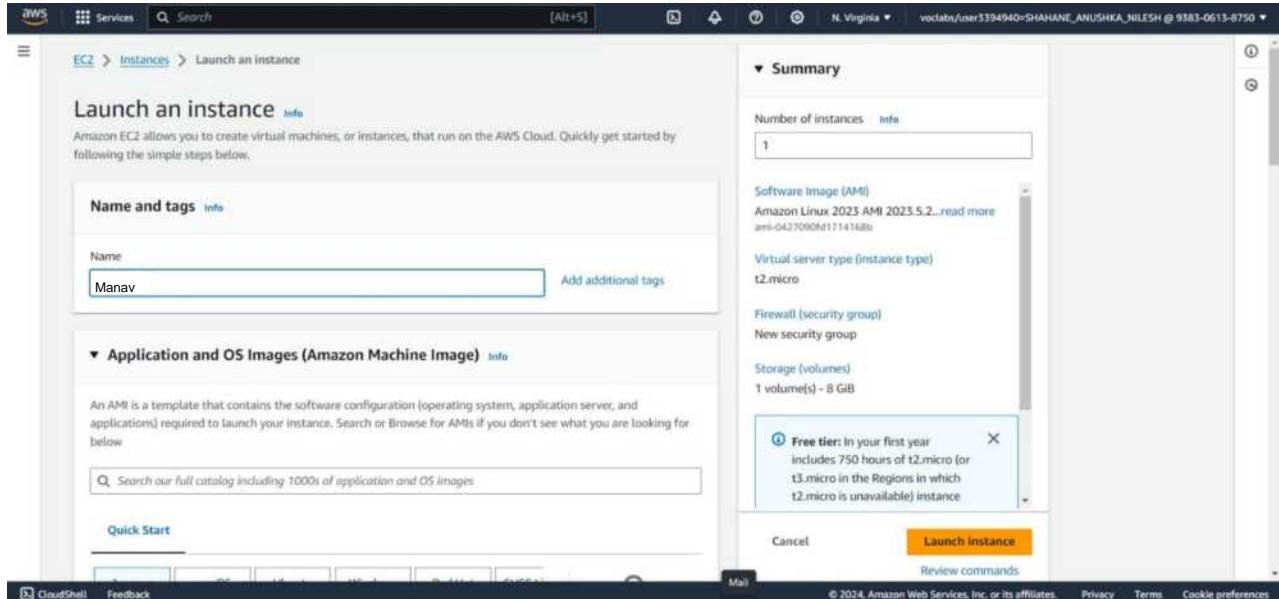


ADVANCE DEVOPS EXPERIMENT 1



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

Manav

Add additional tags

Quick Start



Amazon
Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Li

🔍

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

Description

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

Architecture AMI ID

64-bit (x86) ▾

ami-04a81a99f5ec58529

Verified provider

▼ 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

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

ⓘ 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

The screenshot shows the AWS EC2 Instances Launch an instance success page. At the top, there is a green success banner with the message "Successfully initiated launch of instance (i-0df3904aed5f9e9d9)". Below the banner, there is a "Launch log" link. A "Next Steps" section follows, containing a search bar and a numbered list of 6 items. The first item is "Create billing and free tier usage alerts". The second item is "Connect to your instance". The third item is "Connect an RDS database". The fourth item is "Create EBS snapshot policy". The fifth item is "Create a new RDS database". The sixth item is "Learn more".

This screenshot shows the same EC2 Instances Launch an instance success page as above, but it includes a "Launch log" section. The log details the steps taken during the instance launch: "Initializing requests" (Succeeded), "Creating security groups" (Succeeded), "Creating security group rules" (Succeeded), and "Launch initiation" (Succeeded). The rest of the page structure is identical to the first screenshot.

U anavPunjabi D15A 45

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main area is titled 'Instances (1) Info' and shows a table with one row. The row contains columns for Name (Manav), Instance ID (i-0df3904aed5f9e9d9), Instance state (Running), Instance type (t2.micro), Status check (Initializing), Alarm status (View alarms), Availability Zone (us-east-1b), and Public IP (ec2-54-197-204-120.co...). A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and has a dropdown set to 'All states'. There are also 'Connect', 'Actions', and 'Launch instances' buttons.

This screenshot shows the details for the instance 'i-0df3904aed5f9e9d9' (Manav). At the top, it shows the instance summary with fields like Public IPv4 address (54.197.204.120), Instance state (Running), and Instance type (t2.micro). Below this, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's a section for 'Instance summary' with fields for Instance ID (i-0df3904aed5f9e9d9), IPv6 address (-), Hostname type (IP name: ip-172-31-42-176.ec2.internal), Answer private resource DNS name (IPv4 (A)), and Auto-assigned IP address. To the right, there are sections for Public IPv4 address (54.197.204.120), Private IPv4 addresses (172.31.42.176), Public IPv4 DNS (ec2-54-197-204-120.compute-1.amazonaws.com), and Elastic IP addresses (AWS Compute Optimizer finding).

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

ubuntu@ip-172-31-42-176:~$ ls
ubuntu@ip-172-31-42-176:~$ echo "hello"
hello
ubuntu@ip-172-31-42-176:~$ cat > myfile.txt
This is Advance devops lab
^C
ubuntu@ip-172-31-42-176:~$ cat myfile
cat: myfile: No such file or directory
ubuntu@ip-172-31-42-176:~$ cat myfile.txt
This is Advance devops lab
ubuntu@ip-172-31-42-176:~$
```

Hosting a static website using EC2 instance:

```
*** System restart required ***
Pending kernel upgrade!
Running kernel version:
  6.8.0-1009-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1012-aws.
Last login: Tue Jul 30 08:37:47 2024 from 18.206.107.28
ubuntu@ip-172-31-41-78:~$ sudo su
root@ip-172-31-41-78:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-41-78:/home/ubuntu# systemctl
```

i-0104434d25a50dc8d (Manav1)

PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78

```
[ 12917 /usr/sbin/apache2 -k start
[ 12919 /usr/sbin/apache2 -k start
[ 12921 /usr/sbin/apache2 -k start
```

```
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-78:/home/ubuntu# cd/var/www/html/
bash: cd/var/www/html/: No such file or directory
root@ip-172-31-41-78:/home/ubuntu# cd /var/www/html/
root@ip-172-31-41-78:/var/www/html# /var/www/html#
bash: /var/www/html#: No such file or directory
root@ip-172-31-41-78:/var/www/html#
```

i-0104434d25a50dc8d (Manav1)

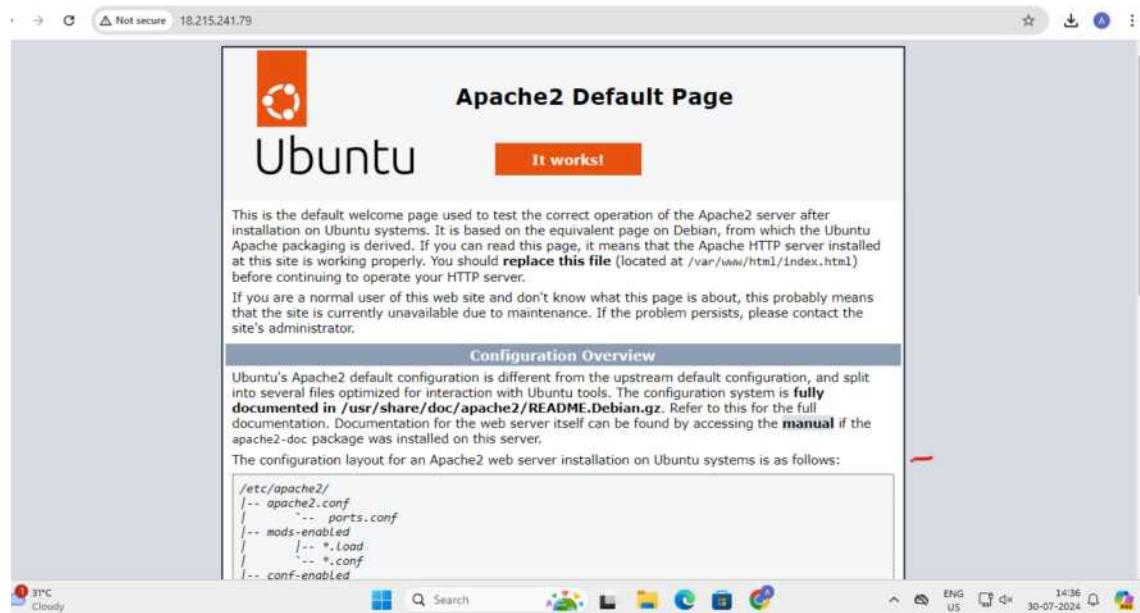
PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78

```
command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: apt install <deb name>
root@ip-172-31-41-78:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-07-30 08:44:17 UTC; 12min ago
     Docs: https://httpd.apache.org/docs/2.4/
      Main PID: 12917 (apache2)
        Tasks: 55 (limit: 1130)
       Memory: 5.3M (peak: 5.4M)
          CPU: 74ms
         CGroup: /system.slice/apache2.service
                 ├─12917 /usr/sbin/apache2 -k start
                 ├─12919 /usr/sbin/apache2 -k start
                 └─12921 /usr/sbin/apache2 -k start

Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-78:/home/ubuntu#
```

i-0104434d25a50dc8d (Manav1)

PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78



U anavPunjabi D15A 45



Hosting using S3 bucket :

The screenshot shows the 'Create bucket' configuration page on the Amazon S3 service. The top navigation bar includes 'Amazon S3 > Buckets > Create bucket'. The main title is 'Create bucket' with an 'Info' link. Below it, a sub-instruction says 'Buckets are containers for data stored in S3.' A 'General configuration' section is open, showing the following settings:

- AWS Region:** US East (N. Virginia) us-east-1
- Bucket type:** [Info](#) (radio button selected for 'General purpose')
- Bucket name:** [Info](#) (text input field containing 'test-123-manav')
- Copy settings from existing bucket - optional:** (checkbox labeled 'Choose bucket')
Format: s3://bucket/prefix

Below the configuration section, there is a note: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'.

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

► Advanced settings

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

[Cancel](#) [Create bucket](#)

| <p><input checked="" type="radio"/> Successfully created bucket "test-123 manav". To upload files and folders, or to configure additional bucket settings, choose View details.</p> <p>View details X</p> <p>Amazon S3 > Buckets</p> <p>► Account snapshot - updated every 24 hours All AWS Regions</p> <p>Storage Lens provides visibility into storage usage and activity trends. Learn more</p> <p>View Storage Lens dashboard</p> | | | | | | | | | |
|---|---------------------------------|---|---------------------------------------|---------------|----------------|---------------------------------|---|---------------------------------------|--|
| <p>General purpose buckets (1) Info All AWS Regions</p> <p>Buckets are containers for data stored in S3.</p> <p><input type="text"/> Find buckets by name</p> <table border="1"> <thead> <tr> <th>Name</th> <th>AWS Region</th> <th>IAM Access Analyzer</th> <th>Creation date</th> </tr> </thead> <tbody> <tr> <td>test-123-manav</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>August 11, 2024, 19:49:09 (UTC+05:30)</td> </tr> </tbody> </table> | Name | AWS Region | IAM Access Analyzer | Creation date | test-123-manav | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 11, 2024, 19:49:09 (UTC+05:30) | <p>Create bucket</p> <p>C Copy ARN Empty Delete</p> <p>< 1 > @</p> |
| Name | AWS Region | IAM Access Analyzer | Creation date | | | | | | |
| test-123-manav | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 11, 2024, 19:49:09 (UTC+05:30) | | | | | | |

Upload succeeded
View details below.

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

Summary

| Destination | Succeeded | Failed |
|---------------------|------------------|-------------------|
| s3://test-123-manav | 1 file, 0 B (0%) | 0 files, 0 B (0%) |

Files and folders Configuration

Files and folders (1 Total, 0 B)

| Name | Folder | Type | Size | Status | Error |
|----------|--------|------------|------|-----------|-------|
| Test.txt | - | text/plain | 0 B | Succeeded | - |

Amazon S3

Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > test-123-anushka > Test.txt

Test.txt Info

Properties Permissions Versions

Object overview

| | |
|---------------------------------------|---|
| Owner | S3 URI |
| awslabsc0w4201293t1653663267 | Copy S3 URI |
| AWS Region | Amazon Resource Name (ARN) |
| US East (N. Virginia) us-east-1 | arn:aws:s3:::test-123-anushka/Test.txt |
| Last modified | Entity tag (Etag) |
| August 11, 2024, 19:58:50 (UTC+05:30) | d41d8cd98f00b204e9800998ecf8427e |
| Size | Object URL |
| - | https://test-123-manav.s3.amazonaws.com/Test.txt |
| Type | |
| txt | |
| Key | |

U anavPunjabi D15A 45

Successfully edited bucket policy.

Amazon S3 > Buckets > test-123-anushka

test-123-manav Info

Objects Properties Permissions Metrics Management Access Points

Permissions overview

Access finding

Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#).
[View analyzer for us-east-1](#)

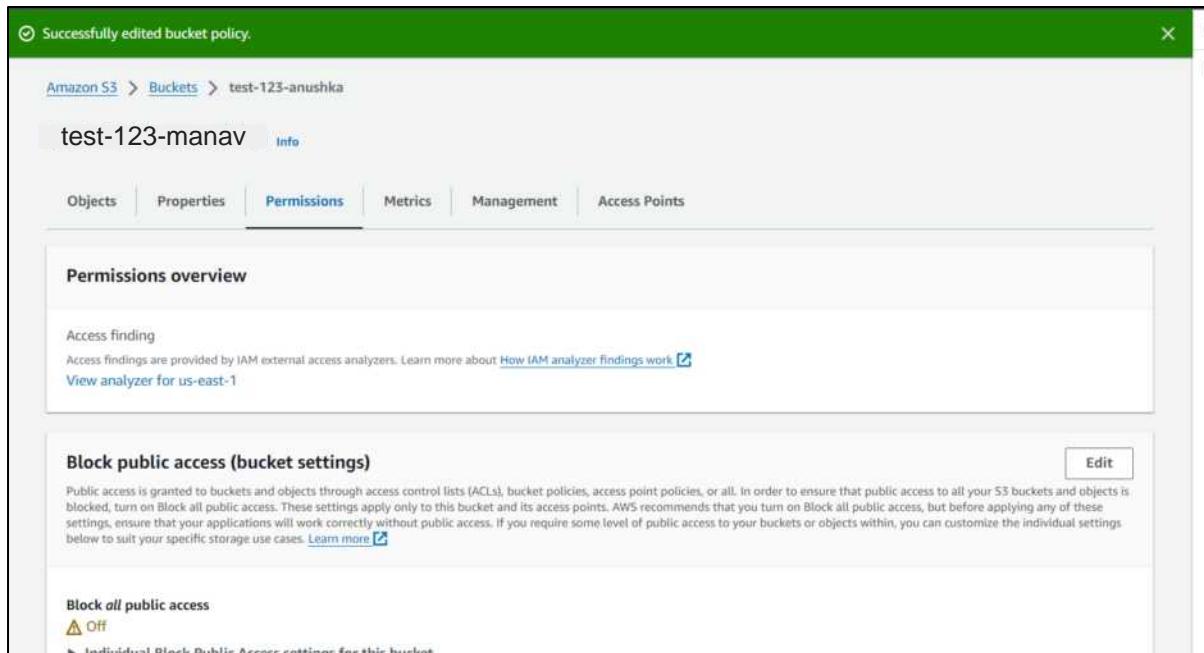
Block public access (bucket settings)

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

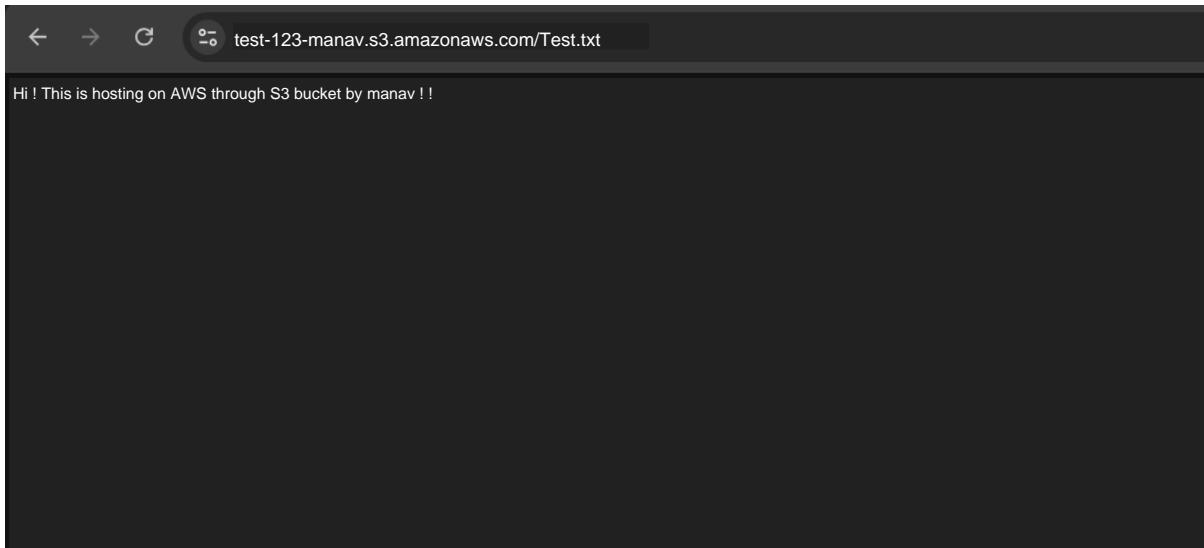
Off  Individual [Block Public Access settings for this bucket](#)

Edit

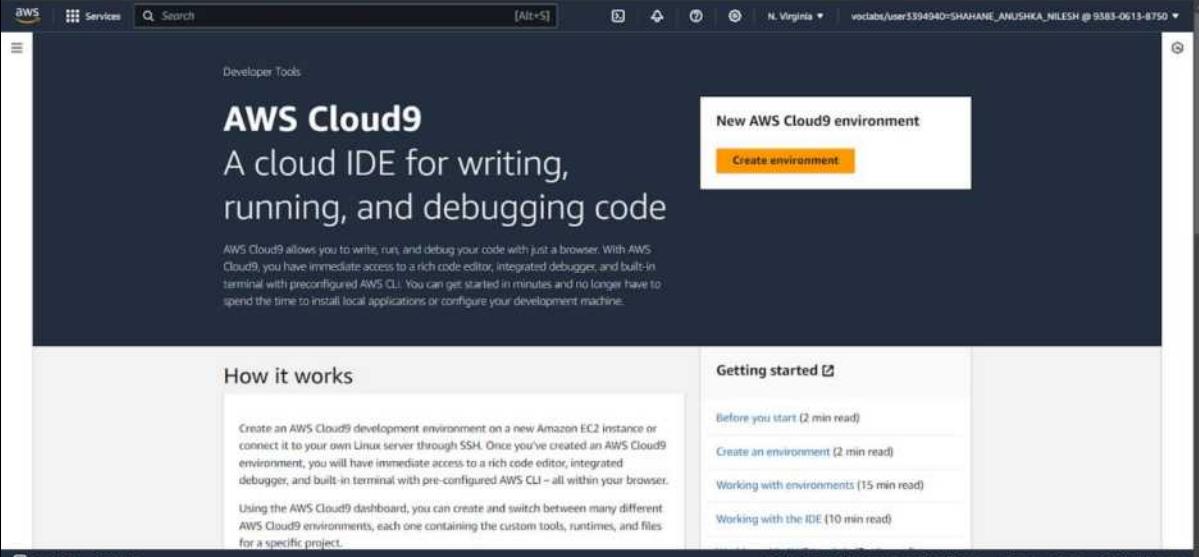


← → ⌂ test-123-manav.s3.amazonaws.com/Test.txt

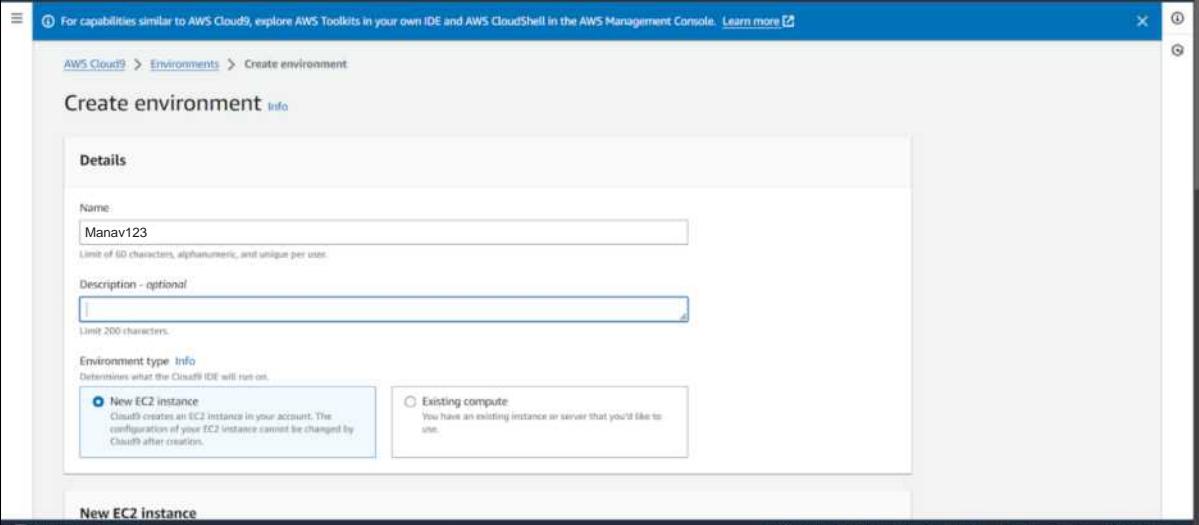
Hi ! This is hosting on AWS through S3 bucket by manav ! !



Hosting using Cloud 9 :



The screenshot shows the AWS Cloud9 homepage. At the top right, there is a call-to-action button labeled "Create environment". Below this, there's a section titled "How it works" which describes the service's purpose: "A cloud IDE for writing, running, and debugging code". A detailed description follows, explaining how Cloud9 allows users to write, run, and debug code directly in their browser using a rich code editor, integrated debugger, and built-in terminal. To the right, there's a sidebar titled "Getting started" with links to "Before you start", "Create an environment", "Working with environments", and "Working with the IDE". At the bottom, standard AWS footer links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences are visible.



This screenshot shows the "Create environment" dialog box. In the "Details" section, the "Name" field is filled with "Manav123". The "Description - optional" field is empty. Under "Environment type", the "New EC2 instance" option is selected, with a note explaining that Cloud9 creates an EC2 instance in the user's account. The "Existing compute" option is also available but not selected. At the bottom left, there is a link to "New EC2 instance". The bottom of the dialog box contains standard AWS footer links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS Cloud9 service in the AWS Management Console. A modal window is open, indicating the process of creating a new environment named 'Manav123'. The message states: 'Creating Manav123 This can take several minutes. While you wait, see Best practices for using AWS Cloud9.' Below the modal, the 'Environments' list shows one entry: 'Manav123' (Status: Open, Type: EC2 instance, Connection: Secure Shell (SSH), Permission: Owner, ARN: arn:aws:sts::938306138750:assumed-role/vocabs/user3394940:PUNJABI_MANAV_NARAIN). The interface includes standard AWS navigation elements like 'Search', 'Services', and 'CloudShell'.

The screenshot displays the AWS Cloud9 IDE interface. The main window shows the 'Welcome' screen with the title 'AWS Cloud9' and the sub-header 'Welcome to your development environment'. It includes a brief description of what AWS Cloud9 offers and a 'Getting started' sidebar with options like 'Create File', 'Upload Files...', and 'Clone from GitHub'. Below the welcome screen is a terminal window titled 'bash - [ip-172-31-72-68 x] immediate'. The terminal prompt is 'vocabs:/environment \$'. The left sidebar shows a file structure with a file named 'README.md'.

The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a code editor with the file `Cloud9.html` open, containing the following HTML and CSS code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Anushka Shahane's Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f0f0f0;
        }
        .navbar {
            background-color: #003345;
            overflow: hidden;
        }
        .navbar a {
            float: left;
            display: block;
            color: #222222;
            text-align: center;
            padding: 14px 20px;
            text-decoration: none;
        }
        .navbar a:hover {
            background-color: #ddd;
            color: #003345;
        }
        .container {
            padding: 20px;
        }
        .header {
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>Welcome</h1>
        <p>This is my personal website hosted on AWS Cloud9.</p>
        <a href="#">View My Profile</a>
    </div>
    <div class="container">
        <ul class="list-group">
            <li>Home</li>
            <li>About</li>
            <li>Contact</li>
        </ul>
    </div>
</body>
</html>
```

On the right, there's a terminal window with the command `curl -v https://ip-172-31-72-68.x.x.x/` entered and its output displayed.

The screenshot shows the AWS Cloud9 IDE interface with a browser preview window. The preview shows a simple website with a header, a container with a list group, and a footer. The browser tab is titled `[B] /Cloud9.html`. The browser window displays the following content:

Welcome to Manav's Website

About Me

Hello! I'm Manav Punjabi. This is a simple example of my personal website hosted on AWS Cloud9.

Content Section

I am a student of VESIT Department:IT Class:D15A/45

The code editor on the left remains the same as in the previous screenshot.

Experiment 2

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

Contents:

1. s3 bucket
2. ec2 instance
3. elastic beanstalk

- s3 bucket

The screenshot shows the 'Create bucket' page in the AWS Management Console. The URL in the address bar is 'Amazon S3 > Buckets > Create bucket'. The main title is 'Create bucket' with an 'Info' link. Below it, a sub-instruction says 'Buckets are containers for data stored in S3.' A 'General configuration' section is open, showing the following settings:

- AWS Region:** US East (N. Virginia) us-east-1
- Bucket type:** General purpose
Description: Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- Directory - New
Description: Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
- Bucket name:** Info
Description: Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.
- Copy settings from existing bucket - optional:** Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

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

► Advanced settings

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

[Cancel](#) [Create bucket](#)

| Amazon S3 > Buckets | | | | | | | | | |
|--|---------------------------------|---|---------------------------------------|---------------------|---------------|----------------------------|---------------------------------|---|---------------------------------------|
| ► Account snapshot - updated every 24 hours All AWS Regions <small>Storage Lens provides visibility into storage usage and activity trends. Learn more</small> | | | | | | | | | |
| View Storage Lens dashboard | | | | | | | | | |
| General purpose buckets Directory buckets | | | | | | | | | |
| General purpose buckets (1) Info All AWS Regions <small>Buckets are containers for data stored in S3.</small> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>AWS Region</th> <th>IAM Access Analyzer</th> <th>Creation date</th> </tr> </thead> <tbody> <tr> <td>test-Manav</td> <td>US East (N. Virginia) us-east-1</td> <td>View analyzer for us-east-1</td> <td>August 12, 2024, 20:04:18 (UTC+05:30)</td> </tr> </tbody> </table> | | Name | AWS Region | IAM Access Analyzer | Creation date | test-Manav | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 12, 2024, 20:04:18 (UTC+05:30) |
| Name | AWS Region | IAM Access Analyzer | Creation date | | | | | | |
| test-Manav | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 12, 2024, 20:04:18 (UTC+05:30) | | | | | | |

Upload Info

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

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

Files and folders (1 Total, 50.0 B)

[Remove](#)

[Add files](#)

[Add folder](#)

All files and folders in this table will be uploaded.

Find by name

< 1 >

| <input type="checkbox"/> | Name | Folder | Type | Size |
|--------------------------|----------|--------|------------|--------|
| <input type="checkbox"/> | test.txt | - | text/plain | 50.0 B |

Destination Info

Destination

s3://test-Manav

Upload succeeded

View details below.

Upload: status

[Close](#)

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

Summary

Destination
s3://test-Manav

Succeeded

1 file, 287.0 B (100.00%)

Failed

0 files, 0 B (0%)

[Files and folders](#)

[Configuration](#)

Files and folders (1 Total, 287.0 B)

Find by name

< 1 >

| Name | Folder | Type | Size | Status | Error |
|-----------|--------|-----------|---------|-----------|-------|
| test.html | - | text/html | 287.0 B | Succeeded | - |

| Properties | Permissions | Versions |
|--|---|----------|
| Object overview | | |
| Owner awslabsc0w3698888l1642940625 | S3 URI s3://test-Manav/test.html | |
| AWS Region US East (N. Virginia) us-east-1 | Amazon Resource Name (ARN) arn:aws:s3:::test-Manav/test.html | |
| Last modified August 12, 2024, 22:33:51 (UTC+05:30) | Entity tag (Etag) 7a3411f1dad97a2779c8dc65580432d2 | |
| Size 287.0 B | Object URL https://test-Manav.s3.amazonaws.com/test.html | |
| Type html | | |
| Key test.html | | |

Edit static website hosting [Info](#)

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

Static website hosting

Disable

Enable

Hosting type

Host a static website
Use the bucket endpoint as the web address. [Learn more](#) 

Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#) 

Objects Properties Permissions Metrics Management Access Points

Permissions overview

Access finding
Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#).

[View analyzer for us-east-1](#)

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

On

► Individual Block Public Access settings for this bucket

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Edit](#) [Delete](#)

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::test-Manav/*"  
    }  
  ]  
}
```



- Launching an EC2 instance

[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 Add additional tags

Quick Start



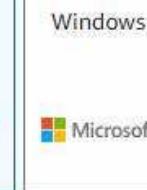
Amazon
Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Li
>
SUS



Browse more AMIs

Including AMIs from:
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

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-04a81a99f5ec58529

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

▼ Configure storage [Info](#)

[Advanced](#)

1x

8

GiB

gp3



Root volume (Not encrypted)



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



[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](#)

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-073a9e2489cd0d33c

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-1**' with the following rules:

Allow SSH traffic from

Helps you connect to your instance

Anywhere

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

[EC2](#) > [Instances](#) > [Launch an instance](#)

Success

Successfully initiated launch of instance ([i-0e39cd326d64588eb](#))

▼ Launch log

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

| Instances (1) Info | | C | Connect | Instance state ▾ | Actions ▾ | Launch instances | ⋮ |
|---|-------------------------------|-------------------|----------------|------------------|--------------|------------------|-------------------|
| <input type="text"/> Find Instance by attribute or tag (case-sensitive) | | | | | | All states ▾ | ⋮ |
| Instance ID = i-0e39cd326d64588eb X | Clear filters | | | | | | |
| <input type="checkbox"/> | Name ↴ | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone |

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

▼ Instance summary [Info](#)

| | | |
|--|---|---|
| Instance ID i-0e39cd326d64588eb (Manav) | Public IPv4 address 34.201.2.60 open address | Private IPv4 addresses: 172.31.13.190 |
| IPv6 address - | Instance state Running | Public IPv4 DNS ec2-34-201-2-60.compute-1.amazonaws.com open address |
| Hostname type IP name: ip-172-31-13-190.ec2.internal | Private IP DNS name (IPv4 only) ip-172-31-13-190.ec2.internal | Elastic IP addresses |
| Answer private resource DNS name | Instance type | |

```
[ec2-user@ip-172-31-13-190 ~]$ ls
[ec2-user@ip-172-31-13-190 ~]$ echo "hello"
hello
[ec2-user@ip-172-31-13-190 ~]$ cat > myfile.txt
this is advanced devops lab
^C
[ec2-user@ip-172-31-13-190 ~]$ cat myfile
cat: myfile: No such file or directory
[ec2-user@ip-172-31-13-190 ~]$ cat myfile.txt
this is advanced devops lab
[ec2-user@ip-172-31-13-190 ~]$ █
```

```
root@ip-172-31-32-173:~# sudo su
root@ip-172-31-32-173:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sql:
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-
0 upgraded, 10 newly installed, 0 to remove and 26 not upgraded.
Need to get 1680 kB/2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
```

```
root@ip-172-31-32-173:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-07-30 08:58:11 UTC; 44s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2619 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 5.4M (peak: 5.5M)
      CPU: 40ms
    CGroup: /system.slice/apache2.service
            ├─2619 /usr/sbin/apache2 -k start
            ├─2621 /usr/sbin/apache2 -k start
            └─2623 /usr/sbin/apache2 -k start

Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

```
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:58:11 ip-172-31-32-173 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-32-173:~# cd /var/www/html
```

The screenshot shows a web browser window displaying the Apache2 Default Page. The page features the Ubuntu logo at the top left, followed by the text "Apache2 Default Page" and "Ubuntu". A red button on the right says "It works!". Below this, there is a message about the default welcome page and its purpose. A section titled "Configuration Overview" provides details about the configuration layout, mentioning the directory structure under /etc/apache2. A code block shows the directory tree:

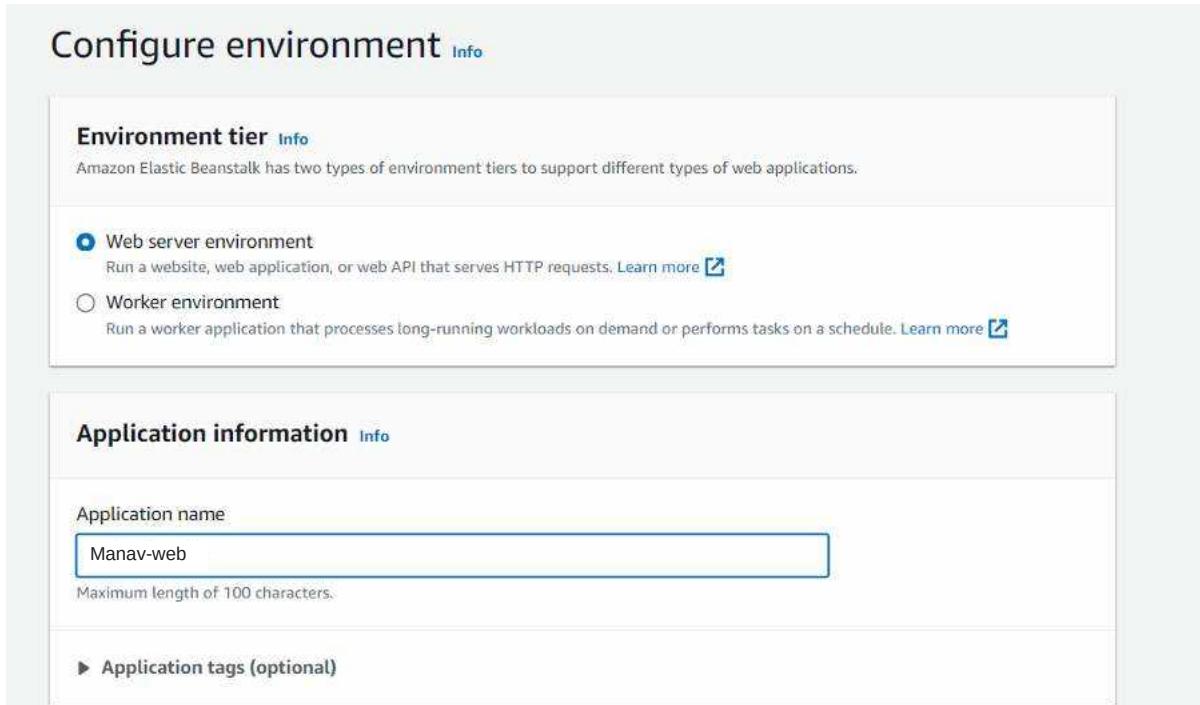
```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
```

The screenshot shows a web browser window displaying a custom index2.html file. The page contains the text "Hi..Manav here" in a large, bold, black font.

Elastic beanstalk



The screenshot shows the Amazon Elastic Beanstalk landing page. At the top left, there's a 'Compute' dropdown menu. The main title is 'Amazon Elastic Beanstalk' in large bold letters, followed by the subtitle 'End-to-end web application management.' Below the title, a brief description states: 'Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.' To the right, there's a 'Get started' button with the subtext 'Easily deploy your web application in minutes.' and a 'Create application' button.



The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk wizard. It has two sections: 'Environment tier' and 'Application information'. In the 'Environment tier' section, 'Web server environment' is selected. In the 'Application information' section, the 'Application name' field contains 'Manav-web'. There is also a section for 'Application tags (optional)' which is currently empty.

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

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

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

Application information Info

Application name

Manav-web

Maximum length of 100 characters.

▶ Application tags (optional)

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

Node.js



Platform branch

Node.js 20 running on 64bit Amazon Linux 2023



Platform version

6.2.0 (Recommended)



Review Info

Step 1: Configure environment

[Edit](#)

Environment information

Environment tier

Application name

Web server environment

Manav-web

Environment name

Application code

Manav-web-env

Sample application

Platform

arn:aws:elasticbeanstalk:eu-north-1::platform/PHP 8.3

running on 64bit Amazon Linux 2023/4.3.2

Step 2: Configure service access

[Edit](#)

Service access Info

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

Service role

EC2 instance profile

| | | |
|----------------|-------------------------|--------------------|
| Lifecycle | Log streaming | Allow URL fopen |
| false | Deactivated | On |
| Display errors | Document root | Max execution time |
| Off | - | 60 |
| Memory limit | Zlib output compression | Proxy server |
| 256M | Off | nginx |
| Logs retention | Rotate logs | Update level |
| 7 | Deactivated | minor |
| X-Ray enabled | | |
| Deactivated | | |

Environment properties

| Key | ▲ Value |
|--|-----------|
| No environment properties There are no environment properties defined | |

Cancel

Previous

Submit

Environment overview

| | |
|-----------|------------------|
| Health | Environment ID |
| ⊖ Unknown | ⌚ e-trkmirvuz |
| Domain | Application name |
| - | Manav-web |

Platform

Change version

| |
|---|
| Platform |
| Node.js 20 running on 64bit Amazon Linux 2023/6.2.0 |
| Running version |
| - |
| Platform state |
| ⌚ Supported |

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud.

This environment is launched with Elastic Beanstalk Node.js Platform.

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploying an Express Application to AWS Elastic Beanstalk](#)
- [Deploying an Express application with clustering to Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

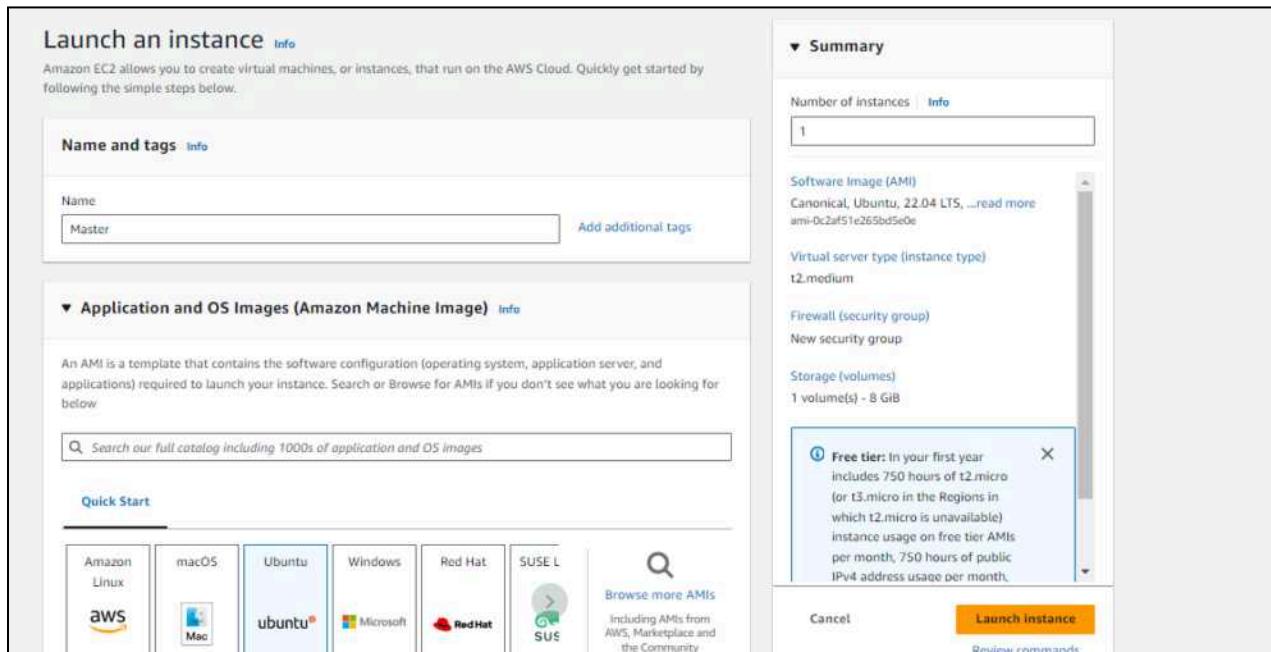
ADVANCE DEVOPS EXP 3

Name:Manav Punjabi
Class:D15A
Roll No:45

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

Step 1:Pre-requisites

1.1 Create 3 EC2 instances,one for the master node and two for the worker nodes.



1.2 Proceed with the following settings and create a new key pair as follows(use the same key pair for all the three nodes)

The screenshot shows the AWS Lambda 'Create Function' configuration interface. It includes sections for 'Instance type', 'Key pair (login)', and 'Network settings'.

Instance type: t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0496 USD per Hour
On-Demand Windows base pricing: 0.0676 USD per Hour
On-Demand RHEL base pricing: 0.0784 USD per Hour
On-Demand SUSE base pricing: 0.1496 USD per Hour

Key pair (login): two-tier-app-k8s

Network settings: Network: vpc-04007898e59a6979f Subnet:

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on

Cancel **Create key pair**

| Instances (1/3) Info | | | | | | | | | | |
|--|---------------------|----------------|---------------|-------------------|-----------------------------|-------------------|---------------------------------------|-----------|--|--|
| Last updated less than a minute ago Instance state ▾ Actions ▾ Launch instances | | | | | | | | | | |
| <input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> All states ▾ | | | | | | | | | | |
| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IP | | |
| Worker-2 | i-0e3930ceb2d892d01 | Running | t2.medium | 2/2 checks passed | View alarms | ap-south-1a | ec2-13-234-226-219.ap... 13.234.22 | | | |
| Worker-1 | i-0d16e01d1824e0e3a | Running | t2.medium | 2/2 checks passed | View alarms | ap-south-1a | ec2-65-0-104-95.ap-so... 65.0.104. | | | |
| Master | i-01ae3d388db90ad73 | Running | t2.medium | 2/2 checks passed | View alarms | ap-south-1a | ec2-13-232-36-34.ap-s... 13.232.36 | | | |

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

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

Instance ID
i-0e3930ceb2d892d01 (Worker-2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is two-tier-app-k8s.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "two-tier-app-k8s.pem"
4. Connect to your instance using its Public DNS:
ec2-13-234-226-219.ap-south-1.compute.amazonaws.com

Example:

ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-234-226-219.ap-south-1.compute.amazonaws.com

```
acer@TMP214-53 MINGW64 ~/Downloads
$ ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-232-36-34.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com (13.232.36.34)' can't be established.
ED25519 key fingerprint is SHA256:uVGEO+FwYefj60j0ft70Sralv8NrzEi/IwxAtBY+EPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Wed Sep 11 14:07:10 UTC 2024

System load: 0.0          Processes:           106
Usage of /: 20.7% of 7.57GB   Users logged in:      0
Memory usage: 5%           IPv4 address for eth0: 172.31.45.227
Swap usage: 0%          

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Step 2: Prepare Nodes

2.1. Update the package manager on all nodes:

```
sudo apt-get update && sudo apt-get upgrade -y
```

```
ubuntu@ip-172-31-22-29:~ Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-22-29:~$ sudo apt-get update && sudo apt-get upgrade -y

ubuntu@ip-172-31-28-127:~ Expanded Security Maintenance for Application
0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-28-127:~$ sudo apt-get update && sudo apt-get upgrade -y
```

```
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2023 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [352 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.8 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2437 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [419 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted a
```

2.2. Disable Swap (Kubernetes requires swap to be off):

```
sudo swapoff -a
```

```
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

```
ubuntu@ip-172-31-22-29:~$ sudo swapoff -a  
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

2.3. Load necessary kernel modules for networking and iptables:

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

```
overlay
```

```
br_netfilter
```

```
EOF
```

```
sudo modprobe overlay
```

```
sudo modprobe br_netfilter
```

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
overlay  
br_netfilter  
EOF  
sudo modprobe overlay  
sudo modprobe br_netfilter  
overlay  
br_netfilter
```

2.4. Configure sysctl settings for Kubernetes networking:

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

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

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

```
EOF
```

```
sudo sysctl --system
```

```

ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1

```

Step 3: Install Docker

Kubernetes uses container runtimes like Docker. Install Docker on all nodes.

```

sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io

```

```

ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Fetched 129 kB in 1s (241 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.17).
curl set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed

```

Configure Docker for Kubernetes:

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
```

```
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
sudo systemctl restart docker
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

Step 4: Install kubeadm, kubelet, kubectl

Install Kubernetes tools on all nodes.

4.1. Add Kubernetes APT repository:

```
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-22-29:~$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main
```

4.2. Install kubeadm, kubelet, and kubectl:

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
```

Step 5: Initialize the Kubernetes Cluster on Master Node

On the master node:

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

```
ubuntu@ip-172-31-22-29:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --v=5
Found multiple CRI endpoints on the host. Please define which one do you wish to
use by setting the 'criSocket' field in the kubeadm configuration file: unix://
/var/run/containerd/containerd.sock, unix:///var/run/crio/crio.sock
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.detectCRISocketImpl
    cmd/kubeadm/app/util/runtime/runtime.go:167
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.DetectCRISocket
    cmd/kubeadm/app/util/runtime/runtime.go:175
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetNodeRegistrationDynamicDefaults
    cmd/kubeadm/app/util/config/initconfiguration.go:118
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetInitDynamicDefaults
    cmd/kubeadm/app/util/config/initconfiguration.go:64
k8s.io/kubernetes/cmd/kubeadm/app/util/config.DefaultedInitConfiguration
    cmd/kubeadm/app/util/config/initconfiguration.go:248
k8s.io/kubernetes/cmd/kubeadm/app/util/config.LoadOrDefaultInitConfiguration
    cmd/kubeadm/app/util/config/initconfiguration.go:282
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newInitData
    cmd/kubeadm/app/cmd/init.go:319
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func3
    cmd/kubeadm/app/cmd/init.go:170
k8s.io/kubernetes/cmd/kubeadm/app/cmd/phases/workflow.(*Runner).InitData
    cmd/kubeadm/app/cmd/phases/workflow/runner.go:183
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func1
```

5.1. Set up kubectl on the master node:

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

```
ubuntu@ip-172-31-22-29:~$ sudo kubeadm config images pull  
sudo kubeadm init  
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf "$HOME/.kube/config"  
sudo chown "$(id -u):$(id -g)" "$HOME/.kube/config"  
  
# Network Plugin = calico  
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml  
  
kubeadm token create --print-join-command --v=5  
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in the kubeadm configuration file: unix:///var/run/containerd/containerd.sock, unix:///var/run/crio/crio.sock  
To see the stack trace of this error execute with --v=5 or higher  
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in the kubeadm configuration file: unix:///var/run/containerd/containerd.sock, unix:///var/run/crio/crio.sock
```

Step 6: Install a Pod Network Add-on

To enable communication between pods, install a pod network plugin like Flannel or Calico.

Install Flannel:

```
kubectl apply -f
```

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

```
ubuntu@ip-172-31-22-29:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml --validate=false  
E0913 15:35:04.261458 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
E0913 15:35:04.261902 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
E0913 15:35:04.263424 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
E0913 15:35:04.263795 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
E0913 15:35:04.265840 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
E0913 15:35:04.266524 19259 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080  
unable to recognize "https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml": Get "http://localhost:8080/api?timeout=32s": dial
```

Step 7: Join Worker Nodes to the Cluster

On the **worker nodes**, run the command provided by the master node during initialization . It looks something like this:

```
sudo kubeadm join <master-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>
```

```
* clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created  
* daemonset.apps/calico-node created  
* deployment.apps/calico-kube-controllers created  
* kubeadm join 172.31.62.216:6443 --token br7fe5.hq28adbn1mu17ky --discovery-token-ca-cert-hash sha256:2bc469a8d14fheb8f879328d2b416fad  
32b29a8505d3f448b98703fff3b814d9
```

Step 8: Verify the Cluster

Once the worker node joins, check the status on the **master node**

```
ubuntu@ip-172-31-45-227:~$ kubectl get nodes
NAME        STATUS   ROLES     AGE      VERSION
ip-172-31-43-211   Ready    <none>    50s     v1.29.0
ip-172-31-45-13    Ready    <none>    34s     v1.29.0
ip-172-31-45-227   Ready    control-plane   5m17s   v1.29.0
ubuntu@ip-172-31-45-227:~$ |
```

ADVANCE DEVOPS EXP 4

Name:Manav Punjabi

Class:D15A

Roll No:45

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

Step 1: Install Kubectl on Ubuntu

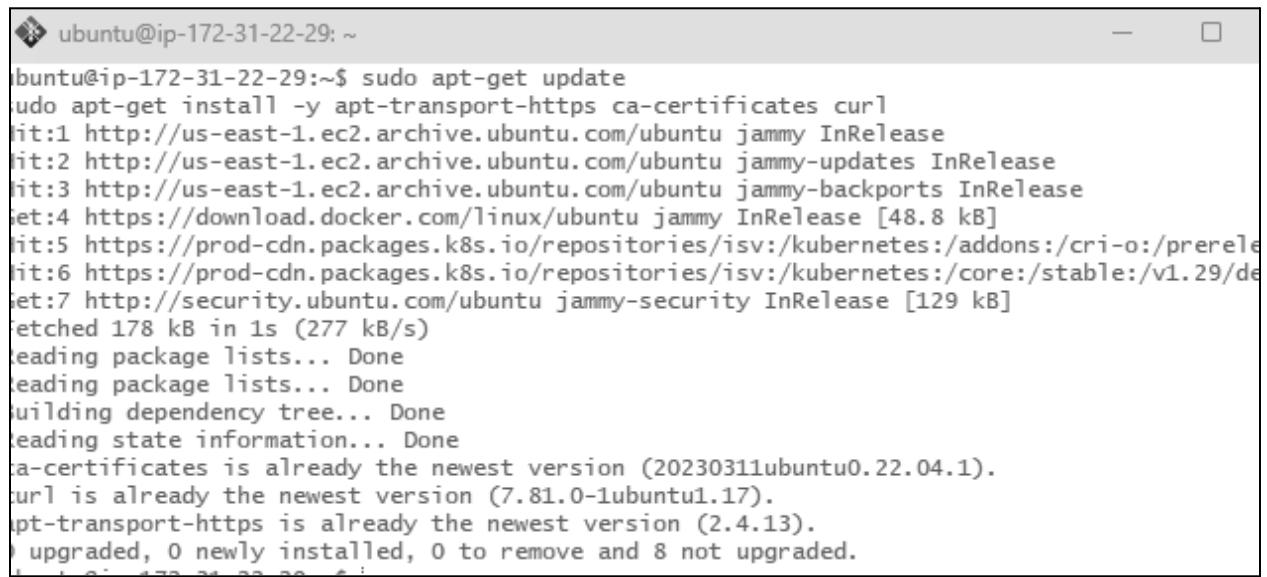
1.1 Add Kubernetes APT repository

First, add the Kubernetes repository to your system.

1. Install prerequisites:

```
sudo apt-get update
```

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



```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
[sudo] password for ubuntu:
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/de
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Fetched 178 kB in 1s (277 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.17).
apt-transport-https is already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
```

2. Add the GPG key for Kubernetes:

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

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

```
ubuntu@ip-172-31-22-29:~$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

3. Add the Kubernetes repository:

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

```
ubuntu@ip-172-31-22-29:~$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring
.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/ku
ernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/
kubernetes-focal main
```

1.2 Install kubectl

Now install kubectl:

```
sudo apt-get update
```

```
sudo apt-get install -y kubectl
```

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb InRelease
Ign:7 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.62.138 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

Verify the installation(extra):

```
kubectl version --client
```

```
ubuntu@ip-172-31-22-29:~$ kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 2: Deploying Your Application on Kubernetes

2.1 Set up Kubernetes Cluster

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

```
kubectl get nodes
```

```
ubuntu@ip-172-31-45-227:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-172-31-43-211   Ready    <none>    50s   v1.29.0
ip-172-31-45-13   Ready    <none>    34s   v1.29.0
ip-172-31-45-227   Ready    control-plane   5m17s  v1.29.0
ubuntu@ip-172-31-45-227:~$ |
```

Step 3: Create the Deployment YAML file

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

```
ubuntu@ip-172-31-45-227:~$ nano nginx-deployment.yaml
```

b)Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).

```
ubuntu@ip-172-31-45-227: ~          nginx-deployment.yaml
GNU nano 6.2
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.3
          ports:
            - containerPort: 80
```

Step 4:Create the Service YAML File

a)Create the YAML File: Create another file named nginx-service.yaml

```
ubuntu@ip-172-31-45-227:~$ nano nginx-service.yaml
```

b)Add the Service Configuration: Copy and paste the following YAML content into the file given below.

```
ubuntu@ip-172-31-45-227: ~          nginx-service.yaml *
GNU nano 6.2
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

Step 5:Apply the YAML Files

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

```
ubuntu@ip-172-31-45-227:~$ kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-service.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
```

b)Verify the Deployment: Check the status of your Deployment,Pods and Services.

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
kubectl get pods
kubectl get services
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2           2          40s
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-6k84m   1/1     Running   0          40s
nginx-deployment-6b4d6fdbf-9d8j6   1/1     Running   0          40s
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
kubernetes     ClusterIP   10.96.0.1    <none>        443/TCP     40m
nginx-service   LoadBalancer   10.106.182.152  <pending>    80:32317/TCP  40s
```

Describe the deployment(Extra)

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  1/1     1           1           14h
ubuntu@ip-172-31-45-227:~$ kubectl describe deployment
Name:            nginx-deployment
Namespace:       default
CreationTimestamp:  Wed, 11 Sep 2024 17:16:17 +0000
Labels:          <none>
Annotations:    deployment.kubernetes.io/revision: 2
Selector:        app=nginx
Replicas:        1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx:latest
      Port:       80/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:
        /usr/share/nginx/html from website-volume (rw)
  Volumes:
    website-volume:
      Type:      ConfigMap (a volume populated by a ConfigMap)
      Name:      nginx-website
      Optional:  false
Conditions:
  Type    Status  Reason
  ----  -----
  Available  True    MinimumReplicasAvailable
  Progressing  True    NewReplicaSetAvailable
OldReplicaSets:  nginx-deployment-6b4d6fdbf (0/0 replicas created)
NewReplicaSet:   nginx-deployment-776b8fd845 (1/1 replicas created)
Events:         <none>
```

Step 6:Ensure Service is Running

6.1 Verify Service: Run the following command to check the services running in your cluster:

```
kubectl get service
```

```
ubuntu@ip-172-31-45-227:~$ kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      16h
nginx     NodePort   10.106.0.176    <none>        80:32618/TCP  76m
nginx-service  NodePort   10.106.182.152  <none>        80:30007/TCP  15h
nginx2     NodePort   10.99.32.156    <none>        80:31421/TCP  8s
```

Step 7:Forward the Service Port to Your Local Machine

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

1. **Forward the Service Port:** Use the following command to forward a local port to the service's target port.

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

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

2. This means port forwarding is now active, and any traffic to localhost:8080 will be routed to the nginx-service on port 80.

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8081:8080
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-776b8fd845-k9cx4  1/1     Running   0          113m
ubuntu@ip-172-31-45-227:~$ kubectl logs nginx-deployment-776b8fd845-k9cx4
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/12 06:35:51 [notice] 1#1: using the "epoll" event method
2024/09/12 06:35:51 [notice] 1#1: nginx/1.27.1
2024/09/12 06:35:51 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/12 06:35:51 [notice] 1#1: OS: Linux 6.5.0-1022-aws
2024/09/12 06:35:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/12 06:35:51 [notice] 1#1: start worker processes
2024/09/12 06:35:51 [notice] 1#1: start worker process 24
2024/09/12 06:35:51 [notice] 1#1: start worker process 25
```

Step 8: Access the Application Locally

1. **Open a Web Browser:** Now open your web browser and go to the following URL:

http://localhost:8080

You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080.

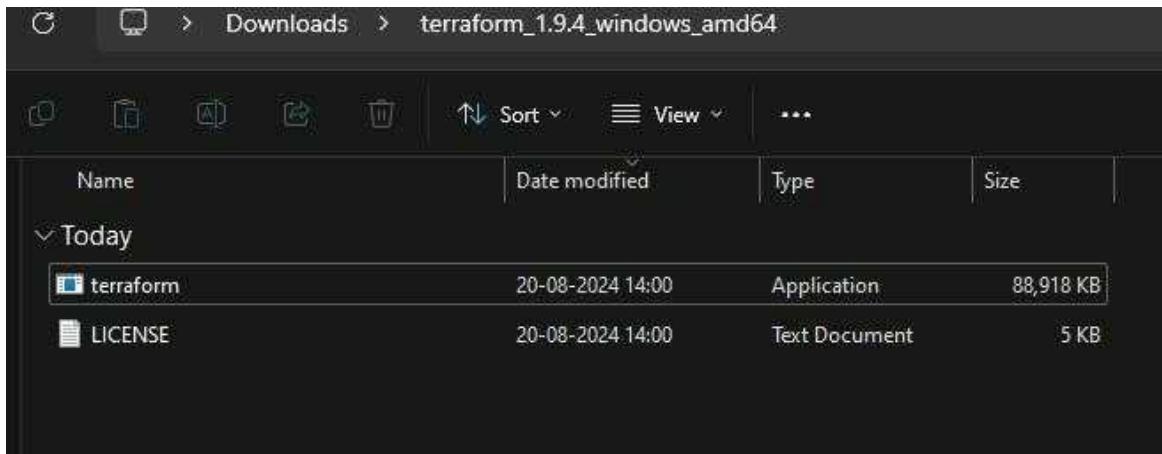
In case the port 8080 is unavailable, try using a different port like 8081

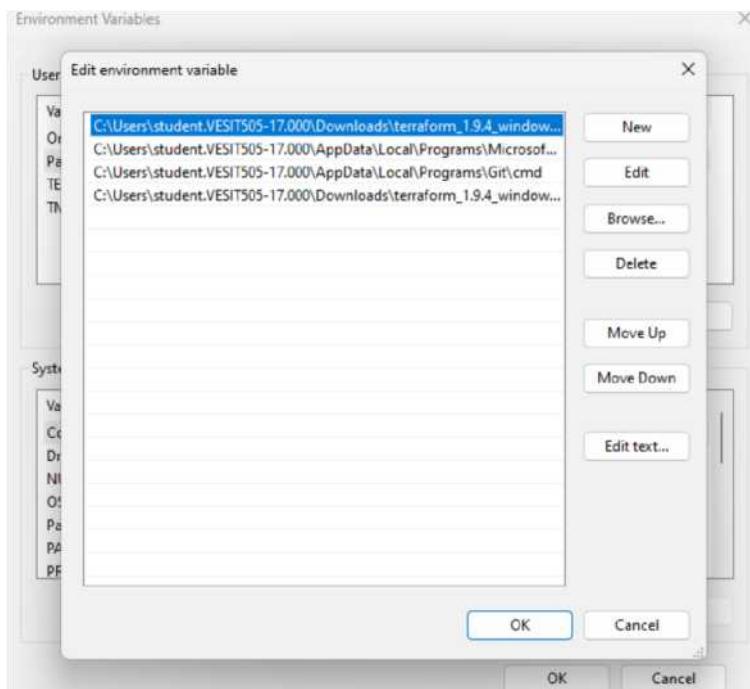


AdvDevops Experiment 5

AIM: Installation and configuration of terraform on Windows

The screenshot shows the HashiCorp Terraform website's "Install Terraform" section for macOS. It features a purple logo icon, a "Install Terraform" button, and a dropdown menu showing "1.9.4 (latest)". Below this, there are two sections: "Package manager" containing a terminal command: "brew tap hashicorp/tap" and "brew install hashicorp/tap/terraform", and "Binary download" with links for "AMD64" and "ARM64". To the right, there's an "About Terraform" summary, a "Featured docs" sidebar with links to "Introduction to Terraform", "Configuration Language", "Terraform CLI", "HCP Terraform", and "Provider Use", and a "Get Started" button.





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

```
PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64> terraform
Usage: terraform [global options] <subcommand> [args]
```

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

Main commands:

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

All other commands:

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

Manav Punjabi D15A 45

```
PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64> terraform --version
Terraform v1.9.4
on windows_amd64
PS C:\Users\student.VESIT505-17.000\Downloads\terraform_1.9.4_windows_amd64> |
```

ADVANCE DEVOPS EXP 6

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

Part A: Creating docker image using terraform

Step 1:Check Docker functionality

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

C:\Users\student>docker

Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

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

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

Check for the docker version with the following command.

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

C:\Users\student>
```

Create a folder named ‘Terraform Scripts’ in which we save our different typesof scripts which will be further used in this experiment.

Step 2:

Creating a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.

Creating a new docker.tf file using Atom editor and write the following contents into.

This will create a Ubuntu Linux container

```
"ψ" docker.tf  ×
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 # Pull the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "3600"]
24 }
```

Step 3: Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform plan

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach           = false
    + bridge          = (known after apply)
    + command         = [
        + "sleep",
        + "3600",
    ]
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data   = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
}
```

```
+ runtime          = (known after apply)
+ security_opts   = (known after apply)
+ shm_size         = (known after apply)
+ 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.

Step 5: Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : “**terraform apply**”

```
● PS C:\Users\Admin\TerraformScripts\Docker> terraform apply

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

Terraform will perform the following actions:

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

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

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

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

Enter a value: yes

```
● docker_image.ubuntu: Creating...
● docker_image.ubuntu: Creation complete after 9s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
● docker_container.foo: Creating...
● docker_container.foo: Creation complete after 2s [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Before Executing Apply step:

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

After Executing Apply step:

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------|--------|--------------|-------------|--------|
| ubuntu | latest | edbfe74c41f8 | 3 weeks ago | 78.1MB |

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
● PS C:\Users\Admin\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach          = false -> null
    - command        = [
        - "sleep",
        - "3600",
    ] -> null
    - cpu_shares     = 0 -> null
    - dns            = [] -> null
    - dns_opts       = [] -> null
    - dns_search     = [] -> null
    - entrypoint     = [] -> null
    - env            = [] -> null
    - gateway        = "172.17.0.1" -> null
    - group_add      = [] -> null
    - hostname       = "01adf07e5918" -> null
    - id             = "01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24" -> 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
    - 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     = "default" -> 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:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  - image_id = "sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - latest   = "sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

```

Destroy complete! Resources: 2 destroyed.

Docker images After Executing Destroy step

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

ADVANCE DEVOPS EXP 7

Name:Manav Punjabi
Class:D15A
Roll No:45

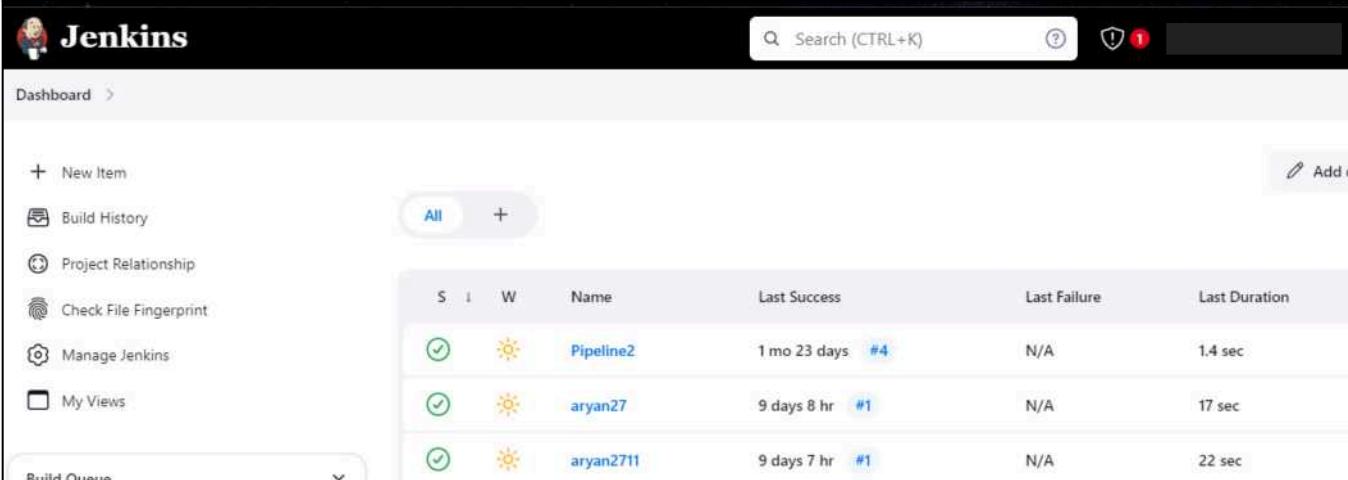
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

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



The screenshot shows the Jenkins dashboard with the following interface elements:

- Header:** Shows the Jenkins logo and a search bar labeled "Search (CTRL+K)".
- Left Sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views".
- Central Area:** A table displaying three Jenkins pipelines:

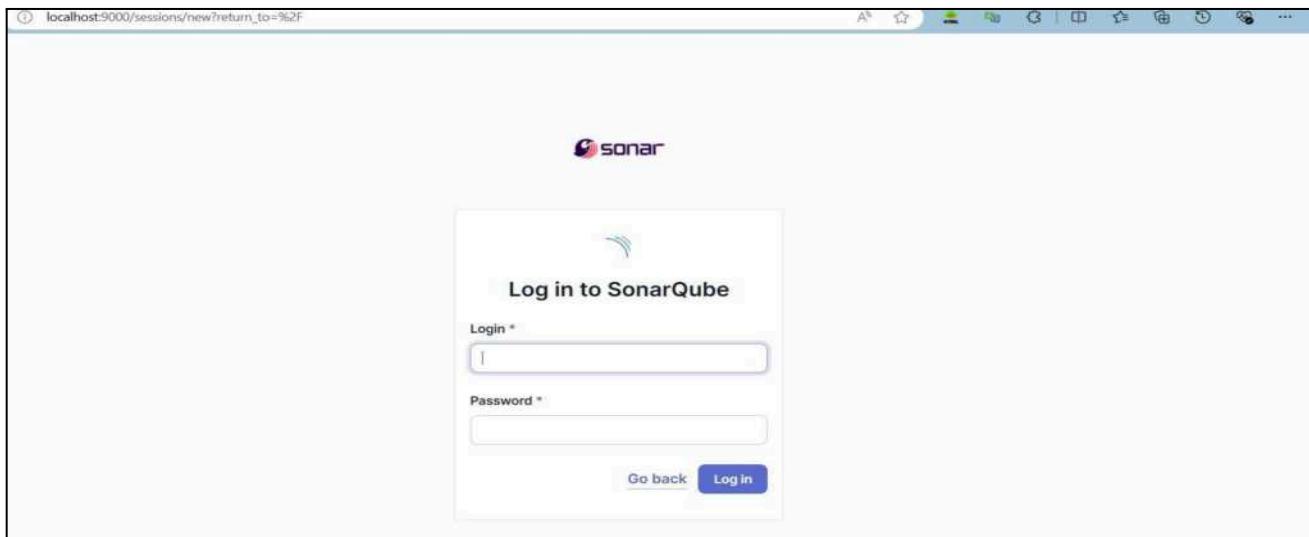
| S | I | W | Name | Last Success | Last Failure | Last Duration |
|---|----|-----------|--------------|--------------|--------------|---------------|
| ✓ | ☀️ | Pipeline2 | 1 mo 23 days | #4 | N/A | 1.4 sec |
| ✓ | ☀️ | aryan27 | 9 days 8 hr | #1 | N/A | 17 sec |
| ✓ | ☀️ | aryan2711 | 9 days 7 hr | #1 | N/A | 22 sec |
- Bottom:** A "Build Queue" dropdown menu.

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

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

```
PS C:\Windows\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
7478e0ac0f23: Pull complete  
90a925ab929a: Pull complete  
7d9a34308537: Pull complete  
80338217a4ab: Pull complete  
1a5fd5c7e184: Pull complete  
7b87d6fa783d: Pull complete  
bd819c9b5ead: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde  
Status: Downloaded newer image for sonarqube:latest  
5ab3928e5e27607e3661d129731e4e600a9019574c7dc2767aa9b3bfdaa941be
```

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



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

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

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 'sonarq'. Below the search bar, there are tabs for 'Updates' (25), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main area displays a table with a single row for the 'SonarQube Scanner 2.17.2' plugin. The table columns are 'Install', 'Name ↴', and 'Released'. The 'Install' button is highlighted with a blue background. The 'Name' column shows 'SonarQube Scanner 2.17.2'. The 'Released' column shows '6 mo 29 days ago'. Below the table, a description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page with the 'Download progress' tab selected. On the left, there is a sidebar with 'Updates' (25), 'Available plugins' (selected), 'Installed plugins', 'Advanced settings', and 'Download progress' (selected). The main area is titled 'Download progress' and shows the status of the SonarQube Scanner plugin. It indicates 'Preparation' with three success items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. It also shows 'SonarQube Scanner' with a green checkmark and 'Success'. Below that, 'Loading plugin extensions' is shown with another green checkmark and 'Success'. At the bottom, there are two buttons: '→ Go back to the top page (you can start using the installed plugins right away)' and '→ Restart Jenkins when installation is complete and no jobs are running'.

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

and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under Name add <project name of sonarqube>, here we have named it as **adv_devops_7_sonarqube**

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

SonarQube servers

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

Environment variables.

SonarQube installations

List of SonarQube installations

Name

adv_devops_7_sonarqube

Server URL

Default is <http://localhost:9000>

<https://localhost:9000>

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

Advanced ▾

7. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for 'Gradle installations', 'SonarScanner for MSBuild installations', 'SonarQube Scanner installations', and 'Ant installations'. Each section has a 'Add [Tool]' button. The 'SonarQube Scanner installations' section is currently selected.

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

The screenshot shows the 'SonarQube Scanner installations' configuration dialog. It allows adding a new installation with a name ('sonarqube_exp7') and checking the 'Install automatically' option. A sub-section for 'Install from Maven Central' is expanded, showing the selected version ('SonarQube Scanner 6.1.0.4477').

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

adv_devops_exp7
» Required field

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

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

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

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

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

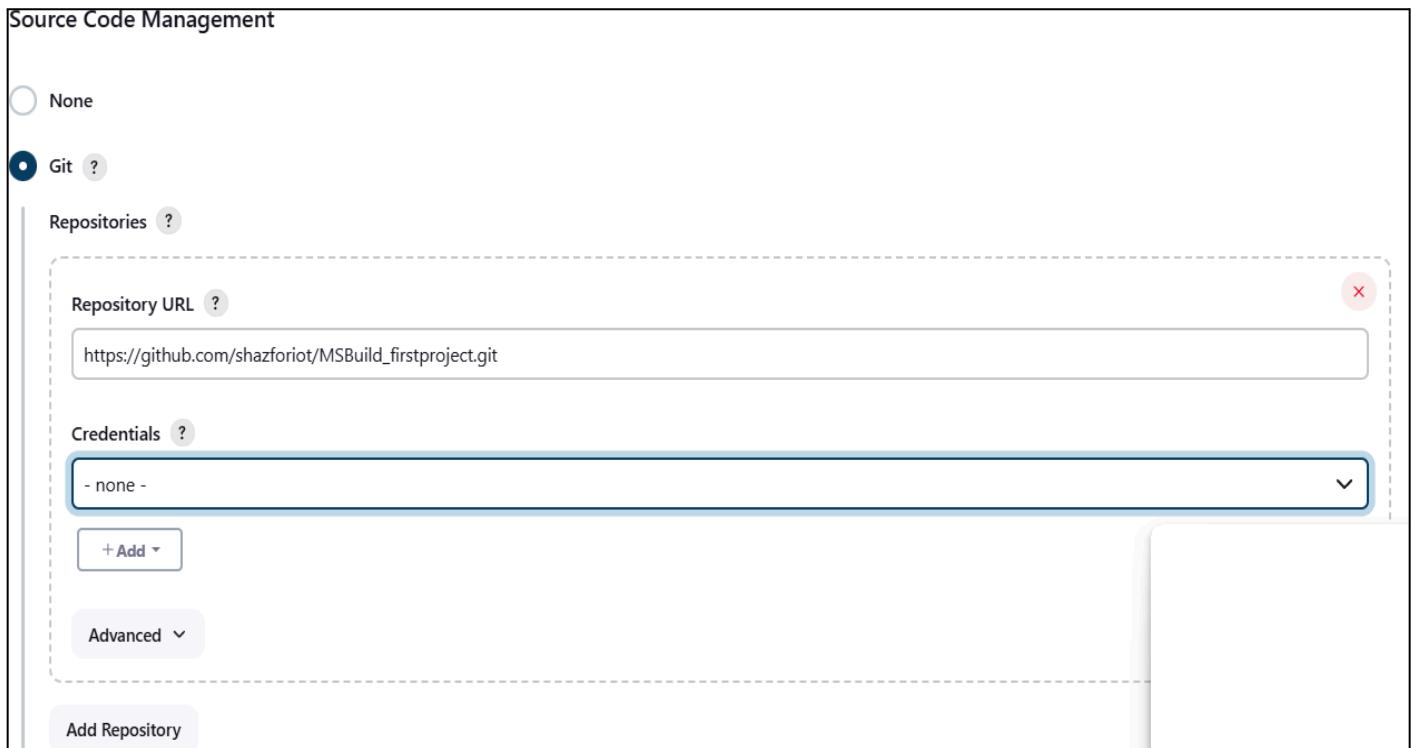
branch Pipeline
Creates a pipeline job for each branch of a GitHub repository, according to detected branches in the SCM repository.

OK

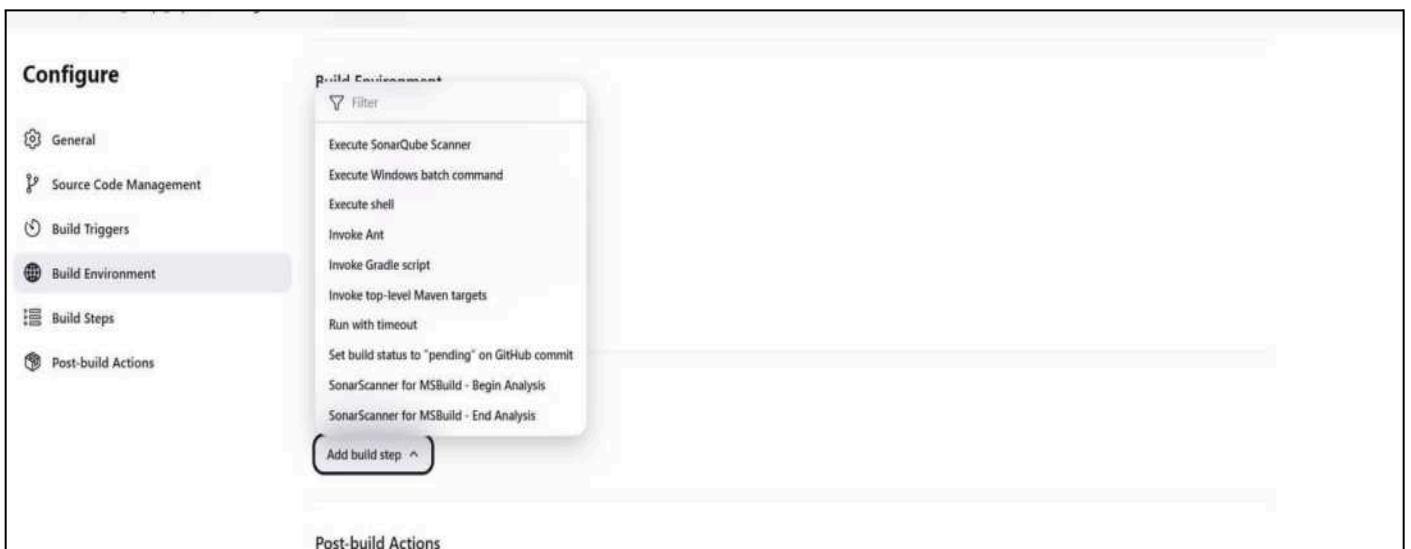
9. Choose this GitHub repository in Source Code Management.

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

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



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



Execute SonarQube Scanner

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

Path to project properties ?
[Empty input field]

Analysis properties ?

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

Additional arguments ?
[Empty input field]

JVM Options ?
[Empty input field]

Then save

adv_devops_exp7

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- SonarQube
- Rename

SonarQube

Permalinks

- Last build (#2), 1 day 20 hr ago
- Last stable build (#2), 1 day 20 hr ago
- Last successful build (#2), 1 day 20 hr ago
- Last completed build (#2), 1 day 20 hr ago

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

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

IF CONSOLE OUTPUT FAILED:

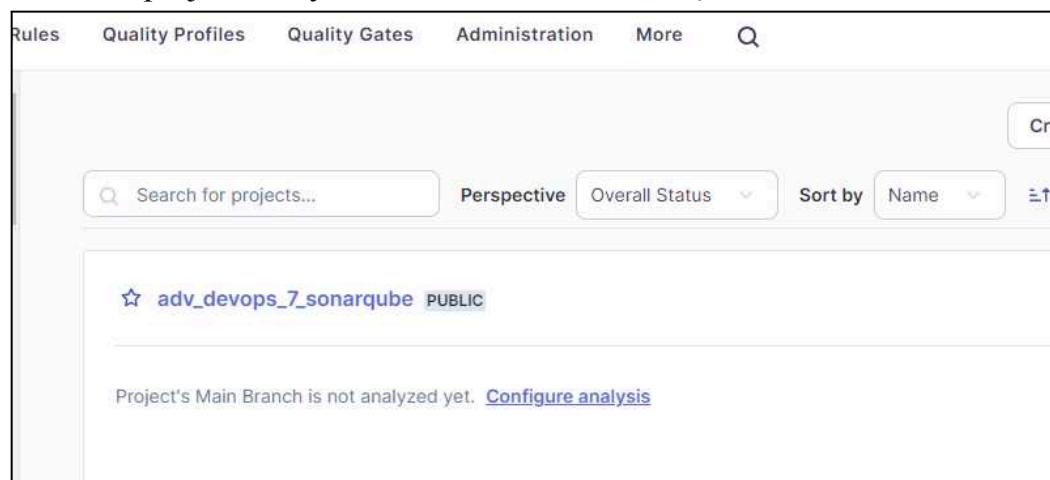
Step 1: Generate a New Authentication Token in SonarQube

1. Login to SonarQube:

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

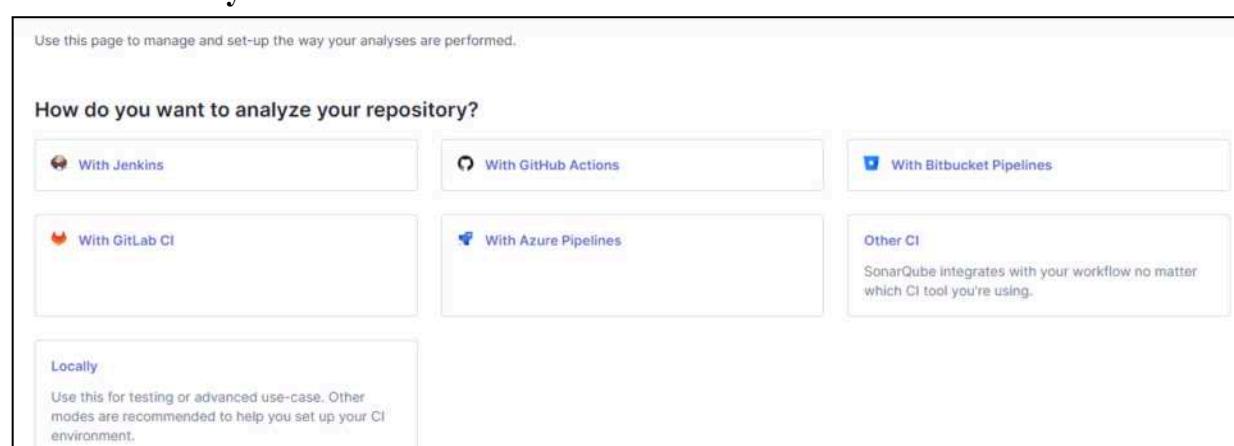
2. Generate a New Token:

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



The screenshot shows the SonarQube web interface. At the top, there is a navigation bar with links for Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation bar is a search bar labeled "Search for projects..." and a dropdown menu for "Perspective". Further down are dropdown menus for "Overall Status" and "Sort by", and a "Name" dropdown. The main content area displays a single project entry: "adv_devops_7_sonarqube PUBLIC". Below the project name, a message states: "Project's Main Branch is not analyzed yet. [Configure analysis](#)".

- Click on **Locally**



The screenshot shows the "How do you want to analyze your repository?" section of the SonarQube configuration page. There are several options: "With Jenkins", "With GitHub Actions", "With Bitbucket Pipelines", "With GitLab CI", "With Azure Pipelines", and "Other CI". The "Locally" option is highlighted with a blue border, indicating it is selected. A tooltip for "Locally" says: "Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment." To the right of the "Other CI" button, there is a note: "SonarQube integrates with your workflow no matter which CI tool you're using."

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

1 Provide a token

[Generate a project token](#) [Use existing token](#)

Token name [?](#) Expires in

"adv_devops_7.sonarqube"

1 year [Generate](#)

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

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

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

1 Provide a token

"adv_devops_7.sonarqube": sqp_bfa5258ea4fd254f00c3d1d4e64205ebefcdd027 [Delete](#)

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

[Continue](#)

Step 2: Update the Token in Jenkins

1. Go to Jenkins Dashboard:

- Open Jenkins and log in with your credentials.

The screenshot shows the Jenkins dashboard with the following details:

- Dashboard:** Shows the Jenkins logo and navigation links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views".
- Build Queue:** A dropdown menu.
- Build History:** A table showing build status (S), last build (L), pipeline (W), name, last success, last failure, and last duration.
- Table Data:**

| S | L | W | Name | Last Success | Last Failure | Last Duration |
|-----------------|------------|-----------|-----------|-----------------|--------------|---------------|
| Green checkmark | Yellow sun | Pipeline2 | Pipeline2 | 1 mo 23 days #4 | N/A | 1.4 sec |
| Green checkmark | Yellow sun | aryan27 | aryan27 | 9 days 8 hr #1 | N/A | 17 sec |
| Green checkmark | Yellow sun | aryan2711 | aryan2711 | 9 days 7 hr #1 | N/A | 22 sec |

2. Go to Dashboard—>Manage Jenkins—>Credentials

The screenshot shows the Jenkins 'Credentials' page under 'Manage Jenkins'. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Credentials. Below the header, the title 'Credentials' is displayed. A table lists one credential entry:

| T | P | Store ↓ | Domain | ID | Name |
|-----------|-----------|---------|----------|-----------------|--------|
| File icon | User icon | System | (global) | sonarqube_token | /***** |

Below the table, a section titled 'Stores scoped to Jenkins' is shown, containing a similar table:

| P | Store ↓ | Domains |
|-----------|-----------|---------|
| File icon | User icon | System |

At the bottom of the page, there are icons for 'Icon' and size options 'S', 'M', and 'L'.

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

The screenshot shows the 'Add Credential' form in Jenkins. The fields are as follows:

- Kind:** Secret text
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Secret:** (redacted)
- ID:** sonarqube-exp7
- Description:** advance devops exp7

At the bottom left is a blue 'Create' button.

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

The screenshot shows the 'Global credentials (unrestricted)' page under 'Manage Jenkins'. The title is 'Global credentials (unrestricted)' and there is a blue '+ Add Credentials' button. Below the title, a note says 'Credentials that should be available irrespective of domain specification to requirements matching.' A table lists the existing credential:

| ID | Name | Kind | Description |
|----------------------------|---------------------|-------------|---------------------|
| File icon sonarqube-exp | advance devops exp7 | Secret text | advance devops exp7 |

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

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

| | |
|-----------------------------|--|
| Name | adv_devops_7_sonarqube |
| Server URL | Default is http://localhost:9000 |
| | http://localhost:9000 |
| Server authentication token | SonarQube authentication token. Mandatory when anonymous access is disabled. |
| | advance devops exp7 |
| + Add ▾ | |

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

Build Environment

Delete workspace before build starts

Use secret text(s) or file(s) ?

Add timestamps to the Console Output

Inspect build log for published build scans

Prepare SonarQube Scanner environment ?

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

| | |
|-------------------------|---|
| advance devops exp7 | ▼ |
| + Add ▾ | |

Execute SonarQube Scanner

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

Path to project properties ?
[Empty text area]

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

Additional arguments ?
[Empty text area]

JVM Options ?
[Empty text area]

12. Run the Jenkins build.

Dashboard > adv_devops_exp7 >

Status (✓) adv_devops_exp7

- </> Changes
- Workspace
- ▷ Build Now
- ⚙ Configure
- trash Delete Project
- SonarQube
- pen Rename

SonarQube Quality Gate

adv_devops_7_sonarqube Passed
server-side processing: Success

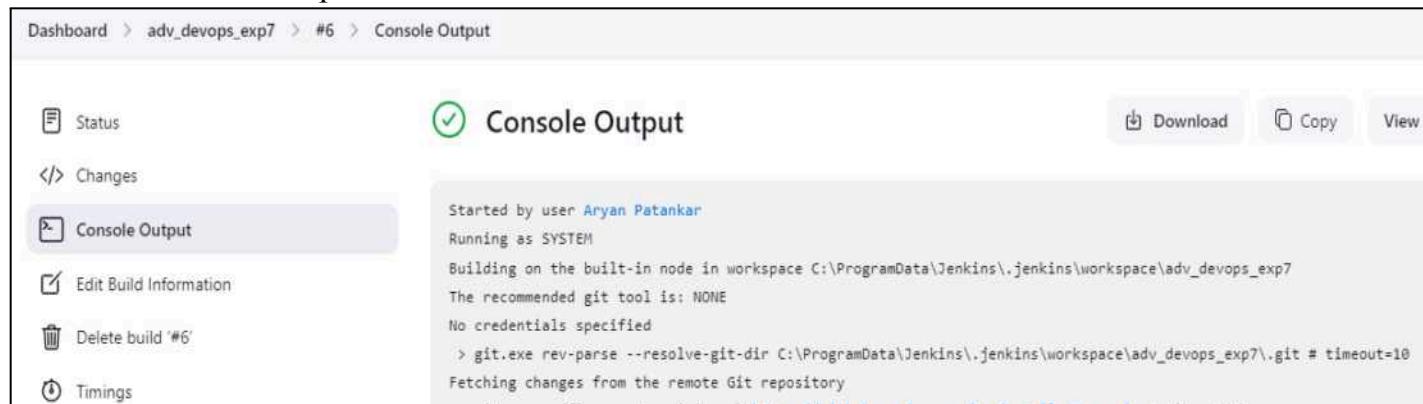
Permalinks

- [Last build \(#6\), 1 min 55 sec ago](#)
- [Last stable build \(#6\), 1 min 55 sec ago](#)
- [Last successful build \(#6\), 1 min 55 sec ago](#)
- [Last failed build \(#5\), 17 min ago](#)
- [Last unsuccessful build \(#5\), 17 min ago](#)
- [Last completed build \(#6\), 1 min 55 sec ago](#)

Build History trend /

#6 | Sep 25, 2024, 10:04 PM

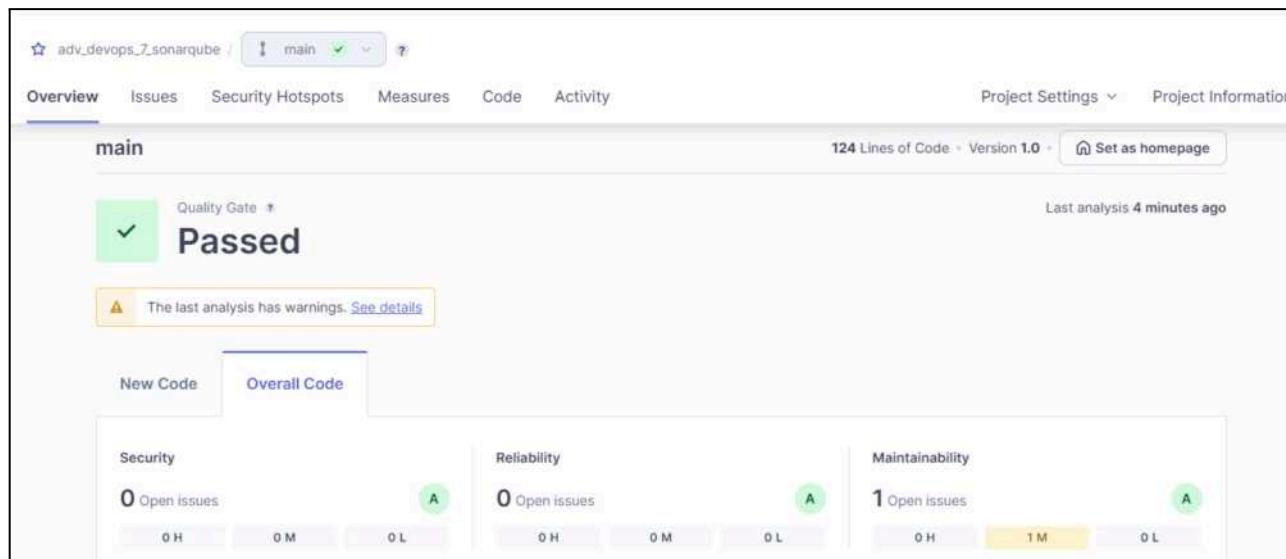
Check the console Output



The screenshot shows the Jenkins 'Console Output' page for build #6 of the 'adv_devops_exp7' project. The left sidebar includes links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#6', and Timings. The main content area displays the build configuration and logs:

```
Started by user Aryan Patankar
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\adv_devops_exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\adv_devops_exp7\.git # timeout=10
Fetching changes from the remote Git repository
```

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



The screenshot shows the SonarQube 'Project Overview' for the 'main' branch of the 'adv_devops_7.sonarqube' project. The top navigation bar includes links for Overview, Issues, Security Hotspots, Measures, Code, Activity, Project Settings, and Project Information. The main content area displays the Quality Gate status as 'Passed' (green checkmark) and the last analysis time as '4 minutes ago'. It also shows the number of lines of code (124 Lines of Code) and the version (Version 1.0). Below this, there are sections for New Code, Overall Code, Security, Reliability, and Maintainability, each showing the count of open issues and severity distribution (0 H, 0 M, 0 L).

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

ADVANCE DEVOPS EXP 8

Name:Manav punjabi

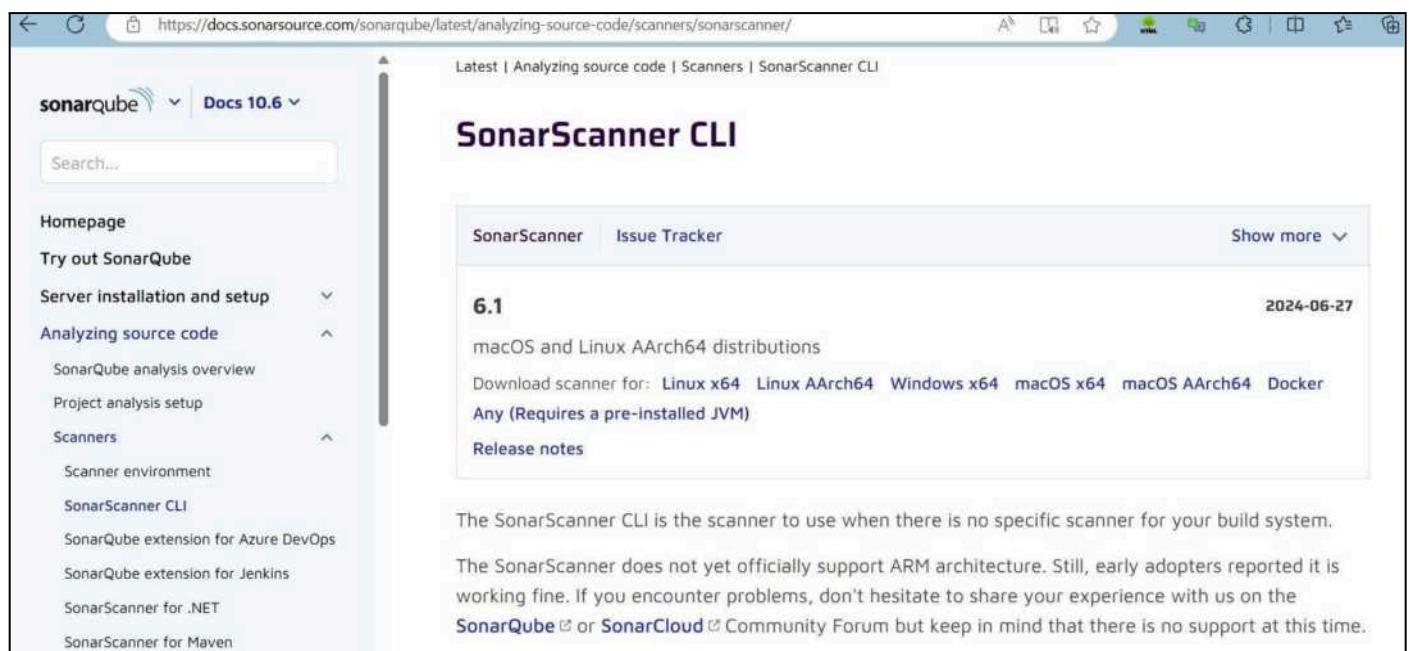
Class:D15A

Roll No:45

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

Step 1: Download sonar scanner

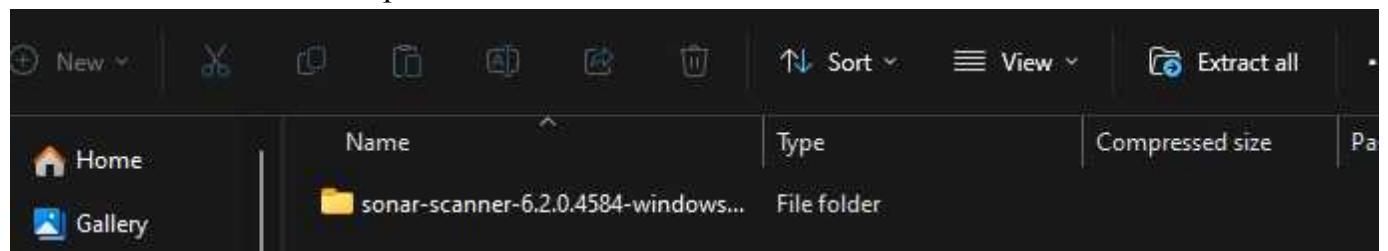
<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan>



The screenshot shows the 'SonarScanner CLI' page from the SonarQube documentation. The left sidebar has a 'Scanners' section expanded, showing options like 'Scanner environment', 'SonarScanner CLI', and 'SonarScanner for .NET'. The main content area displays the 'SonarScanner' tab selected. It features a '6.1' release note from 2024-06-27, which includes links for 'macOS and Linux AArch64 distributions' and download links for various platforms including Linux x64, Windows x64, macOS x64, macOS AArch64, Docker, and Any (Requires a pre-installed JVM). Below the release note, there's a note about ARM support and links to SonarQube and SonarCloud forums.

ner/ Visit this link and download the sonarqube scanner CLI.

Extract the downloaded zip file in a folder.



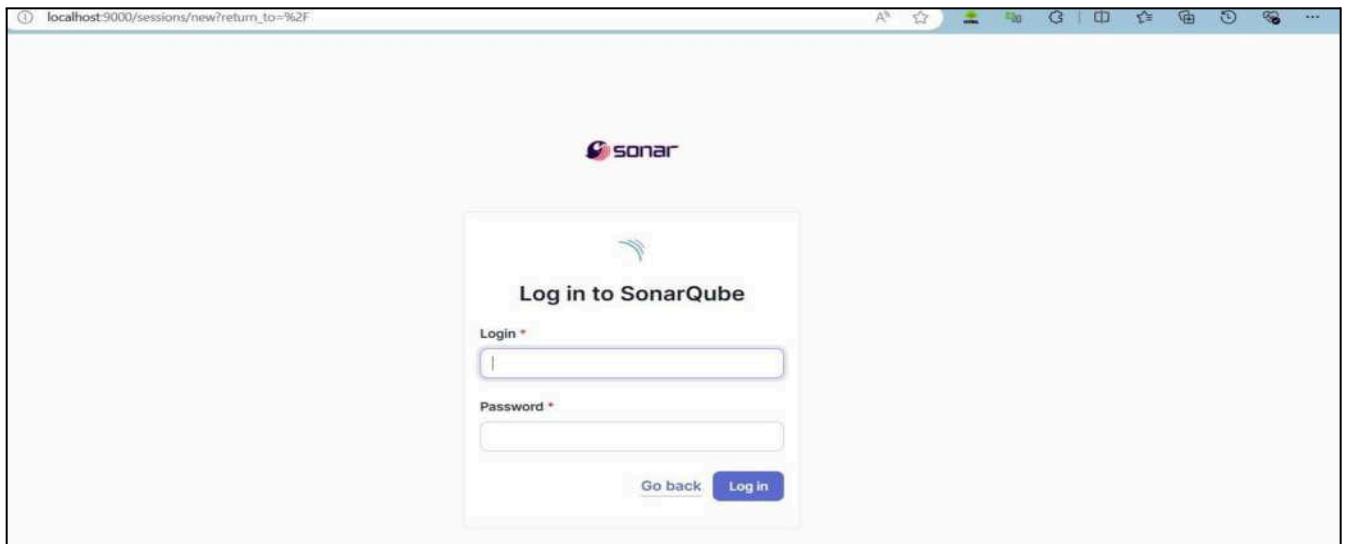
1. Install sonarqube image

Command: **docker pull**

sonarqube

```
C:\Windows\System32>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

- Once the container is up and running, you can check the status of



SonarQube at localhost port 9000.

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

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

How do you want to create your project?

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

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

Import from Azure DevOps Import from Bitbucket Cloud Import from Bitbucket Server
Import from GitHub Import from GitLab
Create a local project

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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

[Cancel](#)

[Next](#)

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 at 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.

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

The screenshot shows the Jenkins dashboard at localhost:8080. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. Below that are sections for 'Build Queue' (empty) and 'Build Executor Status' (one idle node). The main area displays a table of build jobs:

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|----|----------------------|-----------------|-----------------|---------------|
| ✓ | ☀️ | Devops Pipeline | 1 mo 13 days #4 | N/A | 0.61 sec |
| ✓ | ☀️ | devops_exp6_pipeline | 24 days #1 | N/A | 2.2 sec |
| ✓ | ☁️ | maven_exp_6 | 17 days #13 | 17 days #12 | 9.2 sec |
| ✗ | ☁️ | maven_project | 1 mo 13 days #3 | 1 mo 7 days #10 | 12 sec |
| ✓ | ☀️ | myNewJob | 24 days #1 | N/A | 0.49 sec |

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

The screenshot shows the 'Manage Jenkins > Plugins' page. The search bar contains 'sonarq'. The 'Available plugins' section is selected, showing the 'SonarQube Scanner' plugin. It has been released at version 2.17.2 on 6 mo 29 days ago. The description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' An 'Install' button is visible.

The screenshot shows the 'Manage Jenkins > Plugins' page with the 'Download progress' section selected. It shows the preparation steps: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Under 'SonarQube Scanner', it shows 'Success' for both 'SonarQube Scanner' and 'Loading plugin extensions'. There are links to 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

7. Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers**

and enter the details.

Enter the Server Authentication token if needed.

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

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



The screenshot shows the Jenkins configuration interface for SonarQube servers. It includes fields for Name (sonarqube), Server URL (http://localhost:9000), and a dropdown for Server authentication token (set to - none -). There are also 'Add' and 'Advanced' buttons.

| | |
|-----------------------------|---|
| Name | sonarqube |
| Server URL | Default is http://localhost:9000 http://localhost:9000 |
| Server authentication token | - none - + Add Advanced |

8. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

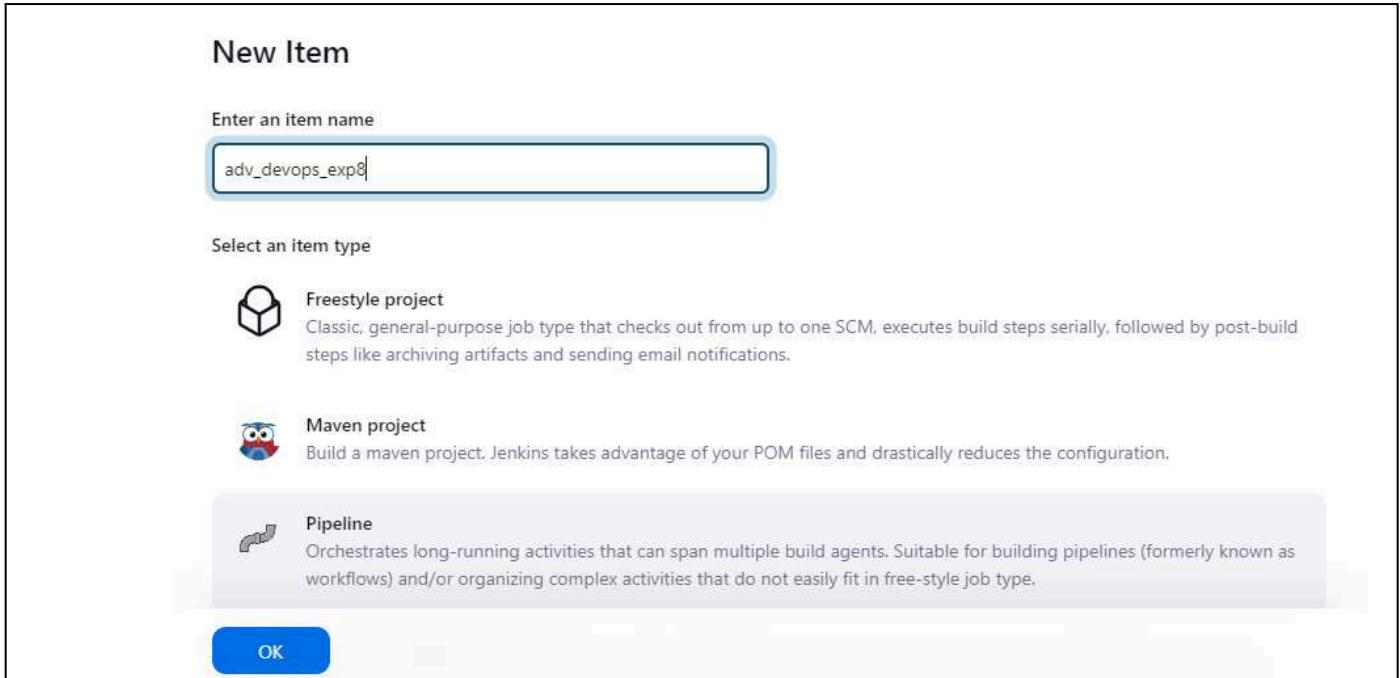
The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for 'Add Git', 'Gradle installations' (with 'Add Gradle'), 'SonarScanner for MSBuild installations' (with 'Add SonarScanner for MSBuild'), 'SonarQube Scanner installations' (with 'Add SonarQube Scanner'), and 'Ant installations'. The 'SonarQube Scanner installations' section is currently selected.

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

The screenshot shows the 'SonarQube Scanner' configuration dialog. It includes fields for 'Name' (set to 'sonarqube_exp8'), 'Install automatically' (checked), 'Install from Maven Central' (Version: 'SonarQube Scanner 6.2.0.4584'), and an 'Add Installer' dropdown. At the bottom is a 'Add SonarQube Scanner' button.

the “Install automatically” option.

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



10. Under Pipeline script, enter the following:

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

  stage('SonarQube analysis') {
    withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenki
ns>') {
      sh """
        <PATH_TO SONARQUBE SCANNER FOLDER>/bin/sonar-scanner \
        -D sonar.login=<SonarQube_USERNAME> \
        -D sonar.password=<SonarQube_PASSWORD> \
        -D sonar.projectKey=<Project_KEY> \
        -D sonar.exclusions=vendor/**,resources/**, **/*.java \
        -D sonar.host.url=<SonarQube_URL>(default: http://localhost:9000/)
      """
    }
  }
}
```

}

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

Definition

Pipeline script

Script ?

```
1 node {  
2 stage('Cloning the GitHub Repo') {  
3 git 'https://github.com/shazforiot/GOL.git'  
4 }  
5  
6 stage('SonarQube analysis') { withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {  
7 sh """  
8 <PATH_TO SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \  
9 -D sonar.login=admin \  
10 -D sonar.password=admin> \  
11 -D sonar.projectKey=sonarqube \  
12 -D sonar.exclusions=vendor/**,resources/**,**/*.java \  
13 -D sonar.host.url=http://localhost:9000  
14 """  
15 }  
16 }  
17 }  
18 }
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

11. Build project

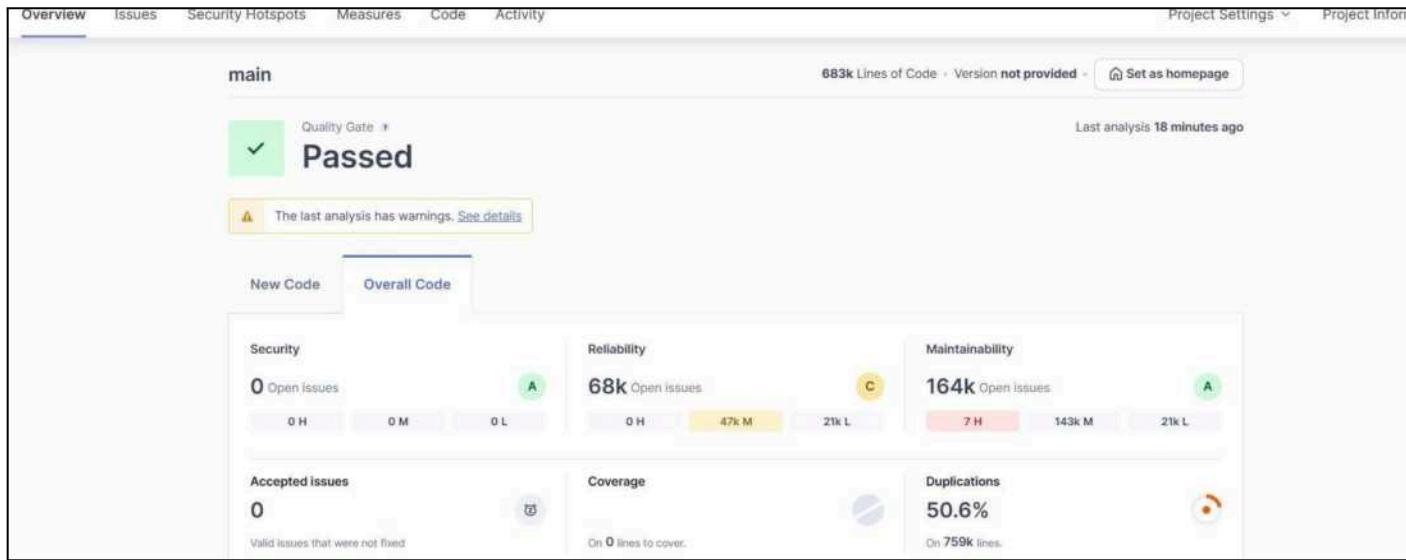
The screenshot shows the Jenkins Pipeline interface for the project 'adv_devops_exp8'. On the left, there's a sidebar with various options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. Below that is a 'Build History' section showing three builds: Sep 18 16:14 (No Changes), Sep 18 16:12 (No Changes), and Sep 18 16:10 (No Changes). The main area is titled 'Stage View' and displays a grid of stages. The first stage, 'Cloning the GitHub Repo', took 3s. The second stage, 'SonarQube analysis', took 40s and failed. The third stage, which appears to be another SonarQube analysis, took 1s and failed. The fourth stage, which appears to be another SonarQube analysis, took 120ms and failed. The overall average stage time is 3s, and the average full run time is approximately 6 minutes and 4 seconds.

12. Check console

The screenshot shows the Jenkins Pipeline Console Output for build #9. The sidebar on the left includes options like Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#9', Timings, Git Build Data, Pipeline Overview, Pipeline Console (selected), Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Console Output' and shows the following log entries:

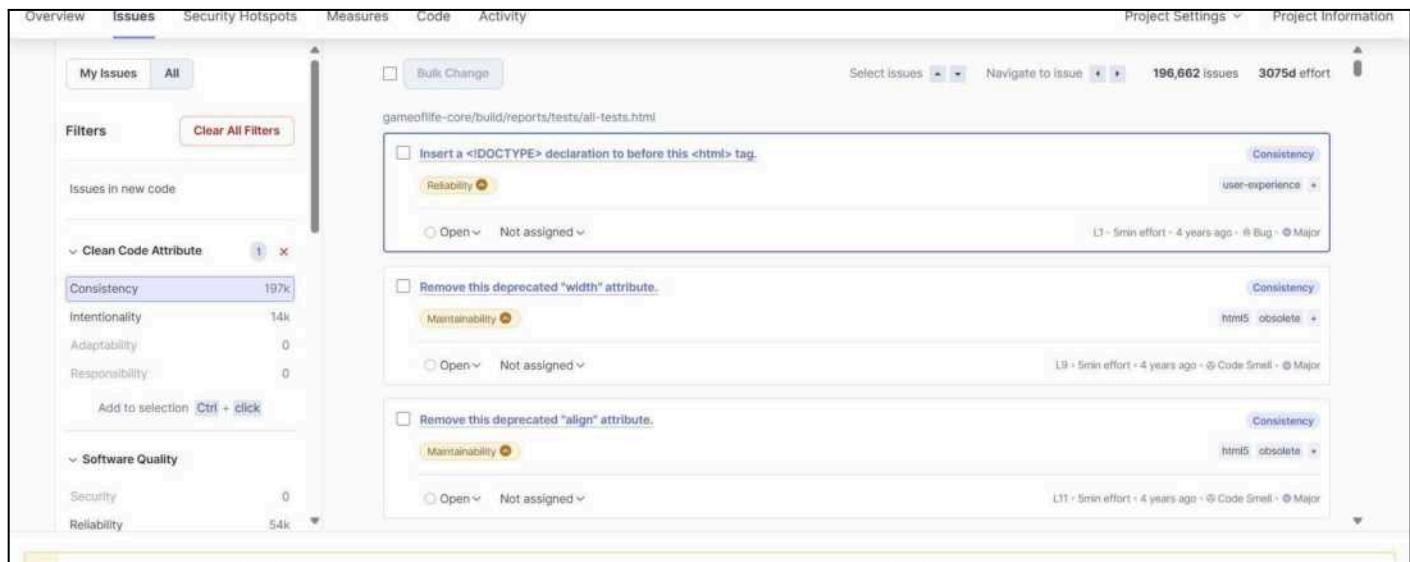
```
Skipping 4.246 KB.. Full Log
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 512. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 248. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 886. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 249. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 662. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 615. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 664. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 913. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 810. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 668. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 548. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 543. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 152. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 152. Keep only the first 100 references.
```

13. Now, check the project in SonarQube



14. Code Problems

- Consistency



● Intentionality

The screenshot shows a software interface for managing code quality and security. At the top, there are tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The Issues tab is selected. In the top right corner, it says "Project Settings" and "Project Information". Below the tabs, there are buttons for "My Issues" and "All". A "Bulk Change" button is also present. On the far right, it shows "13,887 issues" and "59d effort".

On the left side, there is a sidebar with a "Filters" section and a "Clear All Filters" button. Below that, under "Issues in new code", there is a section titled "Clean Code Attribute" with the following items:

- Consistency: 197k
- Intentionality: 14k** (This item is highlighted with a purple background)
- Adaptability: 0
- Responsibility: 0

Below this, there is a link "Add to selection Ctrl + click".

Under "Software Quality", there are two items:

- Security: 0
- Reliability: 14k

The main content area displays three specific issues related to Intentionality:

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

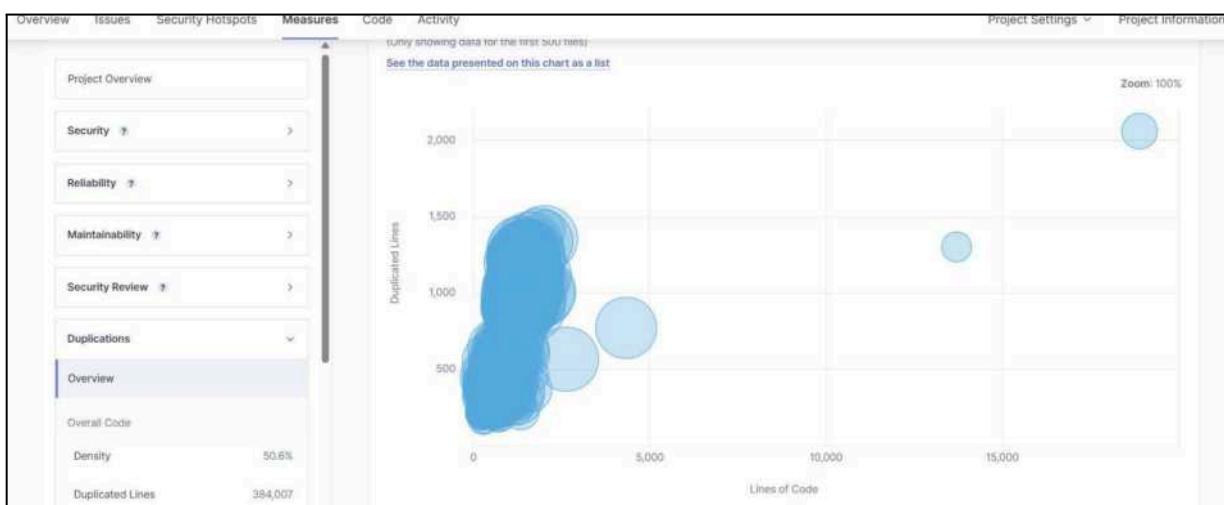
Bugs

The screenshot shows a list of bugs for the 'gameoflife-core' project. The top navigation bar includes 'Bulk Change', 'Select Issues', 'Navigate to Issue', and statistics: '67,624 issues' and '1646d effort'. The first issue is about adding 'lang' and/or 'xml:lang' attributes to an HTML element, categorized under 'Intentionality' (accessibility, wcag2-a). The second issue is about inserting a DOCTYPE declaration before the HTML tag, categorized under 'Consistency' (user-experience). The third issue is about adding <th> headers to a table, also under 'Intentionality' (accessibility, wcag2-a).

Code Smells

The screenshot shows a list of code smells for the 'gameoflife-acceptance-tests/Dockerfile'. The top navigation bar includes 'Bulk Change', 'Select Issues', 'Navigate to Issue', and statistics: '163,781 issues' and '1705d effort'. On the left, there are filters for 'Issues in new code' and 'Clean Code Attribute' (Consistency, Intentionality, Adaptability, Responsibility) and 'Software Quality' (Security, Reliability, Maintainability). The listed code smells include using a specific version tag for the image (Maintainability, No tags) and surrounding variables with double quotes (Maintainability, No tags).

Duplications



- Cyclomatic Complexities

The screenshot shows the SonarQube interface for a project named "gameoflife". The top navigation bar includes links for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The "Measures" tab is currently selected. On the left, a sidebar lists various metrics: Security, Reliability, Maintainability, Security Review, Duplications, Size, Complexity, and Cyclomatic Complexity (which is highlighted in blue). The main content area displays the "Cyclomatic Complexity" report, which has a total count of 1,112. Below this, a list of files is shown, each with its cyclomatic complexity value: gameoflife-acceptance-tests (18), gameoflife-build (18), gameoflife-core (18), gameoflife-deploy (18), gameoflife-web (1,094), and pom.xml (18). A note at the bottom indicates "6 of 6 shown".

| File | Cyclomatic Complexity |
|-----------------------------|-----------------------|
| gameoflife-acceptance-tests | 18 |
| gameoflife-build | 18 |
| gameoflife-core | 18 |
| gameoflife-deploy | 18 |
| gameoflife-web | 1,094 |
| pom.xml | 18 |

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

Experiment 9

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

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

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

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

EC2 > Instances > i-025f1d18f7c8a8cda > Connect to instance

Connect to instance Info

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

EC2 Instance Connect Session Manager SSH client EC2 serial console

All ports are open to all IPv4 addresses in your security group

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

Instance ID: [i-025f1d18f7c8a8cda](#) (nagios-host)

Connection Type:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address: [3.86.198.73](#)

IPv6 address: -

Username:

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

Step 3: Now, run the following commands -

```
sudo su
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
```

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

(ec2-user@ip-172-31-93-157 ~)$ sudo su
[root@ip-172-31-93-157 ec2-user]# sudo yum update
Last metadata expiration check: 0:11:38 ago on Mon Sep 30 16:39:07 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-93-157 ec2-user]# sudo yum install httpd php
Last metadata expiration check: 0:11:51 ago on Mon Sep 30 16:39:07 2024.
Dependencies resolved.

Package           Architecture Version       Repository      Size
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    46 k
php8_3           x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    10 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
pcre2c-logos-httdp noarch     18.0.0-12.amzn2023.0.3  amazonlinux    19 k
httpd-core       x86_64      2.4.62-1.amzn2023   amazonlinux    1.4 M
httpd-filesystem noarch     2.4.62-1.amzn2023   amazonlinux    14 k
httpd-tools       x86_64      2.4.62-1.amzn2023   amazonlinux    81 k
libbrotli        x86_64      1.0.9-4.amzn2023.0.2  amazonlinux    315 k
libsodium         x86_64      1.0.19-4.amzn2023   amazonlinux    176 k

i-025f1d18f7c8a8cda (nagios-host)
Public IPs: 3.86.198.73 Private IPs: 172.31.93.157
```

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

```

php8.3-x86_64.3.10-1.amzn2023.0.1.x86_64
=====
Complete!
[root@ip-172-31-93-157 ec2-user]# sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:12:22 ago on Mon Sep 30 16:39:07 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
=====
Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]: y

i-025f1d18f7c8a8cda (nagios-host)
PublicIP: 3.86.198.73 PrivateIP: 172.31.93.157

```

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

```

Complete!
[root@ip-172-31-93-157 ec2-user]# sudo yum install gd gd-devel
Last metadata expiration check: 0:13:10 ago on Mon Sep 30 16:39:07 2024.
Dependencies resolved.

Transaction Summary
=====
Install 30 Packages

Total download size: 1.39 G
Installed size: 3.86 G
Is this ok [y/N]: y

i-025f1d18f7c8a8cda (nagios-host)
PublicIP: 3.86.198.73 PrivateIP: 172.31.93.157

```

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

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

`sudo adduser -m nagios`

`sudo passwd nagios`

`sudo groupadd nagcmd`

`sudo usermod -a -G nagcmd nagios`

`sudo usermod -a -G nagcmd apache`

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

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

Step 5: Now, run the following commands -

```
mkdir ~/downloads
cd ~/downloads
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz
```

```
root@ip-172-31-93-157 ec2-user]# mkdir ~/downloads
root@ip-172-31-93-157 ec2-user]# cd ~/downloads
root@ip-172-31-93-157 downloads]# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
--2024-09-30 17:00:06-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: "nagios-plugins-2.0.3.tar.gz"

nagios-plugins-2.0.3.tar.gz      100%[=====]  2.54M  6.10MB/s   in 0.4s
2024-09-30 17:00:07 (6.16 MB/s) - "nagios-plugins-2.0.3.tar.gz" saved [2659772/2659772]

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

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

To resolve the error run the following commands -

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar zxvf nagios-4.0.8.tar.gz
tar zxvf nagios-plugins-2.0.3.tar.gz
cd nagios-4.0.8
```

```
[root@ip-172-31-93-157 downloads]# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
--2024-09-30 17:03:04-- http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net) ... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net) port 80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-30 17:03:04-- http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net) ... 204.68.111.105
Resolving existing connection to prdownloads.sourceforge.net:80...
HTTP request sent, awaiting response... 302 Found
Location: http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1 [following]
--2024-09-30 17:03:04-- http://versaweb.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1
Resolving versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net) ... 162.251.232.173
Connecting to versaweb.dl.sourceforge.net (versaweb.dl.sourceforge.net) port 80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.78M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

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

2024-09-30 17:03:05 (2.23 MB/s) - `nagios-4.0.8.tar.gz' saved [1805059/1805059]

--2024-09-30 17:03:05-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org) ... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org) port 80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.58M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz.l1'

nagios-plugins-2.0.3.tar.gz.l1 100%[=====]   2.54M  7.26MB/s  in 0.3s

2024-09-30 17:03:05 (7.26 MB/s) - `nagios-plugins-2.0.3.tar.gz.l1' saved [2659772/2659772]
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 · PrivateIPs: 172.31.93.157

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

```
nagios-plugins-2.0.3/plugins/scripts/check_ifoperstatus.pl
nagios-plugins-2.0.3/plugins/scripts/Makefile.am
nagios-plugins-2.0.3/plugins/scripts/subst.in
nagios-plugins-2.0.3/plugins/scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins/scripts/check_log.sh
nagios-plugins-2.0.3/plugins/scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins/scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins/scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins/scripts/utils.rn.in
nagios-plugins-2.0.3/plugins/scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins/scripts/t/
nagios-plugins-2.0.3/plugins/scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins/scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins/scripts/t/check_file_aget.t
nagios-plugins-2.0.3/plugins/scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins/scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins/scripts/t/utils.t
nagios-plugins-2.0.3/plugins/scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins/scripts/check_wave.pl
nagios-plugins-2.0.3/plugins/scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins/scripts/utils.sh.in
nagios-plugins-2.0.3/plugins/scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins/scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requirements
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requirements
[root@ip-172-31-93-157 downloads]# ]
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 · PrivateIPs: 172.31.93.157

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

```
nagios-plugins-2.0.3/plugins/scripts/Makefile.am
nagios-plugins-2.0.3/plugins/scripts/subst.in
nagios-plugins-2.0.3/plugins/scripts/check_breeze.pl
nagios-plugins-2.0.3/plugins/scripts/check_log.sh
nagios-plugins-2.0.3/plugins/scripts/check_flexlm.pl
nagios-plugins-2.0.3/plugins/scripts/check_rpc.pl
nagios-plugins-2.0.3/plugins/scripts/check_oracle.sh
nagios-plugins-2.0.3/plugins/scripts/utils.rn.in
nagios-plugins-2.0.3/plugins/scripts/check_disk_smb.pl
nagios-plugins-2.0.3/plugins/scripts/t/
nagios-plugins-2.0.3/plugins/scripts/t/check_ifoperstatus.t
nagios-plugins-2.0.3/plugins/scripts/t/check_rpc.t
nagios-plugins-2.0.3/plugins/scripts/t/check_file_aget.t
nagios-plugins-2.0.3/plugins/scripts/t/check_disk_smb.t
nagios-plugins-2.0.3/plugins/scripts/t/check_ifstatus.t
nagios-plugins-2.0.3/plugins/scripts/t/utils.t
nagios-plugins-2.0.3/plugins/scripts/check_mailq.pl
nagios-plugins-2.0.3/plugins/scripts/check_wave.pl
nagios-plugins-2.0.3/plugins/scripts/check_ircd.pl
nagios-plugins-2.0.3/plugins/scripts/utils.sh.in
nagios-plugins-2.0.3/plugins/scripts/check_ifstatus.pl
nagios-plugins-2.0.3/plugins/scripts/check_sensors.sh
nagios-plugins-2.0.3/pkg/
nagios-plugins-2.0.3/pkg/fedora/
nagios-plugins-2.0.3/pkg/fedora/requirements
nagios-plugins-2.0.3/pkg/solaris/
nagios-plugins-2.0.3/pkg/solaris/preinstall
nagios-plugins-2.0.3/pkg/solaris/solpkg
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in
nagios-plugins-2.0.3/pkg/solaris/pkginfo
nagios-plugins-2.0.3/pkg/redhat/
nagios-plugins-2.0.3/pkg/redhat/requirements
[root@ip-172-31-93-157 downloads]# cd nagios-4.0.8
[root@ip-172-31-93-157 nagios-4.0.8]# ]
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 · PrivateIPs: 172.31.93.157

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

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

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

```
[root@ip-172-31-93-157 nagios-4.0.8]# ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -O... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets ${MAKE}... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/egrep -E
checking for ANSI C header files... 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
checking ctype.h usability... yes
checking ctype.h presence... yes
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.75 PrivateIPs: 172.31.93.157

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

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

```
[root@ip-172-31-93-157 nagios-4.0.8]# make all
cd ./base && make
make[1]: Entering directory '/root/downloads/nagios-4.0.8/base'
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o broker.o broker.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o nelmods.o nelmods.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o nerd.o nerd.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:224:12:
workers.c:209:17: warning: '\s' directive argument is null [-Wformat-overflows]
  209 |     log_debug_info(DEBUG_CHECKS, 1, "Found specialized worker(s) for '%s', (%lash && *slash != '/') ? slash : cmd_name");
  |             ~~~~~~
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o checks.o checks.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o config.o config.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o commands.o commands.c
commands.c: In function 'process_passive_service_check':
commands.c:2247:19: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
2247 |     cr.source = command_worker.source_name;
  |             ^
commands.c: In function 'process_passive_host_check':
commands.c:2339:19: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
2339 |     cr.source = command_worker.source_name;
  |             ^
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o events.o events.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o logging.o logging.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o macros-base.o ../common/macros.c
gcc -Wall -I... -g -O2 -DRAVE_CONFIG_B -DNSCORE -c -o netutils.o netutils.c
netutils.c: In function 'my_tcp_connect':
netutils.c:50:47: warning: '\d' directive output may be truncated writing between 1 and 11 bytes into a region of size 6 [-Wformat-truncation=]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.75 PrivateIPs: 172.31.93.157

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

```
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install
cd ./base; make install
make[1]: Entering directory '/root/downloads/nagios-4.0.8/base'
make install-basic
make[2]: Entering directory '/root/downloads/nagios-4.0.8/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/base'
make strip-post-install
make[2]: Entering directory '/root/downloads/nagios-4.0.8/base'
/usr/bin/strip /usr/local/nagios/bin/nagios
/usr/bin/strip /usr/local/nagios/bin/nagiosstats
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/base'
make[1]: Leaving directory '/root/downloads/nagios-4.0.8/base'
cd ./cgi; make install
make[1]: Entering directory '/root/downloads/nagios-4.0.8/cgi'
make install-basic
make[2]: Entering directory '/root/downloads/nagios-4.0.8/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin/ \
done
/usr/bin/install: cannot stat '**.cgi': No such file or directory
make[2]: *** [Makefile:205: install-basic] Error 1
make[2]: Leaving directory '/root/downloads/nagios-4.0.8/cgi'
make[1]: *** [Makefile:197: install] Error 2
make[1]: Leaving directory '/root/downloads/nagios-4.0.8/cgi'
make: *** [Makefile:235: install] Error 2
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios
*** init script installed ***

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

```
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios
*** Init script installed ***

```

```
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/templates.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/timperiods.cfg /usr/local/nagios/etc/objects/timperiods.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -m 664 -o nagios -g nagios sample-config/template-object/swtch.cfg /usr/local/nagios/etc/objects/swtch.cfg
*** config files installed ***

```

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

```
[root@ip-172-31-93-157 nagios-4.0.8]# sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
*** external command directory configured ***

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

```
[root@ip-172-31-93-157 nagios-4.0.8]# 
```

To resolve the errors run the following commands -

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

`rm -rf nagios-4.0.8`

`cd ~/downloads/nagios-4.4.6`

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

`make all`

`sudo make install`

```

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

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

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

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

*****  

Enjoy,  

root@ip-172-31-93-157 nagios-4.4.6] 

```

Step 8: 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
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                2022.anuprita.mhapankar@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

CONTACT GROUPS
[

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

define contactgroup{
    contactgroup_name    admins
    alias               Nagios Administrators
    members             nagiosadmin
}

  Help      Write Out  Where Is  Cut  Execute  Location  Undo  Set Mark  To Bracket  Previous
  Exit      -P Read File  -Replace  -I Paste  -J Justify  -L Go To Line  -U Redo  -C Copy  -Q Where Was  -N Next

```

Step 9: Now run the following commands –
 sudo make install-webconf
 sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
 sudo service httpd restart
 cd ~/downloads
 tar zxvf nagios-plugins-2.0.3.tar.gz

```
- Read the documentation on the Nagios Library at:  
  https://library.nagios.com  
  
before you post a question to one of the mailing lists.  
Also make sure to include pertinent information that could  
help others help you. This might include:  
  
- What version of Nagios you are using  
- What version of the plugins you are using  
- Relevant snippets from your config files  
- Relevant error messages from the Nagios log file  
  
For more information on obtaining support for Nagios, visit:  
  https://support.nagios.com  
*****  
Enjoy.  
  
(root@ip-172-31-93-157 nagios-4.4.6) # sudo nano /usr/local/nagios/etc/objects/contacts.cfg  
(root@ip-172-31-93-157 nagios-4.4.6) # sudo make install-webconf  
/usr/bin/install -d -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf  
if ! 0 -eq 1 ; then \  
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf \  
fi  
*** Nagios/Apache conf file installed ***  
  
(root@ip-172-31-93-157 nagios-4.4.6) # sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
New password:  
Re-type new password:  
Adding password for user nagiosadmin  
(root@ip-172-31-93-157 nagios-4.4.6) #  
  
i-025f1d18f7c8a8cda (nagios-host)  
PublicIP: 3.86.198.73 PrivateIP: 172.31.93.157
```

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

```
(root@ip-172-31-93-157 nagios-4.4.6) # sudo service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
(root@ip-172-31-93-157 nagios-4.4.6) # cd ~/downloads  
tar xvzf nagios-plugins-2.0.3.tar.gz  
nagios-plugins-2.0.3/  
nagios-plugins-2.0.3/perlmods/  
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz  
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz  
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile.in  
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile.am  
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz  
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz  
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz  
nagios-plugins-2.0.3/perlmods/try-Tiny-0.18.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Implementation-0.07.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile  
nagios-plugins-2.0.3/perlmods/Perl-OSType-1.003.tar.gz  
nagios-plugins-2.0.3/perlmods/install_order  
nagios-plugins-2.0.3/perlmods/Nagios-Plugin-0.36.tar.gz  
nagios-plugins-2.0.3/perlmods/Math-Calc-Units-1.07.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Build-0.4007.tar.gz  
nagios-plugins-2.0.3/ABOUT-NLS  
nagios-plugins-2.0.3/configure.ac  
nagios-plugins-2.0.3/Makefile.in  
nagios-plugins-2.0.3/config.h.in  
nagios-plugins-2.0.3/Changelog  
nagios-plugins-2.0.3/AUTHORS  
nagios-plugins-2.0.3/lib/  
nagios-plugins-2.0.3/lib/parse_ini.h  
nagios-plugins-2.0.3/lib/extr4_opts.c  
nagios-plugins-2.0.3/lib/Makefile.in
```

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

Step 10: Compile and install plugins

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

```

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

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

Step 11: To start nagios run the following commands –
sudo chkconfig --add nagios

sudo chkconfig nagios on

Verify using the following command -

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

```

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

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

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

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

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

If there are no errors run the following command –
sudo service nagios start

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

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

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

```
Total Warnings: 0
```

```
Total Errors: 0
```

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

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

```
i-025f1d18f7c8a8cda (nagios-host)
```

```
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157
```

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

Check status using the following command -
sudo systemctl status nagios

```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios)
   Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
     Docs: man:/etc/nagios/nagios.cfg
   Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
   Memory: 2.2M
      CPU: 52ms
      CPU: 52ms
     CGroup: /system.slice/nagios.service
             ├─00009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─00011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─00012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─00013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─00014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─00027 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: wproc: Successfully registered manager as #wproc with query handler
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: wproc: Registry request: name=Core Worker 80011;pid=80011
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: wproc: Registry request: name=Core Worker 80014;pid=80014
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: wproc: Registry request: name=Core Worker 80013;pid=80013
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: wproc: Registry request: name=Core Worker 80012;pid=80012
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[00009]: Successfully launched command file worker with pid 80037
lines: 1-26/26 (END)
```

```
i-025f1d18f7c8a8cda (nagios-host)
```

```
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157
```

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

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

Screenshot of the AWS EC2 Instances page showing a single instance named "nagios-host".

Instances (1/1) Info

Last updated about 1 hour ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

| Name | instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPs |
|-------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|------------|
| nagios-host | i-025f1d18f7c8a8cda | Running | t2.micro | 2/2 checks passed | View alarm | us-east-1c | ec2-3-86- |

i-025f1d18f7c8a8cda (nagios-host)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary

Instance ID: i-025f1d18f7c8a8cda (nagios-host)
 IPv6 address: -
 Hostname type: IP name: ip-172-31-93-157.ec2.internal
 Answer private resource DNS name: -

Public IPv4 address copied
 Instance state: Running
 Private IP DNS name (IPv4 only): ip-172-31-93-157.ec2.internal
 Instance type: t2.micro
 Elastic IP addresses: 172.31.93.157
 Public IPv4 DNS: ec2-3-86-198-73.compute-1.amazonaws.com | open address

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

Step 13: Now visit http://<your_public_ip_address>/nagios Enter correct credentials and then you will see this page.

Screenshot of the Nagios Core web interface.

Nagios® Core

Not secure | 3.86.198.73/nagios/

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Help

Problems

- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages

Reports

- Availability
- Trends (Legacy)
- Alerts
- Metrics
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

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)

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

Nagios® Core™ Version 4.4.6
 April 28, 2020
[Check for updates](#)

Latest News

Don't Miss...

Copyright © 2019-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

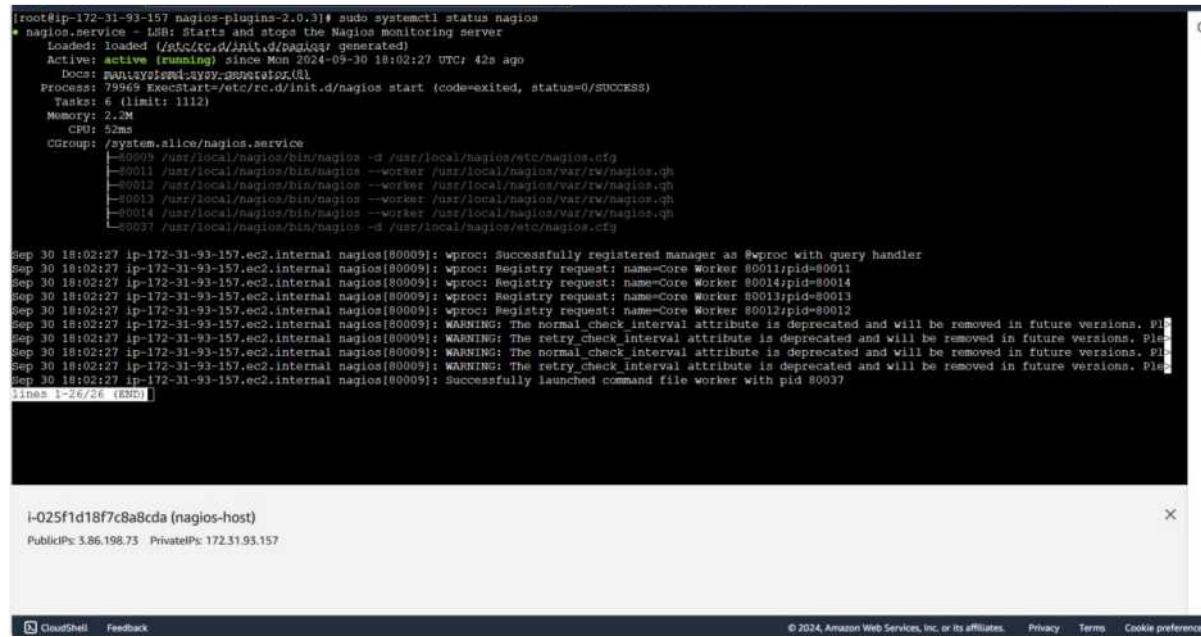
Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, service marks, registered trademarks or registered service marks owned by Nagios Enterprise, LLC. Use of the Nagios mark is governed by the trademark use restrictions.

Nagios | [NAGIOS.ORG](https://nagios.org) | SOURCEFORGE.NET

Experiment 10

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

Step 1: Initially confirm that Nagios is running on the server side. For this run the following command -
sudo systemctl status nagios
on the nagios-host instance.



```
root@ip-172-31-93-157 nagios-plugins-2.0.3]# sudo systemctl status nagios
nagios.service - LSB: starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/systemd/system/nagios.service)
   Active: active (running) since Mon 2024-05-30 18:02:27 UTC; 42s ago
     Docs: man:/etc/nagios3/doc/nagios.cfg
    Process: 79969 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
      Memory: 2.2M
         CPU: 52ms
      CGroup: /system.slice/nagios.service
              ├─79979 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─79981 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
              ├─79982 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
              ├─79983 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
              ├─79984 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
              └─79987 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

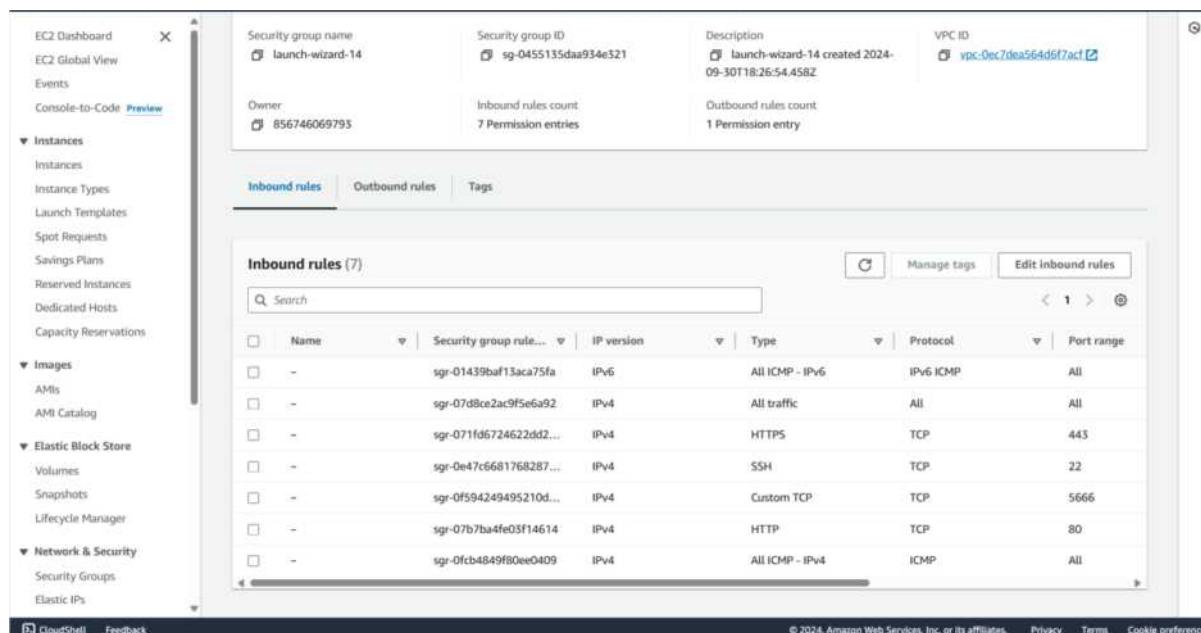
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: wproc: Successfully registered manager as #wproc with query handler
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: wproc: Registry request: name=Core Worker 80011;pid=80011
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: wproc: Registry request: name=Core Worker 80012;pid=80014
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: wproc: Registry request: name=Core Worker 80013;pid=80013
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: wproc: Registry request: name=Core Worker 80012;pid=80012
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use check_retries instead.
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use check_retries instead.
May 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[8000]: Successfully launched command file worker with pid 80037
lines 1-26/26 (END)
```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

Step 2: Once confirmed, make another instance with the same security group as that of nagios-host.
For now, leave this machine as it is, and go back to your nagios-host machine.



| Inbound rules (7) | Name | Security group rule ID | IP version | Type | Protocol | Port range |
|------------------------|------|------------------------|------------|-----------------|-----------|------------|
| sgr-01439ba13aca75fa | - | sgr-01439ba13aca75fa | IPv6 | All ICMP - IPv6 | IPv6 ICMP | All |
| sgr-07d8ce2ac95e6a92 | - | sgr-07d8ce2ac95e6a92 | IPv4 | All traffic | All | All |
| sgr-071fd6724622dd2... | - | sgr-071fd6724622dd2... | IPv4 | HTTPS | TCP | 443 |
| sgr-0e47c6681768287... | - | sgr-0e47c6681768287... | IPv4 | SSH | TCP | 22 |
| sgr-0f594249495210d... | - | sgr-0f594249495210d... | IPv4 | Custom TCP | TCP | 5666 |
| sgr-07b7ba4fe03f14614 | - | sgr-07b7ba4fe03f14614 | IPv4 | HTTP | TCP | 80 |
| sgr-0ftcb4849f80ee0409 | - | sgr-0ftcb4849f80ee0409 | IPv4 | All ICMP - IPv4 | ICMP | All |

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

Step 3: Now run the following command -

ps -ef | grep nagios

```

Active: active (running) since Mon 2024-09-30 18:02:27 UTC; 42s ago
  Docs: man:systemctl(1)
Process: 79569 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
 Tasks: 6 (limit: 1112)
Memory: 2.2M
 CPU: 52ms
 cGroup: /system.slice/nagios.service
├─80009 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
  ├─80011 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
  ├─80012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
  ├─80013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
  ├─80014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
  └─80037 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Successfully registered manager as #wproc with query handler
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80011;pid=80011
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80014;pid=80014
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80013;pid=80013
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: wproc: Registry request: name=Core Worker 80012;pid=80012
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please note that the check_interval attribute is the preferred attribute.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please note that the check_interval attribute is the preferred attribute.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please note that the check_interval attribute is the preferred attribute.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please note that the check_interval attribute is the preferred attribute.
Sep 30 18:02:27 ip-172-31-93-157.ec2.internal nagios[80009]: Successfully launched command file workers with pid 80037

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

```

Step 4: Now, run the following commands -

```

sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```

```

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

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

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

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

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)

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

Change hostgroup_name under hostgroup to linux-servers1

```
GNU nano 5.2                               /usr/local/nagios/etc/objects/monitorshosts/linuxserver.cfg                         Modified: ②
example of how you can create configuration entries to monitor
the local (linux) machine.

#####
# HOST DEFINITION
#####

# Define a host for the local machine

define host{
    use           linux-server          ; Name of host template to use
                                         ; This host definition will inherit all variables that are defined
                                         ; in (or inherited by) the linux-server host template definition.

    host_name     linux-server
    alias         linux-server
    address       3.86.198.73
}

#####

```

```
GNU nano 5.2                               /usr/local/nagios/etc/objects/monitorshosts/linuxserver.cfg                         Modified: ②
check_command      check_local_swap!20!10

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

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

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

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


```

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

Then add this line -

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

```

OS: nano 5.8
File: /usr/local/nagios/etc/objects/commands.cfg
File: /usr/local/nagios/etc/objects/contacts.cfg
File: /usr/local/nagios/etc/objects/timperiods.cfg
File: /usr/local/nagios/etc/objects/templates.cfg

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

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

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

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

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

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

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

  Help   Write Out  Where Is   Cut   Execute   Location   Undo   Set Mark   To Bracket   Previous
  Exit   Read File  Replace   Paste  Justify   Go To Line  Redo   Copy   Where Was   Next

```

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 · PrivateIPs: 172.31.93.157

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

Step 8: Verify configuration files using the following command -
 sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

If there are no errors, run the following command -
 sudo service nagios start

```

(root@ip-172-31-93-157 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPLv3

Website: https://www.nagios.org
Reading configuration data...
Read main config file okay...
WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
WARNING: The normal_check_interval attribute is deprecated and will be removed in future versions. Please use check_interval instead.
WARNING: The retry_check_interval attribute is deprecated and will be removed in future versions. Please use retry_interval instead.
Error: Could not find any host matching 'linuxserver' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 45)
Error: Could not expand members specified in hostgroup (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 45)
  Error processing object config files!

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

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

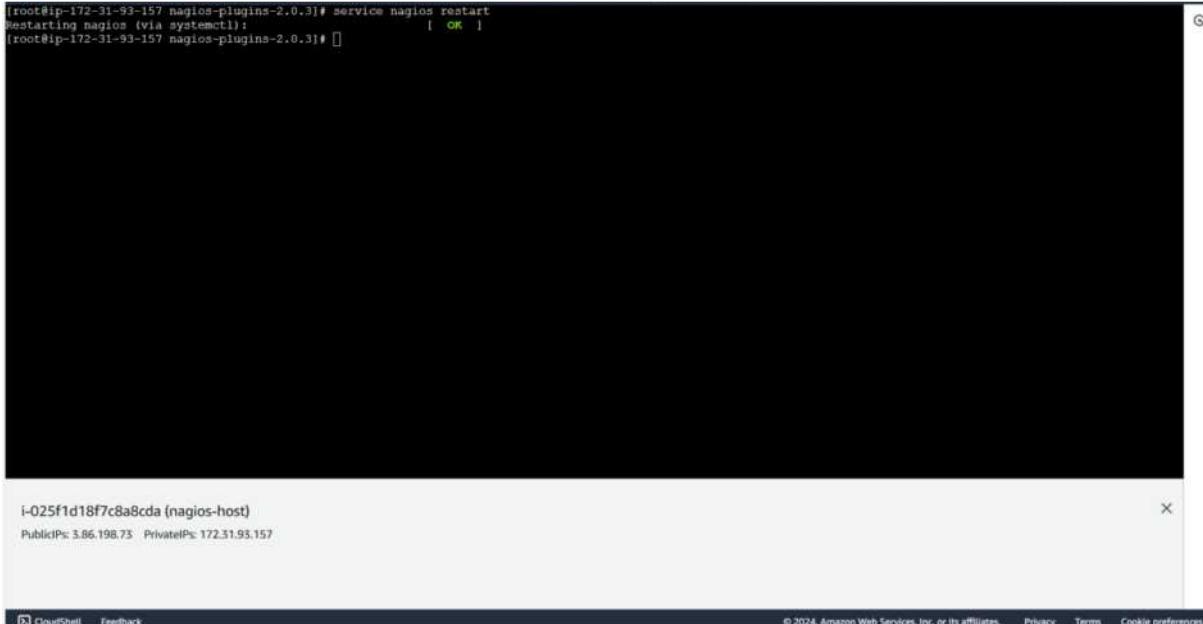
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# []

i-025f1d18f7c8a8cda (nagios-host)

PublicIPs: 3.86.198.73 · PrivateIPs: 172.31.93.157
```

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

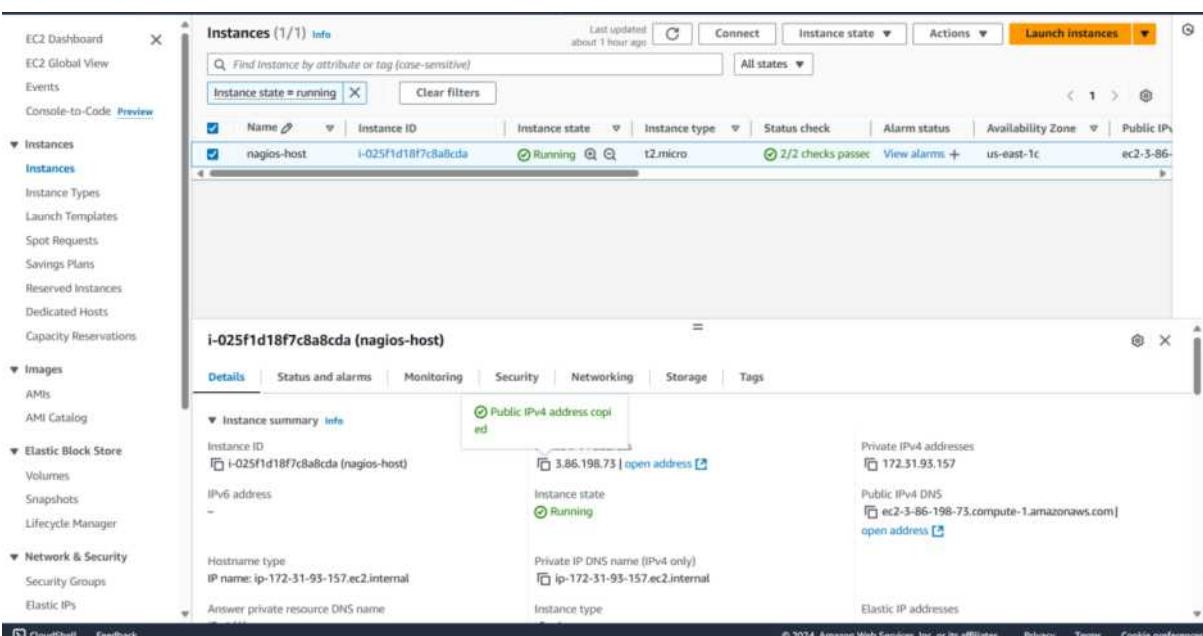
```
[root@ip-172-31-93-157 nagios-plugins-2.0.3]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-93-157 nagios-plugins-2.0.3]#
```



i-025f1d18f7c8a8cda (nagios-host)
PublicIPs: 3.86.198.73 PrivateIPs: 172.31.93.157

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

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



EC2 Dashboard EC2 Global View Events Console-to-Code [Preview](#)

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

Images AMIs AMI Catalog

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs

Instances (1/1) [Info](#) Last updated about 1 hour ago Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IP |
|-------------|---------------------|----------------|---------------|-------------------|---------------|-------------------|-----------|
| nagios-host | i-025f1d18f7c8a8cda | Running | t2.micro | 2/2 checks passed | View alarms + | us-east-1c | ec2-3-86- |

i-025f1d18f7c8a8cda (nagios-host)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary [Info](#) Public IPv4 address copied

| | | | | | |
|---|-----------------------------------|--|---|--|-----------------------|
| Instance ID: i-025f1d18f7c8a8cda (nagios-host) | IPV6 address: - | Private IP: 3.86.198.73 open address | Instance state: Running | Private IP DNS name (IPv4 only): ip-172-31-93-157.ec2.internal | Elastic IP addresses: |
| Hostname type: IP name: ip-172-31-93-157.ec2.internal | Answer private resource DNS name: | Public IP: 172.31.93.157 | Public IP DNS: ec2-3-86-198-73.compute-1.amazonaws.com open address | | |

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

Not secure 3.86.198.73/nagios/

Nagios*

General
Home Documentation

Current Status
General Overview
Map (Legacy)
Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search

Results 1 - 2 of 2 Matching Hosts

| Host | Status | Last Check | Duration | Status Information |
|-----------|--------|---------------------|---------------|---|
| localhost | UP | 09-30-2024 19:13:16 | 0d 0h 8m 33s+ | PING OK - Packet loss = 0%, RTA = 1.02 ms |
| localhost | UP | 09-30-2024 19:11:49 | 0d 1h 11m 20s | PING OK - Packet loss = 0%, RTA = 0.94 ms |

Host Status Details For All Host Groups

Host Status Totals
Last Updated: Mon Sep 30 19:13:49 UTC 2024
Nagios Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin

| All Problems | All Types |
|--------------|-----------|
| 2 | 2 |

Service Status Totals
All Problems All Types

| Ok | Warning | Unknown | Critical | Pending |
|----|---------|---------|----------|---------|
| 6 | 1 | 0 | 1 | 8 |

Limit Results: 100 ▾

Reports Availability Trends (Legacy) Alerts History Summary Histogram (Legacy) Notifications Event Log

System Comments Downtime Process Info Performance Info Scheduling Queue Configuration

Page Top

ADVANCE DEVOPS EXP 11

Name:Manav Punjabi

Class: D15A

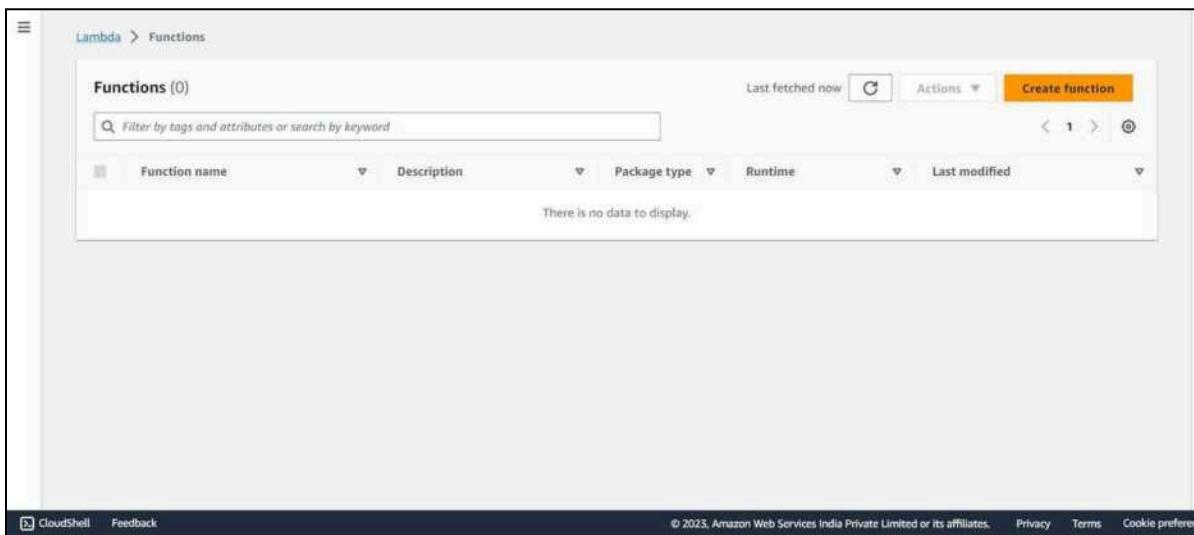
Roll No:45

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

Steps to create an AWS Lambda function

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

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



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

Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones. After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

Lambda > Functions > Create function

Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch Start with a simple Hello World example.

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

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

Basic information

Function name Enter a name that describes the purpose of your function.
myFunctionName
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 18.x

Architecture Info Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

CloudShell Feedback

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Lambda > Functions > Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch Start with a simple Hello World example.

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

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

Basic information

Function name Enter a name that describes the purpose of your function.
myPythonLambdaFunction
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.11

Architecture Info Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info

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

<https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/create/app> © 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Lambda > Functions > Create function

Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch Start with a simple Hello World example.

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

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

Basic information

Function name Enter a name that describes the purpose of your function.
myPythonLambdaFunction
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.11

Architecture Info Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info

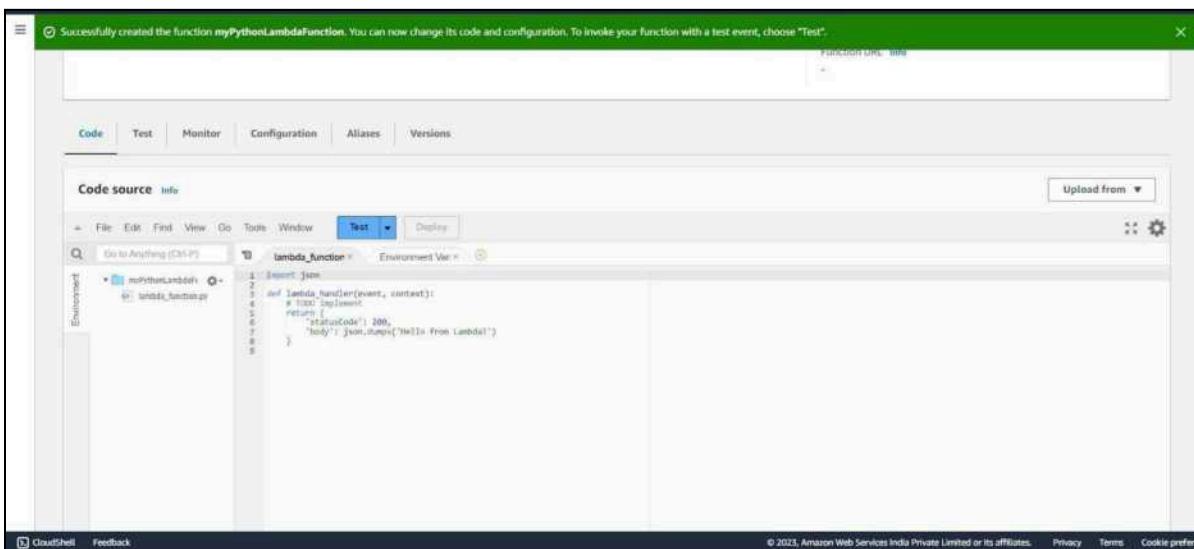
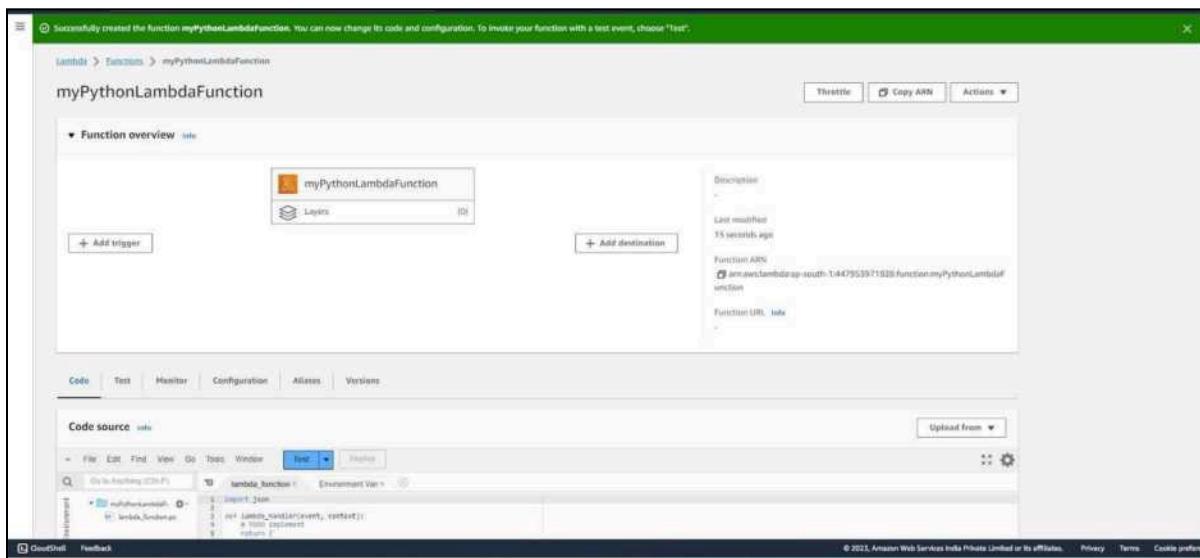
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Change default execution role Advanced settings

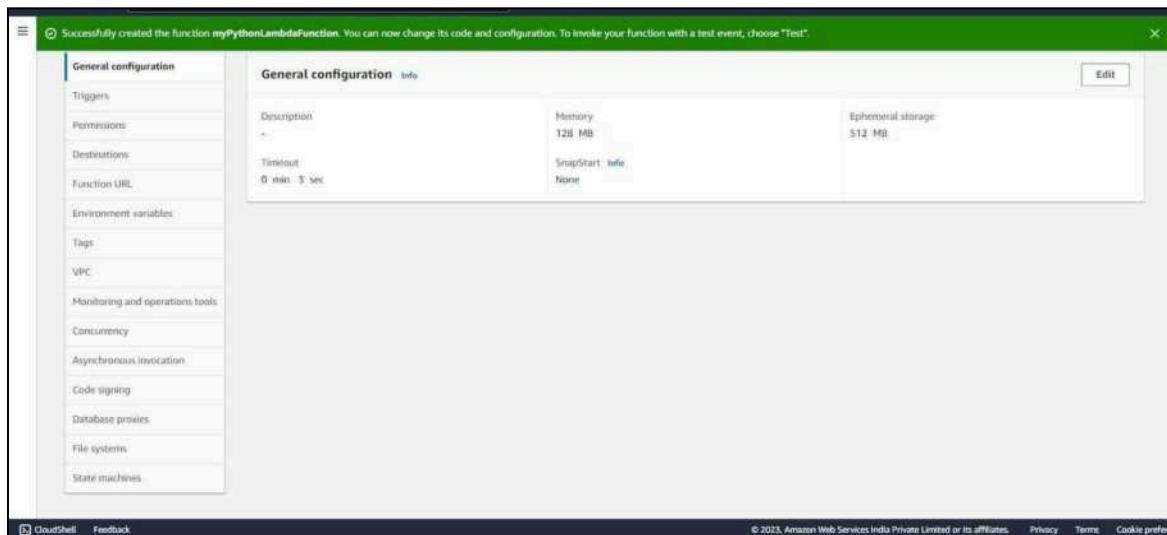
Cancel Create function

Click on the Create button.

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



4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit. Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



The screenshot shows the 'Edit basic settings' page for the 'myPythonLambdaFunction'. The top navigation bar includes the AWS logo, Services, a search bar, and a keyboard shortcut [Alt+S]. The breadcrumb navigation shows: Lambda > Functions > myPythonLambdaFunction > Edit basic settings.

Edit basic settings

Basic settings [Info](#)

Description - optional
A text input field for entering a description, currently empty.

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.
A numeric input field set to 128 MB, with a note below stating: "Set memory to between 128 MB and 10240 MB".

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)
A numeric input field set to 512 MB, with a note below stating: "Set ephemeral storage (/tmp) to between 512 MB and 10240 MB".

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).
A dropdown menu currently set to "None".

Supported runtimes: Java 11, Java 17.

Timeout
A numeric input field for minutes (0) and a numeric input field for seconds (1).

Execution role

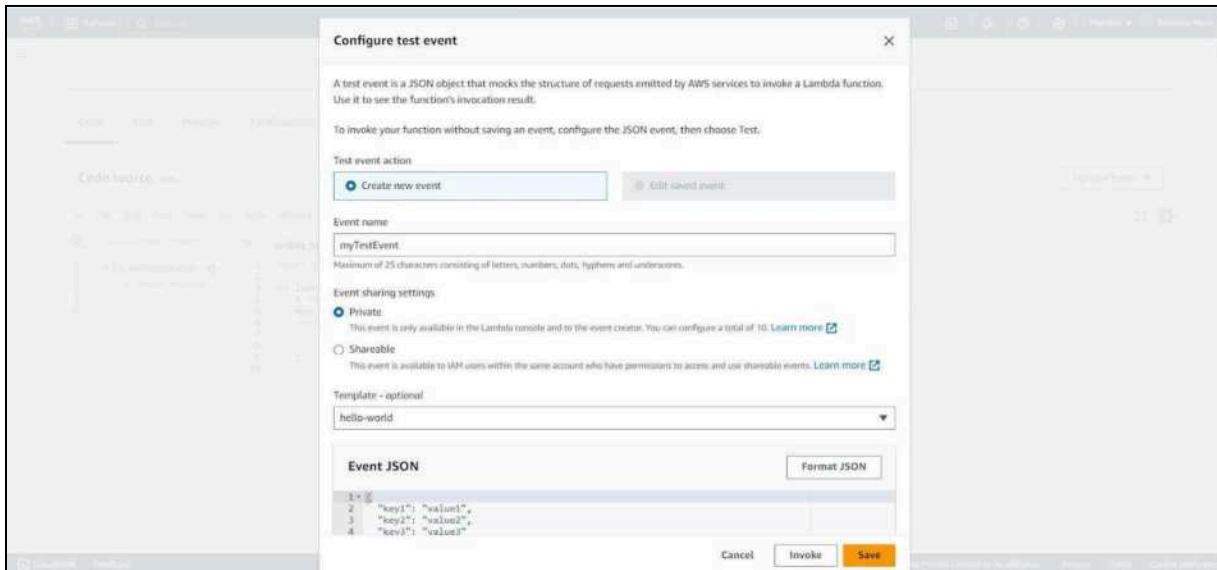
CloudShell Feedback

5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed. Press Ctrl + S to save the file and click Deploy to deploy the changes.

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message 'Changes not deployed'. The main area is titled 'Code source' with a 'Info' link. It contains a code editor window for 'lambda_function.py' which contains the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string="Hello how are you?"
6     return {
7         'statusCode': 200,
8         'body': json.dumps('Hello from Lambda!')
9     }
10
```

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



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

The test event myTestEvent was successfully saved.

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

Go to Anything (Ctrl+P) lambda_function Environment Var Execution result

Execution results Test Event Name myTestEvent

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

Function Logs

```
START RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc Version: $LATEST
END RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc
REPORT RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc Duration: 1.66 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 40 MB Init Duration: 1.66 ms
```

Request ID
7d26f404-f1da-4435-9faf-8dbb2a2733cc

Code properties Info

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The test event myTestEvent was successfully saved.

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

Go to Anything (Ctrl+P) lambda_function Environment Var Execution result

Execution results Test Event Name myTestEvent

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

Function Logs

```
START RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc Version: $LATEST
END RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc
REPORT RequestId: 7d26f404-f1da-4435-9faf-8dbb2a2733cc Duration: 1.66 ms Billed Duration: 2 ms Memory Size: 128 MB Max
```

Request ID
7d26f404-f1da-4435-9faf-8dbb2a2733cc

Code source Info Upload from ▾

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

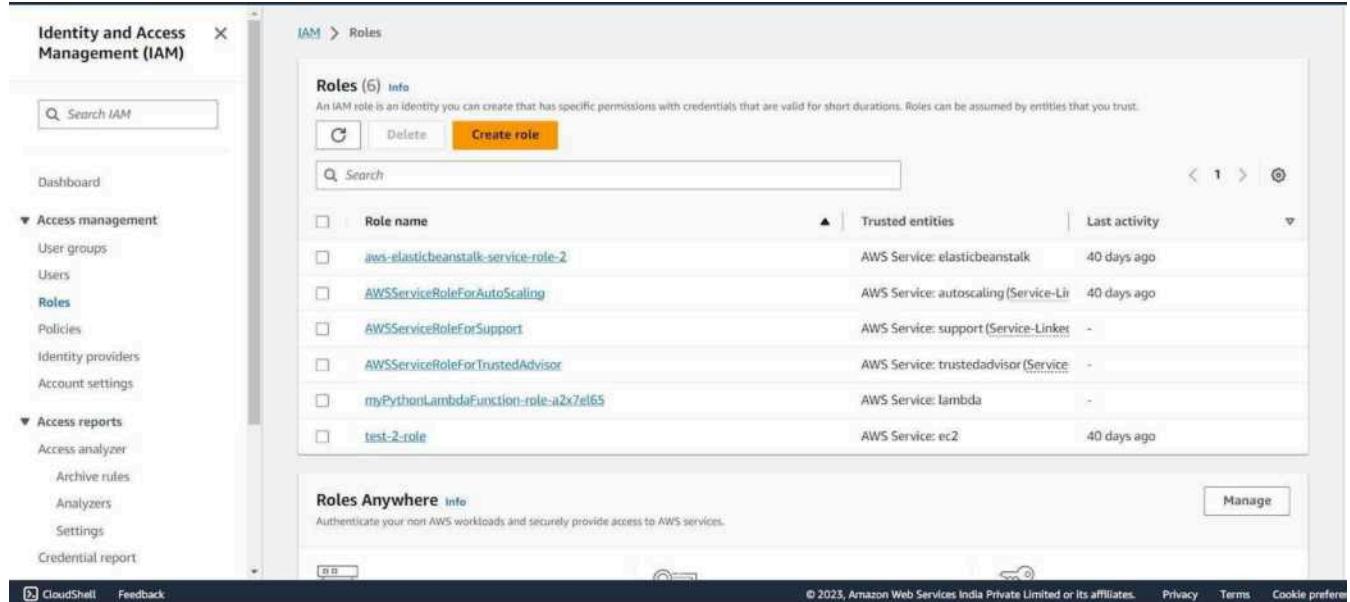
Adv. DevOps Exp. 12

Name-Manav Punjabi

Class-D15A

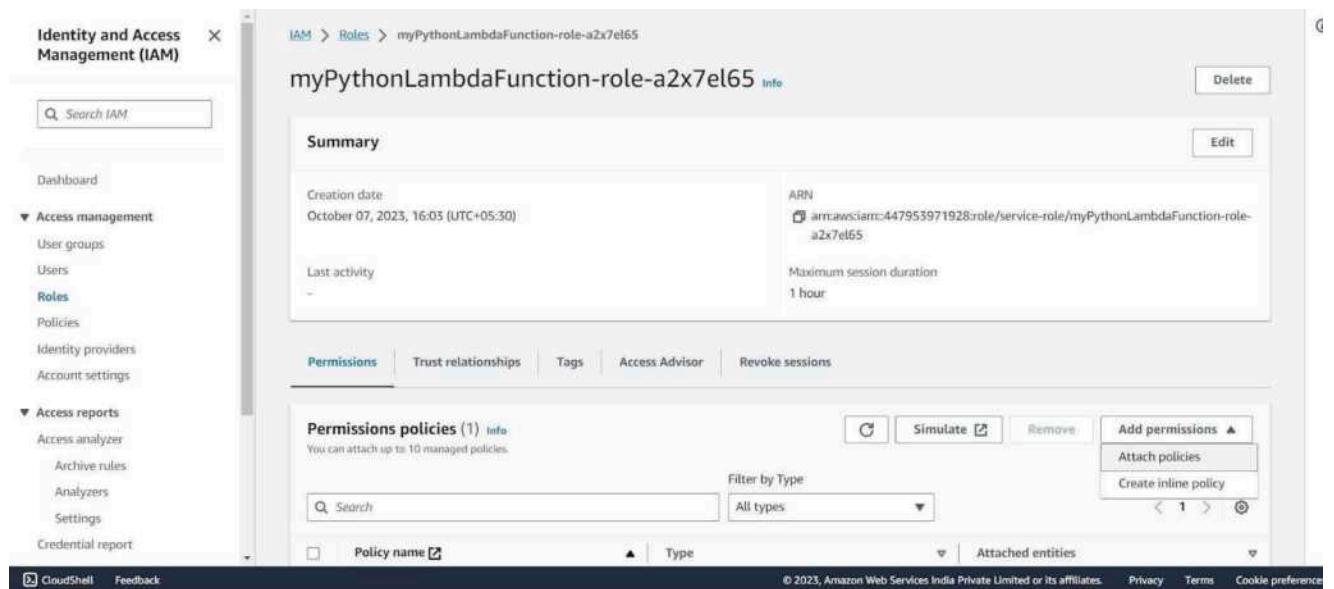
Roll No-45

Step 1: Open up the IAM Console and under Roles, choose the Role we previously created for the Python Lambda Function (You can find your role name configuration of your Lambda function).



The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like Dashboard, Access management (Roles selected), Policies, Identity providers, Account settings, Access reports, and Credential report. The main area displays a table titled 'Roles (6)'. The table has columns for Role name, Trusted entities, and Last activity. The roles listed are: aws-elasticbeanstalk-service-role-2, AWSServiceRoleForAutoScaling, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, myPythonLambdaFunction-role-a2x7el65, and test-2-role. Each row shows the ARN, the entity it trusts (e.g., AWS Service: elasticbeanstalk), and the last time it was used (40 days ago for most). Below the table, there's a section for 'Roles Anywhere' with a 'Manage' button.

Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.



The screenshot shows the details for the role 'myPythonLambdaFunction-role-a2x7el65'. The left sidebar is identical to the previous screenshot. The main page has a 'Summary' section with details like Creation date (October 07, 2023) and ARN (arn:aws:iam::447953971928:role/service-role/myPythonLambdaFunction-role-a2x7el65). Below this is a 'Permissions' tab, which is currently selected. It shows one policy attached: 'Permissions policies (1)'. A dropdown menu on the right allows adding permissions, attaching policies, or creating inline policies. At the bottom, there are buttons for 'Simulate', 'Remove', and 'Add permissions'.

S3-ReadOnly

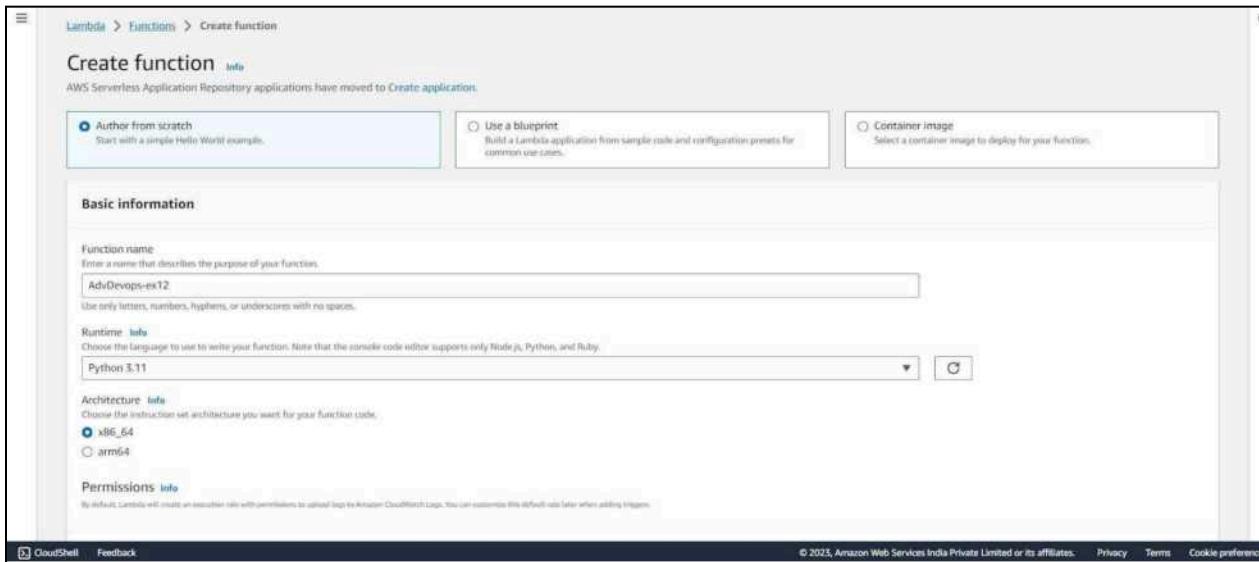
The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The role selected is 'myPythonLambdaFunction-role-a2x7el65'. The search bar at the top contains 'S3read'. A table lists one policy: 'AmazonS3ReadOnlyAccess' (AWS managed), which provides read-only access to all buckets. Buttons for 'Cancel' and 'Add permissions' are visible.

CloudWatchFull

The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The role selected is 'myPythonLambdaFunction-role-a2x7el65'. The search bar at the top contains 'cloudwatchfull'. A table lists two policies: 'CloudWatchFullAccess' and 'CloudWatchFullAccessV2' (both AWS managed), which provide full access to CloudWatch. Buttons for 'Cancel' and 'Add permissions' are visible.

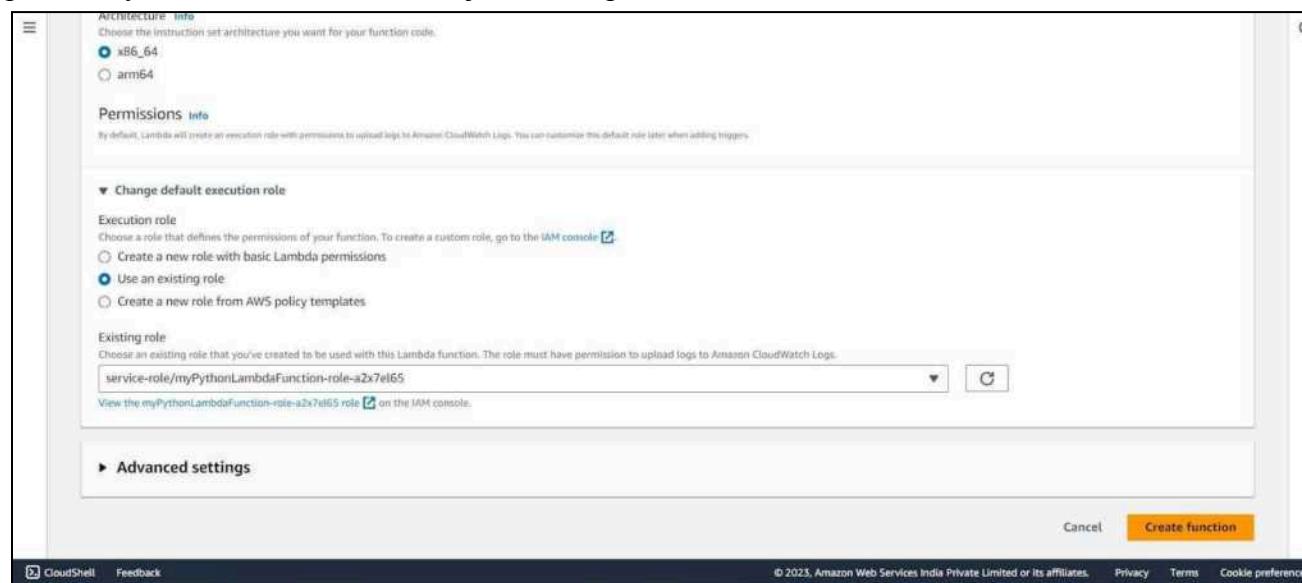
After successful attachment of policy you will see something like this you will be able to see the updated policies.

The screenshot shows the 'Permissions' tab in the AWS IAM console for the role 'myPythonLambdaFunction-role-a2x7el65'. A green banner at the top indicates 'Policy was successfully attached to role.' The 'Permissions' section shows three attached policies: 'AmazonS3ReadOnlyAccess', 'AWSLambdaBasicExecutionRole-c4946a...', and 'CloudWatchFullAccess'. Buttons for 'Simulate', 'Remove', and 'Add permissions' are available.

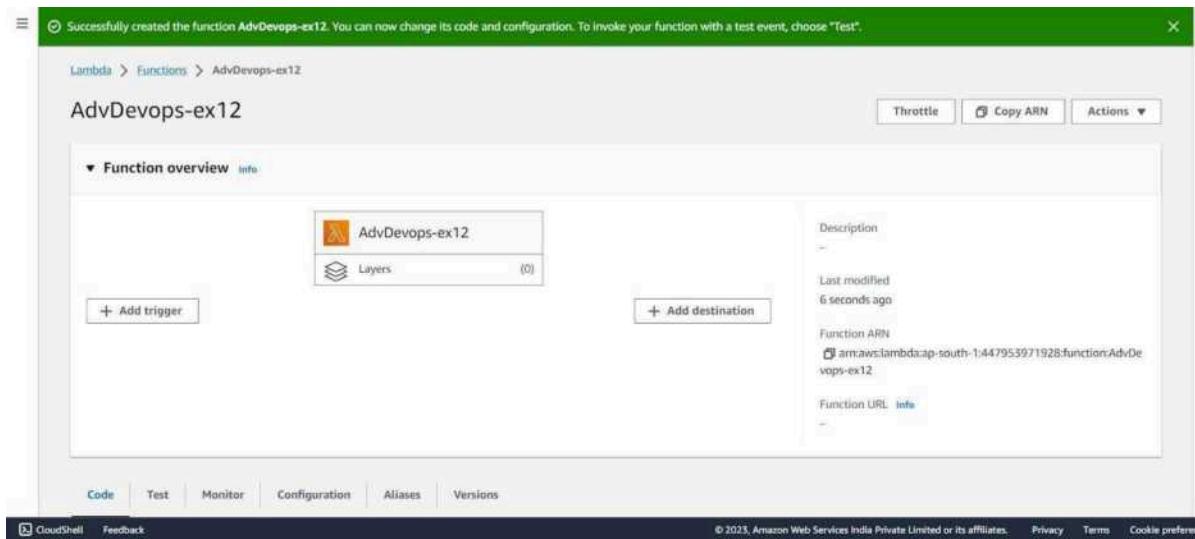


Step 3: Open up AWS Lambda and create a new Python function.

Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.



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.

```
lambda_function
Environment
CloudShell Feedback
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
6
7     s3_client = boto3.client('s3')
8     bucket_name = event['Records'][0]['s3']['bucket']['name']
9     key = event['Records'][0]['s3']['object']['key']
10    key_urllib_parse_unquote_plus = (key, encoding='utf-8')
11    message = 'A file has been added with key ' + key + ' to the bucket ' + bucket_name
12    print(message)
13    response = s3_client.get_object(Bucket=bucket_name, Key=key)
14    contents = response['Body'].read().decode()
15    contents = json.loads(contents)
16
17    print("These are the Contents of the File: \n", contents)
18
19
```

The screenshot shows the AWS Lambda code editor for the function "lambda_function". The code is written in Python and uses the AWS SDK (boto3) to interact with the S3 service. It retrieves a file from an S3 bucket, decodes its content, and then prints both a log message and the JSON contents to the console. The editor interface includes a sidebar for environment variables and a status bar at the bottom indicating "18.5 Python Spaces: 4". The footer includes links for "CloudShell", "Feedback", and copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences".

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 'aws' logo, 'Services' dropdown, a search bar, and a keyboard shortcut '[Alt+S]'. A green banner message says: 'Successfully created the function **AdvDevops-ex12**. You can now change its code and configuration. To invoke your function, choose Test or Deploy.' Below the banner, there are tabs: 'Code' (selected), 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. Under the 'Code' tab, the 'Code source' section is visible, showing the file structure 'AdvDevops-ex12' with 'lambda_function.py' selected. The code editor shows the following Python code:

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
```

Below the code editor, a modal window titled 'Configure test event' is open. 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.' It also says: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', there are two options: 'Create new event' (selected) and 'Edit saved event'. The 'Event name' field is set to 'test'. In the 'Event sharing settings' section, 'Private' 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. At the bottom of the modal, there's an 'Event JSON' input field containing the following JSON:

```
{ "Records": [ { "s3": { "bucket": { "name": "advdevops-ex12" }, "object": { "key": "testfile.txt", "size": 1024 } } } ] }
```

Buttons at the bottom right of the modal include 'Format JSON', 'Cancel', 'Invoke' (disabled), and 'Save'.

And Save it.

Step 7: Open up the S3 Console and create a new bucket.

Amazon S3

▶ Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

Buckets (3) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

[Create bucket](#)

Find buckets by name

| Name | AWS Region | Access | Creation date |
|--|----------------------------------|-----------------------|--------------------------------------|
| elasticbeanstalk-ap-south-1-447953971928 | Asia Pacific (Mumbai) ap-south-1 | Objects can be public | August 7, 2023, 14:24:02 (UTC+05:30) |
| www.hellorachana.com | Asia Pacific (Mumbai) ap-south-1 | Public | July 30, 2023, 15:05:34 (UTC+05:30) |
| www.htmlwebsite.com | Asia Pacific (Mumbai) ap-south-1 | Public | July 30, 2023, 15:49:06 (UTC+05:30) |

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.

Choose bucket

Object Ownership [Info](#)

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.

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](#)

[Edit](#) [Delete](#) [Create event notification](#)

| 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 arclinks](#)

[Edit](#)

Send notifications to Amazon EventBridge for all events in this bucket

Off

Transfer acceleration

Use an accelerated endpoint for faster data transfers. [Learn more](#)

[Edit](#)

Transfer acceleration

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Click on Create Event Notification.

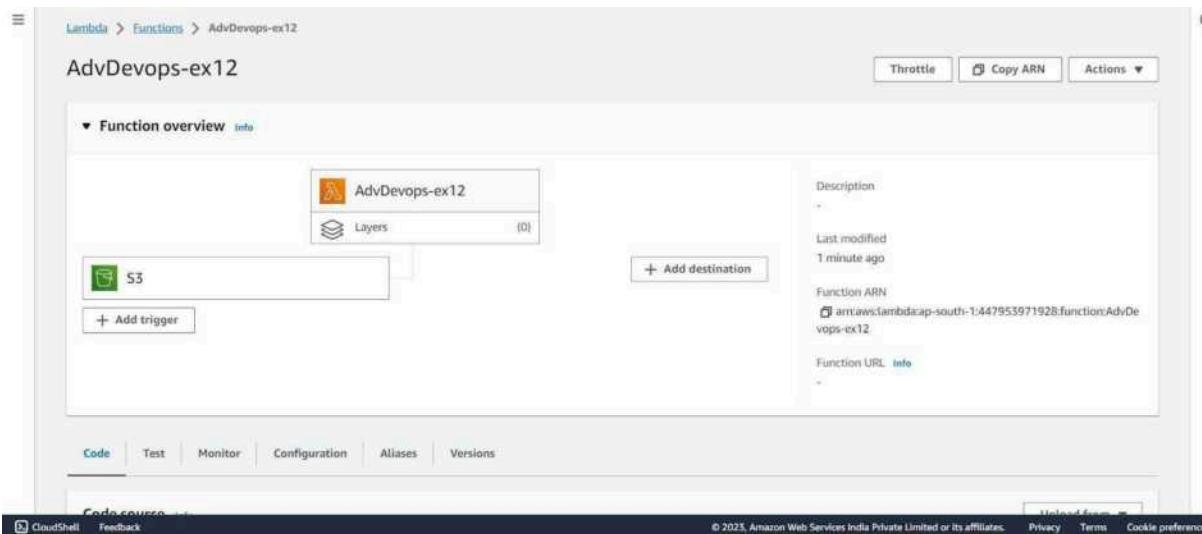
Step 10: Mention an event name and check Put under event types.

The screenshot shows the 'General configuration' section of the AWS S3 console. The 'Event name' field contains 'S3putrequest'. Under 'Event types', the 'Put' checkbox is checked, with the sub-type 's3:ObjectCreated:Put' selected. Other options like 'Post' are unchecked.

Choose Lambda function as destination and choose your lambda function and save the changes.

The screenshot shows the 'Destination' configuration page. It includes a note about granting permissions to publish messages to a destination. Under 'Destination', the 'Lambda function' radio button is selected. In the 'Specify Lambda function' section, 'Choose from your Lambda functions' is selected. A dropdown menu shows 'AdvDevops-ex12' as the chosen function. At the bottom right, there are 'Cancel' and 'Save changes' buttons.

Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



Step 12: Now, create a dummy JSON file locally.

```
{ } dummy.json X
{ } dummy.json > ...
1  {
2    "firstname" : "Shashwat",
3    "lastname" : "Tripathi",
4    "gender" : "Male",
5    "age": 19
6 }
```

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.

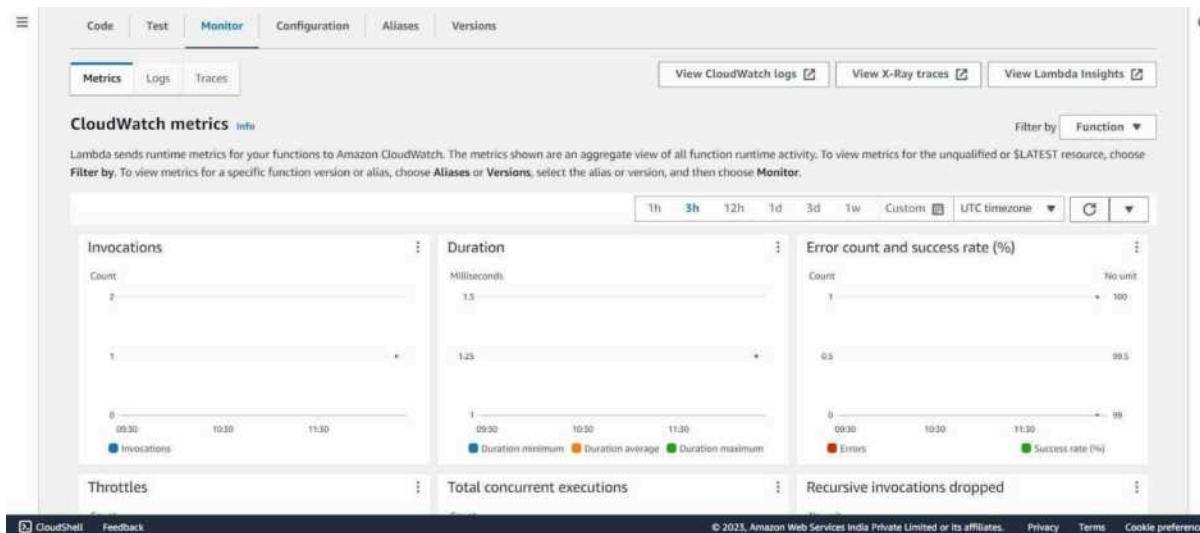
Step 14: Select the dummy data file from your computer and click Upload.

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar containing 'Search [Alt+S]', and a refresh icon. Below the navigation is a breadcrumb trail: 'Amazon S3 > Buckets > advopssexp12 > Upload'. The main area is titled 'Upload' with a 'Info' link. A note at the top says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more' with a link icon. Below this is a large dashed box with the placeholder text 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' A 'Files and folders (1 Total, 89.0 B)' section follows, containing a table with one item: 'dummy.json' (application/json, 89.0 B). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A 'Find by name' search bar and a page navigation bar ('< 1 >') are also present. The 'Destination' section shows 'Destination s3://advopssexp12'. At the bottom, there are 'CloudShell' and 'Feedback' links, and a copyright notice: '© 2023, Amazon Web Services India Private Limited or its affiliates'.

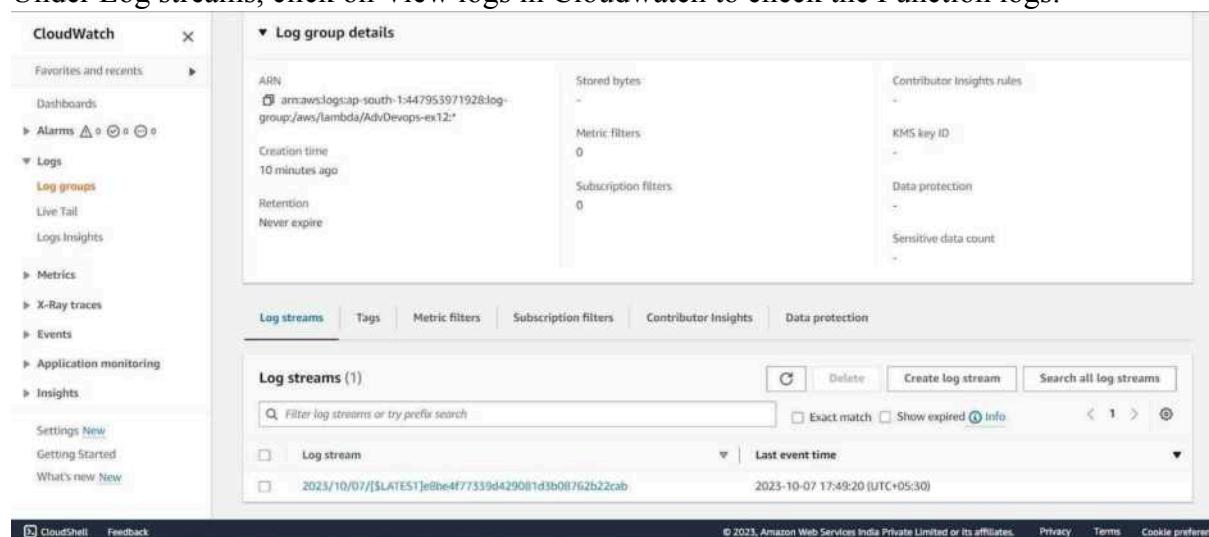
Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

The screenshot shows the AWS Lambda 'Event JSON' editor. It displays a large JSON object with numerous lines of code, representing a test event. The JSON includes fields like 'principalId', 'requestParameters', 'responseElements', and 's3'. The 's3' field contains details about a bucket named 'advopssexp12' with ARN 'arn:aws:s3:::advopssexp12'. The 'object' field within 's3' contains a key 'test%2Fkey', size '1024', ETag '0123456789abcdef0123456789abcdef', and a sequencer '0A1B2C3D4E5F678901'. The JSON is displayed with line numbers from 10 to 38 on the left. A 'Format JSON' button is located in the top right corner.

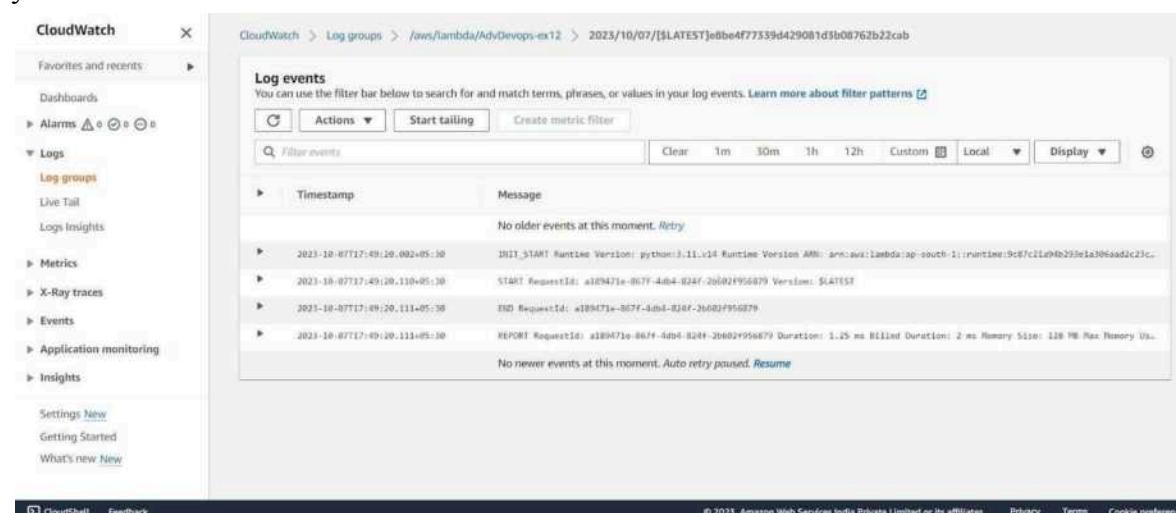
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



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.



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.

Assignment - 1

Ad - DevOps

Q. 1] USE S3 BUCKET AND HOST VIDEO STREAMING.

Step 1: Log in to AWS code

- Go to AWS Management console.
- Enter your login credentials.

Step 2: Create an S3 Bucket.

- In the console, search for S3 in the search bar and select S3 from the results.
- Click Create bucket.
- Give your bucket a unique name (like "My-video-streaming-bucket")
- Choose a region (closer to your audience.)
- Scroll down and uncheck Block all public access
- Confirm by checking the acknowledgement box.
- Click Create bucket.

Step 3: Upload your video file to S3

- Click on your newly created bucket.
- Click the Upload button.
- Add your video file from your computer.
- Click Upload to start the upload process.

Step 4: Set permissions for Public Access.

- Once the video file is uploaded, you need to make it publicly accessible.
- Select your video file in the S3 bucket.
- Click the Actions dropdown and choose Make Public.

- confirm the action by clicking make public again

Step 5: Get the video URL:

- After making the file Public, click on the video file.
- You will see a URL for the video under Object URL. This is the effect direct link to your object.
- Copy this URL.

Conclusion:- While solving this we get an error during bucket Policies. I solved this error using this steps:-

Step 1: Go to Permissions tab of your Bucket:

- Click on the permissions tab of your bucket (not the individual file.)

Step 2: Edit Bucket Policy:

- Click on Bucket Policy.
- Add the following Policy (Replace your-bucket name with your actual bucket name):

JSON

Copy code:

{

"version": "2012-10-17",

"statement": [

{

"sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "*";

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::your-bucket-name/*"

3

3

3

Step 3: Save the Policy.

Final Step : Testing Public Access.

- Once you've made your video public, copy the object URL (you'll find it in video file's)
- Paste that URL into your web browser to see if the video plays.

Q.2]

⇒

Discuss BMW and Hotstar case studies using AWS.

BMW and Hotstar case studies using AWS.

BMW:

1. Connected car features:

- Remote diagnostics for car health
- over-the-air software updates.
- Real-time traffic information.

2. Improving customer experience: They use AWS to enhance how customers interact with their cars.

for example, BMW has added voice-activated assistants and personalized suggestions using services like Amazon Lex and Amazon SageMaker.

3. Better manufacturing: BMW uses AWS to improve its manufacturing. By analyzing data from machines, they can find problems early and increase efficiency.

Hotstar:

1. Manage Huge traffic: During big events like the Indian Premier League (IPL), many people what at once.

2. Quality streaming: AWS services like Amazon Kinesis and Amazon Elastic Transcoder ensure that users get high-quality streaming.

3. Personalized Recommendations: Hotstar used AWS machine learning tools to suggest content to users, making

their viewing experience more enjoyable.

key AWS services used:

- Compute: Amazon EC2, Lambda
- Storage: Amazon S3, EFS
- Database: Amazon RDS, DynamoDB
- Networking: Amazon VPC, Direct Connect
- Analytics: Amazon Kinesis, Redshift
- Machine learning: SageMaker, Rekognition

AWS helps both companies innovate and deliver top-notch services to their customers.

Q.3) Why Kubernetes and advantages and disadvantages of Kubernetes. Explain how Adidas uses Kubernetes.

⇒ why Kubernetes?

Kubernetes is popular because it simplifies the management of containerized applications. It automates tasks such as deployment, scaling and monitoring, making it easier for organizations to manage their applications in a cloud environment.

Advantages of Kubernetes:

1. Portability: Applications can be moved easily between different environments without major changes.

2. Scalability: Kubernetes can automatically scale applications up or down based on traffic and demand.

3. Reliability: It features self-healing capabilities, meaning it can restart failed containers and balance workloads to ensure high availability.
4. Efficiency: It optimizes resource usage by running containers on a single host, improving overall efficiency.

Disadvantages of Kubernetes:

1. Complexity: It can be complicated to set up and manage, especially for those new to container technology.
2. Steep learning curve: Required time and knowledge to fully understand and utilize its features.
3. Resource intensive: It may require more computing resources than simpler solutions, which can increase costs.
4. Management overhead: Requires ongoing management and maintenance, which can add to operational workload.

How Adital uses Kubernetes:

- 1. Faster Development: Streamlined deployment for quicker product launches.
- 2. Operational Efficiency: Automated tasks, reducing managing time and increasing reliability.
- 3. Scalability: Easily handles traffic spikes during peak usage periods.

4. Encouraging innovation: flexible platform for experimenting with new technologies and ideas.

Specific use case at Adidaf:

1. microservices Architecture: independent services for better management and deployment.
2. continuous delivery: supports quick and efficient build, test, and deployment processes.
3. Hybrid cloud deployment: runs apps on both on-premises and cloud for flexibility and cost savings.

Benefits: modernized IT, improved agility, and enhanced innovation, keeping Adidaf competitive.

Q. 9 what are Nagios and explain how Nagios are used in F-O Services?

Ans what is Nagios?

Nagios is an open-source monitoring tool that helps organizations keep track of their IT infrastructure including servers, network, and applications.

Key features of Nagios:

1. Monitoring: tracks the performance and availability of servers, applications, and network devices.
2. Alerts: sends notifications via email or SMS when problems occur, so teams can respond quickly.
3. Reporting: offers detailed reports on system performance and uptime, helping identify trends and areas for improvement.

→ How Nagios used in E-Services:

1. Infrastructure Monitoring: tracks servers and databases, alerting the IT team if any issue arises.

2. Service Availability: monitors web services and APIs for accessibility, enabling quick response to outages.

3. Performance Metrics: collects data on system performance to optimize resources and enhance user experience.

4. Incident Management: integrates with tools to streamline issue resolution, minimizing downtime.

5. User Experience Monitoring: checks application performance from the user's perspective to ensure smooth operation.

Benefits: High availability, improved reliability, and better user experience, essential for maintaining service quality and customer satisfaction.

Adv DevOps Assignment - 2 Q3

Q.1] Create a REST API with Serverless framework
~~Ans~~ Creating REST API with serverless framework is an efficient way to deploy serverless applications that can scale automatically without managing servers.

- 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 application without worrying about underlying infrastructure, enabling focus on code and business logic.
- iii) REST API: - Representational state transfer is architecture style for designing network applications.

Steps for creating REST API for serverless framework:

- 1) Install Serverless framework:
~~You start by installing Serverless framework globally using node package manager. This allows you to manage Serverless applications directly from your terminal.~~
- 2) Creating a Node.js Serverless Project:
A directory is created by your project where you will initialize a Serverless service. This service will house all your lambda function, configuration, and cloud resources.

3) Project Structure:

The Project structure creates essential files like `handler.js` (which contains code for Lambda functions) and `serverless.yml`.

4) Create a REST API Resource:

In the `serverless.yml` file you define function that handles Post requests of HTTP.

5) Deploy the Service:

With the 'sls deploy' command the serverless framework packages your application, up to necessary resources.

6) Testing the API: Once deployed you can test REST API using tools like curl or Postman.

7) Storing data in Dynamo: To store a submitted candidate data you integrated AWS.

8) Adding more functionalities: Adding functionality like 'list all candidate', 'get candidate by ID'

9) AWS IAM permissions

You need to ensure that serverless framework is given right permissions to interact with AWS resources, the Dynamo.

- 10) Monitoring and maintenance
After deployment servers have framework
is given right permissions to interact with
AWS resources like dynamic.
- 11) Case study for Sonarable
Creating your own profile in Sonarable
for testing project quality. We can use
Sonarlin in your Java IntelliJ IDE and
analyze Java code. Analyze Python
Project with Sonarable.
→ Sonarable is an open source platform used
for continuous inspection of quality. It
detects bugs, code smells and security
vulnerability in project across various
programming languages.
- 12) Profile creation in Sonarable.
Quality profiles in Sonarable are essential
configuration that define rules applied
during code analysis. Each Project has a
quality profile for every supported
language with default being 'Java'
way. Profile comes built-in for all
languages. Custom profiles can be
created by copying or extending existing
ones. Copying creates an independent

profile, while extending inroot races for Parent profile and reflows future changes automatically you can activate or deactivate rules, Prioritize certain rules and configure Parameters to tailor Profile to Specific Profiles. Permissions to manage Quality Profile are restricted to users with administrative privileges. Monar allows for the comparison of two profiles to check for differences in activated rules and users can track changes via event tag.

2)

using SonarCloud to analyze Github. SonarCloud is cloud-based counterpart of SonarQube that integrates directly with Github - BitBucket, Azure and Github repositories. To get started with SonarCloud via Github sign up via sonarcloud product page and connect your Github organization or personal account. Once connected, SonarCloud mirrors your Github setup with each project corresponding to Github repository. After setting up the organization choose subscription plan (free for public repos). Next, import repositories into your SonarCloud organization where each Github

including security import issue.

3)

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 the development environment such as IntelliJ IDEA or Eclipse. To set it up, install the Sonarlint plugin, configure the connection with SonarQube or SonarCloud and select the Project Profile to analyze Java code. This approach creates immediate feedback in code quality, promoting clean and maintainable code from beginning.

4)

Analyzing Python Projects with SonarQube.

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

For setup you can use tox, PyTest and coverage and run test in your tox.ini include configurations for Pytest and coverage to generate coverage report in XML format. The build process can also be automated using GitHub Actions, which install dependencies, run tests, and invokes SonarQube scan.

5) Analyzing Node.js Projects with SonarQube.

for Node.js Project SonarQube can analyze Javascript and TypeScript code. Similar to the Python setup, you can configure SonarQube to analyze Node.js Projects by installing the appropriate Plugins and using Sonar Scanners to scan the Projects. SonarQube will check the code against industry standard rules and best practices.

Q.3] In large organizations, centralized operations teams often face many repetitive infrastructure requests, causing delays. Using a service infrastructure model with Terraform can help Product teams manage the infrastructure model with Terraform can help Product team manage independently.

Key benefits of Terraform:-

1. Modularity & Reusability : Terraform modules encapsulate standard configurations (database, compute resources) for reuse across projects.

2. Standardization : Best practices are built into modules, ensuring compliance with organizational policies.

3. Version control - dependencies and reduce disruptions.
4. Documentation: Promotes comprehensive documentation, collaboration, and understanding.

Implementation steps:-

1. Identify infrastructure components: Determine which components (load balancers) to modularize.
2. Develop reusable module: Define configurations with input variables and output for integration.
3. Governance and Best Practices: Set guidelines for module usage, versioning, and encourage team contributions.
4. Testing and validation: Implement testing frameworks to validate module functionality.

Terraform can also integrate with ticketing system like Jira or ServiceNow to automate infrastructure requests, improving workflows and reducing manual interactions. This self-service model increases agility.