

Experiment 1

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

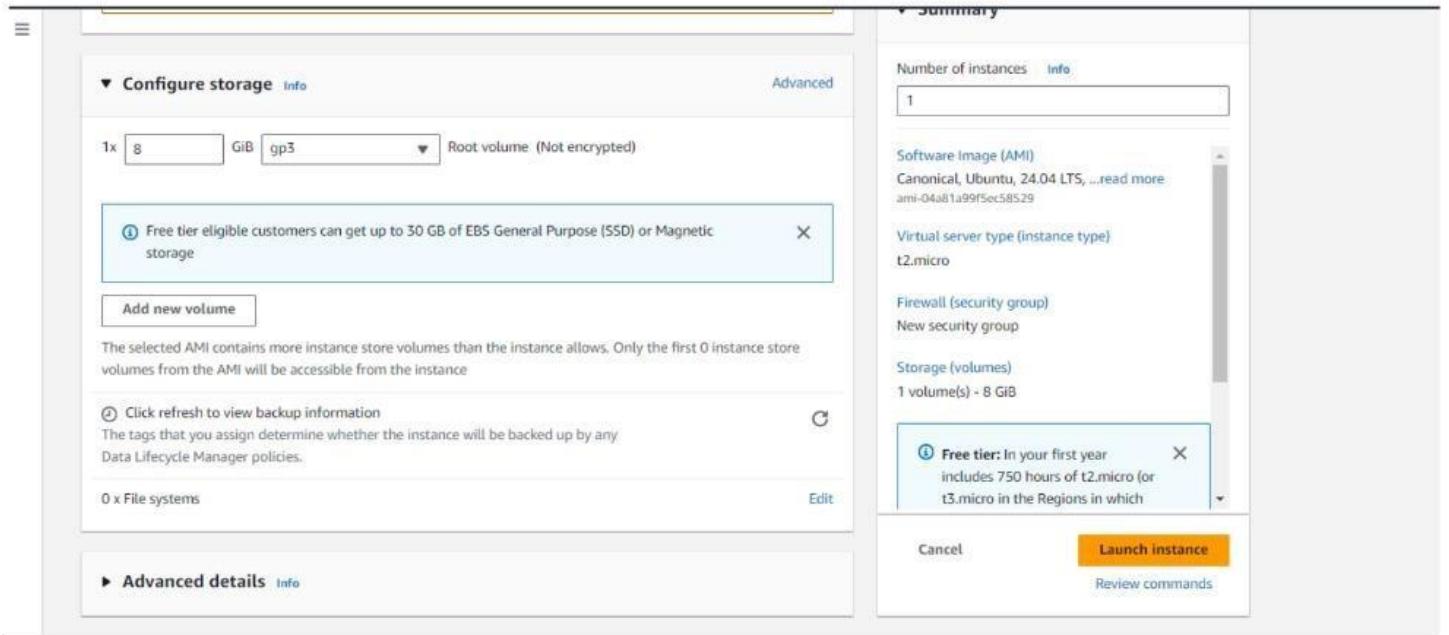
1. Open AWS Academy and select launch instance

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Global View, Events, Console-to-Code Preview, Instances, Images, and Elastic Block Store. The main area has a 'Resources' section with a table showing 1 instance (running), 1 auto scaling group, 0 dedicated hosts, 1 elastic IP, 0 key pairs, 0 load balancers, 2 security groups, 0 snapshots, 1 instance, 0 placement groups, and 1 volume. Below this is a 'Launch instance' section with a large orange 'Launch instance' button. To the right is a 'Service health' section showing 'AWS Health Dashboard' and 'US East (N. Virginia)' status as 'operating normally'. Further right are sections for 'EC2 Free Tier Info', 'Account attributes', 'Default VPC' (set to vpc-0fcb2f3fa22f1a71), 'Settings' (Data protection and security, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences), and 'Additional information'.

2. And then select Ubuntu or Linux.

This screenshot shows the 'Quick Start' section of the AWS AMI selection interface. It features a grid of icons for different operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. Below the grid, a search bar says 'Browse more AMIs' and 'Including AMIs from AWS, Marketplace and the Community'. A specific AMI is highlighted: 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' (ami-04a81a99f5ec58529). The description notes it's 'Free tier eligible'. Other details include 'Virtualization: hvm' and 'ENAv2 enabled: true'. The 'Architecture' dropdown is set to '64-bit (x86)'. To the right, a 'Summary' panel shows 'Number of instances: 1', 'Software Image (AMI): Canonical, Ubuntu, 24.04 LTS...', 'Virtual server type (instance type): t2.micro', 'Firewall (security group): New security group', and 'Storage (volumes): 1 volume(s) - 8 GiB'. A tooltip for the free tier says: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch)'. At the bottom are 'Cancel' and 'Launch instance' buttons.

3. Set the configuration and then connect to the instance which you created.



4. Execute the following commands in the aws console.

Commands :

```
sudo su
```

```
sudo apt install
```

```
sudo apt-get update
```

```
apt install apache2
```

```
systemctl status apache2
```

```
cd /var/www/html/
```

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

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

System information disabled due to load higher than 1.0

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-17-139:~$ sudo su
root@ip-172-31-17-139:/home/ubuntu# sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
i-Of7cedaab7d390e14 (My Web Server)
PublicIPs: 3.91.6.193 PrivateIPs: 172.31.17.139
```

```
root@ip-172-31-17-139:/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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [265 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [63.3 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [3668 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [247 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [107 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 [9220 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [208 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [40.7 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [420 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [318 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [82.9 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5676 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [319 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [134 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.6 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [208 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [208 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [40.7 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 6s (5073 kB/s)
Reading package lists... Done
root@ip-172-31-17-139:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 42 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.lubuntu7 [91.9 kB]
```

```

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 libaprilt64 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.lubuntu7 [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.lubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.lubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblulu5.4-0 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-lubuntu8.4 [1329 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-lubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-lubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-lubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntul [17.8 kB]
Fetched 2083 kB in 0s (25.8 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libaprilt64:amd64.
(Reading database ... 67739 files and directories currently installed.)
Preparing to unpack .../0-libaprilt64_1.7.2-3.1build2_amd64.deb ...
Unpacking libaprilt64:amd64 (1.7.2-3.1build2) ...
Selecting previously unselected package libaprutil1t64:amd64.
Preparing to unpack .../1-libaprutil1t64_1.6.3-1.lubuntu7_amd64.deb ...
Unpacking libaprutil1t64:amd64 (1.6.3-1.lubuntu7) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
Preparing to unpack .../2-libaprutil1-dbd-sqlite3_1.6.3-1.lubuntu7_amd64.deb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.lubuntu7) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../3-libaprutil1-ldap_1.6.3-1.lubuntu7_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.3-1.lubuntu7) ...
Selecting previously unselected package liblulu5.4-0:amd64.
Preparing to unpack .../4-liblulu5.4-0_5.4.6-3build2_amd64.deb ...
Unpacking liblulu5.4-0:amd64 (5.4.6-3build2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../5-apache2-bin_2.4.58-lubuntu8.4_amd64.deb ...
Unpacking apache2-bin (2.4.58-lubuntu8.4) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../6-apache2-data_2.4.58-lubuntu8.4_all.deb ...

Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-17-139:/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 Wed 2024-08-07 14:32:03 UTC; 32s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2487 (apache2)
   Tasks: 55 (limit: 1130)
  Memory: 5.4M (peak: 5.6M)
    CPU: 37ms
   CGroup: /system.slice/apache2.service
           ├─2487 /usr/sbin/apache2 -k start
           ├─2490 /usr/sbin/apache2 -k start
           └─2491 /usr/sbin/apache2 -k start

Aug 07 14:32:03 ip-172-31-17-139 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 07 14:32:03 ip-172-31-17-139 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-17-139:/home/ubuntu# cd /var/www/html/
root@ip-172-31-17-139:/var/www/html# 
```

5. Edit the inbound and outbound rules.

The screenshot shows the AWS EC2 Security Groups Details page for a security group named "sg-0501f07360e1dce47 - launch-wizard-1". The page includes the following details:

- Security group name:** launch-wizard-1
- Security group ID:** sg-0501f07360e1dce47
- Description:** launch-wizard-1 created 2024-08-07T14:21:19.108Z
- VPC ID:** vpc-0ec7dea564d6f7acf
- Owner:** 856746069793
- Inbound rules count:** 1 Permission entry
- Outbound rules count:** 1 Permission entry

The **Inbound rules** section shows a table with one row:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-09fa86395ec8777e3	IPv4	HTTP	TCP	80

EC2 Dashboard

EC2 Global View

Events

Console-to-Code [Preview](#)

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes
- Snapshots
- Lifecycle Manager

Network & Security

- Security Groups
- Elastic IPs

Details

sg-0501f07360e1dce47 - launch-wizard-1

Details

Security group name launch-wizard-1	Security group ID sg-0501f07360e1dce47	Description launch-wizard-1 created 2024-08-07T14:21:19.108Z	VPC ID vpc-0ec7dea564d6f7acf
Owner 856746069793	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Outbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0cc59fddd03e24266	IPv4	HTTP	TCP	80

6. This is the hosted Static Website.

Not secure | 3.91.6.193

Apache2 Default Page

Ubuntu

It works!

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

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

Configuration Overview

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

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

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

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain

Using S3

1. Now visit S3 under the developer tools and create a Bucket and then Click on the Edit Static Website Hosting under the properties tab

aws Services Search [Alt+S] Stockholm SnehaPatra

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-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
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

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

aws Services Search [Alt+S] Stockholm SnehaPatra

Amazon S3 > Buckets

Successfully created bucket "snehabucky"
To upload files and folders, or to configure additional bucket settings, choose View details.

View details X

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 Info All AWS Regions | Directory buckets

General purpose buckets (3) Info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
codepipeline-eu-north-1-233300349215	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 22, 2024, 19:54:58 (UTC+05:30)
elasticbeanstalk-eu-north-1-337909754951	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 22, 2024, 13:27:27 (UTC+05:30)
snehabucky	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 22, 2024, 23:49:21 (UTC+05:30)

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

aws Services Search [Alt+S] Stockholm SnehaPatra

Amazon S3

Amazon S3 > Buckets > snehabucky > Edit static website hosting

Edit static website hosting Info

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

Static website hosting
 Disable
 Enable

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

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

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

Successfully edited static website hosting.

Amazon S3 > Buckets > snehabucky

snehabucky [Info](#)

Objects [Properties](#) Permissions Metrics Management Access Points

Bucket overview

AWS Region Europe (Stockholm) eu-north-1	Amazon Resource Name (ARN) arn:aws:s3:::snehabucky	Creation date August 22, 2024, 23:49:21 (UTC+05:30)
---	---	--

Bucket Versioning

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

Bucket Versioning

Bucket Versioning [Edit](#)
Disabled

Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

2. Upload your html file.

Amazon S3 > Buckets > snehabucky > 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, 261.0 B)

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

Destination [Info](#)

Destination
s3://snehabucky

► Destination details

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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://snehabucky	1 file, 261.0 B (100.00%)	0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Files and folders (1 Total, 261.0 B)

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	261.0 B	Succeeded	-

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

3. And then select on the Edit block public access under the Permissions tab.

The screenshot shows the 'Edit Block public access (bucket settings)' page. On the left, a sidebar lists various S3 features like Buckets, Storage Lens, and Feature spotlight. The main content area is titled 'Block public access (bucket settings)' and contains a detailed description of how public access is granted through access control lists (ACLs), bucket policies, and access point policies. It includes four checkboxes for different access levels:

- 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.

At the bottom right are 'Cancel' and 'Save changes' buttons.

4. Select Object Ownership under Permission Tab

The screenshot shows the 'Edit Object Ownership' page. The sidebar is identical to the previous screenshot. The main content area is titled 'Object Ownership' and contains two radio button options:

- 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.

Below these options is a warning message: "⚠ 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." At the bottom of this section is a checkbox: "I acknowledge that ACLs will be restored." The footer includes links for CloudShell, Feedback, and cookie preferences.

5. Select the file and click on Actions and select the option Make Public using ACL from the dropdown

The screenshot shows the 'Objects' tab of the S3 bucket 'snehabucky'. The sidebar is identical. The main content area displays a list of objects with a table header: Name, Type, Last modified. One object, 'index.html', is selected. A context menu is open over this object, showing options like Copy, Move, Initiate restore, Query with S3 Select, Edit actions, Rename object, Edit storage class, Edit server-side encryption, Edit metadata, Edit tags, and 'Make public using ACL'.

The screenshot shows the AWS S3 console with a green header bar indicating a successful edit of public access. Below the header, a summary table shows the status of edits across three categories: Source, Successfully edited public access, and Failed to edit public access. A link to view details is provided for the successfully edited category. At the bottom, there are tabs for Failed to edit public access and Configuration, and a search bar.

Source	Successfully edited public access	Failed to edit public access
s3://snehabucky	Successfully edited public access 1 object, 261.0 B	Failed to edit public access 0 objects

6. Now you can visit the domain and the website hosted.

The screenshot shows a web browser window displaying a simple HTML page. The URL in the address bar is `snehabucky.s3.eu-north-1.amazonaws.com/index.html`. The page content is "Hello World. I am Sneha Patra From D15A!!".

Dynamic Hosting :

Step 1: Clone the following Github repository: <https://github.com/ljharb/define-data-property>

The screenshot shows the GitHub repository page for 'ljharb / define-data-property'. The repository is public and has 43 commits. The commits are listed in reverse chronological order, starting from 'ljharb [Dev Deps] update @ljharb/tsconfig' (a622c13 - 3 weeks ago) and ending with '[Dev Deps] update @ljharb/tsconfig' (3 weeks ago). The repository has 4 stars, 2 forks, and 0 forks. It includes tags for javascript, data, object, ecmascript, property, enumerable, configurable, accessor, define, and writable.

Step 2: Open Console and run the following command

```
root@ip-172-31-55-145:/home/ubuntu/dynamic/dyanamic_site# npm i
(██████████) :: reify:define-data-property: http fetch GET 200 https://registry.npmjs.org/define-data-prop
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
```

Step 3: Install necessary packages and run the website on port number 3000.

The screenshot shows a web browser displaying the message 'Hey this is Dynamic Website.' This indicates that the website is successfully running on port 3000.

Hey this is about page.

IDE Hosting :

Step 1: Go to AWS Academy and open AWS Cloud9 from developer Tools and select create environment.

The screenshot shows the AWS Cloud9 landing page. At the top right, there is a call-to-action button labeled "Create environment". Below the button, there is a section titled "New AWS Cloud9 environment". The main content area features the AWS Cloud9 logo and the tagline "A cloud IDE for writing, running, and debugging code". A descriptive paragraph explains that AWS Cloud9 allows you to write, run, and debug your code with just a browser. It highlights immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. The page also includes a "How it works" section and a "Getting started" sidebar with links to various documentation pages.

Step 2: Create a environment

The screenshot shows the "Create environment" wizard in AWS Cloud9. The first step, "Details", is selected. In the "Name" field, the value "snehaenvi" is entered. The "Description - optional" field is empty. Under "Environment type", the "New EC2 instance" option is selected, which is highlighted with a blue border. The "Existing compute" option is also available but not selected. Below the environment type section, there is a "New EC2 instance" summary card. At the bottom of the screen, there are navigation links for "CloudShell", "Feedback", and copyright information.



30 minutes



Network settings Info

Connection

How your environment is accessed.

AWS Systems Manager (SSM)

Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)

Accesses environment directly via SSH, opens inbound ports.

▶ VPC settings Info

Tags - optional Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- **AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

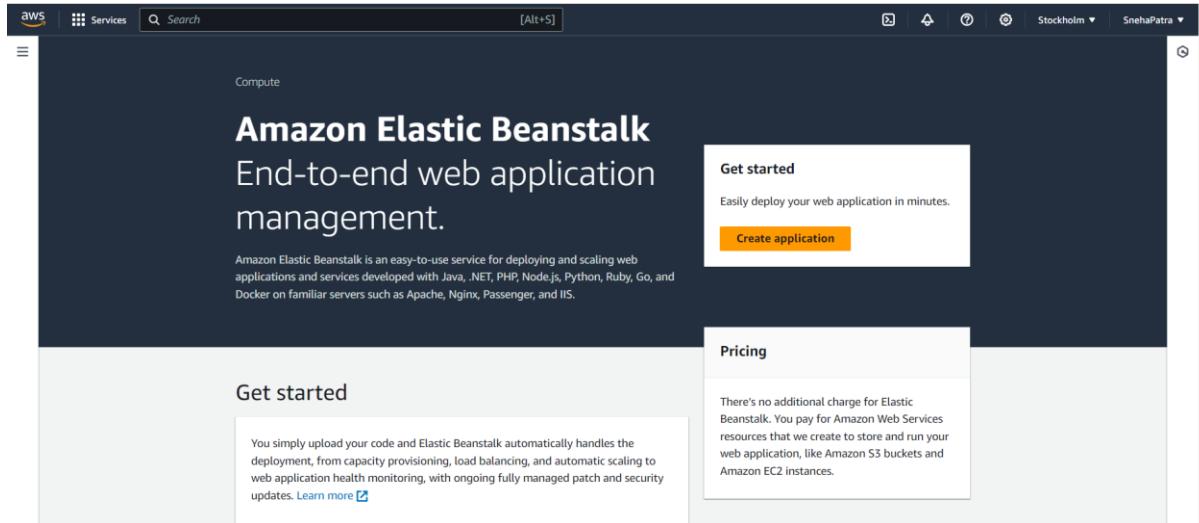
Cancel

Create

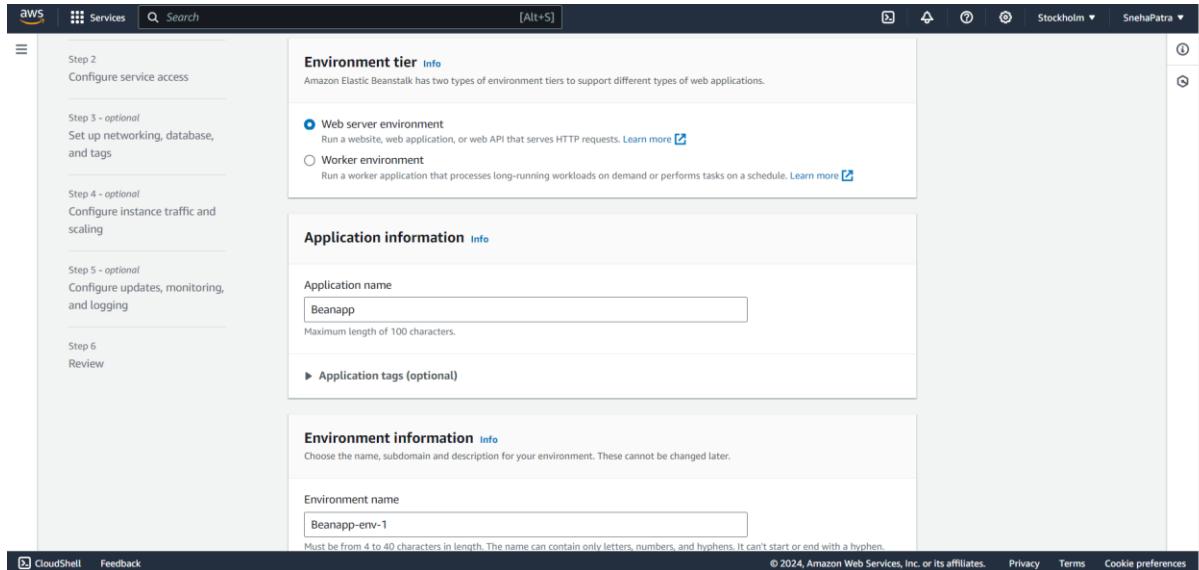
EXPERIMENT 2- BEANSTALK

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

1. Open the aws console and then search Elastic Beanstalk and click on create application.



2. Configure the environment by adding your application name.



3. Choose PHP and click next.

The screenshot shows the AWS Elastic Beanstalk configuration interface. The top navigation bar includes the AWS logo, Services, a search bar, and user information (Stockholm, SnehaPatra). The main content area is divided into sections:

- Platform info**:
 - Platform type: Managed platform (Platforms published and maintained by Amazon Elastic Beanstalk)
 - Custom platform (Platforms created and owned by you. This option is unavailable if you have no platforms)
- Platform**: PHP
- Platform branch**: PHP 8.3 running on 64bit Amazon Linux 2023
- Platform version**: 4.3.2 (Recommended)
- Application code Info**:
 - Sample application
 - Existing version (Application versions that you have uploaded)
 - Upload your code (Upload a source bundle from your computer or copy one from Amazon S3)
- Presets Info**:
 - Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
 - Configuration presets:
 - Single instance (free tier eligible)
 - Single instance (using spot instance)
 - High availability
 - High availability (using spot and on-demand instances)
 - Custom configuration

At the bottom right, there are "Cancel" and "Next" buttons, with "Next" being highlighted in orange.

4. Add your key pair and also select EC2 instance profile.

The screenshot shows the AWS Elastic Beanstalk configuration interface, specifically the "Configure service access" step. The left 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 current step is Step 2.

The main content area is titled "Configure service access" and includes the following fields:

- 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.
Selected role: aws-elasticbeanstalk-service-role
- EC2 key pair**: Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)
Selected key pair: AppKey
- EC2 instance profile**: Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
Selected profile: new-user

At the bottom right, there are "Cancel", "Skip to review", "Previous", and "Next" buttons, with "Next" being highlighted in orange.

5. After selecting skip to review then select submit.

The screenshot shows the AWS Elastic Beanstalk environment configuration page. The main content area displays environment properties:

Key	Value
Allow URL fopen	On
Max execution time	60
Proxy server	nginx
Update level	minor

Below this, there is a section for "Environment properties" which is currently empty, displaying the message "No environment properties".

At the bottom right, there are "Cancel", "Previous", and "Submit" buttons. The "Submit" button is highlighted in orange.

6. Your sample environment is created for you to deploy your application.

The screenshot shows the AWS Elastic Beanstalk environment overview page for "Beanapp-env".

Environment overview:

Health	Environment ID
No Data - View causes	e-ymmkztzg8u

Platform:

Platform
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2

Events: (25) Info

Filter events by text, property or value

At the bottom, there are navigation links for CloudShell, Feedback, and Copyright information.

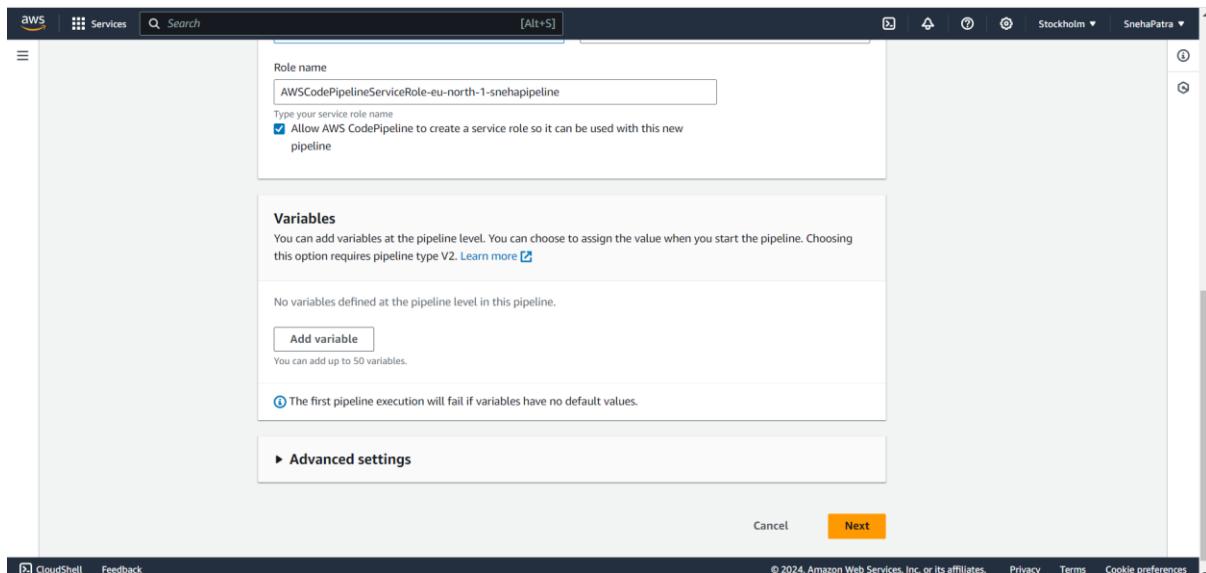
NOW LET US CREATE PIPELINE:

1. Select CodePipeline and create a new pipeline

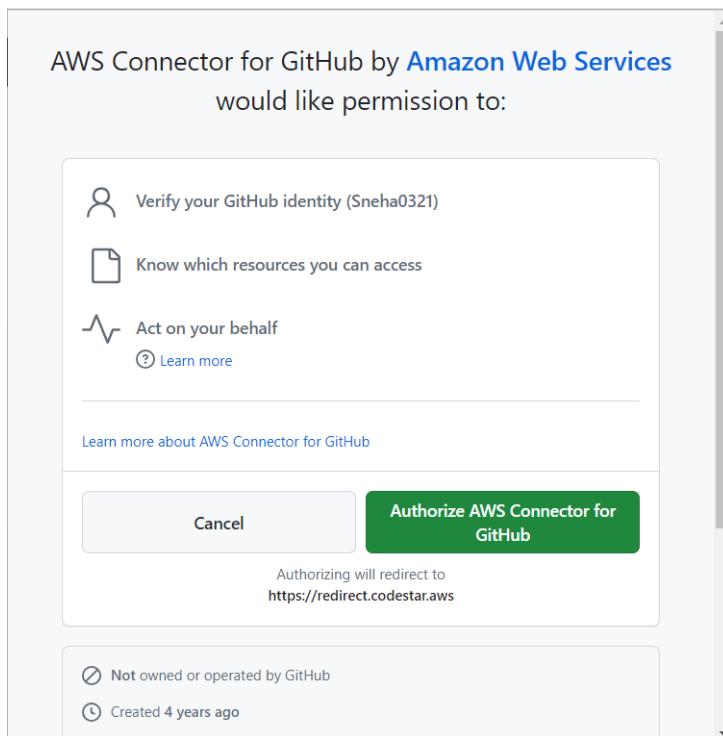
The screenshot shows the AWS CodePipeline Pipelines page. On the left, there is a navigation sidebar with sections like Source, Artifacts, Build, Deploy, Pipeline, and Settings. Under Pipeline, 'Getting started' and 'Pipelines' are listed, with 'Pipelines' being the active tab. The main content area has a heading 'Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more'. Below this is a table with columns: Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A message 'No results' and 'There are no results to display.' is shown. At the top right of the main area is a 'Create pipeline' button. The bottom of the screen shows standard AWS navigation and footer links.

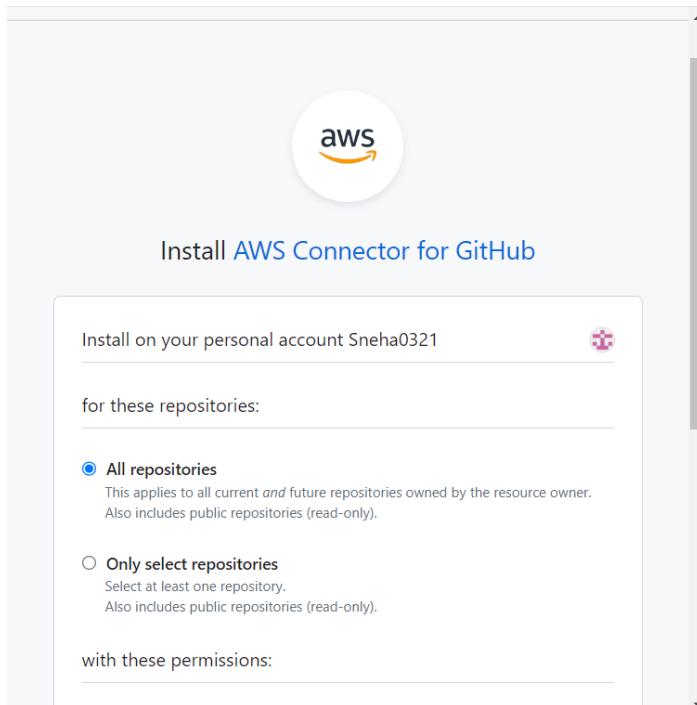
2. Name Your pipeline.

The screenshot shows the 'Choose pipeline settings' step of the AWS CodePipeline pipeline creation wizard. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main panel is titled 'Pipeline settings' and contains a 'Pipeline name' field with the value 'sneahipipeline'. Below it is a note: 'Pipeline type: You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.' Under 'Execution mode', there are three options: 'Superseded' (radio button unselected), 'Queued (Pipeline type V2 required)' (radio button selected), and 'Parallel (Pipeline type V2 required)' (radio button unselected). The bottom of the screen shows a Windows taskbar with various icons and system status information.



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





The screenshot shows the 'Create connection' page for GitHub. The URL in the browser bar is 'eu-north-1.console.aws.amazon.com/codesuite/settings/connections/create/github?re...'. The page has a header with the AWS logo and navigation links for Services, Developer Tools, and Create connection. The main content area is titled 'Connect to GitHub' and contains a 'GitHub connection settings' section. It includes fields for 'Connection name' (set to 'myconnection'), an optional 'App installation' search bar (containing '53902292') with an 'Install a new app' button, and an optional 'Tags' section. At the bottom right is a large orange 'Connect' button. The footer of the page includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice: '© 2024, Amazon Web Services, Inc. or its affiliates.'

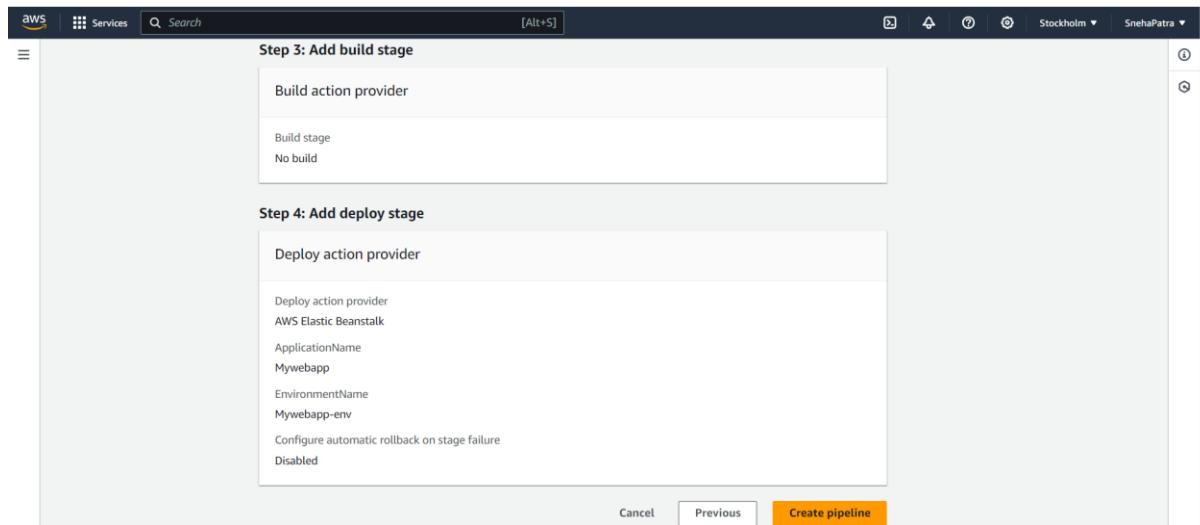
5. After establishing connection select repository and branch name.

The screenshot shows the AWS CodePipeline interface for connecting to GitHub. It includes fields for Repository name (Sneha0321/aws-codepipeline-s3-codedeploy-linux-2.0), Default branch (master), and Output artifact format (CodePipeline default). A green box indicates the GitHub connection is ready for use.

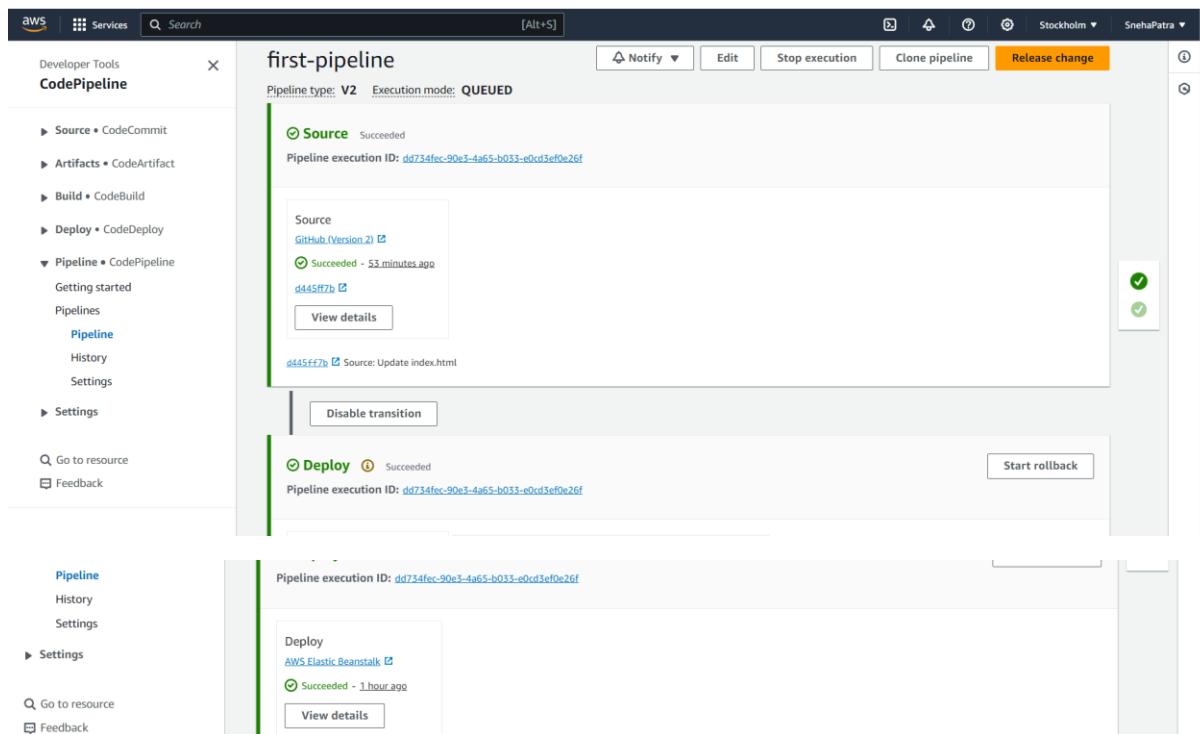
6. Add deploy stage.

This screenshot shows the 'Add deploy stage' step 4 of 5. It includes a note that you cannot skip the stage and must choose a provider. The Deploy section is shown, with Deploy provider set to AWS Elastic Beanstalk, Region set to US East (N. Virginia), and Application name left blank.

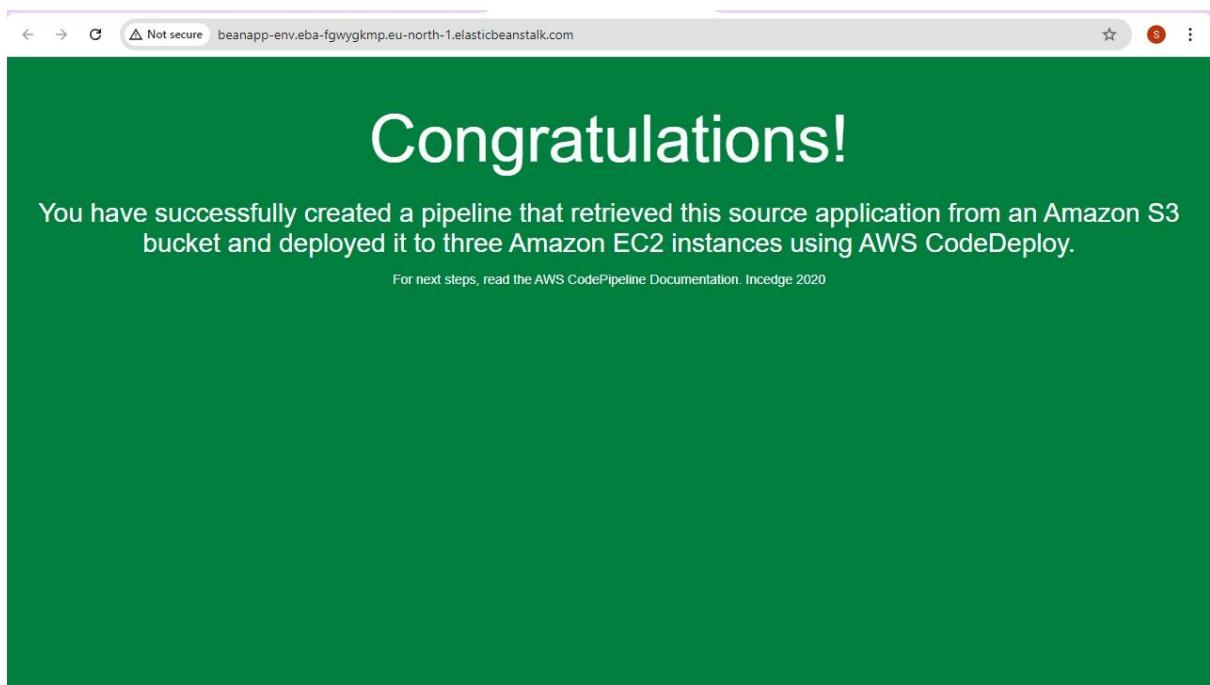
7. Review your pipeline.



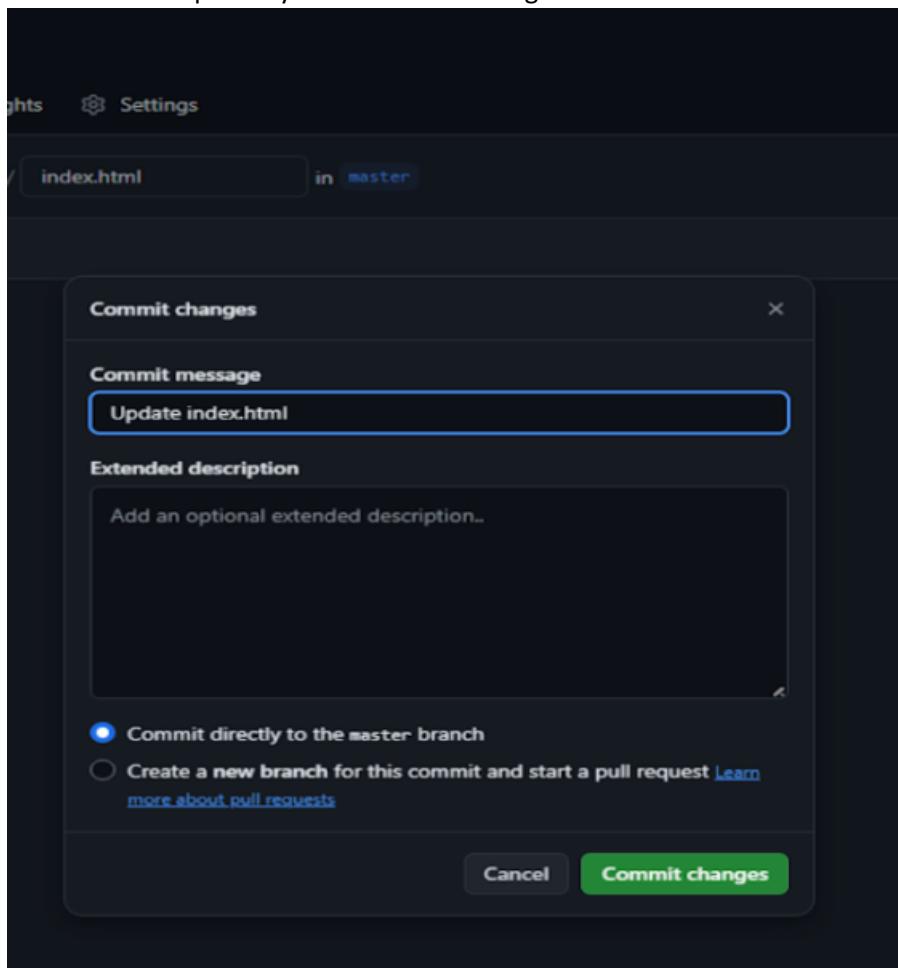
8. Now your pipeline is created.



9. This is the website hosted in our beanstalk environment.



10. Go to the repository and make the changes in the index.html file and commit them



11.The changes that are committed will reflect on url once you refreshed it.

The image shows two screenshots illustrating the deployment process. The top screenshot is from the AWS CodePipeline console, displaying the 'Pipelines' page. It lists a single pipeline named 'first-pipeline' which has just completed a successful execution. The bottom screenshot is a web browser window showing the deployed application at the URL beanapp-env.eba-fgwygkmp.eu-north-1.elasticbeanstalk.com. The page displays a large green banner with the text: 'Congratulations! This is my first deployment-Sneha Patra D15A 41'. Below the banner, a message states: 'You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.' A small note at the bottom of the banner reads: 'For next steps, read the AWS CodePipeline Documentation. Inedge 2020'.

Advanced DevOps Experiment 3

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

Step 1: Go to AWS Academy in services select EC2 and create 3 instance with and name them as master, node1, node2 and remember to select instance type as t2.medium.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Node1	i-04cd1f76e6a0b61a8	Terminated	t2.micro	-	View alarms +	us-east-1c	-
Master	i-0d0503cebb680a65b	Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-34-23-
Node1	i-030b5f6226fb1efdf	Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-3-86-2
Node2	i-02366d960a4501365	Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-3-83-8

EC2 > Instances > i-0365889e6b3d22cdb > Connect to instance

Connect to instance Info

Connect to your instance i-0365889e6b3d22cdb (Mynewapp-env) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Port 22 (SSH) is not authorized
Port 22 (SSH) is currently not authorized by your security group. To use EC2 Instance Connect, you must authorize port 22 for the EC2 Instance Connect service IP addresses in your Region: 18.206.107.24/29.
[Learn more](#)

Instance ID:

Connection Type:

- Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address:

Username:
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

Step 2: Then, Select and connect each instance and run the following commands inside the console of each instance.

- sudo su
- yum install docker -y
- systemctl start docker
- docker –version
- yum repolist

```

aws Services Search [Alt+S] N. Virginia v voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 %
[ec2-user@ip-172-31-44-214 ~]$ sudo su
[root@ip-172-31-44-214 ec2-user]# yum install docker -y
Last metadata expiration check: 0:14:58 ago on Wed Sep 18 13:11:57 2024.
Dependencies resolved.
=====
          Package           Architecture     Version      Repository  Size
=====
Installing:
  docker                  x86_64        25.0.6-1.amzn2023.0.2      amazonlinux  44 M
Installing dependencies:
  containerd               x86_64        1.7.20-1.amzn2023.0.1      amazonlinux  35 M
  iptables-libs             x86_64        1.8.8-3.amzn2023.0.2      amazonlinux  401 k
  iptables-nft              x86_64        1.8.8-3.amzn2023.0.2      amazonlinux  183 k
  libcgroup                x86_64        3.0-1.amzn2023.0.1      amazonlinux  75 k
  libnetfilter_conntrack    x86_64        1.0.8-2.amzn2023.0.2      amazonlinux  58 k
  libnftnl                 x86_64        1.0.1-19.amzn2023.0.2     amazonlinux  30 k
  libnftnl                 x86_64        1.2.2-2.amzn2023.0.2     amazonlinux  84 k
  pigz                     x86_64        2.5-1.amzn2023.0.3      amazonlinux  83 k
  runc                     x86_64        1.1.13-1.amzn2023.0.1     amazonlinux  3.2 M
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm  3.0 MB/s | 401 kB   00:00

i-030d25988270b63ec (Master)
PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

```

```

aws Services Search [Alt+S] N. Virginia v voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 %
=====
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64          9/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64       10/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying   : containerd-1.7.20-1.amzn2023.0.1.x86_64      1/10
Verifying   : docker-25.0.6-1.amzn2023.0.2.x86_64       2/10
Verifying   : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  3/10
Verifying   : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  4/10
Verifying   : libcgroup-3.0-1.amzn2023.0.1.x86_64       5/10
Verifying   : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying   : libnftnl-1.0.1-19.amzn2023.0.2.x86_64     7/10
Verifying   : libnftnl-1.2.2-2.amzn2023.0.2.x86_64     8/10
Verifying   : pigz-2.5-1.amzn2023.0.3.x86_64          9/10
Verifying   : runc-1.1.13-1.amzn2023.0.1.x86_64        10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64      docker-25.0.6-1.amzn2023.0.2.x86_64      iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64      libcgroup-3.0-1.amzn2023.0.1.x86_64      libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnftnl-1.0.1-19.amzn2023.0.2.x86_64        libnftnl-1.2.2-2.amzn2023.0.2.x86_64      pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-44-214 ec2-user]# systemctl start docker
[root@ip-172-31-44-214 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-44-214 ec2-user]# 

i-030d25988270b63ec (Master)
PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

```

Step 3: Now, visit the following link <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/> and scroll down till you find Red-Hat and then select Red-Hat based distributions tab copy all the commands one by one in each console of instance.

Search this site
Documentation
Getting started
Learning environment
Production environment
Container Runtimes
Installing Kubernetes with deployment tools
Bootstrapping clusters with kubeadm
Installing kubeadm
Troubleshooting kubeadm
Creating a cluster with kubeadm
Customizing components with the kubeadm API
Options for Highly Available Topology

Debian-based distributions Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

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

Caution:

- Setting SELinux in permissive mode by running `setenforce 0` and `sed ...` effectively disables it. This is required to allow containers to access the host filesystem; for example, some cluster network plugins require that. You have to do this until SELinux support is improved in the kubelet.
- You can leave SELinux enabled if you know how to configure it but it may require settings that are not supported by kubeadm.

2. Add the Kubernetes `yum` repository. The `exclude` parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running `yum update` as there's a special procedure that must be followed for upgrading Kubernetes. Please note that this repository have packages only for Kubernetes 1.31; for other Kubernetes minor versions, you need to change the Kubernetes minor version in the URL to match your desired minor version (you should also check that you are reading the documentation for the version of Kubernetes that you plan to install).

 Edit this page Create child page Create documentation issue Print entire section

Before you begin

Verify the MAC address and product_uuid are unique for every node

Check network adapters

Check required ports

Swap configuration

Installing a container runtime

Installing kubeadm, kubelet and kubectl

Configuring a cgroup driver

Troubleshooting

What's next

aws | Services | [Alt+S] | N. Virginia ▾ | vodlabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾ |

```
Complete!
[root@ip-172-31-44-214 ec2-user]# systemctl start docker
[root@ip-172-31-44-214 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-44-214 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livelpatch                       Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-44-214 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-44-214 ec2-user]# cat <>EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-44-214 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

i-030d25988270b63ec (Master) X
 PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-44-214 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.

=====
Package           Architecture      Version       Repository   Size
=====
Installing:
kubeadm          x86_64          1.31.1-150500.1.1   kubernetes   11 M
kubectl          x86_64          1.31.1-150500.1.1   kubernetes   11 M
kubelet          x86_64          1.31.1-150500.1.1   kubernetes   15 M
Installing dependencies:
conntrack-tools  x86_64          1.4.6-2.amzn2023.0.2   amazonlinux  208 k
cri-tools         x86_64          1.31.1-150500.1.1   kubernetes   6.9 M
kubernetes-cni   x86_64          1.5.1-150500.1.1   kubernetes   7.1 M
libnetfilter_cthelper x86_64        1.0.0-21.amzn2023.0.2   amazonlinux  24 k
libnetfilter_cttimeout x86_64        1.0.0-19.amzn2023.0.2   amazonlinux  24 k
libnetfilter_queue x86_64        1.0.5-2.amzn2023.0.2   amazonlinux  30 k

Transaction Summary

Install 9 Packages

Total download size: 51 M
Installed size: 269 M
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm 378 kB/s | 24 kB 00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 360 kB/s | 24 kB 00:00
(3/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm 2.7 MB/s | 208 kB 00:00

```

i-030d25988270b63ec (Master)

PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 4/9
Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 5/9
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubectl-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 9/9
Verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying   : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying   : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying   : kubectl-1.31.1-150500.1.1.x86_64 7/9
Verifying   : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64      kubeadm-1.31.1-150500.1.1.x86_64
kubectl-1.31.1-150500.1.1.x86_64                 kubelet-1.31.1-150500.1.1.x86_64      kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-44-214 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-44-214 ec2-user]#

```

i-030d25988270b63ec (Master)

PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

Step 4: Now, run the following command in the master instance -
kubeadm init

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-81-4 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0918 14:26:24.654814 28225 checks.go:946] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificatebin folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] Generating "apiserver serving cert is signed for DNS names [ip-172-31-81-4.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.81.4]"
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-81-4.ec2.internal localhost] and IPs [172.31.81.4 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-81-4.ec2.internal localhost] and IPs [172.31.81.4 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file

```

i-0d0503cebb680a65b (Master)

PublicIPs: 34.239.132.160 PrivateIPs: 172.31.81.4

Step 5: Now, run the following commands in master instance's console –

- a.

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

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

```
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash
sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
```

The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output details the configuration of a bootstrap token, setting up RBAC rules, and updating kubelet configuration. It concludes with a message that the Kubernetes control-plane has initialized successfully and provides instructions for regular users to run `mkcert` commands.

```
[bootstrapping] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrapping] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrapping] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrapping] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrapping] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrapping] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

Step 6: Run this command in node1 and node2 -

```
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
```

The screenshot shows the terminal output of the `kubeadm join` command on a worker node. It lists the packages being installed and verified, including `conntrack-tools`, `kubelet`, `cri-tools`, and `kubernetes-cni`. The command also creates a `kubelet` service and enables it. A warning message about missing `socat` and `tc` files is shown, along with an error message about a context deadline exceeded during the execution phase.

```
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubectl-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 9/9
Verifying  : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying  : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying  : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying  : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying  : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying  : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying  : kubectl-1.31.1-150500.1.1.x86_64 7/9
Verifying  : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying  : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Complete!
[root@ip-172-31-95-221 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-95-221 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-95-221 ec2-user]#
```

```

Installing      : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          6/9
Installing      : kubelet-1.31.1-150500.1.1.x86_64                  7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64                  7/9
Installing      : kubeadm-1.31.1-150500.1.1.x86_64                  7/9
Installing      : kubectl-1.31.1-150500.1.1.x86_64                  8/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64                  9/9
Verifying       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          9/9
Verifying       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64          1/9
Verifying       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64          2/9
Verifying       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64          3/9
Verifying       : cri-tools-1.31.1-150500.1.1.x86_64          4/9
Verifying       : kubeadm-1.31.1-150500.1.1.x86_64          5/9
Verifying       : kubectl-1.31.1-150500.1.1.x86_64          6/9
Verifying       : kubelet-1.31.1-150500.1.1.x86_64          7/9
Verifying       : kubernetes-cni-1.5.1-150500.1.1.x86_64          8/9
Verifying       : kubelet-1.31.1-150500.1.1.x86_64          9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64          kubeadm-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64                  kubelet-1.31.1-150500.1.1.x86_64          kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64    libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-94-95 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-94-95 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:ibbcc9939e095e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
preflight] Running pre-flight checks
[WARNING FileExistsSocat]: socat not found in system path
[WARNING FileExistsTc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info configMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-94-95 ec2-user]#

```

Step 7: Run the following command in master instance console -

kubectl get nodes



```

[root@ip-172-31-01-4 ec2-user]# kubectl get nodes
NAME           STATUS   ROLES     AGE   VERSION
ip-172-31-81-4.ec2.internal   NotReady   control-plane   26m   v1.31.1
[root@ip-172-31-01-4 ec2-user]# kubectl get nodes
NAME           STATUS   ROLES     AGE   VERSION
ip-172-31-81-4.ec2.internal   NotReady   control-plane   26m   v1.31.1
ip-172-31-94-95.ec2.internal   NotReady   <none>    17s   v1.31.1
ip-172-31-95-221.ec2.internal NotReady   <none>    13s   v1.31.1
[root@ip-172-31-01-4 ec2-user]#

```

Advanced DevOps Experiment 4

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

Step 1: Go to AWS Academy in services select EC2 and create 3 instance with and name them as master, node1, node2 and remember to select instance type as t2.medium.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Node1	i-04cd1f7f6e6a0b61a8	Terminated	t2.micro	-	View alarms	us-east-1c	-
Master	i-0d0503cebb680a65b	Running	t2.medium	Initializing	View alarms	us-east-1a	ec2-34-23-
Node1	i-030b5f6226fb1efdf	Running	t2.medium	Initializing	View alarms	us-east-1a	ec2-3-86-2
Node2	i-02366d960a4501365	Running	t2.medium	Initializing	View alarms	us-east-1a	ec2-3-83-8

EC2 Instance Connect Session Manager SSH client EC2 serial console

Port 22 (SSH) is not authorized
Port 22 (SSH) is currently not authorized by your security group. To use EC2 Instance Connect, you must authorize port 22 for the EC2 Instance Connect service IP addresses in your Region: 18.206.107.24/29.
[Learn more.](#)

Instance ID: [i-0365889e6b3d22cdb](#) (Mynewapp-env)

Connection Type:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address: [44.205.155.152](#)

Username: Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

Step 2: Create a new key pair and name it as myKey1 and download as .pem file. Then, open command prompt and go to the directory where the key is downloaded and run the following command

chmod 400 myKey1.pem

ssh -i myKey1.pem [ec2-user@3.88.13.120](#)

Repeat the steps for node1, master and node2

```
ec2-user@ip-172-31-30-94:~
```

```
user@DESKTOP-Q0GK15A MINGW64 ~ (master)
$ cd Downloads
user@DESKTOP-Q0GK15A MINGW64 ~/Downloads (master)
$ chmod 400 myKey1.pem
```

```
user@DESKTOP-Q0GK15A MINGW64 ~/Downloads (master)
$ ssh -i myKey1.pem ec2-user@3.88.13.120
The authenticity of host '3.88.13.120 (3.88.13.120)' can't be established.
ED25519 key fingerprint is SHA256:f1SJDOrz561fHUKw1T49IGXDY76iHNT5isSDB2NWyXQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added '3.88.13.120' (ED25519) to the list of known hosts.
```

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
```

```
Last Login: wed Sep 18 17:26:06 2024 From 18.206.107.29
[ec2-user@ip-172-31-30-94 ~]$
```

Step 3: Then, Select and connect each instance and run the following commands inside the console of each instance.

- sudo su
- yum install docker -y
- systemctl start docker
- docker --version
- yum repolist

```
aws Services Search [Alt+S] N. Virginia v vclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[ec2-user@ip-172-31-44-214 ~]$ sudo su
[root@ip-172-31-44-214 ec2-user]# yum install docker -y
Last metadata expiration check: 0:14:58 ago on Wed Sep 18 13:11:57 2024.
Dependencies resolved.
=====
Package           Architecture   Version        Repository      Size
=====
Installing:
  docker          x86_64         25.0.6-1.amzn2023.0.2
Installing dependencies:
  containerd      x86_64         1.7.20-1.amzn2023.0.1
  iptables-libc
  iptables-nft
  libcgroup
  libnetfilter_conntrack
  libnftn
  pigz
  runc
=====
Transaction Summary
=====
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libc-1.8.8-3.amzn2023.0.2.x86_64.rpm
3.0 MB/s | 401 kB     00:00
i-030d25988270b63ec (Master)
PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214
```

```
aws Services Search [Alt+S] N. Virginia v vclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64
Verifying : iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
Verifying : libnftn-1.0.1-19.amzn2023.0.2.x86_64
Verifying : libnftn-1.2.2-2.amzn2023.0.2.x86_64
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64
=====
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64          docker-25.0.6-1.amzn2023.0.2.x86_64
  iptables-libc-1.8.8-3.amzn2023.0.2.x86_64        libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnftn-1.2.2-2.amzn2023.0.2.x86_64             libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64               pigz-2.5-1.amzn2023.0.3.x86_64
=====
Complete!
[root@ip-172-31-44-214 ec2-user]# systemctl start docker
[root@ip-172-31-44-214 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-44-214 ec2-user]# █
i-030d25988270b63ec (Master)
PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214
```

Step 4: Now, visit the following link <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/> and scroll down till you find Red-Hat and then select Red-Hat based distributions tab copy all the commands one by one in each console of instance.

The screenshot shows the Kubernetes documentation website. The top navigation bar includes links for Documentation, Kubernetes Blog, Training, Partners, Community, Case Studies, Versions, English, and a search bar. On the left, a sidebar menu lists various documentation categories, including 'Getting started' and 'Installing Kubernetes with deployment tools' which is currently selected. The main content area is titled 'Without a package manager' under the 'Red Hat-based distributions' tab. It contains instructions for setting SELinux to permissive mode and adding the Kubernetes yum repository. A 'Caution' section provides notes about SELinux configuration. To the right, there are edit and creation options for the page, and a sidebar with links for 'Before you begin' and other documentation sections.

The screenshot shows an AWS CloudShell terminal window. The terminal prompt is 'root@ip-172-31-44-214 ec2-user#'. The user has run several commands to prepare the system for Kubernetes:

```
complete!
[root@ip-172-31-44-214 ec2-user]# systemctl start docker
[root@ip-172-31-44-214 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-44-214 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                            Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-44-214 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-44-214 ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repo.gpg
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repo.gpg
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-44-214 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

At the bottom of the terminal, it shows the session details: 'i-030d25988270b63ec (Master)', 'Public IPs: 54.152.128.140', and 'Private IPs: 172.31.44.214'.

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-44-214 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.

=====
Package           Architecture      Version          Repository      Size
=====
Installing:
kubeadm          x86_64          1.31.1-150500.1.1   kubernetes      11 M
kubectl          x86_64          1.31.1-150500.1.1   kubernetes      11 M
kubelet          x86_64          1.31.1-150500.1.1   kubernetes      15 M
Installing dependencies:
conntrack-tools  x86_64          1.4.6-2.amzn2023.0.2 amazonlinux    208 k
cri-tools         x86_64          1.31.1-150500.1.1   kubernetes      6.9 M
kubernetes-cni   x86_64          1.5.1-150500.1.1   kubernetes      7.1 M
libnetfilter_cthelper x86_64        1.0.0-21.amzn2023.0.2 amazonlinux    24 k
libnetfilter_cttimeout x86_64        1.0.0-19.amzn2023.0.2 amazonlinux    24 k
libnetfilter_queue x86_64        1.0.5-2.amzn2023.0.2 amazonlinux    30 k

Transaction Summary
Install 9 Packages

Total download size: 51 M
Installed size: 269 M
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm 378 kB/s | 24 kB 00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm 360 kB/s | 24 kB 00:00
(3/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm 2.7 MB/s | 208 kB 00:00

```

i-030d25988270b63ec (Master)

PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 4/9
Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 5/9
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubectl-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 9/9
Verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying   : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying   : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying   : kubectl-1.31.1-150500.1.1.x86_64 7/9
Verifying   : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64
kubectl-1.31.1-150500.1.1.x86_64      kubelet-1.31.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
                                                 kubernetes-cni-1.5.1-150500.1.1.x86_64
                                                 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-44-214 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-44-214 ec2-user]#

```

i-030d25988270b63ec (Master)

PublicIPs: 54.152.128.140 PrivateIPs: 172.31.44.214

Step 5: Now, run the following command in the master instance -
kubeadm init

```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-81-4 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0918 14:26:24.654814 28225 checks.go:946] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificatebin folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] Generating "apiserver serving cert" is signed for DNS names [ip-172-31-81-4.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.81.4]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-81-4.ec2.internal localhost] and IPs [172.31.81.4 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-81-4.ec2.internal localhost] and IPs [172.31.81.4 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file

```

i-0d0503cebb680a65b (Master)

PublicIPs: 34.239.132.160 PrivateIPs: 172.31.81.4

Step 6: Now, run the following commands in master instance's console –

- a.

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

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

```
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash
sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
```

The screenshot shows a CloudWatch Log Stream with the following content:

```
[bootstrapping] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrapping] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrapping] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrapping] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrapping] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrapping] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

Step 7: Run this command in node1 and node2 -

```
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
```

The screenshot shows a CloudWatch Log Stream with the following content:

```
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubectl-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 9/9
Verifying : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying : kubectl-1.31.1-150500.1.1.x86_64 7/9
Verifying : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
 conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64          kubeadm-1.31.1-150500.1.1.x86_64
 kubelet-1.31.1-150500.1.1.x86_64                   kubelet-1.31.1-150500.1.1.x86_64          kubernetes-cni-1.5.1-150500.1.1.x86_64
 libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-95-221 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-95-221 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-95-221 ec2-user]# ]
```

```

Installing      : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Installing      : kubelet-1.31.1-150500.1.1.x86_64
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64
Installing      : kubeadm-1.31.1-150500.1.1.x86_64
Installing      : kubectl-1.31.1-150500.1.1.x86_64
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64
Verifying       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Verifying       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
Verifying       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
Verifying       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
Verifying       : cri-tools-1.31.1-150500.1.1.x86_64
Verifying       : kubeadm-1.31.1-150500.1.1.x86_64
Verifying       : kubectl-1.31.1-150500.1.1.x86_64
Verifying       : kubelet-1.31.1-150500.1.1.x86_64
Verifying       : kubernetes-cni-1.5.1-150500.1.1.x86_64

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64                    kubeadm-1.31.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-94-95 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-94-95 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:ibbcc9939e095e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3018
preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info configMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-94-95 ec2-user]#

```

Step 8: Run the following command in master instance console -

kubectl get nodes

```

[root@ip-172-31-01-4 ec2-user]# kubectl get nodes
NAME           STATUS    ROLES     AGE   VERSION
ip-172-31-81-4.ec2.internal   NotReady   control-plane   26m   v1.31.1
[root@ip-172-31-01-4 ec2-user]# kubectl get nodes
NAME           STATUS    ROLES     AGE   VERSION
ip-172-31-81-4.ec2.internal   NotReady   control-plane   26m   v1.31.1
ip-172-31-94-95.ec2.internal   NotReady   <none>    17s   v1.31.1
ip-172-31-95-221.ec2.internal   NotReady   <none>    13s   v1.31.1
[root@ip-172-31-01-4 ec2-user]#

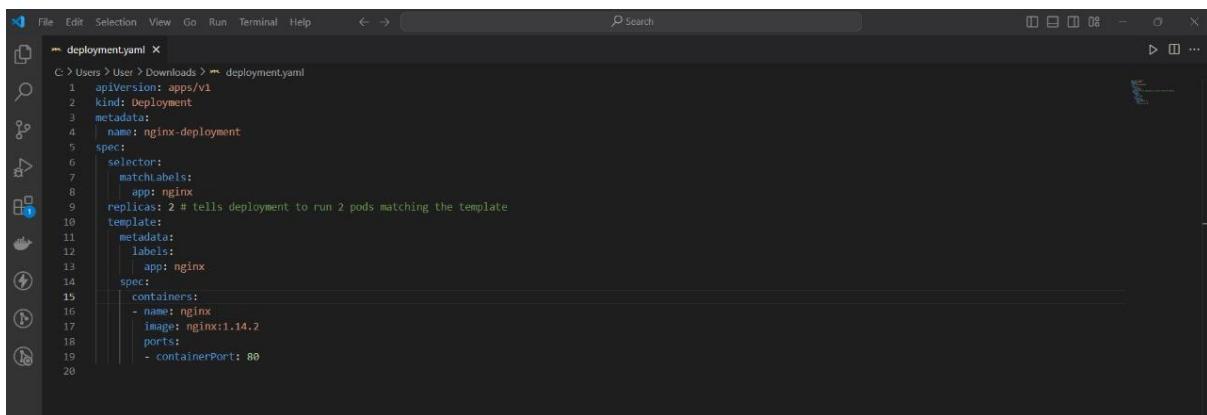
```

Step 9: Once the cluster is set up and running, deploy an Nginx application:

kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

Forward the Nginx service to your localhost so that you can access it using the following command

kubectl port-forward deployment/nginx-deployment 8080:80



Step 10: In a new terminal of Git Bash, run the following command:

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

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

The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
RSA key fingerprint is SHA256:TJ3309vz5B3HnG4t4Qqdy78lnETL50bZwngQ.
This key is not known by the user.
Are you sure you want to continue connecting (yes/no)?指纹识别失败
警告：永久性地将“127.0.0.1”（RSA指纹）添加到已知主机列表。
Last Update: Wed Sep 18 17:24:06 2024 from 18.206.187.29

```
curl: (60) SSL certificate problem: unable to get local issuer certificate  
更多详细信息请参阅: https://curl.haxx.se/ca/cacert.pem
```

```
[nc2-user@ip-172-31-80-94 ~]
```

Step 11: The website is deployed.



Welcome to nginx!

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

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

EXPERIMENT 5:TERRAFORM

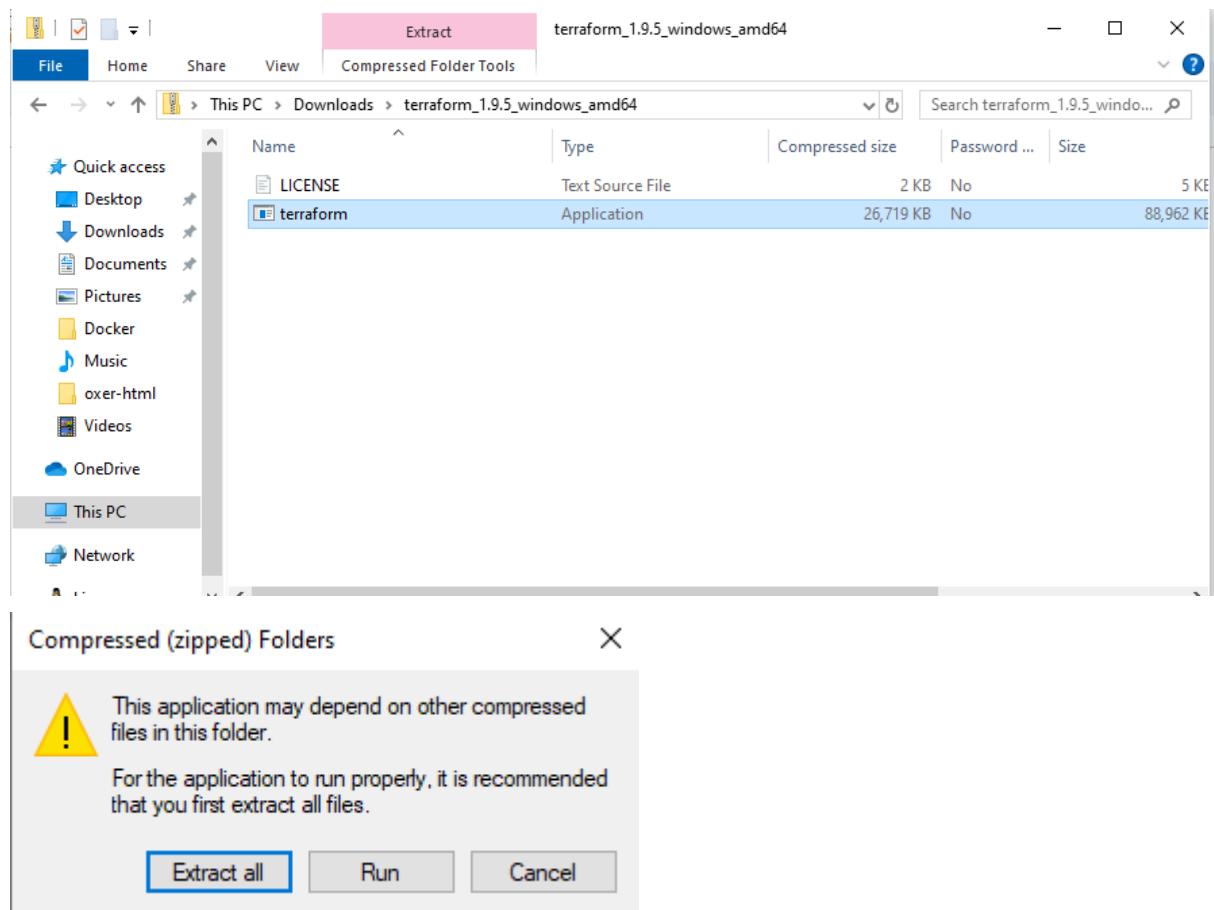
AIM: To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine and Windows.

Installation and Configuration of Terraform in Windows:

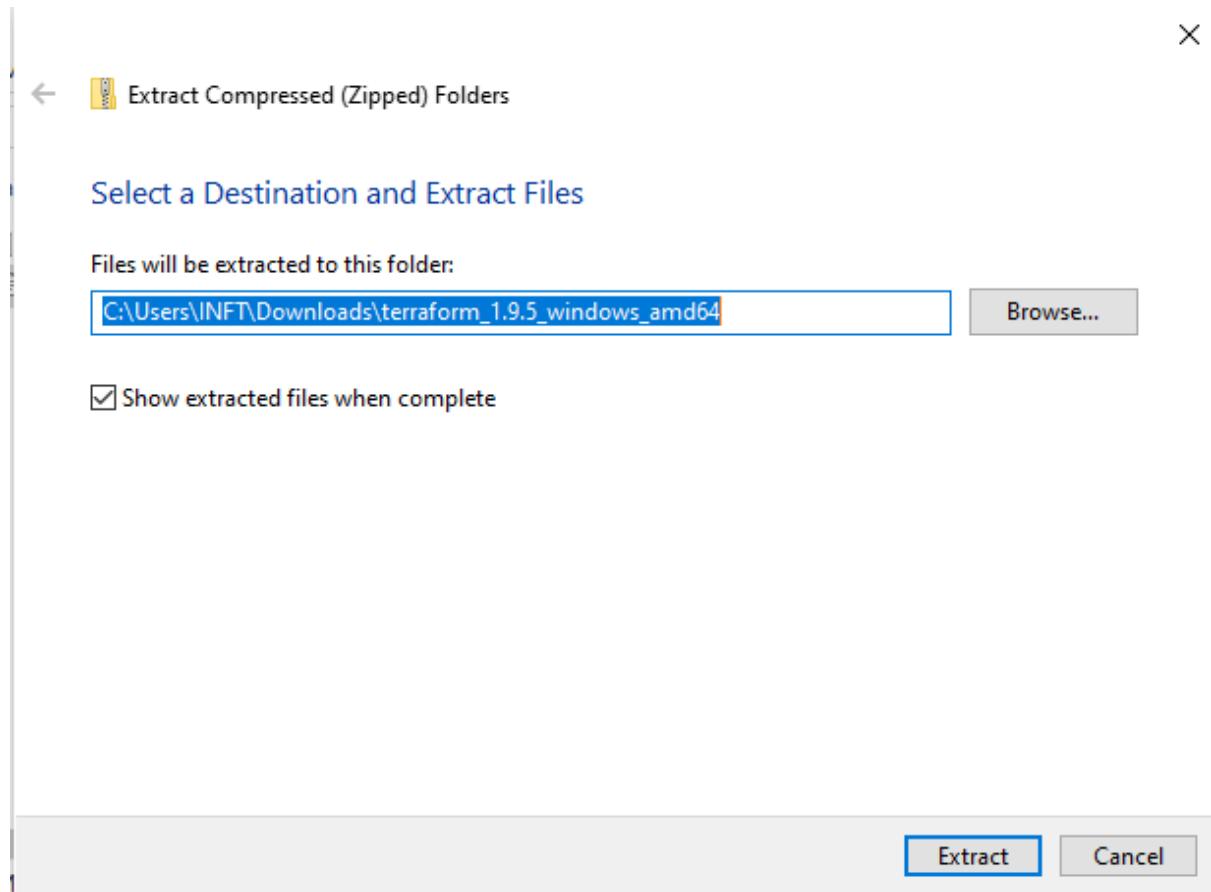
1: To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website: :<https://www.terraform.io/downloads.html>

The screenshot shows two windows side-by-side. The top window is a web browser displaying the Terraform official website at <https://developer.hashicorp.com/terraform/install>. The 'Windows' section is highlighted, showing binary download options for 386 and AMD64 architectures, both version 1.9.5. The bottom window is a file explorer showing the 'Downloads' folder. It lists four files: 'terraform_1.9.5_windows_amd64' (Compressed (zipped), 26,721 KB), 'AIDS_Exp5 (1)' (WPS PDF Document, 305 KB), 'AIDS_Exp5' (WPS PDF Document, 304 KB), and another 'AIDS_Exp5' file (OpenDocument Text, 339 KB). The file names correspond to the ones shown in the browser's terminal window.

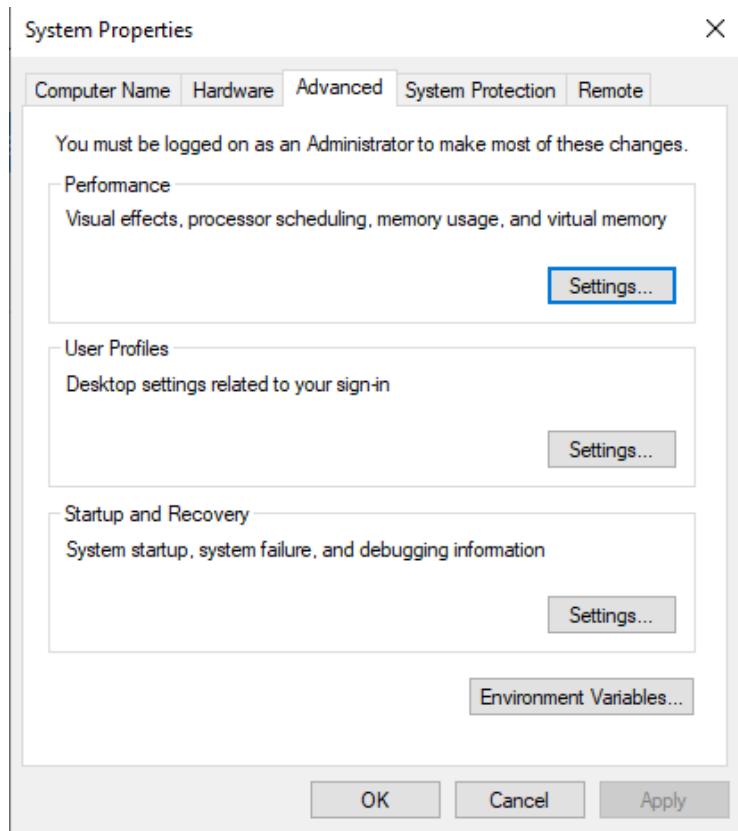
2: Compress the zip file.



2: Extract the downloaded setup file Terraform.exe.



4: Set the System path for Terraform in 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

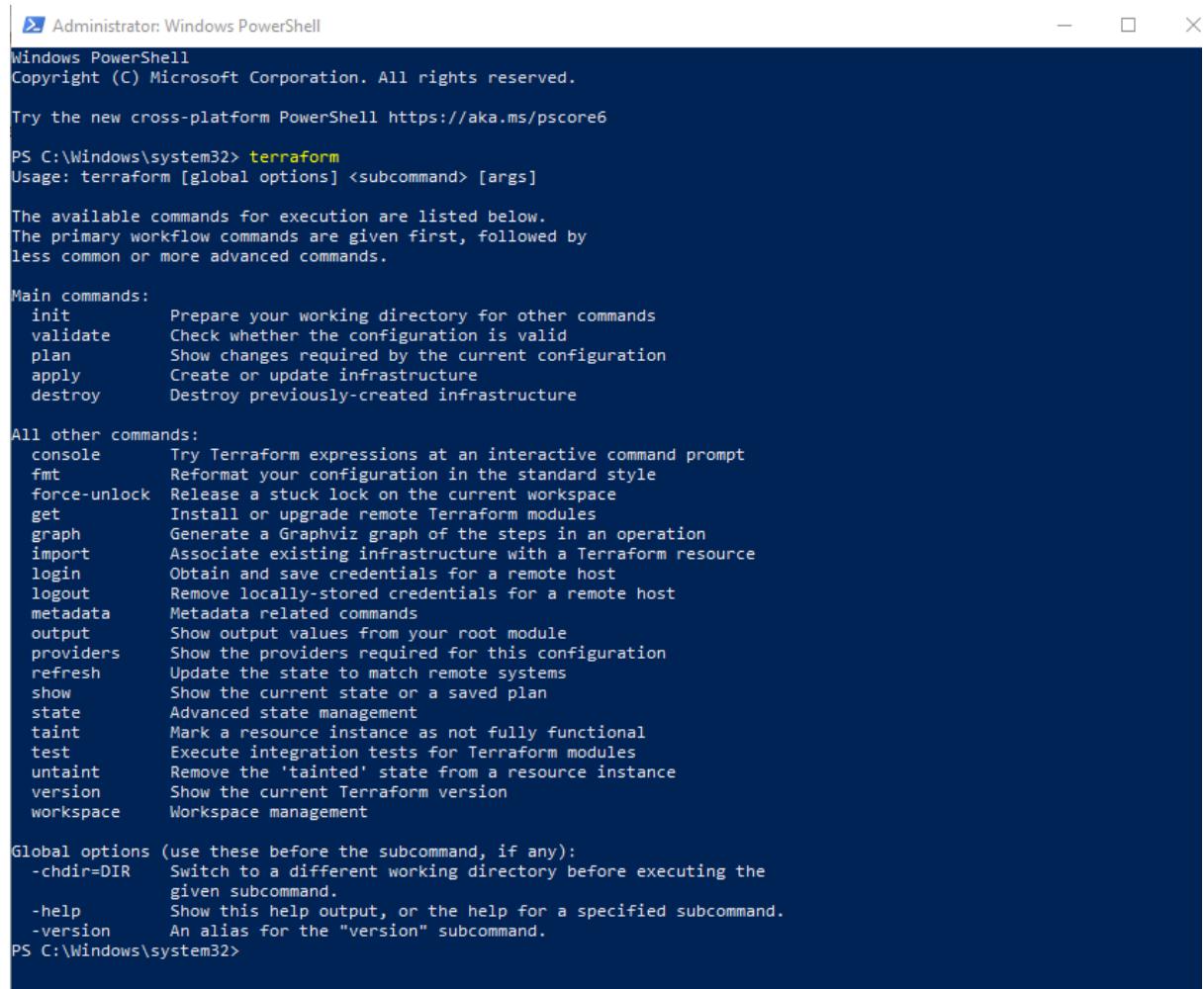
OK Cancel

Edit User Variable

Variable name:	Path
Variable value:	C:\terraform
Browse Directory...	Browse File...

OK Cancel

5 : Open Terraform in PowerShell and check its functionality



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command "terraform" was run, displaying the following help output:

```
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

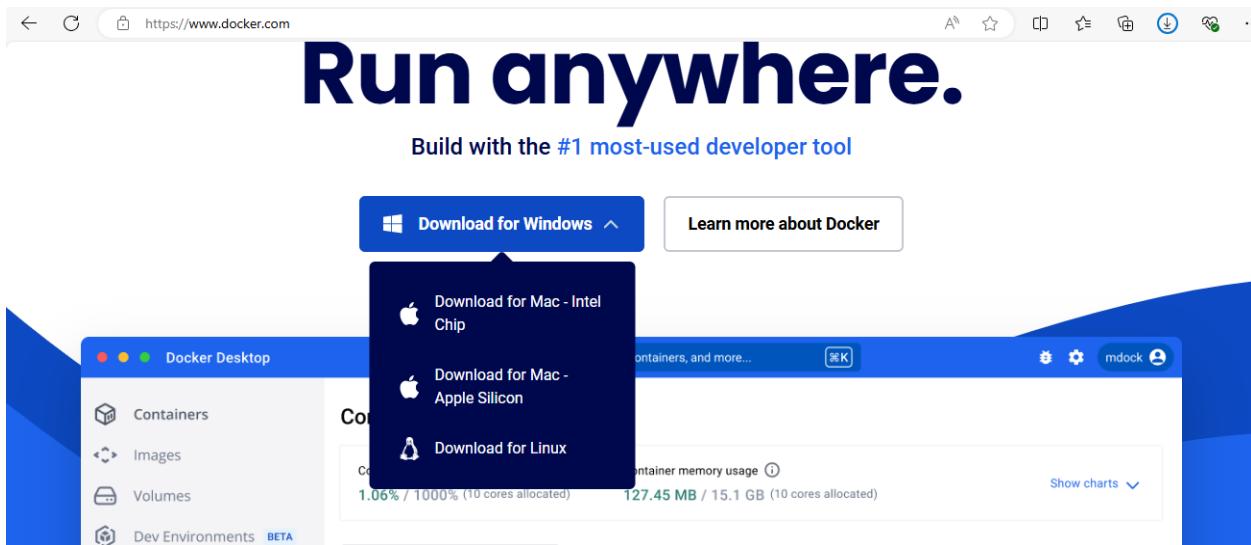
Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
             given subcommand.
  -help       Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.

PS C:\Windows\system32>
```

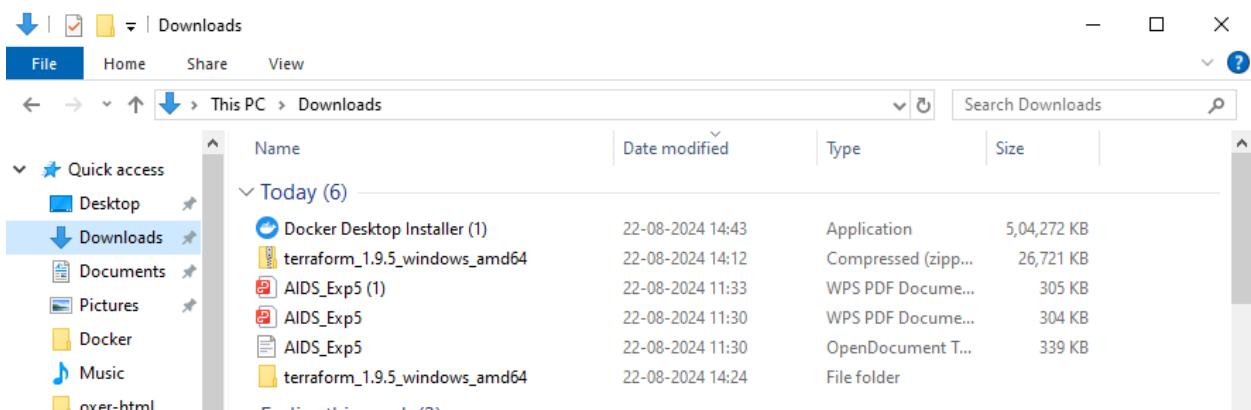
Experiment 6

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

Step 1: Download Docker from www.docker.com



Step 2: Now, Docker is successfully downloaded.





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
```



Installing Docker Desktop 4.33.1 (161083)

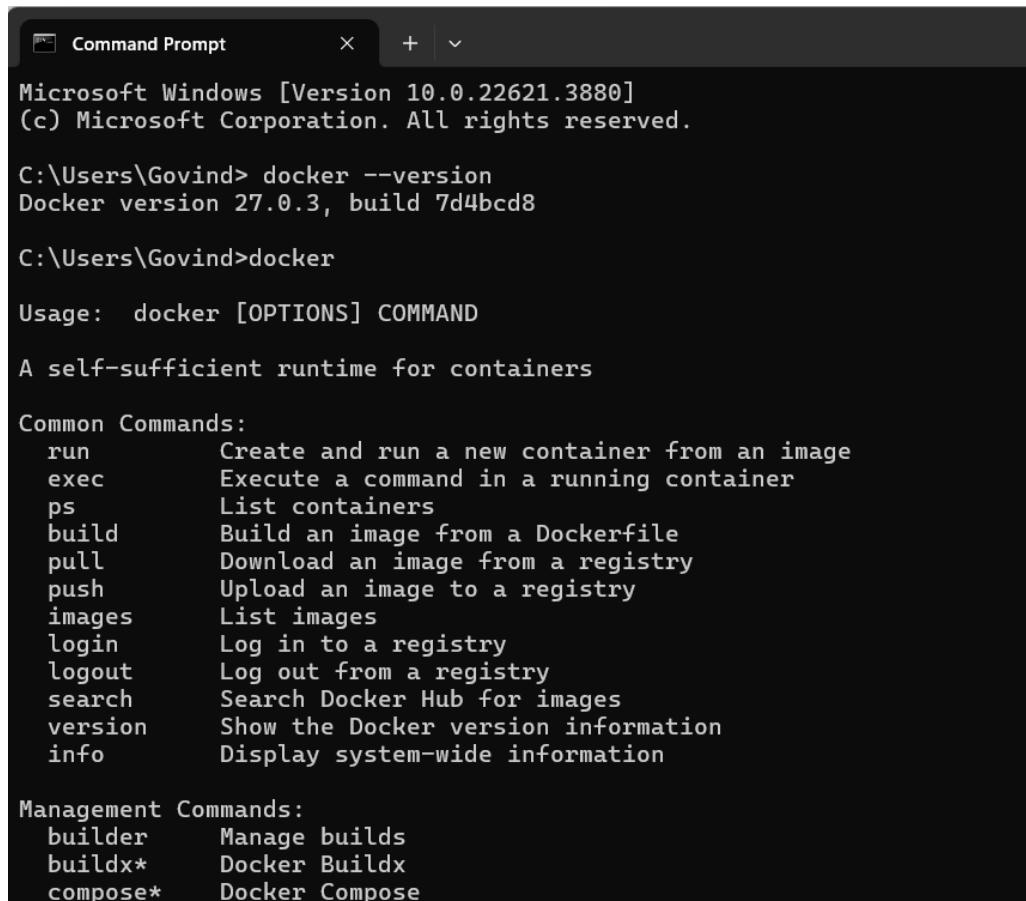


Docker Desktop 4.33.1

Installation succeeded

Close

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



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
Microsoft Windows [Version 10.0.22621.3880]
(c) Microsoft Corporation. All rights reserved.

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

C:\Users\Govind>docker

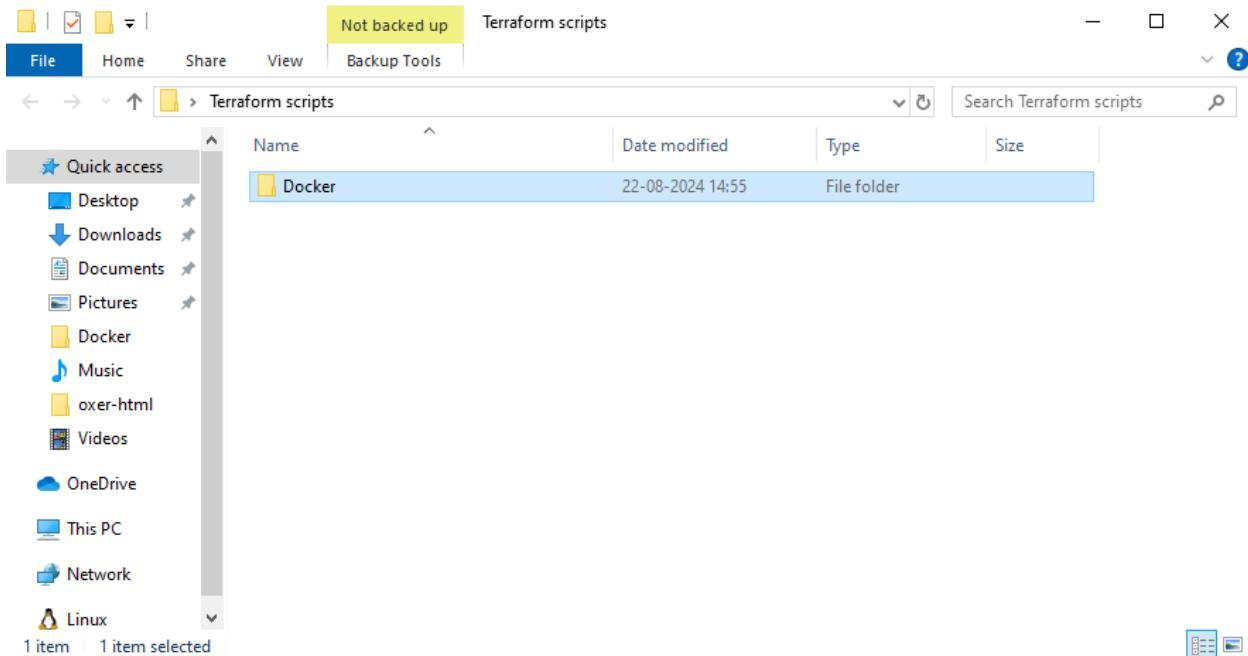
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

Management Commands:
  builder  Manage builds
  buildx*  Docker Buildx
  compose* Docker Compose
```

Step 4: Create a folder Terraform_scripts and inside it create a folder named Docker.



Step 5: create a new folder named 'Terraform' in the 'TerraformScripts' folder. Then create a new terraform_script.tf file using vs code.

Run the following script in the VS Code.

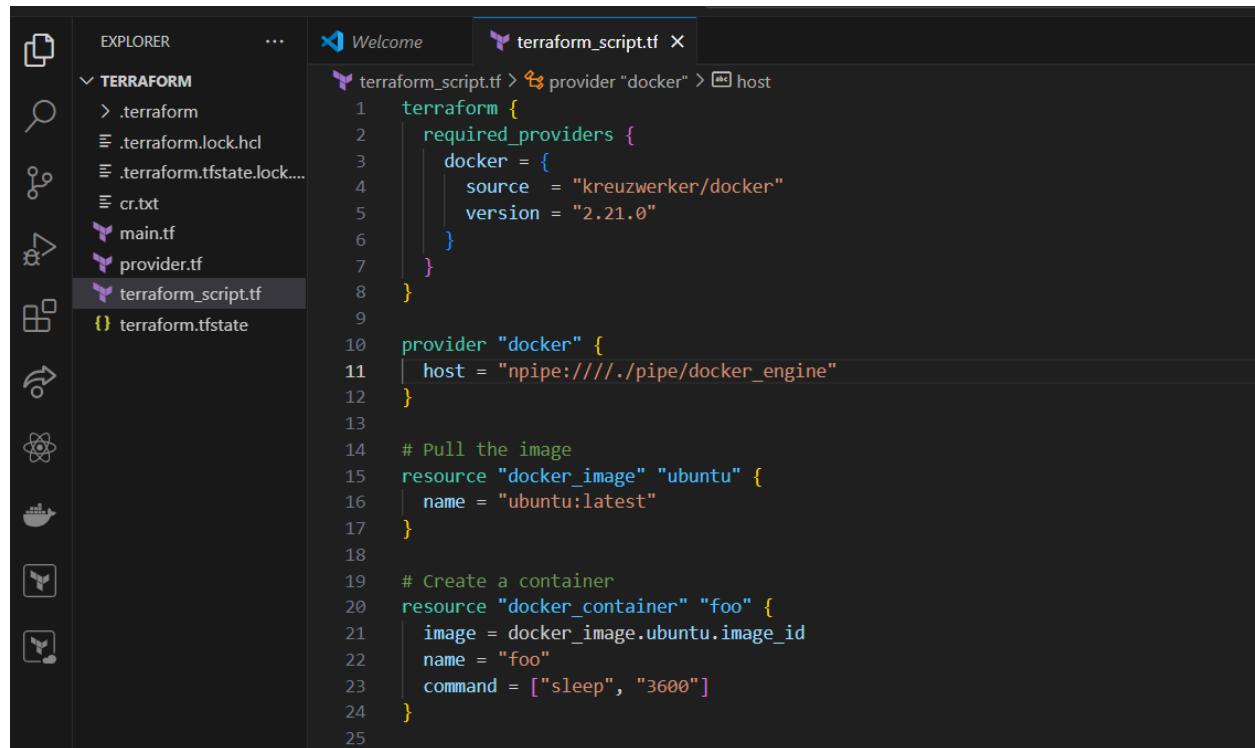
```
terraform {  
    required_providers {  
        docker = {  
            source = "kreuzwerker/docker"  
            version = "2.21.0"  
        }  
    }  
}  
  
provider "docker" {  
    host = "npipe:///./pipe/docker_engine"  
}  
  
# Pull the image
```

```

resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

# Create a container
resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name = "foo"
  command = ["sleep", "3600"]
}

```



```

terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

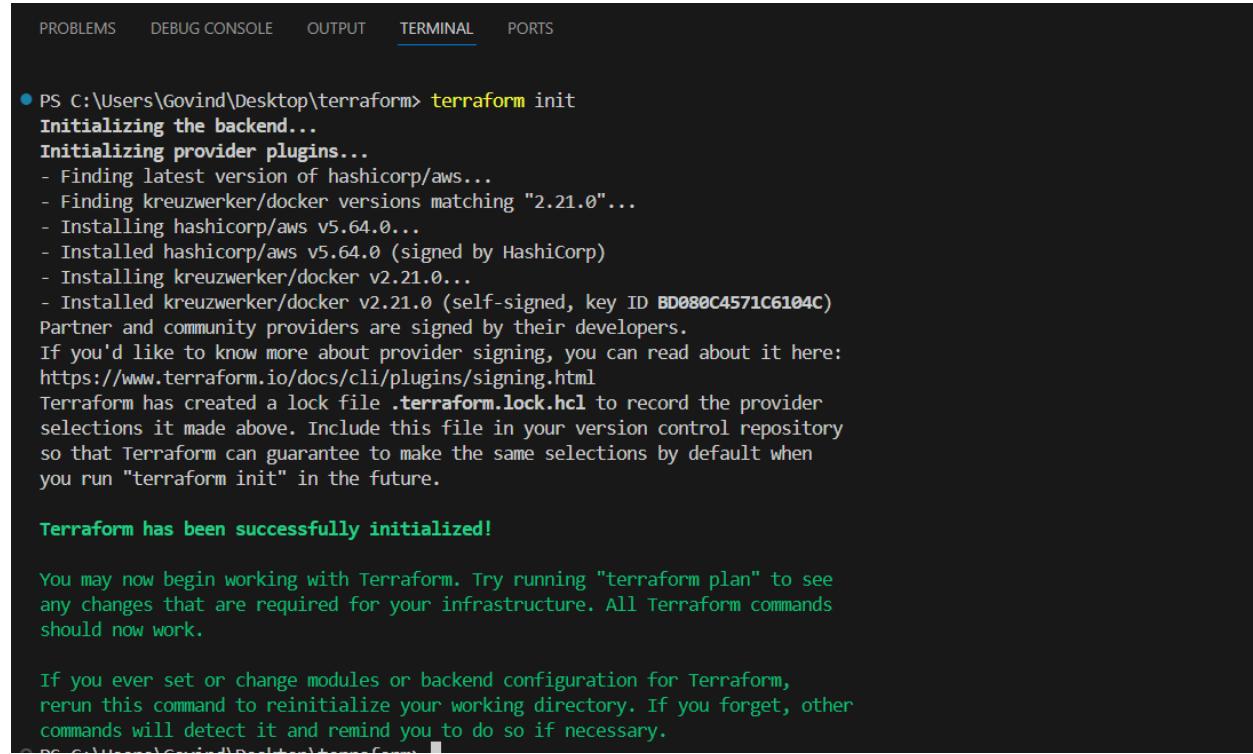
provider "docker" {
  host = "npipe:///pipe/docker_engine"
}

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

# Create a container
resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name = "foo"
  command = ["sleep", "3600"]
}

```

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



The screenshot shows a terminal window with the following content:

```
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

● PS C:\Users\Govind\Desktop\terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing hashicorp/aws v5.64.0...
- Installed hashicorp/aws v5.64.0 (signed by HashiCorp)
- 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\Govind\Desktop\terraform> terraform plan

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

Terraform will perform the following actions:

```
# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "3600",
    ]
    + container_logs  = (known after apply)
    + entrypoint      = (known after apply)
    + env              = (known after apply)
    + exit_code        = (known after apply)
    + gateway          = (known after apply)
    + hostname         = (known after apply)
    + id               = (known after apply)
    + image             = (known after apply)
    + init              = (known after apply)
    + ip_address       = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode         = (known after apply)
    + log_driver        = (known after apply)
    + 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)
```

Ln 25, Col 1 (439 selected) Spaces: 2 UTF-8 CRLF

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\Govind\Desktop\terraform> terraform apply
```

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

Terraform will perform the following actions:

```
# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "3600",
    ]
    + container_logs  = (known after apply)
    + entrypoint      = (known after apply)
    + env              = (known after apply)
    + exit_code        = (known after apply)
    + gateway          = (known after apply)
    + hostname         = (known after apply)
    + id               = (known after apply)
    + image             = (known after apply)
    + init              = (known after apply)
    + ip_address       = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode         = (known after apply)
    + log_driver        = (known after apply)
    + logs              = false
    + 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)
```

Ln 25, Col 1 (439 selected) Spaces: 2 UTF-8 CRLF {

```

+ 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: Still creating... [20s elapsed]
docker_image.ubuntu: Creation complete after 29s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=2f7e8bcraf7e5f75f04f53be0aa80e74a915285dddb826402ebfc7f569e571ebd]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
● PS C:\Users\Govind\Desktop\terraform> terraform providers
Providers required by configuration:
└── provider[registry.terraform.io/kreuzwerker/docker] 2.21.0
    └── provider[registry.terraform.io/hashicorp/aws]

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

● PS C:\Users\Govind\Desktop\terraform> terraform validate
Success! The configuration is valid.

● PS C:\Users\Govind\Desktop\terraform> terraform state list
  docker_container.foo
  docker_image.ubuntu

● PS C:\Users\Govind\Desktop\terraform> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   edbf74c41f8  3 weeks ago  78.1MB
nginx           latest   448ae8f1d2f9  15 months ago 142MB
nginx/docker-extension 0.0.3   41d3d0d7d940  16 months ago 7.53MB

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	edbf74c41f8	3 weeks ago	78.1MB
nginx	latest	448ae8f1d2f9	15 months ago	142MB
nginx/docker-extension	0.0.3	41d3d0d7d940	16 months ago	7.53MB

```
PS C:\Users\Govind\Desktop\terraform> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbf7e4c41f8a3501ce542e137cf28ea04dd03e6df8c9d6519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=2f7e8bcf7e5f75f04f53be0aa80e74a915285ddb826402ebfc7f569e571ebd]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
  - attach          = false -> null
  - command        = [
    - "sleep",
    - "3600",
  ] -> null
  - cpu_shares     = 0 -> null
  - dns             = [] -> null
  - dns_opts        = [] -> null
  - dns_search      = [] -> null
  - entrypoint      = [] -> null
  - env             = [] -> null
  - gateway         = "172.17.0.1" -> null
  - group_add       = [] -> null
  - hostname        = "2f7e8bcf7e5" -> null
  - id              = "2f7e8bcf7e5f75f04f53be0aa80e74a915285ddb826402ebfc7f569e571ebd" -> null
  - image           = "sha256:edbf7e4c41f8a3501ce542e137cf28ea04dd03e6df8c9d6519b6ad761c2598a" -> null
  - init            = false -> null
  - ip_address      = "172.17.0.2" -> null
  - ip_prefix_length = 16 -> null
  - ipc_mode        = "private" -> null
  - links           = [] -> null
  - log_driver       = "json-file" -> null
  - log_opts         = {} -> null
  - logs             = false -> null
  - max_retry_count = 0 -> null
  - memory           = 0 -> null
  - memory_swap      = 0 -> null
  - must_run         = true -> null
  - name             = "foo" -> null
  - network_data     = f
```

Experiment No 7

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

Step 1: Make sure you download all the Prerequisites required :

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

And then run the following command on your window powershell/cmd prompt only once:

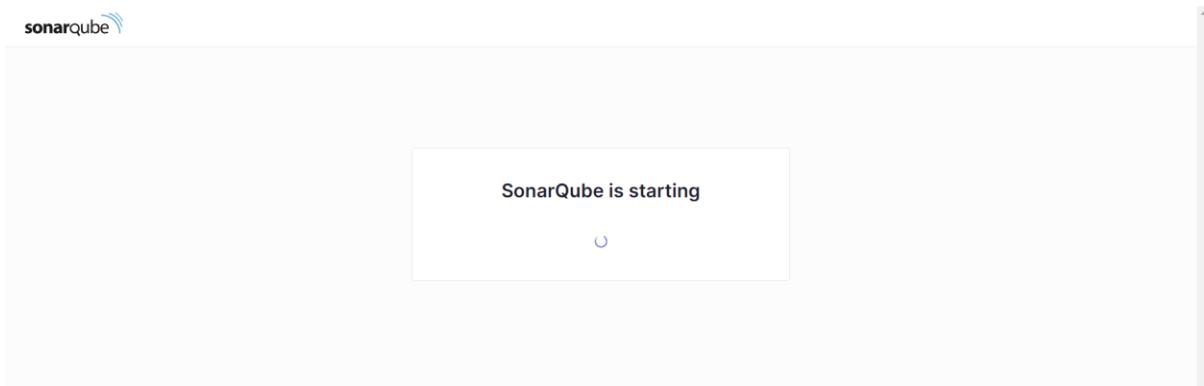
```
$ docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 Sonarqube:latest
```

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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Govind> 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
f7b8d5f7157e7cb2643b22c497a78368667800258325d55d236330b6b6286f85
```

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



Step 3: Login to SonarQube using username admin and password admin then create a local project in SonarQube with the name sonarqube-test.

1 of 2

Create a local project

Project display name *

sonarqube-test



Project key *

sonarqube-test



Main branch name *

main

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

[Cancel](#)

[Next](#)

Step 4: Setup the project to use the global setting and open Jenkins Dashboard.

2 of 2

x

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.

Step 5: Now, Go to Dashboard ->Manage Jenkins -> Plugin Manager and search for SonarQube Scanner under Available plugins for Jenkins and install it.

The screenshot shows the Jenkins Plugin Manager interface. A search bar at the top contains the text "sona". Below the search bar, a table lists available plugins. One plugin is highlighted: "SonarQube Scanner for Jenkins 2.17.2". The plugin details show: "Name" (SonarQube Scanner for Jenkins), "Version" (2.17.2), "Description" (This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.), "Status" (Enabled), and a "Check for updates" toggle switch which is turned on (indicated by a blue circle with a white checkmark). The table also includes columns for "Name", "Status", and "Check for updates".

Plugins

Updates

28

Available plugins

Installed plugins

Advanced settings

Download progress

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

Step 6: Under Jenkins 'Configure System', look for SonarQube Servers and enter the details then Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

The screenshot shows the Jenkins 'Global Tool Configuration' page for 'SonarQube Scanner installations'. A new configuration is being added with the name 'sonarqube scanner'. The 'Install automatically' checkbox is checked. Under 'Install from Maven Central', the version 'SonarQube Scanner 6.2.0.4584' is selected. There is also an 'Add Installer' dropdown. At the bottom are 'Save' and 'Apply' buttons.

Step 7: Create a New Item in Jenkins and choose a freestyle project.

The screenshot shows the 'Create New Item' dialog with 'Enter an item name' set to 'SonarQube'. Below it, a list of project types is shown:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

At the bottom is an 'OK' button.

Step 7: Choose this GitHub repository in Source Code Management.

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

Under Build ->Execute SonarQube Scanner, enter these Analysis

properties. Mention the SonarQube Project Key, Login, Password, and Host URL.

```
sonar.projectKey=sonarqube-test  
sonar.login=admin  
sonar.password=Sneha@100  
sonar.hosturl=http://localhost:9000/
```

The screenshot shows the 'Source Code Management' configuration page. Under the 'Configure' tab, the 'General' section is selected. In the 'Source Code Management' section, the 'Git' option is chosen. The 'Repositories' section contains a single repository entry with the 'Repository URL' set to `https://github.com/shazfiorit/MSBuild_firstproject.git`. The 'Branches to build' section is collapsed. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the 'Build Steps' configuration page. Under the 'Configure' tab, the 'General' section is selected. In the 'Build Steps' section, there is one step named 'Execute SonarQube Scanner'. The 'JDK' dropdown is set to '(Inherit From Job)'. The 'Path to project properties' field contains the following properties:
`sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=Sneha@100
sonar.hosturl=http://sonarqube:9000`

Step 8: Go to `http://localhost:9000/` and enter your previously created username and password then Go to Permissions and grant the Admin user Execute Permissions.

Global Permissions

Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

	Administer System	Execute Analysis	Create
A Administrator admin	<input type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/> Projects

Step 9: Now Build and Run.

Jenkins

SonarQube

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

Build History

#2 Sep 25, 2024, 6:35PM

#1 Sep 25, 2024, 6:15PM

Permalinks

- Last build (#2), 13 min ago
- Last stable build (#2), 13 min ago
- Last successful build (#2), 13 min ago
- Last failed build (#1), 32 min ago
- Last unsuccessful build (#1), 32 min ago
- Last completed build (#2), 13 min ago

Jenkins

SonarQube > #2

Status

#2 (Sep 25, 2024, 6:35:00 PM)

No changes.

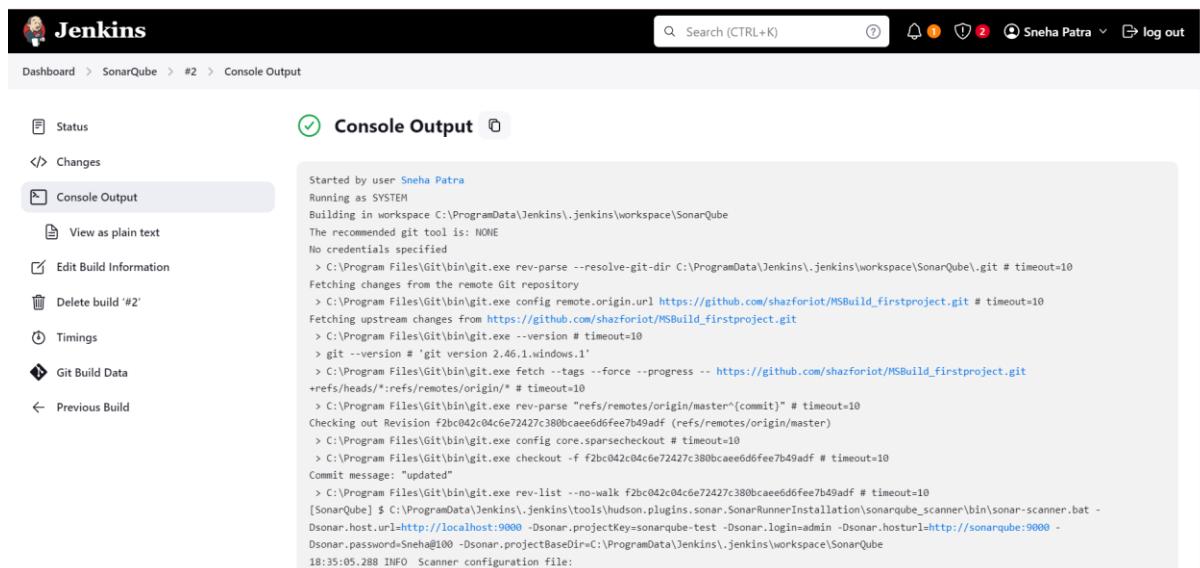
Started by user Sneha Patra

This run spent:

- 5 ms waiting;
- 1 min 16 sec build duration;
- 1 min 16 sec total from scheduled to completion.

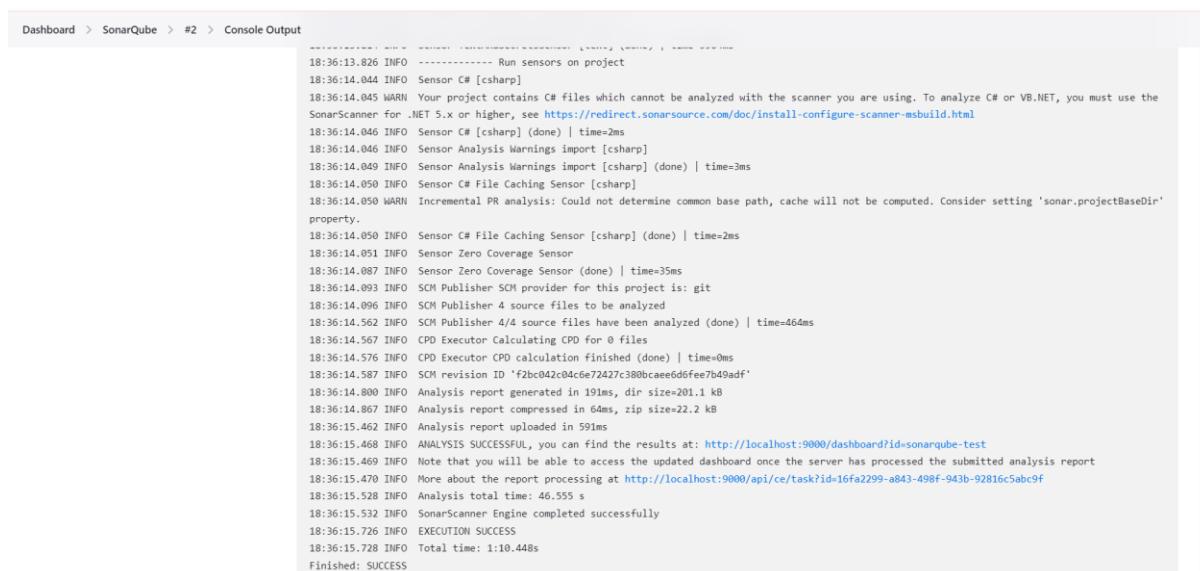
Revision: f2bc042c04c6e72427c380bcae6d6fee7b49adf
Repository: https://github.com/shazforiot/MSBuild_firstproject.git

Step 10: Check the console output.



The screenshot shows the Jenkins interface for a build named "SonarQube". The "Console Output" tab is selected. The log shows the execution of a git fetch command and the configuration of a SonarScanner job. The log ends with a success message.

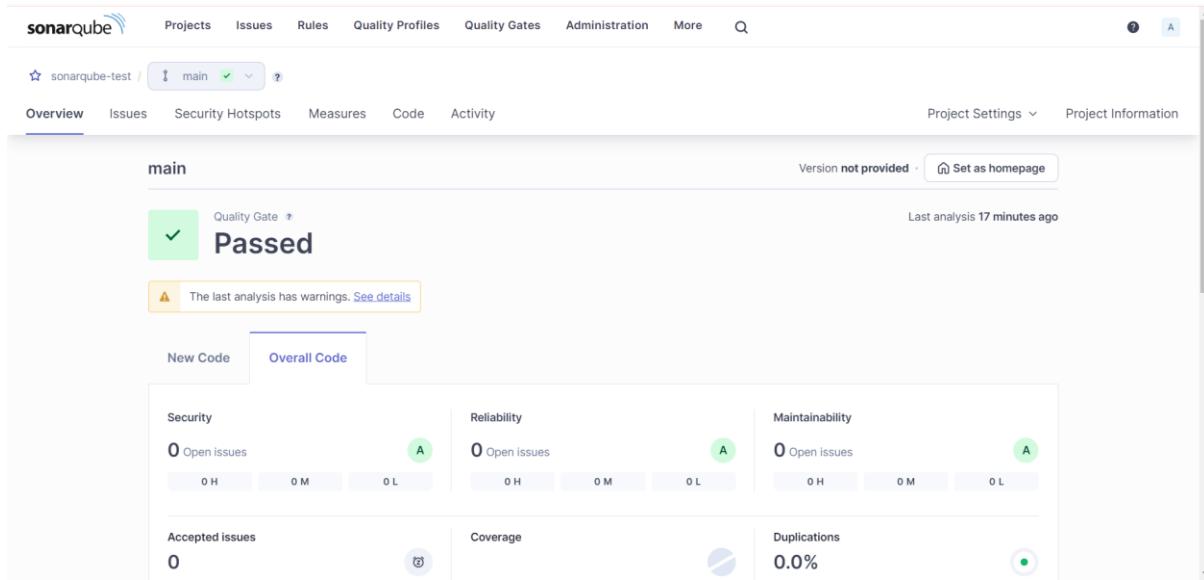
```
Started by user Sneha Patra
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\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
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.46.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
Checking our Revision f2bc042c04c6e72427c380bc4ee6dfee7b49adf (refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f f2bc042c04c6e72427c380bc4ee6dfee7b49adf # timeout=10
Commit message: "updated"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk f2bc042c04c6e72427c380bc4ee6dfee7b49adf # timeout=10
[SonarQube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_scanner\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test -Dsonar.login=admin -Dsonar.hostUrl=http://sonarqube:9000 -
Dsonar.password=Sneha@100 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
18:35:05.288 INFO Scanner configuration file:
18:35:05.288 INFO Scanning completed
```



This screenshot shows a different section of the Jenkins console output for the same build. It displays a detailed log of the analysis process, including sensor imports, warnings about unsupported file types, and the final successful completion of the analysis report.

```
18:36:13.826 INFO ----- Run sensors on project
18:36:14.044 INFO Sensor C# [sharp]
18:36:14.045 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
18:36:14.046 INFO Sensor C# [sharp] (done) | time=2ms
18:36:14.046 INFO Sensor Analysis Warnings import [sharp]
18:36:14.049 INFO Sensor Analysis Warnings import [sharp] (done) | time=3ms
18:36:14.050 INFO Sensor C# File Caching Sensor [sharp]
18:36:14.050 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
18:36:14.050 INFO Sensor C# File Caching Sensor [sharp] (done) | time=2ms
18:36:14.051 INFO Sensor Zero Coverage Sensor
18:36:14.087 INFO Sensor Zero Coverage Sensor (done) | time=35ms
18:36:14.093 INFO SCM Publisher SCM provider for this project is: git
18:36:14.096 INFO SCM Publisher 4 source files to be analyzed
18:36:14.562 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=464ms
18:36:14.567 INFO CPD Executor Calculating CPD for 0 files
18:36:14.576 INFO CPD Executor CPD calculation finished (done) | time=0ms
18:36:14.587 INFO SCM revision ID: f2bc042c04c6e72427c380bc4ee6dfee7b49adf
18:36:14.800 INFO Analysis report generated in 191ms, dir size=201.1 kB
18:36:14.867 INFO Analysis report compressed in 64ms, zip size=22.2 kB
18:36:15.462 INFO Analysis report uploaded in 99ms
18:36:15.468 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
18:36:15.469 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:36:15.470 INFO More about the report processing at http://localhost:9000/api/ce/task?id=16fa2299-a843-498f-943b-92816c5abc9f
18:36:15.528 INFO Analysis total time: 46.555 s
18:36:15.532 INFO SonarScanner Engine completed successfully
18:36:15.726 INFO EXECUTION SUCCESS
18:36:15.728 INFO Total time: 1:10.448s
Finished: SUCCESS
```

Step 11: Once the build is successful, check the project in SonarQube. In this way, we have integrated Jenkins with SonarQube for SAST.



Conclusion:

In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.

Experiment 8

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

Step 1 : Visit the following link to download the SonarScanner CLI -

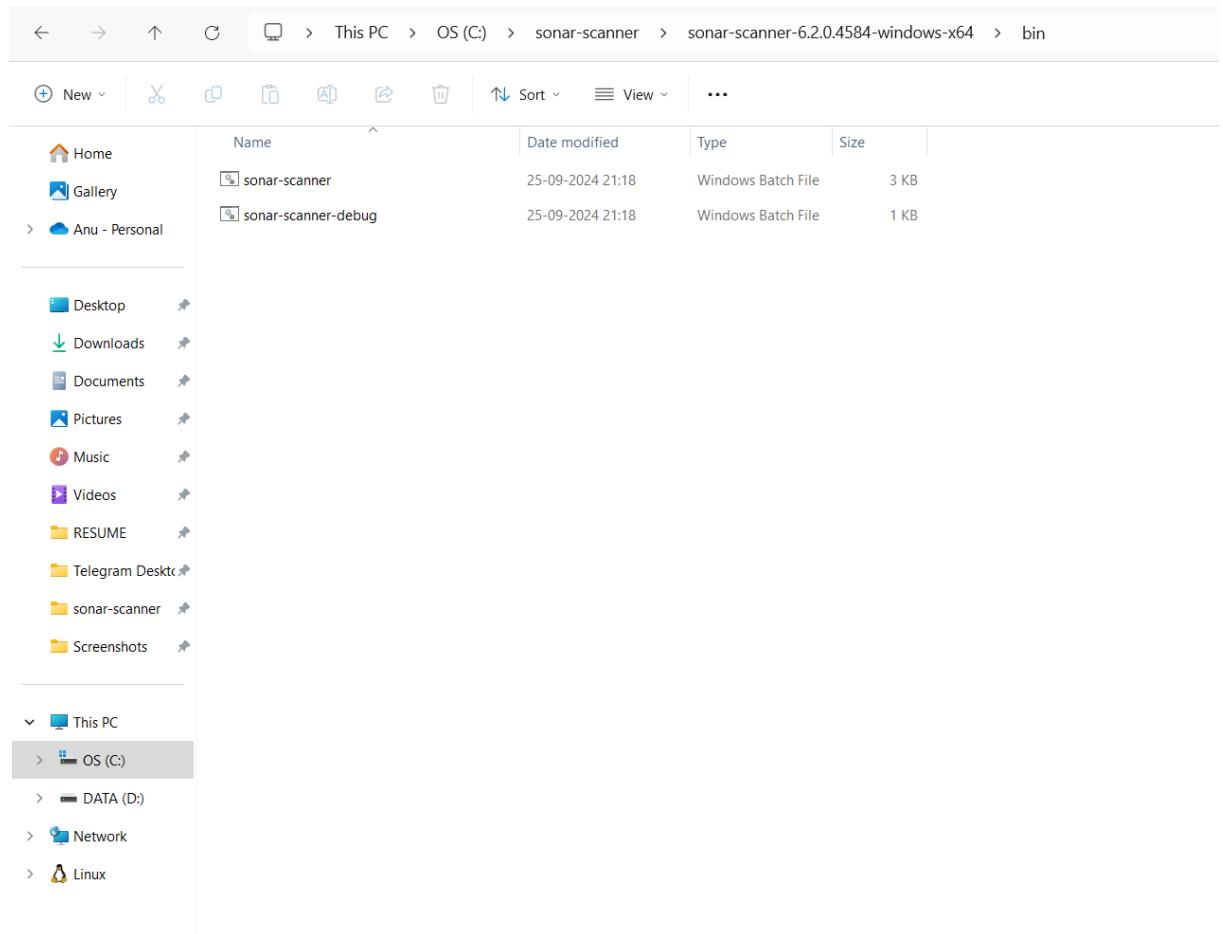
<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/> and then click on Windows x-64 to download the zip file.

The screenshot shows the 'SonarScanner CLI' page under the 'Analyzing source code' section. The main content area displays version 6.2 details, including support for PKCS12 truststore generated with OpenSSL and download links for various platforms. A note states that SonarScanners run on checked-out code. To the right, there's a sidebar titled 'On this page' with links to various configuration and usage topics.

Step 2: Then extract the content in C drive and name the folder sonar-scanner

The screenshot shows a Windows File Explorer window with the path 'OS (C:)'. The 'sonar-scanner' folder is selected. The details pane on the right shows the folder is a type 'File folder' located at 'C:\sonar-scanner' with a date modified of '25-09-2024 21:16'. A yellow folder icon is also visible in the details pane.

Name	Date modified	Type
data	26-11-2023 16:28	File folder
dell	07-05-2023 12:31	File folder
Drivers	07-10-2022 03:08	File folder
e-logo	04-12-2022 20:02	File folder
ghcup	22-07-2023 18:02	File folder
kubectl	13-11-2023 10:20	File folder
MinGW	24-10-2022 11:26	File folder
msys64	24-10-2022 10:51	File folder
PerfLogs	07-05-2022 10:54	File folder
Program Files	25-09-2024 21:28	File folder
Program Files (x86)	18-07-2024 20:15	File folder
tools	22-07-2023 18:24	File folder
Users	24-10-2022 04:17	File folder
Windows	15-09-2024 10:56	File folder
xampp	24-09-2023 20:52	File folder
sonar-scanner	25-09-2024 21:16	File folder



Step 3: Then now open Cmd Prompt and run as administrator and run the following commands –

cd C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin

dir

sonar-scanner.bat

```
Administrator: Command Prompt
C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>dir
Volume in drive C is OS
Volume Serial Number is E83B-22BB

Directory of C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin

25-09-2024 21:18    <DIR>        .
25-09-2024 21:18    <DIR>        ..
25-09-2024 21:18            805 sonar-scanner-debug.bat
25-09-2024 21:18            2,553 sonar-scanner.bat
25-09-2024 21:18            3,358 bytes
25-09-2024 21:18           2 File(s)   3,358 bytes
25-09-2024 21:18          2 Dir(s)  8,509,411,328 bytes free

C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>sonar-scanner.bat
22:44:22.348 INFO Scanner configuration file: C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\..\conf\sonar-scanner.properties
22:44:22.353 INFO Project root configuration file: NONE
22:44:22.369 INFO SonarScanner CLI 6.2.0.4584
22:44:22.370 INFO Java 17.0.12 Eclipse Adoptium (64-bit)
22:44:22.371 INFO Windows 11 10.0 amd64
22:44:22.389 INFO User cache: C:\Users\User\.sonar\cache
22:44:22.827 INFO JRE provisioning: os[windows], arch[amd64]
22:44:23.921 INFO EXECUTION FAILURE
22:44:23.923 INFO Total time: 1.577s
22:44:23.923 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [https://api.sonarcloud.io/analysis/jres?os=windows&arch=amd64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
22:44:23.925 ERROR
22:44:23.926 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.

C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin>
```

Step 4: Open Jenkins and create a pipeline and name the pipeline SonarQube Pipeline

Enter an item name

= Required field

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

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

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

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

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

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

OK

Step 5: In the configuration, under the Pipeline Section write the following Pipeline Script -
node {

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

stage('SonarQube analysis') {
    withSonarQubeEnv('sonarqube') {
        bat "C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat \
            -D sonar.login=admin \
            -D sonar.password=sonarqube \
            -D sonar.projectKey=sonarqube-test \
            -D sonar.exclusions=vendor/**,resources/**,**/*.java \
            -D sonar.host.url=http://127.0.0.1:9000/"
    }
}
}
```

Then click on the save button.

Configure

General

Advanced Project Options

Pipeline

Pipeline**Definition**

Pipeline script

Script ?

```

1+ node {
2+   stage('Cloning the GitHub Repo') {
3+     git 'https://github.com/shafzforiot/MSBuild_firstproject.git'
4+   }
5+   stage('SonarQube analysis') {
6+     withSonarQubeEnv('sonarqube') {
7+       Bat "C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat" \
8+           -D sonar.login=admin \
9+           -D sonar.password=sonarqube \
10+          -D sonar.projectKey=sonarqube-test \
11+          -D sonar.host.url='http://127.0.0.1:9000/' \
12+          -D sonar.host.url='http://127.0.0.1:9000/' \
13+     }
14+   }
15+ }
16

```

 Use Groovy Sandbox ?**Pipeline Syntax****Save****Apply**

REST API Jenkins 2.452.3

Step 6: Now, click on Build Now and the build is successful.**Status****SonarQube Pipeline**

</> Changes

▷ Build Now

⚙ Configure

🗑 Delete Pipeline

🔍 Full Stage View

SonarQube

⚙ Stages

✍ Rename

Pipeline Syntax

 Add description

Disable Project

Stage View

```

21:56:16.244 INFO  ----- Run sensors on project
21:56:16.428 INFO  Sensor C# [csharp]
21:56:16.429 WARN  Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
21:56:16.429 INFO  Sensor C# [csharp] (done) | time=1ms
21:56:16.430 INFO  Sensor Analysis Warnings import [csharp]
21:56:16.432 INFO  Sensor Analysis Warnings import [csharp] (done) | time=2ms
21:56:16.432 INFO  Sensor C# File Caching Sensor [csharp]
21:56:16.432 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
21:56:16.432 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=0ms
21:56:16.433 INFO  Sensor Zero Coverage Sensor
21:56:16.450 INFO  Sensor Zero Coverage Sensor (done) | time=16ms
21:56:16.494 INFO  CPD Executor Calculating CPD for 0 files
21:56:16.494 INFO  CPD Executor CPD calculation finished (done) | time=0ms
21:56:16.530 INFO  SCM revision ID 'f2bc042c04c6e724273080cae6d6fee7b49adfc'
21:56:16.704 INFO  Analysis report generated in 178ms, dir size=200.5 kB
21:56:16.773 INFO  Analysis report compressed in 68ms, zip size=21.9 kB
21:56:16.930 INFO  Analysis report uploaded in 155ms
21:56:16.931 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
21:56:16.931 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:56:16.932 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=aef67fb15-719a-4b23-8f38-5edc7a765dae
21:56:16.940 INFO  Analysis total time: 16.723 s
21:56:16.943 INFO  SonarScanner Engine completed successfully
21:56:17.042 INFO  EXECUTION SUCCESS
21:56:17.044 INFO  Total time: 19.574s
[Pipeline] 
[Pipeline] // withSonarQubeEnv
[Pipeline] 
[Pipeline] // stage
[Pipeline] 
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Step 7: Now, you can visit <http://127.0.0.1:9000/dashboard?id=sonarqube-test> to see the result.

The screenshot shows the SonarQube dashboard for the project 'sonarqube-test'. The main header includes the SonarQube logo, navigation links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the header, the project name 'sonarqube-test' is displayed with a 'main' branch selected. The dashboard has tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information dropdowns. A 'Quality Gate' section at the top right indicates a 'Passed' status with a green checkmark icon. A message below says 'The last analysis has warnings. See details'. The main content area is divided into two sections: 'New Code' and 'Overall Code'. Under 'Overall Code', there are six metrics: Security (0 Open issues), Reliability (0 Open issues), Maintainability (0 Open issues), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (0.0%). Below these, there is a 'Security Hotspots' section with 0 items. The overall status is 'Passed'.

This screenshot shows the SonarQube dashboard after a new code analysis. The main header and project information are identical to the previous screenshot. The 'Quality Gate' section now shows a 'Failed' status with a red exclamation mark icon. A message below says 'Learn how to improve your code base by cleaning only new code.' There are two buttons: 'Take the Tour' and 'Not now'. The main content area under 'Overall Code' shows significantly higher issue counts compared to the first screenshot. Metrics include: Security (0 Open issues), Reliability (68k Open issues), Maintainability (164k Open issues), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6%). The overall status is 'Failed'.

Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, and security vulnerabilities on our sample Java application.

Experiment 9

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

Step 1: Create an EC2 Instance and name it as nagios-host

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, and Instances. The main area displays a table with one row for the instance 'nagios-host'. The instance details are as follows:

- Name:** nagios-host
- Instance ID:** i-0ea55a0966a726e57
- Instance state:** Running
- Instance type:** t2.micro
- Status check:** 2/2 checks passed
- Alarm status:** View alarms
- Availability Zone:** us-east-1a
- Public IPv4 DNS:** ec2-18-212-3-

Below the table, a detailed view for the instance 'i-0ea55a0966a726e57 (nagios-host)' is shown. It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's an 'Instance summary' section with fields like Instance ID, Public IPv4 address, Instance state, and Private IP DNS name.

Step 2: Under the security groups, click on edit inbound rules and set as shown in the figure below

The screenshot shows the AWS Security Groups page. A security group named 'sgr-0fd2456e348e44b06' is selected. The inbound rules table lists the following rules:

Protocol	Port Range	Source	Action
SSH	22	Custom (0.0.0.0)	Allow
HTTP	80	Anywhere (::/0)	Allow
All ICMP - IPv6	All	Anywhere (::/0)	Allow
HTTPS	443	Anywhere (0.0.0.0/0)	Allow
All ICMP - IPv4	All	Anywhere (0.0.0.0/0)	Allow
Custom TCP	0	Anywhere (0.0.0.0/0)	Allow
All traffic	All	Anywhere (0.0.0.0/0)	Allow

At the bottom left, there's a button labeled 'Add rule'.

Step 3: Then select the instance nagios-host and then connect the instance.

Connect to instance Info

Connect to your instance i-025f1d18f7c8a8cda (nagios-host) using any of these options

[EC2 Instance Connect](#)[Session Manager](#)[SSH client](#)[EC2 serial console](#)**All ports are open to all IPv4 addresses in your security group**

All ports are currently open to all IPv4 addresses, indicated by All and 0.0.0.0/0 in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#).

Instance ID

[i-025f1d18f7c8a8cda \(nagios-host\)](#)

Connection Type

 Connect using EC2 Instance Connect

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

 Connect using EC2 Instance Connect Endpoint

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

 Public IPv4 address[3.86.198.73](#) IPv6 address

...

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

 [X](#)[CloudShell](#)[Feedback](#)© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 3: Now, run the following commands -

```
sudo su
```

```
sudo yum update
```

```
sudo yum install httpd php
```

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

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

aws | Services | [Alt+S]

```
[ec2-user@ip-172-31-91-19 ~]$ sudo su
[ec2-user@ip-172-31-91-19 ec2-user]# sudo yum update
Last metadata expiration check: 0:11:42 ago on Mon Sep 30 16:40:21 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-91-19 ec2-user]# sudo yum install httpd php
Last metadata expiration check: 0:12:03 ago on Mon Sep 30 16:40:21 2024.
Dependencies resolved.

=====
Package           Architecture      Version          Repository        Size
=====
Installing:
httpd            x86_64          2.4.62-1.amzn2023
php8.3           x86_64          8.3.10-1.amzn2023.0.1
                                          
Installing dependencies:
apr               x86_64          1.7.2-2.amzn2023.0.2
apr-util          x86_64          1.6.3-1.amzn2023.0.1
generic-logos-httpd    noarch        18.0.0-12.amzn2023.0.3
httpd-core        x86_64          2.4.62-1.amzn2023
httpd-filesystem  x86_64          2.4.62-1.amzn2023
httpd-tools       x86_64          2.4.62-1.amzn2023
libbrotli         x86_64          1.0.9-4.amzn2023.0.2
libsodium          x86_64          1.0.19-4.amzn2023
libxslt           x86_64          1.1.34-5.amzn2023.0.2
mailcap           noarch        2.1.49-3.amzn2023.0.3
nginx-filesystem x86_64          1:1.24.0-1.amzn2023.0.4
php8.3-cli        x86_64          8.3.10-1.amzn2023.0.1
                                          
amazonlinux        48 k
amazonlinux        10 k
amazonlinux        129 k
amazonlinux        98 k
amazonlinux        19 k
amazonlinux        1.4 M
amazonlinux        14 k
amazonlinux        81 k
amazonlinux        315 k
amazonlinux        176 k
amazonlinux        241 k
amazonlinux        33 k
amazonlinux        9.8 k
amazonlinux        3.7 M

=====
i-0ea55a0966a726e57 (nagios-host)
PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19
```

aws Services Q Search [Alt+S] N. Virginia voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾

```
[root@ip-172-31-91-19 ec2-user]# sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:14:35 ago on Mon Sep 30 16:40:21 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 x86_64          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
  libxcrypt-devel  x86_64          4.4.33-7.amzn2023   amazonlinux   32 k
  make             x86_64          1:4.3-5.amzn2023.0.2   amazonlinux   534 k
=====
Transaction Summary
=====
Install 13 Packages
```

i-0ea55a0966a726e57 (nagios-host) X
PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19

aws Services Q Search [Alt+S] N. Virginia voclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾

```
[root@ip-172-31-91-19 ec2-user]# sudo yum install gd gd-devel
Last metadata expiration check: 0:16:13 ago on Mon Sep 30 16:40:21 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      x86_64          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
  graphite2        x86_64          1.3.14-7.amzn2023.0.2   amazonlinux   97 k
  graphite2-devel  x86_64          1.3.14-7.amzn2023.0.2   amazonlinux   21 k
  harfbuzz         x86_64          7.0.0-2.amzn2023.0.1   amazonlinux   868 k
  harfbuzz-devel   x86_64          7.0.0-2.amzn2023.0.1   amazonlinux   404 k
  harfbuzz-icu     x86_64          7.0.0-2.amzn2023.0.1   amazonlinux   18 k
=====
i-0ea55a0966a726e57 (nagios-host) X  
PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19
```

Step 4: Create a new nagios user with its password.

```
sudo adduser -m nagios
sudo passwd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

```
[root@ip-172-31-93-157 ec2-user]# sudo adduser -m nagios
[root@ip-172-31-93-157 ec2-user]# sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-93-157 ec2-user]# sudo groupadd nagcmd
[root@ip-172-31-93-157 ec2-user]# sudo usermod -a -G nagcmd nagios
[root@ip-172-31-93-157 ec2-user]# sudo usermod -a -G nagcmd apache
[root@ip-172-31-93-157 ec2-user]# ]
```

Step 5: Now, run the following commands -

```
mkdir ~/downloads
cd ~/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
tar zxvf nagios-4.0.8.tar.gz
```

```
[root@ip-172-31-93-157 ec2-user]# mkdir ~/downloads
[root@ip-172-31-93-157 ec2-user]# cd ~/downloads
[root@ip-172-31-93-157 downloads]# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.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.gz: No such file or directory
bash: gz: command not found
[root@ip-172-31-93-157 downloads]# wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-09-30 17:00:06-- 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  6.16MB/s   in 0.4s
2024-09-30 17:00:07 (6.16 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]

[root@ip-172-31-93-157 downloads]# tar zxvf nagios-4.0.8.tar.gz
tar (child): nagios-4.0.8.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
[root@ip-172-31-93-157 downloads]# ]
```

To resolve the error run the following commands -

```
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
tar zxvf nagios-4.0.8.tar.gz
tar zxvf nagios-plugins-2.0.3.tar.gz
cd nagios-4.0.8
```

```
(root@ip-172-31-93-157 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
--2024-09-30 17:03:04-- http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net)... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-30 17:03:04-- http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Reusing existing connection to prdownloads.sourceforge.net:80.
HTTP request sent, awaiting response... 302 Found
Location: http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1 [following]
--2024-09-30 17:03:04-- http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1
Resolving versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)... 162.251.232.173
Connecting to versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net)|162.251.232.173|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz          100%[=====]  1.72M  2.21MB/s   in 0.8s

2024-09-30 17:03:05 (2.21 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

--2024-09-30 17:03:05-- 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.l'

nagios-plugins-2.0.3.tar.gz.l 100%[=====]  2.54M  7.26MB/s   in 0.3s

2024-09-30 17:03:05 (7.26 MB/s) - 'nagios-plugins-2.0.3.tar.gz.l' saved [2659772/2659772]
```

This screenshot shows a terminal window within the AWS CloudWatch interface. The user has run several `wget` commands to download the Nagios and Nagios Plugins source code. The output of these commands is displayed in the terminal window, showing the progress and final results of the downloads.

```
nagios-plugins-2.0.3/plugins-scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins-scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins-scripts/utils.pm.in
nagios-plugins-2.0.3/plugins-scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins-scripts/t/
nagios-plugins-2.0.3/plugins-scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins-scripts/t/check_file_age.t
nagios-plugins-2.0.3/plugins-scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins-scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins-scripts/t/utils.t
nagios-plugins-2.0.3/plugins-scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins-scripts/check_wave.pl
nagios-plugins-2.0.3/plugins-scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins-scripts/utils.sh.in
nagios-plugins-2.0.3/plugins-scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requirements
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpk
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requirements
[root@ip-172-31-91-19 downloads]#
```

i-0ea55a0966a726e57 (nagios-host)
PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19

Step 6: Now to run the configuration script run the following command.

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

This screenshot shows the terminal window again, this time executing the configuration step. The user runs the `./configure` command with the `--with-command-group=nagcmd` option. The terminal displays the configuration process, which includes checking for various system components and settings to ensure they are compatible with the build.

```
[root@ip-172-31-91-19 downloads]# cd nagios-4.0.8
[root@ip-172-31-91-19 nagios-4.0.8]# ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $ (MAKE)... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -E... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
```

i-0ea55a0966a726e57 (nagios-host)
PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19

Step 7: Now, to compile the source code run the following command -

`make all`

```

aws Services Search [Alt+S] N. Virginia v vclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-91-19 nagios-4.0.8]# make all
cd ./base && make
make[1]: Entering directory '/root/downloads/nagios-4.0.8/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ..//common/shared.o ..//common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nerd.o nerd.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
    inlined from 'get_worker' at workers.c:224:12:
workers.c:209:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  209 |         log_debug_info(DEBUG_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
  |         ^
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
commands.c: In function 'process_passive_service_check':
commands.c:224:19: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
  224 |     cr.source = command_worker.source_name;
  |     ^
commands.c: In function 'process_passive_host_check':
commands.c:233:19: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
  233 |     cr.source = command_worker.source_name;
  |     ^
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c

```

i-0ea55a0966a726e57 (nagios-host)

PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19

```

aws Services Search [Alt+S] N. Virginia v vclabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
[root@ip-172-31-91-19 nagios-4.0.8]# sudo make install
cd ./base && make install
make[1]: Entering directory '/root/downloads/nagios-4.0.8/base'
make install-basic
make[2]: Entering directory '/root/downloads/nagios-4.0.8/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/base'
make strip-post-install
make[2]: Entering directory '/root/downloads/nagios-4.0.8/base'
/usr/bin/strip /usr/local/nagios/bin/nagios
/usr/bin/strip /usr/local/nagios/bin/nagiosstats
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/base'
make[1]: Leaving directory '/root/downloads/nagios-4.0.8/base'
cd ./cgi && make install
make[1]: Entering directory '/root/downloads/nagios-4.0.8/cgi'
make install-basic
make[2]: Entering directory '/root/downloads/nagios-4.0.8/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
/usr/bin/install: cannot stat '*.cgi': No such file or directory
make[2]: *** [Makefile:205: install-basic] Error 1
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/cgi'
make[1]: *** [Makefile:197: install] Error 2
make[1]: Leaving directory '/root/downloads/nagios-4.0.8/cgi'

```

i-0ea55a0966a726e57 (nagios-host)

PublicIPs: 18.212.3.162 PrivateIPs: 172.31.91.19

```

[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios
*** Init script installed ***
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg
*** Config files installed ***
Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagcmd -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
*** External command directory configured ***
[root@ip-172-31-93-157 nagios-4.0.8]# []

```

To resolve the errors run the following commands -

sudo yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel

rm -rf nagios-4.0.8

cd ~/downloads/nagios-4.4.6

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

make all

sudo make install

```

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

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

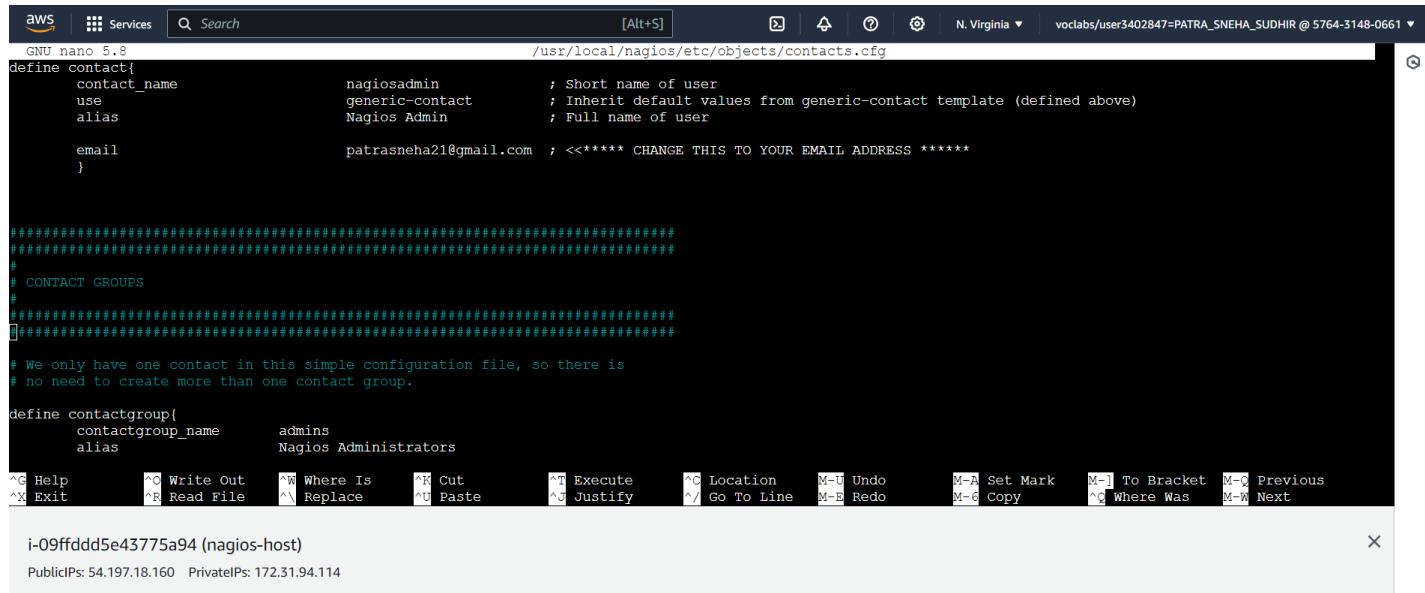
before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:
  - What version of Nagios you are using
  - What version of the plugins you are using
  - Relevant snippets from your config files
  - Relevant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

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

*****
enjoy.
root@nagios:~#
```

Step 8: Edit the config file and change the email address.

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



```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402847=PATRA_SNEHA_SUDHIR @ 5764-3148-0661 ▾
GNU nano 5.8 /usr/local/nagios/etc/objects/contacts.cfg
define contact{
  contact_name          nagiosadmin      ; Short name of user
  use                   generic-contact   ; Inherit default values from generic-contact template (defined above)
  alias                Nagios Admin     ; Full name of user
  email                patrasneha21@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
#
# CONTACT GROUPS
#
#####

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

define contactgroup{
  contactgroup_name      admins
  alias                 Nagios Administrators
```

i-09ffddd5e43775a94 (nagios-host) X

Public IPs: 54.197.18.160 Private IPs: 172.31.94.114

Step 9: Now run the following commands –

`sudo make install-webconf`

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

`sudo service httpd restart`

`cd ~/downloads`

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

```
- Read the documentation on the Nagios Library at:  
  https://library.nagios.com  
  
before you post a question to one of the mailing lists.  
Also make sure to include pertinent information that could  
help others help you. This might include:  
- What version of Nagios you are using  
- What version of the plugins you are using  
- Relevant snippets from your config files  
- Relevant error messages from the Nagios log file
```

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

Enjoy.

```
(root@ip-172-31-93-157 nagios-4.4.6]# sudo nano /usr/local/nagios/etc/objects/contacts.cfg  
(root@ip-172-31-93-157 nagios-4.4.6]# sudo make install-webconf  
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf  
if { 0 -eq 1 }; then \  
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \  
fi  
*** Nagios/Apache conf file installed ***  
(root@ip-172-31-93-157 nagios-4.4.6]# sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
New password:  
Re-type new password:  
Adding password for user nagiosadmin
```

```
[root@ip-172-31-93-157 nagios-4.4.6]# sudo service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
(root@ip-172-31-93-157 nagios-4.4.6]# cd ~/downloads  
tar zxvf nagios-plugins-2.0.3.tar.gz  
nagios-plugins-2.0.3/  
nagios-plugins-2.0.3/perlmod/  
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  
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz  
nagios-plugins-2.0.3/perlmods/Try-Tiny-0.18.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Implementation-0.07.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile  
nagios-plugins-2.0.3/perlmods/Perl-OSType-1.003.tar.gz  
nagios-plugins-2.0.3/perlmods/install_order  
nagios-plugins-2.0.3/perlmods/Nagios-Plugin-0.36.tar.gz  
nagios-plugins-2.0.3/perlmods/Math-Calc-Units-1.07.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Build-0.4007.tar.gz  
nagios-plugins-2.0.3/ABOUT-NLS  
nagios-plugins-2.0.3/configure.ac  
nagios-plugins-2.0.3/Makefile.in  
nagios-plugins-2.0.3/config.h.in  
nagios-plugins-2.0.3/ChangeLog  
nagios-plugins-2.0.3/AUTHORS  
nagios-plugins-2.0.3/lib/  
nagios-plugins-2.0.3/lib/parses_ini.h  
nagios-plugins-2.0.3/lib/extr(opts.c  
nagios-plugins-2.0.3/lib/Makefile.in
```

Step 10: Compile and install plugins

```
cd nagios-plugins-2.0.3  
.configure --with-nagios-user=nagios --with-nagios-group=nagios  
make  
sudo make install
```

```
/usr/bin/install -c -o nagios -g nagios check_dhcp /usr/local/nagios/libexec/check_dhcp  
chown root /usr/local/nagios/libexec/check_dhcp  
chmod ug=rw,u+s /usr/local/nagios/libexec/check_dhcp  
/usr/bin/install -c -o nagios check_icmp /usr/local/nagios/libexec/check_icmp  
chown root /usr/local/nagios/libexec/check_icmp  
chmod ug=rw,u+s /usr/local/nagios/libexec/check_icmp  
make[2]: Nothing to be done for 'install-data-am'.  
make[2]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/plugins-root'  
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/plugins-root'  
taking install in po  
make[1]: Entering directory '/root/downloads/nagios-plugins-2.0.3/po'  
/usr/bin/mkdir -p /usr/local/nagios/share  
installing fr.gmo as /usr/local/nagios/share/locale/fr/LC_MESSAGES/nagios-plugins.mo  
installing de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo  
if test "nagios-plugins" = "gettext-tools"; then \  
    /usr/bin/mkdir -p /usr/local/nagios/share/gettext/po; \  
    for file in Makefile.in.in remove-potdate.sin Makevars.template; do \  
        /usr/bin/install -c -o nagios -g nagios -m 644 ./file \  
            /usr/local/nagios/share/gettext/po/$file; \  
    done; \  
    for file in Makevars; do \  
        rm -f /usr/local/nagios/share/gettext/po/$file; \  
    done; \  
else \  
    : ; \  
fi  
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3/po'  
make[1]: Entering directory '/root/downloads/nagios-plugins-2.0.3'  
make[2]: Entering directory '/root/downloads/nagios-plugins-2.0.3'  
make[2]: Nothing to be done for 'install-exec-am'.  
make[2]: Nothing to be done for 'install-data-am'.  
make[2]: Leaving directory '/root/downloads/nagios-plugins-2.0.3'  
make[1]: Leaving directory '/root/downloads/nagios-plugins-2.0.3'  
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# []
```

Step 11: To start nagios run the following commands –

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

```
sudo chkconfig nagios on
```

Verify using the following command -

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

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo chkconfig --add nagios
sudo chkconfig nagios on
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
  Read object config files okay...

running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
```

If there are no errors run the following command –

```
sudo service nagios start
```

```
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
  Read object config files okay...

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo service nagios start
Starting nagios (via systemctl):
                                         [ OK ]
```

Check status using the following command -

```
sudo systemctl status nagios
```

```

root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios; generated)
   Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
     Docs: man:systemv-generator(8)
  Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
  Memory: 2.2M
    CPU: 52ms
  CGroup: /system.slice/nagios.service
          ├─80009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
          ├─80011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─80012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─80013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─80014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          └─80037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please note that this warning was suppressed by the configuration file.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please note that this warning was suppressed by the configuration file.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please note that this warning was suppressed by the configuration file.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please note that this warning was suppressed by the configuration file.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file worker with pid 80037
lines 1-26/26 (END)

```

Step 12: Go to EC2 instance and copy the public IP address of the instance. Now visit

http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.

Conclusion: Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) was installed and configured.

Experiment 10

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

Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -

```
sudo systemctl status nagios
```

on the nagios-host instance.

```
(root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
   Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
     Docs: man:systemd-sysv-generator(8)
 Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
  Memory: 2.2M
    CPU: 52ms
   CGroup: /system.slice/nagios.service
           └─0009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─0011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─0012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─0013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─0014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─0037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file worker with pid 80037
lines: 1-26/26 (END)
```

Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.

For now, leave this machine as it is, and go back to your nagios-host machine.

Name	Security group rule...	IP version	Type	Protocol	Port range
sgr-01439baf13aca75fa	IPv6	All ICMP - IPv6	IPv6 ICMP	All	
sgr-07d8ce2ac9f5e6a92	IPv4	All traffic	All	All	
sgr-071fd6724622dd2...	IPv4	HTTPS	TCP	443	
sgr-0e47c6681768287...	IPv4	SSH	TCP	22	
sgr-0f594249495210d...	IPv4	Custom TCP	TCP	5666	
sgr-07b7ba4fe05f14614	IPv4	HTTP	TCP	80	
sgr-0fcba4849fb0ee0409	IPv4	All ICMP - IPv4	ICMP	All	

Step 3: Now run the following command -

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

```

Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
Docs: man:systemd-sysv-generator(8)
Process: 79969 ExecStart-/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
Tasks: 6 (limit: 1112)
Memory: 2.2M
CPU: 52ms
CGroup: /system.slice/nagios.service
└─80009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
   ├─80011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
   ├─80012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
   ├─80013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
   ├─80014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
   └─80037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as @wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please see https://nagios-plugins.org/doc/troubleshooting.html#normal_check_interval_deprecated
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please see https://nagios-plugins.org/doc/troubleshooting.html#retry_check_interval_deprecated
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please see https://nagios-plugins.org/doc/troubleshooting.html#normal_check_interval_deprecated
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please see https://nagios-plugins.org/doc/troubleshooting.html#retry_check_interval_deprecated
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file Worker with pid 80037

[root@ip-172-31-93-157 nagios-plugins-2.0.3]# ps -ef | grep nagios
nagios 80009 1 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 80011 80009 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 80012 80009 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 80013 80009 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 80014 80009 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 80037 80009 0 18:02 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root 81960 3110 0 18:35 pts/1 00:00:00 grep --color=auto nagios
[root@ip-172-31-93-157 nagios-plugins-2.0.3]#

```

Step 4: Now, run the following commands -

`sudo su`

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

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

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

```

root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo su
root@ip-172-31-93-157 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts
root@ip-172-31-93-157 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-93-157 nagios-plugins-2.0.3]# cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: missing destination file operand after '/usr/local/nagios/etc/objects/localhost.cfg'
try 'cp --help' for more information.
ash: /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg: No such file or directory
root@ip-172-31-93-157 nagios-plugins-2.0.3]# nano
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
ash: /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg: No such file or directory
root@ip-172-31-93-157 nagios-plugins-2.0.3]# cp /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: missing destination file operand after '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg'
try 'cp --help' for more information.
root@ip-172-31-93-157 nagios-plugins-2.0.3]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-93-157 nagios-plugins-2.0.3]#

```

Step 5: Open `linuxserver.cfg` using the the following command -

`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.

Change hostgroup_name under hostgroup to linux-servers1

```

GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg
example of how you can create configuration entries to monitor
the local (Linux) machine.

#####
# 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    linux-server
    alias        linux-server
    address      3.95.202.23[]
}

#####

^G Help      ^O Write Out   ^W Where Is      ^M Cut       ^T Execute      ^C Location     M-U Undo      M-A Set Mark   M-J To Bracket M-Q Previous
^X Exit      ^R Read File    ^V Replace      ^U Paste      ^J Justify      ^Y Go To Line   M-E Redo      M-B Copy      ^Q Where Was   M-W Next

```

i-025f1d18f7c8a8cda (nagios-host)

```

GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg
check_command      check_local_swap!20!10
}

#####

# Define a service to check SSH on the local machine.
# Disable notifications for this service by default, as not all users may have SSH enabled.

define service{
    use          local-service           ; Name of service template to use
    host_name    linuxserver
    service_description  SSH
    check_command    check_ssh
    notifications_enabled  0
}

#####

# Define a service to check HTTP on the local machine.
# Disable notifications for this service by default, as not all users may have HTTP enabled.

define service{
    use          local-service           ; Name of service template to use
    host_name    linuxserver
    service_description  HTTP
    check_command    check_http
    notifications_enabled  0
}

#####

^G Help      ^O Write Out   ^W Where Is      ^M Cut       ^T Execute      ^C Location     M-U Undo      M-A Set Mark   M-J To Bracket M-Q Previous
^X Exit      ^R Read File    ^V Replace      ^U Paste      ^J Justify      ^Y Go To Line   M-E Redo      M-B Copy      ^Q Where Was   M-W Next

```

Step 6: Open Nagios config file and add the following line -
nano /usr/local/nagios/etc/nagios.cfg

Then add this line -

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

```

GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/templates.cfg

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

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

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

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

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

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

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

#####

^G Help      ^O Write Out   ^W Where Is      ^M Cut       ^T Execute      ^C Location     M-U Undo      M-A Set Mark   M-J To Bracket M-Q Previous
^X Exit      ^R Read File    ^V Replace      ^U Paste      ^J Justify      ^Y Go To Line   M-E Redo      M-B Copy      ^Q Where Was   M-W Next

```

Step 8: Verify configuration files using the following command -

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

If there are no errors, run the following command -

```
sudo service nagios start
```

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# /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...
  WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
  WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
  WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
  WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
  Error: Could not find any host matching 'linuxserver' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 45)
  Error: Could not expand members specified in hostgroup (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 45)
    Error processing object config files!

***> One or more problems was encountered while processing the config files...

Check your configuration file(s) to ensure that they contain valid
directives and data definitions. If you are upgrading from a previous
version of Nagios, you should be aware that some variables/definitions
may have been removed or modified in this version. Make sure to read
the HTML documentation regarding the config files, as well as the
'Whats New' section to find out what has changed.
```

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# 
```

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# 
```

Step 9: After entering the correct credentials, you will see this page.

The screenshot shows the Nagios web interface with the following sections:

- Current Network Status:** Last Updated: Mon Sep 30 19:13:49 UTC 2024. Updated every 30 seconds. Nagios Core™ 4.4.6 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:** Up: 2, Down: 0, Unreachable: 0, Pending: 0. All Problems: 9, All Types: 2.
- Service Status Totals:** Ok: 6, Warning: 1, Unknown: 0, Critical: 1, Pending: 8. All Problems: 2, All Types: 16.
- Host Status Details For All Host Groups:** Limit Results: 100. Host: ihostserver, Status: UP, Last Check: 09-30-2024 19:13:16, Duration: 0d 0h 0m 33s+. Status Information: PING OK - Packet loss = 0%, RTA = 1.82 ms. Host: localhost, Status: UP, Last Check: 09-30-2024 19:01:49, Duration: 0d 1h 11m 22s. Status Information: PING OK - Packet loss = 0%, RTA = 0.64 ms.
- Current Status:** Tactical Overview, Map (Legacy), Hosts, Services, Host Groups Summary, Grid, Service Groups Summary, Grid, Problems, Services (Unhandled), Hosts (Unhandled), Network Outages, Quick Search.
- Reports:** Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, Event Log.
- System:** Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration.

Conclusion: Port, Service monitoring, Windows/Linux server monitoring using Nagios was performed.

Experiment 11

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

Step1: Open up the Lambda Console and click on the Create button.

The screenshot shows the AWS Lambda Functions page. On the left, there's a sidebar with navigation links like Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main area displays a table titled 'Functions (5)' with columns for Function name, Description, Package type, Runtime, and Last modified. Three functions are listed:

- ModLabRole**: Description: updates LabRole to allow it to assume itself. Package type: Zip, Runtime: Python 3.8, Last modified: 2 months ago.
- RedshiftEventSubscription**: Description: Create Redshift event subscription to SNS Topic. Package type: Zip, Runtime: Python 3.8, Last modified: 2 months ago.
- RedshiftOver**: Description: Deletes Redshift Cluster if the. Package type: Zip, Runtime: Python 3.8, Last modified: 2 months ago.

A prominent orange 'Create function' button is located at the top right of the table area.

Step 2: Create a function and name it and select the runtime environment as Python 3.12 and for the role select the existing role LabRole.

This screenshot shows the 'Create function' wizard. The first step, 'Basic information', is displayed. It includes fields for 'Function name' (containing 'lambdasneha') and 'Runtime' (set to 'Python 3.12'). There are three radio button options for creating the function:

- Author from scratch**: Selected. Description: Start with a simple Hello World example.
- Use a blueprint**: Description: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image**: Description: Select a container image to deploy for your function.

This screenshot shows the 'Create function' wizard, specifically the 'Permissions' section. It includes a note about default execution roles and a dropdown for selecting an existing role. The 'Use an existing role' option is selected, and the 'LabRole' is chosen from the dropdown. Below this, there's a section for 'Advanced settings'.

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

Throttle Copy ARN Actions ▾

Export to Application Composer Download ▾

Description

Last modified 11 seconds ago

Function ARN arn:aws:lambda:us-east-1:576431480661:function:lambdasneha

Function URL Info

Step 3: Scroll down to the code source section and then visit Configuration section and click on edit.

Code source Info

File Edit Find View Go Tools Window Test Deploy

```
lambda_function.py
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
```

Step 4: Make the following changes and then click on the save button.

Lambda > Functions > lambdasneha > Edit basic settings

Edit basic settings

Basic settings Info

Description - optional

Memory Info Your function is allocated CPU proportional to the memory configured.

128 MB Set memory to between 128 MB and 10240 MB

Ephemeral storage Info You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB Set ephemeral storage (/tmp) to between 512 MB and 10240 MB

SnapStart Info Reduce start-up time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function is a candidate for SnapStart, Lambda periodically runs your function in the background and measures its execution time.

Step 5: Scroll down to the code source section and then visit Test section. Create a new event and then name the event. Now, click on Test.

Successfully updated the function lambdasneha.

Code source Info

File Edit Find View Go Tools Window Test Deploy

lambda_function Environment

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

Successfully updated the function lambdasneha.

Test event Info

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

Test event action

Create new event Edit saved event

Event name

myevent

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

Event sharing settings

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

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

Template - optional

hello-world

Successfully updated the function lambdasneha.

Template - optional

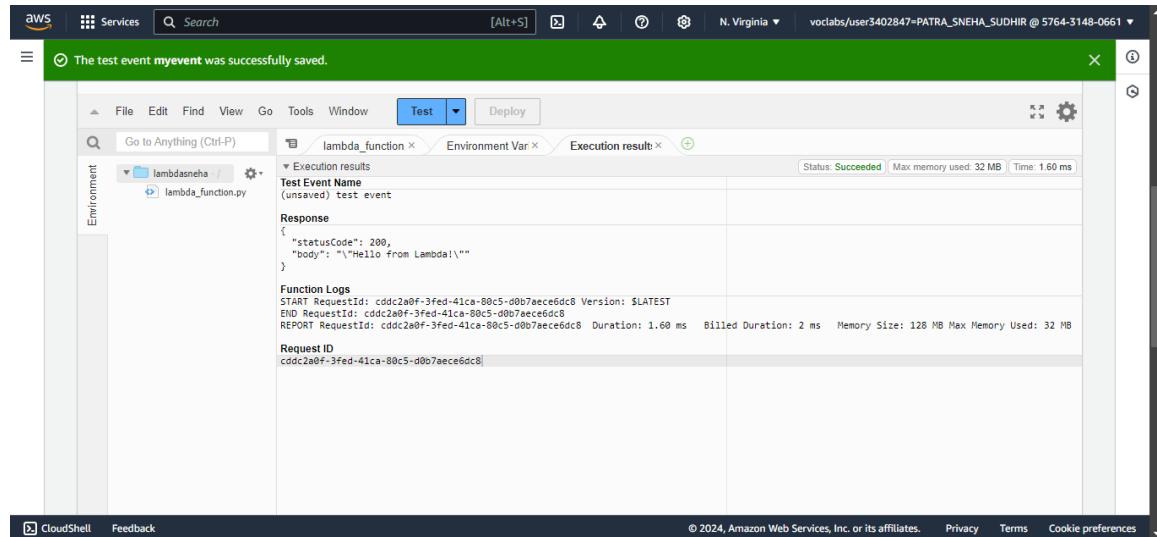
hello-world

Event JSON

Format JSON

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

Step 6: The test results are as shown below.



The screenshot shows the AWS Lambda Test console interface. At the top, a green banner displays the message "The test event myevent was successfully saved." Below the banner, the navigation bar includes File, Edit, Find, View, Go, Tools, Window, Test (which is currently selected), Deploy, and a gear icon. The main area has tabs for Go to Anything (Ctrl-P), lambda_function, Environment Var, and Execution result. The Execution result tab is active, showing the following details:

- Test Event Name:** (unsaved) test event
- Response:**

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

```
START RequestId: cddc2a0f-3fed-41ca-80c5-d0b7aece6dc8 Version: $LATEST
END RequestId: cddc2a0f-3fed-41ca-80c5-d0b7aece6dc8
REPORT RequestId: cddc2a0f-3fed-41ca-80c5-d0b7aece6dc8 Duration: 1.60 ms    Billed Duration: 2 ms    Memory Size: 128 MB Max Memory Used: 32 MB
Request ID
cddc2a0f-3fed-41ca-80c5-d0b7aece6dc8
```

At the bottom of the interface, there are links for CloudShell, Feedback, and a footer with copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Experiment-12

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

Step 1: First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

The screenshot shows the 'Create bucket' interface in the AWS Management Console. Under 'General configuration', the 'Bucket type' is selected as 'General purpose'. The bucket name is 'snehalambdabucket'. Other fields like 'AWS Region' (US East (N. Virginia) us-east-1) and 'Storage class' (Standard) are also present. The 'Bucket name' field is highlighted with a blue border.

The screenshot shows the 'Buckets' page in the AWS Management Console. A green success message at the top states 'Successfully created bucket "snehalambdabucket"'. Below it, there's an 'Account snapshot' card. The main area displays two buckets: 'elasticbeanstalk-us-east-1-576431480661' and 'snehalambdabucket'. The 'snehalambdabucket' row includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The 'snehalambdabucket' was created on August 8, 2024, at 14:24:28 (UTC+05:30).

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-us-east-1-576431480661	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 8, 2024, 14:24:28 (UTC+05:30)
snehalambdabucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 3, 2024, 14:41:44 (UTC+05:30)

Step 2: Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java. I have selected python environment.

The screenshot shows the AWS Lambda 'Create function' wizard. At the top, there are three options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section contains fields for 'Function name' (set to 'snehaimageloader') and 'Runtime' (set to 'Python 3.12'). A success message at the top right states: 'Successfully created the function snehaimageloader. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' The code editor shows a Python lambda function:

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

Step 3:Link the S3 bucket to the Lambda function by setting up a trigger.

Add trigger

Trigger configuration [Info](#)

S3 aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
 [X](#) [C](#)
Bucket region: us-east-1

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

[All object create events](#) [X](#)

Prefix (optional)

CloudShell **Feedback** © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Lambda > Functions > snehaimageloader

snehaimageloader

The trigger snehalambdabucket was successfully added to function snehaimageloader. The function is now receiving events from the trigger.

Function overview [Info](#) [Export to Application Composer](#) [Download](#)

Diagram **Template**

snehaimageloader

Layers (0)

S3 [+ Add destination](#)

[+ Add trigger](#)

Description
-

Last modified
12 minutes ago

Function ARN
[arn:aws:lambda:us-east-1:576431480661:function:snehaimageloader](#)

Function URL [Info](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshot shows the AWS Lambda function configuration page for a function named `snehaimageloader`. The left sidebar has tabs for Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. Under Configuration, the Triggers section is selected, showing one trigger named `S3: snehalambdabucket` which points to the S3 bucket `snehalamdbucket`. There are buttons for Add trigger, Fix errors, Edit, Delete, and Add trigger.

Step 4: Setup rthe required permissions.

The screenshot shows the AWS Lambda Permissions page for the function `snehaimageloader`. The left sidebar lists Lambda services like Dashboard, Applications, Functions, and the current item `snehaimageloader`. The main area shows the `Resource summary` and a `Resource-based policy statements` section. The policy statement table includes:

Statement ID	Principal	Actions
lambda-c0f0e...	s3.amazonaws...	lambda:Invoke...

Step 5: Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

The screenshot shows two stacked screenshots of the AWS S3 console. The top screenshot displays the 'Objects' tab for the 'snehalambdabucket'. It includes a toolbar with 'Create folder' and 'Upload' buttons, a search bar, and a table header for 'Name', 'Type', 'Last modified', and 'Size'. The message 'No objects' is displayed below the table. The bottom screenshot shows the 'Upload' page for the same bucket. It features a large dashed box for dragging files, a 'Files and folders' section listing 'image.jpg' (1 Total, 51.4 KB), and a 'Destination' section. Both screenshots have identical navigation bars at the top and bottom.

The screenshot shows the AWS S3 console with a green success banner at the top stating "Upload succeeded". Below it, the "Summary" section shows the destination "s3://snehalambdabucket" with a "Succeeded" status, indicating "1 file, 51.4 KB (100.00%)". The "Files and folders" tab is selected, displaying a table with one item: "image.jpg" (image/jpeg, 51.4 KB, Succeeded). The CloudWatch logs for the Lambda function show log events for a file upload, including INIT, START, and REPORT requests.

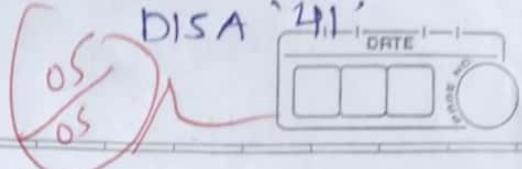
Name	Folder	Type	Size	Status	Error
image.jpg	-	image/jpeg	51.4 KB	Succeeded	-

The screenshot shows the AWS CloudWatch Logs interface for the Lambda function "aws/lambda/snehalambdaloader". It displays log events for a specific execution ID. The log entries include:

- 2024-10-03T09:40:03.528Z INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:18..
- 2024-10-03T09:40:03.660Z START RequestId: 087a7b65-0b18-4be4-a779-2fd7a077d523 Version: \$LATEST
- 2024-10-03T09:40:03.663Z END RequestId: 087a7b65-0b18-4be4-a779-2fd7a077d523
- 2024-10-03T09:40:03.663Z REPORT RequestId: 087a7b65-0b18-4be4-a779-2fd7a077d523 Duration: 2.15 ms Billed Duration: 3 ms Memory...

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.



Advance Devops

ASSIGNMENT NO: 1

Q1]

Use S3 bucket and host video streaming

Ans: Steps to host video streaming on S3 bucket

Prerequisites:

Before you start to host video streaming through S3 bucket, you must register and configure a custom domain (for example, example.com) with Route 53 so that you can configure your CloudFront to use a custom domain name later. Without a custom domain name, your S3 video is publicly and hosted through CloudFront at a URL that looks similar to the following <https://cloudfront.distribution.domain.name/path> to an S3 video.

• Step 1 : Create Bucket

Sign in to AWS console and open Amazon S3 console.

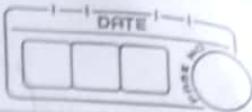
Then Select Bucket → Create Bucket > Enter Bucket name

→ Select Region → Block Public Access settings for this Bucket → Remaining setting defaults → Then choose (create bucket).

Step 2 : Upload a video to the S3 bucket

- Now , in buckets list , choose the name of the bucket that you created in step 1 to upload your file to
- On the objects tab for your bucket , choose Upload
- Then on upload page , under files and folders , choose Add files
- Then select a file to upload , and then choose Open for example , you can upload a video file named
- Choose upload

Then Open Cloudfront and follow next steps.



Step 3: Create a CloudFront origin access identity

- Sign in to AWS and open the CloudFront console
- Select Security section and choose Origin access
- Now under Identities tab, choose Origin access identity
- Enter name and choose create

Step 4: Create a CloudFront distribution

a)

Create a CloudFront distribution

- Choose Distribution → Create distribution
- Choose the domain name → Starts the name of S3 bucket you created
- For origin access, select Legacy access identities
- Under Origin access identity → Origin access identity
- Under Bucket policy, choose Yes update the bucket policy → Select Default cache behavior → Viewer protocol policy → Redirect HTTP to HTTPS.
- Keep the remaining setting to Default
- Now, choose Create distribution

b)

Review the bucket policy.

Step 5: Access the video through the CloudFront distribution

- Go to Distribution in the left panel
- Find the distribution by matching the S3 origin name and then copy the Domain Name and now Open a new tab and paste the copied Domain name
- Now, Return the previous tab, open the S3 console and select the bucket created in step 1 and then choose the video object uploaded in step 2 and then copy the key from the Object overview.

- In the new tab, append / and paste the key to the domain name
 - Your video is now accessible via CloudFront
- Step 6:** Configure your CloudFront distribution to use your custom domain name
- a) Request an SSL certificate
 - b) Add the alternate DNS to your CloudFront.
 - c) Create a DNS record to route traffic from your alternate domain name to your CloudFront distribution's Domain name.
 - d) Check whether IPv6 is enabled for your distribution and create another DNS record if needed
- Step 7:** Access the S3 video through the CloudFront distribution with the custom domain name
- Select Distribution → find distribution by matching S3 origin name → copy the alternate domain name → Paste the domain name.
 - Open S3 → find path to your S3 video → Return to the tab with domain, add / <S3 video path>
 - Access your video at https://cloudfront_domain/S3_video_path.
- Step 8:** View data about requests received by your CloudFront distribution (optional)
Now, your video streaming is live through S3 bucket.

Q3] Discuss BMW and Hotstar case studies using AWS

- Ans.
- The BMW Group whose headquarters is in Munich, Germany, is a global manufacturer of premium automobiles and motorcycles.
 - The company needed to scale its data lake to support the growing demands of internal and external stakeholders. As data wasn't easily accessible..
 - The BMW Group's innovation was slowed down by their own IT infrastructure and the long lead times required to support new initiatives. The BMW needed to develop a solution agile enough to both support the data needs of all the various internal business units and allow this company to move quickly to address the array of emerging use cases its customers demand.
 - The BMW Group re-architected its on-premises data lake to the AWS Cloud, creating a Cloud Data Hub (CDH) that integrates anonymized data from vehicle sensors and other sources.
 - Using AWS services like Amazon S3, Athena, Kinesis Data Firehose, and Glue, BMW streamlined data management and enabled scalable, agile operations for data engineers. The setup also allowed teams to maintain their own DevOps processes, fostering innovation. A modern web portal was implemented to help users discover and query trusted datasets, facilitating new insights.

- The company uses this data to monitor vehicle health indicators such as check control error to identify potential issues across vehicle lines
 - This enable the BMW groups to leverage fleet data ingested, collected and refined from the CDH to better resolve issues, even before they impact customers.
- HOTSTAR**
- Hotstar is an Indian subscription video on-demand streaming service owned by and operated by Star India, a subsidiary of The Walt Disney Company India.
 - In 2019, during ICC World Cup semi-final between India and New Zealand, Hotstar sets a new record of 25.3 million viewers. So, on the game day, the first spike witnessed was from 1.5M to 15M, as India started batting. Then, in between it was usual (10-12M) then Dhoni came to bat, and again sudden spike was noticed in traffic, taking it to 25.3M viewers. But then Dhoni got out, and suddenly there was drastic viewers drop to 4M viewers.
 - The very first challenge was handling 25.3M viewers. Secondly, when users dropped off the match some of them exited from the app entirely and others returned to the homepage and started exploring other content. That leads to an increase in load on homepage services.
 - Hotstar does not use traditional autoscaling from

AWS because there were a lot of challenges like

- Insufficient capacity errors.
- Step size, Autoscaling groups.
- They built their own scaling strategy and At the Back end side, Hotstar uses Amazon Route 53 and Amazon CloudFront are services for the Hotstar streaming video

Q3]

Why kubernetes and advantages and disadvantages of kubernetes. Explain how adidas uses Kubernetes?

Ans.

Imagine you have a bunch of different programs running on your computer. They need to work together and sometimes you want to run more copies of a program when things are busy. This can be hard to do manually, that's where kubernetes comes in.

Advantages of kubernetes

- Scalability - Kubernetes allows developers to easily scale their application up or down as demand fluctuates
- Resource efficiency
Kubernetes helps optimize the use of resources by scheduling containers to run on the most appropriate node based on their resource requirement.
- High availability
Kubernetes provides mechanisms for ensuring that applications are always available.
- Other advantages are Portability, Self-healing,

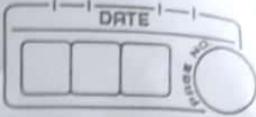
service discovery and load balancing, Extensi

Disadvantages of kubernetes:

- Complexity:
Kubernetes can be complex to set up and manage.
- Learning curve:
Developers and operations teams need to learn how to use Kubernetes effectively.
- Performance overhead:
Kubernetes introduces some overhead in terms of CPU and memory usage, which can impact application performance.
- Some other disadvantages are Security, Dependency on external services, lack of maturity, Complexity of networking.

Case Study: Adidas.

- Adidas is a globally renowned sportswear and athletic footwear company headquartered in Germany.
- In recent years, the ~~adidas~~ team was happy with the software choices from a technology perspective. But accessing all of the tools was a problem. For example, just to get a developer VM, you had to send a request, give the purpose, who's responsible, internal cost center and so on. So the best case is you got your machine in half an hour and the worst case in half week or sometimes even a week was a challenge faced by the ~~adidas~~.
- They found the solution with containerization,

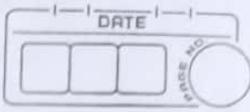


agile development, continuous delivery, and a cloud native platform that includes Kubernetes and Prometheus.

- Just six months after the project began, 100% of the adidas e-commerce site was running on Kubernetes. Load time for the e-commerce site was reduced by half. Release went from every 4-6 weeks to 3-4 times a day. With 4000 pods, 200 nodes, and 80,000 builds per month, adidas is now running 40% of its most critical, impactful systems on its cloud native platform.

Q4] what are Nagios and explain how Nagios are used in E-services.

- Ans.
- Nagios is a powerful monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes.
 - E-services S.R.L is an innovative energy company offering solutions in monitoring, VOIP, call centers and IT solutions. E-services S.R.L. chose to partner with Nagios and become the official representative of Nagios XI in Paraguay.
 - To set up a monitoring system across Paraguay one would have to carefully consider monitoring bandwidth, setting up different levels of support for hosts that are only being watched, a high availability system with a failover system,



an a system that can be accessed from afar.

- Nagios provided E-services and ANDE with real-time solution including:
 - A centralized monitoring system for their entire infrastructure, easing the sysAdmin workload
 - Helpful and intuitive statics that simplify decision making and aid troubleshooting
 - Easy-to-understand graphics and displays
 - Excellent availability, thanks to the mirrored servers with the failover system.

✓

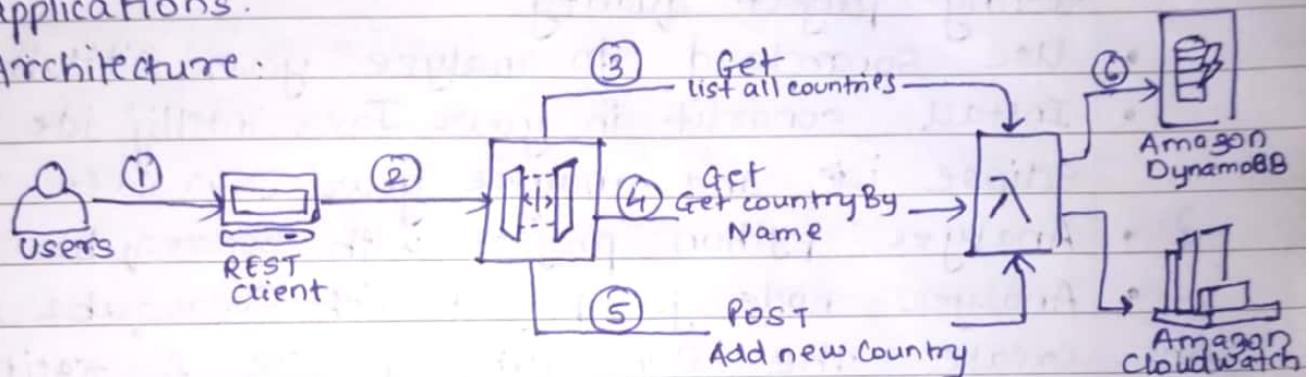
Advance DevOps

Assignment No : 2

Q1] Create a REST API with the Serverless framework.

Ans. AWS is a popular cloud provider that offers a myriad of services to support serverless architectures. One of the most common use cases involves the combination of Lambda, API Gateway, and DynamoDB to build scalable, efficient, and cost-effective applications.

Architecture:



Setting Up the Environment.

- To install the Serverless framework, you need to have Node.js installed on your machine. Once you have installed the serverless framework you can create a new serverless project using a template. Here we will use aws-nodejs template.
- The handler.js file is where we will write our Lambda function code.
- The serverless.yml file is where we will define our infrastructure. In this we will define DynamoDB table to store Data.
- Lambda function will be triggered by different endpoints of our API.
- After defining the serverless.yml file and

all of the functions, we can deploy our serverless application using following commands SLS deploy.

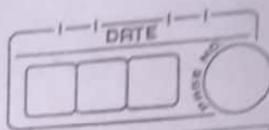
- After the deployment is complete, the serverless framework will output the URLs of API endpoint.

Q2] Case Study for Sonarqube

- Create your own profile in sonarqube for testing project quality.
- Use SonarCloud to analyze your GitHub code
- Install sonarlint in your Java intelliJ ide or eclipse ide and analyze your Java Code
- Analyze python project with Sonarqube
- Analyze node.js project with Sonarqube

Ans: Create the Sonarqube profile for testing project quality

- Then open Intelij setting, find Tools > Sonarlint entry and select + to open the connection wizard
- Enter a name for this connection, select SonarCloud or SonarQube.
- Choose the authentication method.
 - (a) Generate token on Sonarqube or SonarCloud
 - (b) Now add Username + Password : This can be used on Sonarqube connection only
- for SonarCloud only select organisation that you want to connect to.
- SonarQube and SonarCloud can push notification to developers.
- Validate the connection creating by selecting



finishing at the end of the wizard and save the connection in global setting by clicking OK.

- BIND Python Project to SonarQube
- Select SonarLint > Bind project to SonarQube
- choose the correct project from SonarQube
- Analyze the project (Python Project)
- Trigger an analysis by going to Code > Analyze code > SonarLint
- Analyze Node.js project
 - Make sure your Node.js project is properly configured with sonar-project.properties file or equivalent for the analysis to run.

Q3] At a large organization, your centralized operations team may get many repetitive infrastructure requests. You can use Terraform to build a "self-serve" infrastructure model that lets product teams manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying and managing services in your organization, allowing teams to efficiently deploy services in compliance with your organization's practices. Terraform Cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

Soln: At a large organization, implementing a self-service infrastructure model using Terraform can significantly streamline the process of managing infrastructure across different teams. This approach allows product teams to manage their own infrastructure independently while adhering to organizational standards and best practices.

1.] Standardization through Terraform Modules.

By creating and utilizing Terraform modules, organizations can codify their infrastructure deployment and management standards. These modules serve as reusable packages of Terraform configurations that encapsulate common patterns and best practices.

2.] Efficient Deployment:

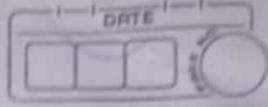
Product team can leverage these standardized modules to quickly deploy services without needing to reinvent the wheel or wait for the centralized operations team to handle every request.

3.] Compliance

By using predefined modules, teams ensure that their deployments comply with the organization's established practices and security guidelines.

4.] Automation.

The use of Terraform modules promotes



automation, reducing manual intervention and potential human errors in infrastructure management.

5.] Version Control.

With modules stored in version control systems like Git, teams can track changes, collaborate on improvements, and maintain a history of infrastructure configuration.

Integration with Ticketing Systems

Terraform Cloud offers integration capabilities that further enhance the self-service model:

1.] Automatic Infrastructure Requests.

Terraform Cloud can integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests. This automation streamlines the process of submitting and tracking infrastructure changes.

2.] Centralized Management.

By centralizing infrastructure management through Terraform Cloud, organizations can maintain better control over who can request and approve infrastructure changes.

3.] Governance.

The integration with ticketing systems allows for better governance of infrastructure requests, ensuring that all changes go through proper approval processes before deployment.

Collaborative Infrastructure Management

As organizations grow and adopt Terraform at scale, they often move towards a collaborative infrastructure management approach:

- 1.] Team-Based Permissions and Team-Based Implementing
- 2.] State and Run History
- 3.] Sensitive information Protection
- 4.] Module Registry

By implementing these features and practices, large organizations can effectively leverage Terraform to build a robust, scalable, and compliant infrastructure management system that supports both centralized control and decentralized team autonomy.