

## Advance Devops:1

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

### **Theory:**

#### **Amazon EC2 (Elastic Compute Cloud)**

Amazon Elastic Compute Cloud (EC2) is a cloud service that offers scalable compute power in the cloud. It enables developers to easily deploy and manage virtual servers, known as instances, providing flexibility in handling varying workloads.

#### **Key Concepts of EC2:**

1. **Instances:** Instances are virtual servers hosted on Amazon's EC2 platform, designed to run applications in the cloud. You can launch instances as needed and scale them up or down depending on your application's requirements.
2. **Amazon Machine Image (AMI):** An AMI is a blueprint used to create an instance. It includes a pre-configured operating system, application server, and application software, allowing for consistent and rapid deployment of instances.
3. **Instance Types:** EC2 offers a range of instance types, each optimized for specific use cases, such as compute-intensive tasks, memory-intensive applications, or storage-heavy operations. Each type varies in its combination of CPU, memory, storage, and network capacity.
4. **Elastic IP Addresses:** Elastic IPs are static IP addresses that can be associated with an EC2 instance. They are useful for maintaining a consistent IP address for your instance, even if the underlying instance changes.
5. **Security Groups:** Security groups function as virtual firewalls, controlling inbound and outbound traffic to and from your EC2 instances. You can define rules based on IP addresses, ports, and protocols to manage the traffic securely.
6. **Auto Scaling:** Auto Scaling enables automatic adjustments in the number of EC2 instances in response to current demand, ensuring that your application performs optimally while minimizing costs.

#### **Amazon S3 (Simple Storage Service)**

Amazon S3 is an object storage service offering high scalability, availability, security, and performance. It is ideal for storing and retrieving large amounts of data from anywhere on the web.

#### **Key Concepts of S3:**

1. **Buckets:** Buckets serve as containers in S3, where all objects are stored. Each bucket is unique across AWS and is used to organize and manage the stored data.
2. **Objects:** Objects are the basic units stored in S3. They consist of the data itself, metadata, and a unique key (identifier). Objects can range from documents to videos, images, and other file types.
3. **Keys:** A key is a unique identifier for an object within a bucket. Each object in S3 has a key that allows you to access and manage it.
4. **Versioning:** S3 supports versioning, which allows you to maintain multiple versions of an object within a bucket. This feature protects against accidental deletions or overwrites by preserving older versions.
5. **Access Control:** S3 provides various mechanisms for controlling access to data, such as bucket policies, access control lists (ACLs), and IAM policies. These tools ensure that only authorized users can access and manage your data.
6. **Lifecycle Management:** S3 lifecycle policies help automate the transition of objects between different storage classes or delete them after a specified period, optimizing storage costs over time.
7. **Storage Classes:** S3 offers multiple storage classes designed for different access patterns, such as S3 Standard for frequently accessed data, S3 Intelligent-Tiering for cost optimization, and S3 Glacier for long-term archival storage.

## AWS Cloud9

AWS Cloud9 is a cloud-based integrated development environment (IDE) that allows you to write, run, and debug code directly from a web browser. It supports various programming languages and comes pre-configured with essential tools and libraries, streamlining the development process.

### Key Features of AWS Cloud9:

1. **Cloud-Based IDE:** AWS Cloud9 provides a fully-featured development environment accessible through a browser, eliminating the need for local IDE installation and setup.
2. **Collaborative Development:** Cloud9 enables real-time collaboration, allowing multiple users to work on the same project simultaneously. Features like chat and simultaneous editing make it ideal for teamwork and pair programming.
3. **Pre-configured Environment:** Cloud9 comes with pre-configured tools and libraries for various programming languages and frameworks, enabling developers to start coding immediately without the hassle of setting up the environment.
4. **Seamless Integration with AWS Services:** Cloud9 integrates seamlessly with AWS services like EC2, S3, and Lambda, allowing developers to deploy and manage applications directly from the IDE.
5. **Terminal Access:** Cloud9 provides full terminal access to the underlying instance, enabling developers to run shell commands and manage their environment directly.

## Implementation:

### EC2 Instance Creation and Static Site Hosting

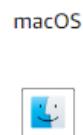
1. **Log in to your AWS account:** Begin by logging into your AWS account through the AWS Management Console.
2. **Navigate to EC2:** From the dashboard, click on the EC2 service to access the instance management console.
3. **Launch an Instance:** Click on the "Launch Instance" button to begin setting up a new EC2 instance. Choose an appropriate AMI, select an instance type, configure instance details, and add storage as needed.
4. **Configure Security Group:** Set up a security group to control inbound and outbound traffic to your instance, ensuring that your application is accessible but secure.
5. **Assign an Elastic IP (Optional):** To maintain a static IP address, you can allocate an Elastic IP and associate it with your instance.
6. **Connect and Deploy:** Once the instance is running, connect to it via SSH or the AWS Cloud9 terminal, and deploy your static website or application by uploading files to the web server directory.
7. **Access Your Site:** After deployment, your static site will be accessible via the instance's public IP address or domain name associated with the Elastic IP.

## Implementation :

EC2 Instance Creation and static site hosting

### 1) Login to your AWS account

The screenshot shows the "Launch an instance" wizard on the AWS EC2 console. The first step, "Name and tags", is selected. A "Name" field contains the value "SNEHAL". An "Add additional tags" link is visible. Below this, a section titled "Application and OS Images (Amazon Machine Image)" is expanded, showing a search bar with the placeholder "Search our full catalog including 1000s of application and OS images".

[Recents](#)[Quick Start](#)[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

##### Amazon Linux 2023 AMI

ami-0ae8f15ae66fe8cda (64-bit (x86), uefi-preferred) / ami-0e36db3a3a535e401 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

#### Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

#### Architecture

64-bit (x86)

#### Boot mode

uefi-preferred

#### AMI ID

ami-0ae8f15ae66fe8cda

Verified provider

#### ▼ Instance type [Info](#) | [Get advice](#)

##### Instance type

t2.micro  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

#### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

##### Key pair name - required

Select

Create new key pair

**▼ Instance type** [Info](#) | [Get advice](#)

**Instance type**

t2.micro      Free tier eligible

Family: t2    1 vCPU    1 GiB Memory    Current generation: true  
 On-Demand Windows base pricing: 0.0162 USD per Hour  
 On-Demand SUSE base pricing: 0.0116 USD per Hour  
 On-Demand RHEL base pricing: 0.026 USD per Hour  
 On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

**▼ Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

[Create new key pair](#)

EC2 > Instances > Launch an instance

Success  
Successfully initiated launch of instance (i-0c3f04602fb41afa7)

**▼ Launch log**

Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

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

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

[Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[Clear filters](#)

<input checked="" type="checkbox"/> Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/> SNEHAL	i-0c3f04602fb41afa7	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1b

**i-0c3f04602fb41afa7 (SNEHAL)**

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

**▼ Instance summary** [Info](#)

Instance ID i-0c3f04602fb41afa7 (SNEHAL)	Public IPv4 address 52.90.12.26   <a href="#">open address</a>	Private IPv4 addresses 172.31.36.47
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-90-12-26.compute-1.amazonaws.com   <a href="#">open address</a>

## Static Website hosting using EC2:

Follow the steps and then run the commands

```
See "man sudo_root" for details.

ubuntu@ip-172-31-41-61:~$ sudo su
root@ip-172-31-41-61:/home/ubuntu# sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-41-61:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [294 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [68.1 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Metadata [3768 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [250 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [108 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9412 B]
```

i-Odda4db17f16307ec (Snehal)

PublicIPs: 54.162.220.58 PrivateIPs: 172.31.41.61

```
Reading package lists... Done
root@ip-172-31-41-61:/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-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 libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 53 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
```

```
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-172-31-41-61:/home/ubuntu$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-08-18 12:30:09 UTC; 30s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2442 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 5.4M (peak: 5.7M)
      CPU: 40ms
     CGroup: /system.slice/apache2.service
             ├─2442 /usr/sbin/apache2 -k start
             ├─2445 /usr/sbin/apache2 -k start
             └─2446 /usr/sbin/apache2 -k start

Aug 18 12:30:09 ip-172-31-41-61 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 18 12:30:09 ip-172-31-41-61 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-61:/home/ubuntu$ cd /var/www/html
root@ip-172-31-41-61:/var/www/html$
```

i-0dda4db17f16307ec (Snehal)

Public IPs: 54.162.220.58 Private IPs: 172.31.41.61

[EC2](#) > [Security Groups](#) > sg-0e7811c687e701e30 - launch-wizard-7

[Actions ▾](#)

### Details

Security group name  
[launch-wizard-7](#)

Security group ID  
[sg-0e7811c687e701e30](#)

Description  
[launch-wizard-7 created 2024-08-18T11:25:33.225Z](#)

VPC ID  
[vpc-08963bc0f8afcd789](#)

Owner  
[608111999703](#)

Inbound rules count  
1 Permission entry

Outbound rules count  
1 Permission entry

[Inbound rules](#)

[Outbound rules](#)

[Tags](#)

[Inbound rules \(1\)](#)



[Manage tags](#)

[Edit inbound rules](#)

Search

< 1 >

## sg-0896d82a58154b33d - launch-wizard-9

Actions ▾

## Details

Security group name	Security group ID	Description	VPC ID
<a href="#">launch-wizard-9</a>	<a href="#">sg-0896d82a58154b33d</a>	<a href="#">launch-wizard-9 created 2024-08-18T12:21:13.480Z</a>	<a href="#">vpc-08963bc0f8afcd789</a>
Owner	Inbound rules count	Outbound rules count	
<a href="#">608111999703</a>	3 Permission entries	1 Permission entry	

[Inbound rules](#)[Outbound rules](#)[Tags](#)

## Inbound rules (3)

[Manage tags](#)[Edit inbound rules](#)

Security group name
<a href="#">launch-wizard-9</a>

Security group ID
<a href="#">sg-0896d82a58154b33d</a>

Description
<a href="#">launch-wizard-9 created 2024-08-18T12:21:13.480Z</a>

VPC ID
<a href="#">vpc-08963bc0f8afcd789</a>

Owner
<a href="#">608111999703</a>

Inbound rules count
3 Permission entries

Outbound rules count
1 Permission entry

[Inbound rules](#)[Outbound rules](#)[Tags](#)

## Outbound rules (1)

[Manage tags](#)[Edit outbound rules](#) [Search](#) [1](#)  

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-06dd7ee61f83e4e88	IPv4	All traffic	All

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 URL `54.162.220.58`. The page title is "Apache2 Default Page". On the left, there's the classic Ubuntu logo. In the center, the word "Ubuntu" is written in a large, lowercase, sans-serif font. To the right of "Ubuntu", there's an orange rectangular button with the white text "It works!". Below the logo and the word "Ubuntu", there's a paragraph of text explaining what this page represents. At the bottom of the page, there's a dark blue horizontal bar containing the text "Configuration Overview". Below this bar, there's a detailed description of the Apache2 configuration layout, mentioning files like `/etc/apache2/apache2.conf`, `/ports.conf`, `/mods-enabled`, `*.Load`, and `*.conf`, and `/conf-enabled`.

```
ubuntu@ip-172-31-40-177:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-40-177:~$ mkdir Snehal
ubuntu@ip-172-31-40-177:~$ cd Snehal
ubuntu@ip-172-31-40-177:~/Snehal$ git clone https://github.com/Snehal1490102/dynamic-web-hosting.git
Cloning into 'dynamic-web-hosting'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 15.04 KiB | 3.76 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@ip-172-31-40-177:~/Snehal$ ls
dynamic-web-hosting
ubuntu@ip-172-31-40-177:~/Snehal$ cd dynamic-web-hosting/
ubuntu@ip-172-31-40-177:~/Snehal/dynamic-web-hosting$ ls
README.md index.js package-lock.json package.json
ubuntu@ip-172-31-40-177:~/Snehal/dynamic-web-hosting$ npm i
Command 'npm' not found, but can be installed with:
sudo apt install npm
ubuntu@ip-172-31-40-177:~/Snehal/dynamic-web-hosting$ sudo apt install npm
Reading package lists... Done
```



[Home](#) | [Products](#) | [Services](#) | [Contact](#)

Nykaa is your go-to destination for all things beauty and personal care. Explore our wide range of products and services.

## Our Products

### Beauty Products



### Skincare Solutions



### Fragrance



### Hair Care



## STATIC WEBSITE HOSTING USING S3 BUCKET:

### Step1: Create bucket

[Amazon S3](#) > [Buckets](#) > Create bucket

### Create bucket Info

Buckets are containers for data stored in S3.

**General configuration**

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

Bucket type Info

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

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

Bucket name Info

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

Copy settings from existing bucket - optional

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

► Advanced settings

**i** After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

Amazon S3 > Buckets

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

[View Storage Lens dashboard](#)

[General purpose buckets](#)

[Directory buckets](#)

General purpose buckets (2) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

[Copy](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">elasticbeanstalk-us-east-1-608111999703</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 8, 2024, 15:22:13 (UTC+05:30)
<a href="#">snehal-123-aws</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 17, 2024, 22:59:32 (UTC+05:30)

## Edit static website hosting [Info](#)

### Static website hosting

#### Static website hosting

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

[Edit](#)

Static website hosting

Enabled

Hosting type

Bucket hosting

**Bucket website endpoint copied**

After you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://snehal-123-aws.s3-website-us-east-1.amazonaws.com>

**i** 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.

index.html

## Block public access (bucket settings)

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

Edit

### Block all public access

⚠ Off

► Individual Block Public Access settings for this bucket

## Step 2: Add resources

### Files and folders (9 Total, 972.7 KB)

Remove

Add files

Add folder

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	sale.png	-	image/png
<input type="checkbox"/>	personal-care.png	-	image/png
<input type="checkbox"/>	Loreal.png	-	image/png
<input type="checkbox"/>	logo.png	-	image/png
<input type="checkbox"/>	kay.png	-	image/png
<input type="checkbox"/>	fragrances.png	-	image/png
<input type="checkbox"/>	dotandkey.png	-	image/png
<input type="checkbox"/>	beauty.png	-	image/png
<input type="checkbox"/>	index.html	-	text/html

⌚ Upload succeeded

View details below.

### Files and folders (9 Total, 972.7 KB)

Find by name

< 1 >

Name	Folder	Type	Size	Status	Error
<a href="#">sale.png</a> ↗	-	image/png	767.3 KB	<span>⌚ Succeeded</span>	-
<a href="#">personal-car...</a> ↗	-	image/png	46.6 KB	<span>⌚ Succeeded</span>	-
<a href="#">Loreal.png</a> ↗	-	image/png	19.3 KB	<span>⌚ Succeeded</span>	-
<a href="#">logo.png</a> ↗	-	image/png	6.4 KB	<span>⌚ Succeeded</span>	-
<a href="#">kay.png</a> ↗	-	image/png	18.7 KB	<span>⌚ Succeeded</span>	-
<a href="#">fragrances.p...</a> ↗	-	image/png	33.8 KB	<span>⌚ Succeeded</span>	-
<a href="#">dotandkey.p...</a> ↗	-	image/png	23.1 KB	<span>⌚ Succeeded</span>	-
<a href="#">beauty.png</a> ↗	-	image/png	50.5 KB	<span>⌚ Succeeded</span>	-
<a href="#">index.html</a> ↗	-	text/html	7.0 KB	<span>⌚ Succeeded</span>	-

## Step 3 : Provide public access

### Edit Block public access (bucket settings) Info

#### Block public access (bucket settings)

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

##### Block all public access

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

###### Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

###### Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

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

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

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

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

#### Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

##### ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

##### ACLs enabled

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

**⚠️** We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.



##### Enabling ACLs turns off the bucket owner enforced setting for Object Ownership

Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.

I acknowledge that ACLs will be restored.

**⌚ Successfully edited public access**  
View details below.

#### Summary

Source  
s3://snehal-123-aws

Successfully edited public access  
**9 objects, 972.7 KB**

Failed to edit public access  
0 objects

[Failed to edit public access](#) | [Configuration](#)

#### **⌚ Failed to edit public access (0)**

Find objects by name

Name	▲	Folder	▼	Type	▼	Last modified	▼	Size	▼	Error	▼
------	---	--------	---	------	---	---------------	---	------	---	-------	---

No objects failed to edit

The screenshot shows the Nykaa website homepage. At the top left is the Nykaa logo in pink. To its right is the text "Welcome to Nykaa". Below the logo is a navigation bar with links for "Home", "Products", "Services", and "Contact". A sub-headline reads, "Nykaa is your go-to destination for all things beauty and personal care. Explore our wide range of products and services." Below this is a section titled "Our Products" with four categories: "Beauty Products", "Skincare Solutions", "Fragrance", and "Hair Care", each accompanied by a small image of related products.

## EC2 Dynamic Site Hosting:

```
root@ip-172-31-55-145:/home/ubuntu/dynamic/dyanamic_site# npm i
[REDACTED] : reify:define-data-property: http fetch GET 200 https://registry.npmjs.org/define-data-property
added 93 packages, and audited 94 packages in 3s

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

found 0 vulnerabilities
root@ip-172-31-55-145:/home/ubuntu/dynamic/dyanamic_site# npm start

> hosting-dynamic-website@1.0.0 start
> nodemon index.js

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server is running on port 3000
```

The screenshot shows a browser window displaying a dynamic website. The address bar shows the URL "54.237.31.193:3000". The page content is a single line of text: "Hey this is Dynamic Website."



## Cloud 9 IDE Site Hosting:

### Step 1: Create Environment

AWS Cloud9 > Environments > Create environment

#### Create environment Info

**Details**

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

Description – *optional*  Limit 200 characters.

Environment type Info Determines what the Cloud9 IDE will run on.

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

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

AWS Cloud9 > Environments > SnehalEnv

#### SnehalEnv

[Delete](#) [Open in Cloud9](#)

Details		<a href="#">Edit</a>
Name SnehalEnv	Owner ARN <code>arn:aws:sts::608111999703:assumed-role/voclabs/user3402712=PATIL_SHRAVANI_ANIL</code>	Status <span style="color: green;">Ready</span>
Description -	Number of members 1	Lifecycle status <span style="color: green;">Created</span>
Environment type EC2 instance		

## Step 2 : Open the Environment IDE

The screenshot shows the AWS Cloud9 Environments page. At the top, there are two informational messages: one about creating the environment and another about AWS Toolkits. Below this is a breadcrumb navigation bar: AWS Cloud9 > Environments. The main area is titled "Environments (1)" and contains a table with one row. The columns are: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The environment listed is "SnehalEnv" (Status: Open), which is an EC2 instance connected via Secure Shell (SSH). The owner is listed as "Owner" with the ARN: arn:aws:sts::608111999703:assumed-role/voclabs/user3402712=PATIL\_SHRAVANI\_ANIL.

The screenshot shows the AWS Cloud9 development environment. On the left is a file browser showing "SnehalEnv - /home" with files "cloud9.html" and "README.md". The main area has a title "Welcome" and sub-section "Developer Tools". Below this is a large "AWS Cloud9" header with the sub-header "Welcome to your development environment". A "Getting started" sidebar offers "Create File" and "Upload Files...". At the bottom is a terminal window titled "bash - [ip-172-31-42-231.x]" showing the command "voclabs:~/environment \$".

## Step 3: Add the code and preview the website

The screenshot shows the AWS Cloud9 development environment. On the left is a file browser showing "SnehalEnv - /home" with files "cloud9.html" and "README.md". The main area has a code editor window for "cloud9.html" containing the following HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile Page</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .navbar {
            background-color: #333;
            overflow: hidden;
        }
        .navbar a {
            float: left;
            display: block;
            color: white;
            text-align: center;
            padding: 14px 20px;
            text-decoration: none;
        }
        .navbar a:hover {
            background-color: #ddd;
            color: black;
        }
        .container { max-width: 800px; margin: auto; padding: 20px; text-align: center; }
        h1 { font-size: 2.5em; margin: 20px 0 26.44 HTML Spaces: 4; color: #555; margin: 10px 0; }
    </style>

```

To the right is a browser window titled "[B] /cloud9.html" showing the preview of the website. The page has a dark header with "Home", "About", and "Contact" links. The main content area has a title "Snehal Patil" and subtitle "Information Technology Student". It also includes a "At VESIT" section and a "Passionate about coding and technology. I am currently pursuing my degree in Information Technology at VESIT, where I am learning to build innovative solutions and".

## User details

User name

Snehal

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

**Provide user access to the AWS Management Console - optional**

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Console password

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

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

## Permissions options

Add user to group

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

Copy permissions

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

Attach policies directly

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



### Get started with groups

Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

[Create group](#)

### ► Set permissions boundary - optional

[Cancel](#)

[Previous](#)

[Next](#)

## Create user group



Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name

Enter a meaningful name to identify this group.

awsgrp

Maximum 128 characters. Use alphanumeric and '+,-,.@-' characters.

## Permissions policies (952)

[Create policy](#)

Filter by Type

 Search

All ty... ▾

&lt; 1 2 3 4 5 6 7 ... 48 &gt;



Policy name	Type	User	Description
<a href="#">AdministratorAccess</a>	AWS managed	Permis...	Provides full access to AWS services an...
<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative permis...
<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative permis...
<a href="#">AlexaForBusinessD...</a>	AWS managed	None	Provide device setup access to AlexaFo...
<a href="#">AlexaForBusinessF...</a>	AWS managed	None	Grants full access to AlexaForBusiness ...
<a href="#">AlexaForBusinessG...</a>	AWS managed	None	Provide gateway execution access to A...
<a href="#">AlexaForBusinessLi...</a>	AWS managed	None	Provide access to Lifesize AVS devices
<a href="#">AlexaForBusinessP...</a>	AWS managed	None	Provide access to Poly AVS devices
<a href="#">AlexaForBusinessR...</a>	AWS managed	None	Provide read only access to AlexaForB...
<a href="#">AmazonAPIGatewa...</a>	AWS managed	None	Provides full access to create/edit/dele...
<a href="#">AmazonAPIGateway...</a>	AWS managed	None	Provides full access to create/edit/dele...

[Cancel](#)[Create user group](#)

## EXPERIMENT NO:2

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

### Theory:

#### **AWS Elastic Beanstalk**

**AWS Elastic Beanstalk** is a versatile Platform as a Service (PaaS) offering from Amazon Web Services (AWS) that simplifies the deployment and management of applications in the cloud. It abstracts much of the underlying infrastructure complexity, allowing developers to focus on building and maintaining their applications while AWS takes care of provisioning resources like EC2 instances, load balancers, and databases.

#### **Key Features of Elastic Beanstalk:**

##### **1. Simplified Application Deployment:**

- Elastic Beanstalk streamlines the deployment process by automatically handling the setup, configuration, and maintenance of the infrastructure required to run your application. This allows developers to concentrate on writing and refining their code rather than managing servers.

##### **2. Broad Language and Framework Support:**

- Elastic Beanstalk supports multiple programming languages and frameworks, such as Java, .NET, Node.js, Python, Ruby, PHP, Go, and Docker. This flexibility makes it easy for developers to deploy applications built with their preferred technologies.

##### **3. Automatic Resource Scaling:**

- Elastic Beanstalk includes automatic scaling capabilities, dynamically adjusting the number of instances running your application based on current demand. This ensures that your application remains responsive and cost-efficient during traffic fluctuations.

##### **4. Robust Health Monitoring:**

- The platform continuously monitors the health of your application and provides detailed metrics and logs. It can automatically replace any unhealthy instances, ensuring high availability and reliability of your application.

##### **5. Multiple Environment Support:**

- Elastic Beanstalk allows you to manage different environments, such as development, staging, and production. You can deploy updates to one environment without impacting others, providing a controlled and organized deployment process.

#### **AWS CodeBuild**

**AWS CodeBuild** is a fully managed continuous integration (CI) service that compiles your source code, runs tests, and produces deployable artifacts. It automates the build process, ensuring consistent compilation and testing across various environments.

#### **Key Features of AWS CodeBuild:**

##### **1. Managed Build Infrastructure:**

- With AWS CodeBuild, there's no need to manage your own build servers. AWS handles all infrastructure management, allowing you to focus on developing and testing your applications.
- 2. Automatic Scaling:**
- CodeBuild automatically scales to meet your needs, handling multiple builds simultaneously. This ensures that even during peak times, your builds are processed quickly and efficiently.
- 3. Customizable Build Environments:**
- You can define custom build environments using Docker images, providing the flexibility to tailor the build environment to meet the specific needs of your application.
- 4. Seamless AWS Integration:**
- CodeBuild integrates effortlessly with other AWS services, such as CodePipeline, CodeCommit, and S3, enabling you to create a complete CI/CD pipeline that automates your entire build and deployment process.
- 5. Cost-Effective Pricing:**
- CodeBuild uses a pay-as-you-go pricing model, charging you only for the build time you consume. This makes it an economical solution for automating your build processes.

## Deploying on S3 Using AWS CodePipeline

**AWS CodePipeline** is a fully managed continuous delivery service that automates the steps required to release your software, from building and testing to deployment. By integrating with services like CodeBuild and S3, CodePipeline provides a streamlined process for deploying applications.

### Key Steps to Deploy on S3 Using AWS CodePipeline:

- 1. Source Code Retrieval:**
  - The pipeline begins with a source stage, where the source code is retrieved from a repository such as AWS CodeCommit, GitHub, or an S3 bucket. This source code is then used in subsequent stages for building and deploying the application.
- 2. Build and Package with CodeBuild:**
  - In the build stage, AWS CodePipeline triggers CodeBuild to compile the source code, run tests, and package the application. The output of this process is a deployable artifact stored in an S3 bucket.
- 3. Deployment to S3:**
  - In the deployment stage, CodePipeline automatically deploys the artifacts generated during the build stage to an S3 bucket. The S3 bucket can be configured to host a static website or store files accessed by your application.
- 4. Automation and Notifications:**
  - CodePipeline can be configured to automatically trigger builds and deployments when changes are made to the source code repository. Additionally, it can send notifications via Amazon SNS to keep you informed about the status of your pipeline.
- 5. Pipeline Monitoring and Logging:**
  - CodePipeline provides tools for monitoring and logging the progress of each stage in your pipeline. This allows you to quickly identify and address any issues that arise during the build or deployment process, ensuring a smooth release cycle.

## Implementation :

### Elastic Beanstalk

#### Step 1: create environment

The screenshot shows the Amazon Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and user information (Stockholm, Snehal\_Patil). Below the header, the page title is "Amazon Elastic Beanstalk" with the subtitle "End-to-end web application management.". A descriptive text block explains that Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications. To the right, there are two main sections: "Get started" (with a "Create application" button) and "Pricing" (stating there's no additional charge). On the left, a "Get started" section provides instructions for uploading code. The overall layout is clean and modern.

This screenshot shows the "Configure environment" step of the Elastic Beanstalk wizard. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled "Configure environment" and contains two sections: "Environment tier" and "Application information". Under "Environment tier", it says "Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications." with options for "Web server environment" (selected) and "Worker environment". Under "Application information", the "Application name" is set to "Snehal123".

## Environment information Info

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

Environment name

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

Domain

Environment description

## 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 values.

### Configuration presets

- Single instance (free tier eligible)
- Single instance (using spot instance)
- High availability
- High availability (using spot and on-demand instances)
- Custom configuration

[Cancel](#)

[Next](#)

[IAM](#) > [Dashboard](#)

## IAM Dashboard

### Security recommendations 1

#### ⚠ Add MFA for root user

Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.

[Add MFA](#)

#### ✓ Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves security.

### IAM resources

Resources in this AWS Account

User groups

Users

Roles

Policies

Identity providers



### AWS Account

#### Account ID

825765388229

#### Account Alias

[Create](#)

#### Sign-in URL for IAM users in this account

<https://825765388229.signin.aws.amazon.com/console>

### Quick Links

[My security credentials](#)

Roles (2) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

 Search

< 1 > ⚙️

<input type="checkbox"/>	Role name	▲ Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linked)	-

Roles Anywhere Info
Manage

Authenticate your non AWS workloads and securely provide access to AWS services.



Access AWS from your non AWS workloads



X.509 Standard



Temporary credentials

Step 1

[Select trusted entity](#)

Step 2

[Add permissions](#)

Step 3

[Name, review, and create](#)
Add permissions InfoPermissions policies (3/946) Info

Choose one or more policies to attach to your new role.

Filter by Type

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess...</a>	AWS managed	Grants account administrative permissions...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkC...</a>	AWS managed	Provide the instance in your custom platf...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkE...</a>	AWS managed	AWS Elastic Beanstalk Service policy for H...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalk...</a>	AWS managed	This policy is for the AWS Elastic Beanstal...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalk...</a>	AWS managed	Provide the instances in your multicontain...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkR...</a>	AWS managed	Grants read-only permissions. Explicitly all...

**Set permissions boundary - optional**

Cancel Previous Next

Role aws-Snehal-ec2-role created.

View role

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
<a href="#">aws-Snehal-ec2-role</a>	AWS Service: ec2	2023-09-18 10:30:00
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	2023-09-18 10:30:00
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	2023-09-18 10:30:00

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Manage

## Step 2 : add your Ec2 key pair and instance profile

Step 1  
[Configure environment](#)

Step 2  
**Configure service access**

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

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

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

Step 6  
[Review](#)

### Configure service access Info

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

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

**Existing service roles**  
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-Snehal-ec2-role ▼

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

Choose a key pair ▼

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

aws-Snehal-ec2-role ▼

**View permission details**

Cancel Skip to review Previous Next

## Review [Info](#)

### Step 1: Configure environment

[Edit](#)

#### Environment information

Environment tier	Application name
Web server environment	Snehal123
Environment name	Application code
Snehal123-env	Sample application

Platform  
arn:aws:elasticbeanstalk:eu-north-1::platform/Python  
3.11 running on 64bit Amazon Linux 2023/4.1.3

### Step 2: Configure service access

[Edit](#)

#### Service access [Info](#)

### Step 2: Configure service access

[Edit](#)

#### Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::825765388229:role/aws-Snehal-ec2-role	-Snehal-ec2-role

### Step 3: Set up networking, database, and tags

[Edit](#)

#### Networking, database, and tags [Info](#)

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

No options configured

#### Tags

## Step 4: Configure instance traffic and scaling

[Edit](#)

### Instance traffic and scaling [Info](#)

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

#### Instances

IMDSv1

Deactivated

#### Capacity

Environment type	Fleet composition	On-demand base
------------------	-------------------	----------------

Single instance	On-Demand instance	0
-----------------	--------------------	---

On-demand above base	Capacity rebalancing	Scaling cooldown
----------------------	----------------------	------------------

0	Deactivated	360
---	-------------	-----

Processor type	Instance types	AMI ID
----------------	----------------	--------

x86_64	t3.micro,t3.small	ami-030d0ebd08fe18778
--------	-------------------	-----------------------

## Step 5: Configure updates, monitoring, and logging

[Edit](#)

### Updates, monitoring, and logging [Info](#)

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

#### Monitoring

System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
--------	--------------------------------------	---

enhanced	—	—
----------	---	---

Log streaming	Retention	Lifecycle
---------------	-----------	-----------

Deactivated	7	false
-------------	---	-------

#### Updates

Managed updates	Deployment batch size	Deployment batch size type
-----------------	-----------------------	----------------------------

Activated	100	Percentage
-----------	-----	------------

Command timeout	Deployment policy	Health threshold
-----------------	-------------------	------------------

**Platform software**

Lifecycle	Log streaming	NumProcesses
false	Deactivated	1
NumThreads	WSGIPath	Proxy server
15	application	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

**Environment properties**

Key	Value
PYTHONPATH	/var/app/venv/staging-LQM1test/bin

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

✓ Environment successfully launched.

Elastic Beanstalk > Environments > Snehal123-env

### Snehal123-env [Info](#)

[Actions ▾](#)
[Upload and deploy](#)

**Environment overview**

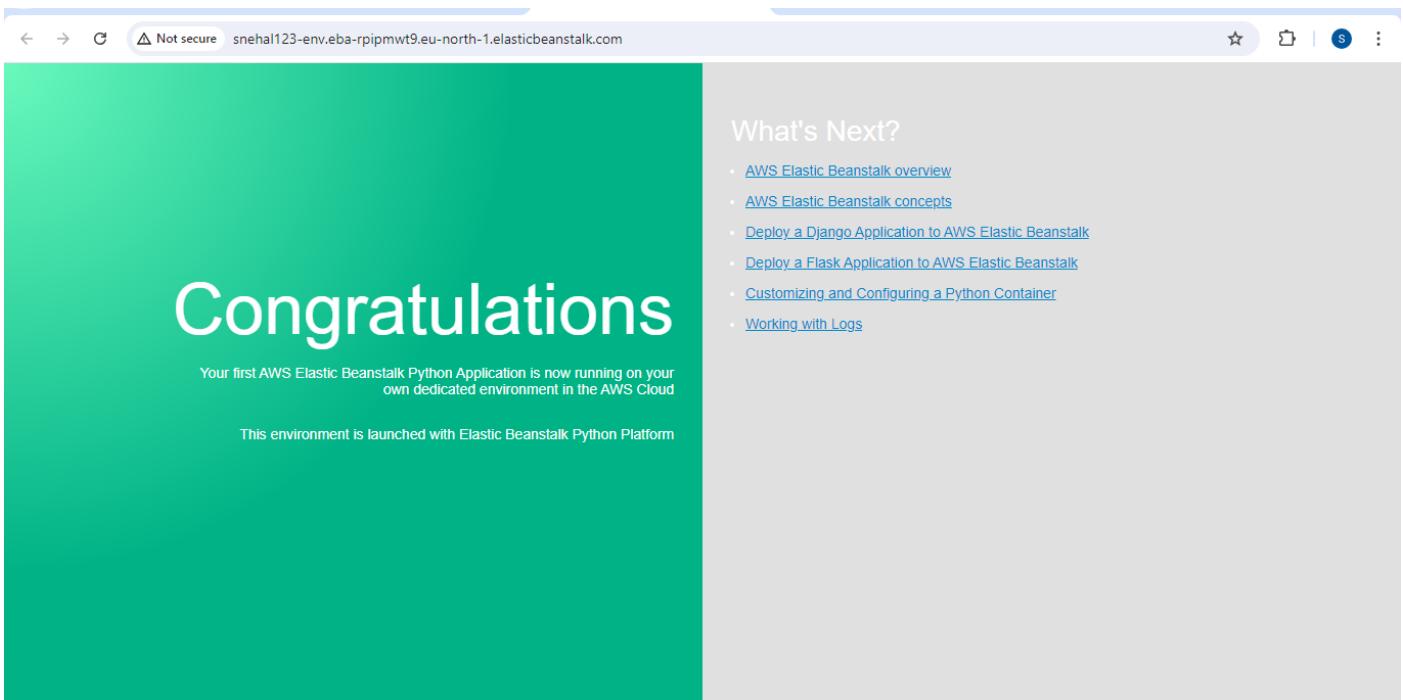
Health	Environment ID
<span style="color: orange;">⚠ Warning</span>	e-nm5tuux6pa
Domain	Application name
Snehal123-env.eba-rpipmwt9.eu-north-1.elasticbeanstalk.com <a href="#">Edit</a>	Snehal123

**Platform** [Change version](#)

Platform	Python 3.11 running on 64bit Amazon Linux 2023/4.1.3
Running version	-
Platform state	<span style="color: green;">✓ Supported</span>

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

### Step 3: Beanstalk environment is created



### Step 4 : add security config and review all settings

#### Configure instance traffic and scaling - *optional* Info

**▼ Instances** Info

Configure the Amazon EC2 instances that run your application.

**Root volume (boot device)**

Root volume type

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

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

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

## EC2 security groups

Select security groups to control traffic.

### EC2 security groups (6)

Filter security groups



<input type="checkbox"/>	Group name	▲	Group ID	▼	Name	▼
<input type="checkbox"/>	default		sg-0a1301dad692be19a			
<input type="checkbox"/>	launch-wizard-1		sg-08f567b852d69d909			
<input type="checkbox"/>	launch-wizard-2		sg-09df1f87b10436fac			
<input type="checkbox"/>	launch-wizard-3		sg-03ee263ce302b1e04			
<input type="checkbox"/>	launch-wizard-4		sg-01d4af8f20286924d			
<input checked="" type="checkbox"/>	launch-wizard-5		sg-0bed7f87bc908ca03			

## ▼ Capacity Info

## Pipeline Creation:

Step 1 : click on create pipeline and give name

> [CodePipeline](#) > [Pipelines](#) > Create new pipeline

## Choose pipeline settings Info

Step 1 of 5

### Pipeline settings

#### Pipeline name

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

pipeline\_Snehal1

No more than 100 characters

#### Pipeline type

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

#### Execution mode

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

Superseded

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

Queued (Pipeline type V2 required)

## Step 2 : Add Your github account and add the file to add to pipeline deployment

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

### Add source stage Info

Step 2 of 5

#### Source

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

GitHub (Version 1)

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

Connected

ⓘ You have successfully configured the action with the provider. X

ⓘ **The GitHub (Version 1) action is not recommended**  
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository  X

Branch  X

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

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

AWS CodePipeline  
Use AWS CodePipeline to check periodically for changes

Cancel Previous Next

### Step 3 : Add deploy config choosing the elastic beanstalk

Step 1  
[Choose pipeline settings](#)

Step 2  
[Add source stage](#)

Step 3  
[Add build stage](#)

Step 4  
**Add deploy stage**

Step 5  
[Review](#)

## Add deploy stage Info

Step 4 of 5

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

### Deploy

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

AWS Elastic Beanstalk

**Region**

Europe (Stockholm)

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

Region

Europe (Stockholm)

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

SourceArtifact

No more than 100 characters

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

Snehal123

**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.

Snehal123-env

Configure automatic rollback on stage failure

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

#### Step 4 : review changes and submit

> [CodePipeline](#) > [Pipelines](#) > Create new pipeline

**Review** Info  
Step 5 of 5

**Step 1: Choose pipeline settings**

Pipeline settings	
Pipeline name	pipeline_Snehal1
Pipeline type	V2
Execution mode	QUEUED
Artifact location	codepipeline-eu-north-1-77720346183
Service role name	AWSCodePipelineServiceRole-eu-north-1-pipeline_Snehal1

#### Step 2: Add source stage

**Source action provider**

Source action provider	GitHub (Version 1)
PollForSourceChanges	false
Repo	Nykaa-E-commerce_css
Owner	Snehal490102
Branch	main

#### Step 3: Add build stage

## Step 2: Add source stage

Source action provider

Source action provider

GitHub (Version 1)

PollForSourceChanges

false

Repo

Nykaa-E-commerce\_css

Owner

Snehal490102

Branch

main

## Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

Snehal123

EnvironmentName

Snehal123-env

Configure automatic rollback on stage failure

Disabled

Cancel

Previous

Create pipeline

# pipeline\_Snehal

Notify ▾

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2 Execution mode: QUEUED

✓ Source Succeeded

Pipeline execution ID: [bd6f8320-0a79-412d-9825-256bb1ec64d0](#)

Source

[GitHub \(Version 1\)](#) ↗

✓ Succeeded - Just now

[b8307259](#) ↗

[View details](#)

[b8307259](#) ↗ Source: Update style.css

[Disable transition](#)

⌚ Deploy ⓘ In progress

[Disable transition](#)

⌚ Deploy ⓘ In progress

Pipeline execution ID: [bd6f8320-0a79-412d-9825-256bb1ec64d0](#)

Deploy

[AWS Elastic Beanstalk](#) ↗

⌚ In progress - Just now

[View details](#)

[b8307259](#) ↗ Source: Update style.css

**Deploy**

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

AWS Elastic Beanstalk ▾

**Region**

US East (N. Virginia) ▾

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

No more than 100 characters

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

Snehal123

**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.

Snehal123-Env

Configure automatic rollback on stage failure

**Step 6 : Check the deployed website at beanstalk link**

Not secure : snehal123-env.eba-riplpmwt9.eu-north-1.elasticbeanstalk.com

**NYKA**

Home | Products | Services | Contact

Nykaa is your go-to destination for all things beauty and personal care. Explore our wide range of products and services.

**Our Products**

**Beauty Products**

**Skincare Solutions**

**Fragrance**

### **Advanced DevOps 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.

##### **Overview of Kubernetes**

Kubernetes is an open-source container orchestration platform developed by Google, designed to automate the deployment, scaling, and management of containerized applications. It provides a robust infrastructure that supports microservices architecture, offering features such as self-healing, scaling, and zero-downtime deployments. Kubernetes can run on various environments, including public clouds (like AWS and Azure), private clouds, and bare metal servers.

##### **Kubernetes Architecture**

Kubernetes architecture is primarily composed of two main components: the Control Plane and the Data Plane.

##### **Control Plane**

The Control Plane manages the overall state of the Kubernetes cluster and includes several key components:

- **kube-apiserver:** The API server acts as the gateway for all interactions with the cluster, processing REST requests and managing the state of the cluster.
- **etcd:** A distributed key-value store that holds the configuration data and state of the cluster, ensuring consistency and availability.
- **kube-scheduler:** Responsible for assigning Pods to worker nodes based on resource availability and other constraints.
- **kube-controller-manager:** Manages controllers that regulate the state of the cluster, ensuring that the desired state matches the actual state.
- **cloud-controller-manager (optional):** Integrates with cloud provider APIs to manage resources specific to the cloud environment.

##### **Data Plane**

The Data Plane consists of the worker nodes that run the containerized applications. Each worker node includes:

- kubelet: An agent that ensures containers are running in Pods. It communicates with the Control Plane to receive instructions.
- kube-proxy: Maintains network rules and facilitates communication between Pods and services.
- Container Runtime: Software responsible for running containers, such as Docker or containerd.

## Core Concepts

Key concepts in Kubernetes include:

- Pods: The smallest deployable units in Kubernetes, which can contain one or more containers.
- Services: Abstracts a set of Pods, providing a stable network endpoint for accessing them.
- Deployments: Define the desired state for Pods and manage their lifecycle, including scaling and updates.

## Installing and Spinning Up a Kubernetes Cluster

To install and set up a Kubernetes cluster, follow these general steps:

1. Choose an Environment: Decide whether to deploy on local machines or a cloud platform. For cloud platforms, services like Google Kubernetes Engine (GKE), Amazon EKS, or Azure AKS can simplify the process.
2. Install Prerequisites: Ensure that you have the necessary tools installed, such as kubectl (the command-line tool for interacting with the cluster) and a container runtime.
3. Set Up the Control Plane: This can be done using tools like kubeadm, which helps bootstrap the cluster by initializing the Control Plane components.
4. Join Worker Nodes: Once the Control Plane is set up, you can join worker nodes to the cluster using the token generated during the initialization.
5. Deploy Applications: After the cluster is up and running, you can deploy your applications using YAML configuration files that define the desired state of your Pods and Services.

## Best Practices

When setting up a Kubernetes cluster, consider the following best practices:

- Resource Management: Define resource requests and limits for Pods to ensure efficient utilization of cluster resources.
- High Availability: Use multiple Control Plane nodes to avoid single points of failure.
- Networking: Implement network policies to secure communication between Pods and manage external access.
- Monitoring and Logging: Integrate monitoring tools and logging solutions to keep track of cluster performance and troubleshoot issues.

By understanding the architecture and following the installation steps and best practices, you can effectively manage a Kubernetes cluster, enabling efficient deployment and scaling of containerized applications.

## Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

The screenshot shows the AWS EC2 Instances page. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
worker-2	i-01d90b250aa03275c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f
Master	i-0adb85729938302cc	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f
worker-1	i-02fc4e862c7b6e71c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f

A modal window titled "Select an instance" is open at the bottom, listing the same three instances.

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

Now copy the ssh from the example and paste it on command prompt.

The screenshot shows the AWS Security Groups page for a specific security group. It displays two inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-08a6e73c0a0d652a6	SSH	TCP	22	Cust... <input type="button" value="Delete"/>	0.0.0.0/0 X
sgr-0155f08b28c2cec31	All traffic	All	All	Cust... <input type="button" value="Delete"/>	0.0.0.0/0 X

An "Add rule" button is visible at the bottom left.

## Connect to instance [Info](#)

Connect to your instance i-043b8a223eab93c13 (master) using any of these options

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

Instance ID

 [i-043b8a223eab93c13 \(master\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Snehalkey.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "Snehalkey.pem"`
4. Connect to your instance using its Public DNS:  
 `ec2-34-229-230-179.compute-1.amazonaws.com`

Example:

 `ssh -i "Snehalkey.pem" ubuntu@ec2-34-229-230-179.compute-1.amazonaws.com`

### Prerequisites:

Ensure the following requirements are met before starting the Kubernetes cluster setup:

- **Ubuntu OS:** Ubuntu Xenial or later is recommended.
- **sudo Privileges:** Administrative access to execute commands.
- **Instance Type:** Use t2.medium or higher for adequate resources.

### Commands to Execute on Both Master and Worker Nodes:

1. After the instances have been created, copy the text given in the example part of each of the three instances into git bash.

```
Windows@DESKTOP-I925HTO MINGW64 ~/Downloads
$ ssh -i "Snehalkey.pem" ubuntu@ec2-44-204-227-4.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-227-4.compute-1.amazonaws.com (44.204.227.4)' can't be established.
ED25519 key fingerprint is SHA256:IGnMIHNbyAzruaz0ospRsy1CcVXZPSJ1iGisJHkwTqE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-227-4.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep 18 09:48:04 UTC 2024

System load: 0.0	Processes: 114
Usage of /: 22.8% of 6.71GB	Users Logged in: 0
Memory usage: 5%	IPv4 address for enx0: 172.31.87.198
Swap usage: 0%	

```
ubuntu@ip-172-31-87-198:~$ sudo su
root@ip-172-31-87-198:/home/ubuntu# Yum install docker -y
Command 'Yum' not found, did you mean:
  command 'gum' from snap gum (0.13.0)
  command 'uum' from deb freewnn-jserver (1.1.1~a021+cvs20130302-7build1)
  command 'num' from deb quickcal (2.4-1)
  command 'zum' from deb perforate (1.2-5.3)
  command 'sum' from deb coreutils (9.4-2ubuntu2)
See 'snap info <snapname>' for additional versions.
```

---

Run the following commands on both the master and worker nodes to prepare them for kubeadm.

### 1. Disable Swap:

- Command: sudo swapoff -a
- Explanation: Disables swap memory, which is required by Kubernetes for proper functioning.

### 2. Create Configuration File for Kernel Modules:

- Commands:

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
```

```
overlay
```

```
br_netfilter
```

```
EOF
```

- Explanation: Loads necessary kernel modules for Kubernetes networking.

```
root@ip-172-31-87-198:/home/ubuntu# sudo swapoff -a
root@ip-172-31-87-198:/home/ubuntu# cat <<EOF | sudo tee /etc/modules-load.d/k8s
.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
```

---

### Configure Sysctl Parameters:

- Command:

```
bash
```

```
Copy code
```

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
```

```
net.bridge.bridge-nf-call-iptables = 1
```

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
net.ipv4.ip_forward = 1
```

```
EOF
```

- Explanation: Sets necessary network parameters for Kubernetes. These settings enable IP forwarding and ensure iptables rules are applied to bridge traffic.

```
root@ip-172-31-87-198:/home/ubuntu# cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
>
> ^C
root@ip-172-31-87-198:/home/ubuntu# cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
```

## Apply Sysctl Parameters:

- Command: sudo sysctl --system
- Explanation: Applies the sysctl parameters without rebooting the system.

```
root@ip-172-31-87-198:/home/ubuntu# sudo sysctl --system
* Applying /usr/lib/sysctl.d/10-apparmor.conf ...
* Applying /etc/sysctl.d/10-console-messages.conf ...
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
* Applying /etc/sysctl.d/10-map-count.conf ...
* Applying /etc/sysctl.d/10-network-security.conf ...
* Applying /etc/sysctl.d/10-ptrace.conf ...
```

## Install CRI-O Runtime:

- **Commands:**

bash

Copy code

```
sudo apt-get update -y
```

```
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg
```

```
sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/" | sudo tee /etc/apt/sources.list.d/cri-o.list
```

```
sudo apt-get update -y
```

```
sudo apt-get install -y cri-o
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable crio --now
```

```
sudo systemctl start crio.service
```

- **Explanation:** Installs and configures CRI-O, a container runtime compatible with Kubernetes.

```
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Fetched 15.0 MB in 1s (12.5 MB/s)

```

## Add Kubernetes APT Repository and Install Packages:

- **Commands:**

```
bash
```

```
Copy code
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update -y
```

```
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
```

```
sudo apt-get update -y
```

```
sudo apt-get install -y jq
```

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

```
sudo systemctl start kubelet
```

- **Explanation:** Adds the Kubernetes APT repository, installs Kubernetes tools (kubelet, kubectl, kubeadm), and ensures the kubelet service is running.

```
root@ip-172-31-87-198:/home/ubuntu# sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ " | sudo tee /etc/apt/sources.list.d/cri-o.list
deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /
```

```
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update -y
sudo apt-get install -y cri-o
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
6 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri
-o:/prerelease:/main/deb InRelease [1206 B]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri
-o:/prerelease:/main/deb Packages [2454 B]
Fetched 130 kB in 0s (282 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Recommended packages:
  kubernetes-cni
The following NEW packages will be installed:
  cri-o
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 19.9 MB of archives.
After this operation, 76.5 MB of additional disk space will be used.
Get:1 https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb cri-o 1.32.0~dev-
2.1 [19.9 MB]
```

```
root@ip-172-31-87-198:/home/ubuntu# sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service
root@ip-172-31-87-198:/home/ubuntu# echo "CRI runtime installed successfully"
CRI runtime installed successfully
root@ip-172-31-87-198:/home/ubuntu# curl -fsSL https://pkgs.k8s.io/core:/stable:
/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-
keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.
k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes
.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io
/core:/stable:/v1.29/deb/
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"

sudo apt-get update -y
sudo apt-get install -y jq
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri
```

```
root@ip-172-31-87-198:/home/ubuntu# sudo systemctl enable --now kubelet
sudo systemctl start kubelet
root@ip-172-31-87-198:/home/ubuntu# sudo kubeadm config images pull
I0918 09:59:24.968795    3577 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.29
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.29.9
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.29.9
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.29.9
[config/images] Pulled registry.k8s.io/kube-proxy:v1.29.9
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.11.1
[config/images] Pulled registry.k8s.io/pause:3.9
[config/images] Pulled registry.k8s.io/etcd:3.5.10-0
root@ip-172-31-87-198:/home/ubuntu# sudo kubeadm init
I0918 09:59:41.619600    3915 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.9
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
```

### **Commands to execute on master node:**

## 1. Pull Kubernetes Control Plane Images:

- Command: `sudo kubeadm config images pull`
  - Explanation: Downloads the required container images for the Kubernetes Control Plane components.

```
root@ip-172-31-87-198:/home/ubuntu# sudo kubeadm config images pull
I0918 09:59:24.968795      3577 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.29
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.29.9
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.29.9
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.29.9
[config/images] Pulled registry.k8s.io/kube-proxy:v1.29.9
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.11.1
[config/images] Pulled registry.k8s.io/pause:3.9
[config/images] Pulled registry.k8s.io/etcd:3.5.10-0
```

## 2. Initialize the Kubernetes Cluster:

- Command: `sudo kubeadm init`
  - Explanation: Initializes the Kubernetes Control Plane on the Master node

```
root@ip-172-31-87-198:/home/ubuntu# sudo kubeadm init
I0918 09:59:41.619600    3915 version.go:256] remote version is much newer: v1.3
1.0; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.9
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0918 09:59:41.935262    3915 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.10" of the container runtime is inconsistent with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-87-198 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local]
[certs] [172.31.87.198] [10.25.0.1 172.31.87.198]
```

## Configure kubectl

- Commands:

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

- Explanation: Sets up kubectl on the Master node to interact with the cluster.

```
|root@ip-172-31-87-198:/home/ubuntu# mkdir -p "$HOME/.kube"
|sudo cp -i /etc/kubernetes/admin.conf "$HOME/.kube/config"
|sudo chown "$(id -u)":"$(id -g)" "$HOME/.kube/config"
```

## Install Network Plugin (Calico):

- Command: kubectl apply -f <https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml>
- Explanation: Deploys Calico as the network plugin for the cluster.

```
root@ip-172-31-87-198:/home/ubuntu# mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
serviceaccount/calico-cni-plugin created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalic
```

---

## Generating Join Command for Worker Nodes:

- Command: kubeadm token create --print-join-command
- Explanation: Generates the command to join worker nodes to the Kubernetes cluster.

```
root@ip-172-31-87-198:/home/ubuntu# kubeadm token create --print-join-command
kubeadm join 172.31.87.198:6443 --token xqllei.w91ed3g6i8omszpr --discovery-token-ca-cert-hash sha256:750e1245a8b1c658b62daad553dbcd27c89ac2bb81aca75f1b686c4da2838885
root@ip-172-31-87-198:/home/ubuntu# kubectl get nodes
NAME      STATUS   ROLES      AGE      VERSION
ip-172-31-80-64  Ready    <none>    15s     v1.29.0
ip-172-31-81-208 Ready    <none>    49s     v1.29.0
ip-172-31-87-198 Ready    control-plane  3m53s   v1.29.0
root@ip-172-31-87-198:/home/ubuntu#
```

---

## Commands to execute on worker nodes:

### 1. Reset Pre-flight Checks:

- Command: sudo kubeadm reset
- Explanation: Resets the Kubernetes setup on worker nodes, useful for ensuring a clean state before joining.

```
root@ip-172-31-81-208:/home/ubuntu# sudo kubeadm reset pre-flight checks
W0918 10:01:40.885084    3738 preflight.go:56] [reset] WARNING: Changes made to
this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: yes
[preflight] Running pre-flight checks
W0918 10:01:46.735217    3738 removeetcdmember.go:106] [reset] No kubeadm config
, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]
```

---

## 2. Join the Cluster:

- Command: sudo kubeadm join 172.31.87.198:6443 --token xqllei.w91ed3g6i8omszpr --discovery-token-ca-cert-hash sha256:750e1245a8b1c658b62daad553dbcd27c89ac2bb81aca75f1b686c4da2838885 --v=5
- Explanation: Joins the worker node to the Kubernetes cluster using the join command generated from the Master node.

```
root@ip-172-31-81-208:/home/ubuntu# sudo kubeadm join 172.31.87.198:6443 --token xqllei.w91ed3g6i8omszpr --discovery-token-ca-cert-hash sha256:750e1245a8b1c658b62daad553dbcd27c89ac2bb81aca75f1b686c4da2838885 --v=5
I0918 10:02:52.014281    3760 join.go:413] [preflight] found NodeName empty; using OS hostname as NodeName
I0918 10:02:52.015029    3760 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/crio/crio.sock
[preflight] Running pre-flight checks
I0918 10:02:52.015127    3760 preflight.go:93] [preflight] Running general checks
I0918 10:02:52.015154    3760 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I0918 10:02:52.015223    3760 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0918 10:02:52.015235    3760 checks.go:104] validating the container runtime
I0918 10:02:52.034714    3760 checks.go:639] validating whether swap is enabled or not
I0918 10:02:52.034785    3760 checks.go:370] validating the presence of executable crioctl
```

## Verify Cluster Connection

### 1. Check Nodes:

- **Command:** kubectl get nodes
- **Explanation:** Verifies that all nodes (Master and Worker) have successfully joined the cluster and are in a ready state.

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get nodes
NAME      STATUS   ROLES      AGE      VERSION
ip-172-31-80-64  Ready    <none>    15s     v1.29.0
ip-172-31-81-208 Ready    <none>    49s     v1.29.0
ip-172-31-87-198 Ready    control-plane 3m53s   v1.29.0
root@ip-172-31-87-198:/home/ubuntu#
```

## ADVANCE DEVOPS EXP: 4

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

### Theory:

What is kubectl?

kubectl is the command-line interface (CLI) used to interact with a Kubernetes cluster. It allows users to manage cluster resources, deploy applications, inspect and manage cluster components, and much more. Using kubectl, you can communicate with the Kubernetes API server to issue commands and queries.

Common kubectl commands:

- kubectl get: View information about resources.
- kubectl describe: Detailed description of resources.
- kubectl create/apply: Create or update resources.
- kubectl delete: Delete resources.

kubectl plays a crucial role in the day-to-day operation of a Kubernetes cluster.

### Basic Concepts in Kubernetes

Before diving into the application deployment process, it's important to understand a few key Kubernetes objects:

1. **Pods:** The smallest deployable unit in Kubernetes. A pod encapsulates one or more containers (usually a single container) that share the same network namespace and storage.
2. **Deployments:** A Kubernetes resource that defines how to create and manage pods. It ensures the specified number of pod replicas are running at any given time and handles updates and rollbacks.
3. **Services:** An abstraction that defines how to access the pods. A service allows you to expose your pods to internal or external clients.
4. **ReplicaSets:** Ensures that a specified number of pod replicas are running at all times. It is managed by a Deployment, but can also be used independently.

## 1.1 Install prerequisites:

```
sudo apt-get update
```

```
sudo apt-get install -y apt-transport-https ca-certificates curl
```

```
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg
sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:7 https://packages.cloud.google.com/apt kubernetes-focal Release
  404 Not Found [IP: 172.253.122.100 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
curl is already the newest version (8.5.0-2ubuntu10.4).
apt-transport-https is already the newest version (2.7.14build2).
ca-certificates is already the newest version (20240203).
gpg is already the newest version (2.4.4-2ubuntu17).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
File '/etc/apt/keyrings/cri-o-apt-keyring.gpg' exists. Overwrite? (y/N) y
deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
```

### 1. Add the GPG key for Kubernetes:

```
sudo curl -fsLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
```

<https://packages.cloud.google.com/doc/apt-key.gpg>

```
root@ip-172-31-87-198:/home/ubuntu# sudo curl -fsLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/doc/apt-key.gpg
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
hash: https://packages.cloud.google.com/doc/apt-key.gpg: No such file or directory
```

### 2. Add the Kubernetes repository:

```
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
root@ip-172-31-87-198:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
```

## 1.2 Install kubectl:

Now install kubectl

Sudo apt-get update

Sudo apt-get install -y kubectl

```
w/ kubernetes.list ~ , do you want to
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update
sudo apt-get install -y kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease
Ign:6 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:7 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.122.102 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release
N: Updating from such a repository can't be done securely, and is therefore disabled by default
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
```

```
root@ip-172-31-87-198:/home/ubuntu# nano nginx-deployment.yaml
root@ip-172-31-87-198:/home/ubuntu# nano nginx-service.yaml
```

## Verifying the installation:

Kubectl version --client

```
*- command not found
root@ip-172-31-87-198:/home/ubuntu# kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

## Step 2: Deploying the Application on Kubernetes

### 2.1 Setting up Kubernetes Cluster

1. If you haven't already set up a Kubernetes cluster (e.g., with kubeadm), use minikube or any managed Kubernetes service (like EKS, GKE, etc.) to get a cluster running.
2. Once your cluster is ready, confirm that all the nodes are successfully connected and operational.

Command: kubectl get nodes

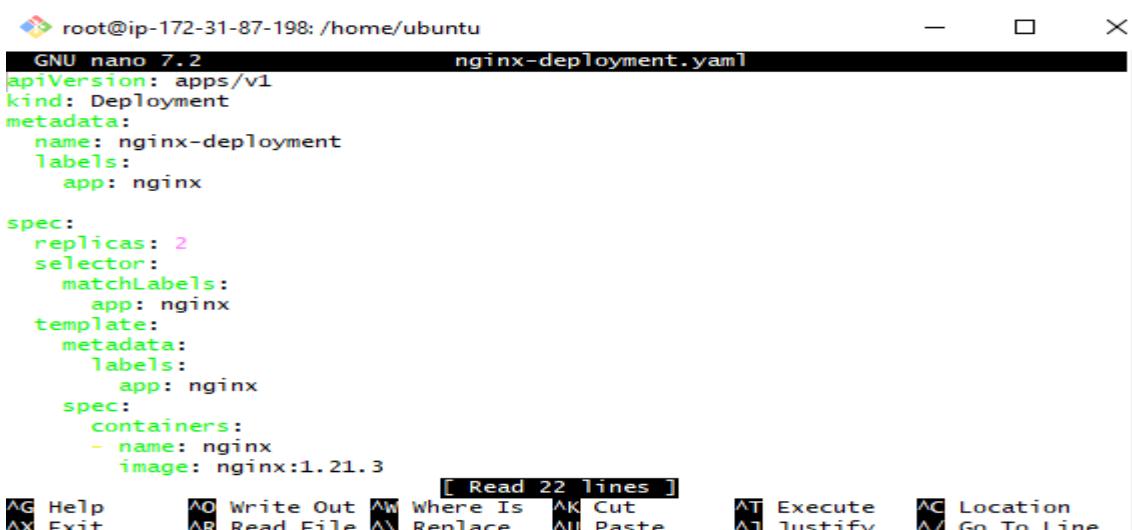
```
root@ip-172-31-87-198:/home/ubuntu# kubectl get nodes
NAME           STATUS  ROLES      AGE   VERSION
ip-172-31-80-64  Ready   <none>    18s   v1.29.0
ip-172-31-81-208 Ready   <none>    35s   v1.29.0
ip-172-31-87-198 Ready   control-plane  43m   v1.29.0
```

### Step 3: Create the Deployment YAML file

- a) Creating the YAML file: Use a text editor to create a file named nginx-deployment.yaml  
And nginx-service.yaml

```
root@ip-172-31-87-198:/home/ubuntu# nano nginx-deployment.yaml
root@ip-172-31-87-198:/home/ubuntu# nano nginx-service.yaml
root@ip-172-31-87-198:/home/ubuntu# kubectl apply of nginx-deployment.yaml
error: Unexpected args: [of nginx-deployment.yaml]
See 'kubectl apply -h' for help and examples
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-service.yaml
service/nginx-service created
```

- b) Adding the Deployment Configuration to nginx-deployment.yaml and nginx-service.yaml



```
root@ip-172-31-87-198:/home/ubuntu
GNU nano 7.2                               nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  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
[ Read 22 lines ]
```

AG Help AO Write Out AW Where Is AK Cut AT Execute AC Location  
AX Exit AR Read File AR Replace AU Paste AJ Justify AL Go To Line



```
root@ip-172-31-87-198:/home/ubuntu
GNU nano 7.2                               nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

## Step 4: Applying the YAML Files

a) Deploying the Application: Use kubectl to create the Deployment and Service from the YAML files.

```
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Verifying the Deployment and also describing the deployment:

Check the status of your Deployment, Pods and Services.

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2           5m57s
root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2           6m39s

root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2           28m
root@ip-172-31-87-198:/home/ubuntu# kubectl describe deployment
Name:            nginx-deployment
Namespace:       default
CreationTimestamp:  Wed, 18 Sep 2024 12:14:59 +0000
Labels:          app=nginx
Annotations:    deployment.kubernetes.io/revision: 1
Selector:        app=nginx
Replicas:       2 desired | 2 updated | 2 total | 2 available | 0 unavailable
Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx:1.21.3
      Port:       80/TCP
      Host Port:  0/TCP
      Environment: <none>
root@ip-172-31-87-198:/home/ubuntu# kubectl get service
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1   <none>        443/TCP   55m
nginx-service   LoadBalancer  10.109.148.186  <pending>   80:30162/TCP  31m
root@ip-172-31-87-198:/home/ubuntu#
```

## Step 6: Ensure Service is Running

6.1 Verify Service: Running the following commands to check the services running in our cluster:

Command: kubectl get service

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get service
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes    ClusterIP  10.96.0.1    <none>       443/TCP       101m
nginx-service LoadBalancer 10.106.17.37 <pending>   80:31687/TCP  105s
root@ip-172-31-87-198:/home/ubuntu# kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Croot@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-n52mq  1/1     Running   0          9m20s
nginx-deployment-6b4d6fdbf-w9qjv  1/1     Running   0          9m20s
root@ip-172-31-87-198:/home/ubuntu# kubectl logs nginx-deployment-6b4d6fdbf-n52mq
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/18 11:34:24 [notice] 1#1: using the "epoll" event method
2024/09/18 11:34:24 [notice] 1#1: nginx/1.21.3
2024/09/18 11:34:24 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2024/09/18 11:34:24 [notice] 1#1: OS: Linux 6.8.0-1012-aws
2024/09/18 11:34:24 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/18 11:34:24 [notice] 1#1: start worker processes
2024/09/18 11:34:24 [notice] 1#1: start worker process 26
2024/09/18 11:34:24 [notice] 1#1: start worker process 27

root@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-bqqg2  1/1     Running   0          35m
nginx-deployment-6b4d6fdbf-ptgmg  1/1     Running   0          35m
```

## Step 7: Forward the Service Port to Your Local Machine

kubectl port-forward allows you to forward a port from your local machine to a port on a service running in the Kubernetes cluster.

Command:

kubectl port-forward service/<service-name> <local-port>:<service-port>

```
root@ip-172-31-87-198:/home/ubuntu# kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

```

root@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-bqqg2   1/1     Running   0          35m
nginx-deployment-6b4d6fdbf-ptgmg  1/1     Running   0          35m
root@ip-172-31-87-198:/home/ubuntu# kubectl logs nginx-deployment-6b4d6fdbf-bqqg
2
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perfo
rm configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-defau
lt.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d
/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf
.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.s
h
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.s
h
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/18 12:34:16 [notice] 1#1: using the "epoll" event method
2024/09/18 12:34:16 [notice] 1#1: nginx/1.21.3
2024/09/18 12:34:16 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2024/09/18 12:34:16 [notice] 1#1: OS: Linux 6.8.0-1012-aws
2024/09/18 12:34:16 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/18 12:34:16 [notice] 1#1: start worker processes
2024/09/18 12:34:16 [notice] 1#1: start worker process 26
2024/09/18 12:34:16 [notice] 1#1: start worker process 27

```

## Step 8: Access the Application

- Open a web browser and navigate to `http://<Node-IP>:<Port>`. You should see the NGINX application running in the Kubernetes cluster.



## Experiment No 5 : Terraform

Aim : Installation and Configuration of Terraform in Windows

The screenshot shows the Terraform website at <https://developer.hashicorp.com/terraform/install>. The left sidebar shows options for different operating systems: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, and Solaris. The main content area is divided into two sections: 'Windows' and 'Linux'. The 'Windows' section shows binary download links for '386' (Version: 1.9.5) and 'AMD64' (Version: 1.9.5). The 'Linux' section shows package manager links for 'Ubuntu/Debian', 'CentOS/RHEL', 'Fedora', 'Amazon Linux', and 'Homebrew'. Below the Linux section is a terminal window showing the command to add the GPG key and update the apt repository.

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearm [REDACTED]o  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
sudo apt update && sudo apt install terraform
```

The screenshot shows the Windows File Explorer with the path 'This PC > Downloads'. The 'Downloads' folder contains four items: 'terraform\_1.9.5\_windows\_amd64' (a compressed zip file), 'AIDS\_Exp5 (1)' (a WPS PDF document), 'AIDS\_Exp5' (another WPS PDF document), and 'AIDS\_Exp5' (an OpenDocument document). The 'terraform\_1.9.5\_windows\_amd64' file is selected.

The screenshot shows the WinRAR file manager with the path 'This PC > Downloads > terraform\_1.9.5\_windows\_amd64'. The contents of the compressed folder are listed in a table:

Name	Type	Compressed size	Password ...	Size
LICENSE	Text Source File	2 KB	No	5 KB
terraform	Application	26,719 KB	No	88,962 KB

## Compressed (zipped) Folders

X



This application may depend on other compressed files in this folder.

For the application to run properly, it is recommended that you first extract all files.

**Extract all**

Run

Cancel

X

← Extract Compressed (Zipped) Folders

### Select a Destination and Extract Files

Files will be extracted to this folder:

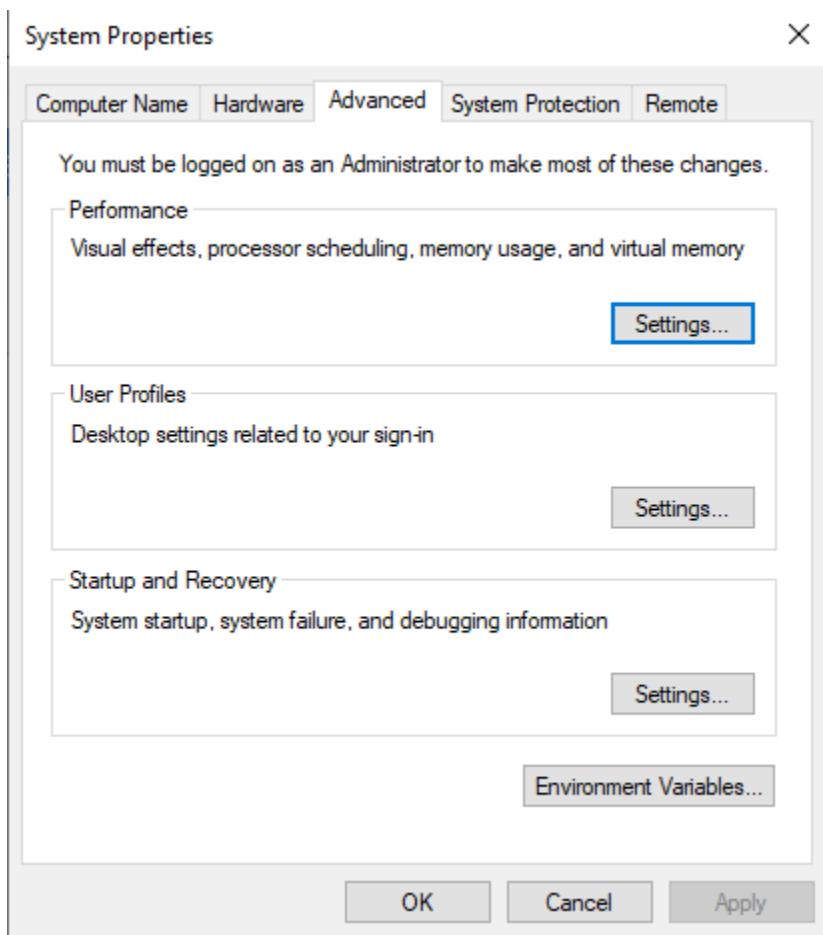
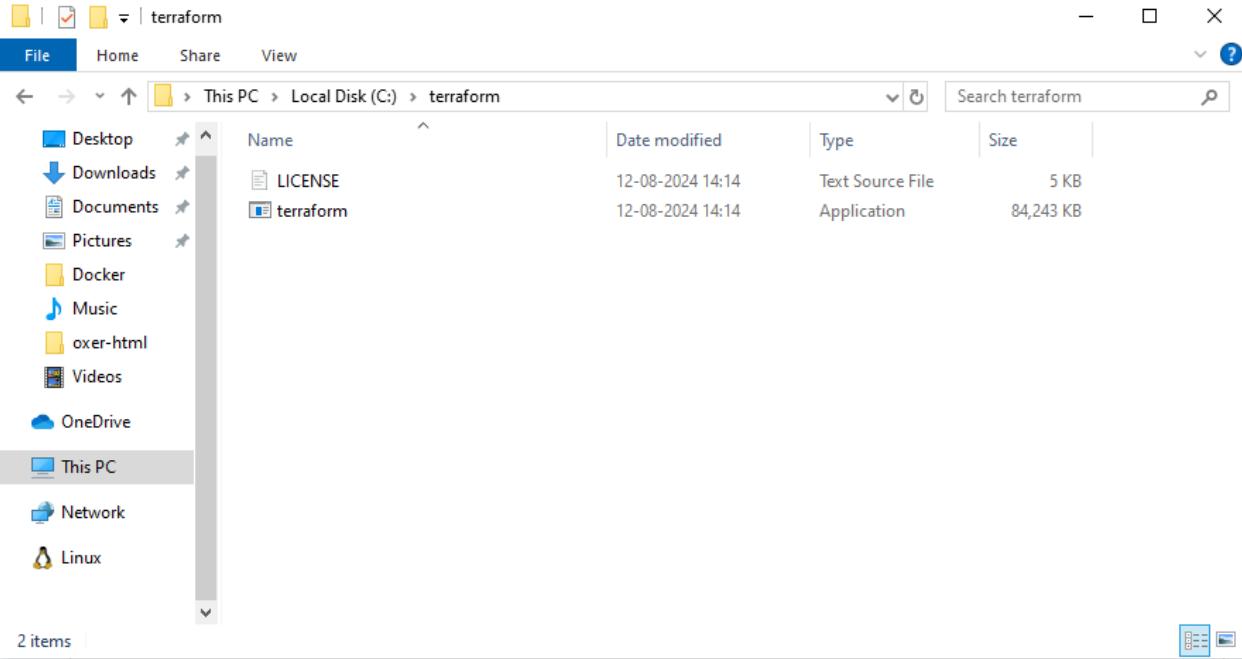
C:\Users\INFT\Downloads\terraform\_1.9.5\_windows\_amd64

**Browse...**

Show extracted files when complete

**Extract**

Cancel



Environment Variables

User variables for INFT

Variable	Value
OneDrive	C:\Users\INFT\OneDrive
Path	C:\Users\INFT\AppData\Local\Microsoft\WindowsApps;C:\Users\I...
TEMP	C:\Users\INFT\AppData\Local\Temp
TMP	C:\Users\INFT\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

New... Edit... Delete

OK Cancel

Edit User Variable

Variable name:	Path
Variable value:	C:\terraform

Browse Directory... Browse File... OK Cancel

Administrator: Windows PowerShell



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

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

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

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

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

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

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

PS C:\Windows\system32>
```

## EXPERIMENT NO. 6

### **Aim:**

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

### **Theory :**

**Terraform** is an open-source tool that enables developers and operations teams to define, provision, and manage cloud infrastructure through code. It uses a declarative language to specify the desired state of infrastructure, which can include servers, storage, networking components, and more. With Terraform, infrastructure changes can be automated, versioned, and tracked efficiently.

### **Building Infrastructure**

When you build infrastructure using Terraform, you define the desired state of your infrastructure in configuration files. For example, you may want to create an S3 bucket or deploy a Docker container on an EC2 instance. Terraform reads these configuration files and, using the specified cloud provider (such as AWS), it provisions the necessary resources to match the desired state.

- **S3 Buckets:** Terraform can create and manage S3 buckets, which are used to store and retrieve data objects in the cloud. You can define the properties of the bucket, such as its name, region, access permissions, and versioning.
- **Docker on AWS:** Terraform can deploy Docker containers on AWS infrastructure. This often involves setting up an EC2 instance and configuring it to run Docker containers, which encapsulate applications and their dependencies.

### **Changing Infrastructure**

As your needs evolve, you may need to modify the existing infrastructure. Terraform makes it easy to implement changes by updating the configuration files to reflect the new desired state. For instance, you might want to change the storage settings of an S3 bucket, add new security policies, or modify the Docker container's configuration.

Terraform's "plan" command helps you preview the changes that will be made to your infrastructure before applying them. This step ensures that you understand the impact of your changes and can avoid unintended consequences.

### **Destroying Infrastructure**

When certain resources are no longer needed, Terraform allows you to destroy them in a controlled manner. This might involve deleting an S3 bucket or terminating an EC2 instance running Docker containers. By running the "destroy" command, Terraform ensures that all associated resources are properly de-provisioned and removed.

Destroying infrastructure with Terraform is beneficial because it helps avoid unnecessary costs associated

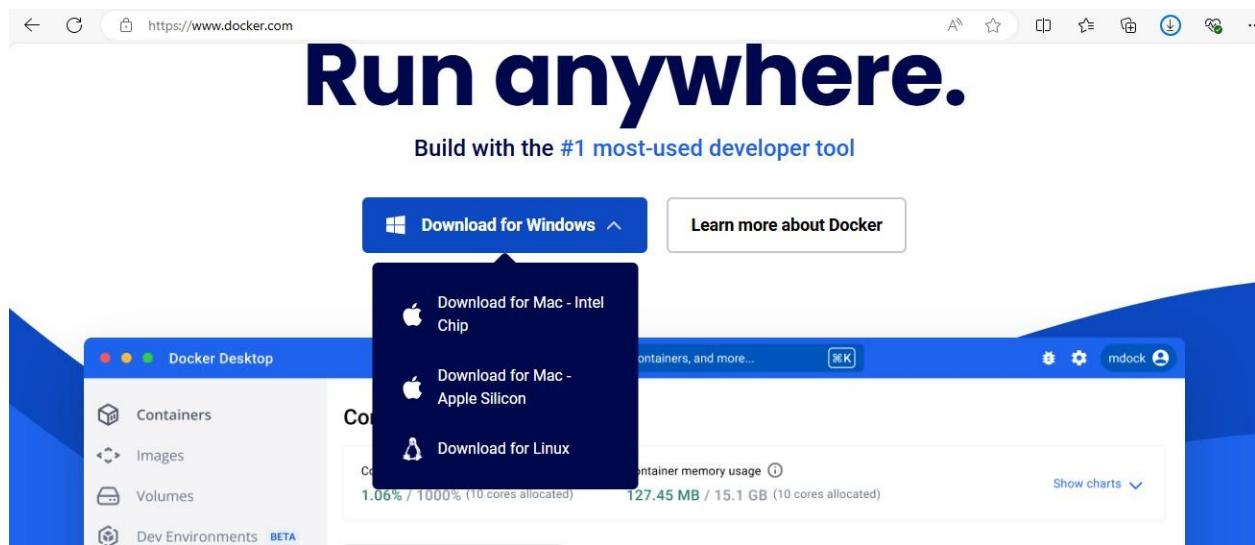
with unused resources and ensures that the environment remains clean and free of clutter.

## Benefits of Using Terraform for AWS Infrastructure

- Consistency:** Terraform ensures that infrastructure is consistent across environments by applying the same configuration files.
- Automation:** Manual processes are reduced, and infrastructure is provisioned, updated, and destroyed automatically based on code.
- Version Control:** Infrastructure configurations can be stored in version control systems (like Git), allowing teams to track changes, collaborate, and roll back if necessary.
- Scalability:** Terraform can manage complex infrastructures, scaling them up or down as needed, whether for small projects or large-scale applications.
- Modularity:** Terraform configurations can be broken down into reusable modules, making it easier to manage and scale infrastructure.

## Installing and Setting Up Docker with Terraform:

### Step 1: Download Docker



### Step 2: Run the Docker installer and complete the installation process.

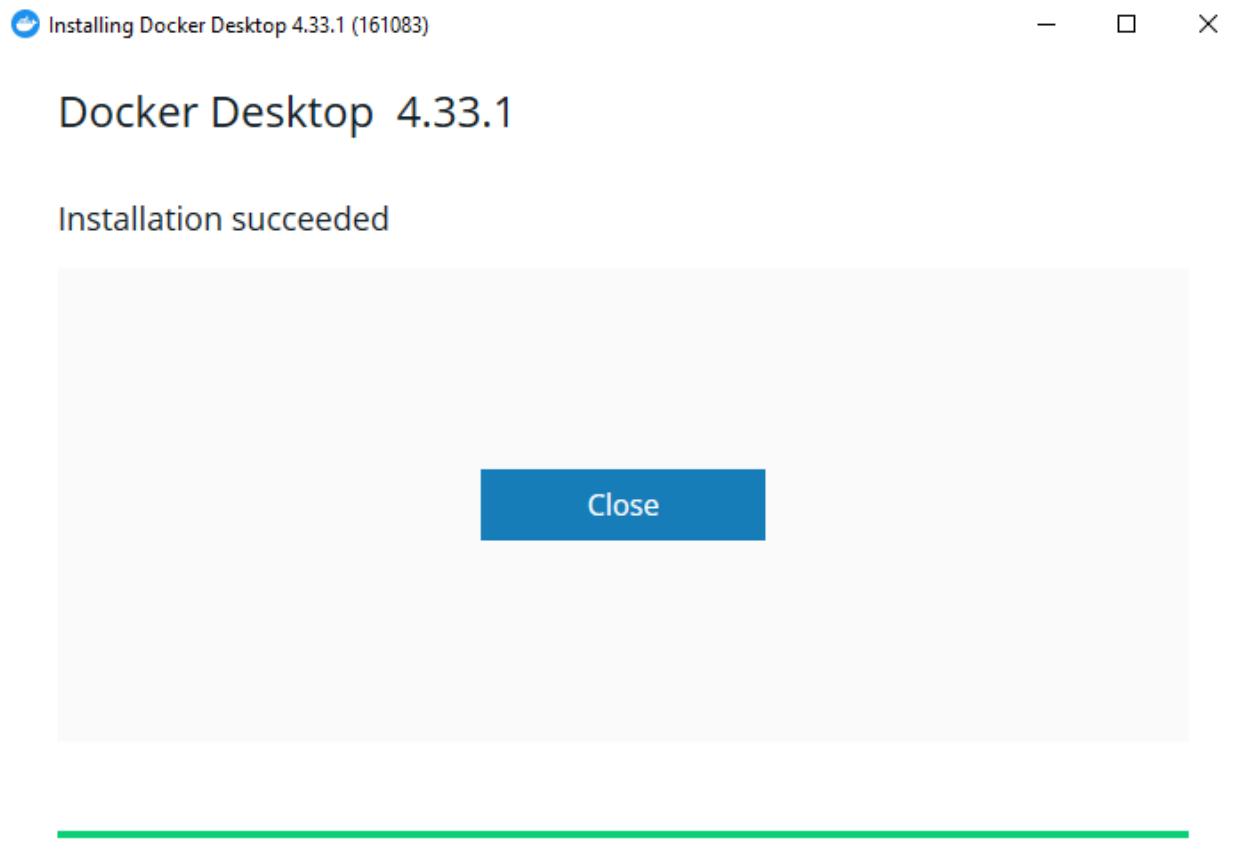
Name	Date modified	Type	Size
Docker Desktop Installer (1)	22-08-2024 14:43	Application	5,04,272 KB
terraform_1.9.5_windows_amd64	22-08-2024 14:12	Compressed (zipp...)	26,721 KB
AIDS_Exp5 (1)	22-08-2024 11:33	WPS PDF Docume...	305 KB
AIDS_Exp5	22-08-2024 11:30	WPS PDF Docume...	304 KB
AIDS_Exp5	22-08-2024 11:30	OpenDocument T...	339 KB
terraform_1.9.5_windows_amd64	22-08-2024 14:24	File folder	

Installing Docker Desktop 4.33.1 (161083)

## Docker Desktop 4.33.1

Unpacking files...

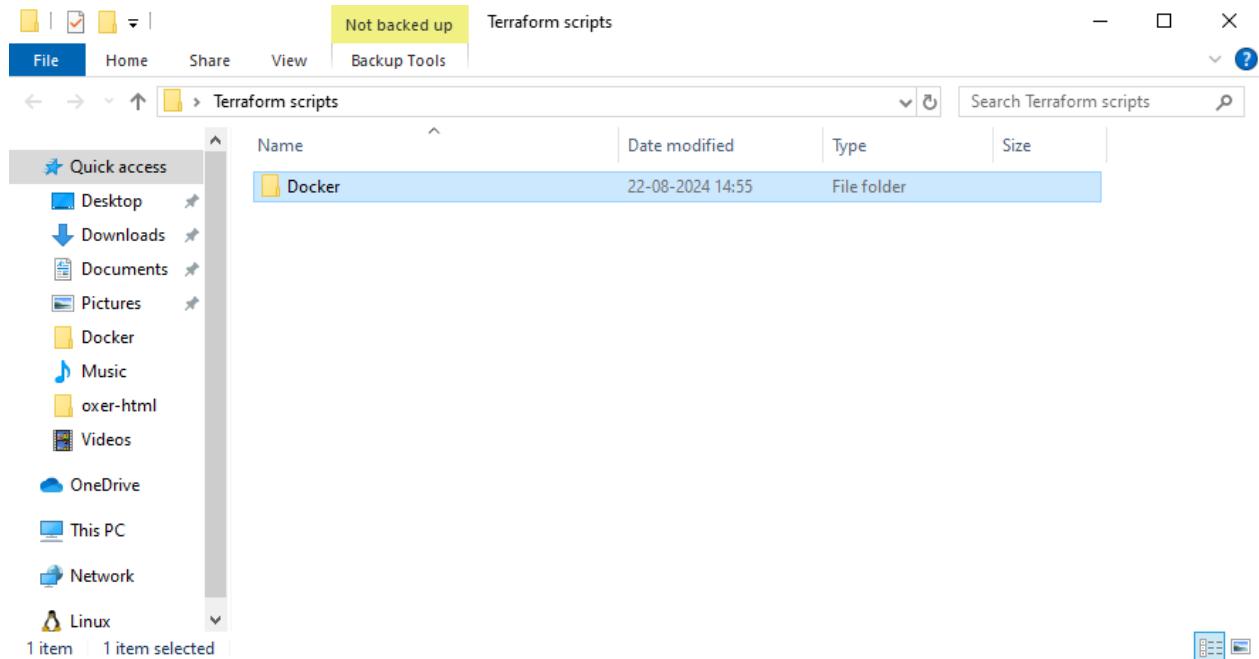
```
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/util.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs
```



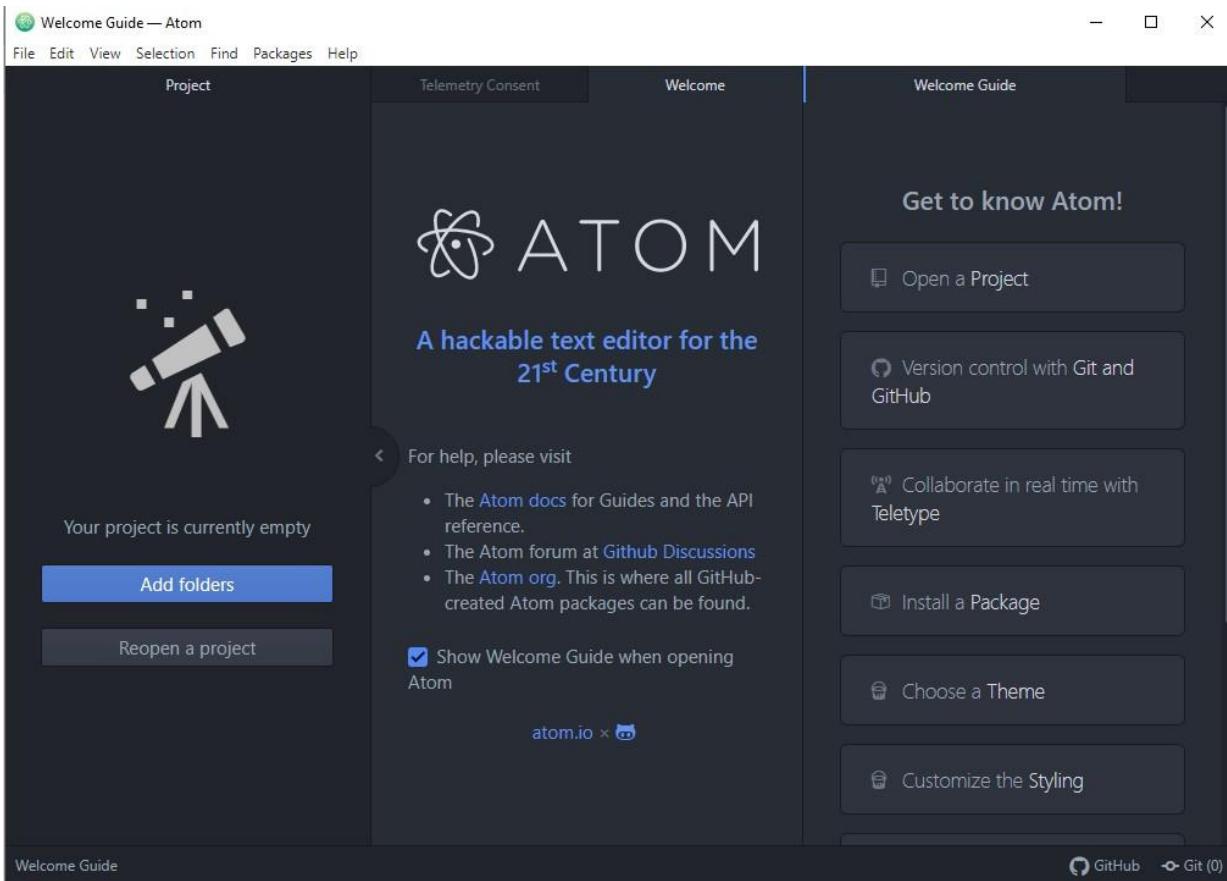
Step 3: Open Command Prompt as an administrator and enter `docker --version` to verify installation.

A screenshot of a Windows Command Prompt window titled "Command Prompt". The prompt shows the user's path as "C:\Users\INFT>". The user has run the command "docker --version", which returned the output "Docker version 27.1.1, build 6312585". Below this, the usage information for the docker command is displayed, followed by a list of common commands and management commands with their descriptions.

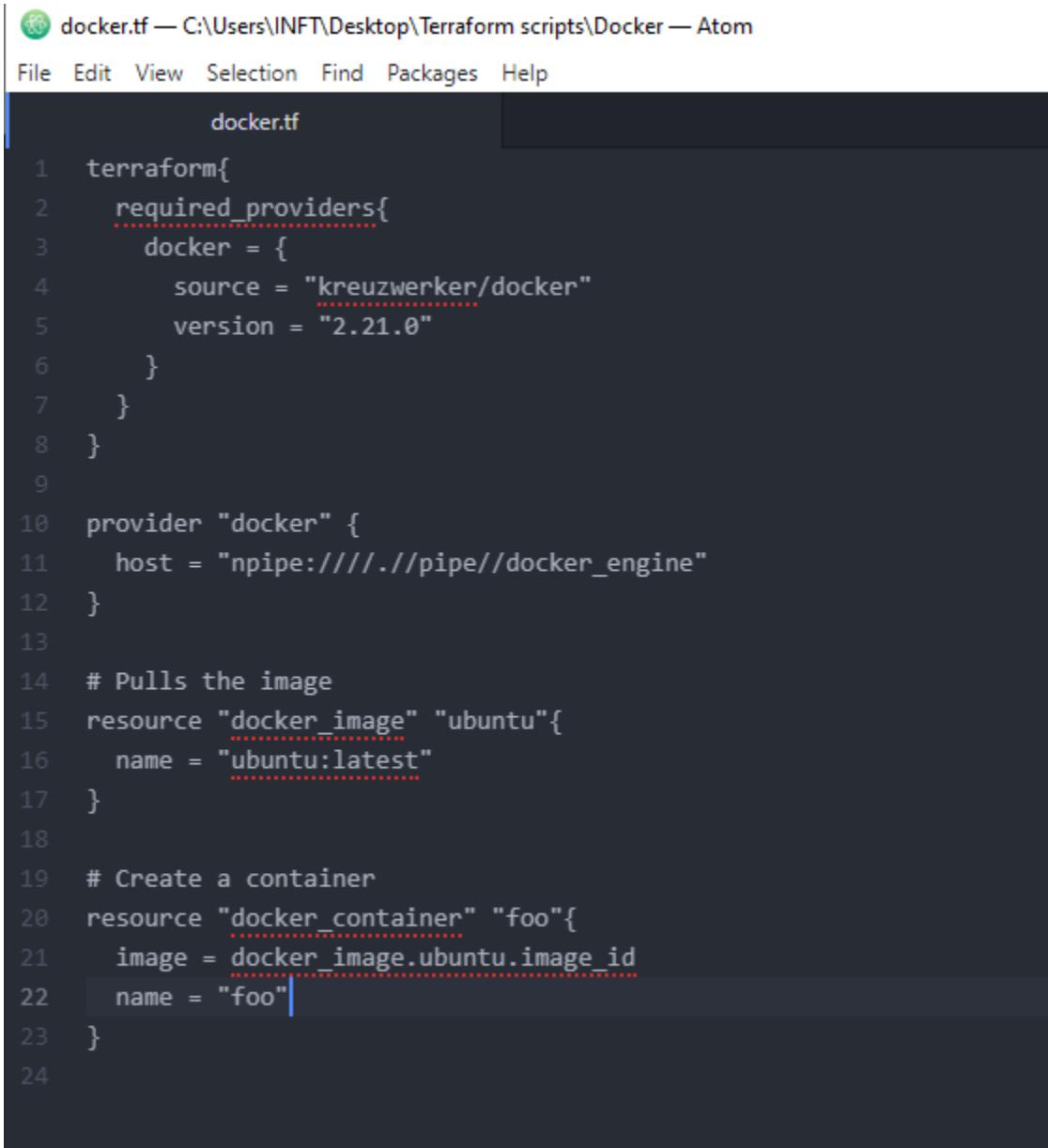
Step 4: Create a directory called Terraform\_Scripts and within it, a subdirectory named Docker



Step 5: Download and install the Atom Editor from [Atom's official site](#).



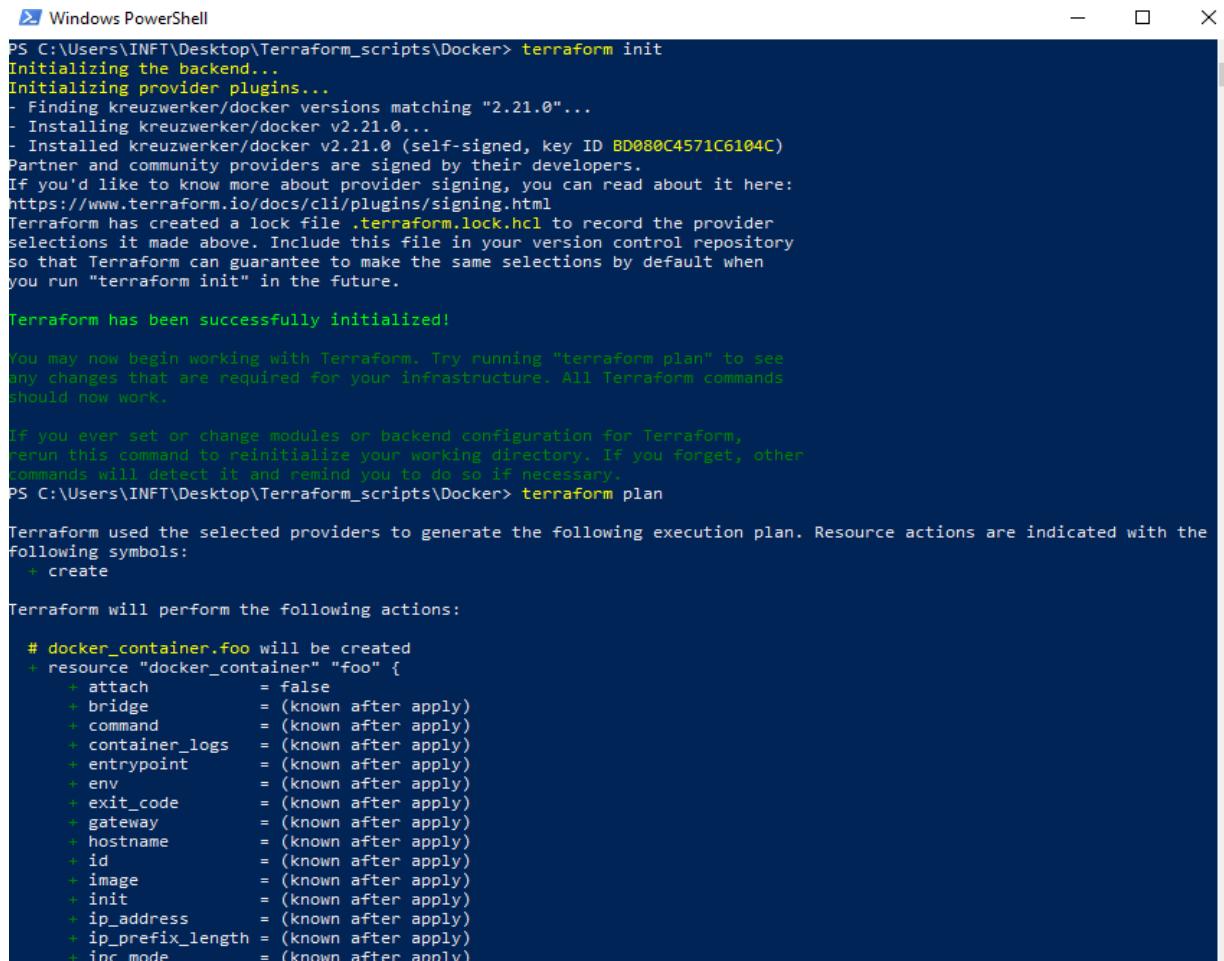
Step 6: Open Atom Editor, create a new document, and input or paste your script.



The screenshot shows the Atom code editor interface with a dark theme. The title bar reads "docker.tf — C:\Users\INFT\Desktop\Terraform scripts\Docker — Atom". The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The main editor area displays a Terraform configuration file:

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

Step 7: In Command Prompt, go to the Terraform\_Scripts directory and execute terraform init, terraform plan, terraform apply, terraform destroy, and docker images.



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the output of several Terraform commands. It starts with "terraform init", which initializes the backend, installs provider plugins, and creates a .terraform.lock.hcl lock file. It then shows "Terraform has been successfully initialized!". Next, "terraform plan" is run, which generates an execution plan. The output indicates that a "docker\_container.foo" resource will be created, specifically a "resource "docker\_container" "foo" {". The plan lists various attributes and their values, such as "attach = false", "bridge = (known after apply)", etc. The PowerShell window has a dark blue background and white text, with some command output in green.

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

Terraform has been successfully initialized!

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

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

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = (known after apply)
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
```

```
Windows PowerShell
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\INFT\Desktop\Terraform_scripts\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         = (known after apply)
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data   = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
    + security_opts  = (known after apply)
    + shm_size        = (known after apply)
    + start           = true
    + stdin_open      = false
    + stop_signal     = (known after apply)
    + stop_timeout    = (known after apply)
    + tty              = false

    + healthcheck (known after apply)

    + labels (known after apply)
}

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

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

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 11s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598/ubuntu:latest]
docker_container.foo: Creating...

Error: container exited immediately

with docker_container.foo,
on docker.tf line 20, in resource "docker_container" "foo":
20: resource "docker_container" "foo" {
```

```
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

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

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

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

Enter a value: yes

docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\INFT\Desktop\Terraform_scripts\Docker>
```

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

## EXPERIMENT NO:7

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

### **Theory:**

**Static Application Security Testing (SAST)** is a method of debugging by examining source code before a program is run. It involves analyzing the application's source code, bytecode, or binary code to identify vulnerabilities and security flaws. SAST tools scan code for common security vulnerabilities such as SQL injection, cross-site scripting (XSS), and buffer overflows, among others.

### **Problems SAST Solves:**

1. **Early Detection of Vulnerabilities:** SAST enables developers to find security flaws early in the development lifecycle, reducing the cost and effort required to fix them later.
2. **Compliance with Security Standards:** It helps organizations comply with various security regulations and standards, such as PCI DSS, OWASP Top Ten, and ISO 27001, by identifying security weaknesses that need to be addressed.
3. **Integration into CI/CD Pipelines:** SAST tools can be integrated into Continuous Integration/Continuous Deployment (CI/CD) pipelines, allowing for automated security checks during the development process.
4. **Comprehensive Coverage:** It scans all code paths and identifies vulnerabilities that may not be detected during dynamic testing (which tests the application while it runs).
5. **Reduction of Technical Debt:** By catching vulnerabilities early, SAST helps prevent the accumulation of technical debt related to security issues, making the codebase more maintainable.
6. **Improved Code Quality:** Besides security, SAST tools often identify coding best practices and help improve overall code quality.
7. **Enhanced Collaboration:** By providing clear reports and insights, SAST tools foster better communication between development and security teams.
8. **Risk Mitigation:** It helps organizations manage risks associated with software vulnerabilities, thereby protecting against data breaches and cyberattacks.

### **Prerequisites:**

- Jenkins installed
- Docker Installed (for SonarQube)

## Steps to integrate Jenkins with SonarQube:

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

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

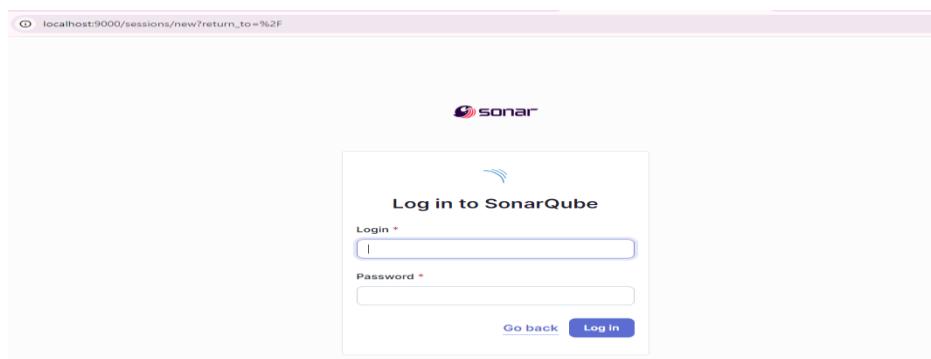
**Command:** docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Users\Windows> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
7df3e28058c6bfc74d745f9f18f0923c82c1fc4058967a5b33907e0010b01ee2
PS C:\Users\Windows>
```

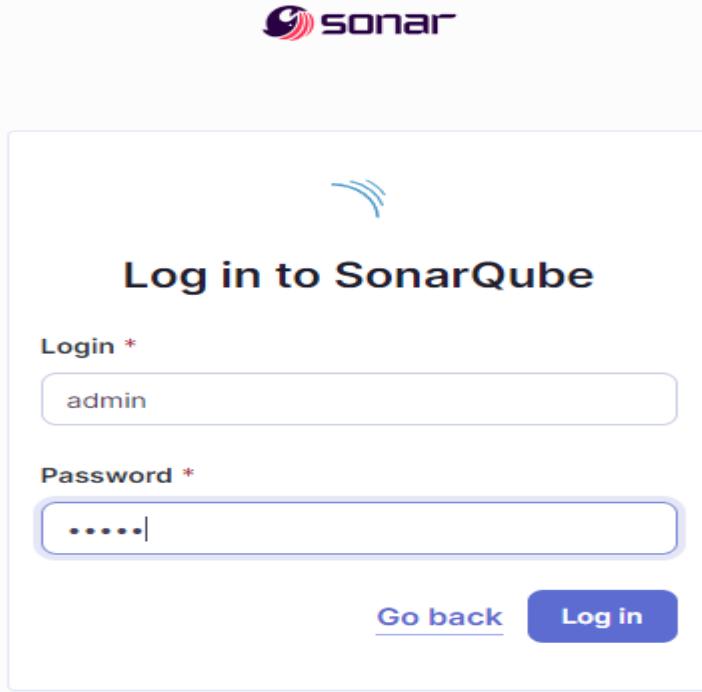
```
PS C:\Users\Windows> docker ps
>>
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7df3e28058c6 sonarqube:latest "/opt/sonarqube/dock..." 5 minutes ago Up 5 minutes 0.0.0.0:9000->9000/tcp sonarqube
PS C:\Users\Windows> ■
```

```
# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
sonar.jdbc.username=snehalsonar
sonar.jdbc.password=snehalsonar
```

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



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



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

The screenshot shows the "Create a local project" wizard on SonarQube. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, and Quality Gates. The current step is "1 of 2". The form has three fields: "Project display name \*" with the value "Snehal-sonarqube", "Project key \*" with the value "Snehal-sonarqube", and "Main branch name \*" with the value "main". Below the form is a note: "The name of your project's default branch [Learn More](#)". At the bottom are "Cancel" and "Next" buttons.

## Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

Previous version

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

Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

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

Recommended for projects following regular versions or releases.

## 6. Generate a token

### Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

#### 1 Provide a token

[Generate a project token](#)

[Use existing token](#)

Token name [?](#)

Expires in

Analyze "Snehal-sonarqube"

30 days

[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.

### Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password leak over the network.

#### Generate Tokens

Name

Type

Expires in

Enter Token Name

Select Token Type

30 days

Generate

New token "snehalsonar" has been created. Make sure you copy it now, you won't be able to see it again!

sqa\_d52d2f5b73ebb76572f9042ed2117d8980481682



7. Setup the project and come back to Jenkins Dashboard.

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

## Download progress

### Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner  Success

Loading plugin extensions  Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

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

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

Enter the Server Authentication token if needed.

### 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

sonarqube



Server URL

Default is http://localhost:9000

http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Dashboard > Manage Jenkins > System > 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: sonarqube

Server URL: http://localhost:9000

Server authentication token: Sonar\_token

**Save** **Apply**

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

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name: sonarqube\_scanner

**Required**

Install automatically ?

Install from Maven Central

Version: SonarQube Scanner 6.2.0.4584

Add Installer

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

New Item

Enter an item name: SonarQube

Select an item type

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

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

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

**OK**

## 11. Choose this GitHub repository in Source Code

Management.

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

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

### Source Code Management

None

Git [?](#)

#### Repositories [?](#)

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

! Please enter Git repository.

##### Credentials [?](#)

- none -

[+ Add ▾](#)

Advanced ▾

#### Branches to build [?](#)

##### Branch Specifier (blank for 'any') [?](#)

[Add Branch](#)

#### Repository browser [?](#)

#### Additional Behaviours

[Add ▾](#)

12. Under Build-> 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' page for a build step. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Configure' and contains sections for 'Path to project properties', 'Analysis properties', 'Additional arguments', and 'JVM Options'. In the 'Analysis properties' section, the following values are set:

```
sonar.projectKey = Snehal-sonarqube
sonar.login = squ_5d57f35ed4e7c0f733830e90700608647df8168d
sonar.sources = HelloWorldCore
sonar.host.url = http://localhost:9000/
```

At the bottom of the configuration page are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins 'Permissions' page. At the top, it displays the user 'Administrator admin'. Below the user information are checkboxes for 'Quality Gates', 'Quality Profiles' (which is checked), and 'Projects'. To the right of the checkboxes are links for 'Download', 'Copy', and 'View as plain text'.

Check the console output.

The screenshot shows the Jenkins 'Console Output' page. It displays the log output from the SonarQube build step. The output shows the process of cloning a GitHub repository and preparing it for analysis by SonarQube. Key lines include:

```
Started by user Snehal Patil
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.42.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
```

```
14:04:53.398 INFO Project root configuration file: NONE
14:04:53.455 INFO SonarScanner CLI 6.2.0.4584
14:04:53.459 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
14:04:53.471 INFO Windows 10 10.0 amd64
14:04:53.542 INFO User cache: C:\WINDOWS\system32\config\systemprofile\.sonar\cache
14:04:55.391 INFO JRE provisioning: os[windows], arch[amd64]
14:04:56.116 INFO Communicating with SonarQube Server 10.6.0.92116
14:04:57.650 INFO Starting SonarScanner Engine...
14:04:57.651 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
14:05:01.587 INFO Load global settings
14:05:01.991 INFO Load global settings (done) | time=402ms
14:05:02.005 INFO Server id: 147B411E-AZIo070pNro_dTnC3uoH
14:05:02.035 INFO Loading required plugins
14:05:02.036 INFO Load plugins index
14:05:02.166 INFO Load plugins index (done) | time=127ms
14:05:02.169 INFO Load/download plugins
14:05:07.332 INFO Load/download plugins (done) | time=5168ms
14:05:08.489 INFO Process project properties
14:05:08.518 INFO Process project properties (done) | time=29ms
14:05:08.560 INFO Project key: Snehal-sonarqube
14:05:08.562 INFO Base dir: C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
14:05:08.565 INFO Working dir: C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\scannerwork
14:05:08.601 INFO Load project settings for component key: 'Snehal-sonarqube'
14:05:08.694 INFO Load project settings for component key: 'Snehal-sonarqube' (done) | time=91ms
14:05:08.906 INFO Load quality profiles
14:05:09.907 INFO Load quality profiles (done) | time=1001ms
```

```
14:05:39.119 INFO Load analysis cache (404) | time=45ms
14:05:39.875 WARN The property 'sonar.login' is deprecated and will be removed in the future. Please use the 'sonar.token' property instead when passing a token.
14:05:39.954 INFO Preprocessing files...
14:05:43.668 INFO 2 languages detected in 23 preprocessed files
14:05:43.671 INFO 0 files ignored because of scm ignore settings
14:05:43.678 INFO Loading plugins for detected languages
14:05:43.679 INFO Load/download plugins
14:05:44.555 INFO Load/download plugins (done) | time=876ms
14:05:44.835 INFO Executing phase 2 project builders
14:05:44.838 INFO Executing phase 2 project builders (done) | time=3ms
14:05:44.870 INFO Load project repositories
14:05:44.941 INFO Load project repositories (done) | time=72ms
14:05:45.005 INFO Indexing files...
14:05:45.007 INFO Project configuration:
14:05:45.083 INFO 23 files indexed
14:05:45.087 INFO Quality profile for cs: Sonar way
14:05:45.088 INFO Quality profile for json: Sonar way
14:05:45.091 INFO ----- Run sensors on module Snehal-sonarqube
14:05:45.236 INFO Load metrics repository
14:05:45.321 INFO Load metrics repository (done) | time=84ms
14:05:47.290 INFO Sensor C# Project Type Information [csharp]
14:05:47.294 INFO Sensor C# Project Type Information [csharp] (done) | time=5ms
14:05:47.296 INFO Sensor C# Analysis Log [csharp]
14:05:47.343 INFO Sensor C# Analysis Log [csharp] (done) | time=48ms
14:05:47.344 INFO Sensor C# Properties [csharp]
```

```

14:05:51.146 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
14:05:51.147 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
14:05:51.147 INFO Sensor Zero Coverage Sensor
14:05:51.162 INFO Sensor Zero Coverage Sensor (done) | time=18ms
14:05:51.170 INFO SCM Publisher SCM provider for this project is: git
14:05:51.174 INFO SCM Publisher 2 source files to be analyzed
14:05:52.968 INFO SCM Publisher 2/2 source files have been analyzed (done) | time=1792ms
14:05:52.973 INFO CPD Executor Calculating CPD for 0 files
14:05:52.984 INFO CPD Executor CPD calculation finished (done) | time=0ms
14:05:52.998 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
14:05:53.651 INFO Analysis report generated in 374ms, dir size=199.1 kB
14:05:53.845 INFO Analysis report compressed in 123ms, zip size=20.5 kB
14:05:54.150 INFO Analysis report uploaded in 299ms
14:05:54.156 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=Snehal-sonarqube
14:05:54.160 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
14:05:54.162 INFO More about the report processing at http://localhost:9000/api/ce/task?id=1b0e8114-e5a0-4256-ba7f-e9eb666d5810
14:05:54.202 INFO Analysis total time: 46.510 s
14:05:54.211 INFO SonarScanner Engine completed successfully
14:05:54.357 INFO EXECUTION SUCCESS
14:05:54.360 INFO Total time: 1:00.973s
Finished: SUCCESS

```

## 14. Once the build is complete, check the project in SonarQube.

The screenshot shows the SonarQube 'main' project overview. The top navigation bar includes links for Overview, Issues, Security Hotspots, Measures, Code, Activity, Project Settings, and Project Information. The main content area displays the following information:

- Quality Gate:** Passed (green checkmark icon)
- New Code:** 0 Open issues (0 H, 0 M, 0 L)
- Overall Code:** 0 Open issues (0 H, 0 M, 0 L)
- Security:** 0 Open issues (0 H, 0 M, 0 L)
- Reliability:** 0 Open issues (0 H, 0 M, 0 L)
- Maintainability:** 0 Open issues (0 H, 0 M, 0 L)
- Activity:** Issues dropdown set to 'Issues'. A message states: 'There isn't enough data to generate an activity graph.'
- Summary:** September 25, 2024 at 2:05 PM (NOT PROVIDED), Quality Gate: Passed, First analysis: 0 Issues • 0.0% Coverage • 0.0% Duplications. A link 'See full history of analyses' is present.

## Advanced DevOps Expt No:08

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

### Theory:

**Static Application Security Testing (SAST)** is a methodology that analyzes source code for security vulnerabilities before the code is compiled, often referred to as white box testing.

### Problems SAST Solves:

- **Early Detection:** Identifies vulnerabilities in the early stages of the Software Development Life Cycle (SDLC), allowing developers to fix issues without breaking builds or passing vulnerabilities to production.
- **Real-Time Feedback:** Provides immediate insights while coding, which helps in addressing issues proactively.
- **Visual Guidance:** Offers graphical representations of vulnerabilities, indicating their locations and providing detailed guidance on remediation.

### Importance of SAST:

- **Efficiency:** Can analyze 100% of the codebase quickly, scanning millions of lines in minutes, unlike manual reviews that are time-consuming.
- **Scalability:** Addresses the challenge of limited security staff by automating vulnerability detection, identifying critical issues like SQL injection and cross-site scripting with high accuracy.

### What is SonarQube?

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

#### Benefits of SonarQube

- Sustainability - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- Increase productivity - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- Quality code - Code quality control is an inseparable part of the process of software development.
- Detect Errors - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.

## Integrating Jenkins with SonarQube:

### Prerequisites:

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

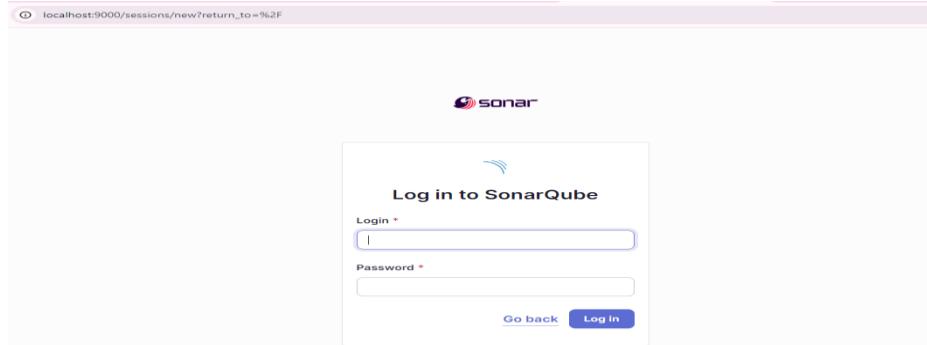
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform

SAST

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command –

```
PS C:\Users\Windows> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
7df3e28058c6bfc74d745f9f18f0923c82c1fc4058967a5b33907e0010b01ee2
PS C:\Users\Windows>
```

3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
4. Login to SonarQube using username admin and password admin.



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

1 of 2

## Create a local project

Project display name \*

 ✓

Project key \*

 ✓

Main branch name \*

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

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

5. Setup the project and come back to Jenkins Dashboard.
6. Create a New Item in Jenkins, choose Pipeline.

### New Item

Enter an item name

Select an item type



Freestyle project

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



Pipeline

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



Multi-configuration project

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

[OK](#)

7. Under Pipeline Script, enter the following -

Under Pipeline Script, enter the following -

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

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                 docker run --rm ^  
9                     -e SONAR_HOST_URL=http://172.20.64.1:9000 ^  
10                    -v ${WORKSPACE.replace('\\', '/')}:/.src ^  
11                     sonarsource/sonar-scanner-cli ^  
12                     -Dsonar.projectKey=sonarqube-test ^  
13                     -Dsonar.sources=. ^  
14                     -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^  
15                     -Dsonar.login=admin ^  
16                     -Dsonar.password=snehalsonar  
17             """  
18         }  
19     }
```

Use Groovy Sandbox ?

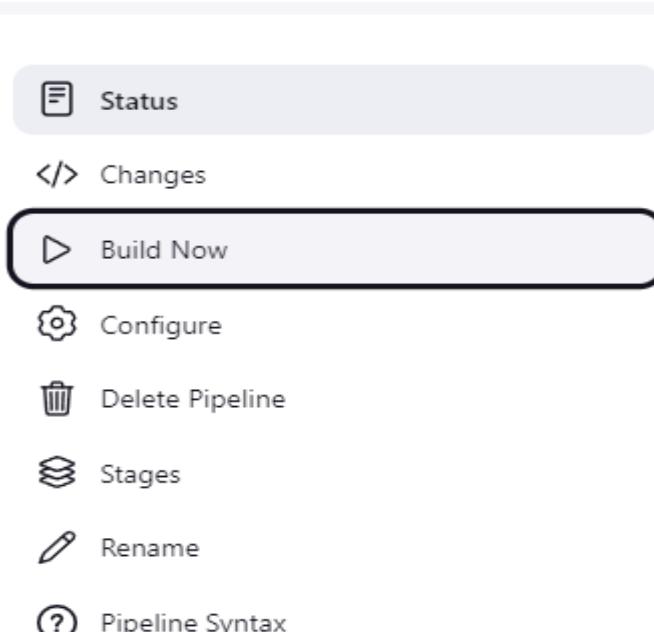
Pipeline Syntax

Save

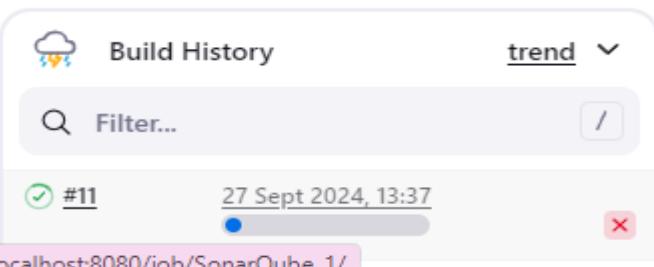
Apply

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

## 8.Run The Build.

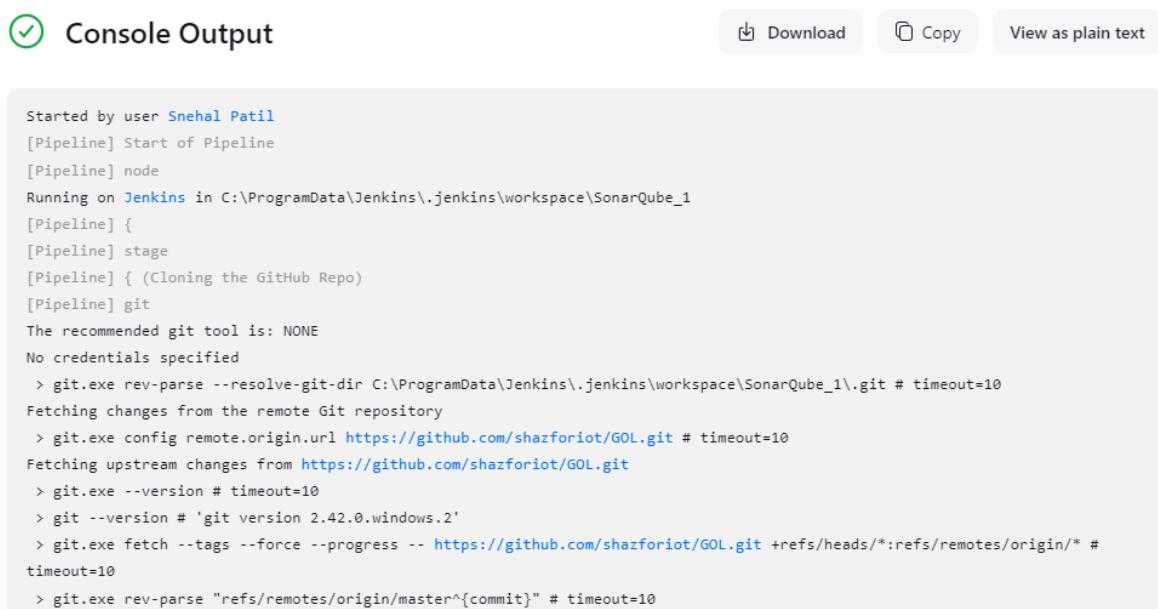


The screenshot shows the Jenkins Pipeline configuration page. At the top, there's a 'Status' button, followed by a 'Changes' section with a 'Build Now' button highlighted with a black border. Below these are options for 'Configure', 'Delete Pipeline', 'Stages', 'Rename', and 'Pipeline Syntax'.

The screenshot shows the Jenkins Build History page. It displays a list of builds, with build #11 selected. The status is shown as 'trend' with a green checkmark. The build was completed on '27 Sept 2024, 13:37'. A pink box highlights the URL 'localhost:8080/job/SonarQube\_1/'.

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



The screenshot shows the Jenkins Console Output page for job 'SonarQube\_1'. The output log is displayed, starting with the user who triggered the build and the pipeline stages. It includes git cloning, credential handling, and a detailed git fetch command.

```
Started by user Snehal Patil
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube_1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube_1\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git version 2.42.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
```

```

09:10:39.180 INFO Analysis report generated in 42838ms, dir size=127.2 MB
09:11:11.618 INFO Analysis report compressed in 32408ms, zip size=29.6 MB
09:11:36.343 INFO Analysis report uploaded in 24600ms
09:11:36.349 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://172.20.64.1:9000/dashboard?id=sonarqube-test
09:11:36.350 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
09:11:36.350 INFO More about the report processing at http://172.20.64.1:9000/api/ce/task?id=176cf1be-9b99-439a-a09b-9a9d6f49a11c
09:11:53.206 INFO Analysis total time: 31:25.145 s
09:11:53.223 INFO SonarScanner Engine completed successfully
09:11:54.386 INFO EXECUTION SUCCESS
09:11:56.253 INFO Total time: 32:07.446s
[Pipeline]
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

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

Snehal-sonarqube / main

**main**

Quality Gate **Passed** Version not provided Set as homepage Last analysis 1 day ago

New Code      Overall Code

Security	Reliability	Maintainability
0 Open issues	0 Open issues	0 Open issues
0 H 0 M 0 L	0 H 0 M 0 L	0 H 0 M 0 L

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

sonarqube-test / main

**main**

New Code      Overall Code

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

Accepted issues	Coverage	Duplications
0	On 0 lines to cover.	50.6%
Valid issues that were not fixed		On 759k lines.

Security Hotspots
3

## 11. Code Problems –

### Issues:

The screenshot shows the SonarQube 'Issues' tab for the 'sonarqube-test / main' project. The top navigation bar includes 'Overview', 'Issues' (selected), 'Security Hotspots', 'Measures', 'Code', and 'Activity'. Project settings and information are also present.

**Filters:** Includes 'My Issues' and 'All' buttons, and a 'Clear All Filters' button. A sidebar displays 'Issues in new code' and a 'Clean Code Attribute' section with metrics: Consistency (33k), Intentionality (14k), Adaptability (0), and Responsibility (0).

**Issues List:** Shows two items under 'gameoflife-core/build/reports/tests/all-tests.html':

- Add "lang" and/or "xml:lang" attributes to this "<html>" element. **Intentionality**: Reliability. Status: Open, Not assigned. L1 = 2min effort • 4 years ago • Bug • Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag. **Consistency**: Reliability. Status: Open, Not assigned. L1 = 5min effort • 4 years ago • Bug • Major.

### Security hotspots:

The screenshot shows the SonarQube 'Security Hotspots' tab for the 'sonarqube-test / main' project. The top navigation bar includes 'Overview', 'Issues' (selected), 'Security Hotspots' (dotted underline), 'Measures', 'Code', and 'Activity'. Project settings and information are also present.

**Review status:** 0.0% Security Hotspots Reviewed. A 'Review' button is highlighted.

**Review priority:** Medium (highlighted) and Low.

**Review details:** A Dockerfile snippet is shown with a warning: 'The tomcat image runs with root as the default user. Make sure it is safe here.'

### Codesmells:

The screenshot shows the SonarQube 'Codesmells' tab for the 'sonarqube-test / main' project. The top navigation bar includes 'Overview', 'Issues' (selected), 'Security Hotspots', 'Measures', 'Code' (highlighted), and 'Activity'. Project settings and information are also present.

**Metrics:** Security (0), Reliability (21k), Maintainability (164k).

**Severity:** Severity (0).

**Type:** Type (1). Sub-categories: Bug (47k), Vulnerability (0), and Code Smell (164k). An 'Add to selection' button is present.

**Issues List:** Shows two items under 'gameoflife-acceptance-tests/Dockerfile':

- Use a specific version tag for the image. **Intentionality**: Maintainability. Status: Open, Not assigned. L1 = 5min effort • 4 years ago • Code Smell • Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Intentionality**: Maintainability. Status: Open, Not assigned. L12 = 5min effort • 4 years ago • Code Smell • Major.

## Complexity:

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Measures' tab is selected. On the left, a sidebar lists various metrics: Maintainability, Security Review, Duplications, Size, Complexity (selected), Cyclomatic Complexity (1,112), and Issues. The main panel displays 'Cyclomatic Complexity' with a value of 1,112 and a link to 'See history'. Below this, a tree view shows complexity distribution across four modules: gameoflife-acceptance-tests, gameoflife-build, gameoflife-core (with a value of 18), and gameoflife-deploy.

## Duplications:

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Measures' tab is selected. On the left, a sidebar lists: Duplications (selected), Overview, Overall Code, Density (0.0%), Duplicated Lines (0), Duplicated Blocks (0), Duplicated Files (0), and Size. The main panel displays 'Duplicated Lines (%)' at 0.0%. To the right, a detailed view of a Dockerfile shows code duplication. The Dockerfile content is as follows:

```
1 shazfo... FROM selenium/standalone-firefox:latest
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

The Dockerfile uses environment variables \$MAVEN\_VERSION and \$MAVEN\_HOME, which are highlighted in red, indicating they are duplicated across multiple lines.

## **Advanced DevOps Lab**

### **Experiment No: 9**

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

#### **Theory:**

##### **What is Nagios?**

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

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

##### **Why We Need Nagios tool?**

Here are the important reasons to use Nagios monitoring tool:

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

##### **Features of Nagios:**

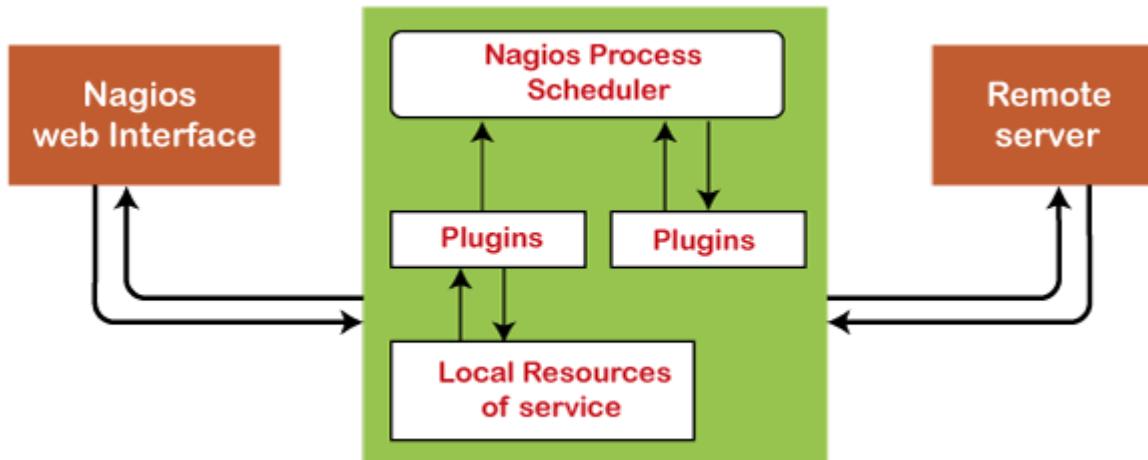
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.

- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

### Nagios Architecture

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



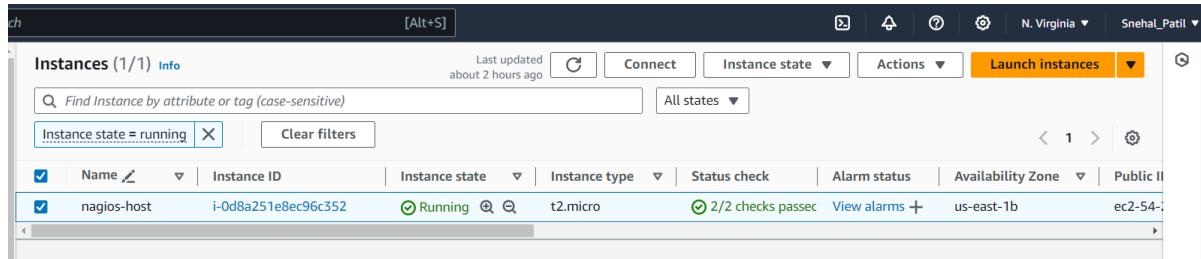
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates

# Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host



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

A screenshot of the AWS Security Groups inbound rules configuration page. It shows three rules: one for SSH (TCP port 22), one for HTTP (TCP port 80), and one for a custom TCP port (5666). All three rules have "Custom" as the source and "0.0.0.0/0" as the destination. There is a "Delete" button next to each rule. A "Add rule" button is at the bottom left.

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.

A screenshot of an EC2 Instance Connect session. The terminal shows the user updating the system and installing httpd and php packages. The output is as follows:

```
'`#`          Amazon Linux 2023
~~\`#####
~~ \#####
~~ \|#
~~ \|/
~~ \|/` https://aws.amazon.com/linux/amazon-linux-2023
~~ \|/` 
~~ \|/` 
[ec2-user@ip-172-31-83-176 ~]$ sudo yum update
Last metadata expiration check: 0:16:11 ago on Thu Sep 26 09:05:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:17:18 ago on Thu Sep 26 09:05:15 2024.
Dependencies resolved.

Package           Architecture      Version       Repository      Size
Installing:
httpd             x86_64          2.4.62-1.amzn2023
php8.3            x86_64          8.3.10-1.amzn2023.0.1      amazonlinux      48 k
amazonlinux      10 k

i-0561bc807cf78beba (nagios-host-snehal)
PublicIPs: 3.82.154.220 PrivateIPs: 172.31.83.176
```

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

sudo yum update

sudo yum install httpd php

sudo yum install gcc glibc glibc-common

sudo yum install gd gd-devel

```
Verifying : php8.3-8.3.10-1.amzn2023.0.1.x86_64          16/25
Verifying : php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64          17/25
Verifying : php8.3-common-8.3.10-1.amzn2023.0.1.x86_64          18/25
Verifying : php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64          19/25
Verifying : php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64          20/25
Verifying : php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64          21/25
Verifying : php8.3-process-8.3.10-1.amzn2023.0.1.x86_64          22/25
Verifying : php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64          23/25
Verifying : php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64          24/25
Verifying : php8.3-xsl-8.3.10-1.amzn2023.0.1.x86_64          25/25

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch    httpd-tools-2.4.62-1.amzn2023.x86_64
libsodium-1.0.19-4.amzn2023.x86_64        libxslt-1.1.34-5.amzn2023.0.2.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64   mod_lua-2.4.62-1.amzn2023.x86_64
php8.3-8.3.10-1.amzn2023.0.1.x86_64     php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64  php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64  php8.3-process-8.3.10-1.amzn2023.0.1.x86_64
php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64  php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-83-176 ~]$
```

i-0561bc807cf78beba (nagios-host-snehal)

Public IPs: 3.82.154.220 Private IPs: 172.31.83.176

```
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:19:41 ago on Thu Sep 26 09:05:15 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
Installing dependencies:
 annobin-docs      noarch        10.93-1.amzn2023.0.1      amazonlinux    92 k
 annobin-plugin-gcc x86_64         10.93-1.amzn2023.0.1      amazonlinux    887 k
 cpp              x86_64         11.4.1-2.amzn2023.0.2      amazonlinux    10 M
 gc               x86_64         8.0.4-5.amzn2023.0.2      amazonlinux    105 k
 glibc-devel       x86_64         2.34-52.amzn2023.0.11     amazonlinux    27 k
 glibc-headers-x86 noarch        2.34-52.amzn2023.0.11     amazonlinux    427 k
 guile22          x86_64         2.2.7-2.amzn2023.0.3      amazonlinux    6.4 M
 kernel-headers    x86_64         6.1.109-118.189.amzn2023  amazonlinux    1.4 M
 libmpc            x86_64         1.2.1-2.amzn2023.0.2      amazonlinux    62 k
 libtool-ltdl      x86_64         2.4.7-1.amzn2023.0.3      amazonlinux    38 k
 libcrypt-devel    x86_64         4.4.33-7.amzn2023        amazonlinux    32 k

Complete!
[ec2-user@ip-172-31-83-176 ~]$ i-0561bc807cf78beba (nagios-host-snehal)
Public IPs: 3.82.154.220 Private IPs: 172.31.83.176
```

```
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:20:53 ago on Thu Sep 26 09:05:15 2024.
Dependencies resolved.
=====
 Package           Architecture   Version            Repository      Size
Installing:
 gd               x86_64         2.3.3-5.amzn2023.0.3      amazonlinux    139 k
 gd-devel          x86_64         2.3.3-5.amzn2023.0.3      amazonlinux    38 k
Installing dependencies:
 brotli           x86_64         1.0.9-4.amzn2023.0.2      amazonlinux    314 k
 brotli-devel     x86_64         1.0.9-4.amzn2023.0.2      amazonlinux    31 k
 bzip2-devel      x86_64         1.0.8-6.amzn2023.0.2      amazonlinux    214 k
 cairo             x86_64         1.17.6-2.amzn2023.0.1      amazonlinux    684 k
 cmake-filesystem x86_64         3.22.2-1.amzn2023.0.4      amazonlinux    16 k
 fontconfig        x86_64         2.13.94-2.amzn2023.0.2    amazonlinux    273 k
 fontconfig-devel  x86_64         2.13.94-2.amzn2023.0.2    amazonlinux    128 k
 fonts-filesystem  noarch        1.12.0.5-12.amzn2023.0.2   amazonlinux    9.5 k
 freetype          x86_64         2.13.2-5.amzn2023.0.1      amazonlinux    423 k
 freetype-devel    x86_64         2.13.2-5.amzn2023.0.1      amazonlinux    912 k
 glib2-devel       noarch        2.74.7-689.amzn2023.0.2    amazonlinux    486 k
 google-noto-fonts-common x86_64         20201206-2.amzn2023.0.2  amazonlinux    15 k
 google-noto-sans-vf-fonts   noarch        20201206-2.amzn2023.0.2  amazonlinux    492 k

Complete!
[ec2-user@ip-172-31-83-176 ~]$ i-0561bc807cf78beba (nagios-host-snehal)
Public IPs: 3.82.154.220 Private IPs: 172.31.83.176
```

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

```
[ec2-user@ip-172-31-83-176 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-83-176 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
[ec2-user@ip-172-31-83-176 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-83-176 ~]$
```

6. Create a new user group

```
sudo groupadd nagcmd
```

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

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

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

```
[ec2-user@ip-172-31-83-176 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-83-176 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-83-176 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-83-176 ~]$
```

8. Create a new directory for Nagios downloads

```
mkdir ~/downloads
```

```
cd ~/downloads
```

```
[ec2-user@ip-172-31-83-176 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-83-176 ~]$ cd ~/downloads
```

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

wget

http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.

gz

wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-83-176 downloads]$ wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.  
gz  
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz  
wget: missing URL  
Usage: wget [OPTION]... [URL]...  
Try `wget --help' for more options.  
-bash: http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.: No such file or directory  
-bash: gz: command not found  
- 2024-09-26 09:33:05 -- http://nagios-plugins.org/download/nagios-plugins-2.0.3.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|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2659772 (2.5M) [application/x-gzip]  
Saving to: 'nagios-plugins-2.0.3.tar.gz'  
  
nagios-plugins-2.0.3.tar.gz      100%[=====]  2.54M  7.18MB/s    in 0.4s  
2024-09-26 09:33:06 (7.18 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]
```

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

tar zxvf nagios-4.0.8.tar.gz

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

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

**12.** Compile the source code.

make all

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

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-91-100 nagioscore-nagios-4.4.6]$ cd /tmp/nagioscore-nagios-4.4.6  
make all  
sudo make install  
sudo make install-init  
sudo make install-config  
sudo make install-commandmode  
cd ./base && make  
make[1]: Entering directory '/tmp/nagioscore-nagios-4.4.6/base'  
make -C ../lib  
make[2]: Entering directory '/tmp/nagioscore-nagios-4.4.6/lib'  
make[2]: Nothing to be done for 'all'.
```

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

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

```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ cat /etc/nagios/nagios.cfg
#####
# nagios.cfg                                     /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                snehalpatil302004@gmail.com   ; <>***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
#
```

15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
*** Nagios/Apache conf file installed ***
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$
```

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
```

17. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$
```

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

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.0.3.tar.gz
```

```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
nagios-plugins-2.0.3/
nagios-plugins-2.0.3/perlmods/
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.in
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.am
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz
```

## 19. Compile and install plugins

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

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

```
make
```

```
sudo make install
```

## 20. Start Nagios

Add Nagios to the list of system services

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

```
sudo chkconfig nagios on
```

Verify the sample configuration files

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

If there are no errors, you can go ahead and start Nagios.

```
sudo service nagios start
```

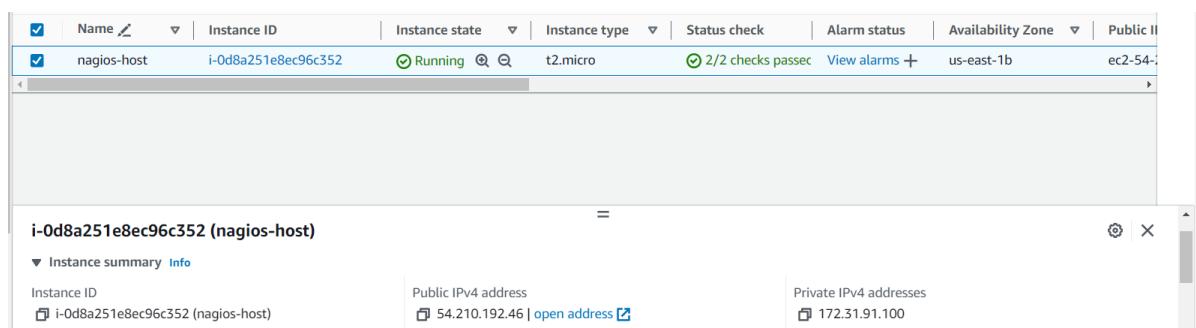
```
[ec2-user@ip-172-31-83-176 ~]$ sudo service nagios start
starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-83-176 ~]$
[ec2-user@ip-172-31-83-176 ~]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/init.d/nagios)
  Active: active (running) since Sat 2024-09-28 17:23:08 UTC; 5min ago
    Docs: https://www.nagios.org/documentation
 Main PID: 34305 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
     CPU: 68ms
    CGroup: /system.slice/nagios.service
            └─34305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

## 21. Check the status of Nagios

```
sudo systemctl status nagios
```

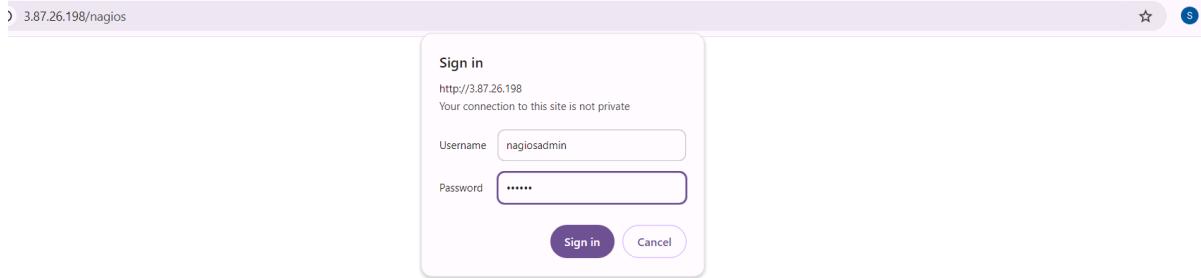
```
[ec2-user@ip-172-31-91-100 nagioscore-nagios-4.4.6]$ sudo systemctl status nagios
sudo systemctl status httpd
● nagios.service - Nagios Core 4.4.6
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
  Active: active (running) since Sat 2024-09-28 17:23:08 UTC; 5min ago
    Docs: https://www.nagios.org/documentation
 Main PID: 34305 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
     CPU: 68ms
    CGroup: /system.slice/nagios.service
            └─34305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

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



23. Open up your browser and look for [http://<your\\_public\\_ip\\_address>/nagios](http://<your_public_ip_address>/nagios)

Enter username as nagiosadmin and password which you set in Step 16.



## Extra:

Faced Problem

Permission Forbidden Error

**Problem:** While accessing the Nagios web interface, I encountered "403 Forbidden" errors.

**Solution:** This issue typically arises due to improper file permissions or SELinux settings. To resolve it:

**Adjust File Permissions:** Ensure that the Nagios directory and its files have the correct permissions. Execute the following commands:

```
sudo chown -R nagios:nagios /usr/local/nagios
```

```
sudo chmod -R 755 /usr/local/nagios
```

```
_m/'  
[ec2-user@ip-172-31-35-105 ~]$ sudo chown -R nagios:nagios /usr/local/nagios  
sudo chmod -R 755 /usr/local/nagios
```

Configure SELinux (if applicable): If SELinux is enabled on your instance, it might block access. To check the status and modify it:

```
sestatus # Check SELinux status
```

```
sudo setenforce 0 # Temporarily disable SELinux
```

```
[ec2-user@ip-172-31-35-105 ~]$ sestatus # Check SELinux status  
sudo setenforce 0 # Temporarily disable SELinux  
SELinux status:                 enabled  
SELinuxfs mount:               /sys/fs/selinux  
SELinux root directory:        /etc/selinux  
Loaded policy name:            targeted  
Current mode:                  permissive  
Mode from config file:         permissive  
Policy MLS status:             enabled  
Policy deny_unknown status:    allowed  
Memory protection checking:   actual (secure)  
Max kernel policy version:    33
```

Apache Configuration: Ensure that the Apache configuration allows access to the Nagios directory. You may need to add the following lines to your /etc/httpd/conf.d/nagios.conf:

```
<Directory "/usr/local/nagios/share">
```

Options None

AllowOverride None

Require all granted

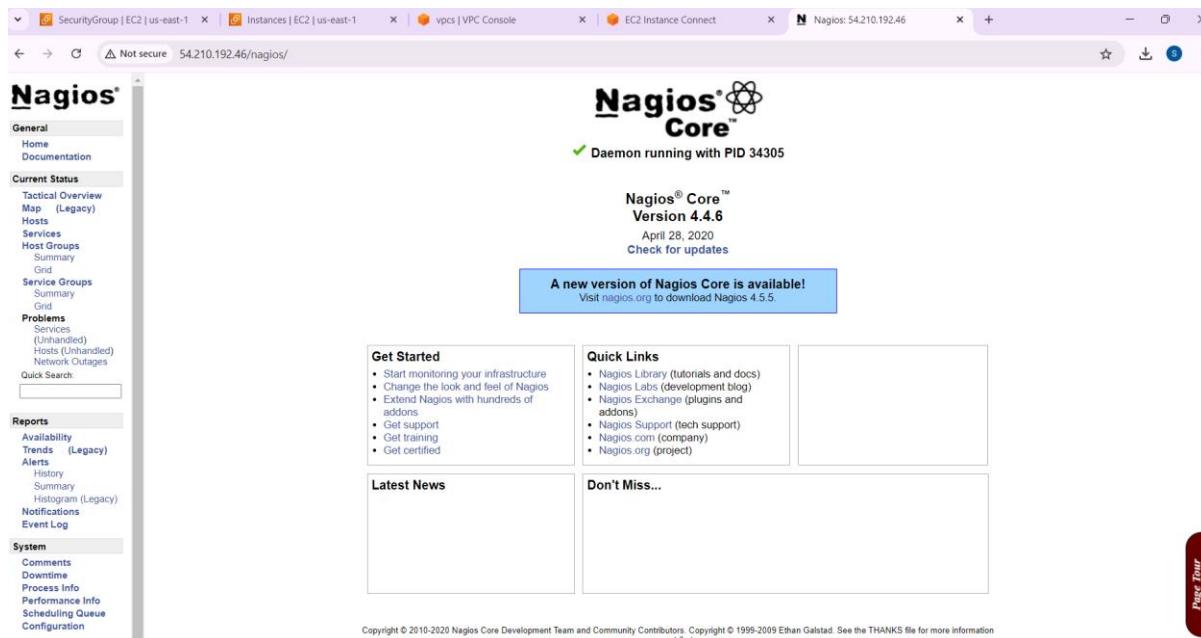
```
</Directory>
```

```
<Directory "/usr/local/nagios/share">
    Options None
    AllowOverride None
    Require all granted
</Directory>
```

After making changes, restart Apache:

```
sudo service httpd restart
```

24. After entering the correct credentials, I am able to see this page.



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

## Conclusion:

We successfully installed Nagios on an Amazon Linux EC2 instance in the AWS Free Tier. By configuring the security group, setting up users and groups, and installing the necessary packages, we prepared the environment for Nagios. After downloading and compiling the software, we configured the web interface and authenticated the admin account. Accessing the Nagios dashboard confirmed a successful setup, enabling effective monitoring of systems and services. This installation lays the groundwork for a robust monitoring solution tailored to your needs.

## Advanced DevOps Lab

### Experiment No: 10

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

Steps:

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running on the server side, run this sudo systemctl status nagios on the “NAGIOS HOST”.

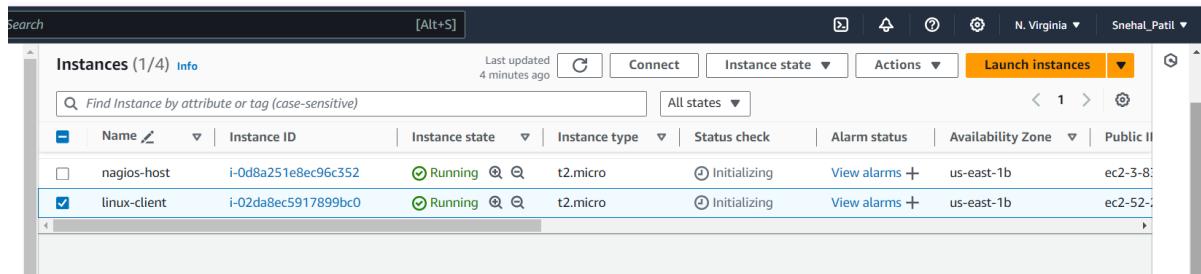
```
[ec2-user@ip-172-31-91-100 ~]$ sudo systemctl status
nagios
● ip-172-31-91-100.ec2.internal
  State: running
  Units: 298 loaded (incl. loaded aliases)
    Jobs: 0 queued
  Failed: 0 units
   Since: Sun 2024-09-29 05:39:54 UTC; 25s ago
  systemd: 252.23-2.amzn2023
 CGroup: /
          ├─init.scope
          │ └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
          ├─system.slice
          │ ├─acpid.service
          │   ├─1957 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate
          │   └─1995 /usr/sbin/acpid -f
```

You can proceed if you get this message.

Before we begin,

2. To monitor a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

Provide it with the same security group as the Nagios Host and name it ‘linux-client’ alongside the host.



For now, leave this machine as is, and go back to your nagios HOST machine.

Extra:

## Apache Server was Not Running

**Problem:** Unable to access the Nagios web interface due to the Apache HTTP server not running.

```
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl status httpd # For CentOS/RHEL
# or
sudo systemctl status apache2 # For Ubuntu/Debian
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)

Unit apache2.service could not be found.
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl status httpd # For CentOS/RHEL
# or
sudo systemctl status apache2 # For Ubuntu/Debian
```

```
Unit apache2.service could not be found.
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Wed 2024-10-02 06:48:57 UTC; 8s ago
       Docs: man:httpd.service(8)
 Main PID: 3401 (httpd)
   Status: "Started, listening on: port 80"
     Tasks: 177 (limit: 1112)
    Memory: 17.8M
      CPU: 58ms
     CGroup: /system.slice/httpd.service
             ├─3401 /usr/sbin/httpd -DFOREGROUND
             ├─3408 /usr/sbin/httpd -DFOREGROUND
             ├─3409 /usr/sbin/httpd -DFOREGROUND
             ├─3410 /usr/sbin/httpd -DFOREGROUND
             └─3411 /usr/sbin/httpd -DFOREGROUND
```

```
Oct 02 06:48:57 ip-172-31-84-219.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 02 06:48:57 ip-172-31-84-219.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 02 06:48:57 ip-172-31-84-219.ec2.internal httpd[3401]: Server configured, listening on: port 80
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-84-219 ~]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-84-219 ~]$
```

3. On the server, run this command

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

```
[ec2-user@ip-172-31-91-100 ~]$ ps -ef | grep nagios
nagios 2007 1 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 2008 2007 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 2009 2007 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 2010 2007 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 2011 2007 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 2012 2007 0 05:39 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 2917 2760 0 05:42 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-100 ~]$
```

#### 4. Become a root user and create 2 folders

```
sudo su
```

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

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

#### 5.

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

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

#### 6. Open linuxserver.cfg using nano and make the following changes

```
nano
```

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

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)

Change address to the public IP address of your LINUX CLIENT.

```
GNU nano 3.0                                     /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

# HOST DEFINITION
#
#####
# Define a host for the local machine
define host {
    use             linux-server          ; Name of host template to use
                                ; This host definition will inherit all variables that are defined
                                ; in (or inherited by) the linux-server host template definition.
    host_name       linuxserver
    alias           localhost
    address         52.202.216.168
}

#####

```

Change hostgroup\_name under hostgroup to linux-servers1

```
GNU nano 5.8                                      /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

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

```

#### 7. Open the Nagios Config file and add the following line

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

```
##Add this line
```

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

## 8. Verify the configuration files

```
[root@ip-172-31-91-100 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

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

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

Checking objects...
  Checked 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
```

You are good to go if there are no errors.

## 9. Restart the nagios service

```
service nagios restart
```

## 10. SSH into the machine or simply use the EC2 Instance Connect feature.

```
applicable law.
```

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

```
ubuntu@ip-172-31-89-25:~$ sudo apt update -y
```

## 11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

```
sudo apt update -y
```

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

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

```
ubuntu@ip-172-31-89-25:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
```

## 12. Open nrpe.cfg file to make changes.

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

Under allowed\_hosts, add your nagios host IP address like so

## 13. Restart the NRPE server

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

## 14. Now, check your nagios dashboard and you'll see a new host being added.

Click on Hosts.

The screenshot shows the Nagios Core 4.4.6 dashboard. The top navigation bar indicates "Not secure" and the URL "3.83.138.228/nagios/". The main header features the "Nagios® Core™" logo. Below the header, a message says "Daemon running with PID 4764". The dashboard is divided into several sections:

- Current Status:** Includes links for "Tactical Overview", "Map (Legacy)", "Hosts", "Services", "Host Groups", "Grid", "Service Groups", "Grid", "Problems", "Services (Unhandled)", "Hosts (Unhandled)", "Normal Outages", and "Quick Search".
- Reports:** Includes links for "Availability", "Trends (Legacy)", "Alerts", "History", "Summary", "Histogram (Legacy)", "Notifications", and "Event Log".
- System:** Includes links for "Comments", "Downtime", "Process Info", and "Performance Info".
- Get Started:** A list of items including: Start monitoring your infrastructure, Change the look and feel of Nagios, Extend Nagios with hundreds of addons, Get support, Get training, and Get certified.
- Quick Links:** A list of links including: Nagios Library (tutorials and docs), Nagios Labs (development blog), Nagios Exchange (plugins and addons), Nagios Support (tech support), Nagios.com (company), and Nagios.org (project).
- Latest News:** A section showing the latest news updates.
- Don't Miss...** A section showing recent or important notifications.

A blue banner at the bottom left of the dashboard area says "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.5."

Now here we can see there is a host added

The screenshot shows the Nagios web interface at the URL 3.87.26.198/nagios/. The main dashboard displays 'Current Network Status' with a timestamp of 09-29-2024 13:18:44 UTC 2024. It includes 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0) and 'Service Status Totals' (Ok: 11, Warning: 1, Unknown: 0, Critical: 4, Pending: 0). Below these are two tables: 'Host Status Details For All Host Groups' and 'Service Status Details For All Host Groups'. The 'Host Status Details' table shows two hosts: 'linuxserver' (UP) and 'localhost' (UP). The 'Service Status Details' table shows various services for both hosts, with some entries in red indicating critical or warning states.

You can click Services to see all services and ports being monitored.

This screenshot shows the 'Service Status Details For All Hosts' table. It lists services for two hosts: 'linuxserver' and 'localhost'. For 'linuxserver', the 'HTTP' service is in a CRITICAL state due to permission errors. The 'Swap Usage' service is also in a CRITICAL state because no swap partition was created. For 'localhost', the 'HTTP' service is in a WARNING state. Other services like PING, SSH, and Root Partition are in OK states.

As you can see, we have our linuxserver up and running. It is showing critical status on

HTTP due to permission errors and swap because there is no partition created.

In this case, we have monitored -

Servers: 1 linux server

Services: swap

Ports: 22, 80 (ssh, http)

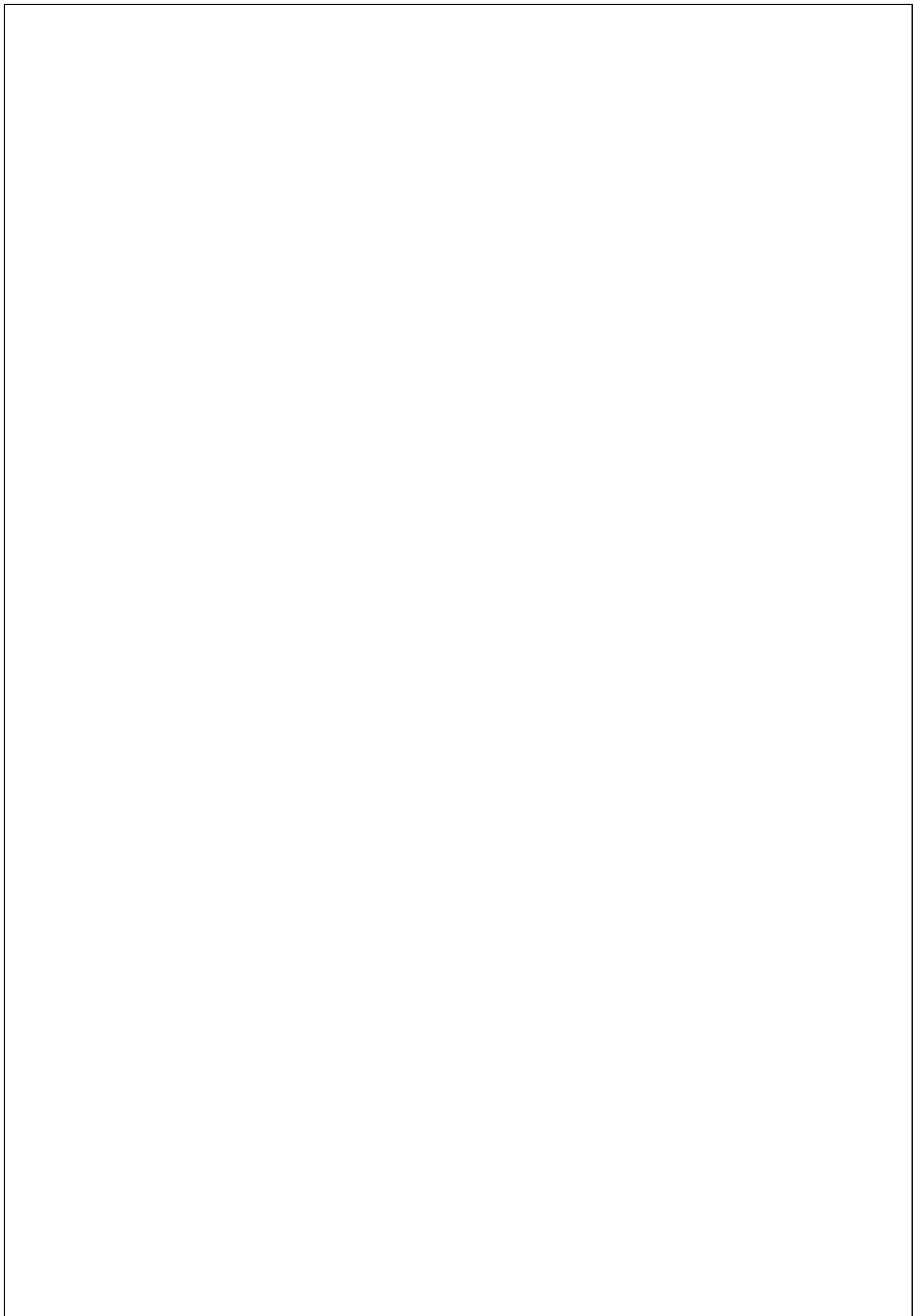
Processes: User status, Current load, total processes, root partition, etc.

Recommended Cleanup:

- Terminate both of your EC-2 instances to avoid charges.
- Delete the security group if you created a new one (it won't affect your bill, you may avoid it)

## Conclusion:

In this experiment, I successfully implemented Nagios for comprehensive port and service monitoring across both Windows and Linux servers. The setup allowed us to effectively track the status of critical services and ports, ensuring optimal performance and availability of server resources.



**Advanced DevOps Lab**

**Experiment No: 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.
- It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

## 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.

## Steps to create an AWS Lambda function

1. Open up the Lambda Console and click on the Create button.

Be mindful of where you create your functions since Lambda is region-dependent.

Functions (7)					
	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	<a href="#">RedshiftOverwatch</a>	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	<a href="#">lambdasnehal</a>	-	Zip	Python 3.12	4 days ago
<input type="checkbox"/>	<a href="#">RedshiftEventSubscription</a>	Create Redshift event subscription to SNS	Zip	Python 3.8	2 months ago

2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.

Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

## Create function Info

Choose one of the following options to create your function.

- Author from scratch  
Start with a simple Hello World example.

- Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.

- Container image  
Select a container image to deploy for your function.

### Basic information

#### Function name

Enter a name that describes the purpose of your function.

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

#### Runtime Info

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



#### Architecture Info

Choose the instruction set architecture you want for your function code.

#### Architecture Info

Choose the instruction set architecture you want for your function code.

- x86\_64

- arm64

### Permissions Info

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

#### ▼ Change default execution role

##### Execution role

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

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

##### Existing role

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



View the LabRole role on the IAM console.

Click on the *Create* button.

3. This process will take a while to finish and after that, you'll get a message that your function was successfully created.

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

Lambda > Functions > lambdasnehal

### lambdasnehal

Throttle
Copy ARN
Actions ▾

Function overview Info

Diagram
Template

lambdasnehal

+ Add trigger
+ Add destination

Description

-

Last modified

33 seconds ago

Function ARN

arn:aws:lambda:us-east-1:608111999703:function:lambdasnehal

Function URL Info

-

The screenshot shows the AWS Lambda code editor interface. At the top, a green banner displays the message: "Successfully created the function lambdasnehal. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below the banner, the navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. The main area shows the code source for the function "lambda\_function". The code is as follows:

```
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     }
```

4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.

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

The screenshot shows the "Edit basic settings" page for the function "lambdasnehal". The top navigation bar includes links for Services, Search, and N. Virginia. The left sidebar shows the function hierarchy: Lambda > Functions > lambdasnehal > Edit basic settings. The main content area is titled "Edit basic settings" and contains the "Basic settings" tab. The configuration fields are as follows:

- Description**: A text input field labeled "Description - optional".
- Memory**: A numeric input field set to 128 MB. A note states: "Your function is allocated CPU proportional to the memory configured."
- Ephemeral storage**: A numeric input field set to 512 MB. A note states: "You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing".
- SnapStart**: A dropdown menu set to "None". A note states: "Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations".
- Supported runtimes**: A note stating "Supported runtimes: Java 11, Java 17, Java 21".
- Timeout**: A note stating "Timeout".

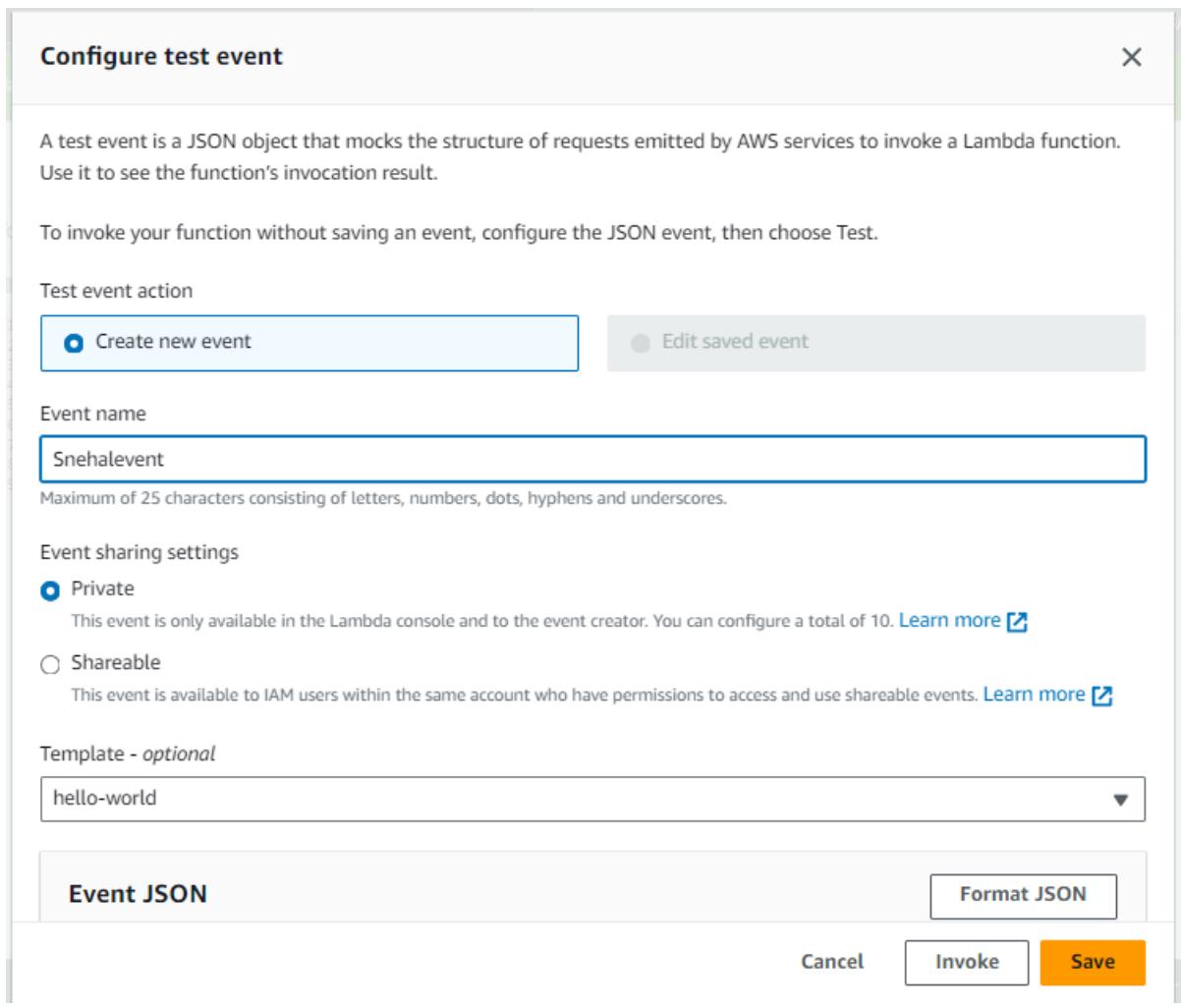
5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed.

Press Ctrl + S to save the file and click Deploy to deploy the changes.

The screenshot shows the AWS Lambda code editor interface. At the top, a green banner displays the message "Successfully updated the function lambdasnehal.". Below the banner, the "Code source" tab is selected, showing the "Info" sub-tab. The main area contains a code editor with the following Python code:

```
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
```

6. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



7. Now click on Test and you should be able to see the results.

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Test' tab is currently selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, Window, a dropdown for Test, Deploy, and an Upload from button. On the left, there's a sidebar labeled 'Environment' with a search bar and a dropdown menu. The main content area displays the 'lambda\_function' folder, which contains 'lambda\_function.py'. Underneath, the 'Execution results' section shows a successful test run. The 'Test Event Name' is set to 'SnehalEvent'. The 'Response' field shows the JSON output: {"statusCode": 200, "body": "Hello from Lambda!"}. The 'Function Logs' section provides detailed log information for the request, including RequestId, Version, Duration, Billed Duration, Memory Size, and Max Memory Used. The 'Request ID' is listed as 919e2620-ce61-462b-8909-3df14af543c.

## Conclusion:

In conclusion, through this implementation, I learned the fundamentals of AWS Lambda and gained a solid understanding of its workflow and functionalities. By creating my first Lambda functions using Python/Java/Node.js, I was able to see firsthand how serverless architecture operates and its benefits in terms of scalability and efficiency. This hands-on experience has deepened my knowledge and equipped me with the skills to build responsive applications while minimizing infrastructure management. Overall, I feel more confident in leveraging AWS Lambda for future projects and exploring its potential further.

**Advanced Devops Lab**  
**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:

- o 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:

- o Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.

- o Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

- o 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:

- o 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:

- o 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

## Create bucket Info

Buckets are containers for data stored in S3.

### General configuration

#### AWS Region

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

Bucket type Info

General purpose

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

Directory

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

Bucket name Info

snehallambdabuck

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

⌚ Successfully created bucket "snehallambdabuck"

[View details](#)

Amazon S3 > Buckets

► Account snapshot - updated every 24 hours All AWS Regions

[View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (2) Info All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

[Create bucket](#)

< 1 > ⌂

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">elasticbeanstalk-us-east-1-608111999703</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 18, 2024, 16:04:27 (UTC+05:30)
<a href="#">snehallambdabuck</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 3, 2024, 14:54:58 (UTC+05:30)

## Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Container image

Select a container image to deploy for your function.

### Basic information

**Function name**

Enter a name that describes the purpose of your function.

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

**Runtime** Info

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



**Architecture** Info

Choose the instruction set architecture you want for your function code.

x86\_64

arm64

[CloudShell](#)
[Feedback](#)

© 2024, Amazon Web Services

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



## Snehalimageloader

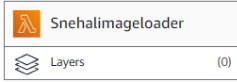
[Throttle](#)
 [Copy ARN](#)
[Actions ▾](#)

**Function overview** Info

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

Diagram

Template


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

Description

Last modified

3 seconds ago

Function ARN

arn:aws:lambda:us-east-1:608111999703:function:Snehalimageloader

Function URL Info

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

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

**Code source** Info

[File](#)
[Edit](#)
[Find](#)
[View](#)
[Go](#)
[Tools](#)
[Window](#)
[Test](#)
[Deploy](#)
[Changes not deployed](#)
[Go to Anything \(Ctrl-P\)](#)
[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
6     bucket_name = event['Records'][0]['s3']['bucket']['name']
7     object_key = event['Records'][0]['s3']['object']['key']
8
9     # Log a message
10    print(f"An Image has been added to the bucket (bucket_name): (object_key)")
11
12    return {
13        'statusCode': 200,
14        'body': json.dumps('Log entry created successfully')
15    }
16
17
18
19

```

## Trigger configuration Info

S3  
aws asynchronous storage

### Bucket

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



Bucket region: us-east-1

### Event types

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

All object create events

### Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

e.g. images/

### Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any [special characters](#) must be URL encoded.

e.g. jpg

[Lambda](#) > [Functions](#) > Snehalimageloader

## Snehalimageloader

Throttle

Copy ARN

Actions ▾

The trigger snehallambdabuck was successfully added to function Snehalimageloader. The function is now receiving events from the trigger.



### Function overview Info

[Export to Application Composer](#)

[Download](#) ▾

Diagram

Template



#### Description

Last modified  
8 minutes ago

Function ARN  
 arn:aws:lambda:us-east-1:608111999703:function:Snehalimageloader

Function URL [Info](#)

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

### General configuration

### Triggers

### Permissions

### Destinations

### Function URL

### Environment variables

### Tags

### VPC

### RDS databases

### Monitoring and operations tools

### Concurrency and recursion detection

### Triggers (1) Info



[Fix errors](#)

[Edit](#)

[Delete](#)

[Add trigger](#)

< 1 >

Find triggers

Trigger



S3: [snehallambdabuck](#)

arn:aws:s3:::snehallambdabuck

Details

RDS databases

Monitoring and operations tools

Concurrency and recursion detection

Asynchronous invocation

Code signing

File systems

State machines

**Resource-based policy statements (1) [Info](#)**

Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.

Statement ID	Principal	PrincipalOrgID	Conditions	Action
lambda-b9cc7374-337f...	s3.amazonaws.com	-	StringEquals, ArnLike	lambda:InvokeFunction

**Auditing and compliance**

AWS CloudTrail can log this function's invocations for operational and risk auditing, governance, and compliance. [Get started](#) on the CloudTrail console.

Amazon S3 > Buckets > snehallambdabuck > Upload

**Upload [Info](#)**

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

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

**Files and folders (1 Total, 16.7 KB)**

All files and folders in this table will be uploaded.

Name	Folder	Type
image.png	-	image/png

**Destination [Info](#)**

Destination  
<s3://snehallambdabuck>

**Upload succeeded**  
View details below.

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

**Summary**

Destination	Succeeded	Failed
s3://snehallambdabuck	1 file, 16.7 KB (100.00%)	0 files, 0 B (0%)

**Files and folders (1 Total, 16.7 KB)**

Name	Folder	Type	Size	Status	Error
image.png	-	image/png	16.7 KB	Succeeded	-

The screenshot shows two views of the AWS CloudWatch service. The top view displays the 'Log groups' page, showing a list of five log groups: '/aws/lambda/RedshiftEventSubscription', '/aws/lambda/RedshiftOverwatch', '/aws/lambda/RoleCreationFunction', '/aws/lambda/SnehalImageLoader', and '/aws/lambda/lambdadasnehal'. The bottom view shows the 'Log events' page for the '/aws/lambda/SnehalImageLoader' log group, dated 2024/10/10. It lists several log entries, including the initiation of a Lambda function, its execution, and the successful upload of an image to an S3 bucket.

## 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. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.