾

Diagram Template

sahil35
Layers (0)

+ Add trigger + Add destination

Description -
Last modified 10 seconds ago
Function ARN arnaws:lambda:us-east-1:767378259677:function:sahil35
Function URL Info -

Code Test Monitor Configuration Aliases Versions

Code source Info Upload from ▾

File Edit Find View Go Tools Window Test Deploy

index.mjs Environment Var

```
index.mjs
1 export const handle = async (event) => {
2   // Implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 };
9 
```

- 6) This function has the following default code, which is used to print "Hello form Lambda!".

The screenshot shows the AWS Lambda function editor. At the top, a message says "Successfully created the function sahilSS. You can now change its code and configuration. To invoke your function with a test event, choose "Test". The main area displays the code source for the function:

```

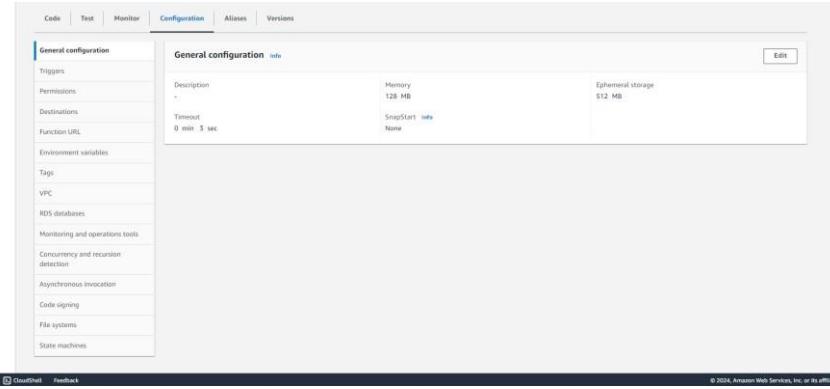
Code source info
File Edit View Go Tools Window Test Deploy
Go to Anything (Ctrl-P)
Environment Environment Variables
index.js
1 export const handler = async (event) => {
2   // Your implementation
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify("Hello from Lambda!")
6   };
7   return response;
8 };

```

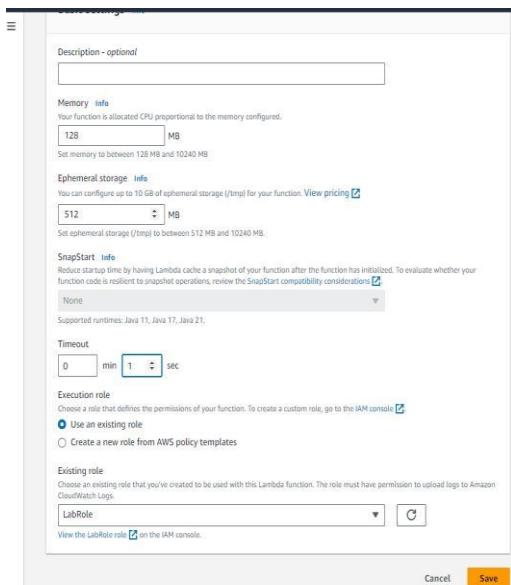
Below the code editor is the "Code properties" section, which includes fields for Package size, SHA256 hash, and Last modified.

Step 3: Set up configurations and test events

- 1) Just above the test code, you would find Configuration, click on it. Then click on Edit.



- 2) Here, change the Timeout to 1 sec. This is the time for which the function can be running before it is forcibly terminated.



3) We can see the executed changes.

The screenshot shows the General configuration tab of the Lambda function configuration page. The settings are identical to the previous screenshot:

- Description**: -
- Memory**: 128 MB
- Ephemeral storage**: 512 MB
- Timeout**: 0 min 1 sec
- SnapStart**: None

4) Switch back to the code tab. Click on the dropdown arrow near test. Then select configure test event.

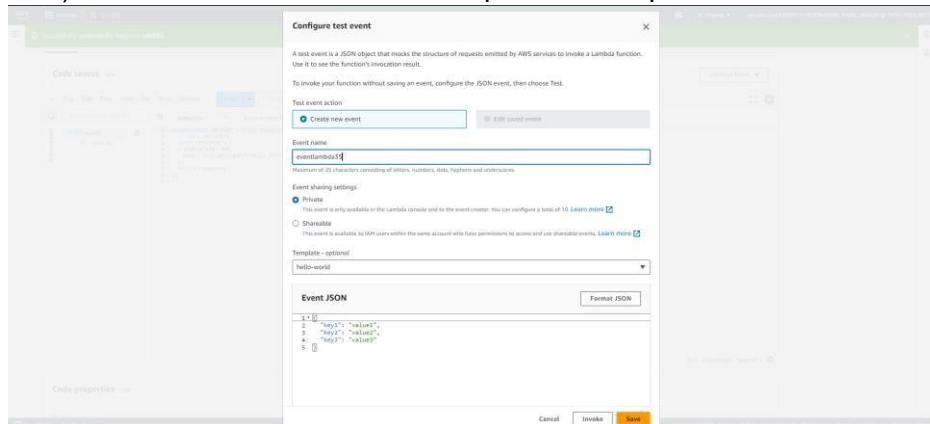
The screenshot shows the AWS Lambda code editor with the 'Test' tab selected. A dropdown menu is open, and the option 'Configure test event' is highlighted. The code editor displays the 'index.mjs' file content:

```

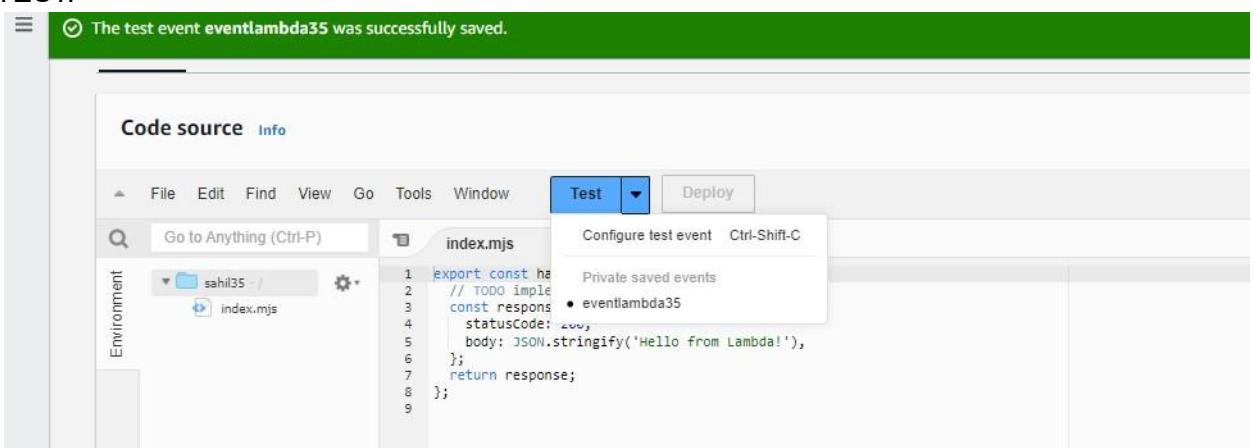
Code source Info
File Edit Find View Go Tools Window Test Deploy
Upload from ▾
Configure test event Ctrl-Shift-C
index.mjs
1 export const handler = async (event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 }
9

```

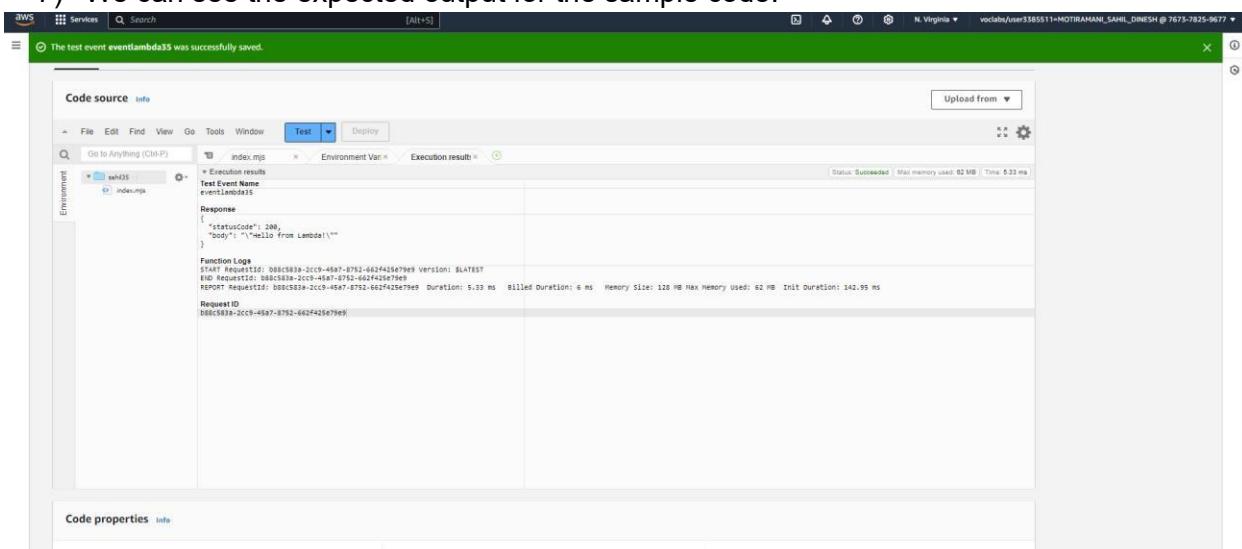
5) Here, create a new event, keep the other options default and save the event.



6) Now, again click on the dropdown. This time, select the event you have created. Then, click on TEST.



7) We can see the expected output for the sample code.



- 8) For a test, declare a string and call it in line 6. After making the changes click on deploy.

The screenshot shows the AWS Lambda Test interface. The code editor contains a file named index.js with the following content:

```

1 const test = "sahil";
2 exports.handler = async (event) => {
3     // TODO implement
4     const response = {
5         statusCode: 200,
6         body: JSON.stringify(test),
7     };
8     return response;
9 }
10

```

The deployment status bar at the top indicates "The test event eventlambdaSS was successfully saved." The "Test" tab is selected in the navigation bar.

- 9) Run the test. We can see that the string we declared has come in the output.

The screenshot shows the AWS Lambda Test interface after running the test. The code editor remains the same as in the previous screenshot. The execution results pane shows the following output:

```

Test Event Name: eventlambdaSS
Response:
{
    "statusCode": 200,
    "body": "sahil"
}

Function Log:
START RequestId: f97c08032-4531-4444-0830-c178663c982f Version: $LATEST
END RequestId: f97c08032-4531-4444-0830-c178663c982f
REPORT RequestId: f97c08032-4531-4444-0830-c178663c982f Duration: 4.06 ms Billed Duration: 5 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 140.97 ms
RequestID: f97c08032-4531-4444-0830-c178663c982f

```

The status bar at the top indicates "Successfully updated the function sahilSS." The "Test" tab is selected in the navigation bar.

Conclusion:

In this experiment, we effectively investigated the AWS Lambda service by developing and configuring Lambda functions using Python, Java, or Node.js. We discovered how to establish a Lambda function, tweak its settings (like modifying the timeout duration), and evaluate the function with personalized events. Throughout this experience, we noted how Lambda manages executions, including timeout handling and producing expected results according to modifications in the code.

Exp:12

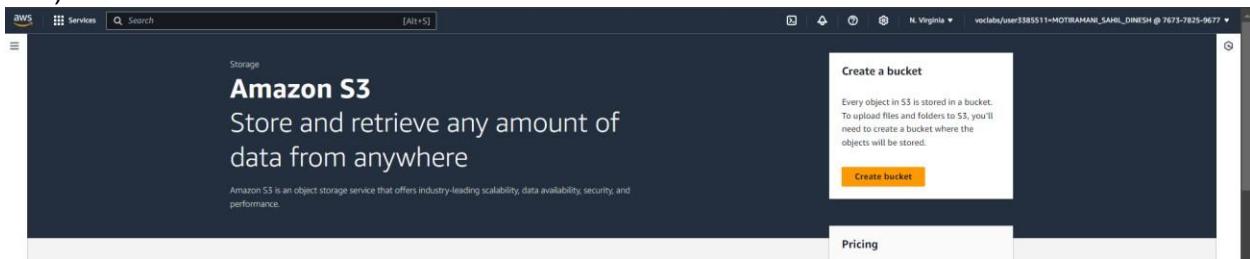
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Prerequisites:

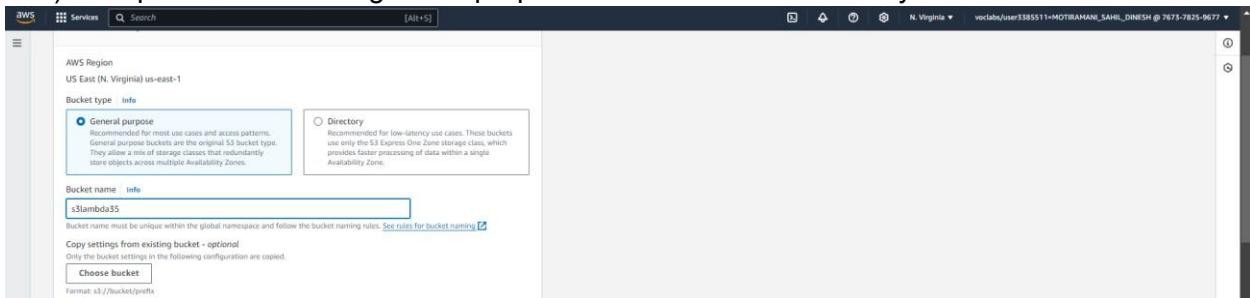
- 1) AWS account (academy preferable)
- 2) Lambda function (created in the previous experiment).

Step 1: Create a s3 bucket.

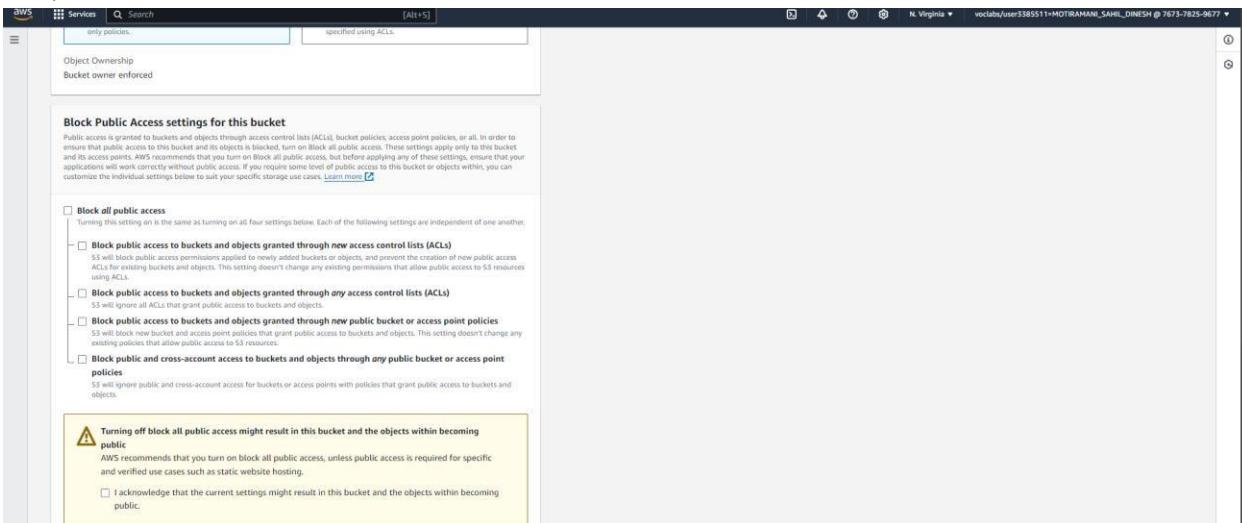
- 1) Search for S3 bucket in the services search. Then click on create bucket.



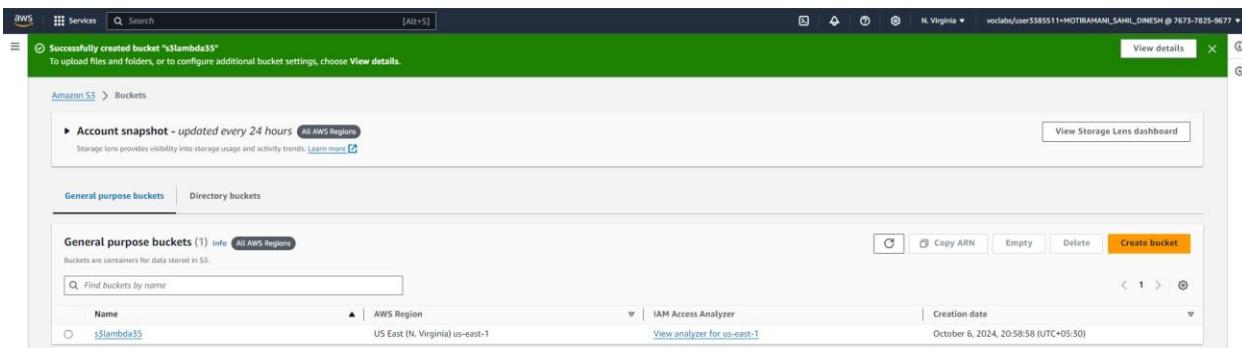
- 2) Keep the bucket as a general purpose bucket. Give a name to your bucket.



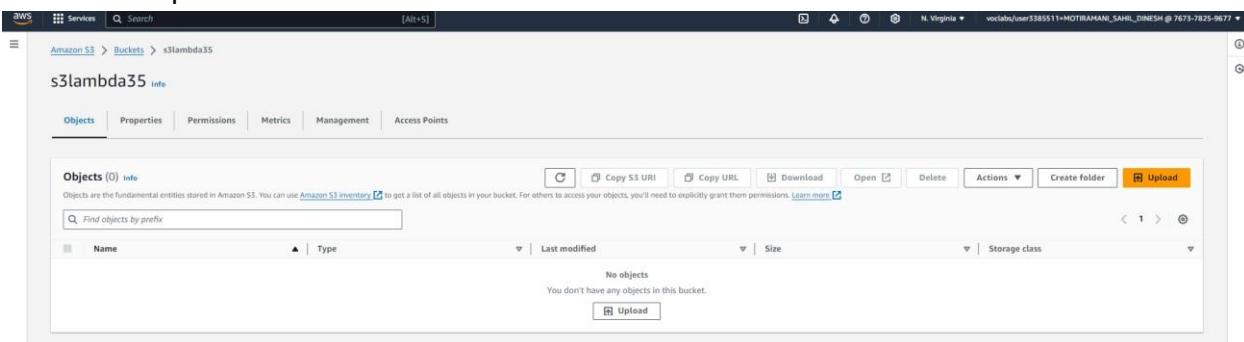
3) Uncheck block all public access.



4) Keeping all other options same, click on create. This would create your bucket. Now click on the name of the bucket.



5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload.



- 6) The image has been uploaded to the bucket.

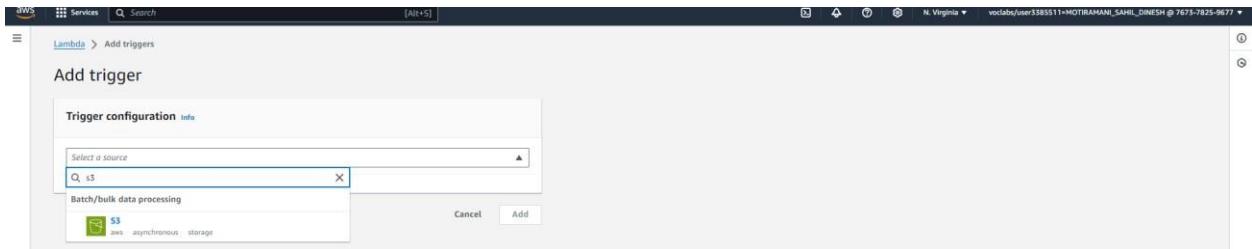
The screenshot shows two screenshots of the AWS S3 console. The top screenshot shows the 'Upload' interface where a file named '1.jpg' is being selected for upload. The bottom screenshot shows the 'Upload status' page indicating that the upload was successful, with 1 file (48.4 KB) having a success rate of 100%.

Step 2: Configure Lambda function

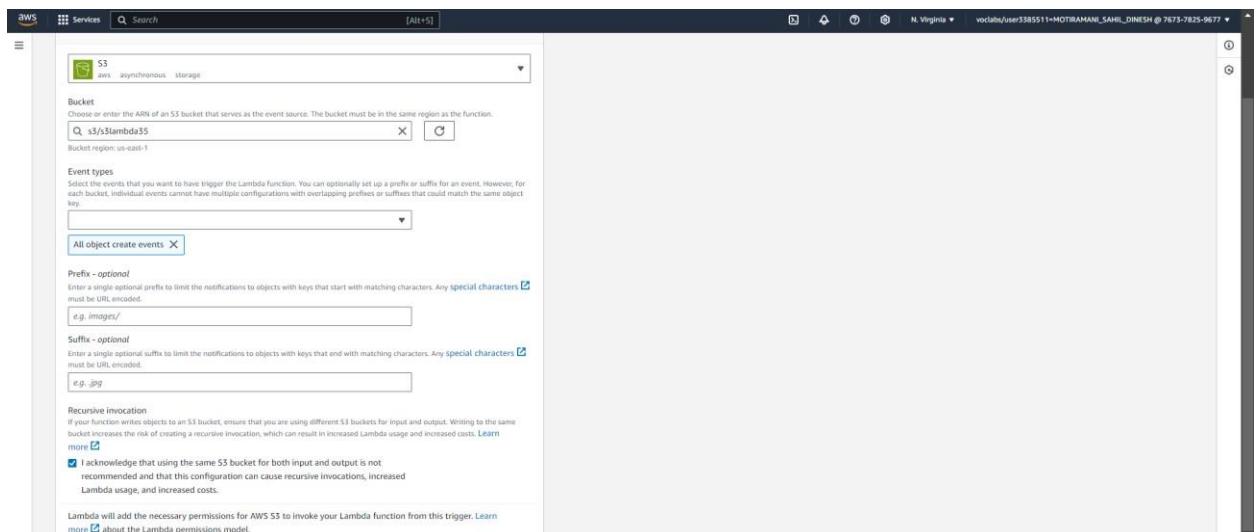
- 1) Go to the lambda function you had created before. (Services → Lambda → Click on name of function). Here, click on add trigger.

The screenshot shows the AWS Lambda console for the 'sahil35' function. It displays the 'Function overview' and 'General configuration' tabs. Under 'General configuration', settings include: Description (empty), Memory (128 MB), Ephemeral storage (512 MB), Timeout (0 min 1 sec), and Snapshot (None). The 'Edit' button is visible at the top right of this section.

- 2) Under trigger configuration, search for S3 and select it.



- 3) Here, select teh S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function.



4. Scroll down to the code section of the function. Add the following javascript code to the code area by replacingthe existing code

```
export const handler = async (event) => {
  if (!event.Records || event.Records.length === 0) {
    console.error("No records found in the event.");
    return { statusCode: 400, body: JSON.stringify('No records
      found in the event')}
  }
};
```

```

}

// Extract bucket name and object key from the event const record = event.Records[0]; const
bucketName = record.s3.bucket.name; const objectKey =
decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle
encoded keys

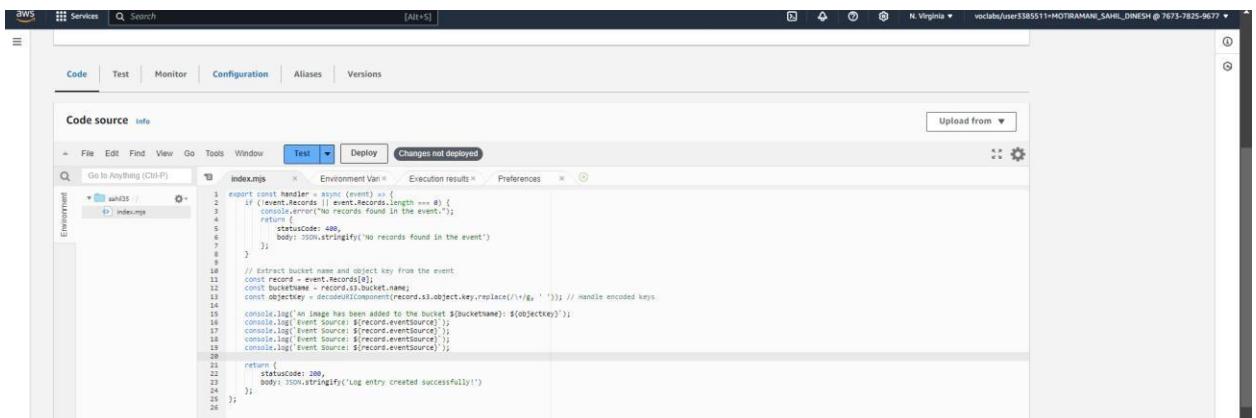
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
console.log(`Event Source: ${record.eventSource}`); console.log(`Event Source:
${record.eventSource}`); console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`); return {

  statusCode: 200, body: JSON.stringify('Log entry
  created successfully!')
};

}

```

This code checks for records in the event, extracts the bucket name and object key, logs the details, and returns a success message if an image is added to the bucket.



Now, click on the dropdown near test, then click on configure test event.

6) Here, select edit saved event. Select the event taht you had created before. Under Event JSON, paste the following code.

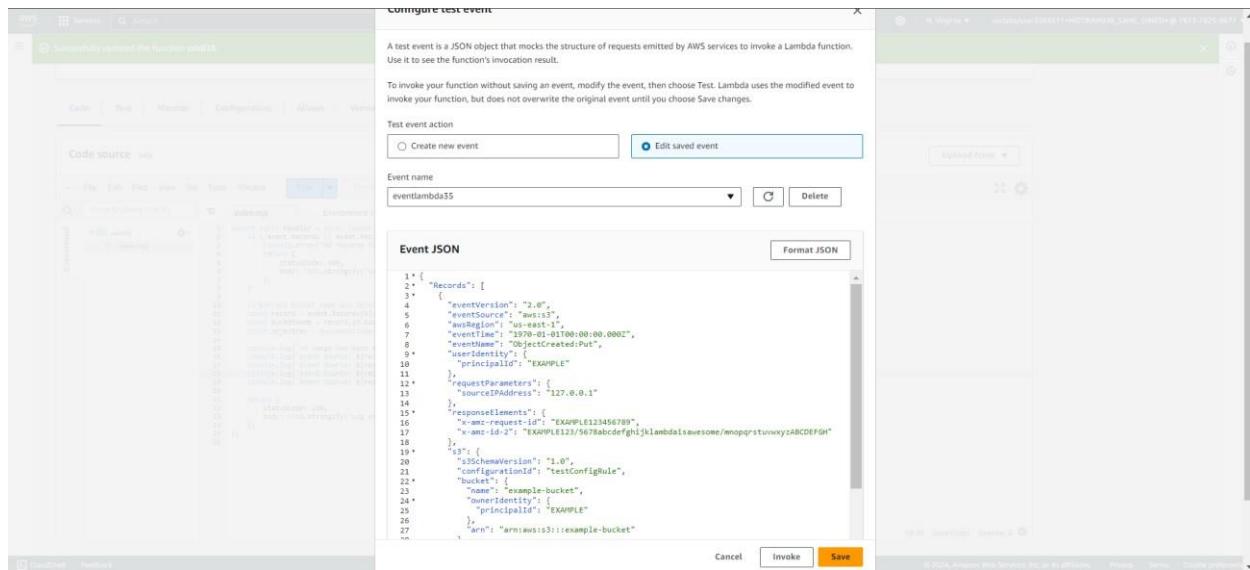
```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789", "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmabaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket", "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-bucket"
        }
      }
    }
  ]
}
```

```

},
"object": {
  "key": "test%2Fkey",
  "size": 1024,
  "eTag": "0123456789abcdef0123456789abcdef",
  "sequencer": "0A1B2C3D4E5F678901"
}
}
}
]
}

```

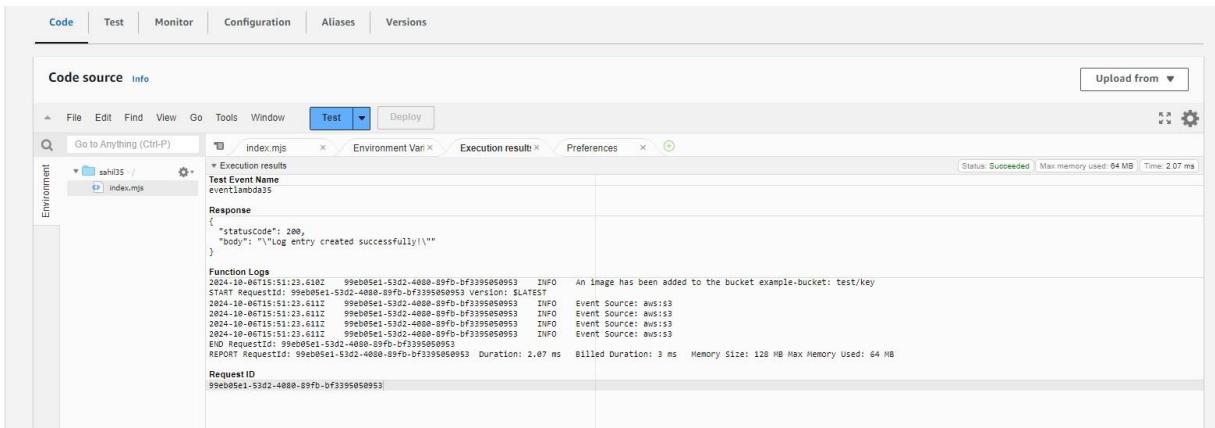
This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.



Save the changes. Then deploy the code changes by clicking on deploy.

7) After deploying, click on Test. The console output shows that ‘an image has been added to the bucket’

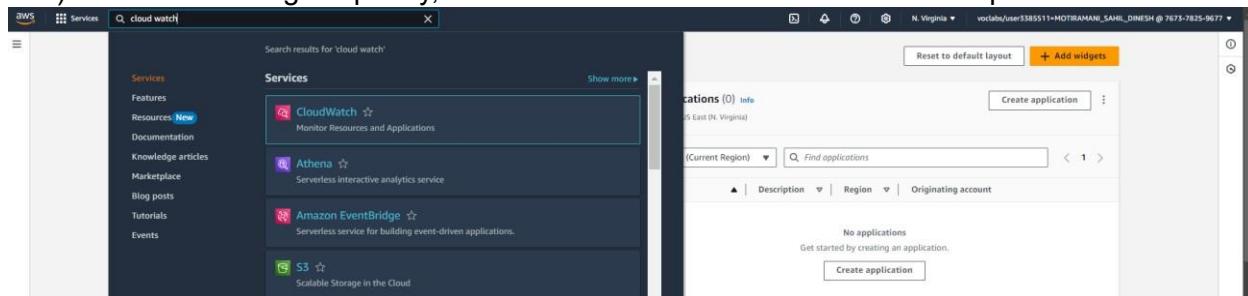
The JSON response shows that the log entry was created successfully.



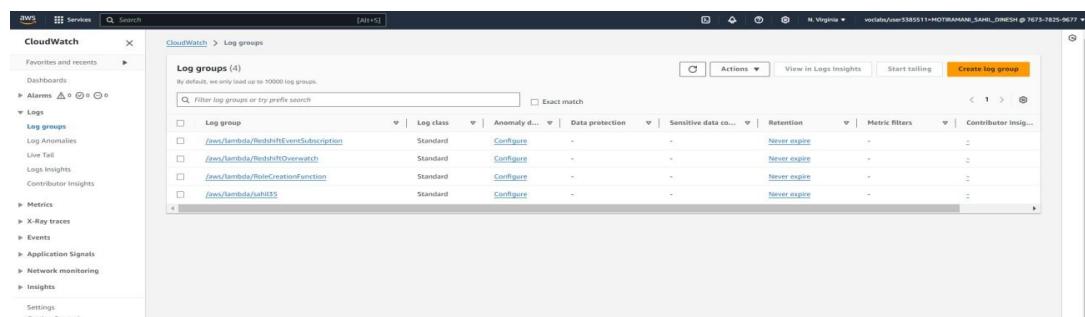
The screenshot shows the AWS Lambda Test interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Test' tab is selected. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, and a dropdown for Environment Variables. The main area displays the execution results for a function named 'index.mjs'. The 'Event' section shows an event named 'event1lambda035'. The 'Response' section shows a JSON object with a status code of 200 and a body containing the message 'log entry created successfully'. The 'Function Logs' section contains several log entries from AWS Lambda, detailing requests and responses. The logs indicate that an image was added to a bucket named 'example-bucket' under the key 'test/key'. The status bar at the bottom right shows 'Status: Succeeded', 'Max memory used: 64 MB', and 'Time: 2.07 ms'.

Step 3: Check the logs

1) To check the logs explicitly, search for CloudWatch on services and open it in a new tab.



2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.



3) Here, under Log streams, select the log stream you want to check.

The screenshot shows the AWS CloudWatch Log Groups interface. The left sidebar navigation includes CloudWatch, Services, and a search bar. Under Logs, the Log groups section is selected. The main content area shows the details for the log group `/aws/lambda/sahil35`. It includes sections for Log class (Info), Standard, ARN, Metric filters (0), Creation time (1 hour ago), Retention (Never expire), and various CloudWatch features like Anomaly detection, Data protection, and Contributor Insights rules. Below this, the Log streams tab is selected, showing three log entries from October 6, 2024, with the last event time being 14:47 UTC.

4) Here again, we can see that 'An image has been added to the bucket'.

The screenshot shows the AWS CloudWatch Log Events interface for the log stream `2024/10/06/[SLATEST]d475d9ade6054435995f9aaaa948219e4`. The left sidebar navigation is identical to the previous screenshot. The main content area shows a table of log events. One entry in the table is highlighted, showing the timestamp `2024-10-06T15:46:47.072Z` and the message `REPORT RequestId: e2260cc0-1951-4ef7-9a1d-5c9204a5dc Duration: 47.61 ms Billed Duration: 48 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 159.25 ms`. This message includes the key information: 'INFO An image has been added to the bucket example-bucket: test/key'.

Conclusion:

In this experiment, we developed and deployed a Lambda function designed to respond to file uploads in an S3 bucket. The function was triggered automatically whenever a new object was added to the bucket, illustrating how AWS services can efficiently automate workflows. The Lambda function extracted and logged key details from the event, such as the bucket's name and the object's key. We tested this by uploading a sample file, and upon reviewing the logs in CloudWatch, we confirmed that the function executed successfully, capturing the upload event.