

ADVANCE DEVOPS EXPERIMENT NO.1

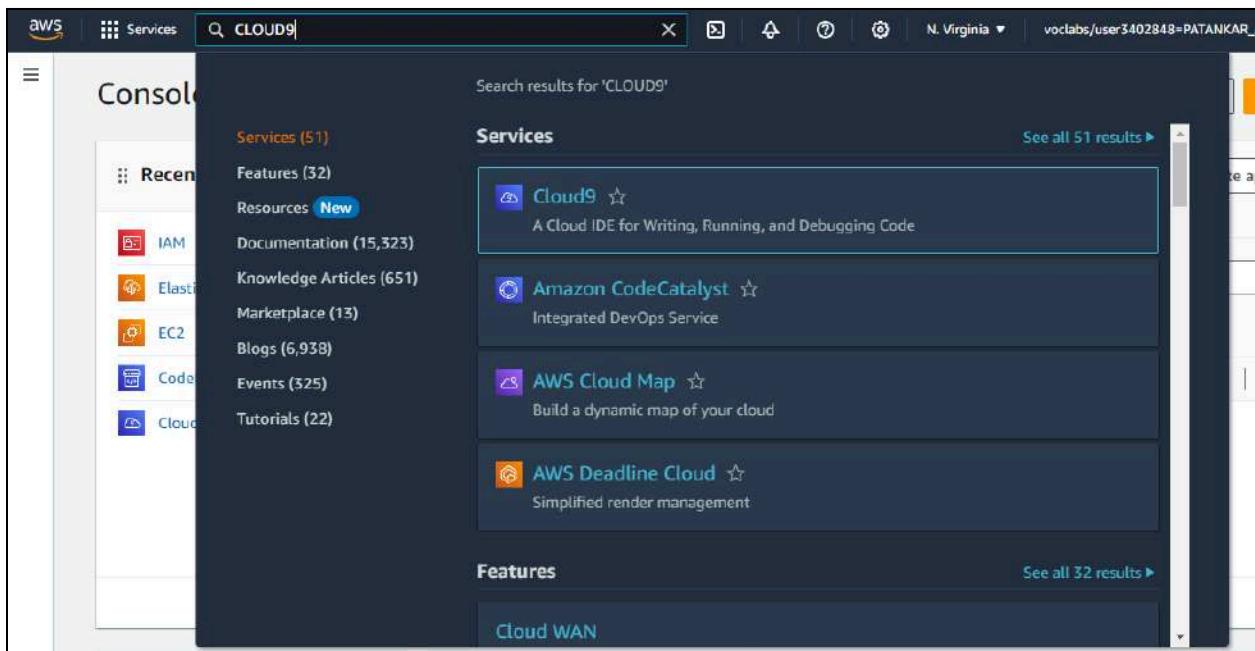
Name: Aryan Anil Patankar
Class:D15A
Roll No:34

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

Cloud9

Steps:

1. Open your AWS account and search for Cloud9 service inside Developer tools. Create a new Cloud9 environment by filling in the required details. Make sure you use an EC2 instance to create your environment.



The screenshot shows the AWS Cloud9 landing page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, search bar, and account information ('N. Virginia' and 'voclabs/user3402848=PATANKAR_ARYAN_ANIL'). Below the header, a sidebar titled 'Developer Tools' is visible. The main content area features the 'AWS Cloud9' logo and the tagline 'A cloud IDE for writing, running, and debugging code'. To the right, a call-to-action box says 'New AWS Cloud9 environment' with a prominent orange 'Create environment' button.

The screenshot shows the 'Create environment' form. On the left, a vertical sidebar lists 'Details', 'Environment type', 'Compute type', 'Code source', 'Code editor', 'File browser', 'Terminal', 'Logs', and 'Metrics'. The 'Details' tab is active. In the 'Name' field, 'Test123' is entered. A note below says 'Limit of 60 characters, alphanumeric, and unique per user.' The 'Description - optional' field is empty. A note below it says 'Limit 200 characters.' Under 'Environment type', the 'Info' link is shown, followed by a note: 'Determines what the Cloud9 IDE will run on.' Two options are available: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' option includes a note: 'Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.'

New EC2 instance

Instance type [Info](#)

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

t2.micro (1 GiB RAM + 1 vCPU)

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

t3.small (2 GiB RAM + 2 vCPU)

Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)

Recommended for production and most general-purpose development.

Additional instance types

Explore additional instances to fit your need.

Platform [Info](#)

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

Amazon Linux 2023



Timeout

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

30 minutes



Network settings [Info](#)

Connection

How your environment is accessed.

AWS Systems Manager (SSM)

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

Secure Shell (SSH)

Accesses environment directly via SSH, opens inbound ports.

► VPC settings [Info](#)

► Tags - optional [Info](#)

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

The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

Successfully created Test123. To get the most out of your environment, see [Best practices for using AWS Cloud9](#)

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

AWS Cloud9 > Environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Test123	Open	EC2 instance	Secure Shell (SSH)	Owner	<input checked="" type="checkbox"/> arn:aws:sts::554378108602:assumed-role/voclabs/user3402848=PATANKAR_ARYAN_ANIL

Search results for 'iam'

Services (11) See all 11 results ►

Features (24)

Resources **New**

Documentation (59,458)

Knowledge Articles (467)

Marketplace (856)

Blogs (1,843)

Events (12)

Tutorials (11)

IAM ☆ Manage access to AWS resources

IAM Identity Center ☆ Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager ☆ Share AWS resources with other accounts or AWS Organizations

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

IAM > Users

Users (0) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

[Create user](#)

User name	Path	Group	Last activity	MFA	Password age
No resources to display					

	<p>User name</p> <input type="text" value="aryan"/> <p>The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)</p> <p><input checked="" type="checkbox"/> Provide user access to the AWS Management Console - optional If you're providing console access to a person, it's a best practice IAM User Access to manage their access in IAM Identity Center.</p> <p>Console password</p> <p><input type="radio"/> Autogenerated password You can view the password after you create the user.</p> <p><input checked="" type="radio"/> Custom password Enter a custom password for the user. *****</p> <ul style="list-style-type: none"> • Must be at least 8 characters long • Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), @ # \$ % ^ & * () _ + - (hyphen) = [] { } [] ^ <p><input type="checkbox"/> Show password</p> <p><input checked="" type="checkbox"/> Users must create a new password at next sign-in - Recommended Users automatically get the IAMUserChangePassword policy to allow them to change their own password.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>User details</p> <table border="1" style="width: 100%;"> <tr> <td>User name aryan</td> <td>Console password type Custom password</td> <td>Require password reset Yes</td> </tr> </table>			User name aryan	Console password type Custom password	Require password reset Yes			
User name aryan	Console password type Custom password	Require password reset Yes						
<p>Permissions summary</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Name IAMUserChangePassword</th> <th>Type</th> <th>Used as</th> </tr> </thead> <tbody> <tr> <td>IAMUserChangePassword</td> <td>AWS managed</td> <td>Permissions policy</td> </tr> </tbody> </table>			Name IAMUserChangePassword	Type	Used as	IAMUserChangePassword	AWS managed	Permissions policy
Name IAMUserChangePassword	Type	Used as						
IAMUserChangePassword	AWS managed	Permissions policy						
<p>Tags - optional</p> <p>Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.</p> <p>No tags associated with the resource.</p> <p>Add new tag</p> <p>You can add up to 50 more tags.</p>								

more [\[2\]](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

i **Get started with groups**
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more \[2\]](#)

[Create group](#)

User name aryanp	Console password type None	Require password reset No
---------------------	-------------------------------	------------------------------

Permissions summary

Name [2]	Type	Used as
No resources		

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

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

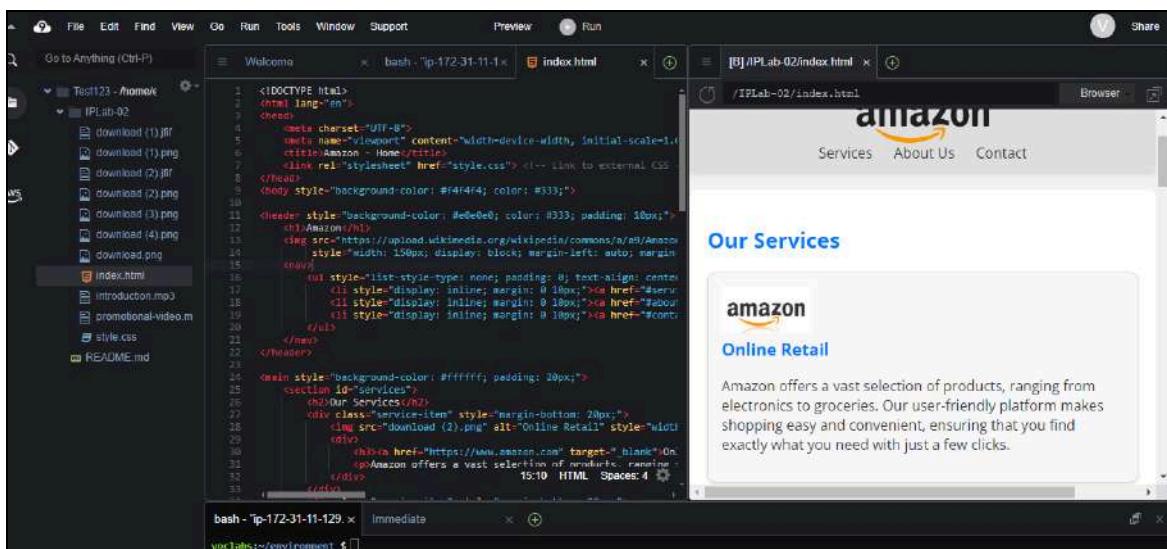
© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie pref](#)

Here the environment has been successfully created

The screenshot shows the AWS Cloud9 Environments page. At the top, there is a blue header bar with a message: "For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#)". Below the header, the page title is "AWS Cloud9 > Environments". The main content area is titled "Environments (1)". It contains a table with one row, labeled "My environments". The table columns are: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The single row shows: Name "Test123", Cloud9 IDE "Open", Environment type "EC2 instance", Connection "Secure Shell (SSH)", Permission "Owner", and Owner ARN "arn:aws:sts::554378108602:assumed-role/voclabs/user3402848=PATANKAR_ARYAN_ANIL". There are also buttons for "Delete", "View details", "Open in Cloud9", and "Create environment".

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Test123	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::554378108602:assumed-role/voclabs/user3402848=PATANKAR_ARYAN_ANIL

2. We have successfully set up and launched our Cloud9 environment. Over here, we can build and develop programs as per our desire. We are also allowed to collaborate with multiple other users and access shared resources.



Further, we are supposed to login from another browser using the credentials of the IAM user, to access the shared cloud9 environment with us. These steps could not be completed because Cloud9 services have been disrupted and there is no access to the IAM user from the remote login.

EC2 INSTANCE

Steps:

- 1.Create a new instance and follow the steps

The screenshot shows two pages from the AWS EC2 interface. The top part is the 'Launch an instance' wizard, where a new instance named 'Aryan' is being created with an Amazon Linux 2023.5.2 AMI, t2.micro instance type, and 8 GiB storage. The bottom part is the 'Amazon Machine Image (AMI)' search results page, showing various OS options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux, along with a search bar and a 'Browse more AMIs' link.

Launch an instance

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

Name: Aryan | Add additional tags

Application and OS Images (Amazon Machine Image)

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

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

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023.5.2... (ami-0ae8f15ae6fe5cda)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Cancel | Launch instance

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type | Free tier eligible

ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

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

Architecture: 64-bit (x86) | **AMI ID**: ami-04a81a99f5ec58529 | **Verified provider**

Quick Start

Recents

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

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

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

▼ **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](#)

▼ **Key pair (login)** [Info](#)

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

Key pair name - *required*

vockey ▾ [Create new key pair](#)

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

Instance type

t2.micro Free tier eligible

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

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0eb15f74eb572c84e

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

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

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

Create security group Select existing security group

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

Allow SSH traffic from Anywhere
Helps you connect to your instance

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-0d949d3b5f417c6b6](#))

▼ Launch log

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

Instances (1/1) Info			Connect	Instance state ▾	Actions ▾	Launch instances
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states ▾		
Instance ID = i-0d949d3b5f417c6b6	X		Clear filters		< 1 >	
<input checked="" type="checkbox"/> Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/> Aryan	i-0d949d3b5f417c6b6	Running		t2.micro	Initializing	View alarms +



Apache2 Default Page

Ubuntu

It works!

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

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

Configuration Overview

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

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

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

```
[1]+  Stopped                  nano index1.html
root@ip-172-31-36-118:/var/www/html# sudo nano index.html
root@ip-172-31-36-118:/var/www/html# sudo start apache2
sudo: start: command not found
root@ip-172-31-36-118:/var/www/html# sudo systemctl start apache2
root@ip-172-31-36-118:/var/www/html# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-08-18 09:03:15 UTC; 22min ago
     Docs: https://httpd.apache.org/docs/2.4/
      Main PID: 3261 (apache2)
        Tasks: 55 (limit: 1130)
       Memory: 5.3M (peak: 5.5M)
          CPU: 128ms
         CGroup: /system.slice/apache2.service
                   ├─3261 /usr/sbin/apache2 -k start
                   ├─3264 /usr/sbin/apache2 -k start
                   └─3265 /usr/sbin/apache2 -k start

Aug 18 09:03:15 ip-172-31-36-118 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 18 09:03:15 ip-172-31-36-118 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-36-118:/var/www/html#
```

```
- See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-36-118:~$ :/home/ubuntu# apt install apache2
:bash: :/home/ubuntu#: No such file or directory
ubuntu@ip-172-31-36-118:~$ cls
Command 'cls' not found, but there are 20 similar ones.
ubuntu@ip-172-31-36-118:~$ sudo su
root@ip-172-31-36-118:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
```

i-0d949d3b5f417c6b6 (Aryan)

PublicIPs: 52.207.231.96 PrivateIPs: 172.31.36.118

← → ⌂ Not secure aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com ☆ ⓘ

Amazon

amazon

Services About Us Contact

Our Services

amazon
Online Retail

Amazon offers a vast selection of products, ranging from electronics to groceries. Our user-friendly platform makes shopping easy and convenient, ensuring that you find exactly what you need with just a few clicks.

prime video

Amazon Prime Membership

ADVANCE DEVOPS EXPERIMENT NO.2

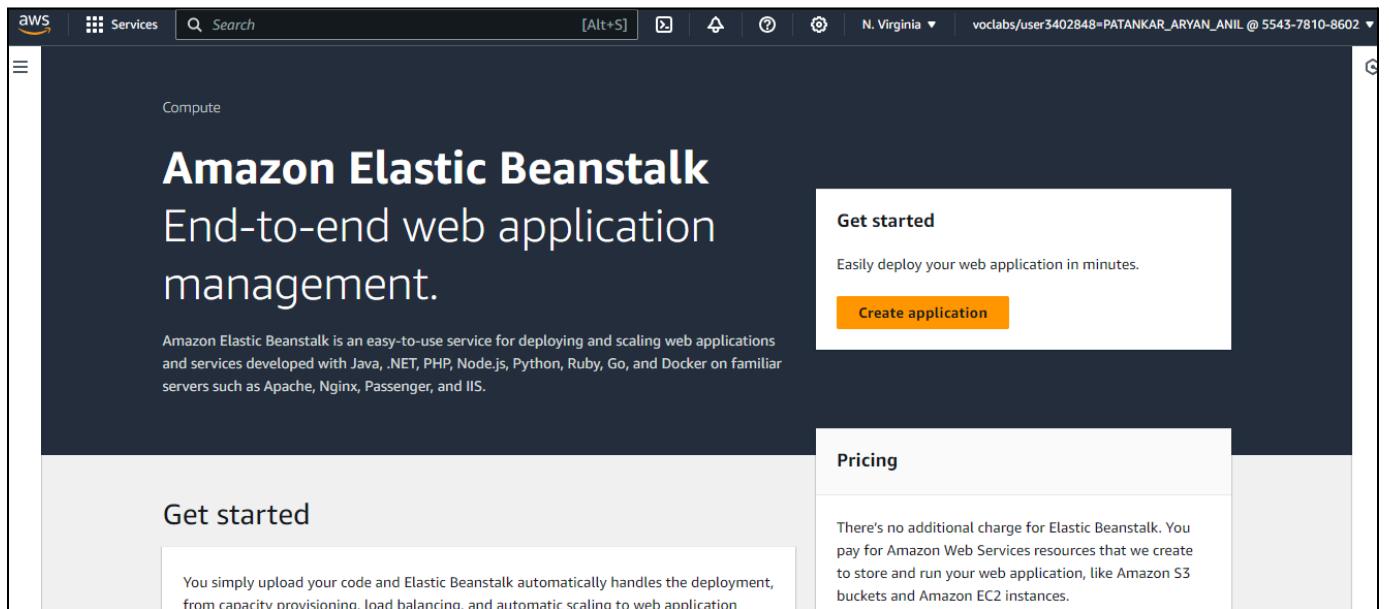
Name: Aryan Anil Patankar

Class:D15A

Roll No:34

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.

Step 1:Search for **Elastic Beanstalk** in the search bar next to services section and you would see the following page.



Step 2: Create a new application and proceed with the following settings

The screenshot shows the 'Configure environment' step of the AWS Elastic Beanstalk setup wizard. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Configure environment', which is currently active. Step 2 is 'Configure service access'. Step 3 is 'optional' and covers networking, database, and tags. Step 4 is 'optional' and covers instance traffic and scaling. Step 5 is 'optional' and covers updates, monitoring, and logging. Step 6 is 'Review'. The main content area is titled 'Configure environment' and contains two sections: 'Environment tier' and 'Application information'. In the 'Environment tier' section, 'Web server environment' is selected (indicated by a blue circle). Below it, 'Worker environment' is described as running a worker application that processes long-running workloads on demand or performs tasks on a schedule. In the 'Application information' section, the 'Application name' field is filled with 'Aryan27'. A note states that the maximum length is 100 characters. There is also a section for 'Application tags (optional)'.

The screenshot shows the 'Platform' configuration step of the AWS Elastic Beanstalk setup wizard. It includes sections for 'Platform type', 'Platform', 'Platform branch', and 'Platform version'. In the 'Platform type' section, 'Managed platform' is selected (blue circle). Below it, 'Custom platform' is described as platforms created and owned by the user, noting that it is unavailable if no platforms are present. The 'Platform' dropdown is set to 'PHP'. The 'Platform branch' dropdown is set to 'PHP 8.3 running on 64bit Amazon Linux 2023'. The 'Platform version' dropdown is set to '4.3.2 (Recommended)'. Each dropdown has a downward arrow icon indicating more options are available.

Application code Info

- Sample application
- Existing version

Application versions that you have uploaded.
- Upload your code

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

Presets Info

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

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

[Cancel](#)

[Next](#)

© 2024, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Term](#)

Step 3: Create a new service role as given below, if an existing service role with the same name does not exist. Proceed with the steps given below.

Service access

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

Service role

Create and use new service role
 Use an existing service role

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

[View permission details](#)

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

▼ 

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

▼ 

[View permission details](#)

▼ Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

(Container default) ▾

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

8 GB

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

100 IOPS

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

125 MiB/s

Step 2
[Configure service access](#)

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

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

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

Step 6
[Review](#)

▼ Monitoring Info

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

System
 Basic
 Enhanced

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

Log streaming
 Activated (standard CloudWatch charges apply.)

Retention
7

Step 4: Review each step along with the selected options and verify that the correct options have been chosen.

Review Info

Step 1: Configure environment

Edit

Environment information	
Environment tier	Application name
Web server environment	Aryan27
Environment name	Application code
Aryan27-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-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
arn:aws:iam::405894863107:role/service-role/aws-elasticbeanstalk-service-role	aws-elasticbeanstalk-ec2-role

Step 3: Set up networking, database, and tags

Edit

Networking, database, and tags Info

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

Network

VPC	Public IP address	Instance subnets
vpc-0bf7d7d872a737f13	false	subnet-035fe38d8d742329e,subnet-0a7c9c6dedec1325d

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

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

Monitoring

System enhanced	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
Log streaming Deactivated	Retention 7	Lifecycle false

Updates

Managed updates Activated	Deployment batch size 100	Deployment batch size type Percentage
---------------------------	---------------------------	---------------------------------------

Platform software

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		

Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		

Environment properties

Key	▲	Value	▼
No environment properties			
There are no environment properties defined			

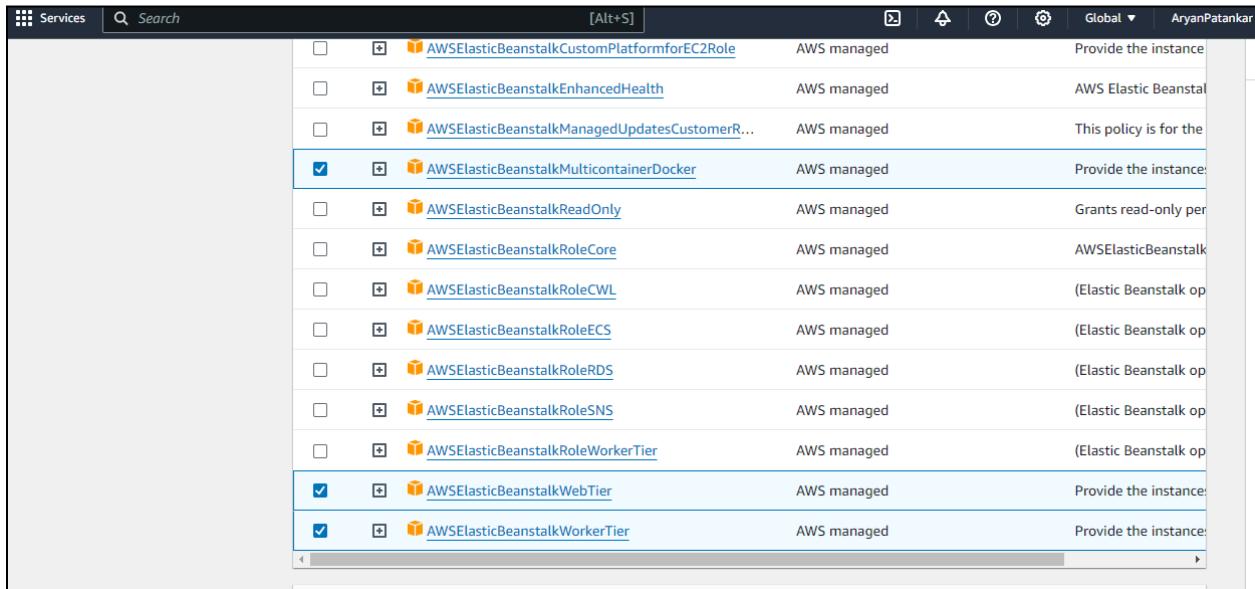
[Cancel](#)[Previous](#)[Submit](#)

Step 5: After clicking on the submit button, you would notice that the Elastic Beanstalk environment is being created and it may take some time for the environment to load completely.

The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar lists 'Applications' and 'Environments'. Under 'Environments', 'Aryan27' is selected, and its sub-options like 'Application versions' and 'Saved configurations' are visible. The main content area displays the 'Aryan27-env' environment details. It includes an 'Environment overview' section with fields for Health (Unknown), Environment ID (e-83gzhjzgmq), Domain (-), Application name (Aryan27), and a 'Platform' section showing PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2. Below this are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab shows 2 items. At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

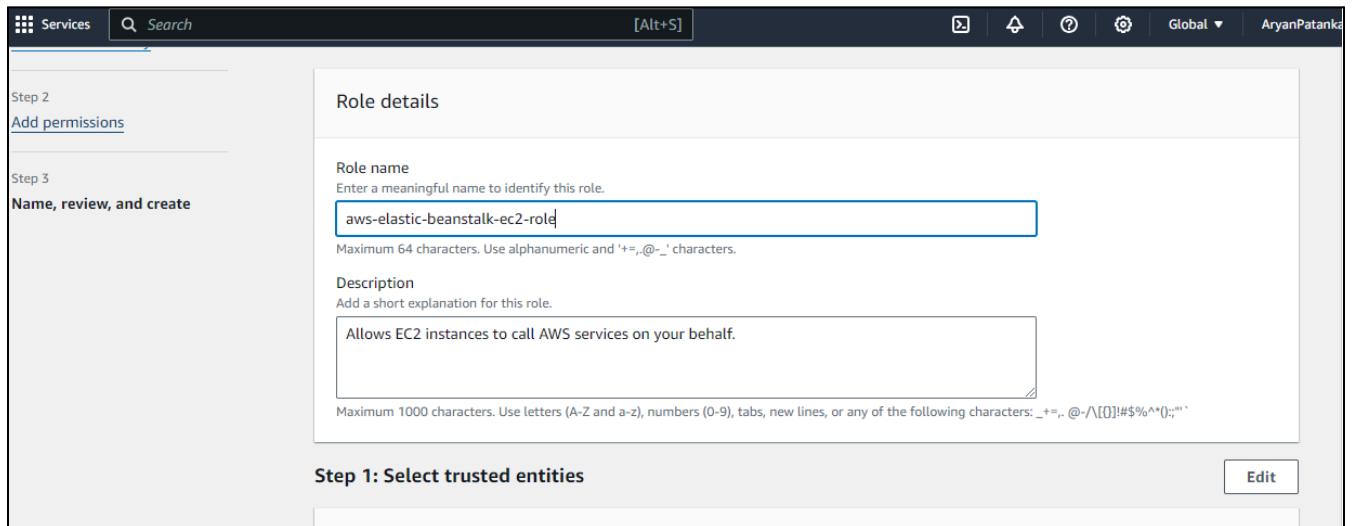
The screenshot shows the 'Trusted entity type' configuration page. It lists five options: 'AWS service' (selected, highlighted in blue), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. Below this is a 'Use case' section with a note about allowing actions from AWS services. At the bottom, a 'Service or use case' dropdown is set to 'EC2'.

Step 6: Meanwhile, if a role is already not defined, then you need to create a new role for the elastic beanstalk and ensure there is a blue checkmark on the following three permissions given below.

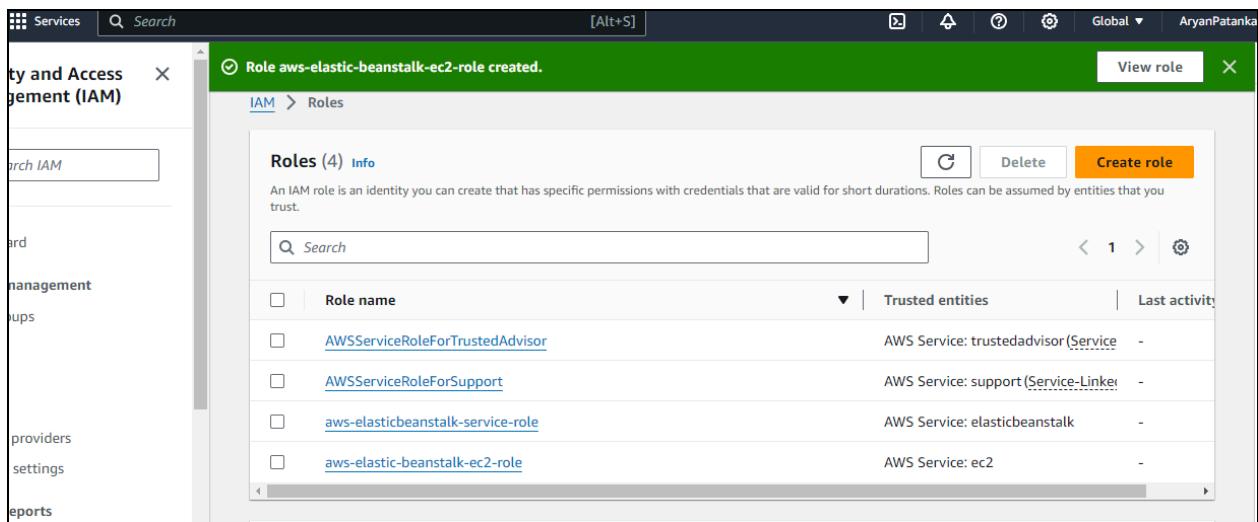


The screenshot shows the AWS IAM Roles list. A search bar at the top left contains 'Search'. On the right, there are global settings and a user name 'AryanPatankar'. The list displays several roles, each with a checkbox, a role icon, a name, and a description. Three specific roles have blue checkmarks in their checkboxes: 'AWSElasticBeanstalkMulticontainerDocker', 'AWSElasticBeanstalkWebTier', and 'AWSElasticBeanstalkWorkerTier'. Other roles listed include 'AWSElasticBeanstalkCustomPlatformforEC2Role', 'AWSElasticBeanstalkEnhancedHealth', 'AWSElasticBeanstalkManagedUpdatesCustomer...', 'AWSElasticBeanstalkReadOnly', 'AWSElasticBeanstalkRoleCore', 'AWSElasticBeanstalkRoleCWL', 'AWSElasticBeanstalkRoleECS', 'AWSElasticBeanstalkRoleRDS', 'AWSElasticBeanstalkRoleSNS', 'AWSElasticBeanstalkRoleWorkerTier', and 'AWSElasticBeanstalkWebTier' again.

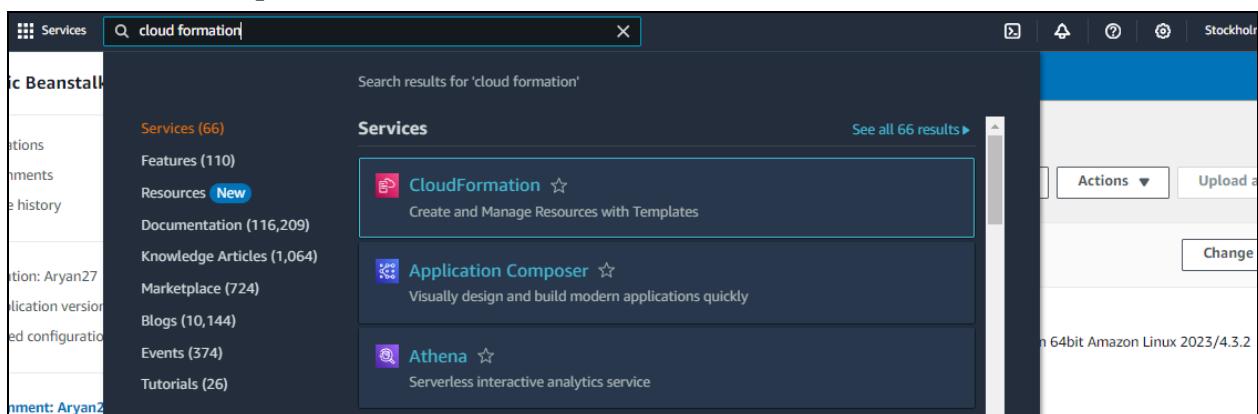
Step 7: Enter a role name and proceed. You would notice the role being successfully created after some time.



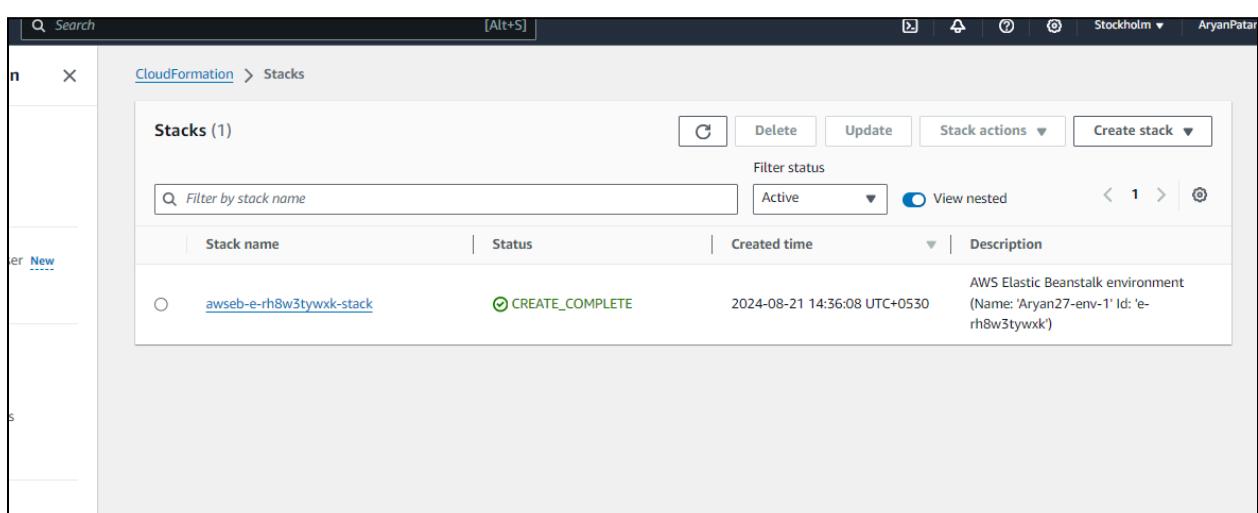
The screenshot shows the 'Role details' section of the AWS IAM Role creation wizard. It is Step 2: 'Add permissions'. The left sidebar shows 'Step 2: Add permissions' and 'Step 3: Name, review, and create'. The main area has a title 'Role details' and fields for 'Role name' and 'Description'. The 'Role name' field contains 'aws-elastic-beanstalk-ec2-role'. The 'Description' field contains 'Allows EC2 instances to call AWS services on your behalf.' Both fields have character limits and allowed characters specified below them. At the bottom, it says 'Step 1: Select trusted entities' and has an 'Edit' button.



Step 8(optional): Search for CloudFormation as it helps you to manage AWS resources in a text file or a template.



Here, the stacks option given below is a collection of AWS resources.



The screenshot shows the AWS CloudFormation console mode interface. On the left, a sidebar navigation includes: Stacks, Stack details, Drifts, StackSets, Exports, Application Composer (selected), and Registry. The main area displays the 'CloudFormation > Stacks > awseb-e-rh8w3tywxk-stack' page. It shows a list of stacks with one item: 'awseb-e-rh8w3tywxk-stack' (Status: CREATE_COMPLETE). The 'Template' tab is selected in the top navigation bar, showing the JSON template content:

```
{ "Outputs": {}, "AWSTemplateFormatVersion": "2010-09-09", "Parameters": { "InstanceTypeFamily": { "NoEcho": "true", "Type": "String", "Description": "WebServer EC2 instance type family" }, "LogPublicationControl": { "NoEcho": "true", "Type": "String", "Description": "If true customer service logs will be published to S3.", "AllowedValues": [ "true", "false" ] } }, "Resources": {} }
```

Below this, the Application Composer interface is shown in 'CloudFormation console mode'. It features a toolbar with 'Canvas' (selected), 'Template', 'Arrange', 'Validate', and 'Update template'. The canvas displays several standard components: AWSEBAutoScalingLaunchConfiguration, AWSEBEIP, AWSEBBeanstalkMetadata, and AWSEBInstanceLaunchWaitHandle. A dashed line connects the AWSEBAutoScalingLaunchConfiguration and AWSEBInstanceLaunchWaitHandle components, indicating a dependency or relationship between them.

Step 9: Now, we search for EC2 in the services section and we notice that an instance of the Elastic Beanstalk app has already been created and it is running.

Step 10: Click on the domain link given below, after which we are redirected to a Congratulations page implying that our sample PHP application has been successfully hosted.

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for 'Aryan27-env-1'. The top navigation bar includes 'Elastic Beanstalk' > 'Environments' > 'Aryan27-env-1'. The main content area is divided into two sections: 'Environment overview' and 'Platform'. The 'Environment overview' section displays 'Health' (Ok), 'Domain' (Aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com), 'Environment ID' (e-rh8w3tywxk), and 'Application name' (Aryan27). The 'Platform' section shows 'Platform' (PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2), 'Running version' (-), and 'Platform state' (Supported). Below these sections are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab is selected, showing 12 events. At the bottom, there are links for 'Events (12)', 'Info', and a refresh button.

The screenshot shows the 'Congratulations!' page for the 'Aryan27-env-1' environment. The page title is 'Congratulations!' and the message states: 'Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud'. It also mentions 'You are running PHP version 8.3.7' and 'This environment is launched with Elastic Beanstalk PHP Platform'. On the right side, there are two sections: 'What's Next?' and 'AWS SDK for PHP'. The 'What's Next?' section lists links to 'AWS Elastic Beanstalk overview', 'Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git', 'Using Amazon RDS with PHP', 'Customizing the Software on EC2 Instances', and 'Customizing Environment Resources'. The 'AWS SDK for PHP' section lists links to 'AWS SDK for PHP home', 'PHP developer center', and 'AWS SDK for PHP on GitHub'.

Step 11: Now, we will be deploying our website using CodePipeline, so follow all the steps given below and proceed.

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

No more than 100 characters

Pipeline type

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

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

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

Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

Service role

Step 12: In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

Step 2 of 5

Source

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

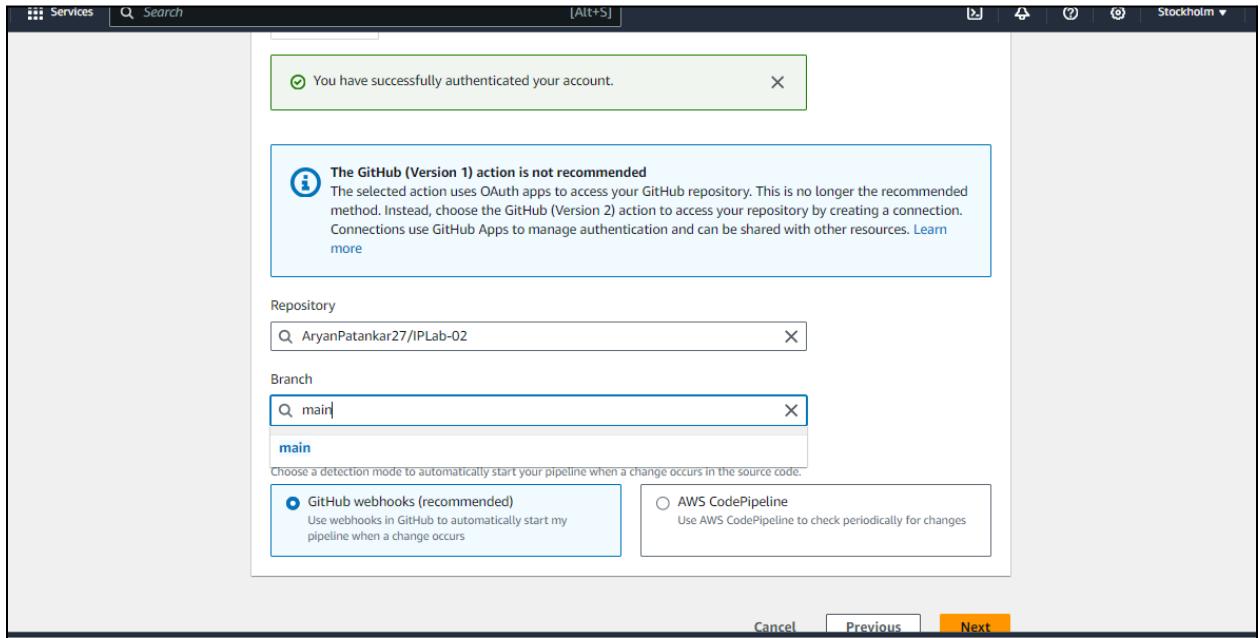
GitHub (Version 1)

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

Connected

ⓘ You have successfully configured the action with the provider.

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



Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 13: Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.

The screenshot shows the 'Deploy' configuration step in an AWS Lambda Pipeline. The form fields are as follows:

- Deploy provider:** AWS Elastic Beanstalk
- Region:** Europe (Stockholm)
- Input artifacts:** SourceArtifact (dropdown menu)
- Application name:** Aryan27
- Environment name:** Aryan27-env-1
- Configure automatic rollback on stage failure:** Unchecked checkbox

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
Aryan27

EnvironmentName
Aryan27-env

Configure automatic rollback on stage failure
Disabled

[Cancel](#) [Previous](#) [Create pipeline](#)

Step 14: Review all the selected steps once.

Review Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
Pipeline_Aryan

Pipeline type
V2

Execution mode
QUEUED

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

Service role name
AWSCodePipelineServiceRole-eu-north-1-Pipeline_Aryan

Step 2: Add source stage

Source action provider

Source action provider

GitHub (Version 1)

PollForSourceChanges

false

Repo

IPLab-02

Owner

AryanPatankar27

Branch

main

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

Aryan27

EnvironmentName

Aryan27-env-1

Configure automatic rollback on stage failure

Disabled

Cancel

Previous

Create pipeline

Step 15: In a few minutes, we will have our pipeline created. Once we have the success message on the

Deploy part, we can go ahead and check our URL provided in the EBS environment.

The screenshot shows the AWS CodePipeline console for a pipeline named "Pipeline_Aryan". The pipeline type is V2 and the execution mode is QUEUED. The pipeline consists of two stages: Source and Deploy. The Source stage is succeeded, with a GitHub (Version 1) action that succeeded 1 minute ago. The Deploy stage is also succeeded. Pipeline execution ID: 5622b57f-b111-4e0e-9bb3-05f6fe3e98d8. There are buttons for Notify, Edit, Stop execution, Clone pipeline, and Release change.

This is the sample website we just created.

The screenshot shows a web browser displaying the Amazon homepage at aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com. The page features the Amazon logo and navigation links for Services, About Us, and Contact. Below the header, there is a section titled "Our Services" featuring logos for Amazon Online Retail and Amazon Prime Video. A descriptive text box states: "Amazon offers a vast selection of products, ranging from electronics to groceries. Our user-friendly platform makes shopping easy and convenient, ensuring that you find exactly what you need with just a few clicks." At the bottom, there is a link to "Amazon Prime Membership".

If you can see this, that means that you successfully created an automated software using CodePipeline.

Using S3 Bucket

Step 16: Now, we will be deploying our website using the S3 bucket. So proceed with the options as given below.

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [Info](#)

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

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

Bucket name [Info](#)
aryan2711

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

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

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.

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.

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

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

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

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

Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access

Encryption type | [Info](#)

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

Bucket Key

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

- Disable
- Enable

► Advanced settings

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

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

i Successfully created bucket "aryan2711". To upload files and folders, or to configure additional bucket settings, choose [View details](#).

View details X

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

General purpose buckets	Directory buckets								
-----------------------------------------	-----------------------------------	--	--	--	--	--	--	--	--

General purpose buckets (3) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
aryan2711	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 21, 2024, 15:42:44 (UTC+05:30)
codepipeline-eu-north-1-365572256475	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 21, 2024, 15:29:16 (UTC+05:30)
elasticbeanstalk-eu-north-1-405894863107	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 21, 2024, 14:00:18 (UTC+05:30)

Step 17: Upload all the files that you want on your website that is to be hosted.

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**.

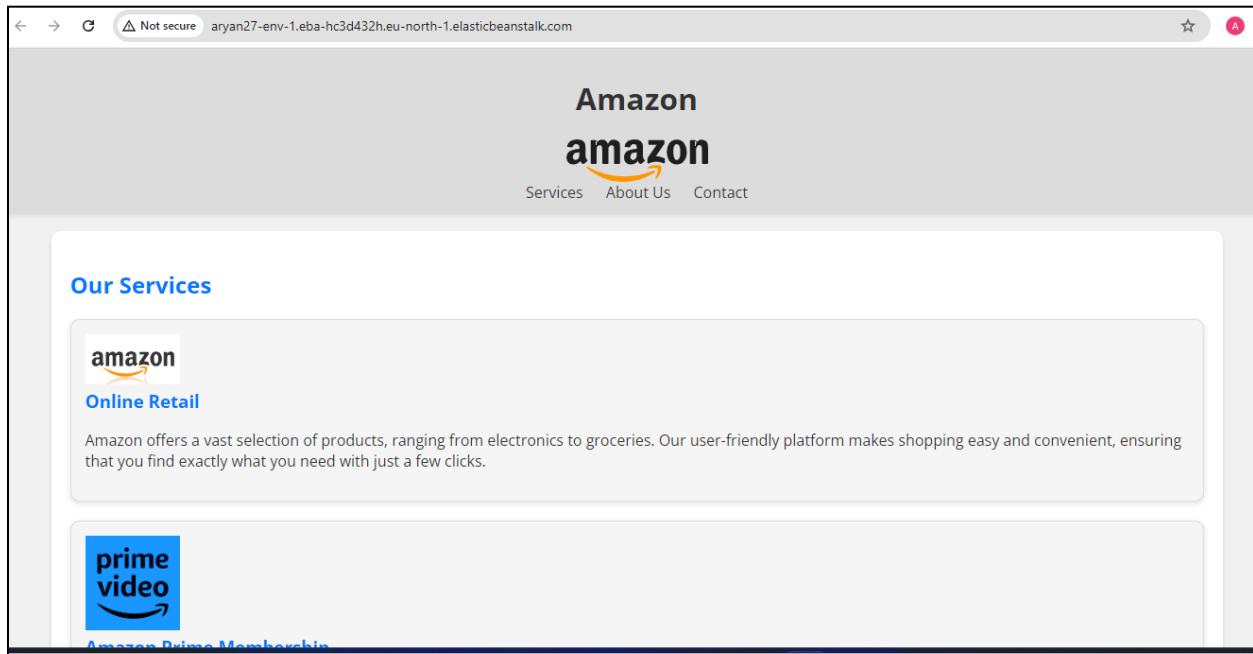
Files and folders (11 Total, 545.5 KB)		
All files and folders in this table will be uploaded.		
<input type="text"/> Find by name		
	Name	Folder
<input checked="" type="checkbox"/>	download (1).jfif	IPLab-02-main/
<input checked="" type="checkbox"/>	download (1).png	IPLab-02-main/
<input checked="" type="checkbox"/>	download (2).jfif	IPLab-02-main/
<input checked="" type="checkbox"/>	download (2).png	IPLab-02-main/
<input checked="" type="checkbox"/>	download (3).png	IPLab-02-main/
<input checked="" type="checkbox"/>	download (4).png	IPLab-02-main/
<input checked="" type="checkbox"/>	download.png	IPLab-02-main/
<input checked="" type="checkbox"/>	index.html	IPLab-02-main/
<input checked="" type="checkbox"/>	introduction.mp3	IPLab-02-main/
<input checked="" type="checkbox"/>	promotional-video.mp4	IPLab-02-main/

☰ Upload succeeded
View details below.

Files and folders Configuration

Files and folders (11 Total, 545.5 KB)					
<input type="text"/> Find by name					
Name	Folder	Type	Size	Status	Error
download (1... []	IPLab-02-main/	image/jpeg	8.4 KB	Successed	-
download (1... []	IPLab-02-main/	image/png	3.3 KB	Successed	-
download (2... []	IPLab-02-main/	image/jpeg	5.5 KB	Successed	-
download (2... []	IPLab-02-main/	image/png	6.0 KB	Successed	-
download (3... []	IPLab-02-main/	image/png	6.1 KB	Successed	-
download (4... []	IPLab-02-main/	image/png	2.4 KB	Successed	-
download.pn... []	IPLab-02-main/	image/png	4.7 KB	Successed	-
index.html []	IPLab-02-main/	text/html	6.1 KB	Successed	-
introduction.... []	IPLab-02-main/	audio/mpeg	158.0 KB	Successed	-
promotional... []	IPLab-02-main/	video/mp4	341.4 KB	Successed	-

Step 18: Here, if the upload of files is successful you would get the following page, meaning your website has been successfully hosted using the S3 bucket.



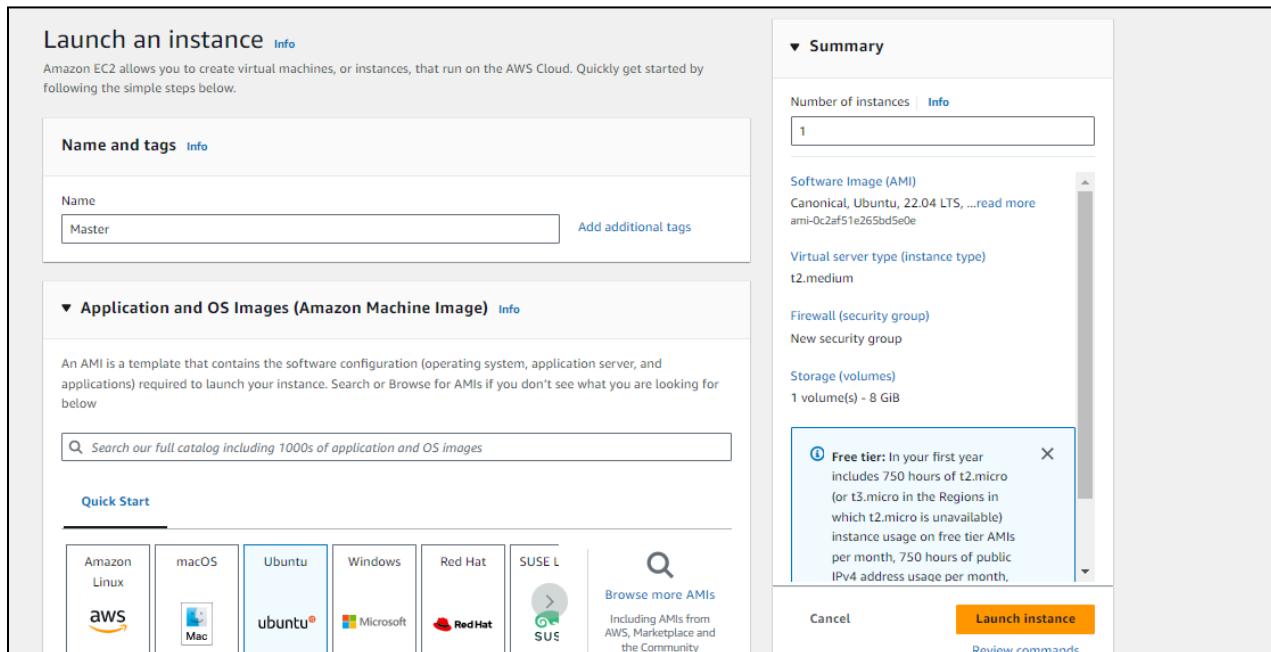
ADVANCE DEVOPS EXP 3

Name:Aryan Anil Patankar
Class:D15A
Roll No:34

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 (selected).
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 (selected).
Create new key pair

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 C Connect Instance state ▾ Actions ▾ Launch instances ▼										
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> All states ▾										
Instance state = running	X	Clear filters								
<input type="checkbox"/>	Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP	
<input type="checkbox"/>	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	
<input type="checkbox"/>	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.	
<input checked="" type="checkbox"/>	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.

The screenshot shows the AWS CloudWatch Metrics interface with the 'SSH client' tab selected. It displays the following information:

- Instance ID:** i-0e3930ceb2d892d01 (Worker-2)
- Instructions:**
 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%             IPv6 address for eth0: fe80::501:1ff:fed2:1000%eth0

Expanded Security Maintenance for Applications is not enabled.

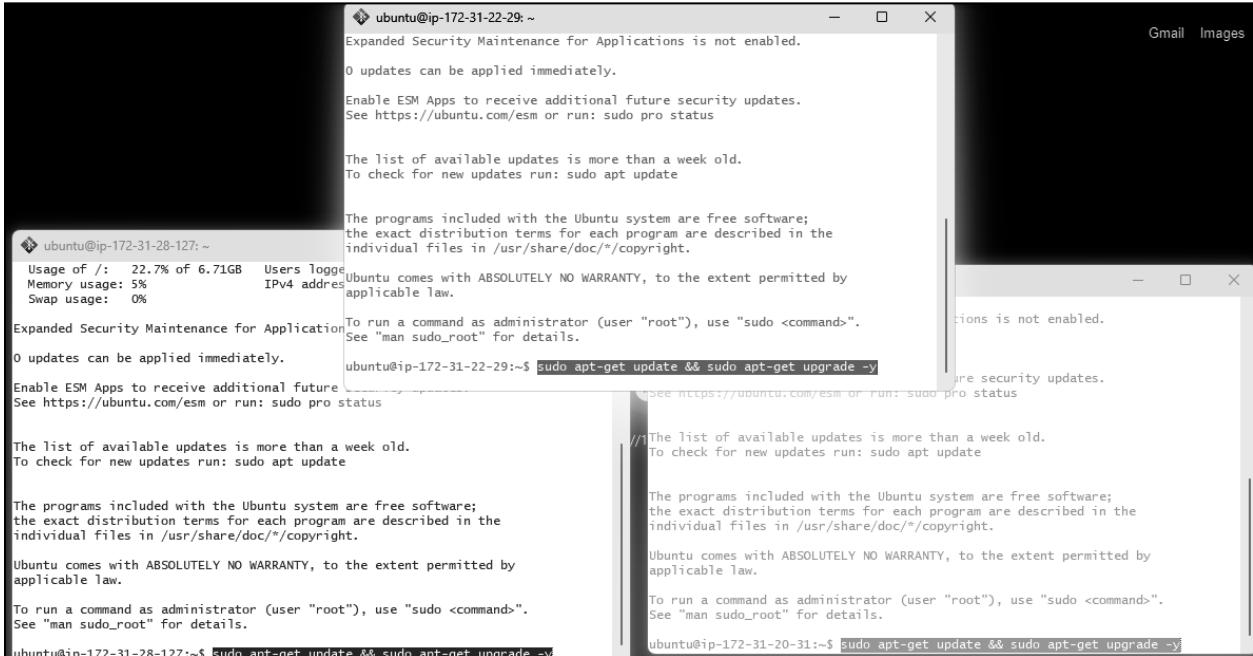
0 updates can be applied immediately.

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
```



The screenshot shows three terminal windows from a Linux desktop environment. Each window displays the command being run and its output.

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update && sudo apt-get upgrade -y
[Output of apt-get update and upgrade]
```

```
ubuntu@ip-172-31-28-127:~$ sudo apt-get update && sudo apt-get upgrade -y
[Output of apt-get update and upgrade]
```

```
ubuntu@ip-172-31-20-31:~$ sudo apt-get update && sudo apt-get upgrade -y
[Output of apt-get update and upgrade]
```



```
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-opt": {
    "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-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
sudo systemctl restart docker
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "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.hq28adbm1mu17ky --discovery-token-ca-cert-hash sha256:2bc469a8d14fbebe0f879328d2b416fad  
32b29a8505d3f448b98703ffff3b014d9
```

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:Aryan Anil Patankar

Class:D15A

Roll No:34

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

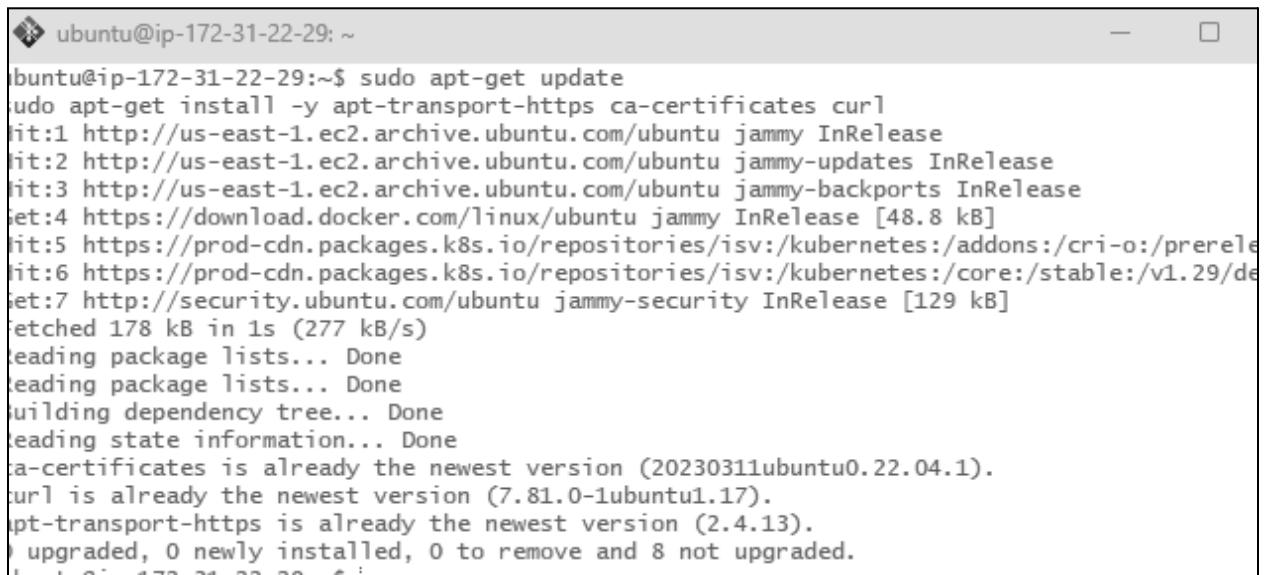
1.1 Add Kubernetes APT repository

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



ADVANCE DEVOPS EXP 5

Name:Aryan Anil Patankar

Class:D15A

Roll No:34

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

Installation for Windows:

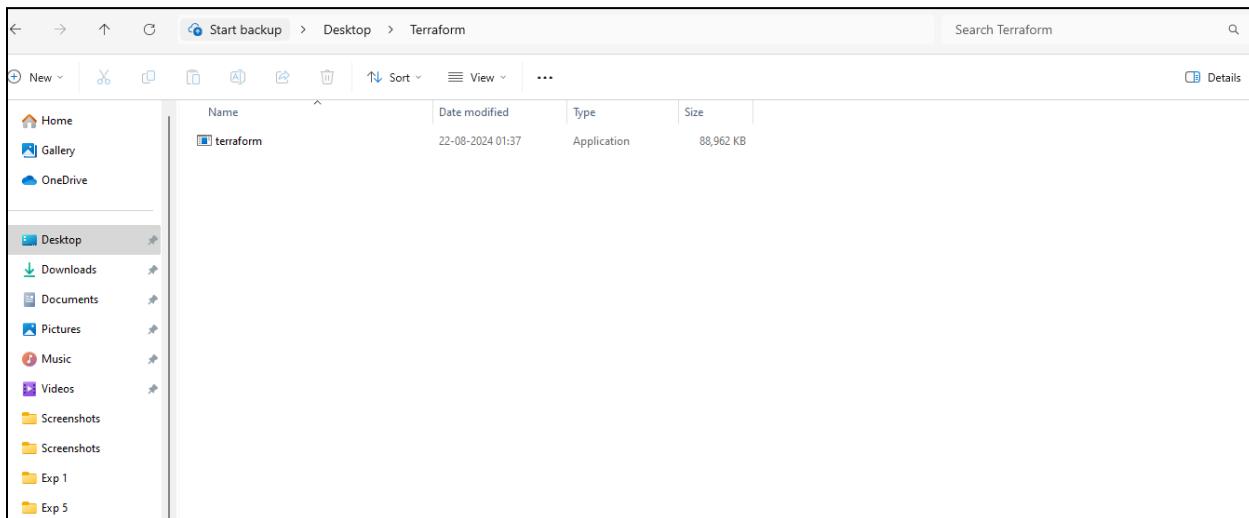
Step 1:Go to the website [terraform.io](https://www.terraform.io) and install Terraform from there..Select the AMD64 option for Windows and download Terraform.

The screenshot shows the Terraform website's 'Install Terraform' page for macOS. On the left, a sidebar lists operating systems: macOS, Windows, Linux, FreeBSD, OpenBSD, and Solaris. The 'macOS' section is selected. In the center, there are two main installation methods: 'Package manager' (using Homebrew) and 'Binary download'. The 'Binary download' section shows two options: 'AMD64' and 'ARM64'. To the right, a sidebar titled 'About Terraform' provides a brief overview and links to 'Featured docs' like Introduction to Terraform, Configuration Language, Terraform CLI, HCP Terraform, and Provider Use. At the bottom right is a 'HCP Terraform' button.

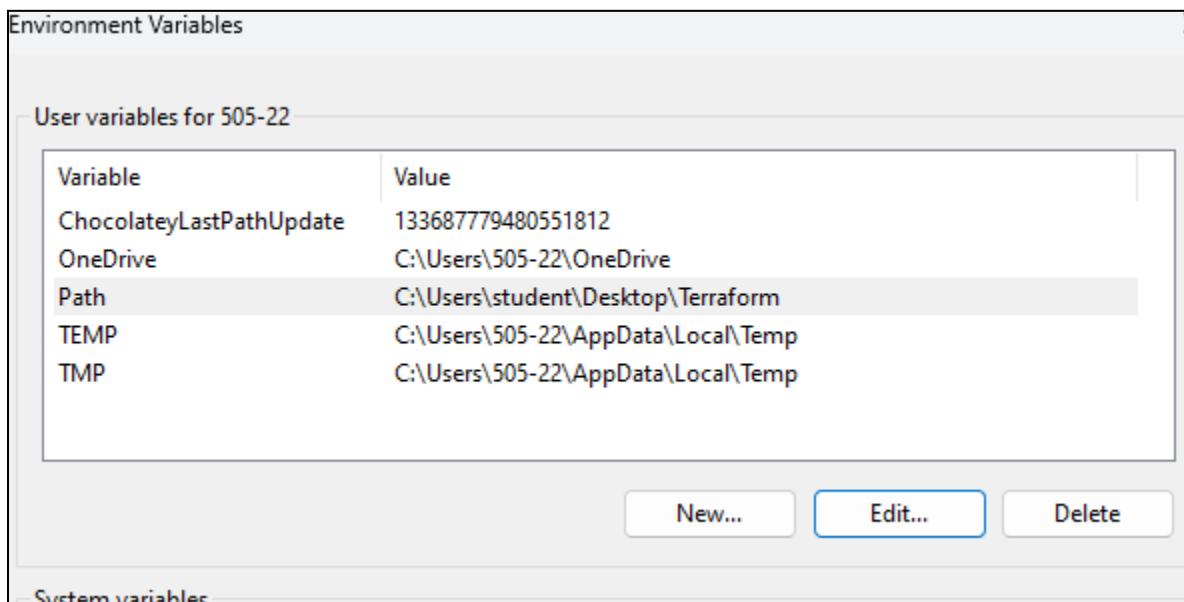
Step 2:Go to the zip file where Terraform is installed.

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
LICENSE	Text Document	2 KB	No	5 KB	63%	20-08-2024 17:35
terraform	Application	26,719 KB	No	88,962 KB	70%	20-08-2024 17:35

Step 3: Since the installed file is a zip file, create a new folder on desktop and copy the installed terraform application there.



Step 4: Now go to search bar, select edit environment variables option, then go to the path option. Now add the file path of the directory wherein we have installed the terraform application.



Step 5: Now go to the folder where we have installed terraform and open Powershell inside it. After this type ‘terraform’ to make sure that terraform has been installed on the system.

The command ‘terraform –version’ simply checks the current version of terraform that has been installed.

```
PS C:\Users\student\Desktop\Terraform> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

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

PS C:\Users\student\Desktop\Terraform> terraform --version
Terraform v1.9.4
on windows_amd64

Your version of Terraform is out of date! The latest version
is 1.9.5. You can update by downloading from https://www.terraform.io/downloads.html
PS C:\Users\student\Desktop\Terraform>
```

ADVANCE DEVOPS EXP 6

Name:Aryan Anil Patankar

Class:D15A

Roll No:34

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

Prerequisite:

- 1) Download and Install Docker Desktop from <https://www.docker.com/>

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>
```

Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

Step 2: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using Atom editor and write the following contents into it to create a Ubuntu Linux container.

Script:

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

```
}
```

```
"\uf0c0 docker.tf  X
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 }
25 "
```

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
```

```
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.

```

Docker images, Before Executing Apply step:

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

Docker images, 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:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  - image_id = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - latest   = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu: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:Aryan Anil Patankar
Class:D15A
Roll No:34

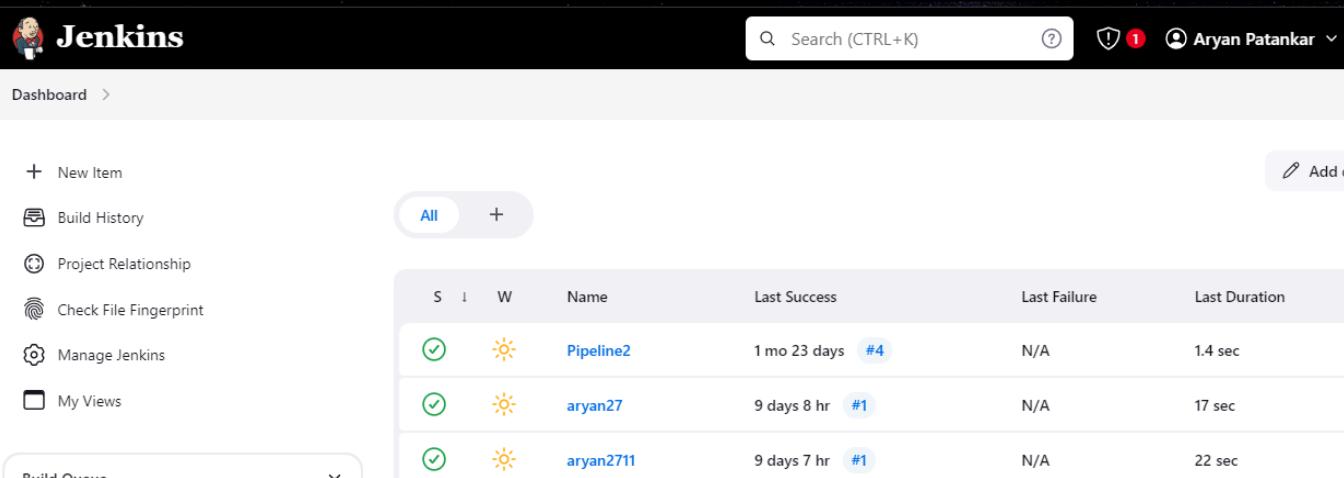
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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:

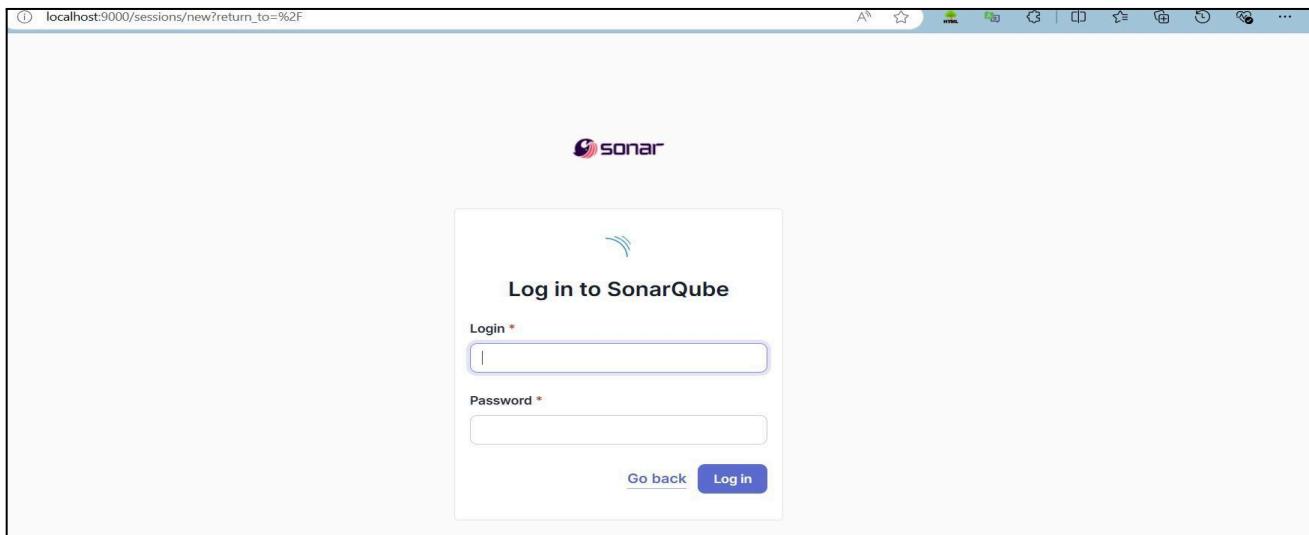
- Header:** Jenkins logo, Search bar (CTRL+K), Notifications (1), User Aryan Patankar.
- Left Sidebar:** Dashboard, + New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Build Queue.
- Center Content:** Pipeline status summary: All (3 Pipelines). Pipelines listed:
 - Pipeline2: Last Success 1 mo 23 days, Last Failure N/A, Last Duration 1.4 sec.
 - aryan27: Last Success 9 days 8 hr, Last Failure #1, Last Duration N/A.
 - aryan2711: Last Success 9 days 7 hr, Last Failure #1, Last Duration 22 sec.

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
```

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



- Login to SonarQube using username admin and password admin.

How do you want to create your project?

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

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

- Import from Azure DevOps
- Import from Bitbucket Cloud
- Import from Bitbucket Server
- Import from GitHub
- Import from GitLab

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

- Create a manual project in SonarQube with the name sonarqube

Create a local project

Project display name *
adv_devops_7_sonarqube

Project key *
adv_devops_7_sonarqube

Main branch name *
main

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

Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

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

Define a specific setting for this project

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

Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will be closed.
Recommended for projects following continuous delivery.

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 Plugins page. In the top right corner, there is a search bar with the text "sonarq". Below the search bar, there is a button labeled "Install". On the left side, there is a sidebar with options: "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings". The main content area displays the "SonarQube Scanner 2.17.2" plugin. It includes tabs for "External Site/Tool Integrations" and "Build Reports". A description below the plugin states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." To the right of the plugin details, it says "Released 6 mo 29 days ago".

The screenshot shows the Jenkins Plugins page with the "Download progress" tab selected. On the left, there is a sidebar with options: "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings". The main content area shows the "Download progress" section. It includes a "Preparation" step with three items: "Checking internet connectivity", "Checking update center connectivity", and "Success". Below that, it shows the "SonarQube Scanner" step with a green checkmark and the word "Success". Underneath, it shows "Loading plugin extensions" with another green checkmark and the word "Success". At the bottom, there are two buttons: one pointing to "Go back to the top page" with the note "(you can start using the installed plugins right away)" and another pointing to "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	X
adv_devops_7_sonarqube	X

Server URL
Default is http://localhost:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

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

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

The screenshot shows the 'SonarQube Scanner installations' configuration page. It allows adding a new scanner with a name (e.g., 'sonarqube_exp7') and the option to 'Install automatically'. A sub-section for 'Install from Maven Central' shows the selected version 'SonarQube Scanner 6.1.0.4477'. There is also an 'Add Installer' button.

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.

OK **branch Pipeline**
Creates a pipeline item of Jenkins projects according to detected branches in one SCM repository

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.

The screenshot shows the 'Source Code Management' configuration page. The 'Git' option is selected. A repository URL is entered as https://github.com/shazforiot/MSBuild_firstproject.git. The credentials dropdown is set to '- none -'. There is an 'Advanced' button and a 'Add Repository' button at the bottom.

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

The screenshot shows the 'Configure' screen with the 'Build Environment' tab selected. A dropdown menu is open, listing various build steps: Execute SonarQube Scanner, Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, Set build status to "pending" on GitHub commit, SonarScanner for MSBuild - Begin Analysis, and SonarScanner for MSBuild - End Analysis. An 'Add build step' button is at the bottom of the list.

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

Status ✓ **adv_devops_exp7**

</> Changes Add description

Workspace Disable Project

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

Configuration Security Projects System Marketplace

	Administer System	Administer	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:

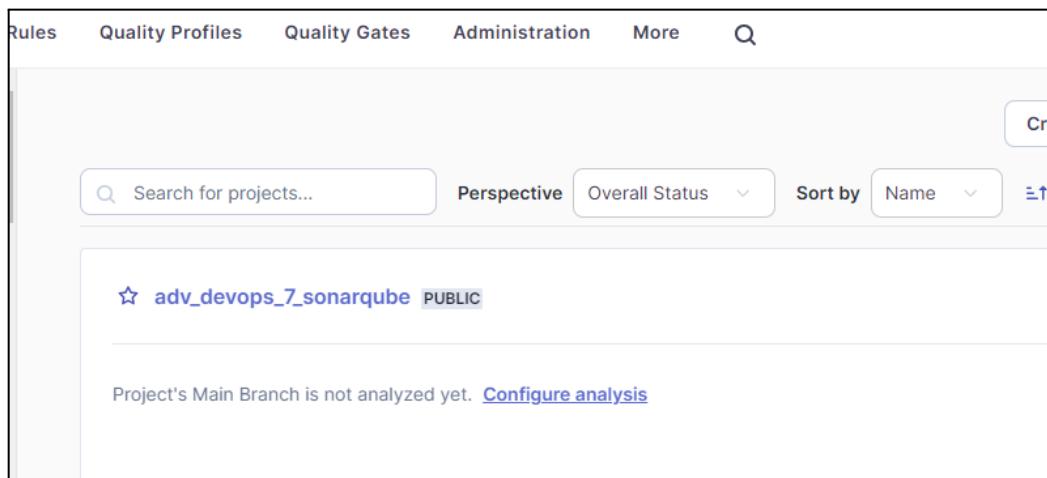
Step 1: Generate a New Authentication Token in SonarQube

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).

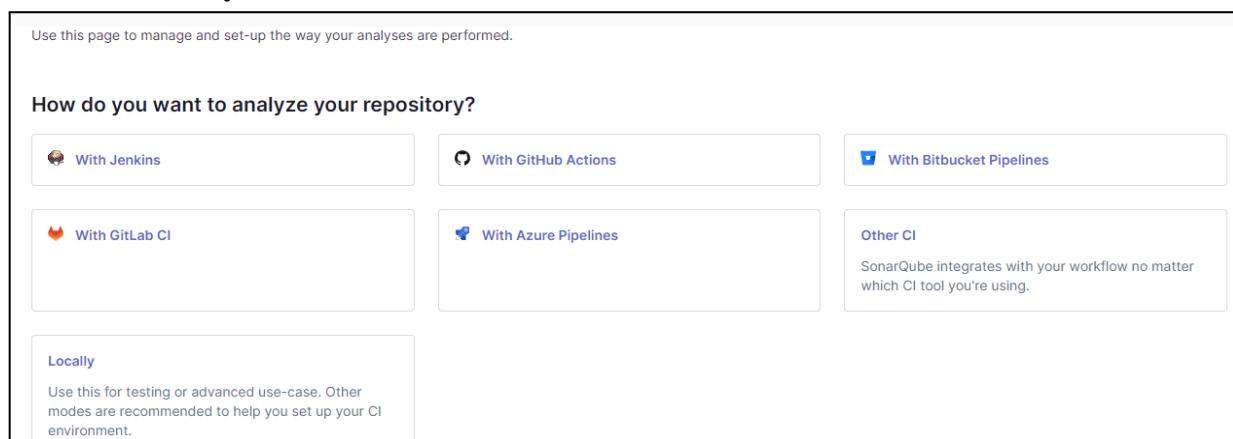
2. Generate a New Token:

- 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 tabs: Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation bar is a search bar with placeholder text "Search for projects...". To the right of the search bar are buttons for "Perspective", "Overall Status", "Sort by", and "Name". A blue button labeled "Create" is visible on the far right. 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." followed by a link "Configure analysis".

- Click on **Locally**



The screenshot shows the SonarQube CI setup interface. At the top, a message says "Use this page to manage and set-up the way your analyses are performed.". Below this, a section titled "How do you want to analyze your repository?" contains several options:

- With Jenkins
- With GitHub Actions
- With Bitbucket Pipelines
- With GitLab CI
- With Azure Pipelines
- Other CI

A callout box highlights the "Locally" option, which is described as "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" box, a note says "SonarQube integrates with your workflow no matter which CI tool you're using."

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

1 Provide a token

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

Token name [?](#) Expires in

"adv_devops_7.sonarqube" 1 year [Generate](#)

Info 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 a summary of recent builds and a "New Item" button.
- Build History:** A table showing build status (Success or Failure), name, last success/failure time, and duration for three projects: Pipeline2, aryan27, and aryan2711.
- Project Relationship:** A link to manage project relationships.
- Check File Fingerprint:** A link to check file fingerprints.
- Manage Jenkins:** A link to manage Jenkins settings.
- My Views:** A link to manage user views.
- Build Queue:** A link to view the build queue.

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

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
		System	(global)	sonarqube_token	/*****

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

P	Store ↓	Domains
	System	(global)

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 filled in are:

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

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

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

The screenshot shows the 'Global credentials (unrestricted)' page under 'Manage Jenkins'. The breadcrumb navigation is: Manage Jenkins > Credentials > System > Global credentials (unrestricted). The title 'Global credentials (unrestricted)' is at the top, and a blue '+ Add Credentials' button is on the right.

Below the title, a note says: 'Credentials that should be available irrespective of domain specification to requirements matching.'

ID	Name	Kind	Description
	sonarqube-exp	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 ?
Project properties file path

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

Additional arguments ?
Additional command-line arguments

JVM Options ?
JVM options for the scanner

12. Run the Jenkins build.

Dashboard > adv_devops_exp7 >

Status (✓) adv_devops_exp7

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

SonarQube Quality Gate

adv_devops_7_sonarqube Passed
server-side processing: Success

Permalinks

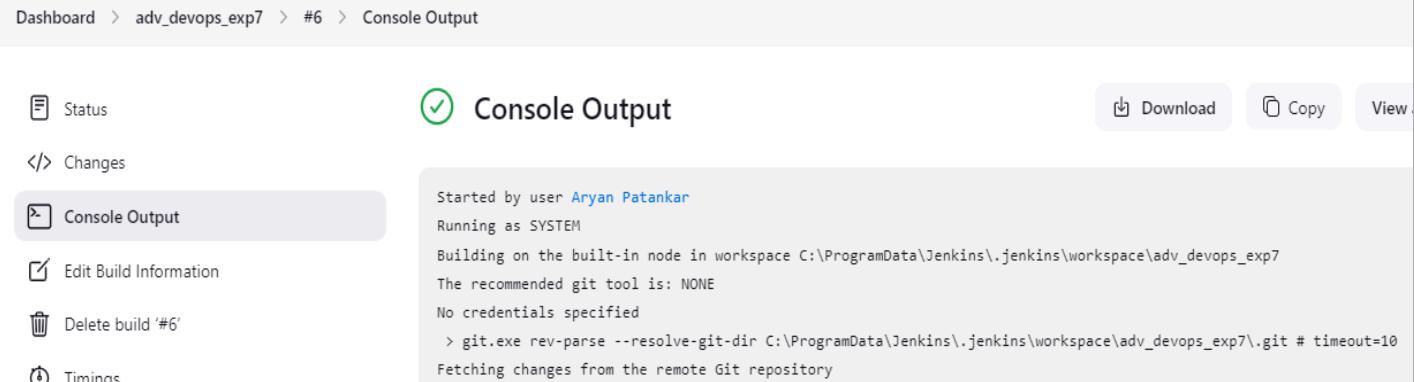
- Last build (#6), 1 min 55 sec ago
- Last stable build (#6), 1 min 55 sec ago
- Last successful build (#6), 1 min 55 sec ago
- Last failed build (#5), 17 min ago
- Last unsuccessful build (#5), 17 min ago
- Last completed build (#6), 1 min 55 sec ago

Build History trend /

Filter... /

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

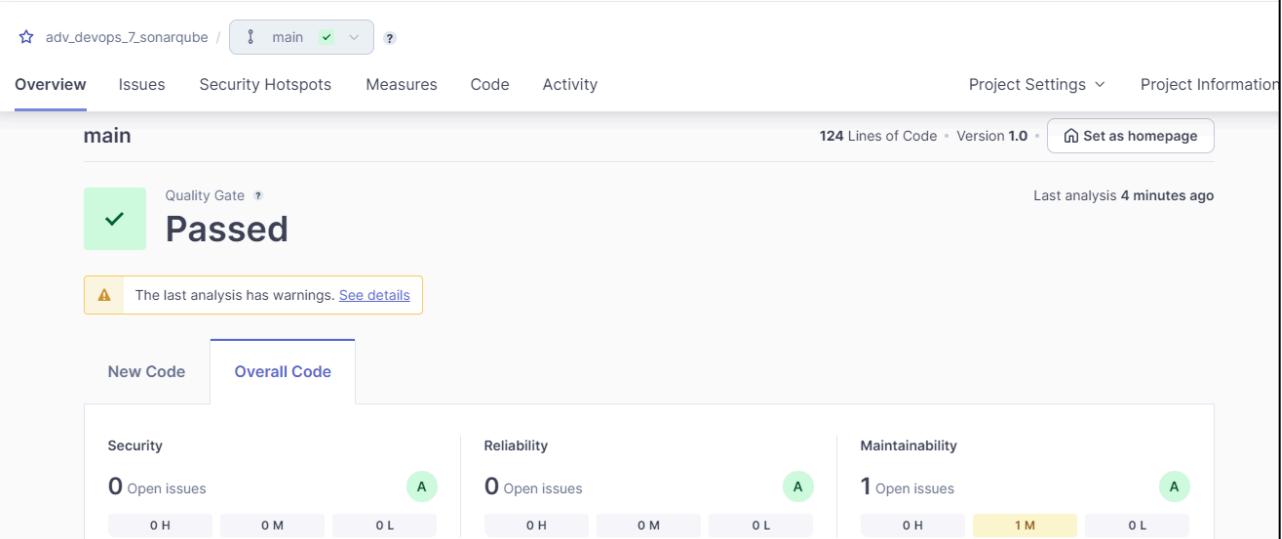
Check the console Output



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

```
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' page 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 shows the Quality Gate status as 'Passed' (green checkmark). It also displays the following metrics:

Category	Value	Status
New Code	0 H, 0 M, 0 L	A
Overall Code	0 H, 0 M, 0 L	A
Security	0 Open issues	A
Reliability	0 Open issues	A
Maintainability	1 Open issue	A

Other details shown include 124 Lines of Code, Version 1.0, and a 'Last analysis 4 minutes ago' message.

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

ADVANCE DEVOPS EXP 8

Name:Aryan Anil Patankar

Class:D15A

Roll No.34

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.

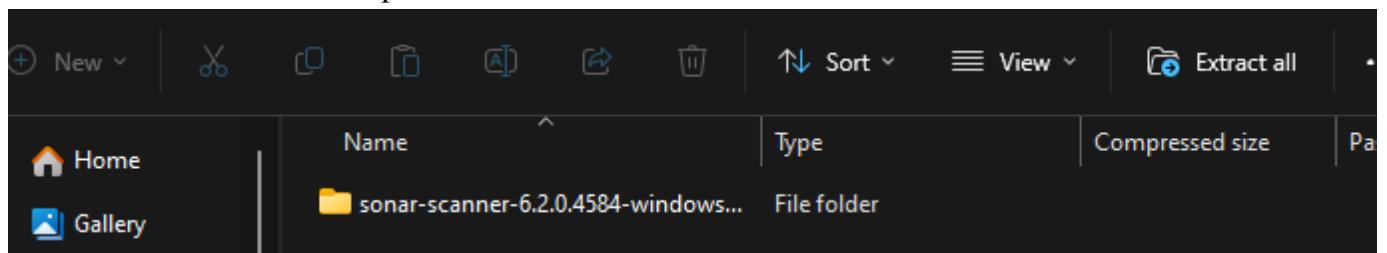
Step 1: Download sonar scanner

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

The screenshot shows a web browser displaying the SonarScanner CLI documentation. The URL in the address bar is <https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan>. The page title is "SonarScanner CLI". On the left, there is a sidebar with navigation links for "Homepage", "Try out SonarQube", "Server installation and setup", "Analyzing source code" (which is expanded to show "SonarQube analysis overview", "Project analysis setup", "Scanners" (expanded to show "Scanner environment", "SonarScanner CLI", "SonarQube extension for Azure DevOps", "SonarQube extension for Jenkins", "SonarScanner for .NET", "SonarScanner for Maven")), and "Docs 10.6". The main content area features a card for "6.1" released on "2024-06-27". It includes sections for "macOS and Linux AArch64 distributions", "Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM)", and "Release notes". Below the card, there is a note about the SonarScanner CLI being the scanner to use when no specific scanner is available, and another note about the SonarScanner not supporting ARM architecture yet.

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

Extract the downloaded zip file in a folder.



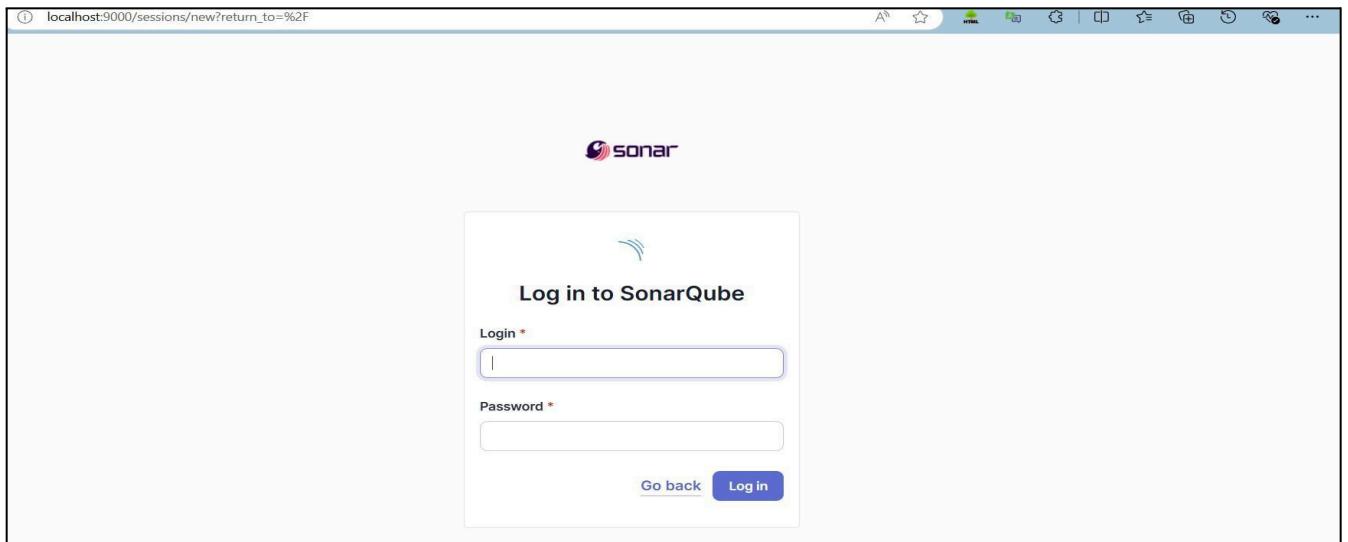
1. Install sonarqube image

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.

A screenshot of the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation, a heading says 'How do you want to create your project?'. It asks if you want to benefit from SonarQube's features like repository import and Pull Request decoration, and suggests creating a project from a favorite DevOps platform. It then lists several import options: 'Import from Azure DevOps' (Setup), 'Import from Bitbucket Cloud' (Setup), 'Import from Bitbucket Server' (Setup), 'Import from GitHub' (Setup), and 'Import from GitLab' (Setup). At the bottom, there's a note about testing or advanced use-cases, followed by a button labeled '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 attention on the Clean as You Code methodology. Learn more: [Defining New Code](#) 

Choose the baseline for new code for this project

Use the global setting

Previous version

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

Define a specific setting for this project

Previous version

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

5. 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 details:

- Left sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", and "Manage Jenkins".
- Build Queue:** Shows "No builds in the queue."
- Build Executor Status:** Shows 1 Idle and 2 Idle nodes, with one node labeled "(offline)".
- Central Table:** Displays a list of build jobs with columns: S (Status), W (Last Success), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Sun icon	Devops Pipeline	1 mo 13 days #4	N/A	0.61 sec
Green checkmark	Sun icon	devops_exp6_pipeline	24 days #1	N/A	2.2 sec
Green checkmark	Cloud icon	maven_exp_6	17 days #13	17 days #12	9.2 sec
Red X	Cloud icon	maven_project	1 mo 13 days #3	1 mo 7 days #10	12 sec
Green checkmark	Sun icon	myNewJob	24 days #1	N/A	0.49 sec
- Bottom:** Icon legend for S (Success), M (Medium), and L (Large).

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

The screenshot shows the Jenkins Manage Jenkins > Plugins page with the following details:

- Left sidebar:** Includes "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings".
- Search Bar:** Contains the search term "sonarq".
- Table:** Shows the "SonarQube Scanner" plugin information.

Install	Name	Released
<input type="checkbox"/>	SonarQube Scanner 2.17.2	6 mo 29 days ago

Details for the SonarQube Scanner plugin:
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 following details:

- Left sidebar:** Includes "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings".
- Current Selection:** "Download progress" (selected).
- Preparation:** A bulleted list of steps:
 - Checking internet connectivity
 - Checking update center connectivity
 - Success
- Progress:** Shows the status of "SonarQube Scanner" and "Loading plugin extensions". Both are marked as "Success".
- Buttons:** "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'. Each section has a corresponding 'Add' button.

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' (checkbox checked), 'Version' (set to 'SonarQube Scanner 6.2.0.4584'), and an 'Add Installer' button.

the “Install automatically” option.

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

New Item

Enter an item name
adv_devops_exp8

Select an item type

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

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

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

OK

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

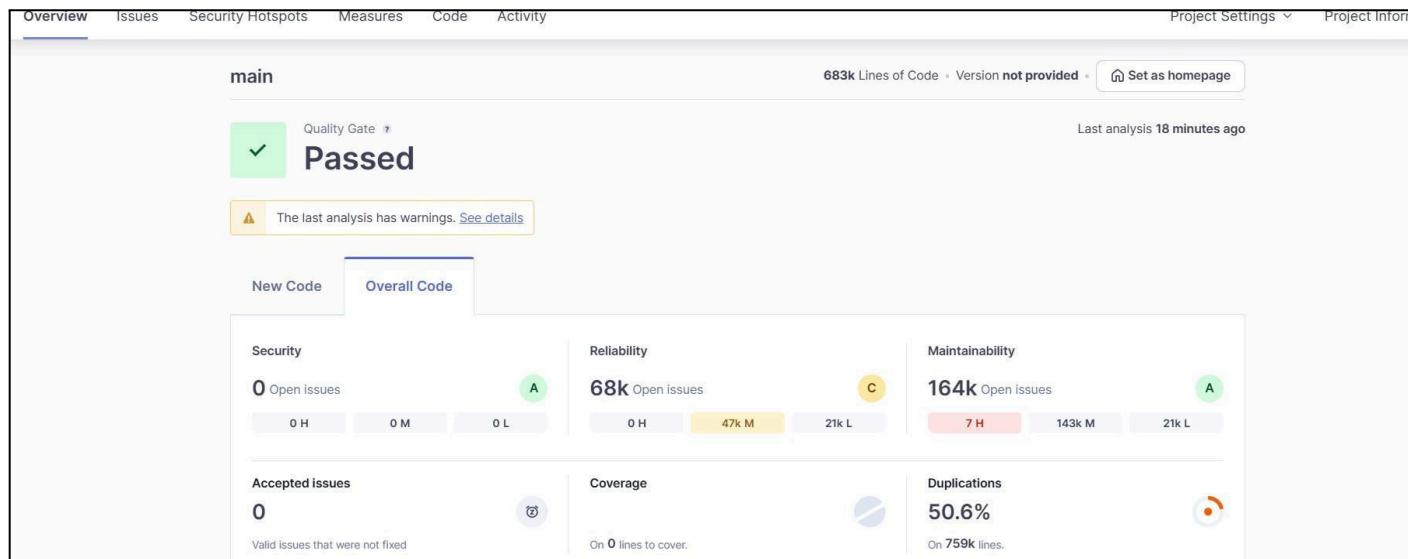
The screenshot shows the 'Stage View' for the pipeline 'adv_devops_exp8'. The view displays three stages: 'Cloning the GitHub Repo' (3s), 'SonarQube analysis' (40s), and 'Build History' (Sep 18, 16:14). The 'Build History' stage has three sub-builds: #9 (2s, green), #8 (2s, red, failed), and #7 (2s, red, failed). The overall average stage time is 3s.

12. Check console

The screenshot shows the 'Console Output' section of the pipeline interface. It displays a log of warnings from JMeter, specifically regarding duplicate references in the 'PropertyControlGui.html' file. The log entries are as follows:

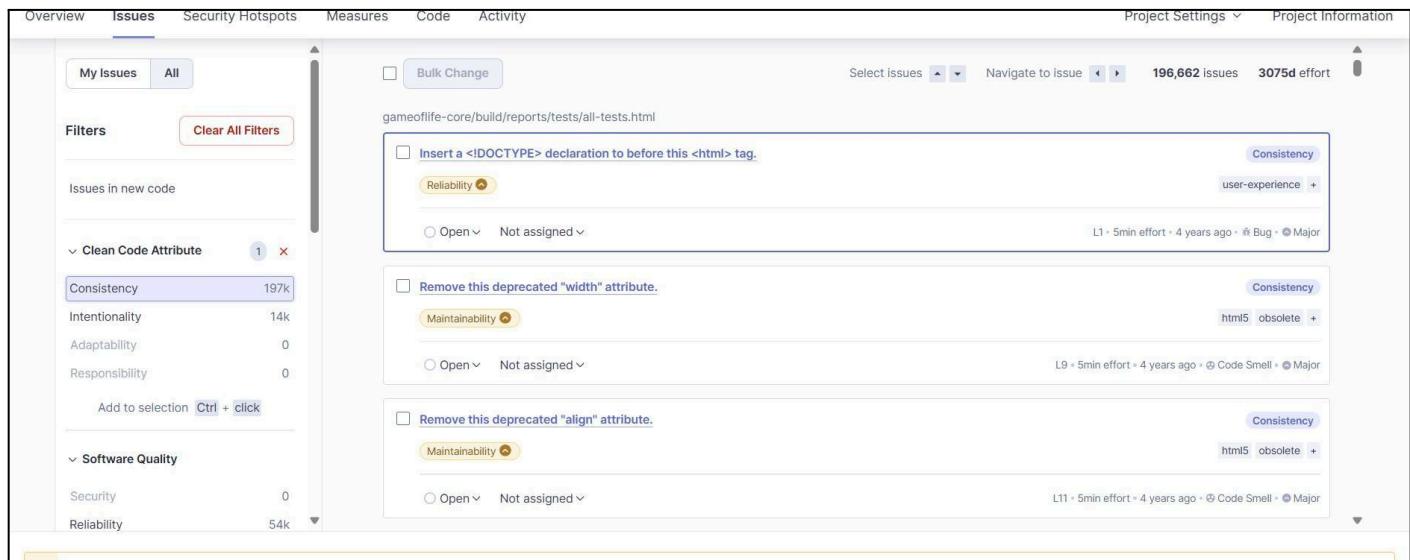
```
Skipping 4,246 KB.. Full Log
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 512. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/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.
16:19:49.752 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's a navigation bar with tabs: Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right side of the header, there are links for Project Settings and Project Information, along with statistics: 13,887 issues and 59d effort.

In the main area, there's a sidebar titled "Filters" with a "Clear All Filters" button. It lists several categories: Issues in new code, Clean Code Attribute (selected), Software Quality, and Reliability. Under "Clean Code Attribute", there are sub-categories: Consistency (197k), Intentionality (14k, highlighted in blue), Adaptability (0), and Responsibility (0). Below these, there's a link to "Add to selection" with a "Ctrl + click" keyboard shortcut.

The main content area displays a list of issues under the heading "gameoflife-acceptance-tests/Dockerfile". Each issue item includes a checkbox, the issue title, a severity level (e.g., Intentionality), a "Maintainability" badge, and a "No tags +" link. There are also dropdown menus for status ("Open" or "Not assigned") and a timestamp indicating when the issue was last modified.

- Use a specific version tag for the image. (Intentionality) Maintainability No tags +
Open Not assigned L1 - 5min effort 4 years ago ⚠️ Code Smell ⚠️ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) Maintainability No tags +
Open Not assigned L12 - 5min effort 4 years ago ⚠️ Code Smell ⚠️ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) Maintainability No tags +
Open Not assigned L12 - 5min effort 4 years ago ⚠️ Code Smell ⚠️ Major

Bugs

Screenshot of a bug reporting interface showing three specific issues:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element.** (Intentionality: accessibility, wcag2-a) Status: Open, Not assigned. Effort: L1 - 2min effort, 4 years ago. Type: Bug, Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency: user-experience) Status: Open, Not assigned. Effort: L1 - 5min effort, 4 years ago. Type: Bug, Major.
- Add "<th>" headers to this "<table>".** (Intentionality: accessibility, wcag2-a) Status: Open, Not assigned. Effort: L9 - 2min effort, 4 years ago. Type: Bug, Major.

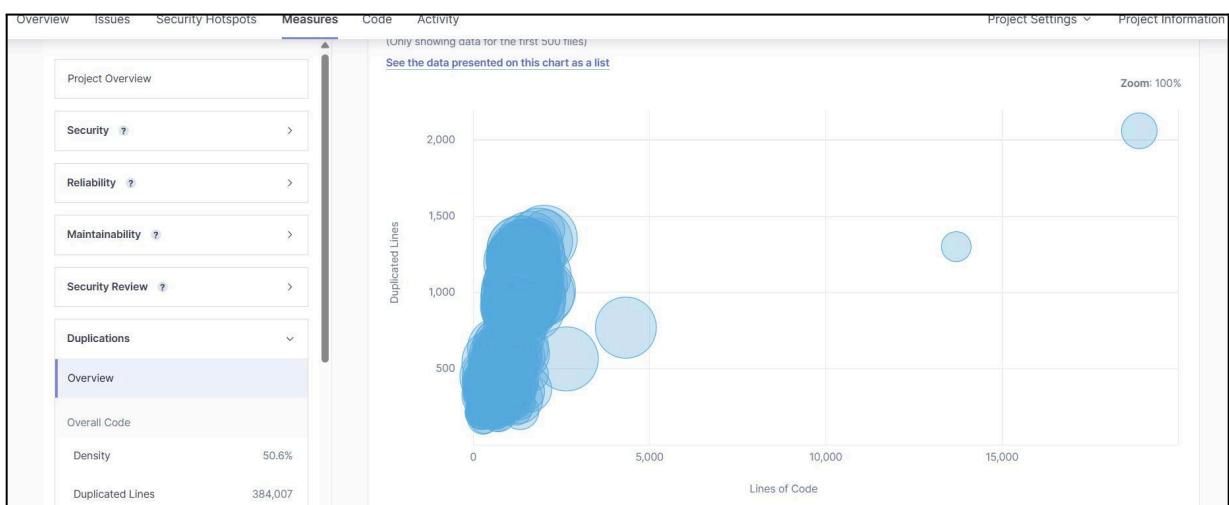
Code Smells

Screenshot of a code smell detection interface showing several issues in Dockerfile:

- Use a specific version tag for the image.** (Intentionality: No tags) Status: Open, Not assigned. Effort: L1 - 5min effort, 4 years ago. Type: Code Smell, Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Intentionality: No tags) Status: Open, Not assigned. Effort: L12 - 5min effort, 4 years ago. Type: Code Smell, Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Intentionality: No tags) Status: Open, Not assigned. Effort: L12 - 5min effort, 4 years ago. Type: Code Smell, Major.

Filters on the left show Clean Code Attribute (Consistency 164k, Intentionality 15, Adaptability 0, Responsibility 0) and Software Quality (Security 0, Reliability 68k, Maintainability 164k).

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 selected. On the left, a sidebar lists various metrics: Security, Reliability, Maintainability, Security Review, Duplications, Size, Complexity, and Cyclomatic Complexity (which is currently selected). The main content area displays the "Cyclomatic Complexity" report, which shows a total of 1,112 issues. The report lists several components and their complexity counts:

Component	Complexity Count
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

At the bottom of the report, it says "6 of 6 shown".

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

ADVANCE DEVOPS EXPERIMENT 9

Name:Aryan Anil Patankar

Class:D15A

Roll No:34

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 Amazon Linux EC2 instance and name it as nagios-host

Instances (1) Info		Last updated 1 minute ago	Connect	Instance state ▾	Actions ▾	Launch instances ▾	Edit filters
		<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states ▾			
<input type="checkbox"/>	Name ▾	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
<input type="checkbox"/>	nagios-host	i-08373a53cb8045f0a	Running Details Logs	t2.micro	Initializing	View alarms +	ap-south-1

Step 2:Edit the following inbound rules of the specified security groups and ensure HTTP,HTTPS,SSH,ICMP are accessible from anywhere

Inbound rules (7)						Edit	Manage tags	Edit inbound rules
						Search		
▼	Security group rule... ▾	IP version	Type	Protocol	Port range			
	sgr-0842dcf237958c987	IPv4	HTTPS	TCP	443			
	sgr-0e3b5fe756fe77f0a	IPv4	All traffic	All	All			
	sgr-07c7572562bdb3...	IPv4	Custom TCP	TCP	0			
	sgr-07882e9275b39c4...	IPv4	HTTP	TCP	80			
	sgr-08540b31df42cc513	IPv4	All ICMP - IPv4	ICMP	All			
	sgr-0dcbe24f99412dcfb	IPv6	Custom TCP	TCP	0			
	sgr-09ccae5af38c85345	IPv6	All ICMP - IPv6	IPv6 ICMP	All			

Step 3: Connect to your EC2 instance via the connect option available in EC2 instances menu

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:23 ago on Thu Sep 26 08:42:17 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
php8.3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 k
Installing dependencies:				
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-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
libxslt	x86_64	1.1.34-5.amzn2023.0.2	amazonlinux	241 k
mod_wsgi	noarch	2.1.49-2.amzn2023.0.3	amazonlinux	33 k

Step 4: Update and install the required packages

Use the following commands:

sudo yum update

sudo yum install httpd php

sudo yum install gcc glibc glibc-common

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:32 ago on Thu Sep 26 08:42:17 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	32 M
Installing dependencies:				
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux	92 k
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux	887 k
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	10 M
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux	105 k
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux	27 k
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 k
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux	1.4 M
libomp	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k

```
[ec2-user@ip-172-31-33-14 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:27 ago on Thu Sep 26 08:42:17 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.12.3-5.amzn2023.0.1	amazonlinux	422 k

Step 5: Create a new nagios user by writing the following commands

```
sudo adduser -m nagios  
sudo passwd nagios
```

```
Complete!  
[ec2-user@ip-172-31-33-14 ~]$ sudo adduser -m nagios  
[ec2-user@ip-172-31-33-14 ~]$ sudo passwd nagios  
Changing password for user nagios.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[ec2-user@ip-172-31-33-14 ~]$ █
```

Step 6: Create a new user group using **sudo groupadd nagcmd** and Add users to the group using the following commands:

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

```
Complete!  
[ec2-user@ip-172-31-33-14 ~]$ sudo adduser -m nagios  
[ec2-user@ip-172-31-33-14 ~]$ sudo passwd nagios  
Changing password for user nagios.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[ec2-user@ip-172-31-33-14 ~]$ sudo groupadd nagcmd  
[ec2-user@ip-172-31-33-14 ~]$ sudo usermod -a -G nagcmd nagios  
[ec2-user@ip-172-31-33-14 ~]$ sudo usermod -a -G nagcmd apache  
[ec2-user@ip-172-31-33-14 ~]$ mkdir downloads  
[ec2-user@ip-172-31-33-14 ~]$ cd downloads  
[ec2-user@ip-172-31-33-14 downloads]$ wget https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz/download?use_mirror=excellmedia  
--2024-09-26 09:15:54-- https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz/download?use_mirror=excellmedia  
a  
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...  
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://downloads.sourceforge.net/project/nagios/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?ts=gAAAAABm9S2KLFW7LwD1QAJ2jNzqmSJwAPA1mQ-eAYK8z5Nmrv ifVkhbsV-qOfPsLUyICC6yvdHu6UeeIyvNzsVGUTir9BeQ%3D%3D&use_mirror=excellmedia&r= [following]
```

Step 7: Create a directory for Nagios downloads using the following commands-

Commands -

```
mkdir ~/downloads  
cd ~/downloads
```

Also download Nagios and plugin source files

Commands -

```
wget  
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz  
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
```

```

Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-26 09:38:43-- https://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
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://excellmedia.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1 [following]
--2024-09-26 09:38:45-- https://excellmedia.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viasf=1
Resolving excellmedia.dl.sourceforge.net (excellmedia.dl.sourceforge.net)... 202.153.32.19, 2401:fb00:0:1fe:8000::5
Connecting to excellmedia.dl.sourceforge.net (excellmedia.dl.sourceforge.net)|202.153.32.19|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz          100%[=====] 1.72M 8.14MB/s   in 0.2s

2024-09-26 09:38:45 (8.14 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

[ec2-user@ip-172-31-33-14 downloads]$ ls
'download?use_mirror=excellmedia'  nagios-4.0.8.tar.gz
[ec2-user@ip-172-31-33-14 downloads]$ tar -xzf nagios-4.0.8.tar.gz
[ec2-user@ip-172-31-33-14 downloads]$ []

```

Step 8-Extract the nagios source file with the following commands

tar zxvf nagios-4.4.6.tar.gz

cd nagios-4.4.6

Then run the configuration script with the following command

/configure --with-command-group=nagcmd

```

Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: ${prefix}/var/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/rc.d/init.d
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

```

Web Interface Options:

```

-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/bin/traceroute

```

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

```
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ ]
```

Step 9-Compile the source code with the following commands
make all

```
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.0.8/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nerd.o nerd.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list':
  inlined from 'get_worker' at workers.c:224:12:
workers.c:209:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  209 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd name);
    |         ^~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
commands.c: In function 'process_passive_service_check':
commands.c:2247:19: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
```

Step 10-Install binaries,init script and sample config files

Commands -

./sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
*** Config files installed ***

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

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

*** External command directory configured ***

[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ ]
```

Step 11-Edit the Config File to Change the Email Address

Commands -

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

- Change the email address in the contacts.cfg file to your preferred email

Step 12-Configure the Web Interface

Commands -

sudo make install-webconf

```

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                nagios@localhost ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

# we only have one contact in this simple configuration file, so there is

```

File: /usr/local/nagios/etc/objects/contacts.cfg

Step 13-Create a Nagios Admin Account

Commands -

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

- You will be prompted to enter and confirm the password for the nagiosadmin user

```

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                vaishnal16305@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

# we only have one contact in this simple configuration file, so there is

```

File: /usr/local/nagios/etc/objects/contacts.cfg

Step 14-. Extract the Plugins Source File

Commands -

cd ~/downloads

tar zxvf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

```

*** External command directory configured ***

[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf

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

```

Step 15-19. Compile and Install Plugins

Commands -

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

```
make
```

```
sudo make install
```

```
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Step 16-Start Nagios

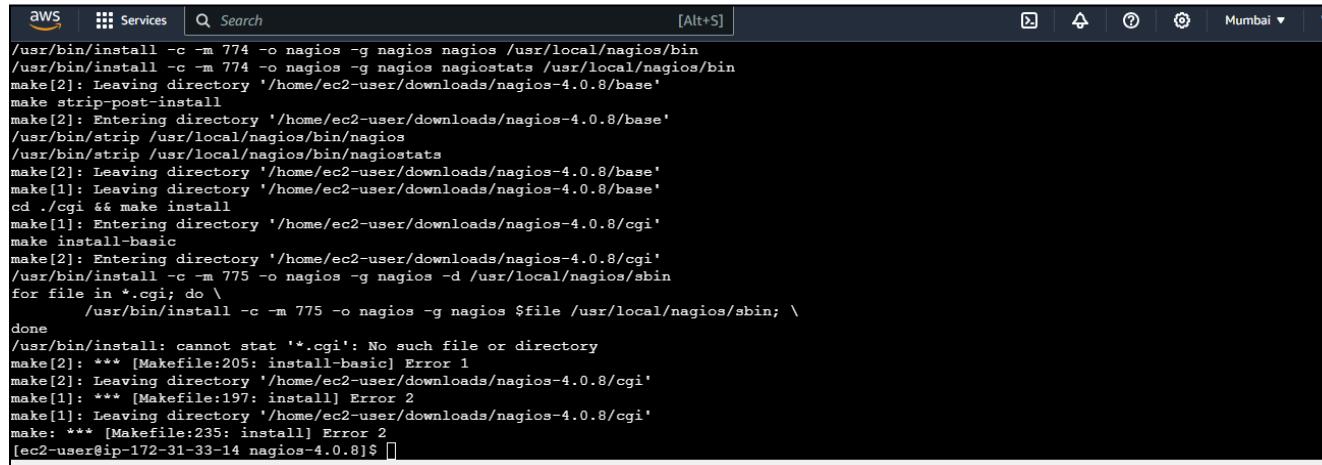
Commands -

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

```
sudo chkconfig nagios on
```

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

```
sudo systemctl start nagios
```



```
/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 '/home/ec2-user/downloads/nagios-4.0.8/base'
make strip-post-install
make[2]: Entering directory '/home/ec2-user/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 '/home/ec2-user/downloads/nagios-4.0.8/base'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.0.8/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/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 $file /usr/local/nagios/sbin; \
done
/usr/bin/install: cannot stat '*.cgi': No such file or directory
make[2]: *** [Makefile:205: install-basic] Error 1
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8/cgi'
make[1]: *** [Makefile:197: install] Error 2
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8/cgi'
make: *** [Makefile:235: install] Error 2
[ec2-user@ip-172-31-33-14 nagios-4.0.8]$
```

Step 17-Access Nagios Web Interface

- Copy the Public IP address of your EC2 instance.
- Open your browser and navigate to <http://nagios>.
- Enter the username nagiosadmin and the password you set in Step 16.

Nagios® Core™



Unable to get process status

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

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

General

- [Home](#)
- [Documentation](#)

Current Status

- [Tactical Overview](#)
- [Map \(Legacy\)](#)
- [Hosts](#)
- [Services](#)
- [Host Groups](#)
 - [Summary](#)
 - [Grid](#)
- [Service Groups](#)
 - [Summary](#)
 - [Grid](#)
- Problems**
- [Services \(Unhandled\)](#)
- [Hosts \(Unhandled\)](#)
- [Network Outages](#)

Quick Search:

Reports

- [Availability](#)
- [Trends \(Legacy\)](#)
- [Alerts](#)
- [History](#)
- [Summary](#)
- [Histogram \(Legacy\)](#)

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- [Nagios Library](#) (tutorials and docs)
- [Nagios Labs](#) (development blog)
- [Nagios Exchange](#) (plugins and addons)
- [Nagios Support](#) (tech support)
- [Nagios.com](#) (company)
- [Nagios.org](#) (project)

Latest News

Don't Miss...

ADVANCE DEVOPS EXPERIMENT 10

Name:Aryan Anil Patankar
Class;D15A
Roll No:34

1) Launch an instance

Launch an ec2 instance.

Select Ubuntu as the os give a meaningful name of the instance.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the instance name is set to 'exp10client'. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Quick Start' tab is selected, showing options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Ubuntu' option is highlighted. The summary on the right indicates 1 instance, using the Canonical, Ubuntu, 24.04 AMI, with a t2.micro virtual server type, launch-wizard-5 security group, and 1 volume(s) - 8 GiB storage. A note about the free tier is also present.

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

Application and OS Images (Amazon Machine Image) Info

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li SUS

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

Summary

Number of instances Info 1

Software Image (AMI) Canonical, Ubuntu, 24.04, ami-0e86e20dae9224db8

Virtual server type (instance) t2.micro

Firewall (security group) launch-wizard-5

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

Free tier: In your first 750 hours of t2.micro usage in the Regions in which it's available, you can launch up to 10 free-tier AMIs per month, receive a public IPv4 address upon launch, 30 GB of EBS storage, 1 million I/Os, 1 GB of traffic, and 100 GB of bandwidth to and from the internet.

Cancel

Select the same security group as given in exp9.

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

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

Recents Quick Start

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (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 Username | ⓘ Verified provider

64-bit (x86) ▾ ami-0e86e20dae9224db8 ubuntu

▼ Summary

Number of instances: 1

Software Images: Canonical, Ubuntu, Red Hat Enterprise Linux, SUSE Linux Enterprise Server

Virtual server type: t2.micro

Firewall (security group): launch-wizard-1

Storage (volume type): 1 volume(s) - 80 GB

Free tier: 750 hours in the Region, unavailability tier AMI public IP per month, million requests per month, 100 GB internet bandwidth

Cancel

Make sure to select the same key-pair login used in the exp9 machine.

The screenshot shows the AWS Launch Wizard configuration interface. On the left, there are two main sections: "Key pair (login)" and "Network settings".

Key pair (login): A dropdown menu is set to "nagios_exp_9". To its right is a button labeled "Create new key pair".

Network settings: This section includes:

- A "Network" dropdown set to "vpc-07b6966cbfba88ee3".
- A "Subnet" dropdown with the note "No preference (Default subnet in any availability zone)".
- An "Auto-assign public IP" dropdown set to "Enable".
- A note about "Additional charges apply when outside of free tier allowance".
- A "Firewall (security groups)" dropdown with two options: "Create security group" (radio button unselected) and "Select existing security group" (radio button selected).
- A "Common security groups" dropdown set to "Select security groups".

On the right side of the interface, there is a vertical sidebar with the following information:

- Software**: Canonical, ami-0e86e20
- Virtual server**: t2.micro
- Firewall (security groups)**: launch-wiz
- Storage (volume)**: 1 volume(s)
- Free tier**: 750 hours, the unatiered public monitoring mill 100 instances
- Cancel**

click on launch instance.

Now connect with this client machine using the ssh through your terminal(open a new terminal in your local machine and we will need both of the terminals open)

The screenshot shows the AWS Instances page with the following details:

Instances (1/5) Info

Last updated 2 minutes ago

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
Master	i-0ab175e9c60cc3a23	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-82-156-160.com...
node-1	i-08ad30b7114767ca2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-85-110-80.comp...
node-2	i-03c70d364fb762af5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-226-209-38.co...
nagios_host_e...	i-0820376be204a7fcf	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-224-175-95.co...
exp10client	i-0994ca5a178801a54	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-54-173-58-143.co...

EC2 > Instances > i-0994ca5a178801a54 > Connect to instance

Connect to instance Info

Connect to your instance i-0994ca5a178801a54 (exp10client) using any of these options

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

Instance ID
i-0994ca5a178801a54 (exp10client)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is nagios_exp_9.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "nagios_exp_9.pem"
4. Connect to your instance using its Public DNS:
ec2-54-173-58-143.compute-1.amazonaws.com

Command copied

ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com

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

Cancel

Note to change the path of the .pem file.

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

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

PS C:\Users\Lenovo> ssh -i "C:\Users\Lenovo\Downloads\nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com

The authenticity of host 'ec2-54-173-58-143.compute-1.amazonaws.com (54.173.58.143)' can't be established.
ED25519 key fingerprint is SHA256:IA3XH7f011spk084wDcZFmqRgNn0iJZ7itI2pBMmHP4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-58-143.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

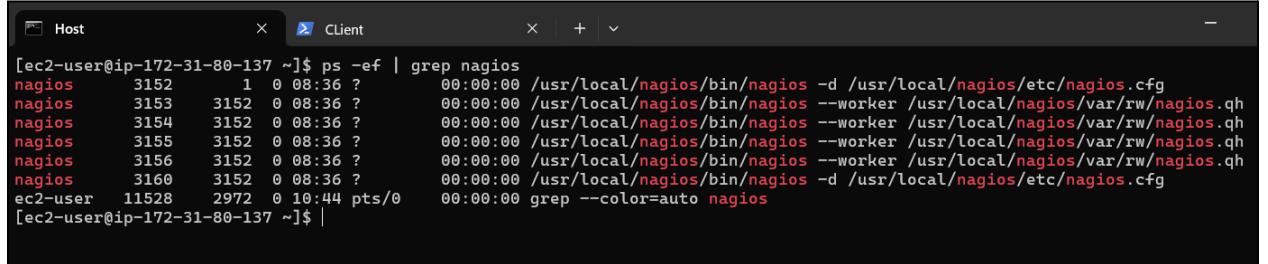
System information as of Sat Sep 28 10:43:28 UTC 2024

System load:  0.01      Processes:          107
Usage of /:   22.8% of 6.71GB  Users logged in:     0
Memory usage: 19%           IPv4 address for enX0: 172.31.82.77
```

2) Go to nagios host machine (Host machine)

Perform the following commands

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



```
[ec2-user@ip-172-31-80-137 ~]$ ps -ef | grep nagios
nagios    3152     1  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    3153   3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3154   3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3155   3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3156   3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3160   3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  11528  2972  0 10:44 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-137 ~]$
```

```
sudo su
```

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

```
[root@ip-172-31-80-137 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-137 ec2-user]# ls
```

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

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

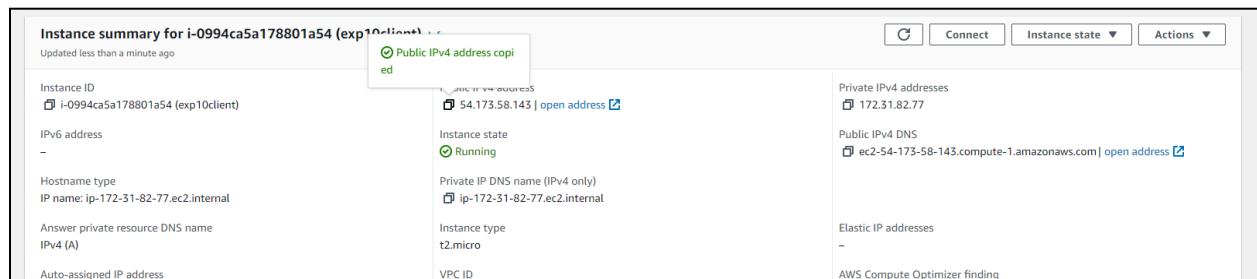
```
[root@ip-172-31-80-137 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

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

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Change hostname and alias to linuxserver

Change address to public ip address of client instance (Ubuntu instance) you can get the ip address by clicking on the instance id on the instances section there you will get the public ipv4 address



```

# HOST DEFINITION
#
#####
# Define a host for the local machine

define host {
    use          linux-server           ; Name of host template to use
                                         ; This host definition will in>
                                         ; in (or inherited by) the lin>
    host_name    linuxserver
    alias        linuxserver
    address     54.173.58.143
}

```

Change hostgroup_name to linux-servers1

```

# Define an optional hostgroup for Linux machines

define hostgroup {
    hostgroup_name      linux-servers1      ; The name of the hostgroup
    alias               Linux Servers        ; Long name of the group
    members             localhost           ; Comma separated list of host>
}
|
```

Change the occurrences of hostname further in the document from localhost to linuxserver
example like:

host_name	localhost
------------------	------------------

changed to

define service {	local-service	; Name of service template
use	linuxserver	
host_name		
service_description	PING	
check_command	check_ping!100.0,20%!500.0,60%	
}		

This is the last one

```

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

```

now ctrl+O and enter to save and then ctrl+X for exiting.
 Open nagios configuration file and add the line shown below
 nano /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

##Add this line below the opened nano interface where similar lines are commented.

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

```

GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
:cfg_file=/usr/local/nagios/etc/objects/commands.cfg
:cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
:cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
:cfg_file=/usr/local/nagios/etc/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/

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The SCIs read object definitions from
# the cache instead of reading them from disk.
:cache_dir=/tmp/nagios_cache

```

ctrl+o and enter for saving and ctrl+x to exit nano editor.

Verify configuration files

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

```
[root@ip-172-31-80-137 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL
```

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

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

```
Checking objects...
```

```
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...
```

```
Total Warnings: 0
Total Errors: 0
```

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# |
```

Restart nagios service.

```
service nagios restart
```

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-80-137 ec2-user]# |
```

- 3) Go to client machine (ubuntu machine)

Perform the following commands

```
sudo apt update -y
```

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

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

```
ubuntu@ip-172-31-82-77:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #1: sshd[990,1101]
ubuntu @ user manager service: systemd[996]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-77:~$ |
```

Open the nrpe.cfg file in nano editor

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

Under allowed_hosts, add the nagios host ip address (public)

```
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd.
nrpe_user=nagios

#
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd.
nrpe_group=nagios

#
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd.
allowed_hosts=127.0.0.1,54.224.175.95

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
```

again save and exit the nano editor.

4) Go to nagios dashboard and click on hosts

The screenshot shows the Nagios Core dashboard. On the left, there's a sidebar with links for General, Current Status, Problems, Reports, and System. The 'Current Status' section is expanded, showing links for Tactical Overview, Map, Hosts, Services, Host Groups, and Service Groups. The main content area displays the Nagios Core logo and version information (Version 4.5.5, September 17, 2024). It also shows a green checkmark indicating the daemon is running with PID 13935. Below this are three boxes: 'Get Started' (with bullet points like 'Start monitoring your infrastructure'), 'Latest News' (empty), and 'Don't Miss...' (empty). A 'Quick Links' box on the right contains links to Nagios Library, Nagios Labs, Nagios Exchange, Nagios Support, Nagios.com, and Nagios.org. At the bottom, there's a copyright notice and a link to the extend page.

Not secure | 54.224.175.95/nagios/

Nagios® Core™

✓ Daemon running with PID 13935

Nagios® Core™
Version 4.5.5
September 17, 2024
Check for updates

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

Latest News

Don't Miss...

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)

Copyright © 2010-2024 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, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

<https://go.nagios.com/nagioscore/extend>

Click on hosts

The screenshot shows the 'Tactical Overview' page, which is part of the 'Current Status' section. It features a sidebar with links for Hosts, Services, and Host Groups. The main content area is currently empty, indicated by a large white space.

5) Click on linux server

Nagios®

Current Network Status

Last Updated: Sat Sep 28 11:33:24 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems	All Types		
0	2		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0
All Problems	All Types			
4	16			

Host Status Details For All Host Groups

Limit Results: 100 ▾

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 11:29:10	0d 0h 8m 36s	PING OK - Packet loss = 0%, RTA = 1.18 ms
localhost	UP	09-28-2024 11:32:18	0d 3h 53m 7s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

Reports

- Availability
- Trends
- Alerts
- History
- Summary
- Histogram
- Notifications
- Event Log

Nagios®

General

Home Documentation

Current Status

Tactical Overview Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

Reports

- Availability
- Trends
- Alerts
- History
- Summary
- Histogram
- Notifications
- Event Log

System

Comments Downtime Process Info Performance Info Scheduling Queue Configuration

Host Information

Last Updated: Sat Sep 28 11:33:24 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

Host
linuxserver
(linuxserver)

Member of
No hostgroups

54.173.58.143

Host State Information

Host Status:	UP (for 0d 0h 8m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.18 ms
Performance Data:	rta=1.18400ms;3000.000000;5000.000000;0.000000 pl=0%;80,100,100
Current Attempt:	1/10 (HARD state)
Last Check Time:	09-28-2024 11:29:10
Check Type:	ACTIVE
Check Latency / Duration:	0.00 - 0.005 seconds
Next Scheduled Active Check:	09-28-2024 11:34:10
Last State Change:	09-28-2024 11:24:48
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	09-28-2024 11:33:37 (0d 0h 0m 2s ago)

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

Host Comments

Add a new comment

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it							

6) Click on nagios services

The screenshot shows the Nagios web interface. At the top, there's a navigation bar with links like Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid, and Service Groups. Below this is a 'Nagios' header with a sub-header 'Current Network Status'. It displays the last update time (Sep 29 11:33:58 UTC 2024), the update interval (every 90 seconds), and the Nagios Core version (4.5.5 - www.nagios.org). It also shows that the user is logged in as 'nagiosadmin'. Below this are links for View History, View Notifications, and View Host Status Detail.

The main content area is titled 'Service Status Details For All Hosts'. It lists services across two hosts: 'linuxserver' and 'localhost'. For each host, it shows services grouped by type (e.g., Current Load, Current Users, HTTP, PING, Root Partition, SSH, Swap Usage, Total Processes) and their status (OK, CRITICAL, WARNING, UNKNOWN). Each service entry includes the last check time, duration, attempt count, and a detailed status information section. For example, the 'HTTP' service on 'localhost' is shown as 'WARNING' with a detailed message about a forbidden response.

Host		Service		Status	Last Check	Duration	Attempt	Status Information			
linuxserver	Current Load	OK		OK	09-28-2024 11:30:25	0d 0h 8m 33s	1/4	OK - load average: 0.01, 0.00, 0.00			
	Current Users	OK		OK	09-28-2024 11:31:03	0d 0h 7m 55s	1/4	USERS OK - 2 users currently logged in			
	HTTP	CRITICAL		CRITICAL	09-28-2024 11:29:40	0d 0h 4m 18s	4/4	connect to address 54.173.58.143 and port 80: Connection refused			
	PING	OK		OK	09-28-2024 11:32:18	0d 0h 6m 40s	1/4	PING OK - Packet loss = 0%, RTA = 1.03 ms			
	Root Partition	OK		OK	09-28-2024 11:32:55	0d 0h 6m 3s	1/4	DISK OK - free space / 6105 MB (75.23% inode=98%)			
	SSH	OK		OK	09-28-2024 11:33:33	0d 0h 5m 25s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-Subuntu13.4 (protocol 2.0)			
	Swap Usage	CRITICAL		CRITICAL	09-28-2024 11:32:10	0d 0h 1m 48s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size			
	Total Processes	OK		OK	09-28-2024 11:29:48	0d 0h 9m 10s+	1/4	PROCS OK - 37 processes with STATE = R/Z/D/T			
localhost	Current Load	OK		OK	09-28-2024 11:29:39	0d 3h 53m 5s	1/4	OK - load average: 0.02, 0.01, 0.00			
	Current Users	OK		OK	09-28-2024 11:30:17	0d 3h 52m 27s	1/4	USERS OK - 2 users currently logged in			
	HTTP	WARNING		WARNING	09-28-2024 11:29:46	0d 2h 49m 12s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time			
	PING	OK		OK	09-28-2024 11:31:32	0d 3h 5m 12s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms			
	Root Partition	OK		OK	09-28-2024 11:32:09	0d 3h 50m 35s	1/4	DISK OK - free space / 6105 MB (75.23% inode=98%)			
	SSH	OK		OK	09-28-2024 11:32:47	0d 3h 49m 57s	1/4	SSH OK - OpenSSH_9.7 (protocol 2.0)			
	Swap Usage	CRITICAL		CRITICAL	09-28-2024 11:31:24	0d 3h 12m 34s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size			
	Total Processes	OK		OK	09-28-2024 11:29:02	0d 3h 14m 56s	1/4	PROCS OK - 37 processes with STATE = R/Z/D/T			

Conclusion:

In this lab, we successfully configured a monitoring setup between a Nagios host machine (referred to as "exp9 machine") and a client machine (created specifically for this experiment). The goal was to set up Nagios to monitor a remote Linux server, which involved configuring both the Nagios host and client machine (Ubuntu instance) in an EC2 environment.

ADVANCE DEVOPS EXP 11

Name:Aryan Anil Patankar

Class: D15A

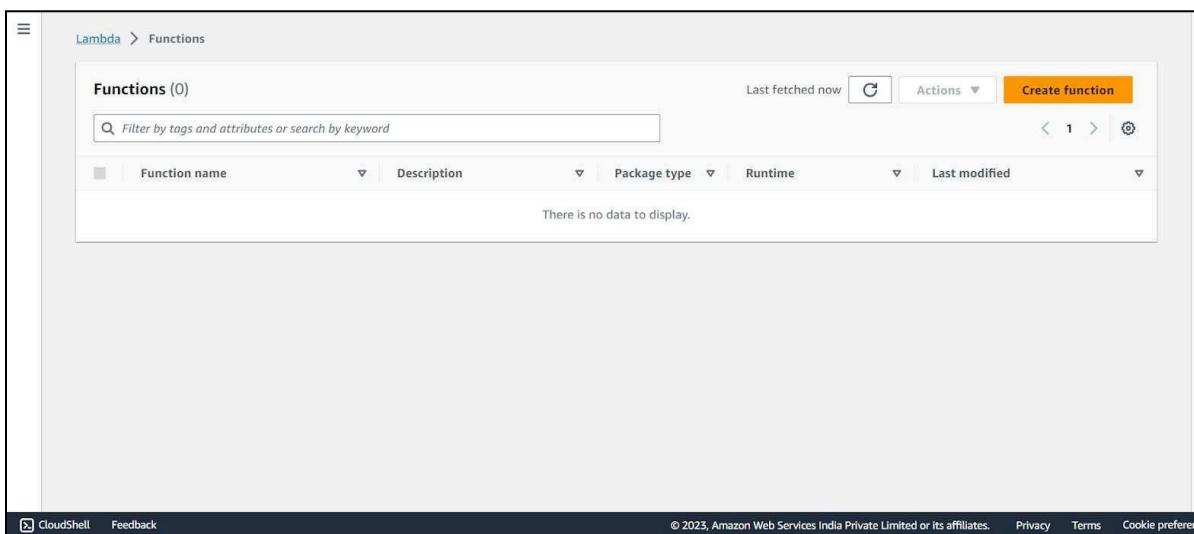
Roll No:34

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.



The screenshot shows the AWS Lambda Functions console. At the top, there's a breadcrumb navigation 'Lambda > Functions'. Below it is a search bar with placeholder text 'Filter by tags and attributes or search by keyword'. To the right of the search bar are buttons for 'Last fetched now' (with a refresh icon), 'Actions' (with a dropdown arrow), and a prominent orange 'Create function' button. Underneath the search bar is a table header with columns: 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. A message 'There is no data to display.' is centered in the table body. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

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.

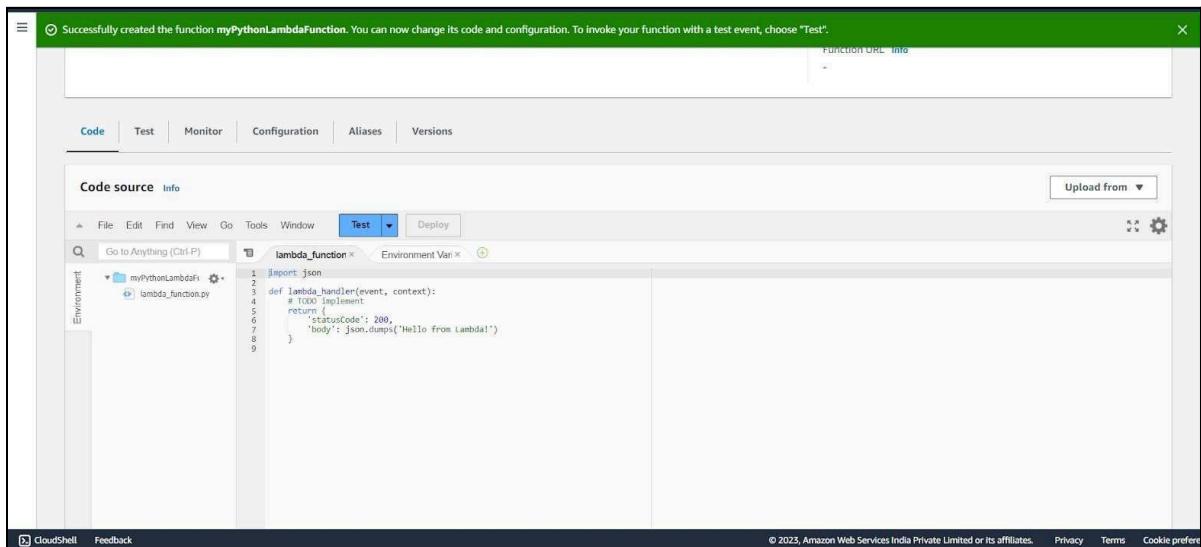
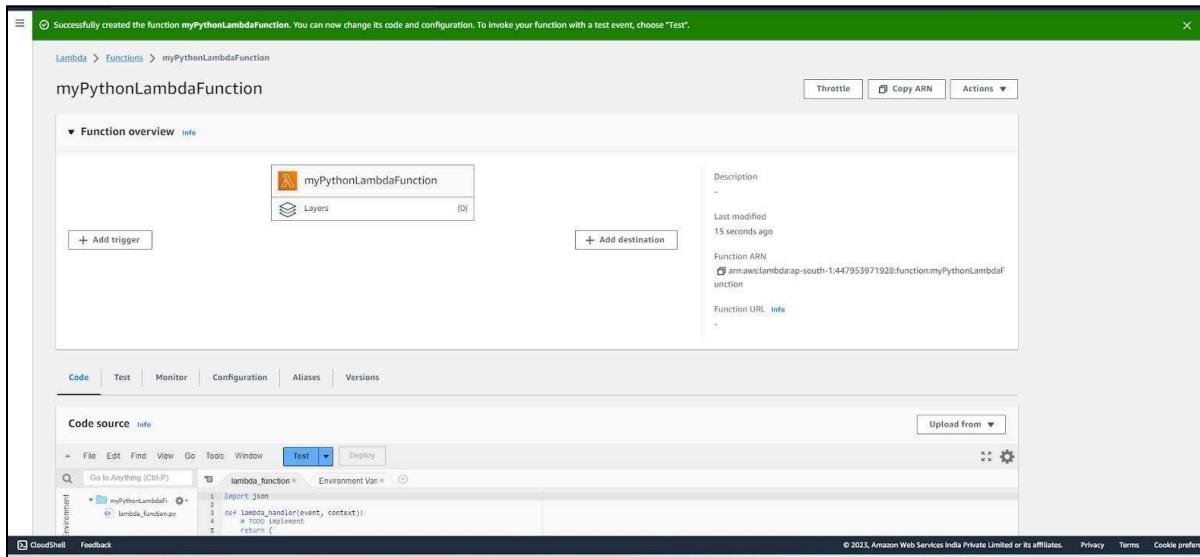
The screenshot shows the 'Create function' wizard on the AWS Lambda console. The top navigation bar includes 'Lambda > Functions > Create function'. The main title 'Create function' has an 'Info' link. A note says 'AWS Serverless Application Repository applications have moved to [Create application](#)'. Below are three options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section contains fields for 'Function name' (set to 'myFunctionName'), 'Runtime' (set to 'Node.js 18.x'), and 'Architecture' (set to 'x86_64'). At the bottom are 'CloudShell', 'Feedback', and footer links.

This screenshot is identical to the one above, but the 'Function name' field is now set to 'myPythonLambdaFunction'. The rest of the configuration remains the same: Node.js runtime and x86_64 architecture.

This screenshot shows the 'Create function' wizard again, but the 'Function name' field is now set to 'myPythonLambdaFunction'. The 'Runtime' is set to 'Python 3.11' and the 'Architecture' is set to 'x86_64'. The 'Permissions' section is partially visible at the bottom.

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 AWS Lambda function configuration interface. On the left, a sidebar lists various configuration options: General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, Monitoring and operations tools, Concurrency, Asynchronous invocation, Code signing, Database proxies, File systems, and State machines. The 'General configuration' tab is selected. The main panel displays the 'General configuration' section with the following details:

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart: Info None	0 min 3 sec

At the bottom of the page, there are links for CloudShell, Feedback, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

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
[Empty input field]

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.
[Input field: 128 MB]
Set memory to between 128 MB and 10240 MB.

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)
[Input field: 512 MB]
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](#).
[Dropdown menu: None]
Supported runtimes: Java 11, Java 17.

Timeout
[Input fields: 0 min 1 sec]

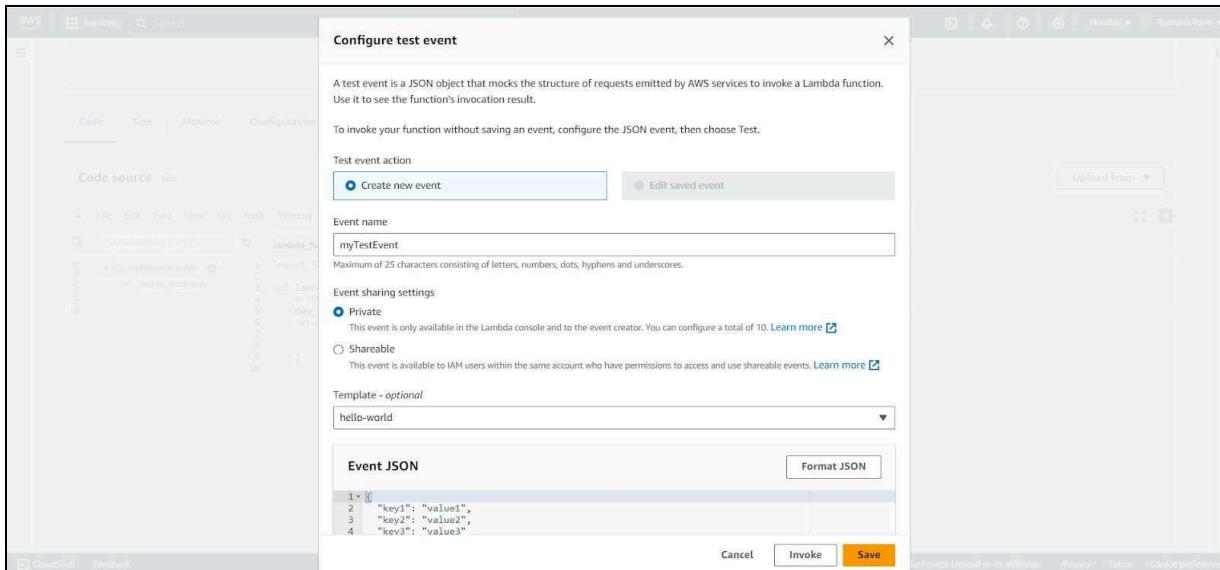
Execution role

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.

```
import json
def lambda_handler(event, context):
    # TODO implement
    new_string="Hello! how are you?"
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

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: 110.05 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 Memory Used: 40 MB Init Duration: 110.05 ms  
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-Aryan Anil Patankar

Class-D15A

Roll No-34

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 sidebar with navigation links like Dashboard, Access management (Roles selected), Policies, Identity providers, Account settings, and Access reports. 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 service it's associated with, and the last time it was used.

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

The screenshot shows the detailed view of the 'myPythonLambdaFunction-role-a2x7el65' role. The left sidebar is identical to the previous screenshot. The main area shows the 'Summary' tab with details like Creation date (October 07, 2023, 16:03 (UTC+05:30)), Last activity (none), ARN (arn:aws:iam::447953971928:role/service-role/myPythonLambdaFunction-role-a2x7el65), and Maximum session duration (1 hour). Below this is the 'Permissions' tab, which lists one policy: 'Permissions policies (1)'. It includes a search bar, a 'Filter by Type' dropdown set to 'All types', and buttons for 'Add permissions', 'Attach policies', and 'Create inline policy'. At the bottom, there are buttons for 'CloudShell' and 'Feedback'.

S3-ReadOnly

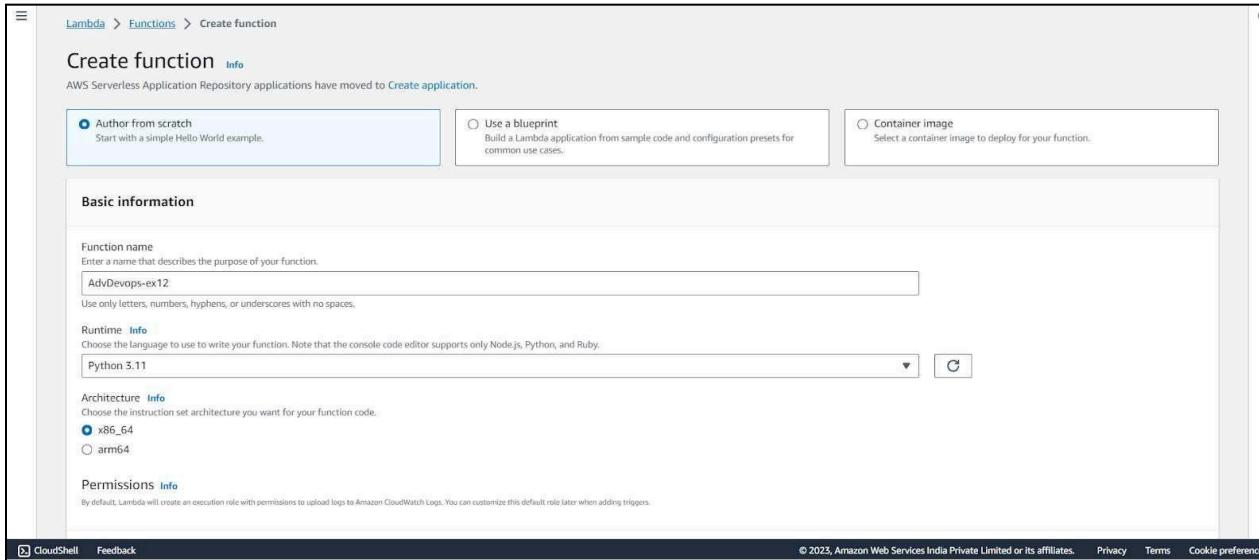
The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The path is IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions. The title is 'Attach policy to myPythonLambdaFunction-role-a2x7el65'. The 'Current permissions policies' section shows one policy: 'AmazonS3ReadOnlyAccess'. The 'Other permissions policies' section is filtered by 'S3read' and shows two results: 'AmazonS3ReadOnlyAccess' (AWS managed) and 'CloudWatchFullAccess' (AWS managed). A search bar at the top right contains 'S3read'. Buttons at the bottom right include 'Cancel' and 'Add permissions'.

CloudWatchFull

This screenshot is identical to the one above, but it includes the 'CloudWatchFullAccess' policy in the list of available policies under 'Other permissions policies'. The search bar now shows 'cloudwatchfull'.

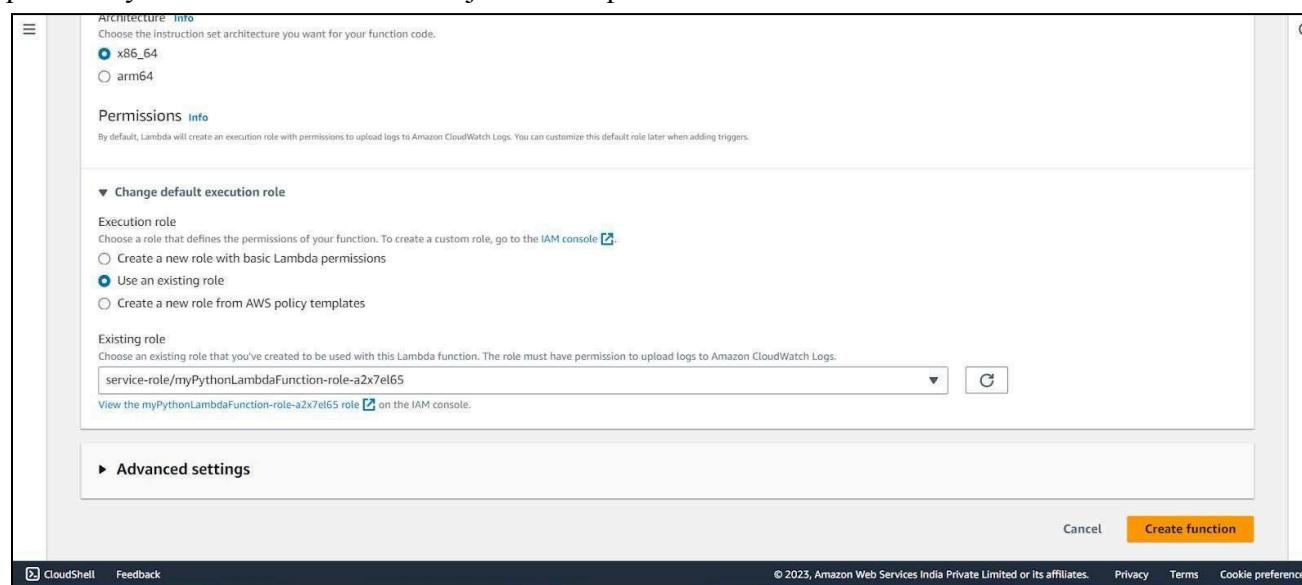
After successful attachment of policy you will see something like this you will be able to see the updated policies.

The screenshot shows the 'Permissions' tab in the IAM console for the 'myPythonLambdaFunction-role-a2x7el65' role. A success message 'Policy was successfully attached to role.' is displayed. The 'Permissions' section lists three attached policies: 'AmazonS3ReadOnlyAccess', 'AWSLambdaBasicExecutionRole-c4946a...', and 'CloudWatchFullAccess'. The 'Attached entities' column shows the count '1' for each. A 'Permissions boundary' section is also present.

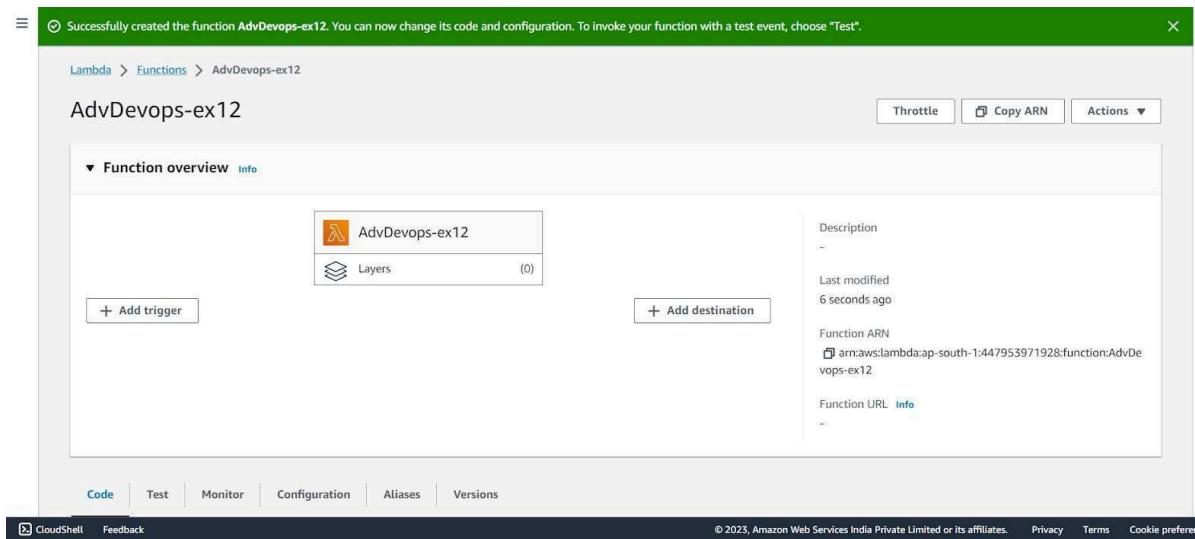


Step 3: Open up AWS Lambda and create a new Python function.

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.x Environment Var x
lambda_function
Environment
Go to Anything (Ctrl-P)
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 = f'An 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
```

© 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, a green banner says "Successfully created the function AdvDevops-ex12. You can now change its code and configuration. To invoke your function, choose Test." Below the banner, there are tabs: Code (selected), Test, Monitor, Configuration, Aliases, and Versions. Under the Code tab, the "Code source" section is active, showing the file structure of "AdvDevops-ex12" with "lambda_function.py". The code editor displays the following Python code:

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

Below the code editor is a "Test" button, a dropdown menu, and a "Deploy" button. A message "Changes not deployed" is displayed. A search bar shows "lambda_function" and a tooltip "Configure test event Ctrl-Shift-C".

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", the "Create new event" option is selected. The "Event name" field is set to "test". Under "Event sharing settings", the "Private" option is selected, with a note: "This event is only available in the Lambda console and to the event creator. You can configure a total of 10." The "Shareable" option is also present with a note: "This event is available to IAM users within the same account who have permissions to access and use shareable events." Under "Template - optional", the value "s3-put" is shown. At the bottom of the modal are "Format JSON", "Cancel", "Invoke", and "Save" buttons.

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 pricing](#)

[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
Disabled

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 with the event name set to 'S3putrequest'. It also shows optional prefix ('images/') and suffix ('.jpg') fields. Below this is the 'Event types' section, which includes an 'Object creation' group. Under 'Object creation', the 'Put' option (s3:ObjectCreated:Put) is checked, while 'Post' (s3:ObjectCreated:Post) is unchecked. At the bottom of the page, there are CloudShell and Feedback links, and a copyright notice for Amazon Web Services India Private Limited.

Event name
S3putrequest

Event name can contain up to 255 characters.

Prefix - optional
Limit the notifications to objects with key starting with specified characters.
images/

Suffix - optional
Limit the notifications to objects with key ending with specified characters.
.jpg

Event types
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

All object create events
s3:ObjectCreated:*

Put
s3:ObjectCreated:Put

Post
s3:ObjectCreated:Post

CloudShell Feedback © 2023, Amazon Web Services India Private Limited

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

The screenshot shows the 'Destination' section. A note states that before publishing messages to a destination, permissions must be granted. The 'Destination' dropdown is set to 'Lambda function'. Under 'Specify Lambda function', the 'Choose from your Lambda functions' option is selected. The Lambda function dropdown contains 'AdvDevops-ex12'. At the bottom right are 'Cancel' and 'Save changes' buttons.

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

Destination
Choose a destination to publish the event. [Learn more](#)

Lambda function
Run a Lambda function script based on S3 events.

SNS topic
Fanout messages to systems for parallel processing or directly to people.

SQS queue
Send notifications to an SQS queue to be read by a server.

Specify Lambda function

Choose from your Lambda functions

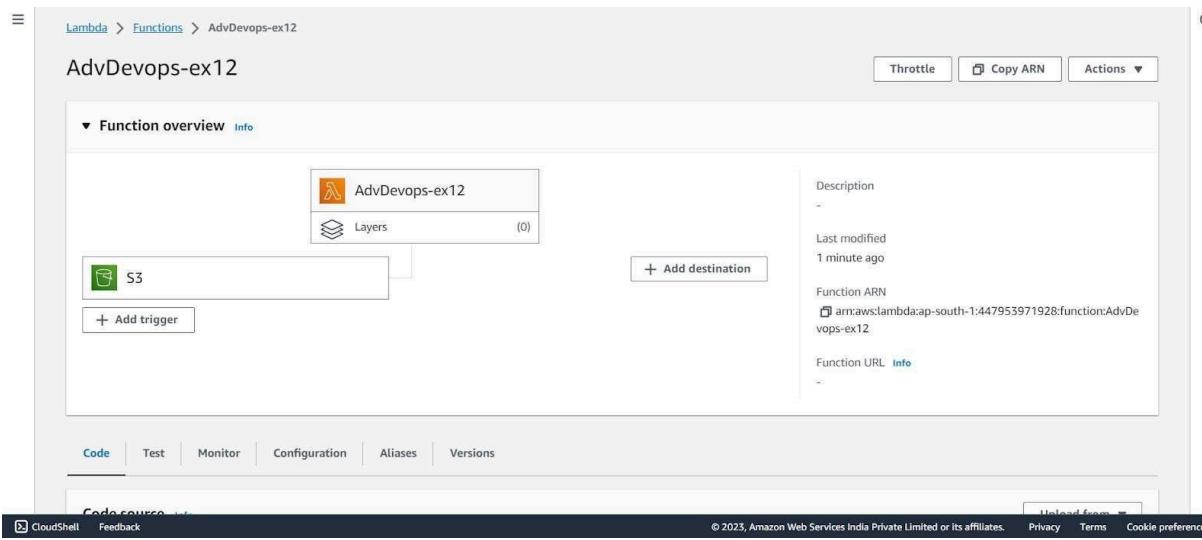
Enter Lambda function ARN

Lambda function

AdvDevops-ex12

Cancel Save changes

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', and a keyboard shortcut '[Alt+S]'. 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 dashed box with the instruction 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' A table titled 'Files and folders (1 Total, 89.0 B)' lists one item: 'dummy.json' (application/json, 89.0 B). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar 'Find by name' is present. The 'Destination' section shows 'Destination' set to '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.

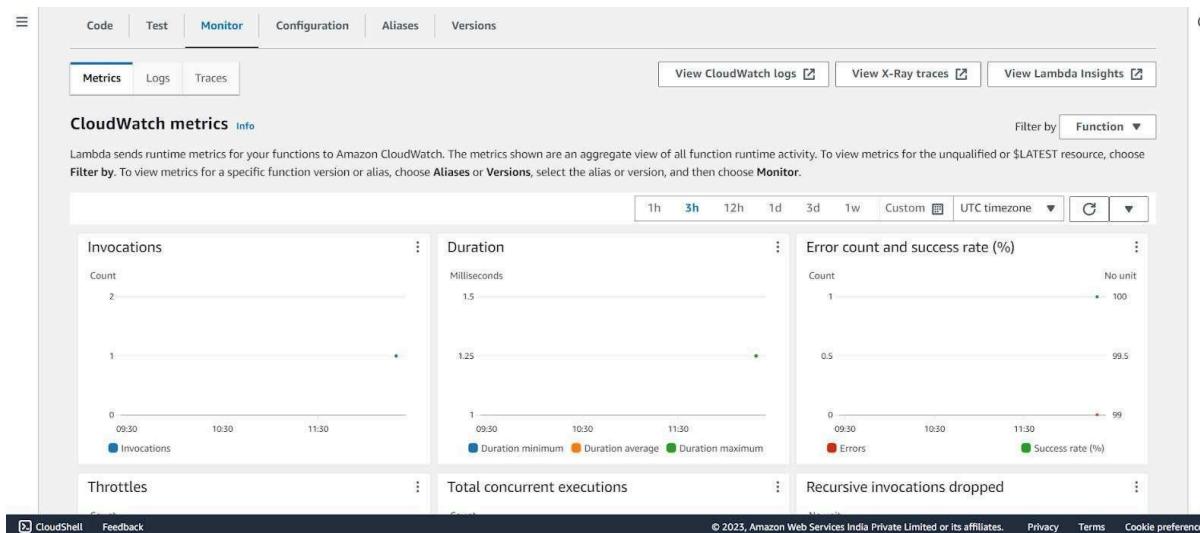
The screenshot shows the Lambda Test Configuration JSON editor. The title is 'Event JSON' and there is a 'Format JSON' button. The JSON code is as follows:

```

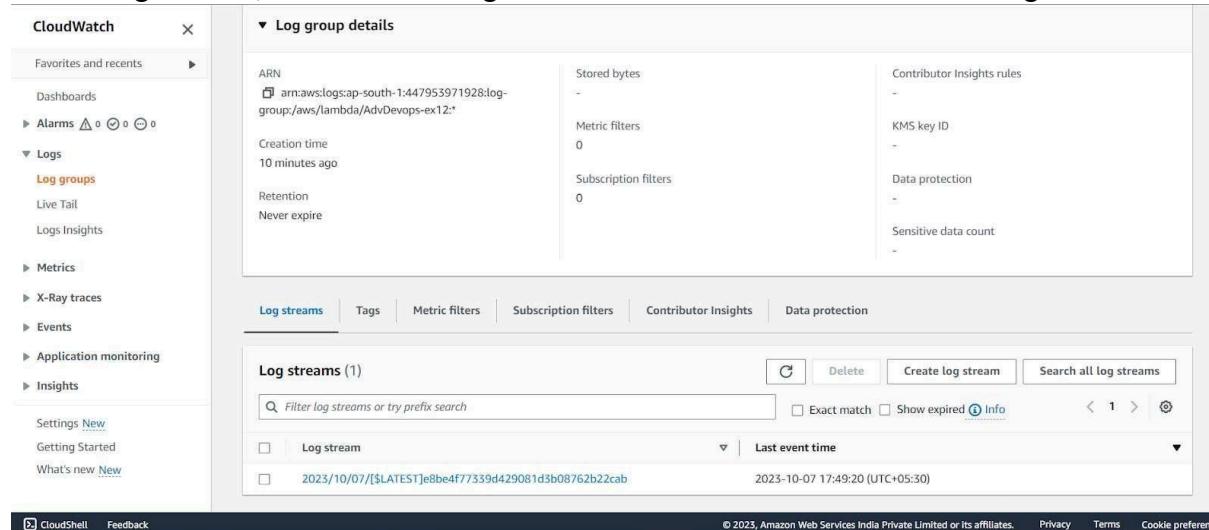
10     "principalId": "EXAMPLE"
11   },
12   "requestParameters": {
13     "sourceIPAddress": "127.0.0.1"
14   },
15   "responseElements": {
16     "x-amz-request-id": "EXAMPLE123456789",
17     "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmnaqrstuvwxyzABCDEFGHIJKLMN"
18   },
19   "s3": {
20     "s3SchemaVersion": "1.0",
21     "configurationId": "testConfigRule",
22     "bucket": {
23       "name": "advopssexp12",
24       "ownerIdentity": {
25         "principalId": "EXAMPLE"
26       },
27       "arn": "arn:aws:s3:::advopssexp12"
28     },
29     "object": {
30       "key": "test%2Fkey",
31       "size": 1024,
32       "eTag": "0123456789abcdef0123456789abcdef",
33       "sequencer": "0A1B2C3D4E5F678901"
34     }
35   }
36 }
37 ]
38 }

```

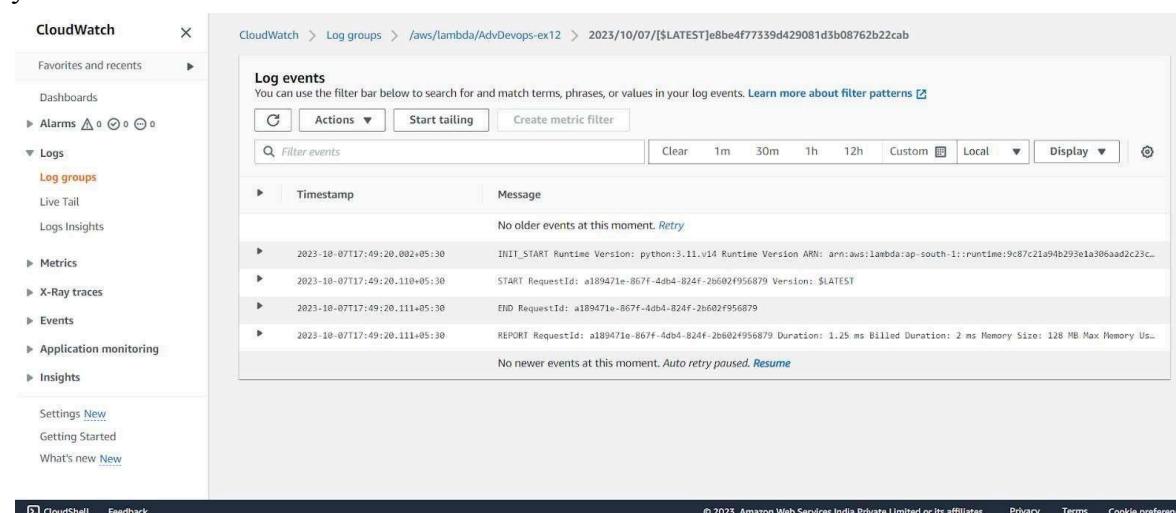
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.



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.

(05)
%

Aryan Patankar

DISA

34

Advance Devops Assignment 1

Q.1 Use S3 bucket and host video streaming

Ans:

1. Create an S3 bucket

i) Login to the AWS console

ii) Navigate to Amazon S3, click Create Bucket and set a unique name

iii) Configure permissions for public access if necessary

2. Upload video file

i) Go to your bucket and upload video files (eg. MP4)

ii) Set public access permissions for the uploaded file

3. Generate Pre-signed URLs

i) If you want to restrict access, generate pre-signed URLs using the AWS CLI or SDK for limited-time access

4. Set up a CloudFront for streaming

• Create a CloudFront distribution with your S3 bucket as the origin

• Use the CloudFront domain to improve video distribution and delivery performance

5. Configure Video Player

• Embed the video using an HTML5 video player or Javascript player:

<video width = "600" controls>

<source src = "https://your-bucket-name.s3.amazonaws.com/your-video.mp4"
type = "video/mp4">

</video>

6. Monitor and Optimize

• Use CloudWatch to monitor S3 and CloudFront performance

• Compress video files or convert them to HLS for better streaming

7. Security

• Use CloudFront signed URLs or signed cookies for access control.

• Enable HTTPS and encryption for secure delivery.

8. Manage Costs

- Track storage and delivery costs in the AWS Billing Dashboard

Q.2) Discuss BMW and Hotstar case studies using AWS

Ans. BMW case study

1. Data collection and management: BMW collects vast amount of data from millions of connected cars, leveraging AWS IoT to manage and analyze the data effectively.
2. Real-Time Data Processing: BMW uses AWS services like Amazon Kinesis to process real-time telemetry data, providing instant insights for services like navigation, traffic alerts and remote diagnostics.
3. Scalability: AWS scalable cloud infrastructure allows BMW to handle the growing data volumes from its global fleet of connected vehicles without compromising performance.
4. Machine Learning and AI: BMW utilizes AWS's machine learning services, including Amazon SageMaker for predictive analytics and personalized driving experience, optimizing vehicle performance based on user behavior.
5. Security Compliance: AWS offers end-to-end encryption, protecting sensitive vehicle and driver data, ensuring compliance with global data protection regulations.
6. Innovation in Autonomous Driving: BMW leverages AWS's high performance computing (HPC) and data analytics capabilities to drive innovations in autonomous driving technology.

• Disney+ Hotstar case study

1. High Scalability During Peak Events: AWS allows Disney+ Hotstar to scale on demand, handling upto 25 million concurrent viewers during live events like the Indian Premier League (IPL).
2. Content Delivery Network (CDN): AWS CloudFront ensures that Hotstar's video content is delivered with low latency and minimal buffering.

enhancing user experience globally.

3. Cost Efficiency: AWS pay-as-you-go pricing model helps Disney+ Hotstar reduce infrastructure costs, scaling resources only when needed during peak viewing times.

4. Global Reach with Edge Locations: With AWS's network of edge locations, Hotstar ensures fast and reliable content delivery to viewers, regardless of their geographic locations.

5. Live Streaming and VOD: AWS enables seamless streaming of both live events and video-on-demand (VOD) content, delivering high-quality videos to millions of users on various devices.

6. Real Time Data Analytics: Hotstar uses AWS analytics such as Amazon Kinesis and Amazon CloudWatch to monitor viewer engagement and platform performance in real-time, enabling quick optimizations.

Q-3

Why Kubernetes and advantages of Kubernetes and its disadvantages. Explain how Adidas uses Kubernetes.

Ans. Kubernetes is an open source platform designed to automate deployment, scaling and management of containerized applications.

Advantages of Kubernetes:

1. Portability: Applications can be moved easily between different environments (development, testing, production) without major changes.

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

3. Reliability: It features self-healing capability, meaning it can restart failed containers and balance workloads to ensure high availability.

4. Self-healing: Kubernetes monitors health of containers and automatically restarts/replaces containers if they fail.

5. Efficiency: It optimizes resource usage by running multiple containers on

on a single host, improving overall efficiency.

Disadvantages of Kubernetes

1. Complexity: It can be calculated and complicated to set up and manage, especially for those new to container technology.
2. Steep Learning Curve: Requires 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 Adidas Uses Kubernetes

Adidas has adopted Kubernetes to enhance its IT infrastructure and improve its ability to respond to market needs. Here's how they benefit from Kubernetes:

1. Faster Application Development: Kubernetes streamlines the deployment process, allowing Adidas to bring new products and features to market quickly.
2. Operational Efficiency: It automates many manual tasks, reducing the amount of time and effort required to manage applications and increasing reliability.

3. Scalability for Demand: During peak sales periods, Kubernetes helps Adidas scale its applications to handle increased customer traffic smoothly.

4. Encouraging Innovation: With a flexible platform, Adidas can experiment with new technologies and business ideas without significant risk.

specific use case at Adidas:

- Microservices Architecture: Adidas breaks down application into smaller, independent services that can be deployed and managed individually.
- Continuous Delivery: Kubernetes supports a continuous delivery pipeline, making it easier to build, test and deploy applications quickly.

Q.4. What are Nagios and explain how Nagios are used in E-service

Ans. Nagios is an open-source monitoring platform designed to oversee system, networks and infrastructure. It helps organization identify and resolve IT infrastructure problems before they impact critical business processes.

Nagios used in E-services:

Publicly available services such as HTTP, FTP, SMTP, etc. These services are network accessible services like web servers, email servers while private services need intermediary agents for monitoring.

Nagios uses plugins to monitor E-services many of which come pre-installed and additional plugins can be found online or developed by users. To monitor a service, a host must first be defined in Nagios configuration files. Once host is defined services like HTTP, FTP or SSH can be monitored by associating them with specific plugins. Nagios provide alert if services fail to respond within defined time frames or if errors are detected.

When issues arise, Nagios integrates with incident management tools to streamline the process of resolving problems. This helps team quickly address and fix issues, minimize downtime.

Nagios can monitor the performance of applications from the user's perspective ensuring that response times are fast and services are running smoothly.

By implementing Nagios, e-services can maintain high availability, enhance reliability and ensure a positive experience for users. This monitoring capability is crucial in today's digital landscape, where any downtime can lead to lost revenue and customer dissatisfaction.

Advance Devops Assignment 2.

34

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 applications 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 CLI globally using node package manager (npm). This allows you to manage serverless applications directly from your terminal.

2) Creating a Node.js serverless project:

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

3) Project Structure:

The project scaffold 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 serverless framework packages your applications, uploads necessary resources to AWS and set up the infrastructure

6) Testing the API: Once deployed you can test REST API using tools like curl or Postman by making post requests to generated API.

7) Storing data in DynamoDB: To store submitted candidate data you integrate AWS DynamoDB as a database

8) ~~AWS~~ Adding more functionalities: Adding functionalities like 'list all candidates, get candidates by ID'

9) AWS IAM Permissions

You need to ensure that serverless framework is given right permissions to interact with AWS resources like dynamoDB

10) Monitoring and maintenance

After deployment serverless framework provides service information like deployed endpoints, API key, log streams.

Q.2 Case study for SonarQube

Creating your own profile in SonarQube for testing project quality. Use SonarQube to analyze your GitHub code. Install SonarLint in your Java IntelliJ IDE and analyze Java code. Analyze Python project with SonarQube.

→ SonarQube is an open source platform used for continuous inspection of code.

quality. It detects bugs, code smells and security vulnerabilities in project across various programming languages.

1) Profile creation in SonarQube:

Quality profiles in SonarQube are essential configurations that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'sonar way' 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 inherit rules from parent profile and reflects future changes automatically. You can activate or deactivate rules, prioritize certain rules and configure parameters to tailor profile to specific projects. Permissions to manage quality profile are restricted to users with administrative privileges. SonarQube allows for the comparison of two profiles to check for differences in activated rules and user can track changes via event log. Quality profiles can also be imported from other instances via backup and restore. To ensure profiles include new rules its important to check against updated built in profiles or use SonarQube rules page.

2) Using SonarCloud to analyze GitHub code:

SonarCloud is cloud-based counterpart of SonarQube that integrates directly with GitHub, BitBucket, Azure and GitLab repositories. To get started with SonarCloud via GitHub signup 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 repo). Next, import repositories into your SonarCloud organization where each GitHub

repo becomes a sonarcloud project. Define 'new code' to focus on recent changes and choose between automatic analysis or CI-based analysis. Automatic analysis happens directly in sonarcloud, while CI based analysis integrates with your build process once the analysis is complete result can be viewed in both sonar-cloud and 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 ensures immediate feedback on 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 tool like coverage.py to generate the coverage port. To enable coverage adjust your build process so that coverage tool runs before sonar scanner and ensure report file is saved in different path.

For setup you can use tox, PyTest and coverage.py to configure and run test. In your tox.ini include configurations for pytest and coverage to generate coverage report in XML format. The build process can also be automated using GitHub Actions, which install dependencies, runs tests and invoke SonarQube scan. Ensure report in Cobertura XML format and place where scanner can access it.

5) Analyzing Node.js projects with SonarQube.

For Node.js project SonarQube can analyze JavaScript and TypeScript code. Similar to the Python setup, you can configure SonarQube to analyze Node.js projects by installing the appropriate plugin and using SonarScanner to scan the projects. SonarQube will check the code against industry standard rules and best practices, flagging issues related to security vulnerabilities, bugs and performance optimization.

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

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

• The Need for self-service infrastructure:

In large organizations, centralized operations teams often face an overwhelming number of repetitive requests. This can lead to delays in service delivery and frustration among product teams who need to move quickly. A self-service model allows teams to provision and manage their infrastructure without relying on the operations team for every request.

- Benefits of Using Terraform
 - 1. Modularity and Reusability:
 - Terraform modules encapsulate standard configurations for various infrastructure components (e.g. networks, databases, compute resources).
 - Teams can reuse these modules across different projects, reducing redundancy and minimizing the risk of errors.
 - 2. Standardization
 - By defining best practices within modules, organizations can ensure that all deployments comply with internal policies and standards.
 - This consistency helps maintain security and operational integrity across the organization.
 - 3. Increased Efficiency
 - Product teams can deploy services quickly by using pre-defined modules, significantly reducing the time spent on infrastructure setup.
 - This allows team to focus on developing features rather than managing infrastructure.
 - 4. Integration with Ticketing Systems
 - Terraform Cloud can integrate with ticketing systems like ServiceNow to automate the generation of infrastructure requests.
 - This integration streamlines workflows by allowing teams to initiate requests directly from their ticketing platform, reducing manual intervention.
- Implementation steps
1. Identify Infrastructure Components
 - Begin by identifying which components of your infrastructure can be modularized (e.g. VPCs, security groups, load balancers)
 2. Develop Terraform modules.
 - Create reusable modules that define the desired configurations and

Resources:

- Ensure each module includes input variables for customization and outputs for integration with other modules.
- 3. Establish Governance and Best Practices:
 - Define guidelines for module usage, versioning, and documentation to ensure clarity and maintainability.
 - Encourage teams to contribute to module development and share improvements.
- 4. Testing and validation
 - Implement a testing framework to validate module functionality before deployment.
 - Use tools like terraform plan to preview changes and catch potential issues early.
- Best practices for module management
 - Utilize the terraform registry:
 - Leverage existing community modules from the Terraform Registry to avoid reinventing solutions and ensure adherence to best practices.
 - Version control: Implement versioning for your module to track changes over time. This helps manage dependencies effectively and minimize disruptions during updates.
 - Documentation: Maintain comprehensive documentation for each module, including usage examples, input/output descriptions and any dependencies.
 - Encourage collaboration: Foster a culture of collaboration by sharing modules across teams. This promotes consistency in deployments and facilitates knowledge within the organization.

By adopting a self-service infrastructure model with Terraform, organizations can empower product teams to efficiently manage their own infrastructure while ensuring compliance with established standards.

This approach not only streamlines processes but also enhances agility in responding to changing business needs. Ultimately, it leads to a more responsive IT environment that supports innovation ^{and} growth within the organization.