

Name : Himesh Pathai

Div : D10A

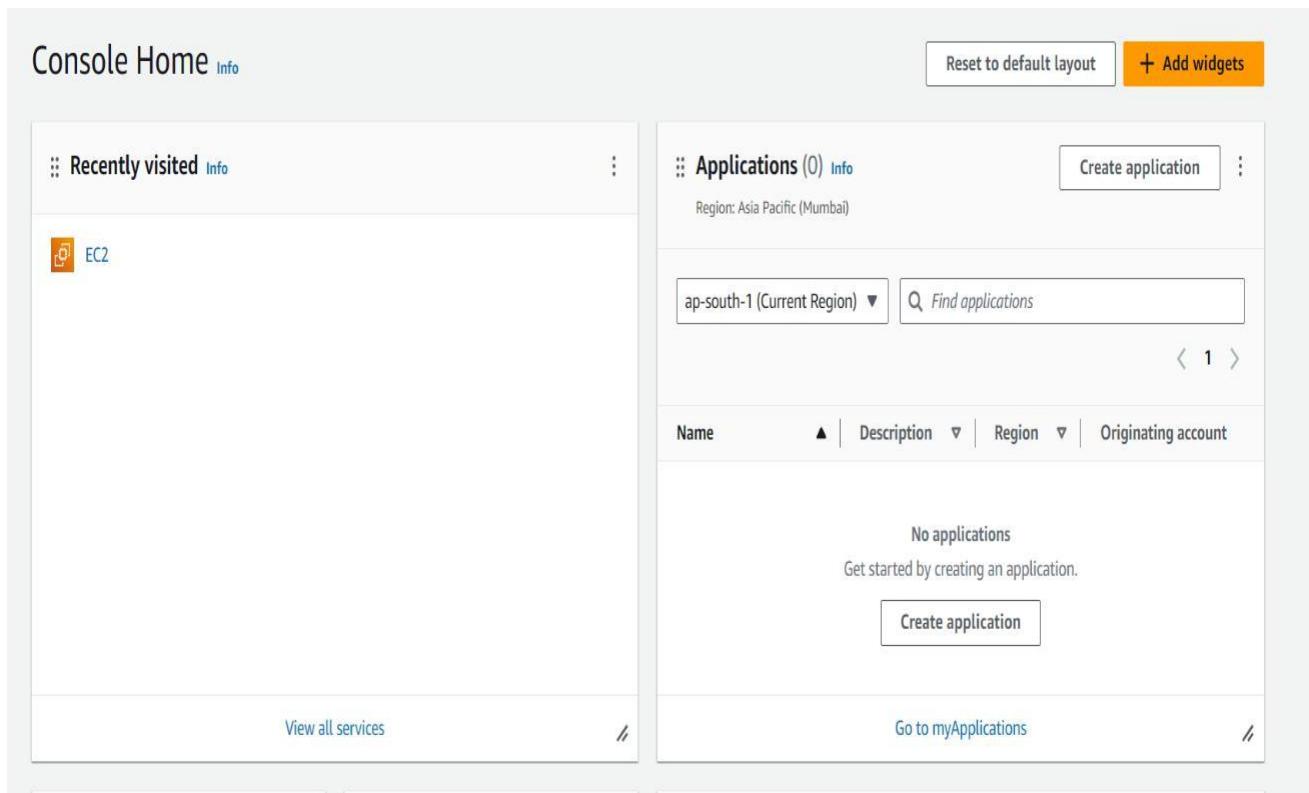
Roll No. : 35

## EXPERIMENT NO. 1

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

### EC2 Instance Creation and static site hosting

1. Login to your AWS account



2. Click on EC2 and then create an instance by clicking on instances

The screenshot shows the AWS EC2 Home page. On the left, there's a summary of resources: Instances (running) 0, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. Below this is a 'Launch instance' section with a large orange 'Launch instance' button and a smaller 'Migrate a server' link. A note says 'Note: Your instances will launch in the Asia Pacific (Mumbai) Region'. To the right is a 'Service health' section showing the region as 'Asia Pacific (Mumbai)' and the status as 'This service is operating normally.' Further right is an 'EC2 Free Tier' section with a link to 'View Global EC2 resources' and a 'View all AWS Free Tier offers' link. At the bottom right is an 'Account attributes' section with a 'Default VPC' entry (vpc-0b8f24d2c64f977f) and a 'Settings' link.

### 3. After an instance is created wait for it to come to Running state

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations (New), Images (selected), AMIs, AMI Catalog, and Elastic Block Store. The main content area is titled 'Instances (1/1) Info' and shows one instance: 'aws 1' (Instance ID: i-0e6218a10f73b4de7). The instance is listed as 'Running' (Status check: Initializing), with its type as 't2.micro'. It is located in the 'ap-south-1b' Availability Zone and has a Public IPv4 address of 'ec2-52-66-25'. There are buttons for 'Connect', 'Actions', and 'Launch instances'.

### 4. After doing that you will see this UI

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

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

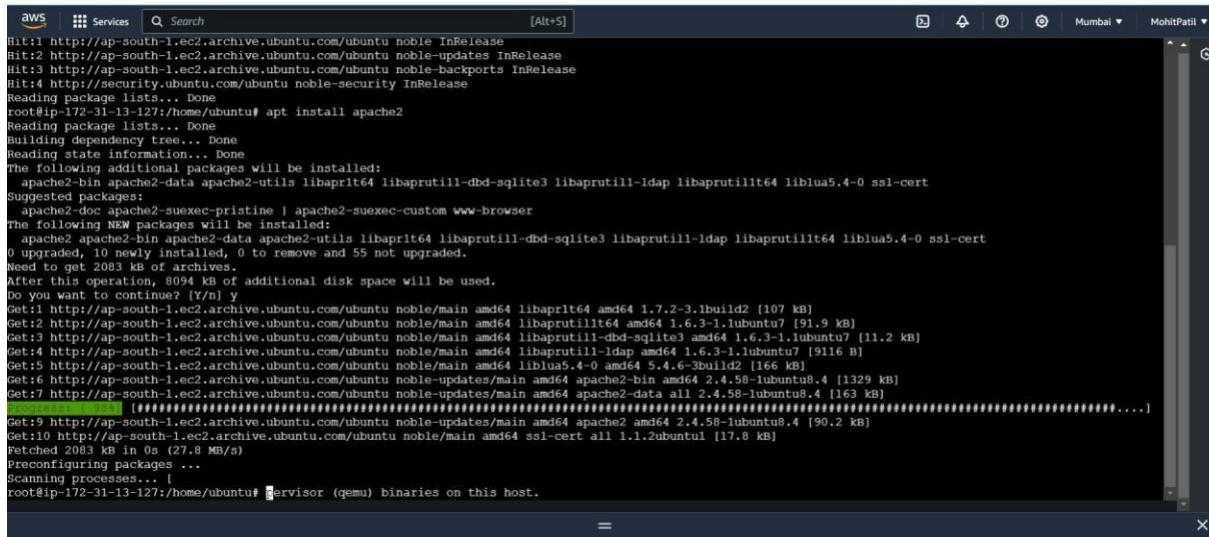
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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

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

ubuntu@ip-172-31-13-127:~$ sudo su
root@ip-172-31-13-127:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [296 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

## 5. Follow these steps and then run these commands



```
aws Services Search [Alt+S]
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
root@ip-172-31-13-127:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libaprilt64 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 55 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprilt64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils all 2.4.58-1ubuntu8.4 [163 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
Fetched 2083 kB in 0s (27.8 MB/s)
Preconfiguring packages...
Scanning processes... [root@ip-172-31-13-127:/home/ubuntu# ]ervisor (qemu) binaries on this host.
```

## 6. After that the ip-address which was given while running the instance, copy that and paste that on chrome, make sure that it is http and not https



The screenshot shows a web browser window with the Apache2 Default Page for Ubuntu. The page features the Ubuntu logo and the text "Apache2 Default Page". A red button labeled "It works!" is prominently displayed. Below the button, there is descriptive text about the default welcome page and configuration overview.

**Apache2 Default Page**

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

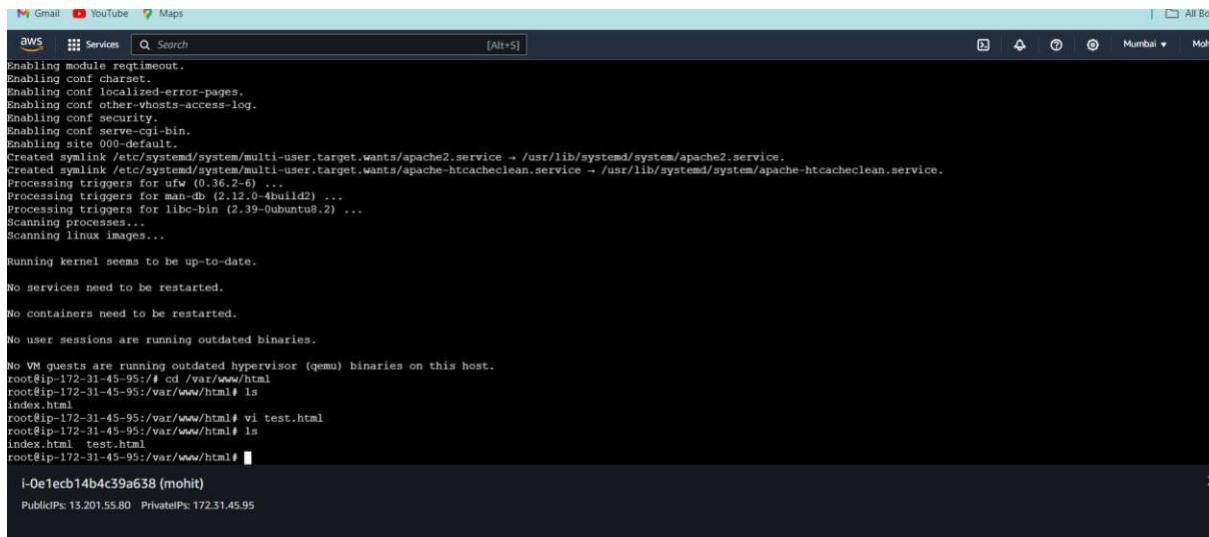
**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.Load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

## 7. Create a file using vi command and save it using :wq



```
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-hosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu0.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

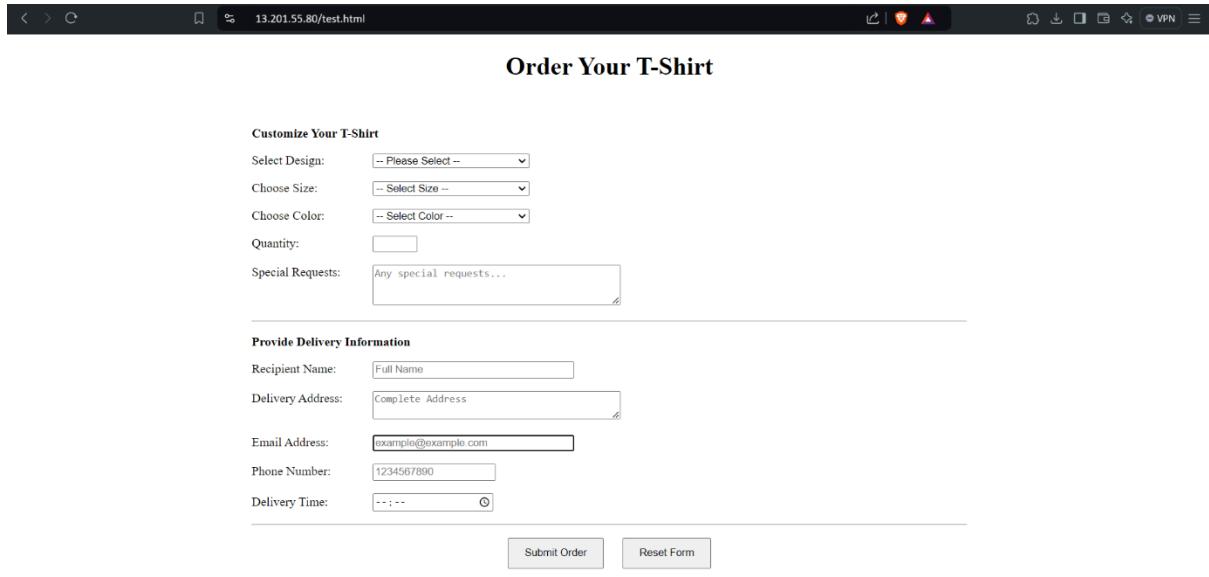
No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-45-95:~# cd /var/www/html
root@ip-172-31-45-95:/var/www/html# ls
index.html
root@ip-172-31-45-95:/var/www/html# vi test.html
root@ip-172-31-45-95:/var/www/html# ls
index.html test.html
root@ip-172-31-45-95:/var/www/html# l
l-0eTech14b4c39a638 (mohit)
PublicIPs: 13.201.55.80 PrivateIPs: 172.31.45.95
```

## 8. After saving that file go that page where ubuntu is listed and then on the link add “/your\_file\_name.html” and then whatever you saved on that file will be displayed



The screenshot shows a web browser window with the URL `13.201.55.80/test.html` in the address bar. The page content is as follows:

**Order Your T-Shirt**

**Customize Your T-Shirt**

Select Design:

Choose Size:

Choose Color:

Quantity:

Special Requests:

---

**Provide Delivery Information**

Recipient Name:

Delivery Address:

Email Address:

Phone Number:

Delivery Time:

---

# Static Hosting using S3 bucket

## Step1: Create bucket

The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section is active. The 'Bucket name' field contains 'himeshbucket'. 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](#)'.

AWS Region: Asia Pacific (Mumbai) ap-south-1

Bucket name:  [Info](#)

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

Format: s3://bucket/prefix

The screenshot shows the 'Edit static website hosting' page for the 'mohitpatilbucket' bucket. The 'Static website hosting' section is active. The 'Enable' radio button is selected. The 'Hosting type' section shows 'Host a static website' selected, with a note: 'Use the bucket endpoint as the web address. [Learn more](#)'. The 'Redirect requests for an object' option is also present with its own note.

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.

## Step 2: Add resources

The screenshot shows the AWS S3 console interface. At the top, there is a green success message: "Upload succeeded. View details below." Below this, the "Details" section shows a file named "index.html" with a status of "Succeeded". The "Download" and "Edit" buttons are also visible. The main area displays a table titled "Files and folders (59 Total, 176.4 KB)" containing various files and their details. The columns include Name, Folder, Type, Size, Status, and Last modified.

Name	Folder	Type	Size	Status	Last modified
background.jpg	ip-all-hosters...	image/jpeg	19.3 KB	Succeeded	...
classified.jpg	ip-all-hosters...	image/jpeg	7.8 KB	Succeeded	...
images.jpg	ip-all-hosters...	image/jpeg	5.9 KB	Succeeded	...
index.html	ip-all-hosters...	text/html	8.1 KB	Succeeded	...
style.css	ip-all-hosters...	text/css	995.8 B	Succeeded	...
wallpaper.jpg	ip-all-hosters...	image/jpeg	48.8 KB	Succeeded	...
COMMIT_001...	ip-all-hosters...	-	13.8 B	Succeeded	...
config	ip-all-hosters...	-	337.8 B	Succeeded	...
description	ip-all-hosters...	-	73.8 B	Succeeded	...
readme	ip-all-hosters...	-	21.8 B	Succeeded	...

The screenshot shows the AWS S3 bucket properties page for "himeshbucket". The "Properties" tab is selected. The "Bucket overview" section displays basic information: AWS Region (Asia Pacific (Mumbai) ap-south-1), Amazon Resource Name (ARN) (arn:aws:s3:::himeshbucket), and Creation date (August 21, 2024, 20:58:25 UTC+05:30). The "Bucket Versioning" section indicates that versioning is disabled. The "Multi-factor authentication (MFA) delete" section also indicates that MFA delete is disabled. The "Tags" section shows an empty tag list.

Screenshot of the AWS S3 console showing the "Make public: status" page. A green header bar indicates "Successfully edited public access". The main summary table shows 39 objects, 176.4 KB total size, and 0 failed edits. Below the summary is a table titled "Failed to edit public access (0)" which is currently empty.

Name	Folder	Type	Last modified	Size	Error
No objects failed to edit					

### Step 3 : visit hosted website

Screenshot of a web browser displaying a custom T-shirt ordering form. The URL is 13.201.55.80/test.html. The form is divided into two sections: "Customize Your T-Shirt" and "Provide Delivery Information".

**Customize Your T-Shirt**

Select Design:

Choose Size:

Choose Color:

Quantity:

Special Requests:

---

**Provide Delivery Information**

Recipient Name:

Delivery Address:

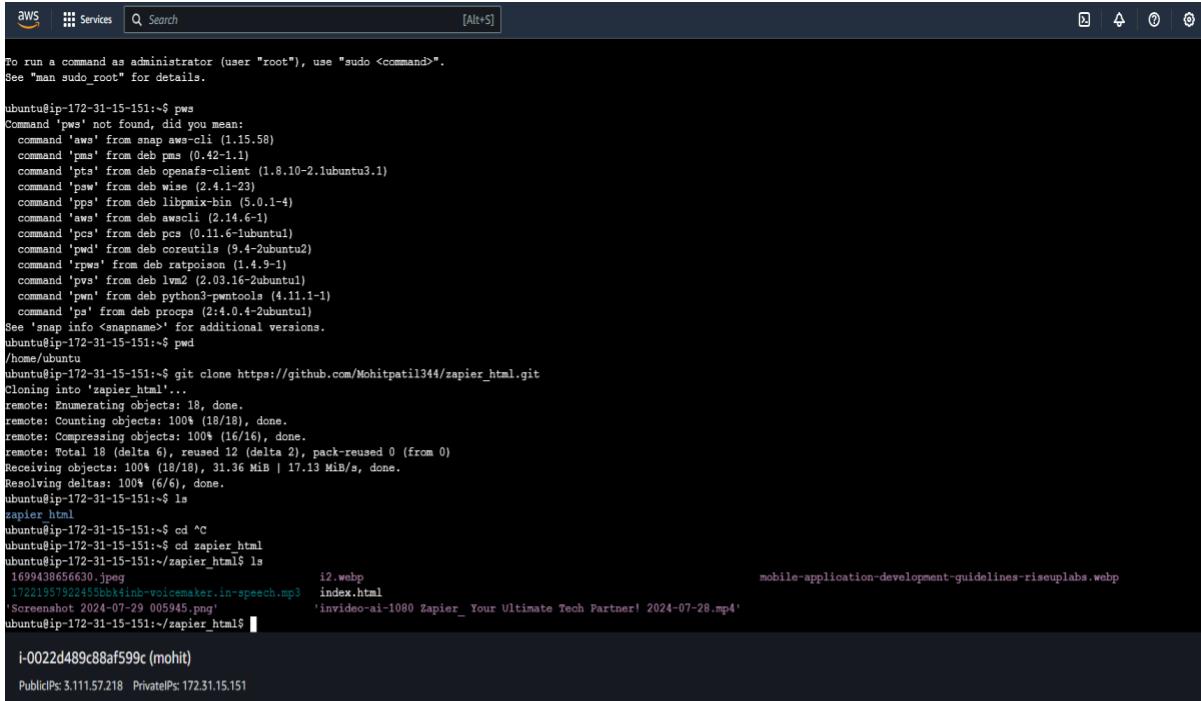
Email Address:

Phone Number:

Delivery Time:

# EC2 Dynamic Site Hosting

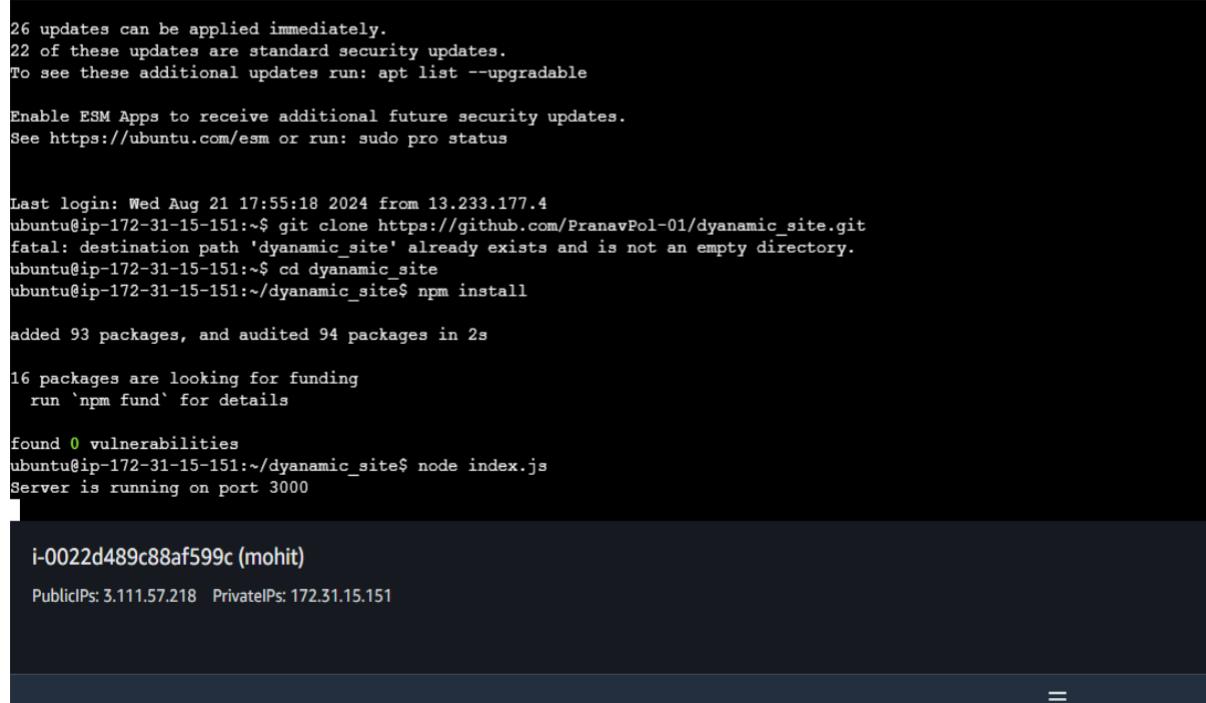
Step 1 : Open Console and clone the github repository



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

ubuntu@ip-172-31-15-151:~$ pws
Command 'pws' not found, did you mean:
  command 'aws' from snap aws-cli (1.15.58)
  command 'pms' from deb pmx (0.42-1.1)
  command 'pts' from deb openafs-client (1.8.10-2.1ubuntu3.1)
  command 'pws' from deb wise (2.4.1-23)
  command 'pps' from deb libpmix-bin (5.0.1-4)
  command 'aws' from deb awscli (2.14.6-1)
  command 'pcs' from deb pacu (0.11.6-1ubuntu1)
  command 'pwd' from deb coreutils (9.4-2ubuntu2)
  command 'pws' from deb ratpoison (1.4.9-1)
  command 'pwn' from deb python3-pwntools (4.11.1-1)
  command 'ps' from deb procps (2:4.0.4-2ubuntu1)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-15-151:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-15-151:~$ git clone https://github.com/Mohitpatil344/zapier_html.git
Cloning into 'zapier_html'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 18 (delta 6), reused 12 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (18/18), 31.36 MiB | 17.13 MiB/s, done.
Resolving deltas: 100% (6/6), done.
ubuntu@ip-172-31-15-151:~$ ls
zapier_html
ubuntu@ip-172-31-15-151:~$ cd zapier_html
ubuntu@ip-172-31-15-151:~/zapier_html$ ls
1699438656630.jpeg          i2.webp
17221957322455bh4inh-wolcemaker_in-speech.mp3 index.html      mobile-application-development-guidelines-riseuplabs.webp
'Screenshot 2024-07-29 005945.png' 'invideo-ai-1080 Zapier_ Your Ultimate Tech Partner! 2024-07-28.mp4'
ubuntu@ip-172-31-15-151:~/zapier_html$ 
```

i-0022d489c88af599c (mohit)  
PublicIPs: 3.111.57.218 PrivateIPs: 172.31.15.151



```
26 updates can be applied immediately.
22 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: Wed Aug 21 17:55:18 2024 from 13.233.177.4
ubuntu@ip-172-31-15-151:~$ git clone https://github.com/PranavPol-01/dyanamic_site.git
fatal: destination path 'dyanamic_site' already exists and is not an empty directory.
ubuntu@ip-172-31-15-151:~$ cd dyanamic_site
ubuntu@ip-172-31-15-151:~/dyanamic_site$ npm install

added 93 packages, and audited 94 packages in 2s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-15-151:~/dyanamic_site$ node index.js
Server is running on port 3000

i-0022d489c88af599c (mohit)
PublicIPs: 3.111.57.218 PrivateIPs: 172.31.15.151
```

## Step 2 : Install necessary Packages and run website on port 3000



## Cloud 9 IDE Site Hosting

The screenshot shows the AWS Cloud9 homepage. At the top right, there is a white box with the text "New AWS Cloud9 environment" and a blue "Create environment" button.

### Details

Name: himesh  
Description - optional:  
Environment type:  New EC2 instance  
 Existing compute

### New EC2 instance

Instance type:  t2.micro (1 GiB RAM + 1 vCPU)  
 t3.small (2 GiB RAM + 2 vCPU)  
 m5.large (8 GiB RAM + 2 vCPU)  
 Additional instance types  
Platform: Amazon Linux 2023  
Timeout: 30 minutes

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

AWS Cloud9 > Environments

**Environments (1)**

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
himesh	<a href="#">Open</a>	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::554378108602:assumed-role/voclabs/user3402848=PATANKAR_ARYAN_ANIL

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

Test123 - homek

Welcome

Developer Tools

# AWS Cloud9

## Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications using the AWS Cloud9 IDE.

Getting started

Create File

Upload Files...

Clone from GitHub

bash - ip-172-31-11-129 x Immediate vortabs:~/environment \$

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

Test123 - homek

IPLab-02

- download (1).jfif
- download (1).png
- download (2).jfif
- download (2).png
- download (3).png
- download (4).png
- download.png
- index.html
- introduction.mp3
- promotional-video.m4v
- style.css
- README.md

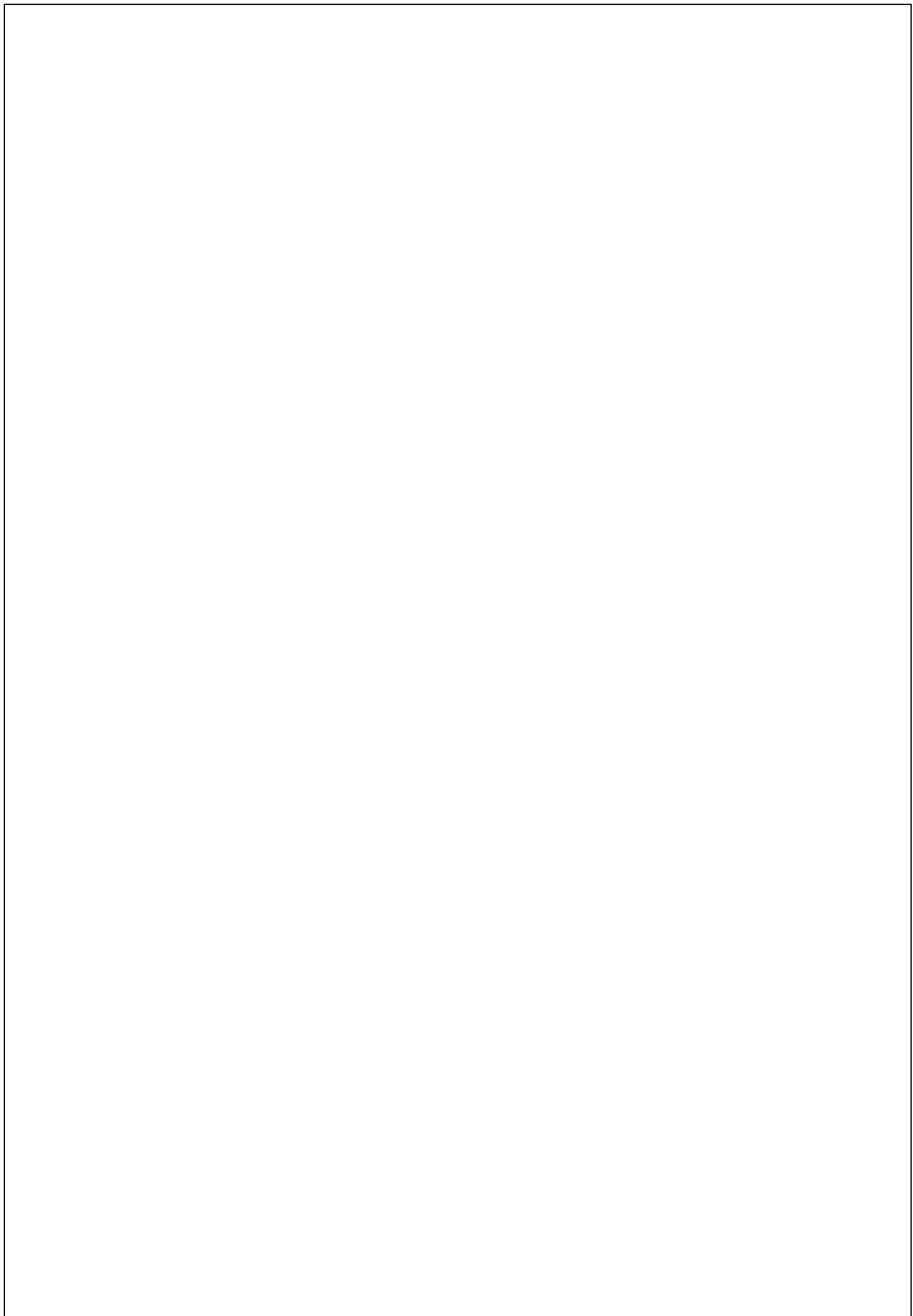
Welcome bash - ip-172-31-11-1 x index.html [B] /IPLab-02/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Amazon - Home</title>
    <link rel="stylesheet" href="style.css">
</head>
<body style="background-color: #f4f4f4; color: #333; padding: 10px;">
    <h1>Amazon</h1>
    
    <nav>
        <ul style="list-style-type: none; padding: 0; text-align: center;">
            <li><a href="#">Services</a></li>
            <li><a href="#">Our Services</a></li>
            <li><a href="#">Amazon Offers a vast selection of products, ranging from books to electronics, apparel to home goods, and more!</a></li>
        </ul>
    </nav>
    <main style="background-color: #f4f4f4; padding: 20px;">
        <section id="services">
            <h2>Our Services</h2>
            <div class="service-item" style="margin-bottom: 20px;">
                <img alt="Online Retail" style="width: 100px; height: 100px; border-radius: 50%; margin-right: 10px;">
                <a href="https://www.amazon.com" target="_blank">Amazon offers a vast selection of products, ranging from books to electronics, apparel to home goods, and more!</a>
            </div>
        </section>
    </main>
</body>
```

bash - ip-172-31-11-129 x Immediate vortabs:~/environment \$

Flipkart

Our Products



Name : Himesh Pathai

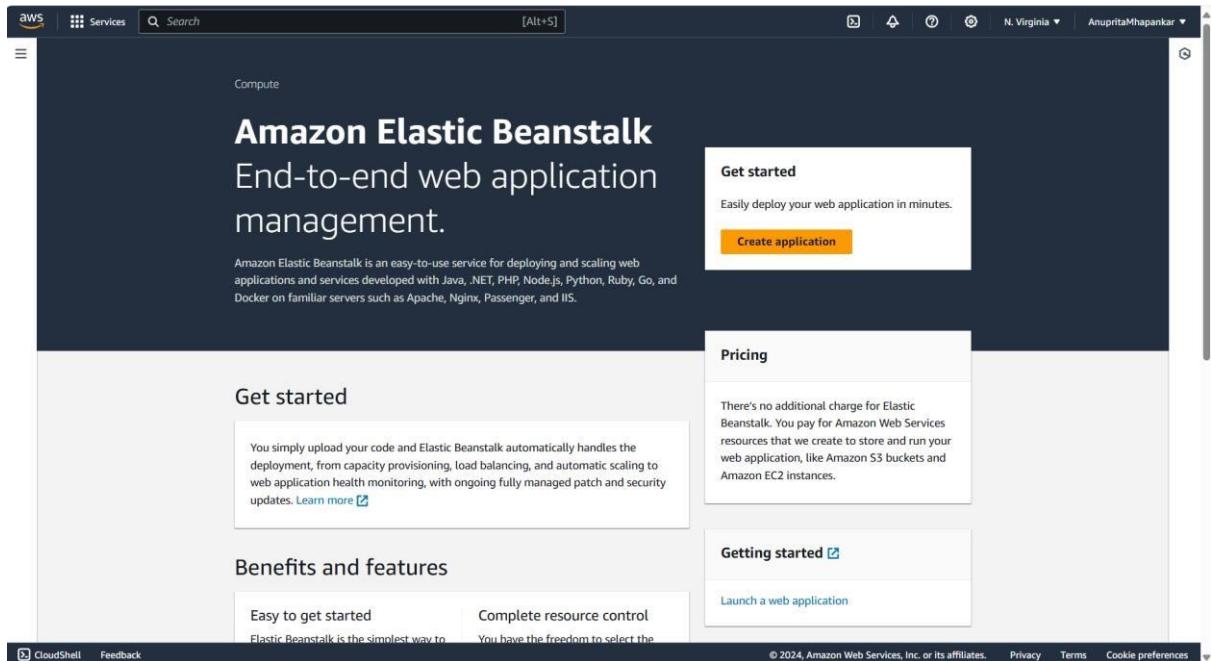
Div : D10A

Roll No. : 35

## Experiment 2

### Using Beanstalk

#### 1. Search Elastic Beanstalk from Developer Tools



The screenshot shows the AWS Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a services menu, a search bar, and account information for 'N. Virginia' and 'AnupritaMhapankar'. Below the header, the page title 'Amazon Elastic Beanstalk' and subtitle 'End-to-end web application management.' are displayed. A descriptive paragraph explains that Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. On the right side, there are two main call-to-action boxes: 'Get started' (with a 'Create application' button) and 'Pricing' (describing no additional charge). On the left, there's a 'Get started' section with a box explaining the deployment process and a 'Benefits and features' section with two items: 'Easy to get started' (Elastic Beanstalk is the simplest way to...) and 'Complete resource control' (You have the freedom to select the...). At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

#### 2. Click on create application and configure the environment

Configure environment [Info](#)

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

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

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

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

Application name  
  
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.

### 3. Choose PHP from the dropdown menu and click next

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

Platform [Info](#)

Platform type

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

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

Platform

Platform branch

Platform version

Application code [Info](#)

**Sample application**

Existing version  
Application versions that you have uploaded.

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

Presets [Info](#)  
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default

4. From the dropdown menu select the key pair and instance profile

The screenshot shows the 'Configure service access' step in the AWS Elastic Beanstalk setup process. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access, currently selected), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), and Step 5 (optional: Configure updates, monitoring, and logging). Step 6 (Review) is at the bottom.

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

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

**Service role name:** Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.  
aws-elasticbeanstalk-service-role [View permission details](#)

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

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

At the bottom are buttons: Cancel, Skip to review, Previous, and a large orange Next button.

5. Review the changes made and click on Submit

The screenshot shows the AWS Lambda configuration interface. In the 'Environment properties' section, there is a table with one row: 'No environment properties'. A message below the table states 'There are no environment properties defined'. At the bottom right of the configuration panel, there are 'Cancel', 'Previous', and 'Submit' buttons.

## Pipeline Creation :

### 1. Fork a github repo for aws codepipeline.

The screenshot shows the GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0'. The repository is public and has 20 commits. The 'About' section describes it as a sample for creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial. It includes links to the README, Code of conduct, Apache-2.0 license, Activity, 4 stars, 3 watching, 445 forks, and a report repository button. The 'Releases' section indicates 'No releases published'. The 'Packages' section indicates 'No packages published'.

## 2. Go to developer tools and select CodePipeline and create a new pipeline

The screenshot shows the AWS CodePipeline Pipelines page. On the left, there is a sidebar with a tree view of pipeline components: Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), Getting started, Pipelines (selected), and Settings. Below these are links to Go to resource and Feedback. The main content area has a header "Pipelines info" with buttons for Create pipeline, Notify, View history, Release change, and Delete pipeline. A search bar is at the top. Below it is a table with columns: Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A message "No results" and "There are no results to display." is centered. At the bottom, there are CloudShell and Feedback links, and a footer with copyright information.

## 3. Name your pipeline and select the desired service role

The screenshot shows the "Pipeline settings" step in the AWS CodePipeline wizard. On the left, a sidebar lists steps: Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main area is titled "Pipeline settings". It includes fields for "Pipeline name" (set to "himesh") and "Pipeline type" (set to "Queued (Pipeline type V2 required)"). A note says "You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model." Under "Execution mode", "Queued" is selected. Under "Service role", "New service role" is selected with the role name "AWSCodePipelineServiceRole-us-east-1-anupritaPipelineNew". A checkbox "Allow AWS CodePipeline to create a service role so it can be used with this new pipeline" is checked. At the bottom, there are CloudShell and Feedback links, and a footer with copyright information.

S | Services | Search | [Alt+S] | N. Virginia | AnupritaMhapankar | ⓘ | ⓘ

Service role

New service role  
Create a service role in your account

Existing service role  
Choose an existing service role from your account

Role name  
AWSCodePipelineServiceRole-us-east-1-anupritaPipeline

Type your service role name

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

**Variables**

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

Add variable  
You can add up to 50 variables.

ⓘ The first pipeline execution will fail if variables have no default values.

► Advanced settings

Cancel | **Next**

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

4. In the source stage select Github v2 as the provider and then connect your github

The screenshot shows the AWS CodeSuite interface for creating a connection to GitHub. At the top, the URL is https://us-east-1.console.aws.amazon.com/codesuite/settings/connections/creat... . The navigation bar includes the AWS logo, Services, a search icon, and a 'More' dropdown. Below the navigation, the breadcrumb trail shows 'Developer Tools > ... > Create connection'. The main title is 'Create a connection' with an 'Info' link. A sub-section titled 'Create GitHub App connection' also has an 'Info' link. A 'Connection name' field contains 'MyConnection'. Below it, a section labeled 'Tags - optional' is shown. At the bottom right is a large orange 'Connect to GitHub' button. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice for 2024, Amazon Web Services, Inc. or its affiliates.

Authorize AWS Connector for GitHub - Personal - Microsoft Edge

https://github.com/login/oauth/authorize?client\_id=lv1.ab636337c58c3ec...

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

Verify your GitHub identity (Anuprita579)

Know which resources you can access

Act on your behalf

[Learn more](#)

---

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

[Cancel](#) **Authorize AWS Connector for GitHub**

Authorizing will redirect to  
<https://redirect.codestar.aws>

Not owned or operated by GitHub

Created 4 years ago



https://github.com/apps/aws-connector-for-github/installations/new/per...



## Install AWS Connector for GitHub

Install on your personal account Anuprita Mhapankar 

for these repositories:

**All repositories**

This applies to all current *and* future repositories owned by the resource owner.  
Also includes public repositories (read-only).

**Only select repositories**

Select at least one repository.  
Also includes public repositories (read-only).

with these permissions:

- Read** access to issues and metadata
- Read and write** access to administration, code, commit statuses, pull

5. Once the connection is established from the drop down menu select the repository and the branch

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

Connection  
Choose an existing connection that you have already configured, or create a new one and then return to this task.  
arn:aws:codeconnections:us-east-1:557690619479:connection/fbe678f5-a05 X or [Connect to GitHub](#)

Ready to connect  
Your GitHub connection is ready for use.

Repository name  
Choose a repository in your GitHub account.  
himesh/aws-codepipeline-s3-codedeploy-linux-2.0 X  
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

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

Output artifact format  
Choose the output artifact format.

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

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

Trigger

## 6. Skip the build stage

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

Region  
US East (N. Virginia)

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

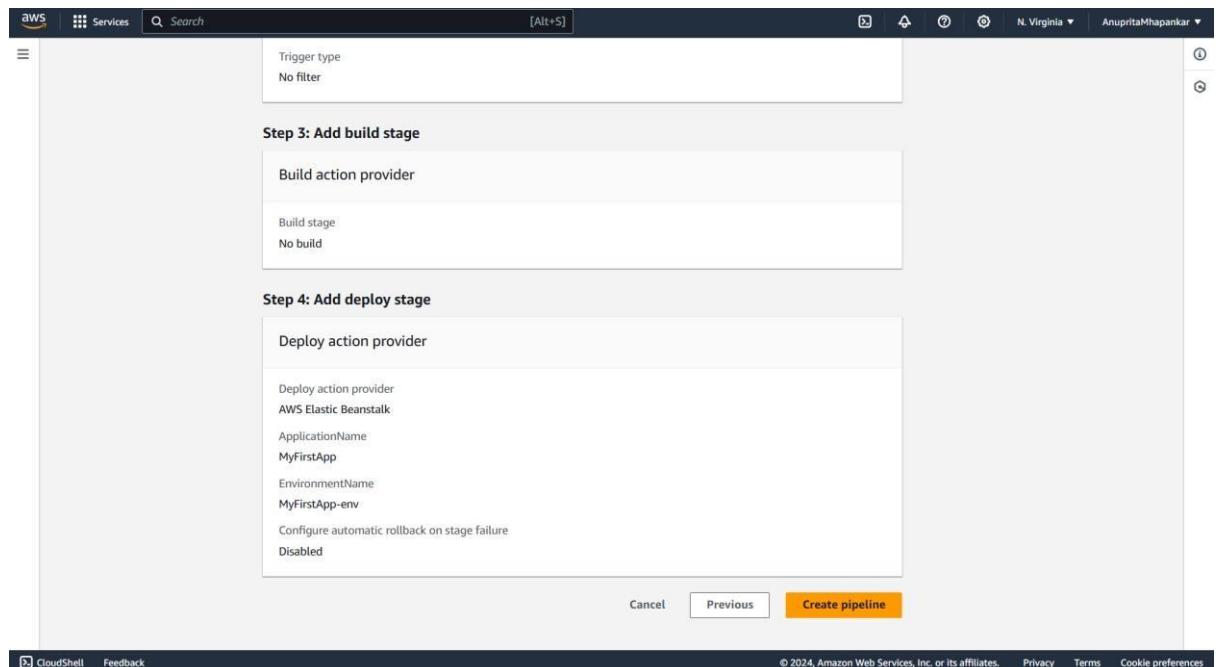
Application name  
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.  
myBeanApp

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.  
MyBeanApp-env

Configure automatic rollback on stage failure

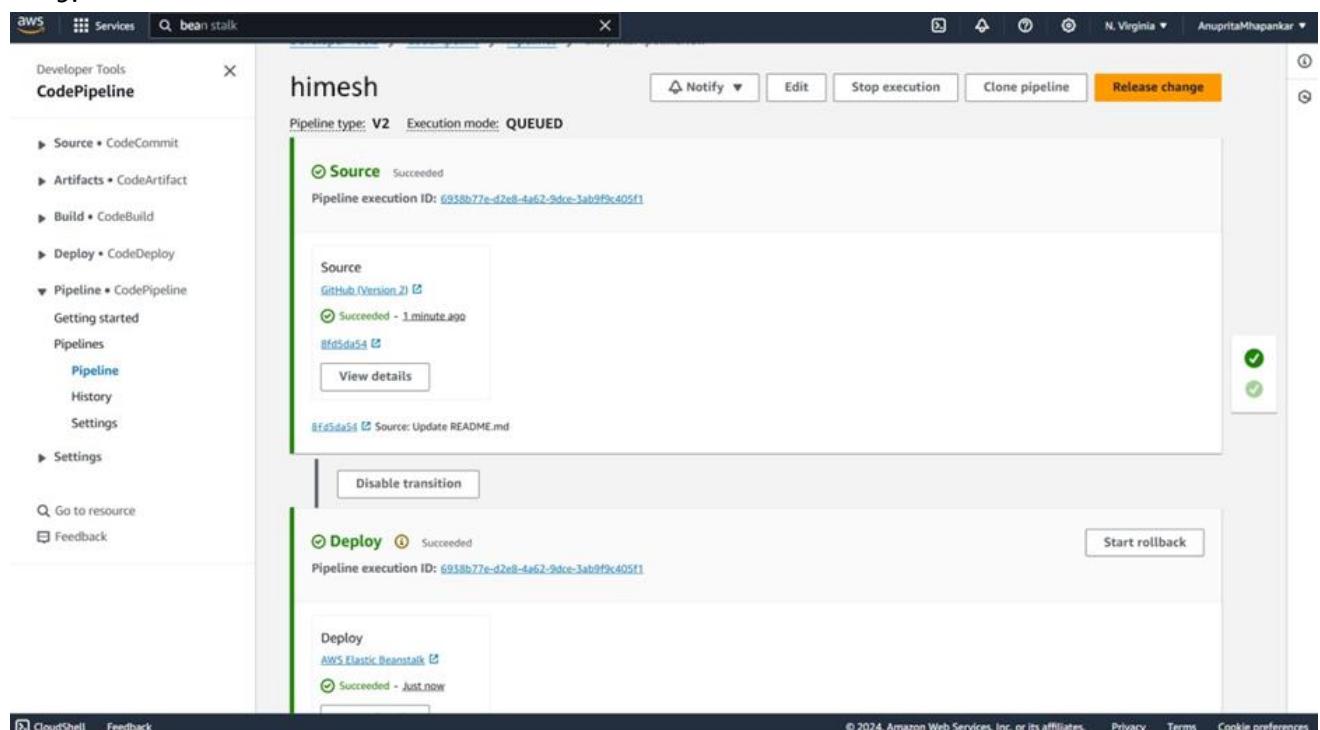
Cancel Previous Next

7. Review the settings and click on create pipeline

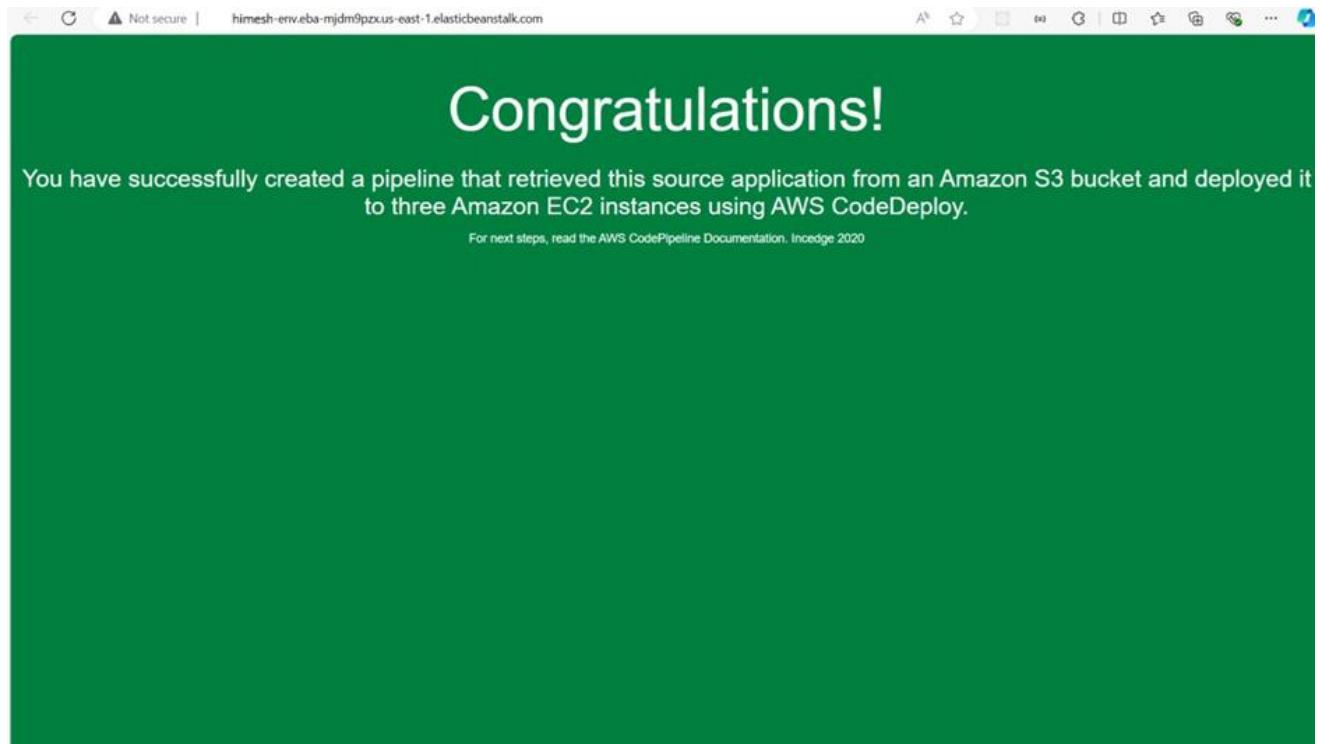


8. Check the URL provided in the EBS environment.

9.



9. The website is hosted from the forked repo in our beanstalk environment



10. Now, Edit index.html file and then commit the changes

Commit changes ×

---

Commit message

Update index.html

Extended description

Add an optional extended description..

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

---

Cancel Commit changes

11. Visit the deployed link again, the changes will be reflected in the website.

The screenshot shows a web browser window with the URL `himesh-env.eba-mjdm9pzx.us-east-1.elasticbeanstalk.com`. The page is a modified version of the Flipkart homepage. At the top, there is a navigation bar with icons for back, forward, search, and other browser functions. The main header features the **Flipkart** logo with its signature yellow 'f' icon. Below the header is a search bar with the placeholder "Search..." and a red "Search" button. The main content area is divided into sections: "Our Products" and "Our Services". The "Our Products" section contains four cards, each representing a product: Product 1 (laptop), Product 2 (iPhone), Product 3 (headphones), and Product 4 (keyboard and mouse). Each card includes a small image, the product name, a brief description, the price (\$19.99, \$29.99, \$39.99, or \$49.99), and a blue "Add to Cart" button. The "Our Services" section lists "Online Retail" and "Flipkart Plus Membership". The overall layout is clean and follows the familiar design of the original Flipkart site.

Name : Himesh Pathai  
Div : D10A  
Roll No. : 35

## Advanced DevOps Lab Experiment:3

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

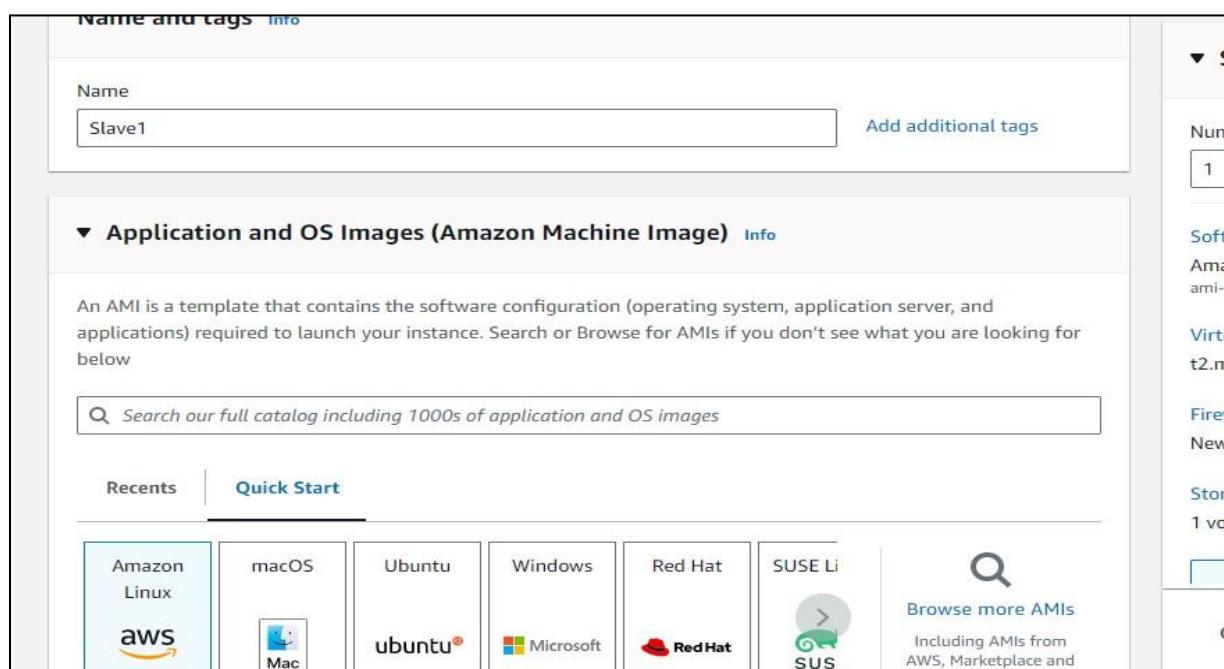
### Theory:

To understand Kubernetes Cluster Architecture and how to install and spin up a Kubernetes cluster on Linux machines or cloud platforms, it's essential to grasp the fundamental components and design principles of Kubernetes.

### Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

(Name 1 as Master, the other 2 as Slave1 and Slave2)



1. Created a master and 2 slaves:

Instances (3) <a href="#">Info</a>		Last updated	less than a minute ago	<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance state ▾</a>	<a href="#">Actions ▾</a>	<a href="#">Launch instances</a>	<a href="#">▼</a>
		<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states ▾			
<input type="checkbox"/>	Name <a href="#">✎</a>	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon		
<input type="checkbox"/>	Master <a href="#">✎</a>	i-02d0bd51d43449e29	<span>Running</span> <a href="#">?</a> <a href="#">?</a>	t2.medium	<span>2/2 checks passed</span> <a href="#">View alarms</a> +	<a href="#">View alarms</a> +	us-east-1b		
<input type="checkbox"/>	Slave2	i-07acb8c5081bec929	<span>Running</span> <a href="#">?</a> <a href="#">?</a>	t2.medium	<span>2/2 checks passed</span> <a href="#">View alarms</a> +	<a href="#">View alarms</a> +	us-east-1b		
<input type="checkbox"/>	Slave1	i-0fb88878237380e6	<span>Running</span> <a href="#">?</a> <a href="#">?</a>	t2.medium	<span>2/2 checks passed</span> <a href="#">View alarms</a> +	<a href="#">View alarms</a> +	us-east-1b		

2. Now click on connect to instance, then click on SSH client.

3. Now copy the ssh from the example and paste it on command prompt.(I used gitbash)

EC2 > Instances > i-02d0bd51d43449e29 > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-02d0bd51d43449e29 (Master) using any of these options

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

Instance ID  
 i-02d0bd51d43449e29 (Master)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is kubernetes.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "kubernetes.pem"  
4. Connect to your instance using its Public DNS:  
 ec2-54-164-13-87.compute-1.amazonaws.com

Example:  
 ssh -i "kubernetes.pem" ubuntu@ec2-54-164-13-87.compute-1.amazonaws.com

## Commands:

4. Now since you are on GitBash, first type sudo su to perform the command as a root user.

## 5. After this type on all 3 machines

Yum install docker -y

```
[ec2-user@ip-172-31-84-37 ~]$ sudo su
[root@ip-172-31-84-37 ec2-user]# yum install docker -y
Last metadata expiration check: 0:18:22 ago on Thu Aug 29 08:52:52 2024.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size   |
| ======           | ======       | ======       | ======     | ====== |
| Installing:      |              |              |            |        |
| docker           | x86_64       | 25.0.6-1.amzn2023.0.1 | amazonlinux | 44 M  |
| Installing dependencies: |          |              |            |        |
| containerd        | x86_64       | 1.7.20-1.amzn2023.0.1 | amazonlinux | 35 M  |
| iptables-libc     | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 401 k |
| iptables-nft      | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
| libcgroup         | x86_64       | 3.0-1.amzn2023.0.1  | amazonlinux | 75 k  |
| libnetfilter_conntrack | x86_64       | 1.0.8-2.amzn2023.0.2 | amazonlinux | 58 k  |
| libnftnl          | x86_64       | 1.0.1-19.amzn2023.0.2 | amazonlinux | 30 k  |
| libnftnl          | x86_64       | 1.2.2-2.amzn2023.0.2 | amazonlinux | 84 k  |
| pigz              | x86_64       | 2.5-1.amzn2023.0.3  | amazonlinux | 83 k  |
| runc              | x86_64       | 1.1.11-1.amzn2023.0.1 | amazonlinux | 3.0 M |
```

```
Running scriptlet: docker-25.0.6-1.amzn2023.0.1.x86_64
Installing : docker-25.0.6-1.amzn2023.0.1.x86_64
Running scriptlet: docker-25.0.6-1.amzn2023.0.1.x86_64
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64
Verifying : docker-25.0.6-1.amzn2023.0.1.x86_64
Verifying : iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64
Verifying : runc-1.1.11-1.amzn2023.0.1.x86_64

Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64      docker-25.0.6-1.amzn2023.0.1.x86_64      iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgroup-3.0-1.amzn2023.0.1.x86_64      libnetfilter_conntrack-1.0.8-2.amzn2023.0.
libnftnl-1.0.1-19.amzn2023.0.2.x86_64        libnftnl-1.2.2-2.amzn2023.0.2.x86_64      pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
```

6. To start the docker on master and slave perform this command: Systemctl start docker

## Extra

7. To check if docker is Installed successfully:

Docker -v or Docker --version

```
[root@ip-172-31-84-37 ec2-user]# systemctl start docker
[root@ip-172-31-84-37 ec2-user]# sudo su
[root@ip-172-31-84-37 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-84-37 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
```

## 8. Now to install kubeadm on master and slaves :

Installing kubeadm:

Go the official documentation off kubeadm.

The screenshot shows the Kubernetes Documentation website with the 'Documentation' tab selected. The main content area is titled 'Installing kubeadm'. It includes a brief introduction, a sidebar with navigation links, and a 'Before you begin' section. A 'kubeadm' logo is visible on the right.

Kubernetes Documentation / Getting started / Production environment  
/ Installing Kubernetes with deployment tools / Bootstrapping clusters with kubeadm  
/ Installing kubeadm

## Installing kubeadm

This page shows how to install the `kubeadm` toolbox. For information on how to create a cluster with kubeadm once you have performed this installation process, see the [Creating a cluster with kubeadm](#) page.

This installation guide is for Kubernetes v1.31. If you want to use a different Kubernetes version, please refer to the following pages instead:

- [Installing kubeadm \(Kubernetes v1.30\)](#)
- [Installing kubeadm \(Kubernetes v1.29\)](#)
- [Installing kubeadm \(Kubernetes v1.28\)](#)
- [Installing kubeadm \(Kubernetes v1.27\)](#)

**Before you begin**

## 9. Scroll down and select Red Hat based distributions:

The screenshot shows the 'Red Hat-based distributions' section of the documentation. It includes a 'Without a package manager' section and a step-by-step guide for setting SELinux to permissive mode.

Debian-based distributions      Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

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

10. Now copy the command on all 3 machines:

Set SELinux to permissive mode:

These instructions are for Kubernetes 1.31.

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

11. Now copy all the commands on the GitBash on all the 3 machines:

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo cat
<<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes] name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/ enabled=1 gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

#Install kubelet, kubeadm and kubectl: sudo yum install -y kubelet kubeadm

kubectl --disableexcludes=kubernetes #(Optional) Enable the kubelet service

before running kubeadm:

sudo systemctl enable --now kubelet

```
Installing : kubelet-1.31.0-150500.1.1.x86_64
Running scriptlet: kubectl-1.31.0-150500.1.1.x86_64
Verifying  : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Verifying  : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
Verifying  : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
Verifying  : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
Verifying  : socat-1.7.4.2-1.amzn2023.0.2.x86_64
Verifying  : cri-tools-1.31.1-150500.1.1.x86_64
Verifying  : kubeadm-1.31.0-150500.1.1.x86_64
Verifying  : kubectl-1.31.0-150500.1.1.x86_64
Verifying  : kubelet-1.31.0-150500.1.1.x86_64
Verifying  : kubernetes-cni-1.5.0-150500.2.1.x86_64

Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
kubeadm-1.31.0-150500.1.1.x86_64
kubelet-1.31.0-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-84-37 ec2-user]# sudo systemctl enable --now kubelet
```

## 12. Type yum repolist to check the repository of kubernetes

```
[root@ip-172-31-84-143 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                            Amazon Linux 2023 repository
kernel-livepatch                        Amazon Linux 2023 Kernel Livepatch repository
kubernetes                             Kubernetes
```

## EXTRA

### Got an error in initialization kubeadm

```
[root@ip-172-31-31-240 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0908 11:25:45.820964    2320 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": desc = connection error: desc = "transport: Error while dialing: dial unix /var/run/containerd/containerd.sock: [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
[ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...` to see the stack trace of this error execute with --v=5 or higher
```

### Error was resolved:

(after again starting from scratch)

## 13. Initialize the kubeadm by the command kubeadm init only on master:

Kubeadm initialized successfully:

```
[root@ip-172-31-26-66 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
      [WARNING FileExisting-socat]: socat not found in system path
      [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0912 06:07:49.475553    28037 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.10" as the default image for kubelet
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-26-66.ec2.internal.cluster.local] and IPs [10.96.0.1 172.31.26.66]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
```

## 14. After this we will get 3 things:

- The directory
- Some export Statement
- The most important thing - the token to connect the slaves with the master.

15. Copy them one by one and paste it on the slaves:

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

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

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

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

You should now deploy a pod network to the cluster.

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

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

```
kubeadm join 172.31.26.66:6443 --token grw4r4.gb3kkhb7392dnvjp \  
--discovery-token-ca-cert-hash sha256:b61f1de7eedb2c0dc0cc237d4629e9631920b63dd6634c3e22e76aaa36d01920
```

16. After pasting type kubectl get nodes:

The nodes are connected successfully:

```
ubuntu@ip-172-31-17-23:~$ kubectl get nodes  
NAME     STATUS   ROLES      AGE     VERSION  
ip-172-31-17-23   Ready    control-plane   3m56s   v1.29.0  
ip-172-31-18-12   Ready    <none>       37s    v1.29.0  
ip-172-31-26-153   Ready    <none>       24s    v1.29.0  
ubuntu@ip-172-31-17-23:~$ kubectl get nodes  
NAME     STATUS   ROLES      AGE     VERSION  
ip-172-31-17-23   Ready    control-plane   9m34s   v1.29.0  
ip-172-31-18-12   Ready    <none>       6m15s   v1.29.0  
ip-172-31-26-153   Ready    <none>       6m2s    v1.29.0  
ubuntu@ip-172-31-17-23:~$ |
```

Name : Himesh Pathai  
Div : D10A  
Roll No. : 35

## Advanced DevOps Lab

### Experiment 4

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

#### Theory:

Kubernetes, often referred to as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Originally developed by Google, it has become the industry standard for managing container workloads due to its flexibility and robust features.

#### Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

(Name 1 as Master, the other 2 as Slave1 and Slave2)

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function. In the 'Name and tags' section, the name 'Slave1' is entered. In the 'Application and OS Images (Amazon Machine Image)' section, the 'Quick Start' tab is selected, showing options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux Enterprise Server (SLES). A search bar is available to find specific AMIs. On the right side of the modal, there are detailed settings for memory, timeout, and environment variables, along with a 'Create Function' button at the bottom.

2. Now click on connect to instance, then click on SSH client.

3. Now copy the ssh from the example and paste it on command prompt.(I used gitbash)

EC2 > Instances > i-02d0bd51d43449e29 > Connect to instance

## Connect to instance Info

Connect to your instance i-02d0bd51d43449e29 (Master) using any of these options

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

Instance ID  
i-02d0bd51d43449e29 (Master)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is kubernetes.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "kubernetes.pem"
4. Connect to your instance using its Public DNS:  
ec2-54-164-13-87.compute-1.amazonaws.com

Example:

ssh -i "kubernetes.pem" ubuntu@ec2-54-164-13-87.compute-1.amazonaws.com

```
Running scriptlet: docker-25.0.6-1.amzn2023.0.1.x86_64
Installing : docker-25.0.6-1.amzn2023.0.1.x86_64
Running scriptlet: docker-25.0.6-1.amzn2023.0.1.x86_64
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64
Verifying : docker-25.0.6-1.amzn2023.0.1.x86_64
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64
Verifying : runc-1.1.11-1.amzn2023.0.1.x86_64

Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64           docker-25.0.6-1.amzn2023.0.1.x86_64           iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64        libcgroup-3.0-1.amzn2023.0.1.x86_64         libnetfilter_conntrack-1.0.8-2.amzn2023.0.
libnftnl-1.0.1-19.amzn2023.0.2.x86_64          libnftnl-1.2.2-2.amzn2023.0.2.x86_64        pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
```

Commands:

4. Now since you are on GitBash, first type sudo su to perform the command as a root user.

## 5. After this type on GitBash

**Yum install docker -y**

Package	Architecture	Version	Repository	Size
<b>Installing:</b>				
docker	x86_64	25.0.6-1.amzn2023.0.1	amazonlinux	44 M
<b>Installing dependencies:</b>				
containerd	x86_64	1.7.20-1.amzn2023.0.1	amazonlinux	35 M
iptables-lib	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux	401 k
iptables-nft	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux	183 k
libcgroup	x86_64	3.0-1.amzn2023.0.1	amazonlinux	75 k
libnetfilter_conntrack	x86_64	1.0.8-2.amzn2023.0.2	amazonlinux	58 k
libnftnl	x86_64	1.0.1-19.amzn2023.0.2	amazonlinux	30 k
libnfntnl	x86_64	1.2.2-2.amzn2023.0.2	amazonlinux	84 k
pigz	x86_64	2.5-1.amzn2023.0.3	amazonlinux	83 k
runc	x86_64	1.1.11-1.amzn2023.0.1	amazonlinux	3.0 M

6. To start the docker perform this command: Systemctl start docker

7. To check if docker is Installed successfully:

Docker -v or Docker --version

[root@ip-172-31-84-37 ec2-user]# systemctl start docker	
[root@ip-172-31-84-37 ec2-user]# sudo su	
[root@ip-172-31-84-37 ec2-user]# yum repolist	
repo id	repo name
amazonlinux	Amazon Linux 2023 repository
kernel-livepatch	Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-84-37 ec2-user]# docker --version	
Docker version 25.0.5, build 5dc9bcc	

8. Now to install kubeadm :

Installing kubeadm:

Go the official documentation off kubeadm.

The screenshot shows the Kubernetes Documentation website with the URL <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>. The page title is "Installing kubeadm". It includes a sidebar with navigation links like Documentation, Getting started, and Production environment. The main content area provides instructions for installing kubeadm, mentioning it's for Kubernetes v1.31. It lists several sub-sections under "Before you begin" and "After you begin". A "kubeadm" logo is present on the right.

9. Scroll down and select Red Hat based distributions:

Debian-based distributions

Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

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

10. Now copy the command:

Set SELinux to permissive mode:

These instructions are for Kubernetes 1.31.

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

11. Now copy all the commands on the GitBash:

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/ enabled=1 gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF

#Install kubelet, kubeadm and kubectl: sudo yum install -y kubelet

kubeadm kubectl --disableexcludes=kubernetes #(Optional) Enable the

kubelet service before running kubeadm:

sudo systemctl enable --now kubelet
```

```
Installing      : kubeadm-1.31.0-150500.1.1.x86_64
Installing      : kubectl-1.31.0-150500.1.1.x86_64
Running scriptlet: kubectl-1.31.0-150500.1.1.x86_64
Verifying       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Verifying       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
Verifying       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
Verifying       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
Verifying       : socat-1.7.4.2-1.amzn2023.0.2.x86_64
Verifying       : cri-tools-1.31.1-150500.1.1.x86_64
Verifying       : kubeadm-1.31.0-150500.1.1.x86_64
Verifying       : kubectl-1.31.0-150500.1.1.x86_64
Verifying       : kubelet-1.31.0-150500.1.1.x86_64
Verifying       : kubernetes-cni-1.5.0-150500.2.1.x86_64

Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
kubeadm-1.31.0-150500.1.1.x86_64
kubelet-1.31.0-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-84-37 ec2-user]# sudo systemctl enable --now kubelet
```

## 12. Type yum repolist to check the repository of kubernetes

```
[root@ip-172-31-84-143 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                            Amazon Linux 2023 repository
kernel-livepatch                        Amazon Linux 2023 Kernel Livepatch repository
kubernetes                             Kubernetes
```

## EXTRA

### Got an error in initialization kubeadm

```
[root@ip-172-31-31-240 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0908 11:25:45.820964    2320 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": desc = connection error: desc = "transport: Error while dialing: dial unix /var/run/containerd/containerd.sock: [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
[ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...` to see the stack trace of this error execute with --v=5 or higher
```

### Error was resolved:

(after again starting from scratch)

## 13. Initialize the kubeadm by the command kubeadm init :

Kubeadm initialized successfully:

```
[root@ip-172-31-26-66 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
      [WARNING FileExisting-socat]: socat not found in system path
      [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0912 06:07:49.475553    28037 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.10" as the default image for kubelet
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-26-66.ec2.internal svc.cluster.local] and IPs [10.96.0.1 172.31.26.66]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
```

## 14. After this we will get 3 things:

- The directory
- Some export Statement
- The most important thing - the token to connect the slaves with the master.

## 15. Copy them

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

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

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

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

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

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

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

```
kubeadm join 172.31.26.66:6443 --token grw4r4.gb3kkhb7392dnvjp \
--discovery-token-ca-cert-hash sha256:b61f1de7eedb2c0dc0cc237d4629e9631920b63dd6634c3e22e76aaa36d01920
```

## 16. After pasting type kubectl get nodes:

The nodes are connected successfully:

```
ubuntu@ip-172-31-17-23:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-17-23 Ready    control-plane   3m56s   v1.29.0
ip-172-31-18-12 Ready    <none>        37s    v1.29.0
ip-172-31-26-153 Ready    <none>        24s    v1.29.0
ubuntu@ip-172-31-17-23:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-17-23 Ready    control-plane   9m34s   v1.29.0
ip-172-31-18-12 Ready    <none>        6m15s   v1.29.0
ip-172-31-26-153 Ready    <none>        6m2s    v1.29.0
ubuntu@ip-172-31-17-23:~$ |
```

## 17. Create two YAML files named nginx-deployment.yaml and nginx-service.yaml (I used nano editor for the same)

```
ubuntu@ip-172-31-17-23:~$ nano nginx-deployment.yaml
ubuntu@ip-172-31-17-23:~$ nano nginx-service.yaml
```

18. Then add the deployment and service configuration in it, respectively:

Deployment:

```
GNU nano 6.2                                     nginx-deployment.yaml *
name: nginx-deployment
labels:
  app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.3
          ports:
            - containerPort: 80

^G Help      ^O Write Out     ^W Where Is      ^K Cut      ^T Execute      ^C Location
^X Exit      ^R Read File     ^\ Replace       ^U Paste      ^J Justify      ^/ Go To Line
```

Service:

```
GNU nano 6.2                                     nginx-service.yaml *
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer

^G Help      ^O Write Out     ^W Where Is      ^K Cut      ^T Execute      ^C Location
^X Exit      ^R Read File     ^\ Replace       ^U Paste      ^J Justify      ^/ Go To Line
```

19. Now since we have configured our files we would now proceed for applying both the deployment and the service files.

Deployment :

```
ubuntu@ip-172-31-17-23:~$ kubectl apply -f nginx-deployment.yaml  
deployment.apps/nginx-deployment created
```

Service:

```
ubuntu@ip-172-31-17-23:~$ kubectl apply -f nginx-service.yaml  
service/nginx-service created
```

20. After deployment its time for verifying the same:

For deployment:

```
ubuntu@ip-172-31-17-23:~$ kubectl get deployments  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx 1/1 1 1 14m  
nginx-deployment 2/2 2 2 39s
```

For services:

```
ubuntu@ip-172-31-17-23:~$ kubectl get services  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S)  
(S) AGE ClusterIP 10.96.0.1 <none> 443/  
kubernetes 70m TCP 80 10.109.245.143:80  
nginx 37m NodePort 10.99.247.105:80<pending> 80:3  
0306/TCP 36s nginx-service LoadBalancer 10.99.247.105:80<pending> 80:3  
1130/TCP 36s
```

For pods:

```
ubuntu@ip-172-31-17-23:~$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
nginx-7854ff8877-mxrqg 1/1 Running 0 15  
nginx-deployment-6b4d6fdbf-5rb6h 1/1 Running 0 65  
nginx-deployment-6b4d6fdbf-6q2jj 1/1 Running 0 65
```

**Extra:**

```
ubuntu@ip-172-31-17-23:~$ kubectl get namespaces
NAME        STATUS   AGE
default     Active   55m
kube-node-lease Active   55m
kube-public  Active   55m
kube-system  Active   55m
```

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

```
ubuntu@ip-172-31-17-23:~$ kubectl port-forward service/nginx 8080:
80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

22. You can open the browser and check on

<http://localhost:8080>

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

**Name : Himesh Pathai**

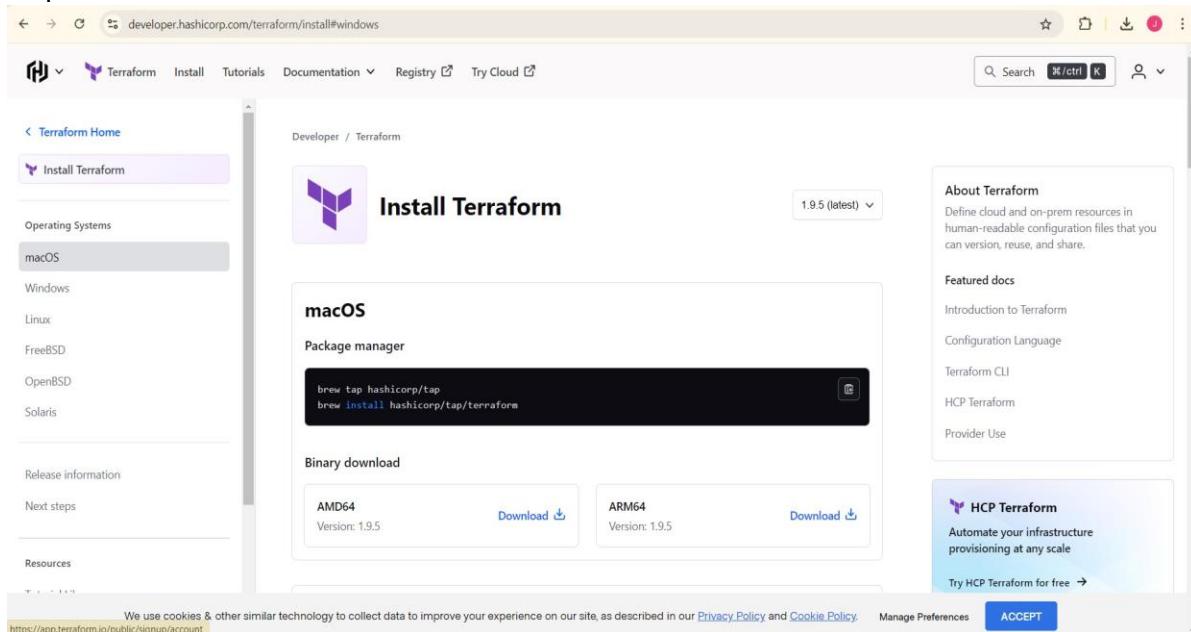
**Roll No. : 35**

**Div. : D15A**

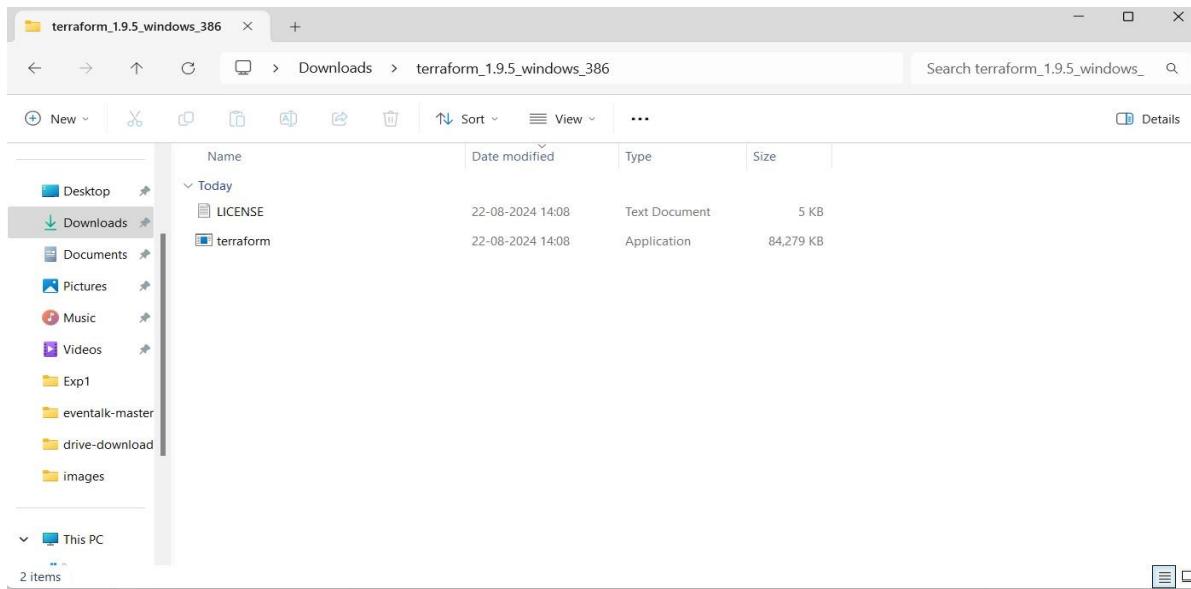
### **Advance DevOps Exp : 5**

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

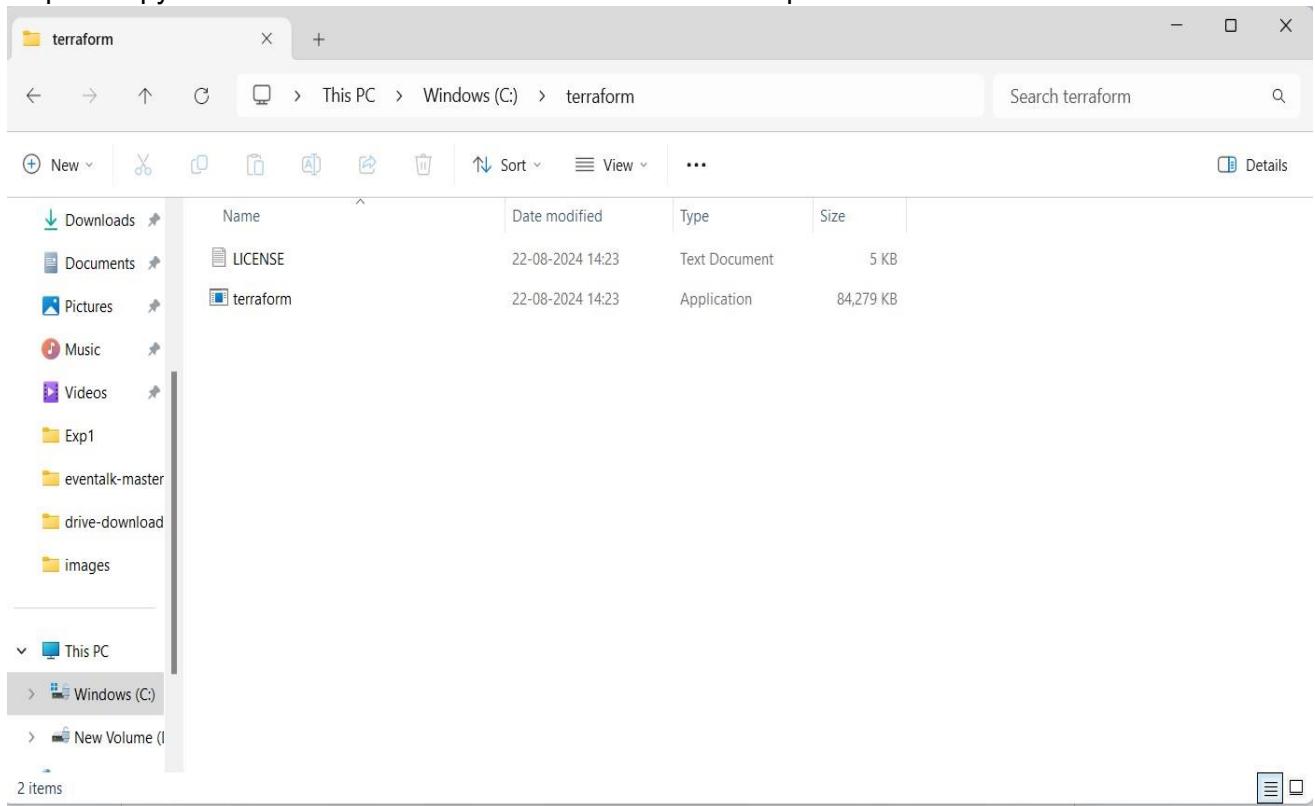
**Step1: Download Terraform from the official website**



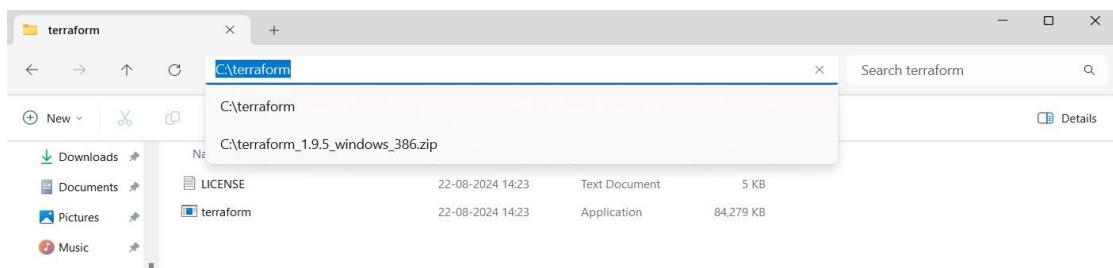
The screenshot shows the HashiCorp Terraform website at developer.hashicorp.com/terraform/install#windows. The left sidebar has a 'macOS' tab selected under 'Operating Systems'. The main content area is titled 'Install Terraform' and shows instructions for macOS. It includes a 'Package manager' section with a terminal command: 'brew tap hashicorp/tap' and 'brew install hashicorp/tap/terraform'. Below that is a 'Binary download' section with links for 'AMD64' and 'ARM64' versions. A sidebar on the right contains sections like 'About Terraform', 'Featured docs', and 'HCP Terraform'.



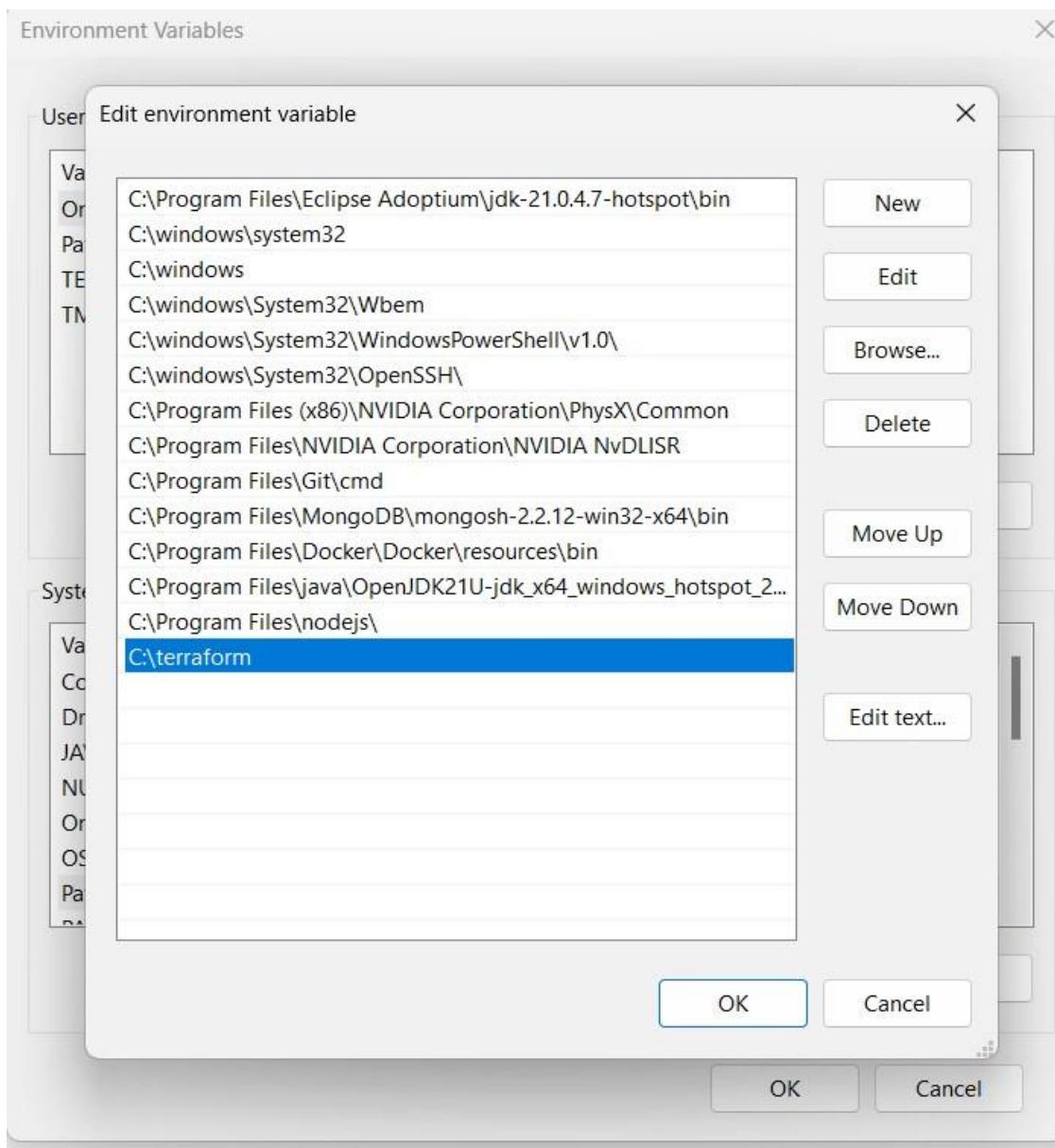
## Step 2: Copy and extract Terraform from the downloads and paste it in the C drive



## Step 3: Copy the file path to paste in the environment variables



#### Step 4: Set the environment variables for terraform



#### Step 5: Check whether the terraform is installed

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\himes>terraform --version
Terraform v1.9.5
on windows_386

C:\Users\himes>
```

**Name : Himesh Pathai**

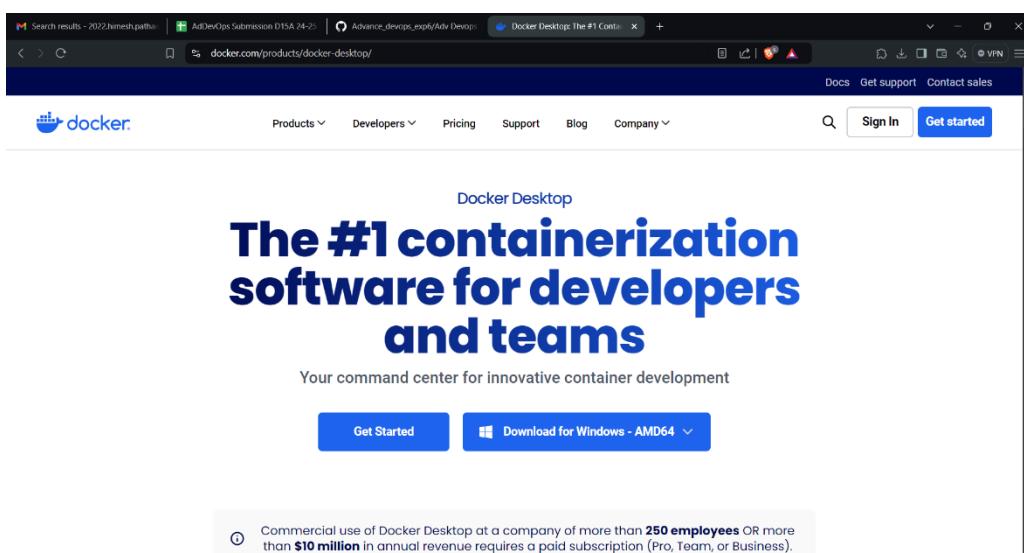
**Roll No. : 35**

**Div. : D15A**

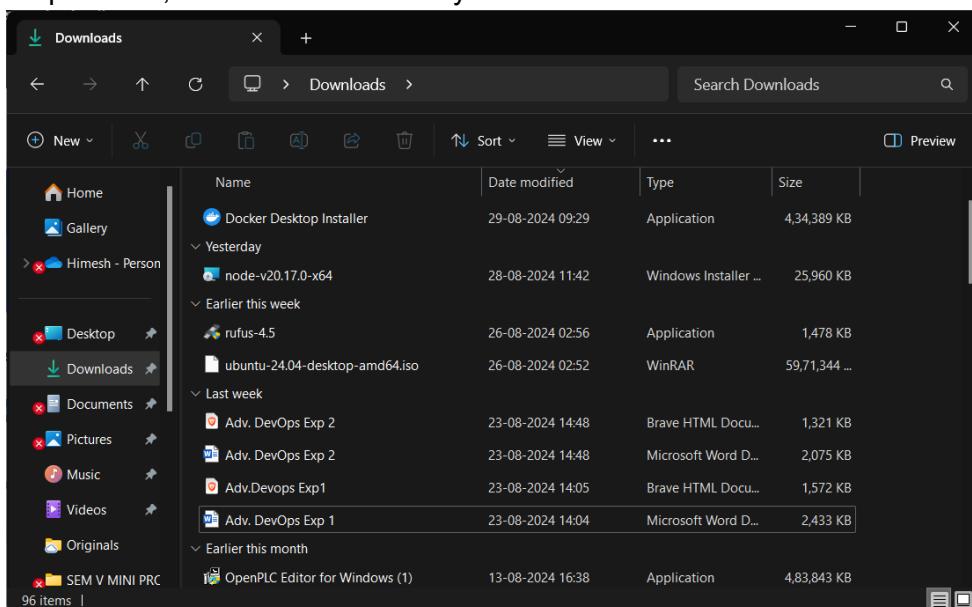
## **Advance DevOps Exp : 6**

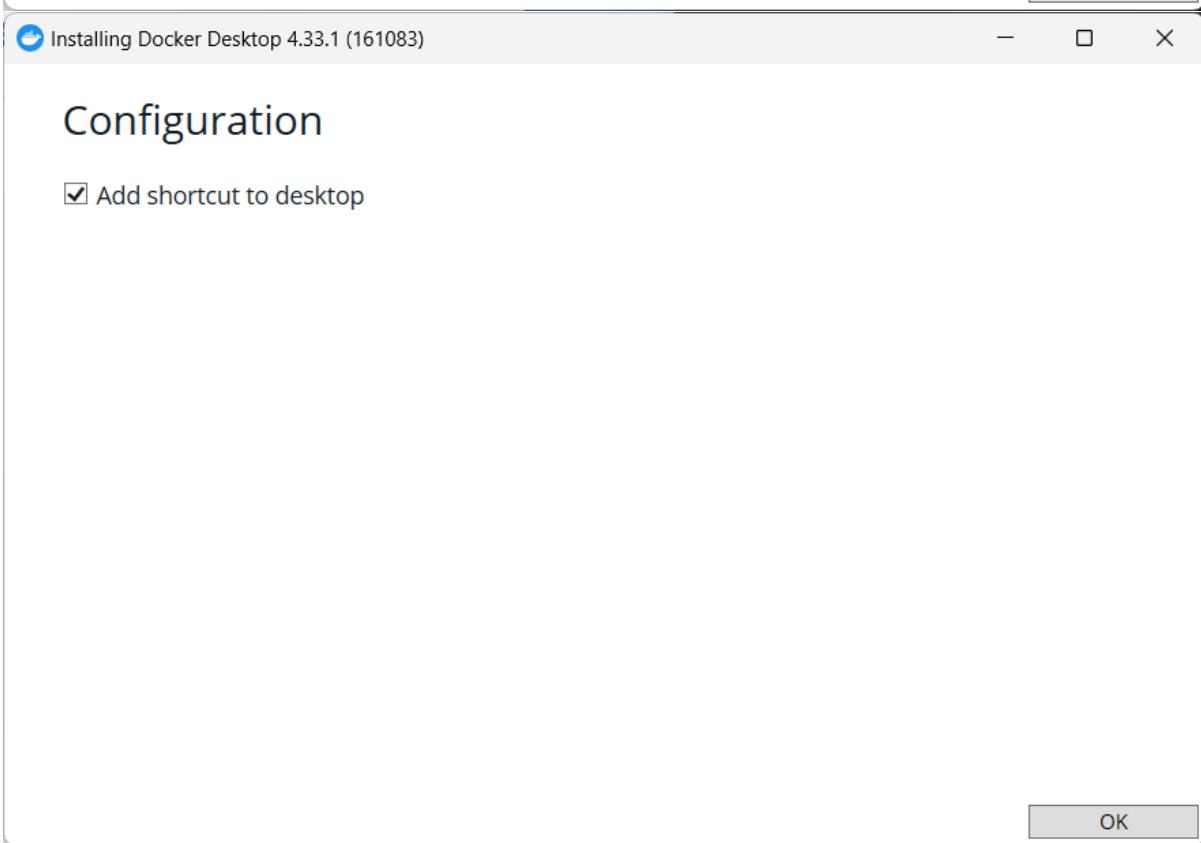
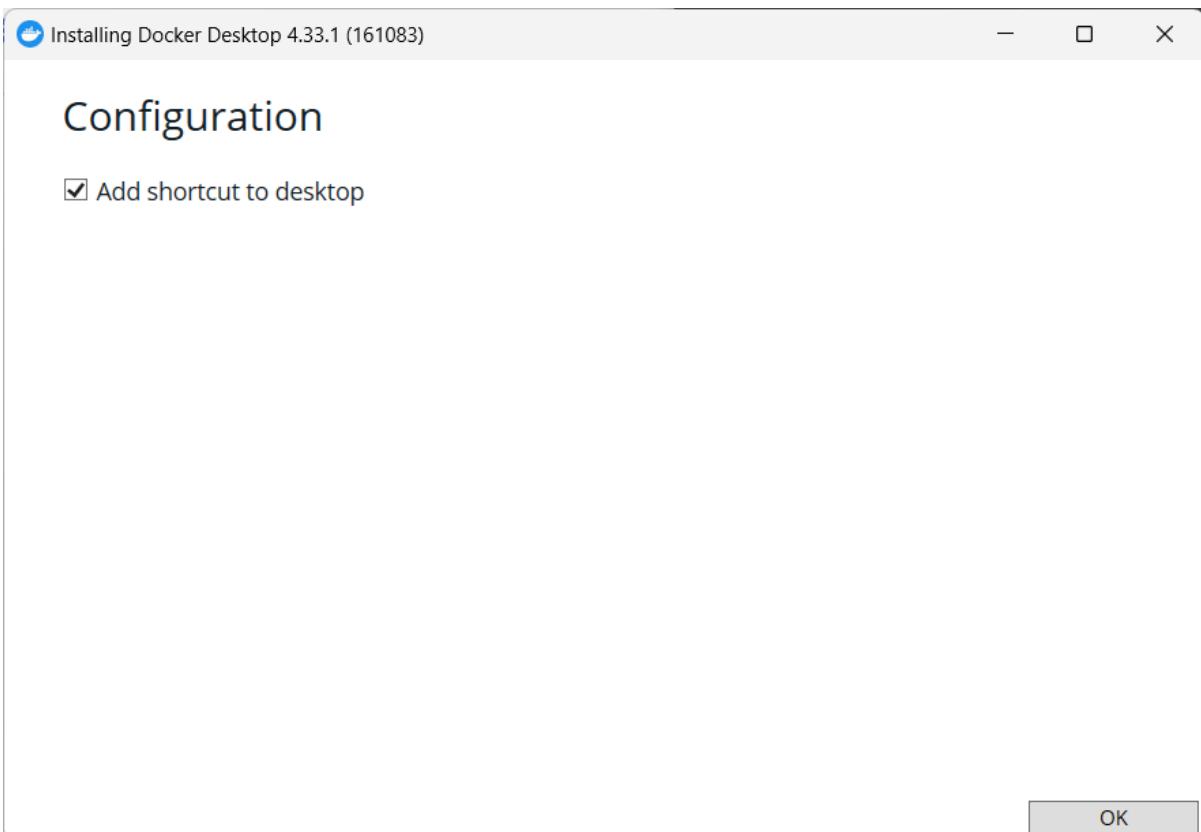
**Aim:** To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker)

Step 1: Download Docker from [www.docker.com](https://www.docker.com)

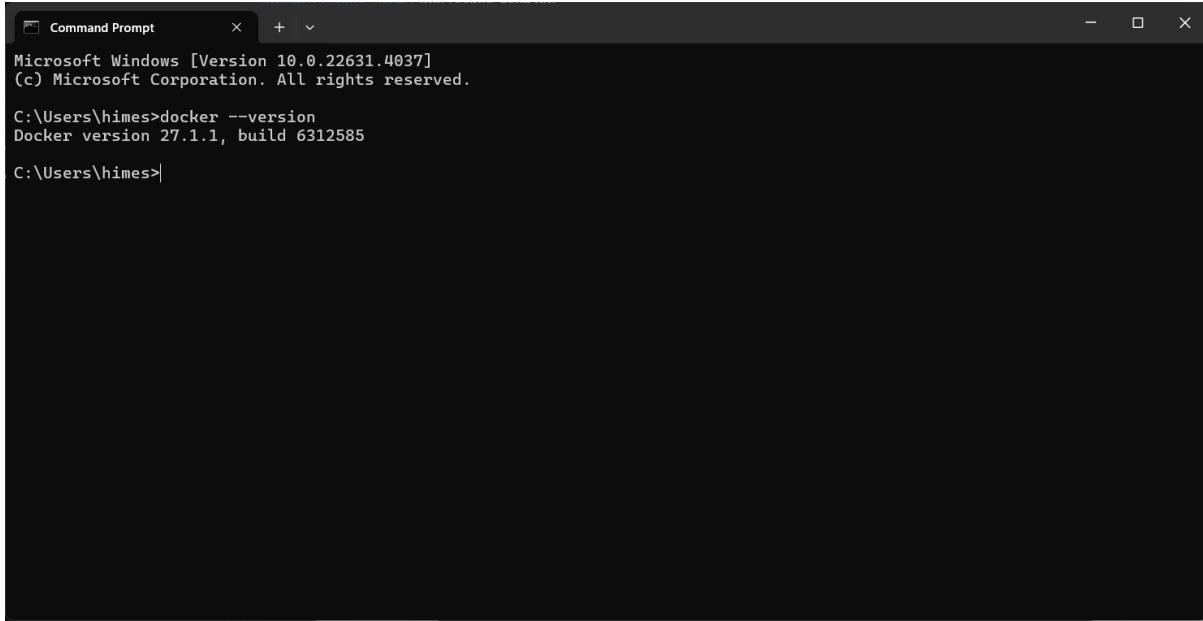


Step 2: Now, Docker is successfully downloaded.





Step 3: Open Command Prompt and enter the command docker –version, to check whether the docker is successfully installed.

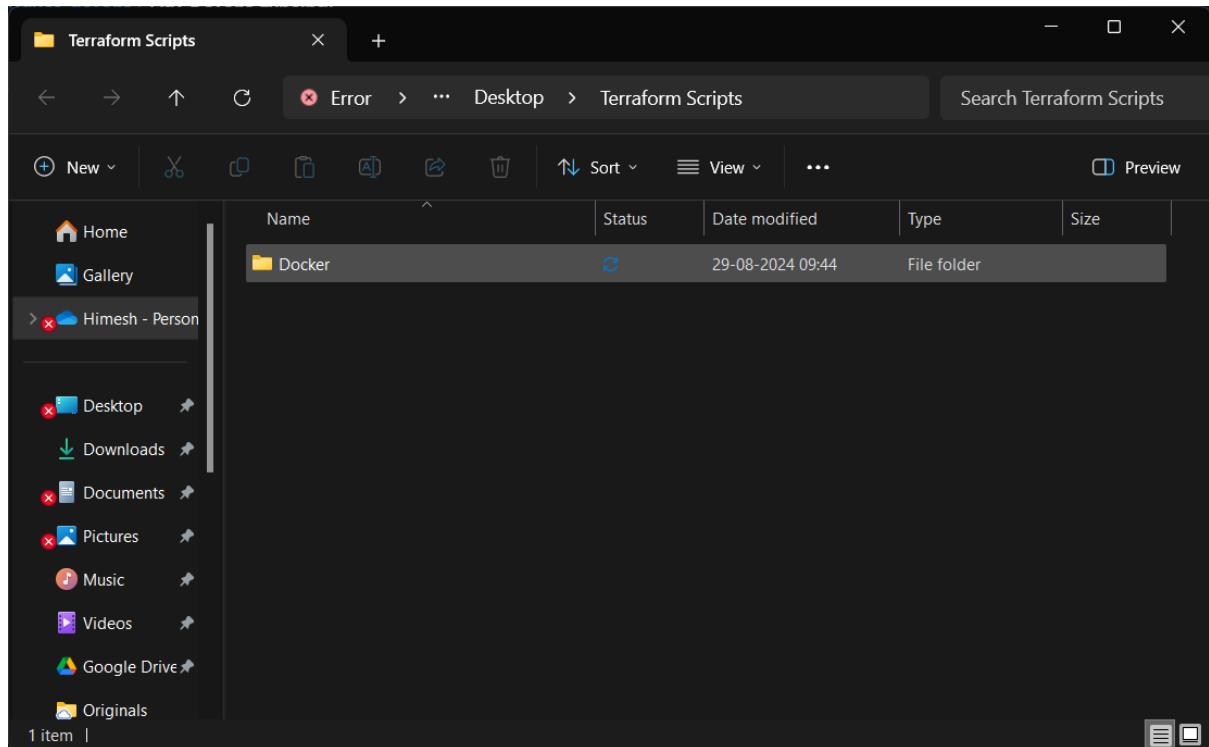


```
Command Prompt
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

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

C:\Users\himes>
```

Step 4: Create a folder Terraform\_scripts and inside it create a folder named Docker.



Step 5: create a new folder named ‘Terraform’ in the ‘TerraformScripts’ folder. Then create a new terraform\_script.tf file using vs code.

Run the following script in the VS Code.

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

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has a 'DOCKER' section expanded, showing a file named 'terraform\_script.tf'. The main editor tab is also titled 'terraform\_script.tf'. The code in the editor is:

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

The status bar at the bottom shows 'Ln 25, Col 1' and other settings like 'Spaces: 4', 'UTF-8', 'CRLF', 'Plain Text', and 'Go Live'.

Step 6: Open Windows Explorer and run the following command `terraform init`, `terraform plan`, `terraform apply`, `terraform destroy`, `terraform provider`, `terraform validate`, `terraform state list` and `docker images`.

The screenshot shows a Windows Command Prompt window titled 'Command Prompt'. The command `cd docker` is entered, followed by `terraform init`. The output shows the initialization process, including finding and installing the 'docker' provider version 2.21.0, creating a lock file, and successfully initializing Terraform. It also provides instructions for subsequent commands and provider signing.

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts>cd docker  
C:\Users\himes\OneDrive\Desktop\TerraformScripts\docker>terraform init  
Initializing the backend...  
Initializing provider plugins...  
- Finding kreuzwerker/docker versions matching "2.21.0"...  
- Installing kreuzwerker/docker v2.21.0...  
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)  
Partner and community providers are signed by their developers.  
If you'd like to know more about provider signing, you can read about it here:  
https://www.terraform.io/docs/cli/plugins/signing.html  
Terraform has created a lock file .terraform.lock.hcl to record the provider  
selections it made above. Include this file in your version control repository  
so that Terraform can guarantee to make the same selections by default when  
you run "terraform init" in the future.  
  
Terraform has been successfully initialized!  
  
You may now begin working with Terraform. Try running "terraform plan" to see  
any changes that are required for your infrastructure. All Terraform commands  
should now work.  
  
If you ever set or change modules or backend configuration for Terraform,  
rerun this command to reinitialize your working directory. If you forget, other  
commands will detect it and remind you to do so if necessary.  
C:\Users\himes\OneDrive\Desktop\TerraformScripts\docker>
```

```
Command Prompt - + ▾
C:\Users\himes\OneDrive\Desktop\TerraformScripts\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         = [
    + "sleep",
    + "3600",
  ]
  + 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)
}

Command Prompt - + ▾
C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>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         = [
    + "sleep",
    + "3600",
  ]
  + 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
}
```

```
Command Prompt + ▾
+-----+
+ read_only      = false
+ remove_volumes = true
+ restart        = "no"
+ rm             = false
+ runtime         = (known after apply)
+ security_opts  = (known after apply)
+ shm_size       = (known after apply)
+ start          = true
+ stdin_open     = false
+ stop_signal    = (known after apply)
+ stop_timeout   = (known after apply)
+ tty             = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

[Output from right terminal window]
C:\Users\micha\Downloads\Terraform\Ubuntu>terraform init
Terraform uses the selected providers to generate the configuration
for this workspace. To inspect, edit, or refresh this configuration at any time,
use the "terraform" command on the command-line.

Terraform will probe the following actions:
+ 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)
}

Terraform successfully initialized. You may now begin working with
your Terraform configuration files. Type "terraform apply" to see
how Terraform will transform your infrastructure, or "terraform
plan" to validate your configuration without changing state.

[Output from left terminal window]
Command Prompt + ▾
+-----+
+ read_only      = false
+ remove_volumes = true
+ restart        = "no"
+ rm             = false
+ runtime         = (known after apply)
+ security_opts  = (known after apply)
+ shm_size       = (known after apply)
+ start          = true
+ stdin_open     = false
+ stop_signal    = (known after apply)
+ stop_timeout   = (known after apply)
+ tty             = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

```
Command Prompt × + ▾ - □ ×

}

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

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Creation complete after 15s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598ubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=a5c5d23b0cb0e6c71d8b5ecdc395a48bc3fdaf608fc3058df946209f0886952b]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>
```

```
Command Prompt × + ▾ - □ ×

C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>terraform providers

Providers required by configuration:
└── provider[registry.terraform.io/kreuzwerker/docker] 2.21.0

Providers required by state:
provider[registry.terraform.io/kreuzwerker/docker]

C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>terraform validate
Success! The configuration is valid.
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>terraform state list
docker_container.foo
docker_image.ubuntu
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts\Docker>SS|
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   edbfe74c41f8  3 weeks ago  78.1MB
```

```
C:\Users\himes\OneDrive\Desktop\TerraformScripts>S|
```

# ADVANCE DEVOPS EXP 7

Name : **Himesh Pathai**  
Class : **D15A**  
Roll No. : **35**

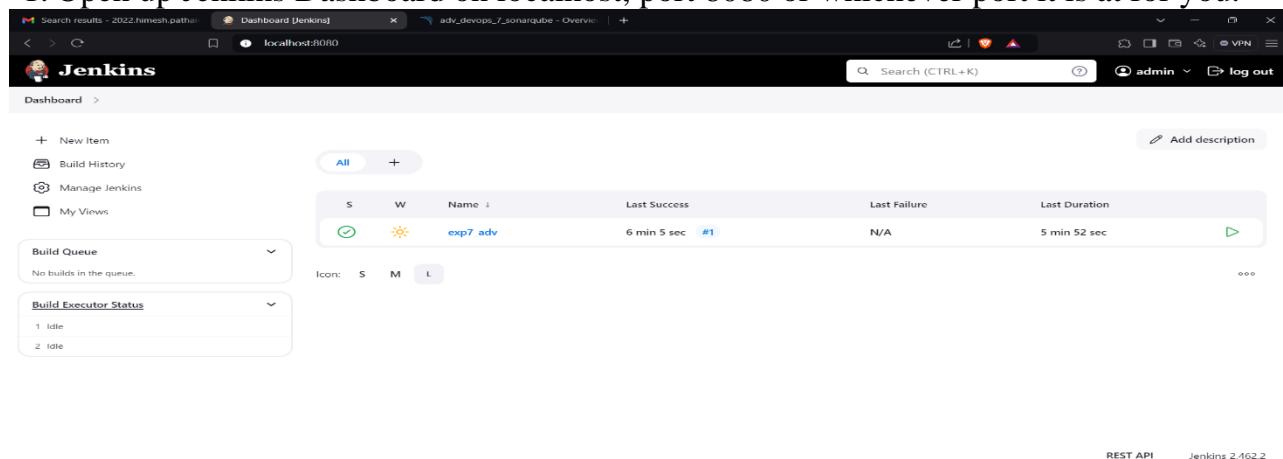
**Aim:** To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

## Integrating Jenkins with SonarQube:

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

## Steps to integrate Jenkins with SonarQube

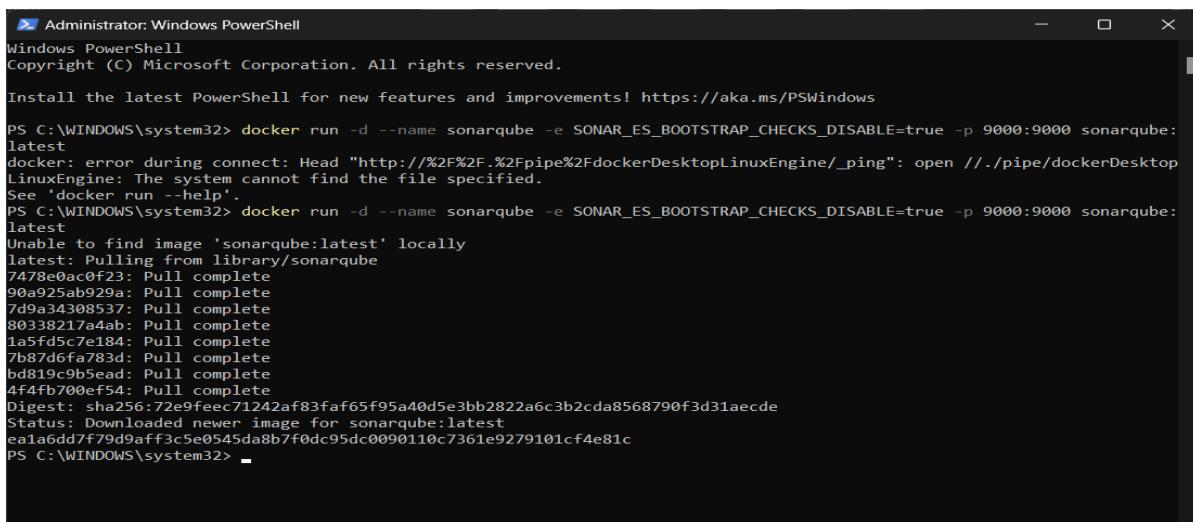
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



The screenshot shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The main area displays the 'exp7\_adv' project, which has a status of 'Success' (green), last run at '6 min 5 sec', and a build number of '#1'. Below the project details, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 idle, 2 idle). On the left sidebar, there are links for '+ New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The top right corner shows the user 'admin' and a 'log out' link. The bottom right corner indicates the version 'Jenkins 2.162.2'.

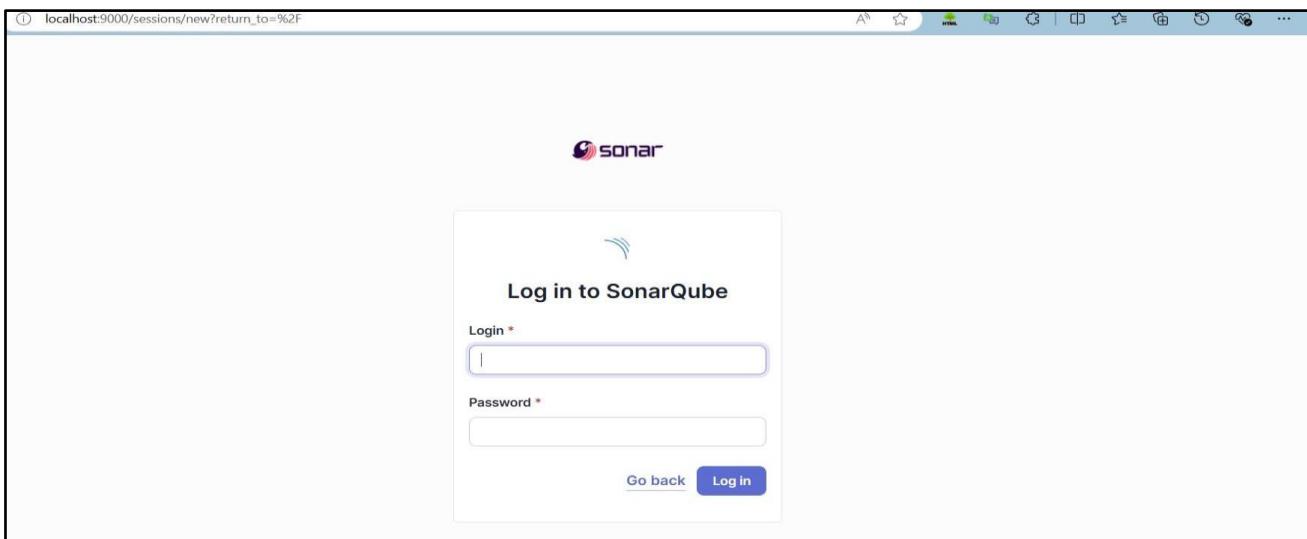
2. Run SonarQube in a Docker container using this command -

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



```
Administrator: Windows PowerShell  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
PS C:\WINDOWS\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:  
latest  
docker: error during connect: Head "http://%2F%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktop  
LinuxEngine: The system cannot find the file specified.  
See 'docker run --help'.  
PS C:\WINDOWS\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:  
latest  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
7478e0ac0f23: Pull complete  
90a925ab929a: Pull complete  
/d9a34308537: 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  
ea1a6dd7f79d9aff3c5e0545da8b7f0dc95dc0090110c7361e9279101cf4e81c  
PS C:\WINDOWS\system32>
```

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



- Login to SonarQube using username admin and password admin.

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.

Create a local project

- Create a manual project in SonarQube with the name sonarqube

1 of 2

Create a local project

Project display name \*

Project key \*

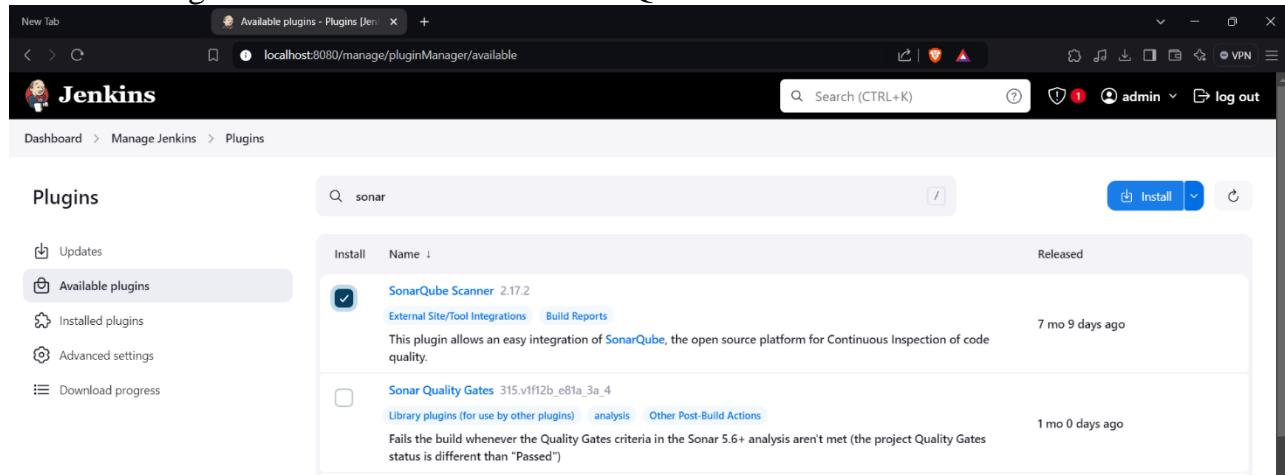
Main branch name \*

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

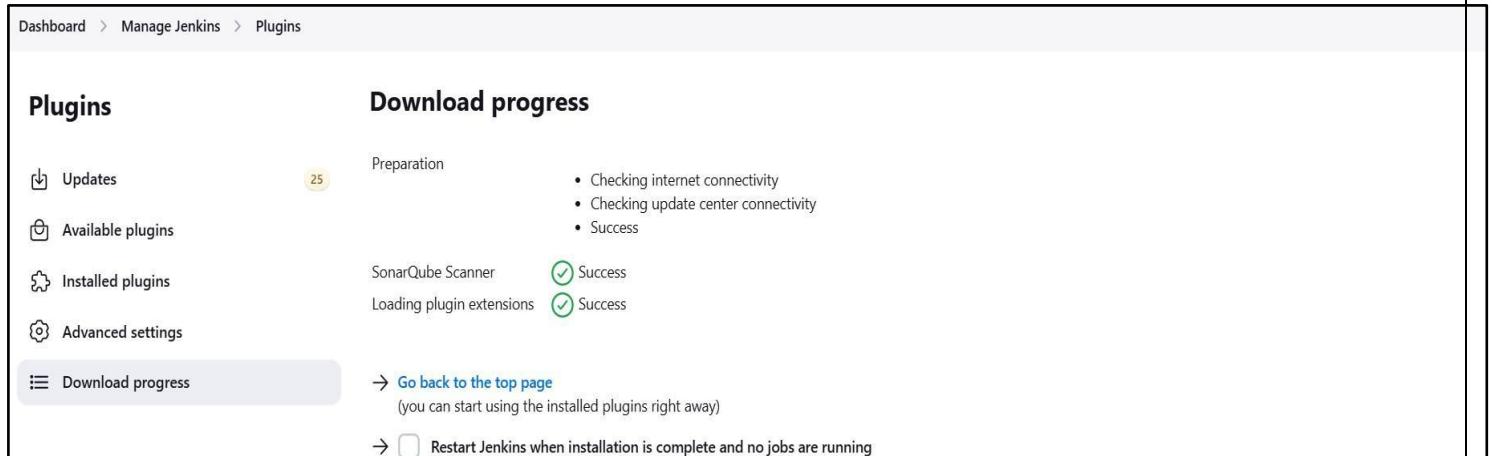
**⚠️ Embedded database should be used for evaluation purposes only**  
The embedded database will not scale, it will not support upgrading to newer versions of :

Setup the project and come back to Jenkins Dashboard.

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



The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text "sonar". Below the search bar, a list of plugins is displayed. The first item in the list is "SonarQube Scanner 2.17.2", which is checked for installation. The description for this plugin states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." To the right of the plugin list, there is a "Released" column and a timestamp "7 mo 9 days ago". Below the main list, another plugin, "Sonar Quality Gates 315.vlf12b\_e81a\_3a\_4", is listed with a note: "Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than 'Passed')". The timestamp for this plugin is "1 mo 0 days ago". At the bottom right of the plugin manager interface, there is a blue "Install" button.



The screenshot shows the Jenkins download progress page. On the left, a sidebar menu includes "Updates", "Available plugins" (which is selected), "Installed plugins", "Advanced settings", and "Download progress" (which is highlighted). The main content area is titled "Download progress" and shows the status of the SonarQube Scanner plugin. It indicates "Preparation" with three bullet points: "• Checking internet connectivity", "• Checking update center connectivity", and "• Success". Below this, it shows "SonarQube Scanner" with a green checkmark and the word "Success". Further down, it shows "Loading plugin extensions" with a green checkmark and the word "Success". At the bottom of the page, there are two links: "→ Go back to the top page (you can start using the installed plugins right away)" and "→  Restart Jenkins when installation is complete and no jobs are running".

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

and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube>, here we have named it as **adv\_devops\_7\_sonarqube**

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

### SonarQube servers

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

Environment variables

#### SonarQube installations

List of SonarQube installations

##### Name

adv\_devops\_7.sonarqube

##### Server URL

Default is <http://localhost:9000>

<https://localhost:9000>

##### Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

Advanced ▾

7. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. It includes sections for 'Gradle installations', 'SonarScanner for MSBuild installations', 'SonarQube Scanner installations', and 'Ant installations'. Each section has a 'Add [Tool Name]' button.

## Dashboard > Manage Jenkins > Tools

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

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section, specifically the 'SonarQube Scanner installations' configuration. A new configuration is being created with the name 'sonarqube\_exp7'. The 'Install automatically' checkbox is checked. The 'Install from Maven Central' section is expanded, showing the selected version 'SonarQube Scanner 6.2.0.4584'. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows a browser window for a Jenkins instance at `localhost:8080/view/all/newJob`. The title bar says "New Item [Jenkins]". The main content is titled "New Item" with a sub-section "Enter an item name" containing the value "exp7\_adv". A red error message below it reads "» A job already exists with the name 'exp7\_adv'". Below this, there's a section titled "Select an item type" with three options: "Freestyle project", "Pipeline", and "Multi-configuration project". The "Freestyle project" option is highlighted with a light gray background. Its description states: "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." At the bottom of the form is an "OK" button.

9. Choose this GitHub repository in Source Code Management.

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

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

The screenshot shows the Jenkins configuration interface for a job named 'exp7\_adv'. The left sidebar has 'Source Code Management' selected. The main panel is titled 'Git' and shows the configuration for a repository. The 'Repository URL' is set to <https://github.com/sunny-shaw/retail-store.git>. The 'Branch Specifier' is set to `*/main`. There are 'Save' and 'Apply' buttons 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 Jenkins configuration interface for a job. The left sidebar has 'Build Environment' selected. A dropdown menu is open over the 'Build Steps' section, showing various options: 'Execute SonarQube Scanner', '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 dropdown is a button labeled 'Add build step ^'.

**Execute SonarQube Scanner**

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

**Path to project properties** ?

**Analysis properties** ?

```
sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.sources=.
```

**Additional arguments** ?

**JVM Options** ?

Then save

The screenshot shows the Jenkins interface for the 'exp7\_adv' job. The 'Build Now' button is highlighted with a red box. The page includes navigation links like 'Dashboard > exp7\_adv >', a sidebar with options like 'Status', 'Changes', 'Workspace', 'Configure', 'Delete Project', 'SonarQube', and 'Rename'. A 'Build History' section shows a single build entry from 'Sep 26, 2024, 11:07 AM'. On the right side, there's a 'SonarQube' icon and a 'Permalinks' section. The URL in the browser is 'localhost:8080/job/exp7\_adv/'.

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

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

## ***IF CONSOLE OUTPUT FAILED:***

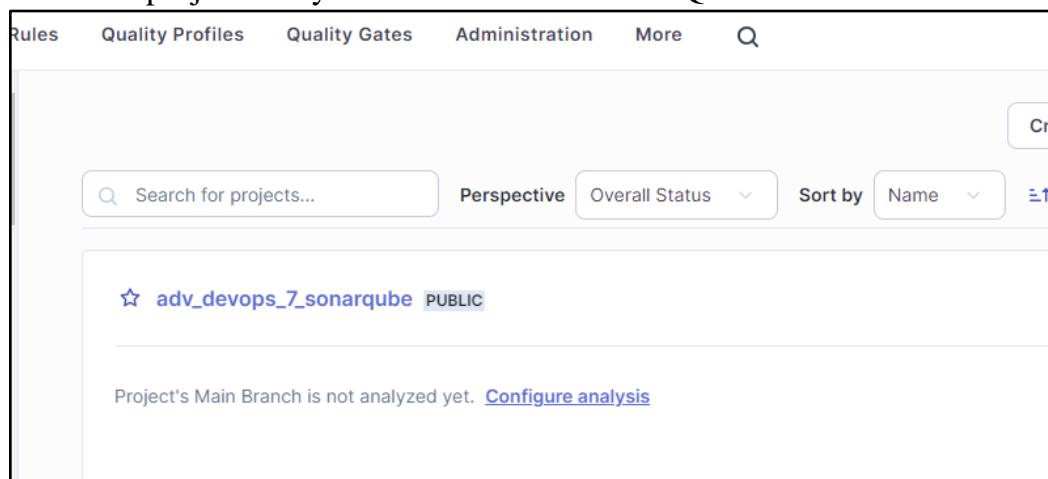
### **Step 1: Generate a New Authentication Token in SonarQube**

#### **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).

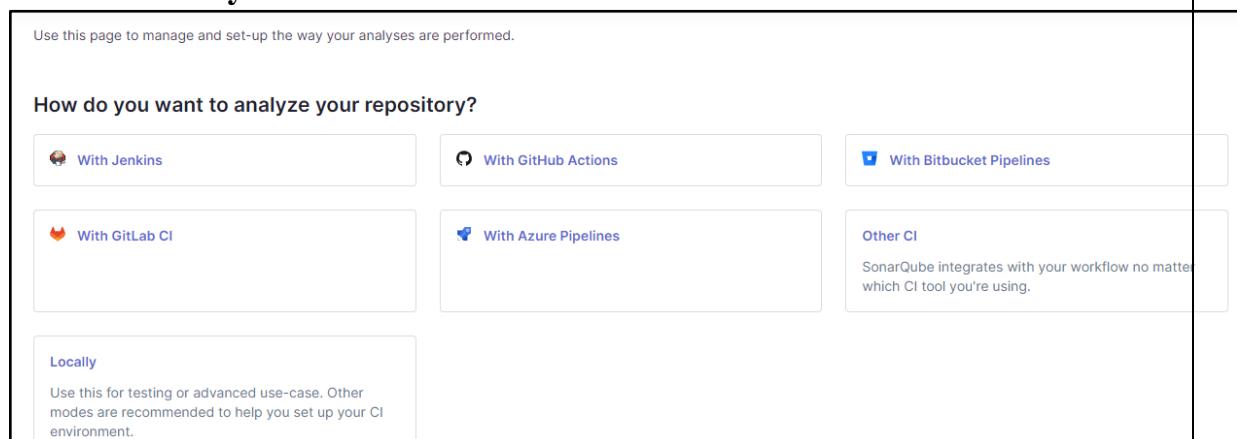
#### **2. Generate a New Token:**

- Go to the project that you have created on SonarQube.



The screenshot shows the SonarQube web interface. At the top, there is a navigation bar with tabs: Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation bar, there is a search bar labeled "Search for projects..." and a "Perspective" dropdown set to "Overall Status". To the right of the search bar are buttons for "Sort by" and "Name". A list of projects is displayed, with one project entry visible: "adv\_devops\_7\_sonarqube PUBLIC". Below the project list, a message states "Project's Main Branch is not analyzed yet." followed by a link "Configure analysis".

- Click on **Locally**



The screenshot shows the "How do you want to analyze your repository?" configuration page. The page has several options:

- "With Jenkins"
- "With GitHub Actions"
- "With Bitbucket Pipelines" (with a checked checkbox)
- "With GitLab CI"
- "With Azure Pipelines"
- "Locally" (selected, indicated by a checked checkbox) - A note below says: "Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment."
- "Other CI" - A note says: "SonarQube integrates with your workflow no matter which CI tool you're using."

- Further, Generate a Project token with the following details and click on generate.

**1 Provide a token**

Generate a project token
Use existing token

Token name ?
Expires in

1 year
Generate

ⓘ Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your [user account](#). See the [documentation](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

- Copy the token you get here and save it securely as we would need it in Jenkins.

The screenshot shows the SonarQube interface for generating a project token. At the top, there are two buttons: "Generate a project token" and "Use existing token". Below them is a form with "Token name" set to "adv\_devops\_7\_sonarqube", "Expires in" set to "1 year", and a "Generate" button. A tooltip message explains that the token is specific to the current project and can be revoked from the user account. Below the form, a note states that the token is used to identify the user during analysis and can be revoked if compromised. The background shows the SonarQube navigation bar and a project overview page.

## Step 2: Update the Token in Jenkins

- 1. Go to Jenkins Dashboard:**
  - Open Jenkins and log in with your credentials.
- 2. Go to Dashboard—>Manage Jenkins—>Credentials**

The screenshot shows the Jenkins 'Credentials' page under 'Manage Jenkins'. A specific credential entry is highlighted:

T	P	Store	Domain	ID	Name
		System	(global)	sonarqube_token	/*****

Below this, a section titled 'Stores scoped to Jenkins' is shown, listing a single 'System' store with domain '(global)'. There are also icons for S, M, and L.

3. Click on **global** under the domains part of Stores scoped to Jenkins section.Further click on add credentials.Proceed with the following details.Make sure to copy the token generated earlier in sonarqube and give any suitable name as the ID.

The screenshot shows the Jenkins 'New credentials' page under 'Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. A new credential is being created with the following details:

Kind	Secret text
Scope	Global (Jenkins, nodes, items, all child items, etc)
Secret	.....
ID	himesh
Description	this exp 7

A blue 'Create' button is at the bottom left.

4.After clicking on create we see that the given token has been added in Jenkins credentials.

The screenshot shows the Jenkins Global credentials (unrestricted) page. At the top, there is a search bar and a user menu for 'admin'. Below the header, the breadcrumb navigation shows: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). A blue button labeled '+ Add Credentials' is visible on the right. The main table displays one credential entry:

ID	Name	Kind	Description
himesh	this exp 7	Secret text	this exp 7

Below the table, there is a legend for icons: S (Small), M (Medium), and L (Large). At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.2'.

5.Now go to **Manage Jenkins**—>**System**—>**SonarQube servers** and proceed with the following details. Reference the authentication token generated in the previous step.

The screenshot shows the 'SonarQube servers' configuration page. It includes the following sections:

- SonarQube servers**: A note stating that checked boxes allow job administrators to inject SonarQube server configurations as environment variables.
- Environment variables**: A checked checkbox.
- SonarQube installations**: A link to a list of installations.
- Name**: A text input field containing 'adv\_devops\_7\_sonarqube'.
- Server URL**: A text input field with the default value 'http://localhost:9000' and an optional value 'http://localhost:9000'.
- Server authentication token**: A text input field containing 'advance devops exp7'.
- + Add**: A button to add more servers.

## 6.Check the SonarQube Scanner Environment and add the server authentication token

Build Environment

- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Prepare SonarQube Scanner environment ?  
Server authentication token  
SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

Execute SonarQube Scanner

JDK ?  
JDK to be used for this SonarQube analysis

Path to project properties ?

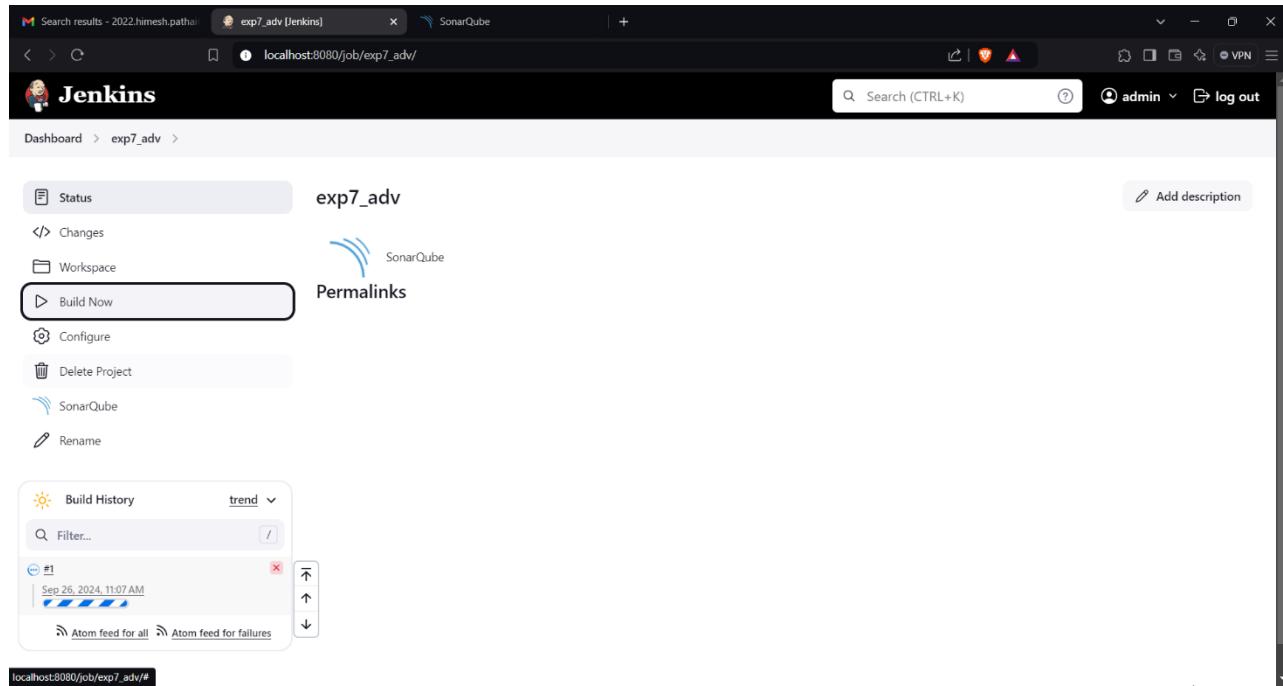
Analysis properties ?  

```
sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
-Dsonar.login=sqp_568834b7b5e77a92843e4b3072e044643ce921c1
sonar.sources=.
```

Additional arguments ?

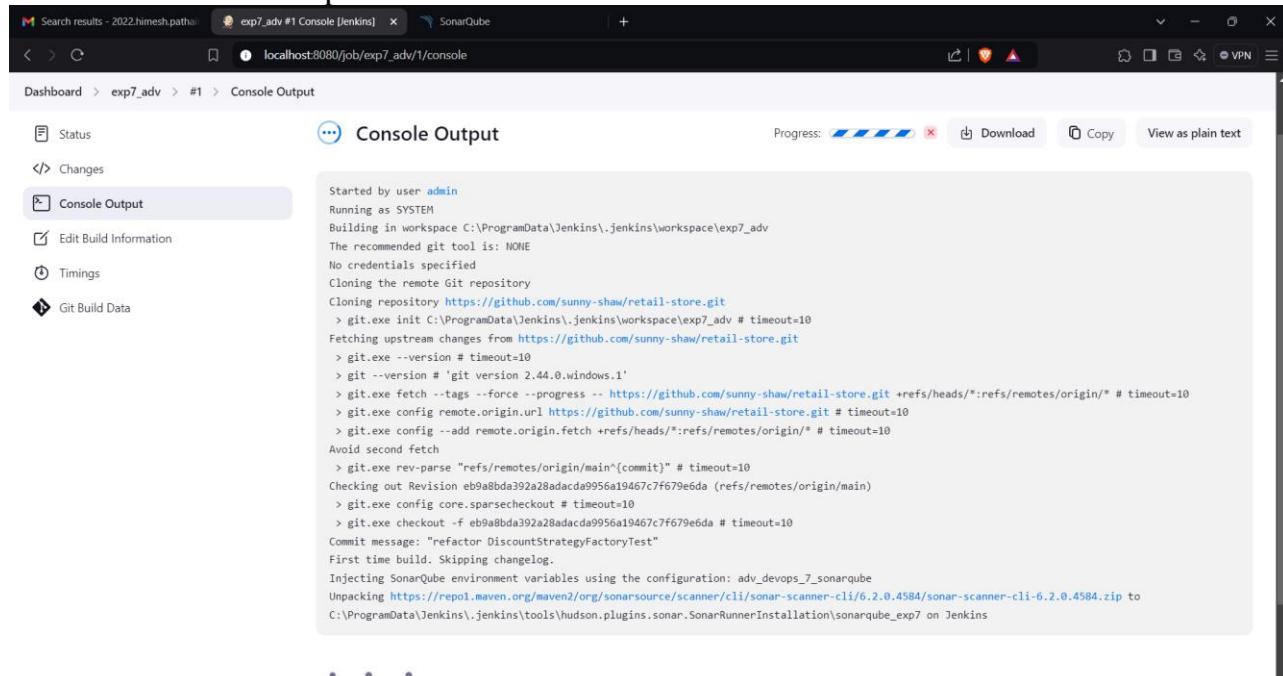
JVM Options ?

## 12.Run the Jenkins build.



The screenshot shows the Jenkins interface for the job 'exp7\_adv'. The 'Build Now' button is highlighted with a red box. The 'Build History' section shows one build (#1) from Sep 26, 2024, at 11:07 AM. Below the history are links for 'Atom feed for all' and 'Atom feed for failures'.

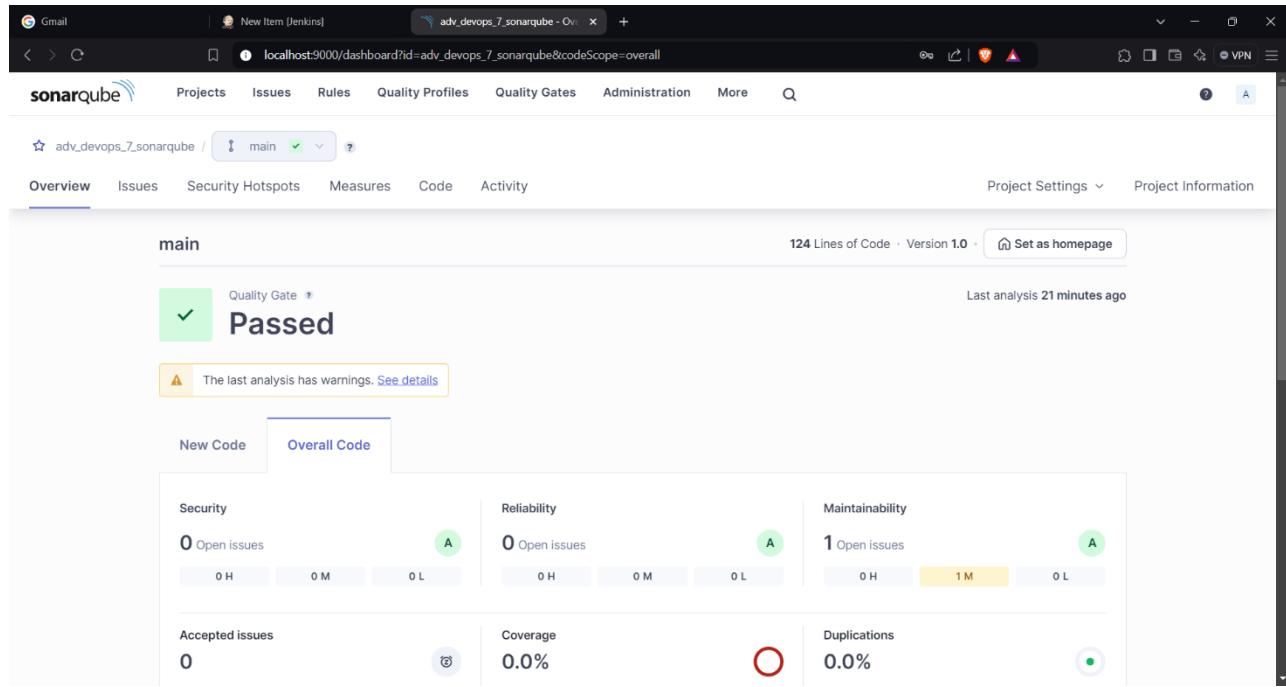
## Check the console Output



The screenshot shows the Jenkins interface for the job 'exp7\_adv', specifically the 'Console Output' tab. The log output is as follows:

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\exp7_adv
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/sunny-shaw/retail-store.git
> git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\exp7_adv # timeout=10
Fetching upstream changes from https://github.com/sunny-shaw/retail-store.git
> git.exe --version # timeout=10
> git --version # 'git' version 2.44.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/sunny-shaw/retail-store.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/sunny-shaw/retail-store.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision eb9a8bda392a28adacda9956a19467c7f679e6da (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f eb9a8bda392a28adacda9956a19467c7f679e6da # timeout=10
Commit message: "refactor DiscountStrategyFactoryTest"
First time build. Skipping changelog.
Injecting SonarQube environment variables using the configuration: adv_devops_7_sonarqube
Unpacking https://repo1.maven.org/maven2/org/sonarsource/scanner/cli/sonar-scanner-cli/6.2.0.4584/sonar-scanner-cli-6.2.0.4584.zip to C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_exp7 on Jenkins
```

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



The screenshot shows the SonarQube main dashboard for the 'main' branch. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation is a breadcrumb trail showing the project path: adv\_devops\_7.sonarqube / main. The main content area displays a large green 'Passed' status with a checkmark icon. A yellow warning box indicates that the last analysis has warnings, with a 'See details' link. The dashboard is divided into several sections: Security (0 Open issues), Reliability (0 Open issues), Maintainability (1 Open issue), Accepted issues (0), Coverage (0.0%), and Duplications (0.0%). The coverage section is highlighted with a red circle. At the bottom right, there are buttons for 'Set as homepage' and a 'Project Settings' dropdown.

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

Name : **Himesh Pathai**

Class : **D15A**

Roll No. : **35**

## **Experiment 8**

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

Step 1: Log in to sonarqube portal and create a local project.

The screenshot shows the 'Create a local project' form in SonarQube. It includes fields for 'Project display name' (sonarqube-pipeline), 'Project key' (sonarqube-pipeline), and 'Main branch name' (main). A note below specifies the default branch. Buttons for 'Cancel' and 'Next' are at the bottom.

The screenshot shows the 'Set up project for Clean as You Code' step in SonarQube. It asks to choose a baseline for new code, with 'Use the global setting' selected. It also provides options for 'Previous version' and 'Define a specific setting for this project'. The 'Previous version' option is highlighted with a gray background.

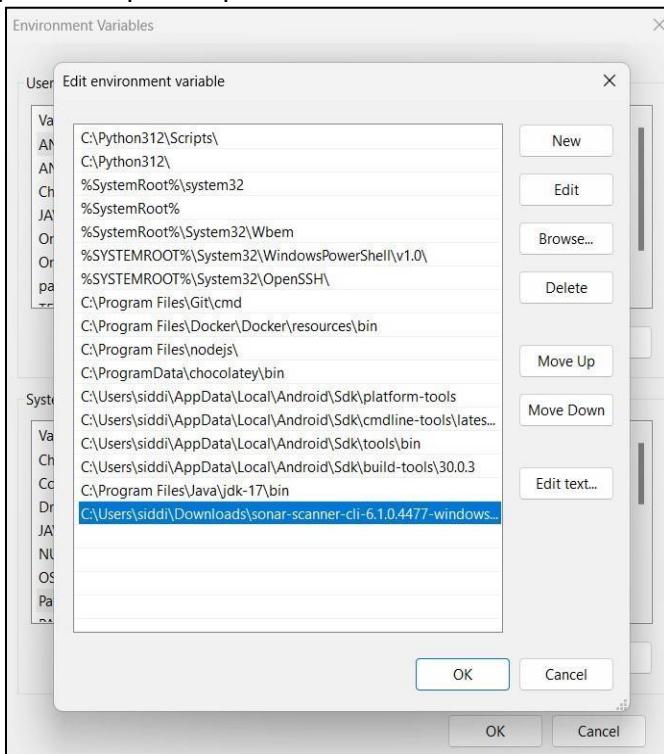
## Step 2: Go to [download sonarscanner](#) to download sonar scanner

The screenshot shows the SonarScanner CLI documentation page on the SonarQube website. The left sidebar contains links for SonarQube analysis overview, Project analysis setup, Scanners (with Scanner environment, SonarScanner CLI, SonarQube extension for Azure DevOps, SonarQube extension for Jenkins, SonarScanner for .NET, SonarScanner for Maven, SonarScanner for Gradle, SonarScanner for NPM, SonarScanner for Ant (Deprecated), SonarScanner for Python (Beta), and Analysis parameters). The main content area displays three releases:

- 6.2** (2024-09-17): Support PKCS12 truststore generated with OpenSSL. Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM). Release notes.
- 6.1** (2024-06-27): macOS and Linux AArch64 distributions. Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM). Release notes.
- 6.0** (2024-06-04): New bootstrapping mechanism and JRE provisioning with SonarQube 10.6+ and SonarCloud. Download scanner for: Linux x64, Windows x64, macOS x64, Docker Any (Requires a pre-installed JVM).

A red box highlights the "Windows x64" link under the 6.2 release.

After the download is complete, extract the file and copy the path to bin folder Go to environment variables, system variables and click on path Add a new path, paste the path copied earlier.



### Step 3: Create a New Item in Jenkins, choose Pipeline.

Dashboard > All > New Item

### New Item

Enter an item name  
sonarqube-pipeline

Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace so you can have multiple things of the same name as long as they are in different folders.

OK

Dashboard > sonarqube-pipeline > Configuration

### Configure Pipeline

Definition: Pipeline script

Script :

```
1+ node {
2+   stage('Cloning the GitHub Repo') {
3+     git 'https://github.com/shazforiot/GOL.git'
4+   }
5+   stage('SonarQube analysis') {
6+     withSonarQubeEnv('sonarqube') {
7+       bat """
8+         C:\\\\Users\\\\siddi\\\\Downloads\\\\sonar-scanner-cli-6.1.0.4477-windows-x64\\\\sonar-scanner-6.1.0.4477-windows-x64\\\\bin\\\\sonar-scanner
9+         -Dsonar.login=admin ^
10+        -Dsonar.password=MahiVish ^
11+        -Dsonar.projectKey=sonarqube-pipeline ^
12+        -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^
13+        -Dsonar.host.url=http://localhost:9000/
14+
15+     }
16+   }
17+ }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

## Step 4: Save the pipeline and build it.

The screenshot shows the SonarQube Pipeline interface. At the top, there's a navigation bar with 'Dashboard > sonarqube-pipeline >'. Below it is a sidebar with various options: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, Pipeline Syntax, Build History, and Permalinks. The main area is titled 'Stage View' and displays two stages: 'Cloning the GitHub Repo' (9s) and 'SonarQube analysis' (3min 53s). A summary bar indicates an average stage time of 9s and a full run time of ~7min 49s. Below this, two builds are shown: build #2 (Sep 26 20:42) which was successful ('No Changes') and build #1 (Sep 26 20:24) which failed ('No Changes'). The failed build has a red background. To the right, a list of build history items is provided:

- Last build (#2), 9 min 1 sec ago
- Last stable build (#2), 9 min 1 sec ago
- Last successful build (#2), 9 min 1 sec ago
- Last failed build (#1), 26 min ago
- Last unsuccessful build (#1), 26 min ago
- Last completed build (#2), 9 min 1 sec ago

## Console output:

The screenshot shows the 'Console Output' section of the SonarQube Pipeline interface. The sidebar on the left includes options like Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#2', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Console Output' and shows the following log output:

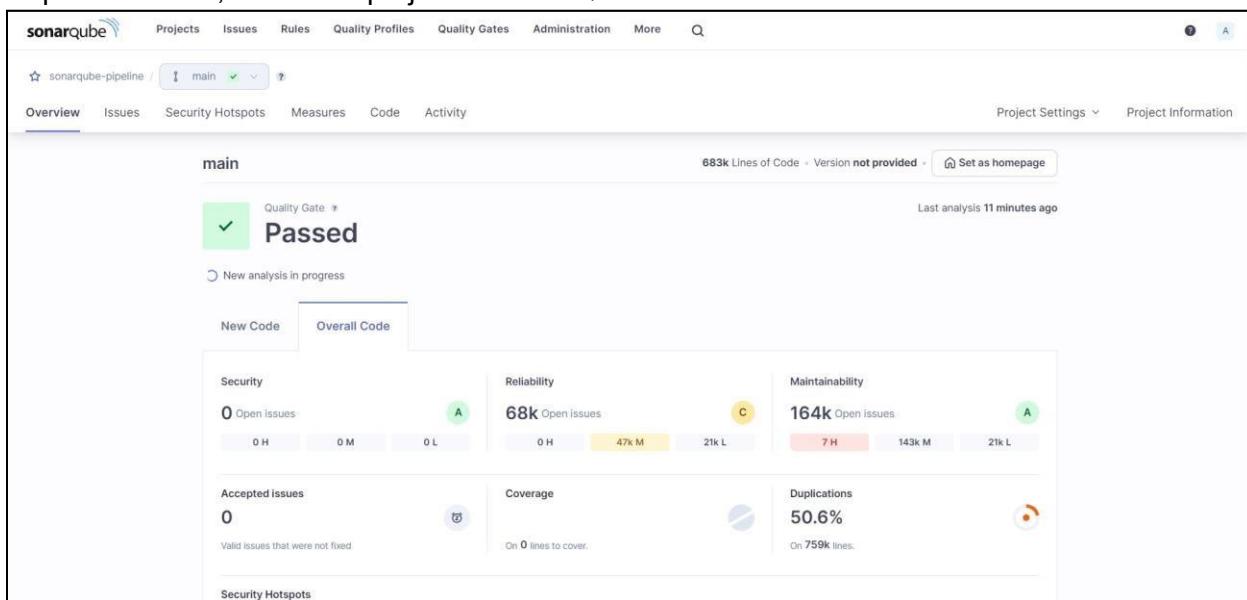
```
Skipping 4.248 KB. Full Log
20:49:35.711 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 40. Keep only the first 100 references.
20:49:35.712 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 65. Keep only the first 100 references.
20:49:35.712 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 14. Keep only the first 100 references.
20:49:35.712 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 17. Keep only the first 100 references.
20:49:35.712 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 1487. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 226. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 229. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 225. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 226. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 424. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 17. Keep only the first 100 references.
20:49:35.812 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 17. Keep only the first 100 references.
```

```

20:50:01.832 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-pipeline
20:50:01.832 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:50:01.832 INFO More about the report processing at http://localhost:9000/api/ce/task?id=159a9d05-1f5f-4e17-bd27-3643a32a836a
20:50:12.108 INFO Analysis total time: 7:37.235 s
20:50:12.110 INFO SonarScanner Engine completed successfully
20:50:12.849 INFO EXECUTION SUCCESS
20:50:12.851 INFO Total time: 7:44.878s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## Step 5: After that, check the project in SonarQube



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

SonarQube Project Overview for sonarqube-pipeline

**Measures** tab selected.

**Reliability** section:

- Maintainability
- Security Review
- Duplications
- Size
- Complexity

**Issues** section:

- Overall Code
- Open Issues: 210,549
- Confirmed Issues: 0
- Accepted Issues: 0
- False Positive Issues: 0

**Open Issues** section:

- gameoflife-acceptance-tests: 4 issues
- gameoflife-build: 0 issues
- gameoflife-core: 603 issues
- gameoflife-deploy: 0 issues
- gameoflife-web: 209,940 issues
- pom.xml: 2 issues

6 files total.

SonarQube Issues View for sonarqube-pipeline

**Issues** tab selected.

**Filters** section:

- My Issues
- All

**Bulk Change** button.

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

- Insert a <!DOCTYPE> declaration to before this <html> tag.  
Reliability  
Consistency  
user-experience
- Remove this deprecated "width" attribute.  
Maintainability  
HTML5 obsolete
- Remove this deprecated "align" attribute.  
Maintainability  
HTML5 obsolete
- Remove this deprecated "align" attribute.  
Maintainability  
HTML5 obsolete

196,662 issues | 3,075d effort

SonarQube Issues Page (gameoflife-acceptance-tests/Dockerfile)

Filters: Intentionality

Issues in new code:

- Clean Code Attribute:
  - Intentionality: 14k
  - Consistency: 197k
  - Adaptability: 0
  - Responsibility: 0
- Software Quality:
  - Security: 0
  - Reliability: 14k
  - Maintainability: 15

Bulk Change: Use a specific version tag for the image.

Select issues: 13,887 issues | 59d effort

Project Settings | Project Information

Introducing Clean Code Attributes (1 of 5):

Clean Code attributes are the characteristics that your code must have to be considered Clean Code. You can now filter by these attributes to evaluate why your code is breaking away from being clean.

Next

SonarQube Issues Page (gameoflife-core/build/reports/tests/all-tests.html)

Filters: Intentionality

Issues in new code:

- Clean Code Attribute:
  - Intentionality: 14k
  - Consistency: 54k
  - Adaptability: 0
  - Responsibility: 0
- Software Quality:
  - Security: 0
  - Reliability: 14k
  - Maintainability: 15

Bulk Change: Add "lang" and/or "xml:lang" attributes to this "<html>" element.

Select issues: 13,872 issues | 59d effort

Project Settings | Project Information

Introducing Clean Code Attributes (1 of 5):

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

Intentionality: Reliability, accessibility, wcag2-a

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

Intentionality: Reliability, accessibility, wcag2-a

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

Intentionality: Reliability, accessibility, wcag2-a

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

Intentionality: Reliability, accessibility, wcag2-a

SonarQube Issues Overview

Project: sonarqube-pipeline / main

Filters: My Issues, All, Clear All Filters

Issues in new code

Clean Code Attribute:

- Consistency: 164k
- Intentionality: 15 (selected)
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

Software Quality:

- Security: 0
- Reliability: 14k
- Maintainability: 15 (selected)

Add to selection Ctrl + click

Bulk Change: Select issues, Navigate to issue, 15 issues, 44min effort

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL1 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL12 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL12 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedNo tags +

SonarQube Issues Overview

Project: sonarqube-pipeline / main

Filters: Software Quality

Severity: ?

Type:

- Bug: 0
- Vulnerability: 0
- Code Smell: 15 (selected)

Add to selection Ctrl + click

Scope: ?

Status: ?

Security Category: ?

Creation Date: ?

Bulk Change: Select issues, Navigate to issue, 15 issues, 44min effort

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL1 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL12 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedL12 ~ 5min effort ~ 4 years ago ~ ⚡ Code Smell ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality: Maintainability (yellow) No tags +
  - Open Not assignedNo tags +

SonarQube Project: sonarqube-pipeline / main

Overview Issues Security Hotspots Measures Code Activity

0.0% Security Hotspots Reviewed

3 Security Hotspots

Review priority: Medium

Permission: The tomcat image runs with root as the default user. Make sure it is safe here.

Review priority: Low

Encryption of Sensitive Data: The tomcat image runs with root as the default user. Make sure it is safe here.

Others: The tomcat image runs with root as the default user. Make sure it is safe here.

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk? What's the risk? Assess the risk How can I fix it? Activity

gameoflife-web/Dockerfile

FROM tomcat:8-jre8  
The tomcat image runs with root as the default user. Make sure it is safe here.  
RUN rm -rf /usr/local/tomcat/webapps/\*  
COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war  
EXPOSE 8080  
CMD ["catalina.sh", "run"]

Open in IDE

SonarQube Project: sonarqube-pipeline / main

Overview Issues Security Hotspots Measures Code Activity

Reliability

Maintainability

Security Review

Duplications

Overall Code

Density: 50.6%

Duplicated Lines: 384,007

Duplicated Blocks: 42,808

Duplicated Files: 979

Size

sonarqube-pipeline

Duplicated Lines (%): 50.6% See history

	Duplicated Lines (%)	Duplicated Lines
gameoflife-acceptance-tests	0.0%	0
gameoflife-build	0.0%	0
gameoflife-core	9.6%	374
gameoflife-deploy	0.0%	0
gameoflife-web	50.9%	383,633
pom.xml	0.0%	0

The screenshot shows the SonarQube web interface for the 'sonarqube-pipeline' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation is a breadcrumb trail: sonarqube-pipeline / main. The main content area has tabs for Overview, Issues, Security Hotspots, Measures (which is selected), Code, and Activity. On the right side, there are Project Settings and Project Information dropdowns.

The left sidebar contains sections for Security Review, Duplications, Overall Code, Size, and Complexity. The Complexity section is expanded, showing Cyclomatic Complexity at 1,112. The main panel displays a tree view of the project structure under 'sonarqube-pipeline'. The tree includes nodes for gameoflife-acceptance-tests, gameoflife-build, gameoflife-core, gameoflife-deploy, gameoflife-web, and pom.xml. The gameoflife-web node is expanded, showing 1,094 files. A message at the bottom indicates '6 of 6 shown'.

Name : **Himesh Pathai**

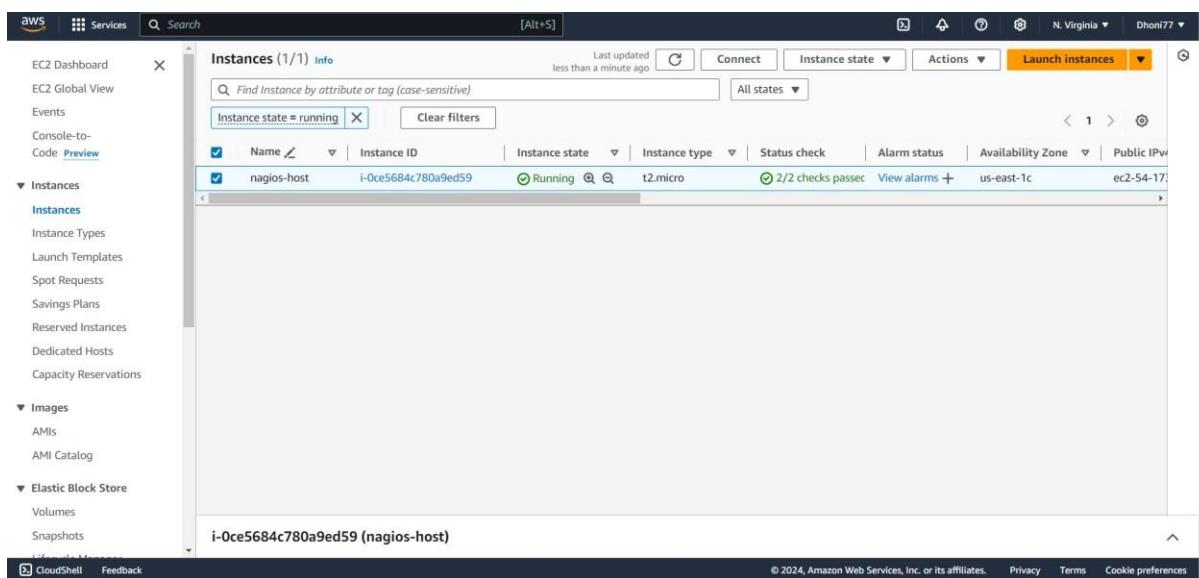
Class : **D15A**

Roll No. : **35**

## **EXPERIMENT 9**

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

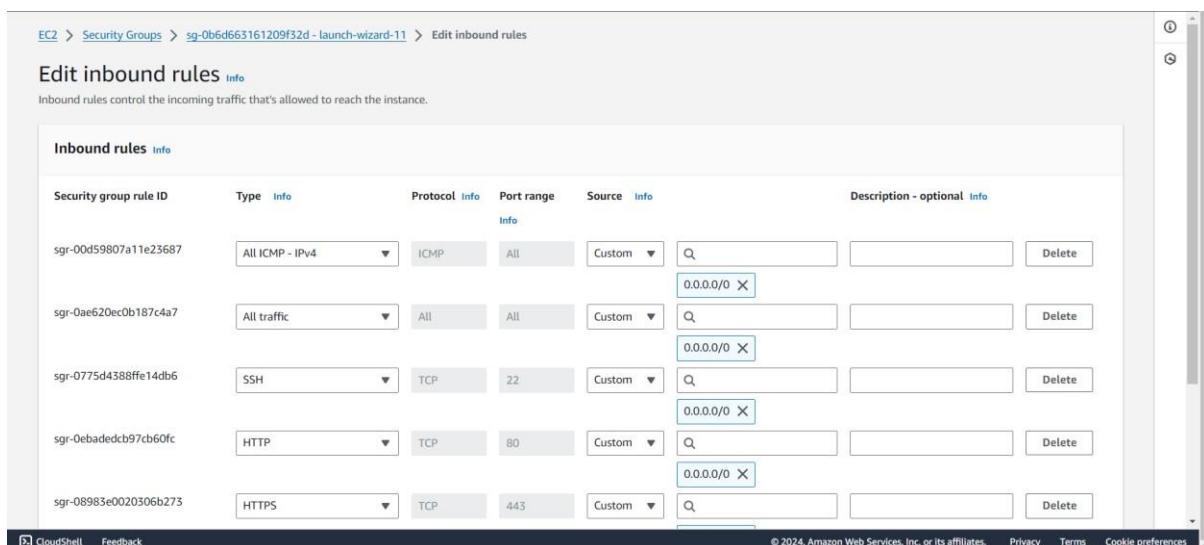
### **1. Create an Amazon Linux EC2 Instance**



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main area displays a table for 'Instances (1/1)'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. One row is selected, showing 'nagios-host' with the instance ID 'i-0ce5684c780a9ed59', which is 'Running', 't2.micro', '2/2 checks passed', 'us-east-1c', and 'ec2-54-17'. At the bottom of the table, it says 'i-0ce5684c780a9ed59 (nagios-host)'. The footer includes links for CloudShell, Feedback, and copyright information from 2024.

### **2. Configure Security Group**

- Ensure HTTP, HTTPS, SSH, and ICMP are open from everywhere.
- Edit the inbound rules of the specified Security Group



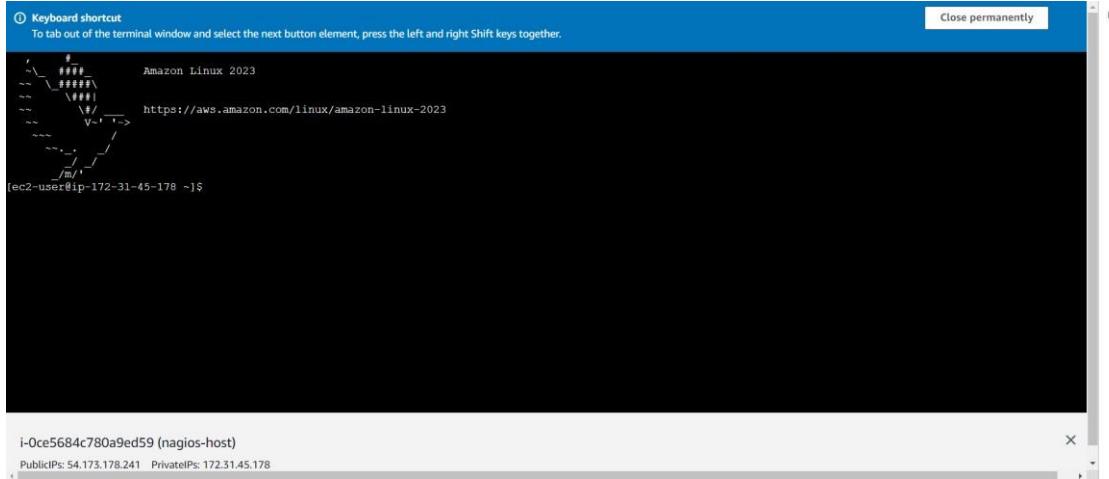
The screenshot shows the AWS Security Groups page. The URL is 'EC2 > Security Groups > sg-0b6d663161209f32d - launch-wizard-11 > Edit inbound rules'. The page title is 'Edit inbound rules' with a 'Info' link. A note below says 'Inbound rules control the incoming traffic that's allowed to reach the instance.' Below this is a table titled 'Inbound rules' with an 'Info' link. The table columns are: Security group rule ID, Type (with an 'Info' link), Protocol (with an 'Info' link), Port range (with an 'Info' link), Source (with an 'Info' link), and Description - optional (with an 'Info' link). There are five rows in the table:

- sgr-00d59807a11e23687: Type: All ICMP - IPv4, Protocol: ICMP, Port range: All, Source: Custom, Description: (empty)
- sgr-0ae620ec0b187c4a7: Type: All traffic, Protocol: All, Port range: All, Source: Custom, Description: (empty)
- sgr-0775d4388ffe14db6: Type: SSH, Protocol: TCP, Port range: 22, Source: Custom, Description: (empty)
- sgr-0ebadedcb97cb60fc: Type: HTTP, Protocol: TCP, Port range: 80, Source: Custom, Description: (empty)
- sgr-08983e0020306b273: Type: HTTPS, Protocol: TCP, Port range: 443, Source: Custom, Description: (empty)

At the bottom, there are 'CloudShell', 'Feedback', and copyright information from 2024.

You have to edit the inbound rules of the specified Security Group for this.

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



```
curl https://aws.amazon.com/linux/amazon-linux-2023
```

### 4. Update the package indices and install the following packages using yum sudo yum update

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum update
Last metadata expiration check: 0:01:31 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-254 ~]$
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:59 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
Dependencies resolved.
-----
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
php8_3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 k
Installing dependencies:				
apc	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apc-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libxml	x86_64	1.1.34-5.amzn2023.0.2	amazonlinux	241 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.4	amazonlinux	9.8 M
php8_3-cli	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	3.7 M
php8_3-common	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	737 k
php8_3-process	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	45 k

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:41 ago on Wed Oct 2 05:48:47 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package          Architecture Version      Repository  Size
=====
Installing:
  gcc             x86_64    11.4.1-2.amzn2023.0.2   amazonlinux 32 M
  glibc           noarch    10.93-1.amzn2023.0.1   amazonlinux 92 M
  glibc-common    x86_64    10.93-1.amzn2023.0.1   amazonlinux 887 M
  glibc-devel     x86_64    11.4.1-2.amzn2023.0.2   amazonlinux 10 M
  gc              x86_64    8.0.4-5.amzn2023.0.2   amazonlinux 105 M
  glibc-devel     x86_64    2.34-52.amzn2023.0.11  amazonlinux 27 M
  glibc-headers-x86 x86_64    2.34-52.amzn2023.0.11  amazonlinux 427 M
  guile22        x86_64    2.2.7-2.amzn2023.0.3   amazonlinux 6.4 M
  kernel-headers x86_64    6.1.109-118.189.amzn2023  amazonlinux 1.4 M
  libmpc          x86_64    1.2.1-2.amzn2023.0.2   amazonlinux 62 M
  libtool-ltdl    x86_64    2.4.7-1.amzn2023.0.3   amazonlinux 38 M
  libxcrypt-devel x86_64    4.4.33-7.amzn2023  amazonlinux 32 M
  make            x86_64    1:4.3-5.amzn2023.0.2  amazonlinux 534 M
=====
Transaction Summary
=====

```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:46 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
=====
Package          Architecture Version      Repository  Size
=====
Installing:
  gd              x86_64    2.3.3-5.amzn2023.0.3   amazonlinux 139 M
  gd-devel        x86_64    2.3.3-5.amzn2023.0.3   amazonlinux 38 M
  brotli          x86_64    1.0.9-4.amzn2023.0.2   amazonlinux 314 M
  brotli-devel    x86_64    1.0.9-4.amzn2023.0.2   amazonlinux 31 M
  bzip2-devel     x86_64    1.0.8-6.amzn2023.0.2   amazonlinux 214 M
  cairo           x86_64    1.17.6-2.amzn2023.0.1  amazonlinux 694 M
  cmake-filesystem x86_64    3.22.2-1.amzn2023.0.4  amazonlinux 16 M
  fontconfig       x86_64    2.13.94-2.amzn2023.0.2  amazonlinux 273 M
  fontconfig-devel x86_64    2.13.94-2.amzn2023.0.2  amazonlinux 128 M
  fonts-filesystem noarch    1:2.0.5-12.amzn2023.0.2  amazonlinux 9.5 M
  freetype         x86_64    2.13.2-5.amzn2023.0.1  amazonlinux 423 M
  freetype-devel   x86_64    2.13.2-5.amzn2023.0.1  amazonlinux 912 M
  glib2-devel      x86_64    2.74.7-68.amzn2023.0.2  amazonlinux 486 M
  google-noto-fonts-common noarch    20201206-2.amzn2023.0.2  amazonlinux 15 M
  google-noto-sans-vf-fonts noarch    20201206-2.amzn2023.0.2  amazonlinux 492 M
  graphite2        x86_64    1.3.14-7.amzn2023.0.2  amazonlinux 97 M
  graphite2-devel  x86_64    1.3.14-7.amzn2023.0.2  amazonlinux 21 M
  harfbuzz         x86_64    7.0.0-2.amzn2023.0.1  amazonlinux 868 M
  harfbuzz-devel   x86_64    7.0.0-2.amzn2023.0.1  amazonlinux 404 M
  harfbuzz-icu     x86_64    7.0.0-2.amzn2023.0.1  amazonlinux 18 M
=====

```

## 5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

sudo adduser -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-40-254 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-40-254 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-254 ~]$ █
```

## 6. Create a new user group sudo groupadd nagcmd

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$ █
```

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

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

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-40-254 ~]$ █
```

## **8. Create a new directory for Nagios downloads** `mkdir ~/downloads` `cd ~/downloads`

```
[ec2-user@ip-172-31-40-254 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-40-254 downloads]$ █
```

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

```
Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz wget
```

```
[ec2-user@ip-172-31-40-254 downloads]$ Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
--2024-10-02 06:15:45-- https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)[45.56.123.251]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'

nagios-plugins-2.3.3.tar.gz      0%[=====]   632.00K  3.02MB/s
nagios-plugins-2.3.3.tar.gz    23%[=====>]  2.65M  8.10MB/s  in 0.3s
nagios-plugins-2.3.3.tar.gz  100%[=====]  2.65M  8.10MB/s  in 0.3s

2024-10-02 06:15:46 (8.10 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]

[ec2-user@ip-172-31-40-254 downloads]$
```

<https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz>

```
[ec2-user@ip-172-31-40-254 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-10-02 06:17:24-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.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: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz          0%[=====]   0      0 --.-KB/s
nagios-4.4.6.tar.gz          4%[==>]    495.62K 2.40MB/s
nagios-4.4.6.tar.gz          30%[=====>]   3.26M 7.99MB/s
nagios-4.4.6.tar.gz          63%[=====>]   6.91M 11.0MB/s
nagios-4.4.6.tar.gz          96%[=====>]  10.46M 12.6MB/s
nagios-4.4.6.tar.gz     100%[=====>]  10.81M 12.9MB/s  in 0.8s

2024-10-02 06:17:25 (12.9 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

[ec2-user@ip-172-31-40-254 downloads]$
```

## **10. Use tar to unzip and change to that directory.**

```
tar zxvf nagios-4.4.6.tar.gz cd  
nagios-4.4.6
```

```
[ec2-user@ip-172-31-40-254 downloads]$ tar zxvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/ChangeLog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
nagios-4.4.6/autoconf-macros/ax_nagios_get_files
nagios-4.4.6/autoconf-macros/ax_nagios_get_inetd
nagios-4.4.6/autoconf-macros/ax_nagios_get_init
nagios-4.4.6/autoconf-macros/ax_nagios_get_os
nagios-4.4.6/autoconf-macros/ax_nagios_get_paths
nagios-4.4.6/autoconf-macros/ax_nagios_get_ssl
nagios-4.4.6/base/
nagios-4.4.6/base/.gitignore
nagios-4.4.6/base/Makefile.in
nagios-4.4.6/base/broker.c
```

## 11. Run the configuration script with the same group name you previously created.

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

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./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 we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $[MAKE]... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
```

## 12. Compile the source code.

make all

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netmods.o netmods.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o .../common/shared.o .../common/shared.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflows=]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
   |         ^
   |         ~~~~~
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o .../common/macros.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netutils.o netutils.c
netutils.c: In function 'my_tcp_connect':
netutils.c:50:47: warning: '%d' directive output may be truncated writing between 1 and 11 bytes into a region of size 6 [-Wformat-truncation=]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
   |     ~~~~~
netutils.c:50:46: note: directive argument in the range [-2147483648, 65535]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
netutils.c:50:9: note: 'sprintf' output between 2 and 12 bytes into a destination of size 6
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o notifications.o notifications.c
```

```
make install-classicui
- This installs the classic theme for the Nagios
  web interface

*** Support Notes *****
If you have questions about configuring or running Nagios,
please make sure that you:
  - Look at the sample config files
  - Read the documentation on the Nagios Library at:
    https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:
  - What version of Nagios you are using
  - What version of the plugins you are using
  - Relevant snippets from your config files
  - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:
  https://support.nagios.com

*****
Enjoy.
```

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

```
./sudo make install sudo make
```

```
install-init sudo make install-
```

```
config sudo make install-
```

```
commandmode
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
-bash: ./sudo: No such file or directory
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

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

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

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

## 14. Edit the config file and change the email address.

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

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
Just one contact defined by default - the Nagios admin (that's you)
This contact definition inherits a lot of default values from the
'generic-contact' template which is defined elsewhere.

define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin        ; Full name of user
    email             bhagyeshpatil0702@gmail.com; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

CONTACT GROUPS

We only have one contact in this simple configuration file, so there is
no need to create more than one contact group.

define contactgroup {
```

## 15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

**16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.** sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

**17. Restart Apache** sudo systemctl restart httpd

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

cd ~/downloads tar zxvf nagios-

plugins-2.3.3.tar.gz cd nagios-

plugins-2.3.3

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ cd ~/downloads
tar zxvf nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.3.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Module-File-Type-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/ABOUT-NLS
nagios-plugins-2.3.3/configure.ac
nagios-plugins-2.3.3/Makefile.in
nagios-plugins-2.3.3/config.h.in
nagios-plugins-2.3.3/ChangeLog
nagios-plugins-2.3.3/AUTHORS
nagios-plugins-2.3.3/lib/
nagios-plugins-2.3.3/lib/parse_ini.h
nagios-plugins-2.3.3/lib/extr_opts.c
nagios-plugins-2.3.3/lib/Makefile.in
```

**18. Go back to the downloads folder and unzip the plugins zip file.** ./configure --with-nagios-user=nagios --with-nagios-group=nagios make sudo make install

```
file. ./configure --with-nagios-user=nagios --with-nagios-
group=nagios make sudo make install
```

```
ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
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 to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
```

**19. Compile and install plugins** sudo chkconfig --add nagios sudo  
 chkconfig nagios on  
 sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg sudo  
 systemctl start nagios

```
ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo systemctl start nagios
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.

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.
```

## 20. Check the status of Nagios

```
things look okay - No serious problems were detected during the pre-flight check
ec2-user@ip-172-31-45-178 nagios-plugins-2.3.3]$ sudo systemctl status nagios
nagios.service - Nagios Core 4.4.6
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
  Active: active (running) since Wed 2024-10-02 05:37:36 UTC; 14s ago
    Docs: https://www.nagios.org/documentation
  Process: 67990 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 67991 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67992 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
     CPU: 16ms
    CGroup: /system.slice/nagios.service
           └─67992 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─67993 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67994 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67995 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67996 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─67997 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: core query handler registered
```

## 23. Open up your browser and look for <http://<your public ip address>/nagios>

## Sign in

http://54.173.178.241

Your connection to this site is not private

Username

Password

**Sign in**

**Cancel**

The screenshot shows the Nagios Core 4.4.6 dashboard. At the top, it displays the Nagios Core logo and a green checkmark indicating the daemon is running with PID 67992. Below this, it shows the version information: Nagios® Core™ Version 4.4.6, dated April 28, 2020, with a link to check for updates. A blue banner at the top right announces a new version of Nagios Core is available, with a link to download Nagios 4.5.5.

The left sidebar contains a navigation menu with sections: General, Current Status, Problems, Reports, and System. The 'Current Status' section is currently selected and shows links for Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Summary, Service Groups, and Grid. The 'Problems' section shows a summary of unhandled services, hosts, and network outages. The 'Reports' section includes Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, and Event Log. The 'System' section includes Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

The main content area is divided into several panels: 'Get Started' (with a list of bullet points), 'Quick Links' (with links to Nagios Library, Nagios Labs, Nagios Exchange, Nagios Support, Nagios.com, and Nagios.org), 'Latest News' (empty), and 'Don't Miss...' (empty). At the bottom, there is a copyright notice: Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

**Name : Himesh Pathai**  
**Class : D15A**  
**Roll No. : 35**

## **EXPERIMENT 10**

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

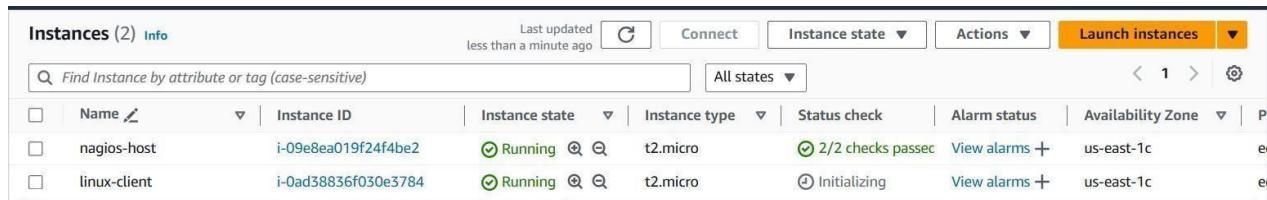
### **Procedure:-**

**Check if the nagios service is running by executing following command**  
sudo systemctl status nagios

```
ubuntu@ip-172-31-89-161:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 16:08:58 UTC; 1min 2s ago
     Docs: https://www.nagios.org/documentation
 Process: 15743 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 15753 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 15764 (nagios)
   Tasks: 6 (limit: 1130)
  Memory: 2.4M (peak: 3.2M)
    CPU: 29ms
   CGroup: /system.slice/nagios.service
           ├─15764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─15765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─15769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: core query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: echo service query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: help for the query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15765;pid=15765
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15766;pid=15766
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15767;pid=15767
```

**Now, create a new EC2 instance on AWS**



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
nagios-host	i-09e8ea019f24f4be2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c
linux-client	i-0ad38836f030e3784	Running	t2.micro	Initializing	View alarms +	us-east-1c

**Now perform the following commands on nagios-host EC2 instance. On the server, run this command**

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

```
ubuntu@ip-172-31-89-161:~$ ps -ef | grep nagios
nagios 15764 1 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 15765 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15766 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15767 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15768 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15769 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ubuntu 15957 1342 0 16:13 pts/0 00:00:00 grep --color=auto nagios
ubuntu@ip-172-31-89-161:~$
```

Sudo su

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

```
ubuntu@ip-172-31-89-161:~$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/home/ubuntu#
```

### Copy localhost.cfg file to the mentioned location

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

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts': No such file or directory
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# sudo mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects#
```

### Open the nano editor for localhost.cfg file and make these changes. Add the Ip address of the linux-client for the address field.

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

```

GNU nano 7.2                               /usr/local/nagios/etc/nagios.cfg
#####
# HOST DEFINITION
#
#####

# Define a host for the local machine

define host {

    use          linux-server ; Name of host template
    ; This host definition is (or inherits) from the "linux-server" template

    host_name    linuxserver
    alias        linuxserver
    address      52.207.253.18
}

#####

# HOST GROUP DEFINITION

^G Help           ^O Write Out      ^W Where Is      ^K Cut          ^T Exit
^X Exit          ^R Read File      ^\ Replace       ^U Paste         ^J Ju

```

**Note - Here replace hostname with linuxserver**

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

**Add the following line to the nagios.cfg file**

cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/

```

# 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/

```

**After making the changes in nagios.cfg file now check validate the file by typing the following command in the terminal.**

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts#
```

**Now restart the service by using this command**

```
service nagios restart
```

```

root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# service nagios restart
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 17:36:35 UTC; 19s ago
     Docs: https://www.nagios.org/documentation
 Process: 1870 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1872 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1874 (nagios)
   Tasks: 8 (limit: 1130)
  Memory: 3.0M (peak: 3.2M)
    CPU: 24ms
   CGroup: /system.slice/nagios.service
           ├─1874 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─1875 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1876 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1877 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1878 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1879 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/etc/nagios.cfg
           ├─1880 /usr/local/nagios/libexec/check_ping -H 52.207.253.18 -w 3000.0,80% -c 5000.0,100% -p 5
           └─1881 /usr/bin/ping -n -U -w 30 -c 5 52.207.253.18

Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: core query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: echo service query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: help for the query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Registry request: name=Core Worker 1875,pid=1875
lines 1-26

```

**Now using this command update the apt repository of ubuntu (linux-client), install gcc, nagios-nrpe-server and nagios-plugin sudo apt update -y sudo apt install gcc -y**

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

**Now open nrpe.cfg file and add the ip address of the nagios host as shown. To open the nrpe.cfg file copy this command.**

```

# Supported.
#
# Note: The daemon only does rudimentary checking
# address. I would highly recommend adding entries
# file to allow only the specified host to connect
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running
#       as a module in Apache.
allowed_hosts=127.0.0.1,54.167.169.0

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE
# to specify arguments to commands that are executed
# if the daemon was configured with the --enable-command
# option.

```

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

**Now restart nrpe server by using this command**

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

Now, check nagios dashboard, you should see linuxserver up and running, if not

The screenshot shows the Nagios Core 4.4.6 dashboard with the following key components:

- Current Network Status:** Last Updated: Wed Oct 2 18:47:25 UTC 2024. Nagios® Core™ 4.4.6 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:** Up: 1, Down: 0, Unreachable: 0, Pending: 0. All Problems: 0, All Types: 1.
- Service Status Totals:** Ok: 6, Warning: 1, Unknown: 0, Critical: 1, Pending: 0. All Problems: 2, All Types: 8.
- Status Summary For All Host Groups:** Host Group: Linux Servers (linux-servers). Host Status Summary: 1 UP. Service Status Summary: 6 OK, 1 WARNING, 1 UNHANDLED, 1 CRITICAL, 1 UNHANDLED.
- Left Sidebar (Current Status):** Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems, Reports, System.
- Left Sidebar (Reports):** Availability, Trends (Legacy), Alerts, Notifications, Event Log.
- Left Sidebar (System):** Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration.

Name : **Himesh Pathai**

Class : **D15A**

Roll No. : **35**

## **EXPERIMENT - 11**

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

### **Theory:**

#### **AWS Lambda**

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). Users of AWS Lambda create functions, self-contained applications written in one of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner. The Lambda functions can perform any kind of computing task, from serving web pages and processing streams of data to calling APIs and integrating with other AWS services.

The concept of “serverless” computing refers to not needing to maintain your own servers to run these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure for you. And so “serverless” doesn’t mean that there are no servers involved: it just means that the servers, the operating systems, the network layer and the rest of the infrastructure have already been taken care of so that you can focus on writing application code.

#### **Features of AWS Lambda**

- AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
- It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
- You don’t need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
- AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.

## Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket. And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

## Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served, a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container with our function to serve those requests.

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code. AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

## Stateless Functions

The above execution model makes Lambda functions effectively stateless. This means that every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event.

However, due to the optimization noted above, the actual Lambda function is invoked only once per container instantiation. Recall that our functions are run inside containers. So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.

For example, the `createNewDbConnection` method below is called once per container instantiation and not every time the Lambda function is invoked. The `myHandler` function on the other hand is called on every invocation.

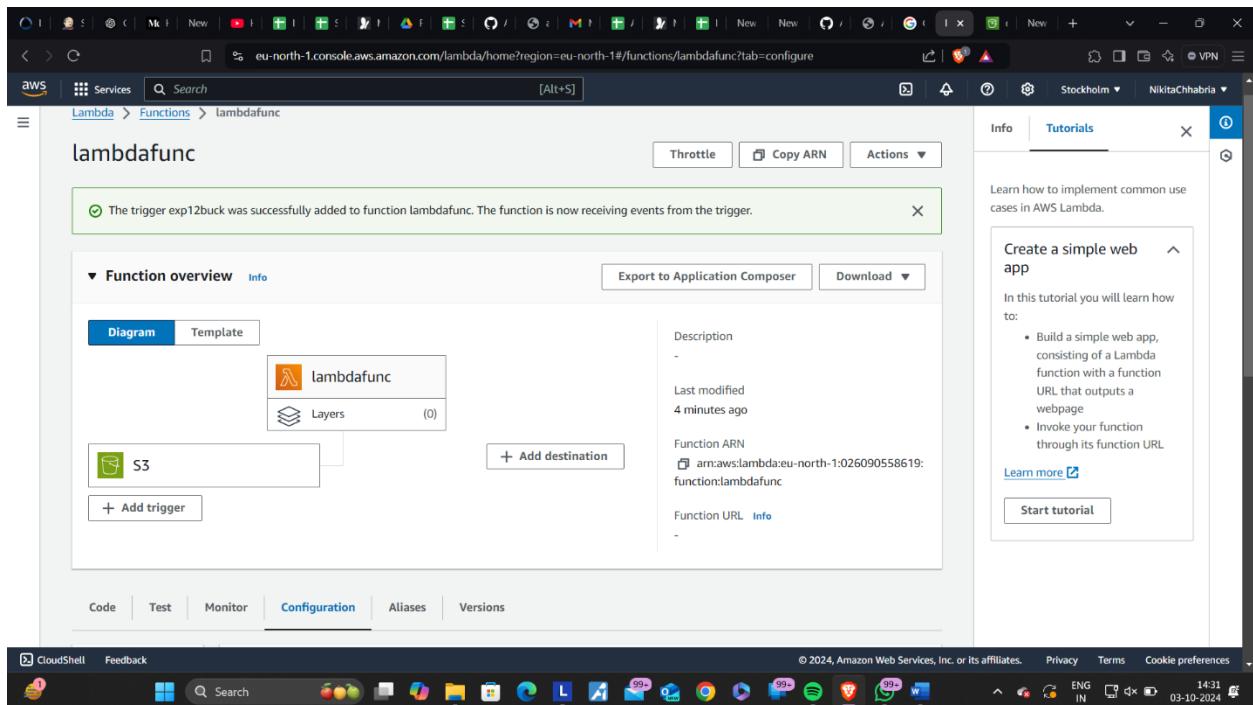
## Common Use Cases for Lambda

Due to Lambda's architecture, it can deliver great benefits over traditional cloud computing setups for applications where:

1. Individual tasks run for a short time;
2. Each task is generally self-contained;
3. There is a large difference between the lowest and highest levels in the workload of the application.

Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events. For example, you could employ Lambda to do some work every time an item in DynamoDB is created or updated, thus making it a good fit for things like notifications, counters and analytics.



### Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

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

Test event action

Create new event     Edit saved event

Event name

himeshbuck

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

Event sharing settings

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

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

Template - optional

hello-world

Template - optional

hello-world

Event JSON

Format JSON

```
1 [ {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 } ]
```

aws Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl+P)

lambda\_function

Environment

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

Cloud9

Code Test Monitor Configuration Aliases Versions

General configuration Triggers Permissions Destinations Function URL Environment variables Tags VPC

General configuration Info

Description - Memory 128 MB Ephemeral storage 512 MB

Timeout 0 min 3 sec

AWS Compute Optimizer Opt in to see memory recommendations for your Lambda functions. [View details](#)

Edit

aws Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9

Basic settings Info

Description - optional

Memory Info Your function is allocated CPU proportional to the memory configured. 128 MB Set memory to between 128 MB and 10240 MB.

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

Timeout 0 min 1 sec

Execution role Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

Use an existing role

Create a new role from AWS policy templates

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

service-role/mypythonlambdafunction-role-kexdw12k

View the mypythonlambdafunction-role-kexdw12k role on the IAM console.

Cancel Save

Feedback Looking for language selection? Find it in the new Unified Settings

83°F Haze

Code Test Monitor Configuration Aliases Versions

Code source Info

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

Upload from

Environment

lambda\_function

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO: Implement this function
5     new_string="Hey,This is Kajal and this is my AWS Lambda Function!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10
```

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO: Implement this function
5     new_string="Hey,This is Kajal and this is my AWS Lambda Function!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10
```

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

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

Test event action

Create new event       Edit saved event

Event name

mytestevent

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

Event sharing settings

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

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

Template - optional

hello-world

Event JSON

```

1 < [{
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 }]
```

Format JSON

Code source [Info](#)

File Edit Find View Go Tools Window [Test](#) Deploy Changes not deployed

Upload from

lambda\_function Environment

Go to Anything (Ctrl-P)

lambda\_function.py

Execution results

Test Event Name komalevent

Status: Succeeded Max memory used: 33 MB Time: 1.90 ms

Response

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

Function Logs

```
START RequestId: 3c07521e-c0c3-4431-afca-991cd27e7724 Version: $LATEST
END RequestId: 3c07521e-c0c3-4431-afca-991cd27e7724
REPORT RequestId: 3c07521e-c0c3-4431-afca-991cd27e7724 Duration: 1.90 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 33 MB
```

Request ID

3c07521e-c0c3-4431-afca-991cd27e7724

## Conclusion:

AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database updates, and scales based on the demand.

Name : **Himesh Pathai**

Class : **D15A**

Roll No. : **35**

## **EXPERIMENT NO - 12**

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

### **Theory:**

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

### **Workflow:**

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

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

#### **2. Create the Lambda Function:**

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

#### **3. Set Up Permissions:**

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

#### **4. Configure S3 Trigger:**

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

#### **5. Test the Setup:**

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

## Outcomes:

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. The 'General configuration' section is visible, showing the AWS Region set to 'Europe (Stockholm) eu-north-1'. The 'Bucket type' dropdown is open, with 'General purpose' selected (indicated by a blue border). Other options like 'Directory' are also shown. The 'Bucket name' field contains 'exp12buck'. Below it, a note states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'.

The screenshot shows the configuration page for the Lambda function 'lambdafunc'. In the 'Function overview' section, it is noted that the trigger 'exp12buck' was successfully added. The 'S3' trigger is listed under 'Destinations'. The 'Configuration' tab is selected at the bottom. On the right side, there is a 'Tutorials' sidebar with a 'Create a simple web app' section, which includes a 'Start tutorial' button.

```
lambda_function ✘ Environment Var ✘ +
```

```
1 import json
2
3 def lambda_handler(event, context):
4     # Extract bucket name and object key from the event
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     # Log a message
9     print(f"An Image has been added to the bucket {bucket_name}: {object_key}")
10
11    return {
12        'statusCode': 200,
13        'body': json.dumps('Log entry created successfully')
14    }
15
```

The screenshot shows the AWS Lambda console interface. The main area displays the code for a Lambda function named 'lambda\_function'. The function uses Python and triggers on S3 events, specifically for new objects in a bucket named 'exp12buck'. A sidebar on the right provides a tutorial titled 'Create a simple web app', which includes steps for building a Lambda function with a function URL and invoking it.

eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc?tab=configure

aws Services Search [Alt+S]

Code Test Monitor Configuration Aliases Versions

General configuration Triggers

Triggers (1) Info

Trigger S3: exp12buck arn:aws:s3:::exp12buck

Find triggers C Fix errors Edit Delete Add trigger

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

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

Learn more Start tutorial

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 14:31 ENG IN 03-10-2024

The screenshot shows three stacked browser windows demonstrating the integration of AWS Lambda with S3. The top window displays the AWS S3 console after a file upload, showing a success message and a summary table. The middle window shows the CloudWatch Logs console for the Lambda function, displaying log events related to the file upload. The bottom window shows the AWS Lambda function configuration page.

**S3 Upload Success:**

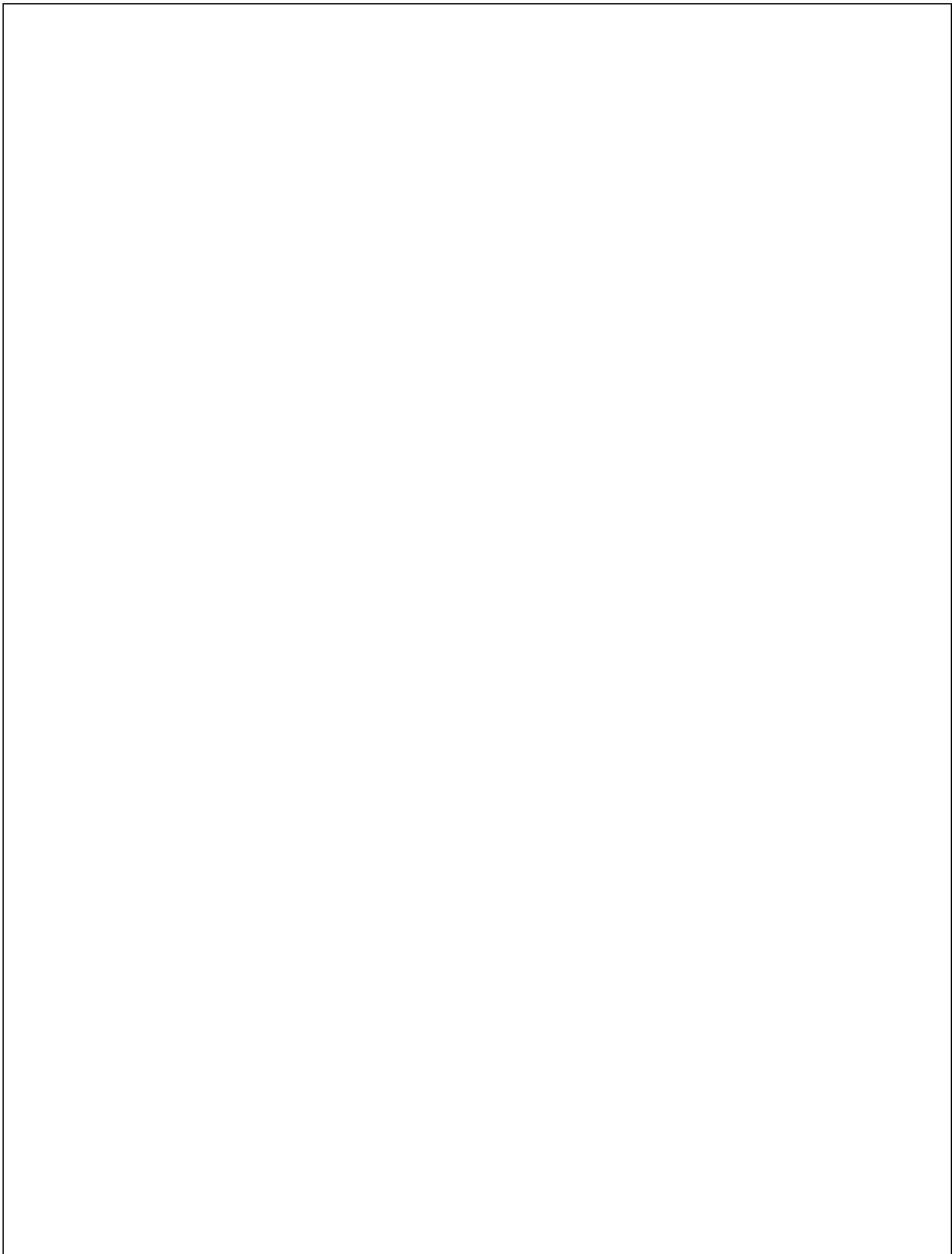
Destination	Status	Details
s3://exp12buck	Succeeded	1 file, 41.0 KB (100.00%)
	Failed	0 files, 0 B (0%)

**CloudWatch Log Events:**

Timestamp	Message
2024-10-03T14:32:51.304+05:30	INIT START Runtime Version: nodejs:20.v39 Runtime Version ARN: arn:aws:lambda:eu-north-1::runtime:ad9b28ae231dfc4c3325e183024ccb4d9de1aa14796d98295f898140041...
2024-10-03T14:32:51.454+05:30	START RequestId: abbd7b5f-9a9c-419a-8490-ed167f46eb05 Version: \$LATEST
2024-10-03T14:32:51.467+05:30	An image has been added to the bucket lambdafunc: image.png
2024-10-03T14:32:51.468+05:30	REPORT RequestId: abbd7b5f-9a9c-419a-8490-ed167f46eb05 Duration: 12.70 ms Billed Duration: 13 ms Memory Size: 128 MB Max Memory Used: 62 MB Init Duration: 14...

## Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket.



## Ad DevOps Assignment -1

Q  
of  
65

- A) Use S3 bucket and hls video streaming.  
→ To use Amazon S3 bucket for video streaming, we need to use S3 buckets as a container and CloudFront as a Content Delivery Network (CDN).

Step 1:- Set up Amazon S3 bucket.

- 1) Search for S3 on the Services section. Click on it, then click on it. Then click on Create Bucket. This will direct you to the Bucket creation page. Here, give a name to your bucket. It is better to block all public access so that unauthorized people do not use the video.

Maintaining the other options as default, click on Create Bucket.

- 2) The bucket has been created. Now we need to add our video in the bucket. For that, click on the name of the bucket, this will redirect you to the Objects screen which shows the objects of your bucket. Click on Upload.

Then select Add files (you can also drag and drop your file). An mp4 extension file is needed as we need to host a video. And the upload.

Step 2:- Set up CloudFront

- 1) As the video is being uploaded. Search for CloudFront on the Services tab & open it in a new tab.
- 2) On the left pane, under Security, you will find Origin Access. Click on it, then click on Identities (legacy). Click on Create Origin Access Identity. Now, go back to Distribution on the left pane & click on Create.

A CloudFront distribution -

- 3) Here, in the origin field, select the S3 bucket where video is uploaded. Under Origin access, select access identities. Here, Under Origin access identities, the identity that you have created. Under Bucket, Select Yes, update bucket policy.

### Step 3: Accessing the hosted video

- 1) Once the distribution is deployed, copy the domain of your distribution.
- 2) Now, go to the S3 bucket and click on its name. Click on the name of the video you have uploaded. You will find a key, copy that.
- 3) Combine the domain name of the distribution and of the video to make your final link of the video that is streamed.  
~~<domain name of distribution> / <key of video>~~

### Q2] Discuss BMW and hotstar case studies using AWS.

→ **BMW:** Overview: BMW uses Amazon Web Services (AWS) to build and scale its connected car platform. AWS provides services like real-time traffic updates, remote vehicle diagnostics. AWS enables BMW to handle millions of data requests daily from over 20 million connected vehicles, leveraging tools like Amazon Elastic Kubernetes Service (EKS) for scalability and Amazon Kinesis for real-time data analysis.

## BMW case study using AWS:

### Overview:

BMW utilizes Amazon Web Services (AWS) to power its connected car platform, delivering a suite of digital services such as real-time traffic updates, remote vehicle control, and diagnostic capabilities.

### Key Points:

#### 1 Scalability:

- BMW leverages AWS's data services, such as Amazon S3 for storage, Amazon EMR for big data processing, and Amazon Athena for querying, to analyze enormous amounts of vehicle & user data.

#### 2 Data Analytics:

- BMW leverages AWS's data services, such as Amazon S3 for storage, Amazon EMR for big data processing, and Amazon Athena for querying, to analyze enormous amounts of vehicle & user data.

#### 3 Security:

- Security is paramount in BMW's architecture, especially when dealing with sensitive vehicle and user data. AWS Identity & Access Management (IAM) & AWS Key Management Service (KMS) are employed to securely manage access controls & encryption.

#### 4 Supply Chain Transparency (PastChain Platform):

- To improve traceability & transparency in its supply chain, BMW has built the PastChain platform on Amazon EKS & blockchain technology.

5.

### Cost Optimization:

- BMW uses Amazon EC2 Auto Scaling to adjust compute capacity based on real-time demand, preventing overprovisioning & reducing operational costs.

6.

### Employee Upskilling & Innovation:

- AWS plays a pivotal role in training BMW's workforce. BMW aims to upskill 5,000 engineers & certify 2,000 employees in cloud services like machine learning & data analytics.

~~These~~

Through AWS, BMW accelerates the development of electric and autonomous vehicles & enhance digital customer services, aligning with its long-term strategy for connected, efficient, & sustainable automotive.

~~HOTSTAR:~~

~~Overview:~~ Disney+ Hotstar is a popular Indian subscription streaming service owned by Star India, a subsidiary of The Walt Disney Company India. It offers two paid plans: VIP, focusing on domestic content & sports, and Premium, featuring international content from platforms like HBO & Showtime.

1. Amazon Route 53

The name itself suggests that at port no. 53, the AWS provides the DNS services to its application. It easily and effectively connects the EC2 instances or Amazon S3 buckets, & it also provides the routing information to the outer side of the infrastructure. This makes AWS more user friendly.

## 2. Amazon ELB

Provides Scalability & reliability to computing capacity in AWS cloud which makes less to hardware & more to developing & deploying application on the cloud.

## 3. Amazon CloudFront

Low latency & high transfer speed of 5700 Gbps for transfer is somehow possible through CloudFront as it proxies content delivery Network Services (CDN).

## 4. Amazon S3

Storing the data & fetching them as per need is the advantage of the AWS services.

Q3) Why is Kubernetes & advantages & disadvantages of Kubernetes. Explain How today uses Kubernetes.

→ Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, & management of containerized applications. It is designed to manage large clusters of containers efficiently.

### Key features:

1. Automation: Automates deployment, Scaling and operation of application containers across clusters of hosts.
2. Portability: Works across on-premises, hybrid or cloud environments, ensuring flexibility in infrastructure choices.
3. Self-Healing: Detects & replaces failed containers, reschedules them automatically.
4. Service Discovery: Facilitates easy discovery of services within a cluster without hard-coding network settings.

5

Load Balancing: automatically distributes network traffic across the different containers.

Advantages:

1. Scalability: easily scales applications horizontally based on real-time demand.
2. Portability: allows applications to be moved seamlessly between environments (e.g., from testing to production).
3. Resilience: offers self-healing capabilities, replacing failed containers without downtime.

Disadvantages of Kubernetes:

1. Complexity: Kubernetes has a steep learning curve & is difficult to set up and manage, especially for beginners.
2. Overhead: can introduce overhead in terms of computing operational costs, requiring more resources for orchestration.
3. Security Management: while powerful, Kubernetes requires strong expertise to ensure proper security configurations.

Adidas

In recent years, the adidas team was happy with its software choices from a technology perspective—but accessing all the tools was a problem.

Challenge: In recent years, the adidas team was happy with its software choices from a technology perspective—but accessing all of the tools was a problem.

Solution: To improve the process, "we started from a developer point of view" & looked for ways to shorten the time it took to get a project up and running.

running and into the adidas infrastructure, says Senior Director of Platform Engineering Fernando Corrigo. They found the soil with containerization, agile development, continuous delivery, & a cloud native platform that includes Kubernetes & Prometheus.

- Kubernetes, an open-source container orchestration platform, has provided several benefits to Adidas:
  - Scalability & flexibility: Kubernetes enables Adidas to scale its application & service dynamically to meet fluctuating demand. During high-traffic events like product releases or major sports events, the platform automatically allocates additional resources to ensure smooth operations.
  - Faster Development & Deployment: Kubernetes simplifies the deployment & management of containerized applications. This allows Adidas to develop & release new features & updates more rapidly.
  - High Availability & Reliability: Kubernetes ensures high availability by automatically distributing workloads across multiple containers & nodes. This means Adidas can provide uninterrupted online shopping experiences to customers worldwide.
  - Resource Efficiency: With Kubernetes, Adidas can optimize resource utilization. Containers share the underlying infrastructure efficiently, reducing waste & resource consumption.

~~→~~ **Q1 C D:** Kubernetes integrates seamlessly with CI/CD. Adidas can auto-mate the testing & deployment of code, ensuring that new features & update are delivered quickly & reliably.

**Impact:** just six months after the project began, one of the adidas e-commerce site was running on Kubernetes. Load time for the e-commerce site was reduced. Releases went from every 4-6 weeks to 3-4 times.

**Conclusion:** Kubernetes is a powerful & flexible platform for container orchestration that can simplify the management, & scaling of your applications while enhancing their reliability & resource efficiency. Its extensive ecosystem & active community make it a compelling choice for modern application deployment management.

~~a n)~~ what are Nagios & explain how Nagios works in F-services?

→ Nagios is an open-source monitoring system that enables organizations to monitor their IT infrastructure, including servers, network devices, & applications. It helps ensure systems are running smoothly by providing real-time monitoring, alerting & reporting capabilities.

\* How Nagios is used in F-services

1. **Infrastructure Monitoring:** Nagios continuously monitors servers, databases & network devices in an environment to ensure they are operational. This includes checking the availability of web servers, databases,

and other critical components.

2. Performance Monitoring: It tracks various performance metrics such as CPU load, memory usage, disk space, & network traffic.
  3. Alerting & Notifications: Nagios can send alerts via email, SMS, or other communication methods when it detects issues.
  4. Service Monitoring: It monitors specific services such as web applications, APIs, and email servers to ensure they are functioning correctly.
  5. Log Monitoring - Nagios can analyze log files for error messages or anomalies, providing insight into potential issues within the e-services.
  6. Integration with Other Tools: Nagios can be integrated with other monitoring and management tools, allowing for comprehensive oversight of the IT environment and enabling more sophisticated alerting & reporting mechanisms.
  7. Customizable Dashboards: It provides dashboards that give a visual overview of the monitored infrastructure, helping teams to quickly assess the health of their e-services.
- By using Nagios, e-service provider can ensure high availability & performance of their applications, enhancing user satisfaction & trust in their services.

2.

~~Advance DevOps Assignment 2~~

DATE:

Create a REST API with serverless framework.

Creating REST API with serverless framework is an efficient way to deploy serverless applications that can scale automatically without managing servers in Serverless framework: A powerful tool that deployment of services and serverless applications across various cloud providers such as AWS, Azure and Google cloud.

iii) Serverless architecture: This design model allows developers to build applications without worrying about underlying infrastructure, enabling focus on code & business logic.

iv) REST API: Representational State Transfer is architecture style for designing network applications.

Steps for Creating REST API for serverless framework

1) Install Serverless framework:

You start by installing serverless framework globally using node package manager (npm). This allow you to manage serverless applications directly from your terminal.

2) Creating a Node in Serverless Project :

A directory is created for your project, where you will initialize a serverless service (project). This service will house all your lambda functions configurations and cloud resources. Using -tlo command serverless create you set up template for AWS Node.js microservices that will eventually deploy to AWS lambda.

- 3) Project Structure:-  
 The project scaffold creates essential files like handled.js (which contains code for lambda functions) and serverless.yml.
- 4) Create a REST API Resource:  
 In the serverless.yml file you define function that handles part requests of HTTP.
- 5) Deploy the Service:  
 With the 'sls deploy' command serverless framework packages your application, uploads necessary resources to AWS and set up the infrastructure.
- 6) Testing the API: Once deployed you can test REST API using tools like curl or Postman by making post request to generated API.
- 7) Storing data in Dynamodb: To store submitted candidate data you integrate AWS DYNAMO DB as a database.
- 8) Adding more functionalities: Adding functionalities like 'list all candidates', 'get candidates by ID'.
- 9) AWS IAM Permissions:  
 You need to ensure that serverless framework is given right permissions to interact with AWS resources like Dynamodb.
- 10) Monitoring and Maintenance  
 After deployment serverless framework provides service information like deployed endpoints, API key, log streams.
- Q2 Case Study for SonarQube  
 Creating your own profile in SonarQube for testing

DATE:

Project quality. Use SonarQube to analyse your GitHub code. Install sonarQube in your SonarQube to analyse Java IntelliJ IDE and analyse Java code. Analyze Python project with SonarQube.

SonarQube is an open source platform used for continuous inspection of quality. It detects bugs, code smells and security vulnerabilities in project across various programming languages.

## 17 Profile Creation in SonarQube:

Quality profiles in SonarQube are essential configurations that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'Sonar Way'. Profiles come built-in for all languages. Custom profiles can be created by copying or extending existing ones. Copying creates an independent profile, while extending inherit rules from parent profile & reflects future changes automatically. You can activate or deactivate rules, prioritize certain rules and configure parameters to tailor profile to specific projects. Permissions to manage quality profile are restricted to users with administrative privileges. SonarQube allows for the comparison of two profiles to check for differences in activated rules and users can track changes via event log. Quality profiles to specific projects permissions to can also be imported from other instances via backup and restore.

To ensure profiles include now rule its important to check against updated built in profiles or use sonarqube rules page.

### 27) Using Sonarcloud to analyze Github code:

Sonarcloud is cloud-based counterpart of Sonar Cube that integrates directly with Github, BitBucket, Azure and GitHub repositories. To get started with sonarcloud via Github sign up via sonarcloud product page and connect your Github organizations or personal account. Once connected, sonarcloud merges your Github setup with each project corresponding to Github repos. After setting up the organization choose subscription plan (free for public repos). Next import repositories into your Sonarcloud organizations where each Github repo becomes a Sonarcloud project. Define 'new code' to focus on recent changes and choose between automatic analysis or CI-based analysis. Automatic analysis happens directly in Sonarcloud while CI based analysis integrates with your build process once the analysis is complete results can be viewed in both Sonarcloud and Github including security impact issue.

### 37) Sonarlint in Java IDE :

Sonarlint is an IDE that performs on-the-fly code analysis as you write code. It helps developers in the development environment.

such as IntelliJ idea or Eclipse. To set it up install the sonarlint plugin, configure the connection with SonarQube or SonarCloud and Select the Project Profile to analyse Java code. This approach ensures immediate feedback on code quality. Promoting clean & maintainable code from beginning.

#### 4) Analyzing Python Projects with SonarQube:

SonarQube supports Python test coverage reporting but it requires third party tool like Coverage. Py to generate the coverage report. To enable coverage adjust your build process so that coverage tools run before Sonar scanner and ensures report file is saved in diff. path.

For set up, you can use ~~PYTEST~~, and coverage. Py to configure and run test. In your tox.ini include configurations for Pytest and coverage to generate coverage report in XML format. The build process can also be automated using GitHub Actions, which install dependencies runs tests and invokes SonarQube scan. Ensure report in Cobertura XML format and place where scanner can access it.

#### 5) Analyzing Node.js Projects with SonarQube

For Node.js Project SonarQube can analyse JavaScript and TypeScript code. Similar to the Python setup, you can configure SonarQube to analyse node.js projects by installing the appropriate plugin and using Sonar Scanner to

Scan the Projects. SonarQube will check the code against Industry Standard rules and best practices, flagging issues related to security vulnerabilities, bugs and performance optimization.

3. At a large organization, your centralized operation team may get many repetitive infrastructure requests. You can use Terraform to build a self-service infrastructure model that lets product team manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying & managing services in your organizations, allowing teams to efficiently ~~deploy services~~ in compliance with your organization's practices. Terraform cloud can also integrate with ticketing system like ServiceNow to automatically generate new infrastructure requests.

Implementing a 'self-service' infrastructure model using Terraform can transform how large organization manage their infrastructure independently. Organization can enhance efficiency, reduce bottlenecks and ensure compliance with established needs.

- The need for self-service infrastructure:  
In large organization, centralized operations teams often face an overwhelming number of

repetitive requests. This can lead to delay in service delivery and frustration among product teams who need to move quickly. A self-service model allows teams to provision and manage their infrastructure without relying on the operations team for every request.

#### • Benefits of using Terraform.

##### 1. Modularity & Reusability :

- Terraform modules encapsulate standard configurations for various infrastructure components (e.g. networks, databases, compute resources).
- Teams can reuse these modules across different projects, reducing redundancy and minimizing the risk of error.

##### 2. Standardizations

- By defining best practices within modules, organizations can ensure that all deployments comply with internal policies and standards.
- This consistency helps maintain security and operational integrity across the organization.

##### 3. Increased Efficiency :

- Product teams can deploy services quickly by using pre-defined modules, significantly reducing the time spent on infrastructure setup.
- This allows teams to focus on developing features rather than managing infrastructure.

##### 4. Integration with ticketing systems

- Terraform cloud can integrate with ticketing systems like ServiceNow.

- to automate the generation of infrastructure requests.
- This integration streamlines workflows by allowing teams to initiate requests directly from their ticketing platform, reducing manual intervention.

### → Implementation steps

1. Identify Infrastructure components
  - Begin by identifying which components of your infrastructure can be modularized (e.g. VPCs, security groups, load balancers)
2. Develop Terraform modules.
  - Create reusable modules that define the desired configuration and resources.
  - Ensure each module includes input variables for customizations and outputs for integration with other modules.
3. Establish Governance and Best Practices:
  - Define guidelines for module usage, versioning and documentation to ensure clarity and maintainability.
  - Encourage teams to contribute to module development and share improvements.
4. Testing and Validation.
  - Implement a testing framework to validate module functionality before deployment.
  - Best practices for module management.
  - Utilize the Terraform Registry.
  - Leverage existing community modules from the

- Terraform Registry: Use the Terraform Registry to avoid reinventing solutions and ensure adherence to best practices.
- Version control: Implement versioning for your modules to track changes over time. This helps manage dependencies effectively and minimizes disruptions during updates.
- Documentation: Maintain comprehensive documentation for each module, including usage examples, input/output descriptions, and any dependency details.
- Encourage collaboration: Foster a culture of collaboration by sharing modules across teams. This promotes consistency in deployments and facilitates knowledge within the organization.  
~~By adopting a self-service infrastructure model with Terraform organizations can empower product teams to efficiently manage their own infrastructure while ensuring compliance with established standards. This approach not only streamlines processes but also enhances agility in responding to changing business needs. Ultimately, it leads to a more responsible IT environment that supports innovation and growth within the organization.~~

J.