

Adv. Devops Experiment no. 1

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

Aim:

Part A) To host static and dynamic website on AWS and EC2.

Part B) To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Theory:

AWS EC2 (Elastic Compute Cloud)

1. What is it? AWS EC2 provides scalable virtual servers (instances) in the cloud. You can use EC2 to run applications and services without managing physical hardware.

2. Key Features:

- **Instance Types:** Various types optimized for compute, memory, storage, or GPU capabilities.
- **Auto Scaling:** Automatically adjusts the number of instances based on traffic.
- **Elastic Load Balancing (ELB):** Distributes incoming traffic across multiple instances.
- **Security:** Integrates with AWS Identity and Access Management (IAM) and provides security groups and network ACLs.
- **Pricing:** Pay-as-you-go or reserved instances for long-term commitments.

3. Common Use Cases:

- Hosting websites and web applications.
- Running big data analytics.
- Building development and test environments.
- Running high-performance computing (HPC) applications.

AWS S3 (Simple Storage Service)

1. What is it? AWS S3 is an object storage service that provides highly scalable and durable storage for a wide variety of data types.

2. Key Features:

- **Storage Classes:** Multiple classes like Standard, Intelligent-Tiering, One Zone-IA, Glacier for different use cases and cost savings.
- **Data Durability:** Designed for 99.999999999% durability over a given year.
- **Scalability:** Automatically scales to accommodate data growth.
- **Security:** Data encryption (at rest and in transit), and access control policies.
- **Versioning:** Keeps multiple versions of an object to recover from accidental deletions.

3. Common Use Cases:

- Backup and restore.
- Data archiving.
- Application data storage.
- Content distribution and hosting.

AWS Cloud9

1. What is it? AWS Cloud9 is a cloud-based Integrated Development Environment (IDE) that allows you to write, run, and debug code from a web browser.

2. Key Features:

- **Environment:** Comes with pre-configured environments (Ubuntu-based) that include necessary tools and libraries.
- **Collaboration:** Multiple users can collaborate in real-time within the same IDE environment.
- **Integrated Tools:** Built-in terminal, debugger, and support for various programming languages.
- **AWS Integration:** Seamless integration with AWS services for deploying applications directly from the IDE.
- **Customizable:** Allows custom configurations and preferences for your development environment.

3. Common Use Cases:

- Developing applications in a managed environment.
- Collaborative coding and debugging sessions.
- Learning and experimenting with code in a cloud-based IDE.

1. Static Website hosting using EC2(Ubuntu):

1) Instance creation and configuration:

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
Rohan's Ubuntu Server Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

 [Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Key pair (login) Info

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

Key pair name - *required*
  [Create new key pair](#)

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)

vpc-0f5e8abf0225b8a45

Subnet | [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#)

[Select existing security group](#)

We'll create a new security group called '**launch-wizard-3**' with the following rules:

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server	
<input type="checkbox"/> Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server	

<input type="checkbox"/>	Name 	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Rohan's Ubuntu...	i-0af1b42335baa28df	 Running  	t2.micro

Instance summary for i-0af1b42335baa28df (Rohan's Ubuntu Server) [Info](#)

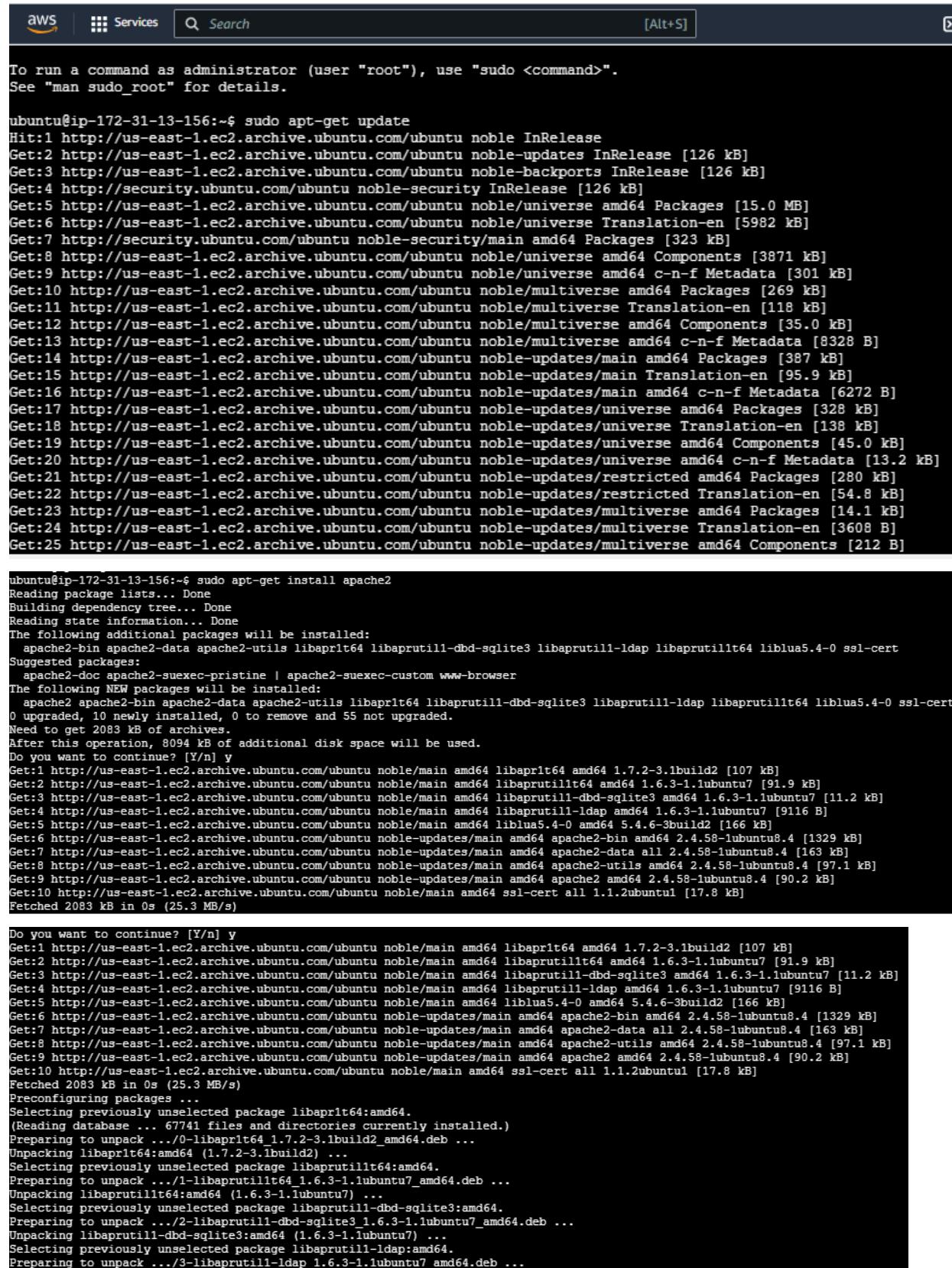
Updated less than a minute ago

[C](#) [Connect](#) [Instance state](#) [Actions](#)

Instance ID  i-0af1b42335baa28df (Rohan's Ubuntu Server)	Public IPv4 address  3.236.201.192 [open address]	Private IPv4 addresses  172.31.13.156
IPv6 address -	Instance state  Running	Public IPv4 DNS  ec2-3-236-201-192.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-13-156.ec2.internal	Private IP DNS name (IPv4 only)  ip-172-31-13-156.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding  Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address  3.236.201.192 [Public IP]	VPC ID  vpc-0f5e8abf0225b8a45	Auto Scaling Group name -
IAM Role -	Subnet ID  subnet-0a7cc411812f446e5	
IMDSv2 Required	Instance ARN  arn:aws:ec2:us-east-1:656404017537:instance/i-0af1b42335baa28df	

2) Connect to the instance and execute commands:

1. sudo apt-get update
2. sudo apt-get install apache2
3. systemctl status apache2



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

ubuntu@ip-172-31-13-156:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [323 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [387 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [95.9 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [6272 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [328 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [138 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [13.2 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [280 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [54.8 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]

ubuntu@ip-172-31-13-156:~$ sudo apt-get 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 libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 55 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://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]
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 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-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 (25.3 MB/s)

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]
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 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-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 (25.3 MB/s)

Preconfiguring packages ...
Selecting previously unselected package libapr1t64:amd64.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-libapr1t64_1.7.2-3.1build2_amd64.deb ...
Unpacking libapr1t64: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 ...
```

```

ubuntu@ip-172-31-13-156:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Thu 2024-08-22 17:23:12 UTC; 11min ago
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2399 (apache2)
     Tasks: 55 (limit: 1130)
    Memory: 5.4M (peak: 5.6M)
       CPU: 70ms
      CGroup: /system.slice/apache2.service
              ├─2399 /usr/sbin/apache2 -k start
              ├─2402 /usr/sbin/apache2 -k start
              └─2403 /usr/sbin/apache2 -k start

Aug 22 17:23:12 ip-172-31-13-156 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 22 17:23:12 ip-172-31-13-156 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-13-156:~$ sudo su
root@ip-172-31-13-156:/home/ubuntu# cd /var/www/html/
bash: cd: /var/www/html/: No such file or directory
root@ip-172-31-13-156:/home/ubuntu# cd /var/www/html/
root@ip-172-31-13-156:/var/www/html#

```

3) Editing the inbound rules:

Security Groups (1/1)

Security Groups (1/1)					
<input type="button" value="Create security group"/> Actions ▾ Export security groups to CSV ▾					
<input type="text" value="Find resources by attribute or tag"/> <input type="button" value="Clear filters"/>					
<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
<input checked="" type="checkbox"/>	-	sg-04536e9fd071d79ea	launch-wizard-3	vpc-0f5e8abf0225b8a45	launch-wizard-3 created 2024-08-22T

Inbound rules (1/1)

Inbound rules (1/1)						
<input type="button" value="Manage tags"/> <input type="button" value="Edit inbound rules"/> Actions ▾						
<input type="text" value="Search"/> <input type="button" value="Clear filters"/>						
<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input checked="" type="checkbox"/>	-	sgr-0f27b5ab0f675045d	IPv4	SSH	TCP	22

Edit inbound rules

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

Inbound rules						
Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0f27b5ab0f675045d	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>
-	HTTP	TCP	80	Anywhere...	<input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>
-	HTTPS	TCP	443	Anywhere...	<input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>

4) Go to the public IP link in new tab.

The screenshot shows a web browser window with the URL 3.236.201.192. The page is titled "Apache2 Default Page". It features the Ubuntu logo and the word "Ubuntu". A red button on the right says "It works!". Below the logo, there is a paragraph of text explaining the default welcome page. Another paragraph below it states that the site is currently unavailable due to maintenance. A section titled "Configuration Overview" provides information about the configuration layout of the Apache2 server on Ubuntu, mentioning files like apache2.conf, ports.conf, mods-enabled, and * Load. A code block shows the directory structure: /etc/apache2/ |-- apache2.conf | |-- ports.conf |-- mods-enabled | |-- * Load

1. Dynamic hosting using EC2(Amazon Linux):

1) Instance creation:

The screenshot shows the "Launch an instance" wizard in the AWS EC2 console. The first step, "Name and tags", is completed with the name "Rohan's Server". The "Add additional tags" button is visible. The second step, "Application and OS Images (Amazon Machine Image)", is partially visible below. The wizard guides the user through creating a virtual machine instance.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE I

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-066784287e358dad1 (64-bit (x86), uefi-preferred) / ami-023508951a94f0c71 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

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

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-066784287e358dad1

Verified provider

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

Instance type

t2.micro Free tier eligible

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

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

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

Key pair name - *required*

AWSLinux

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)
vpc-0f5e8abf0225b8a45

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#) [Select existing security group](#)

We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance 0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

⌚ Click refresh to view backup information ⟳
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

2) Connect to the instance:

Execute commands:

1. sudo su -
2. yum update -y
3. yum install -y httpd
4. systemctl status httpd
5. mkdir aws_assg3
6. cd aws_assg3
7. For this experiment we have created a website which we have uploaded on Github.com.
8. Copy the Download Link for the .zip file of the website.
9. using the wget command, download the zip file to the folder.
10. unzip the main.zip file and navigate into the "OnlinePrintingServices-main" folder using the cd command.
11. move all the contents from the folder to "/var/www/html/"

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

Last login: Wed Aug 21 20:17:27 2024 from 18.206.107.28
[ec2-user@ip-172-31-51-115 ~]$ sudo su -
[root@ip-172-31-51-115 ~]# yum update -y
Last metadata expiration check: 0:10:51 ago on Wed Aug 21 20:12:36 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-51-115 ~]# yum install -y httpd
Last metadata expiration check: 0:11:57 ago on Wed Aug 21 20:12:36 2024.
Dependencies resolved.

=====
Package           Architecture     Version          Repository      Size
=====
Installing:
httpd            x86_64          2.4.62-1.amzn2023   amazonlinux    48 k
Installing dependencies:
apr              x86_64          1.7.2-2.amzn2023.0.2   amazonlinux   129 k
apr-util          x86_64          1.6.3-1.amzn2023.0.1   amazonlinux   98 k
generic-logos-httpd noarch        18.0.0-12.amzn2023.0.3   amazonlinux   19 k
httpd-core        x86_64          2.4.62-1.amzn2023   amazonlinux   1.4 M
=====
httpd-core        x86_64          2.4.62-1.amzn2023   amazonlinux   1.4 M
httpd-filesystem noarch        2.4.62-1.amzn2023   amazonlinux   14 k
httpd-tools       x86_64          2.4.62-1.amzn2023   amazonlinux   81 k
libbrotli         x86_64          1.0.9-4.amzn2023.0.2   amazonlinux   315 k
mailcap           noarch        2.1.49-3.amzn2023.0.3   amazonlinux   33 k
=====
Installing weak dependencies:
apr-util-openssl x86_64          1.6.3-1.amzn2023.0.1   amazonlinux   17 k
mod_http2         x86_64          2.0.27-1.amzn2023.0.3   amazonlinux   166 k
mod_lua           x86_64          2.4.62-1.amzn2023   amazonlinux   61 k
=====
Transaction Summary
=====
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm          256 kB/s | 17 kB  00:00
(2/12): apr-1.7.2-2.amzn2023.0.2.x86_64.rpm                      1.8 MB/s | 129 kB  00:00
(3/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm                  1.2 MB/s | 98 kB  00:00
(4/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm     1.0 MB/s | 19 kB  00:00
(5/12): httpd-2.4.62-1.amzn2023.x86_64.rpm                     2.3 MB/s | 48 kB  00:00
(6/12): httpd-filesystem-2.4.62-1.amzn2023.noarch.rpm             657 kB/s | 14 kB  00:00
(7/12): httpd-core-2.4.62-1.amzn2023.x86_64.rpm                 34 MB/s | 1.4 MB  00:00
(8/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm                14 MB/s | 315 kB  00:00
(9/12): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm                 1.4 MB/s | 33 kB  00:00
(10/12): mod_http2-2.0.27-1.amzn2023.0.3.x86_64.rpm             5.9 MB/s | 166 kB  00:00
(11/12): mod_lua-2.4.62-1.amzn2023.x86_64.rpm                   3.5 MB/s | 61 kB  00:00
(12/12): httpd-tools-2.4.62-1.amzn2023.x86_64.rpm                708 kB/s | 81 kB  00:00
```

```
Total 8.7 MB/s | 2.3 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64 3/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 4/12
Installing : httpd-tools-2.4.62-1.amzn2023.x86_64 5/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 6/12
Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Installing : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Installing : httpd-core-2.4.62-1.amzn2023.x86_64 7/12
Installing : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 8/12
Installing : mod_lua-2.4.62-1.amzn2023.x86_64 9/12
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 10/12
Installing : httpd-2.4.62-1.amzn2023.x86_64 11/12
Running scriptlet: httpd-2.4.62-1.amzn2023.x86_64 12/12
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying : httpd-2.4.62-1.amzn2023.x86_64 5/12
Verifying : httpd-core-2.4.62-1.amzn2023.x86_64 6/12
Verifying : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Verifying : httpd-tools-2.4.62-1.amzn2023.x86_64 8/12

Verifying : httpd-tools-2.4.62-1.amzn2023.x86_64 9/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
Verifying : mod_lua-2.4.62-1.amzn2023.x86_64 12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64      generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.62-1.amzn2023.x86_64        httpd-core-2.4.62-1.amzn2023.x86_64        httpd-filesystem-2.4.62-1.amzn2023.noarch      httpd-tools-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64  mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.27-1.amzn2023.0.3.x86_64      mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
root@ip-172-31-51-115 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Active: inactive (dead)
      Docs: man:httpd.service(8)
root@ip-172-31-51-115 ~]# mkdir aws_assg3
root@ip-172-31-51-115 ~]# cd aws_assg3
root@ip-172-31-51-115 aws_assg3]# wget https://github.com/Rohan-Lalchandani08/OnlinePrintingServices.git
--2024-08-21 20:32:34- https://github.com/Rohan-Lalchandani08/OnlinePrintingServices.git
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/Rohan-Lalchandani08/OnlinePrintingServices [following]
--2024-08-21 20:32:34- https://github.com/Rohan-Lalchandani08/OnlinePrintingServices
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'OnlinePrintingServices.git'

Saving to: 'OnlinePrintingServices.git'

OnlinePrintingServices.git [ => ] 277.94K ---KB/s in 0.007s

2024-08-21 20:32:35 (39.4 MB/s) - 'OnlinePrintingServices.git' saved [284608]

[root@ip-172-31-51-115 aws_assg3]# ls -lrt
total 280
-rw-r--r--. 1 root root 284608 Aug 21 20:32 OnlinePrintingServices.git
[root@ip-172-31-51-115 aws_assg3]# wget https://github.com/Rohan-Lalchandani08/OnlinePrintingServices/archive/refs/heads/main.zip
--2024-08-21 20:35:21-- https://github.com/Rohan-Lalchandani08/OnlinePrintingServices/archive/refs/heads/main.zip
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/Rohan-Lalchandani08/OnlinePrintingServices/zip/refs/heads/main [following]
--2024-08-21 20:35:21-- https://codeload.github.com/Rohan-Lalchandani08/OnlinePrintingServices/zip/refs/heads/main
Resolving codeload.github.com (codeload.github.com)... 140.82.113.9
Connecting to codeload.github.com (codeload.github.com)|140.82.113.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'main.zip'

main.zip [ => ] 450.42K ---KB/s in 0.02s

2024-08-21 20:35:21 (28.4 MB/s) - 'main.zip' saved [461233]

[root@ip-172-31-51-115 aws_assg3]# ls -lrt
total 732
-rw-r--r--. 1 root root 284608 Aug 21 20:32 OnlinePrintingServices.git
-rw-r--r--. 1 root root 461233 Aug 21 20:35 main.zip
```

```
-rw-r--r--. 1 root root 284608 Aug 21 20:32 OnlinePrintingServices.git
-rw-r--r--. 1 root root 461233 Aug 21 20:35 main.zip
[root@ip-172-31-51-115 aws_assg3]# unzip main.zip
Archive: main.zip
e89afae28997f8992f8d48da34c717b34da37c24
    creating: OnlinePrintingServices-main/
    extracting: OnlinePrintingServices-main/about.html
    inflating: OnlinePrintingServices-main/audio.mp3
    extracting: OnlinePrintingServices-main/banner.png
    inflating: OnlinePrintingServices-main/brochure.jpg
    extracting: OnlinePrintingServices-main/buisnesscard.png
    inflating: OnlinePrintingServices-main/contact.html
    extracting: OnlinePrintingServices-main/facebook.png
    extracting: OnlinePrintingServices-main/flyers.png
    inflating: OnlinePrintingServices-main/index.html
    extracting: OnlinePrintingServices-main/insta.png
    extracting: OnlinePrintingServices-main/poster.png
    inflating: OnlinePrintingServices-main/printer logo.png
    inflating: OnlinePrintingServices-main/x.png
[root@ip-172-31-51-115 aws_assg3]# ls -lrt
total 748
drwxr-xr-x. 2 root root 16384 Aug 20 04:47 OnlinePrintingServices-main
-rw-r--r--. 1 root root 284608 Aug 21 20:32 OnlinePrintingServices.git
-rw-r--r--. 1 root root 461233 Aug 21 20:35 main.zip
[root@ip-172-31-51-115 aws_assg3]# cd OnlinePrintingServices-main
[root@ip-172-31-51-115 OnlinePrintingServices-main]# ls -lrt
total 480
-rw-r--r--. 1 root root 14913 Aug 20 04:47 x.png
-rw-r--r--. 1 root root 17827 Aug 20 04:47 'printer logo.png'
-rw-r--r--. 1 root root 68827 Aug 20 04:47 poster.png

-rw-r--r--. 1 root root 17827 Aug 20 04:47 'printer logo.png'
-rw-r--r--. 1 root root 68827 Aug 20 04:47 poster.png
-rw-r--r--. 1 root root 28213 Aug 20 04:47 insta.png
-rw-r--r--. 1 root root 2821 Aug 20 04:47 index.html
-rw-r--r--. 1 root root 102141 Aug 20 04:47 flyers.png
-rw-r--r--. 1 root root 11110 Aug 20 04:47 facebook.png
-rw-r--r--. 1 root root 936 Aug 20 04:47 contact.html
-rw-r--r--. 1 root root 34395 Aug 20 04:47 buisnesscard.png
-rw-r--r--. 1 root root 86675 Aug 20 04:47 brochure.jpg
-rw-r--r--. 1 root root 69398 Aug 20 04:47 banner.png
-rw-r--r--. 1 root root 29997 Aug 20 04:47 audio.mp3
-rw-r--r--. 1 root root 2 Aug 20 04:47 about.html
[root@ip-172-31-51-115 OnlinePrintingServices-main]# mv * /var/www/html/
[root@ip-172-31-51-115 OnlinePrintingServices-main]# cd /var/www/html/
[root@ip-172-31-51-115 html]# ls -lrt
total 480
-rw-r--r--. 1 root root 14913 Aug 20 04:47 x.png
-rw-r--r--. 1 root root 17827 Aug 20 04:47 'printer logo.png'
-rw-r--r--. 1 root root 68827 Aug 20 04:47 poster.png
-rw-r--r--. 1 root root 28213 Aug 20 04:47 insta.png
-rw-r--r--. 1 root root 2821 Aug 20 04:47 index.html
-rw-r--r--. 1 root root 102141 Aug 20 04:47 flyers.png
-rw-r--r--. 1 root root 11110 Aug 20 04:47 facebook.png
-rw-r--r--. 1 root root 936 Aug 20 04:47 contact.html
-rw-r--r--. 1 root root 34395 Aug 20 04:47 buisnesscard.png
-rw-r--r--. 1 root root 86675 Aug 20 04:47 brochure.jpg
-rw-r--r--. 1 root root 69398 Aug 20 04:47 banner.png
-rw-r--r--. 1 root root 29997 Aug 20 04:47 audio.mp3
-rw-r--r--. 1 root root 2 Aug 20 04:47 about.html
[root@ip-172-31-51-115 html]# ]
```

3) Editing the Inbound Rules:

Click on “Edit Inbound Rules” Button:

The screenshot shows three stacked interface panels. The top panel is a list of security groups, filtered by 'Security group name = launch-wizard-2'. It shows one entry: 'sg-0c0fbdd01d426cf3f' with 'Name: launch-wizard-2', 'VPC ID: vpc-0f5e8abf0225b8a45', and 'Description: launch-wizard-2 created 2024-08-21T'. The middle panel is titled 'Inbound rules (1)' and lists a single rule: 'sgr-0726ca15cc9357960' (Type: SSH, IP version: IPv4, Protocol: TCP, Port range: 22, Source: 0.0.0.0/0). The bottom panel is titled 'Inbound rules' and provides detailed settings for the selected rule: Type: SSH, Protocol: TCP, Port range: 22, Source: Custom (0.0.0.0/0), and Destination: 0.0.0.0/0. It also includes tabs for 'Info', 'Protocol Info', 'Port range Info', 'Source Info', and 'Description - optional Info'.

Now, Add new Rules as follows:

This screenshot shows the 'Inbound rules' interface after adding three new rules. The rules are listed as follows:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0726ca15cc9357960	SSH	TCP	22	Custom (0.0.0.0/0)	
-	HTTP	TCP	80	Anywhere...	Web Port
-	HTTPS	TCP	443	Anywhere...	Web Port

4) Check the status of httpd and then enable & start httpd using the following commands:

```
systemctl status httpd  
systemctl enable httpd  
systemctl start httpd
```

Now open the public ipv4 address allocated to the EC2 instance we created in new tab. We will be able to see the Website.

```
[root@ip-172-31-51-115 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Active: inactive (dead)
      Docs: man:httpd.service(8)
[root@ip-172-31-51-115 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-51-115 html]# start httpd
-bash: start: command not found
[root@ip-172-31-51-115 html]# systemctl start httpd
[root@ip-172-31-51-115 html]#
```

```
[root@ip-172-31-51-115 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Active: active (running) since Wed 2024-08-21 20:59:52 UTC; 3s ago
    Docs: man:httpd.service(8)
   Main PID: 28176 (httpd)
     Status: "Started, listening on: port 80"
       Tasks: 177 (limit: 1112)
      Memory: 12.9M
        CPU: 63ms
      CGroup: /system.slice/httpd.service
              ├─28176 /usr/sbin/httpd -DFOREGROUND
              ├─28177 /usr/sbin/httpd -DFOREGROUND
              ├─28178 /usr/sbin/httpd -DFOREGROUND
              ├─28179 /usr/sbin/httpd -DFOREGROUND
              └─28180 /usr/sbin/httpd -DFOREGROUND
```

```
Aug 21 20:59:52 ip-172-31-51-115.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Aug 21 20:59:52 ip-172-31-51-115.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Aug 21 20:59:52 ip-172-31-51-115.ec2.internal httpd[28176]: Server configured, listening on: port 80
```

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL `54.197.101.170`. To the left of the address bar are navigation icons (back, forward, search). To the right are a lock icon (Not secure), a file icon (IJCRT2203092.pdf), and a support link (Support for Latitude...).
- Page Content:**
 - Logo and Company Name:** A printer icon with the text "PRINTING COMPANY" below it.
 - Title:** "Print It - Online Printing Services for everyone"
 - Navigation Links:** Buttons for "About Us" and "Contact Us". Below these are four links:
 - [Introduction Audio](#)
 - [Promotional Video](#)
 - [Company Details](#)
 - [Services We Offer](#)
 - Text:** "Your one-stop solution for all printing needs."
 - Section Header:** "Introduction Audio" followed by a media player control bar with a play button, a progress bar showing 0:00 / 0:01, and other controls.
 - Section Header:** "Promotional Video" followed by a video player window. The video thumbnail shows a group of people at a business meeting. The video controls include a "Stock Videos" link, a play button, a timestamp (0:00), a "Watch later" button, and a "Share" button.

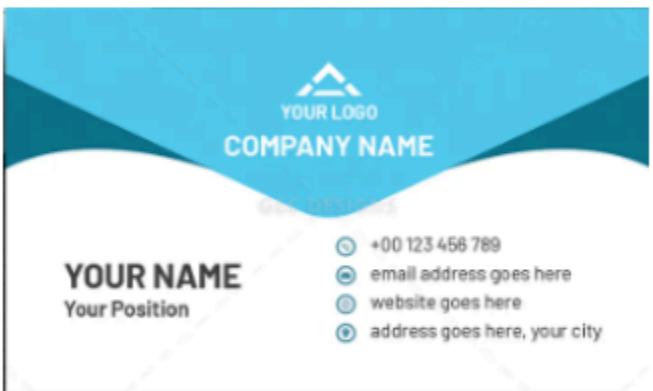
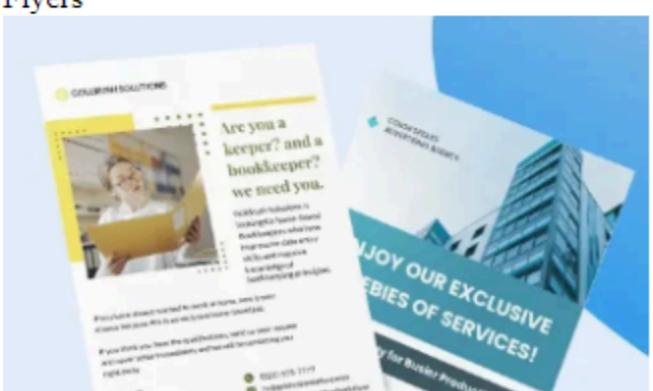
← → ⌛ Not secure 54.197.101.170

IJCRT2203092.pdf Support for Latitude...

▶

Company Details

Services We Offer

- Business Cards
- Flyers

3) Using S3 Bucket:

- 1) Search for s3 bucket and go to the link and click on create bucket.

The screenshot shows the Amazon S3 landing page under the 'Storage' category. The main heading is 'Amazon S3' with the tagline 'Store and retrieve any amount of data from anywhere'. Below the tagline, a brief description states: 'Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.' To the right, there is a white callout box with the heading 'Create a bucket' and the subtext: 'Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.' A prominent orange 'Create bucket' button is at the bottom of this box.

- 2) Configure the settings:

The screenshot shows the 'General configuration' step in the S3 bucket creation wizard. It includes fields for 'AWS Region' (set to 'US East (N. Virginia) us-east-1'), 'Bucket type' (with 'General purpose' selected), 'Bucket name' (set to 'rohans3bucket'), and 'Copy settings from existing bucket - optional' (disabled). The 'Choose bucket' button is visible, and the format is specified as 'Format: s3://bucket/prefix'.

The screenshot shows the 'Object Ownership' step in the S3 bucket creation wizard. It includes two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The description for 'ACLs disabled' states: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' The description for 'ACLs enabled' states: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.' At the bottom, the 'Object Ownership' and 'Bucket owner enforced' settings are confirmed.

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

- Disable
 Enable

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

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

Bucket Key

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

- Disable
 Enable

General purpose buckets (1) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

[Create bucket](#)

Find buckets by name

< 1 >

Name	AWS Region	IAM Access Analyzer	Creation date
rohans3bucket1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 23, 2024, 21:32:27 (UTC+05:30)

3) Upload the files.

Files and folders (13 Total, 456.9 KB)

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

All files and folders in this table will be uploaded.

Find by name

< 1 2 >

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	x.png	-	image/png
<input type="checkbox"/>	about.html	-	text/html
<input type="checkbox"/>	audio.mp3	-	audio/mpeg
<input type="checkbox"/>	banner.png	-	image/png
<input type="checkbox"/>	brochure.jpg	-	image/jpeg
<input type="checkbox"/>	buisnesscard.png	-	image/png
<input type="checkbox"/>	contact.html	-	text/html
<input type="checkbox"/>	facebook.png	-	image/png
<input type="checkbox"/>	flyers.png	-	image/png
<input type="checkbox"/>	index.html	-	text/html

4) Go to your bucket destination link.

Destination [Info](#)

Destination
<s3://rohans3bucket1>

▼ Destination details
Bucket settings that impact new objects stored in the specified destination.

Bucket Versioning When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended user actions and application failures. Learn more	Default encryption type If an encryption type isn't specified, bucket settings for default encryption are used to encrypt objects when storing them in Amazon S3. Learn more	Object Lock When enabled, objects in this bucket might be prevented from being deleted or overwritten for a fixed amount of time or indefinitely. Learn more
Enabled	Server-side encryption with Amazon S3 managed keys (SSE-S3)	Disabled

5) Uncheck the block public access.

[Amazon S3](#) > [Buckets](#) > [rohans3bucket1](#) > [Edit Block public access \(bucket settings\)](#)

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

Block public access (bucket settings)

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

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

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

6) Go to the properties section and scroll down to static website hosting and enable it.

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.

Error document - *optional*
This is returned when an error occurs.

7) Go to the permissions section and see for ACL's option and enable it.

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

ACLs disabled (recommended)
All objects in this bucket are owned by this account.
Access to this bucket and its objects is specified using only policies.

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

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

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

I acknowledge that ACLs will be restored.

Object Ownership

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer
The object writer remains the object owner.

- 8) Now, go to the objects section and select all objects and click on action button and select "Make public using ACL"

Name	Type	Last modified	Size
about.html	html	August 23, 2024, 21:37:28 (UTC+05:30)	
audio.mp3	mp3	August 23, 2024, 21:37:29 (UTC+05:30)	
banner.png	png	August 23, 2024, 21:37:30 (UTC+05:30)	
brochure.jpg	jpg	August 23, 2024, 21:37:31 (UTC+05:30)	
buisnesscard.png	png	August 23, 2024, 21:37:32 (UTC+05:30)	
contact.html	html	August 23, 2024, 21:37:33 (UTC+05:30)	
facebook.png	png	August 23, 2024, 21:37:34 (UTC+05:30)	
flyers.png	png	August 23, 2024, 21:37:35 (UTC+05:30)	
index.html	html	August 23, 2024, 21:37:36 (UTC+05:30)	
insta.png	png	August 23, 2024, 21:37:37 (UTC+05:30)	
poster.png	png	August 23, 2024, 21:37:38 (UTC+05:30)	67.2 KB Standard
printer logo.png	png	August 23, 2024, 21:37:39 (UTC+05:30)	17.4 KB Standard
x.png	png	August 23, 2024, 21:37:27 (UTC+05:30)	14.6 KB Standard

- 9) Copy the URL of the index.html file and go to the copied link in the new tab of the browser.

Name	Type	Last modified
contact.html	html	August 23, 2024, 21:37:33 (UTC+05:30)
facebook.png	png	August 23, 2024, 21:37:34 (UTC+05:30)
flyers.png	png	August 23, 2024, 21:37:35 (UTC+05:30)
index.html	html	August 23, 2024, 21:37:36 (UTC+05:30)
insta.png	png	August 23, 2024, 21:37:37 (UTC+05:30)
poster.png	png	August 23, 2024, 21:37:38 (UTC+05:30)
printer logo.png	png	August 23, 2024, 21:37:39 (UTC+05:30)
x.png	png	August 23, 2024, 21:37:27 (UTC+05:30)

10) Website is successfully deployed.

The screenshot shows a web browser window with the URL rohans3bucket1.s3.amazonaws.com/index.html. The page has a header with a printer icon and the text "PRINTING COMPANY". Below the header is a navigation bar with "About Us" and "Contact Us" buttons. A list of links includes "Introduction Audio", "Promotional Video", "Company Details", and "Services We Offer". A subtext below the links reads "Your one-stop solution for all printing needs." A section titled "Introduction Audio" features a video player showing a progress bar at 0:00 / 0:01. A section titled "Promotional Video" shows a thumbnail of a video titled "Successful business meeting of people discussing stock...". The video thumbnail includes a "Stock Video" logo, a play button, and sharing options for "Watch later" and "Share".

Part B) AWS Cloud9:

- 1) Search for IAM in the search box.

The screenshot shows the AWS Cloud9 search interface. A search bar at the top contains the text 'IAM'. Below it, a sidebar lists various categories: Services (11), Features (24), Resources (New), Documentation (59,444), Knowledge Articles (457), Marketplace (864), Blogs (1,846), Events (12), and Tutorials (1). The main content area displays search results for 'IAM'. It includes a 'Services' section with a card for 'IAM' (Manage access to AWS resources) and cards for 'IAM Identity Center' (Manage workforce user access) and 'Resource Access Manager' (Share AWS resources with other accounts or AWS Organizations). Below this is a 'Features' section with cards for 'Groups' (IAM feature) and 'Roles' (IAM feature). A sidebar on the right shows a list of actions: 'No ap...', 'Create...', and 'Data unavailable'.

- 2) Click on create user.

The screenshot shows the AWS IAM 'Users' page. The left sidebar has a 'Users' section under 'Access management'. The main content area shows a table titled 'Users (0) Info' with a note: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar and a 'Create user' button are at the top right. The table has columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Access key ID. A message 'No resources to display' is shown at the bottom.

3) Configuring the IAM role.

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Specify user details

User details

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

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

Are you providing console access to a person?

User type

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

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

Console password

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

Custom password

Are you providing console access to a person?

User type

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

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

Console password

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

Custom password
Enter a custom password for the user.
.....

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

Show password

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

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

Cancel **Next**

Step 1

[Specify user details](#)

Step 2

[Set permissions](#)

Step 3

[Review and create](#)

Step 4

[Retrieve password](#)

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group

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

Copy permissions

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

Attach policies directly

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

 **Get started with groups**

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

[Create group](#)

► **Set permissions boundary - *optional***

[Cancel](#)

[Previous](#)

[Next](#)

 **User created successfully**

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

X

Step 1

[Specify user details](#)

Step 2

[Set permissions](#)

Step 3

[Review and create](#)

Step 4

[Retrieve password](#)

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

[Email sign-in instructions](#)

Console sign-in URL

 <https://861276120101.signin.aws.amazon.com/console>

User name

 Rohan_IAM

Console password

 ***** [Show](#)

[Cancel](#)

[Download .csv file](#)

[Return to users list](#)

4) After creating IAM role, go to User groups, and click on create group.

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is titled "Identity and Access Management (IAM)" and contains the following navigation items:

- Dashboard
- Access management
 - User groups
 - Users
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access Analyzer
 - External access
 - Unused access
 - Analyzer settings
 - Credential report
 - Organization activity
- Service control policies

The main content area is titled "User groups (0) Info" and includes a search bar and a table header with columns: Group name, Users, Permissions, and Creation time. A message at the bottom states "No resources to display".

5) Configuring the user group.

The screenshot shows the "Create user group" page within the AWS IAM console. The left sidebar is identical to the previous screenshot. The main form is titled "Name the group" and contains a "User group name" field with the value "Rohan_IAM_Group". Below this, there is an optional section titled "Add users to the group - Optional (1/1) Info" which lists a single user "Rohan_IAM" with a checkmark next to their name. At the bottom, there is an optional section titled "Attach permissions policies - Optional (4/948) Info" with a "Filter by Type" button.

Screenshot of the AWS IAM User Groups page. A user group named "Cloud9" has been created and assigned to a user named "Rohan_IAM". Policies attached to the group include "AWSCloud9Administrator", "AWSCloud9EnvironmentMember", "AWSCloud9SSMInstanceProfile", and "AWSCloud9User".

Add users to the group - *Optional* (1/1) Info

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name Rohan_IAM

Groups Last activity Creation time

0 None 2 minutes ago

Attach permissions policies - *Optional* (4/948) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Filter by Type

Cloud9 All types 4 matches

Policy name	Type	Used as	Description
AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS ...
AWSCloud9EnvironmentMember	AWS managed	None	Provides the ability to be invited into ...
AWSCloud9SSMInstanceProfile	AWS managed	None	This policy will be used to attach a rol...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

Create user group

6) After creating a user group, search for Cloud9 in the search box.

Screenshot of the AWS search results for "Cloud9". The search results are categorized into Services and Features.

Search results for 'Cloud9'

Services (51)

- Features (32)
- Resources **New**
- Documentation (15,326)
- Knowledge Articles (652)
- Marketplace (13)
- Blogs (6,954)
- Events (325)
- Tutorials (22)

Services

- Cloud9 ★ A Cloud IDE for Writing, Running, and Debugging Code
- Amazon CodeCatalyst ★ Integrated DevOps Service
- AWS Cloud Map ★ Build a dynamic map of your cloud
- AWS Deadline Cloud ★ Simplified render management

See all 51 results ▶

Features

- Cloud WAN VPC feature
- Namespaces AWS Cloud Map feature

See all 32 results ▶

Data unavailable

7) Click on create environment.

The screenshot shows the AWS Cloud9 homepage under the 'Developer Tools' section. The main heading is 'AWS Cloud9: A cloud IDE for writing, running, and debugging code'. Below the heading is a brief description of what AWS Cloud9 allows you to do. To the right, there's a call-to-action box with the text 'New AWS Cloud9 environment' and a prominent orange 'Create environment' button. On the left, there's a 'How it works' section with a detailed description of how to create environments and switch between them, along with a 'Learn more' link. On the far right, there's a 'Getting started' sidebar with links to various documentation pages.

8) Configuring the environment.

The screenshot shows the 'Create environment' configuration page. At the top, there's a breadcrumb navigation: 'AWS Cloud9 > Environments > Create environment'. The main title is 'Create environment' with a 'Info' link. Below that is a 'Details' section with fields for 'Name' (containing 'demo 1') and 'Description - optional'. There's also a note about character limits. Under 'Environment type', there are two options: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' section includes a 'New EC2 instance' title and a 'Instance type' dropdown with three options: 't2.micro (1.7 GB RAM + 1 vCPU)', 't2.small (3.8 GB RAM + 2 vCPU)', and 't2.medium (7.5 GB RAM + 3 vCPU)'. Each option has a small 'Info' link next to it.

New EC2 instance

Instance type Info

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)

Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)

Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)

Recommended for production and most general-purpose development.

Additional instance types

Explore additional instances to fit your need.

Platform Info

This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023



Timeout

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

30 minutes



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

9) Environment is successfully created.

The screenshot shows the AWS Cloud9 interface. At the top, there's a green banner with a success message: "Successfully created demo 1. To get the most out of your environment, see Best practices for using AWS Cloud9". Below this is a blue banner with a note about AWS Toolkits. The main area is titled "Environments (1)" and shows a table with one row. The table columns are: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The single row contains: "demo 1", "Open", "EC2 instance", "Secure Shell (SSH)", "Owner", and a long ARN starting with "arn:aws:sts:...".

The screenshot shows the AWS Cloud9 IDE interface. The title bar says "us-east-1.console.aws.amazon.com/cloud9/ide/91871a92cae641439b1844ecdeef3da0?region=us-east-1". The menu bar includes File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run. On the left, there's a sidebar with a search bar ("Go to Anything (Ctrl-P)"), a file tree showing "demo 1 - /home/ec2-user/c9/README.md", and an AWS logo. The main area has a dark header with "Welcome" and "Developer Tools". Below that is a large "AWS Cloud9" logo and the text "Welcome to your development environment". A "Getting started" sidebar on the right lists "Create File", "Upload Files...", and "Clone from GitHub". At the bottom, there's a terminal window titled "bash - *p-172-31-45-14.e x" with the prompt "vocabs:~/environment \$".

Screenshot of a terminal and browser interface showing the development of a web page.

The terminal window (left) shows the file structure under `/IPLab-02`, including files like `about.html`, `audio.mp3`, `banner.png`, etc., and the content of the `index.html` file being edited:

```
4      <meta charset="UTF-8">
5      <meta name="viewport" content=
6          <title>Online Printing Services
7      </head>
8      <body>
9          <table>
10             <tr>
11                 <td>
12                 <td><h1><mark>Print It</h1>
13             </tr>
14         </table>
15         <hr>
16         <button onclick="location.href='about.html'">About Us</button>
17         <button onclick="location.href='contact.html'">Contact Us</button>
```

The browser window (right) displays the rendered HTML with the heading "Print It - Online Printing Services for everyone", a printer icon, and navigation links for "About Us" and "Contact Us". A sidebar on the right lists additional links:

- [Introduction Audio](#)
- [Promotional Video](#)
- [Company Details](#)
- [Services We Offer](#)

Adv. Devops Experiment no. 2

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

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

Theory:

Amazon Elastic Beanstalk is a Platform-as-a-Service (PaaS) offered by AWS that simplifies the deployment, management, and scaling of applications. It abstracts the underlying infrastructure, allowing developers to focus on writing code rather than managing servers and infrastructure.

Key Features

1. Easy Deployment:

- **Deployment Options:** Supports multiple deployment methods, including the AWS Management Console, CLI, and SDKs. You can deploy applications using ZIP files, Docker images, or from a source control repository.
- **Managed Platform Updates:** Automatically handles platform updates and patches for the underlying infrastructure.

2. Application Management:

- **Environment Management:** Provides pre-configured environments for popular application platforms like Node.js, Python, Java, .NET, PHP, Ruby, and Docker.
- **Monitoring and Logging:** Integrated with Amazon CloudWatch and AWS X-Ray for monitoring performance and logging. Offers health monitoring and application logs through the Elastic Beanstalk console.

3. Auto Scaling and Load Balancing:

- **Auto Scaling:** Automatically adjusts the number of instances based on traffic and resource utilization.

- **Load Balancing:** Distributes incoming application traffic across multiple instances to ensure high availability.

4. Customization and Flexibility:

- **Configuration Options:** Allows customization through configuration files (e.g., `.ebextensions`) and environment variables. You can configure instance types, scaling policies, and load balancer settings.
- **Integration with AWS Services:** Easily integrates with other AWS services like Amazon RDS (Relational Database Service), S3 (Simple Storage Service), and IAM (Identity and Access Management).

5. Environment Types:

- **Single Instance Environment:** Suitable for development and testing where high availability and fault tolerance are not required.
- **Load Balanced Environment:** Designed for production with auto-scaling, load balancing, and high availability.

Common Use Cases

- **Web Applications:** Deploy and manage web applications with a scalable backend and frontend.
- **APIs:** Host RESTful APIs or microservices with auto-scaling and high availability.
- **Development and Testing:** Quickly spin up environments for development and testing purposes.

Implementation:

- 1) Search for Elastic Beanstalk in the search box and click on create application.

The screenshot shows the AWS Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and a 'Compute' dropdown. Below the header, the title 'Amazon Elastic Beanstalk' is prominently displayed with the subtitle 'End-to-end web application management.' A descriptive paragraph explains that Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications. To the right, there's a 'Get started' section with a 'Create application' button, a 'Pricing' section stating there's no additional charge, and a 'Benefits and features' section. At the bottom right, there's a 'Getting started' link.

- 2) Configuring the environment.

This screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk setup wizard. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main area is titled 'Configure environment'. It contains three sections: 'Environment tier' (with options for Web server environment or Worker environment, currently set to Web server environment), 'Application information' (with an 'Application name' field containing 'RohanWebApp' and a note about the maximum length of 100 characters), and 'Environment information' (with a note about choosing a name, subdomain, and description that cannot be changed later). There are also 'Application tags (optional)' and 'Environment tags' sections at the bottom.

Platform Info

Platform type

Managed platform

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

Custom platform

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

Platform

Python

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

Platform version

4.1.3 (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 instances)

High availability

High availability (using spot and on-demand instances)

Custom configuration

- 3) Click on create service role and for selecting EC2 instance profile we need to create an IAM policy for it.

Step 1
[Configure environment](#)

Step 2
Configure service access

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

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

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

Step 6
[Review](#)

Configure service access Info

Service access

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

Service role

Create and use new service role
 Use an existing service role

Service role name
Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

[View permission details](#)

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

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

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) **Next**

- 4) Search for IAM in the search box and click on “roles” on the left hand side bar and click on create role.

The screenshot shows the AWS IAM search results page. The search term 'IAM' has been entered into the search bar at the top. The results are categorized into Services and Features.

- Services (11)**
 - IAM** Manage access to AWS resources
 - IAM Identity Center** Manage workforce user access to multiple AWS accounts and cloud applications
 - Resource Access Manager** Share AWS resources with other accounts or AWS Organizations
 - AWS App Mesh** Easily monitor and control microservices
- Features (24)**
 - Groups** IAM feature
 - Roles** IAM feature

At the bottom of the page, there is a note: "AWS IAM Access Analyzer now offers policy checks for public and critical resource access. 2 months ago".

The screenshot shows the AWS IAM Roles management page. The left sidebar is identical to the previous search results page, showing the 'Roles' section under 'Access management'.

The main content area displays a table of existing roles:

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-

Below the table, there are sections for "Roles Anywhere" and "Temporary credentials".

- Roles Anywhere**: Info: Authenticate your non AWS workloads and securely provide access to AWS services. Options include "Manage" and "Access AWS from your non AWS workloads". Description: Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.
- X.509 Standard**: Info: Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.
- Temporary credentials**: Info: Use temporary credentials with ease and benefit from the enhanced security they provide.

5) Configuring the IAM role.

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2

Choose a use case for the specified service.
Use case
 EC2
Allows EC2 instances to call AWS services on your behalf.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2

Choose a use case for the specified service.
Use case
 EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

6) We need to select three permission policies as shown below.

Permissions policies (3/947) Info			
Choose one or more policies to attach to your new role.			
<input type="text" value="elasticbean"/> Filter by Type <input type="button" value="X"/> All types 14 matches			
Policy name	Type	Description	
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permission...	
<input type="checkbox"/> AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance in your custom plat...	
<input type="checkbox"/> AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstalk Service policy for ...	
<input type="checkbox"/> AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	This policy is for the AWS Elastic Beanst...	
<input checked="" type="checkbox"/> AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontain...	
<input type="checkbox"/> AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only permissions. Explicitly ...	
<input type="checkbox"/> AWSElasticBeanstalkRoleCore	AWS managed	AWSElasticBeanstalkRoleCore (Elastic Be...	
<input type="checkbox"/> AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk operations role) Allow...	
<input type="checkbox"/> AWSElasticBeanstalkRoleECS	AWS managed	(Elastic Beanstalk operations role) Allow...	
<input type="checkbox"/> AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk operations role) Allow...	
<input type="checkbox"/> AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk operations role) Allow...	
<input type="checkbox"/> AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allow...	
<input checked="" type="checkbox"/> AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server...	
<input checked="" type="checkbox"/> AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker env...	

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=.,@~`^<>`{|}#\$/%&*`~`

Step 1: Select trusted entities

Trust policy

```

1 ~ [l
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10         "Service": [
11           "ec2.amazonaws.com"
12         ]
13       }
14     }
15   ]
16 ]

```

Step 1: Select trusted entities

Trust policy

```

1- [{}]
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": {
10-         "Service": [
11-           "ec2.amazonaws.com"
12-         ]
13-       }
14-     }
15-   ]
16- ]

```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

- 7) IAM Role for EC2 instance is successfully created, go back to “Configuring Service access” for elastic beanstalk and select the EC2 instance profile.

Role aws-elasticbeanstalk-ec2-role created. [View role](#)

IAM > Roles

Roles (4) Info

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

<input type="checkbox"/> Role name	Trusted entities	Last activity
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	-
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSRoleForSupport	AWS Service: support (Service-Linker)	-
AWSRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linker)	-

8) Select default settings as shown below in the next section.

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

Step 3 - optional
Set up networking, database, and tags

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

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

Step 6
[Review](#)

Set up networking, database, and tags - *optional* [Info](#)

Virtual Private Cloud (VPC)

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

vpc-0f5e8abf0225b8a45 | (172.31.0.0/16)

[Create custom VPC](#)

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
 Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Filter instance subnets					
-	Availability Zone	Subnet	▲	CIDR	Name
<input checked="" type="checkbox"/>	us-east-1a	subnet-01c3ce9cb...	▲	172.31.32.0/20	

[scaling](#)

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

Step 6
[Review](#)

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
 Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Filter instance subnets					
-	Availability Zone	Subnet	▲	CIDR	Name
<input checked="" type="checkbox"/>	us-east-1a	subnet-01c3ce9cb...	▲	172.31.32.0/20	
<input checked="" type="checkbox"/>	us-east-1f	subnet-055b0c11b...	▲	172.31.64.0/20	
<input type="checkbox"/>	us-east-1e	subnet-05c5390d8...	▲	172.31.48.0/20	
<input type="checkbox"/>	us-east-1c	subnet-0733979c0...	▲	172.31.80.0/20	
<input type="checkbox"/>	us-east-1b	subnet-0a7cc4118...	▲	172.31.0.0/20	
<input type="checkbox"/>	us-east-1d	subnet-0fd9b1dd1...	▲	172.31.16.0/20	

Database [Info](#)
 Integrate an RDS SQL database with your environment. [Learn more](#)

Database Info

Integrate an RDS SQL database with your environment. [Learn more](#)

Database subnets

If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

Choose database subnets (6)

Filter database subnets

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-1a	subnet-01c3ce9cb...	172.31.32.0/20	
<input type="checkbox"/>	us-east-1f	subnet-055b0c11b...	172.31.64.0/20	
<input type="checkbox"/>	us-east-1e	subnet-05c5390d8...	172.31.48.0/20	
<input type="checkbox"/>	us-east-1c	subnet-0733979c0...	172.31.80.0/20	
<input type="checkbox"/>	us-east-1b	subnet-0a7cc4118...	172.31.0.0/20	
<input type="checkbox"/>	us-east-1d	subnet-0fd9b1dd1...	172.31.16.0/20	

Enable database

Restore a snapshot - optional

Restore an existing snapshot from a previously used database.

Snapshot

None

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

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

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

Step 6
[Review](#)

Configure instance traffic and scaling - optional

Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

(Container default)

Size

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

8 GB

IOPS

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

100 IOPS

Throughput

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

125 MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

5 minute

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 Instances

Monitoring interval

5 minute

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more [\[?\]](#)

IMDSv1

With the current setting, the environment enables only IMDSv2.

Deactivated

EC2 security groups

Select security groups to control traffic.

EC2 security groups (4)			
<input type="text"/> Filter security groups			
	Group name	▲	Group ID
<input checked="" type="checkbox"/>	default		sg-009f9ea9079c28fd9
<input type="checkbox"/>	launch-wizard-1		sg-094cf5e04d89a25f
<input type="checkbox"/>	launch-wizard-2		sg-0c0fbdd01d426cf3f
<input type="checkbox"/>	launch-wizard-3		sg-04536e9fd071d79ea

▼ Capacity [Info](#)

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type

Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. Learn more [\[?\]](#)

Single instance

Instances

1 Min

1 Max

Fleet composition

Spot Instances are launched at the lowest available price. Learn more [\[?\]](#)

On-Demand instance

Spot instance

Maximum spot price

The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.

Default

Set your maximum price

On-Demand base

The minimum number of On-Demand Instances that your Auto Scaling group provisions before considering Spot Instances as your environment scales out.

0

The minimum number of On-Demand instances that your Auto Scaling group provisions before considering Spot instances as your environment scales out.

0

On-Demand above base

The percentage of On-Demand Instances as part of any additional capacity that your Auto Scaling group provisions beyond the On-Demand base instances.

0

%

Capacity rebalancing

Specifies whether to enable the capacity rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when `EnableSpot` is true in the `aws:ec2:instances` namespace, and there is at least one Spot Instance in your Auto Scaling group.

Turn on capacity rebalancing

Architecture

The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. [Learn more](#)

x86_64

This architecture uses x86 processors and is compatible with most third-party tools and libraries.

arm64 - new

This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

Instance types

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

Choose x86 instance types ▾

t3.micro X t3.small X

AMI ID

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

ami-05218d60cd8f94600

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

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

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

Step 6
Review

Configure updates, monitoring, and logging - *optional* Info

▼ Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The `EnvironmentHealth` custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

System

Basic

Enhanced

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

Keep logs after terminating environment

▼ Managed platform updates Info

9) Uncheck Managed Updates, Activated.

▼ **Managed platform updates** [Info](#)

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates Activated

Weekly update window
Thursday at 09 : 17 UTC

Update level
Minor and patch

Instance replacement
If enabled, an instance replacement will be scheduled if no other updates are available.
 Activated

▼ **Email notifications** [Info](#)

Enter an email address to receive email notifications for important events from your environment. [Learn more](#) 

Email

▼ **Rolling updates and deployments** [Info](#)

▼ Rolling updates and deployments Info

Application deployments

Choose how Amazon Elastic Beanstalk propagates source code changes and software configuration updates. [Learn more](#) 

Deployment policy

All at once



Batch size type

Percentage

Fixed

Deployment batch size

100

% instances at a time

Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime. [Learn more](#) 

Rolling update type

Deactivated



Deployment preferences

Customize health check requirements and deployment timeouts.

Ignore health check

Don't fail deployments due to health check failures.

▼ Platform software [Info](#)

Configure the options available to your specific platform. These include the proxy server and OS environment properties. [Learn more](#)



Container options

Proxy server

Nginx▼

Amazon X-Ray

Amazon X-Ray is a service that collects data about the requests and responses that your application serves and receives. You can use the tools that X-Ray offers to view and filter the data that it provides to identify potential issues and optimization opportunities.

X-Ray daemon

(service charges may apply.)

Activated

S3 log storage

Configure the instances in your environment to upload rotated logs to Amazon S3. [Learn more](#)

Rotate logs

(standard S3 charges apply.)

Activated

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. [Learn more](#)

Log streaming

CloudWatch Metrics streams required

Configure the instances in your environment to upload rotated logs to Amazon S3. [Learn more](#)

Rotate logs
(standard S3 charges apply.)
 Activated

Instance log streaming to CloudWatch logs
Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. [Learn more](#)

Log streaming
(standard CloudWatch charges apply.)
 Activated

Retention
7

Lifecycle
Keep logs after terminating envir... ▾

Environment properties
The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value	Remove
PYTHONPATH	/var/app/venv/staging-LQM1lest/bin	<input type="button" value="Remove"/>

10) Environment is successfully launched.

The screenshot shows the AWS Elastic Beanstalk console with the environment 'RohanWebApp-env-1' successfully launched. The interface includes a sidebar with navigation links like Applications, Environments, Change history, Application versions, Saved configurations, Environment (selected), Go to environment, Configuration, Events, Health, Logs, Monitoring, Alarms, Managed updates, Tags, and Recent environments (RohanWebApp-env-1). The main content area displays the environment overview, platform details (Python 3.11 running on 64bit Amazon Linux 2023/4.1.3), and a list of events. The events table shows three entries from August 25, 2024, at 00:10:33 UTC+5:30, all of which are INFO level and indicate successful launch and instance addition.

Time	Type	Details
August 25, 2024 00:10:33 (UTC+5:30)	INFO	Successfully launched environment: RohanWebApp-env-1
August 25, 2024 00:10:32 (UTC+5:30)	INFO	Application available at RohanWebApp-env-1.eba-uc4er9m.us-east-1.elasticbeanstalk.com.
August 25, 2024 00:10:24 (UTC+5:30)	INFO	Adding instance i-07c34c41921052b9b' to your environment.

Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

Code Deploy and Code Pipeline:

- 1) Search for Codepipeline in the search box and click on create pipeline.

The screenshot shows the AWS CodePipeline console. On the left, there is a navigation sidebar titled "Developer Tools" with a section for "CodePipeline". Under "CodePipeline", there are several items: "Source • CodeCommit", "Artifacts • CodeArtifact", "Build • CodeBuild", "Deploy • CodeDeploy", "Pipeline • CodePipeline" (which is expanded to show "Getting started" and "Pipelines"), and "Settings". Below the sidebar, there are links for "Go to resource" and "Feedback". The main content area has a breadcrumb navigation: "Developer Tools > CodePipeline > Pipelines". A banner at the top says "Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more". Below the banner, there is a table header for "Pipelines" with columns: Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A search bar is located above the table. The table body displays the message "No results" and "There are no results to display." At the bottom right of the table area, there is a "Create pipeline" button.

- 2) Configuring the pipeline.

The screenshot shows the "Choose pipeline settings" step of the AWS CodePipeline 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 content area has a title "Choose pipeline settings" with a "Step 1 of 5" indicator. It contains a "Pipeline settings" section. Under "Pipeline name", there is a text input field containing "Rohan_Pipeline" with the placeholder "Enter the pipeline name. You cannot edit the pipeline name after it is created." and the note "No more than 100 characters". Under "Pipeline type", there is a note: "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." Below this, there is an "Execution mode" section with three options: "Superseded" (radio button is empty), "Queued (Pipeline type V2 required)" (radio button is checked), and "Parallel (Pipeline type V2 required)" (radio button is empty). The "Queued" option has a note: "Executions are processed one by one in the order that they are queued." The "Parallel" option has a note: "Executions don't wait for other runs to complete before starting or finishing." At the bottom, there is a "Service role" section which is currently empty.

S | Services | Search | [Alt+S] | ☰ | 🔍 | ? | 🌐

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name: AWSCodePipelineServiceRole-us-east-1-Rohan_Pipeline

Type your service role name

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

Variables

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

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

You can add up to 50 variables.

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

Advanced settings

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add source stage Info

Step 2 of 5

Source

Source provider

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

GitHub (Version 1)

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

Connected

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

ⓘ **The GitHub (Version 1) action is not recommended**

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository

Repository

Rohan-Lalchandani08/OnlinePrintingServices

Branch

main

Change detection options

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

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

AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

3) Skip the build stage.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage **Add build stage** Info

Step 4 Add deploy stage

Step 5 Review

Add build stage *optional*

Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

4) Choose Elastic Beanstalk as the Deploy Provider.

Step 2
[Add source stage](#)

Step 3
[Add build stage](#)

Step 4
[Add deploy stage](#)

Step 5
[Review](#)

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

Deploy

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

AWS Elastic Beanstalk ▾

Region
US East (N. Virginia) ▾

Input artifacts
Choose an input artifact for this action. [Learn more](#) ↗
No more than 100 characters ▾

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

Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
RohanWebApp-env-1 X

Configure automatic rollback on stage failure

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Review Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
Rohan_Pipeline

Pipeline type
V2

Execution mode
QUEUED

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-us-east-1-Rohan_Pipeline

Variables

Name	Default value	Description
No variables		

Step 2: Add source stage

Source action provider

Source action provider

GitHub (Version 1)

PollForSourceChanges

false

Repo

OnlinePrintingServices

Owner

Rohan-Lalchandani08

Branch

main

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

RohanWebApp

EnvironmentName

RohanWebApp-env-1

Configure automatic rollback on stage failure

Disabled

Cancel

Previous

Create pipeline

5) Pipeline has been created.

Screenshot of the AWS CodePipeline console showing a newly created pipeline named "Rohan_Pipeline".

The pipeline type is V2 and the execution mode is QUEUED. The pipeline consists of two stages: Source and Deploy.

- Source Stage:** Succeeded. Pipeline execution ID: bfb65899-6728-4fc0-9971-9639480a06da.
 - GitHub (Version 1) - Succeeded (e89afae2) - 3 minutes ago. View details.
- Deploy Stage:** Succeeded. Pipeline execution ID: bfb65899-6728-4fc0-9971-9639480a06da. Start rollback.

Buttons available: Notify, Edit, Stop execution, Clone pipeline, Release change.

Screenshot of a web browser displaying a website for "Print It - Online Printing Services for everyone".

The page features a logo of a printer and the text "PRINTING COMPANY".

Navigation links include "About Us" and "Contact Us".

A sidebar menu lists:

- Introduction Audio
- Promotional Video
- Company Details
- Services We Offer

The main content area displays the text "Your one-stop solution for all printing needs." followed by a section titled "Introduction Audio" with a play button showing 0:00 / 0:01.

Below this is a section titled "Promotional Video" featuring a video player thumbnail showing a business meeting. The video description reads: "Successful business meeting of people discussing stock...". Buttons for "Watch later" and "Share" are visible.

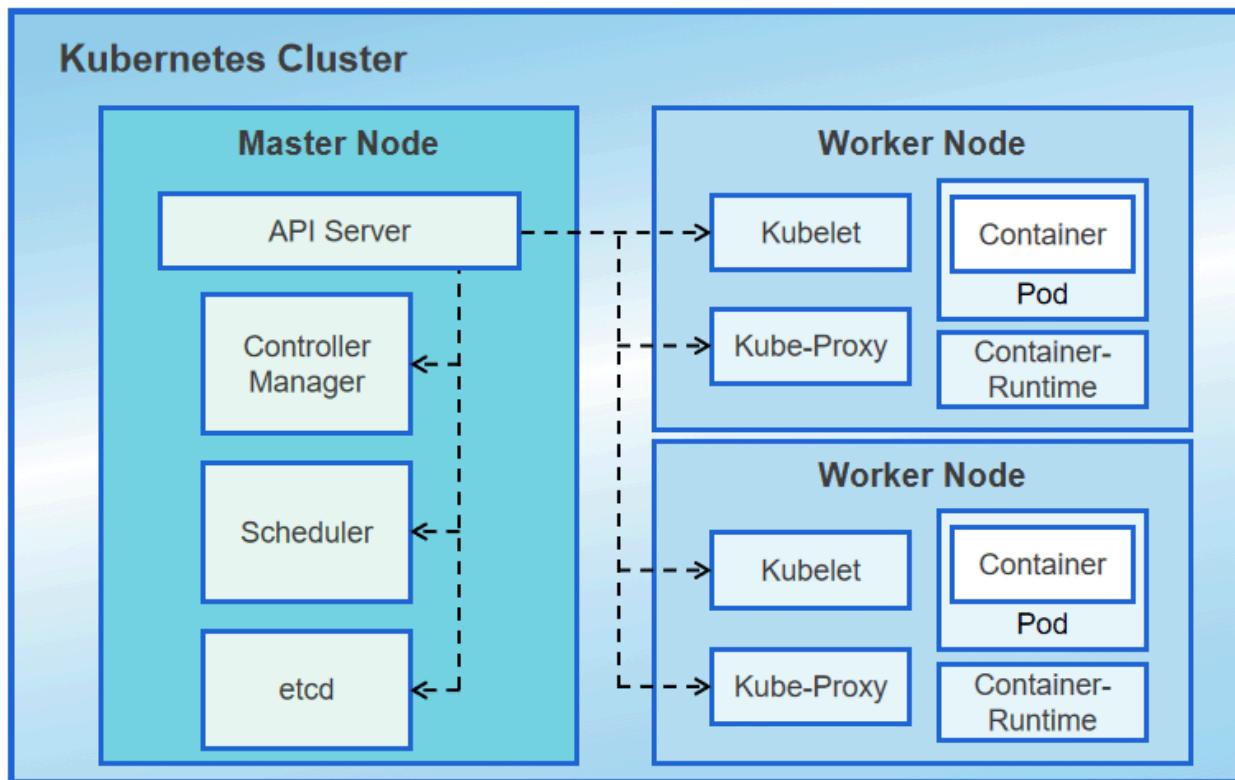
Adv. Devops Experiment no. 3

Name: Rohan Lalchandani
Class: D15A Roll no.: 25

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

Theory:

Kubernetes cluster architecture is designed to manage containerized applications efficiently. It consists of two main components: the **Control Plane** and the **Worker Nodes**.



1. Control Plane

The control plane is responsible for managing the overall state of the cluster. It makes global decisions about the cluster, such as scheduling applications, and detects/responds to cluster events (like a container failure).

- **API Server:** The entry point for all administrative tasks in the cluster. It exposes the Kubernetes API, acting as the front end of the control plane.
- **etcd:** A distributed key-value store that stores all the cluster's data. It is a highly reliable store for all cluster state and configuration.
- **Controller Manager:** Monitors the cluster state and performs routine tasks like handling node failures, maintaining the correct number of replicas for pods, and balancing the load across the cluster.
- **Scheduler:** Assigns newly created pods to nodes based on resource availability, affinity rules, and other policies.

2. Worker Nodes

Worker nodes run the containerized applications (pods). Each node has a set of components that communicate with the control plane to receive and execute tasks.

- **Kubelet:** The agent that runs on every node in the cluster. It ensures that containers are running in pods and communicates with the API server to get the desired state of the node.
- **Container Runtime:** The software responsible for running containers, such as Docker, containerd, or CRI-O. It manages the lifecycle of containers on the node.
- **Kube Proxy:** Ensures networking rules are applied to allow communication between the different components in the cluster, handling services and load balancing.

3. Cluster Networking

Kubernetes provides an abstracted networking layer for communication between pods. Key features include:

- **Pod-to-Pod Communication:** Every pod in a cluster can communicate with every other pod without using NAT (Network Address Translation).

- **Service Abstraction:** Services provide a stable endpoint to access a group of pods. This decouples the frontend from backend pods, offering load balancing and failure recovery.

4. Storage

Kubernetes uses persistent storage systems like persistent volumes (PVs) and persistent volume claims (PVCs) to manage data storage needs.

Implementation:

1. Create three EC2 Ubuntu Instances - Master, Worker 1 and Worker 2.

The screenshot shows the AWS CloudFormation console interface. A new stack named "K8s-Stack" is being created. The "Template" tab is selected, displaying the CloudFormation template code. The "Outputs" tab shows the outputs for the stack, including "Master Public IP" and "Worker 1 Public IP". The "Resources" tab lists the resources created by the stack, such as "AWS::CloudWatchLogs::LogGroup" and "AWS::Lambda::Function".

2. Select the following AMI image.

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture: 64-bit (x86)

AMI ID: ami-0e86e20dae9224db8

Username: ubuntu

Verified provider

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...read more
ami-0e86e20dae9224db8

Virtual server type (instance type): t2.medium

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

3. Select t2.medium in instance type.

Why?

t2.medium has 2 vCPUs and 4 GB of RAM, which allows it to run more pods and handle larger or more resource-intensive containers compared to **t2.micro** (1 vCPU, 1 GB of RAM).

Instance type: t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Compare instance types

Key pair (login): aws_ubuntu

Create new key pair

Network settings

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...read more
ami-0e86e20dae9224db8

Virtual server type (instance type): t2.medium

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

4. Create security group

In it type the group name and description as my-master-sg

VPC - required | Info
vpc-0c73ad858cebf5faa (default)
172.31.0.0/16

Subnet | Info
No preference

Auto-assign public IP | Info
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required
my-master-sg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#@[]+=&;!\$*

Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...read more
ami-0e86e20dae9224db8

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Review commands

5. Edit the inbound rules and add a new rule to accept “All traffic” as shown below.

Description - required | Info
my-master-sg

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info
ssh

Protocol | Info
TCP

Port range | Info
22

Remove

Source type | Info
Anywhere

Source | Info
Add CIDR, prefix list or security

Description - optional | Info
e.g. SSH for admin desktop

0.0.0.0/0 X

▼ Security group rule 2 (All, All, 0.0.0.0/0)

Type | Info
All traffic

Protocol | Info
All

Port range | Info
All

Remove

Source type | Info
Custom

Source | Info
Add CIDR, prefix list or security

Description - optional | Info
e.g. SSH for admin desktop

0.0.0.0/0 X

Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...read more
ami-0e86e20dae9224db8

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which you launch this instance

Cancel Launch instance Review commands

6. Create Worker 1 and Worker 2 also with same settings and select the existing security group of “my-master-sg” in them. Connect all the instances.

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area displays a table titled "Instances (3) info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The instances listed are: Worker 1 (i-04a3cf028e1290254, Running, t2.medium, Initializing, View alarms, us-east-1c, ec2-35-...), Worker 2 (i-0ee6249a6f40632c6, Running, t2.medium, Initializing, View alarms, us-east-1c, ec2-35-...), and Master (i-093cde3353624675c, Running, t2.medium, 2/2 checks passed, View alarms, us-east-1c, ec2-34-...). Below the table is a modal window titled "Select an instance". The bottom navigation bar includes CloudShell, Feedback, and links to © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

7. Now run “sudo su” to execute commands as root user.
“cd” to change directory
then “apt-get update”

The screenshot shows a terminal session on the Master instance (i-093cde3353624675c). The session starts with a message about Ubuntu being free software. It then shows the user running "sudo su" to become root. The user then runs "cd" to change the directory. Finally, the user runs "apt-get update". The terminal output is as follows:

```
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-172-31-91-125:~$ sudo su
root@ip-172-31-91-125:/home/ubuntu# cd
root@ip-172-31-91-125:~# apt-get updates
E: Invalid operation updates
root@ip-172-31-91-125:~# 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 [377 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 [81.4 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4516 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [269 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [113 kB]

i-093cde3353624675c (Master)
PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125
```

8. Install docker by “apt-get install docker.io -y”

```
aws Services Search [Alt+S] N. Virginia vocabs/user3402855=LALCHANDANI_ROHAN_ANIL@6564-0401-7537 ▾
root@ip-172-31-91-125:~# apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 139 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willsync1 [4450 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (81.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
i-093cde3353624675c (Master)
```

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

```
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
aws Services Search [Alt+S] N. Virginia vocabs/user3402855=LALCHANDANI_ROHAN_ANIL@6564-0401-7537 ▾
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up dns-root-data (2023112702-willsync1) ...
Setting up bridge-utils (1.7.1-1ubuntu2) ...
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu4.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

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

9. enable, start and check status of docker if it's active.

The screenshot shows the AWS CloudShell interface. At the top, there are tabs for 'aws' and 'Services'. A search bar contains 'Search' and a keybinding '[Alt+S]'. The main area displays terminal output for Docker configuration:

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-91-125:~# service docker restart
root@ip-172-31-91-125:~# systemctl enable docker
root@ip-172-31-91-125:~# systemctl start docker
root@ip-172-31-91-125:~# systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sun 2024-09-22 13:59:43 UTC; 1min 5s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 2479 (dockerd)
   Tasks: 9
  Memory: 24.6M (peak: 24.8M)
    CPU: 222ms
   CGroup: /system.slice/docker.service
           └─2479 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 22 13:59:42 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:42.921053877Z" level=info msg="Starting up"
Sep 22 13:59:42 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:42.922042636Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, using /etc/resolv.conf"
Sep 22 13:59:42 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:42.974054473Z" level=info msg="[graphdriver] using prior storage driver: overlay2"
Sep 22 13:59:42 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:42.974206157Z" level=info msg="Loading containers: start."
Sep 22 13:59:43 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:43.138929302Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.1/16"
Sep 22 13:59:43 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:43.193826483Z" level=info msg="Loading containers: done."
Sep 22 13:59:43 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:43.213251021Z" level=info msg="Docker daemon" commit="24.0.7-0ubuntu4.1" graphdriver="overlay2"
Sep 22 13:59:43 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:43.213311717Z" level=info msg="Daemon has completed initialization"
Sep 22 13:59:43 ip-172-31-91-125 dockerd[2479]: time="2024-09-22T13:59:43.241077478Z" level=info msg="API listen on /run/docker.sock"
Sep 22 13:59:43 ip-172-31-91-125 systemd[1]: Started docker.service - Docker Application Container Engine.
```

Below the terminal, the instance details are shown:

i-093cde3353624675c (Master)
PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

At the bottom right, there are links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

10. Now we have to install kubernetes, for that we will refer the documentation at <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

Got to the “Debian-based distributions” and follow the instructions below to install kubeadm

```
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg
--dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
systemctl restart kubelet
```

aws Services Search [Alt+S] N. Virginia v vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537

```
root@ip-172-31-91-125:~# sudo apt-get install -y apt-transport-https ca-certificates curl gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
gpg is already the newest version (2.4.4-2ubuntu17).
gpg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu10.4 [341 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2ubuntu10.4 [333 kB]
Fetched 904 kB in 0s (24.7 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Preparing to unpack .../curl_8.5.0-2ubuntu10.4_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.4) over (8.5.0-2ubuntu10.1) ...
Preparing to unpack .../libcurl3t64_8.5.0-2ubuntu10.4_amd64.deb ...

```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

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

aws Services Search [Alt+S] N. Virginia v vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537

```
Preparing to unpack .../libcurl4t64_8.5.0-2ubuntu10.4_amd64.deb ...
Unpacking libcurl4t64:amd64 (8.5.0-2ubuntu10.4) over (8.5.0-2ubuntu10.1) ...
Preparing to unpack .../libcurl3t64-gnutls_8.5.0-2ubuntu10.4_amd64.deb ...
Unpacking libcurl3t64-gnutls:amd64 (8.5.0-2ubuntu10.4) over (8.5.0-2ubuntu10.1) ...
Setting up apt-transport-https (2.7.14build2) ...
Setting up libcurl4t64:amd64 (8.5.0-2ubuntu10.4) ...
Setting up libcurl3t64-gnutls:amd64 (8.5.0-2ubuntu10.4) ...
Setting up curl (8.5.0-2ubuntu10.4) ...
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-172-31-91-125:~# sudo mkdir -p /etc/apt/keyrings
root@ip-172-31-91-125:~# curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
root@ip-172-31-91-125:~# echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

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

```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb /  
root@ip-172-31-91-125:~# 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]  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]  
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 132 kB in 0s (288 kB/s)  
Reading package lists... Done  
root@ip-172-31-91-125:~# sudo apt-get install -y kubelet kubeadm kubectl  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  conntrack cri-tools kubernetes-cni  
The following NEW packages will be installed:  
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni  
0 upgraded, 6 newly installed, 0 to remove and 136 not upgraded.  
Need to get 87.4 MB of archives.  
After this operation, 314 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]  
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]  
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]  
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]  
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]  
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

```
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
aws services Search [Alt+S] N. Virginia vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537  
Selecting previously unselected package conntrack.  
(Reading database ... 68112 files and directories currently installed.)  
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...  
Unpacking conntrack (1:1.4.8-1ubuntu1) ...  
Selecting previously unselected package cri-tools.  
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...  
Unpacking cri-tools (1.31.1-1.1) ...  
Selecting previously unselected package kubeadm.  
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...  
Unpacking kubeadm (1.31.1-1.1) ...  
Selecting previously unselected package kubectl.  
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...  
Unpacking kubectl (1.31.1-1.1) ...  
Selecting previously unselected package kubernetes-cni.  
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...  
Unpacking kubernetes-cni (1.5.1-1.1) ...  
Selecting previously unselected package kubelet.  
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...  
Unpacking kubelet (1.31.1-1.1) ...  
Setting up conntrack (1:1.4.8-1ubuntu1) ...  
Setting up kubeadm (1.31.1-1.1) ...  
Setting up cri-tools (1.31.1-1.1) ...  
Setting up kubernetes-cni (1.5.1-1.1) ...  
Setting up kubectl (1.31.1-1.1) ...  
Setting up kubelet (1.31.1-1.1) ...  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...
```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

```
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
aws services Search [Alt+S] N. Virginia vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-172-31-91-125:~# sudo apt-mark hold kubelet kubeadm kubectl  
kubelet set on hold.  
kubeadm set on hold.  
kubectl set on hold.  
root@ip-172-31-91-125:~# sudo systemctl enable --now kubelet
```

Now only in **Master Node**, execute the “kubeadm init” to initialize the cluster.

Your Kubernetes control-plane has initialized successfully!

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

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

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

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

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

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

```
kubeadm join 172.31.91.125:6443 --token ueue2q7.i3wahpw8072hh8ms \
--discovery-token-ca-cert-hash sha256:64b48d7aa08c0dafbeba2879e1ae9a55e5a58504c9b2dc0871c4af03910b494
root@ip-172-31-91-125:~# sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
```

i-093cde3353624675c (Master)

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

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

11. Now copy the 3 commands from the regular user and 1 command from root user and one by one execute them.

Execute the same set of steps from 1 to 11 in Worker 1 and Worker 2 except the kubeadm init command.

Processing triggers for man-db (2.12.0-4build2) ...

Scanning processes...

Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
root@ip-172-31-91-125:~# mkdir -p $HOME/.kube
root@ip-172-31-91-125:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-91-125:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-91-125:~# export KUBECONFIG=/etc/kubernetes/admin.conf
root@ip-172-31-91-125:~# systemctl restart kubelet
```

12. Execute “kubectl get nodes” in master to get a list of nodes present. Initially there will be no nodes present.

```
aws | Services | Q Search [Alt+S] | N. Virginia | vclabs/user3402855=LALCHANDANI_ROHAN_ANIL@6564-0401-7537 ▾

Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-91-125:~# mkdir -p $HOME/.kube
root@ip-172-31-91-125:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-91-125:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-91-125:~# export KUBECONFIG=/etc/kubernetes/admin.conf
root@ip-172-31-91-125:~# systemctl restart kubelet
root@ip-172-31-91-125:~# kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-91-125   NotReady  control-plane  4m8s   v1.31.1
```

Copy the join command from Master and execute it in Worker 1 and Worker 2 to join them to the cluster.

```
aws | Services | Q Search [Alt+S] | N. Virginia | vclabs/user3402855=LALCHANDANI_ROHAN_ANIL@6564-0401-7537 ▾

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-172-31-84-210:~# kubeadm join 172.31.91.125:6443 --token uee2q7.i3wahpw807hh8ms \
--discovery-token-ca-cert-hash sha256:64b48d7aa808c0dafbea2879e1ae9a55e5a58504c9b2dc0871c4af03910b494
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 500.852828ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-84-210:~#
```

i-04a3cf028e1290254 (Worker 1)

PublicIPs: 35.174.111.130 PrivateIPs: 172.31.84.210

AWS Services Search [Alt+S] N. Virginia v vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537

```
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
kubelet set on hold.  
kubeadm set on hold.  
kubectl set on hold.  
root@ip-172-31-92-94:~#  
kubeadm join 172.31.91.125:6443 --token uee2q7.i3wahpw8072hh8ms \  
--discovery-token-ca-cert-hash sha256:64b48d7aa808c0dafbea2879e1ae9a55e5a58504c9b2dc0871c4af03910b494  
[preflight] Running pre-flight checks  
[preflight] Reading configuration from the cluster...  
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'  
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"  
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"  
[kubelet-start] Starting the kubelet  
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s  
[kubelet-check] The kubelet is healthy after 501.859381ms  
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap  
  
This node has joined the cluster:  
* Certificate signing request was sent to apiserver and a response was received.  
* The Kubelet was informed of the new secure connection details.  
  
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.  
root@ip-172-31-92-94:~#
```

i-0ee6249a6f40632c6 (Worker 2)
PublicIPs: 35.171.163.251 PrivateIPs: 172.31.92.94

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

Again execute kubectl get nodes to see if they have joined. You can see the status of nodes to be Not Ready, we have to make them Ready.

AWS Services Search [Alt+S] N. Virginia v vocabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537

```
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-172-31-91-125:~# mkdir -p $HOME/.kube  
root@ip-172-31-91-125:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
root@ip-172-31-91-125:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config  
root@ip-172-31-91-125:~# export KUBECONFIG=/etc/kubernetes/admin.conf  
root@ip-172-31-91-125:~# systemctl restart kubelet  
root@ip-172-31-91-125:~# kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ip-172-31-91-125 NotReady control-plane 4m8s v1.31.1  
root@ip-172-31-91-125:~# kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ip-172-31-84-210 NotReady <none> 21s v1.31.1  
ip-172-31-91-125 NotReady control-plane 20m v1.31.1  
ip-172-31-92-94 NotReady <none> 13s v1.31.1  
root@ip-172-31-91-125:~#
```

i-093cde3353624675c (Master)
PublicIPs: 34.205.203.24 PrivateIPs: 172.31.91.125

Now, we have to manifest “calico.yaml” file to make the status ready for that execute:

```
curl https://raw.githubusercontent.com/projectcalico/calico/v3.28.2/manifests/calico.yaml -O  
ls  
kubectl apply -f calico.yaml
```

```
aws | Services | Q Search [Alt+S] | N. Virginia | vclabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537 ▾  
root@ip-172-31-91-125:~# curl https://raw.githubusercontent.com/projectcalico/calico/v3.28.2/manifests/calico.yaml -O  
% Total % Received % Xferd Average Speed Time Time Current  
Dload Upload Total Spent Left Speed  
100 247k 100 247k 0 0 1372k 0 --:--:--:--:--:-- 1377k  
root@ip-172-31-91-125:~# ls  
Command 'ld' not found, but can be installed with:  
apt install binutils  
root@ip-172-31-91-125:~# ls  
calico.yaml snap  
root@ip-172-31-91-125:~# kubectl apply -f calico.yaml  
poddisruptionbudget.policy/calico-kube-controllers created  
serviceaccount/calico-kube-controllers created  
serviceaccount/calico-node created  
serviceaccount/calico-cni-plugin created  
configmap/calico-config created  
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created  
i-093cde3353624675c (Master)  
PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125
```

```
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
aws | Services | Q Search [Alt+S] | N. Virginia | vclabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537 ▾  
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/ippreservations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created  
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created  
clusterrole.rbac.authorization.k8s.io/calico-node created  
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created  
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created  
clusterrolebinding.rbac.authorization.k8s.io/calico-node created  
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created  
daemonset.apps/calico-node created  
deployment.apps/calico-kube-controllers created
```

Now the status becomes ready.

EXTRA: Renaming the role names:

kubectl label nodes <node ip address> node-role.kubernetes.io/<nodename>=<nodename>

Replace node ip address with your worker 1 or 2 ip address.

Type the name of the node in nodename.

The screenshot shows a terminal session in AWS CloudShell. The user runs 'kubectl get nodes' to view the current node configuration. Then, they run 'kubectl label nodes ip-172-31-84-210 node-role.kubernetes.io/worker1=Worker1' to label the first node. They repeat this for the second node. Finally, they run 'kubectl get nodes' again to verify that the roles have been updated correctly. The terminal also shows the AWS logo, search bar, and various status icons.

```
aws | Services | Q Search [Alt+S] | X | 🔍 | ⓘ | ⚙️ | N. Virginia | vodlabs/user3402855=LALCHANDANI_ROHAN_ANIL @ 6564-0401-7537 | G

root@ip-172-31-91-125:~# kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-172-31-84-210 Ready    <none>    29m   v1.31.1
ip-172-31-91-125 Ready    control-plane   49m   v1.31.1
ip-172-31-92-94 Ready    <none>    29m   v1.31.1
root@ip-172-31-91-125:~# kubectl label nodes ip-172-31-84-210 node-role.kubernetes.io/worker1=Worker1
node/ip-172-31-84-210 labeled
root@ip-172-31-91-125:~# kubectl label nodes ip-172-31-92-94 node-role.kubernetes.io/worker2=Worker2
node/ip-172-31-92-94 labeled
root@ip-172-31-91-125:~# kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-172-31-84-210 Ready    worker1   29m   v1.31.1
ip-172-31-91-125 Ready    control-plane   49m   v1.31.1
ip-172-31-92-94 Ready    worker2   29m   v1.31.1
root@ip-172-31-91-125:~#
```

i-093cde3353624675c (Master) X

PublicIPs: 34.203.203.24 PrivateIPs: 172.31.91.125

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

Adv. Devops Experiment no. 4

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

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

Theory:

Implementation:

1. Create an EC2 Ubuntu Instance - Master.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the instance is named 'Master'. In the 'Application and OS Images (Amazon Machine Image)' section, the AMI selected is 'Canonical, Ubuntu, 24.04, amd64'. The 'Virtual server type (instance type)' is set to 't2.medium'. Under 'Storage (volumes)', there is one volume of 8 GiB. The 'Summary' section shows 1 instance selected. At the bottom, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

2. Select the following AMI image.

The screenshot shows the AWS Quick Start interface. On the left, there's a grid of quick links for various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The Ubuntu link is highlighted with a blue border. Below this grid, there's a section for "Amazon Machine Image (AMI)" featuring the Ubuntu Server 24.04 LTS (HVM), SSD Volume Type. This section includes details like the AMI ID (ami-0e86e20dae9224db8), architecture (64-bit (x86)), and provider information (Verified provider). To the right, a "Summary" panel is open, showing settings for launching one instance. The "Virtual server type (instance type)" is set to t2.medium. At the bottom right of the summary panel is a prominent orange "Launch instance" button.

3. Select t2.medium in instance type.

Why?

t2.medium has 2 vCPUs and 4 GB of RAM, which allows it to run more pods and handle larger or more resource-intensive containers compared to **t2.micro** (1 vCPU, 1 GB of RAM).

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Compare instance types

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8

Virtual server type (instance type)

t2.medium

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Launch instance

4. Create security group

In it type the group name and description as my-master-sg

VPC - required

vpc-0c73ad858cebf5faa (default)
172.31.0.0/16

Subnet

No preference

Create new subnet

Auto-assign public IP

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required

my-master-sg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#,@[]+=&;!\$*

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...read more
ami-0e86e20dae9224db8

Virtual server type (instance type)

t2.medium

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Launch instance

5. Edit the inbound rules and add a new rule to accept “All traffic” as shown below.

The screenshot shows the AWS CloudFormation console. On the left, there's a sidebar with a search bar and a list of services. The main area is titled "Description - required" with the value "my-master-sg". Under "Inbound Security Group Rules", there are two entries:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: ssh, Protocol: TCP, Port range: 22. Source type: Anywhere, Source: 0.0.0.0/0. Description: e.g. SSH for admin desktop.
- Security group rule 2 (All, All, 0.0.0.0/0)**: Type: All traffic, Protocol: All, Port range: All. Source type: Custom, Source: 0.0.0.0/0. Description: e.g. SSH for admin desktop.

On the right, the "Summary" section shows 1 instance, using the Canonical, Ubuntu, 24.04 AMI, t2.medium instance type, and 1 volume(s) - 8 GiB storage. A tooltip indicates a free tier of 750 hours for t2.micro or t3.micro instances. At the bottom, there are "Cancel", "Launch instance", and "Review commands" buttons.

6. Now connect to the instance by copying the “ssh” command and executing in Git Bash.

The screenshot shows the "Connect to instance" dialog. It lists the instance ID "i-0b9acd9256582f063 (Master)". Below it are instructions for connecting via SSH:

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is `aws_ubuntu.pem`.
- Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "aws_ubuntu.pem"
- Connect to your instance using its Public DNS:
`ec2-52-87-160-202.compute-1.amazonaws.com`

Below these instructions is an example command:
`ssh -i "aws_ubuntu.pem" ubuntu@ec2-52-87-160-202.compute-1.amazonaws.com`

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

7. Install docker, kubernetes by steps did in 3rd experiment.

8. After that we have to create 2 files as shown below.

```
root@ip-172-31-86-153:~# nano nginx-deployment.yaml  
root@ip-172-31-86-153:~# nano nginx-service.yaml
```

nginx-deployment.yaml file :

The screenshot shows a terminal window with the title "nginx-deployment.yaml". The file content is a YAML configuration for a Kubernetes Deployment:

```
root@ip-172-31-86-153: ~  
GNU nano 7.2  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nginx-deployment  
  labels:  
    app: nginx  
spec:  
  replicas: 2 # Number of pod replicas  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
    spec:  
      containers:  
      - name: nginx  
        image: nginx:latest  
      ports:  
      - containerPort: 80
```

The terminal window includes a menu bar with "File", "Edit", "Search", "Help", and "View". At the bottom, there is a toolbar with various keyboard shortcut keys for file operations like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, and Copy.

nginx-service.yaml file :

```
root@ip-172-31-86-153:~# nano nginx-service.yaml
GNU nano 7.2                                         nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80 # Port on the service
    targetPort: 80 # Port on the container
  type: LoadBalancer # For cloud environments, or use ClusterIP for internal traffic only
[ Read 12 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^L Go To Line M-E Redo
M-A Set Mark M-6 Copy
```

Apply the files

```
root@ip-172-31-86-153:~# kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-service.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
root@ip-172-31-86-153:~# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2       2           2           12s
```

Check the status of the services and pods if they are running properly

```
root@ip-172-31-86-153:~# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-54b9c68f67-5f8vb   1/1     Running   0          23s
nginx-deployment-54b9c68f67-sbxr6   1/1     Running   0          23s
root@ip-172-31-86-153:~# kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1   <none>        443/TCP   30m
nginx-service   LoadBalancer  10.110.82.177  <pending>    80:30319/TCP 60s
```

Expose the port by this command

```
root@ip-172-31-86-153:~# kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
```

Check the status of the port

```
root@ip-172-31-86-153:~# kubectl get services
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1    <none>       443/TCP       111m
nginx          NodePort   10.101.242.191  <none>       80:30905/TCP  6m33s
nginx-service  LoadBalancer  10.110.82.177  <pending>    80:30319/TCP  81m
```

Now go to the browser and copy the “public dns” of the master instance created, paste it in the browser put a colon and then type the port number of the nginx service, in this case it is 30905

Example: <http://52.87.160.202:30905/>



Adv. Devops Experiment no. 5

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

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

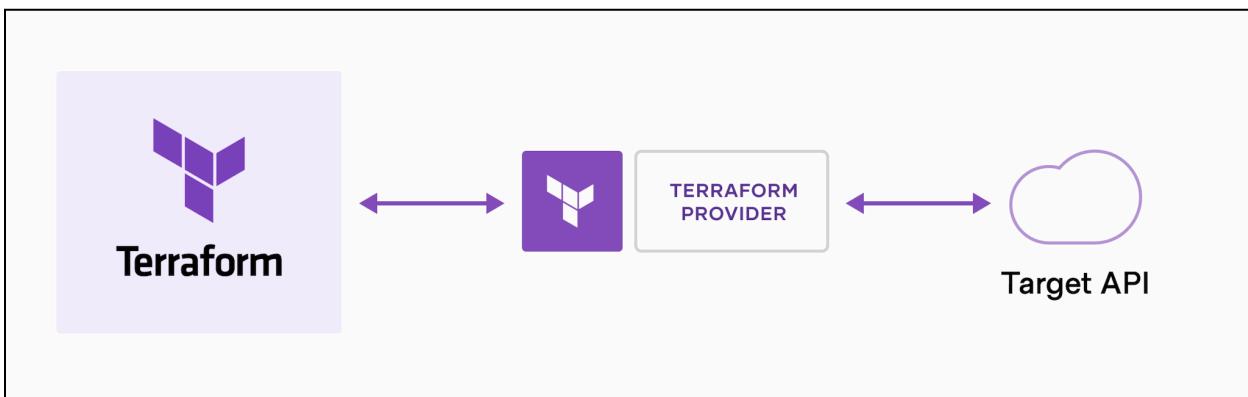
Theory:

Terraform:

HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructure throughout its lifecycle. Terraform can manage low-level components like compute, storage, and networking resources, as well as high-level components like DNS entries and SaaS features.

How does Terraform work?

Terraform creates and manages resources on cloud platforms and other services through their application programming interfaces (APIs). Providers enable Terraform to work with virtually any platform or service with an accessible API.



Key Features

1. Infrastructure as Code (IaC)

- **Declarative Configuration:** Users describe the desired final state of infrastructure, and Terraform determines the steps to achieve that state.
- **Version Control:** Infrastructure configurations can be versioned and treated similarly to application code, allowing for easy tracking of changes and collaboration.
- **Reusability:** Modules and configurations can be reused across different projects, promoting consistency and reducing duplication.

2. Multi-Cloud Support

- **Providers:** Terraform supports a wide range of cloud providers such as AWS, Azure, Google Cloud Platform, and many others, including on-premises solutions.
- **Abstraction:** It provides a consistent workflow across different providers, enabling users to manage heterogeneous environments seamlessly.
- **Portability:** Easy to migrate or replicate infrastructure across different cloud environments.

3. Resource Management

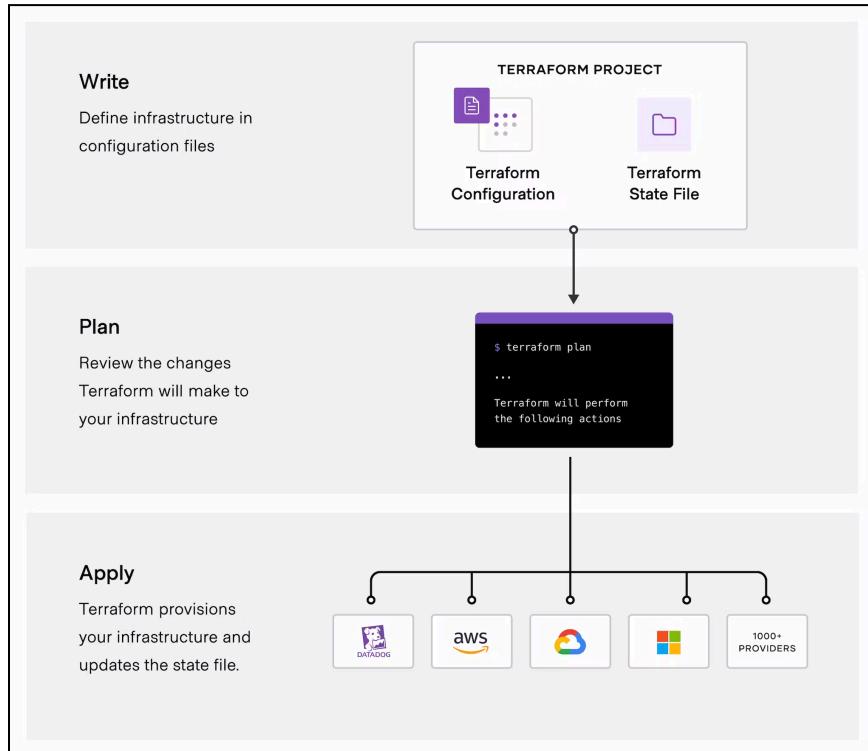
- **Dependency Graphing:** Terraform automatically understands and manages dependencies between resources, ensuring correct order of operations.
- **Plan and Apply:** The `terraform plan` command shows a preview of changes before they are applied, while `terraform apply` executes the changes, providing a safe and predictable workflow.

4. Extensibility

- **Modules:** Users can create and use modules to organize and encapsulate infrastructure configurations, promoting modularity and best practices.
- **Community and Ecosystem:** A vibrant community contributes modules, plugins, and support, enhancing Terraform's capabilities and usability.
- **Custom Plugins:** Ability to develop custom providers and plugins to extend functionality as needed.

Common Use Cases

- **Provisioning Cloud Infrastructure:** Setting up and managing cloud resources like virtual machines, networks, and storage.
- **Managing Multi-Cloud Environments:** Coordinating resources across different cloud platforms for redundancy and optimization.
- **Infrastructure Migration:** Facilitating the migration of resources from one environment to another through codified configurations.

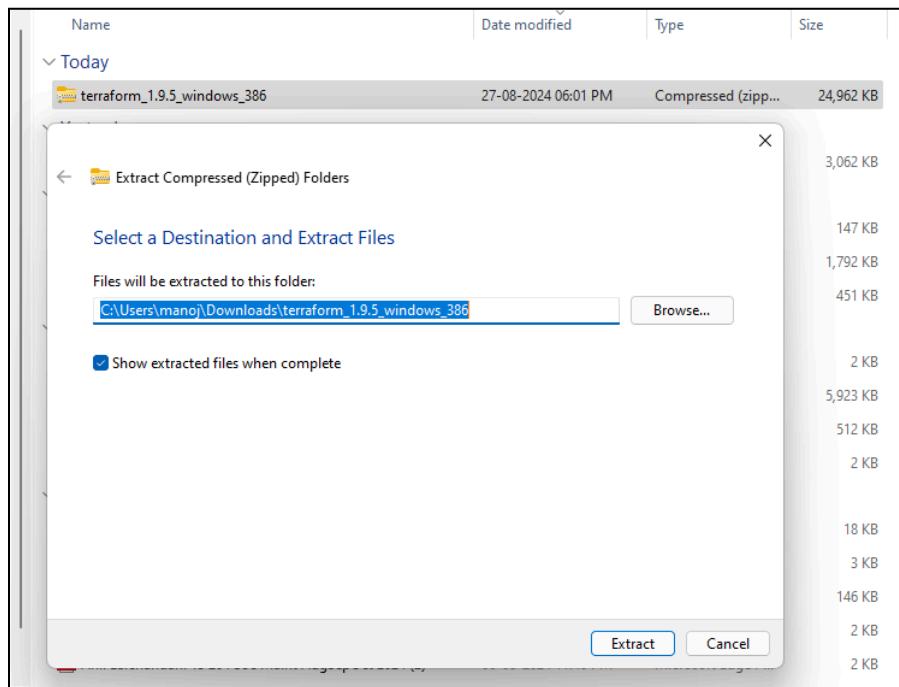


Implementation:

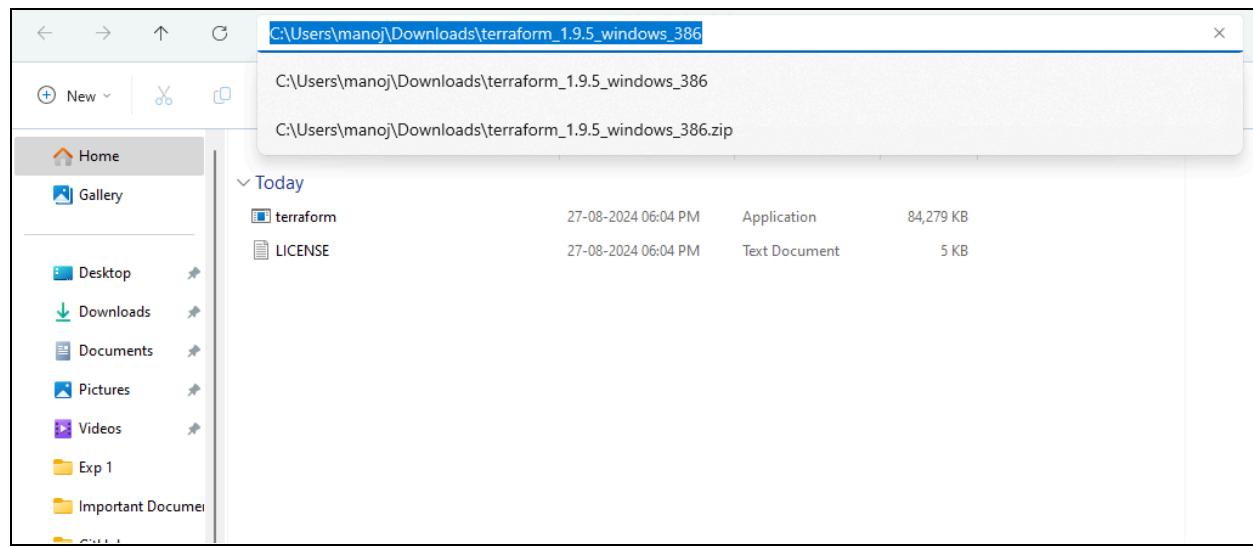
1) Go to website "<https://www.terraform.io/downloads.html>". Select your OS, in this case we are installing on Windows, download 386.

The screenshot shows the Terraform website's download section. On the left, a sidebar lists operating systems: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, and Solaris. The main content area has sections for "Binary download" (AMD64 and ARM64) and "Windows" (386 and AMD64). The "Windows" section is expanded, showing "Binary download" for "386" and "AMD64". Both have "Download" buttons. To the right, there's an "About Terraform" summary, "Featured docs" (Introduction to Terraform, Configuration Language, Terraform CLI, HCP Terraform, Provider Use), and a "HCP Terraform" section. At the bottom, there's a cookie consent banner with "Manage Preferences" and "ACCEPT" buttons.

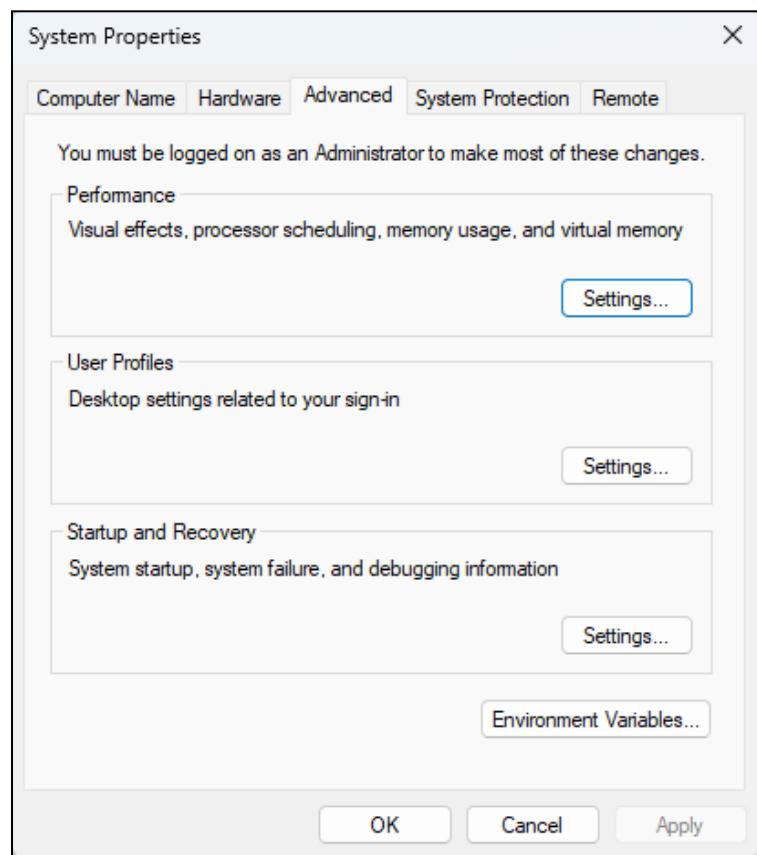
2) Extract the downloaded zip folder.



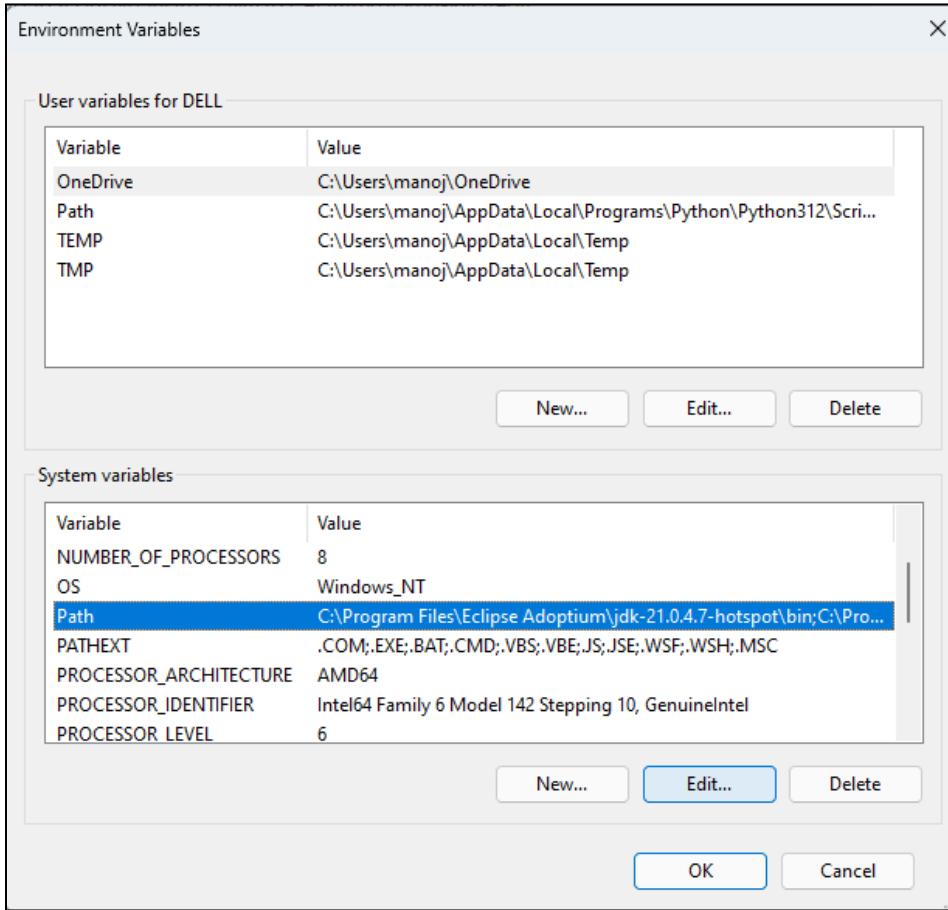
3) Go to the extracted folder and copy the path of the folder as shown below.



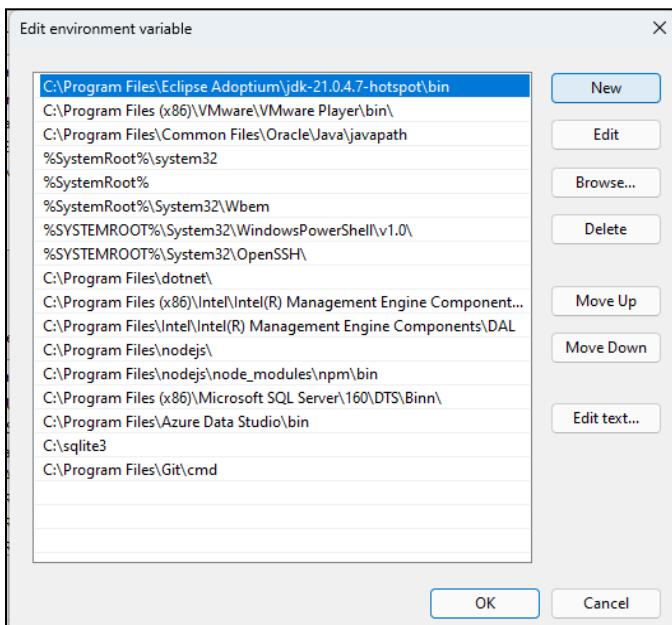
4) Search for “Environment variables” in the Start menu. In the dialog box click on environment variables.



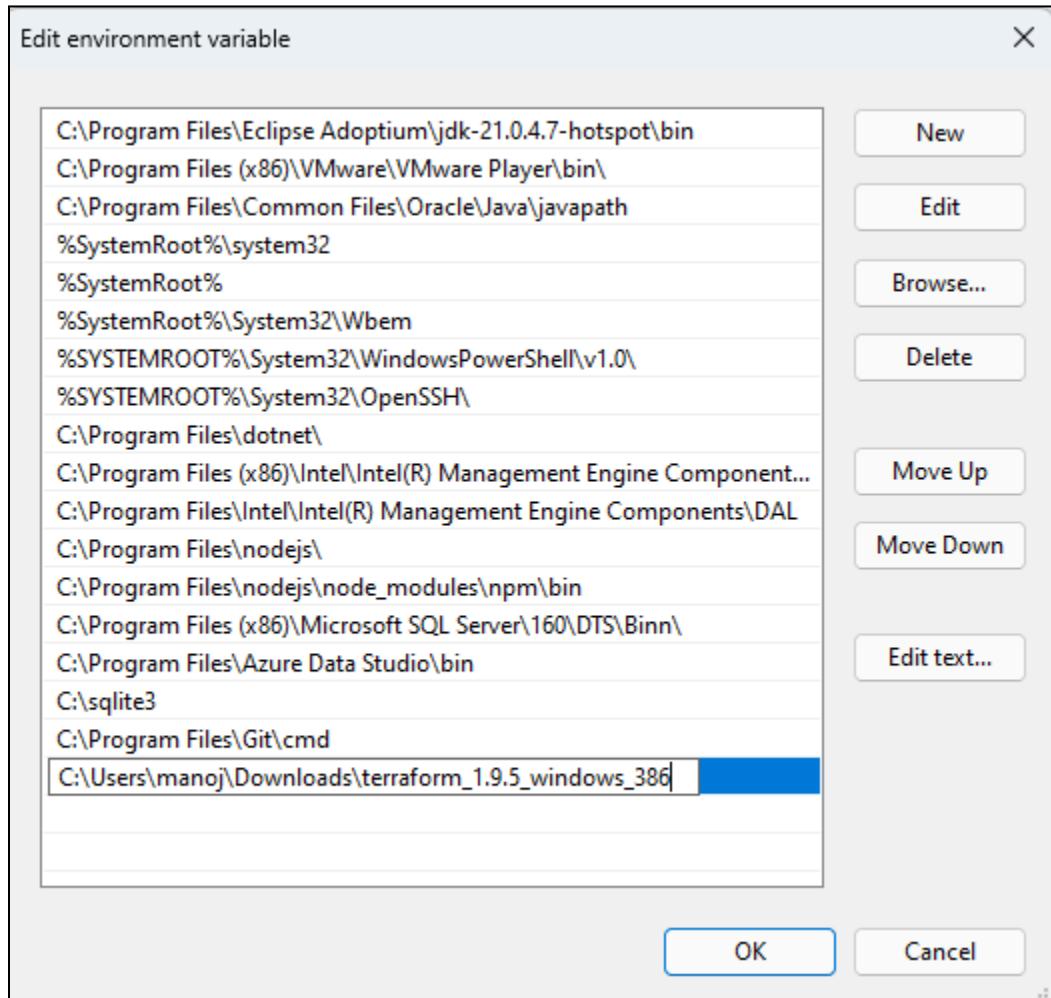
5) Click on path and click on edit as shown below.



6) Click on New.



7) Paste the path of the extracted folder and click on ok.



8) Open cmd and run the command “terraform --version” to check if it is successfully installed.

The screenshot shows a Microsoft Windows Command Prompt window. The title bar says 'Command Prompt'. The window displays the following text:

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

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

Adv. Devops Experiment no. 6

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

Aim: To Build, change, and destroy AWS infrastructure Using Terraform (S3 bucket or Docker).

Theory:

Terraform is an Infrastructure as Code (IaC) tool designed to manage and provision cloud and on-premises infrastructure through code. It offers a consistent and declarative way to describe and automate infrastructure deployments across various platforms.

IaC is a methodology where infrastructure is managed and provisioned using code, rather than manual processes.

1. Core Components of Terraform:

a) Providers:

Providers are plugins that Terraform uses to interact with various infrastructure services. Each provider is responsible for understanding API interactions with a specific service or platform (e.g., AWS, Azure, Google Cloud, Docker).

b) Resources

Resources are the fundamental units of infrastructure managed by Terraform. They represent components such as virtual machines, databases, or networking elements.

c) Modules

Modules are reusable containers of Terraform configuration that are used to create multiple instances of a resource or to encapsulate complex configurations. Modules can be created and shared to standardize and simplify deployments.

d) Variables

Variables allow you to parameterize your Terraform configurations. They help in customizing configurations without modifying the main configuration files directly.

e) Outputs

Outputs are used to extract information from Terraform configurations and make it available to other configurations or external systems.

2. Terraform Workflow

Terraform follows a specific workflow to manage infrastructure:

a) Write

You define your infrastructure requirements using Terraform configuration files (.tf files) written in HCL or JSON.

b) Plan

Terraform generates an execution plan to show what actions will be taken to reach the desired state defined in your configuration files. This step helps in previewing changes before applying them.

c) Apply

Terraform applies the changes required to reach the desired state of the configuration. This step creates, updates, or deletes infrastructure resources as necessary.

d) Destroy

Terraform removes all infrastructure defined in the configuration files. This step is useful for cleaning up resources when they are no longer needed.

3. State Management

Terraform maintains a state file (terraform.tfstate) that records the current state of your infrastructure. This state file is crucial for:

- **Tracking Resources:** It maps the resources defined in the configuration files to real-world infrastructure.
- **Planning Changes:** It helps Terraform determine what changes are needed by comparing the state file to the desired configuration.
- **Concurrency Control:** It ensures that multiple users or systems do not make conflicting changes to the infrastructure.

Implementation:

1. Download and install docker from <https://www.docker.com/>

For Windows Click on AMD64 version. Check if successfully installed in cmd using docker

docker --version

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

C:\Users\manoj>docker

Usage: docker [OPTIONS] COMMAND

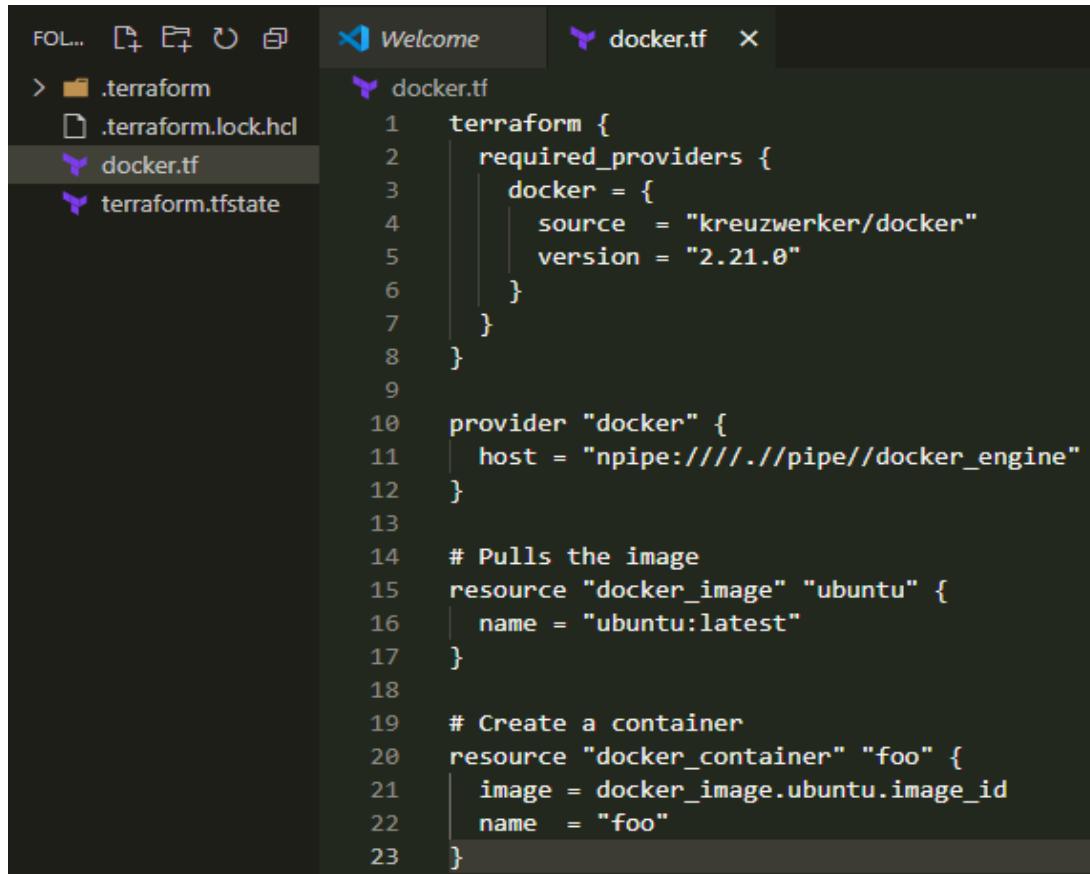
A self-sufficient runtime for containers

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
  container    Manage containers
  context      Manage contexts
  debug*       Get a shell into any image or container
```

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

2. Create a folder TerraformScripts and in it create a folder named docker. Open it in VS code and create a new file named docker.tf and write the following code.



```
FOL... ⌂ ⌂ ⌂ ⌂ ⌂
> .terraform
  .terraform.lock.hcl
  docker.tf
  terraform.tfstate

Welcome docker.tf

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

3. Open the terminal, make sure the path is set to the docker folder and execute
a) terraform init

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

Terraform has been successfully initialized!

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

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

b) terraform plan

```
PS D:\TerraformScripts\Docker> terraform plan
```

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

```
+ create
```

```
Terraform will perform the following actions:
```

```
# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = (known after apply)
  + container_logs = (known after apply)
  + entrypoint      = (known after apply)
  + env             = (known after apply)
  + exit_code       = (known after apply)
  + gateway         = (known after apply)
  + hostname        = (known after apply)
  + id              = (known after apply)
  + image           = (known after apply)
  + init            = (known after apply)
  + ip_address      = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode        = (known after apply)
  + log_driver      = (known after apply)
  + logs            = false
  + must_run        = true
  + name            = "foo"
  + network_data    = (known after apply)
  + read_only       = false
  + remove_volumes = true
  + restart         = "no"
  + rm              = false
  + runtime         = (known after apply)
  + security_opts   = (known after apply)
```

```
+ 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.
```

(Extra)

c) terraform validate

```
PS D:\TerraformScripts\Docker> terraform validate
Success! The configuration is valid.
```

d) terraform apply

```
PS D:\TerraformScripts\Docker> terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = [
    + "/bin/bash",
    + "-c",
    + "while true; do sleep 3600; done",
  ]
  + 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)
```

```

+ network_data      = (known after apply)
+ read_only         = false
+ remove_volumes   = true
+ restart           = "no"
+ rm                = false
+ runtime            = (known after apply)
+ security_opts     = (known after apply)
+ shm_size          = (known after apply)
+ start              = true
+ stdin_open         = false
+ stop_signal        = (known after apply)
+ stop_timeout       = (known after apply)
+ tty                = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

```

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

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

```

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Creation complete after 37s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=984860ac4d11d2b4bed665180e05f69408c3e0f24227c2b0386e77dc63568188]

```

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

4. Docker images before terraform apply command

```
PS D:\TerraformScripts\ Docker> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

5. Docker images after terraform apply command

```
PS D:\TerraformScripts\ Docker> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	edbfe74c41f8	5 weeks ago	78.1MB

6. terraform destroy

```
PS D:\TerraformScripts\ Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=984860ac4d11d2b4bed665180e05f69408c3e0f24227c2b0386e77dc63568188]

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        = [
        - "/bin/bash",
        - "-c",
        - "while true; do sleep 3600; done",
    ] -> 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       = "984860ac4d11" -> null
    - id             = "984860ac4d11d2b4bed665180e05f69408c3e0f24227c2b0386e77dc63568188" -> null
    - image          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - init           = false -> null
    - ip_address     = "172.17.0.2" -> null
    - ip_prefix_length = 16 -> null
    - ipc_mode       = "private" -> null
    - links          = [] -> null
    - log_driver     = "json-file" -> null
    - log_opts       = {} -> null
    - logs           = false -> null
}
```

```

- 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      = [
    {
        - gateway           = "172.17.0.1"
        - global_ipv6_prefix_length = 0
        - ip_address         = "172.17.0.2"
        - ip_prefix_length   = 16
        - network_name       = "bridge"
        # (2 unchanged attributes hidden)
    },
],
] -> null
- network_mode      = "bridge" -> null
- privileged         = false -> null
- publish_all_ports = false -> null
- read_only          = false -> null
- remove_volumes    = true -> null
- restart            = "no" -> null
- rm                 = false -> null
- runtime            = "runc" -> null
- security_opts     = [] -> null
- shm_size           = 64 -> null
- start              = true -> null
- stdin_open         = false -> null
- stop_timeout       = 0 -> null
- storage_opts       = {} -> null
- sysctls            = {} -> null
- tmpfs              = {} -> null
- tty                = false -> null
# (8 unchanged attributes hidden)
}

```

```

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

```

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

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```

docker_container.foo: Destroying... [id=984860ac4d11d2b4bed665180e05f69408c3e0f24227c2b0386e77dc63568188]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

```

Destroy complete! Resources: 2 destroyed.

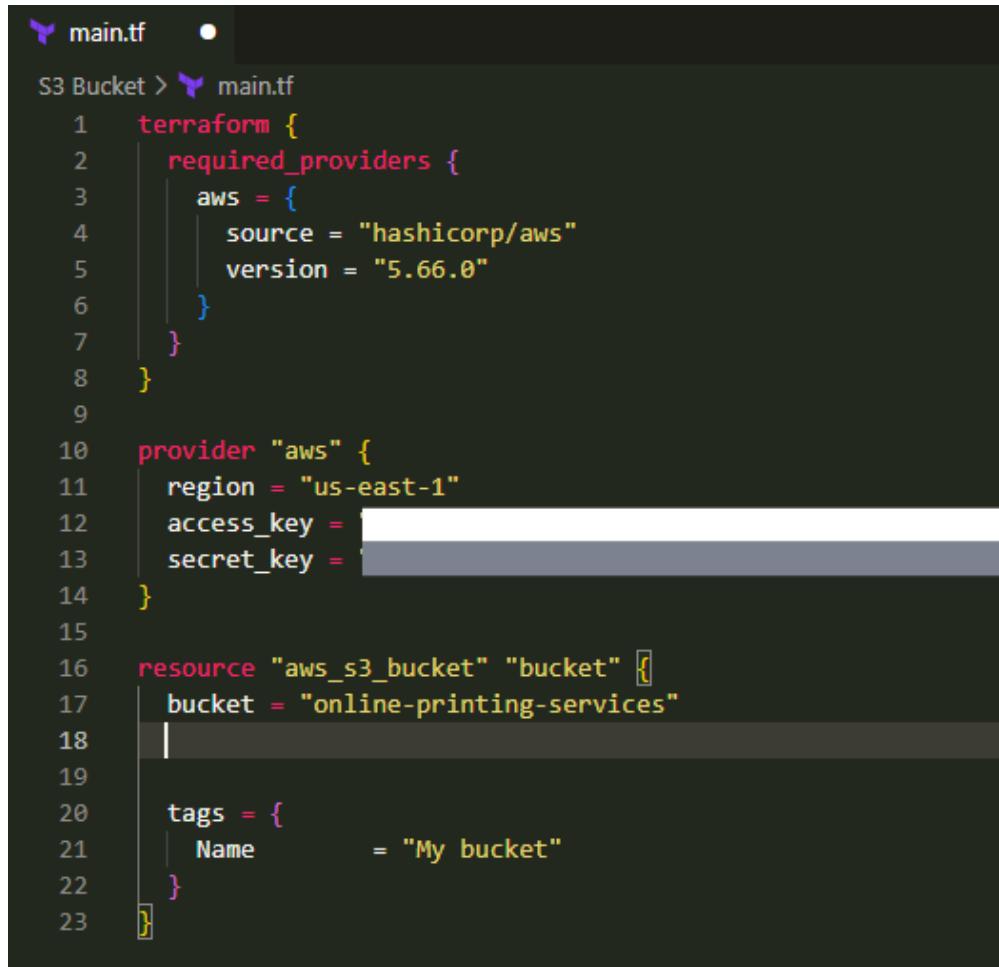
7. Docker images after terraform destroy

```
Destroy complete! Resources: 2 destroyed.  
PS D:\TerraformScripts\Docker> docker images  
REPOSITORY      TAG          IMAGE ID   CREATED    SIZE
```

S3 bucket using terraform:

1. Create another folder “**S3 Bucket**” in “**TerraformScripts**” folder open it in VS code and create a “main.tf file” and write the following code, taken from terraform documentation.

- a) For access_key and secret_key, Go to your “AWS Management Console” and go to security credentials option by click on you profile name at top right corner of console.
- b) Click on generate access key, copy and paste the key in the code.



```
main.tf
S3 Bucket > main.tf
1  terraform {
2    required_providers {
3      aws = {
4        source  = "hashicorp/aws"
5        version = "5.66.0"
6      }
7    }
8  }
9
10 provider "aws" {
11   region = "us-east-1"
12   access_key =
13   secret_key =
14 }
15
16 resource "aws_s3_bucket" "bucket" [
17   bucket = "online-printing-services"
18 ]
19
20   tags = {
21     Name      = "My bucket"
22   }
23 }
```

2) Open terminal, move to the “S3 Bucket” folder and execute terraform init.

```
PS D:\TerraformScripts> cd '.\S3 Bucket\'  
PS D:\TerraformScripts\S3 Bucket> terraform init  
Initializing the backend...  
Initializing provider plugins...  
- Finding hashicorp/aws versions matching "5.66.0"...  
- Installing hashicorp/aws v5.66.0...  
- Installed hashicorp/aws v5.66.0 (signed by HashiCorp)  
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.
```

2. Execute terraform validate to check if your code is correct or not.

```
PS D:\TerraformScripts\S3 Bucket> terraform validate  
Success! The configuration is valid.
```

3. terraform plan

```
PS D:\TerraformScripts\S3 Bucket> 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:

# aws_s3_bucket.bucket will be created
+ resource "aws_s3_bucket" "bucket" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "OnlinePrintingServices"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy              = false
    + hostedZoneId              = (known after apply)
    + id                        = (known after apply)
    + objectLockEnabled         = (known after apply)
    + policy                    = (known after apply)
    + region                    = (known after apply)
    + requestPayer               = (known after apply)
    + tags                      = {
        + "Name" = "My bucket"
    }
    + tags_all                  = {
        + "Name" = "My bucket"
    }
    + websiteDomain              = (known after apply)
    + websiteEndpoint            = (known after apply)

    + cors_rule (known after apply)

    + grant (known after apply)
}

+ grant (known after apply)

+ lifecycle_rule (known after apply)

+ logging (known after apply)

+ objectLockConfiguration (known after apply)

+ replicationConfiguration (known after apply)

+ serverSideEncryptionConfiguration (known after apply)

+ versioning (known after apply)

+ replicationConfiguration (known after apply)

+ serverSideEncryptionConfiguration (known after apply)

+ versioning (known after apply)

+ website (known after apply)
}

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

4. terraform apply

```
PS D:\TerraformScripts\S3 Bucket> 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:

# aws_s3_bucket.bucket will be created
+ resource "aws_s3_bucket" "bucket" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "online-printing-services"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy            = false
    + hosted_zone_id           = (known after apply)
    + id                       = (known after apply)
    + object_lock_enabled       = (known after apply)
    + policy                   = (known after apply)
    + region                   = (known after apply)
    + request_payer             = (known after apply)
    + tags                     = {
        + "Name" = "My bucket"
    }
    + tags_all                 = {
        + "Name" = "My bucket"
    }
    + website_domain           = (known after apply)
    + website_endpoint          = (known after apply)

    + cors_rule (known after apply)

    + grant (known after apply)
```

```
+ grant (known after apply)

+ lifecycle_rule (known after apply)

+ logging (known after apply)

+ object_lock_configuration (known after apply)

+ replication_configuration (known after apply)

+ server_side_encryption_configuration (known after apply)

+ versioning (known after apply)

+ website (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

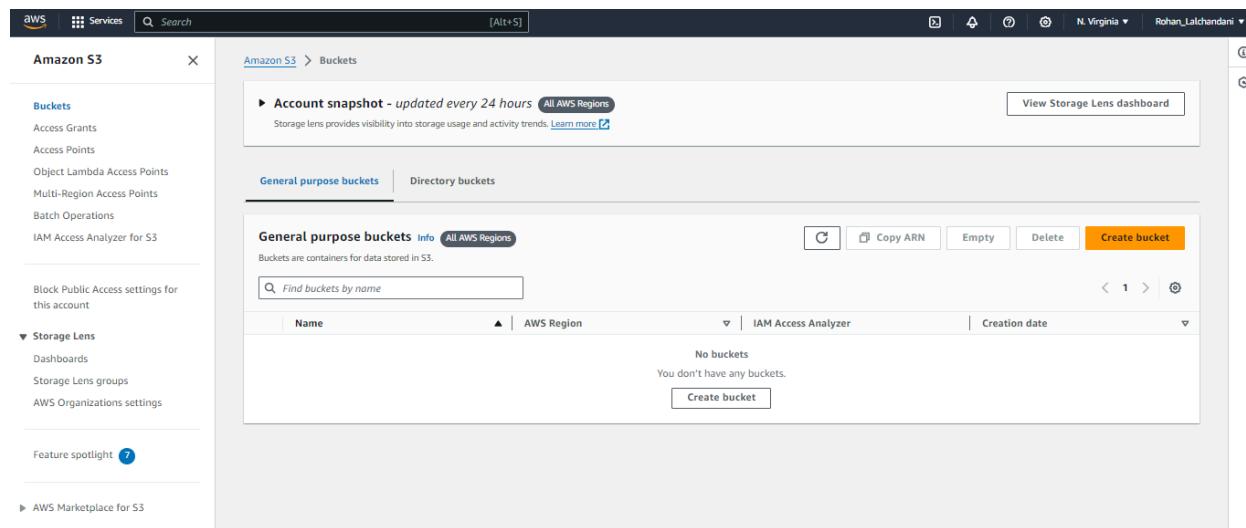
Enter a value: yes

```
aws_s3_bucket.bucket: Creating...
aws_s3_bucket.bucket: Creation complete after 7s [id=online-printing-services]
```

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

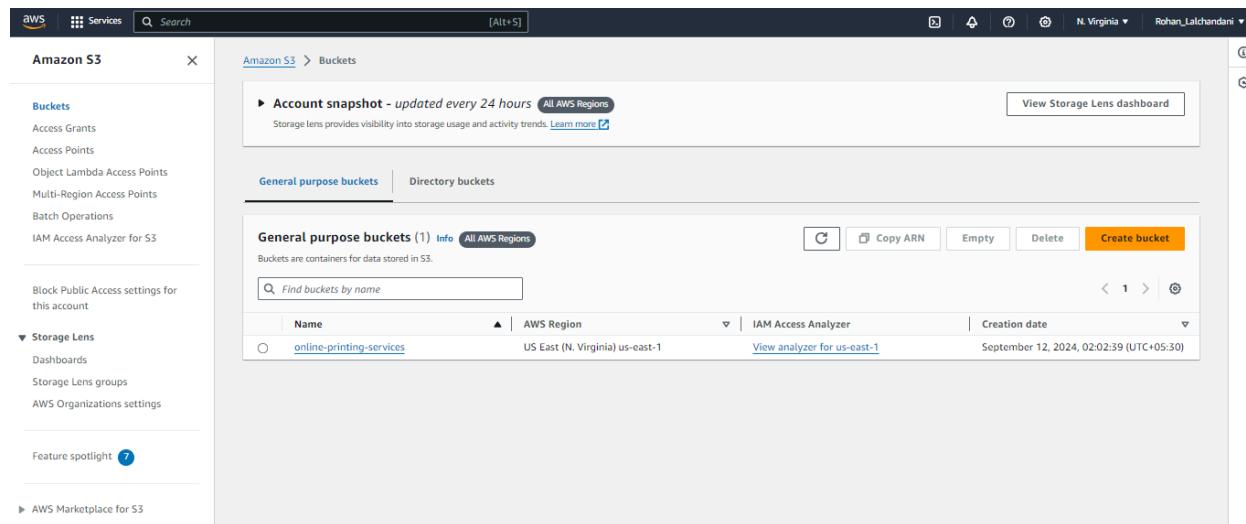
```
PS D:\TerraformScripts\S3 Bucket> []
```

5. Buckets in console before terraform apply.



The screenshot shows the AWS S3 service console. The left sidebar includes options like Buckets, Storage Lens, and Feature spotlight. The main content area displays an account snapshot and a table for General purpose buckets. The table has columns for Name, AWS Region, IAM Access Analyzer, and Creation date. It shows one entry: 'No buckets' with the message 'You don't have any buckets.' and a 'Create bucket' button.

6. Buckets in console after terraform apply.



The screenshot shows the AWS S3 service console after a Terraform apply. The left sidebar remains the same. The main content area displays an account snapshot and a table for General purpose buckets. The table now shows one entry: 'online-printing-services' located in 'US East (N. Virginia) us-east-1'. It also includes a link to 'View analyzer for us-east-1' and a creation date of 'September 12, 2024, 02:02:39 (UTC+05:30)'.

(Extra)

Hosting a website using s3 bucket in terraform

1. Create a new folder “S3 Website” under “TerraformScripts”.

Create four files “main.tf”, “providers.tf”, “variables.tf”, “output.tf” and write the following code.

main.tf :

This Terraform code creates and configures an AWS S3 bucket for hosting static website files with public access.

1. **S3 Bucket:** Creates an S3 bucket (`aws_s3_bucket.demo-bucket`) using a variable for the name.
2. **Ownership Controls:** Configures ownership of the objects in the bucket, making the bucket owner the preferred owner (`BucketOwnerPreferred`).
3. **Public Access Block:** Disables blocking of public access (`aws_s3_bucket_public_access_block`), allowing public access settings.
4. **Bucket ACL:** Sets the bucket's access control list (ACL) to allow public read access (`public-read`).
5. **Bucket Policy:** Adds a policy to the bucket allowing public read access (`s3:GetObject`) to all objects in the bucket.
6. **Template Files Module:** Fetches local files from a directory for hosting using the `hashicorp/dir/template` module.
7. **Website Configuration:** Configures the bucket to host a static website, specifying `index.html` as the main document.
8. **S3 Object:** Uploads files to the S3 bucket for website hosting, with each object having its key (file name) and content.

The screenshot shows a code editor interface with several tabs and files. On the left, a sidebar lists files: .terraform, webfiles (with index.html), main.tf (selected), output.tf, providers.tf, and variables.tf. The main editor area displays the contents of main.tf.

```
1 resource "aws_s3_bucket" "demo-bucket" {
2   bucket = var.my_bucket_name # Name of the S3 bucket
3 }
4
5
6 resource "aws_s3_bucketOwnershipControls" "example" {
7   bucket = aws_s3_bucket.demo-bucket.id
8   rule {
9     objectOwnership = "BucketOwnerPreferred"
10  }
11 }
12
13
14 resource "aws_s3_bucketPublicAccessBlock" "example" {
15   bucket = aws_s3_bucket.demo-bucket.id
16
17   blockPublicAcls      = false
18   blockPublicPolicy    = false
19   ignorePublicAcls    = false
20   restrictPublicBuckets = false
21 }
22
23
24 # AWS S3 bucket ACL resource
25 resource "aws_s3_bucketAcl" "example" {
26   dependsOn = [
27     aws_s3_bucketOwnershipControls.example,
28     aws_s3_bucketPublicAccessBlock.example,
29   ]
30
31   bucket = aws_s3_bucket.demo-bucket.id
32   acl    = "public-read"
```

The screenshot shows a code editor interface with several tabs and files visible. On the left, there's a sidebar with icons for file operations like New, Open, Save, and Undo. Below that is a tree view of the project structure:

- > .terraform
- webfiles
 - index.html
 - .terraform.lock.hcl
- main.tf
- output.tf
- providers.tf
- variables.tf

The main editor area has five tabs at the top: main.tf (which is the active tab), providers.tf, variables.tf, and output.tf. The main.tf tab contains the following Terraform code:

```
25 resource "aws_s3_bucket_acl" "example" {
30   bucket = aws_s3_bucket.demo-bucket.id
31   acl    = "public-read"
32 }
33
34
35
36
37 resource "aws_s3_bucket_policy" "host_bucket_policy" {
38   bucket = aws_s3_bucket.demo-bucket.id # ID of the S3 bucket
39
40   # Policy JSON for allowing public read access
41   policy = jsonencode({
42     "Version" : "2012-10-17",
43     "Statement" : [
44       {
45         "Effect" : "Allow",
46         "Principal" : "*",
47         "Action" : "s3:GetObject",
48         "Resource": "arn:aws:s3:::${var.my_bucket_name}/*"
49       }
50     ]
51   })
52 }
53
54
55 module "template_files" [
56   source = "hashicorp/dir/template"
57
58   base_dir = "${path.module}/webfiles"
59 ]
```

```
main.tf
55 module "template_files" {
56   base_dir = "${path.module}/webfiles"
57 }
58
59
60
61 # https://registry.terraform.io/modules/hashicorp/dir/template/latest
62
63
64 resource "aws_s3_bucket_website_configuration" "web-config" {
65   bucket = aws_s3_bucket.demo-bucket.id # ID of the S3 bucket
66
67   # Configuration for the index document
68   index_document {
69     suffix = "index.html"
70   }
71 }
72
73
74 # AWS S3 object resource for hosting bucket files
75 resource "aws_s3_object" "Bucket_files" {
76   bucket = aws_s3_bucket.demo-bucket.id # ID of the S3 bucket
77
78   for_each      = module.template_files.files
79   key          = each.key
80   content_type = each.value.content_type
81
82   source       = each.value.source_path
83   content      = each.value.content
84
85   # ETag of the S3 object
86   etag         = each.value.digests.md5
87 }
```

providers.tf :

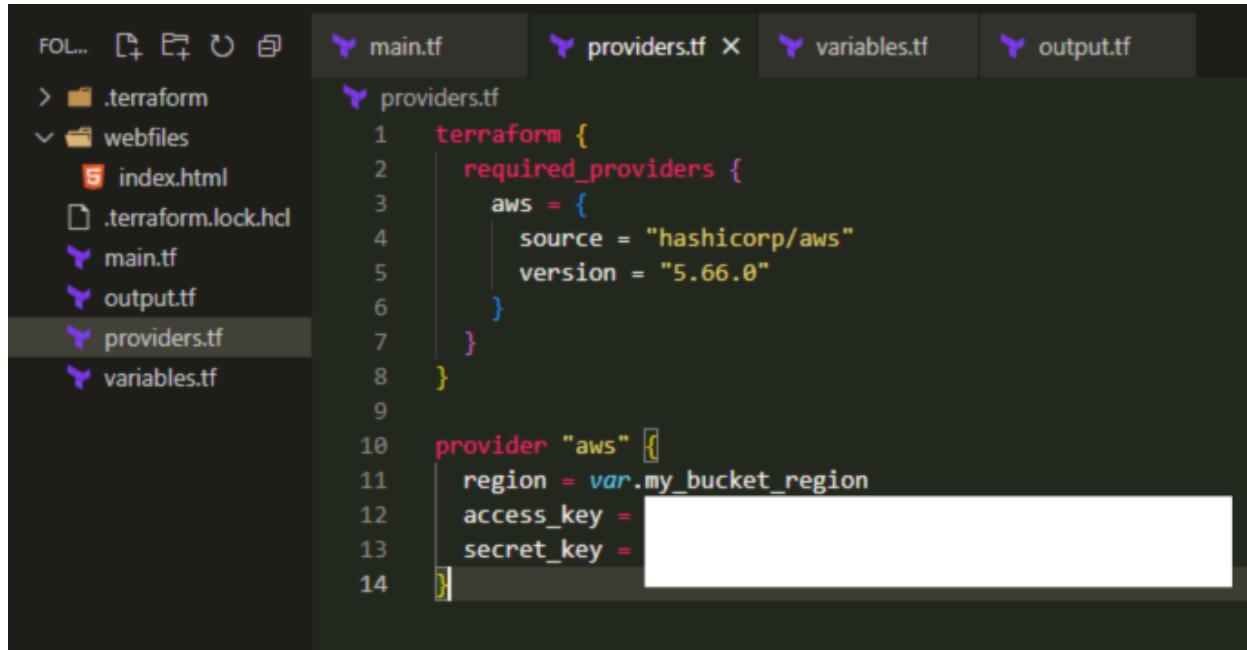
This Terraform snippet is defining the AWS provider and specifying its version for the configuration. Here's what it does:

1. Terraform Block:

- Specifies that the AWS provider (`hashicorp/aws`) version 5.66.0 is required.

2. Provider Configuration:

- Configures the AWS provider with the region set by the `my_bucket_region` variable.
- The `access_key` and `secret_key` fields are placeholders for AWS credentials, which are needed for authentication when interacting with AWS services.

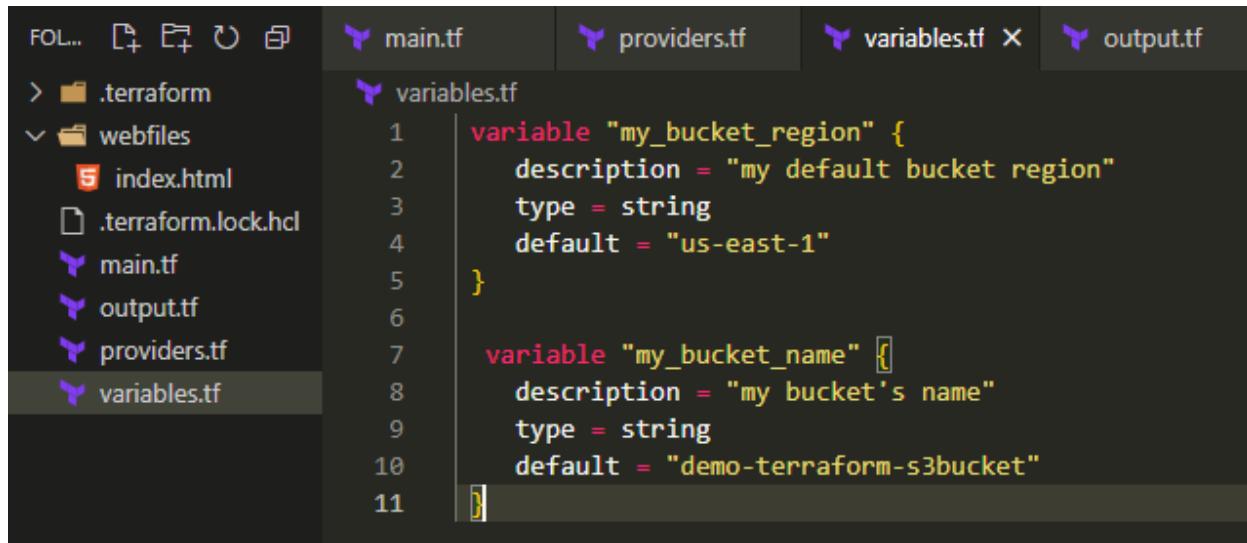


```
main.tf providers.tf variables.tf output.tf

provider "aws" {
  region = var.my_bucket_region
  access_key =
  secret_key =
}
```

variables.tf :

These variables allow flexible configuration, so you can easily change the region and bucket name without modifying the main code.



```
variable "my_bucket_region" {
  description = "my default bucket region"
  type = string
  default = "us-east-1"
}

variable "my_bucket_name" {
  description = "my bucket's name"
  type = string
  default = "demo-terraform-s3bucket"
}
```

output.tf :

This **output block** in Terraform is used to display the **URL of the static website** hosted on the S3 bucket after the configuration is applied.

- **output "website_url"**: Defines an output variable named `website_url` that will be shown after the Terraform execution.
- **description**: Describes the purpose of this output, which is to display the website's URL.

- **value**: Uses the `website_endpoint` attribute from the `aws_s3_bucket_website_configuration` resource (referenced as `web-config`). This contains the URL of the website hosted in the S3 bucket.

Once Terraform completes, it will print the website's URL for easy access.

The screenshot shows a code editor interface with several tabs and a sidebar. The tabs at the top are: main.tf, providers.tf, variables.tf, output.tf (which is currently selected), and output.tf. In the sidebar on the left, there are icons for .terraform, webfiles (with index.html), .terraform.lock.hcl, main.tf, output.tf (selected), providers.tf, and variables.tf. The main pane displays the contents of the output.tf file:

```
1  output "website_url" [
2    description = "My website URL"
3    value        = aws_s3_bucket_website_configuration.web-config.website_endpoint
4 ]
```

2. Open terminal and run terraform init.

```
PS D:\TerraformScripts\S3 Website> terraform init
Initializing the backend...
Initializing modules...
Downloading registry.terraform.io/hashicorp/dir/template 1.0.2 for template_files...
- template_files in .terraform\modules\template_files
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.66.0"...
Initializing modules...
Downloading registry.terraform.io/hashicorp/dir/template 1.0.2 for template_files...
- template_files in .terraform\modules\template_files
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.66.0"...
Downloading registry.terraform.io/hashicorp/dir/template 1.0.2 for template_files...
- template_files in .terraform\modules\template_files
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.66.0"...
- template_files in .terraform\modules\template_files
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.66.0"...
- template_files in .terraform\modules\template_files
- Installing hashicorp/aws v5.66.0...
- Installing hashicorp/aws v5.66.0...
- Installed hashicorp/aws v5.66.0 (signed by HashiCorp)
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.

3. Execute terraform plan.

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
```

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 D:\TerraformScripts\S3 Website> **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:

```
# aws_s3_bucket.demo-bucket will be created
+ resource "aws_s3_bucket" "demo-bucket" {
    + acceleration_status      = (known after apply)
    + acl                      = (known after apply)
    + arn                      = (known after apply)
    + bucket                   = "demo-terraform-s3bucket"
    + bucket_domain_name       = (known after apply)
    + bucket_prefix             = (known after apply)
    + bucketRegionalDomainName = (known after apply)
    + force_destroy             = false
    + hosted_zone_id           = (known after apply)
    + id                       = (known after apply)
    + object_lock_enabled       = (known after apply)
    + policy                   = (known after apply)
    + region                   = (known after apply)
    + request_payer             = (known after apply)
    + tags_all                 = (known after apply)
    + website_domain            = (known after apply)
    + website_endpoint          = (known after apply)

    + cors_rule (known after apply)
    + grant (known after apply)
    + lifecycle_rule (known after apply)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
+ lifecycle_rule (known after apply)

+ logging (known after apply)

+ object_lock_configuration (known after apply)

+ replication_configuration (known after apply)

+ server_side_encryption_configuration (known after apply)

+ versioning (known after apply)

+ website (known after apply)
}

# aws_s3_bucket_acl.example will be created
+ resource "aws_s3_bucket_acl" "example" {
    + acl      = "public-read"
    + bucket   = (known after apply)
    + id       = (known after apply)

    + access_control_policy (known after apply)
}

# aws_s3_bucket_ownership_controls.example will be created
+ resource "aws_s3_bucket_ownership_controls" "example" {
    + bucket   = (known after apply)
    + id       = (known after apply)

    + rule {
        + object_ownership = "BucketOwnerPreferred"
    }
}

# aws_s3_bucket_policy.host_bucket_policy will be created
+ resource "aws_s3_bucket_policy" "host_bucket_policy" {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
+ resource "aws_s3_bucket_policy" "host_bucket_policy" {
+   bucket = (known after apply)
+   id     = (known after apply)
+   policy = jsonencode(
+   {
+     Statement = [
+       {
+         Action    = "s3:GetObject"
+         Effect   = "Allow"
+         Principal = "*"
+         Resource  = "arn:aws:s3:::demo-terraform-s3bucket/*"
+       },
+     ]
+     Version  = "2012-10-17"
+   }
+ )
}

# aws_s3_bucket_public_access_block.example will be created
+ resource "aws_s3_bucket_public_access_block" "example" {
+   block_public_acls      = false
+   block_public_policy     = false
+   bucket                 = (known after apply)
+   id                     = (known after apply)
+   ignore_public_acls     = false
+   restrict_public_buckets = false
}

# aws_s3_bucket_website_configuration.web-config will be created
+ resource "aws_s3_bucket_website_configuration" "web-config" {
+   bucket          = (known after apply)
+   id              = (known after apply)
+   routing_rules   = (known after apply)
+   website_domain  = (known after apply)
+   website_endpoint = (known after apply)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
+ website_endpoint = (known after apply)

+ index_document {
    + suffix = "index.html"
}

+ routing_rule (known after apply)
}

# aws_s3_object.Bucket_files["index.html"] will be created
+ resource "aws_s3_object" "Bucket_files" {
    + acl                  = (known after apply)
    + arn                  = (known after apply)
    + bucket               = (known after apply)
    + bucket_key_enabled   = (known after apply)
    + checksum_crc32      = (known after apply)
    + checksum_crc32c     = (known after apply)
    + checksum_sha1        = (known after apply)
    + checksum_sha256      = (known after apply)
    + content_type          = "text/html; charset=utf-8"
    + etag                 = "706be2e258c7c90ddaa6d23b17cefa4a"
    + force_destroy         = false
    + id                   = (known after apply)
    + key                  = "index.html"
    + kms_key_id           = (known after apply)
    + server_side_encryption = (known after apply)
    + source               = "./webfiles/index.html"
    + storage_class         = (known after apply)
    + tags_all              = (known after apply)
    + version_id            = (known after apply)
}
}
```

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ website_url = (known after apply)
```

4. Execute terraform apply

The screenshot shows a terminal window with the following content:

```
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 D:\TerraformScripts\S3 Website> 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:  
  
# aws_s3_bucket.demo-bucket will be created  
+ resource "aws_s3_bucket" "demo-bucket" {  
    + acceleration_status      = (known after apply)  
    + acl                      = (known after apply)  
    + arn                      = (known after apply)  
    + bucket                   = "demo-terraform-s3bucket"  
    + bucket_domain_name       = (known after apply)  
    + bucket_prefix             = (known after apply)  
    + bucketRegionalDomainName = (known after apply)  
    + force_destroy             = false  
    + hosted_zone_id           = (known after apply)  
    + id                       = (known after apply)  
    + object_lock_enabled       = (known after apply)  
    + policy                   = (known after apply)  
    + region                   = (known after apply)  
    + request_payer             = (known after apply)  
    + tags_all                 = (known after apply)  
    + website_domain            = (known after apply)  
    + website_endpoint          = (known after apply)  
  
    + cors_rule (known after apply)  
    + grant (known after apply)  
    + lifecycle_rule (known after apply)  
    + logging (known after apply)  
}
```

The screenshot shows a terminal window with the following content:

```
+ logging (known after apply)  
  
+ object_lock_configuration (known after apply)  
  
+ replication_configuration (known after apply)  
  
+ server_side_encryption_configuration (known after apply)  
  
+ versioning (known after apply)  
  
+ website (known after apply)  
}  
  
# aws_s3_bucket_acl.example will be created  
+ resource "aws_s3_bucket_acl" "example" {  
    + acl      = "public-read"  
    + bucket   = (known after apply)  
    + id       = (known after apply)  
  
    + access_control_policy (known after apply)  
}  
  
# aws_s3_bucket_ownership_controls.example will be created  
+ resource "aws_s3_bucket_ownership_controls" "example" {  
    + bucket   = (known after apply)  
    + id       = (known after apply)  
  
    + rule {  
        + object_ownership = "BucketOwnerPreferred"  
    }  
}  
  
# aws_s3_bucket_policy.host_bucket_policy will be created  
+ resource "aws_s3_bucket_policy" "host_bucket_policy" {  
    + bucket   = (known after apply)  
    + id       = (known after apply)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
+ id      = (known after apply)
+ policy = jsonencode(
  {
    + Statement = [
      + {
        + Action    = "s3:GetObject"
        + Effect   = "Allow"
        + Principal = "*"
        + Resource  = "arn:aws:s3:::demo-terraform-s3bucket/*"
      },
    ]
    + Version  = "2012-10-17"
  }
)
}

# aws_s3_bucket_public_access_block.example will be created
+ resource "aws_s3_bucket_public_access_block" "example" {
  + block_public_acls      = false
  + block_public_policy     = false
  + bucket                  = (known after apply)
  + id                      = (known after apply)
  + ignore_public_acls      = false
  + restrict_public_buckets = false
}

# aws_s3_bucket_website_configuration.web-config will be created
+ resource "aws_s3_bucket_website_configuration" "web-config" {
  + bucket          = (known after apply)
  + id              = (known after apply)
  + routing_rules   = (known after apply)
  + website_domain  = (known after apply)
  + website_endpoint = (known after apply)

  + index_document {
    + suffix = "index.html"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
+ suffix = "index.html"
}

+ routing_rule (known after apply)
}

# aws_s3_object.Bucket_files["index.html"] will be created
+ resource "aws_s3_object" "Bucket_files" {
    + acl                  = (known after apply)
    + arn                  = (known after apply)
    + bucket               = (known after apply)
    + bucket_key_enabled   = (known after apply)
    + checksum_crc32      = (known after apply)
    + checksum_crc32c     = (known after apply)
    + checksum_sha1        = (known after apply)
    + checksum_sha256      = (known after apply)
    + content_type          = "text/html; charset=utf-8"
    + etag                 = "706be2e258c7c90ddaa6d23b17cefa4a"
    + force_destroy         = false
    + id                   = (known after apply)
    + key                  = "index.html"
    + kms_key_id           = (known after apply)
    + server_side_encryption = (known after apply)
    + source                = "./webfiles/index.html"
    + storage_class          = (known after apply)
    + tags_all              = (known after apply)
    + version_id             = (known after apply)
}
}
```

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ website_url = (known after apply)
```

Do you want to perform these actions?

Terraform will perform the actions described above.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

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

```
aws_s3_bucket.demo-bucket: Creating...
aws_s3_bucket.demo-bucket: Creation complete after 6s [id=demo-terraform-s3bucket]
aws_s3_bucket_ownership_controls.example: Creating...
aws_s3_object.Bucket_files["index.html"]: Creating...
aws_s3_bucket_policy.host_bucket_policy: Creating...
aws_s3_bucket_public_access_block.example: Creating...
aws_s3_bucket_website_configuration.web-config: Creating...
aws_s3_bucket_ownership_controls.example: Creation complete after 1s [id=demo-terraform-s3bucket]
aws_s3_bucket_public_access_block.example: Creation complete after 1s [id=demo-terraform-s3bucket]
aws_s3_object.Bucket_files["index.html"]: Creation complete after 1s [id=index.html]
aws_s3_bucket_acl.example: Creating...
aws_s3_bucket_policy.host_bucket_policy: Creation complete after 1s [id=demo-terraform-s3bucket]
aws_s3_bucket_website_configuration.web-config: Creation complete after 2s [id=demo-terraform-s3bucket]
aws_s3_bucket_acl.example: Creation complete after 1s [id=demo-terraform-s3bucket,public-read]
```

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

```
website_url = "demo-terraform-s3bucket.s3-website-us-east-1.amazonaws.com"
```

5. The “curl demo-terraform-s3bucket.s3-website-us-east-1.amazonaws.com” command is used to make an HTTP request to the specified URL, which corresponds to the static website hosted on an AWS S3 bucket. Here's what it does:

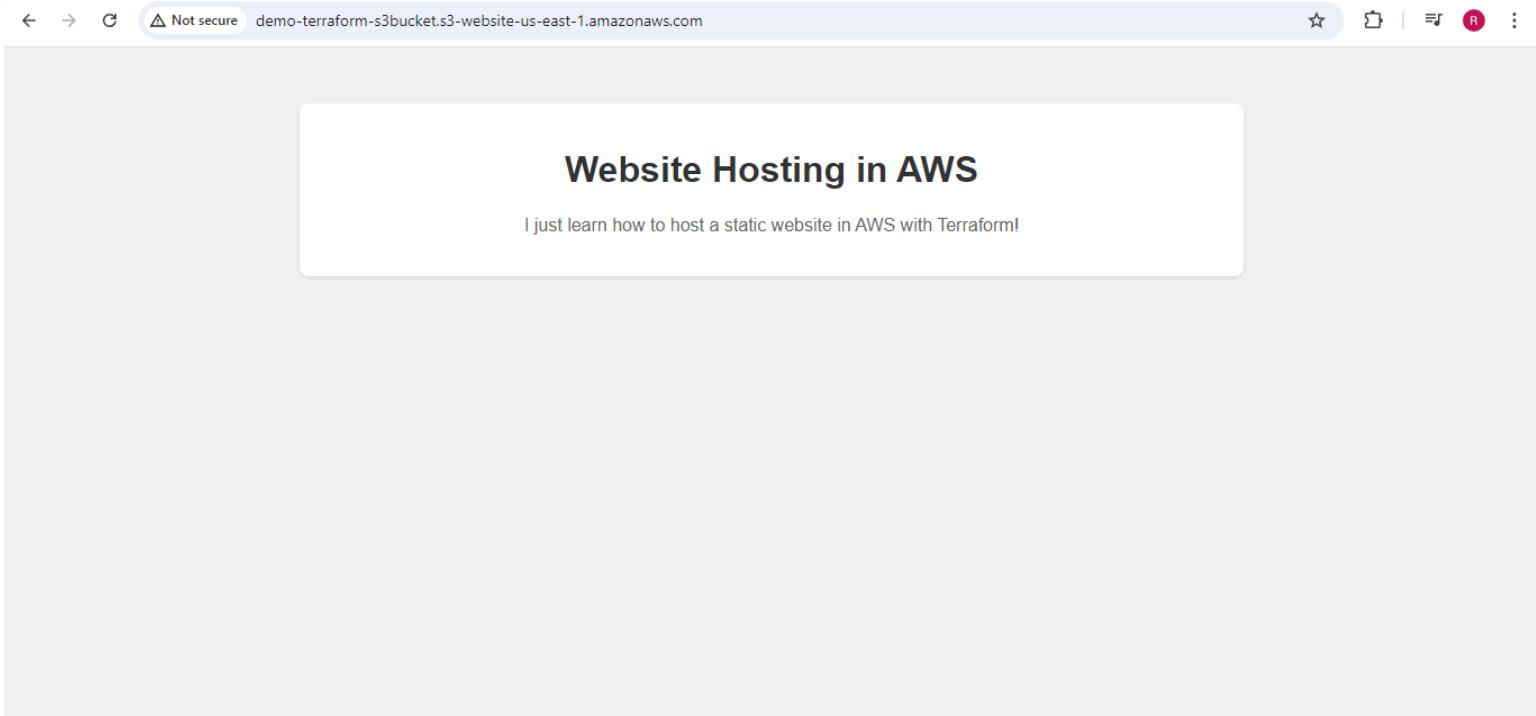
- **curl**: A command-line tool to transfer data from or to a server, commonly used to fetch web pages or APIs.

It fetches the content of the static website (likely an HTML page like `index.html`) stored in the S3 bucket and displays the response in the terminal.

```
PS D:\TerraformScripts\S3 Website> curl demo-terraform-s3bucket.s3-website-us-east-1.amazonaws.com
```

```
StatusCode      : 200
StatusDescription : OK
Content          : <!DOCTYPE html>
                  <html lang="en">
                    <head>
                      <meta charset="UTF-8">
                      <meta name="viewport" content="width=device-width, initial-scale=1.0">
                      <title>Website Hosting in AWS</title>
                      <style...>
RawContent       : HTTP/1.1 200 OK
                  x-amz-id-2: gqFsh0XNgxgx004fpo0wFbu0gY++NLD0QFMz+hIsKG24/jwo64P6K8SKFTEpXlgTJhoYk4xm89Q=
                  x-amz-request-id: KYT3KHRDSSYFT5ZE
                  Content-Length: 990
                  Content-Type: text/html; charset=utf...
Forms            : {}
Headers          : {[x-amz-id-2, gqFsh0XNgxgx004fpo0wFbu0gY++NLD0QFMz+hIsKG24/jwo64P6K8SKFTEpXlgTJhoYk4xm89Q=], [x-amz-request-id, KYT3KHRDSSYFT5ZE], [Content-Length, 990], [Content-Type, text/html; charset=utf-8]...}
Images           : {}
InputFields      : {}
Links            : {}
ParsedHtml       : mshtml.HTMLDocumentClass
RawContentLength : 990
```

6. Go to the browser and enter the url to see the hosted webpage.



7. Use “terraform destroy” to delete the bucket.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS D:\TerraformScripts\S3 Website> terraform destroy
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-terraform-s3bucket]
aws_s3_bucket_policy.host_bucket_policy: Refreshing state... [id=demo-terraform-s3bucket]
aws_s3_bucket_website_configuration.web-config: Refreshing state... [id=demo-terraform-s3bucket]
aws_s3_bucket_public_access_block.example: Refreshing state... [id=demo-terraform-s3bucket]
aws_s3_bucket_ownership_controls.example: Refreshing state... [id=demo-terraform-s3bucket]
aws_s3_object.Bucket_files["index.html"]: Refreshing state... [id=index.html]
aws_s3_bucket_acl.example: Refreshing state... [id=demo-terraform-s3bucket,public-read]

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:

# aws_s3_bucket.demo-bucket will be destroyed
- resource "aws_s3_bucket" "demo-bucket" {
    - arn                  = "arn:aws:s3:::demo-terraform-s3bucket" -> null
    - bucket               = "demo-terraform-s3bucket" -> null
    - bucket_domain_name   = "demo-terraform-s3bucket.s3.amazonaws.com" -> null
    - bucketRegionalDomainName = "demo-terraform-s3bucket.s3.us-east-1.amazonaws.com" -> null
    - force_destroy         = false -> null
    - hosted_zone_id       = "Z3AQ8STGFVJSTF" -> null
    - id                   = "demo-terraform-s3bucket" -> null
    - object_lock_enabled   = false -> null
    - policy                = jsonencode(
        {
            - Statement = [
                - {
                    - Action     = "s3:GetObject"

```

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.

There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```
aws_s3_bucket_website_configuration.web-config: Destroying... [id=demo-terraform-s3bucket]
aws_s3_object.Bucket_files["index.html"]: Destroying... [id=index.html]
aws_s3_bucket_policy.host_bucket_policy: Destroying... [id=demo-terraform-s3bucket]
aws_s3_bucket_acl.example: Destroying... [id=demo-terraform-s3bucket,public-read]
aws_s3_bucket_acl.example: Destruction complete after 0s
aws_s3_bucket_ownership_controls.example: Destroying... [id=demo-terraform-s3bucket]
aws_s3_bucket_public_access_block.example: Destroying... [id=demo-terraform-s3bucket]
aws_s3_object.Bucket_files["index.html"]: Destruction complete after 1s
aws_s3_bucket_ownership_controls.example: Destruction complete after 1s
aws_s3_bucket_public_access_block.example: Destruction complete after 2s
aws_s3_bucket_website_configuration.web-config: Destruction complete after 2s
aws_s3_bucket_policy.host_bucket_policy: Destruction complete after 2s
aws_s3_bucket.demo-bucket: Destroying... [id=demo-terraform-s3bucket]
aws_s3_bucket.demo-bucket: Destruction complete after 1s
```

Destroy complete! Resources: 7 destroyed.

PS D:\TerraformScripts\S3 Website>

Adv. Devops Experiment no. 7

Name: Rohan Lalchandani

Class: D15A Roll no.: 25

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

Theory:

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled.

It's also known as white box testing.

Why SAST is Important in DevOps ?

Early Detection of Vulnerabilities: SAST helps identify vulnerabilities early in the development phase, allowing developers to fix issues before the code is deployed to production.

Shifts Security Left: Incorporating SAST in DevOps practices supports the "Shift Left" approach in security, where testing starts earlier in the pipeline rather than waiting until the final stages.

Automation-Friendly: SAST tools can be integrated with CI/CD pipelines, ensuring automated security checks on every code commit or pull request.

SAST in DevOps Workflow Example

1. **Code Commit:** Developers push code changes to a version control system like Git.
2. **Automated Build:** The CI system (e.g., Jenkins) automatically triggers a build and kicks off the SAST scan as part of the pipeline.
3. **SAST Scan:** The SAST tool scans the code and reports vulnerabilities if present.
Some popular SAST tools for DevOps include:
4. **Build Failures or Warnings:** If critical vulnerabilities are found, the build fails or issues warnings, depending on the security policies in place.
5. **Developer Feedback:** Developers receive feedback, either through IDE plugins, the CI/CD dashboard, or via notifications, so they can address the issues quickly.
6. **Security Approval:** Once all critical vulnerabilities are resolved, the build proceeds, and the application can move to the next stage of deployment.

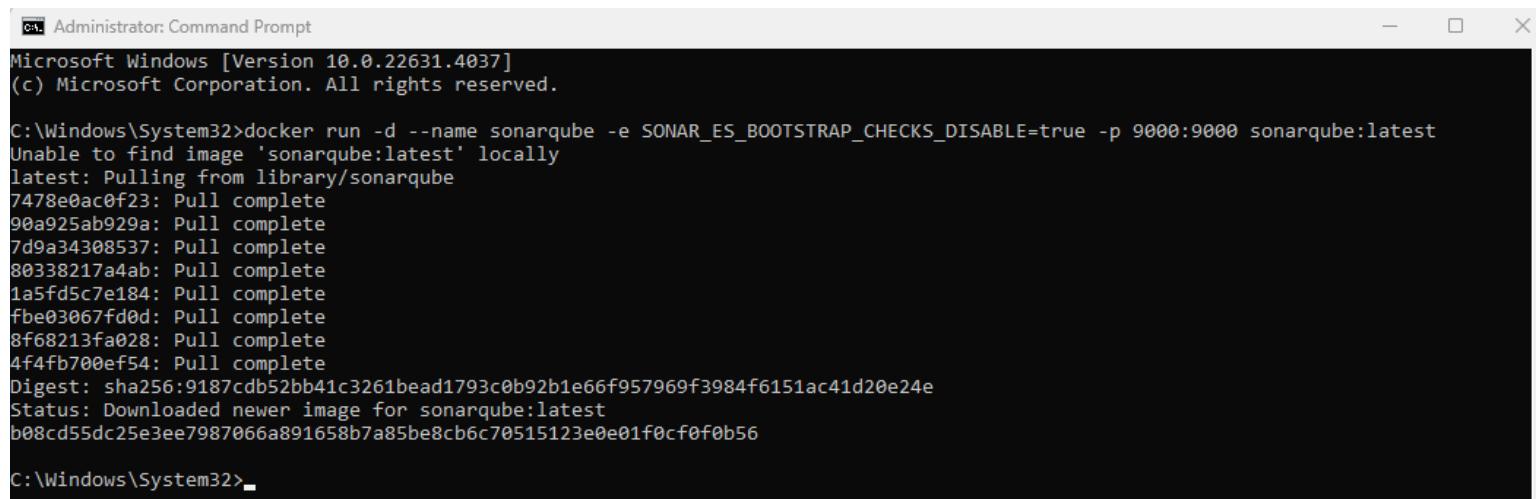
Implementation:

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

The screenshot shows the Jenkins dashboard interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information for 'Rohan Lalchandani'. Below the header, the main content area has a sidebar on the left with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main panel features a 'Welcome to Jenkins!' message, a 'Start building your software project' button, and a 'Create a job' button. It also includes sections for 'Set up a distributed build' with links to 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

Step 2: Run Sonarqube's image using the following command.

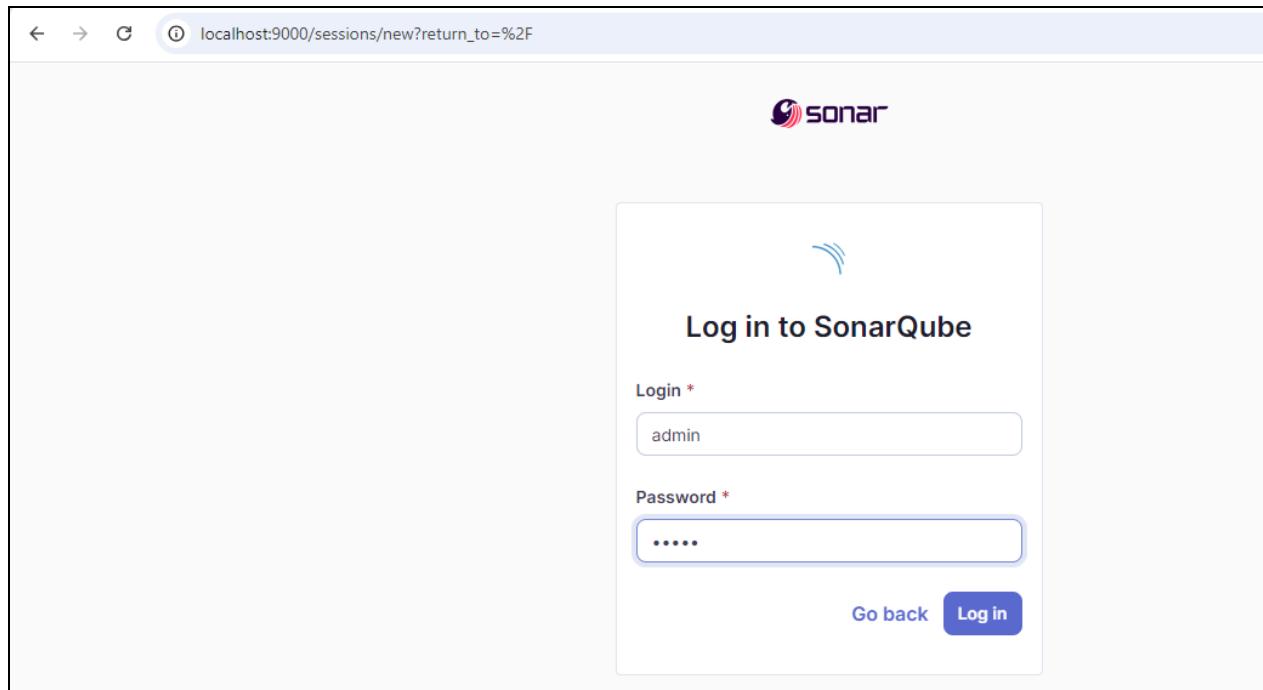
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000  
sonarqube:latest
```



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

```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.22631.4037]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
7478e0ac0f23: Pull complete  
90a925ab929a: Pull complete  
7d9a34308537: Pull complete  
80338217a4ab: Pull complete  
1a5fd5c7e184: Pull complete  
fbe03067fd0d: Pull complete  
8f68213fa028: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:9187cdb52bb41c3261bead1793c0b92b1e66f957969f3984f6151ac41d20e24e  
Status: Downloaded newer image for sonarqube:latest  
b08cd55dc25e3ee7987066a891658b7a85be8cb6c70515123e0e01f0cf0f0b56  
  
C:\Windows\System32>
```

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



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

How do you want to create your project?

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

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

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

⚠️ Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database.

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

localhost:9000/projects/create?mode=manual

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

1 of 2

Create a local project

Project display name *

 (edit)

Project key *

 (edit)

Main branch name *

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

Cancel Next

⚠️ Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

Setup the project by selecting the following options.

localhost:9000/projects/create?mode=manual&setncd=true

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

2 of 2

Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

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

Define a specific setting for this project

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

Now, for setting up sonarqube in Jenkins we need to have the login and password, for that we need to generate token.

Go to My Account > Security

The screenshot shows the SonarQube web interface at the URL `localhost:9000/tutorials?id=sonarqube`. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. A sidebar on the right is titled "Administrator" and contains links for My Account and Log out. The main content area is titled "Analysis Method" and contains a sub-section "Use this page to manage and set-up the way your analyses are performed."

Type the details and click on generate token, copy the token and save it in a notepad.

The screenshot shows the SonarQube "Security" page at the URL `localhost:9000/account/security`. The top navigation bar includes links for Profile, Security (which is selected), Notifications, and Projects. The main content area is titled "Generate Tokens". It features a form with fields for "Name" (with placeholder "Enter Token Name"), "Type" (with dropdown "Select Token Type" and "30 days" selected), and a "Generate" button. A success message box states "New token "Rohan_25" has been created. Make sure you copy it now, you won't be able to see it again!" Below the message is a token value "squ_cfcf524f12368dd3db0ba4422d89747e0299355f" with a copy icon. A table below lists tokens, showing one entry: "Name": "Rohan_25", "Type": "User", "Project": "", "Last use": "Never", "Created": "October 3, 2024", "Expiration": "November 2, 2024", and a "Revoke" button.

Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'Sonarqube scanner'. Below the search bar, there are tabs for 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. A list of available plugins is displayed, with 'SonarQube Scanner 2.17.2' highlighted. To the right of the plugin name are buttons for 'Install' and 'Install after restart'. A tooltip for 'Install' says 'Released'. Below the plugin details, a description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' and indicates it was released 7 months and 16 days ago.

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'SonarQube Scanner'. Below the search bar, there are tabs for 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. A list of available plugins is displayed, with 'SonarQube Scanner' shown as downloaded successfully. A message below the plugin says: 'Downloaded Successfully. Will be activated during the next boot'. There is also a link to 'Go back to the top page'.

Go to Installed plugins to see if sonarqube scanner is successfully installed

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'sonar'. Below the search bar, there are tabs for 'Available plugins' (selected), 'Installed plugins' (selected), and 'Advanced settings'. A list of installed plugins is displayed, with 'SonarQube Scanner for Jenkins 2.17.2' listed. The plugin is marked as 'Enabled'. To the right of the plugin name are a toggle switch (set to 'On') and a red 'X' button.

Go to Manage Jenkins > Credentials

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins. Below the header, there's a section titled "when the machine boots." followed by several configuration items:

- Appearance**: Configure the look and feel of Jenkins.
- Security**: Secure Jenkins; define who is allowed to access/use the system.
- Credentials**: Configure credentials (this item is highlighted).
- Credential Providers**: Configure the credential providers and types.
- Users**: Create/delete/modify users that can log in to this Jenkins.

Click on System

The screenshot shows the Jenkins Credentials management page. The URL is localhost:8080/manage/credentials/. The page title is "Credentials". The breadcrumb navigation is: Dashboard > Manage Jenkins > Credentials. The main content area is titled "Stores scoped to Jenkins" and shows a table of stores:

P	Store	Domains
	System	(global)

At the bottom left, there are filter buttons for Icon, S, M, and L.

Click on Global Credentials

The screenshot shows the Jenkins System store page. The URL is localhost:8080/manage/credentials/store/system/. The page title is "System". The breadcrumb navigation is: Dashboard > Manage Jenkins > Credentials > System. The main content area shows a table of credentials:

Domain	Description
	Global credentials (unrestricted) Credentials that should be available irrespective of domain specification to requirements matching.

At the bottom left, there are filter buttons for Icon, S, M, and L.

Click on Add Credentials

The screenshot shows the Jenkins Global credentials (unrestricted) page. At the top right, there is a blue button labeled '+ Add Credentials'. Below it, a message says 'This credential domain is empty. How about [adding some credentials?](#)'.

Now paste the generated token from sonarqube in the secret box and fill in other details as shown below.

The screenshot shows the Jenkins New credentials page. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains a long string of asterisks. The 'ID' field is set to 'generate_token'. The 'Description' field is also set to 'generate_token'. At the bottom left is a blue 'Create' button.

Token has been added successfully to the Jenkins

The screenshot shows the Jenkins Global credentials (unrestricted) page again. The newly added credential is listed in the table:

ID	Name	Kind	Description
generate_token	generate_token	Secret text	generate_token

Configure SonarQube Scanner in Jenkins

- Go to Manage Jenkins > System. Scroll down to Sonarqube servers, check on environment variables.
- Select he token from drop down menu.
- Click on save.

The screenshot shows the Jenkins management interface at localhost:8080/manage/configure. The path in the navigation bar is Dashboard > Manage Jenkins > System. Under the 'SonarQube servers' heading, there is a note: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' A checkbox labeled 'Environment variables' is checked. Below this is the 'SonarQube installations' section, which includes a link to 'List of SonarQube installations' and a button to 'Add SonarQube'.

This screenshot shows the same Jenkins management interface as the previous one, but it has zoomed into the 'SonarQube installations' section. It displays a configuration for a single instance named 'sonarqube_Rohan'. The fields shown are: Name (sonarqube_Rohan), Server URL (http://localhost:9000), and Server authentication token (generate_token). There is also a '+ Add' button and an 'Advanced' dropdown.

Configure SonarQube Scanner in Jenkins

- Go to Manage Jenkins > Global Tool Configuration.
- Scroll down to SonarQube Scanner.
- Choose the latest version and select Install automatically

The screenshot shows the Jenkins 'Configure Tools' page at localhost:8080/manage/configureTools/. The 'SonarQube Scanner installations' section is active. A new configuration is being added for 'sonarqube_Rohan'. The 'Install automatically' checkbox is checked. The 'Version' dropdown is set to 'SonarQube Scanner 6.2.1.4610'. There is also an 'Add Installer' link. At the bottom are 'Save' and 'Apply' buttons.

Go to your Jenkins Dashboard and Create a new Job

The screenshot shows the Jenkins dashboard at localhost:8080. The 'Welcome to Jenkins!' message is displayed. On the left, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. On the right, under 'Start building your software project', there is a 'Create a job' button which is highlighted with a red box. Other options include 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and a link to 'Learn more about distributed builds'.

Enter name of the project and select freestyle project

The screenshot shows the Jenkins 'New Job' creation interface. At the top, there's a search bar and a user profile for 'Rohan Lalchandani'. Below it, the 'Dashboard' link is visible. The main area has a title 'Enter an item name' with a text input field containing 'sonarqube_Rohan'. A red box highlights the 'Freestyle project' section, which includes a icon of a folder, the text 'Freestyle project', and a description: '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.' Below this are other options: 'Pipeline' (with a circular icon) and 'Multi-configuration project' (with a gear icon).

Under Configure put the github link in source code management
https://github.com/shazforiot/MSBuild_firstproject.git

The screenshot shows the Jenkins configuration page for the 'sonarqube_Rohan' job. The left sidebar lists 'General', 'Source Code Management' (which is selected and highlighted in grey), 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main configuration area is titled 'Configure'. It shows a radio button for 'None' and another for 'Git' (which is selected). Under 'Git', there's a 'Repositories' section with a 'Repository URL' input field containing 'https://github.com/shazforiot/MSBuild_firstproject.git'. Below it are 'Credentials' (set to '- none -') and an 'Advanced' dropdown. At the bottom of the configuration area are 'Save' and 'Apply' buttons.

In build steps Select “Execute SonarQube scanner”

The screenshot shows the Jenkins configuration page for a job named "sonarqube_Rohan". The "Build Environment" tab is selected. In the "Build Steps" section, a dropdown menu is open under "Add build step". The "Execute SonarQube Scanner" option is highlighted with a red box.

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Add build step ^

Filter

Execute SonarQube Scanner

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

SonarScanner for MSBuild - Begin Analysis

SonarScanner for MSBuild - End Analysis

Type the following in analysis properties:

- sonar.projectKey=my_project_name
- sonar.login=your_generated_token
- sonar.sources=HelloWorldCore
- sonar.host.url=http://localhost:9000

The screenshot shows the Jenkins configuration interface for a job named 'sonarqube_Rohan'. The left sidebar has sections: General, Source Code Management, Build Triggers, Build Environment, **Build Steps**, and Post-build Actions. The 'Build Steps' section is expanded, showing a step titled 'Execute SonarQube Scanner'. It includes fields for 'JDK' (set to 'Inherit From Job'), 'Path to project properties' (empty), 'Analysis properties' containing the configuration:
sonar.projectKey=sonarqube_Rohan
sonar.login=squ_cfcf524f12368dd3db0ba4422d89747e0299355f
sonar.source=HelloWorldCore
sonar.host.url=http://localhost:9000, and 'Additional arguments' (empty). At the bottom are 'Save' and 'Apply' buttons.

Now we need to enable access to sonarqube.
For that go to the sonarqube website
click on administration and then global permissions.

The screenshot shows the SonarQube administration interface at 'localhost:9000/admin/settings'. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, **Administration** (which is highlighted with a red box labeled '1.'), More, and a search icon. Below the navigation is a secondary menu with 'General' selected, showing options for Users, Groups, and Global Permissions. A sub-menu for 'Edit global settings' is open, with 'Global Permissions' highlighted with a red box labeled '2.'. The main content area contains sections for Analysis Scope, Authentication, and DevOps Platform Integrations, along with a 'Duplications' section.

Click on Anyone in the checkbox as shown below.

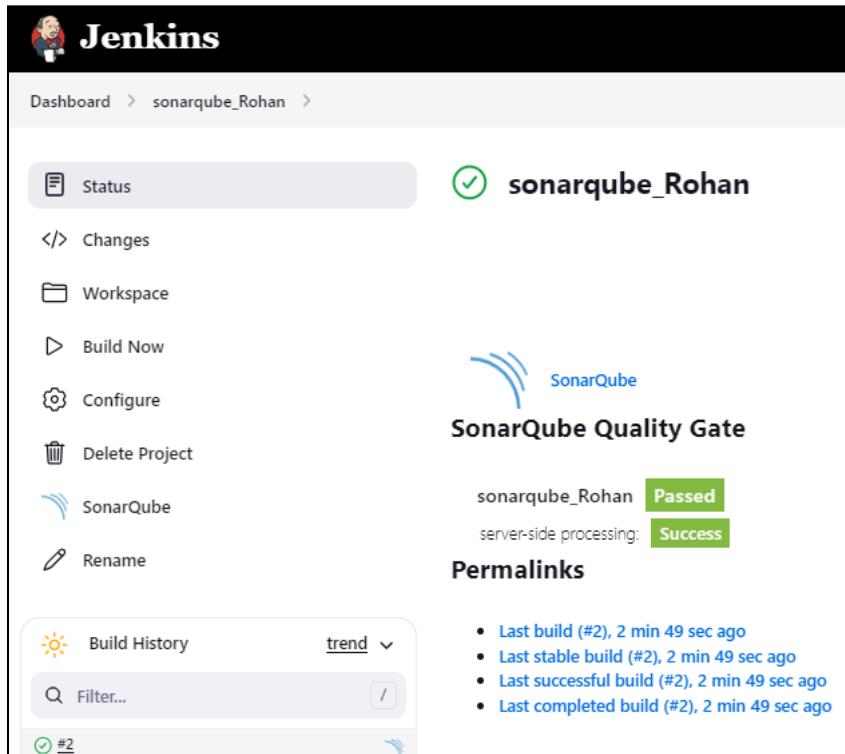
The screenshot shows the SonarQube Administration interface under the Security tab. It lists three groups: 'sonar-administrators', 'sonar-users', and 'Anyone'. For each group, there are checkboxes for 'Administer System', 'Administer', 'Execute Analysis', and 'Create'. There are also checkboxes for 'Quality Gates' and 'Quality Profiles'. A red box highlights the 'Execute Analysis' checkbox for the 'Anyone' group, which is checked.

Group	Administer System	Administer	Execute Analysis	Create
sonar-administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
sonar-users	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Anyone <small>DEPRECATED</small>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Go to the Jenkins Dashboard and click on “Build Now”

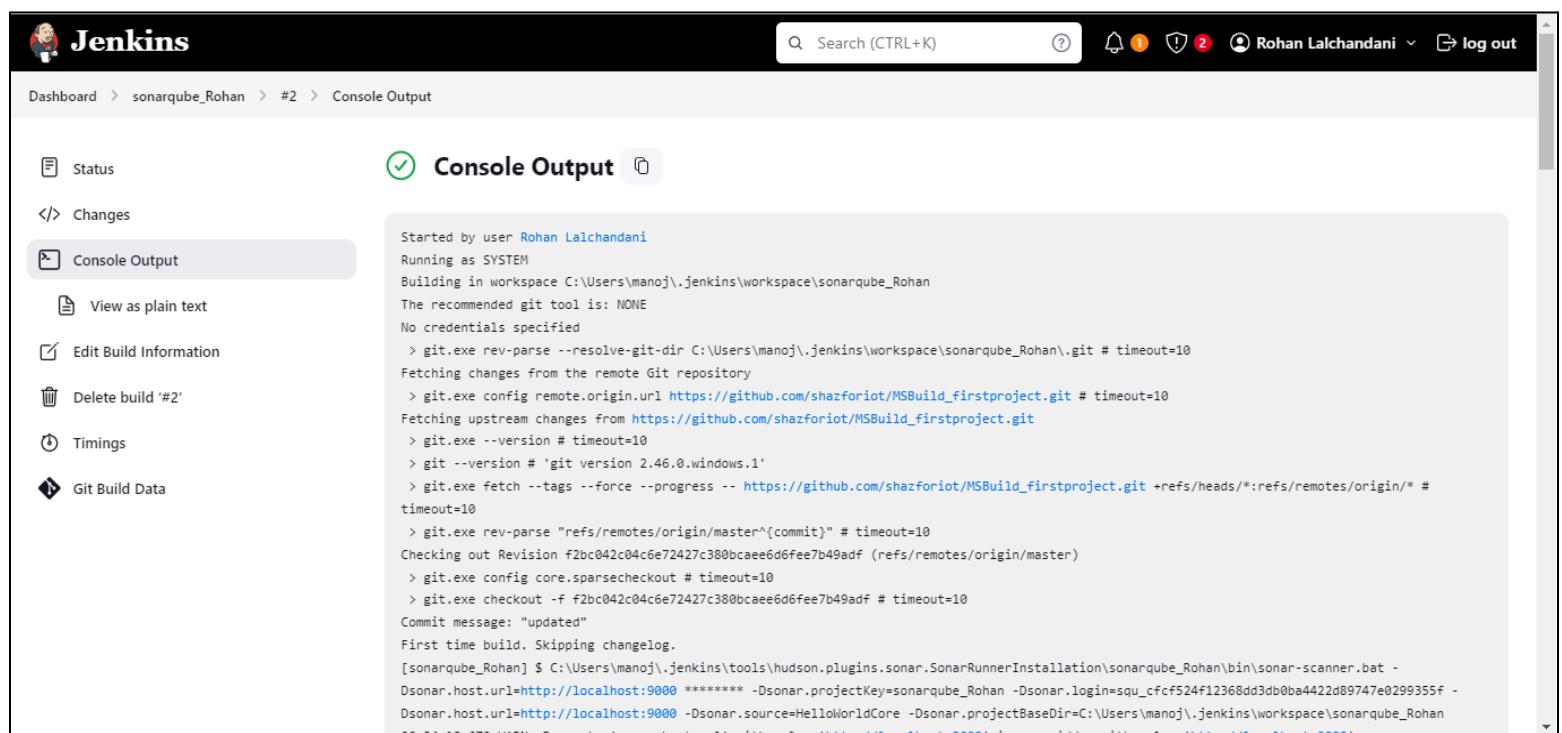
The screenshot shows the Jenkins dashboard for the project 'sonarqube_Rohan'. The left sidebar has links for 'Status', 'Changes', 'Workspace', 'Build Now' (which is highlighted with a red box), 'Configure', 'Delete Project', 'SonarQube', and 'Rename'. The main area shows the project name 'sonarqube_Rohan' and a 'Permalinks' section with a SonarQube icon. The top right corner shows the user 'Rohan Lalchandani' and a 'log out' link.

The build has been completed successfully.



A screenshot of the Jenkins interface showing the SonarQube Quality Gate results for the project "sonarqube_Rohan". The status is "Passed" with a green checkmark icon. Below it, "server-side processing" is also listed as "Success". The "Permalinks" section shows the last four builds, all of which are successful. On the left sidebar, there are links for Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. A "Build History" section is visible at the bottom left.

Go to the console output.



A screenshot of the Jenkins interface showing the "Console Output" page for build #2 of the project "sonarqube_Rohan". The status is "Passed" with a green checkmark icon. The console output window displays the command-line logs of the build process, including the execution of git commands to clone the repository and run the SonarQube scanner. The left sidebar shows other options like Status, Changes, and View as plain text.

```
Started by user Rohan Lalchandani
Running as SYSTEM
Building in workspace C:\Users\manoj\.jenkins\workspace\sonarqube_Rohan
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\Users\manoj\.jenkins\workspace\sonarqube_Rohan\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/*
timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
[sonarqube_Rohan] $ C:\Users\manoj\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_Rohan\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 ***** -Dsonar.projectKey=sonarqube_Rohan -Dsonar.login=squ_cfcf524f12368dd3db0ba4422d89747e0299355f -
Dsonar.host.url=http://localhost:9000 -Dsonar.source=HelloWorldCore -Dsonar.projectBaseDir=C:\Users\manoj\.jenkins\workspace\sonarqube_Rohan
```

Scroll down click on the link as shown below to view the SonarQube analysis report.

```
02:34:34.415 INFO Analysis report compressed in 44ms, zip size=23.8 kB
02:34:34.448 INFO Analysis report uploaded in 51ms
02:34:34.451 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube\_Rohan
02:34:34.451 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
02:34:34.451 INFO More about the report processing at http://localhost:9000/api/ce/task?id=91e88697-16f7-4df8-9bcc-69a576460a6c
02:34:34.468 INFO Analysis total time: 18.278 s
02:34:34.471 INFO SonarScanner Engine completed successfully
02:34:34.558 INFO EXECUTION SUCCESS
02:34:34.559 INFO Total time: 21.879s
Finished: SUCCESS
```

You can see the code has successfully passed.

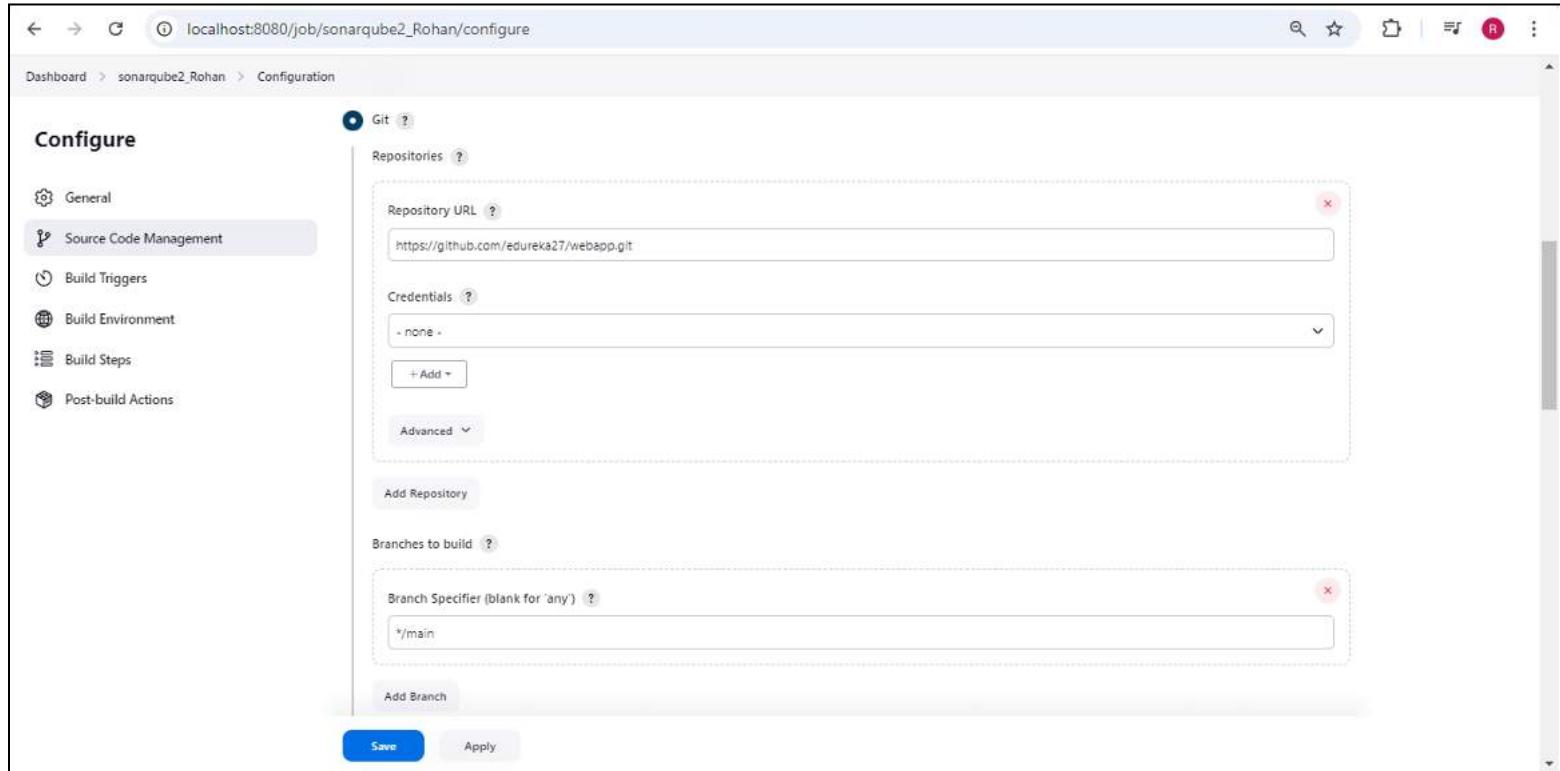
The screenshot shows the SonarQube dashboard for the project 'sonarqube_Rohan'. The main header includes the SonarQube logo, navigation links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More, along with a search bar and user profile icons. Below the header, the breadcrumb navigation shows 'sonarqube_Rohan / main'. The top navigation bar has tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity, with 'Overview' being the active tab. To the right of the tabs are Project Settings and Project Information dropdowns. A prominent green checkmark icon indicates a 'Passed' status for the analysis. A yellow warning box states 'The last analysis has warnings. See details'. The main content area displays various quality metrics in cards:

- New Code** (Overall Code selected): Security (0 Open issues), Reliability (0 Open issues), Maintainability (0 Open issues).
- Accepted issues**: 0 Valid issues that were not fixed.
- Coverage**: On 0 lines to cover.
- Duplications**: 0.0% on 86 lines.

On the right side of the dashboard, there is a vertical sidebar with several small icons and sections labeled 'Project', 'Code', 'Issues', 'Measures', 'Activity', and 'Administration'.

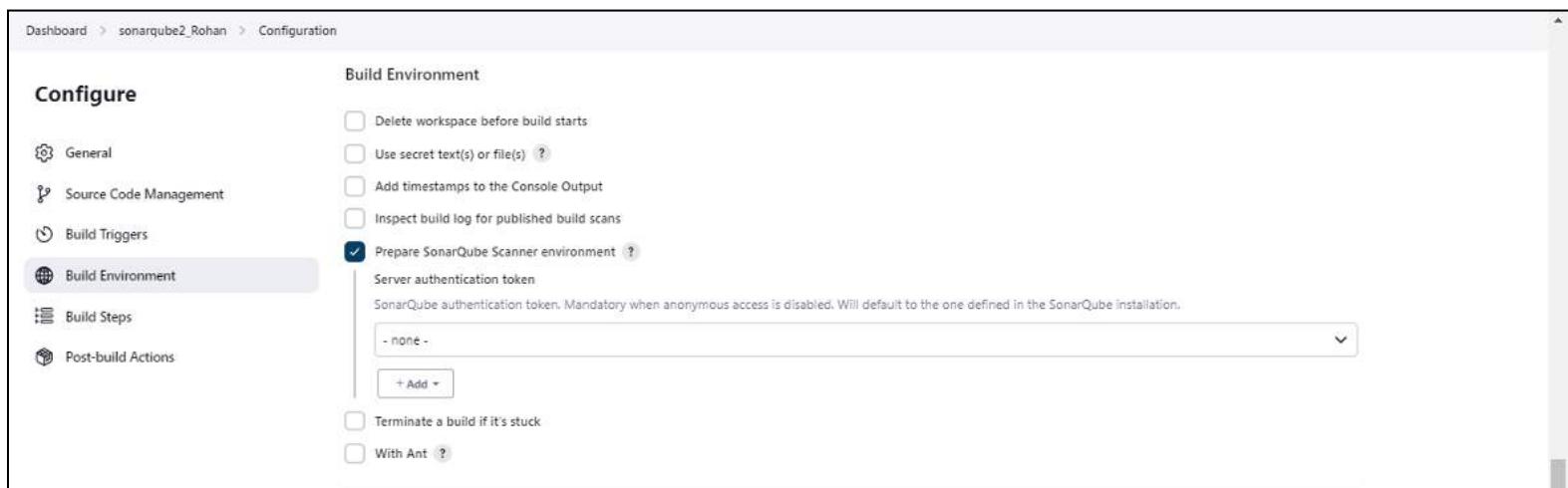
(EXTRA) : Building and Analysis of Maven project.

1. Create a maven project and paste the github link in source code management
<https://github.com/edureka27/webapp.git>



The screenshot shows the Jenkins configuration interface for a job named 'sonarqube2_Rohan'. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Source Code Management' section is currently selected. The main panel is titled 'Git' and contains fields for 'Repository URL' (set to 'https://github.com/edureka27/webapp.git') and 'Branches to build' (set to '*/*'). There are 'Save' and 'Apply' buttons at the bottom.

2. Check on prepare sonarqube scanner



The screenshot shows the Jenkins configuration interface for a job named 'sonarqube2_Rohan'. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Build Environment' section is currently selected. The main panel contains several checkboxes under 'Build Environment': 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', and 'Prepare SonarQube Scanner environment' (which is checked). Below this, there is a section for 'Server authentication token' with a dropdown menu set to '- none -'. There are also checkboxes for 'Terminate a build if it's stuck' and 'With Ant'. At the bottom of the panel are 'Save' and 'Apply' buttons.

3. Give the maven_path and click on save.

Build Steps

≡ Invoke top-level Maven targets ? ✖

Maven Version

Maven_Path

Goals

clean install sonar:sonar

Advanced ▾

Build completed successfully.

Dashboard > sonarqube2_Rohan >

Status ✓ **sonarqube2_Rohan**

</> Changes

📁 Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

SonarQube

✍️ Rename

 SonarQube

SonarQube Quality Gate

MyWebApp Maven Webapp Passed

server-side processing: Success

Permalinks

Build History trend ▾

Filter... /

🕒 #9 ⚡

Oct 3, 2024, 10:15 AM

- Last build (#9), 7 min 33 sec ago
- Last stable build (#9), 7 min 33 sec ago
- Last successful build (#9), 7 min 33 sec ago
- Last failed build (#8), 8 min 54 sec ago
- Last unsuccessful build (#8), 8 min 54 sec ago
- Last completed build (#9), 7 min 33 sec ago

Go to the console output and scroll down click on the analysis report link and you will be able to see the sonarqube analysis report.

The screenshot shows the SonarQube interface for a project named "MyWebApp Maven Webapp". The "Issues" tab is selected. On the left, there's a sidebar with filters for "Clean Code Attribute" (Consistency: 2, Intentionality: 1, Adaptability: 0, Responsibility: 0) and "Software Quality" (Security: 0, Reliability: 3). The main area displays three issues:

- src/main/webapp/index.jsp**:
 - Add a <title> tag to this page. (Reliability: 0) - Consistency: user-experience
 - Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability: 0) - Consistency: user-experience
 - Add "lang" and/or "xml:lang" attributes to this "<html>" element. (Reliability: 0) - Intentionality: accessibility wcag2-a

A message at the bottom states: "Embedded database should be used for evaluation purposes only".

Adv. Devops Experiment no. 8

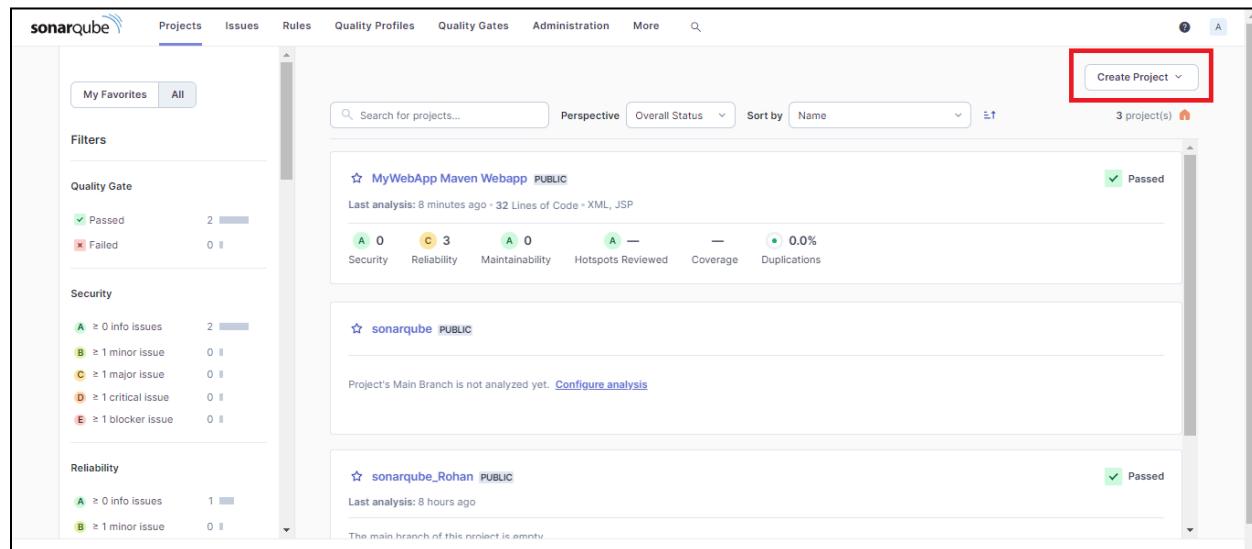
Name: Rohan Lalchandani

Class: D15A Roll no.: 25

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

Implementation:

1. Download SonarQube by following steps of experiment 7, login using credentials and create a new project.



The screenshot shows the SonarQube web interface. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. On the right side of the header, there's a 'Create Project' button with a dropdown arrow, which is highlighted with a red box. Below the header, there's a 'Filters' section on the left containing sections for Quality Gate (Passed: 2, Failed: 0), Security (0 info issues, 0 minor issues, 0 major issues, 0 critical issues, 0 blocker issues), and Reliability (0 info issues, 1 minor issue). The main area displays three projects: 'MyWebApp Maven Webapp' (PUBLIC, Last analysis: 8 minutes ago, 32 Lines of Code - XML, JSP, 0 info issues, 3 critical issues, 0 major issues, 0 blocker issues, 0.0% coverage, 0.0% duplications), 'sonarqube' (PUBLIC, Last analysis: 8 hours ago, Main Branch not analyzed yet), and 'sonarqube_Rohan' (PUBLIC, Last analysis: 8 hours ago, Main branch of this project is amity).

2. Type the name of the project as shown.

The screenshot shows a web browser window with the URL `localhost:9000/projects/create?mode=manual`. The page title is "Create a local project". The form fields are as follows:

- Project display name ***: Pipeline (highlighted with a green border)
- Project key ***: Pipeline (highlighted with a green border)
- Main branch name ***: main

Below the form, there is a note: "The name of your project's default branch [Learn More](#)". At the bottom are two buttons: "Cancel" and "Next".

3. Use the global setting.

The screenshot shows the SonarQube interface for creating a new project. The URL in the address bar is `localhost:9000/projects/create?mode=manual&setncd=true`. The page title is "Set up project for Clean as You Code". A note at the top states: "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. You Code methodology. Learn more: [Defining New Code](#)". Below this, a section titled "Choose the baseline for new code for this project" contains two options:

- Use the global setting (highlighted with a red box)
- Define a specific setting for this project
 - Previous version (highlighted with a light gray background)
 - Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

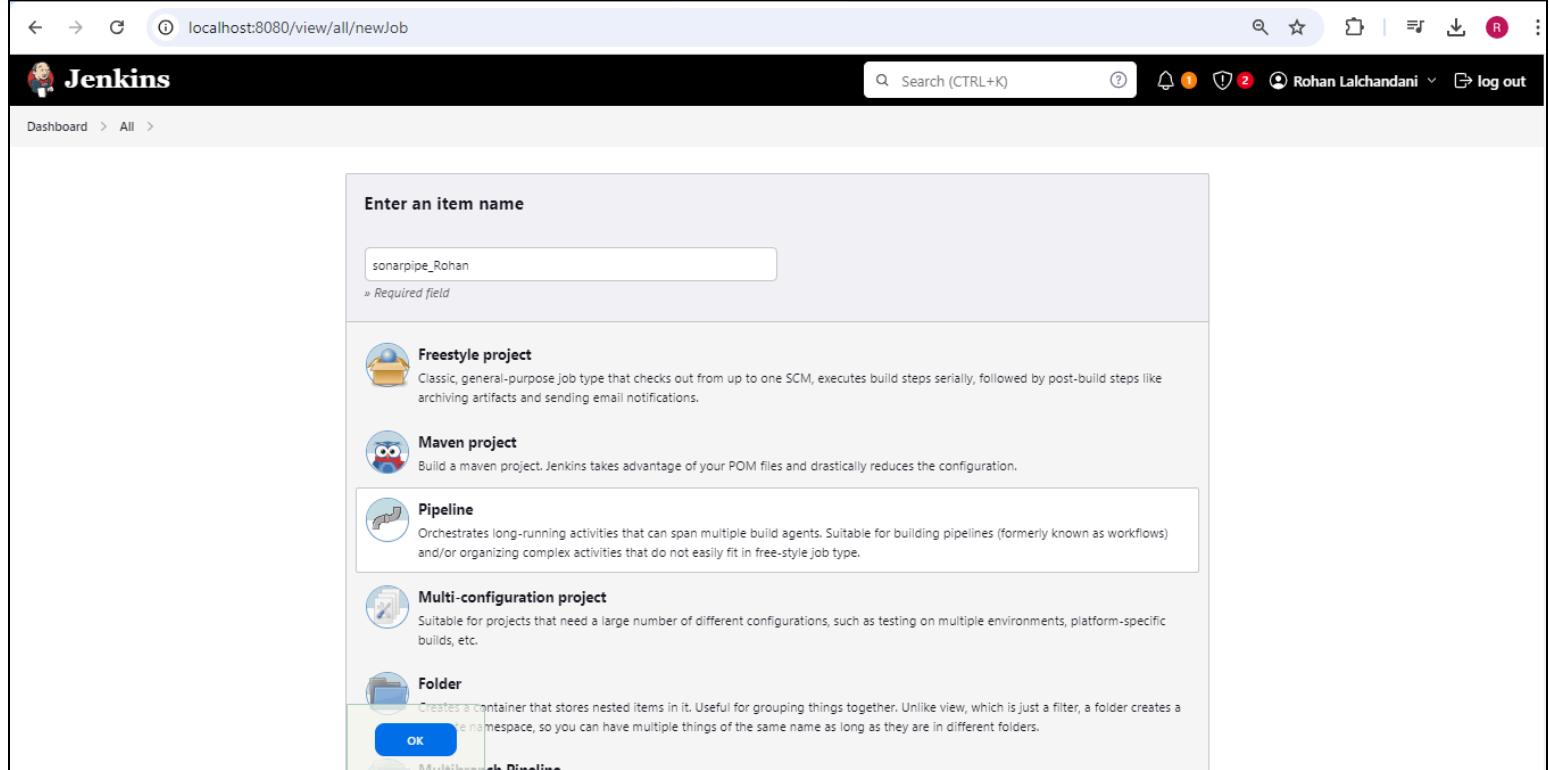
4. See if your project is created in the dashboard.

The screenshot shows the SonarQube dashboard with the "Projects" tab selected. On the left, there are filters for "Quality Gate" (Passed: 2, Failed: 0) and "Security" (0 info issues, 0 minor issues, 0 major issues, 0 critical issues, 0 blocker issues). The main area displays the analysis results for the "MyWebApp Maven Webapp" project, which is PUBLIC and has a green "Passed" status. It shows the following metrics:

- Security: A 0
- Reliability: C 3
- Maintainability: A 0
- Hotspots Reviewed: A —
- Coverage: —
- Duplications: 0.0%

A note below states: "Last analysis: 11 minutes ago - 32 Lines of Code - XML, JSP". To the right, another project, "Pipeline", is listed as PUBLIC with the note: "Project's Main Branch is not analyzed yet".

5. Now go to Jenkins, and create a pipeline.



The screenshot shows the Jenkins web interface at localhost:8080/view/all/newJob. A modal window titled "Enter an item name" is open, with the value "sonarpipe_Rohan" entered into the input field. Below the input field, there is a note: "» Required field". A list of project types is displayed: "Freestyle project", "Maven project", "Pipeline" (which is selected and highlighted with a blue border), "Multi-configuration project", and "Folder". Each item has a brief description and a small icon. At the bottom of the modal, there is an "OK" button and a link to "Advanced".

6. Enter the pipeline script

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

Pipeline

Definition

Pipeline script

```

1+ node {
2+   stage('Cloning the GitHub Repo') {
3+     git 'https://github.com/shafzoriot/GOL.git'
4+   }
5+   stage('SonarQube analysis') {
6+     withSonarQubeEnv('sonarqube') {
7+       bat """
8+         docker run --rm ^
9+           -e SONAR_HOST_URL=http://172.20.64.1:9000 ^
10+          -v ${WORKSPACE.replace('\\', '/')}/:usr/src ^
11+            sonarsource/sonar-scanner-cli ^
12+            -Dsonar.projectKey=sonarqube-test ^
13+            -Dsonar.sources=. ^
14+            -Dsonar.exclusions=*/.resources/,*.java ^
15+            -Dsonar.login=admin ^
16+            -Dsonar.password=squ_cfcf524f12368dd3db0ba4422d89747e0299355f
17+
18+     }
19+   }
}

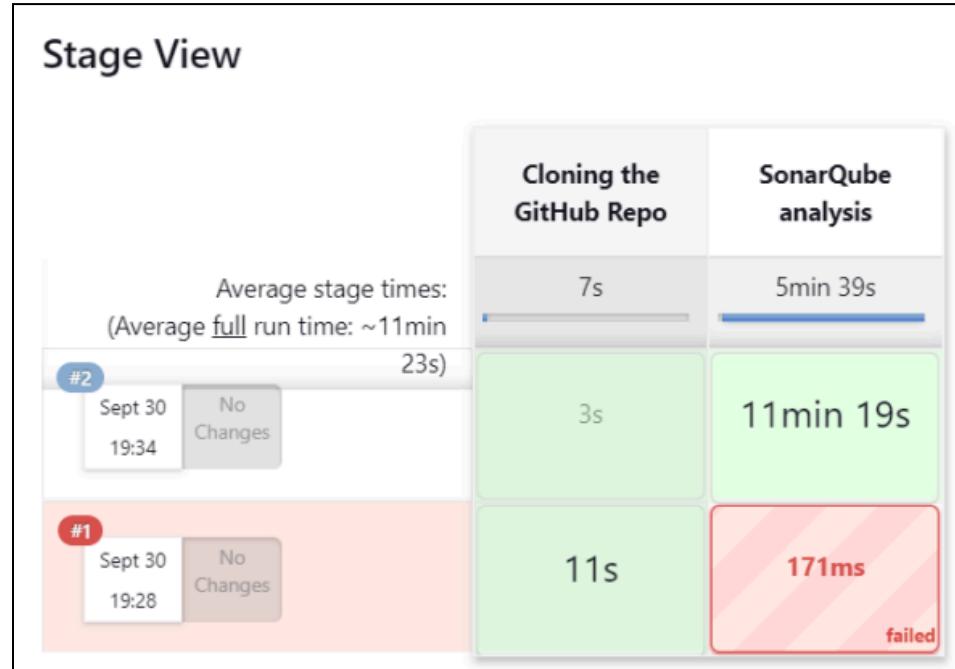
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Save](#) [Apply](#)

7. See the Stage view and console output the build has completed successfully.



Jenkins

Dashboard > SonarQube_pipeline > #2

Status Changes Console Output Edit Build Information Delete build #2 Timings Git Build Data Pipeline Overview Pipeline Console Replay Pipeline Steps unknown

Console Output

Skipping 4.247 KB... Full Log

```
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 145. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 506. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 759. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 785. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 659. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 511. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 578. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 707. Keep only the first 100 references.  
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for
```

8. See the analysis report in SonarQube.

Sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Quality Gate: **Passed** Last analysis 26 minutes ago

The last analysis has warnings. See details

New Code Overall Code

New Code: Since September 26, 2024 Started 4 days ago

New Issues Accepted Issues

New Issues	Accepted Issues
0	0

Required = 0 Valid issues that were not fixed

Coverage Duplications Security Hotspots

Coverage	Duplications	Security Hotspots
●	●	● A

My Issues All

Bulk Change Select issues ▾ Navigate to issue ▾ 210,549 issues 3135d effort

Filters gameoflife-acceptance-tests/Dockerfile

Issues in new code

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image. Intentionality

Maintainability ⓘ No tags +

Open Not assigned L1 · 5min effort · 4 years ago · ⚡ Code Smell ⚡ Major

Clean Code Attribute

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Software Quality

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

Maintainability ⓘ No tags +

Open Not assigned L12 · 5min effort · 4 years ago · ⚡ Code Smell ⚡ Major

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

Maintainability ⓘ No tags +

ADVANCE DEVOPS EXP 9

Name : Rohan Lalchandani

Class : D15A Roll no : 25

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

1. Create an Amazon Linux EC2 Instance

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts. The main pane displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. Three instances are listed: 'Webpage-env' (Running, t3.micro, 3/3 checks passed, ap-south-1a, ec2-3-1), 'nagios-host' (Running, t2.micro, 2/2 checks passed, ap-south-1b, ec2-15-1), and 'linux-client1' (Running, t2.micro, 2/2 checks passed, ap-south-1b, ec2-13-1). A search bar at the top allows filtering by instance attribute or tag.

2. Configure Security Group

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

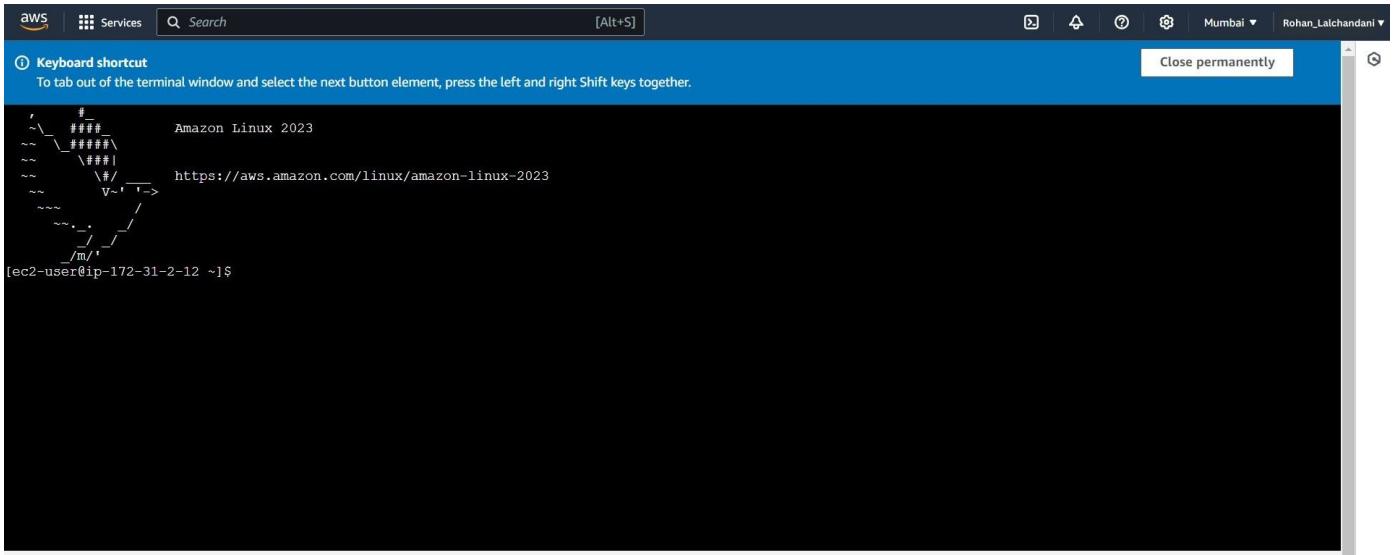
The screenshot shows the 'Edit inbound rules' page for a specific security group. It lists five existing rules:

- sgr-00d59807a11e25687: All ICMP - IPv4, ICMP protocol, All port range, Custom source (0.0.0.0/0), Description: Delete
- sgr-0ae620ec0b187c4a7: All traffic, All protocol, All port range, Custom source (0.0.0.0/0), Description: Delete
- sgr-0775d4388ffe14db6: SSH, TCP protocol, 22 port range, Custom source (0.0.0.0/0), Description: Delete
- sgr-0ebadedcb97cb60fc: HTTP, TCP protocol, 80 port range, Custom source (0.0.0.0/0), Description: Delete
- sgr-08983e0020306b273: HTTPS, TCP protocol, 443 port range, Custom source (0.0.0.0/0), Description: Delete

At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

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

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



AWS Lambda terminal window showing a file download progress bar for 'Amazon Linux 2023' from 'https://aws.amazon.com/linux/amazon-linux-2023'. The progress bar is at approximately 10% completion.

```
[ec2-user@ip-172-31-2-12 ~]$
```

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

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

sudo yum install httpd php

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

=====
Package           Architecture Version       Repository      Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    48 M
php8.3           x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    10 M
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2  amazonlinux   129 M
apr-util          x86_64      1.6.3-1.amzn2023.0.1  amazonlinux   98 M
generic-logos-httpd  noarch    18.0.0-12.amzn2023.0.3  amazonlinux   19 M
httpd-core        x86_64      2.4.62-1.amzn2023      amazonlinux   1.4 M
httpd-filesystem  noarch    2.4.62-1.amzn2023      amazonlinux   14 M
httpd-tools        x86_64      2.4.62-1.amzn2023      amazonlinux   81 M
libbrotli          x86_64      1.0.9-4.amzn2023.0.2  amazonlinux   315 M
libsodium          x86_64      1.0.19-4.amzn2023      amazonlinux   176 M
libxml2            x86_64      1.1.34-5.amzn2023.0.2  amazonlinux   241 M
mailcap            noarch    2.1.49-3.amzn2023.0.3  amazonlinux   33 M
nginx             noarch    1:1.24.0-1.amzn2023.0.4  amazonlinux   9.8 M
nginx-filesystem  noarch    8.3.10-1.amzn2023.0.1  amazonlinux   3.7 M
php8.3-cli         x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   737 M
php8.3-common      x86_64      8.3.10-1.amzn2023.0.1  amazonlinux   45 M
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:02:41 ago on Wed Oct 2 05:48:47 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.

Transaction Summary
```

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 M
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 K
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 K
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 K
libcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 K
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 K

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:46 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.

Transaction Summary
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 M
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 M
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-fs	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 M
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 K
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 K
fonts-fs	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 M
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	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 M
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

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

sudo adduser -m nagios sudo passwd nagios

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

6. Create a new user group

sudo groupadd nagcmd

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

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

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

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

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

9. Use wget to download the source zip files.

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

```
(ec2-user@ip-172-31-40-254 downloads)$ Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz  
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz  
--2024-10-02 06:15:45-- https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz  
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251  
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2782610 (2.7M) [application/x-gzip]  
Saving to: 'nagios-plugins-2.3.3.tar.gz'  
  
nagios-plugins-2.3.3.tar.gz      0%[=====] 632.00K 3.02MB/s  
nagios-plugins-2.3.3.tar.gz    23%[=====] 2.65M 8.10MB/s  in 0.3s  
nagios-plugins-2.3.3.tar.gz  100%[=====] 2.65M 8.10MB/s  in 0.3s  
  
2024-10-02 06:15:46 (8.10 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]  
[ec2-user@ip-172-31-40-254 downloads]$ ]
```

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

```
(ec2-user@ip-172-31-40-254 downloads)$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz  
--2024-10-02 06:17:24-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz  
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00:f03c:92ff:fef7:45ce  
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 11333414 (11M) [application/x-gzip]  
Saving to: 'nagios-4.4.6.tar.gz'  
  
nagios-4.4.6.tar.gz      0%[=====] 495.62K 2.40MB/s  
nagios-4.4.6.tar.gz    4%[==>] 3.26M 7.99MB/s  
nagios-4.4.6.tar.gz  30%[=====] 6.91M 11.0MB/s  
nagios-4.4.6.tar.gz  63%[=====] 10.46M 12.6MB/s  
nagios-4.4.6.tar.gz  96%[=====] 10.81M 12.9MB/s  in 0.8s  
  
2024-10-02 06:17:25 (12.9 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]  
[ec2-user@ip-172-31-40-254 downloads]$ ]
```

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

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

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

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

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

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./configure --with-command-group=nagcmd  
checking for a BSD-compatible install... /usr/bin/install -c  
checking build system type... x86_64-pc-linux-gnu  
checking host system type... x86_64-pc-linux-gnu  
checking for gcc... gcc  
checking whether the C compiler works... yes  
checking for C compiler default output file name... a.out  
checking for suffix of executables...  
checking whether we are cross compiling... no  
checking for suffix of object files... o  
checking whether we are using the GNU C compiler... yes  
checking whether gcc accepts -g... yes  
checking for gcc option to accept ISO C89... none needed  
checking whether make sets $MAKE... yes  
checking whether ln -s works... yes  
checking for strip... /usr/bin/strip  
checking how to run the C preprocessor... gcc -E  
checking for grep that handles long lines and -e... /usr/bin/grep  
checking for egrep... /usr/bin/grep -E  
checking for ANSI C header files... yes  
checking whether time.h and sys/time.h may both be included... yes  
checking for sys/wait.h that is POSIX.1 compatible... yes  
checking for sys/types.h... yes  
checking for sys/stat.h... yes  
checking for stdlib.h... yes  
checking for string.h... yes  
checking for memory.h... yes  
checking for strings.h... yes  
checking for inttypes.h... yes  
checking for stdint.h... yes  
checking for unistd.h... yes  
checking arpa/inet.h usability... yes  
checking arpa/inet.h presence... yes  
checking for arpa/inet.h... yes
```

12. Compile the source code. make all

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

```

*** Support Notes *****

If you have questions about configuring or running Nagios,
please make sure that you:

- Look at the sample config files
- Read the documentation on the Nagios Library at:
  https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:

- What version of Nagios you are using
- What version of the plugins you are using
- Relevant snippets from your config files
- Relevant error messages from the Nagios log file

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

Enjoy.

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

./sudo make install sudo make

install-init sudo make installconfig

sudo make installcommandmode

```

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

*** Config files installed ***

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

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

*** External command directory configured ***

```

14. Edit the config file and change the email address. sudo nano /usr/local/nagios/etc/objects/contacts.cfg

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

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

}

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

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

define contactgroup {
    # Help      ^C Write Out   ^W Where Is   ^K Cut          ^A Execute   ^C Location   M-1 Undo   M-A Set Mark   M-] To Bracket   M-Q Previous
    X Exit      ^R Read File   ^X Replace   ^V Paste       ^Y Justify   ^G Go To Line   M-2 Redo   M-C Copy      M-` Where Was   M-W Next
}
```

15. Configure the web interface.

sudo make install-webconf

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

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

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

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

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

17. Restart Apache

sudo systemctl restart httpd

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

cd ~/downloads tar zxvf

nagiosplugins-2.3.3.tar.gz cd

nagiosplugins-2.3.3

```
[ec2-user@ip-172-31-40-254 nagios-4.1.6]$ sudo systemctl restart httpd
[nagios@ip-172-31-40-254 nagios-4.1.6]$ cd ~/downloads
tar xvzf nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile_in
nagios-plugins-2.3.3/perlmods/Module-MetaData-0.00014.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.00014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/class-Accessor-0.34.tar.gz
nagios-plugins-2.3.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/#ABOUT-NLS
nagios-plugins-2.3.3/#AUTHORS
nagios-plugins-2.3.3/#ChangeLog
nagios-plugins-2.3.3/#ConfigFile_in
nagios-plugins-2.3.3/#config.h.in
nagios-plugins-2.3.3/#changelog
nagios-plugins-2.3.3/#AUTHORS
nagios-plugins-2.3.3/lib/
nagios-plugins-2.3.3/lib/parse_ini.h
nagios-plugins-2.3.3/lib/extr_opts.c
nagios-plugins-2.3.3/lib/Makefile.in
```

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

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

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

19. Compile and install plugins sudo chkconfig --add nagios sudo chkconfig nagios on

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

```

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

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.

```

20. Check the status of Nagios

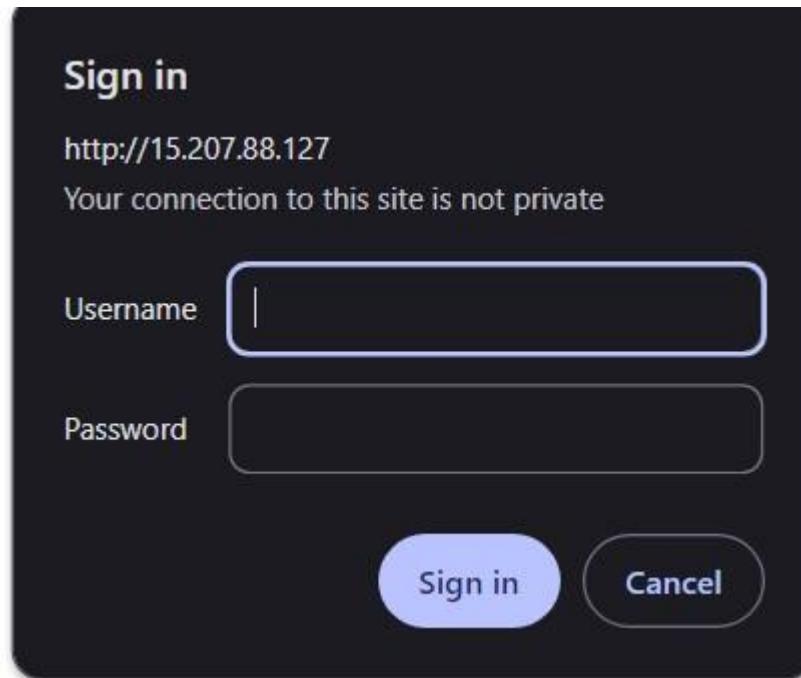
```

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

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

```

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



Not secure 15.207.88.127/nagios/

Nagios® Core™

Process running with PID 90965

Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Page Tour

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Grid

Problems

- Services
- (Unresolved)
- 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

Advance DevOps Exp 10

Name : Rohan Lalchandani

Class : D15A Roll no : 25

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

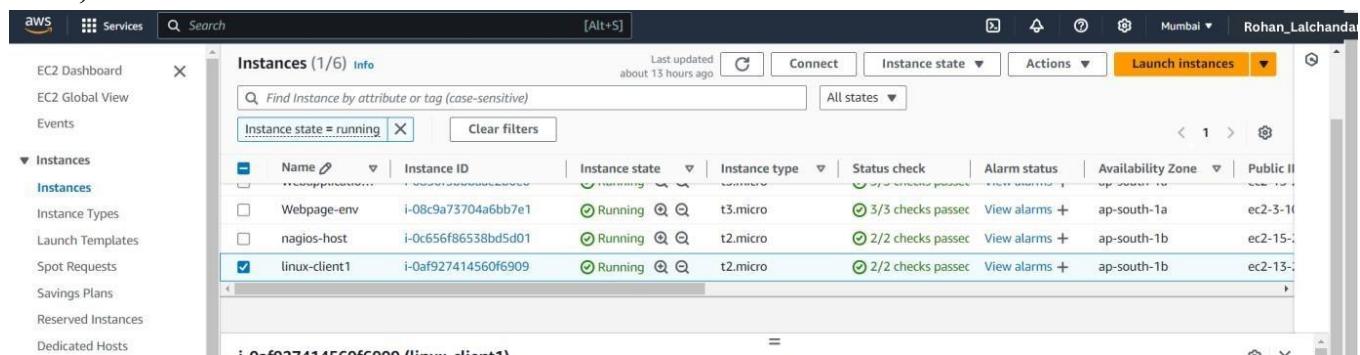
Procedure:-

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

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

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

Now, create a new EC2 instance on AWS



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

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

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

Sudo su

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

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

Copy localhost.cfg file to the mentioned location

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

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

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

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

```

GNU nano 7.2                                     /usr/local/nagios/etc/nagios.cfg
#####
#
# HOST DEFINITION
#
#####
# Define a host for the local machine

define host {

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

    host_name    linuxserver
    alias        linuxserver
    address      52.207.253.18
}

#####
#
# HOST GROUP DEFINITION

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

```

Note - Here replace hostname with linuxserver

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

Add the following line to the nagios.cfg file

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

```

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

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

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

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

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

```

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

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

```
License: GPL
```

```
Website: https://www.nagios.org
```

```
Reading configuration data...
```

```
    Read main config file okay...
```

```
    Read object config files okay...
```

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

```
Checking objects...
```

```
    Checked 16 services.
```

```
    Checked 2 hosts.
```

```
    Checked 2 host groups.
```

```
    Checked 0 service groups.
```

```
    Checked 1 contacts.
```

```
    Checked 1 contact groups.
```

```
    Checked 24 commands.
```

```
    Checked 5 time periods.
```

```
    Checked 0 host escalations.
```

```
    Checked 0 service escalations.
```

```
Checking for circular paths...
```

```
    Checked 2 hosts
```

```
    Checked 0 service dependencies
```

```
    Checked 0 host dependencies
```

```
    Checked 5 timeperiods
```

```
Checking global event handlers...
```

```
Checking obsessive compulsive processor commands...
```

```
Checking misc settings...
```

```
Total Warnings: 0
```

```
Total Errors: 0
```

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

Now restart the service by using this command

```
service nagios restart
```

```

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

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

```

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

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

```

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

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

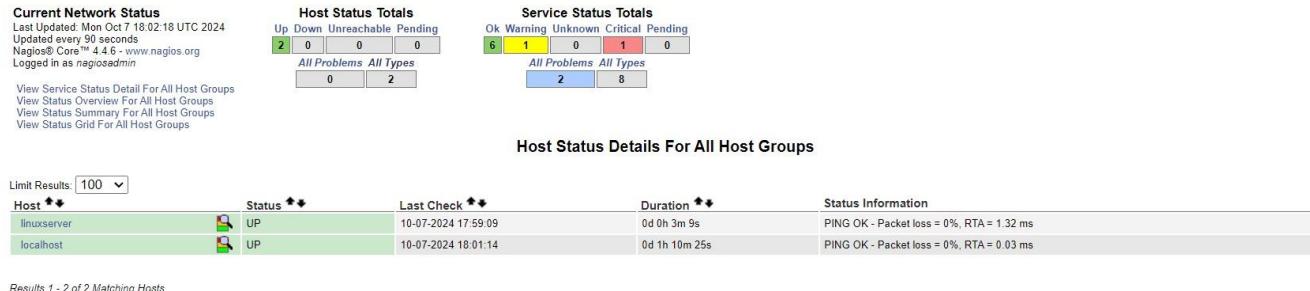
```

sudo nano /etc/nagios/nrpe.cfg

Now restart nrpe server by using this command

sudo systemctl restart nagios-nrpe-server

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



Adv. DevOps Experiment 11

Name : Rohan Lalchandani

Class : D15A Roll no : 25

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

AWS Lambda

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

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

Features of AWS Lambda

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

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket.

And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We’ll go over this in detail later on in this guide.

Execution Model

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

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code.

AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

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

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

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

Common Use Cases for Lambda

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

1. Individual tasks run for a short time;
 2. Each task is generally self-contained;
 3. There is a large difference between the lowest and highest levels in the workload of the application.
- Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request.

Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to

the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data

events. For example, you could employ Lambda to do some work every time an item in DynamoDB is created or updated, thus making it a good fit for things like notifications, counters and analytics.

Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1. Choose a Function Creation Method: Select Author from scratch.

2. Configure the Function:

Function name: Enter a name for your function (e.g., MyFirstLambda).

Runtime: Choose Python 3.x (the latest available version).

Permissions: Choose Create a new role with basic Lambda permissions (this creates a role with the necessary permissions).

3. Click on Create function.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The top navigation bar includes 'Services', a search bar, and account information for 'Rohan_Lalchandani'. The main page title is 'Create function' with an 'Info' link. Below the title, a note says 'Choose one of the following options to create your function.' Three options are listed: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section contains fields for 'Function name' (set to 'Lambda-Func'), 'Runtime' (set to 'Python 3.12'), and 'Architecture' (set to 'x86_64'). The 'Permissions' section is partially visible at the bottom. On the right side, there's a sidebar titled 'Create a simple web app' with a 'Learn more' link and a 'Start tutorial' button. The sidebar also contains a note about common use cases in AWS Lambda.

Step 2: Write Your Lambda Function Code

In the Function code section, you will see a code editor. Replace the default code with the following Python code:

```
python Copy code def lambda_handler(event, context): #  
This function returns a greeting message name =  
event.get('name', 'World') return {  
    'statusCode': 200, 'body': f'Hello, {name}!' }
```

This function reads a name from the event and returns a greeting message. If no name is provided, it defaults to "World".

The screenshot shows the AWS Lambda console interface. On the left, there's a navigation bar with 'Services' and a search bar. The main area shows a function named 'Lambda-Func'. The 'Function overview' tab is selected, displaying a diagram with a single function node labeled 'Lambda-Func', 'Layers (0)', and buttons for '+ Add trigger' and '+ Add destination'. To the right, there are sections for 'Description', 'Last modified' (2 minutes ago), 'Function ARN' (arn:aws:lambda:ap-south-1:529088256210:function:Lambda-Func), and 'Function URL' (Info). At the top right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. A 'Tutorials' sidebar on the right provides links to 'Create a simple web app' and 'Learn more'.

The screenshot shows the code editor for the 'lambda_function' file. The code is as follows:

```
1 def lambda_handler(event, context):  
2     # This function returns a greeting message  
3     name = event.get('name', 'World')  
4     return {  
5         'statusCode': 200,  
6         'body': f'Hello, {name}!'  
7     }  
8  
9
```

Step3: 1. Configure a Test Event:

Click on the Test button.

In the Configure test event dialog, give your event a name (e.g., TestEvent). Replace the default JSON with the following:

```
{  
  "name": "Lambda User"  
}
```

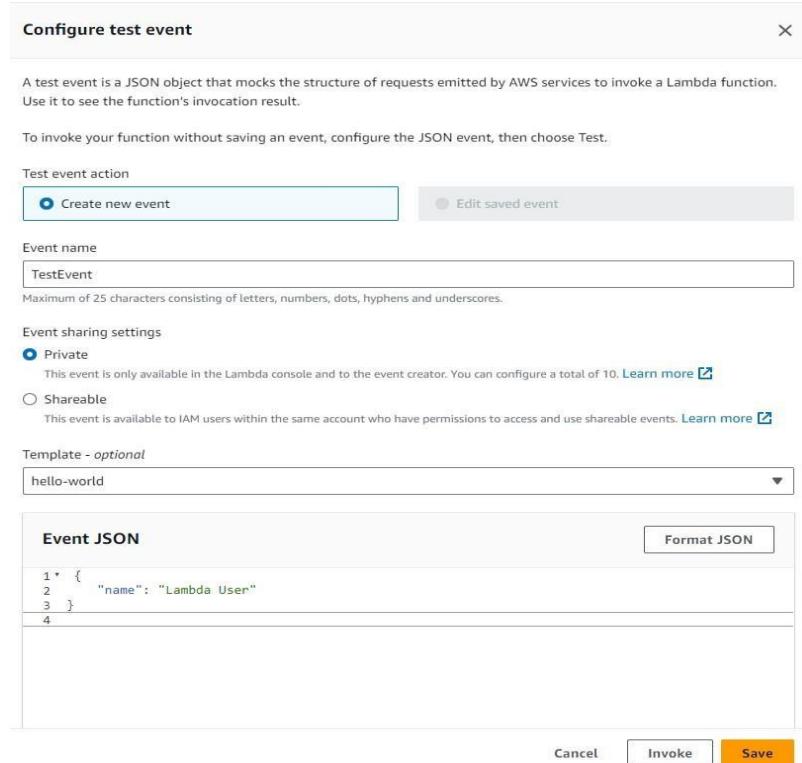
2. Run the Test:

Click on the Test button again to execute your Lambda function.

You should see the execution results below the code editor, including the response: json

Copy code

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



The screenshot shows the AWS Lambda function editor interface. On the left, there's a sidebar with 'Environment' and 'CloudShell'. The main area has tabs for 'Code source' and 'Info'. The 'Code source' tab is active, showing a code editor with Python code for a lambda function named 'lambda_function'. The code defines a handler 'lambda_handler' that returns a JSON response with 'statusCode' 200 and 'body' 'Hello from Lambda!'. Below the code editor are sections for 'Execution results', 'Function Logs', and 'Request ID'. The 'Execution results' section shows a successful execution with status 'Succeeded', max memory used '32 MB', and time '1.55 ms'. The 'Function Logs' section shows log entries for START, END, and REPORT events. The 'Request ID' section shows the unique request identifier.

The screenshot shows the AWS Lambda function configuration page. At the top, it displays the function name 'function:Lambda-Func' and a 'Function URL' with a link. Below this, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected, showing the same code editor as the previous screenshot. The code editor displays the same Python code for the 'lambda_function' handler.

Conclusion:

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

Adv. DevOps Exp. 12

Name : Rohan Lalchandani

Class : D15A Roll no : 25

Step 1: Open the IAM (user)

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and various navigation links like Dashboard, User groups, Users, Roles, Policies, Identity providers, Account settings, Access analyzer, Archive rules, Analyzers, Settings, and Credential report. The main area is titled 'Roles (6) Info' and contains a table with the following data:

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-service-role-2	AWS Service: elasticbeanstalk	40 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Link)	40 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Links)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Links)	-
myPythonLambdaFunction-role-a2x7el65	AWS Service: lambda	-
test-2-role	AWS Service: ec2	40 days ago

Below the table, there's a section titled 'Roles Anywhere' with a 'Manage' button.

Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

The screenshot shows the 'myPythonLambdaFunction-role-a2x7el65' role summary page. The sidebar is identical to the previous one. The main area has a 'Summary' section with details like Creation date (October 07, 2023, 16:05 (UTC+05:30)), Last activity (recent), ARN (arn:aws:iam::44753971928:role/service-role/myPythonLambdaFunction-role-a2x7el65), and Maximum session duration (1 hour). Below this is a 'Permissions' tab with a 'Permissions policies (1) Info' section. It shows one policy attached: 'S3-ReadOnly'. There are buttons for 'Edit', 'Delete', 'Add permissions', 'Attach policies', and 'Create inline policy'.

S3-ReadOnly

The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The path is IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions. The title is 'Attach policy to myPythonLambdaFunction-role-a2x7el65'. A sidebar on the left lists 'Current permissions policies (1)'. The main area is titled 'Other permissions policies (882)' and contains a search bar with 'S3read' and a filter 'All types' showing '1 match'. A single policy, 'AmazonS3ReadOnlyAccess', is listed as 'AWS managed' with a description 'Provides read only access to all bucket...'. At the bottom are 'Cancel' and 'Add permissions' buttons.

CloudWatchFull

The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The path is IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions. The title is 'Attach policy to myPythonLambdaFunction-role-a2x7el65'. A sidebar on the left lists 'Current permissions policies (2)'. The main area is titled 'Other permissions policies (881)' and contains a search bar with 'cloudwatchfull' and a filter 'All types' showing '2 matches'. Two policies are listed: 'CloudWatchFullAccess' and 'CloudWatchFullAccessV2', both described as 'AWS managed' with descriptions 'Provides full access to CloudWatch.' and 'Provides full access to CloudWatch.'. At the bottom are 'Cancel' and 'Add permissions' buttons.

After successful attachment of policy you will see something like this you will be able to see the updated policies.

The screenshot shows the AWS Identity and Access Management (IAM) console. A green banner at the top indicates that a policy has been successfully attached to a role. Below the banner, the 'Permissions' tab is selected in the navigation bar. The 'Permissions policies' section lists three managed policies: 'AmazonS3ReadOnlyAccess', 'AWSLambdaBasicExecutionRole', and 'CloudWatchFullAccess'. The 'Attached entities' column shows that each policy is attached to one entity.

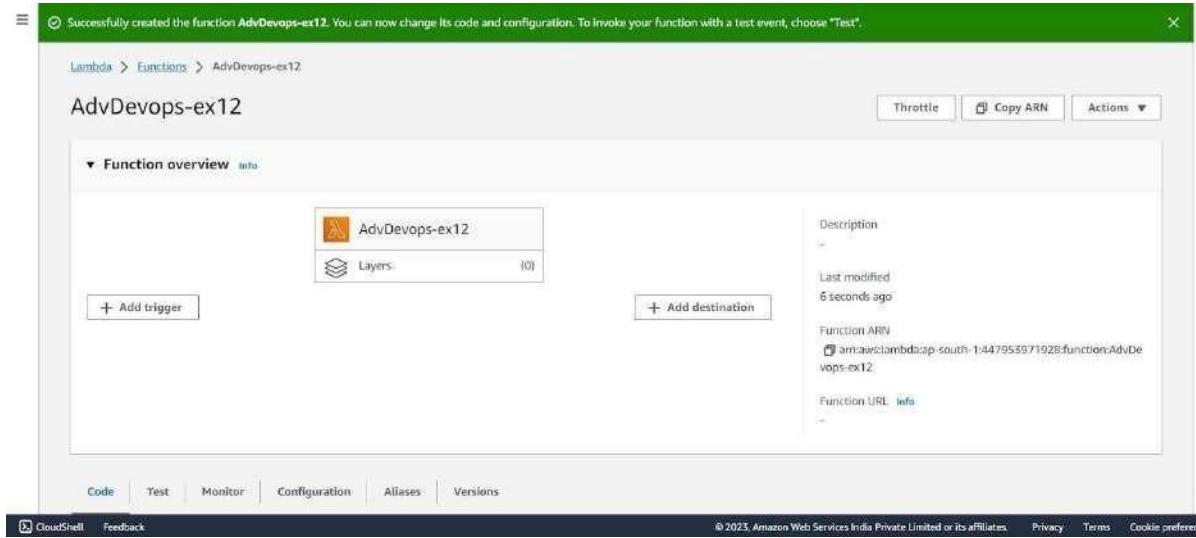
Step 3: Open up AWS Lambda and create a new Python function.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The first step, 'Author from scratch', is selected. The 'Basic information' section includes fields for 'Function name' (set to 'AdvDevops-ec12'), 'Runtime' (set to 'Python 3.11'), and 'Architecture' (set to 'x86_64'). Under 'Permissions', it notes that the function will receive an execution role with permissions to upload logs to Amazon CloudWatch Logs. The 'Create function' button is visible at the bottom right.

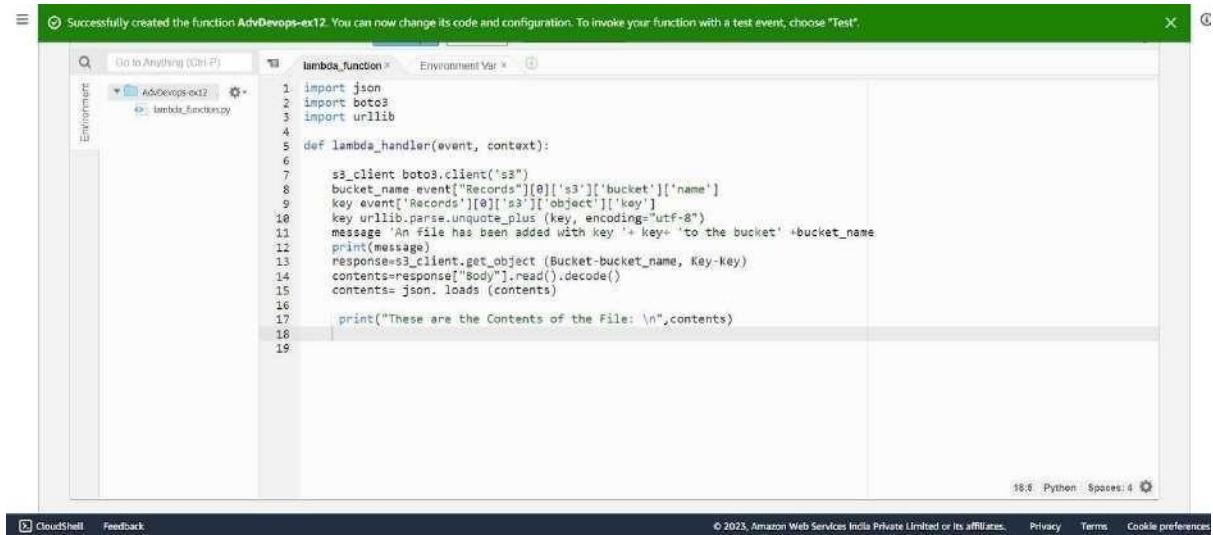
Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.

The screenshot shows the 'Create function' wizard in the AWS Lambda console, specifically the 'Execution role' step. It asks to choose an existing role with permission to upload logs to Amazon CloudWatch Logs. The dropdown menu shows the previously created role: 'service-role/myPythonLambdaFunction-role-a2x7el65'. The 'Create function' button is visible at the bottom right.

Step 4: The function is up and running.



Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.



Step 6: Click on Test and choose the 'S3 Put' Template.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a keyboard shortcut '[Alt+S]'. A green success message box says: 'Successfully created the function AdvDevops-ex12. You can now change its code and configuration. To invoke your function, click on the Test tab.' Below the message, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected, showing the 'Code source' section with an 'Info' link. A toolbar above the code editor includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is highlighted), 'Deploy', and a status message 'Changes not deployed'. The code editor displays a file named 'lambda_function.py' with the following content:

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
```

To the left of the code editor is a sidebar with 'Environment' settings, showing a folder named 'AdvDevops-ex12' containing a file 'lambda_function.py'. Above the code editor is a search bar with 'lambda_function' and a 'Configure test event' button with 'Ctrl-Shift-C' keyboard shortcut.

A modal window titled 'Configure test event' is open in the foreground. It contains instructions: 'A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.' Below this, there's a 'Test event action' section with two options: 'Create new event' (selected) and 'Edit saved event'. The 'Event name' field is set to 'test'. A note below it says: 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.' Under 'Event sharing settings', the 'Private' option is selected, with a note: 'This event is only available in the Lambda console and to the event creator. You can configure a total of 10.' The 'Shareable' option is also present with a note: 'This event is available to IAM users within the same account who have permissions to access and use shareable events.' In the 'Template - optional' section, 's3-put' is selected from a dropdown. At the bottom of the modal are buttons for 'Format JSON', 'Cancel', 'Invoke' (disabled), and 'Save'.

And Save it.

Step 7: Open up the S3 Console and create a new bucket.

Buckets (3) Info
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
elasticbeanstalk-ap-south-1-442953971928	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 7, 2023, 14:24:02 (UTC+05:30)
www.hellorechnera.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:05:54 (UTC+05:30)
www.htmlwebsite.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:49:06 (UTC+05:30)

View Storage Lens dashboard ④

C Copy ARN Empty Delete Create bucket

Q Find buckets by name

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Step 8: With all general settings, create the bucket in the same region as the function.

Amazon S3 > Buckets > Create bucket

Create bucket Info
Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.

AWS Region

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

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

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Step 9: Click on the created bucket and under properties, look for events.

Event notifications (0)
Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
				No event notifications

Choose Create event notification to be notified when a specific event occurs.

[Create event notification](#)

Amazon EventBridge
For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or see [EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket

Off

Transfer acceleration
Use an accelerated endpoint for faster data transfers. [Learn more](#)

Transfer acceleration
Disabled

Click on Create Event Notification.

Step 10: Mention an event name and check Put under event types.

General configuration

Event name
 Event name can contain up to 255 characters.

Prefix - optional
Limit the notifications to objects with key starting with specified characters.

Suffix - optional
Limit the notifications to objects with key ending with specified characters.

Event types
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

All object create events
s3:ObjectCreated:
 Put
s3:ObjectCreated:Put

Post
s3:ObjectCreated:Post

Choose Lambda function as destination and choose your lambda function and save the changes.

Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

Destination
Choose a destination to publish the event. [Learn more](#)

Lambda function
Run a Lambda function script based on S3 events.

SNS topic
Fanout messages to systems for parallel processing or directly to people.

SQS queue
Send notifications to an SQS queue to be read by a server.

Specify Lambda function

Choose from your Lambda functions

Enter Lambda function ARN

Lambda function
AdvDevops-ex12

Cancel **Save changes**

CloudShell Feedback © 2023, Amazon Web Services

Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.

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

Lambda > Functions > Lambda-Func

Lambda-Func

The trigger elasticbeanstalk-ap-south-1-529088256210 was successfully added to function Lambda-Func. The function is now receiving events from the trigger.

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

Diagram [Template](#)

Description
Last modified 3 days ago

Function ARN arn:aws:lambda:ap-south-1:529088256210:function:Lambda-Func

Function URL [Info](#)

Actions Throttle, Copy ARN, Actions

Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

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

[Learn more](#) [Start tutorial](#)

Code | Test | Monitor | Configuration | Aliases | Metrics

Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.

Step 14: Select the dummy data file from your computer and click Upload.

The screenshot shows the AWS S3 'Upload' interface. At the top, the navigation bar includes 'Services' and a search bar. Below it, the path 'Amazon S3 > Buckets > advopssexp12 > Upload' is visible. The main area is titled 'Upload' with a sub-link 'Info'. A large text box with a dashed border is labeled 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this, a table titled 'Files and folders (1 Total, 89.0 B)' lists 'dummy.json' with details: Name, Folder, Type (application/json), and Size (89.0 B). Buttons for 'Remove', 'Add files', and 'Add folder' are at the top of the table. A search bar 'Find by name' is also present. The 'Destination' section shows the target as 's3://advopssexp12'. At the bottom, there are links for 'CloudShell', 'Feedback', and a copyright notice: '© 2023, Amazon Web Services India Private Limited or its affiliates'.

Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

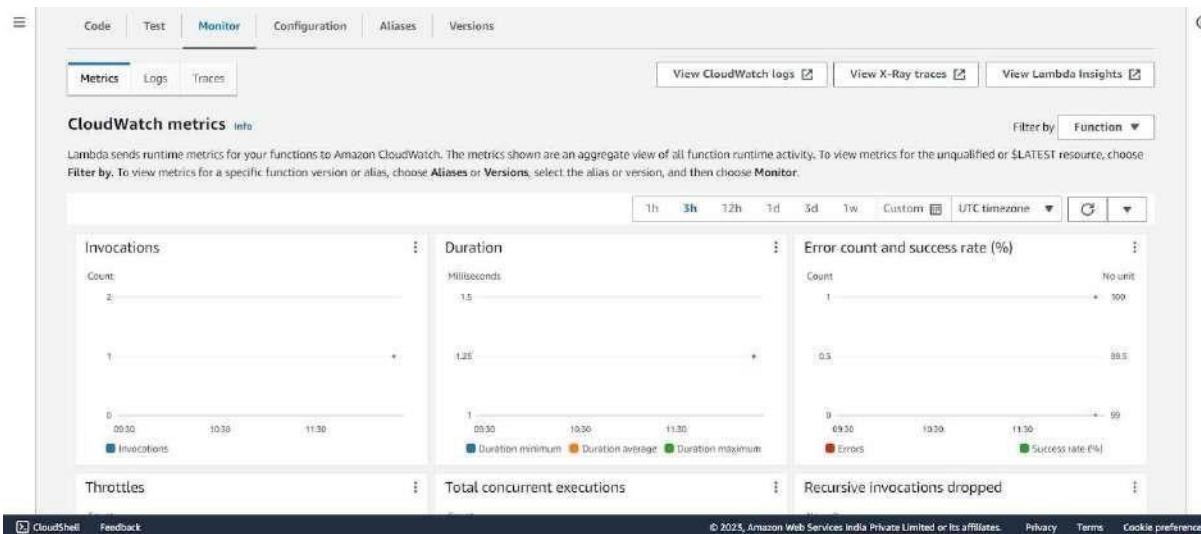
The screenshot shows the AWS Lambda 'Event JSON' editor. The JSON code is displayed in a monospaced font with line numbers from 10 to 38. The code defines an event object with fields like 'principalId', 'requestParameters', 'responseElements', and 's3'. The 's3' field contains detailed information about the uploaded file, including 'key' (test%2Fkey), 'size' (1024), 'eTag' (0123456789abcdef0123456789abcdef), and 'sequencer' (0A182C3D4E5F678901). A 'Format JSON' button is located in the top right corner of the editor.

```

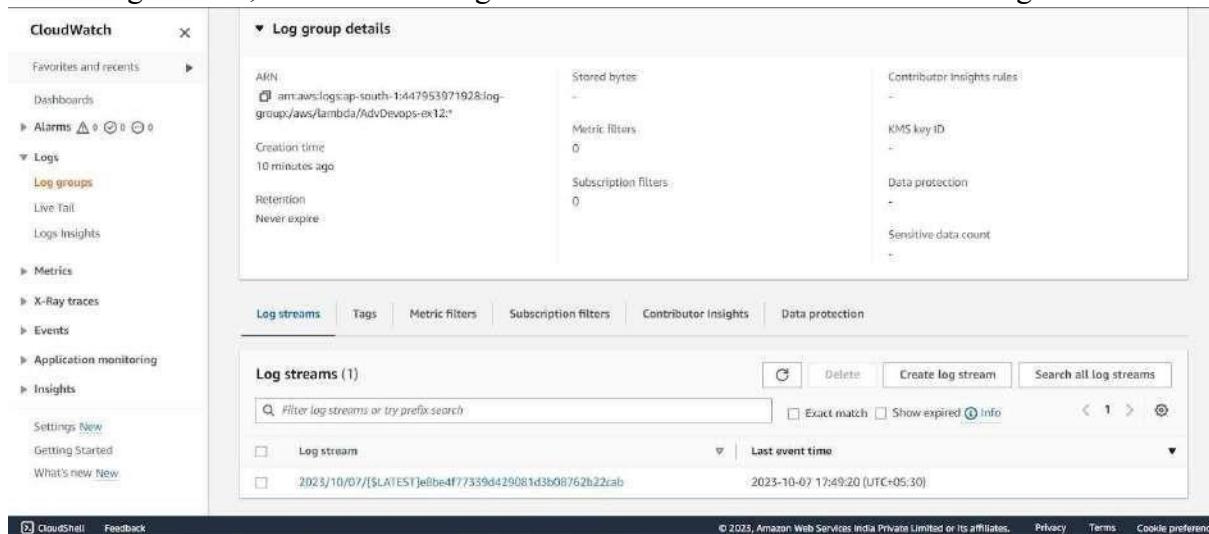
10     "principalId": "EXAMPLE"
11   },
12   "requestParameters": {
13     "sourceIPAddress": "127.0.0.1"
14   },
15   "responseElements": {
16     "x-amz-request-id": "EXAMPLE123456789",
17     "x-amz-id-2": "EXAMPLE123/56789bcdefghijklmbdaisawesome/mnopqrstuvwxyzABCDEFGH"
18   },
19   "s3": {
20     "s3SchemaVersion": "1.0",
21     "configurationId": "testConfigRule",
22     "bucket": {
23       "name": "advopssexp12",
24       "ownerIdentity": {
25         "principalId": "EXAMPLE"
26       },
27       "arn": "arn:aws:s3:::advopssexp12"
28     },
29     "object": {
30       "key": "test%2Fkey",
31       "size": 1024,
32       "eTag": "0123456789abcdef0123456789abcdef",
33       "sequencer": "0A182C3D4E5F678901"
34     }
35   }
36 }
37 ]
38 }

```

Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar has a 'Logs' section expanded, showing 'Log groups'. The main area shows a table of log events for the '/aws/lambda/Lambda-Func' log group. The table has columns for 'Timestamp' and 'Message'. The first message is 'No older events at this moment. [Retry](#)'. Subsequent messages show Lambda runtime events like INIT, START, REPORT, and END requests, each with a unique RequestId and timestamp.

Timestamp	Message
2024-10-11T05:23:33.473Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:ap-south-1::runtime:188d9ca2e2714ff5637bd2bb..
2024-10-11T05:23:33.568Z	START RequestId: 4662c213-1c33-4099-a4ac-ef71af92f36a Version: \$LATEST
2024-10-11T05:23:33.587Z	END RequestId: 4662c213-1c33-4099-a4ac-ef71af92f36a
2024-10-11T05:23:33.587Z	REPORT RequestId: 4662c213-1c33-4099-a4ac-ef71af92f36a Duration: 1.93 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memor..
2024-10-11T05:23:55.230Z	START RequestId: fb1457e5-f5cd-4787-bc2b-141c111b7271 Version: \$LATEST
2024-10-11T05:23:55.247Z	END RequestId: fb1457e5-f5cd-4787-bc2b-141c111b7271
2024-10-11T05:23:55.247Z	REPORT RequestId: fb1457e5-f5cd-4787-bc2b-141c111b7271 Duration: 1.55 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memor..
2024-10-11T05:25:23.104Z	START RequestId: lbbbb172-9698-4cf7-912b-d7f9d38c7748 Version: \$LATEST
2024-10-11T05:25:23.106Z	END RequestId: lbbbb172-9698-4cf7-912b-d7f9d38c7748
2024-10-11T05:25:23.106Z	REPORT RequestId: lbbbb172-9698-4cf7-912b-d7f9d38c7748 Duration: 1.63 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memor..

Conclusion: Thus, we have created a Lambda function which logs “An Image has been added” once you add an object to a specific bucket in S3.

Name : Rohan

Name : Rohan Lalchandani

Class : DIS A Roll no. 25

Subject : Adv. DevOps (Exp - Assignment - 1)

Date : 27/09/2024

(83
85)

(45)

Video streaming using S3 Bucket.

Step 1 : Set up your S3 Bucket.

- 1.(a) Sign in to AWS Management Console
- (b) Navigate to S3 & click on "Create Bucket".
- (c) Name your bucket (eg my-video-streaming-bucket) and choose a region.
- (d) Ensure the bucket name is unique across all of S3.

2. Configure Bucket Permissions

- (a) Set up the bucket to be private to secure your content.
- (b) Use AWS Identity & Access Management (IAM) to create a user with permissions to access S3.
- (c) Optionally, configure a bucket policy or access control lists (ACLs) if you want public access for streaming.

~~Step 2 : Configure Bucket Permissions~~

~~Step 2 : Upload your video files~~

1. Upload Video files

- (a) In S3 bucket, click on "Upload" and choose video files.
- (b) Consider using formats like MP4 for better compatibility with streaming.

2. Set Metadata for streaming (optional)

(46)

Step 3 : To step up Video Streaming :

1. Use Amazon CloudFront

- (a) Set up a CloudFront distribution to serve your video content more efficiently.
- (b) Point the CloudFront origin to your S3 bucket.
- (c) Configure caching behaviors based on requirements.

2. Set protocol

- (a) Choose whether to use HLS (HTTP Live Streaming) or DASH (Dynamic Adaptive Streaming over HTTP).
- (b) Use AWS Media Convert to convert your video files into streaming-compatible formats if necessary.

Step 4 : Integrate Video Player

1. Select a video player.

- (a) Choose a web video player that supports streaming (e.g. video.js, JW Player).

2. Embed the player.

- (a) Use HTML to embed the video player on your website.

- (b) Set the source of the video to your CloudFront URL or S3 URL.

Step 5 : Test your setup

1. Test video playback:

Access website & play video to ensure everything works correctly.

Discuss BMW & Hot Star Case studies using AWS.

BMW Case Study on AWS

BMW used AWS to accelerate innovation, enhance customer experiences and create data-driven services. Their use case centers on the BMW Connected Drive & BMW Cloud Data Hub.

Key Components

1. BMW Connected Drive :

Connected Services : AWS allows BMW to offer real-time connected services (including in-car navigation, telematics & driver assistance).

2. Data Processing : BMW uses AWS to process millions of requests daily, analyzing large amounts of data generated by vehicles in real-time.

3. Scalability : The use of Amazon S3, Amazon EC2, and other AWS services enable BMW to scale infrastructure based on growing number of connected vehicles.

2. BMW Cloud Data Hub :

1. Global Data Platform : BMW built its cloud data hub on AWS to consolidate manage & analyse vehicles & user data across the globe.

2. Amazon S3 & Amazon Redshift :

BMW uses Amazon S3 for scalable storage & Amazon Redshift for analysing vast datasets related to vehicle performance.

(13)

Benefits :

1. Reduced Development Time
2. Global Scale & Availability
3. Cost-effectiveness.

Hotstar (Disney + Hotstar) : Case Study on AWS

Hotstar, one of India's largest streaming platforms, used AWS to stream high definition video to millions of concurrent users especially during live events such as IPL cricket matches. Their main challenge was scalability during the events, where millions of users could be streaming simultaneously.

Key Components :

1. Scalability During peak events

• Elastic Compute (Amazon EC2) :

~~Hotstar uses Amazon EC2 for elastic, scalable computer power, especially during peak streaming events like IPL or cricket matches.~~

2. Auto Scaling :

AWS auto scaling allows hotstar to dynamically adjust its infrastructure based on user demand.

2. Content Delivery Using Amazon EC2.

+ Global CDN : Hotstar uses Amazon Cloudfront to deliver video content worldwide, ensuring low latency & high throughput.

+ Edge Locations : By caching content at AWS edge locations across the globe, Hotstar can reduce the load on its servers & deliver content more efficiently.

3. Data Analytics & ML

(a) Amazon Kinesis : Uses this to monitor real time viewing statistics & engagement metrics during live streams.

(b) Amazon S3 & Redshift :

Hotstar stores user data and analytics in Amazon S3 and processes this data using Amazon Redshift to derive insights on viewer preference.

Benefits :

- 1) Handling Massive Traffic
- 2) Low Latency Streaming
- 3) Faster Time to Market
- 4) Improved User Experience.

Q.3.

Why Kubernetes?

Kubernetes (often abbreviated as K8s) is an open-source platform designed to automate deploying, scaling and operating containerized applications. Containers help package applications and their dependencies, ensuring consistency across environments. Kubernetes takes containerization further by orchestrating the containers, making it easier to manage large-scale distributed systems.

Advantages :

1. Scalability : It allows horizontal scaling of containerized applications, making it easy to handle fluctuating workloads & large-scale deployment.
2. Portability : Kubernetes is a platform-agnostic meaning applications can run on any cloud provider.
3. Self Healing : Kubernetes automatically monitors the health of nodes & containers, restarting failed containers & redistributing workloads to ensure service continuity.
4. Automation : Kubernetes automates various operational tasks such as container scheduling, scaling & load balancing reducing manual intervention & simplifying management.

Disadvantages :

1. Complexity : Kubernetes has a steep learning curve & requires considerable expertise to manage effectively. The system is complex especially for small teams or beginners in container orchestration.
2. Overhead : Running Kubernetes adds operational overhead, requiring significant resources to maintain infrastructure like control planes, nodes & networks.
3. Monitoring & Debugging : Debugging issues in Kubernetes can be difficult due to its distributed nature.

How Adidas uses Kubernetes ?

Adidas, the global sportswear company adopted Kubernetes as a part of its digital transformation strategy, especially to modernize its e-commerce platform and build a cloud-native infrastructure.

Here's how Adidas uses Kubernetes :

1. Microservices architecture :-

Adidas moved from a monolithic application architecture to microservices using Kubernetes. This allows them to develop, deploy & scale individual services independently.

2. Scalability -

Addidas' ecommerce platform experiences high traffic during product launches or sales. Kubernetes helps scale applications on-demand by automatically increasing resources based on traffic load.

3. Cloud Native & Multi-Cloud Strategy

Addidas uses Kubernetes to maintain cloud-native infrastructure. This abstraction layer helps them be cloud-agnostic, allowing deployments across multiple cloud providers.

Q.4. What is Nagios?

Nagios is an open-source monitoring tool designed to track performance, availability and health of IT infrastructure, including servers, network devices, applications & services. It helps administrators identify & resolve issues before they affect end users ensuring continuous service availability.

Key features :

- 1) Monitoring of infrastructure
- 2) Alerts
- 3) Reporting & logs
- 4) Plugins

How Nagios is used in E-services?

E-services refer to digital platforms / services over the internet such as e-commerce platforms, online banking or cloud-based applications.

1. Monitoring Service Availability:

- (a) Service Uptime monitoring - It continuously monitors the availability of e-services like web servers (eg. Apache, nginx)

(b) Web Application monitoring - For e-services, Nagios checks critical components.

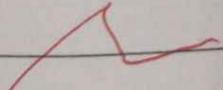
2. Network Monitoring

- (a) Tracking network health - It monitors network devices (routers, switches, etc) - bandwidth usage & response time to ensure good network performance.

3. Performance Monitoring

4. Security Monitoring

5. Alerting & Notification Systems.



Create a REST API with serverless framework.
Creating REST API with serverless framework is an efficient way to deploy serverless applications that can scale automatically without managing services. and serverless applications (i) Serverless framework :

A powerful tool that deployment of services and serverless applications across various cloud providers such as AWS, Azure and Google Cloud.

(ii) Serverless architecture :

This design model allows developers to build applications without worrying about underlying infrastructure enabling focus on code & business logic.

(iii) REST API : Representational state transfer is web architecture style for designing network applications.

Steps :

- 1) Install serverless framework
- 2) Creating a Node.js serverless project
- 3) To Create a REST API resource
- 4) Deploy the service.
- 5) Testing the API.
- 6) Storing data in Dynamo DB
- 7) Adding more functionalities like To list all, get candidates by ID
- 8) AWS IAM permissions : You need to ensure serverless framework serverless framework is given right permissions to interact with AWS resources.
- 9) Monitoring & maintenance : After deployment serverless framework provides service information like deployed endpoints, API key, log streams.

Q.2. Case study for SonarQube

Ans

i) SonarQube is an open-source platform for testing project quality. Use SonarQube to analyze code.

ii) It detects bugs, code smells and security vulnerabilities in project across various programming languages.

iii) Profile creation in SonarQube

Quality profiles are essential configuration that define rules applied during code analysis. Each project has a quality profile for every language with default being 'Sonar way' profile.

iv) Using SonarCloud to analyze github code

SonarCloud is counterpart cloud-based part of SonarQube that directly integrates with Github, Bitbucket, Azure, etc. To get started with SonarCloud via Github sign up via SonarCloud product page and connect your Github organization account. Once connected, SonarCloud will mirror your Github setup with each project corresponding to Github repository. Automatic analysis happens automatically in SonarCloud.

v) Sonarlint in Java IDE:

Sonarlint is an IDE that performs on-the-fly code analysis as you write code. It helps developers detect bugs, security vulnerabilities and code smells directly in development environment such as Eclipse, IntelliJ, etc.

IntelliJ, Idea or Eclipse.

- 4) Analyzing Python project with Sonargube :
Sonargube supports python test coverage reporting but it requires 3rd party tool like coverage.py to generate the coverage part.
To enable coverage adjust your build process so that process tool runs before sonarScanner and ensure coverage report file is saved in different path.
- 5) Analyzing Node.js project with Sonargube :
For Node.js project sonargube can analyze Javascript and TypeScript code. Similar to python setup you can configure sonargube to analyze node.js projects by installing appropriate plugins.

Implementing "a self service" infrastructure model using terraform can transform how large organizations manage their infrastructure independently, organizations can enhance efficiency, reduce bottlenecks and ensure compliance with established needs.

Benefits of using terraform :

- 1) Modularity & reusability
- 2) Standardization
- 3) Increased efficiency
- 4) Integration with ticketing systems